Robotica (2024), **42**, pp. 3263–3280
doi:10.1017/S026357472400119X

RESEARCH ARTICLE

Curriculum reinforcement learning-based drifting along a general path for autonomous vehicles

Kai Yu[1], Mengyin Fu[1,2], Xiaohui Tian[1], Shuaicong Yang[1] and Yi Yang[1]

[1]School of Automation, Beijing Institute of Technology, Beijing, China
[2]School of Automation, Nanjing University of Science and Technology, Nanjing, China
Corresponding author: Yi Yang; Email: yang_yi@bit.edu.cn

Received: 13 March 2024; **Revised:** 12 June 2024; **Accepted:** 2 August 2024; **First published online:** 2 December 2024

Keywords: autonomous vehicles; drifting control; reinforcement learning; curriculum learning; randomization

Abstract

Expert drivers possess the ability to execute high sideslip angle maneuvers, commonly known as drifting, during racing to navigate sharp corners and execute rapid turns. However, existing model-based controllers encounter challenges in handling the highly nonlinear dynamics associated with drifting along general paths. While reinforcement learning-based methods alleviate the reliance on explicit vehicle models, training a policy directly for autonomous drifting remains difficult due to multiple objectives. In this paper, we propose a control framework for autonomous drifting in the general case, based on curriculum reinforcement learning. The framework empowers the vehicle to follow paths with varying curvature at high speeds, while executing drifting maneuvers during sharp corners. Specifically, we consider the vehicle's dynamics to decompose the overall task and employ curriculum learning to break down the training process into three stages of increasing complexity. Additionally, to enhance the generalization ability of the learned policies, we introduce randomization into sensor observation noise, actuator action noise, and physical parameters. The proposed framework is validated using the CARLA simulator, encompassing various vehicle types and parameters. Experimental results demonstrate the effectiveness and efficiency of our framework in achieving autonomous drifting along general paths. The code is available at https://github.com/BIT-KaiYu/drifting.

1. Introduction

Drifting is a cornering maneuver characterized by significant lateral slip. During drifting, the driver induces the vehicle into a state of oversteer, causing the tires to lose traction and resulting in substantial sideslip angles. This technique is commonly observed in motorsport events, where drivers execute rapid turns while maintaining a drifting state. As an extreme sport, drifting has attracted considerable research interest, particularly in the field of autonomous drifting [1–3]. The exploration of autonomous drifting holds great potential for advancing autonomous driving technology by expanding handling limits and improving vehicle stability. However, controlling autonomous drifting presents significant challenges due to its inherent complexity and potential risks. This is particularly true when navigating general paths that exhibit a diverse range of curvature variations, which are commonly encountered in real-world driving environments.

Previous studies have employed vehicle models of varying fidelity to analyze vehicle stability during cornering and demonstrate the existence of drift equilibria, laying the foundation for autonomous drifting control [4–6]. Notably, Velenis et al. [7] and Goh et al. [8] have developed controllers for achieving steady-state drifting in simulations and experiments, respectively. To track a reference drifting trajectory along a sharp bend, Zhang et al. [9] proposed a comprehensive framework for path planning and motion control. In the context of drifting maneuvers, they employed a rule-based algorithm to generate reference

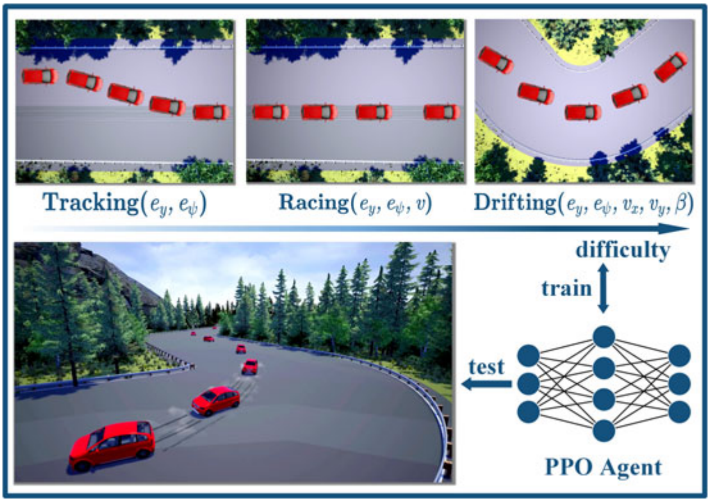
Tracking(e_y, e_ψ)
Racing(e_y, e_ψ, v)
Drifting(e_y, e_ψ, v_x, v_y, β)
difficulty
train
test
PPO Agent

Figure 1. *Our framework employs curriculum reinforcement learning to train an agent for autonomous drifting along general paths. The curriculum is designed by considering the various vehicle states involved in the task.*

vehicle states and control actions. Additionally, Chen et al. [10] developed a hierarchical dynamic drifting controller capable of uniformly tracking a general path during both drifting maneuvers and typical cornering maneuvers. However, the applicability of their approach to general paths is limited, as they rely on carefully designed road shapes to facilitate the calculation of drift equilibria.

The nonlinearity of the model and the difficulty of obtaining accurate parameters present challenges for achieving autonomous drifting using traditional methods. However, learning-based methods, particularly reinforcement learning, provide a solution for such cases. Reinforcement learning (RL) is a data-driven approach that involves learning through interactions with the environment [11]. Since explicit mathematical models of the system are not required, reinforcement learning has been successfully applied to solve various complex control tasks [12–14]. Domberg et al. [15] employed a highly accurate vehicle simulation to train a vehicle for steady-state drifting and subsequently transferred the learned controller to a physical model car. In their work, Cai et al. [16] introduced a reinforcement learning-based controller for autonomous drifting along complex paths. They formulated the problem as a trajectory tracking task by learning from human player operations.

Considering the existing works, performing drifting maneuvers along general paths using model-based control methods presents challenges due to the difficulties in accurately modeling vehicle dynamics. When applying reinforcement learning to autonomous drifting, the presence of multiple objectives makes direct training time-consuming and challenging. Curriculum learning (CL) has been proven effective in accelerating reinforcement learning training across various applications [17]. However, designing and sequencing the curriculum for autonomous driving tasks have not been thoroughly studied. Therefore, this paper presents a novel curriculum reinforcement learning-based control framework for autonomous drifting along general paths, as illustrated in Figure 1. The main contributions of this paper are as follows:

- A control framework for autonomous drifting along general paths is proposed, taking into account vehicle dynamics and leveraging reinforcement learning. Our framework enables the vehicle to follow complex paths at high speeds while executing drifting maneuvers during sharp corners.
- CL is introduced to accelerate the convergence rate and improve the final performance. By considering the vehicle's dynamics and customizing the reward function, the overall task is

divided into three stages of increasing difficulty: low-speed trajectory tracking, high-speed path following, and high-speed drifting.

- To enhance the generalization ability of our controller, we introduce randomization into sensor observation noise, actuator action noise, and physical parameters.
- Experiments conducted in CARLA demonstrate the effectiveness and efficiency of our framework even when encountering vehicles of diverse sizes, masses, and friction coefficients.[1]

The remainder of this paper is organized as follows: Section 2 provides an overview of the related work. Section 3 elucidates the background information concerning drifting and reinforcement learning. Our proposed method is presented in Section 4, encompassing the RL-based framework, the curriculum setup, and the randomization techniques employed. The experimental results are presented in Section 5. Finally, Section 6 draws the conclusion.

2. Related work

As a typical vehicle motion observed under extreme conditions, drifting has been extensively studied. Prior research on autonomous drifting can be categorized into two groups: traditional methods and learning-based methods.

a) Traditional methods: The investigation of traditional methods encompasses two components: modeling and control. In the modeling aspect, Inagaki et al. [4] analyzed vehicle dynamic stability during critical cornering using a two-state model for lateral dynamics. Ono et al. [5] employed a bicycle model with nonlinear tyres to demonstrate that vehicle spin phenomena can be attributed to the saturation of rear side forces. In order to abandon the constraint of small slip and steering angles, a nonlinear four-wheel vehicle model was used by Edelmann et al. [18]. Additionally, Hindiyeh et al. [19] presented an analysis of drifting using a three-state bicycle model that incorporates longitudinal dynamics.

Various control methods are employed in drifting control, such as feedforward-feedback control [20], linear quadratic regulator (LQR) [21, 22], model predictive control (MPC) [23–25], etc. Zhang et al. [9] employed a rule-based algorithm to generate an input sequence for steering and torque, enabling the vehicle to track a reference drifting trajectory along a sharp bend. Chen et al. [10] integrated drifting and typical cornering control using the dynamic drifting inverse model to track a customized path during both drifting and typical cornering maneuvers.

Overall, traditional methods for drifting control typically rely on precise mathematical models and system dynamics equations. However, these models struggle to accurately capture the highly nonlinear and dynamic behaviors that are characteristic of drifting along general paths.

b) Learning-based methods: In contrast to traditional methods, learning-based methods can establish direct input–output relationships, often requiring minimal or no explicit model information. Acosta et al. [26] introduced a data-driven approach for drifting control, utilizing feedforward neural networks and a model predictive controller. The internal parameters and drifting steady-state references are learned directly from a real driver using supervised machine learning. Zhou et al. [27] employed a combination of a relatively simple vehicle model and a neural network to compensate for model errors, achieving precise drifting control using MPC.

Different from methods based on neural networks, RL-based drifting control methods, due to their interactive learning nature, possess the capability to fully exploit the maximum potential of the vehicle. Cutler et al. [28] employed a model-based RL algorithm to achieve steady-state drifting motion. Initial policies were learned from data obtained using traditional control methods. Domberg et al. [15] employed a highly accurate vehicle simulation to train the vehicle for steady-state drifting and subsequently transferred the learned controller to a physical model car. In [16], the model-free RL algorithm,

[1] https://youtu.be/94Dez-1yN-U

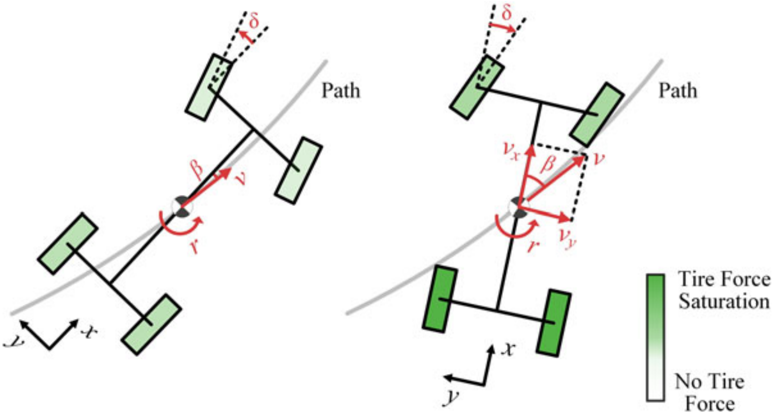

Figure 2. *Comparison of typical cornering (left) and drifting (right).*

soft actor-critic (SAC), was used to train a drifting controller in CARLA. An experienced driver provided reference states for both training and evaluation by operating the vehicle on different maps and recording corresponding trajectories.

What sets our approach apart from the aforementioned methods is that our control framework does not require any vehicle mathematical models. Furthermore, to address the challenges associated with multiple objectives in drifting along general paths, we incorporate CL to divide the task into multiple stages of increasing difficulty.

3. Background

3.1. Autonomous drifting

Drifting differs significantly from typical cornering when considering vehicle maneuvering, as depicted in Figure 2. To an observer, drifting can typically be identified by three distinctive characteristics [29]. The first and foremost essential is a large sideslip angle β, indicating a substantial angular difference between the vehicle's travel direction and its heading direction. This corresponds to the vehicle exhibiting significant lateral sliding. The sideslip angle can be calculated by:

$$\beta = \arctan\left(\frac{v_y}{v_x}\right), \tag{1}$$

where v_x and v_y, respectively, represent the vehicle's longitudinal velocity and lateral velocity in the body-fixed coordinate system at the center of gravity.

The second characteristic is large drive torques. For rear-wheel drive vehicle, large drive torques result in the saturation of tire friction at the rear wheels. The tire forces are given by the "friction circle" relationship:

$$F_x{}^2 + F_y{}^2 = (\mu F_z)^2, \tag{2}$$

where F_x, F_y, F_z, and μ, represent the longitudinal force, the lateral force, the normal load on the rear tire, and the coefficient of friction, respectively. The equation establishes the principle that the total force generated at the tire must not exceed the maximum force available at the tire due to friction.

The final characteristic is countersteer, which refers to steering the vehicle in the opposite direction of the intended turn. This implies that the steering angle δ and the vehicle's yaw rate r have opposite signs. Countersteer assists in stabilizing the vehicle and preventing it from spinning out or losing control completely.

During drifting, there are complex interactions among multiple forces. When a vehicle drifts, the lateral, longitudinal, and vertical forces acting on the front and rear wheels undergo changes.

These forces are coupled together and impact the vehicle's dynamic behavior. The coupling of these forces increases the complexity of the vehicle's behavior, necessitating a comprehensive consideration of the effects of multiple forces and corresponding adjustments during the drifting process. Therefore, solving the task of autonomous drifting using traditional methods is a challenging endeavor [30].

3.2. Reinforcement learning

Reinforcement learning is a branch of machine learning aimed at enabling an agent to learn optimal decision-making strategies through interaction with the environment [11]. In reinforcement learning, the agent learns by observing the state of the environment, taking different actions, and receiving rewards or penalties from the environment. The core idea of reinforcement learning is to maximize cumulative rewards by optimizing action selection through trial and error and feedback.

As a classical formalization for sequential decision-making problems, Markov decision process (MDP) provides a mathematical model to formalize reinforcement learning problems. It consists of a set of states S, a set of actions A, transition probabilities $P(s_{t+1} \mid s_t, a_t)$, a reward function $R: S \times A \to \mathbb{R}$, and a discount factor $\gamma \in [0, 1]$.

In reinforcement learning, a policy π refers to the action selection mechanism employed by an agent in a specific environmental state. It establishes the mapping from states to actions, determining which action the agent should take in different states. Specifically, at each discrete time step t, the agent observes a certain feature representation of the environmental state, $s_t \in S$. Then, the agent selects an action based on the current state: $a_t \sim \pi(\cdot \mid s_t)$. As a consequence of its action, the agent receives a numerical reward $r_t = R(s_t, a_t)$ and the environment's new state s_{t+1}, which is sampled based on the transition function. The objective of reinforcement learning training is to find a policy π that maximizes the discounted sum of rewards over an infinite horizon:

$$\pi^* = \arg\max_{\pi} \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right], \tag{3}$$

where $\mathbb{E}^{\pi}[\cdot]$ is computed as the expectation over the distribution of sequences $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \cdots)$ obtained under the policy π and transition P.

3.3. PPO and GAE

Proximal policy optimization (PPO) [31] is one of the most popular reinforcement learning algorithms. It interleaves the collection of new transitions with policy optimization. After a batch of new transitions (s_t, a_t, r_t, s_{t+1}) is collected, optimization is performed to maximize the objective:

$$L(\theta) = \hat{\mathbb{E}}_t\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right], \tag{4}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the probability ratio. Here, ϵ is a hyperparameter that controls the amount of clipping. In PPO, the unconstrained objective function encourages the agent to take the actions that have positive advantage while clipping limits too large of the policy update.

In (4), the advantage function $\hat{A}_t$ is necessary for the policy gradient and it can usually be estimated by:

$$\hat{A}^{\pi}(s_t, a_t) = \hat{Q}^{\pi}(s_t, a_t) - \hat{V}^{\pi}(s_t), \tag{5}$$

where $\hat{Q}^{\pi}(s_t, a_t)$ is the estimation of the action-value function $Q^{\pi}(s_t, a_t) = \mathbb{E}[R_t \mid s_t, a_t]$ and $\hat{V}^{\pi}(s_t)$ is the approximation of the state-value function $V^{\pi}(s_t) = \mathbb{E}[R_t \mid s_t]$. Monte Carlo methods are often used to construct $\hat{Q}^{\pi}(s_t, a_t)$,

$$\hat{Q}^{\pi}(s_t, a_t) = \sum_{l=0}^{\infty} \gamma^l r_{t+l}. \tag{6}$$

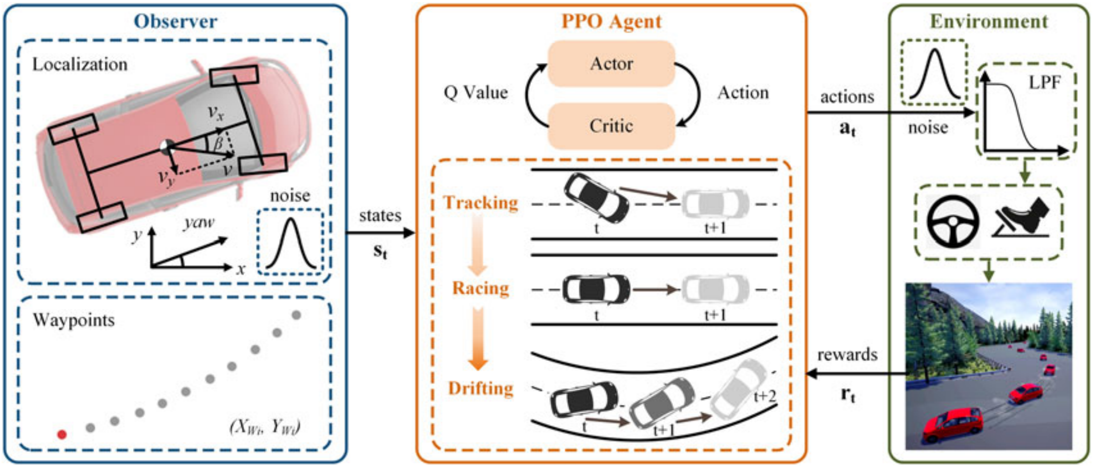

Figure 3. *Proposed learning-based drifting framework.*

Generalized advantage estimator (GAE) [32] is one style of policy gradient implementation that adjusts the bias-variance trade-off. It is defined as the exponentially weighted average of multi-step estimators in the following way:

$$\hat{V}_t^{GAE(\gamma,\lambda)} = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V, \tag{7}$$

where δ_t^V is the TD error:

$$\delta_t^V = r_t + \gamma V(s_{t+1}) - V(s_t). \tag{8}$$

4. Methodology

4.1. Problem formulation

We formulate the drifting control problem as MDP. The goal is to achieve high-speed path following while enabling the vehicle to execute drifting maneuvers during cornering. To achieve high-speed drifting with an autonomous vehicle, four main objectives need to be considered: (1) avoiding collisions with road boundaries, (2) precisely following the reference trajectory, (3) maintaining high speeds, and (4) executing effective drifting maneuvers in corners.

The learning-based framework is shown in Figure 3. The controller's inputs include the vehicle's states and the reference trajectory. The action commands generated by the controller are passed through a low-pass filter before being executed by the vehicle. Rewards received from the environment provide feedback for the controller's learning process. PPO with GAE is used to train the agent.

1) State space: In MDP, states represent the conditions or situations in which a system or agent may exist. They are crucial as they provide key information for the agent and influence the actions it takes. The design of states should reflect the impact of the agent's actions on the environment, but complex states can increase the complexity of the problem.

In order to implement autonomous drifting of a vehicle, the state space needs to encompass information about the vehicle's state and the environmental state. The vehicle state observation is shown in Figure 4, which consists of the continuous variables of the lateral position from the reference trajectory e_y, the heading angle error e_ψ, forward and side velocities (v_x, v_y), and sideslip angle β. We denote the state space S as:

$$S = \{e_y, e_\psi, v_x, v_y, \beta, \Gamma\}, \tag{9}$$

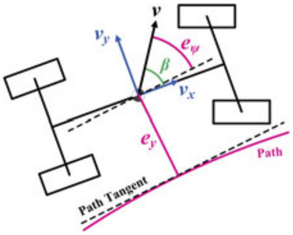

Figure 4. *Observation for vehicle states.*

where Γ contains 10 (x, y) positions in the reference trajectory ahead. For the general case, they are represented in the body frame coordinate system. e_y and e_ψ characterize whether the vehicle stays on the expected trajectory. v_x is used to measure the vehicle's travel speed, while v_y indicates whether the vehicle is undergoing lateral motion. β is the angle between the vehicle and the direction of travel. A larger sideslip angle indicates that the vehicle is drifting laterally, while a smaller sideslip angle suggests that the vehicle is more stable on the trajectory. All state variables are individually normalized to [0, 1] based on their respective maximum and minimum values.

2) Action space: The control actions are the steering angle and the throttle, denoted as:

$$A = \{\delta, \tau\}. \tag{10}$$

In CARLA [33], the steering angle is normalized within the range of $[-1, 1]$, and the throttle within $[0, 1]$. To enhance training efficiency, we further constrain the throttle within the range of $[0.6, 1]$. Through experimentation, we have discovered that employing the unfiltered steering command directly from the RL agent results in the vehicle frequently swaying from side to side. This observation aligns with a similar finding presented in [34]. To achieve a smoother motion, the actions are first passed through a low-pass filter before being executed by the vehicle.

3) Reward function: The reward function plays a crucial role as a signal that indicates the desirability of actions taken by the agent in specific states. It provides valuable feedback to the agent, enabling it to comprehend the consequences of its actions and adjust its behavior accordingly. In line with our objectives, we have appropriately formulated the high-speed drifting problem by introducing a novel reward function:

$$r = r_c + r_{e_y} + r_{e_\psi} + r_v + r_s. \tag{11}$$

The first part r_c is designed with a focus on safety. We aim to penalize the vehicle for colliding with road boundaries. CARLA is equipped with collision sensors to detect such incidents:

$$r_c := \begin{cases} 0 & \text{no collision} \\ -a & \text{collision} \end{cases} \tag{12}$$

In order to enable the vehicle to track the reference trajectory, r_{e_y} and r_{e_ψ} are designed separately. We motivate the vehicle to get as close as possible to the reference trajectory. Therefore, the position following reward is defined as:

$$r_{e_y} := \exp\left(-k_1 \left|e_y\right|\right). \tag{13}$$

The angle following reward is designed to penalize a large heading angle error, thereby promoting the vehicle to maintain a stable posture throughout its movement:

$$r_{e_\psi} := k_2 \exp\left(-k_3 \left|e_\psi\right|\right) \tag{14}$$

To facilitate high-speed driving, positive rewards are provided to the vehicle's speed v, measured in m/s, thereby incentivizing higher speeds in the agent's behavior. However, it is important to consider that the encouraged speed may have an upper limit due to the decrease in speed during drifting maneuvers.

Hence, the speed reward r_v encourages the vehicle to achieve higher speed, up to a predefined maximum threshold:

$$r_v := \begin{cases} k_4 \exp\left(-k_5 \left| v - b \right| \right) & \text{if } v < b \\ k_4 & \text{otherwise} \end{cases} \tag{15}$$

As previously mentioned, drifting is a cornering technique wherein the vehicle deliberately sustains a high sideslip angle while operating at high speeds. Simultaneously, the sideslip angle reflects the lateral stability of the vehicle [35]. Therefore, in straight roads, a large sideslip angle is not encouraged. Then the sideslip angel reward is defined as:

$$r_s := \begin{cases} k_6 \beta & |\kappa| > c_1 \\ -k_7 \beta & |\kappa| < c_2 \\ 0 & c_2 \leq |\kappa| \leq c_1 \end{cases}, \tag{16}$$

where κ represents the curvature value of the reference trajectory. The estimation of the value is performed by utilizing three waypoints and applying filtering processes [36].

4.2. Curriculum setup

Due to the presence of multiple objectives, training a controller for autonomous drifting directly poses significant challenges and can consume considerable time. In our specific context, we adopt a stage-based learning procedure for our agent inspired by CL [37]. CL is a training strategy in the context of machine learning. It involves exposing the model to progressively challenging data, beginning with simpler examples and gradually advancing to more complex ones. In comparison to direct training on the entire set of tasks, curriculum reinforcement learning has shown its effectiveness in improving convergence rate and achieving improved final performance [38–40].

We divide the training procedure into three stages based on the different objectives involved, such that the agent is guided to learn increasingly complex behaviors. In each stage, we maintain the action space, state space, and environment dynamics unchanged. However, the initial policy and reward functions are modified based on the specific task at hand.

- *Stage 1:* We initiate the training procedure by addressing the low-speed trajectory tracking task. In this task, the autonomous agent is trained to accurately track a reference trajectory at low speeds. The objective is to minimize deviations from the desired trajectory while ensuring heading stability. By emphasizing this initial stage, the agent acquires fundamental control skills, laying a solid foundation for more complex maneuvers in subsequent stages. The reward function defined for this stage is as follows:

$$r_1 = r_c + r_{e_y} + r_{e_\psi}. \tag{17}$$

- *Stage 2:* We shift our focus to the high-speed path following task, in which the agent is trained to track the path at increased speeds. This stage introduces higher velocities and more dynamic challenges compared to the initial stage. To effectively accomplish this task, the agent's control system must demonstrate the capability to execute actions with both precision and responsiveness. A defined reward function is utilized for this stage to provide guidance and assess the agent's performance:

$$r_2 = r_c + r_{e_y} + r_{e_\psi} + r_v. \tag{18}$$

- *Stage 3:* In the concluding stage, we delve into the high-speed drifting task. The agent is trained to perform precise and controlled drift maneuvers at high speeds. Drifting involves intentionally inducing and managing oversteer, enabling the vehicle to slide laterally while maintaining control. Furthermore, we impose stringent criteria on the vehicle's lateral stability, emphasizing

Table I. *Standard deviation of applied gaussian observation noise.*

Observation	Standard deviation
Vehicle positions	± 0.02m
Vehicle orientations	± 0.003 rad
Vehicle velocities	± 0.03 m/s

its ability to sustain control and stability. In this stage, the reward function r encompasses all the rewards defined throughout the training procedure. It is designed to provide positive reinforcement for the agent's successful execution of precise and controlled drift maneuvers at high speeds while maintaining control and stability.

When it comes to knowledge transfer, policy transfer is employed to enable the agent to leverage the knowledge acquired from one intermediate task and apply it to another task [41]. This approach allows for the reuse and transfer of learned policies, enabling the agent to build upon previously acquired knowledge and skills as it progresses through different stages of the curriculum. Specifically, the initial stage starts with a randomized policy, indicating a lack of prior knowledge or specific strategy. In the subsequent stages, the initial policy selection is based on the training outcomes from the previous stage. Correspondingly, the parameters related to policy exploration undergo changes when transitioning between stages.

4.3. Randomization

In order to enhance the robustness and generalization ability of the controller, domain randomization is incorporated into the training procedure. Domain randomization involves the introduction of random variations and perturbations into the training environment or simulation to generate a diverse set of training scenarios [42–45]. Specifically, we introduce randomization to the sensor observation noise, actuator action noise, and physical parameters.

1) Observation noise: To faithfully simulate the noise encountered in real-world scenarios, gaussian noise is introduced to the sensor observations. The magnitude of the noise is determined based on the achieved accuracy through Precise Point Positioning (PPP) within a Real-Time Kinematic (RTK) network. The specific noise levels are detailed in Table I.

2) Action noise: We introduce gaussian noise to all actions to simulate an inaccurate actuation system. In addition, we achieve policy exploration by modifying the standard deviation of the noise. As the training process advances, the standard deviation of the gaussian noise linearly decreases. This approach allows the agent to explore a diverse set of actions while steadily improving its control strategy.

3) Physics randomization: The physical parameters are sampled at the beginning of each episode and remain constant throughout the entire episode. Randomization is introduced by using the test baseline values as the reference point. Table II provides a comprehensive list of these randomized values. These particular parameters are chosen due to their substantial influence on the dynamic performance of the vehicle.

5. Experiments

5.1. Experimental setup

We conduct our experiments using CARLA [33], an open-source autonomous driving simulator known for its flexible environment and realistic physics. We select the "Audi A2" as the simulated vehicle model. As shown in Figure 5, A map featuring corners with varying levels of difficulty for turns serves as the reference path.

Table II. *Ranges of physics parameter randomizations.*

Parameter	**Scaling factor range**
Vehicle mass	Uniform([0.8, 1.2])
Tire friction	Uniform([0.8, 1.2])
Center of mass in x direction	Uniform([0.95, 1.05])
Center of mass in z direction	Uniform([0.95, 1.05])

Table III. *Hyperparameters.*

Hyperparameter	**Value**
Iteration size	2048
Num. epochs	10
Optimizer	Adam
Policy network learning rate	0.0003
Value network learning rate	0.001
Discount factor (γ)	0.99
Clipping parameter (ϵ)	0.2
GAE discount (λ)	0.95

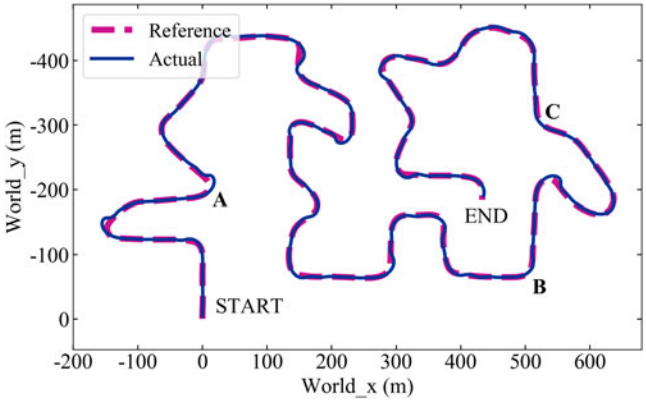

Figure 5. *Measured vs. reference path for experimental run.*

The policy is trained using the PPO algorithm with GAE. The PPO agent is built upon the actor-critic framework, where both the policy network and the value function network are constructed with two hidden layers. Each hidden layer consists of 64 units and utilizes the activation function *tanh*. The hyperparameters for the training are listed in Table III. The training is conducted on a desktop equipped with a Xeon E3-1275 CPU and a RTX 2080 GPU.

5.2. Drifting performance

1) Quantitative results: To demonstrate the effectiveness of our approach, we conduct experiments using vehicles with various combinations of tire friction (F) and mass (M). The baseline for both training and testing is set to $F3.5M1.8$, indicating a friction coefficient of 3.5 for all four tires and a vehicle mass of 1800 kg. We compare our results with the algorithm proposed by Cai et al. [16]. To quantitatively evaluate the performance, the following metrics are employed:

Table IV. *Quantitative evaluation for different methods. ↓ means smaller numbers are better. ↑ means larger numbers are better.*

		C.T.E.↓	**H.A.E.↓**	**MAX-V↑**	**AVG-V↑**	**MAX-S↑**	**AVG-S-S↓**	**AVG-S-C↑**	**SMOY↓**	**SMOS↓**
Setup	**Methods**	**(m)**	**(°)**	**(km/h)**	**(km/h)**	**(°)**	**(°)**	**(°)**		
F2.8M1.8	Cai	-	-	100.24	74.12	**26.23**	3.30	**11.32**	18.082	0.099
	PPO	1.71	8.12	103.62	73.37	22.28	2.92	8.24	9.134	0.033
	CL-PPO	**1.32**	**7.05**	**105.02**	**74.57**	20.58	**2.86**	8.44	**9.062**	**0.031**
F3.5M1.8	Cai	-	-	97.84	74.87	**25.77**	3.38	**9.72**	21.552	0.102
	PPO	1.24	7.78	104.08	76.22	19.80	**2.20**	7.05	10.799	0.034
	CL-PPO	**1.08**	**5.94**	**104.89**	**76.68**	20.47	2.72	7.77	**9.129**	**0.030**
F3.5M1.5	Cai	-	-	102.63	**79.37**	**29.30**	3.52	**10.00**	21.139	0.101
	PPO	1.69	8.40	110.64	78.92	19.40	2.98	8.19	10.600	0.033
	CL-PPO	**1.30**	**7.05**	**111.84**	79.34	18.72	**2.71**	8.18	**9.873**	**0.032**

- *C.T.E.* represents the average cross-track error, which measures the deviation between the vehicle's actual position and the desired path in a test. Similarly, *H.A.E.* represents the average heading angle error.
- *MAX-V* and *AVG-V* represent the maximum velocity and the average velocity during a test, respectively. In our context, achieving higher velocities is encouraged.
- *MAX-S* represents the maximum slip angle achieved by the vehicle. *AVG-S-S* and *AVG-S-C* represent the average slip angle during straight roads and sharp corners, respectively. On straight roads, a small slip angle is desirable as it indicates good lateral stability. Conversely, in sharp corners, a larger slip angle is encouraged as it allows for exploration of the vehicle's control limits.
- *SMOY* and *SMOS* are used to evaluate the smoothness of the vehicle's yaw rate and steering, respectively. To calculate these metrics, rolling standard deviations are computed independently for each of them, using a rolling window size of 5.

We evaluate the performance of each trained agent by calculating the average metrics across 10 repeated experiments. The testing results are summarized in Table IV. It is important to note that we refrained from conducting error comparisons with the approach proposed by Cai et al. due to the disparities in reference trajectories.

In all setups, our agent (CL-PPO) achieves the highest *MAX-V*, reaching a maximum of 111.84 km/h. When it comes to slip angle, our agent achieves a maximum of 20.58°, which is slightly lower than Cai's. Nevertheless, our agent demonstrates an advantage in stabilizing vehicle sideslip, as evidenced by smaller slip angles on straight roads. Additionally, our agent consistently achieves smaller values for *SMOY* and *SMOS* across all experimental setups. This indicates that our agent excels in maintaining the vehicle's lateral stability and steerability. Overall, these results highlight the effectiveness of our agent in enhancing the performance and stability of the vehicle.

The experiments conducted on various combinations of vehicle mass and tire friction demonstrate the adaptability of our intelligent agent. It is evident that reducing the mass and friction coefficient both lead to increased velocities and slip behavior. However, they also introduce challenges in control, resulting in larger tracking errors. Fortunately, our agent exhibits excellent performance in maintaining vehicle stability.

To evaluate the robustness of two agents, we present the key data from 10 experiments in Figure 6. The same conclusion can be drawn that our agent achieves higher velocity and smaller slip angle. The concentration of data indicates that our algorithm exhibits better robustness. This can be attributed to the smoothness and stability of our control.

2) Qualitative results: Figure 7 visualizes the performance of the learned agents in the setting $F3.5M1.8$, supplemented with crucial measurements. In order to showcase the adaptability of our

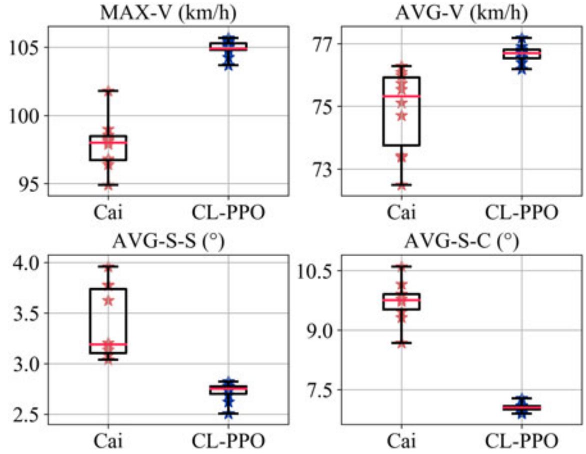

Figure 6. *Evaluation comparisons for the setting $F3.5M1.8$.*

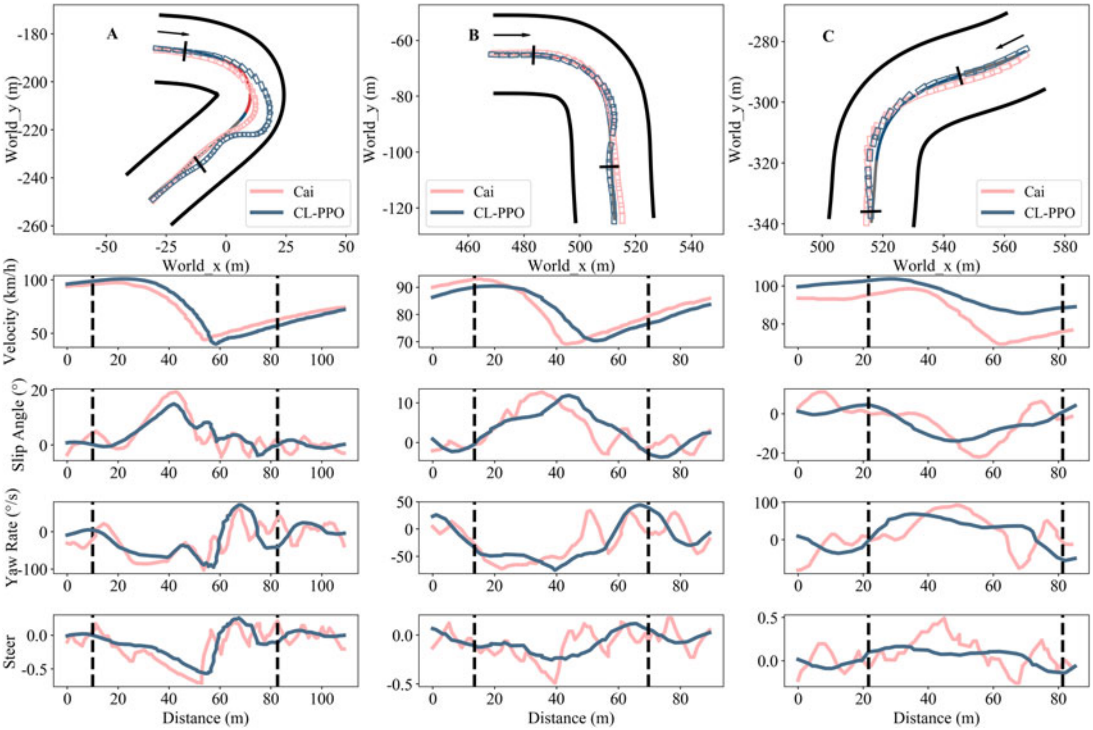

Figure 7. *Comparison of drifting maneuvers between our agent and the Cai's. The dashed lines serve as the starting and ending line for drifting maneuver. A negative number of the steering command means a right turn.*

agent to diverse road scenarios, three distinct locations in Figure 5 are chosen. A, B, and C represent sharp-angle bends, right-angle bends, and obtuse-angle bends, respectively.

The top plots depict the vehicle's motion path in relation to the reference path. It is evident that our agent successfully tracks the reference path closely in scenarios B and C. However, in scenario A, the learned agent exhibits a wider deviation in both position and heading from the reference path.

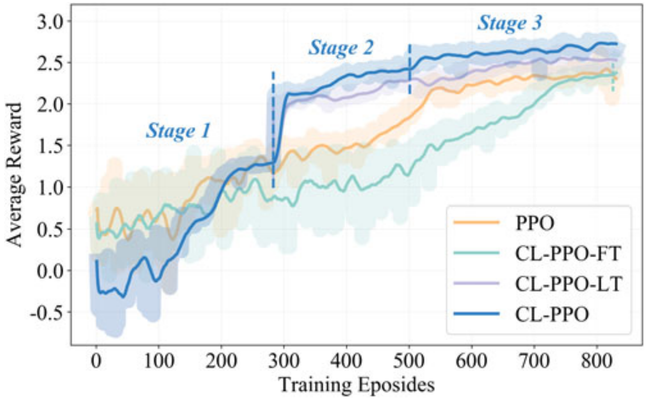

Figure 8. *Comparison of different training methods. The dashed lines represent the switching of training stages.*

This discrepancy can be attributed to the smaller turning radius, which results in a higher centrifugal force and more significant vehicle sliding.

The velocity subplot reveals that our agent surpasses Cai's by achieving higher velocities, particularly in scenario C. Our analysis of the velocity and slip angle curves aligns with established findings in the field. During drifting, as the slip angle gradually increases, a greater lateral force is required to maintain dynamic stability. However, due to force saturation on the rear wheels, the longitudinal force diminishes, resulting in a decrease in vehicle velocity. In scenarios A and B, we observe a relatively higher velocity decay, which can be attributed to the extended drifting process over a longer distance.

Drifting maneuvers involve significant variations in yaw rate, indicating the engagement of vehicles in extreme control actions. Analysis of the steering data reveals that in scenario A, characterized by a smaller turning radius, the steering command is increased intentionally to induce oversteer, thereby facilitating the vehicle's drifting. Conversely, even slight variations in steering command have a notable impact on the yaw rate in scenarios B and C. As the vehicle exits the drifting maneuvers, the steering command returns to its neutral position.

In summary, our agent demonstrates the ability to effectively control the vehicle during drifting maneuvers across various bends. The variations in vehicle states and control align with observations derived from the analysis of high sideslip maneuvers performed by professional drivers [46]. Moreover, compared to Cai's results, our approach exhibits smoother vehicle states and improved drifting behavior.

5.3. Curriculum learning

To evaluate the impact of CL, we employ different training methods for the drifting policy. In the case of CL-PPO, we follow a three-stage training approach. In contrast, the policy trained without CL begins training directly with the ultimate rewards. For comparison purposes, two additional experimental setups are devised: CL-PPO-FT, where the first two stages are trained jointly, and CL-PPO-LT, where the last two stages are trained jointly. It is crucial to load the pretrained policy from the previous stage and reinitialize the weights of the exploration term before commencing each new training stage.

Figure 8 shows the learning curves of different methods. Comparing CL-PPO and CL-PPO-FT highlights the importance of dividing the training into Stage 1 and Stage 2. Starting from the second stage, CL-PPO demonstrates higher rewards compared to CL-PPO-LT, showcasing the significance of dividing Stage 2 and Stage 3. In general, CL-PPO exhibits a relatively faster convergence rate and achieves higher total rewards compared to PPO. Additionally, the data in Table IV demonstrates that our method exhibits certain improvements in terms of final performance compared to PPO.

Table V. *Quantitative evaluation for different stages. ↓ means smaller numbers are better. ↑ means larger numbers are better.*

Setup	C.T.E.↓ (m)	H.A.E.↓ (°)	MAX-V↑ (km/h)	AVG-V↑ (km/h)	MAX-S↑ (°)	AVG-S-S↓ (°)	AVG-S-C↑ (°)	SMOY↓	SMOS↓
Stage 1	**0.18**	**2.60**	45.23	42.70	5.26	**0.24**	2.70	**1.441**	**0.008**
Stage 2	0.64	4.78	95.46	72.80	18.58	1.94	7.06	10.662	0.037
Stage 3	1.38	6.79	**106.60**	**77.20**	**19.43**	2.22	**7.56**	8.240	0.027

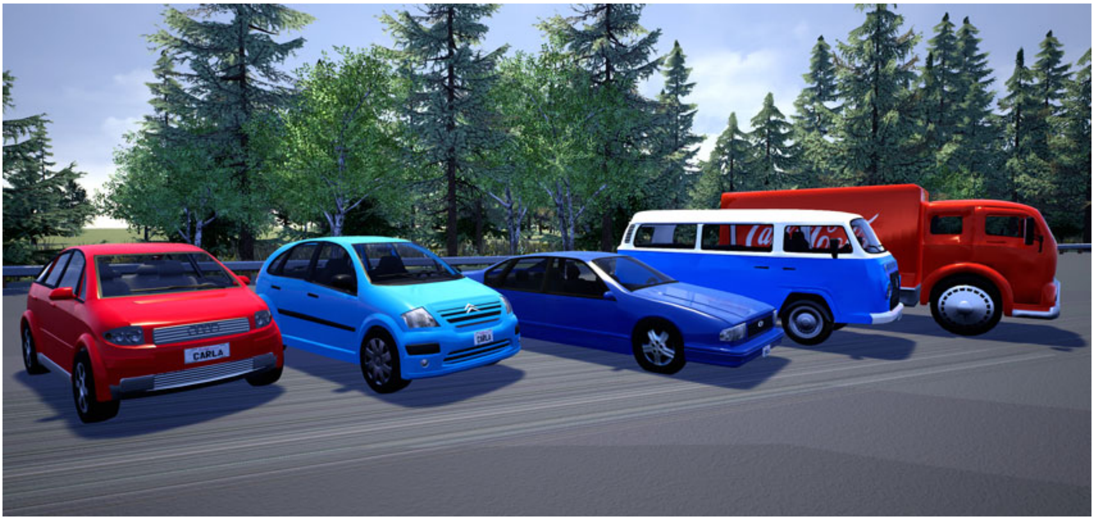

Figure 9. *Different vehicles for generalization ability test.*

To demonstrate the validity of our three-stage training design, which progressively approximates the overall task objective, we tested the policies obtained from each stage. The test results are shown in Table V. As can be seen, the first stage achieves excellent trajectory tracking, characterized by minimal deviation from the reference trajectory and exhibits superior vehicle stability. The results from the second and third stages show significant improvements in speed and slip angle compared to the first stage. However, the third stage outperformed the second stage in terms of speed, slip angle, and vehicle stability. With the progression of training, the performance gradually improved, further demonstrating the validity of our curriculum design.

5.4. Generalization analysis

1) Unseen vehicle types: To evaluate the generalization ability of the proposed controller, a variety of vehicles are selected for testing, as shown in Figure 9. The corresponding physical parameters are presented in Table VI. In the table, “RPM” stands for “Revolutions Per Minute,” which refers to the number of times the engine’s crankshaft rotates in one minute. When comparing the Citroen C3 to the benchmark Audi A2, both vehicles have a comparable size, but the Citroen C3 stands out with a lower mass and maximum RPM. On the other hand, the Chevrolet Impala, a full-sized sedan, has a longest chassis, resulting in an expanded turning radius that presents a significant steering challenge. The Volkswagen T2, characterized as a van model, has a restricted maximum RPM, which limits its acceleration capability. Lastly, “Carlacola” is a truck model featured within Carla, known for its notable physical dimensions and substantial mass. Consequently, it poses higher demands on vehicle control and state stability.

Table VI. *Different vehicles with their respective physical parameters for generalization ability test.*

Vehicle	**Audi A2**	**Citroen C3**	**Chevrolet Impala**	**Volkswagen T2**	**Carlacola**
Suspension	urban	urban	urban	van	truck
Length (m)	3.72	3.98	5.37	4.52	5.20
Width (m)	1.68	1.69	1.83	1.73	2.62
Height (m)	1.54	1.61	1.42	2.03	2.48
Tire friction	3.50	3.50	3.50	3.50	3.50
Mass (t)	1.80	1.00	1.50	2.30	5.50
Max RPM	10,000	5730	5730	2000	5730

Table VII. *Quantitative evaluation for different vehicles. ↓ means smaller numbers are better. ↑ means larger numbers are better.*

	C.T.E.↓	**H.A.E.↓**	**MAX-V↑**	**AVG-V↑**	**MAX-S↑**	**AVG-S-S↓**	**AVG-S-C↑**	**SMOY↓**	**SMOS↓**
Vehicle	**(m)**	**(°)**	**(km/h)**	**(km/h)**	**(°)**	**(°)**	**(°)**		
Audi A2	1.24	7.78	104.89	76.68	20.47	2.72	7.05	10.799	0.034
Citroen C3	1.90	11.05	116.78	78.63	27.49	4.26	9.75	10.169	0.036
Chevrolet Impala	1.28	9.19	106.37	74.71	21.25	3.60	8.87	9.461	0.034
Volkswagen T2	1.18	4.20	63.41	57.25	6.83	1.19	0.74	5.132	0.019
Carlacola	0.98	5.02	73.00	60.43	7.66	1.33	1.95	5.869	0.022

We conduct drifting tests on the identical path, and the results are presented in Table VII. Due to the general nature of our design, all tested vehicles can fully leverage their capabilities to accomplish drifting tasks without the need for reference vehicle states. The Citroen C3 achieves a maximum velocity of 116.78 km/h and a maximum slip angle of 27.49°. This performance can be attributed to its lower mass compared to other vehicles. The Chevrolet Impala demonstrates results comparable to the Audi A2, showcasing the effective adaptability of our controller to extended vehicle dimensions. The Volkswagen T2, characterized by its minimal maximum RPM, achieves superior *SMOY* and *SMOS*, despite its lower velocity and slip angle. When dealing with the large dimensions and substantial mass of "Carlacola," our controller also demonstrates notable effectiveness.

Experiments conducted with various vehicles demonstrate that our controller performs well when faced with vehicles having different combinations of size, mass, and maximum RPM, indicating strong generalization ability.

2) Unseen path: To demonstrate the generalization ability of our controller on previously unseen paths, we conduct tests on a new path with varying curvature, as shown in Figure 10. The vehicle used is configured as $F3.5M1.8$. The experimental results indicate that our controller follows the desired path successfully.

Figure 11 presents significant measurements obtained during the test. A maximum slip angle of 16.46° is observed at 25.3 s. Between 48 and 51 s, the vehicle engages in extended drifting, leading to a noticeable deviation from the desired path. Notably, the slip angle remains close to zero during long straight sections, indicating the effectiveness of our controller in maintaining lateral stability and minimizing vehicle slip. The measured velocity aligns with our previous analysis, exhibiting rapid deceleration upon entering a slip condition and quick acceleration during the exit from drifting. Throughout the test, the average velocity reaches 82.28 km/h, with a maximum of 119.82 km/h achieved. Remarkably, the steering command remains close to zero during straight path segments, showcasing excellent stability. Even during continuous turning between 30 and 50 s, our measurements show no significant oscillatory behavior. The effectiveness and stability demonstrated on the unseen path serve as validation of the generalization ability of our control framework.

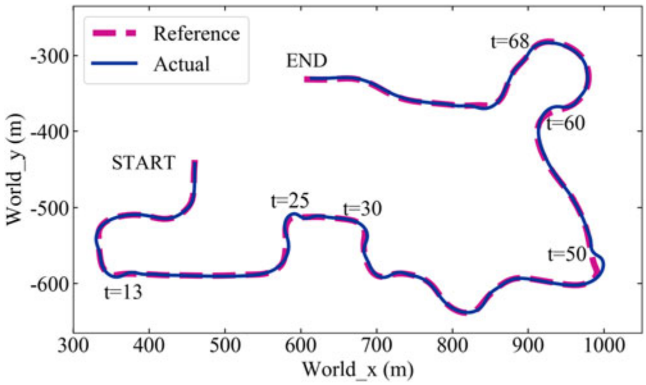

Figure 10. *Measured vs. reference path for generalization ability test.*

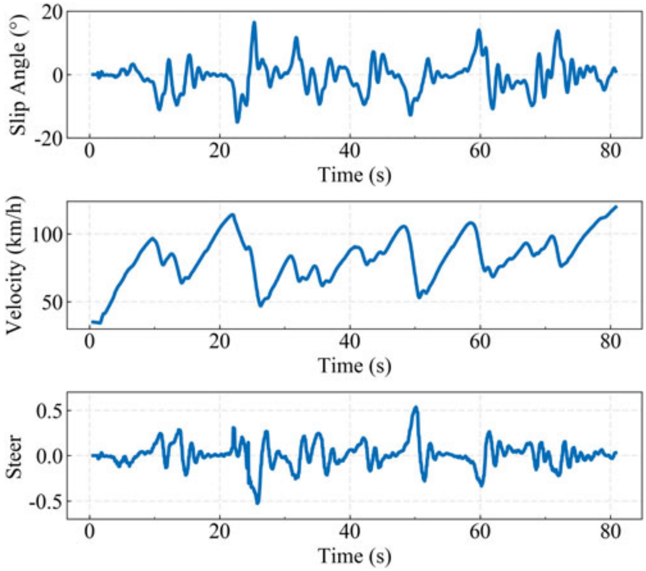

Figure 11. *Important measurements vs. time for generalization ability test.*

6. Conclusion

In this paper, a control framework based on curriculum reinforcement learning is designed to enable vehicles to perform high-speed driving and drifting maneuvers along general paths. The framework considers the dynamics of the vehicle and incorporates CL to break down the task into multiple stages of increasing difficulty. Additionally, we introduce domain randomization techniques to enhance the robustness of the learned policies. We validate the proposed framework in CARLA using various vehicle types and parameters. The experimental results demonstrate the effectiveness and generalization ability of our framework, as well as the beneficial impact of CL on improving the convergence rate and overall performance. In the future, we will implement and evaluate the simulation-to-reality policy on real vehicles, thus bridging the gap between simulation and real-world applications.

Author contributions. Kai Yu contributed to this research by participating in the design of the experimental methods, conducting the actual survey research, performing data analysis, visualizing the experimental results, and contributing to the writing of the manuscript. Mengyin Fu provided the research content and secured the funding support for the project. Xiaohui Tian contributed to the design of the methods and analysis of the results. Shuaicong Yang participated in the software design and experimental analysis. Yi Yang supervised and guided the research project and contributed to the review and revision of the manuscript.

Financial support. This work is partly supported by the National Natural Science Foundation of China (grant no. NSFC 61973034, 62233002, U1913203, 61903034 and CJSP Q2018229).

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

[1] X. Tian, S. Yang, Y. Yang, W. Song and M. Fu, "A multi-layer drifting controller for all-wheel drive vehicles beyond driving limits," *IEEE/ASME Trans. Mechatron*. **29**(2), 1–11 (2023). doi: 10.1109/TMECH.2023.3298660

[2] J. Y. Goh, M. Thompson, J. Dallas and A. Balachandran, "Beyond the stable handling limits: Nonlinear model predictive control for highly transient autonomous drifting," *Veh. Syst. Dyn*. **62**(10), 2590–2613 (2024).

[3] H. Dong, H. Yu and J. Xi, "Phase portrait analysis and drifting control of unmanned tracked vehicles," *IEEE Trans. Intell. Veh*. **10**(1), 1–16 (2024). doi: 10.1109/TIV.2024.3356608

[4] S. Inagaki, I. Kushiro and M. Yamamoto, "Analysis on vehicle stability in critical cornering using phase-plane method," *JSAE Rev*. **2**(16), 216–216 (1995).

[5] E. Ono, S. Hosoe, H. D. Tuan and S. Doi, "Bifurcation in vehicle dynamics and robust front wheel steering control," *IEEE Trans. Control Syst. Technol*. **6**(3), 412–420 (1998).

[6] C. Voser, R. Y. Hindiyeh and J. C. Gerdes, "Analysis and control of high sideslip manoeuvres," *Veh. Syst. Dyn*. **48**(S1), 317–336 (2010).

[7] E. Velenis, D. Katzourakis, E. Frazzoli, P. Tsiotras and R. Happee, "Steady-state drifting stabilization of rwd vehicles," *Control. Eng. Pract*. **19**(11), 1363–1376 (2011).

[8] J. Y. Goh, T. Goel and J. C. Gerdes, "Toward automated vehicle control beyond the stability limits: drifting along a general path," *J. Dyn. Sys., Meas., Control*. **142**(2), 1–13 (2020).

[9] F. Zhang, J. Gonzales, S. E. Li, F. Borrelli and K. Li, "Drift control for cornering maneuver of autonomous vehicles," *Mechatronics*. **54**(1), 167–174 (2018)

[10] G. Chen, X. Zhao, Z. Gao and M. Hua, "Dynamic drifting control for general path tracking of autonomous vehicles," *IEEE Trans. Intell. Veh*. **8**(3), 2527–2537 (2023). doi: 10.1109/TIV.2023.3235007

[11] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction* (MIT Press, USA, 2018).

[12] L. Rose, M. C. Bazzocchi and G. Nejat, "A model-free deep reinforcement learning approach for control of exoskeleton gait patterns," *Robotica*. **40**(7), 2189–2214 (2022).

[13] D. Zhang, R. Ju and Z. Cao, "Reinforcement learning-based motion control for snake robots in complex environments," *Robotica*. **42**(4), 947–961 (2024).

[14] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," *Nature*. **620**(7976), 982–987 (2023).

[15] F. Domberg, C. C. Wembers, H. Patel and G. Schildbach, "Deep drifting: Autonomous drifting of arbitrary trajectories using deep reinforcement learning," *2022 IEEE International Conference on Robotics and Automation (ICRA)*, Philadelphia, USA (IEEE, 2022) pp. 7753–7759

[16] P. Cai, X. Mei, L. Tai, Y. Sun and M. Liu, "High-speed autonomous drifting with deep reinforcement learning," *IEEE Robot. Autom. Lett*. **5**(2), 1247–1254 (2020).

[17] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor and P. Stone, "Curriculum learning for reinforcement learning domains: a framework and survey," *J. Mach. Learn. Res*. **21**(1), 7382–7431 (2020).

[18] J. Edelmann and M. Plöchl, "Handling characteristics and stability of the steady-state powerslide motion of an automobile," *Regul. Chaotic Dyn*. **14**(6), 682–692 (2009).

[19] R. Y. Hindiyeh and J. C. Gerdes, "Equilibrium analysis of drifting vehicles for control design," *Dynamic Systems and Control Conference*, Philadelphia, USA (2009) pp. 181–188.

[20] M. Baars, H. Hellendoorn and M. Alirezaei, "Control of a scaled vehicle in and beyond stable limit handling," *IEEE Trans. Veh. Technol*. **70**(7), 6427–6437 (2021).

[21] M. Khan, E. Youn, I. Youn and L. Wu, "Steady state drifting controller for vehicles travelling in reverse direction," *2018 15th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, Islamabad, Pakistan (IEEE, 2018) pp. 263–268.

[22] M. T. Peterson, T. Goel and J. C. Gerdes, "Exploiting linear structure for precision control of highly nonlinear vehicle dynamics," *IEEE Trans. Intell. Veh*. **8**(2), 1852–1862 (2022).

[23] G. Bellegarda and Q. Nguyen, "Dynamic vehicle drifting with nonlinear mpc and a fused kinematic-dynamic bicycle model," *IEEE Contr. Syst. Lett*. **6**(1), 1958–1963 (2021)

[24] C. Hu, X. Zhou, R. Duo, H. Xiong, Y. Qi, Z. Zhang and L. Xie, "Combined fast control of drifting state and trajectory tracking for autonomous vehicles based on mpc controller," *2022 IEEE International Conference on Robotics and Automation (ICRA)*, Philadelphia, USA *(IEEE*, 2022) pp. 1373–1379.

[25] T. P. Weber and J. C. Gerdes, "Modeling and control for dynamic drifting trajectories," *IEEE Trans. Intell Veh*. **9**(2), 1–11 (2023). doi: 10.1109/TIV.2023.3340918

[26] M. Acosta and S. Kanarachos, “Teaching a vehicle to autonomously drift: a data-based approach using neural networks,” *Knowl. Based. Syst*. **153**(1), 12–28 (2018)

[27] X. Zhou, C. Hu, R. Duo, H. Xiong, Y. Qi, Z. Zhang, H. Su and L. Xie, “Learning-based mpc controller for drift control of autonomous vehicles,” *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, Macau, China *(IEEE*, 2022) pp. 322–328.

[28] M. Cutler and J. P. How, “Autonomous drifting using simulation-aided reinforcement learning,” *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden (IEEE, 2016) pp. 5442–5448.

[29] R. Y. Hindiyeh. *Dynamics and Control of Drifting in Automobiles* (Stanford University, USA, 2013)

[30] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi and R. Mangharam, “Autonomous vehicles on the edge: A survey on autonomous vehicle racing,” *IEEE O J Intell. Transp. Syst*. **3**(1), 458–488 (2022)

[31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O. Klimov, “Proximal policy optimization algorithms,” arXiv: 1707.06347, 1-12 (2017)

[32] J. Schulman, P. Moritz, S. Levine, M. Jordan and P. Abbeel, “High dimensional continuous control using generalized advantage estimation,” arXiv: 1506.02438, 1-14 (2015)

[33] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez and V. Koltun, “Carla: An open urban driving simulator,” *Conference on Robot Learning*, California, USA (PMLR, 2017) pp. 1–16.

[34] L. Chen, Y. He, Q. Wang, W. Pan and Z. Ming, “Joint optimization of sensing, decision-making and motion-controlling for autonomous vehicles: A deep reinforcement learning approach,” *IEEE Trans. Veh. Technol*. **71**(5), 4642–4654 (2022).

[35] W. Sun, X. Wang and C. Zhang, “A model-free control strategy for vehicle lateral stability with adaptive dynamic programming,” *IEEE Trans. Ind. Electron*. **67**(12), 10693–10701 (2019).

[36] I. Bae, J. H. Kim and S. Kim, “Steering rate controller based on curvature of trajectory for autonomous driving vehicles,” *2013 IEEE Intelligent Vehicles Symposium (IV)*, Gold Coast City, Australia (IEEE, 2013) 1381–1386.

[37] Y. Bengio, J. Louradour, R. Collobert and J. Weston, “Curriculum learning,” *Proceedings of the 26th Annual International Conference on Machine Learning*, Montreal, Canada (2009) pp. 41–48.

[38] N. Rudin, D. Hoeller, P. Reist and M. Hutter, “Learning to walk in minutes using massively parallel deep reinforcement learning,” *Conference on Robot Learning*, Auckland, New Zealand (PMLR, 2022). 91–100.

[39] H.-C. Wang, S.-C. Huang, P.-J. Huang, K.-L. Wang, Y.-C. Teng, Y.-T. Ko, D. Jeon and I.-C. Wu, “Curriculum reinforcement learning from avoiding collisions to navigating among movable obstacles in diverse environments,” *IEEE Robot. Autom. Lett*. **8**(5), 2740–2747 (2023).

[40] D. Hoeller, N. Rudin, D. Sako and M. Hutter, “Anymal parkour: Learning agile navigation for quadrupedal robots,” *Sci. Robot*. **9**(88), eadi7566 (2024).

[41] X. Wang, Y. Chen and W. Zhu, “A survey on curriculum learning,” *IEEE Trans. Pattern Anal. Mach. Intell*. **44**(9), 4555–4576 (2022).

[42] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada (IEEE, 2017) pp. 23–30.

[43] X. B. Peng, M. Andrychowicz, W. Zaremba and P. Abbeel, “Sim-to-real transfer of robotic control with dynamics randomization,” *2018 IEEE International Conference on Robotics and Automation (ICRA),* Brisbane, Australia (IEEE, 2018) pp. 3803–3810.

[44] Y. Chen, C. Zeng, Z. Wang, P. Lu and C. Yang, “Zero-shot sim-to-real transfer of reinforcement learning framework for robotics manipulation with demonstration and force feedback,” *Robotica*. **41**(3), 1015–1024 (2023).

[45] R. Xiao, C. Yang, Y. Jiang and H. Zhang, “One-shot sim-to-real transfer policy for robotic assembly via reinforcement learning with visual demonstration,” *Robotica*. **42**(4), 1074–1093 (2024).

[46] R. Y. Hindiyeh and J. C. Gerdes, “A controller framework for autonomous drifting: Design, stability, and experimental validation,” *J. Dyn. Sys., Meas., Control*. **136**(5), 051015 (2014).

Cite this article: K. Yu, M. Fu, X. Tian, S. Yang and Y. Yang (2024). “Curriculum reinforcement learning-based drifting along a general path for autonomous vehicles”, Robotica **42**, 3263–3280. https://doi.org/10.1017/S026357472400119X