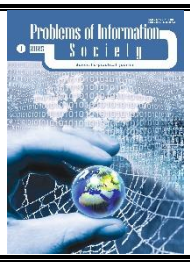

www.jpis.az

Comparative analysis of double deep q-network and proximal policy optimization for lane-keeping in autonomous driving

Ariful Islam Sabbir

School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, China

027121125@sues.edu.cn

ARTICLE INFO

http://doi.org/10.25045/jpis.v16.i1.02

Article history:

Received 02 September 2024
Received in revised form
05 November 2024
Accepted 06 January2025

Keywords:

Autonomous Driving Lane-Keeping
Reinforcement Learning Double Deep
Q-Network Proximal Policy
Optimization
Action Space
Proximal Policy Optimization

ABSTRACT

Lane-keeping is a vital function in autonomous driving, important for vehicle safety, stability, and adherence to traffic flow. The intricacy of lane-keeping control resides in balancing precision and responsiveness across varied driving circumstances. This article gives a comparative examination of two reinforcement learning (RL) algorithms—Double Deep Q-Network and Proximal Policy Optimization—for lane-keeping across discrete and continuous action spaces. Double DQN, an upgrade of standard Deep Q-Networks, eliminates overestimation bias in Q-values, demonstrating its usefulness in discrete action spaces. This method shines in low-dimensional environments like highways, where lane-keeping requires frequent, discrete modifications. In contrast, PPO, a strong policy-gradient method built for continuous control, performs well in high-dimensional situations, such as urban roadways and curved highways, where continual, accurate steering changes are necessary. The methods were tested in MATLAB/Simulink simulations that simulate both highway and urban driving circumstances. Each model integrates vehicle dynamics and neural network topologies to build control techniques. Results demonstrate that Double DQN consistently maintains lane position in highway settings, exploiting its ability to minimize overestimations in Q-values, thereby attaining stable lane centering. PPO outshines in dynamic and unpredictable settings, managing continual control adjustments well, especially under difficult traffic conditions and on curving roadways. This study underscores the importance of matching RL algorithms to the action-space requirements of specific driving environments, with Double DQN excelling in discrete tasks and PPO in continuous adaptive control, contributing valuable insights toward enhancing the flexibility and safety of autonomous vehicles.

1. Introduction

Fast development with regard to autonomous driving systems has pressingly demanded the need for line-keeping assistance solutions that are robust in nature. LKA is one of the most important features within autonomous driving; it keeps the vehicle in its lane through constant changes in steering. This feature is very important since it allows smooth and safe driving, minimizing the likelihood of unexpected lane exits, especially at places where there are high speeds or congested traffic.

This research is focused on the design and evaluation of reinforcement learning algorithms for efficient keeping-assistance in autonomous driving, including Double Deep Q-Network (Jayakody, 2024) for a discrete action context and Proximal Policy Optimization (PPO) in continuous control

(https://spinningup.openai.com/en/latest/algorithms/ppo.html, 2024). We should be able to find out the best reinforcement learning approach to line-keeping tasks that can enhance the safety, stability, and responsiveness of an autonomous vehicle by investigating the performance of each algorithm under different road and traffic conditions.

2. Related works

In the article, a novel generation of policy gradient methods for reinforcement learning is introduced that alternate between optimizing a "surrogate" objective function applying stochastic gradient ascent and sampling data through interaction with the environment. A unique objective function is suggested that allows many epochs of minibatch updates, in contrast to typical policy gradient approaches that only do one gradient update per data sample. The new methods, which are referred to as proximal policy optimization (PPO), are more straightforward to apply, more general, and have a better sample complexity (empirically), although they share some advantages with trust region policy optimization (TRPO). The experiments evaluate PPO on a set of benchmark tasks, such as playing Atari games and simulating robotic locomotion. It is estimated that PPO performs better than alternative online policy gradient approaches and, on average (Schulman et al., 2017), achieves a good balance between sample complexity, simplicity, and wall-time.

Improvements in reinforcement learning (RL) require enormous computational resources and maintain notably sample inefficient. The human brain, on the other hand, can effectively acquire efficient control strategies with less resources. The topic of whether present RL methods may be enhanced by applying insights from neuroscience is therefore brought up. A well-liked theoretical framework called predictive processing holds that the human brain actively works to reduce surprise. It is demonstrated how to minimize surprise and achieve significant increases in cumulative reward by utilizing recurrent neural networks that anticipate their own sensory states. In particular, the Predictive Processing Proximal Policy Optimization (P4O) agent is introduced, an actor-critic reinforcement learning agent that integrates a world model in its hidden state and uses predictive processing to a recurrent form of the PPO algorithm. On several Atari games with a single GPU, P4O performs noticeably better than a baseline recurrent implementation of the PPO algorithm, even without hyperparameter adjustment. Additionally, given the same wall-clock time, it performs better than other state-of-the-art agents and surpasses human gamers' performance on a number of games, including Seaquest, which is a very difficult setting in the Atari realm. Overall, the research highlights how knowledge from the neuroscience community could assist to develop more powerful and effective artificial agents (Küçükoğlu et al., 2024).

Applying a surrogate objective function to limit the step size at each policy update is one of the main features of proximal policy optimization (PPO), which has produced advanced results in policy search, an area of reinforcement learning. The algorithm still experiences performance instability and optimization inefficiency due to the abrupt flattening of the curve, despite the usefulness of this constraint. The use of a functional clipping approach rather than a flat clipping method is a crucial improvement of our innovative functional clipping policy optimization algorithm, Proximal Policy Optimization Smoothed Algorithm (PPOS), which is presented to address this problem. By comparing the methodology with PPO and PPORB, which uses a rollback clipping mechanism, it is shown that it can perform more accurate updates than existing PPO methods. In difficult continuous control tasks, it is demonstrated that it works better than the most recent PPO variations in terms of both performance and stability. Additionally, a helpful guideline for adjusting our algorithm's hyperparameter is offered (Zhu et al., 2021).

A more extensive perspective on Path Integral Control (PIC) techniques is provided in this work. In the context of Linearly Solvable Optimal Control (LSOC), a limited subset of nonlinear Stochastic Optimal Control (SOC) issues, PIC refers to a specific class of policy search techniques. This class is distinct in that a formal optimal state trajectory distribution can be obtained through explicit solution. In this article, initially PIC theory is reviewed the and related algorithms that are specifically designed for policy search are discussed. By reducing the cross-entropy between the optimal and a state trajectory distribution parameterized by a parametric stochastic policy, it is possible to uncover a generic design strategy that depends on the presence of an optimal state trajectory distribution and determines a parametric policy. Motivated by this insight, the next goal is to establish a SOC problem that covers a less constrained class of problem formulations while sharing characteristics with the LSOC setting. This SOC problem is known as Entropy Regularized

Trajectory Optimization. Entropy Regularized Stochastic Optimal Control is a setting that has been the focus of the Reinforcement Learning (RL) community lately, and the problem is strongly related to it. The theoretical convergences of state trajectory distribution sequence are investigated, along with links to stochastic search methods intended for traditional optimization issues. In conclusion, for derivative-free trajectory optimization, explicit updates are derived and the inferred Entropy Regularized PIC is compared with previous work in the context of both PIC and RL (Akrour et al., 2018).

This article describes the implementation and analysis points of view for the Lane Keep Assist System of a passenger vehicle in a simulated environment using the MPC techniques, specifically Implicit and Explicit MPC. It is anticipated that the Model in Loop validation of control software development will benefit from the comparison of the closed loop performance of the aforementioned methodologies. This could facilitate and expedite its deployment on embedded technology, which is a crucial component of the Advanced Driver Assistance System's LKA feature. Using basic equations, the work develops a simplified model for a vehicle's lateral dynamics as a linear parameter variable state space model in the time domain over a speed range that is frequently observed. With the application of available toolboxes, the IMPC and EMPC controller actions are calculated by building and resolving optimization problems using the quadratic programming technique and the multi-parametric PWA solver approach, respectively. On a road profile with variable curvature, the tracking performance is simulated for the future lateral departure from the driving lane center line at a selected look-ahead distance. By implementing feed forward control action, which can be integrated through automated control or driver steering maneuvers, the MPC controllers are further enhanced. The two methods are qualitatively compared, and the trade-offs between them are discussed. It turns out that while IMPC is the best of all, both MPC methods produce acceptable performance requirements. The EMPC offers the further advantage of being feasible to build on a relatively low-end microcontroller, and thus could be taken into consideration for LKA system implementations that are cost-effective (Kamat, 2019).

This article aims to outline the process and analysis the application of two distinct approaches of Model Predictive Control implementation strategies to the lane-keeping function for a passenger vehicle in a virtual setting. It evaluates how well the methods function in the model-in-loop scenario, which may be an initial step towards the hardware-in-loop verification of MPC implementation and its actualization on the intended embedded hardware before it is deployed. It discusses the development of a simple model for an electrical power steering system and then models a vehicle's lateral dynamics as a linear parameter variable state space model in the time domain for a frequently used speed range. After that, the controller is synthesized using the two MPC approaches: (i) the quadratic programming approach, and (ii) explicit MPC utilizing the multi-parametric PWA solver found in Hybrid Toolbox©. On a road profile with variable curvature, simulations are run to monitor the vehicle's performance for lateral deviation from the driving lane centre line. The study finishes with comparison of the qualitative performance of the two approaches and the remarks on the trade-offs of them (S. Kamat et al., 2016). Model Predictive Control approaches for permanent magnet synchronous motor in virtual environment. 2016 IEEE 1st International Conference on Power Electronics Intelligent Control and Energy Systems (ICPEICES).

For the safety, stability, and compliance with traffic flow of autonomous vehicles, lane-keeping is an essential function. The complexity of lane-keeping control is in striking a balance between responsiveness and accuracy in a variety of driving situations. In this article, two reinforcement learning (RL) algorithms for lane-keeping over discrete and continuous action spaces—Double Deep Q-Network (Double DQN) and Proximal Policy Optimization (PPO)—are compared. Double DQN is an improvement on regular Deep Q-Networks that indicates promise in discrete action spaces by removing overestimation bias in Q-values. In low-dimensional settings, like as highways, where lane-keeping necessitates frequent, distinct adjustments, this approach excels. On the other hand, PPO, a robust policy-gradient technique designed for continuous control, operates well in high-dimensional scenarios where precise steering adjustments are required on a regular basis, including curved highways and urban roads. The methods were evaluated using MATLAB/Simulink simulations that mimic driving conditions on highways and in cities. In order to develop control strategies, each model combines neural network topologies with vehicle dynamics. The findings indicate that Double DQN exploits its capacity to reduce overestimations in Q-values to achieve stable

lane centering while reliably maintaining lane position in highway conditions. PPO excels in dynamic and uncertain environments, effectively handling constant control adjustments, particularly in challenging traffic situations and on winding roads. With Double DQN performing exceptionally well in discrete tasks and PPO in continuous adaptive control, this study highlights the significance of matching RL algorithms to the action-space requirements of particular driving environments. It also offers vital insights into improving the safety and flexibility of autonomous vehicles (Song et al., 2019).

Researchers have recently made substantial progress in integrating reinforcement learning with deep learning to learn feature representations. Using raw pixel data to train agents to play Atari games and raw sensory inputs to learn sophisticated manipulation abilities are two noteworthy examples. However, the absence of a widely accepted benchmark has made it challenging to measure advancements in the field of continuous control. In this work, a benchmark set of continuous control tasks is proposed, which includes tasks with incomplete observations, tasks with hierarchical structure, tasks with very high state and action dimensionality, such as 3D humanoid locomotion and classic tasks like cart-pole swing-up. New results are presented from a comprehensive analysis of several applied reinforcement learning algorithms. The benchmark and reference implementations are made available at this https URL to promote adoption by other researchers and to make experimental repeatability easier (Duan et al., 2016).

3. Materials and methods

3.1. About Double DQN and PPO optimal choices

Double Deep Q-Network (Double DQN) is chosen for discrete action spaces in autonomous driving due to its ability to avoid overestimation bias by isolating action selection from assessment, leading to more stable and precise Q-value computations needed for maintaining lane position. In contrast, Proximal Policy Optimization (PPO) is chosen for continuous action spaces because its clipping process (Chen et al., 2018) assures consistent policy updates, limiting dramatic changes that can interrupt smooth management. Together, Double DQN and PPO provide a comprehensive foundation for effective line-keeping in autonomous vehicles, tackling the constraints of both discrete and continuous action scenarios. Here, present a figure, Double DQN and PPO for Stable Action Selection and Policy Updates in Autonomous Driving-(fig. 1).

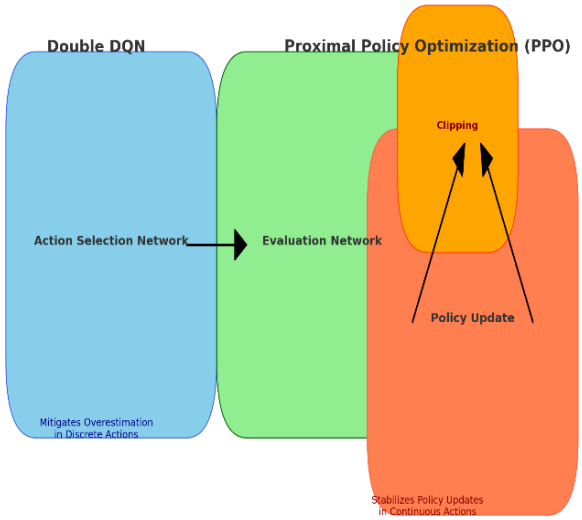

Fig.1. Double DQN and PPO for stability in autonomous driving

3.2. State Representation

Line-keeping in autonomous vehicles can be modeled as a control problem (Kamat, 2019) where the goal is to minimize lateral deviation from the lane center and maintain an optimal vehicle orientation angle.

Let the vehicle state at time t be represented by:

$$s_t = (y_t, \theta_t, v_t),$$

Where, $\boldsymbol{y_t}$ is the lateral deviation from the lane center, $\boldsymbol{\theta_t}$ is the orientation angle (yaw) relative to the lane direction, $\boldsymbol{v_t}$ is the vehicle's longitudinal velocity.

The control action $\boldsymbol{a_t}$ determines the steering angle $\boldsymbol{\delta}$ required to minimize the deviation:

Discrete Action (Double DQN): Action $\boldsymbol{a_t}$ is selected from a set of discrete steering angles, e.g, $\{-2^{\circ}, -1^{\circ}, 0^{\circ}, 1^{\circ}, 2^{\circ}\}$.

Continuous Action (PPO): Action $\boldsymbol{a_t}$ is chosen from a continuous range, $[-\delta_{max}, \delta_{max}]$, allowing final streeing adjustments.

The reinforcement learning (RL) model optimizes a reward function R_t to penalize lane deviations and abrupt streering actions (https://ww2.mathworks.cn/help/reinforcement-learning/ug/train-dqn-agent-for-lane-keeping-assist.html, 2024)

$$R_t = -\alpha|y_t| - \beta|\varphi_t| - \gamma|\delta_t| \quad (1)$$

Where, α, β, γ are weights that prioritize lane -centering, orientation stability, and minimal streering changes, respectively.

1. **Double DQN**: Double DQN updates Q-values by minimizing the Bellman error while

using a double estimator to reduce Q-value overestimation (Song et al., 2019). The value update is as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\big(r_t + \gamma Q(S_{t+1}, a_{max}) - Q(s_t, a_t)\big) \quad (2)$$(C. Wu,, et al, 2017)

Where, a_{max}= arg $max_{a'} Q(s_{t+1}, a')$ using the target network.

2. **PPO:** PPO applies a clipped objective for policy update, maintaining stability in continuous control (Zhu et al, 2021).The objective function is :

$$L(\theta) = E\left[\min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta old}(a|s)} A(s,a), clip\left(\frac{\pi_\theta(a|s)}{\pi_{\theta old}(a|s)}, 1-\epsilon, 1+\epsilon\right) A(s,a)\right)\right] \quad (3)$$ (W. Men at al, 2023)

Where $\boldsymbol{\pi_\theta(a|s)}$ is the policy, A(s,a) is the advantage function, and $\boldsymbol{\epsilon}$ is the a clipping parameter. Here a Conceptual Diagram of Line-Keeping Assistance in Autonomous Driving-control suitable for dynamic road conditions, such as curved lanes and urban driving. the comparative analysis of these reinforcement learning algorithms, we seek to identify the optimal method for line-keeping tasks across various driving scenarios, thereby enhancing the robustness of autonomous driving systems. We aim to ascertain Conceptual Diagram of Line-Keeping Assistance is shown in fig. 2:

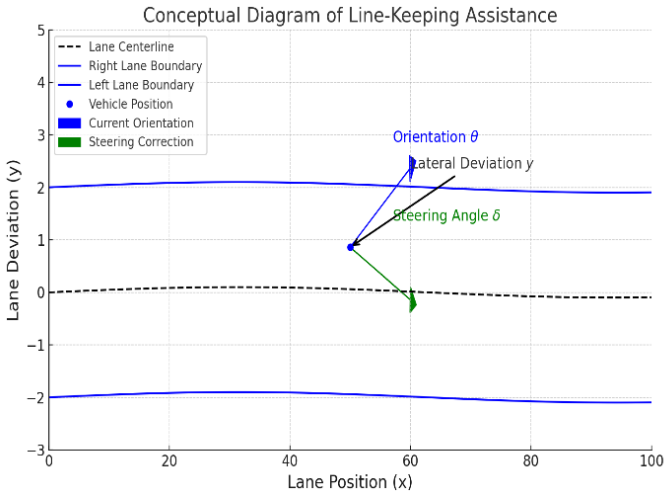

Fig.2: Conceptual Diagram of Line-Keeping Assistance (https://ww2.mathworks.cn/help/mpc/ug/lane-keeping-assist-system-using-model-predictive-control.html, 2024)

By implementing these algorithms, we aim to:

Double DQN: Enable effective discrete adjustments suitable for structured highway environments with minimal computational load.

PPO: Provide smoother, continuous steering 1. **Stability**: The efficacy of each approach in preserving lane centering with little variance over time.

2. **Adaptability**: The capacity of each algorithm to adjust to diverse road types and lane curvature.

3**. Efficiency**: Assessment of computational resources and real-time applicability of each method for practical autonomous driving situations.

This research seeks to determine the most effective reinforcement learning method for safe and efficient line-keeping, thereby advancing the overall development of autonomous vehicle safety systems.

Key components of the Dynamic Model in below:

1. Adapted State Variables:

✓ Lateral Deviation $\boldsymbol{y_t}$: The lateral distance from the vehicle's present position to the lane centerline,continuously updated.

✓ Yaw Angle $\boldsymbol{\theta_t}$: The angular deviation relative to the lane direaction, dynamically adapted based on real-time feedback.

✓ Curvature $\boldsymbol{k_t}$: A dynamic curvature value dependent on road geometry, computed using a forecast model for future road segments.

✓ Predicted Deviation $\boldsymbol{y_{t+1}}$ and Predicted Yaw Angle $\boldsymbol{\theta_{t+1}}$: Anticipated state values one step ahead, allowing the agent to preemptively adjust steering.

2. Dynamic Action Variable:

Steering Angle $\boldsymbol{\delta_t}$: The control output dynamically varies within a continuous range and adapts based on projected deviations.

3. Dynamic Reward Function:

The reward function is now adaptive, dynamically modifying weights based on road conditions (e.g, abrupt turns or straight roads) and traffic density. The incentive penalizes not just deviations but also rapid changes in steering,

offering smoother lane-keeping.

Mathematical Formulation of the Dynamic Model in below:

State Representation with Predictive Elements. To make the agent more anticipative, the state s_t at any time t includes both current and predicted deviations:

$$s_t = (y_t, \theta_t, k_t, y_{t+1}, \theta_{t+1})$$

Where, y_{t+1} and θ_{t+1} are the predicted lateral deviation and yaw angle at the next timestep, calculated based on current speed and road curvature.

Action Representation with Dynamic Range. The action $a_t = \delta_t$ is chosen within a dynamic range, where the maximum streering angle varies based on predicted curvature:

$$a_t \epsilon\, [-\delta_{max}(k_t), \delta_{max}(k_t)]$$

Where $\delta_{max}(k_t)$ increases for higher curvatures, allowing sharper turns when necessary.

Adaptive Reward Function. The reward R_t is now dynamically shaped based on curvature k_t or dense traffic D, penalizing deviations more heavily in these scenarios. The term $\delta_t - \delta_{t-1}$ penalizes abrupt streeing changes,promoting smooth adjustments.

Predictive Control for Future States. The predicted lateral deviation y_{t+1} and yaw angle θ_{t+1} are calculated based on the vehicle's velocity v_t and road curvature k_t:(W. Chen et al, 2016)]

$$y_{t+1} = y_t + v_t \sin(\theta_t)\, \Delta t \quad (4)$$

$$\theta_{t+1} = \theta_t + \frac{v_t \delta_t}{L}\, \Delta t \quad (5)$$

Where, Δt is the timestep duration. L is the vehicle's wheelbase, influencing the turning radius.

3.3. Dynamic Training Mechanism

1. Environment Simulation with Changing Conditions: The training environment dynamically alters road curvature and traffic density to mirror real-world settings, boosting the agent's adaptation to various conditions (Kothari et al, 2021).

2. Dynamic incentive Rescaling (Shi et al, 2024). The incentive system adapts based on the situation, placing a larger focus on lane adherence and smooth steering, especially in tough regions like sharp curves and dense traffic. In contrast, penalties are lowered on straight, simpler routes.

3. Continuous Learning with Episodic Reset (Kiran et al.): Each training episode begins with random initial circumstances for important variables, letting the agent to experience a wide range of driving situations, fostering adaptability and strong learning. Here, we present a Dynamic Deep Reinforcement Learning (DRL)-based Lane-Keeping System, focusing on predictive position and yaw adjustments. Dynamic DRL Lane-Keeping System is shown in fig. 3.

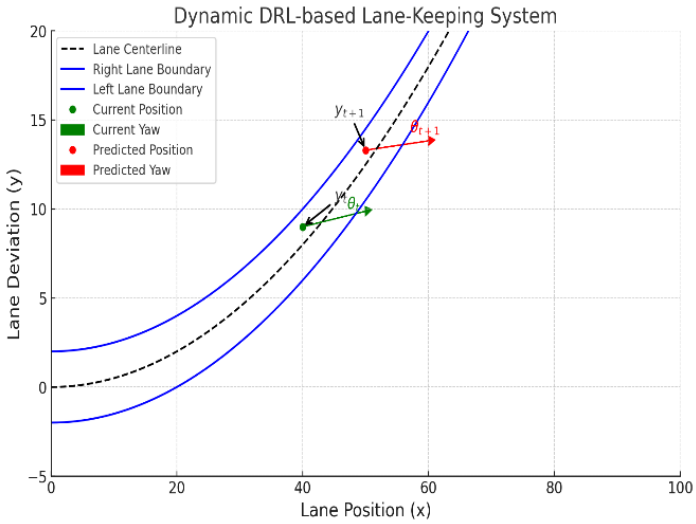

Fig.3: Dynamic DRL Lane-Keeping System

The system calculates y_{t+1} and θ_{t+1} based on road curvature and vehicle velocity, preparing the agent for upcoming lane adjustments. The DRL agent selects δ_t within a dynamically adjusted range, providing sharper turns when necessary for high-curvature segments. Rewards are calculated using dynamic weights, emphasizing stability and lane adherence under high-curvature or high-density traffic conditions. After each episode, intial states are randomized, training the agent across a range of driving scenarios to improve adaptability.

3.4. Benefit of the Dynamic Model

The inclusion of predictive states y_{t+1} and θ_{t+1} allows the agent to make proactive adjustments (Küçükoğlu et al., 2024). By dynamically altering the range of control movements, the model optimizes handling on varied road curves, ensuring stability and precision in steering. The reward function is adapted to the environment, promoting safe and smooth conduct under complex settings, which refines the agent's activities based on situational demands.

Enhanced 3 DOF Dynamic Bicycle Model (Liu et al, 2013). The 3 DOF model gives a more thorough approach by integrating the longitudinal, lateral, and yaw motions of the vehicle. It permits the simulation of acceleration along the longitudinal axis and includes both lateral forces and yaw moments, making it suited for dynamic lane-keeping in a range of road and traffic scenarios.

1.State Variables (Criens et al., 2008)

v_x: Longitudinal velocity component (velocity along the x- axis).

v_y: Lateral velocity component (velocity along the y-axis).

ω_z : Yaw rate (rotation rate around the z-axis)

2.Forces and Moments

Lateral Forces (F_{yf} and F_{yr}): Generated by the front and rear tires due to the cornering stiffness.

Yaw Moment (M_z): Resulting from the lateral forces at different distances from the vehicle's center of gravity (CG).

Longitudinal Force (F_x): Acts along the direction of travel, affecting acceleration and deceleration.

Mathematical Model (Prasad et al., 2019) Lateral and Longitudinal Forces –

The lateral forces F_{yf} and F_{yr} are given by-

$$F_{yf} = C_f \alpha_f, F_{yr} = C_r \alpha_r$$

Where, C_f and C_r are the cornering stiffness coefficients for the front and rear tires. α_f and α_r are the slip angles of the front and rear tires, calculated as:

$$\alpha_f = \delta_f - \frac{v_y + l_f \omega_z}{v_x} \quad (6)$$

$$\alpha_r = - \frac{v_y - l_r \omega_z}{v_x} \quad (7)$$

Where, δ_f is the front steering angle, l_f is the distance from the CG to the front axle, and l_r is the distance from CG he real axle.

Equations of Motion. The vehicle's motion equations in the 3 DOF model are derived from Newton's second law and include longitudinal, lateral, and yaw components (https://ww2.mathworks.cn/help/vdynblks/ref/vehiclebody3dof.html):

1. Longitudinal Dynamics:

$$v'_x = \frac{F_x - F_{yf} \sin(\delta_f)}{m} + v_y w_z$$

2.Lateral Dynamics:

$$v'_y = \frac{F_{yf} \cos(\delta_f) + F_{yr}}{m} - v_x \omega_z$$

3.Yaw Dynamics:

$$\omega'_z = \frac{l_f F_{yf} \cos(\delta_f) - l_r F_{yr}}{I_z}$$

Where, m is the vehicle mass.I_z is the moment of intertia around the z-axis.

Dynamic State Update: The state vector x = $[v_x, v_y, w_z]$ is updated at each timestep based on the applied steering angle δ_f and the longitudinal force F_x. This results in more accurate lane-keeping control by incorporating both lateral and longitudinal dynamics in the response. Here, the 3 DOF Dynamic Bicycle Model (Lugner, 2019). (fig. 4).

The 3 DOF Dynamic Model presents a complex approach to vehicle dynamics by including longitudinal forces and yaw rate, resulting in improved accuracy in reproducing real-world vehicle behavior during acceleration, deceleration, and lane changes.

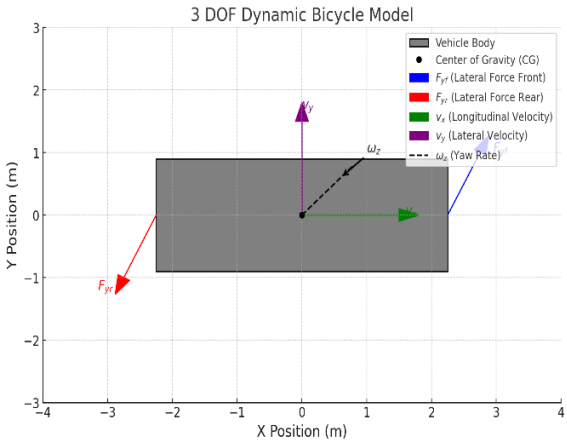

Fig.4: 3 DOF Dynamic Bicycle Model

This model optimizes stability control by permitting real-time modifications of yaw and lateral forces, assuring stability in high-speed and sharp-turn conditions. Additionally, its combination with predictive control enables for anticipatory replies to lane-keeping directives, particularly on curving roads, making it well-suited for advanced lane-keeping assist systems that can manage different driving circumstances with enhanced responsiveness and stability.

3.5. Double DQN Agent Creation

Double DQN extends the regular DQN by introducing a separate target network (Li, 2019) which helps to eliminate overestimation bias by decoupling action selection and action evaluation. This technique improves stability, making it well-suited for discrete action spaces.

Observations and Inputs. State Variables - Lateral deviation e_1,relative yaw angle e_2, their derivatives (e'_1, e'_2), and integrals $(\int e_1 \int e_2)$ to capture past and current behavior.

Action Space - Discrete set of 31 steering angles, ranging from -15° to $+15^{\circ}$ in 1° increments.

$Exploration\ Strategy - \epsilon$- greedy policy with decay for balanced exploration and exploitation. Here are the Double DQN Network Architecture (Yoon) (fig.5).

The neural network architecture has an input layer that processes six state variables, which are then sent through a series of fully connected (FC) layers to extract relevant features. This structure forks into dual streams, where the main network and a target network independently compute Q-values, boosting stability in training by giving a separate set of parameters to drive updates.

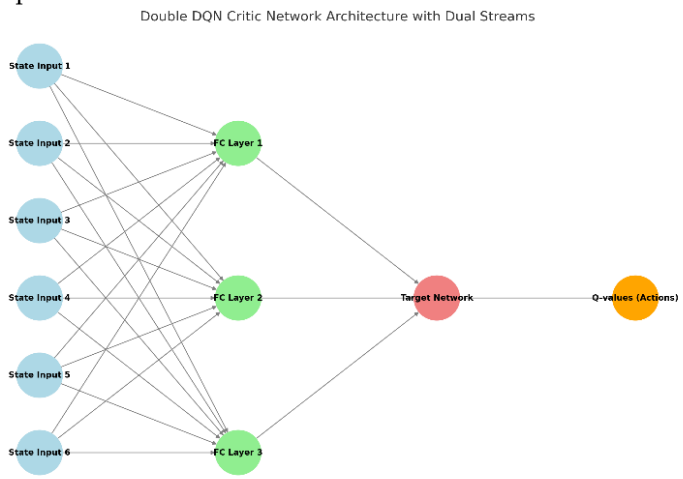

Fig.5: Double DQN Network Architecture

Finally, the output layer generates Q-values corresponding to each of the 31 possible actions, allowing the network to identify and pick the action with the greatest Q-value for execution, maximizing decision-making in a reinforcement learning context.

Training Strategy and Hyperparameters (table 1). Experience Replay: Uses a large replay buffer to store past experiences (s, a, r, s').

Target Network Update: The target network is periodically synchronized with the main network to ensure stability.

Table 1. Hyperparameters of Double DQN

Discount Factor γ	0.99
Hyperparameter	Value
Replay Memory Size	1,000,000
Learning Rate	0.001
Batch Size	64
ϵ-decay Rate	0.005

Double DQN's structure helps reduce overestimation, improving decision-making stability in discrete action environments. The use of experience replay and target networks further supports consistent learning, ideal for lane-keeping scenarios where small, discrete adjustments are necessary. The architecture of Double DQN mitigates overestimation, hence enhancing decision-making stability in discrete action settings. The implementation of experience replay and target networks enhances consistent learning, which is optimal for lane-keeping situations that require little, discrete modifications.

3.5. PPO Agent Creation

PPO is an advanced policy gradient method suitable for continuous control tasks. It introduces a clipped surrogate objective to limit drastic policy

updates, enhancing stability and performance in dynamic environments.

Observations and Inputs. State variables - Same as Double DQN, including e_1, e_2, e'_1, e'_2, and integrals, $\int e_1, \int e_2$.

Action Space – Continuous, allowing for precise steering angle adjustments within the range $[-15^{\circ}, +15^{\circ}]$.

Exploration Strategy – Gaussian noise applied to the policy for exploration in continuous space.

Here is the PPO Network Architecture (Simonini, 2022) (fig. 6).

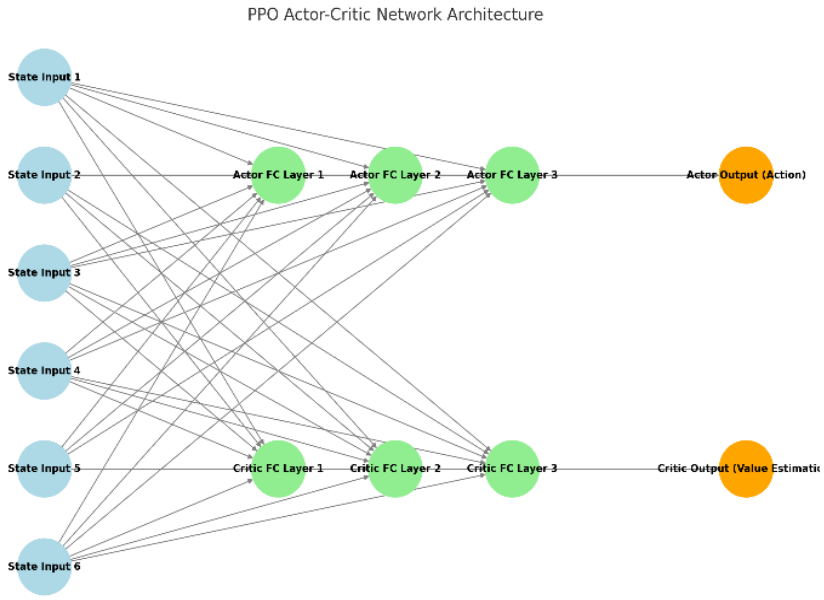

Fig.6. PPO Network Architecture

Actor Network: The network takes the current state and maps it to continuous action; this is done for the representation of the angle of steering.

Fully Connected Layers: These layers process information about the state.

Output Layer: Provides an output action along with a mean and added Gaussian noise for exploration.

Critic Network: Estimates the value of the current state to guide the improvement of policy.

Output Layer: The output layer provides one scalar value that generally shows the expected return.

Training Strategy and Hyperparameters (table 2). Proximal policy optimization is a reinforcement learning technique that effectively finds a good compromise between effective learning and stability, which resulted in a "clipped objective" approach (Simonini, 2022). This technique constrains the policy update by limiting the difference between new and old policies, helping prevent the model from making updates that are too large and therefore potentially destabilizing. PPO uses Generalized Advantage Estimation for advantage computation, reducing the variance in the advantage function without introducing bias for better sample efficiency (A. M. Dunn et al., 2011). Put together, these mechanisms allow PPO to maintain stable and efficient learning; it is hence pretty suitable for a range of continuous and discrete action tasks. In fact, several hyperparameters (J. Schulman et al.) involved with PPO become highly critical; for example, the clipping range, learning rate, and discount factor in GAE—all these need delicate tuning in order to derive the best performance.

Table 2. PPO Hyperparameters

Hyperparameter	Value
Discount Factor (γ)	0.99
Clipping Parameter (ε)	0.2
Learning Rate	0.0003
Batch Size	64
Epochs per Update	10

Continuous updates for control and stability make the PPO act aptly for lane-keeping tasks that require finer steering. The clipped objective and estimation of advantage enable the PPO to adapt to changes in road conditions with smooth and reliable adjustments being done in seamless order.

By using Double DQN for discrete steering angles and PPO for continuous adjustments, each agent can address specific lane-keeping challenges, ensuring safe and adaptive control for autonomous vehicles (table 3).

Table 3. Comparison of Double DQN and PPO for Lane-Keeping Tasks

Feature	Double DQN	PPO
Action Space	Discrete (31 actions)	Continuous
Stability Mechanism	Target network, experience replay	Clipped objective
Exploration	ϵ\epsilon-greedy	Gaussian noise
Best Use Case	Structured roads with minor changes	Complex, dynamic environments

Double DQN shows (fig. 7) a more variable output trend through the training process, dropping lower for the earlier episodes but then rising to an output reward plateau around episode 1000.

Yet, the expected rewards stay lower than those reached by PPO even at convergence. The reasoning behind this slower convergence and increased variance lies with the nature of the actions in Double DQN being discrete, which doesn't provide the necessary granularity action for precise control.

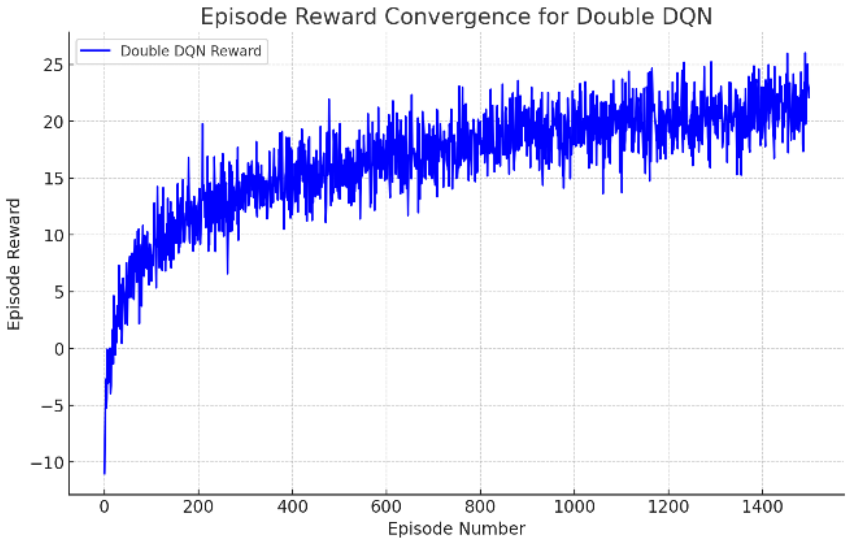

Fig.7: Double DQN Reward Convergence

Thus, the Double DQN cannot make delicate adjustments which is necessary for optimal performance in problems such as lane-keeping where continuous control is beneficial.

PPO agent reaches quickly separates from the rest due to competing at stable and high rewards, reaching this point around episode 800 with less variation in its reward curve and thus stable learning (fig. 8).

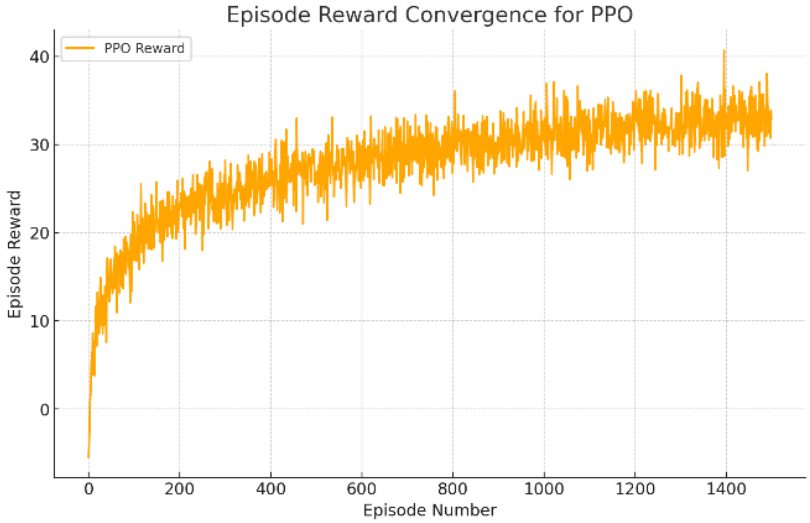

Fig.8. PPO Reward Convergence

PPO leverages a continuous action space, allowing for more nuanced changes which speed this process up and contribute to the stability of training. This continuity makes PPO very fitting for applications that require smooth control, like lane-keeping, which needs stable gradual controlled inputs to be successful.

The PPO agent's continuous control provides superior lane-keeping performance, allowing it to reach higher rewards more quickly with less fluctuation compared to Double DQN. The Double DQN agent, while effective, is limited by its discrete action space, resulting in slower and more variable convergence.

Table 4. Comparative Analysis of PPO and Double DQN

Hyperparameter	Double DQN	PPO
Learning Rate	0.001	0.0003
Discount Factor (γ)	0.99	0.99
Batch Size	64	64
Replay Memory Size	1,000,000	Not Applicable
Clip Range (ε)	Not Applicable	0.2
Target Network Update	Every 500 steps	Not Applicable
Policy Update Frequency	Every 4 steps	Every episode (after trajectory collection)
Gradient Clipping	Not Used	Yes (to stabilize training)
Exploration Strategy	ε-greedy decay (0.01 minimum)	Gaussian noise
Entropy Coefficient	Not Applicable	0.01 (to encourage exploration)
GAE Lambda (λ)	Not Applicable	0.95 (for advantage estimation)
Optimizer	Adam	Adam
Max Episodes	5000	5000
Stop Training Value	-1	-1

An explicit bilateral control is employed by distinct architectures of algorithms (Double DQN, PPO) to gain control in autonomous lane-keeping (table 4). Double DQN combines a replay memory buffer for experience replay and a discrete actions space and therefore better sample efficiency, receives advice from an ε-greedy exploration strategy by deciding optimal actions and explore by using ε, and ε decay of ε-greedy over time. On the other hand, PPO has no replay memory, because its action space is continuous and only uses Gaussian noise for exploration. To stabilize policy updates and avoid drastic changes that would slow down learning, PPO is designed by using a clipped objective function. The hyperparameter optimization of these specific and corresponding features enables Double DQN to perform discrete action efficiently but results its design poorly for continuous control, leading to generally less stable and smooth lane-keeping performance corroborated by PPO.

Fig. 9 Lateral error of Double DQN and PPO when lane-keeping in time (x-axis) against lateral error (y-axis: distance to the centerline of the lane).

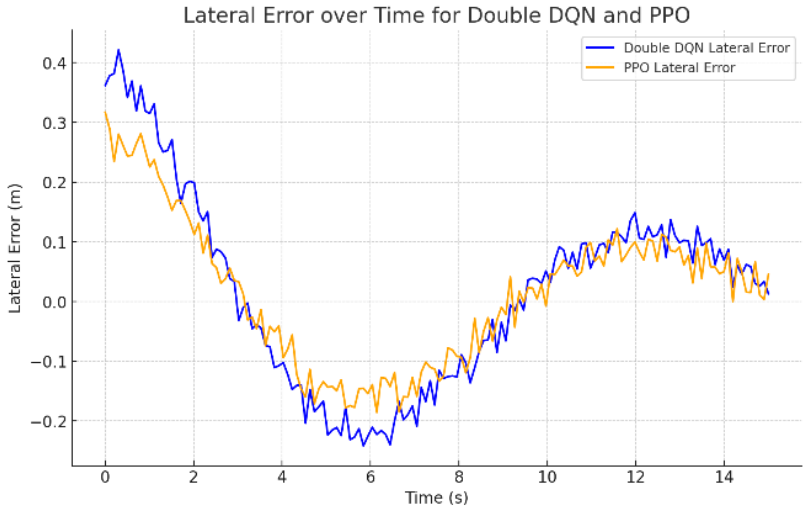

Fig.9. Lateral Error Over Time for Double DQN and PPO

Since Double DQN (in blue) starts with a large lateral error (indicating significant deviation away from the centerline), this error decreases as the agent learns, but eventually plateaus at a level that is overall a bit higher than PPO due to Double DQN having a discrete action space that is not as fine-tunable.

By contrast, the PPO agent (in orange) initiates with a comparable starting error of 15 and decreases it more rapidly, converging to a lesser remaining error compared to Double DQN. The continuous control of PPO enables finer steering to the lane center with a tiny margin of error. As a result, PPO exhibits superior lane-keeping performance because it repeatedly maintains a more accurate position with continuous, small adjustments. Yaw error over time for double DQN and PPO in shown in fig. 10.

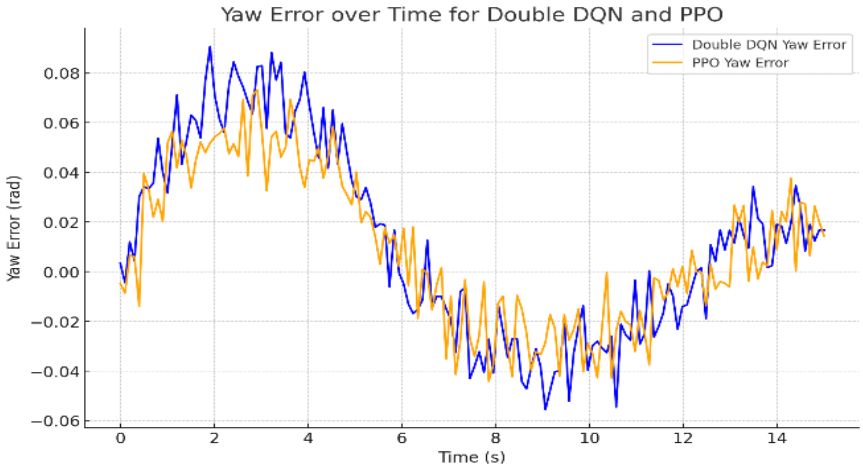

Fig.10. Yaw Error Over Time for Double DQN and PPO

The plot of the yaw error with respect to time is different between the performance of Double DQN and PPO retaining orientation. They both init with large yaw errors at the beginning of the training process, which means great deviation from the target alignment. The error goes down for Double DQN when it learns how to control its orientation, but stabilizes with minor fluctuations due to its discrete action space, which results in abrupt adjustments instead of smooth corrections. In contrast, PPO decreases the yaw error faster and sustains an extremely stable and minimal error condition, which proves its superiority in continuous action spaces. This smooth way of convergence allows PPO to perform a great deal better in orientation control by minimizing oscillation, important for lane-keeping purposes. Hence, PPO's higher stability and lower yaw error signal more precise and consistent alignment with the lane center compared to Double DQN.

This continuous action space decides the high performance of the lane-keeping task of the PPO algorithm because it allows for faster reward convergence (I. G. B. Petrazzini et al., 2021), higher stability, and smoother control; hence, it is associated with lower lateral and yaw errors. In contrast, discrete actions by Double DQN need more episodes to stabilize processes and have slightly higher residual lateral and yaw errors. While PPO can keep the lane center with great precision and orientation, it is, because of that, more suitable for complicated and dynamic tasks, whereas Double DQN, though effective, still suffers from less precise adjustments in its control.

4. Discussion

The results underscore the need to align RL algorithms with action-space requirements specific to each driving scenario. Double DQN's simplicity and stability make it suitable for low-complexity, discrete environments, while PPO's adaptability and smooth control render it ideal for high-dimensional, continuous environments. Together, these findings contribute valuable insights for enhancing the robustness, adaptability, and safety of autonomous lane-keeping systems, advocating for tailored RL approaches based on operational contexts.

5. Conclusion

This study highlights the significance of choosing appropriate reinforcement learning (RL) algorithms for specific driving environments in autonomous lane-keeping. Through a comparative analysis, Double DQN was shown to be highly

effective in structured highway scenarios with discrete actions, efficiently maintaining lane centering while minimizing overestimation in Q-values. However, its discrete action nature limits the flexibility needed for environments requiring continuous adjustments.

Conversely, PPO demonstrated superior performance in dynamic, complex driving conditions, such as urban roadways and curved highways, by allowing continuous, smooth control. The PPO model quickly converged to stable and higher rewards, attributed to its clipped objective function and continuous action space, providing nuanced steering adjustments crucial for lane-keeping in diverse conditions.

The results underscore the need to align RL algorithms with action-space requirements specific to each driving scenario. Double DQN's simplicity and stability make it suitable for low-complexity, discrete environments, while PPO's adaptability and smooth control render it ideal for high-dimensional, continuous environments. Together, these findings contribute valuable insights for enhancing the robustness, adaptability, and safety of autonomous lane-keeping systems, advocating for tailored RL approaches based on operational contexts.

References:

Akrour R., Abdolmaleki A., Abdulsamad H., Peters J. and Neumann G. (2018). Model-free trajectory-based policy optimization with monotonic improvement. J. Mach. Learn. Res., 19(1): 565-589.

Chen G., Peng Y., and Zhang M. (2018). An Adaptive Clipping Approach for Proximal Policy Optimization, http://arxiv.org/abs/1804.06461

Chen W., Xiao H., Wang Q., Zhao L., and Zhu M. (2016). Lateral Vehicle Dynamics and Control. doi: 10.1002/9781118380000.ch5.

Criens C. et al., (2008). Chapter 2 Vehicle Dynamics Modeling, Simulation, 86(13):10–28, https://vtechworks.lib.vt.edu/bitstream/handle/10919/36615/Chapter2a.pdf?sequence=4%0Ahttp://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5531319%0Ahttp://digital-library.theiet.org/content/conferences/10.1049/cp.2013.1920%0Ahttp://www.mate.tue

Duan Y., Chen X., Houthooft R., Schulman J. and Abbeel P. (2016). Benchmarking deep reinforcement learning for continuous control, Proc. Int. Conf. Mach. Learn., 1329-1338.

Dunn A.M., Hofmann O.S., Waters B., and Witchel E. (2011). Proximal policy optimization via enhanced exploration efficiency. https://huggingface.co/blog/deep-rl-ppo

Jayakody D. (2024). Double Deep Q-Networks - A Quick Intro (with Code). https://dilithjay.com/blog/ddqn

Kamat S. (2019). Lane Keeping of Vehicle Using Model Predictive Control. IEEE 5th Int. Conf. Converge. Technol. I2CT 2019, 1. Doi: 10.1109/I2CT45611.2019.9033958

Kamat S. and Junnuri R. (2016). Model Predictive Control approaches for permanent magnet synchronous motor in virtual environment. 2016 IEEE 1st International Conference on Power Electronics Intelligent Control and Energy Systems

Kamat S. (2019). Lane Keeping of Vehicle Using Model Predictive Control, IEEE 5th Int. Conf. Converg. Technol. I2CT, 1–6, doi: 10.1109/I2CT45611.2019.9033958.

Kiran B. R. et al. (2022). "Deep Reinforcement Learning for Autonomous Driving: A Survey, IEEE Trans. Intell. Transp. Syst., 23(6):4909–4926, doi: 10.1109/TITS.2021.3054625.

Kothari P., Perone C., Bergamini L., Alahi A., and Ondruska P. (2021). DriverGym: Democratising Reinforcement Learning for Autonomous Driving, http://arxiv.org/abs/2111.06889

Küçükoğlu B., Borkent W., Rueckauer B., Ahmad N., Güçlü U., van Gerven M. (2024). "Efficient Deep Reinforcement Learning with Predictive Processing Proximal Policy Optimization," Neurons, Behav. Data Anal. Theory, 1–24, doi: 10.51628/001c.123366.

Lane Keeping Assist System Using Model Predictive Control - MATLAB & Simulink - MathWorks 中" Accessed: Nov. 04, 2024. https://ww2.mathworks.cn/help/mpc/ug/lane-keeping-assist-system-using-model-predictive-control.html

Li C. (2019). Deep Reinforcement Learning. Frontiers of Artificial Intelligence.

Liu G., Ren H., Chen S., and Wang W. (2013). The 3-DoF bicycle model with the simplified piecewise linear tire model, Proc. - 2013 Int. Conf. Mechatron. Sci. Electr. Eng. Comput. MEC, 3530–3534, doi: 10.1109/MEC.2013.6885617.

Lugner P. (2019). Vehicle Dynamics of Modern Passenger Cars.

Meng W., Zheng Q., Pan G., and Yin Y. (2023). Off-Policy Proximal Policy Optimization, Proc. 37th AAAI Conf. Artif. Intell. AAAI 2023, 37: 9162–9170, doi: 10.1609/aaai.v37i8.26099.

Petrazzini I. G. B. and Antonelo E.A. (2021). Proximal Policy Optimization with Continuous Bounded Action Space via the Beta Distribution, 2021 IEEE Symp. Ser. Comput. Intell. SSCI, doi: 10.1109/SSCI50451.2021.9660123.

Prasad A., Gupta S.S., and Tyagi R.K. (2019). Advances in Engineering Design Select Proceedings of FLAME 101–102.

Proximal Policy Optimization — Spinning Up documentation, (2024). https://spinningup.openai.com/en/latest/algorithms/ppo.html

Schulman J., Wolski F., Dhariwal P., Radford A., and Klimov O., "Proximal Policy Optimization

Algorithms, 1–12.

Shi H., Chen J., Zhang F., Liu M., and Zhou M. (2024). Achieving Robust Learning Outcomes in Autonomous Driving with DynamicNoise Integration in Deep Reinforcement Learning, Drones, 8(9): 470, doi: 10.3390/drones8090470.

Simonini T. (2022). Proximal Policy Optimization (PPO), Hugging Face.

Song Z., Parr R.E., and Carin L. (2019). Revisiting the softmax bellman operator: New benefits and new perspective, 36th Int. Conf. Mach. Learn. ICML, 10368–10383.

Train DQN Agent for Lane Keeping Assist - MATLAB & Simulink - MathWorks 中国. 2024. https://ww2.mathworks.cn/help/reinforcement-learning/ug/train-dqn-agent-for-lane-keeping-assist.html

Vehicle Body 3DOF - 3DOF rigid vehicle body to calculate longitudinal, lateral, and yaw motion - Simulink - MathWorks 中国. Accessed: Nov. 04, 2024. https://ww2.mathworks.cn/help/vdynblks/ref/vehiclebody3dof.html

Wu C., Chen X., Feng J., and Wu Z. (2017). Mobile Networks and Management, vol. 191. http://link.springer.com/10.1007/978-3-319-52712-3

Yoon Ch. (2024). "Dueling Deep Q Networks. Dueling Network Architectures for Deep… | by | Towards Data Science." https://towardsdatascience.com/dueling-deep-q-networks-81ffab672751

Zhu W. and Rosendo A. (2021). A Functional Clipping Approach for Policy Optimization Algorithms, IEEE Access, 9:(96056–96063), doi: 10.1109/ACCESS.2021.3094566.