2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Receding Horizon Reinforcement Learning with Autoregressive Model for Motion Control of Autonomous Vehicles

Xin Yin, Haotian Cao, Xinglong Zhang, Qingwen Ma, Xin Xu, Haibin Xie

Abstract— **This paper presents a model-based reinforcement learning (MBRL) approach with a receding horizon mechanism to optimize the lateral trajectory-tracking performance of autonomous vehicles (AVs). Accurate modeling of complex vehicle dynamics and adaptation to dynamic environments with limited data pose significant challenges for MBRL in AV control. To address these challenges, we propose sample-efficient algorithms that leverage autoregressive modeling to adapt from limited data while managing complex vehicle dynamics. Unlike traditional methods reliant on fixed models, our approach uses the temporal reasoning of autoregressive (AR) models to compensate for the residual dynamics, which effectively approximates the local effects of nonlinearities and disturbances. Integrated with real-time sensor data, the residual generation model is continuously refined via incremental learning in a closed-loop framework, enhancing adaptability. This architecture, combining physical modeling with data-driven residuals, maintains interpretability and improves responsiveness in complex scenarios. CarSim simulations demonstrate superior performance over other state-of-the-art learning-based predictive controllers and classical methods for AV lateral control. Real-world validation on a HongQi electric vehicle (HQEV) confirms the algorithm's effectiveness, showing significant improvements over classical model predictive control (MPC). This approach holds substantial potential for advanced driver-assistance systems (ADAS) and fully autonomous driving, enabling precise control under diverse conditions.**

I. Introduction

Motion control is a pivotal subsystem for autonomous vehicles (AVs), facilitating precise trajectory tracking and safe collision avoidance. However, significant challenges arise from inaccuracies in modeling AV dynamics and environmental uncertainties, particularly on unstructured roads characterized by diverse and unpredictable conditions. Effective model-based control methods in such scenarios require the accurate prediction of vehicle dynamics under varying environmental factors, including slopes and surface types. Furthermore, maintaining both the stability and interpretability of the predictive model is essential. Incorporating advanced predictive reasoning into a stable and transparent dynamics model can significantly enhance control performance under uncertainty. This capability facilitates reliable autonomous navigation in complex terrains, such as mining sites.

This work was supported in part by the National Natural Science Foundation of China under Grants 62533021, T2521006, and U24A20279.

Xin Yin, Haotian Cao, Xinglong Zhang, Qingwen Ma, Xin Xu, and with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China. ocetyx1@163.com, caohaotian@nudt.edu.cn, zhangxinglong18@nudt.edu.cn, maqingwen@mail.nwpu.edu.cn, xuxin@nudt.edu.cn, xiehaibin@nudt.edu.cn , (Corresponding author: Haotian Cao)

Achieving high accuracy in vehicle lateral dynamics modeling without compromising computational efficiency remains a significant challenge for AVs, especially on unstructured roads, where substantial modeling uncertainties exist. In vehicle lateral dynamics, models with high degrees of freedom (DOF) have been proposed [1], [2], [3]. Although higher-DOF models provide improved accuracy, their increased computational complexity hinders real-time control efficiency. The widely adopted 2-DOF lateral vehicle dynamics model, valued for its simplicity and computational efficiency, is limited in its ability to capture complex environmental factors (e.g., varying road conditions and slopes) or vehicle roll dynamics. Integrating the 2-DOF model with advanced algorithms capable of temporal reasoning can significantly enhance dynamic responsiveness in complex environments, maintaining model interpretability and computational efficiency.

Traditional lateral control methods for AVs, such as Stanley [4], sliding model control (SMC) [5], linear quadratic regulator (LQR) [6], and model predictive control (MPC) [7], lack the capability to learn from environmental uncertainties. Reinforcement learning (RL) provides a powerful approach for online optimization, allowing agents to dynamically refine control policies through continuous environmental interaction, leading to strong adaptability. RL excels at processing continuous data streams, adapting to non-stationary environments, and operating effectively without complete prior knowledge. However, RL faces several challenges in complex dynamic environments, including limitations in real-time modeling, prediction capability, and computational efficiency in complex dynamic environments.

To address these limitations, we propose a novel model-based reinforcement learning (MBRL) framework with a receding horizon (RH) mechanism for AVs motion control that incorporates an autoregressive (AR) model to enhance predictive accuracy. This framework leverages the AR model's capability to forecast future dynamic residuals from historical data, thereby improving sample efficiency and control performance in complex environments. The integration of the AR model, known for its effectiveness in time-series prediction [8], with the MBRL framework enhances predictive control under environmental uncertainties. The main contributions of this work are summarized as follows:

(i) The MBRL with AR (MBRL-AR) framework with the RH mechanism, which integrates temporal reasoning into vehicle dynamics modeling, is developed in this paper. This framework employs the AR model to forecast lateral dynamics by capturing sequential dependencies in vehicle

states, enabling precise prediction of trajectory deviations under varying conditions. Moreover, unlike traditional MPC solving a finite-horizon open-loop optimal control sequence online based on a nonlinear system, the proposed framework delivers a state-feedback control policy adaptable in real-time. It is expected to enhance the control system's adaptability, efficiency, and accuracy for trajectory tracking in uncertain environments.

(ii) We propose a dynamic residual generation mechanism that dynamically compensates for unmodeled dynamics, such as nonlinear tire behavior or external disturbances, not captured by the 2-DOF vehicle model. By generating corrective residuals in real time, this mechanism enhances both control performance and adaptability, while maintaining the interpretability and computational efficiency of the physical model.

(iii) The proposed framework is validated through extensive simulations, demonstrating superior tracking accuracy and adaptability compared to state-of-the-art learning predictive control methods, including receding horizon actor-critic learning (RHACL) [14], digital receding horizon actor-critic learning (DRHACL) [31], and classical baselines such as MPC [7] and LQR [6]. Real-world experiments, conducted on a HongQi electric vehicle (HQEV) platform, further confirm its effectiveness and feasibility, with MPC, recognized as the most widely adopted and practically robust algorithm in AV motion control due to its mature optimization capabilities and constraint handling, which was selected as the primary benchmark against the proposed approach, exhibiting marked improvements in error reduction and resilience under challenging country-road conditions.

The remainder of this paper is organized as follows. Section II reviews the related work on BMRL-AR for AVs. Section III outlines the modeling of vehicle system dynamics and introduces the AR model-based RL strategy. Section IV elaborates on implementing AR model-based RL for predicting dynamics residuals and generating a near-optimal control policy using actor-critic neural networks (NNs). Section V presents the results from both simulation and real-world experimental studies. Finally, Section VI provides conclusions.

II. Related work

Recent research has increasingly focused on improving model-based control algorithms by optimizing their performance in unknown environments and under uncertain model parameters [9], [10]. These studies emphasize that integrating online prediction and optimization with dynamic modeling represents a critical research direction for advancing robotics [11]. RL provides a robust framework for online optimization, enabling agents to dynamically refine control policies through continuous interaction with the environment, resulting in strong adaptive capabilities. MBRL further improves sample efficiency by incorporating predictive system dynamics models, effectively managing complex physical interactions [12]. However, significant challenges persist in complex dynamic environments, including real-time modeling, accurate prediction, cross-scenario generalization, and computational efficiency. Notably, achieving effective prediction and reasoning under environmental variability, ensuring generalizability across diverse scenarios, and maintaining high algorithmic efficiency remain critical open challenges that demand innovative solutions.

RL has become a prominent framework for AV control, typically categorized into model-free and model-based approaches. Model-free methods such as Soft Actor–Critic employ a maximum-entropy objective to improve convergence stability and exploration efficiency, yet suffer from high sample complexity and safety concerns during real-world interaction [36]. To address these limitations, hybrid architectures combine RL with classical feedback control, often learning a residual correction atop a nominal controller to enhance stability and adaptability [37].To improve computational and sample efficiency, several studies [32], [33] have explored model-based reinforcement learning, where a learned or known system model is used to generate synthetic rollouts for policy training. By incorporating model predictions into the learning loop, these approaches significantly reduce real-world interaction requirements and have been validated in autonomous driving tasks under various operational scenarios. Further improvements in sample efficiency have been achieved by combining imitation learning with MBRL. Specifically, Xu et al. [35] proposed a guided policy search-based MBRL framework for urban driving tasks, where expert demonstrations guide the initial policy optimization. Experimental results indicated an approximately 100-fold improvement in sample efficiency compared to conventional RL approaches. To enhance robustness under dynamic uncertainties, Wang et al. [34] introduced a System Identification Transformer and an Adaptive Dynamics Model trained across diverse simulated dynamic conditions. By explicitly modeling dynamic variability, the framework improves policy safety and reliability under changing system parameters.

In the context of AVs, recent advancements in MBRL have demonstrated considerable potential. Lian et al. proposed explicit state feedback control laws derived from actor-critic learning (ACL), which support both offline deployment and online learning while providing greater computational efficiency than model predictive control (MPC) [13]. Zhang et al. developed a receding-horizon RL (RHRL) method for discrete-time two-degree-of-freedom (2-DOF) dynamics and validated it in real-world experiments [14]. Concurrently, Lu et al. extended the RHRL framework to continuous-time formulations for trajectory tracking control on AVs [15]. Ma and Yin et al. further advanced RHRL by embedding a zero-sum differential game, improving robustness to uncertainties [16]. Nonetheless, existing methods for autonomous vehicles lack predictive awareness of environmental changes, constraining predictive control performance. This motivates the development of efficient real-time prediction algorithms integrated with vehicle system dynamics to infer dynamics residuals under complex uncertainties, thereby enhancing

TABLE I

THE PARAMETER DEFINITIONS OF AVS

Parameters	Definitions
m	The total mass of the autonomous vehicle
I_z	The yaw moment of inertia
l_f	The distance from the CoG to the front axle
l_r	The distance from the CoG to the rear axle
C_f	The cornering stiffnesses of front tires
C_r	The cornering stiffnesses of rear tires
u	The steering wheel angle
v_x	The longitudinal velocity in the local coordinate system
Ψ	The yaw angle in the global coordinate system
e_y	The lateral position error
e_Ψ	The yaw angle error

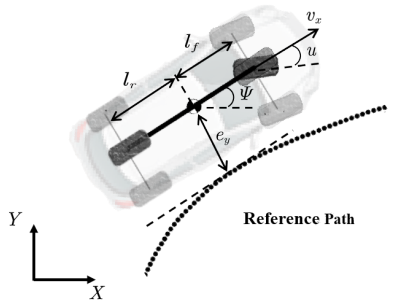

Fig. 1. 2-DoF vehicle lateral dynamics.

MBRL control performance.

AR models have achieved state-of-the-art performance in various generative tasks, owing to their inherent strengths in capturing temporal coherence, computational efficiency, and training stability. In image generation, models such as LlamaGen, NextStep-1, and Hi-MAR have delivered remarkable results [17], [18]. Similarly, innovative approaches for 3D mesh generation have been introduced by PolyGen and DeepMesh [19], [20]. Likewise, large language models (LLMs), including the GPT, LLaMA, and DeepSeek series, have demonstrated exceptional capabilities in text generation [21], [22]. These successes underscore the strong potential of AR models for predictive reasoning tasks. An emerging research direction involves integrating AR mechanisms into online modeling frameworks for intelligent systems. This integration enables agents to reason about dynamic residuals in uncertain environments, thereby improving modeling accuracy and enhancing predictive control performance [23].

III. PROBLEM FORMULATION

Following the parameter definitions in Tab. I, the well-established 2-DOF vehicle lateral dynamics model [30] is formulated as:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B_1}u + \boldsymbol{B_2}\omega_{des} \tag{1}$$

with

$$\begin{aligned} \boldsymbol{A}^{\mathsf{T}} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & -\frac{2(C_f+C_r)}{mv_x} & 0 & -\frac{2(C_f l_f - C_r l_r)}{I_z v_x} \\ 0 & \frac{2(C_f+C_r)}{m} & 0 & \frac{2(C_f+C_r)}{I_z} \\ 0 & -\frac{2(C_f l_f - C_r l_r)}{mv_x} & 1 & -\frac{2(C_f l_f^2 + C_r l_r^2)}{I_z v_x} \end{bmatrix}, \\ \boldsymbol{B_1} &= \begin{bmatrix} 0 \\ \frac{C_f}{m} \\ 0 \\ \frac{2C_f l_f}{I_z} \end{bmatrix}, \boldsymbol{B_2} = \begin{bmatrix} 0 \\ -\frac{2(C_f l_f - C_r l_r)}{mv_x} - v_x \\ 0 \\ -\frac{2(C_f l_f^2 + C_r l_r^2)}{I_z v_x} \end{bmatrix}, \end{aligned} \tag{2}$$

where $\omega_{des} = \dot{\Psi}_{des}$, and $\boldsymbol{x}$ is described as $[e_y\ \dot{e}_y\ e_\Psi\ \dot{e}_\Psi]^{\mathsf{T}}$.

The lateral dynamics of AVs can be formulated by incorporating a lumped disturbance term, which compensates for velocity variations, model uncertainties, and environmental factors [16], as follows:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B_1}u + \boldsymbol{B_2}\omega_{des} + \boldsymbol{\Delta_x}. \tag{3}$$

Initially neglecting the feedforward control term [6], the vehicle lateral dynamics can be simplified to

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B_1}u + \boldsymbol{\Delta_x}. \tag{4}$$

Define $\boldsymbol{A} = \bar{\boldsymbol{A}}$ and $\boldsymbol{B_1} = \bar{\boldsymbol{B}}_{\boldsymbol{1}}$, where $\bar{\boldsymbol{A}}$ and $\bar{\boldsymbol{B}}_{\boldsymbol{1}}$ represent the nominal values (or approximations) of $\boldsymbol{A}$ and $\boldsymbol{B_1}$, respectively. A p-order AR generator is constructed to estimate the evolution of $\boldsymbol{\Delta_x}$ over the prediction horizon. The mathematical expression is as follows:

$$\boldsymbol{\Delta_x}(t) = \sum_{i=1}^{p} \boldsymbol{\phi_i}\boldsymbol{\Delta_x}(t-i) + \varepsilon, \tag{5}$$

where $\boldsymbol{\Delta_x}(t)$ represents the observed value of the time series at time t; $\sum_{i=1}^{p} \boldsymbol{\phi_i}\boldsymbol{\Delta_x}(t-i)$ represents the weighted sum of the lag terms of the previous p periods; ε is the white noise error, and its median value in the prediction time domain is zero. Then the AR model can be simplified as:

$$\begin{aligned} \begin{bmatrix} \Delta_x(t+p) \\ \Delta_x(t+p-1) \\ \vdots \\ \Delta_x(t+1) \end{bmatrix} &= \mathcal{A} \begin{bmatrix} \Delta_x(t+p-1) \\ \Delta_x(t+p-2) \\ \vdots \\ \Delta_x(t) \end{bmatrix}, \\ \mathcal{A} &= \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_{p-1} & \phi_p \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \end{aligned} \tag{6}$$

Remark 1: Solving the characteristic equation $\det(\boldsymbol{\lambda}\mathbf{I} - \mathcal{A}) = 0$ yields the eigenvalues $\boldsymbol{\lambda}$ associated with the system matrix $\mathcal{A}$. The system is stable if and only if all eigenvalues lie strictly within the complex unit disk, thereby ensuring the stability of the corresponding state equation. Consequently, the stability of the linear system implies the stability of the AR state equation.

The control objective is to design a model-based reinforcement learning (MBRL) method that optimizes the control input u for the system in (3) subject to mechanical steering constraints $|u| \leq \frac{\pi}{4}$, while minimizing lateral tracking errors.

IV. MBRL-AR Model Approach

This section presents an online update method for a stabilized AR model. The updated AR model is integrated with a known, stable physical model to form the hybrid dynamic model given in (3). This hybrid formulation allows for residual inference over the prediction horizon and facilitates efficient computation of near-optimal control strategies.

A. AR Model

Stability is a prerequisite for ensuring reliable prediction and control performance under dynamic conditions. To enhance the prediction accuracy of the baseline 2-DOF vehicle dynamics model while maintaining stability, we develop a hybrid dynamics model that integrates the physical model with an AR residual generation component. The AR model, updated online, is designed to capture unmodeled dynamics while adhering to stability constraints. To assess the model's accuracy and stability, we define the prediction error of state derivatives as follows:

$$\begin{aligned} L_\Delta &= (\sum_{i=1}^{p} \hat{\boldsymbol{\phi}}_{\boldsymbol{i}} \boldsymbol{\Delta}_{\boldsymbol{x}}(t-i) - \boldsymbol{\Delta}_{\boldsymbol{x}}(t))^2 + \alpha s(\kappa(|\tilde{\phi}| - \beta))^2, \\ s &= \ln(1 + e^{\boldsymbol{\Delta}_{\boldsymbol{x}}(\boldsymbol{t})}), \tilde{\phi} = \sqrt{\hat{\phi}^2 + \epsilon}, \end{aligned} \tag{7}$$

where α, β, ϵ, and κ are hyperparameters. Specifically, the parameter α determines the global weight for AR coefficient convergence, κ regulates the steepness of the penalty when coefficients deviate beyond the convergence boundary, β adjusts the convergence margin of the AR model, and ϵ controls the smoothness of the AR coefficients within the convergence domain. $\alpha s(\kappa(|\tilde{\phi}| - \beta))^2$ constrains the AR coefficients to maintain model stability while approximating system dynamics residuals, as the stability conditions specified in Remark 1. The AR coefficients are updated using gradient-based optimization, with the gradient of the loss function given by:

$$\begin{aligned} \frac{\partial L_\Delta}{\partial \hat{\boldsymbol{\phi}}} &= 2\boldsymbol{\Delta}_{\boldsymbol{x}} L_\Delta + 2\alpha\kappa s(Z)\chi(Z)\frac{\hat{\boldsymbol{\phi}}}{\sqrt{\hat{\boldsymbol{\phi}}^2 + \epsilon}}, \\ Z &= \kappa(|\tilde{\phi}| - \beta), \chi(Z) = \frac{1}{\sqrt{1 - \exp(-Z)}} \end{aligned} \tag{8}$$

Once the AR model is updated online and its stability is verified, the refined coefficients are incorporated into the prediction horizon to generate accurate state predictions.

B. MBRL Approach with RH Mechanism

The objective of the RH optimal control problem is to minimize the following performance index:

$$\begin{aligned} V(\boldsymbol{x}) &= \int_{t_0}^{t_f} L_R(\boldsymbol{x}, u)\, d\tau + S_R(\boldsymbol{x_{tf}}), \\ S_R(\boldsymbol{x_{tf}}) &= \boldsymbol{x}_{\boldsymbol{tf}}^{\mathsf{T}} \boldsymbol{P} \boldsymbol{x_{tf}},\ L_R(\boldsymbol{x}, u) = \boldsymbol{x}^{\mathsf{T}} \boldsymbol{Q} \boldsymbol{x} + u^{\mathsf{T}} R u, \end{aligned} \tag{9}$$

where $\boldsymbol{P} \in \mathbb{R}^{4\times4}$, $\boldsymbol{Q} \in \mathbb{R}^{4\times4}$, and $\boldsymbol{R} \in \mathbb{R}^{1\times1}$ are the terminal cost matrix, weight matrix of errors, and weight matrix of the control, respectively, which are symmetric positive definite matrices. The terminal time is defined as $t_f = t_0 + t_s$, where t_0 is the initial time and t_s represents the time interval length.

In the implementation of the proposed MBRL framework by actor-critic structure, we solve the AR coefficient using a single-layer NN. After updating the AR model, the optimal control law u^* and the value function V^* are computed using initial stabilizing and successive actor-critic NNs.

$$V^* = \boldsymbol{W}_{\boldsymbol{c}}^{\mathsf{T}} \boldsymbol{\varphi}(\boldsymbol{x}) + \varepsilon_c(\boldsymbol{x}), \tag{10a}$$

$$u^* = \eta_a \tanh(\boldsymbol{W}_{\boldsymbol{a}}^{\mathsf{T}} \boldsymbol{\xi}(\boldsymbol{x}) + \varepsilon_a(\boldsymbol{x})), \tag{10b}$$

where $\boldsymbol{\varphi}(\boldsymbol{x})$ and $\boldsymbol{\xi}(\boldsymbol{x})$ indicate the activation functions, satisfying $\boldsymbol{\varphi}(\mathbf{0}) = \mathbf{0}$ and $\boldsymbol{\xi}(\mathbf{0}) = \mathbf{0}$, respectively. The NN weights are defined as $\boldsymbol{W_c} \in \mathbb{R}^N$ for the optimal critic function, and $\boldsymbol{W_a} \in \mathbb{R}^M$ for the control law. L and N represent the number of single-layer neurons in the critic and actor, respectively. The residual errors of the critic and actor NNs are denoted as $\varepsilon_c(\boldsymbol{x})$ and $\varepsilon_a(\boldsymbol{x})$, respectively. The activation function $\tanh(\cdot)$ is utilized to constrain the output of the actor NN, with the maximum of the control law $\eta_a > 0$.

The estimated cost function $\hat{V}(\boldsymbol{x})$ and control solution $\hat{u}$ are designed using the identifier weights of the critic and actor NNs as follows:

$$\hat{V}(\boldsymbol{x}) = \hat{\boldsymbol{W}}_{\boldsymbol{c}}^{\mathsf{T}} \boldsymbol{\varphi}(\boldsymbol{x}) \tag{11a}$$

$$\hat{u}(\boldsymbol{x}) = \eta_a \tanh(\hat{\boldsymbol{W}}_{\boldsymbol{a}}^{\mathsf{T}} \boldsymbol{\xi}(\boldsymbol{x})) \tag{11b}$$

where $\hat{\boldsymbol{W}}_{\boldsymbol{c}} \in \mathbb{R}^N$ and $\hat{\boldsymbol{W}}_{\boldsymbol{a}} \in \mathbb{R}^M$ denote the estimated weights for the critic and actor NNs, respectively.

Based on equations (4), (9), and (11a), the temporal difference (TD) error $\boldsymbol{e}_{td}$ and the terminal cost error can be obtained as follows:

$$\begin{aligned} \boldsymbol{e}_{td} &= \boldsymbol{\varphi}_{\boldsymbol{x}}^{\mathsf{T}}(\boldsymbol{x}) \hat{\boldsymbol{W}}_{\boldsymbol{c}} (\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B_1} u + \boldsymbol{\Delta}_{\boldsymbol{x}}) + L_R(\boldsymbol{x}, \hat{u}) \\ e_{t_f} &= \hat{\boldsymbol{W}}_{\boldsymbol{c}}^{\mathsf{T}} \boldsymbol{\varphi}(\boldsymbol{x}_{t_f}) - \boldsymbol{x}_{t_f}^{\mathsf{T}} \boldsymbol{P} \boldsymbol{x}_{t_f}. \end{aligned} \tag{12}$$

Consider the objective function that minimizes both the temporal difference error $\boldsymbol{e_{td}}$ and the terminal cost error e_{t_f} by adjusting the identifier weights of the critic network, as

$$E_c = \frac{1}{2} \boldsymbol{e}_{td}^{\mathsf{T}} \boldsymbol{e}_{td} + \frac{1}{2} e_{t_f}^2. \tag{13}$$

Subsequently, the identifier weights of the critic NN are updated using the gradient descent rule.

Define the control error e_a as the difference between the estimated control input (11b) for the system (4) and the optimal control law. That is,

$$e_a = \hat{\boldsymbol{W}}_{\boldsymbol{a}}^{\mathsf{T}} \boldsymbol{\xi}(\boldsymbol{x}) + \frac{1}{2} R^{-1} \boldsymbol{B_1}^{\mathsf{T}} \boldsymbol{\varphi}_{\boldsymbol{x}}^{\mathsf{T}}(\boldsymbol{x}) \hat{\boldsymbol{W}}_{\boldsymbol{c}}. \tag{14}$$

The objective function, which can be minimized by updating the identifier actor NN weights, is defined as

$$E_a = \frac{1}{2} e_a^2. \tag{15}$$

The update rule for the identifier actor NN weights, based on the gradient descent method. A comprehensive outline detailing the proposed methodology's execution is furnished within Algorithm 1.

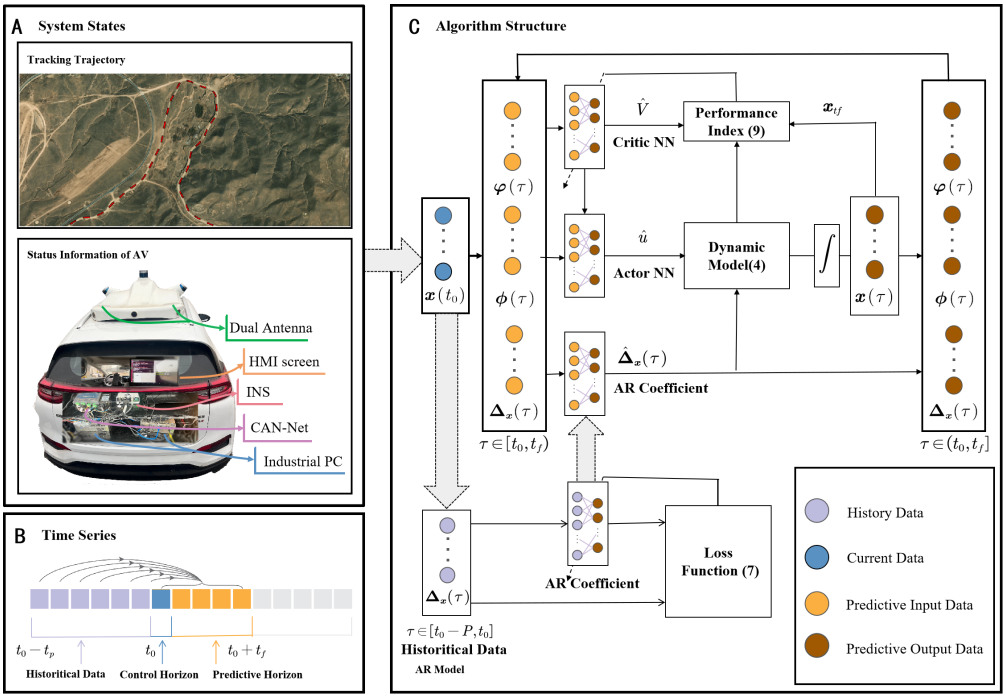

Fig. 2. Schematic of the MBRL-AR for AV Lateral Control.

TABLE II

ASSESSMENT INDEXES OF CONTROL PERFORMANCE

Index	**MBRL-AR**	RHACL[14]	DRHACL[31]	LQR[6]	MPC[7]
IAE	**45.67**	87.29	106.23	93.412	63.02
MAE	**0.027**	0.056	0.060	0.036	0.049
MLE	**0.044**	0.153	0.195	0.118	0.048
$\star$ (s)	1.47	**1.01**	7.61	2.13	6.21

"$\star$" denotes the online computational time of 1000 steps (s).

Remark 2: The prediction model within the receding horizon is stable, provided that the linear dominant part of the vehicle dynamics is linearly stable and the dynamics residual term, solved based on AR coefficients, satisfies the stability condition Remark 1. For the MBRL algorithm presented in continuous time, stability can be achieved through proper design of its parameters. The theoretical proofs of stability and convergence for the MBRL algorithm, which incorporates a convergent dynamic residual term in continuous time, are provided in [15], [16]. Notably, RHRL can stabilize inherently unstable systems, such as the inverted pendulum, through appropriate hyperparameter tuning [38]. By adjusting horizon length and learning parameters, the receding-horizon framework constrains instability and ensures closed-loop convergence despite unstable open-loop dynamics.

V. EXPERIMENTS

In this section, we establish a comprehensive CarSim simulation environment to evaluate the proposed algorithm against other state-of-the-art methods, thereby demonstrating its superior performance. Furthermore, the feasibility of the proposed algorithm is validated in a real-world environment.

To ensure algorithm stability and practical applicability, the parameters of the proposed algorithm were designed as follows. The error constraints used in the simulation are defined as follows:

$$\begin{aligned} x_1 = e_y \in [-5, 5]\text{m},\ x_2 = e_\varphi \in [-\pi/3, \pi/3]\text{rad}, \\ x_3 = \dot{e}_y \in [-10, 10]\text{m/s},\ x_4 = \dot{e}_\varphi \in [-\pi, \pi]\text{rad/s}. \end{aligned} \tag{16}$$

Based on these bars, the hyperparameter values of the algorithm to ensure its stability [15], [16] are listed below:

$$\begin{aligned} &\eta_a = \frac{\pi}{4},\ t_p = 1s,\ t_s = 0.02s,\ t_c = 0.02s,\ T = 50s, \\ &p = 2, \alpha = 10,\ \beta = 0.6,\ \epsilon = 1e-8,\ \kappa = 20, R = 0.5, \\ &Q = \text{diag}\{1, 1, 1, 1\},\ P = \text{diag}\{0.5, 0.5, 0.5, 0.5\}. \end{aligned} \tag{17}$$

We designed the activation function as follows:

$$\boldsymbol{\varphi} = \boldsymbol{x_c} \otimes \boldsymbol{x_t}, \quad \boldsymbol{\xi} = \boldsymbol{x_a} \otimes \boldsymbol{x_t}, \tag{18}$$

where

$$\begin{aligned} &\boldsymbol{x_c} = [x_1^2, x_2^2, x_3^2, x_4^2, x_1x_2, x_1x_3, x_1x_4, x_2x_3, x_2x_4, x_3x_4]^\mathsf{T}, \\ &\boldsymbol{x_a} = \boldsymbol{x},\ \boldsymbol{x_t} = [1, t, t^2]^\mathsf{T}, [x_1; x_2; x_3; x_4] \triangleq [e_y; \dot{e}_y; e_\Psi; \dot{e}_\Psi]. \end{aligned} \tag{19}$$

The Kronecker product is denoted as $\otimes$. The metrics for evaluating control performance [16], including the integral of absolute error (IAE), maximum lateral error (MLE), maximum angular error (MAE), and the root mean square of lateral errors (RMSLE), are defined as follows:

$$\begin{aligned} &\text{IAE} = \int_0^T \|\boldsymbol{x}\| dt, \text{RMSLE} = \frac{1}{K}\sqrt{\sum_{i=1}^{K} x_1^2(i)}, \\ &\text{MLE} = \max_{\tau \in [T_0, T]} |x_1|, \text{MAE} = \max_{\tau \in [T_0, T]} |x_3|. \end{aligned} \tag{20}$$

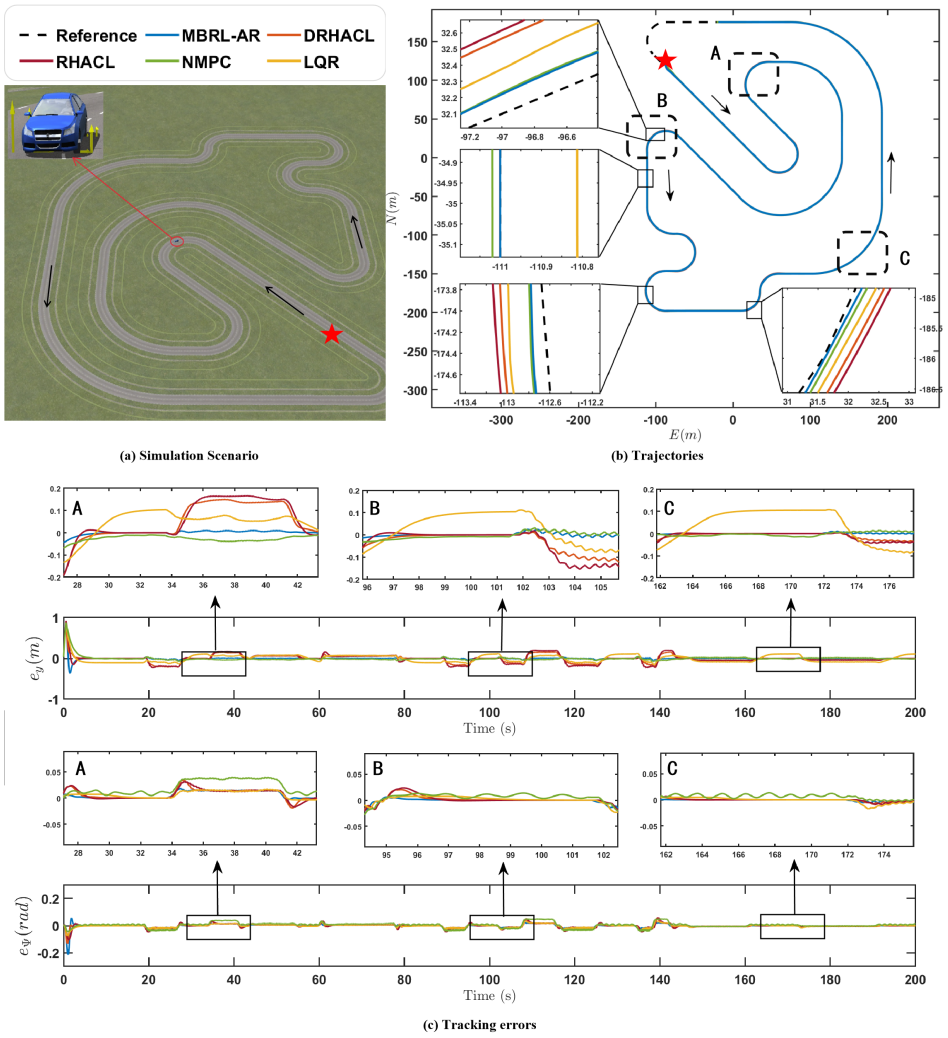

Fig. 3. The comprehensive trajectory tracking simulation (The red star marks the starting point of the AV, with thick black arrows indicating its movement direction). The AV travels at a speed of 36 km/h.

The constructed comprehensive simulation scenario, featuring continuous high curvatures, is shown in Fig. 3(a). The test vehicle completed 200 seconds within this simulated environment, and Fig. 3(b) displays the trajectories generated by the proposed MBRL-AR alongside the comparison methods, including RHACL, DRHACL, LQR, and MPC. For the comparative analysis, DRHRL, RHRL, and MBRL-AR, which utilized similar parameters, each algorithm was initialized with a 2 m lateral error and simulated for 10000 steps at a speed of 36 km/h. Fig. 3(c) presents their corresponding tracking errors, with zoomed-in insets highlighting critical segments for better comparisons. The proposed MBRL-AR exhibits more stable convergence and lower error amplitudes, mitigating model uncertainties through temporal logic reasoning. Overall, these zoomed views underscore the framework's robustness in high-curvature maneuvers, where the AR-enhanced predictions enable proactive steering adjustments, minimizing error amplification from unmodeled dynamics.

The performance metrics are summarized in Table II. where MBRL-AR exhibits the lowest IAE (45.67) and MAE (0.027), outperforming RHACL (IAE: 87.29, MAE: 0.056), DRHACL (IAE: 106.23, MAE: 0.060), LQR (IAE: 93.412, MAE: 0.036), and MPC (IAE: 63.02, MAE: 0.049) by 48-57% in IAE relative to the next best (MPC). This edge stems from the AR model's adaptive residual forecasting, which enables proactive corrections beyond fixed-model baselines (LQR, MPC) and improves temporal dependency capture over other RL methods (RHACL, DRHACL), resulting in faster convergence in nonlinear regimes. While the proposed method shows slightly higher online computation than RHACL, this is attributed to the AR coefficient updates via gradient descent for real-time residual refinement, unlike RHACL's offline actor-critic policy, which lacks such dynamic modeling overhead. In summary, simulation results demonstrate that the proposed method effectively adapts to dynamic model variations across different tracking trajectories (e.g., straight-line, S-turn, U-turn). This adaptability leads to superior control performance, characterized by high precision, robustness, and self-adaptation.

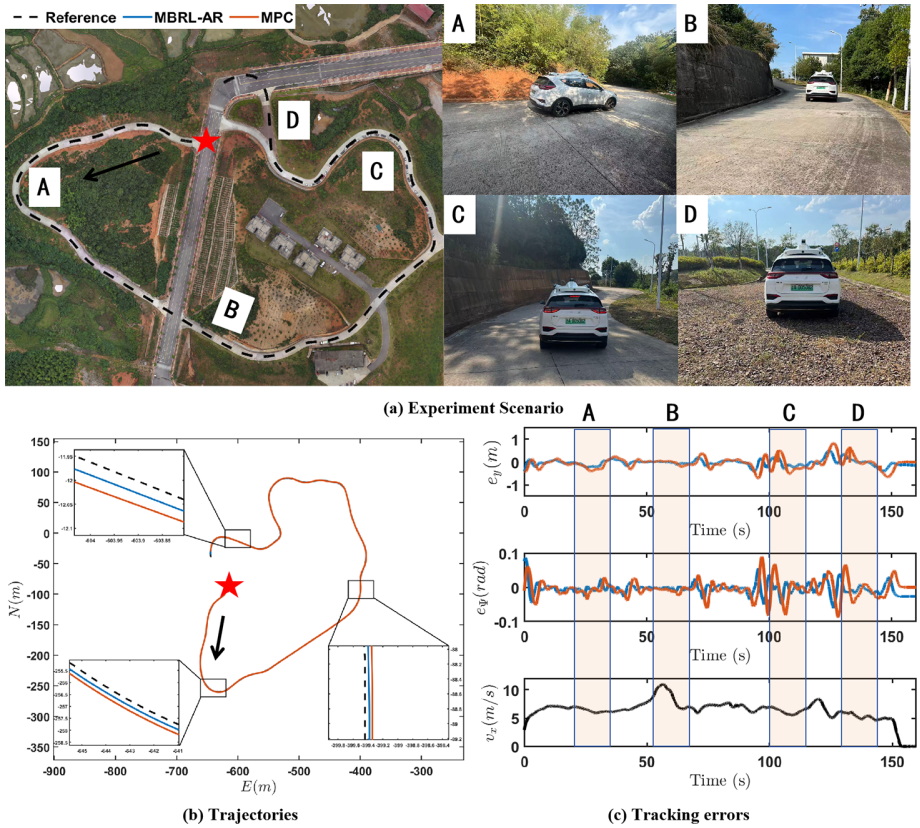

Fig. 4. The experiment on paved road (The red star denotes the starting point of AV, with thick black arrows indicating its movement direction.)

Real-world experiments were conducted on the HongQi electric vehicle (HQEV) platform Fig. 2(a), which was equipped with an industrial computer (Intel Core i9-12900K) operating at 50 Hz, a PP7-E1 inertial measurement unit (IMU), and GNSS-502 dual antennas. A PID controller was employed to track the longitudinal vehicle speed. The evaluation of the algorithm's performance was carried out in a country-road environment containing substantial challenges, including a 20-degree maximum slope, a maximum curvature of 1/12 m^{-1}, and multiple steep slopes and sharp turns, as depicted in Fig. 4(a). The vehicle successfully converged to the target trajectory, as shown in Fig. 4(b). The control error and vehicle speed profiles are presented in Fig. 4(c). The hyperparameters used in the real vehicle test were identical to those employed during policy training. Notably, a PID controller was set to take over if the algorithm became unstable; however, this safety mechanism was never triggered throughout dozens of test runs. On steep unstructured roads with significant uncertainties, the proposed algorithm achieves smaller IAE (MBRL-AR 13.182 vs. MPC 23.298), MLE (MBRL-AR 0.518 vs. MPC 0.842), and MAE (MBRL-AR 0.087 vs. MPC 0.090) compared to the MPC algorithm [7], owing to the proposed AR residuals being incrementally updated from real-time sensor data to adeptly handle unmodeled disturbances such as wheel slip on loose gravel and wind gusts, which temporarily challenge the baseline MPC's rigid predictions. This adaptive mechanism ensures smoother error profiles and enhanced resilience, contrasting with MPC's static approach that struggles under dynamic environmental shifts. Results demonstrate the practicality of our algorithm in highly uncertain, unstructured road conditions and its potential for real-world vehicle applications.

VI. Conclusions

This paper introduces an MBRL motion control framework, integrating an RH mechanism with AR modeling to address the dynamic residuals of AVs. Specifically, it proposes an AR model-based RL algorithm that leverages AR modeling to efficiently generate near-optimal control policies within a receding horizon framework, iteratively refining the strategy through real-time system interactions to offer a robust solution against uncertainties, outperforming both advanced algorithms like RHACL, DRHACL, and classical methods such as LQR and MPC with superior adaptability, precision, and computational efficiency. Validated on the HQEV platform, the approach demonstrates its practical effectiveness and feasibility, with empirical results showcasing reduced tracking errors and enhanced control performance under challenging conditions.

References

[1] P. Riekert,T. Schunck, "Zur fahrmechanik des gummibereiften kraftfahrzeugs," Ingenieur-Archiv, vol. 11, pp. 210-224, 1940.

[2] L. Segel, "Theoretical prediction and experimental substantiation of the response of the automobile to steering," control Proceedings of the Institution of Mechanical Engineers: Automobile Division, vol. 10, no. 1, pp: 310-330, 1956.

Algorithm 1 The Pseudocode of MBRL-AR

Notations: β_N: the maximum iterations for each receding-horizon. t_p, t_c: the length of the time interval and control time in each time interval. T: the simulation time.

Initialize $\hat{\boldsymbol{W}}_{\boldsymbol{c}}$, $\hat{\boldsymbol{W}}_{\boldsymbol{a}}$ and $\boldsymbol{x}(t_0)$, load $\boldsymbol{\phi}$;
while $t_0 \le T$ **do**
  $i = 1$
  while $i \le \beta_m$ **do**
    $t = t_0$
    while $t \le t_0 + t_p$ **do**
      Calculate $\hat{u}$ via (11b);
      $j = 1$;
      while $j \le \beta_m$ **do**
        Update laws $\dot{\hat{\boldsymbol{W}}}_{\boldsymbol{c}}$, $\dot{\hat{\boldsymbol{W}}}_{\boldsymbol{a}}$ by (13), (15). Then, compute $\hat{\boldsymbol{W}}_{\boldsymbol{c}}$, $\hat{\boldsymbol{W}}_{\boldsymbol{a}}$.
        $j = j + 1$;
      end while
      Update the AR coefficient via (7)
      $t = t + T_s$;
    end while
    $i = i + 1$;
  end while
  $t_0 = t_0 + t_c$;
end while

[3] N. Spielberg, M. Brown, N. Kapania, "Neural network vehicle models for high-performance automated driving," Science robotics,vol. 4, no. 28, pp: eaaw1975, 2019.

[4] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann et al., "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[5] Q. Ma, X. Zhang, X. Xu, Y. Yang and E. Q. Wu, "Self-learning sliding mode control based on adaptive dynamic programming for nonholonomic mobile robots," *ISA transactions*, vol. 142, pp. 136-147, 2023.

[6] H. Jung, D. Jung and S. B. Choi, "LQR control of an all-wheel drive vehicle considering variable input constraint". *IEEE Transactions on Control Systems Technology*, vol. 30, no. 1, 2021.

[7] P. Hang, X. Xia, G. Chen, and X. Chen, "Active safety control of automated electric vehicles at driving limits: A tube-based MPC approach," *IEEE Transactions on Transportation Electrification*, vol. 8, no.1, pp. 1338-1349, 2021.

[8] N. Ke, A. Singh, A. Touati, A. Goyal, Y. Bengio, D. Parikh, D. Batra, "Modeling the long term future in model-based reinforcement learning," *International Conference on Learning Representations,* 2018.

[9] T. Miki, J. Lee, J. Hwangbo, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science robotics*, vol. 7, no. 62, pp: eabk2822, 2022.

[10] M. O'Connell,G. Shi,X. Shi, "Neural-fly enables rapid learning for agile flight in strong winds" *Science Robotics*, vol. 7, no. 66, eabm6597, 2022.

[11] P. Wensing,M. Posa,Y. Hu, "Optimization-based control for dynamic legged robots". *IEEE Transactions on Robotics*, vol. 40, pp: 43-63, 2023.

[12] M. Medany, L. Piglia, L. Achenbach, et al., "Model-based reinforcement learning for ultrasound-driven autonomous microrobots," Nature Machine Intelligence, pp. 1–15, 2025.

[13] C. Lian, X. Xu, H. Chen and H. He, "Near-optimal tracking control of mobile robots via receding-horizon dual heuristic programming," *IEEE transactions on cybernetics*, vol. 46, no. 11, pp. 2484-2496, 2015.

[14] X. Zhang, Y. Lu, W. Z. Li, and X. Xu, "Receding horizon reinforcement learning algorithm for lateral control of intelligent vehicles," *Acta Autom. Sin.*, vol. 45, pp. 1-12, 2022.

[15] Y. Lu, W. Li, X. Zhang, "Continuous-time receding-horizon reinforcement learning and its application to path-tracking control of autonomous ground vehicles," *Optimal Control Applications and Methods*, vol. 44, no. 3, pp. 1129-1147, 2023.

[16] Q. Ma, X. Yin, X. Zhang, X. Xu and X. Yao, "Game-theoretic Receding-Horizon Reinforcement Learning for Lateral Control of Autonomous Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 10, 2024.

[17] P. Sun, Y. Jiang, S. Chen, et al., "Autoregressive model beats diffusion: Llama for scalable image generation," arXiv:2406.06525, 2024.

[18] T. Li, Y. Tian, H. Li, et al., "Autoregressive image generation without vector quantization," Advances in Neural Information Processing Systems, vol. 37, pp. 56424–56445, 2024.

[19] C. Nash, Y. Ganin, S. M. A. Eslami, and P. Battaglia, "PolyGen: An autoregressive generative model of 3D meshes," International Conference on Machine Learning (ICML), 2020.

[20] R. Zhao, "DeepMesh: Auto-regressive artist-mesh creation with reinforcement learning," International Conference on Computer Vision (ICCV), 2025.

[21] S. Bubeck, V. Chandrasekaran, R. Eldan, et al., "Sparks of artificial general intelligence: Early experiments with GPT-4," arXiv preprint arXiv:2303.12712, 2023.

[22] H. Touvron, T. Lavril, G. Izacard, et al., "LLaMA: Open and efficient foundation language models," arXiv:2302.13971, 2023.

[23] J. Cen, C. Yu, H. Yuan, Y. Jiang, S. Huang, J. Guo, X. Li, Y. Song, H. Luo, F. Wang, D. Zhao, H. Chen, "WorldVLA: Towards Autoregressive Action World Model," arXiv:2506.21539, 2025.

[24] M. Bansal, A. Krizhevsky, A. Ogale., "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," *arXiv preprint arXiv:*1812.03079, 2018.

[25] B. Paden, M. Čáp, S. Yong, D. Yershov, E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*,vol. 1, no. 1, pp. 33-55, 2016.

[26] R Rajamani, "Vehicle dynamics and control," *Springer Science and Business Media,* 2011.

[27] Y. Chen, S. Li, X. Tang; K. Yang; D. Cao, X. Lin, "Interaction-aware decision-making for autonomous vehicles," *IEEE Transactions on Transportation Electrification,* vol. 9, no. 3, pp. 4704-4715, 2023.

[28] R. Sutton, A. Barto. "Reinforcement learning: An introduction," *Cambridge: MIT press,* 1998.

[29] P. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, "Continuous control with deep reinforcement learning,"*arXiv preprint arXiv,* 1509.02971, 2015.

[30] A. Artuñedo, M. Moreno-Gonzalez, J. Villagra, "Lateral control for autonomous vehicles: A comparative evaluation," *Annual Reviews in Control*, vol. 57, no. 100910, 2024.

[31] X. Zhang, W. Li, X. Xu and W. Jiang, "A digital receding-horizon learning controller for nonlinear continuous-time systems," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 8136-8141, 2020.

[32] Wu, Jingda, Zhiyu Huang, and Chen Lv. "Uncertainty-aware model-based reinforcement learning: Methodology and application in autonomous driving." IEEE Transactions on Intelligent Vehicles 8.1 (2022): 194-203.

[33] Song, Shaoyu, et al. "Data efficient reinforcement learning for integrated lateral planning and control in automated parking system." Sensors 20.24 (2020): 7297.

[34] Wang, Sean J., Honghao Zhu, and Aaron M. Johnson. "Pay attention to how you drive: Safe and adaptive model-based reinforcement learning for off-road driving." 2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024.

[35] Xu Zhuo, Jianyu Chen, and Masayoshi Tomizuka. "Guided policy search model-based reinforcement learning for urban autonomous driving." arXiv preprint arXiv:2005.03076 (2020).

[36] Haarnoja, Tuomas, et al. "Soft actor-critic algorithms and applications." arXiv preprint arXiv:1812.05905 (2018).

[37] Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, Sergey Levine "Residual Reinforcement Learning for Robot Control"

[38] Li L, Li D, Song T, et al. Actor–critic learning control with regularization and feature selection in policy gradient estimation[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 32(3): 1217-1227.