

DOI:10.11918/202504061

LPOA-MPC 的车辆横纵向轨迹跟踪协同优化控制方法

李晓龙^{1,2}, 黄鹤^{1,2}, 杨澜³, 王会峰¹, 高涛³

(1. 长安大学 电子与控制工程学院, 西安 710064; 2. 西安市智慧高速公路信息融合与控制重点实验室(长安大学), 西安 710064; 3. 长安大学 信息工程学院, 西安 710064)

摘要: 针对车辆模型预测控制(model predictive control, MPC)轨迹跟踪控制器中权重矩阵参数选取困难, 导致车辆轨迹跟踪控制的稳定性与精度不足的问题, 本文提出了一种融合多种机制的拉丁-鹈鹕算法, 用于优化控制车辆横、纵向联合模型预测轨迹跟踪控制器权重矩阵参数。首先, 基于车辆单轨模型, 分别设计了车辆横向 MPC 控制器和纵向的 MPC 上位控制器、基于加速度-驱动力逆动力学模型的下位控制器; 其次, 针对鹈鹕算法(Pelican optimization algorithm, POA)在解空间内搜索效率低的问题, 提出了一种拉丁-鹈鹕算法(Latin Pelican optimization algorithm, LPOA), 引入灰狼算法的等级狩猎机制重构 POA 的猎物定位模型, 通过 α 鹈鹕引导策略提高算法收敛速度; 同时, 融入动态随机搜索策略, 利用其重尾分布特性增强算法在迭代后期的局部极值逃逸能力; 最后, 基于 LPOA 的寻优能力, 分别优化横、纵向 MPC 控制器权重矩阵参数, 并通过 Carsim 和 Simulink 平台联合仿真验证所提出的横向、纵向以及横纵联合优化控制方法。研究表明, 本文提出的 LPOA-MPC 控制器在车辆横向、纵向以及横纵联合控制中, 均能够有效提高车辆轨迹跟踪控制稳定性与控制精度。

关键词: 横纵联合; 轨迹跟踪; 模型预测控制; 鹈鹕算法; 参数整定

中图分类号: TP301.6

文献标志码: A

文章编号: 0367-6234(2026)05-0090-13

Cooperative optimization of vehicle lateral and longitudinal trajectory tracking based on LPOA-MPC

LI Xiaolong^{1,2}, HUANG He^{1,2}, YANG Lan³, WANG Huifeng¹, GAO Tao³

(1. School of Electronic and Control Engineering, Chang'an University, Xi'an 710064, China; 2. Key Laboratory of Intelligent Expressway Information Fusion and Control (Chang'an University), Xi'an 710064, China; 3. School of Information Engineering, Chang'an University, Xi'an 710064, China)

Abstract: Aiming at the problem that it is difficult to select the weight matrix parameters in the vehicle model predictive control (MPC) trajectory tracking controller, which makes the stability and accuracy of the vehicle trajectory tracking control insufficient, this research propose a latin-pelican algorithm (LPOA) that integrates multiple mechanisms to optimize the weight matrix parameters of the vehicle lateral and longitudinal joint model predictive trajectory tracking controller. Firstly, the vehicle transverse MPC controller, the longitudinal MPC upper controller and the lower controller based on the acceleration-drive inverse dynamics model are designed respectively based on the vehicle single-track model; Secondly, a Latin Pelican Optimization Algorithm is proposed to improve the efficiency of the pelican algorithm's searching in the solution space. The hierarchical hunting mechanism of the gray wolf algorithm is introduced to reconfigure the prey localization model of the POA, and the convergence speed of the algorithm is improved by the α -pelican guidance strategy. Thus, a dynamic stochastic search strategy is incorporated to enhance the algorithm's ability to escape from local extremes in the late iteration by using its heavy-tailed distribution characteristics. Finally, the parameters of the horizontal and vertical MPC controller weight matrices are optimized using the optimization capability of LPOA; and the proposed horizontal, vertical, and horizontal-longitudinal joint optimization control methods are verified through co-simulation on CarSim and Simulink platforms. Results show that the LPOA-MPC controller proposed in this research can effectively improve the stability and accuracy of vehicle trajectory tracking control in horizontal, longitudinal and transverse-longitudinal joint control.

Keywords: transverse-longitudinal association; trajectory tracking; model predictive control; pelican algorithm; parameter tuning

收稿日期: 2025-04-22; 录用日期: 2025-08-02; 网络首发日期: 2025-10-24

网络首发地址: <https://link.cnki.net/urlid/23.1235.T.20251024.1407.004>

基金项目: 国家自然科学基金面上项目(52572353); 中央高校基本科研业务费资助项目(300102326501); 陕西省留学人员科技活动择优资助项目(2023001)

作者简介: 李晓龙(2002—), 男, 硕士研究生; 黄鹤(1979—), 男, 教授, 博士生导师

通信作者: 黄鹤, huanghe@chd.edu.cn

轨迹跟踪控制是车辆运动控制的核心技术之一,其目标是车辆基于当前的定位和状态信息,结合规划与决策系统,在保证车辆的行驶稳定性以及乘客乘坐体验的前提下,使其能够精准和实时地按照参考轨迹行驶。车辆的轨迹跟踪控制根据控制目标的不同,又可以分为横向控制、纵向控制以及横纵联合控制。车辆的纵向控制通过控制车辆的制动系统和发动机转速实现对车辆速度和前后车或障碍物距离的控制;横向控制通过控制车辆的前轮转角实现车辆保持期望的行车路线;横纵联合控制则综合考虑横向和纵向控制以实现全面控制。

目前国内外学者广泛使用的车辆轨迹跟踪控制方法,主要包括线性二次型最优控制^[1]、自抗扰控制^[2]、模型预测控制算法(model predictive control, MPC)^[3]等方法。模型预测控制因其基于系统动力学模型的优化特性,被广泛应用于非线性、多约束的控制任务。然而,MPC 控制器的性能在很大程度上依赖于权重矩阵参数的合理选择,权重矩阵在 MPC 的优化过程中,起着平衡不同性能指标的作用。因此,合理设置参数对于确保控制器在各种复杂工况下的稳定性和跟踪精度至关重要。但是,在当前 MPC 设计中,调节权重矩阵的方法主要有两种:经验法和多次仿真调试的试错法。经验法通常依赖于设计者的直觉和先验经验,适用于特定领域或已知的标准任务,但过于依赖工程师对系统的了解深度,具有较强的主观性,缺乏系统性和普适性。而试错法虽然能够通过反复仿真验证来确定较为合适的权重矩阵参数,但该方法需要耗费大量时间和精力进行调试,周期较长,效率低下,且获得的参数通常仅适用于特定工况,适应范围相对有限,在面对各种复杂多变的实际应用场景时,难以实现有效的泛化应用。因此,如何根据动态环境条件,自动、精准地调整 MPC 控制器中的权重矩阵参数,成为亟待解决的重要问题。

近年来,群体智能算法通过模拟自然界中生物群体的协同行为,展现出强大的全局搜索和优化能力,为传统控制策略中参数选择困难的问题开辟了新的解决思路和方法^[4]。Zhou 等^[5]提出了一种基于自适应鲸鱼优化算法的模糊 PID 轨迹跟踪控制器,通过引入自适应鲸鱼优化算法来优化模糊 PID 控制器参数,从而显著提高了控制器的控制精度。Li 等^[6]提出了一种基于蝴蝶优化算法的自动驾驶车辆的轨迹跟踪控制方法,并在不同速度条件下进行了试验验证,结果表明,该方法能够显著提高自动驾驶车辆的控制性能及其可靠性。李晓龙等^[7]提出了一种多策略优化算法,用于对横向模型预测轨

迹跟踪控制器的权重矩阵进行优化,显著改善了控制系统在横向轨迹跟踪过程中的性能,有效降低了车辆的最大横向误差,但其仅研究了横向轨迹跟踪控制,忽略了实际驾驶过程中纵向控制对轨迹跟踪精度的影响,无法完全满足复杂驾驶环境的要求。

鹈鹕优化算法(pelican optimization algorithm, POA)是 Trojovsk 等^[8]模仿鹈鹕在自然界狩猎过程中的狩猎策略和行为提出的一种新型群体智能优化方法,其将种群中鹈鹕的位置视为候选解,将目标函数值视为食物的质量,旨在通过模拟鹈鹕的觅食行为来找到最优解。POA 相较于其他智能算法在探索方面非常有效,但在复杂多模态优化问题中存在收敛速度衰减和局部极值逃逸能力弱的问题^[9]。为了扩大了 POA 算法的适用场景,本文提出了一种融合多种机制的拉丁-鹈鹕算法(latin pelican optimization algorithm, LPOA),并将 LPOA 与分层协同优化理论相结合,建立横向与纵向模型预测控制(MPC)的权重矩阵参数协同优化车辆轨迹跟踪控制模型,以提高轨迹跟踪控制器的控制性能。

1 车辆模型的建立

1.1 车辆动力学模型

本文建立的车辆动力学模型如图 1 所示。

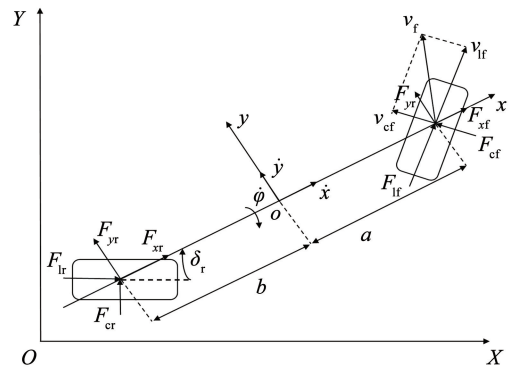


图 1 车辆单轨模型

Fig. 1 Vehicle monorail model

其中, a, b 为前、后轴到质心的距离, I_z 为绕 z 轴转动惯量, m 为车身质量。 F_{lf}, F_{lr} 为前后轮纵向力, F_{cf}, F_{cr} 为前后轮的横向力。 F_{xf}, F_{xr} 为前后轮 x 方向的力, F_{yf}, F_{yr} 为前后轮 y 方向的力。由牛顿第二定律,可得如下平衡方程:

$$x \text{ 轴: } m\ddot{x} = m\dot{y}\dot{\varphi} + 2F_{xf} + 2F_{xr} \quad (1)$$

$$y \text{ 轴: } m\ddot{y} = -m\dot{x}\dot{\varphi} + 2F_{yf} + 2F_{yr} \quad (2)$$

$$\text{绕 } z \text{ 轴: } I_z\ddot{\varphi} = 2aF_{yf} - 2bF_{yr} \quad (3)$$

为了进一步简化分析,采用小角度假设,即假设车辆轮胎侧偏角较小,轮胎侧向力和侧偏刚度可以近似用线性函数描述,可得车辆模型如下:

$$\begin{cases} m\ddot{y} = -m\dot{x}\dot{\varphi} + 2\left[C_{cf}\left(\delta_f - \frac{\dot{y} + a\dot{\varphi}}{\dot{x}}\right) - C_{cr}\frac{\dot{y} - b\dot{\varphi}}{\dot{x}}\right] \\ m\ddot{x} = m\dot{y}\dot{\varphi} + 2\left[C_{lr}s_f + C_{cf}\left(\delta_f - \frac{\dot{y} + a\dot{\varphi}}{\dot{x}}\right)\delta_f + C_{lr}s_r\right] \\ I_z\ddot{\varphi} = 2\left[aC_{cf}\left(\delta_f - \frac{\dot{y} + a\dot{\varphi}}{\dot{x}}\right) + bC_{cr}\frac{\dot{y} - b\dot{\varphi}}{\dot{x}}\right] \\ \dot{Y} = \dot{x}\sin\varphi + \dot{y}\cos\varphi \\ \dot{X} = \dot{x}\cos\varphi - \dot{y}\sin\varphi \end{cases} \quad (4)$$

式中: δ_f 为前轮转角; s_f, s_r 分别为前、后轮滑移率; C_{lf}, C_{lr} 分别为前、后轮胎纵向刚度; C_{cf}, C_{cr} 分别为前、后轮胎侧偏刚度。

1.2 状态量偏差的建立

为了使车辆能够精准地跟踪给定的参考轨迹, 建立车辆与参考轨迹之间的动态误差反馈模型, 能够帮助分析车辆与目标轨迹之间的偏差, 并为进一步优化控制性能提供重要依据。

由图 2 所示的误差模型, 可得如下车辆横向和航向跟踪误差方程:

$$e_\varphi = \varphi - \varphi_r \quad (5)$$

$$v_s = \frac{1}{1 - r_o e_y} [v_x \cos(e_\varphi) + v_y \sin(e_\varphi)] \quad (6)$$

$$\dot{e}_y = v_x \sin(e_\varphi) + v_y \cos(e_\varphi) \quad (7)$$

$$\dot{e}_\varphi = \dot{\varphi} - r_o v_s \quad (8)$$

式中: v_x, v_y 为车辆速度的纵向和横向速度分量, v_s 为车辆沿轨迹的速度, φ_r 为道路切向与大地坐标系 X 轴的夹角, r_o 为道路曲率。为了简化计算, 假设航向误差 e_φ 的值很小, 则可近似认为 $\sin(e_\varphi) \approx e_\varphi$, $\cos(e_\varphi) \approx 1$, 则式(7)和式(8)简化为:

$$\dot{e}_y = v_x e_\varphi + v_y \quad (9)$$

$$\dot{e}_\varphi = \dot{\varphi} - r_o v_x \quad (10)$$

纵向速度的跟踪误差为^[10]:

$$e_v = v_r - v_s \quad (11)$$

$$\dot{e}_v = \dot{v}_r - \dot{v}_s \quad (12)$$

$$e_s = \int_{t_1}^{t_2} e_v dt \quad (13)$$

式中: v_r 为纵向参考速度, e_v 为纵向速度误差, \dot{e}_v 为纵向加速度误差, e_s 为纵向位移误差。

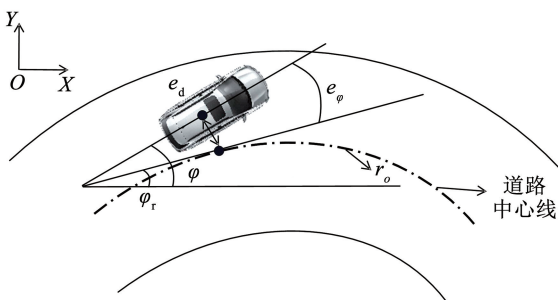


图 2 轨迹跟踪误差模型

Fig. 2 Trajectory tracking error model

2 横纵联合控制器设计

2.1 横向控制器设计

式(4)车辆模型中状态方程为 $\dot{\xi} = f(\xi, u)$, 控制变量为 $u = \delta_f$, 状态变量为 $\xi = [\dot{y}, \dot{x}, \varphi, \dot{\varphi}, Y, X]^T$, 对非线性车辆动力学模型进行线性化处理, 可得到对应的线性事变状态空间方程如下:

$$\dot{\xi} = \mathbf{A}(t)\xi(t) + \mathbf{B}(t)u(t) \quad (14)$$

式中: $\mathbf{A}(t) = \frac{\partial f}{\partial \xi}|_{\xi_t, u_t}$ 为 f 对 ξ 的雅可比矩阵, $\xi(t)$ 为系统 t 时刻状态量, $u(t)$ 系统 t 时刻输入量, $\mathbf{B}(t) = \frac{\partial f}{\partial u}|_{\xi_t, u_t}$ 为 f 对 u 的雅可比矩阵。

经过离散化处理, 得到状态空间表达式为

$$\xi(k+1) = \mathbf{A}(k)\xi(k) + \mathbf{B}(k)u(k) \quad (15)$$

式中: $\mathbf{A}(k) = \mathbf{I} + T_s \mathbf{A}(t)$, $\mathbf{B}(k) = T_s \mathbf{B}(t)$, 其中, \mathbf{I} 为单位矩阵, T_s 为采样周期。

将系统的输出量设置为 $\eta = [\varphi \ Y]^T$, 进行离散化处理, 得到输出状态空间表达式为

$$\eta(k) = \mathbf{C}_k \xi(k) \quad (16)$$

式中, $\mathbf{C}_k = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T$ 。

令 $\tilde{\xi}(k+1) = [\xi(k) \ u(k-1)]^T$, 可以得到如下状态空间表达式:

$$\tilde{\xi}(k+1) = \tilde{\mathbf{A}}_k \tilde{\xi}(k) + \tilde{\mathbf{B}}_k \Delta u(k) \quad (17)$$

$$\tilde{\eta}(k) = \tilde{\mathbf{C}}_k \tilde{\xi}(k) \quad (18)$$

式中: $\tilde{\mathbf{A}}_k = \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{0}_{1 \times 6} & \mathbf{I} \end{bmatrix}$, $\tilde{\mathbf{B}}_k = \begin{bmatrix} \mathbf{B}_k \\ \mathbf{I} \end{bmatrix}$, $\tilde{\mathbf{C}}_k = [\mathbf{C}_k \ \mathbf{0}_{2 \times 1}]$ 。

系统的输出量表达式为

$$\mathbf{Y}(k) = \boldsymbol{\psi}_k \tilde{\xi}(k) + \boldsymbol{\Theta}_k \Delta \mathbf{U}(k) \quad (19)$$

式中各矩阵如下:

$$\mathbf{Y}(k) = \begin{bmatrix} \tilde{\eta}(k+1) \\ \tilde{\eta}(k+2) \\ \vdots \\ \tilde{\eta}(k+N_c) \\ \vdots \\ \tilde{\eta}(k+N_p) \end{bmatrix}, \quad \boldsymbol{\psi}_k = \begin{bmatrix} \tilde{\mathbf{C}}_k \tilde{\mathbf{A}}_k \\ \tilde{\mathbf{C}}_k \tilde{\mathbf{A}}_k^2 \\ \vdots \\ \tilde{\mathbf{C}}_k \tilde{\mathbf{A}}_k^{N_c} \\ \vdots \\ \tilde{\mathbf{C}}_k \tilde{\mathbf{A}}_k^{N_p} \end{bmatrix},$$

$$\Delta \mathbf{U}(k) = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_c) \end{bmatrix},$$

$$\Theta_k = \begin{bmatrix} \tilde{C}_k \tilde{B}_k & 0 & 0 & 0 \\ \tilde{C}_k \tilde{A}_k \tilde{B}_k & \tilde{C}_k \tilde{B}_k & 0 & 0 \\ \dots & \dots & \ddots & \dots \\ \tilde{C}_k \tilde{A}_k^{N_c-1} \tilde{B}_k & \tilde{C}_k \tilde{A}_k^{N_c-2} \tilde{B}_k & \dots & \tilde{C}_k \tilde{B} \\ \tilde{C}_k \tilde{A}_k^{N_c} \tilde{B}_k & \tilde{C}_k \tilde{A}_k^{N_c-1} \tilde{B}_k & \dots & \tilde{C}_k \tilde{A}_k \tilde{B}_k \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{C}_k \tilde{A}_k^{N_p-1} \tilde{B}_k & \tilde{C}_k \tilde{A}_k^{N_p-2} \tilde{B}_k & \dots & \tilde{C}_k \tilde{A}_k^{N_p-N_c-1} \tilde{B}_k \end{bmatrix}$$

根据上述系统模型,可得目标函数如下^[11]:

$$J(\tilde{\xi}(t), \mathbf{u}(t-1), \Delta \mathbf{U}(t)) = \sum_{i=1}^{N_p} \|\tilde{\boldsymbol{\eta}}(t+i|t) - \tilde{\boldsymbol{\eta}}_{\text{ref}}(t+i|t)\|_Q^2 + \sum_{i=1}^{N_c-1} \|\Delta \mathbf{u}(t+i|t)\|_R^2 + \rho \varepsilon^2 \quad (20)$$

式中: $\boldsymbol{\eta}(t+i|t)$ 为 t 时刻预测的 $t+i$ 时刻系统输出量, ρ 为权重系数, ε 为松弛因子, $\boldsymbol{\eta}_{\text{ref}}$ 为系统状态参考值, Q, R 为权重矩阵。

为提高式(20)目标函数的求解速率,将其转化为二次规划形式,从而能够利用数值优化算法快速求解。

2.2 纵向控制器设计

在轨迹跟踪的纵向速度控制架构中,分为上位和下位控制器,目的是将高层的决策优化与底层的执行控制分离,从而简化系统设计并提高控制性能。上位控制器的任务是基于车辆动力学模型和轨迹跟踪目标,生成最优的纵向加速度指令 a_{des} 。下位控制器将上位控制器生成的期望加速度 a_{des} 转化为具体的执行器指令(如节气门开度或制动主缸压力)。上位控制器与下位控制器的分层式设计实现了纵向动力学系统的解耦控制。纵向MPC控制框图如图3所示。

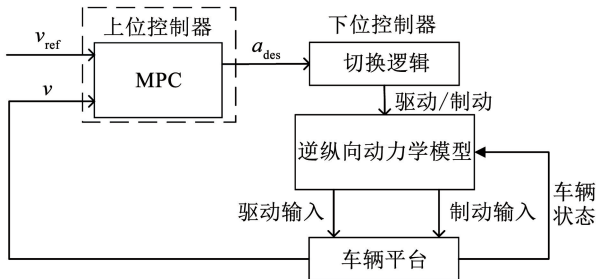


图3 纵向MPC控制框图

Fig.3 Longitudinal MPC control block diagram

2.2.1 基于MPC的上位控制器设计

在车辆的纵向控制系统中,可以认为横摆运动较小,则纵向控制可以被简化为积分和一阶惯性系统,具体表示为:

$$\begin{cases} \dot{v} = a \\ \dot{a} = \frac{K}{\tau}(a_{\text{des}} - a) \end{cases} \quad (21)$$

式中: K 为系统增益,本文取值为1, τ 为时间常数,本文取值为0.5。

根据上述运动特性,则可将车辆纵向运动的连续系统状态空间方程表示为

$$\dot{\mathbf{x}} = \mathbf{A}_1 \mathbf{x} + \mathbf{B}_1 \mathbf{u}_1 \quad (22)$$

式中:状态向量 $\mathbf{x} = [v \ a]^T$, $\mathbf{u}_1 = a_{\text{des}}$ 为系统输入, $\mathbf{A}_1, \mathbf{B}_1$ 矩阵如下:

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 \\ 0 & -1/\tau \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} 0 \\ K/\tau \end{bmatrix}$$

经离散处理,得到离散系统状态方程

$$\mathbf{x}(k+1) = \mathbf{A}_k \mathbf{x}(k) + \mathbf{B}_k \mathbf{u}(k) \quad (23)$$

式中 k 为当前采样时刻。其中, $\mathbf{A}_k, \mathbf{B}_k$ 矩阵如下:

$$\mathbf{A}_k = \begin{bmatrix} 0 & 1 \\ 0 & 1 - T_s/\tau \end{bmatrix}, \mathbf{B}_k = \begin{bmatrix} 0 \\ KT_s/\tau \end{bmatrix}$$

式中 T_s 为采样周期。将车辆速度 v 作为系统输出,可表示为

$$\mathbf{y}(k) = \mathbf{C}_1 \mathbf{x}(k) \quad (24)$$

式中 $\mathbf{C}_1 = [1 \ 0]$ 。

系统的控制目标是速度跟踪精度,定义以下目标函数:

$$J(\mathbf{x}(t), \mathbf{u}(t-1), \Delta \mathbf{u}(k)) = \sum_{i=1}^{N_p} \|\mathbf{y}_p(k+i|k) - \mathbf{y}_{\text{ref}}(k+i|k)\|_Q^2 + \sum_{i=1}^{N_c} \|\Delta \mathbf{u}(k+i)\|_R^2 \quad (25)$$

式中: $t-1$ 为上一采样时刻, $\mathbf{y}_p(k+i|k)$ 为控制输出预测值, $\mathbf{y}_{\text{ref}}(k+i|k)$ 为控制输出参考值, $(k+i|k)$ 为根据 k 时刻的采样信息预测得到的 $k+i$ 时刻状态, Q, R 为权重矩阵。

2.2.2 基于加速度-驱动力逆动力学模型的下位控制器

获得最优加速度指令后,下位控制器需完成动力域到执行机构的非线性映射。本文采用基于加速度-驱动力逆动力学模型建立下位控制器,主要包括驱动、制动的控制切换逻辑设计以及执行机构映射。为了实现车辆行驶过程中加减速的平滑切换,驱动和制动的控制切换逻辑需满足以下关键要素:1)驱动和制动不能同时进行;2)驱动和制动之间不能频繁切换。定义临界加速度 a_0 为

$$a_0 = \frac{F_{\alpha 0} - F}{m} \quad (26)$$

同时,为了适应动态工况,引入了车速和路面附着系数修正因子,其计算公式为

$$a_0 = a(1 - k_v v - k_\mu \mu) \quad (27)$$

式中： F_{a0} 为节气门开度为 0 时的驱动力； F 为车辆行驶过程中总阻力，包含车辆的滚动阻力及空气阻力； k_v 为车速修正因子，用于抑制高速工况下空气阻力的非线性影响； k_μ 为路面附着系数修正因子，在低路面附着系数时增大，以避免打滑。在上述基础上引入缓冲加速度 $\tau = 0.05 \text{ m/s}^2$ 来避免驱动和制动的频繁切换，具体如下：

$$\begin{cases} a_{\text{des}} - a_0 > \tau & \text{驱动控制} \\ a_{\text{des}} - a_0 \leq \tau & \text{制动控制} \end{cases} \quad (28)$$

3 鹈鹕优化策略

3.1 基本鹈鹕算法

3.1.1 初始化

根据具体问题的下限和上限随机初始化鹈鹕种群个体位置，每个鹈鹕个体都表示为一个候选解，用矩阵表示为：

$$X = \begin{bmatrix} \mathbf{x}_{1,1} & \cdots & \mathbf{x}_{1,j} & \cdots & \mathbf{x}_{1,D} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{x}_{i,1} & \cdots & \mathbf{x}_{i,j} & \cdots & \mathbf{x}_{i,D} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{x}_{N,1} & \cdots & \mathbf{x}_{N,j} & \cdots & \mathbf{x}_{N,D} \end{bmatrix} \quad (29)$$

$$\begin{aligned} \mathbf{x}_{ij} &= \text{rand} \times (\mathbf{U}_j - \mathbf{L}_j) + \mathbf{L}_j, \\ i &= 1, 2, \dots, N, j = 1, 2, \dots, D \end{aligned} \quad (30)$$

式中： \mathbf{x}_{ij} 为第 i 只鹈鹕个体在第 j 维的位置， \mathbf{U}_j 和 \mathbf{L}_j 分别为第 j 维的上、下界， N 为种群规模， D 为问题维数。适应度矩阵为

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix} = \begin{bmatrix} f([\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \dots, \mathbf{x}_{1,D}]) \\ \vdots \\ f([\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \dots, \mathbf{x}_{i,D}]) \\ \vdots \\ f([\mathbf{x}_{N,1}, \mathbf{x}_{N,2}, \dots, \mathbf{x}_{N,D}]) \end{bmatrix} \quad (31)$$

3.1.2 猎物更新

为了提高 POA 的全局搜索能力，通过随机选择机制，在种群中随机选择一个鹈鹕个体作为猎物，对其进行捕猎。其计算公式为

$$X_F = X(K, :) \quad (32)$$

式中： X_F 为猎物位置， K 为 $(0, N)$ 中的随机整数值。

3.1.3 逼近猎物阶段(探索阶段)

在逼近猎物阶段，鹈鹕向着猎物所在水面移动，数学模型如下：

$$\mathbf{x}_{ij}^{p1} = \begin{cases} \mathbf{x}_{ij} + \sigma \cdot (\mathbf{P}_j - I \cdot \mathbf{x}_{ij}), & F_p < F_i \\ \mathbf{x}_{ij} + \sigma \cdot (\mathbf{x}_{ij} - \mathbf{P}_j), & \text{else} \end{cases} \quad (33)$$

$$\mathbf{x}_i = \begin{cases} \mathbf{x}_i^p, & F_i^p < F_i \\ \mathbf{x}_i, & \text{else} \end{cases} \quad (34)$$

式中： \mathbf{x}_{ij}^{p1} 为更新后的位置， σ 为 $(0, 1)$ 内随机数， I 为 $[1, 2]$ 内随机整数值， F_i^p 为 \mathbf{x}_{ij}^{p1} 的适应度值， \mathbf{P}_j 为猎物在第 j 维的位置， F_p 为猎物适应度值。

3.1.4 掠过水面阶段(勘探阶段)

鹈鹕飞行至猎物所在水面上方，随后展开翅膀掠过水面，将猎物捕获于喉袋之中。通过这种飞行方式对猎物进行精准定位和捕捉，其数学模型如下：

$$\mathbf{x}_{ij}^{p2} = \mathbf{x}_{ij} + R \cdot (1 - t/T) \cdot (2 \cdot \beta - 1) \cdot \mathbf{x}_{ij} \quad (35)$$

$$\mathbf{x}_i = \begin{cases} \mathbf{x}_i^p, & F_i^p < F_i \\ \mathbf{x}_i, & \text{else} \end{cases} \quad (36)$$

式中： \mathbf{x}_{ij}^{p2} 为更新后的位置， β 为 $[0, 1]$ 内随机数， R 为 $[1, 2]$ 内随机整数值， t 为当前迭代次数， T 为最大迭代次数， F_i^p 为 \mathbf{x}_{ij}^{p2} 的适应度值。

式(33)和式(36)中采用了贪婪法，即仅当更新后的位置具有更优适应度时才接受位置更新，否则保留原位置不变。该策略有助于在迭代过程中持续保持当前较优解，提升算法的局部搜索能力和收敛速度。

3.2 拉丁-鹈鹕算法(LPOA)

为了使 POA 在复杂问题求解中获得更优的寻优效果和更快的收敛速度，本文提出以下改进策略。

3.2.1 基于正交-拉丁超立方采样的种群初始化

在智能算法的种群初始化阶段，传统初始化分布是随机数生成的序列，遵循均匀概率分布。但生成的序列存在较大偏差，导致在搜索空间内分布不均，产生局部聚集，无法有效覆盖搜索空间^[12]。拉丁超立方采样(Latin hypercube sampling, LHS)的主要思想是通过分层将解空间划分为相等的子空间，并使用概率密度函数在每个分层中选择一个随机样本。将这个概念推广到任意维度，可以确保每个样本在包含其轴向超平面中是唯一的^[13]。然而，基于LHS的初始化方法受到样本数量和解空间范围的限制。当种群中个体数量有限或解空间范围较大时，其性能受到较大影响，此时基于随机数的初始化方法更为有利^[14]。因此，在LHS的基础上引入一定程度的随机性来避免初始化和迭代过程容易出现局部最优问题。为此，本文提出一种结合正弦函数与LHS的种群初始化方法，具体计算公式为

$$X_i = \sin\left(\pi \times \left(\frac{1}{2} - \mathbf{ls}_i\right) \times p\right) \times (\mathbf{U}_j - \mathbf{L}_j) + \mathbf{L}_j \quad (37)$$

式中： X_i 为种群初始化后的第 i 个个体， \mathbf{ls}_i 为对应的LHS序列， p 为随机数，取值范围为 $[0, 1]$ ，用来控制初始化的随机性。

3.2.2 融合灰狼算法的猎物位置更新

鹈鹕算法将搜索空间内一个随机个体作为猎物对其进行捕食，随后根据该位置来更新其他个体位

置。虽然这种方法扩大了搜索空间并保证了算法的全局搜索能力,但也会导致收敛速度变慢。这是因为随机选择的猎物位置可能与全局最优解相差甚远,从而使算法在搜索过程中需要耗费更多迭代次数和计算资源,才能逐步逼近最优解。因此,为了在保证算法全局搜索能力的基础上提高其收敛速度,本文设计了一种融合灰狼算法的猎物位置更新策略。具体而言,选取适应度值最优的3个鹧鹑个体,从高到低依次为 α 、 β 、 δ 鹧鹑。这3个鹧鹑个体在种群中扮演着领导者角色,其位置信息代表当前搜索过程中的优质解。此外,为了在引入优质个体信息的同时,保留一定的随机性以维持算法的探索能力,另外选取一个随机鹧鹑个体作为 ω 鹧鹑。计算 ω 鹧鹑与 α 、 β 、 δ 的3个鹧鹑的相对距离的平均值,并将该平均距离值作为新的猎物位置。这一策略不仅充分利用了种群中优秀个体的信息,引导算法向更优区域搜索,还通过随机个体的引入避免了算法过早陷入局部最优,从而在全局搜索和局部开发之间取得了良好的平衡。具体数学模型如下^[15]:

$$\begin{cases} D_\alpha = |C_1 \cdot X_\alpha - X| \\ D_\beta = |C_2 \cdot X_\beta - X| \\ D_\delta = |C_3 \cdot X_\delta - X| \end{cases} \quad (38)$$

$$\begin{cases} X_1 = X_\alpha - A_1 \cdot D_\alpha \\ X_2 = X_\beta - A_2 \cdot D_\beta \\ X_3 = X_\delta - A_3 \cdot D_\delta \end{cases} \quad (39)$$

$$X_F = \frac{X_1 + X_2 + X_3}{3} \quad (40)$$

式中: D_α 、 D_β 和 D_δ 为 ω 鹧鹑到 α 、 β 和 δ 鹧鹑间的相对距离; X_1 、 X_2 和 X_3 分别对应 α 、 β 和 δ 鹧鹑位置; X 为 ω 鹧鹑的位置; C_1 、 C_2 、 C_3 为随机扰动系数向量; A_1 、 A_2 、 A_3 为系数向量; X_F 为更新后的猎物位置。

3.2.3 动态随机搜索策略

动态随机搜索技术^[16]是一种新颖的随机搜索方法,在自适应随机搜索技术的基础上赋予其局部搜索能力,具体分为全局搜索阶段和局部搜索阶段,将全局搜索过程中的最优值作为局部搜索的初始值^[17]。该技术可以增强 POA 算法的局部搜索能力,目的是通过较小的步长搜寻在当前最优值周围解空间中是否存在适应度更优的点。具体伪代码如下。

Algorithm: Dynamic random search technique

- 1: Initialize number of maximum iteration (N), the step size α_0 , maximum epochs E , current solution $x_c = x_\alpha$, epoch = 0, $j = 0$, function $F(x)$, best value F_{best} , current value F_c .
- 2: Reset internal counter $m = 0$
- 3: Produce a random vector x_v in $[-\alpha_j, \alpha_j]$

- 4: epoch = epoch + 1
- 5: new value $F_{new} = F(x_c + x_v)$
- 6: if $F_{new} < F_{best}$ then
- 7: $x_\alpha = x_c + x_v$, $F_{best} = F_{new}$, $m = m + 1$. Go to step 19.
- 8: end if
- 9: if $F_{new} < F_c$ then
- 10: $x_c = x_c + x_v$, $F_c = F_{new}$, $m = m + 1$. Go to step 19.
- 11: end if
- 12: $F_{new} = F(x_c - x_v)$
- 13: if $F_{new} < F_{best}$ then
- 14: $x_\alpha = x_c + x_v$, $F_{best} = F_{new}$, $m = m + 1$. Go to step 19.
- 15: end if
- 16: if $F_{new} < F_c$ then
- 17: $x_c = x_c + x_v$, $F_c = F_{new}$, $m = m + 1$. Go to step 19.
- 18: end if
- 19: If $m < N$, then go to step 3.
- 20: $j = j + 1$, $\alpha_j = \alpha_{j-1} \times 0.5$
- 21: If epoch = E , then quit. Otherwise, go to step 2.

3.3 优化策略的测试

为了系统且全面地评估提出的改进策略在多模态优化问题中的寻优性能优势,将 LPOA 与现有的 POA、MFO、GWO 和 BES 算法在具有典型特征的基准测试函数上进行综合对比分析,测试函数涵盖了单峰函数、多峰函数和复合测试函数。其中,单峰函数用于评估算法在简单、平滑搜索空间中的收敛速度和精度;多峰函数用于测试算法在复杂、多极值点环境中的全局搜索能力和跳出局部最优的能力;复合测试函数则综合了单峰和多峰函数的特点,用于考察算法在更接近实际复杂问题中的综合性能。测试结果如图4所示。

在对基准测试函数的寻优过程中,图4中的收敛曲线清晰地表明,LPOA 相较于其他对比算法全局寻优能力和收敛速度方面具有明显优势。在针对单峰函数(图4(a))的测试中,MFO 过早地陷入了局部陷阱,相比之下,LPOA 展现了更为迅速且稳定的收敛特性,能够在较短的时间内准确地找到全局最优解。在多峰函数(图4(b))的测试中,测试函数的复杂性显著提高,存在多个局部最优解,BES 和 GWO 在该多峰问题的求解过程中,由于局部搜索的局限性,容易停留在某一局部最优解,未能充分探索整个解空间。与之相反,LPOA 凭借其改进策略,在多模态优化问题中展现出强大的全局探索能力,能够有效避免陷入局部极值,并且在落入局部最优解时能够迅速跳出,进而向全局最优解收敛。通过对多个测试函数的实验分析,本文验证了改进策略在提升 POA 性能方面的显著效果。

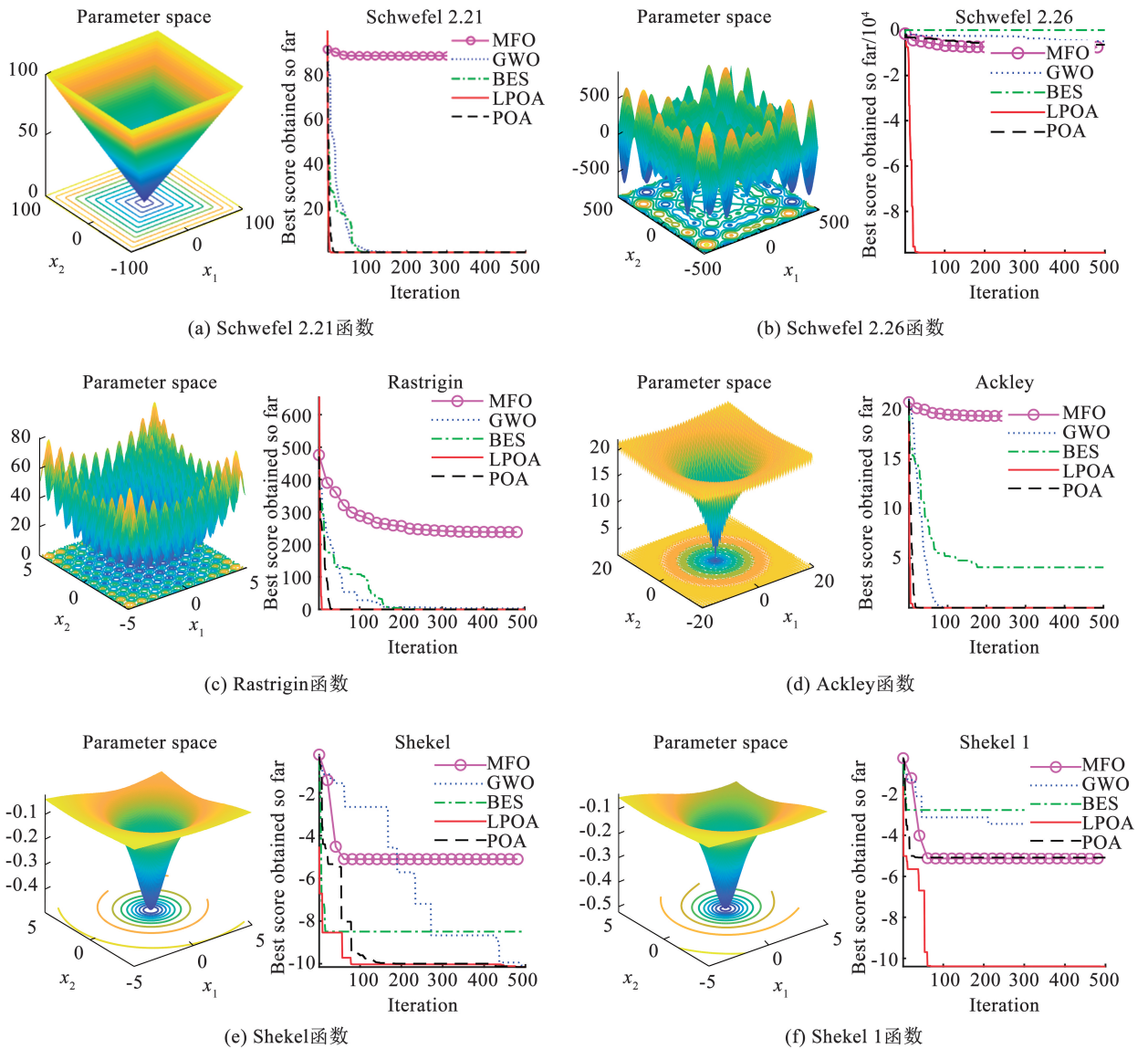


图 4 测试函数曲线

Fig. 4 Test function curve

为了验证改进后的 LPOA 算法是否会因为加入多种机制导致额外的计算负载,影响算法实时性能,本文在相同参数设置下(种群规模为 50,最大迭代次数为 500)分别采用 POA 与 LPOA 在典型测试函数上进行性能对比。设置统一的收敛阈值,记录算法达到该阈值所需的迭代次数及累计计算时间。实验结果见表 1。

由表 1 结果可以看出, LPOA 算法在各测试函数上迭代次数的收敛速度均显著优于 POA 算法。尽管 LPOA 在每次迭代中融入了多种机制,但整体计算时间与 POA 相比仍有所提升,说明 LPOA 在提升全局寻优性能的同时,其计算负载的增加不会对系统带来实时性风险。

表 1 收敛结果

Tab. 1 Convergence results

函数名称	收敛阈值	算法	迭代次数	计算时间/s
Schwefel 2.21	1×10^{-50}	POA	476	1.418
		LPOA	192	1.213
Schwefel 2.26	-8 500	POA	433	1.138
		LPOA	3	0.796
Rastrigin	1×10^{-100}	POA	109	0.882
		LPOA	24	0.773
Ackley	1×10^{-13}	POA	120	0.873
		LPOA	52	0.874
Shekel	-10	POA	167	0.893
		LPOA	43	0.918
Shekel 1	-10.4	POA	479	1.372
		LPOA	126	1.028

4 LPOA 优化的 MPC 控制器设计

4.1 LPOA 优化的 MPC 控制器参数整定过程

LPOA 优化车辆横、纵协同 MPC 轨迹跟踪控制器的参数整定流程,如图 5 所示。

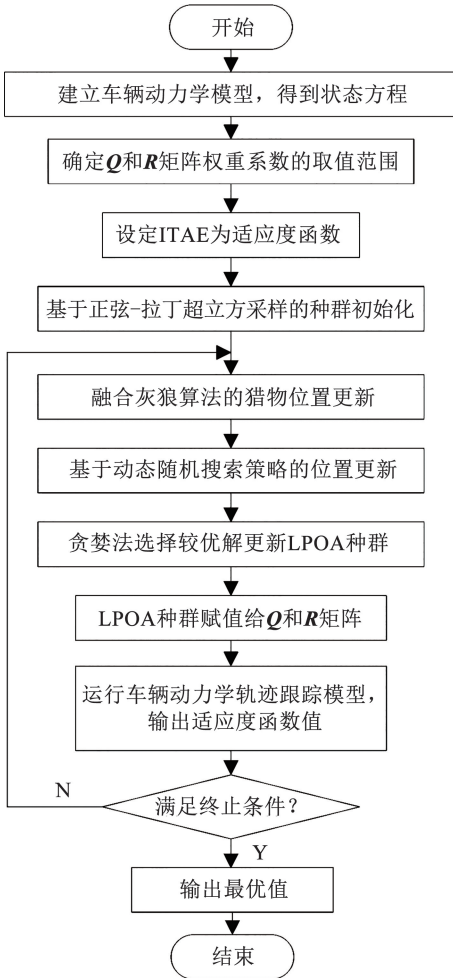


图 5 LPOA-MPC 参数整定过程

Fig.5 Parameter tuning process of LPOA-MPC

4.2 LPOA 优化横纵向 MPC

利用 LPOA 优化纵向 MPC 的控制器结构如图 6 所示。

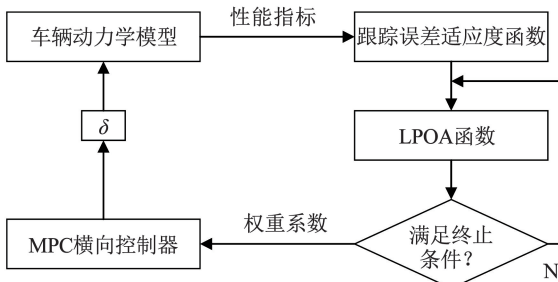


图 6 横向控制器结构

Fig.6 Lateral controller structure

横向误差和航向角误差是评估车辆轨迹跟踪精度的关键性能指标。因此,为确保优化结果的准确

性和稳定性,对横向误差和航向角误差进行归一化处理,归一化的目的是消除不同量纲对权重分配的影响,从而避免在控制器优化过程中出现不平衡现象。同时,为了综合考虑两个误差量的影响,通过加权组合将两个误差量整合成一个综合评价指标,具体如下:

$$\varphi_e = \frac{\varphi - \varphi_{ref} - \varphi_{e_min}}{\varphi_{e_max} - \varphi_{e_min}} \quad (41)$$

$$y_e = \frac{y - y_{ref} - y_{e_min}}{y_{e_max} - y_{e_min}} \quad (42)$$

$$e = ky_e + (1 - k)\varphi_e$$

式中: φ_e 为归一化后的航向角误差, φ 为当前航向角, φ_{ref} 为目标航向角, φ_{e_max} 、 φ_{e_min} 分别为航向角误差的最大值和最小值, y_e 为归一化后的横向误差, y 为当前横向位置, y_{ref} 为目标横向位置, y_{e_max} 、 y_{e_min} 分别为横向误差的最大值和最小值, e 为综合跟踪误差, k 为加权调整系数。

ITAE 是一种用于评价控制系统性能的指标,结合了误差的大小和持续时间,尤其关注误差长时间存在时的影响。ITAE 指标在系统性能优化中具有显著优势,其核心在于对持续时间较长的误差赋予更高权重,从而有效地缩短系统误差存在的时间,并提升系统响应速度。在控制系统优化时,将 ITAE 指标作为目标函数,用于指导控制器的设计或调节,既可使控制系统快速响应,又能减少系统的长期误差。其具体数学表达式为

$$E_{ITAE} = \int |e| \cdot t dt \quad (43)$$

式中 t 为仿真时间。

4.3 LPOA 优化纵向 MPC

利用 LPOA 优化纵向 MPC 的控制器结构如图 7 所示。

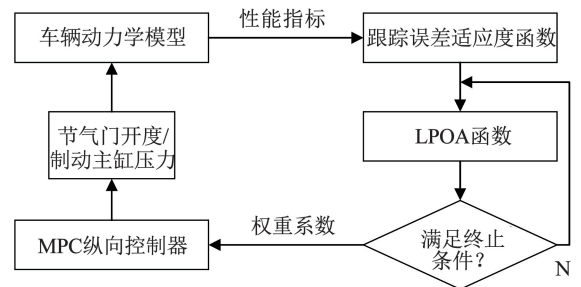


图 7 纵向控制器结构

Fig.7 Longitudinal controller structure

在 LPOA 优化纵向 MPC 控制器控制系统中,本文同样采用了式(43)所示的基于纵向速度误差的 ITAE 指标作为纵向跟踪误差适应度函数。通过 ITAE 指标对速度误差在时间维度上的加权,实现对

系统长期误差的有效抑制。

在此基础上,本文进一步将优化后的横向控制器与纵向控制器进行集成。该联合控制结构通过实

现横向控制器与纵向控制器的协同优化,有效提升了车辆在复杂路径工况下的全局轨迹跟踪精度及运行稳定性。横、纵联合轨迹跟踪控制流程如图 8 所示。

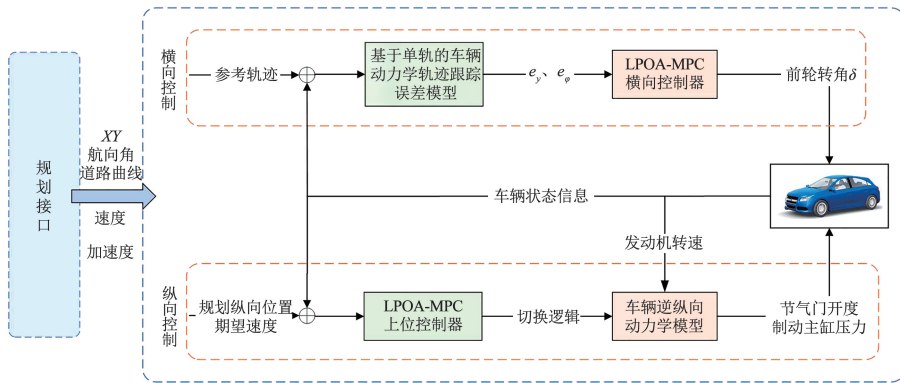


图 8 横、纵联合轨迹跟踪控制流程

Fig. 8 Combined horizontal and longitudinal trajectory tracking control

5 仿真结果分析

5.1 仿真车辆与控制器参数

为确保 MPC 控制器在物理可行性与安全性下运行,本文对控制量设定如表 2 所示约束 (v_{max} 为式(48)最大安全速度),仿真车辆参数见表 3。

MPC 控制器的控制性能主要受预测时域 N_p 、控制时域 N_c 、权重矩阵 Q 和 R 的影响。为评估本文提出的 LPOA-MPC 控制器的优化效果,在仿真实验中,优化前后的控制器预测时域 N_p 、控制时域 N_c 保

持一致。横向控制器选取 $N_p = 20, N_c = 8$;横向控制器优化前的 Q, R 矩阵为 $Q = \text{diag}(10, 10), R = 1$;横向控制器优化后的 Q, R 矩阵为 $Q = \text{diag}(80.8999, 47.0302), R = 2.3286$;其中, Q_{11} 为横向误差权重, Q_{22} 为航向角误差权重。纵向控制器选取 $N_p = 25, N_c = 10$;纵向控制器优化前的 Q, R 矩阵为 $Q = \text{diag}(100, 100), R = 1$;纵向控制器优化后的 Q, R 矩阵为 $Q = \text{diag}(68.3779, 22.8030), R = 4.1940$;其中, Q_{11} 为速度误差权重, Q_{22} 为加速度误差权重。

表 2 控制量约束

Tab. 2 Control variables constraints

前轮转角 $\delta/(^\circ)$	前轮转角增量 $\Delta\delta/(^\circ)$	质心侧偏角 $\beta/(^\circ)$	加速度 $/(m \cdot s^{-2})$	安全速度 $v_{safe}/(m \cdot s^{-1})$	速度 $/(m \cdot s^{-1})$
-10 ~ 10	-0.47 ~ 0.47	-12 ~ 12	-5 ~ 5	$\min(30, v_{max})$	$0 \sim v_{safe}$

表 3 车辆模型参数

Tab. 3 Vehicle model parameters

车辆质量	质心至前轴距离	质心至后轴距离	绕 z 轴转动惯量	车辆质心高度	轮胎纵向刚度	轮胎侧偏刚度
m/kg	a/m	b/m	$I_z/(kg \cdot m^2)$	H/m	$C_1/(N \cdot \text{rad}^{-1})$	$C_c/(N \cdot \text{rad}^{-1})$
1 723	1.232	1.468	4 175	0.54	65 800	61 520

5.2 横向控制模型仿真测试

为了验证 LPOA-MPC 车辆横向轨迹跟踪控制器的有效性和可行性,搭建基于 MATLAB/Simulink 与 CarSim 的联合仿真环境,选取双移线工况作为参考轨迹进行仿真验证。轨迹方程如下:

$$Y(X) = \frac{d_{y1}}{2}(1 + \tanh(z_1)) - \frac{d_{y2}}{2}(1 + \tanh(z_2)) \quad (44)$$

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

本节中仅考虑横向控制,设定车辆的纵向速度匀速不变,为全面评估所设计横向控制算法的跟踪性能与动态响应能力,本文在双移线工况下分别选取 36、72 km/h 作为仿真测试速度,分别代表低速与中高速工况。其中,前者用于评估路径跟踪精度,后者用于验证系统在高速动态条件下的控制稳定性和动态响应能力,具有良好的代表性,仿真结果如图 9 所示。

在 36、72 km/h 两种速度工况下,对 MPC 和 LPOA-MPC 在横向轨迹跟踪控制性能方面进行对比分析。由图 9(a) 横向误差曲线可以看出, LPOA-

MPC 在两种速度工况下的横向误差的波动幅度均明显小于 MPC。在 36 和 72 km/h 工况下, LPOA-MPC 相比 MPC 的最大误差幅值分别降低了 48.0% 和 30.2%, 这说明 LPOA-MPC 控制器能够有效提升控制器的横向轨迹跟踪精度。由图 9(b) 航向角误差

差可以看出, LPOA-MPC 的偏差变化更加平稳, 峰值误差较 MPC 更小, 说明其在对目标方向的跟踪时稳定性更高。从图 9(c) 中的横摆角速度误差中可知, LPOA-MPC 控制器能够有效抑制车辆的横摆变化, 提升车辆行驶中的平稳性和驾乘体验。

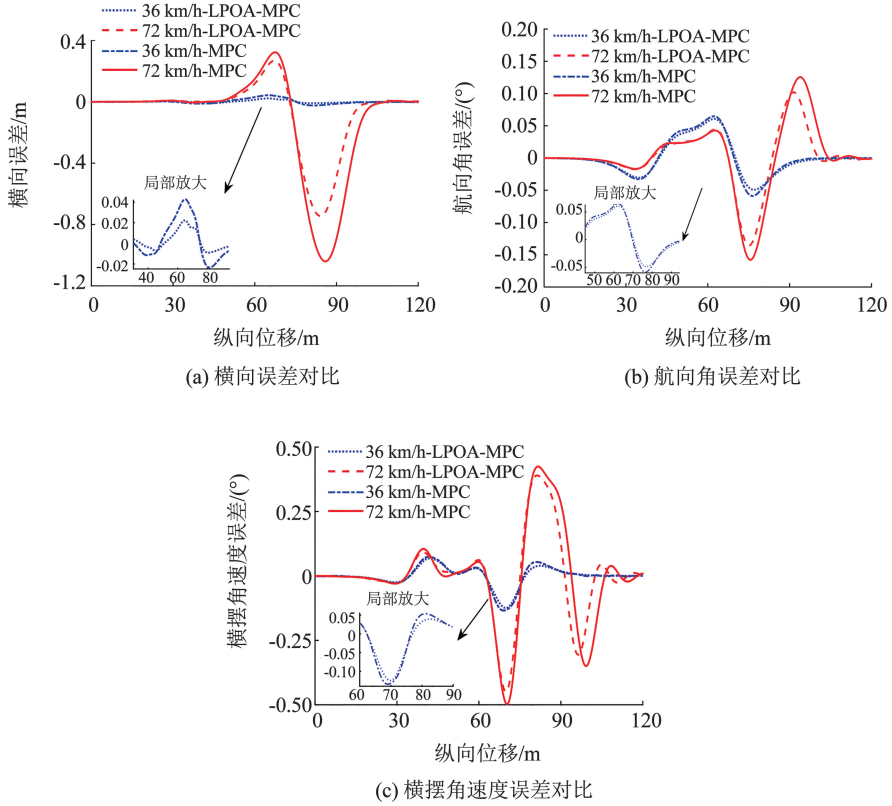


图 9 横向参数对比

Fig. 9 Horizontal parameter comparison

综上所述, 本文提出的 LPOA-MPC 控制器不仅能够有效减少车辆偏离期望轨迹的横向误差和偏航角误差, 还能够降低横摆角的剧烈波动, 从而提升驾驶舒适性和安全性, 更可适应复杂的动态环境, 为车辆的横向轨迹跟踪控制提供更优的解决方案。

5.3 纵向控制模型仿真测试

为了测试纵向 MPC 控制器在复杂车速变化场景下的纵向轨迹跟踪性能, 采用 Zx28 分段线性速度轨迹^[18]作为基准测试工况(图 10)。Zx28 速度轨迹曲线包含多种不同突变加速度的动态特征(1.5、1.0、0.5、0、-1.0、-2.0、-3.0 m/s²), 可以系统地评估控制器在非连续加速度激励下的动态响应特性。

为定量比较控制器的跟踪性能, 采用平均绝对误差(mean absolute error, MAE) 和均方根误差(root mean square error, RMSE) 作为评价指标, 其定义如下:

$$A = \frac{1}{M} \sum_{i=1}^M |e_i| \quad (45)$$

$$R = \sqrt{\frac{1}{M} \sum_{i=1}^M e_i^2} \quad (46)$$

式中: A 为平均绝对误差, R 为均方根误差, M 为总采样点数, i 为采样时刻, e_i 为第 i 个采样时刻的误差。

在 Zx28 速度曲线工况下, 针对 LPOA 优化前后的 MPC 控制器的纵向控制性能进行对比分析, 其仿真结果如图 11 所示。

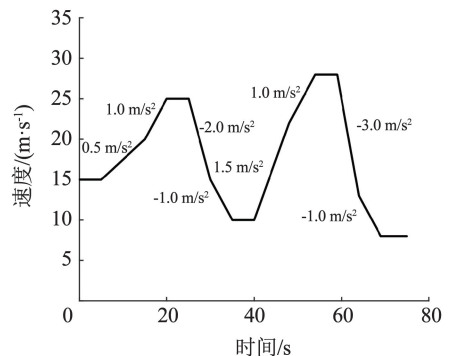


图 10 参考速度轨迹

Fig. 10 Reference speed trajectory

图 11 中: V_{ref} 为 Zx28 分段速度轨迹曲线, $V_{LPOA-MPC}$ 为 LPOA-MPC 控制器速度曲线, V_{MPC} 为 MPC 控制器速度曲线, a_{ref} 为 Zx28 分段加速度轨迹曲线, $a_{LPOA-MPC}$ 为 LPOA-MPC 控制器加速度曲线, a_{MPC} 为 MPC 控制器加速度曲线, LPOA-MPC Error 为 LPOA-

MPC 控制器速度误差曲线, MPC Error 为 MPC 控制器速度误差曲线。

为了进一步对控制效果进行量化评价, 本文采用式(45)的 MAE 与式(46)的 RMSE 作为速度跟踪性能评价指标, 并据此得到表 4 所示的量化数据。

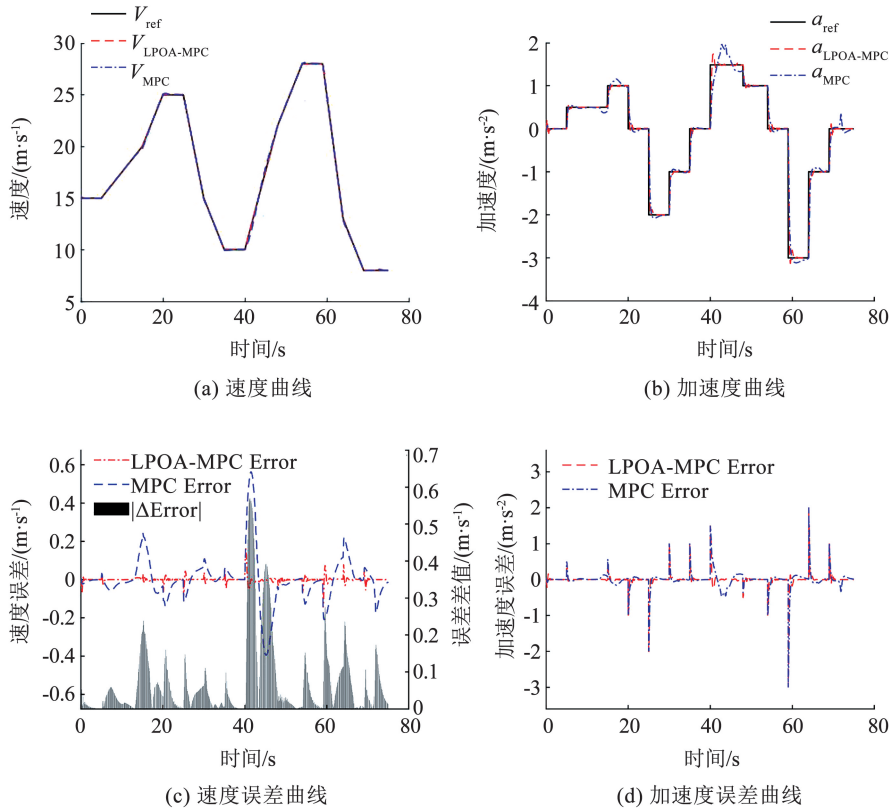


图 11 纵向参数对比

Fig. 11 Longitudinal parameter comparison

表 4 速度跟踪性能对比

Tab. 4 Comparison of speed tracking performance

控制器	速度($m \cdot s^{-1}$)		加速度($m \cdot s^{-2}$)		位移/m	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
MPC	0.563	0.128	2.997	0.253	1.438	0.343
LPOA-MPC	0.169	0.017	0.169	0.017	0.010	0.061

由图 11(a)和(b)可以看出, 尽管 LPOA 优化前后的 MPC 纵向控制器在纵向速度跟踪过程中均表现出良好的跟踪性能, 但总体上 LPOA-MPC 纵向控制器在控制过程中跟踪精度更高, 展现了更优的控制性能。由图 11(c)可以看出, LPOA-MPC 相比 MPC 纵向控制器速度误差曲线波动幅值明显减小, 整体更加紧密地围绕在零误差基准线附近, 说明其与参考速度轨迹更加吻合, 可以有效减少速度累计误差。图 11(d)结果显示, LPOA-MPC 纵向控制器在对加速度的跟踪过程中更加平稳、精确, 并且在加

速度频繁突变的情况下, 能够更迅速、准确地做出响应, 从而提升系统的动态响应性能和稳定性。表 4 的数据也清晰地反映出 LPOA-MPC 相比 MPC 纵向控制器控制性能的显著提升。

综上所述, LPOA-MPC 相比 MPC 纵向控制器在对复杂速度曲线跟踪过程中, 其跟踪精度、稳定性以及动态响应能力等方面均具有显著提升。

5.4 横纵联合控制模型仿真测试

为了验证提出的基于 LPOA-MPC 横纵联合协同优化轨迹跟踪控制器的控制性能及通用性, 本文选取国家标准 GB/T 6323—2014《汽车操纵稳定性试验方法》中规定的蛇形工况作为参考轨迹。该工况作为一种典型的动态操纵稳定性测试场景, 能够有效评估车辆在连续变道过程中的横向稳定性、转向响应特性以及横纵向动力学耦合效应。通过“S”形路径模拟车辆在紧急避障或高速变道等复杂驾驶场景中的运动行为, 具有较高的工程和学术研究价值。具体轨迹方程如下:

$$Y_{\text{ref}} = \begin{cases} 0, & X \leq 20 \\ 0.625 \left(1 + \sin \left(\frac{\pi}{40} (X + 40) \right) \right), & 20 < X \leq 60 \\ 1.25 \cos \left(\frac{\pi}{30} X \right), & 60 < X \leq 240 \\ 0.625 \left(1 + \cos \left(\frac{\pi}{40} X \right) \right), & 240 < X \leq 280 \\ 0, & 280 < X \end{cases} \quad (47)$$

车辆在弯道行驶时, 车辆最大安全车速 v_{max} 受到道路工况限制, 当离心力与路面摩擦力平衡时, 可以得到基于侧滑的最大安全车速为:

$$v_{\text{max}} = \sqrt{\frac{g\mu}{r_o}} \quad (48)$$

式中: μ 为路面附着系数, r_o 为道路曲率。

基于式(48)所确定的基于侧滑条件的最大安全车速, 本文设计车辆的初始参考速度为 15 m/s, 即车辆以 15 m/s 的初始速度匀速行驶, 随后通过匀减速将速度降低至 10 m/s, 并在该速度下保持匀速行驶以通过连续弯道, 最后通过匀加速将速度恢复至 15 m/s, 完成整个蛇形工况的轨迹跟踪。车辆在跟踪蛇形轨迹工况过程中的横、纵向位置信息如图 12 所示。图 12 中: $V_{\text{reference}}$ 为参考速度轨迹曲线, $V_{\text{LPOA-MPC}}$ 为 LPOA-MPC 控制器速度曲线, V_{MPC} 为 MPC 控制器速度曲线, LPOA-MPC Error 为 LPOA-MPC 控

制器速度误差曲线, MPC Error 为 MPC 控制器速度误差曲线。

由图 12(a)和(b)可以看出, LPOA 优化前后的 MPC 横、纵联合轨迹跟踪控制器均能够在变速度条件下实现对蛇形工况的有效跟踪。但相比之下, LPOA-MPC 的轨迹曲线与参考轨迹的贴合度更高, 尤其在连续弯道过渡阶段, 其横向误差幅值被有效抑制在 ± 0.03 m 以内, 且未出现明显的滞后或超调现象。图 12(c)与(d)结果显示, 与 MPC 控制器相比, LPOA-MPC 的航向角误差和横摆角速度误差的振荡幅值均有明显降低, 说明优化后的控制器能够更快适应路面曲率变化, 在保证跟踪精度的前提下, 显著提升了横摆动力学的稳定性。由图 12(e)与(f)可以看出, LPOA-MPC 控制器在匀减速与匀加速阶段的过渡更为平滑, 尤其在当速度由 10 m/s 恢复至 15 m/s 时, LPOA-MPC 控制器误差收敛时间显著缩小, 说明优化后的 MPC 控制器在速度变化时, 能够快速进行控制响应, 此外, 误差插值的定量对比显示, LPOA-MPC 的速度最大误差降低了 39.1%。

综上所述, LPOA-MPC 控制器在变速度下跟踪蛇形轨迹时, 表现出更强的横向控制精度和纵向速度协调能力。LPOA-MPC 控制器在提升车辆轨迹跟踪精度、改善操纵稳定性以及降低横纵向耦合干扰等方面均具有显著优势。

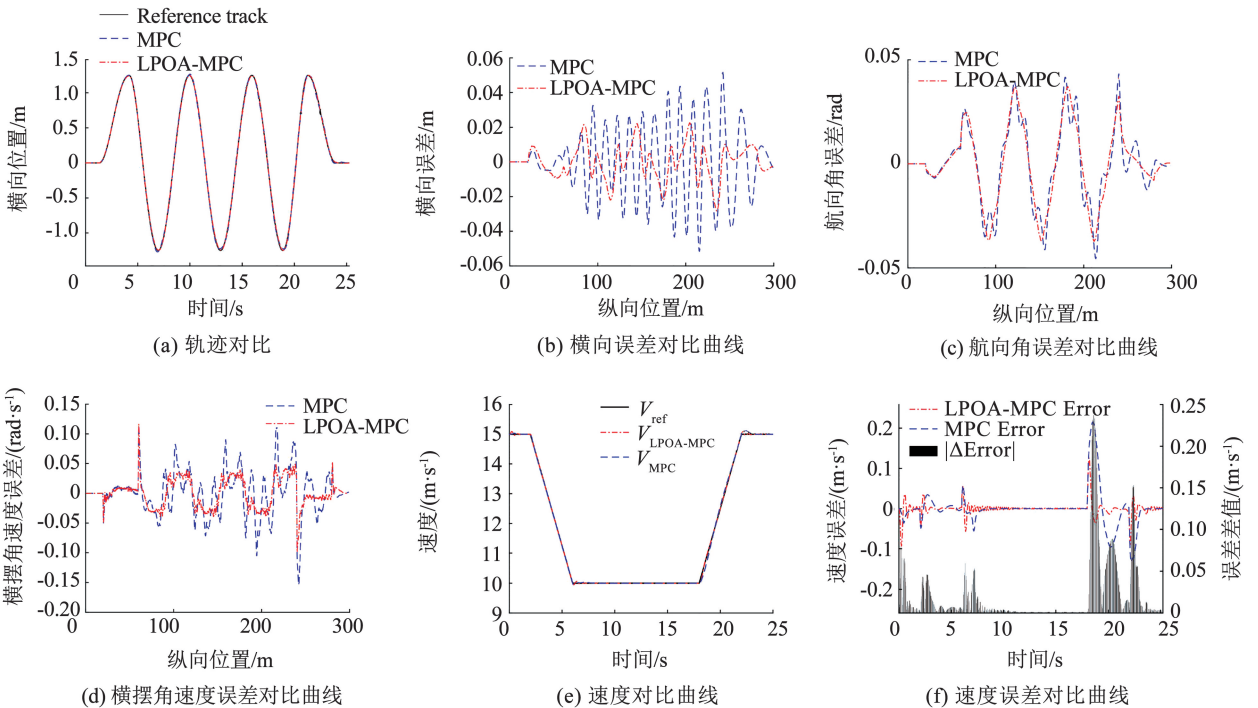


图 12 横、纵联合参数对比

Fig. 12 Comparison of lateral and longitudinal joint parameters

6 结 语

本文基于车辆动力学模型,提出了一种融合多种机制的 LPOA 算法,用于优化横、纵向 MPC 控制器的权重参数。针对 POA 在解空间内搜索效率低下的问题,融入多种机制以增强其寻优能力和收敛速度。在基准测试函数中,LPOA 相较于 MFO、BES、GWO 等算法展现出更快的收敛速度与更强的全局优化能力,尤其在复杂多峰函数上有效避免了陷入局部最优的问题。同时,分别在双移线、复杂速度曲线以及蛇形工况下,测试验证 LPOA-MPC 横向、纵向以及纵横联合控制器的性能。仿真结果表明,LPOA-MPC 控制器在变速度条件下能够实现车辆对参考轨迹的精准跟踪,并可有效提升车辆行驶过程中的稳定性和动态响应能力,具有一定的实用性。

参 考 文 献

- [1] YUAN Tengfei, ZHAO Rongchen. LQR-MPC based trajectory tracking controller of autonomous vehicle subject to coupling effects and driving state uncertainties[J]. *Sensors*, 2022, 22(15): 5556. DOI:10.3390/s22155556
- [2] LI Sihang, ZHANG Jinhui, GU Shaomeng. Discrete time trajectory tracking control for four-mecanum-wheeled mobile vehicle: An variable gain ADRC method[J]. *IEEE Robotics and Automation Letters*, 2024(9): 7771. DOI:10.1109/LRA.2024.3432354
- [3] LI Zhuo, GUO Yunlong, WANG Gang, et al. Level curve tracking via robust RL-guided model predictive control[J]. *IEEE/CAA Journal of Automatica Sinica*, 2024, 11(12): 2512. DOI:10.1109/JAS.2024.124350
- [4] 蒋凌云, 魏庆来, 张峰华, 等. 基于改进粒子群优化算法的 PID 控制器参数整定[J]. *控制工程*, 2024, 31(3): 470
JIANG Lingyun, WEI Qinglai, ZHANG Fenghua, et al. PID controller parameter tuning based on improved particle swarm optimization algorithm[J]. *Control Engineering of China*, 2024, 31(3): 470. DOI: 10.14107/j.cnki.kzgc.20210776
- [5] ZHOU Bin, WANG Yuhang, ZI Bin, et al. Fuzzy adaptive whale optimization control algorithm for trajectory tracking of a cable-driven parallel robot[J]. *IEEE Transactions on Automation Science and Engineering*, 2024, 21(4): 5149. DOI: 10.1109/TASE.2023.3309049
- [6] LI Haoran, LU Yunpeng, LI Yaqiu, et al. Optimization of model predictive control for autonomous vehicles through learning-based weight adjustment[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2024(8): 1. DOI:10.1109/TITS.2024.3445598
- [7] 李晓龙, 黄鹤, 杨澜, 等. 多策略鹈鹕算法优化的车辆轨迹跟踪 MPC 控制器[J/OL]. (2024-11-13) [2025-01-08]. <https://doi.org/10.16183/j.cnki.jsjtu.2024.316>
LI Xiaolong, HUANG He, YANG Lan, et al. Multi-strategy pelican algorithm optimized MPC controller for vehicle trajectory tracking[J/OL]. (2024-11-13) [2025-01-08]. <https://doi.org/10.16183/j.cnki.jsjtu.2024.316>
- [8] TROJOVSKÝ P, DEGHANI M. Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications[J]. *Sensors (Basel, Switzerland)*, 2022, 22(3): 885. DOI: 10.3390/s22030855
- [9] RAJ A, PUNIA P, KUMAR P. A novel hybrid pelican-particle swarm optimization algorithm (HPPSO) for global optimization problem[J]. *International Journal of System Assurance Engineering and Management*, 2024, 15(8): 3878. DOI: 10.1007/s13198-024-02386-9
- [10] CHEN Shuping, XIONG Guangming, CHEN Huiuan, et al. MPC-based path tracking with PID speed control for high-speed autonomous vehicles considering time-optimal travel[J]. *Journal of Central South University*, 2020, 27(12): 3702. DOI: 10.1007/s11771-020-4561-1
- [11] JIN Long, LIU Longqi, WANG Xingxia, et al. Physical-informed neural network for MPC-based trajectory tracking of vehicles with noise considered[J]. *IEEE Transactions on Intelligent Vehicles*, 2024, 9(3): 4493. DOI: 10.1109/TIV.2024.3358229
- [12] AGUSHAKA J O, EZUGWU A E. Initialisation approaches for population-based metaheuristic algorithms: A comprehensive review[J]. *Applied Sciences*, 2022, 12(2): 896. DOI: 10.3390/app12020896
- [13] MOUSAVIRAD S J, BIDGOLI A A, RAHNAMAYAN S, et al. Tackling deceptive optimization problems using opposition-based DE with center-based latin hypercube initialization[C]// 14th International Conference on Computer Science and Education (ICCSSE). Toronto, Canada: IEEE, 2019: 394. DOI: 10.1109/iccsse.2019.8845360
- [14] WANG Zhen, ZHAO Dong, HEIDARI A A, et al. Improved latin hypercube sampling initialization-based whale optimization algorithm for COVID-19 X-ray multi-threshold image segmentation[J]. *Scientific Reports*, 2024, 14(1): 13239. DOI: 10.1038/s41598-024-63739-9
- [15] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. *Advances in Engineering Software*, 2014, 69(3): 46. DOI: 10.1016/j.advengsoft.2013.12.007
- [16] HAMZACEBI C, KUTAY F. Continuous functions minimization by dynamic random search technique[J]. *Applied Mathematical Modelling*, 2007, 31(10): 2189. DOI:10.1016/j.apm.2006.08.015
- [17] DENG Lingyun, LIU Sanyang. A multi-strategy improved slime mould algorithm for global optimization and engineering design problems[J]. *Computer Methods in Applied Mechanics and Engineering*, 2023, 404: 115764. DOI: 10.1016/j.cma.2022.115764
- [18] 朱敏. 智能车辆纵向速度跟踪与控制方法研究[D]. 北京: 北京理工大学, 2016
ZHU Min. Longitudinal speed tracking and control method for intelligent vehicles[D]. Beijing: Beijing Institute of Technology, 2016