Original Article

Institution of MECHANICAL ENGINEERS
Journal of AUTOMOBILE ENGINEERING

Proc IMechE Part D:
J Automobile Engineering
1–18

DOI: 10.1177/09544070251351672
journals.sagepub.com/home/pid
Sage

Distributed iterative reinforcement learning predictive control of truck platoons

Shuyou Yu[1,2], Zepeng Liu[1], Yunyong Li[1], Hong Chen[3] and Qifang Liu[1]

Abstract

Addressing the issue of insufficient real-time computational capability of the centralized controller in solving multi-objective and multi-constraint nonlinear optimization problems for truck platoons, this paper proposes a synchronous distributed model predictive control strategy based on the Predecessor-Leader-Following communication topology. This approach transforms the global optimization problem of the platoon into local optimization problems for each truck, allowing all following trucks to solve their own optimization problems in parallel. Addressing the challenges of behavior prediction arising from the strong coupling characteristics of truck dynamics, a five-degree-of-freedom nonlinear dynamics model that captures both lateral and longitudinal coupling is developed to predict truck behavior. Additionally, a lane-keeping model is formulated to ensure that the longitudinal velocity of the trucks in the platoon matches that of the lead truck, while keeping the trucks within the designated lane. To reduce computational burden, a distributed iterative reinforcement learning predictive control scheme based on actor-critic networks is introduced. Co-simulation results using Matlab/Simulink and TruckSim demonstrate that the proposed strategy ensures both longitudinal velocity tracking and lateral lane-keeping performance, while providing better computational efficiency than conventional nonlinear model predictive control algorithms.

Keywords

Truck platoon, distributed model predictive control, lateral and longitudinal coupling, lane-keeping, reinforcement learning

Date received: 7 November 2024; accepted: 2 June 2025

Introduction

Throughout the last decades, the rapid and widespread adoption of vehicles raises significant concerns about energy security and traffic issues.[1,2] According to the National Highway Traffic Safety Administration, about 84% of traffic accidents are attributed to human factors.[3] The technology of autonomous vehicle platoons has the potential to significantly reduce the risk of accidents caused by driver fatigue or error.[4–7] Furthermore, the vehicle platoons can significantly reduce air resistance between vehicles. This reduces exhaust emissions and fuel consumption, while increasing road throughput.[8–10]

The accuracy of the nominal model in representing vehicle dynamic characteristics significantly affects vehicle handling stability under high-speed conditions.[11] In existing studies on vehicle platoon modeling, vehicles are often simplified as linear point-mass models, such as the single integrator model, double integrator model, or third-order model.[12–14] However, these models ignore the dynamic characteristics of vehicle systems, which makes them unsuitable under complex driving conditions. A nonlinear longitudinal dynamics model incorporating engine dynamics, rolling resistance, and aerodynamic drag is proposed.[15,16] This model captures the vehicle's longitudinal dynamics more accurately. Nevertheless, these models primarily

[1]Department of Control Science and Engineering, Jilin University, Changchun, China
[2]State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun, China
[3]College of Electronics and Information Engineering, Tongji University, Shanghai, China

Corresponding author:
Qifang Liu, Department of Control Science and Engineering, Jilin University, Room 421, Basic Experimental Building, No. 5988 Renmin Street, Changchun City, Jilin Province, China.
Email: liuqf@jlu.edu.cn

focus on longitudinal dynamics and fail to represent vehicle lateral dynamics in scenarios such as cornering or lane changes, and are therefore only applicable to vehicle platoons traveling on straight roads. For platoons traveling on curved roads, it is necessary to account for lateral dynamics.[17] A nonlinear bicycle model is employed to describe lateral dynamics, and a predictive controller is designed to ensure both longitudinal tracking performance and lateral stability.[18] Similarly, a lateral controller based on a lane-keeping model is developed to ensure that vehicles in the platoon remain within the designated lane.[19] However, these lateral models typically assume constant longitudinal velocity and focus only on lateral and yaw motions, neglecting the influence of longitudinal dynamics. In practice, coupling between longitudinal and lateral motions during cornering exists, and ignoring this coupling degrades the performance of platoon.[20] Moreover, compared with conventional passenger vehicles, trucks have greater mass, wheelbase, turning radius, and moment of inertia.[21] Under conditions involving high lateral/longitudinal acceleration or low road adhesion coefficients, the coupling effect and tire nonlinearities become more pronounced. Consequently, the development of a unified model that accurately characterizes the coupling between lateral and longitudinal dynamics, incorporates the nonlinear behavior of tires, and comprehensively reflects both longitudinal tracking accuracy and lateral stability is of critical importance.

Control of vehicle platoon is classified into centralized and distributed control. The centralized approach extends traditional single-vehicle control strategies, using a central unit to coordinate all vehicles' behaviors. In contrast, distributed control eliminates centralized coordination, with each vehicle employing its own controller for autonomous decision-making. This approach offers greater reliability, adaptability, and robustness, especially in scenarios with restricted communication range and large platoon sizes. As a result, distributed control has become the predominant methodology in vehicle platoon systems. A longitudinal distributed control strategy for connected automated vehicles (CAVs) under communication cyberattacks is proposed.[22] A distributed sliding mode control scheme is developed to ensure coordinated behavior within vehicle platoons.[23] A distributed model reference adaptive control strategy is designed to tackle inherent uncertainties in heterogeneous multi-agent systems.[24] Distributed model predictive control (DMPC), capable of handling multi-input multi-output systems and multi-objective constrained optimization problems, has attracted considerable attention in recent years and has been successfully applied to control of vehicle platoons.[25–27] A distributed model predictive control algorithm is proposed to ensure γ-gain stability of vehicle platoons.[28] A distributed model predictive controller for nonlinear vehicle platoons is developed to guarantee string stability.[29] A distributed model predictive control scheme is formulated to ensure local stability while satisfying multi-criteria string stability requirements.[30] Current DMPC implementations for vehicle platoons are still restricted by computational inefficiency and suboptimality in solving constrained optimization problems. These challenges become more severe when considering nonlinear vehicle dynamics with coupled longitudinal and lateral characteristics. Under such conditions, the DMPC algorithm may fail to compute feasible solutions within the required sampling intervals. Therefore, efficiently solving optimization problems in DMPC for vehicle platoons while accounting for the coupling between longitudinal and lateral dynamics remains a critical technical bottleneck in intelligent transportation systems.

Reinforcement learning (RL), as an advanced policy optimization methodology, has shown considerable promise in complex system control and has been widely adopted in intelligent transportation systems. Particularly, multi-agent reinforcement learning (MARL) has demonstrated effective multi-vehicle coordination capabilities in control of vehicle platoons.[31] A MARL-based cooperative adaptive cruise control (CACC) strategy is proposed to optimize platoon stability and energy efficiency for CAVs.[32] A distributed control architecture is proposed, where a deep reinforcement learning agent optimizes vehicle platoon acceleration on curved roads through iterative interaction with the lateral controller.[33] A guided deep deterministic policy gradient (DDPG) framework is proposed to enhance the convergence efficiency of RL-based controllers.[34,35] CACC is reformulated as a decentralized MARL task, eliminating the need for centralized controllers during both training and deployment to improve scalability and robustness.[36] However, the aforementioned MARL-based approaches have these limitations: (1) reliance on third-order integrator kinematic models, capturing only position, velocity, and acceleration relationships, ignoring vehicle dynamics coupling and tire nonlinearities; (2) lack of constraint-handling mechanisms inherent to DMPC frameworks.

Reinforcement learning has demonstrated superior exploration capabilities in high-dimensional solution spaces, making it a powerful tool for tackling non-convex optimization problems.[37] The RL-DMPC integrated framework combines the advantage of RL in solving non-convex optimization problems with the strength of DMPC in constraint handling.[38] Current research has made preliminary attempts at integration: a distributed algorithm that integrates deep reinforcement learning with DMPC is proposed, where deep reinforcement learning is utilized for reference trajectory generation, and DMPC is employed to track the trajectories while ensuring collision avoidance among vehicles.[39] However, this integration does not effectively address the computational inefficiency inherent in DMPC. A distributed learning-based predictive control framework is proposed to generate DMPC's closed-loop control policies for multi-robot coordination.[40] Nevertheless, their reliance on fixed communication topologies

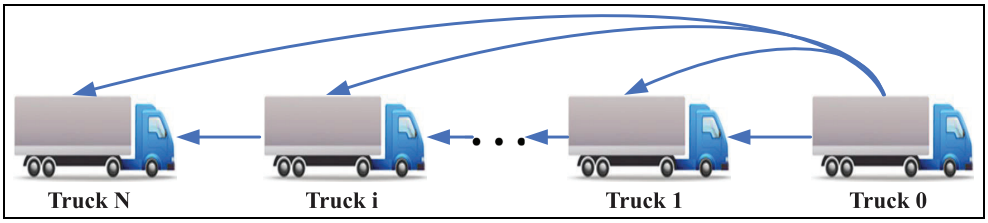

Figure 1. Predecessor-Leader-Following topology.

conflicts with the dynamic networking needs of vehicle platoons. To the best of the authors' knowledge, no systematic study has explored the co-design of RL and DMPC for vehicle platoon controls.

In this paper, a nonlinear model of vehicle platoon is proposed, which combines the five-degree-of-freedom (5-DOF) dynamic model with the lane-keeping model. Then, a distributed model predictive control strategy for vehicle platoons is introduced. Furthermore, an iterative reinforcement learning predictive control (RLPC) algorithm is suggested to effectively handle the constrained optimization problem for each following vehicle. The effectiveness of these algorithms is verified by co-simulation using MATLAB/Simulink and TruckSim. The primary contributions of this paper include:

(1) A 5-DOF nonlinear dynamics model and a lane-keeping model have been employed to construct a vehicle platoon model that effectively captures the coupled lateral and longitudinal characteristics of the vehicles, as well as the nonlinear behavior of tires;
(2) A distributed model predictive controller for vehicle platoons considering lateral and longitudinal coupling is proposed, which achieves cooperative of both lateral and longitudinal coordinated control of vehicle platoons;
(3) This paper proposes an iterative RLPC algorithm based on the actor-critic neural network. The algorithm is designed to generate an explicit closed-loop DMPC policy capable of handling non-convex constrained optimization problems for platoon vehicles in real time. Co-simulation experiments show that the developed controller successfully achieves the lateral and longitudinal control objectives of the vehicle platoon. In comparison to the conventional nonlinear model predictive control (NMPC) algorithm, the introduced iterative RLPC algorithm has been demonstrated to exhibit superior computational efficiency.

The rest of this paper is structured as follows: Section "Problem setup" provides a detailed problem description, covering the communication topology, the vehicle platoon model, and the objectives of vehicle platoon control. Section "Distributed model predictive control strategy" describes a distributed model predictive controller tailored for the vehicle platoon. Section "Iterative reinforcement learning predictive control scheme" introduces the iterative RLPC algorithm. Section "Simulation" shows results of co-simulation experiments using Matlab/Simulink and TruckSim. The conclusion is drawn in Section "Conclusion."

Table 1. Symbols for the vehicle platoon system.

Symbol	Description
x_i	Position of the i^{th} vehicle
v_i^x	Longitudinal velocity of the i^{th} vehicle
v_i^y	Lateral velocity of the i^{th} vehicle
$\dot{\varphi}$	The yaw rate of the i^{th} vehicle
w_i^f	The front wheel angular velocity of the i^{th} vehicle
w_i^r	The rear wheel angular velocity of the i^{th} vehicle
I_i^z	The inertia moment around the z-axis
a_i	The distances from front axle to mass center
b_i	The distances from rear axle to mass center
R_e	The rolling radius of the wheel

Problem setup

In the vehicle platoon, the leading vehicle is numbered 0, while the following vehicles are numbered $1 \cdots N$. The leading vehicle, operated by a human driver, is capable of handling emergencies or unexpected events, which contributes to the overall safety of the platoon. This paper investigates the cooperative control of vehicle platoons in highway scenarios and employs a Predecessor-Leader-Following (PLF) communication topology, as illustrated in Figure 1, where each following vehicle communicates with its preceding vehicle in the platoon. This topology demonstrates low communication latency, making it well-suited for highway environments. The required symbols for the vehicle platoon system are presented in Table 1.

Vehicle dynamics

In this paper, a 5-DOF vehicle dynamics model (Figure 2) is adopted to represent a two-axle truck, incorporating additional degrees of freedom associated with wheel rotation.[41]

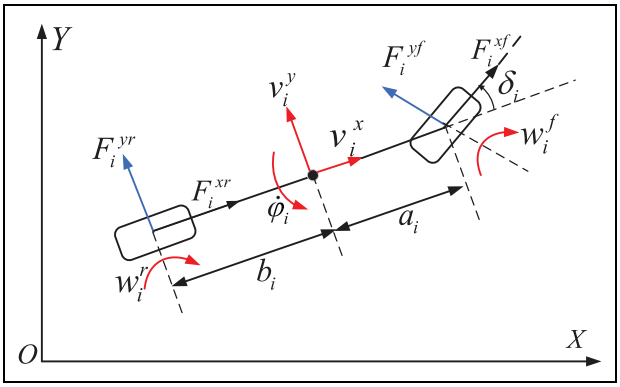

Figure 2. 5-DOF vehicle dynamics model.

The classical 3-DOF dynamics model considers the vehicle's motion in the longitudinal, lateral, and yaw directions, which is expressed as follows:

$$\begin{cases} m_i\dot{v}_i^x - m_i v_i^y\dot{\varphi}_i = F_i^{xf}\cos\delta_i - F_i^{yf}\sin\delta_i + F_i^{xr} \\ m_i\dot{v}_i^y - m_i v_i^x\dot{\varphi}_i = F_i^{xf}\sin\delta_i + F_i^{yf}\cos\delta_i + F_i^{yr} \\ I_i^z\dot{\varphi}_i = (F_i^{xf}\sin\delta_i + F_i^{yf}\cos\delta_i)a_i - F_i^{yr}b_i \end{cases} \tag{1}$$

where v_i^x and v_i^y denote the longitudinal and lateral velocities of the i^{th} vehicle, $\dot{\varphi}_i$ denotes its yaw rate, m_i denotes its mass, F_i^{xf} and F_i^{xr} denote the longitudinal forces of the front and rear tires, F_i^{yf} and F_i^{yr} denote the lateral forces of the front and rear tires, a_i and b_i denote the distances from the center of mass to the front and rear axles, δ_i denotes the front wheel steering angle, and I_i^z denotes the moment of inertia of the i^{th} vehicle.

The forces on the wheel are shown in Figure 3, and the corresponding dynamic equations are as follows:

$$\begin{cases} \dot{w}_i^f = \frac{T_i^d - R_e F_i^{xf}}{J_i^f} \\ \dot{w}_i^r = \frac{T_i^d - R_e F_i^{xr}}{J_i^r} \end{cases} \tag{2}$$

where w_i^f and w_i^r denote the angular velocities of the front and rear wheels of the i^{th} vehicle, J_i^f and J_i^r denote the moments of inertia of the front and rear wheels, R_e denotes the effective rolling radius of the wheel, and T_i^d denotes the driving/braking torque.

Combing the classical 3-DOF dynamics model with (2), then a 5-DOF dynamics model is obtained:

$$\begin{cases} \dot{v}_i^x = v_i^y\dot{\varphi}_i + \frac{F_i^{xf}\cos\delta_i - F_i^{yf}\sin\delta_i + F_i^{xr}}{m_i} \\ \dot{v}_i^y = -v_i^x\dot{\varphi}_i + \frac{F_i^{xf}\sin\delta_i + F_i^{yf}\cos\delta_i + F_i^{yr}}{m_i} \\ \ddot{\varphi}_i = \frac{(F_i^{xf}\sin\delta_i + F_i^{yf}\cos\delta_i)a_i - F_i^{yr}b_i}{I_i^z} \\ \dot{w}_i^f = \frac{T_i^d - R_e F_i^{xf}}{J_i^f} \\ \dot{w}_i^r = \frac{T_i^d - R_e F_i^{xr}}{J_i^r}. \end{cases} \tag{3}$$

The state variables of this model include the longitudinal velocity, lateral velocity, yaw rate, front wheel angular velocity, and rear wheel angular velocity of the vehicle. The control inputs are the front wheel steering angle and driving/braking torque.

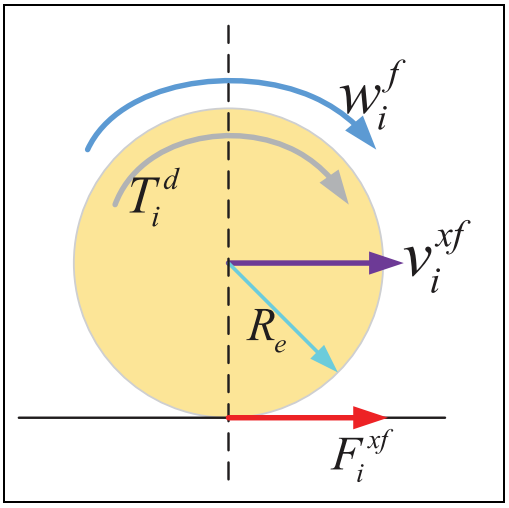

Figure 3. Forces on the wheel.

As a crucial component of vehicle dynamics, an accurate tire model plays a significant role in controller design. In this paper, the Magic Formula is employed to calculate the tire force,[42] which is:

$$\begin{cases} F_i^x = D\sin(C\arctan(Bk_i - E(Bk_i \\ \quad -\arctan Bk_i))) \\ F_i^y = D\sin(C\arctan(B\alpha_i - E(B\alpha_i \\ \quad -\arctan B\alpha_i))) \end{cases} \tag{4}$$

where B, C, D, and E denote the stiffness, shape, peak, and curvature factors, respectively. The terms k_i and α_i denote the slip ratio and slip angle of the tire, and F_i^x and F_i^y denote the longitudinal and lateral forces of the tire.

The slip ratio of front and rear wheel are defined as follows:

$$\begin{cases} k_i^f = \frac{w_i^f\cdot R_e - v_i^{xf}}{|v_i^{xf}|} \\ k_i^r = \frac{w_i^r\cdot R_e - v_i^{xr}}{|v_i^{xr}|}. \end{cases} \tag{5}$$

The slip angle of front and rear wheel are defined as follows:

$$\begin{cases} \alpha_i^f = \mathrm{sgn}\left(v_i^{xf}\right)\cdot\arctan\left(\frac{v_i^{yf}}{v_i^{xf}}\right) \\ \alpha_i^r = \mathrm{sgn}\left(v_i^{xr}\right)\cdot\arctan\left(\frac{v_i^{yr}}{v_i^{xr}}\right) \end{cases} \tag{6}$$

where v_i^{xf} and v_i^{xr} denote the longitudinal velocities of the front and rear wheels in the tire coordinate system, v_i^{yf} and v_i^{yr} denote the lateral velocities of the front and rear wheels in the tire coordinate system. Furthermore,

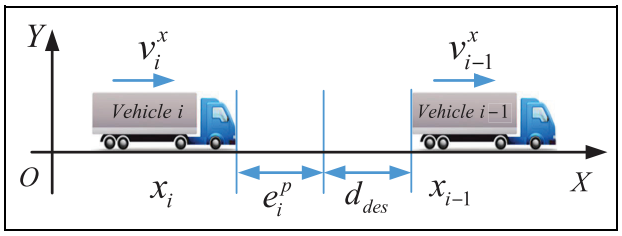

Figure 4. Constant inter-vehicle spacing.

$$\begin{cases} v_i^{xf} = v_i^{x,1}\cos(\delta_i) + v_i^{y,1}\sin(\delta_i) \\ v_i^{yf} = -v_i^{x,1}\sin(\delta_i) + v_i^{y,1}\cos(\delta_i) \\ v_i^{xr} = v_i^{x,2} \\ v_i^{yr} = v_i^{y,2} \end{cases} \tag{7}$$

where $v_i^{x,1}$ and $v_i^{x,2}$ denote the longitudinal velocities of the front and rear wheels in the vehicle coordinate system, $v_i^{y,1}$ and $v_i^{y,2}$ denote the lateral velocities of the front and rear wheels in the vehicle coordinate system, and

$$\begin{cases} v_i^{x,1} = v_i^x \\ v_i^{y,1} = v_i^y + \dot{\varphi}_i \cdot a_i \\ v_i^{x,2} = v_i^x \\ v_i^{y,2} = v_i^y - \dot{\varphi}_i \cdot b_i. \end{cases} \tag{8}$$

The lane-keeping model

The longitudinal position error of the i^{th} vehicle in the platoon is defined as follows:

$$e_i^p = x_i - (x_{i-1} - d_{des}) \tag{9}$$

where x_i and x_{i-1} denote the longitudinal positions of the i^{th} vehicle and its predecessor, respectively. This paper adopts the constant inter-vehicle spacing (Figure 4) policy,[43] that is, $d_{des} = d_0$.

As shown in Figure 5, e_i^y denotes the lateral position error between the vehicle and the center line of the lane, and e_i^φ denotes the heading angle error, which is calculated as:

$$e_i^\varphi = \varphi_{i,des} - \varphi_i \tag{10}$$

where φ_i and $\varphi_{i,des}$ denote the heading angle of the vehicle and the tangential angle of the lane, respectively.

Therefore, the relationships for Vehicle-to-Road and Vehicle-to-Vehicle are as follows[44]:

$$\begin{cases} \dot{e}_i^p = v_i^x - v_{i-1}^x \\ \dot{e}_i^y = v_i^x e_i^\varphi - v_i^y - L\dot{\varphi}_i \\ \dot{e}_i^\varphi = \dot{\varphi}_{i,des} - \dot{\varphi}_i \end{cases} \tag{11}$$

where $\dot{\varphi}_{i,des} = v_i^x/R$ denotes the desired yaw rate of the i^{th} vehicle, L denotes the look-ahead distance, and R denotes the radius of curvature at the look-ahead point on the road.

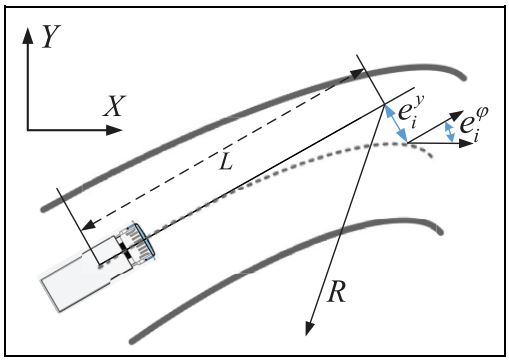

Figure 5. Lane-keeping model.

Assumption 1: In this study, the following assumptions are made regarding the communication system: (1) vehicles in the platoon maintain clock synchronization; (2) inter-vehicle communication is ideal, without channel fading, packet loss, or communication delay; (3) all sensor measurements are noise-free.

Assumption 2: The leading vehicle, operated by a human driver, has its longitudinal position and velocity known, and the road curvature is known.

Control objective of vehicle platoons

Each following vehicle in the platoon, operating within a distributed control framework, collects state data via on-board sensors and V2V communication. The control objectives of vehicle platoons are as follows:

(1) All vehicles in the platoon maintain the same velocity as the leading vehicle while keeping a safe distance to the front and rear vehicles.

$$\begin{cases} \lim\limits_{t\to\infty} \|v_i^x(t) - v_0^x(t)\| = 0 \\ \lim\limits_{t\to\infty} \|x_{i-1}(t) - x_i(t) - d_{des}\| = 0. \end{cases} \tag{12}$$

(2) The trajectories of vehicles in the platoon should align with the prescribed lane, that is, the lateral position error and heading angle error of the i^{th} vehicle relative to the lane should be minimized as much as possible.

$$\begin{cases} \lim\limits_{t\to\infty} \|e_i^y(t)\| = 0 \\ \lim\limits_{t\to\infty} \|e_i^\varphi(t)\| = 0. \end{cases} \tag{13}$$

Distributed model predictive control strategy

This section proposes a distributed model predictive controller that accounts for the coupling between

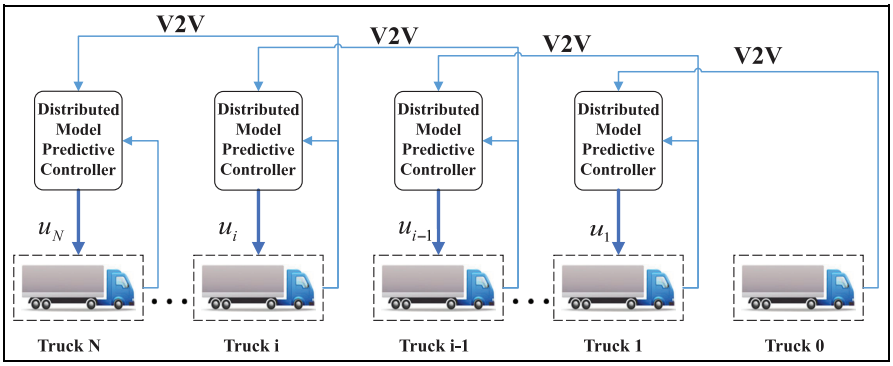

Figure 6. The distributed control framework of the vehicle platoon.

longitudinal and lateral dynamics. The schematic diagram of the proposed control architecture is shown in Figure 6.

Integrated vehicle platoon model

By integrating the 5-DOF dynamic model (3) with the lane-keeping model (11), an integrated vehicle platoon model is derived as follows:

$$\begin{cases} \dot{v}_i^x = v_i^y \dot{\varphi}_i + \frac{F_i^{xf}\cos\delta_i - F_i^{yf}\sin\delta_i + F_i^{xr}}{m_i} \\ \dot{v}_i^y = -v_i^x \dot{\varphi}_i + \frac{F_i^{xf}\sin\delta_i + F_i^{yf}\cos\delta_i + F_i^{yr}}{m_i} \\ \ddot{\varphi}_i = \frac{\left(F_i^{xf}\sin\delta_i + F_i^{yf}\cos\delta_i\right)a_i - F_i^{yr}b_i}{I_i^z} \\ \dot{w}_i^f = \frac{T_i^d - R_e F_i^{xf}}{J_i^f} \\ \dot{w}_i^r = \frac{T_i^d - R_e F_i^{xr}}{J_i^r} \\ \dot{e}_i^p = v_i^x - v_{i-1}^x \\ \dot{e}_i^y = v_i^x e_i^\varphi - v_i^y - L\dot{\varphi}_i \\ \dot{e}_i^\varphi = \dot{\varphi}_{i,des} - \dot{\varphi}_i. \end{cases} \tag{14}$$

The state of the vehicle platoon is defined as follows:

$$x_i = \left[\, v_i^x \quad v_i^y \quad \dot{\varphi}_i \quad w_i^f \quad w_i^r \quad e_i^p \quad e_i^y \quad e_i^\varphi \,\right]^T. \tag{15}$$

The output of the vehicle platoon is defined as follows:

$$y_i = \left[\, v_i^x \quad e_i^p \quad e_i^y \quad e_i^\varphi \,\right]^T. \tag{16}$$

The control inputs include the driving/braking torque and the front wheel steering angle, as follows:

$$u_i = \left[\, T_i^d \quad \delta_i \,\right]^T. \tag{17}$$

Therefore, system (14) can be rewritten as follows:

$$\begin{cases} \dot{x}_i = \bar{f}_i(x_i, u_i) \\ y_i = C_i x_i \end{cases} \tag{18}$$

where $x_i \in R^8$, $u_i \in R^2$,

$$C_i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

With a sampling time of T_s, system (18) is discretized as:

$$\begin{cases} x_i(k+1) = f_i(x_i(k), u_i(k)) \\ y_i(k) = C_i x_i(k). \end{cases} \tag{19}$$

Distributed model predictive controller with coupled longitudinal and lateral dynamics

Based on the integrated vehicle platoon model, a cooperative controller is designed. In the distributed control framework, each following vehicle simultaneously solves its own local optimization problem.

The desired outputs of system (19) are denoted as follows:

$$y_{i,des}(k) = \left[v_{i,des}^x(k) \quad e_{i,des}^p(k) \quad e_{i,des}^y(k) \quad e_{i,des}^\varphi(k)\right]^T \tag{20}$$

where $v_{i,des}^x$ denotes the desired longitudinal velocity of the i^{th} vehicle, with $v_{i,des}^x = v_0^x$. The terms $e_{i,des}^p$, $e_{i,des}^y$ and $e_{i,des}^\varphi$ denote the desired longitudinal position error, lateral position error, and yaw angle error, where $e_{i,des}^p = 0$, $e_{i,des}^y = 0$, and $e_{i,des}^\varphi = 0$.

The tracking error of the i^{th} vehicle is defined as follows:

$$e_i(k) = y_i(k) - y_{i,des}(k).$$

The control sequence over the prediction horizon N_p is defined as follows:

$$U_i(k|k) = \left[u_i{}^T(k|k), u_i{}^T(k+1|k), \cdots, u_i{}^T\left(k+N_p-1|k\right)\right]^T \tag{21}$$

At time k, the optimization problem to be solved by the i^{th} vehicle is formulated as follows:

Problem 1

$$\underset{U_i(k)}{minmize}\, J_i(e_i(k|k), U_i(k|k)) \tag{22a}$$

s.t.

$$x_i(k+j+1|k) = f_i(x_i(k+j|k), u_i(k+j|k)) \tag{22b}$$

$$y_i(k+j|k) = C_i x_i(k+j|k) \tag{22c}$$

$$y_i(k|k) = y_i(k) \tag{22d}$$

$$T^d_{i,min} \leqslant T^d_i(k+j|k) \leqslant T^d_{i,max} \tag{22e}$$

$$\delta_{i,min} \leqslant \delta_i(k+j|k) \leqslant \delta_{i,max} \tag{22f}$$

$$e_i\big(k+N_p|k\big) = 0 \tag{22g}$$

where

$$\begin{aligned} &J_i(e_i(k), U_i(k)) \\ &= \sum_{j=0}^{N_p-1} \Big(\|e_i(k+j|k)\|^2_{Q_i} + \|u_i(k+j|k)\|^2_{R_i} \Big) \end{aligned} \tag{23}$$

The terms Q_i and R_i are symmetric positive definite weight matrices. The terms $T^d_{i,\min}$ and $T^d_{i,\max}$ denote the minimum and maximum torques, where $T^d_{i,\min} = -T^d_{i,\max}$, and $\delta_{i,\min}$ and $\delta_{i,\max}$ denote the minimum and maximum front wheel steering angles, with $\delta_{i,\min} = -\delta_{i,\max}$.

The terminal equality constraint (22g) ensures convergence of the predicted state to the equilibrium point at the end of the control horizon. In the absence of external disturbances and modeling uncertainties, the control inputs beyond the horizon can be set to zero, maintaining the system remains at the equilibrium point.[45,46] The cost function (23) is defined as follows:

$$\begin{aligned} J_i &= \sum_{j=0}^{N_p-1} (\|e_i(k+j|k)\|^2_{Q_i} + \|u_i(k+j|k)\|^2_{R_i}) \\ &= \sum_{j=0}^{\infty} (\|e_i(k+j|k)\|^2_{Q_i} + \|u_i(k+j|k)\|^2_{R_i}) \end{aligned} \tag{24}$$

The cost function in Problem 1 is defined over an infinite horizon. If a solution exists, denoted by $U^*_i(k|k)$, the corresponding predictive control law at time step k is defined as follows:

$$\kappa(x_i(k)) := [\, I_{2\times 2}\ 0\ \cdots\ 0\,]_{2\times 2N_p} U^*_i(k|k), \tag{25}$$

The system under control can be characterized as follows:

$$\begin{aligned} x_i(k+1) &= f_i(x_i(k), \kappa(x_i(k))), \qquad k \geqslant 0 \\ y_i(k) &= C_i x_i(k) \end{aligned} \tag{26}$$

Lemma 1: Suppose that[46]

(a) At $k = 0$, there exists a feasible solution to the constrained optimization Problem 1.
(b) The output y_i exhibits zero-state observability.

For nominal systems that exclude external disturbances and model uncertainties, the following holds:

(1) For any $k > 0$, Problem 1, updated using the state measurement $x_i(k)$, admits a solution.
(2) The closed-loop system (26), composed of (25), is nominally asymptotically stable.

The terminal equality constraint increases computational complexity and may render the optimization problem infeasible. To address this, constraint (22g) is reformulated as a soft constraint, which ensuring both computational efficiency and the feasibility of Problem 1, while driving the terminal state to converge to the equilibrium point, thereby guaranteeing the asymptotic stability of the closed-loop system.[45,46] Meanwhile, the cost function of Problem 1 is modified as follows:

$$\begin{aligned} &J_i(e_i(k), U_i(k)) \\ &\quad = \sum_{j=0}^{N_p-1} \Big(\|e_i(k+j|k)\|^2_{Q_i} + \|u_i(k+j|k)\|^2_{R_i} \Big) \\ &\qquad + \|e_i\big(k+N_p\big)\|^2_{P_i} \end{aligned} \tag{27}$$

where P_i denotes the terminal penalty matrix. In this paper, based on empirical data, the terminal penalty matrix is selected as $P_i = 10Q_i$.

Iterative reinforcement learning predictive control scheme

The distributed control framework decomposes the global optimization problem into a series of local optimization problems. However, for nonlinear systems described by (19), solving the non-convex problem at each sampling instant is generally time-consuming. As the system state and control dimensions increase, the computational burden grows significantly. An iterative RLPC algorithm is proposed in this section to efficiently solve the non-convex problem, as shown in Figure 7.

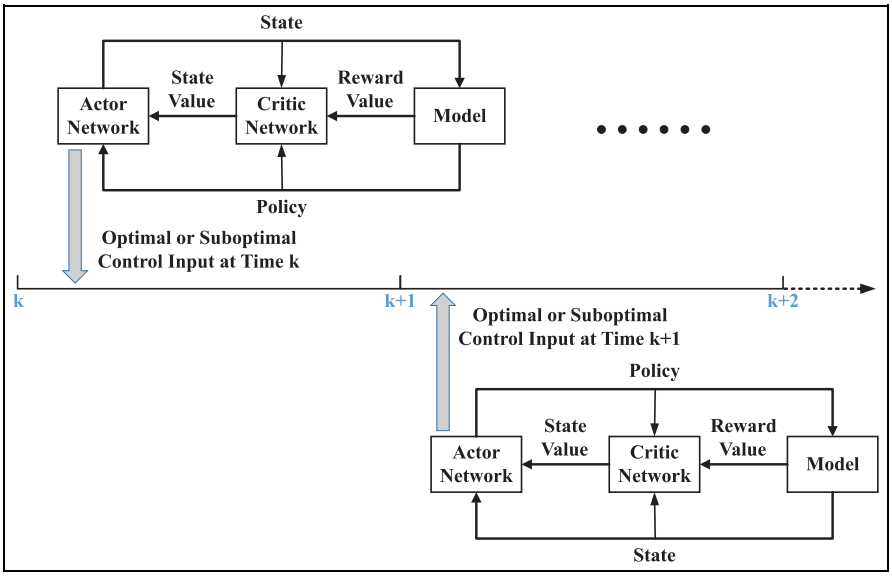

Figure 7. Diagram of the iterative RLPC algorithm.

Finite-horizon iterative RLPC algorithm

The iterative RLPC algorithm integrates policy iteration with the actor-critic architecture in RL, thereby replacing conventional numerical solvers (e.g. IP methods). At each sampling time, the algorithm solves a constrained optimization problem to obtain an optimal or sub-optimal control sequence over the prediction horizon. Specifically, within the prediction horizon $j \in [0, N_p - 1]$, N_p actor networks approximate the optimal control sequence $U_i^*(e_i(k+j|k))$, while N_p critic networks approximate the derivative $\lambda_i^*(e_i(k+j|k))$ of the optimal cost function $J_i^*(e_i(k+j|k))$ with respect to the error $e_i(k+j|k)$.

The stage cost is defined as follows:

$$r_i(e_i(k+j|k), u_i(k+j|k)) = \|e_i(k+j|k)\|_{Q_i}^2 + \|u_i(k+j|k)\|_{R_i}^2 . \tag{28}$$

Suppose that there exists a optimal control policy in Problem 1. According to the Bellman's Optimality Principle, the optimal cost function of the system satisfies the following discrete-time HJB equation[47,48]:

$$\begin{cases} J_i^*(e_i(k+j|k)) = \\ \min\limits_{\|\bar{U}^{-1}u_i\|_\infty \leqslant 1} \begin{pmatrix} r(e_i(k+j|k), u_i(k+j|k)) \\ + J_i^*(e_i(k+j+1|k)) \end{pmatrix}, \\ J_i^*\left(e_i\left(k+N_p|k\right)\right) = \|e_i\left(k+N_p|k\right)\|_{P_i}^2 . \end{cases} \tag{29}$$

The optimal control $u_i^*(e_i(k+j|k))$ satisfies:

$$u_i^*(e_i(k+j|k)) = \underset{\|\bar{U}^{-1}u_i\|_\infty \leqslant 1}{\operatorname{argmin}} \begin{pmatrix} r_i(e_i(k+j|k), u_i(k+j|k)) \\ + J_i^*(e_i(k+j+1|k)) \end{pmatrix}, \tag{30}$$

where $\bar{U} = \operatorname{diag}(T_{i,max}^d, \delta_{i,max})$ denotes the control input constraint matrix. It is worth noting that the constraint $\|\bar{U}^{-1}u_i\|_\infty \leqslant 1$ is equivalent to

$$\max\left(\frac{\left|T_i^d\right|}{T_{i,max}^d}, \frac{|\delta_i|}{\delta_{i,max}}\right) \leqslant 1,$$

which indicates that the control inputs satisfy the given constraints.

In each prediction horizon, due to the high computational burden of accurately solving the nonlinear HJB equation (29), a finite-horizon iterative RLPC algorithm is developed to approximate an optimal or suboptimal control policy. Within the prediction horizon $[k, k+N_p-1]$, the finite-horizon iterative RLPC algorithm is initialized with a cost function $J_i^0(e_i(k+j|k)) = 0$. Then, for $l = 0, 1, \cdots$ and $j \in [0, N_p - 1]$, the control input $u_i^l(e_i(k+j|k))$ is calculated as follows:

$$u_i^l(e_i(k+j|k)) = \underset{\|\bar{U}^{-1}u_i\|_\infty \leqslant 1}{\operatorname{argmin}} \begin{pmatrix} r_i(e_i(k+j|k), u_i(k+j|k)) \\ + J_i^l(e_i(k+j+1|k)) \end{pmatrix}. \tag{31}$$

The cost function $J_i^{l+1}(e_i(k+j|k))$ is updated as follows:

$$\begin{cases} J_i^{l+1}(e_i(k+j|k)) = r(e_i(k+j|k), u_i(k+j|k)) \\ \qquad + J_i^l(e_i(k+j+1|k)), \\ J_i^l(e_i(k+N_p)) = \|e_i(k+N_p)\|_{P_i}^2 . \end{cases} \tag{32}$$

Theorem 1: Let u_i^l and J_i^l be defined by (31) and (32). If $J_i^0(e_i(k+j|k)) = 0$, then as the number of iterations l approaches infinity, u_i^l converges to u_i^*, and J_i^l to J_i^*.

Theorem 1 proves the convergence of the iterative finite-horizon RLPC algorithm under the assumption of infinite iterations and the initial condition $J_i^0(e_i(k+j|k)) = 0$. However, this assumption is conservative in practice, particularly for convex problems where optimal solutions are typically attainable within finite iterations.

To reduce the computational burden of on-board systems while maintaining tracking performance, it is essential to determine a priori both the maximum number of iterations and the convergence threshold. Let the convergence threshold be denoted by $\varepsilon > 0$. According to Theorem 1, within the prediction horizon $j \in [0, N_p - 1]$, there exists an iteration number l such that:

$$\left|J_i^{l+1}(e_i(k+j|k)) - J_i^l(e_i(k+j|k))\right| \leqslant \varepsilon. \tag{33}$$

The convergence threshold ε quantifies the acceptable deviation between the suboptimal and optimal solutions. When the convergence criterion (33) is satisfied, the solution u_i^l is considered ε-optimal, denoted by $u_i^{\varepsilon *}$, with the corresponding cost $J_i^{\varepsilon *}$. Instead of seeking the global optimal solution to the non-convex problem, the algorithm aims for the ε-optimal solution $u_i^{\varepsilon *}$ that satisfies (33), thereby balancing optimality and computational efficiency.

The main procedures of the iterative RLPC algorithm is summarized as Algorithm 1.

Algorithm 1. Iterative RLPC algorithm

Step 1: Initialize: $l = 0$, $j = 0$, $J_i^0(e_i(k+j|k)) = 0$, $\varepsilon > 0$, and maximum number of iterations $l_{\max}$;
Step 2: Calculate $u_i^l(e_i(k+j|k))$ using (31);
Step 3: Generate the next state $e_i(k+j+1|k)$ using (19);
Step 4: Calculate $J_i^{l+1}(e_i(k+j|k))$ using (32);
Step 5: If $j = N_p - 1$, return; else, set $j = j + 1$ and go back to Step 2;
Step 6: If $l = l_{\max}$ or $\left|J_i^{l+1}(e_i(k+j|k)) - J_i^l(e_i(k+j|k))\right| < \varepsilon$, $\forall j \in [0, N_p - 1]$, return; else set $l = l + 1$, $j = 0$ and go back to Step 2.

Remark 1: In the iterative RLPC approach, the finite-horizon iterative RLPC algorithm, as described above, is used to obtain an ε-optimal control policy $u_i^{\varepsilon *}$ within each prediction horizon.

Efficient solving of iterative RLPC algorithm based on neural networks

To mitigate computational and storage burdens, neural networks are commonly employed to approximate value functions and policies in continuous state spaces. In this section, as an effective and practical realization of the iterative RLPC scheme, neural networks, in conjunction with kernel-based basis functions, are integrated into the iterative RLPC algorithm.

In this paper, radial basis function networks are chosen for both the actor and critic networks. The structure of the actor network is as follows[48]:

$$\begin{aligned} u_i^l(e_i(k+j|k)) &= \bar{U}\Gamma\left(\sum_{m=1}^{M_a} \omega_{a,l}^{[m]}(k+j|k)\psi^{[m]}(e_i(k+j|k))\right) \\ &= \bar{U}\Gamma\left(W_{a,l}(k+j|k)^T\Psi(e_i(k+j|k))\right) \end{aligned} \tag{34}$$

where $\Gamma(\cdot)$ is a monotonic odd function and $\|\Gamma(\cdot)\| \leqslant 1$. The first-order derivatives of the $\bar{U}$ and Γ are bounded. The term M_a denotes the number of center points in the hidden layer of the actor network, The term $\omega_{a,l}^{[m]}(k+j|k) \in R^2$ denotes the weight vector between the m^{th} center point and the output layer of the j^{th} actor network when the iteration number is l. The term $\psi^{[m]}(e_i(k+j|k))$ denotes the activation function of the m^{th} center point in the hidden layer of actor network, $W_{a,l}(k+j|k)$ denotes the weight matrix of the j^{th} actor network.

The structure of the critic network is defined as follows:

$$\begin{aligned} \lambda_i^l(e_i(k+j|k)) &= \sum_{m=1}^{M_c} \omega_{c,l}^{[m]}(k+j|k)\phi^{[m]}(e_i(k+j|k)) \\ &= W_{c,l}(k+j|k)^T\Phi(e_i(k+j|k)) \end{aligned} \tag{35}$$

where M_c is the number of center points of the hidden layer of critic network, $\omega_{c,l}^{[m]}(k+j|k) \in R^4$ denotes the weight vector between the m^{th} center point and the output layer of the j^{th} critic network when the iteration number is l. The term $\phi^{[m]}(e_i(k+j|k))$ denotes the activation function of the m^{th} center point in the hidden layer of critic network, $W_{c,l}(k+j|k)$ denotes the weight matrix of the j^{th} critic network.

In all neural networks, the parameters to be determined include the center points of the hidden layer and the weights from the hidden layer to the output layer.

In this paper, the center points of the hidden layer are randomly selected within the input variable range and remain constant. Thus, the parameter to be estimated is the weight connecting the hidden layer to the output layer.

In the iterative RLPC algorithm based on neural networks, the actor neural network and the critic neural network respectively perform policy updates and evaluations in Algorithm 1 through weight adjustments.[48]

(1) weight update of actor network

$$\begin{aligned} &W_{a,l}^{p+1}(k+j|k) \\ &= \left(\Psi(e_i(k+j|k))\Psi(e_i(k+j|k))^T\right)^{-1}\Psi(e_i(k+j|k)) \\ &\times\left(\Gamma^{-1}\left(\bar{U}^{-1}\left(\begin{array}{l}-\frac{1}{2R_i}\left(\frac{\partial e_i(k+j+1|k)}{\partial u_i^{l,p}(e_i(k+j|k))}\right)^T \\ \times W_{c,l}(k+j+1|k)^T \\ \times\Phi(e_i(k+j+1|k))\end{array}\right)\right)\right)^T, \\ &j\in[0,N_p-2] \end{aligned} \tag{36}$$

and

$$\begin{aligned} &W_{a,l}^{p+1}(k+j|k) \\ &= \left(\Psi(e_i(k+j|k))\Psi(e_i(k+j|k))^T\right)^{-1}\Psi(e_i(k+j|k)) \\ &\times\left(\Gamma^{-1}\left(\bar{U}^{-1}\left(\begin{array}{l}-\frac{1}{2R_i}\left(\frac{\partial e_i(k+j+1|k)}{\partial u_i^{l,p}(e_i(k+j|k))}\right)^T \\ \times 2P_i e_i(k+j+1|k))\end{array}\right)\right)\right)^T, \\ &j=N_p-1 \end{aligned} \tag{37}$$

where $W_{a,l}^{p+1}(k+j|k)$ denotes the weight matrix of the j^{th} actor network during the l^{th} policy evaluation and p^{th} policy update, $W_{c,l}(k+j|k)$ denotes the weight matrix of the j^{th} critic network in the l^{th} policy evaluation, $u_i^{l,p}(e_i(k+j|k))$ denotes the output of the j^{th} actor network during the l^{th} policy evaluation and the p^{th} policy update, $\Gamma^{-1}(\cdot)$ is the inverse function of $\Gamma(\cdot)$.

(2) weight update of critic network

$$\begin{aligned} &W_{c,l+1}(k+j|k) \\ &= (\Phi(e_i(k+j|k))\Phi(e_i(k+j|k))^T)^{-1}\Phi(e_i(k+j|k)) \\ &\quad\times(2Q_i e_i(k+j|k)+\left(\frac{\partial e_i(k+j+1|k)}{\partial e_i(k+j|k)}\right)^T \\ &\quad\times W_{c,l}(k+j+1|k)^T\Phi(e_i(k+j+1|k)))^T, \\ &\quad j\in[0,N_p-2] \end{aligned} \tag{38}$$

and

$$\begin{aligned} &W_{c,l+1}(k+j|k) \\ &= \left(\Phi(e_i(k+j|k))\Phi(e_i(k+j|k))^T\right)^{-1}\Phi(e_i(k+j|k)) \\ &\times\left(\begin{array}{l}2Q_i e_i(k+j|k)+\left(\frac{\partial e_i(k+j+1|k)}{\partial e_i(k+j|k)}\right)^T \\ \times 2P_i e_i(k+j+1|k)\end{array}\right)^T, \\ &j=N_p-1 \end{aligned} \tag{39}$$

where $W_{c,l+1}(k+j|k)$ denotes the weight matrix of the j^{th} critic network in the $l+1^{th}$ policy evaluation.

Lemma 2: In the iterative RLPC algorithm based on neural networks, the weights of the actor networks are iteratively updated according to (36) and (37), while the weights of the critic networks are updated according to (38) and (39). As the number of iterations l approaches infinity, u_i^l converges to u_i^*, J_i^l to J_i^*, and λ_i^l to λ_i^*.[47,48] Consequently, the output of the actor network corresponds to the optimal solution of Problem 1.

Proof: A note that J_i is continuously differentiable with respect to u_i. When the cost function is minimized, the optimal solution $u_i^*(k+j|k)$ should satisfy:

$$\frac{\partial J_i^*(e_i(k+j|k))}{\partial u_i^*(k+j|k)} = 0. \tag{40}$$

The derivative of $u_i^*(k+j|k)$ on the right side of the first equation in (29) can be obtained as follows:

$$\begin{aligned} &\frac{\partial\left(r_i\left(e_i(k+j|k),u_i^*(k+j|k)\right)+J_i^*(e_i(k+j+1|k))\right)}{\partial u_i^*(k+j|k)} \\ &= \frac{\partial r_i\left(e_i(k+j|k),u_i^*(k+j|k)\right)}{\partial u_i^*(k+j|k)} \\ &+\left(\frac{\partial e_i(k+j+1|k)}{\partial u_i^*(k+j|k)}\right)^T\frac{\partial J_i^*(e_i(k+j+1|k))}{\partial e_i(k+j+1|k)} \\ &= 0. \end{aligned} \tag{41}$$

Substituting (28) into (41) gives:

$$\begin{aligned} u_i^*(k+j|k) = &-\frac{1}{2R_i}\left(\frac{\partial e_i(k+j+1|k)}{\partial u_i^*(k+j|k)}\right)^T \\ &\times\lambda_i^*(e_i(k+j+1|k)) \end{aligned} \tag{42}$$

where

$$\lambda_i^*(e_i(k+j|k)) = \frac{\partial J_i^*(e_i(k+j|k))}{\partial e_i(k+j|k)}. \tag{43}$$

Let (34) be equal to the right side of (42), from which the weight update of the actor network, given by (36), follows. As observed from (36), the output $\lambda_i^l(e_i(k+j+1|k)) = W_{c,l}(k+j+1|k)^T\Phi(e_i(k+j+1|k))$ of the $j+1^{th}$ critic network is required for updating the weight of the j^{th} actor network. Therefore, (36) can be employed to update the j^{th} actor network for

$\forall j \in [0, N_p - 2]$. When $j = N_p - 1$, the following expression is obtained from (43):

$$\lambda_i^l(e_i(k+j+1|k)) = 2P_i e_i(k+j+1|k)$$

When $j = N_p - 1$, replacing $W_{c,l}(k+j+1|k)^T \Phi(e_i(k+j+1|k))$ in (36) with $2P_i e_i(k+j+1|k)$ yields the weight update rule (37) for the actor network at $N_p - 1$.

By differentiating $e_i(k+j|k)$ on the right side of (29), the following expression is obtained:

$$\begin{aligned}
&\lambda_i^*(e_i(k+j|k)) \\
&= \left(\frac{\partial\left(r_i\left(e_i(k+j|k), u_i^*(k+j|k)\right)\right)}{\partial e_i(k+j|k)}\right) \\
&\quad + \left(\frac{\partial\left(J_i^*(e_i(k+j+1|k))\right)}{\partial e_i(k+j|k)}\right) \\
&= \frac{\partial r_i\left(e_i(k+j|k), u_i^*(k+j|k)\right)}{\partial e_i(k+j|k)} \\
&\quad + \left(\frac{\partial u_i^*(k+j|k)}{\partial e_i(k+j|k)}\right)^T \frac{\partial r_i\left(e_i(k+j|k), u_i^*(k+j|k)\right)}{\partial u_i^*(k+j|k)} \\
&\quad + \left(\frac{\partial u_i^*(k+j|k)}{\partial e_i(k+j|k)}\right)^T \left(\frac{\partial e_i(k+j+1|k)}{\partial u_i^*(k+j|k)}\right)^T \\
&\quad \times \frac{\partial J_i^*(e_i(k+j+1|k))}{\partial e_i(k+j+1|k)} \\
&\quad + \left(\frac{\partial e_i(k+j+1|k)}{\partial e_i(k+j|k)}\right)^T \frac{\partial J_i^*(e_i(k+j+1|k))}{\partial e_i(k+j+1|k)}.
\end{aligned} \tag{44}$$

Substituting (41) into (44) gives:

$$\begin{aligned}
&\lambda_i^*(e_i(k+j|k)) \\
&= \frac{\partial r_i\left(e_i(k+j|k), u_i^*(k+j|k)\right)}{\partial e_i(k+j|k)} \\
&\quad + \left(\frac{\partial e_i(k+j+1|k)}{\partial e_i(k+j|k)}\right)^T \frac{\partial J_i^*(e_i(k+j+1|k))}{\partial e_i(k+j+1|k)} \\
&= 2Q_i e_i(k+j|k) \\
&\quad + \left(\frac{\partial e_i(k+j+1|k)}{\partial e_i(k+j|k)}\right)^T \lambda_i^*(e_i(k+j+1|k)).
\end{aligned} \tag{45}$$

To ensure that the output of the critic network approximates $\lambda_i^*(e_i(k+j|k))$, (35) is equated to the right side of (45). Accordingly, the weight update rule for the j^{th} critic network, where $j \in [0, N_p - 2]$, is given by (38). Similarly, the $(N_p - 1)^{th}$ critic network updates its weight according to (39). Moreover, the convergence of the iterative RLPC algorithm is guaranteed, that is, as the number of iterations l approaches infinity, u_i^l converges to u_i^*, J_i^l to J_i^*, and λ_i^l to λ_i^*. #.

Lemma 2 indicates that the output of the actor network converges to the optimal solution as the number of iterations l tends to infinity. However, due to the constraints of computational resources, infinite iterations are not feasible. Therefore, a balance between computational efficiency and performance must be achieved when selecting the maximum number of iterations and the convergence threshold ε. A smaller ε improves performance but increases computational burden, while a larger ε enhances computational efficiency at the expense of performance.

The maximum number of network weight updates for the critic within each prediction horizon, denoted as $l_{\max}$, and for the actor, denoted as $p_{\max}$. The terms $\Delta W_a(\varepsilon)$ and $\Delta W_c(\varepsilon)$ represent the convergence thresholds for the actor and critic network weights, respectively. Therefore, the termination condition of the iterative RLPC algorithm based on neural networks is defined as follows:

$$\|W_{c,l+1} - W_{c,l}\| \leqslant \Delta W_c(\varepsilon). \tag{46}$$

Defining actor$(k+j|k)$ and critic$(k+j|k)$ as the j^{th} actor network and critic network at time k, respectively, the ε-optimal control sequence is as follows:

$$\begin{aligned}
&U_i^{\varepsilon*}(k) = \\
&\left\{u_i^{\varepsilon*}(k|k), u_i^{\varepsilon*}(k+1|k), \cdots, u_i^{\varepsilon*}\left(k+N_p-1|k\right)\right\}.
\end{aligned} \tag{47}$$

Then, a ε-optimal solution to Problem 1 at time $k+1$ is:

$$\begin{aligned}
U_i^{\varepsilon*}(k+1) = \{&u_i^{\varepsilon*}(k+1|k+1), u_i^{\varepsilon*}(k+2|k+1), \cdots, \\
&u_i^{\varepsilon*}\left(k+N_p-1|k\right), 0\}.
\end{aligned} \tag{48}$$

Therefore, when solving Problem 1 at each time step, the weights of each actor and critic network are initialized according to the following equation:

$$\text{actor}(k+j|k+1) = \begin{cases} \text{actor}(k+j|k), \\ \quad j \in [1, N_p - 1] \\ zeros(M_a, 2), \\ \quad j = N_p \end{cases} \tag{49}$$

$$\text{critic}(k+j|k+1) = \begin{cases} \text{critic}(k+j|k), \\ \quad j \in [1, N_p - 1] \\ zeros(M_c, 2), \\ \quad j = N_p \end{cases} \tag{50}$$

where $zeros(M_a, 2)$ and $zeros(M_c, 2)$ denote the zero matrices of size $M_a \times 2$ and $M_c \times 2$, respectively. The main procedures of the iterative RLPC algorithm are summarized in Algorithm 2.

Remark 2: The parameters l_{max} and p_{max} are regarded as constants, independent of the prediction horizon N_p. As a result, the computational burden is primarily determined by the matrix dimension, which is

Algorithm 2. Iterative RLPC algorithm based on neural networks

1: **Input**: The maximum iteration numbers $l_{\max}$ and $p_{\max}$; the weight convergence thresholds $\Delta W_a(\varepsilon)$ and $\Delta W_c(\varepsilon)$; the initial states of the i^{th} vehicle.
2: **Output**: ε-optimal control input $u_i^{\varepsilon*}(k|k)$.
3: **Initialization**: Based on equations (49) and (50), initialize the weight matrices for the networks actor($k|k$), actor($k+1|k$), $\cdots$, actor($k+N_p-1|k$), critic($k|k$), critic($k+1|k$), $\cdots$, critic($k+N_p-1|k$). Set $l=0$.
4: **repeat**
5: **for** $j=0,1,\cdots,N_p-1$ **do**
6: $p=0$;
7: **repeat**
8: Calculate $u_i^l(e_i(k+j|k))$ using formula (34);
9: Calculate the next time step's $e_i(k+j+1|k)$ using (19);
10: Update the weights of the actor neural network using formulas (36) and (37);
11: Set $p=p+1$;
12: **until** $p=p_{\max}$ or $\|W_{al}^{p+1}(k+j|k)-W_{al}^{p}(k+j|k)\| \leqslant \Delta W_a(\varepsilon)$;
13: Calculate $u_i^l(e_i(k+j|k))$ using formula (34);
14: Calculate the next time step's $e_i(k+j+1|k)$ using (19);
15: Update the weights of the critic neural network using formulas (38) and (39);
16: **end for**
17: $l=l+1$;
18: **until** $l=l_{\max}$ or $\|W_{c,l}(k+j|k)-W_{c,l-1}(k+j|k)\| \leqslant \Delta W_c(\varepsilon), \forall j \in [0, N_p-1]$.
19: Calculate the output of the actor network actor($k|k$) using formula (34), which corresponds to the ε-optimal control input $u_i^{\varepsilon*}(k|k)$. Apply $u_i^{\varepsilon*}(k|k)$ to the i^{th} vehicle, then set $k=k+1$,update Problem 1, and return to the initialization step for re-solving.

approximately equal to the system output dimension n. The computational complexity of the iterative RLPC algorithm is $O(n^2N_p)$. In contrast, NMPC typically employs polynomial-time algorithms, such as the interior point (IP) algorithm, resulting in a complexity of $O(n^{3.5}N_p^2)$.[48] Therefore, the proposed iterative RLPC algorithm demonstrates a significant advantage in computational efficiency over NMPC.

Remark 3: A terminal equality constraint, added to the cost function as a soft constraint, is utilized to ensure the stability of the proposed algorithm. In Algorithm 2, the ε-optimal control sequence is not entirely applied to vehicles, only its first element is applied. In the process of rolling optimization, each network is initialized into a feasible solution according to (49) and (50).

Remark 4: Unlike static neural network-based model predictive control algorithms that rely on offline training, the proposed iterative RLPC algorithm updates its parameters online, reducing the reliance on large offline datasets and improving generalization.

Simulation

This section validates the effectiveness of the proposed distributed iterative RLPC scheme for vehicle platoons with coupled lateral-longitudinal dynamics through MATLAB/Simulink-TruckSim co-simulation. Based on the technical specifications of the DF SKYLINE KJ1V commercial vehicle, a high-fidelity dual-axle, fully loaded vehicle model is developed in TruckSim. Key dynamic parameters are listed in Tables 2 and 3. The configuration includes a $270\,kW$ peak power engine with an AT automatic transmission, and a steering gear ratio of $\beta = 25$: 1. Environmental conditions follow the International Standard Atmosphere (ISA) model, with air density $\rho = 1.225\ \mathrm{kg/m^3}$.

Within the iterative RLPC algorithm, the activation function vectors $\Psi(e_i)$ and $\Phi(e_i)$ for each hidden layer of actor and critic neural networks are both specified as Gaussian radial basis functions:

$$\begin{aligned}\Psi(e_i) &= \left(\exp^{-\|e_i-e_i^1\|^2\kappa^2}; \exp^{-\|e_i-e_i^2\|^2\kappa^2}; \cdots; \exp^{-\|e_i-e_i^{M_a}\|^2\kappa^2}\right)\\ \Phi(e_i) &= \left(\exp^{-\|e_i-e_i^1\|^2\kappa^2}; \exp^{-\|e_i-e_i^2\|^2\kappa^2}; \cdots; \exp^{-\|e_i-e_i^{M_c}\|^2\kappa^2}\right)\end{aligned} \tag{51}$$

where $\kappa = 1.1$, and the number of center points in each hidden layer of the actor and critic networks is set to $M_a = M_c = 5$. The terms $\left(e_i^1; e_i^2; \cdots; e_i^{M_a}\right)$ and $\left(e_i^1; e_i^2; \cdots; e_i^{M_c}\right)$ denote the center points in the hidden layers of the actor and critic networks, respectively. Each center point is a four-dimensional vector matching the input dimensions (velocity error, longitudinal position error, lateral position error, and heading angle error), with components randomly sampled from $[-3,3]$, $[-3,3]$, $[-1,1]$, and $[-0.1,0.1]$, respectively. The function $\Gamma(\cdot)$ denotes the hyperbolic tangent function, that is,

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{52}$$

The initial weights for each actor network and critic neural network, denoted $W_{a,0}^0$ and $W_{c,0}$, are randomly selected from the range $[-0.5, 0.5]$. The maximum number of iterations is set to $l_{\max} = 4$ and $p_{\max} = 4$ in Case 1, and $l_{\max} = 8$ and $p_{\max} = 8$ in Case 2. The weight

Table 2. Parameters of the i^{th} vehicle.

Parameter	Value	Parameter	Value
m_i	18000 kg	I_i^z	130421.8 $kg \cdot m^2$
a_j	3.5 m	b_j	1.5 m
J_i^f	24 $kg \cdot m^2$	J_i^f	48 $kg \cdot m^2$
R_e	0.51 m		

Table 3. Parameters of the Magic Formula.

Tire force	B	C	D	E
F_i^{xf}	8.434	1.813	21370	0.6593
F_i^{xr}	8.434	1.813	42020	0.6593
F_i^{yf}	5.228	2.42	21430	0.9869
F_i^{yr}	5.228	2.42	42140	0.9869

convergence threshold is $\Delta W_a(\varepsilon) = \Delta W_c(\varepsilon) = 10^{-2}$. Other controller parameters are shown in Table 4.

Case 1: Iterative RLPC algorithm ($N_p = 3$)

At the initial time, the positions of the vehicles are given in Table 5. The road adhesion coefficient is set to 0.85. The leading vehicle begins with an initial velocity of 20 m/s, maintaining a constant velocity, then decelerating to 15 m/s, and finally sustaining a constant velocity. All following vehicles are initialized at 21 m/s, with zero lateral displacement and heading angle errors. The vehicle platoon travels along a straight road, enters a curve, and then returns to a straight road. The maximum curvature of the road is 0.01, with the curvature profile shown in Figure 8.

In Figure 9(a) to (c), the longitudinal velocity, position, and trajectory of the vehicle platoon are shown. In Figure 10(a) to (c), the longitudinal position error, heading angle error, and lateral position error of the following vehicles are shown.

Table 4. Parameters of the controller.

Parameter	Value
Sampling time T_s	0.01 s
Weight matrix Q_i	$10^5 \times diag(2, 70, 40, 40)$
Weight matrix R_i	$diag(0.06, 3 \times 10^6)$
$T_{i,\min}^d, T_{i,\max}^d$	$-10000,\ 10000\ (N \cdot m)$
$\delta_{i,\min}, \delta_{i,\max}$	$-0.1,\ 0.1\ (rad)$
Fixed spacing d_0	11 m

Table 5. Initial vehicle position information.

Vehicle number	Initial position
Leading vehicle	$(64, 0)$
Following vehicle 1	$(47, 0)$
Following vehicle 2	$(30, 0)$
Following vehicle 3	$(13, 0)$

(1) Longitudinal tracking: In Figure 9(a), the simulation results demonstrate that following vehicles in the platoon quickly track the leading vehicle's velocity and maintain consistency. During curved driving, longitudinal velocity is influenced by lateral velocity due to coupled lateral and longitudinal dynamics. As vehicles enter the curve at different times, velocity disturbances propagate through the platoon. This requires following vehicles to simultaneously handle both their own coupled dynamics and incoming disturbances, leading to bounded fluctuations in longitudinal velocity. Figure 10(a) confirms that the proposed iterative RLPC algorithm maintains the expected inter-vehicle spacing, with the longitudinal position errors of following vehicles asymptotically converging to zero. Notably, during curved driving, coupled dynamics and disturbance propagation cause bounded fluctuations in inter-vehicle spacing around the desired value.

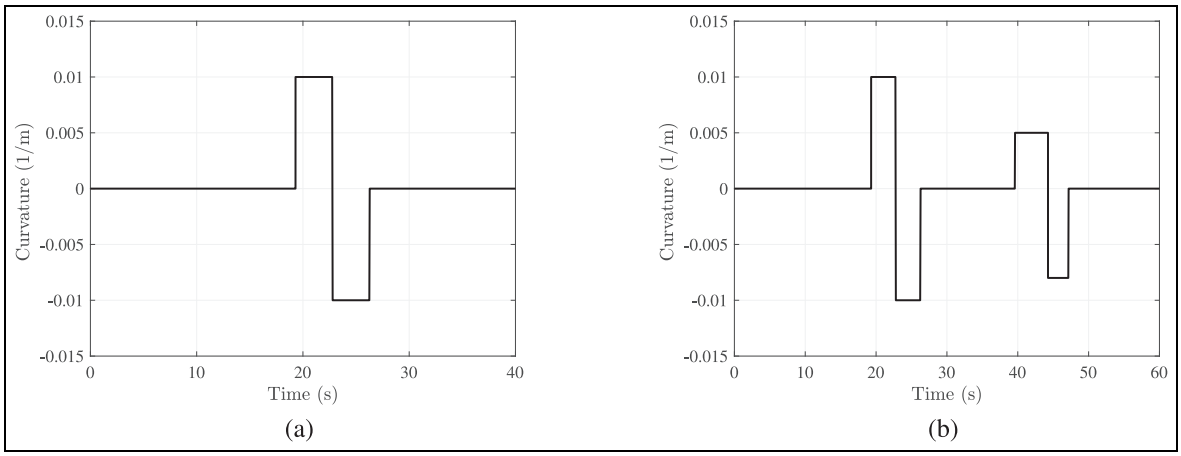

Figure 8. The road curvature: (a) Case 1 and (b) Case 2.

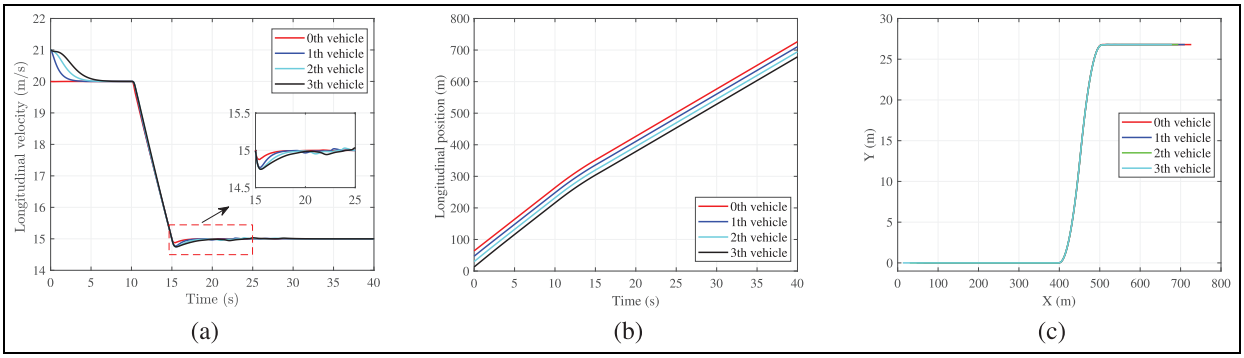

Figure 9. Iterative RLPC with $N_p = 3$: (a) longitudinal velocity, (b) longitudinal position, and (c) trajectory.

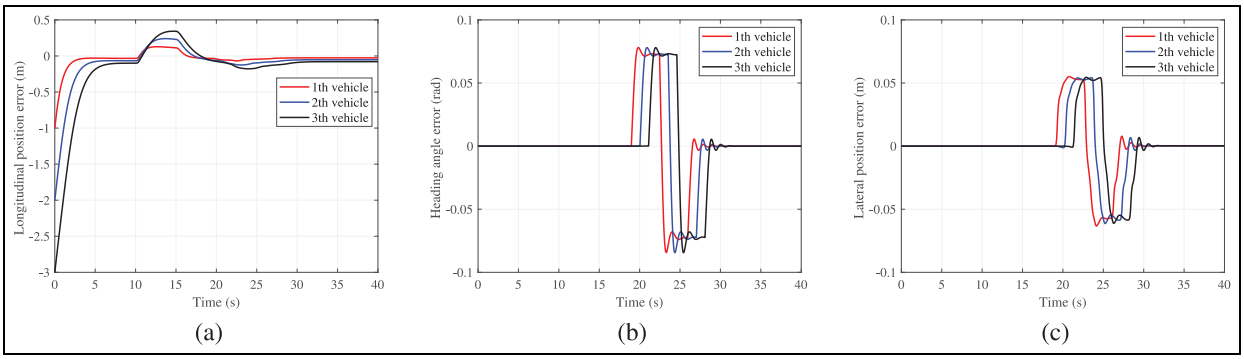

Figure 10. Iterative RLPC with $N_p = 3$: (a) longitudinal position error, (b) heading angle error, and (c) lateral position error.

Table 6. Comparison of computational time for the iterative RLPC algorithm.

Computational time	Vehicle 1	Vehicle 2	Vehicle 3
Average	0.0041 s	0.0039 s	0.0039 s
Maximum	0.0064 s	0.0059 s	0.0059 s

(2) Safety and trajectory consistency: In Figure 9(b) and (c), it is shown that, under the proposed controller, the vehicles' trajectories remain consistent and no collisions occur within the platoon.

According to the "Technical Standard of Highway Engineering," the width of a highway lane is 3.75 m and that of the emergency lane is 3.5 m. With truck widths ranging from 2 to 2.4 m, the maximum allowable lateral position error is 0.675 m to avoid crossing lane boundaries. As shown in Figure 10(b) and (c), the heading angle error and lateral position error of the following vehicles remain zero when the platoon is on a straight road. On curved roads, these errors change but remain within the allowable range, confirming that the proposed controller ensures the safety of the vehicle platoon.

In Figure 11(a) and (b), it is indicated that the torque and front wheel steering angle of the following vehicles

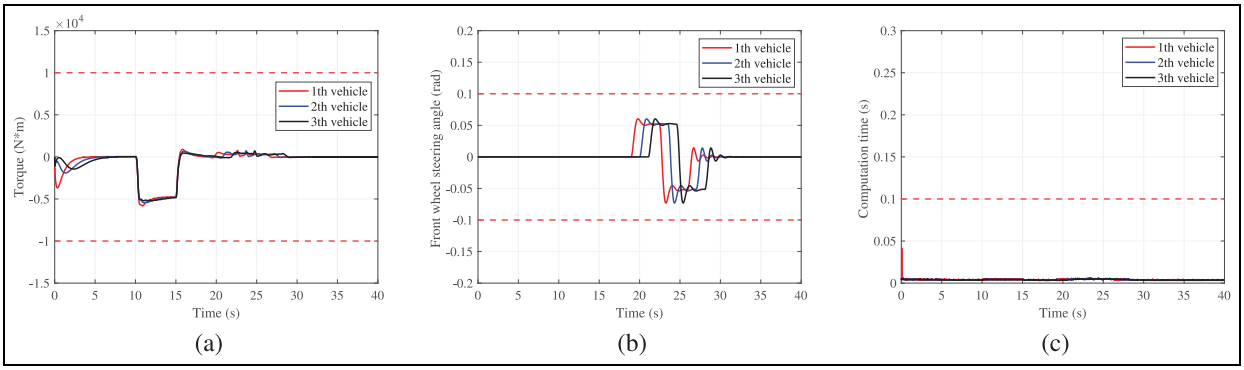

Figure 11. Iterative RLPC with $N_p = 3$: (a) torque, (b) front wheel steering angle, and (c) computational time.

Table 7. The design velocity and the corresponding minimum radius of curvature for highways.

Design velocity (km/h)	100	80	60
Minimum radius of curvature (m)	700	400	200

in the platoon satisfy the actuator constraints. Figure 11(c) presents the computational time of the following vehicles under the iterative RLPC algorithm. Table 6 provides both the average and maximum computational times for each following vehicle, all of which are below the 0.01-s sampling threshold.

Case 2: Comparison of iterative RLPC and NMPC algorithms ($N_p = 7$)

According to the "Technical Standard of Highway Engineering," the corresponding relationship between velocity and road curvature is shown in Table 7.

In Figures 12 to 15, a comparative analysis of the simulation results for the coupled controller under two different algorithms is presented. In Figures 12 and 14, the simulation results based on the iterative RLPC algorithm are shown, and in Figures 13 and 15, the simulation results obtained by utilizing the conventional NMPC algorithm are depicted.

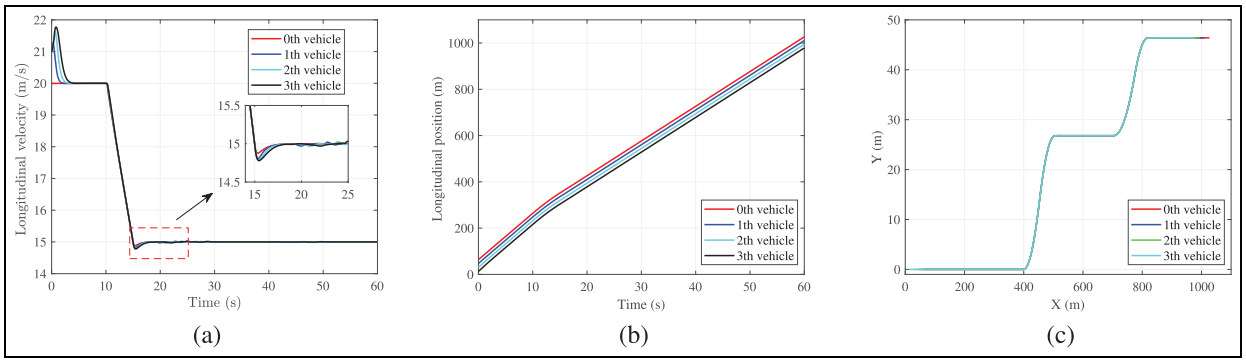

Figure 12. Iterative RLPC with $N_p = 7$: (a) longitudinal velocity, (b) longitudinal position, and (c) trajectory.

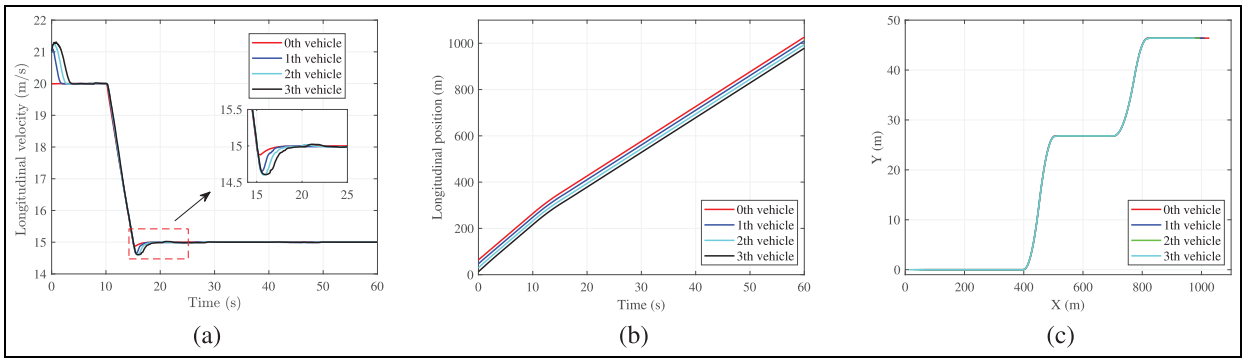

Figure 13. NMPC with $N_p = 7$: (a) longitudinal velocity, (b) longitudinal position, and (c) trajectory.

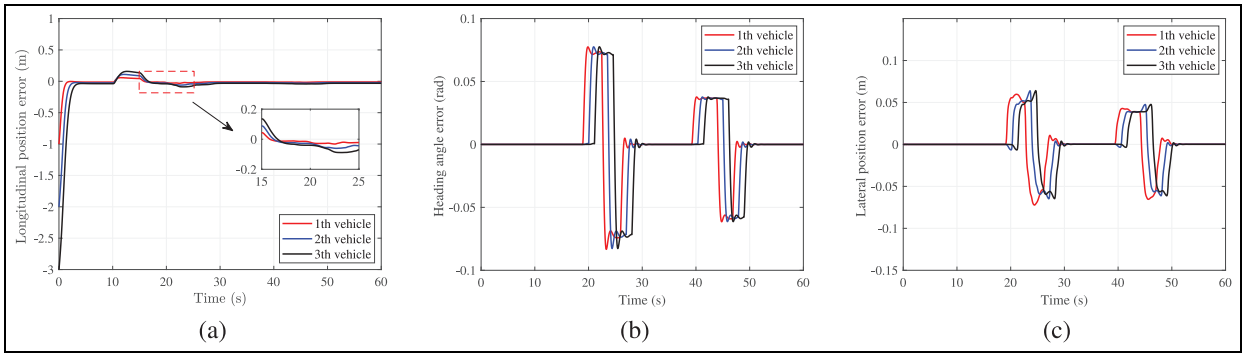

Figure 14. Iterative RLPC with $N_p = 7$: (a) longitudinal position error, (b) heading angle error, and (c) lateral position error.

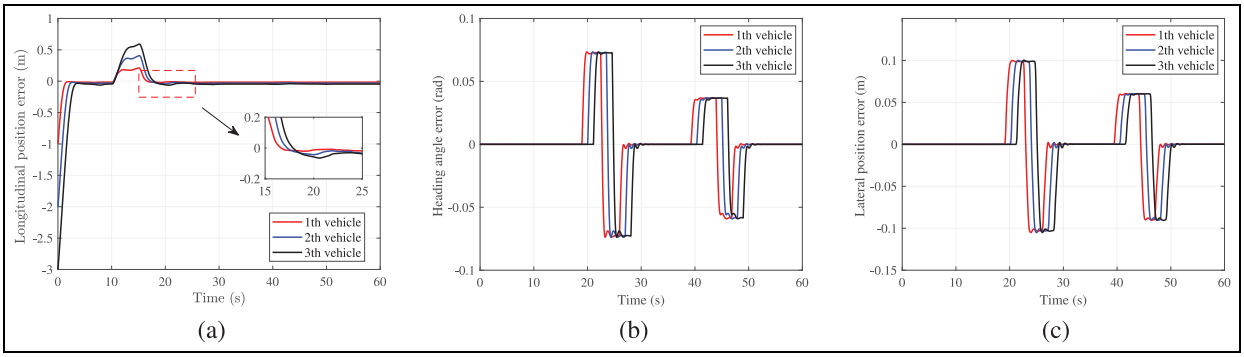

Figure 15. NMPC with $N_p = 7$: (a) longitudinal position error, (b) heading angle error, and (c) lateral position error.

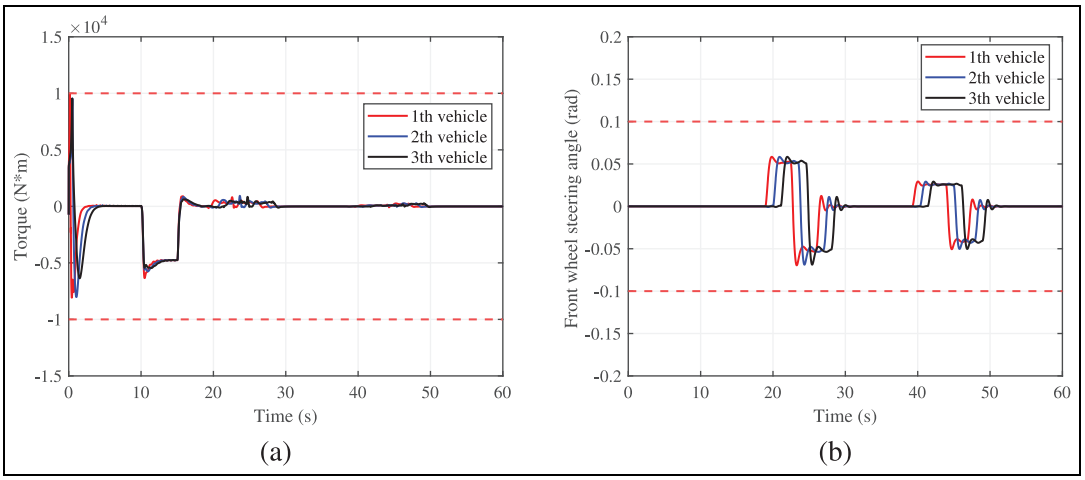

Figure 16. Iterative RLPC with $N_p = 7$: (a) torque and (b) front wheel steering angle.

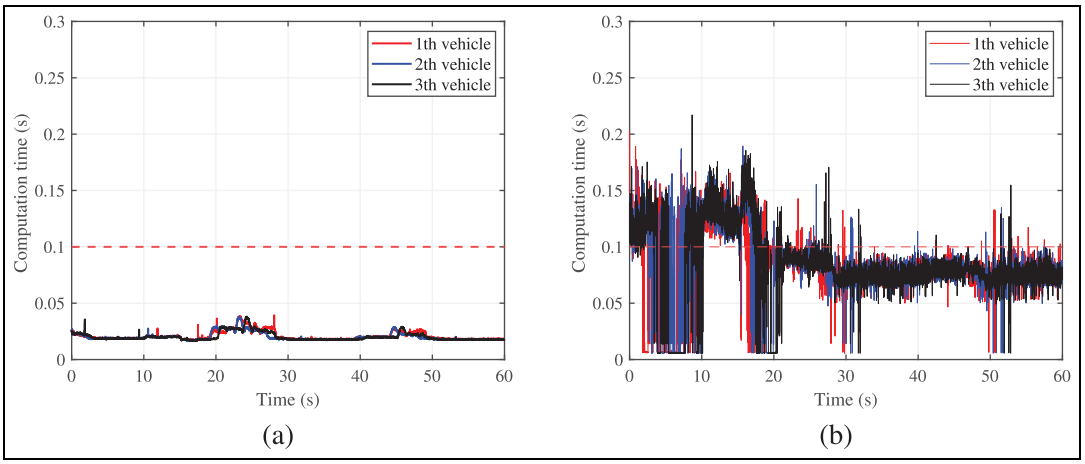

Figure 17. Comparison of computational time with $N_p = 7$: (a) iterative RLPC and (b) NMPC.

As shown in subfigure (a) of Figures 12 to 15, the iterative RLPC algorithm demonstrates superior performance compared to the NMPC algorithm in both longitudinal velocity tracking and inter-vehicle spacing maintenance, despite oscillations observed in both algorithms due to coupled dynamics and disturbance propagation. Subfigures (b) and (c) of Figures 12 and 13 further confirm that both algorithms effectively prevent collisions within the platoon. Moreover, subfigures (b) and (c) of Figures 14 and 15 indicate that the iterative RLPC algorithm achieves higher lateral tracking accuracy, with a maximum lateral error of only 0.07 m, well below the limit specified in the "Technical Standard of Highway Engineering."

Table 8. Comparison of average computational time between the two algorithms.

Algorithm	Vehicle 1	Vehicle 2	Vehicle 3
Iterative RLPC	0.0206 s	0.0200 s	0.0200 s
NMPC	0.0821 s	0.0827 s	0.0833 s

As shown in Figure 16, the iterative RLPC algorithm ensures that the torque and front wheel steering angle of the following vehicles in the platoon satisfy the actuator constraints. As shown in Figure 17 and Table 8, the RLPC algorithm outperforms the NMPC algorithm in both average and maximum computational time. Notably, the NMPC algorithm, solved via MATLAB's *fmincon*, fails to find feasible solutions within the prescribed 0.01-s sampling time, whereas the iterative RLPC algorithm consistently obtains feasible solutions within this time frame.

In summary, the proposed iterative RLPC algorithm not only reduces the computational burden compared to the conventional NMPC algorithm, but also exhibits superior tracking performance.

Conclusion

A coordinated control architecture was proposed for high-speed, fully loaded vehicle platoons, integrating longitudinal tracking with lateral lane-keeping. A five-degree-of-freedom vehicle dynamics model, capturing coupled longitudinal-lateral characteristics and tire nonlinearities, was incorporated with a lane-keeping model to form an integrated platoon model. Subsequently, a synchronous distributed model predictive control strategy was developed, using an iterative RLPC algorithm to solve non-convex constrained optimization problems in real time. Co-simulation using MATLAB/Simulink and TruckSim demonstrated that the proposed strategy achieved accurate longitudinal and lateral control with a lower computational burden compared to conventional NMPC algorithms. Prior to real-world implementation, the nominal model parameters must be calibrated against actual vehicle data. Future work will focus on the development of lightweight neural networks to further reduce computational burden and enhance robustness against model parameter uncertainties.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (No. 62473167) and the Natural Science Foundation of Jilin Province (No. 20240402079GH).

ORCID iDs

Shuyou Yu https://orcid.org/0000-0002-3258-6494
Zepeng Liu https://orcid.org/0009-0006-9515-2040

References

1. Chen T, Cai Y, Chen L, et al. Trajectory and velocity planning method of emergency rescue vehicle based on segmented three-dimensional quartic Bezier curve. *IEEE Trans Intell Transp Syst* 2023; 24(3): 3461–3475.
2. Iancu DT and Florea AM. An improved vehicle trajectory prediction model based on video generation. *Stud Inform Control* 2023; 32(1): 25–36.
3. Huang S, Ren W and Chan SC. Design and performance evaluation of mixed manual and automated control traffic. *IEEE Trans Syst Man Cybern A Syst Hum* 2000; 30(6): 661–673.
4. Hung NV, Luu DL, Dong NS, et al. Reducing time headway for cooperative vehicle following in platoon via information flow topology. *J Control Eng Appl Inform* 2024; 26(2): 77–87.
5. Gülden B and Emirler MT. Investigation of different communication topologies for cooperative adaptive cruise control systems. *Int J Automot Sci Technol* 2024; 8(1): 150–158.
6. Iancu DT, Nan M, Ghita SA, et al. Trajectory prediction using video generation in autonomous driving. *Stud Inform Control* 2022; 31(1): 37–48.
7. Yu Z, Tang X, Lv X, et al. Predictive control via augmented Lagrange function for autonomous vehicles trajectory tracking with dynamic quantization. *J Control Eng Appl Inform* 2024; 26(2): 14–24.
8. Viet Hung N, Luu DL, Pham QT, et al. Comparative analysis of different spacing policies for longitudinal control in vehicle platooning. *Proc IMechE, Part D: J Automobile Engineering*. Epub ahead of print 28 August 2024. DOI: 10.1177/09544070241273985.
9. Luu L, Phan TL, Pham HT, et al. Stability of adaptive cruise control of automated vehicle platoon under constant time headway policy. *Int J Automot Sci Technol* 2024; 8(3): 397–403.
10. Tiganasu A, Lazar C, Caruntu CF, et al. Comparative analysis of advanced cooperative adaptive cruise control algorithms for vehicular cyber physical systems. *J Control Eng Appl Inform* 2021; 23(1): 82–92.
11. Li L, Yao X and Shi S. An analysis on the effects of tire nonlinearity on autonomous platoon stability. *Automot Eng* 2019; 41(9): 1065–1072.
12. Koushkbaghi S, Safi M, Amani AM, et al. Byzantine-resilient second-order consensus in networked systems. *IEEE Trans Cybern* 2024; 54(9): 4915–4927.
13. Hao H and Barooah P. Stability and robustness of large platoons of vehicles with double-integrator models and nearest neighbor interaction. *Int J Robust Nonlinear Control* 2013; 23(18): 2097–2122.
14. Wang Q, Guo G and Cai B. Distributed receding horizon control for fuel-efficient and safe vehicle platooning. *Sci China Technol Sci* 2016; 59(12): 1953–1962.

15. Gao F, Hu X, Li SE, et al. Distributed adaptive sliding mode control of vehicular platoon with uncertain interaction topology. *IEEE Trans Ind Electron* 2018; 65(8): 6352–6361.
16. Guo G and Yuan WL. Bidirectional platoon control of Arduino cars with actuator saturation and time-varying delay. *J Control Eng Appl Inform* 2017; 19(1): 37–48.
17. Mercorelli P. Using fuzzy PD controllers for soft motions in a car-like robot. *Adv Sci Technol Eng Syst J* 2018; 3(6): 380–390.
18. Zhang D, Li K and Wang J. A curving ACC system with coordination control of longitudinal car-following and lateral stability. *Veh Syst Dyn* 2012; 50(7): 1085–1102.
19. Lan J, Zhao D and Tian D. Robust cooperative adaptive cruise control of vehicles on banked and curved roads with sensor bias. In: *2020 American control conference (ACC)*, Denver, CO, USA, 1–3 July 2020, pp.2276–2281. New York: IEEE.
20. Zhao J, Li R, Zhang G, et al. Supervisor-based hierarchical adaptive MPC for yaw stabilization of FWID-EVs under extreme conditions. *IEEE Trans Intell Transp Syst* 2025; 26(2): 1852–1863.
21. Chen T, Cai Y, Chen L, et al. Sideslip angle fusion estimation method of three-axis autonomous vehicle based on composite model and adaptive cubature kalman filter. *IEEE Trans Transp Electrif* 2023; 10(1): 316–330.
22. Li Z, Zhou Y, Zhang Y, et al. Enhancing vehicular platoon stability in the presence of communication cyberattacks: a reliable longitudinal cooperative control strategy. *Transp Res Part C Emerg Technol* 2024; 163: 104660.
23. Wu Y, Li SE, Cortés J, et al. Distributed sliding mode control for nonlinear heterogeneous platoon systems with positive definite topologies. *IEEE Trans Control Syst Technol* 2019; 28(4): 1272–1283.
24. Lui DG, Petrillo A and Santini S. Distributed model reference adaptive containment control of heterogeneous multi-agent systems with unknown uncertainties and directed topologies. *J Franklin Inst* 2021; 358(1): 737–756.
25. Chen T, Xu X, Cai Y, et al. QPSOMPC-based chassis coordination control of 6WIDAGV for vehicle stability and trajectory tracking. *J Franklin Inst* 2025; 362(2): 107458.
26. Zhao J, Li R, Zheng X, et al. Constrained fractional-order model predictive control for robust path following of FWID-AGVs with asymptotic prescribed performance. *IEEE Trans Veh Technol* 2025; 74: 2692–2705.
27. Hedman M and Mercorelli P. FFTSMC with optimal reference trajectory generated by MPC in robust robotino motion planning with saturating inputs. In: *2021 American control conference (ACC)*, New Orleans, LA, USA, 25–28 May 2021, pp.1470–1477. New York: IEEE.
28. Li H, Shi Y and Yan W. Distributed receding horizon control of constrained nonlinear vehicle formations with guaranteed γ-gain stability. *Automatica* 2016; 68: 148–154.
29. Dunbar WB and Caveney DS. Distributed receding horizon control of vehicle platoons: stability and string stability. *IEEE Trans Automat Contr* 2011; 57(3): 620–633.
30. Zhou Y, Wang M and Ahn S. Distributed model predictive control approach for cooperative car-following with guaranteed local and string stability. *Transp Res B Methodol* 2019; 128: 69–86.
31. Xu Y, Shi Y, Tong X, et al. A multi-agent reinforcement learning based control method for connected and autonomous vehicles in a mixed platoon. *IEEE Trans Veh Technol* 2024; 73: 16160–16172.
32. Hua M, Chen D, Jiang K, et al. Communication-efficient MARL for platoon stability and energy-efficiency co-optimization in cooperative adaptive cruise control of CAVs. *IEEE Trans Veh Technol* 2025; 74: 6076–6087.
33. Li M, Wang B, Wang S, et al. Serial distributed reinforcement learning for enhanced multi-objective platoon control in curved road coordinates. *Expert Syst Appl* 2025; 269: 126493.
34. Cai Y, Zhan L, Sun X, et al. Research on obstacle avoidance strategy of automated heavy vehicle platoon in high-speed scenarios. *Proc IMechE, Part D: J Automobile Engineering*. Epub ahead of print 16 September 2024. DOI: 10.1177/09544070241276062.
35. Chen J, Wu X, Lv Z, et al. Collaborative control of vehicle platoon based on deep reinforcement learning. *IEEE Trans Veh Technol* 2024; 73: 14399–14414.
36. Chen D, Zhang K, Wang Y, et al. Communication-efficient decentralized multi-agent reinforcement learning for cooperative adaptive cruise control. *IEEE Trans Intell Veh* 2024; 9: 6436–6449.
37. Wang F, Wang X and Sun S. A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization. *Inf Sci* 2022; 602: 298–312.
38. D'Alfonso L, Giannini F, Franzè G, et al. Autonomous vehicle platoons in urban road networks: a joint distributed reinforcement learning and model predictive control approach. *IEEE CAA J Automatica Sin* 2024; 11(1): 141–156.
39. Shen X and Borrelli F. Reinforcement learning and distributed model predictive control for conflict resolution in highly constrained spaces. In: *2023 IEEE intelligent vehicles symposium (IV)*, Anchorage, AK, USA, 4–7 June 2023, pp.1–6. New York: IEEE.
40. Zhang X, Pan W, Li C, et al. Toward scalable multirobot control: fast policy learning in distributed MPC. *IEEE Trans Robot* 2025; 41: 1491–1512.
41. Wang X. *Coupling bifurcation of vehicle driving torque and steering angle and solution of driving stability region*. PhD Thesis, Jilin University, China, 2014.
42. Pacejka HB and Bakker E. The magic formula tyre model. *Veh Syst Dyn* 1992; 21(S1): 1–18.
43. Swaroop D and Hedrick JK. Constant spacing strategies for platooning in automated highway systems. *J Dyn Syst Meas Control* 1999; 121(3): 462–470.
44. Pang Y, Zhu X, He T, et al. AI-assisted self-powered vehicle-road integrated electronics for intelligent transportation collaborative perception. *Adv Mater* 2024; 36(36): 2404763.
45. Chen H and Allgöwer F. A quasi-infinite horizon nonlinear predictive control scheme for stable systems: application to a CSTR. *IFAC Proc Vol* 1997; 30(9): 529–534.
46. Chen H. *Model predictive control*. Beijing: Science Press, 2013.
47. Wang D, Xin P, Zhao M, et al. Intelligent optimal control of constrained nonlinear systems via receding-horizon heuristic dynamic programming. *IEEE Trans Syst Man Cybern Syst* 2023; 54(1): 287–299.
48. Xu X, Chen H, Lian C, et al. Learning-based predictive control for discrete-time nonlinear systems with stochastic disturbances. *IEEE Trans Neural Netw Learn Syst* 2018; 29(12): 6202–6213.