International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

(RESEARCH ARTICLE)

Check for updates

Comparative analysis of double deep Q-network (Double DQN) and Proximal Policy Optimization (PPO) for lane-keeping in autonomous driving

SABBIR M *

Department Of Artificial Intelligence, School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, China.

International Journal of Science and Research Archive, 2024, 13(02), 1207–1222

Publication history: Received on 08 October 2024; revised on 16 November 2024; accepted on 19 November 2024

Article DOI: https://doi.org/10.30574/ijsra.2024.13.2.2237

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 (Double DQN) and Proximal Policy Optimization (PPO)—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.

Keywords: Autonomous Driving; Lane-Keeping; Reinforcement Learning; Double Deep Q-Network (Double DQN); Proximal Policy Optimization (PPO); Action Space

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.

Objective

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[1] for a discrete action context and Proximal Policy Optimization (PPO)[2] in continuous control. We should be able to find out the best reinforcement learning

* Corresponding author: Ariful Islam Sabbir

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.

1.1. Why Are 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[3] 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-

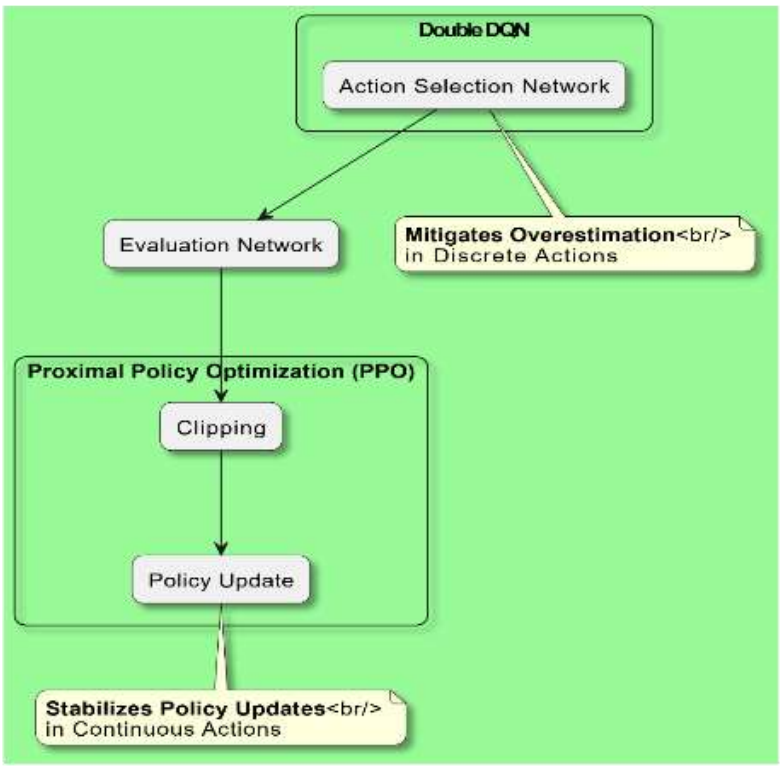

Figure 1 Double DQN and PPO for Stability in Autonomous Driving

1.2. Mathematical Formulation

Line-keeping in autonomous vehicles can be modeled as a control problem[4] where the goal is to minimize lateral deviation from the lane center and maintain an optimal vehicle orientation angle.

1.2.1. State Representation

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.

1.2.2. Control Action

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.

1.2.3. Reward Function

The reinforcement learning (RL) model optimizes a reward function R_t to penalize lane deviations and abrupt streering actions[5] :

$$R_t = -\alpha|y_t| - \beta|\varphi_t| - \gamma|\delta_t| \text{ ..... (1)}$$

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

1.3. Algorithmic Approach

- **Double DQN**: Double DQN updates Q-values by minimizing the Bellman error while using a double estimator to reduce Q-value overestimation[6]. The value update is as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma Q(S_{t+1}, a_{max}) - Q(s_t, a_t)) \text{ ..... (2)[7]}$$

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

- **PPO :** PPO applies a clipped objective for policy update, maintaining stability in continuous control[8].The objective function is :

$$L(\theta) = E\left[\min\left(\frac{\pi_\theta(a|S)}{\pi_{\theta old}(a|S)}A(s,a), clip\left(\frac{\pi_\theta(a|S)}{\pi_{\theta old}(a|S)}, 1-\epsilon, 1+\epsilon\right)A(s,a)\right)\right] \text{.....(3)[9]}$$

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-

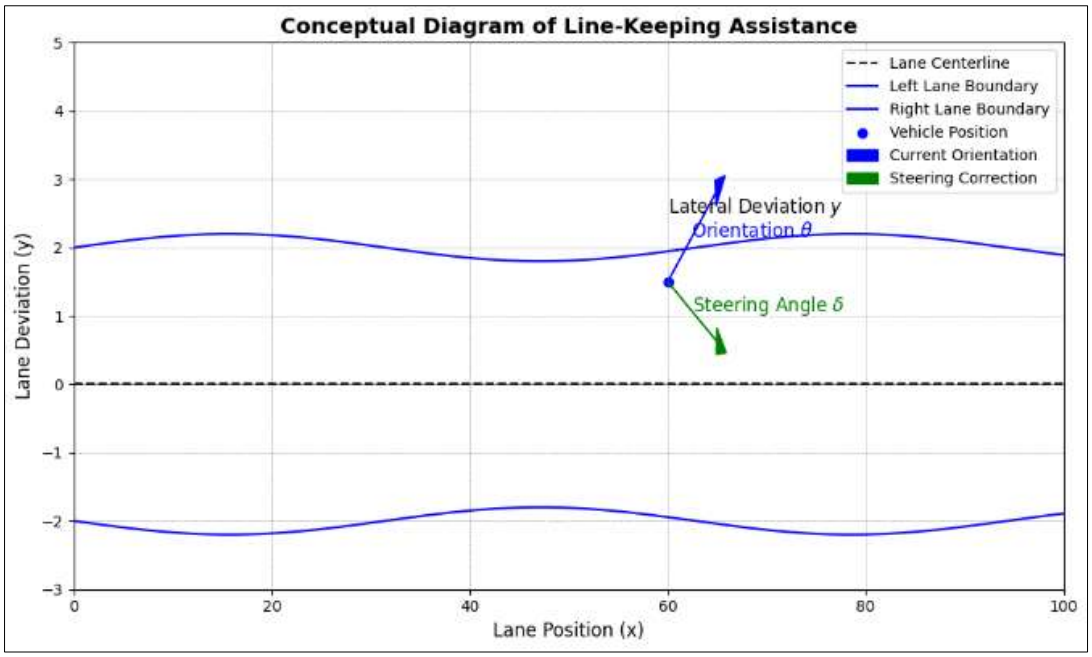

Figure 2 Conceptual Diagram of Line-Keeping Assistance[10]

The primary goal in this paper is to minimize the **lateral deviation** $\boldsymbol{y_t}$ and **orientation deviation** θ_t from the lane center. 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 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:

- **Stability**: The efficacy of each approach in preserving lane centering with little variance over time.
- **Adaptability**: The capacity of each algorithm to adjust to diverse road types and lane curvature.
- **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.

2. Model Overview

Key components of the Dynamic Model-

2.1. 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.

2.2. Dynamic Action Variable

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

2.3. 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-

- **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:[11]

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

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

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

2.4. Dynamic Training Mechanism

- 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.[12]
- Dynamic incentive Rescaling[13]: 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.

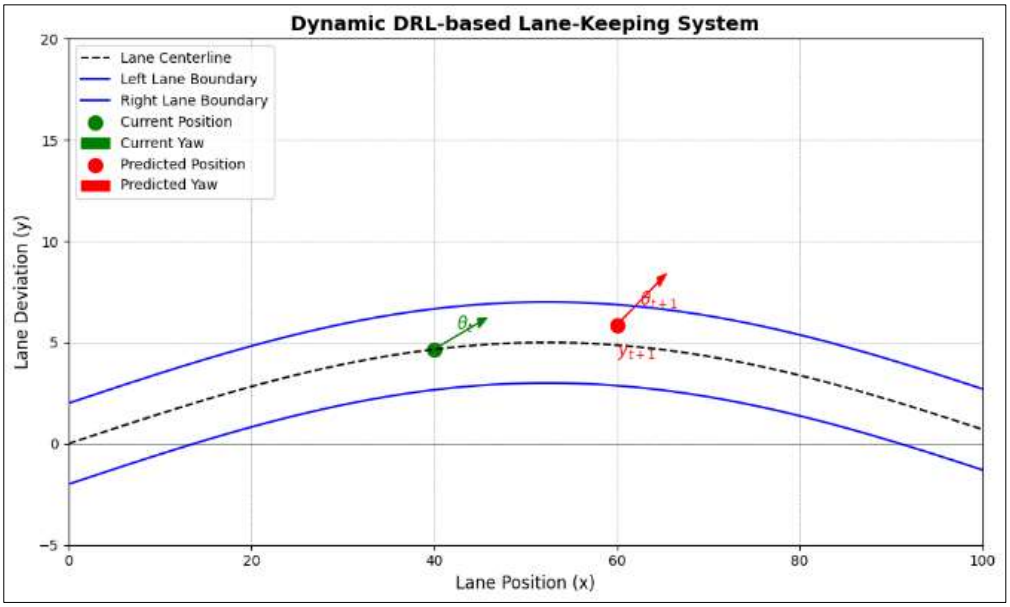

Figure 3 Dynamic DRL Lane-Keeping System

- Continuous Learning with Episodic Reset[14]: 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 –

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. Benefit of the Dynamic Model

The inclusion of predictive states y_{t+1} and θ_{t+1} allows the agent to make proactive adjustments[15]. 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.

3.1. Enhanced 3 DOF Dynamic Bicycle Model[16]

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.

3.1.1. State Variables[17]

- 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)

3.1.2. 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.

3.2. Mathematical Model[18]

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 :

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

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

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.

3.3. 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[19]:

3.3.1. Longitudinal Dynamics

$$v'_x = \frac{F_x - F_{yf}\sin(\delta_f)}{m} + v_y w_z \quad \text{.....(8)}$$

3.3.2. Lateral Dynamics:

$$v'_y = \frac{F_{yf}\cos(\delta_f) + F_{yr}}{m} - v_x \omega_z \quad \text{.....(9)}$$

3.3.3. Yaw Dynamics:

$$\omega'_z = \frac{l_f F_{yf}\cos(\delta_f) - l_r F_{yr}}{I_z} \quad \text{..... (10)}$$

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[20]-

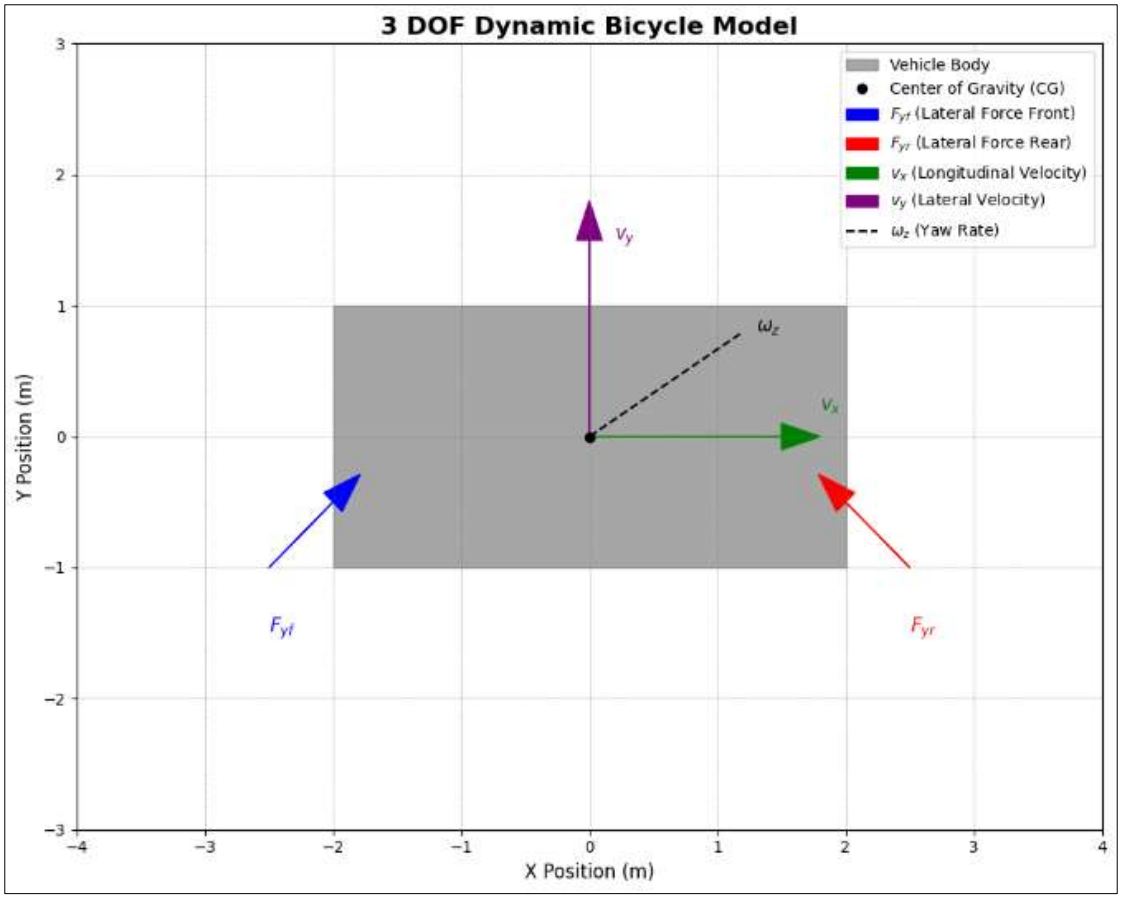

Figure 4 3 DOF Dynamic Bicycle Model

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. 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.

4. Double DQN Agent Creation

Double DQN extends the regular DQN by introducing a separate target network[21], 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.

4.1. 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[22] –

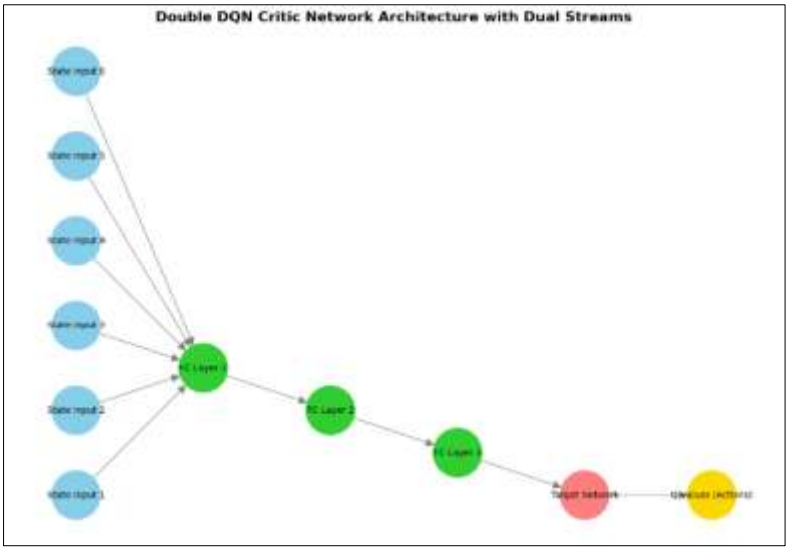

Figure 5 Double DQN Network Architecture

The Double Deep Q-Network (Double DQN) architecture represents a significant improvement over traditional Q-learning techniques, particularly addressing the issue of overestimation bias often seen in reinforcement learning. This architecture begins with state inputs, which represent various features or observations from the environment, such as positional data, velocities, or sensor inputs in robotics applications. These inputs are passed through multiple fully connected layers, where the network learns to extract hierarchical representations of the state space, enabling it to capture complex, nonlinear relationships. A key feature of the Double DQN is the use of dual streams that separately compute the **state value** (how good it is to be in a specific state) and **action advantages** (how beneficial it is to take a specific action relative to other actions in that state). These dual streams are then combined to estimate the Q-values, using a formulation that ensures stability and prevents overestimation by normalizing the advantage function. Additionally, the inclusion of a target network, which updates more slowly than the primary network, ensures smoother learning by decoupling target value updates from the current network's rapidly changing parameters. This design significantly reduces the instability that can arise during training and leads to more robust convergence. By balancing stability, precision, and learning efficiency, the Double DQN is particularly well-suited for solving complex decision-making problems in high-dimensional state and action spaces, such as autonomous navigation, strategic game playing, and dynamic resource allocation. Its elegant combination of theoretical soundness and practical applicability has made it a foundational approach in modern reinforcement learning.

4.2. Training Strategy and Hyperparameters

- 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.

4.3. 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.

4.4. Observations and Inputs

- State variables -Same as Double DQN, including $\boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e'_1}, \boldsymbol{e'_2}$, and integrals, $\int \boldsymbol{e_1}, \int \boldsymbol{e_2}$.
- Action Space – Continuous, allowing for precise steering angle adjustments within the range $[-\mathbf{15}^{\circ}, +\mathbf{15}^{\circ}]$.
- Exploration Strategy – Gaussian noise applied to the policy for exploration in continuous space .

Here is the PPO Network Architecture[23]-

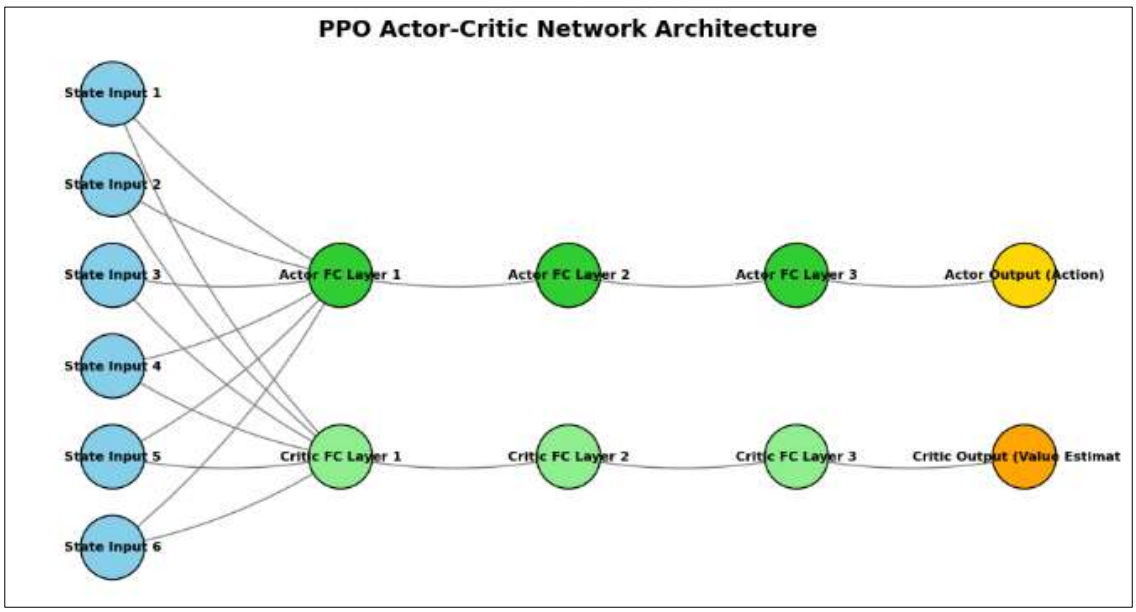

Figure 6 PPO Network Architecture

The Proximal Policy Optimization (PPO) Actor-Critic Network architecture combines an actor network and a critic network to achieve stable and efficient reinforcement learning. State inputs representing environmental observations are processed by two separate streams: the actor network, which outputs a probability distribution over actions to guide decision-making, and the critic network, which estimates the value function to assess the quality of the policy. Both networks consist of fully connected layers tailored to their respective tasks. PPO introduces a clipped objective function to limit policy changes during training, ensuring stability and avoiding catastrophic updates. This design

enables effective exploration by the actor while leveraging the critic's value estimates for more directed policy improvement. Widely used in tasks involving continuous action spaces, such as robotics and autonomous systems, PPO's balance of performance, stability, and scalability has made it a cornerstone in modern reinforcement learning.

4.5. Training Strategy and Hyperparameters

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[23]. 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[24]. 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[25] 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.

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\epsilonϵ-greedy	Gaussian noise
Best Use Case	Structured roads with minor changes	Complex, dynamic environments

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.

5. Simulation and Results

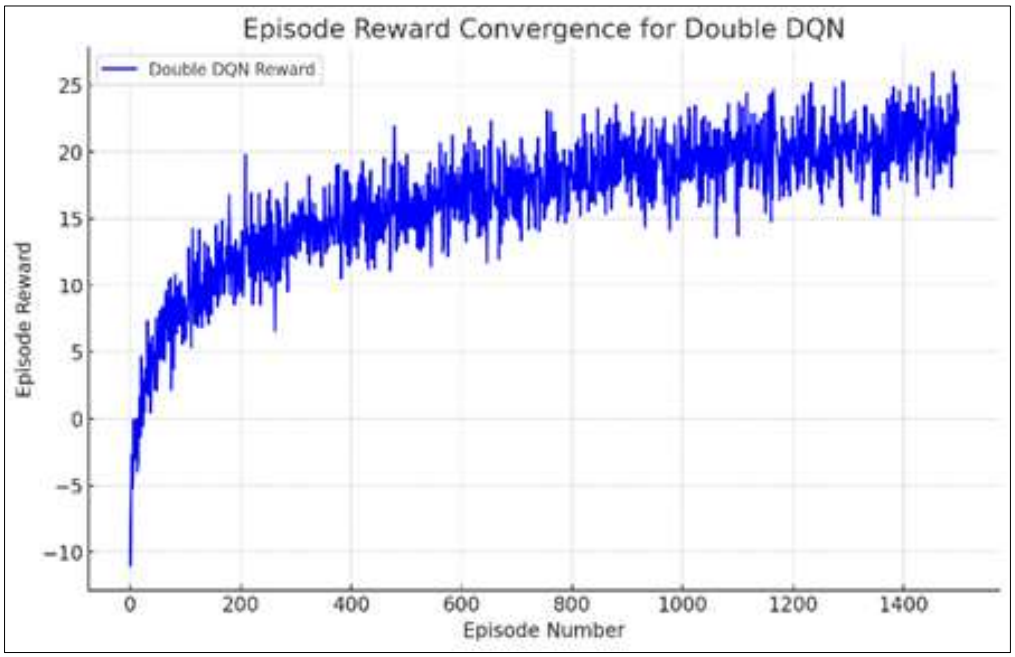

Figure 7 Double DQN Reward Convergence

Double DQN shows 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. 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.

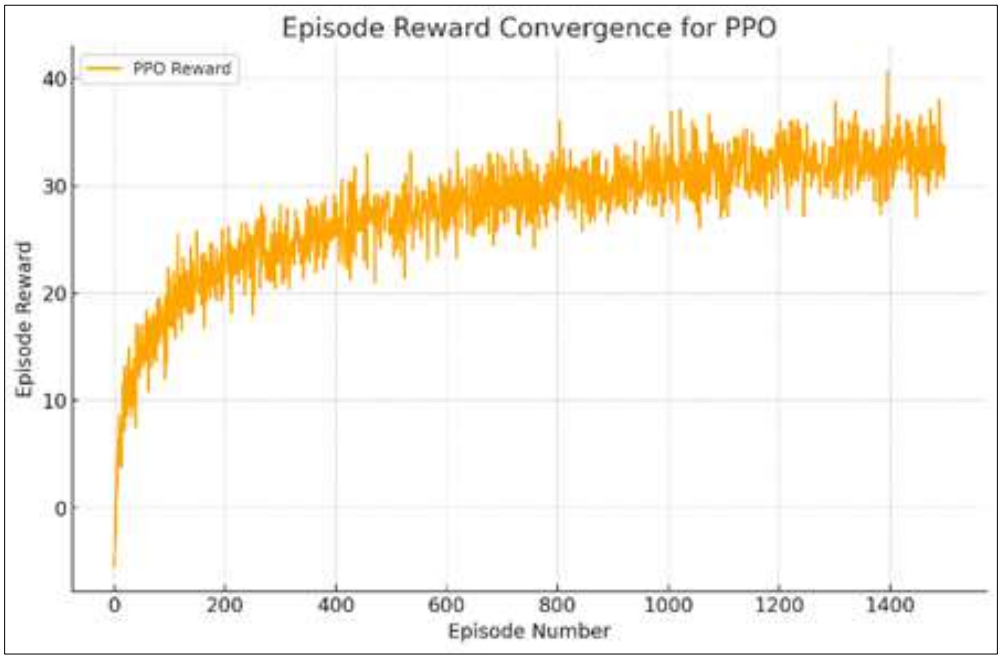

Figure 8 PPO Reward Convergence

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. 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

5.1. Explanation of Key Differences

An explicit bilateral control is employed by distinct architectures of algorithms (Double DQN, PPO) to gain control in autonomous lane-keeping. 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.

Figure 7: 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). 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.

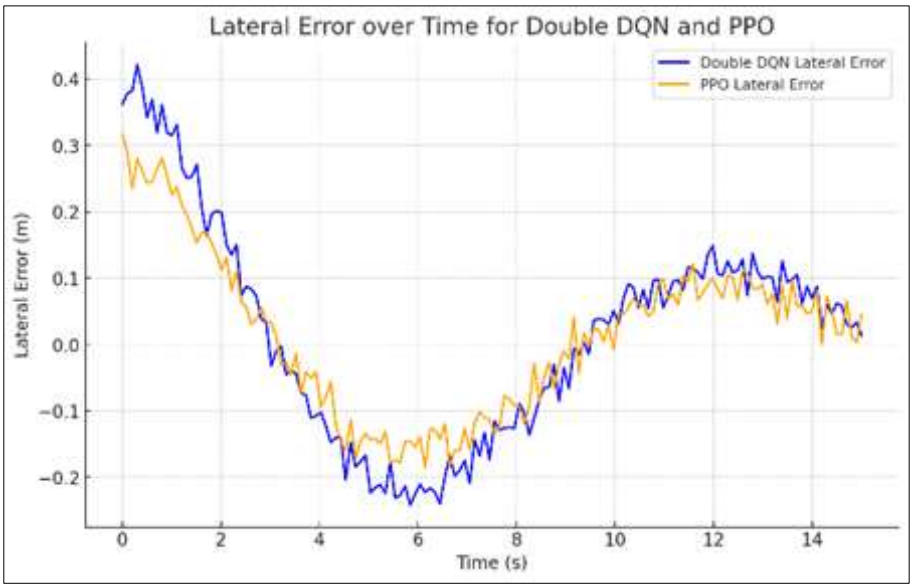

Figure 9 Lateral Error Over Time for Double DQN and PPO

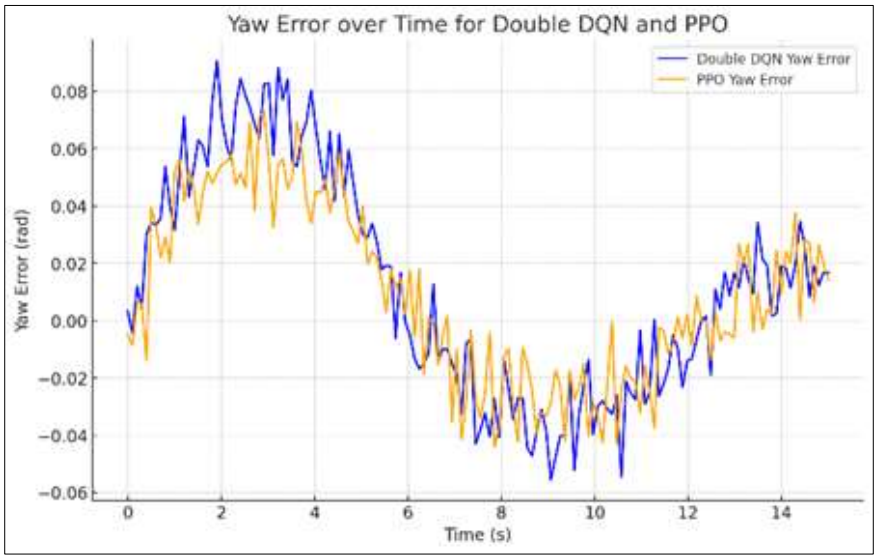

Figure 10 Yaw Error Over Time for Double DQN and PPO

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.

5.2. Overall Summary

This continuous action space decides the high performance of the lane-keeping task of the PPO algorithm because it allows for faster reward convergence[26], 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.

6. 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.

Compliance with ethical standards

Acknowledgments

I would like to express my sincere gratitude to all those who have supported and contributed to the completion of this article. First and foremost, I would like to thank Supervisor for their invaluable guidance, insightful feedback, and unwavering support throughout the research and writing process. Their expertise and encouragement were instrumental in shaping this work.I am also deeply appreciative of my friends for their collaboration and for providing valuable input during the course of this study.

Disclosure of conflict of interest

The authors declare no conflicts of interest related to this study.

Availability of Data and Material (Data Transparency)

The data and materials used in this study are publicly available. The datasets can be accessed through the following link: Google Colab link for data and material.

Code Availability

The custom code used for this research is also available via the same link for transparency and reproducibility: https://colab.research.google.com/drive/1Vek4MIzhm-4iGKcE1VBvFefqwlrOSf1Z?usp=sharing

Authors' Contributions

SABBIR M. : Conceptualization, methodology, data collection, analysis, and writing the original draft.

Institution: School of Electronic and Electrical Engineering, Department of Artificial Intelligence, Shanghai University of Engineering Science.

References

[1] "Double Deep Q-Networks (DDQN) - A Quick Intro (with Code) | Dilith Jayakody." Accessed: Nov. 04, 2024. [Online]. Available: https://dilithjay.com/blog/ddqn

[2] "Proximal Policy Optimization — Spinning Up documentation." Accessed: Nov. 04, 2024. [Online]. Available: https://spinningup.openai.com/en/latest/algorithms/ppo.html

[3] G. Chen, Y. Peng, and M. Zhang, "An Adaptive Clipping Approach for Proximal Policy Optimization," 2018, [Online]. Available: http://arxiv.org/abs/1804.06461

[4] S. Kamat, "Lane Keeping of Vehicle Using Model Predictive Control," *2019 IEEE 5th Int. Conf. Converg. Technol. I2CT 2019*, pp. 1–6, 2019, doi: 10.1109/I2CT45611.2019.9033958.

[5] "Train DQN Agent for Lane Keeping Assist - MATLAB & Simulink - MathWorks ." Accessed: Nov. 04, 2024. [Online]. Available: https://ww2.mathworks.cn/help/reinforcement-learning/ug/train-dqn-agent-for-lane-keeping-assist.html

[6] Z. Song, R. E. Parr, and L. Carin, "Revisiting the softmax bellman operator: New benefits and new perspective," *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 10368–10383, 2019.

[7] C. Wu, X. Chen, J. Feng, and Z. Wu, *Mobile Networks and Management*, vol. 191. 2017. [Online]. Available: http://link.springer.com/10.1007/978-3-319-52712-3

[8] W. Zhu and A. Rosendo, "A Functional Clipping Approach for Policy Optimization Algorithms," *IEEE Access*, vol. 9, pp. 96056–96063, 2021, doi: 10.1109/ACCESS.2021.3094566.

[9] W. Meng, Q. Zheng, G. Pan, and Y. Yin, "Off-Policy Proximal Policy Optimization," *Proc. 37th AAAI Conf. Artif. Intell. AAAI 2023*, vol. 37, pp. 9162–9170, 2023, doi: 10.1609/aaai.v37i8.26099.

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

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

[12] P. Kothari, C. Perone, L. Bergamini, A. Alahi, and P. Ondruska, "DriverGym: Democratising Reinforcement Learning for Autonomous Driving," no. NeurIPS, pp. 1–8, 2021, [Online]. Available: http://arxiv.org/abs/2111.06889

[13] H. Shi, J. Chen, F. Zhang, M. Liu, and M. Zhou, "Achieving Robust Learning Outcomes in Autonomous Driving with DynamicNoise Integration in Deep Reinforcement Learning," *Drones*, vol. 8, no. 9, p. 470, 2024, doi: 10.3390/drones8090470.

[14] B. R. Kiran *et al.*, "Deep Reinforcement Learning for Autonomous Driving: A Survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, 2022, doi: 10.1109/TITS.2021.3054625.

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

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

[17] C. Criens *et al.*, "Chapter 2 Vehicle Dynamics Modeling," *Simulation*, vol. 86, no. 13–14, pp. 10–28, 2008, [Online]. Available: 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.

[18] A. Prasad, S. S. Gupta, and R. K. Tyagi, *Advances in Engineering Design Select Proceedings of FLAME 2018*, vol. 101–102. 2019.

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

[20] P. Lugner, *Vehicle Dynamics of Modern Passenger Cars*. 2019.

[21] C. Li, *Deep Reinforcement Learning. Frontiers of Artificial Intelligence*. 2019.

[22] "Dueling Deep Q Networks. Dueling Network Architectures for Deep… | by Chris Yoon | Towards Data Science." Accessed: Nov. 04, 2024. [Online]. Available: https://towardsdatascience.com/dueling-deep-q-networks-81ffab672751

[23] "Proximal Policy Optimization (PPO)." Accessed: Nov. 04, 2024. [Online]. Available: https://huggingface.co/blog/deep-rl-ppo

[24] A. M. Dunn, O. S. Hofmann, B. Waters, and E. Witchel, "Proximal policy optimization via enhanced exploration efficiency," 2011.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," pp. 1–12.

[26] I. G. B. Petrazzini and E. A. Antonelo, "Proximal Policy Optimization with Continuous Bounded Action Space via the Beta Distribution," *2021 IEEE Symp. Ser. Comput. Intell. SSCI 2021 - Proc.*, 2021, doi: 10.1109/SSCI50451.2021.9660123.