192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

e-Prime - Advances in Electrical Engineering, Electronics and Energy 10 (2024) 100810

Contents lists available at ScienceDirect

e-Prime - Advances in Electrical Engineering, Electronics and Energy

journal homepage: www.elsevier.com/locate/prime

A comprehensive review on safe reinforcement learning for autonomous vehicle control in dynamic environments

Rohan Inamdar, S. Kavin Sundarr, Deepen Khandelwal, Varun Dev Sahu, Nitish Katal *

School of Electronics Engineering, Vellore Institute of Technology, Chennai, India

ARTICLE INFO

Keywords:
Reinforced learning
Trajectory planning
Autonomous vehicles
Autonomous control

ABSTRACT

To operate safely in a dynamic environment, autonomous vehicles must possess the same level of predictive driving abilities as human drivers and must be capable of anticipating the future actions of other dynamic objects in the environment, especially those of neighboring vehicles. The development of safe autonomous vehicles (AVs) poses a challenging task as it requires algorithms that can make real-time decisions in unpredictable circumstances. Reinforcement learning (RL) presents a promising approach for AV control, as it utilizes trial and error to enable optimal decision-making. However, traditional RL algorithms are unsuitable for safety-critical applications, as they may explore unsafe actions, potentially resulting in accidents. Safe reinforcement learning (SRL) algorithms have been developed to address this issue, prioritizing safe and reliable decisions. These algorithms incorporate constraints to prevent unsafe actions or utilize techniques to estimate action risk and avoid actions deemed excessively risky. Despite computational challenges, SRL holds significant promise for AV control, and is likely to play a crucial role in developing safe and reliable systems. SRL methods are critical for the general adoption of autonomous vehicles by guaranteeing their safety and reliability. These algorithms utilize methods like uncertainty and risk estimation along with penalty functions, to avoid excessively risky actions and have the potential to significantly reduce accidents and build public trust in autonomous driving. However, there are challenges that need to be addressed, such as the dynamic nature of real-world traffic, high computational costs, and the diversity of road design; and these varying conditions make the designing, testing, and validating of SRL algorithms difficult. Despite these challenges, SRL presents a promising solution, through integrating new sensing technologies and machine learning techniques, to develop safe, efficient, and environmentally friendly transportation systems.

1. Introduction

Autonomous vehicles (AVs), commonly known as self-driving cars, constitute a radical advancement in modern transportation systems. These vehicles have sophisticated sensor arrays, cameras, and state-of-the-art information processing systems that enable them with the capability to perceive their surroundings, make real-time decisions, and navigate independently. The deployment of autonomous vehicles holds the promise of enhancing road safety, reducing traffic congestion, facilitating more effortless mobility for individuals with disabilities, and promoting energy efficiency.

Reinforcement learning (RL) has emerged as a particularly promising sub-domain in machine learning, providing a robust and adaptive method for training autonomous cars. By utilizing a system of rewards and penalties, RL enables these vehicles to learn and refine their behaviors through iterative interactions with the environment. This dynamic learning process allows autonomous cars to continually enhance their decision-making capabilities, establishing RL as a powerful tool in developing and optimizing self-driving technology. The main objective is to achieve a specific goal of efficient and safe navigation. RL offers a unique advantage in handling complex, dynamic, and uncertain real-world scenarios, making it an ideal framework for developing autonomous vehicles. By leveraging RL, these vehicles can continuously evolve and optimize their responses to diverse and challenging situations, marking a significant step forward in the evolution of modern transportation systems.

Fig. 1 demonstrates the timeline of the various technological advancements that have contributed to the development of autonomous

* Corresponding author.
E-mail addresses: rohan.inamdar2021@vitstudent.ac.in (R. Inamdar), kavinsundarr.s2021@vitstudent.ac.in (S.K. Sundarr), deependheeraj.khandelwal2021@vitstudent.ac.in (D. Khandelwal), varundev.sahu2021@vitstudent.ac.in (V.D. Sahu), nitishkatal@gmail.com, nitish.katal@vit.ac.in (N. Katal).

https://doi.org/10.1016/j.prime.2024.100810
Received 7 December 2023; Received in revised form 3 July 2024; Accepted 9 October 2024
Available online 11 October 2024
2772-6711/© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

vehicles. Early developments followed a rule-based approach, followed by the advancements in modern control systems and neural networks in the 1980's. Later, in the years 2000 onwards, the probabilistic approaches emerged, and now current research focuses on the use of reinforcement learning algorithms, human-machine interactions, etc., to ensure safe navigation of these autonomous vehicles in the environment.

Initially, the development of autonomous vehicles relied on rule-based systems, where vehicles followed predetermined navigation rules within controlled environments [2]. Subsequently, sensor-based control [3] was introduced, integrating sensors like LiDAR, RADAR, and cameras into autonomous vehicles to perceive their surroundings. The data collected by these sensors forms real-time decision-making processes, like obstacle avoidance and lane-following systems. As autonomous vehicles evolved, researchers explored the integration of machine learning and neural networks [4] into perception and control systems. This use of sensor data facilitated faster and more adaptive real-time decision-making, particularly in tasks such as obstacle avoidance and lane following. Further investigations were carried out on probabilistic approaches [4-6]. These approaches incorporate probabilistic methods, such as Bayesian filters and Markov models, to address uncertainty in perception and decision-making. The primary focus was on modeling the uncertainty inherent in sensor data and predicting the likelihood of various states of the vehicle. The integration of probabilistic approaches significantly enhanced the robustness of autonomous systems.

In the evolution of autonomous vehicle development, the shift towards Deep Reinforcement Learning (DRL) has marked a transformative milestone. In 2013, DRL initially demonstrated its capabilities by training algorithms for the video game '*Atari.*' This revolutionary framework laid the basis for the subsequent introduction of Deep *Q*-Network (DQN) in 2015 [7]. DQN seamlessly merged Reinforcement Learning and Deep Learning (DL), achieving remarkable success in seven games, and surpassing human performance in three games.

Amidst this technological progression, the significance of safety in autonomous car development cannot be overstated. The integration of state-of-the-art technologies must align with the fundamental necessity of safeguarding users and the public. As autonomous vehicles navigate complex scenarios, their reliability becomes critical in preventing collisions and minimizing risks to passengers, pedestrians, and fellow road users. While the potential of RL to impart complex driving techniques is apparent, it introduces unique challenges in ensuring responsible and safe learning. The ongoing search for safe RL methods, supported by robust algorithms and regulatory frameworks, is essential. These requirements extend throughout the learning process, including strategies to mitigate risky behaviors during training, mechanisms to address uncertainties and rare occurrences, and rigorous protocols to validate the security of learned policies before deployment.

This literature review aims to thoroughly examine the current landscape of Safe Reinforcement Learning techniques for autonomous vehicles. The paper aims to cover various topics around safe RL, including value function approximation, policy optimization, model-based reinforcement learning, safe exploration techniques, and the ethical and legal implications of these advancements.

2. Foundations of reinforcement learning

Reinforcement Learning stands as a key subdomain within machine learning, holding considerable promise for autonomous vehicle development. Reinforcement learning has emerged as a powerful tool for instructing autonomous systems on effective decision-making and control mechanisms, especially in challenging scenarios. These algorithms leverage the challenging scenarios and policies to iteratively learn an optimal policy that guides the decision-making processes of the agent.

Prominent RL algorithms such as *Q*-learning, policy gradient methods, and advanced models like Deep *Q*-Networks, Proximal Policy Optimization (PPO), and Trust Region Policy Optimization (TRPO) represent the diverse set of approaches. The selection of a specific algorithm, along with its configurations, depends upon the unique characteristics and complexities of the given problem. This well-known adaptability highlights the importance of modifying the reinforcement learning methodologies to the specific demands of autonomous vehicle scenarios, suggesting the complicated relationship between algorithmic selection and problem-specific considerations. Fig. 2 shows the various states in developing and training autonomous driving systems using reinforcement learning. Deep reinforcement learning combines deep neural networks and reinforcement learning to create self-driving systems that can make decisions in real-world situations [8–17].

2.1. Markov decision processes (MDPs)

A vehicle operating in a dynamic environment and executing actions (such as steering, accelerating, and braking) to navigate and optimize a cumulative reward (such as arriving at a destination safely and efficiently) can be described as autonomous driving. In this setting, modeling and resolving RL challenges require a fundamental understanding of MDPs [18]. Mathematically, RL can be modeled by MDP [18], which includes State (s), Action (a), Transition Probability (p) and Reward Function (r). An MDP represents a set of states, denoted as S, defines all possible situations or configurations of the environment.

- *Possible Actions*: There is a set of possible actions, denoted as A, that the agent can take in each state. Actions represent the choices available to the agent to influence the environment.
- *Transition Probability:* Often denoted as, $P(s' \mid s, a)$, defines the probability of transitioning from one state (s) to another state (s') when the agent takes action (a). It characterizes how the environment behaves in response to agent actions.
- *Reward Function:* Denoted as $R(s, a, s')$, defines the immediate reward the agent receives when it transitions from state (s) to state

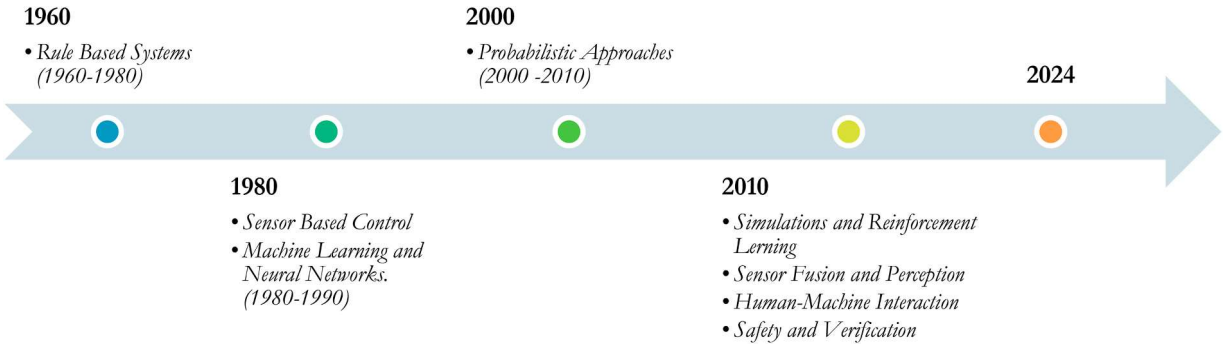


Fig. 1. Autonomous Vehicles Timeline [1].

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

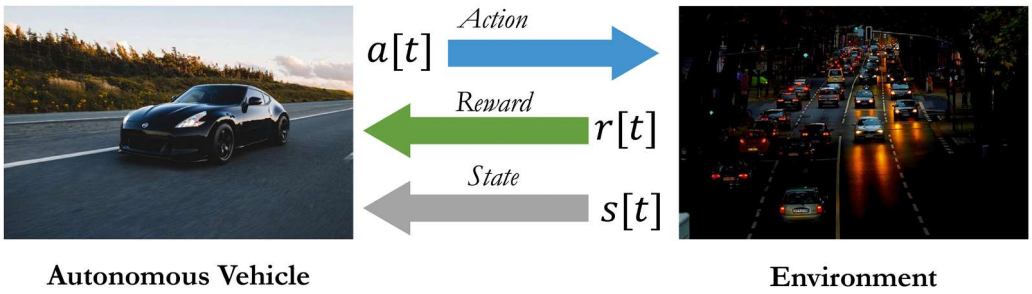


Fig. 2. Action, Reward, and State in developing and training autonomous driving systems using reinforcement learning.

(s') by taking action (a). It quantifies the desirability of a particular state-action pair.

- *Policy*: Denoted as $\pi(a \mid s)$, defines the agent's strategy for selecting actions in each state. It specifies which action to take based on the current state. MDPs tries to find an optimal policy, indicated by the symbol $\pi*$, that maximizes the predicted cumulative reward over time. The best policy can be found by solving MDPs using a variety of algorithms and techniques.

The use of MDPs for autonomous cars aims to find the best course of action that will maximize rewards over time. Through interactions with the environment, various reinforcement learning algorithms, including *Q*-learning, Proximal Policy Optimization (PPO), and Deep *Q*-Networks (DQN), can be used to learn the best policy. These algorithms enhance driving safety and behavior by adjusting the policy in response to input (rewards). The control systems of autonomous vehicles can be developed and optimized by modeling autonomous driving decisions as an MDP. This allows the vehicles to navigate complex and dynamic environments while meeting safety requirements and accomplishing their goals.

Markov Decision Processes serve as an organized and formalized framework for precisely defining the communication dynamics between an autonomous car and its surroundings. This formalism brings a level of clarity and ease of understanding to the decision-making process by providing a structured description of states, actions, transitions, and rewards. The complexities of autonomous driving require a robust strategy for managing uncertainties arising from factors such as sensor noise, dynamically changing road conditions, and the unpredictable behaviors of other drivers. MDPs address this challenge by incorporating probabilistic reward functions and stochastic transitions, enabling autonomous cars to make well-informed decisions even when faced with uncertain conditions. When MDP-based techniques are coupled with reinforcement learning algorithms, they offer a dynamic framework that empowers autonomous cars to continuously adjust and enhance their driving techniques over time. This adaptability becomes essential in effectively handling a diverse spectrum of evolving driving situations, ensuring that autonomous vehicles can navigate safely and efficiently in the face of real-world uncertainties.

MDPs serve as a foundational platform for advancing research and development in autonomous driving. They enable the exploration of complicated algorithms for decision-making, contributing to the creation of more adept and safe autonomous vehicles. MDPs provide a structured framework for identifying optimal policies that maximize anticipated cumulative rewards. This enables the training of autonomous cars to make decisions aligned with long-term objectives, such as reaching a destination swiftly while avoiding accidents and traffic violations. One key attribute of MDPs is their capacity to offer transparency in decision-making. This transparency ensures that the learned policies adhere to ethical and safety standards. Researchers can scrutinize and assess these policies, fostering the development of autonomous systems that are not only capable but also accountable and transparent in their decision-making processes. Fig. 3 shows the block diagram of the Markov decision processes used to find an optimal policy operator close to maximizing the expected cumulative reward over time.

2.2. Bellman equations

Bellman Equations play a key role in dynamic programming and reinforcement learning algorithms, providing a robust framework for making informed decisions within Markov Decision Processes (MDPs) [19]. These equations, formulated as recursive expressions, serve to decompose the value associated with a given state or state-action pair into two fundamental components: the *immediate reward* and the *expected value of subsequent states*. Bellman equations encapsulate the

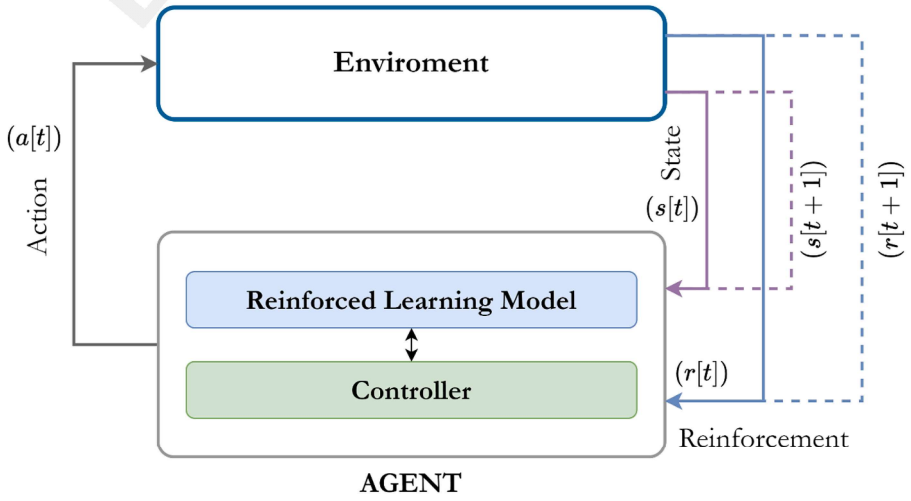


Fig. 3. Markov decision processes aim to find an optimal policy operator close to maximizing the expected cumulative reward over time [18].

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

dynamic nature of decision-making processes in MDPs by breaking down the overall value into these constituent elements, which enables algorithms to systematically assess and optimize decisions based on a comprehensive understanding of both immediate gains and anticipated future outcomes. This recursive nature facilitates the iterative refinement of decision-making strategies, allowing for the continual improvement of autonomous systems within the MDP framework.

A. *Value Function Bellman Equation* (*for states*): The Value Function Bellman Equation provides a mathematical representation for computing the value of a state, denoted as $V(s)$ which is essential in reinforcement learning algorithms and dynamic programming for autonomous systems. The value of a state, $V(s)$, signifies the expected cumulative reward that an agent can accumulate when initiating its journey from that specific state and adhering to the optimal policy, denoted as $\pi*$. This expectation encapsulates the notion of the agent's ability to navigate the environment effectively, factoring in both immediate rewards and the anticipated long-term benefits dictated by the optimal policy. These equations serve as a foundational element in formulating and optimizing strategies for autonomous systems, enabling them to make informed decisions that align with the principal objectives and policies within the MDP framework. The Bellman equation for the value function for overall actions a is expressed as:

$$V\ (s) = \max[P\ (s'|s,\ a) *\ (R(s,\ a,\ s') +\ \gamma V\ (s'))] \tag{1}$$

where γ is the discount factor, which determines the importance of future rewards.

B. *Q-Value (Action-Value) Bellman Equation* (*for state-action pairs*)*:* It is denoted as $Q(s,\ a)$, and provides a mathematical expression for computing the expected cumulative reward that an agent can accumulate. Specifically, it defines the expected cumulative reward when the agent initiates its journey from a particular state, takes a specific action, and subsequently follows the optimal policy, denoted as $\pi *$, thereafter. $Q(s,a)$ encapsulates the agent's anticipation of cumulative rewards, in terms of both the immediate consequences of taking action a in state s and the expected rewards from adhering to the optimal policy thereafter. Q-value is essential in reinforcement learning algorithms and dynamic programming. It enables autonomous systems to systematically assess and optimize their actions based on a comprehensive understanding of the long-term outcomes of specific state-action pairs. Through the iterative refinement of Q-values, autonomous systems can continually enhance their decision-making strategies and adapt to varying environmental conditions. The Bellman equation for the Q-value is expressed as:

$$Q(s,\ a) =\ P\ (s'|s,\ a) *\ (R(s,\ a,\ s') +\ \gamma \max(Q(s',\ a'))) \tag{2}$$

Solving Bellman equations through direct solutions or iterative methods like Value Iteration or Q-Learning algorithms enables the agent to discover the optimal policy, optimizing its expected cumulative reward over time. In MDPs, Bellman equations provide a comprehensive and formalized framework for modeling decision-making in autonomous vehicles. This formalism facilitates organized and understandable problem definition and is a robust tool for navigating complicated, unpredictable, and dynamic environments. In the context of autonomous vehicles, adhering to safety, effectiveness, and compliance with traffic laws is given priority. Bellman equations, characterized by stochastic reward functions and probabilistic transition models, adeptly account for uncertainty, enabling vehicles to make well-informed decisions even in ambiguous situations.

2.3. Exploration vs exploitation trade-off

The exploration-exploitation trade-off presents a fundamental challenge in designing safe RL methods. This trade-off requires that the agent decides whether to explore new actions to gather valuable information or exploit known actions to maximize immediate rewards. Finding a delicate balance between exploration and exploitation is critical for the implementation of RL algorithms. Autonomous vehicles face the imperative of navigating this trade-off judiciously. To develop optimal strategies, AVs must explore various behaviors, allowing them to gather insights and adapt to diverse scenarios. However, this exploration must be conducted cautiously to avoid risky actions that could compromise safety. Various techniques have been devised to address the exploration and exploitation trade-off in this context. These include the epsilon-greedy exploration strategy, which involves choosing the optimal action with a high probability and exploring with a small probability. Additionally, methods like soft actor-critic provide a framework for balancing exploration and exploitation through probabilistic action selection. Furthermore, safe exploration strategies are employed to ensure that exploration activities do not jeopardize the safety of the autonomous vehicle. These strategies incorporate measures to mitigate risks during the exploration phase, aligning with the principal goal of developing robust and secure autonomous systems [11,14,20]. Fig. 4 shows the tradeoff between the exploration and exploitation.

In the exploration phase, autonomous vehicles actively gather information about their surroundings, encompassing road conditions, the behaviors of other drivers, and potential obstructions. This phase is crucial for constructing an accurate environmental model, as it provides essential knowledge that aids in the identification and mitigation of potential dangers. For instance, navigating challenging roads or unfamiliar routes enables the vehicle to anticipate difficulties, enhancing its ability to respond adeptly to unforeseen circumstances. The data collected during exploration is instrumental in training machine learning models within perception and decision-making systems, contributing to improved model performance through informative and diverse datasets.

Subsequently, exploitation comes into play as the autonomous vehicle utilizes its accumulated knowledge to make informed judgments. Exploitation prioritizes driving habits that are predictable and safe, aligning with the principal goal of ensuring safety and adherence to traffic regulations. Utilizing its expertise, the autonomous vehicle maintains a safe following distance and refrains from engaging in risky maneuvers.

While the primary objective of an autonomous vehicle is to offer efficient and rapid navigation to a destination, exploitation aids in focusing on this goal. It involves optimizing the vehicle's performance, incorporating aspects like route planning and effective traffic management based on its current understanding of its surroundings. The dynamic interaction between exploration and exploitation is essential for the autonomous vehicle to continually enhance its decision-making abilities and effectively navigate diverse and evolving driving scenarios.

3. Challenges in applying reinforcement learning to autonomous vehicles

The integration of reinforcement learning into autonomous vehicles

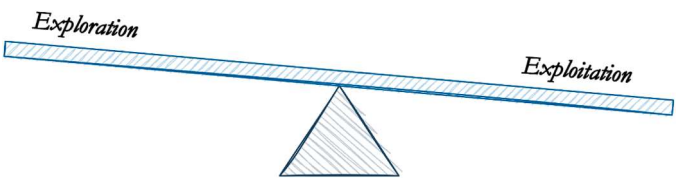


Fig. 4. Balance between exploration and exploitation to achieve specific goals and safety requirements of the autonomous system.

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

holds excellent promise but poses formidable challenges. Several fundamental problems must be addressed to successfully incorporate RL into autonomous driving systems, including safety considerations, sample inefficiency, and the ability to handle rare and unforeseen situations. Safety takes precedence in autonomous driving, given the potential risks associated with accidents or system failures [11,14]. Rigorous testing and validation procedures are imperative to ensure the safe behavior of autonomous driving systems across a diverse spectrum of scenarios. These systems must adhere to traffic rules and signals in variable environments, necessitating the development of accurate algorithms for interpreting and following traffic regulations. Handling rare, unpredictable events, such as accidents or extreme weather conditions, poses a particular challenge, requiring robust responses from the autonomous vehicle.

RL algorithms deal with sample inefficiency, requiring substantial data for effective policy learning. Collecting real-world driving data is time-consuming, expensive and entails inherent risks. Simulations offer a solution by generating additional training data, yet designing realistic and accurate simulators presents its own set of complexities. Closing the domain gap between simulation and real-world driving conditions remains challenging to ensure the efficacy of RL models trained in simulation when deployed in reality.

Generalizing learned behaviors to diverse and novel scenarios is essential for autonomous vehicles. AVs must navigate various urban environments with distinct traffic rules and norms, requiring effective knowledge transfer. A common approach involves pre-training in simulated environments and fine-tuning in real-world settings. Moreover, autonomous vehicles must handle rare and unusual scenarios, even if not encountered during training, emphasizing the need for robustness in their capabilities. The ultimate objective is to ensure the safe and reliable operation of autonomous vehicles in diverse real-world scenarios, optimizing for efficiency and generalization. This involves addressing challenges related to safety, sample inefficiency, and adaptability to unforeseen circumstances in the pursuit of seamless integration of RL into autonomous driving systems.

4. Safe reinforcement learning techniques

Safe reinforcement learning (SRL) approaches are crucial for training autonomous systems in a way that reduces the risk of accidents and guarantees compliance with safety regulations. Safe RL techniques incorporate the safety constraints into the training framework, which includes limiting the exploration space, a reward function that penalizes the risky actions, model-based approaches that simulate the various scenarios before executing the actions in the real world, etc. Some of these approaches are discussed in the following subsections.

4.1. Value function approximation

Value Function Approximation is an effective technique within RL, enabling the field to address intricate and high-dimensional problems. This technique plays a key role in enhancing the efficiency and scalability of RL algorithms, particularly in scenarios where the exhaustive computation of the value function is computationally expensive. In RL, the value function, denoted as Q, serves as a quantitative measure of the expected return that an agent can anticipate when starting from a specific state $(V(s))$ or a state-action pair $(Q(s,\ a))$. This value function is instrumental in the decision-making process, offering the agent a means to evaluate the desirability of different states or state-action pairs [21,22]. This assists the RL algorithms in making informed decisions while solving high-dimensional problems where the number of states or state-action pairs are huge. Thus, it enables the agents to navigate and learn more efficiently while handling complex and large-scale problems. Also, the RL algorithms have the flexibility to learn, adapt and improve their decision making using the feedback mechanism in terms of rewards or penalties based on the agent's actions. The agent can optimize its actions by learning the optimal policies that maximizes the expected rewards.

Let $\widetilde{Q}(s,a)$ represent the approximated function that determines the penalized future return an agent can anticipate after taking an action a at state s.

$$\widetilde{Q}(s,a) \approx \mathbb{E}\Big[\sum \gamma^t\ \mathscr{R}(s_t,a_t)\ |\ s_0=s,\ a_0=a\Big] \tag{3}$$

where, $\mathbb{E}$ is the expectation operator, $\gamma \in (0,1)$ is the discount factor with decreasing value for future rewards, $\mathscr{R}(s_t,a_t)$ is the reward received at time t when agent takes the action a at state s and $s_0,\ a_0$ represents the initial state and action.

The value function approximation offers the benefits of improved *computational efficiency*, as it reduces the requirement for storing and updating each state; *better generalization* for unforeseen situations, as it enables the agent to anticipate the values for unseen states; and *scalability*, as the algorithms will be able to handle vast state spaces, where tabular approaches would fail. The benefits of value approximation functions still face certain challenges such as, approximation errors, overfitting, bias and fairness, and the tradeoff between exploration and exploitation. Some of the value function approximation techniques such as Deep Q-networks and Double DQN are discussed in more detail as follows.

A. Deep Q-Network

For most of the real-world problems, it is unfeasible to obtain the Q-function table containing the Q values of each combination of state s and action a. Function approximators provide a feasible solution to address this problem. To estimate the Q-values, $Q(s,a;\theta) \approx Q^*(s,a)$, by minimizing the loss function, given mathematically as [23]:

$$L_i(\theta_i) = \mathbb{E}_{s,a,r,s'\sim\rho(\cdot)}\left[(y_i - Q(s,a;\theta))^2\right] \tag{4}$$

where, y_i is the temporal difference and is given as $y_i = r + \gamma max_{a'} Q(s', a';\theta_{i-1})$, term $y_i - Q$ is the temporal difference error, ρ is the behavior distribution over transitions $\{s,a,r,s'\}$ in the environment.

Deep Q-Network (DQN) is a foundational framework in deep reinforcement learning (deep RL) known for its proficiency in handling complex problems with high-dimensional input spaces. At its core, DQN employs a neural network to approximate the Q-function, representing the expected cumulative reward for specific actions in given states. To enhance stability during training, DQN utilizes techniques such as experience replay, which samples from stored past experiences and target networks, introducing a separate network with delayed updates. DQN's adaptability is further extended by incorporating safety considerations, allowing the algorithm to penalize actions leading to unacceptable conditions. Safety restrictions, often implemented through reward shaping, ensure that DQN prioritizes learning policies adhering to predefined safety guidelines. DQN has demonstrated applications in real-world environments, particularly in autonomous systems and robotics, showcasing its effectiveness in navigating dynamic and complex spaces.

$$Q(s,\ a) = (1\ -\ \alpha) * Q(s,\ a) + \alpha\ *\ [r\ +\ \gamma \max(Q(s',\ a'))] \tag{5}$$

where, $Q(s,\ a)$ is the Q-value of state s and action a, representing the expected cumulative reward when taking action a in state s. α (alpha) is the learning rate, which controls how much the Q-values are updated in each iteration and has a value between 0 and 1. r is the immediate reward received after taking action a in state s. γ (gamma) is the discount factor, representing the importance of future rewards and has a value between 0 and 1. s' is the next state resulting from taking action a in state s. a' is the action that maximizes the Q-value in the next state s' (i.e., $a' = argmax(Q(s',\ a'))$).

Deep Q-Network (DQN) serves as a valuable technique for

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

autonomous cars to acquire and adapt driving rules, employing a deep neural network to estimate *Q*-values for diverse driving actions across varying conditions. These actions involve crucial maneuvers such as lane changes, turns, braking, and acceleration. DQN framework integrates the principles of exploration and exploitation, authorizing the autonomous vehicles to actively balance learning new behaviors with leveraging effective strategies. This ensures the vehicle explores new possibilities while still relying on established safe and efficient actions. During the exploration phase, the vehicle operates randomly, gathering information to understand its surroundings, while exploitation involves selecting actions with the highest estimated *Q*-values to maximize rewards.

DQN-equipped autonomous cars exhibit continuous learning and adaptation, crucial for navigating diverse road conditions, interpreting traffic patterns, and responding to unforeseen circumstances. The flexibility of DQN models facilitates transfer learning, enabling the application of knowledge gained in one driving situation or environment to enhance performance in another. This dynamic approach empowers autonomous vehicles to evolve their driving capabilities continually.

Fig. 5 shows the processes of training and utilizing a DQN for reinforcement learning. The deep Q-learning algorithm has two phases: a) sampling phase and b) training phase. In the sampling phase. The algorithm is given below as:

Algorithm: Deep Q-learning Algorithm [29]

Initialize: D (replay memory), N (capacity) and Q-values with random weights
for *each iteration* **do**
  Select random action a with probability ρ
  Otherwise, **select** $a = max_{a'} Q(s', a'; \theta_{i-1})$,
  Execute action a
  Calculate reward r and new state s'
  Store transition $\{s, a, r, s'\}$ to D
  Sample random experience $\{s_i, a_i, r_i, s_i'\}$ from D
  Set $y_i = \begin{cases} r_i & \text{if } s' \text{ is terminal state} \\ r_i + \gamma max_{a'} Q(s', a'; \theta_{i-1}) & \text{otherwise} \end{cases}$
  Minimize the loss function. $L_i(\theta_i) = \mathbb{E}_{s,a,r,s' \sim \rho(\cdot)}\left[\left(y_i - Q(s, a, ; \theta)\right)^2\right]$
End

In 2015, the use of deep reinforcement learning in autonomous vehicles was explored [7]. The study involved using the DQN algorithm on the Atari 2600 video game and Alpha Go to control autonomous vehicles. The results showed that the implemented algorithm was able to improve decision-making and learning in unexpected situations. Despite the safety regulations in the automobile industry, which favor supervised learning, DQN is seen as a promising solution for autonomous vehicles in the future.

In [24], an alternative solution to the urban driving problem was addressed by introducing a threshold lexicographic *Q*-learning deep learning approach, which is a multi-objective deep reinforcement learning technique. The multi-objective DQN agent is trained to drive on highways with multiple lanes and at crossroads. It learns to safely change lanes and yield according to traffic laws. In addition, an enhancement to the DQN architecture for factored Markov Decision Processes is proposed, which further improves the *Q*-function. This approach has been shown to increase data efficiency by learning factored *Q*-functions and using them as auxiliary features during training. The agent has the ability to navigate junctions and crowded highways safely while adhering to traffic regulations.

In [25], a framework for obstacle avoidance in autonomous vehicles using the deep *Q*-network (DQN) approach and the CARLA simulator was developed. The aim of the work was to ensure safety while minimizing costs by training the DQN algorithm in a virtual environment before testing it in real-life scenarios. The autonomous vehicle drives on a road simulated by the CARLA simulator [26], which also simulates the motion of an obstacle vehicle parked on the road. Inputs such as target position, real-time position, velocity, and information from the LiDAR point cloud were used, while outputs consist of action settings involving acceleration, braking, and steering. The DQN behavior trend improves as the number of training episodes increases.

In [27], the DQNs have been employed for the development of a lane-changing system. The system integrates a request-respond mechanism for governing the mandatory and optional lane changes by devising the separate request and respond groups and incorporating a central agent to streamline the training process. Both these groups were trained individually, where the request group has been specifically trained by considering the states of the groups and the respond group has been trained considering the superimposing actions of the group. The methodology offered better lane maneuver and is also computationally inexpensive.

In [28], proposes the development of analytical methods for the

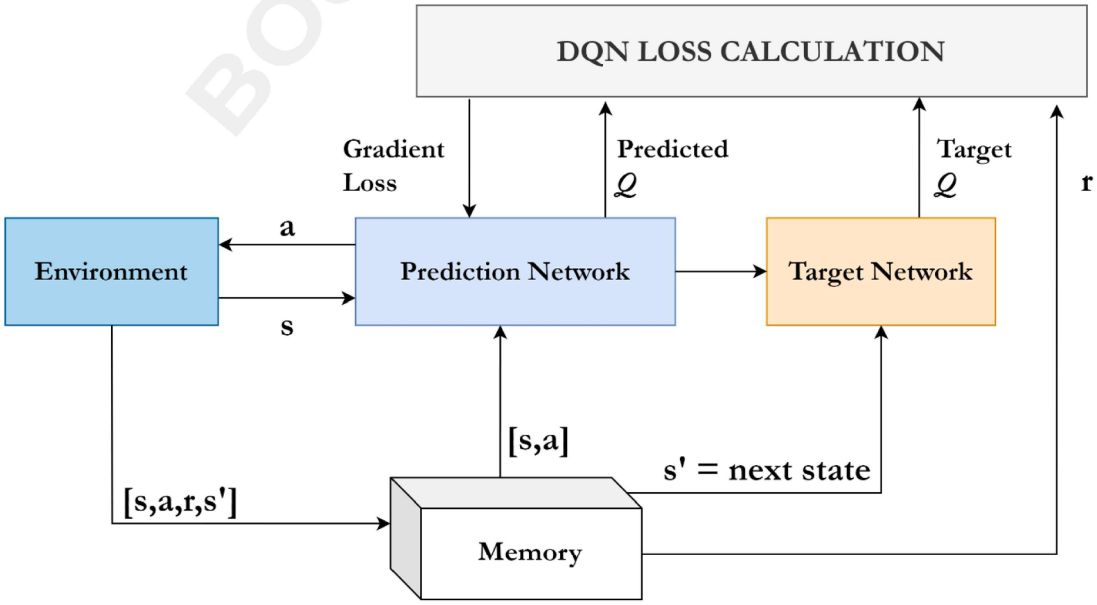


Fig. 5. Block diagram of training and utilizing a DQN for reinforcement learning [30].

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

formulation of the reward function for vehicle velocity control. The problem has been posed as the optimization problem, where the objective is to determine the expected optimal velocity by optimizing the reward functions for deep *Q*-networks.

The use of DQNs offers flexibility to the autonomous vehicles to adapt to various driving scenarios and learn from the gained experiences, this also aids in improving the decision-making capabilities when dealt with new experiences and ensuring efficient and safer navigation of these autonomous driving systems.

B. *Double DQN and Dueling DQN*

Deep *Q*-networks are faced with the problem of learning unrealistic high action values while determining the temporal-distance from the target; it is also evident in the equation, $y_i = r + \gamma max_{a'} Q(s', a'; \theta_{i-1})$, where a **max** operator has been used. These overestimations may or may not impact the performance of the agents, yet obtaining the over-optimistic values might lead to biases towards higher action values. If these overestimations are not uniform, it might lead the agent to learn more and might adversely impact the quality of the resultant policy. Double Q-networks have been introduced to reduce these over-estimations by using two networks to independently estimate the target and action values [31,32]. Double DQN rectifies the overestimation bias of regular DQN. In order to enable steadier and effective learning, dueling DQN divides the value function into state and reward streams. Using these methods can increase safety by giving more precise value estimates. Fig. 6 shows the processes of training and utilizing a DDQN for reinforcement learning.

$$Q(s,\ a)\ =\ (1-\alpha) * Q(s,\ a) + \alpha * \left[r + \gamma\ * Q_{target}(s',\ \mathbf{argmax}(Q(s',\ a')))\right] \tag{6}$$

Algorithm: Double Q-learning Algorithm [31,32]

Initialize: Q_θ (Main network), $Q_{\theta'}$ (Target network), D (replay memory) and $\alpha \ll 1$
for *each iteration* **do**
  for *each environment* **do**
    Observe state s and **select** action $a \sim \pi(a,s)$
    Observe action a, and **observe** next state s' and reward $r = R(s,a)$
    Store $\{s, a, r, s'\}$ in replay memory **D**
  for *each update* **do**
    Sample $e = (s, a, r, s') \sim D$
    Compute target Q value
      $Q^*(s,a) \approx r + \gamma\ Q_\theta(s',\ argmax_{a'}(s', a'))$
    Evaluate loss function $(Q^*(s,a) - Q_\theta(s,a))^2$
  Update target network parameters
$\theta' \leftarrow \alpha * \theta + (1 - \alpha) * \theta'$
End

In order to reduce the need for substantial retraining in new situations, DDQN models can help with transfer learning, which allows knowledge obtained in one driving scenario to be applied to another, where autonomous cars with DDQN models can continually adjust to new information and changing circumstances, which makes it easier for AVs to react to changing traffic situations. The main network is used to estimate *Q*-values for actions in a certain state during training. On the other hand, the value of the action chosen by the main network is estimated by the target network. In difficult and dynamic driving situations, this '*double*' estimation aids in lowering *Q*-value overestimation, improving policy learning and decision-making. The DDQN promises learning effectiveness, stability, and safety of reinforcement learning agents, making it a valuable technique for training autonomous vehicles.

In [34], an automated decision-making technique to regulate the speed of a vehicle is proposed using reinforcement learning, specifically double *Q*-learning, to control the speed of a car based on the environment created by realistic driving data. The methodology integrates a

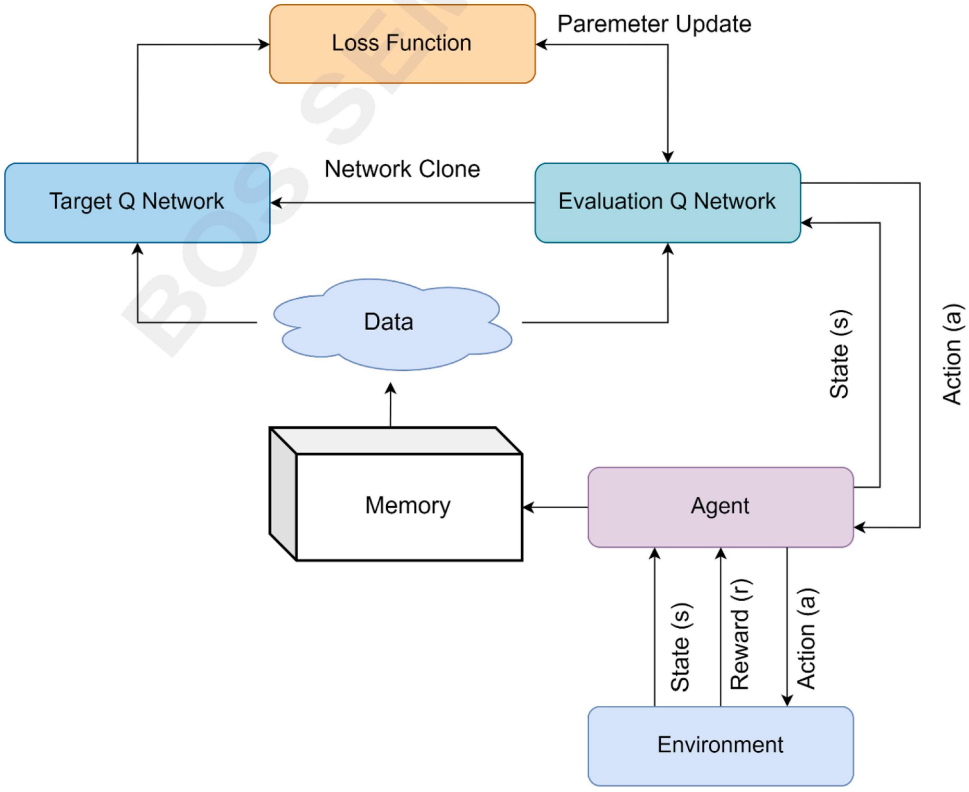


Fig. 6. Block diagram of training and utilizing a DDQN for reinforcement learning [33].

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

perception approach that is proposed to design the environment, based on direct perception approach. It is found that certain Atari 2600 games exhibit instability using the popular *Q*-learning method, which is why double Q-learning is used instead. In the experiments it has been found that the double DQN outperformed the DQN algorithm in terms of both value accuracy and policy quality. The score of DDQN was 271.73 % higher than that of DQN.

In [35], a new approach based on double *Q*-learning was implemented for automated driving systems. By using DDQNs, the highest anticipated value for any number of random variables can be predicted, and the maximum expected value will be underestimated rather than exaggerated. Additionally, the authors used an A^* based optimization technique to enhance routing for automated driving. The suggested technique, based on double *Q*-Learning and A^*, was tested on a test environment with randomly placed obstacles. The results were compared with those obtained using *Q*-learning alone, and the suggested technique was found to perform better in terms of distance traveled and collisions with barriers.

In [36], authors implemented reinforcement algorithms for path finding on Raspberry Pi. The goal of the study was to determine the best path while avoiding collision with static barriers. To achieve this, *Q*-learning and Deep *Q*-learning algorithms were used. The method calculates and navigates the shortest path between a source key point and a destination key point in order to carry out the intended delivery. The study implemented these two algorithms on a robot. Deep *Q*-Network (DQN) was implemented as it solves the space and time issues that arise with *Q*-learning algorithms. With *Q*-learning, the number of states increases the amount of memory and processing time needed to save and update the table. The main network and target network of the DQN algorithm replace the standard *Q*-table and for each *n*-steps, the weights from the main network are copied to the target network.

In [37], a comparative study on the conventional, double, dueling, and priority replay DQNs is presented for the navigation of the autonomous vehicles on the highways. The work explores the role of training algorithms on the convergence rates and the experimental evaluation of the performance of these DQNs, and it has been reported that the dueling DQNs and policy-regulated DQNs offered much better maneuvering in highway driving scenarios.

In [38], an improved dueling *Q*-network based on edge-enhanced graph attention RL has been proposed for the decision-making in autonomous vehicles at road intersections. The EGARL framework utilizes the interaction amongst the agents to make rational decisions. The framework models the environment as an edge-enhanced graph attention model and includes the representation of the topologies and interaction in the driving scenario; and the deep reinforcement learning is employed to make driving decisions considering the current state and the features extracted by the graph.

In [39], a dueling double deep *Q*-network is proposed for learning the dispatch policy by interacting with the digital twin for determining the optimal routing of an automated guided vehicle. The routing problem has been modeled as an MDP and the use of dueling DDQN is proposed for learning the best policies of A^* and dynamic collision avoidance. This results in short delays, better stability and reduced power demands.

Double deep *Q*-networks can aid in improving the overall performance of the autonomous vehicles by addressing the problem of overestimation in standard DQNs by using two separate networks and also assure more accurate and reliable estimation of the *Q*-values and action values. This enables the agents to make informed decisions effectively and ensure a higher level of safety in complex driving scenarios.

4.2. Policy optimization methods

Policy optimization methods are a class of reinforcement learning algorithms that directly optimize the policy function, which specifies the agent's behavior by mapping states or state-action pairs to actions. Policy optimization techniques look for the policy that directly maximizes the projected cumulative reward as opposed to other RL approaches that concentrate on estimating value functions. These techniques have gained acceptance because of their efficiency in dealing with continuous control problems and high-dimensional action environments [40].

Autonomous cars can be taught to identify the best policies, or strategies, for making decisions in complex and changing settings through the application of a class of policy optimization methods. The aim of these techniques is to maximize cumulative rewards by directly optimizing the policy function that associates states with actions. By tackling difficult, long-term choice problems and striking a balance between exploration and exploitation, it can enhance decision-making. It may be applied to the training of autonomous cars for continuous control jobs like acceleration and steering. For precision control, it assists the car in determining deterministic driving plans.

In [42], a deep deterministic policy gradient algorithm based on lateral and longitudinal control for autonomous driving is presented. The aim was to address the problems of traditional path planning algorithms such as instability, lack of robustness, and safety issues, especially during complex road conditions. A combined Reinforcement Learning with Deep Deterministic Policy Gradient (DDPG) Algorithm was proposed that assures that the cars are more stable and reduces the time required for effective learning. The combined approach helps the agent in making complex decisions in a complex environment. (Fig. 7)

A. *Proximal Policy Optimization (PPO)*:

PPO is a technique for policy optimization intended for secure and reliable reinforcement learning [43]. In order to ensure that the policy does not alter much between updates, it uses a clipped surrogate objective. PPO is useful for applications like autonomous driving where dangerous behaviors must be avoided due to its safety and stability characteristics. The agent's policy is learned and represented by the *actor*, which is a neural network or a part of a neural network. The method by which the agent chooses what to do in a certain state is specified by the policy. During training, actions are sampled using the probability distribution that the actor produces over the possible actions. Apart from the *actor*, there is another neural network called the *critic*, and its job is to estimate the value function. Assuming the agent adheres to its present policy, the value function calculates the predicted cumulative reward (return) from a certain state.

$$A(s,a) = Q(s,a) - v(s) \tag{7}$$

Here $v(s)$ is the approximate value of the average of all Q values

Autonomous cars can be trained using PPO to enhance their decision-making rules. PPO provides stability and safety assurances within the framework of reinforcement learning, making it an excellent choice for training autonomous cars. PPO is capable of efficiently managing continuous action spaces. The goal of the reward system is to promote effective and safe driving. It may cover aspects like keeping to the designated lanes, keeping a safe following distance, adhering to traffic laws, and using as little energy as possible. PPO-based models allow AVs to continuously learn from and adjust to novel situations, shifting traffic patterns, and changing road conditions. By engaging with the environment, the AVs gather data iteratively, which is then used to update the policy. PPO algorithms employ the surrogate objective to optimize the policy network parameters.

In [44], a twin-*Q* soft actor-critic (SAC) mechanism has been proposed for the navigation of the tractor-trailer vehicles around the roundabouts of varying diameter. The work tries to address the problem that a trailer has a tighter radius than the front wheels of the truck, if the steering angle is not optimal, it may lead the trailer to collide with the kern of the roundabout. The usage of twin-*Q* SAC aids in maneuvering the truck-trailer near to the center of the road and thus was able to avoid the collision with the roundabout with a success rate of 73 %. In [45] a

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

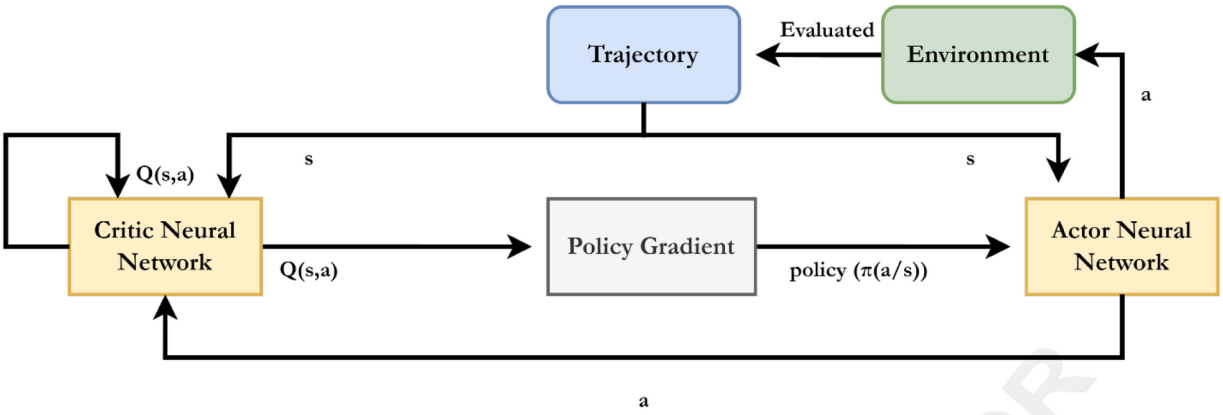


Fig. 7. Block diagram of PPO used in autonomous vehicles [41].

novel adversarial PPO is reported by integrating the adaptive KL penalty. In [46], the PPO has been implemented for the path following for an autonomous vehicle. The PPO has been employed to determine the optimal design parameters such that the maneuvering vehicle offers an optimal transient response.

In [47], a weight varying MPC is proposed using deep reinforcement learning. The work proposes the development of multi-objective Bayesian optimization to determine the Pareto-optimal gains of the MPC controller and these pre-optimized weights are used to govern the RL actions within a safe learning space. This procedure ensures that even an untrained agent will be able to guarantee safe and optimal behavior.

The use of PPO algorithms in the autonomous vehicles will enable continuous improvements in the driving behaviors and also enable the vehicles to adapt to the new scenarios and improve their efficiency. Thus, the PPO algorithms in the autonomous vehicles guarantee improved efficiency, safe driving behavior and reliability.

B. *Trust Region Policy Optimization (TRPO)*:

When safety is a key priority, TRPO [48] is particularly helpful because it optimizes policies under a trust region restriction that limits how far the policy can diverge from the preceding policy. This constraint encourages stability and helps prevent significant policy revisions that could result in risky behaviors.

TRPO is renowned for its capacity to limit the maximum change in the policy, ensuring steady and secure policy updates. TRPO facilitates transfer learning, enabling the application of knowledge acquired in one driving scenario or setting to another. This makes it possible to quickly adjust to changing driving situations without requiring significant retraining. TRPO is a good choice for AV training because of its trust region limitation, which guarantees safe and progressive policy changes. In real-world situations, the TRPO algorithm can aid autonomous vehicles in making better driving judgments, navigating intricate and dynamic settings, and prioritizing efficiency and safety. AVs gather data through interactions with the environment repeatedly using TRPO algorithms, and then they change the policy depending on the trust region limitation. TRPO aims to maximize the total rewards while staying within the trust region.

In [49], an RL-based controller was developed for the vehicle path following at high speeds. At high speeds, assuring good control is quite challenging, as the vehicles have to deal with the non-linear nature of the system dynamics and also there is a strong coupling between the longitudinal and lateral control. To deal with these problems at high speeds, a deep soft actor-critic based RL controller was introduced, along with the development of a lane curvature estimator. The work included the utilization of the TRPO for the minimization of the cost function to learn the policy. It was found that the RL based controller offers a better response when compared to imitation based learning and model-based optimal control methods.

In [50], to address the safety concerns in conventional RL, a methodology based on twin delayed deep determinizing policy gradient was proposed to assure safe actions. The method ensures that the agent adjusts to the safety correction factor to determine an optimal tradeoff between exploration and exploitation phase. Also, the method is model free and does not require any prior information, and the use of safety correction factor assures safe behavior during training and deployment.

In [51], a tiered framework is proposed for high-speed maneuvers in multi-lane autonomous driving. The framework amalgamates the DRL with rule-based methods, where at the higher level the DRL governs the instantaneous high-speed maneuvers and at the lower-level rule-based logic is implemented for car following and three-stage lane changes. The reported framework offers the advantage of hybridizing the DRL with rule-based methods. The work also integrates traffic flow adaptive strategy for considering the traffic flow operations into the framework. The reported framework offered superior performance in terms of higher speeds while adhering to safety and comfort constraints, when compared to the pure-DRL and rule-based IDM-MOBIL model.

In [52], a graph Laplacian approach was proposed for computing the feature matrix from the periodically sampled data in the simulation environment to determine the optimal policy weights. Once, the weights were obtained, the final action is determined via weighted voting and greedy optimal policy calculation. The method had shown good results in cases when the ego vehicles take unexpected turns in simulated environments.

In [53], the problem of path planning under unpredictable circumstances in off-road environments was addressed using RL by means of low-dimensional occupancy grid maps. The agent was allowed to learn the features of the environment in a supervised manner using an adaptive curriculum learning methodology, befitting the agent's efficiency and generalization adeptness.

The use of trust region policy optimization methods constrains the policy updates within the trust region, thus assuring safe and stable actions by preventing the drastic behavior. This also aids in achieving a better exploration and exploitation tradeoff and facilitates safe and accurate decision making. Also, the use of TRPO aids in reducing the sensitivity of the learning process to hyperparameter tuning, assuring that the algorithm is robust and easier to implement in real-world scenarios.

4.3. *Model-based reinforcement learning*

Model-Based Reinforcement Learning (MBRL) [54,55] is a subfield of reinforcement learning that involves the use of learned or pre-defined models of the environment to assist an RL agent in making decisions and optimizing its policy. Unlike model-free RL, where agents directly learn from interactions with the environment, MBRL leverages models

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

to simulate and predict the consequences of actions, which can lead to more sample-efficient and data-efficient learning.

One of the prime steps in MBRL is the development of the learning models that capture the dynamics of the environment, including the state transitions and reward probabilities, based on the experience gained via observations and interactions, the transition model, $s' \sim p(s'|\, s,a)$, predicts the next state s' based on the current state s and action a; and the reward model, $r \sim r(\, s,a,s')$, predicts the reward at the next state s'. The environment is simulated within the model, and the MBRL agents are able to compute the potential of various actions without any real-world intervention.

The techniques like policy rollout and Monte-Carlo simulations are used in the simulations. Under policy rollout, the agent selects the actions within the simulated environment, whereas in Monte Carlo simulations, multiple simulations are evaluated for the current state and the average value of the predicted outcomes is considered. These predicted outcomes are used by the MBRL agents to optimize their policies. In model-based RL algorithms, the policy is directly optimized based on the learned model, given mathematically as:

$$\max_{\pi} \mathbb{E}\left[\sum \gamma^t\, r\left(s_t, a_t, s_t'\right) \mid \pi, M\right] \tag{8}$$

where, π is the policy, M is the learned model, γ is the discount factor, and t is the time step.

In [57], authors presented the development of deep hierarchical reinforcement learning for autonomous driving with distinct behaviors. A general architecture that uses hidden state embedding to identify control actions for autonomous driving straight from data collected from various sensors is presented and employs a deep reinforcement learning algorithm-based end-to-end autonomous driving system for heterogeneous traffic on a highway. Instead of relying on camera views, autonomous driving agents perform better when considering the relative location and speed of nearby cars by using two forms of sensor data to operate an autonomous vehicle in a simulated environment, including camera pictures and the relative positions and velocities of nearby cars to the target vehicle.

A. *Model Predictive Control (MPC)*:

Model-based control (MPC) is a frequently utilized control method in autonomous vehicles [58]. It simulates future trajectories using a learned or pre-defined model of the environment and chooses actions that maximize a given objective while adhering to safety constraints. MPC is useful for real-time control in situations requiring safety. Fig. 8 shows the block diagram of an MPC based framework for autonomous vehicle which consists of optimization and prediction model developed to provide the best output.

MPC uses a dynamic model of the vehicle to predict how various control inputs (such as steering, acceleration, and braking) would affect the AV's motion over a short time by taking the motion and kinematics of the vehicle into account. MPC functions across a finite amount of future time in order to determine the best course of action based on the present situation, the vehicle model, and the cost function, by repeatedly solving an optimization problem. The process is repeated with an updated state estimate once the control sequence has been optimized and the vehicle has been subjected to the first control action. The ability of the AV to make flexible and knowledgeable judgments is guaranteed by this real-time execution and prediction mechanism. MPC is frequently combined with other control techniques, such as PID controllers to guarantee smooth and steady operation.

In [59], an event-triggered MPC is proposed for the path following problem in autonomous driving to address the computation and communication costs. The work proposes an efficient MPC (eMPC) framework that utilizes the techniques like prioritized experience replay (PER) and long-short term memory (LSTM) networks to learn the optimal event triggering policy following a model-free approach. The eMPC framework incorporates three RL methods of DDQN, SAC and PPO to solve it. It has been found that PPO-EMPC with LSTM and DDQN-EMPC with PER & LSTM offered a superior performance.

In [60], an imitation learning-based approach has been proposed for the design of a differentiable MPC for lane keeping in autonomous driving. A tiered learning-based approach has been formulated considering the safety, comfort and human driver imitation as the objectives. The study also incorporates the consideration of the open loop system behavior along with the closed loop behavior that results in more robust learning and more human like driving behavior.

In [61], an RL based nonlinear MPC has been proposed for the lane keeping task in autonomous driving. The work considers the use of tabular *Q*-learning to determine the actions. The robustness of the methodology has been evaluated in simulations and also in real-world scenarios, where a scaled version of the vehicle is considered with large deviations in the sensor measurements and other external interferences. The RL-NMPC assured good robustness and performance in both the scenarios.

In [62], a two-stage decision-making framework is proposed for learning the left-turn policies at unsignalized intersections. The top layer implements a soft actor-critic mechanism for planning a safe trajectory, whereas at the bottom level MPC has been implemented to ensure the feasibility of the left-turn maneuver. The SAC layer is a behavior planning layer and ensures that the agent learns the optimal policies for generating the velocity and yaw angles for the ego vehicle ensuring non-conservative still safe maneuvers and is also responsible for implementing collision avoidance with respect to intersecting vehicles. The MPC ensures the motion planning for the left-turns and also

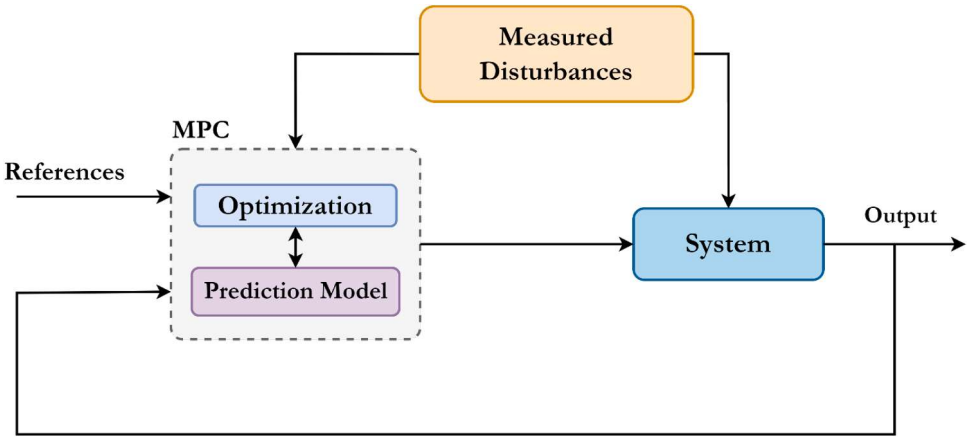


Fig. 8. Block diagram of MPC based framework in autonomous vehicles [56].

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

considers the driving comfort during the route optimization phase. It has been reported that the developed two-stage hierarchical framework offers superior behavior when compared with model free RL methods.

In [63], an interactive RL framework is proposed for evaluating the decision making of the ego vehicle in real-time by cogitating a human driver in the loop that offers efficient and safe navigation in both multi-lane and congested driving situations. The human driver evaluates the behavior of the ego vehicle concurrently and intervenes to amend the actions, the ego vehicle is rewarded if it corrects the action based on human feedback whereas poses a penalty if it doesn't follow the commands. The MPC governs the longitudinal and lateral movement of the vehicle. The framework also implements a safe driving envelope for the safe exploration and exploitation phase to guarantee safety.

The use of model predictive control in autonomous driving allows for the dynamic and adaptive decision making, by not only considering the current state of the vehicle but also by anticipating the future states. The control scheme is optimized continually, allowing the autonomous vehicles to navigate complex and uncertain road scenarios.

B. *Probabilistic Models for Uncertainty Estimation*:

Probabilistic models are frequently used in model-based RL to calculate the degree of prediction uncertainty [64,65]. As it enables the system to avoid actions with high uncertainty or conduct more careful actions in uncertain conditions, uncertainty estimate is crucial for safe decision-making and includes methods like Bayesian neural networks (BNNs) and Gaussian processes [66].

AVs analyze risks associated to various driving behaviors or actions. Through the estimation of the probability of different events, including crashes or moving infractions, probabilistic models measure the risk. When the AV is faced with unclear or uncertain circumstances, it makes decisions based on probabilistic models. Based on the predicted probability of various events, the AV can choose courses of action that reduce the projected risk. It is possible to calculate the likelihood that a specific trajectory or direction will not collide using probabilistic models. This is especially crucial for organizing complicated motions or navigating through crowded areas. Using probabilistic models for uncertainty estimates enables AVs to make safer and better-informed judgments.

In [67], multi-objective Bayesian optimization has been employed for determining the Pareto optimal set of controller parameters for designing a weight varying MPC controller. The DRL agent tries to learn the optimal policy from the Pareto optimal set of solutions, as these parameters are pre-optimized using Bayesian optimization, this assures that the untrained agent will not be incorporating any unsafe behavior during the training phase.

In [68], an uncertainty-aware model-based RL planning algorithm has been reported for resolving the prediction, planning and control objectives in autonomous driving. The variable dynamics of the traffic scene are captured using an action conditioned stochastic model. The framework utilizes an MPC controller for the maneuver control and uses a pessimistic trajectory optimizer to determine the weighted trajectories across all sampled actions and execute the one that has the least sum of return. The work also uses a greedy trajectory optimizer that averages the reward for the ensemble agents and then determines the action trajectory.

In [69], an ensemble quantile network approach is proposed for the uncertainty estimation. The method pools the distributed RL with ensemble methods. The actions are learned by the quantile function to estimate the aleatoric uncertainty, whereas the ensemble agents are trained using the Bayesian optimization to determine the epistemic uncertainty. It is reported that the method counters the risk and time efficiency using estimated aleatoric uncertainty for traffic scenarios at road intersections.

In [70], a combination of interactive multiple model algorithm with stochastic MPC is proposed for safe motion planning of autonomous vehicles in the presence of dynamic objects in urban driving. The method tries to address the issues with multiple objects tracking that may exhibit dynamic behavior. The framework utilizes LQR and interactive multiple model algorithm to anticipate the trajectory of the multiple dynamic objects in an iterative manner. The framework also integrates the collision avoidance features to ensure safe driving by considering the dynamic objects.

In [71], a safe model-based RL method is proposed to address the collision free trajectory planning in uncertain autonomous vehicles. The collision avoidance is guaranteed by calculating the robust control barrier function (CBF) by considering the model uncertainty as gaussian process regression. The CBF function is used to ensure safe exploration and actions; and the gaussian process regression is used to improve the sample efficiency. The control system tries to learn the RL policies of collision avoidance, input saturation, and threshold over velocity using the proposed robust CB function.

In [72], a method for devising the uncertainty prediction horizon for stochastic non-linear model predictive controller for a highly non-linear system has been proposed using polynomial chaos expansion (PCE). The method eases the uncertainty computation for the system dynamics and aids in safe trajectory maneuver and reduced computational complexity by limiting the uncertainty propagation time using PCE.

The probabilistic methods allow the consideration of the uncertain parameters in design problem which are crucial for real-world applications. By including these probabilistic methods, the overall robustness and reliability of control systems can be improved by making more informed decisions and adapting to changing environments. These methods provide a systematic and effective way to model and quantify uncertainty to improve performance and stability.

These safe RL methods aid in overcoming the difficulties posed by safety critical applications like autonomous driving. They offer techniques to guarantee that RL agents and autonomous systems develop and carry out policies that put safety first, refrain from risky behavior, and respect safety limits. Making safe and dependable autonomous systems also requires integrating these methods with sophisticated modeling, data augmentation, and real-world validation procedures.

5. Ensuring safety in autonomous vehicles

The development and use of autonomous cars must prioritize ensuring their safety. Using a variety of methods and strategies is necessary to achieve safety. This section discusses the techniques for limited reinforcement learning, safe exploration strategies, and reward shaping and function approximation.

5.1. Reward shaping and function approximation

Reinforcement learning uses a method called reward shaping to change the reward function in a way that makes it easier for an RL agent to learn a desired behavior. To guarantee that the agent learns safe behavior, safety-critical activities must be implemented. In order to motivate the agent to put safety first, reward shaping can be used to specifically punish or reward particular behaviors or situations. Reward shaping may be used to persuade an agent to follow a certain course of action or display a certain behavior when there are several ways to accomplish a job. Exploration may be dangerous in some circumstances, particularly in fields where there is a chance of disastrous mistakes. Exploration can be guided by reward shaping in ways that minimize such risks.

Two key ideas in reinforcement learning, reward shaping and function approximation [73], are essential for training RL agents. These methods are frequently employed to improve the effectiveness of learning and direct agents toward desired behaviors.

A. *Intrinsic Motivation and Curiosity-Driven Exploration*: An agent is rewarded for both completing external goals and exploring and learning about its environment as part of the intrinsic motivation

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

technique [74]. An example of curiosity-driven exploration is when an agent is motivated by the novelty or unpredictability of experiences. Exploration can be encouraged by intrinsic motivation in a way that reduces unsafe or hazardous behavior. By encouraging curiosity, the agent can learn about the environment's dangers and boundaries without experiencing disastrous failures. This approach is used in autonomous vehicles where the agent is encouraged to explore and learn about its surroundings.

AVs that have an inbuilt motivation are able to investigate unfamiliar or unknown scenarios that may not be addressed by conventional reward-based learning. Through encouraging exploration, it can also help with situations where rewards are few. AVs may have an innate desire to investigate new road environments, handle challenging traffic situations, or observe how other drivers behave. This promotes flexibility and lifelong learning.

Curiosity-driven exploration in autonomous vehicles [74] can be used to research the impact of various control actions in a variety of driving scenarios, test the limits of the maneuver, or explore an unexplored area. More effective learning can result from curiosity-driven exploration, particularly in situations when the autonomous vehicle is unclear about the implications of its decisions. It aids in the AV's environment learning and policy discovery.

B. *Imposing Constraints on the Reward Function* - Reward shaping entails changing the reward function to increase incentives for safe behavior or reduce risky behavior [74]. The reward function might be restricted in order to expressly punish unwanted behaviors or conditions. Autonomous cars can be trained to put safety first by modifying the reward function to discourage risky behavior. For instance, the reward function may impose fines for traffic infractions such as speeding or crashes [75]. Reinforcement learning for autonomous driving frequently employs this strategy to teach cars to obey traffic regulations and steer clear of collisions.

One method used in reinforcement learning for autonomous vehicles (AVs) to direct the learning process and guarantee that AVs follow certain safety, moral, or performance standards is to impose limitations on the reward function. Constraints defined inside the reward function allow the designers to mold the AV's behavior and choices. For instance, autonomous cars must keep a safe following distance from other automobiles. In order to lessen the possibility of rear-end crashes, reward limitations may punish tailgating or following too closely. Constraint optimization and reward shaping are two optimization strategies that AVs might employ. In these situations, the goal of the reward function is to promote desired behavior, while the goal of the limitations is to stop undesired conduct. This method guarantees that the AV will behave in a way that complies with legal and human standards, making autonomous vehicles safer.

5.2. Safe exploration strategies

When training agents for real-world applications where safety is vital, reinforcement learning is a critical component. These tactics are designed to make sure that RL agents explore and learn while abstaining from acts that can have negative or hazardous consequences.

A. *Bayesian Optimization and Active Learning*: Bayesian optimization is a probabilistic method of optimization that takes uncertainty into account [76]. It can be used in RL to direct exploration by simulating environmental uncertainty. In Bayesian optimization the agent may methodically explore the environment while taking uncertainty into account. This assists the agent in avoiding dangerous situations and gathering useful information for learning. It can be used for tasks like safely optimizing control strategies or tweaking the hyperparameters of autonomous cars [77].

Calibration of cameras or LIDAR sensors may be adjusted with the help of Bayesian optimization. Actively choosing the most useful calibration settings helps AVs minimize sensor noise and improve perception accuracy. When designing the best routes and trajectories for autonomous vehicles (AVs), Bayesian optimization can help take into consideration the environment's unpredictability and changing traffic circumstances. AVs can travel more effectively by actively choosing courses that appear promising. Active learning can help AVs perceive and detect objects better. AVs can improve their situational awareness by deliberately selecting where to point their sensors (e.g., concentrating on regions with high uncertainty or possible obstructions). AVs may decide in real time where to concentrate their processing resources due to active learning, which guarantees that the most significant data is handled as soon as possible which is particularly important for AVs operating in dynamic environments.

B. *Safe Reinforcement Learning with Human Guidance*: Human guidance involves incorporating input and human expertise into the RL algorithms [78]. Humans can be brought into the loop during training to correct risky behavior. Autonomous cars can receive fast feedback from human managers when human guidance is present, reducing the likelihood of detrimental acts. Accidents during training can be avoided by human intervention. The training of autonomous systems, like self-driving automobiles, where human safety drivers can take over as needed.

In order to guarantee that AVs learn to make safe and dependable judgments, it integrates the advantages of machine learning with human knowledge and monitoring. Imitation learning [60], in which the AV picks up skills by imitating the actions of a human driver, can be facilitated by human instruction. Human demonstrations provide the AV's policy with expert guidance during training. A human driver's conduct may be directly replicated by AVs using human direction and recorded data. With behavior cloning, an AV mimics the movements of a human demonstration through supervised learning.

A reconciliation is attained between the capabilities of autonomous systems and the safety knowledge of human operators in safe reinforcement learning with human guidance [78]. It guarantees that AVs learn to make appropriate judgments in real-world situations and that they are trained and deployed with a strong emphasis on safety.

5.3. Constrained reinforcement learning

Constrained Reinforcement Learning (CRL) [79] is a subfield of reinforcement learning that focuses on training agents to operate within predefined safety constraints. In conventional RL, agents optimize a reward function without explicitly taking safety into account. By introducing restrictions that must not be violated during learning, CRL, in contrast, makes sure that the agent's activities stay within reasonable bounds. This is particularly crucial in fields like autonomous driving, robotics, and healthcare where safety is of the utmost importance.

A. *Constrained Policy Optimization*: By explicitly considering safety constraints during optimization, autonomous vehicles can make sure that their actions remain within acceptable bounds, lowering the risk of accidents. Constrained policy optimization [79,80] is frequently used in autonomous systems to enforce safety limits on actions. It involves optimizing the policy while adhering to safety constraints, which can be specified in terms of acceptable risk or performance bounds.

Constrained Policy Optimization ensures that AVs respect safety and performance restrictions while learning and making decisions. Additionally, AVs must fulfill a number of performance requirements, such as adhering to speed restrictions and hitting fuel economy goals. Constrained policy optimization ensures that AVs follow these guidelines while deciding how to drive. Limitations can be dynamically changed to reflect how the AV interprets its

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

surroundings. For instance, safety restrictions could be loosened in lower-risk circumstances but tightened in complicated traffic scenarios. Constrained policy optimization offers a sensible method to strike a compromise between safety and performance requirements when it is difficult to create reward functions that do so. This allows the AV to explore efficiently.

B. *Constrained Markov Decision Processes*: Constrained Markov Decision Processes (CMDPs) [81] are an extension of standard MDPs that incorporate behavioral constraints. The objective is to identify policies that satisfy these constraints and optimize performance. Performance and safety trade-offs are specifically addressed in CMDPs. Agents are taught rules that strike a balance between attaining goals and averting risky behaviors or conditions. CMDPs are suitable in a variety of autonomous vehicle scenarios where safety is a major concern.

Reinforcement learning for autonomous vehicles uses CMDPs to simulate and handle situations where constraint satisfaction and performance optimization are crucial. AVs frequently have to operate under performance parameters, such as meeting fuel economy goals or adhering to speed restrictions. A means of incorporating these performance standards into the training process is through CMDPs. CMDPs offer a solution for addressing this trade-off and facilitating effective exploration when it is difficult to create a reward function that strikes a balance between performance and safety. CMDPs can direct autonomous vehicles to prioritize safe moves over non-essential operations, such as emergency braking or evasive maneuvers when safety is at stake. CMDPs augment the cost function used in policy optimization with terms that penalize constraint violations. This encourages the AV to choose actions that reduce the likelihood of violating the limitations.

6. Benchmarking and evaluation of safe rl for autonomous vehicles

Benchmarking and assessing the effectiveness of safe reinforcement learning algorithms for autonomous cars requires the use of simulators and test environments. These settings act as controlled environments for testing different facets of autonomous driving. CARLA, SUMO, LGSVL, and Apollo's simulation platform are a few popular test environments and simulators for autonomous vehicle development. Simulators should accurately resemble actual driving circumstances, including the state of the roads, the flow of traffic, and sensor models. Realism ensures that simulation-trained RL agents can use their knowledge in the actual world. It's important to have a wide range of circumstances at the disposal when assessing an agent's generalization and adaptability to cover urban, suburban, highway, and extreme weather. Safety features and fail-safe systems should be included in simulators to prevent potentially harmful collisions and acts during testing. Simulators should enable researchers to create unique situations, introduce flaws, and adjust numerous parameters to evaluate the performance of the agent in various settings.

A variety of criteria must be used to evaluate the performance and safety of autonomous RL-based cars in order to offer a thorough evaluation. This statistic evaluates an RL agent's tendency to steer clear of objects, people, or other cars. A high percentage of accident avoidance suggests safer driving practices. Adversarial conditions present the agent with difficult circumstances that are intended to test their resiliency and measure how well the RL agent manages hostile situations, including unforeseen barriers or hostile drivers. The agent's capacity to adjust to novel and unexplored events that were not included in its training data is measured by generalization measures. Table 1 compares the various advantages and disadvantages of the safe RL methods employed in autonomous vehicles. To assess the RL agent's generalization skills, various road configurations [82-84], traffic patterns [85-92], and weather variables [93] are used in testing such as:

i. Counting traffic infractions such as speeding and running red lights.
ii. Examining the smoothness of a vehicle's control operations to determine how it affects passenger comfort.
iii. Distance to Safety Limits: Measuring how close the agent is during testing to established safety limits or restrictions.
iv. Evaluation of the efficiency of emergency braking under urgent circumstances.

7. Case studies and research applications

Reinforcement learning based end-to-end autonomous driving [94, 95] has drawn a lot of attention. This method uses RL to directly learn the complete driving pipeline, from perception to control, from unprocessed sensor input, such as camera images and by learning from human examples, where RL is used to train agents to mimic human driving behavior [96,97]. These agents can automatically navigate and generalize from driving data involving humans. Before deploying RL agents in actual vehicles, they have been trained in safe, controlled conditions using simulations, such as the CARLA simulator [25]. Researchers have created RL algorithms that allow cars navigate difficult situations, obey traffic laws, and communicate with other cars. End-to-end RL for autonomous driving has difficulties generalizing to different road conditions and being safe.

Optimizing traffic flow, lowering congestion, and increasing overall transportation effectiveness are all part of urban traffic management. To overcome these difficulties, Multi-Agent Reinforcement Learning (MARL) [98,99] is being used. MARL algorithms have been applied to improve intersection traffic signal timings. MARL can ease congestion and enhance traffic flow by enabling communication and coordination between traffic signals [100-102]. On highways and city streets, agents in a multi-agent RL system can manage lane assignment and traffic merging, to ease congestion and boost traffic efficiency. In order to deliver dependable and effective services, public transportation systems like buses and trams are optimized using MARL. This includes modifying the schedule in real-time to reflect changes in passenger demand. Scalability, coordination, and addressing dynamic traffic circumstances are difficult issues with urban traffic management using MARL. Fig. 9 shows the higher-level abstraction of the end-to-end autonomous driving framework

The fundamental perception layer that AVs need to understand their surroundings is provided by computer vision and based key choices on how the environment is seen, RL expands on this perspective. Together, these technologies enable the development of a full autonomous driving system. Compressing the input image into a representative feature vector is one of the trickiest problems. It is essential to ensure the reliability and security of RL-based control systems, including autonomous cars [103]. When a malicious actor tries to influence an agent's decision-making process by tampering with the input data or taking advantage of flaws in the RL algorithm, this is referred to as an adversarial attack. RL agents that can withstand hostile attacks, such as hacking and sensor manipulation. To increase the security of RL-based control systems, methods like adversarial training, robust training, and input preprocessing must be investigated [104].

RL agents should be trained in hostile perturbation settings. AVs may learn to withstand attacks and make wiser choices when facing adversaries with the aid of adversarial training and create reliable state estimate methods that can handle tampered or noisy sensor data. Accurate perception is essential while facing hostile attacks. AV communication systems should have their cybersecurity strengthened to fend off unwanted access and interference. Encryption and secure communication methods can safeguard data transfer and control and to aid the RL agent in identifying and responding to adversarial assaults, it is required to supplement training data with hostile instances and provide methods for monitoring and alerting in real-time that are capable of identifying and countering hostile attempts on the AV's control system. Fig. 10 shows

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

Table 1
A comprehensive overview of safe reinforcement learning methods in autonomous driving.

S. No.	Category	Method	Environment	Task	Learning Approach	Key Safety Features	Performance Metrics	Ref.
1	**DQN**	DQN for Atari & Alpha Go	Simulated (Atari 2600, Alpha Go)	General decision-making	Off-policy	Unexpected situation handling	Improved decision-making and learning	[24]
2		Lexicographic Q-Learning DQN	Simulated (highways, crossroads)	Multi-objective driving (lane change, yielding)	On-policy	Factored Q functions, adherence to traffic laws	Navigation in complex environments, safety compliance	[25]
3		DQN & CARLA Simulator	Simulated (CARLA)	Obstacle avoidance	Off-policy	Cost-efficient training in virtual environment	Improved behavior with training	[26]
4		DQN with Request-Respond Mechanism	Simulated	Lane changing (mandatory & optional)	On-policy	Separate request & respond groups, central agent	Efficient lane maneuvers, reduced computation	[27]
5		Analytical Reward Function for DQN	–	Velocity control	–	Optimized reward functions	Expected optimal velocity	[28]
6	**DDQN & Dueling DQN**	Double DQN with Integrated Perception	Simulated (naturalistic driving data)	Speed control	Off-policy	Improved value accuracy and policy quality	271.73 % higher score than DQN	[34]
7		Double DQN with A* optimization	Simulated (random obstacles)	Route planning	Off-policy	Collision avoidance, improved routing	Higher distance traveled, fewer collisions than Q-learning	[35]
8		Q-learning & DQN for path finding	Physical (Raspberry Pi robot)	Collision avoidance, pathfinding	Off-policy	Efficient memory & processing, shortest path navigation	Reduced memory usage, shorter travel time compared to Q-learning	[36]
9		Comparative DQN study for highway navigation	Simulated (highways)	Lane maneuvering	On-policy	Duelling & policy-regulated DQNs offer better performance	Convergence rate, maneuvering performance	[37]
10		Improved Dueling DQN with EGARL for intersections	Simulated (intersections)	Decision-making at intersections	On-policy	Graph attention for interaction modeling, RL for decision-making	–	[38]
11		Dueling DDQN for AGV routing	Simulated (digital twin)	Optimal routing for AGV	Off-policy	Collision avoidance, short delays, stability	Reduced delays, improved stability, power efficiency	[39]
13	**PPO**	DDPG for lateral & longitudinal control	Simulated & real-world	Combined lateral & longitudinal control	On-policy	Improved stability, faster learning	–	[44]
14		Twin-Q SAC for tractor-trailer navigation	Simulated (roundabouts)	Roundabout navigation	On-policy	Collision avoidance with tight turns	73 % success rate in avoiding curb collisions	[45]
15		PPO for path following	Simulated	Optimal path following	On-policy	Optimal transient response	–	[46]
16		Weight-varying MPC with deep RL	Simulated	Safe and optimal behavior	Off-policy	Pre-optimized weights for safe exploration	–	[47]
17	**TRPO**	TRPO	Deterministic	Real-world, simulated	Off-policy	Trust region constraint	Policy stability, transfer learning	[48]
18		Deep SAC with TRPO	Stochastic	Simulated	On-policy	Lane curvature estimation, cost minimization with TRPO	Tracking performance, collision avoidance	[49]
19		Twin DDPG with safety factor	Stochastic	–	On-policy	Exploration-exploitation trade-off with safety factor	Safety guarantee level, reward maximization	[50]
20		Tiered DRL with rule-based methods	Hybrid (DRL & rule-based)	Simulated	Hybrid (on- & off-policy)	Rule-based safety for low-level control, DRL for high-speed maneuvers	Maneuver success rate, safety violations	[51]
21		Graph Laplacian for feature matrix	Deterministic	Simulated	Offline (pre-learning)	Unexpected turn handling	Decision-making accuracy, collision avoidance	[52]
22		RL with low-dimensional occupancy grid maps	Stochastic	Simulated (off-road)	On-policy	Adaptive curriculum learning for efficient feature learning	Path planning success rate, obstacle avoidance	[53]
23	**MPC**	eMPC with DDQN, SAC, PPO [-]	Simulated	Path following	Off-policy	Prioritized experience replay (PER), LSTM networks	PPO-eMPC with LSTM & DDQN-eMPC with PER & LSTM offered best performance	[59]
24		Differentiable MPC with imitation learning [-]	Simulated	Lane keeping	On-policy	Safety, comfort, human driver imitation	Robust closed-loop and open-loop behavior	[60]
25		RL-based Nonlinear MPC [-]	Simulated & real-world	Lane keeping	Model-free (tabular Q-learning)	Robustness to sensor noise and external interferences	Good performance in both simulated and real-world scenarios	[61]
26		Two-stage framework with SAC & MPC [-]	Simulated	Left-turn at unsignalized intersections	On-policy & MPC	Collision avoidance, non-conservative safe maneuvers, comfort	Outperforms model-free RL methods	[62]

(*continued on next page*)

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

Table 1 (*continued*)

S. No.	Category	Method	Environment	Task	Learning Approach	Key Safety Features	Performance Metrics	Ref.
27		Interactive RL with MPC and human-in-the-loop [-]	Real-world & simulated	Navigation in complex & congested environments	On-policy with human feedback	Safe driving envelope, human intervention	Efficient and safe navigation in various scenarios	[63]
28	**Probabilistic Models for Uncertainty Estimation**	DRL with multi-objective Bayesian optimization for weight-varying MPC [-]	Simulated	Optimal controller tuning	Off-policy	Pre-optimized controller weights for safe exploration	–	[67]
29		Uncertainty-aware model-based RL planning with MPC [-]	Simulated	Prediction, planning, and control	On-policy	Pessimistic trajectory optimization, MPC	Reduced risk with efficient decision-making	[68]
30		Ensemble quantile network for uncertainty estimation with distributed RL [-]	Simulated	Uncertainty estimation and action learning	Off-policy	Ensemble methods for epistemic uncertainty, quantile network for aleatoric uncertainty	Risk mitigation and time efficiency	[69]
31		Interactive multiple model with stochastic MPC for dynamic objects [-]	Simulated	Safe motion planning with moving objects	On-policy	Interactive tracking, collision avoidance features	–	[70]
32		Safe model-based RL with robust control barrier function (CBF)	Simulated	Collision-free trajectory planning	On-policy	CBF for guaranteed safety, Gaussian process for sample efficiency	–	[71]
33		PCE-based uncertainty prediction for stochastic non-linear MPC	–	Trajectory planning for highly non-linear systems	–	Reduced uncertainty computation, safe maneuvers	Reduced computational complexity	[72]

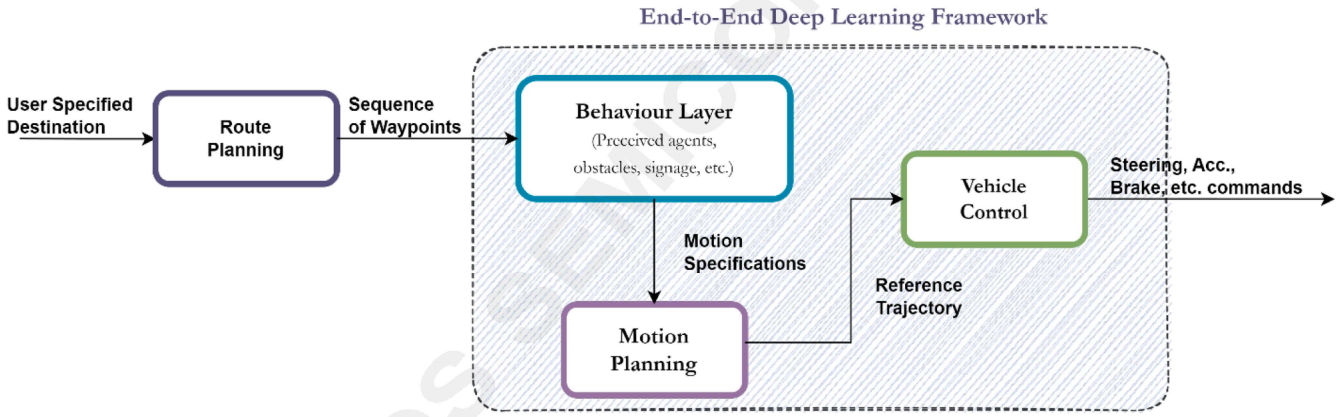


Fig. 9. Schematic representation of higher level end-to-end autonomous driving framework [94,95].

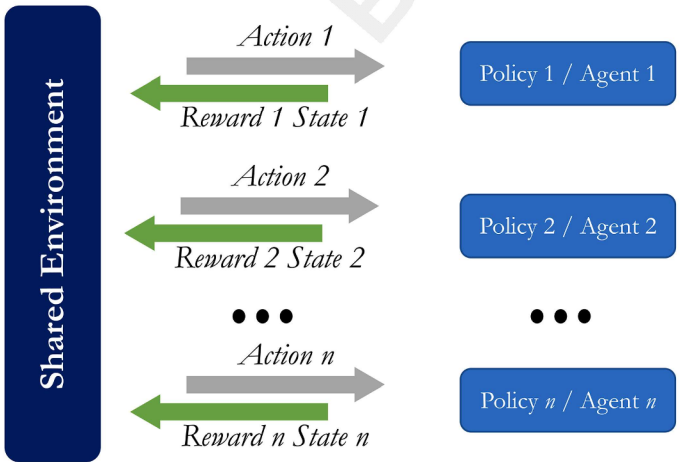


Fig. 10. Representation of the multi-Agent RL operating in a shared environment with various agents with different policies [98].

the block diagram of the multi-Agent RL operation in a shared environment with various agents with different policies.

8. Ethical and legal considerations

When creating and implementing RL-based autonomous systems, especially in safety-critical sectors like autonomous vehicles, ethical and legal considerations are of utmost importance [105,106]. Some key aspects to consider are:

A. *Accountability and Liability in RL-based Autonomous Systems*: It is difficult to clearly define responsibility and liability for mishaps or unfavorable outcomes involving RL-based autonomous systems [107]. Who should be held accountable in the event that an autonomous car makes a decision that results in an accident—the manufacturer, the software developer, or the vehicle owner? [108] It is crucial to create a legal framework that tackles these problems. Governments and regulatory organizations must establish regulations and standards for determining who is at fault and who bears liability in incidents involving autonomous cars. These frameworks

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

must take into consideration the intricacies of real-world learning, where behaviors develop through interaction and learning. The insurance sector needs to adjust to the particular difficulties brought on by RL based autonomous vehicles.

B. *Transparency and Explainability of RL Decisions*: The ability to understand RL models is essential for comprehending the decision-making procedure [109]. Transparent models can aid users and regulators in comprehending the rationale behind a certain course of action. One can learn more about how the RL model made its decisions by using explain ability approaches like saliency maps, attention processes, or feature importance analyses. Transparency and performance might be difficult to balance, while some sophisticated RL models may perform well, they give little transparency. It's difficult to balance performance and explainability in the appropriate way all the time.

C. *Regulatory Framework and Standards for Safe Autonomous Vehicles:* Autonomous vehicles must abide by all current safety rules and traffic laws [110]. It is crucial to define safety requirements particular to RL-based systems. Risk assessment, validation, testing procedures, and safety-critical behavior should all be covered by these standards. Large volumes of data, including sensor data and position data, are gathered by autonomous cars. In order to safeguard users' privacy and sensitive data, data privacy and security regulations must be implemented. To make sure AVs fulfill safety and performance criteria, autonomous vehicles should go through stringent testing and certification procedures. Testing under many scenarios, environments, and weather conditions is part of this. Given that autonomous vehicles can operate internationally, worldwide regulatory norm synchronization is necessary to establish a common framework for producers and consumers.

To ensure the responsible development and use of autonomous vehicles, governments, researchers, and manufacturers must work together to address ongoing challenges such as balancing innovation and safety, defining accountability and liability frameworks, ensuring transparency in decision-making, and establishing robust regulatory standards.

9. Future directions and open challenges

A. *Scalability to Real-world Complexity*: Scaling RL-based systems to handle the entire complexity of real-world situations is one of the biggest difficulties faced by self-driving technologies, as it approaches mainstream use [111,112]. Dealing with various and changing traffic situations, complex urban infrastructure, and interactions with other road users are all included in this. Future research should concentrate on creating RL techniques that can successfully scale up and function dependably under these challenging circumstances.

B. *Incorporating Uncertainty and Risk in RL Agents*: Self-driving cars must maneuver through circumstances where the results are not always predictable since real-world surroundings and roads are by their very nature are unpredictable [113]. Future research should examine cutting-edge methods for incorporating and controlling risk and uncertainty in RL agents. To provide the highest levels of safety, this involves probabilistic modeling, constrained optimization, and techniques for handling rare and catastrophic situations.

C. *Hybrid Approaches: Combining RL with Rule-based Systems*: A promising direction for self-driving technology is hybrid methods [114-116] that integrate RL with rule-based systems. Rule-based systems (RBS) can offer adaptability and learning from data, whereas RBS can offer explicit safety assurances and domain-specific knowledge. In order to fully utilize the benefits of both paradigms, researchers should look into how both methodologies can be combined seamlessly. A critical component of this research is the creation of strategies for smoothly switching between learnt behaviors and rule-based ones.

These issues must be resolved in order to develop safe, scalable, and adaptable autonomous cars. Further advancements in AI and reinforcement learning research are also necessary. Thus, the autonomous vehicles are one step closer to realizing their potential to transform transportation, boost traffic safety, and enhance everyone's mobility. For an ideal scenario with zero accidents and every car running on AI on the streets will follow all the rules.

10. Conclusion

This paper deals with the exploration of the intersection of autonomous vehicles and reinforcement learning, with a specific focus on the importance of safe RL for the future of autonomous vehicles. To increase the safety of RL-based autonomous vehicles, a variety of techniques have been discussed, such as value function approximation, policy optimization techniques, and model-based RL. These methods seek to maintain a balance between exploration and exploitation while observing safety restrictions. SRL methods guarantee that these agents are trained to obligate safety and avoid risky behaviors. Also, it is crucial to consider both the ethical and legal implications of RL-based autonomous systems. To ensure the proper deployment of autonomous cars, concerns of accountability, transparency, and regulatory frameworks must be addressed. Scalability to real-world complexity, including uncertainty and risk, hybrid techniques, interpretable AI, and tackling ethical and legal issues are just a few of the future paths and open difficulties that have been mentioned.

CRediT authorship contribution statement

Rohan Inamdar: Writing – original draft, Visualization, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **S. Kavin Sundarr:** Writing – original draft, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Deepen Khandelwal:** Investigation, Formal analysis, Data curation. **Varun Dev Sahu:** Investigation, Formal analysis, Data curation. **Nitish Katal:** Writing – review & editing, Visualization, Supervision, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

Authors declare that they have no conflict of interests.

Appendix

A1. Kinematic Model of the Vehicle under Control

The development of various control schemes involves considering the kinematic models of the vehicles. Bicycle model is a classical model used for simplified representation of the vehicles motion. In the model it is assumed that the front two wheels of the vehicle are lumped together and are placed at the center of the front axle as shown in Fig. A1.1. The vehicle states are defined by 3 variables of position, orientation and velocity. The position (x, y) defines the coordinates of the center of gravity of the vehicle, the orientation ψ defines the angle between the vehicle's heading direction and the x-axis, and the velocity v represents the velocity of the vehicle. The model has two control inputs, the steering angle δ_f, which defines the angle of the

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

front wheels with respect to the direction of the vehicle and the acceleration a which defines the rate of change of the velocity of the vehicle. Thus, the equations for the kinematic bicycle model of the vehicle can be obtained as [117-119].

$$\dot{x} = v\ cos(\psi + \beta) \tag{a.1}$$

$$\dot{y} = v\ sin(\psi + \beta) \tag{a.2}$$

$$\dot{\psi} = \frac{v}{l_r}\ sin(\beta) \tag{a.3}$$

$$\dot{v} = a \tag{a.4}$$

$$\beta = \tan^{-1}\left(\frac{l_r}{l_f + l_r}\tan(\delta_f)\right) \tag{a.5}$$

Where, β is the slip angle between the direction of the current velocity and the longitudinal axis of car. l_f and l_r represent the wheel base of the car measured between the front and rear axles.

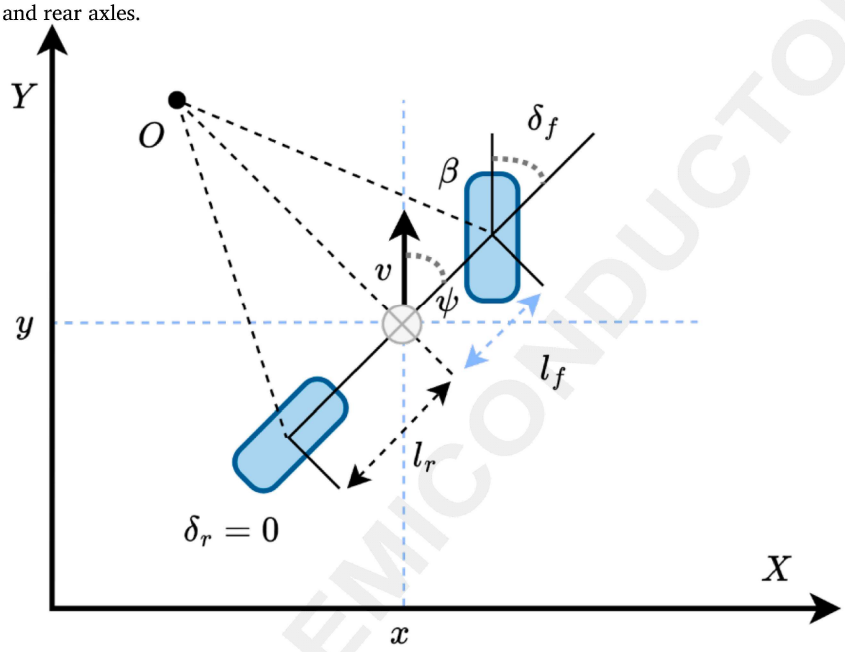


Fig. A1.1. Kinematic bicycle model of a car [117].

A2. Forces Acting on Vehicle Model under Control

A moving vehicle is acted upon by the various longitudinal, lateral and vertical forces [119], as shown in Fig. A2.1.

Fig. A2.1. Forces acting on a vehicle under control.

i. Longitudinal Forces

a. *Thrust Force*: This is the force generated by the engine or the electric motors to propel the vehicle forward.

b. *Aerodynamic Drag*: It is the resistive force acting on the vehicle and is opposite to the direction of the motion of the vehicle. It is dependent on the factors like frontal area $\mathscr{A}$, density of the air ρ, drag coefficient $\mathscr{C}_g$, and velocity v; given mathematically as:

$$F_D = \frac{1}{2}\rho\,\mathscr{C}_g\mathscr{A}v^2 \tag{a.6}$$

c. *Rolling Resistance*: This is the resistance caused by the deformation of the tires and the surface of the road and opposes the motion of the vehicle, it is given by equation (a.7), where $\mathscr{C}_R$ is the coefficient of rolling resistance, m is the mass of the vehicle and g is the acceleration due to gravity.

$$F_R = \mathscr{C}_R mg \tag{a.7}$$

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

ii. Lateral Forces

a. *Tire Forces*: These forces are generated during steering action and are due to the interaction of the tires and road surface. They are given by F_{YF} and F_{YR} for the front and real axle, and can be approximated as in equation (a.8), where $\mathscr{C}_{SF},\ \mathscr{C}_{SR}$ represents the cornering stiffness and $\alpha_F,\ \alpha_R$ is corresponding slip angle.

$$F_{YF} = \mathscr{C}_{SF}\alpha_F$$

$$F_{YR} = \mathscr{C}_{SR}\alpha_R \tag{a.8}$$

b. *Centrifugal Forces*: This the outward force acting on the vehicle at turns, and is proportional to the speed and curvature of the turn, given as in equation (a.9), where $\mathscr{R}$ is the radius of the turn.

$$F_C = \frac{mv^2}{\mathscr{R}} \tag{a.9}$$

iii. Vertical Forces

a. *Normal Forces*: This is the force exerted by the tires of the vehicle opposite to the gravitational force.

b. *Gravitational Force*: It is downward force acting on the vehicle and is given as $F_g = mg$.

A3. Optimal Trajectory Planning

In path planning, the aim is to optimize the trajectory that with minimum deviation and can be achieved by precisely formulating a cost function, a set of constraints and a choice of optimizer.

i. Cost function and Constraints

The formulation of the cost function aims at minimizing the error between the desired and the actual path and also incorporates the penalization of certain actions like high control inputs to ensure safety. The formulated cost function includes the position and orientation errors, control effort and other objectives like the speed deviation or the obstacle avoidance can also be included as per the requirements of the developer. A generalized cost function, J over the time T can be expressed mathematically as:

$$J = \int_0^T \left[\left\{(x_d(t) - x(t))^2\right\} + \left\{(y_d(t) - y(t))^2\right\} + q_\psi\left\{(\psi_d(t) - \psi(t))^2\right\} + \alpha_1\delta^2(t) + \alpha_2 a^2(t)\right]dt \tag{a.10}$$

where, $x(t),\ y(t)$ are the actual and $x_d(t),\ y_d(t)$ are the desired coordinates for the position at time t, $\psi(t),\ \psi_d(t)$ are the actual and desired orientation of the vehicle, $\delta(t)$ is the steering ancle, $a(t)$ is the acceleration and $q_\psi,\ \alpha_1,\ \alpha_2$ are the weighting factors.

In the cost function the first two terms $\left\{(x_d(t) - x(t))^2\right\} + \left\{\left(y_d(t) - y(t)\right)^2\right\}$ are used to ensure the accurate tracking of the path, the 3rd term $\left\{(\psi_d(t) - \psi(t))^2\right\}$ aims at minimizing the deviations in the orientation of the vehicle, such that the vehicle maintains a correct heading, the last terms $\alpha_1\delta^2(t) + \alpha_2 a^2(t)$ penalizes the high control inputs such that steering and the acceleration is smooth to ensure passenger comfort and vehicle stability.

Additionally, the constraints on the steering angle and the acceleration must be ensured for safe operation of the vehicle, and are given in equation (a.10). The steering constraint ensures that the steering angle mist not go beyond the maximum allowable limits and the acceleration constrains ensures that there are no aggressive maneuvers.

$$|\delta(t)| \le \delta_{MAX}\ \&\ |a(t)| \le a_{MAX} \tag{a.11}$$

ii. Optimizer

The above formulated problem can be solved by using two prominent methods like model predictive control and reinforcement learning.

a. Model Predictive Control (MPC)

The model predictive control tries to optimize the control inputs by using the vehicle model to predict the future states over a finite horizon, and can be formulated as in equation a.11.

$$\min_u \sum_{k=0}^{N-1} \left[(x_k - x_d)^T Q(x_k - x_d) + (\psi_k - \psi_d)^2 + u_k^T R u_k\right]$$

subjectto :

$$x_{k+1} = f(x_k,\ u_k)$$

$$|\delta(t)| \le \delta_{MAX},\ |a(t)| \le a_{MAX} \tag{a.12}$$

Where, $x_k,\ u_k$ are the states and control input at time step k, Q and R are the weighting metrices and N is the prediction horizon.

b. Reinforcement Learning

In the reinforcement learning the agent tries to learn the optimal control policies π via interaction with the environment. Algorithms such as DDPG, etc. can ensure that the safety contains are satisfied. The problem can be expressed as:

$$\min_\pi \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

subjectto :

$$s_{k+1} = f(s_k,\ a_k)$$

$$|\delta(t)| \le \delta_{MAX},\ |a(t)| \le a_{MAX} \tag{a.13}$$

Where, the $s_k,\ a_k$ are the states and the actions at time t, r is the reward function and γ is the penalty function.

A4. Some Practical Considerations

i. Sensor Noise

All the measurements are subject to sensor noise, this can significantly impact the accuracy of the estimation of the correct vehicle states. The methods like Kalman filtering or particle filtering can be used to minimize the measurement noise and improve the accuracy of state estimations. For example, mathematically the prediction and updating of the states using Kalman Filtering can be given as in Fig. A4.1. In the figure

ii. Dynamic Objects in Driving Environment

As the driving environment is dynamic, thus the path planning algorithms must take the dynamic objects into consideration by incorporating the obstacle detection and prediction modules in the path planning and control framework. A simple dynamic obstacle can be represented as $x_o^{\cdot} = v_o\cos(\psi_o),\ \dot{y_o} = v_o\sin(\psi_o)$, where $x_o,\ y_o$ are the position coordinates, v_o is the velocity and ψ_o is the heading direction.

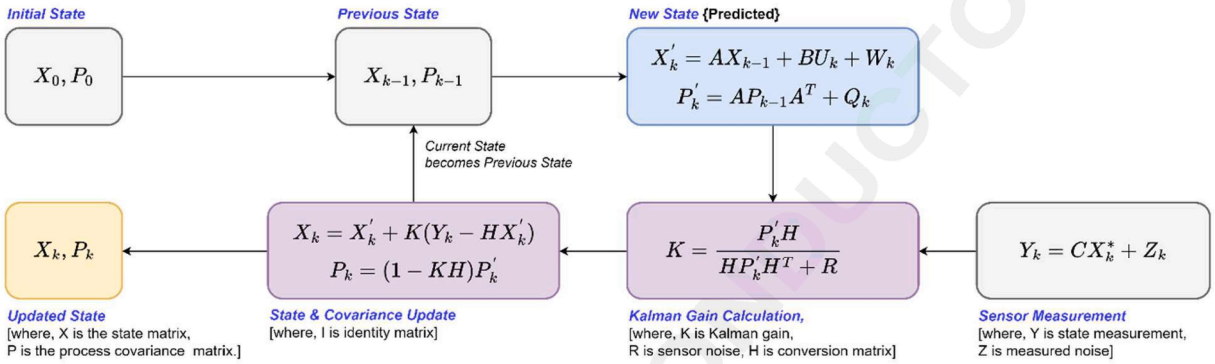


Fig. A4.1. Overview of Kalman filter [120].

Data availability

No data was used for the research described in the article.

References

[1] A. Dornadula, Autonomous driving, both close and far from ubiquity, Skynet Today (2018). Jun. 30, https://www.skynettoday.com/editorials/autonomous_vehicles.

[2] Frank Hoffmann, Gerd Pfister, Evolutionary learning of a fuzzy control rule base for an autonomous vehicle, in: Proceedings of the Fifth International Conference IPMU: Information Processing and Management of Uncertainty in Knowledge Based Systems, Granada, Spain, 1996.

[3] Mike Daily, J. Harris, D. Keirsey, D. Olin, D. Payton, Kurt Reiser, J. Rosenblatt, D. Tseng, V. Wong, Autonomous cross-country navigation with the ALV, in: Proceedings of the 1988 IEEE International Conference on Robotics and Automation, IEEE, 1988, pp. 718–726.

[4] Stefan Annell, Alexander Gratner, Lars Svensson, Probabilistic collision estimation system for autonomous vehicles, in: Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2016.

[5] Daniel Pagac, Eduardo Mario Nebot, Hugh Durrant-Whyte, An evidential approach to map-building for autonomous vehicles, IEEE Trans. Robot. Autom. 14 (4) (1998) 623–629.

[6] Jesse Levinson, Jake Askeland, Jennifer Dolson, Sebastian Thrun, Traffic light mapping, localization, and state detection for autonomous vehicles, in: Proceedings of the 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 5784–5791.

[7] Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. "Playing atari with deep reinforcement learning." arXiv preprint (2013).

[8] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.

[9] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare, Joelle Pineau, An introduction to deep reinforcement learning, Foundations Trends® Machine Learn. 11 (3–4) (2018) 219–354.

[10] Liviu Marina, Andreea Sandu, Deep reinforcement learning for autonomous vehicles-state of the art, Bull. Transilvania Univ. Brasov. Series I (2017) 195–202.

[11] B.Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Yogamani, Patrick Pérez, Deep reinforcement learning for autonomous driving: a survey, IEEE Trans. Intell. Transport. Syst. 23 (6) (2021) 4909–4926.

[12] Sallab, Ahmad EL, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. "Deep reinforcement learning framework for autonomous driving." arXiv preprint (2017).

[13] Wang, Sen, Daoyuan Jia, and Xinshuo Weng. "Deep reinforcement learning for autonomous driving." arXiv preprint (2018).

[14] Szilárd. Aradi, Survey of deep reinforcement learning for motion planning of autonomous vehicles, IEEE Trans. Intell. Transport. Syst. 23 (2) (2020) 740–759.

[15] Jianyu Chen, Bodi Yuan, Masayoshi Tomizuka, Model-free deep reinforcement learning for urban autonomous driving, in: Proceedings of the 2019 IEEE intelligent transportation systems conference (ITSC), IEEE, 2019.

[16] Wei Xia, Huiyun Li, Baopu Li, A control strategy of autonomous vehicles based on deep reinforcement learning, in: Proceedings of the 2016 9th International Symposium on Computational Intelligence and Design (ISCID) 2, IEEE, 2016.

[17] Abida Khanum, Chao-Yang Lee, Chu-Sing Yang, Involvement of deep learning for vision sensor-based autonomous driving control: a review, IEEE Sens J (2023).

[18] Qiying Hu, Wuyi Yue, Markov Decision Processes With Their Applications, 14, Springer Science & Business Media, 2007.

[19] R.E. Bellman, S.E. Dreyfus, Applied Dynamic Programming, 2050, Princeton university press, 2015.

[20] Coggan, Melanie. "Exploration and exploitation in reinforcement learning." Research supervised by Prof. Doina Precup, CRA-W DMP Project at McGill University (2004).

[21] Michael L. Littman, Value-function reinforcement learning in Markov games, Cogn. Syst. Res. 2 (1) (2001) 55–66.

[22] Justin Boyan, Andrew Moore, Generalization in reinforcement learning: safely approximating the value function, Adv. Neural Inf. Process Syst. 7 (1994).

[23] "Introduction to RL and Deep Q Networks," TensorFlow. [Link].

[24] Li, Changjian, and Krzysztof Czarnecki. "Urban driving with multi-objective deep reinforcement learning." arXiv preprint (2018).

[25] Wasinee Terapaptommakol, Danai Phaoharuhansa, Pramote Koowattanasuchat, Jartuwat Rajruangrabin, Design of obstacle avoidance for autonomous vehicle using Deep Q-network and CARLA simulator, World Electric Vehicle J. 13 (12) (2022) 239.

[26] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, Vladlen Koltun, CARLA: an open urban driving simulator, in: Proceedings of the Conference on robot learning, PMLR, 2017, pp. 1–16.

[27] Jian Guo, Istvan Harmati, Lane-changing system based on deep Q-learning with a request–respond mechanism, Expert Syst. Appl. 235 (2024) 121242.

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

[28] Yixu He, Yang Liu, Lan Yang, Xiaobo Qu, Exploring the design of reward functions in deep reinforcement learning-based vehicle velocity control algorithms, Transport. Lett. (2024) 1–15.

[29] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, et al., Deep q-learning from demonstrations, in: Proceedings of the AAAI conference on artificial intelligence 32, 2018.

[30] Lincan Li, Chiew Foong Kwong, Qianyu Liu, Jing Wang, A smart cache content update policy based on deep reinforcement learning, Wireless Commun. Mobile Comput. 2020 (2020) 1–11.

[31] Hado. Hasselt, Double Q-learning, Adv. Neural Inf. Process Syst. 23 (2010).

[32] Hado Van Hasselt, Arthur Guez, David Silver, Deep reinforcement learning with double q-learning, in: Proceedings of the AAAI conference on artificial intelligence 30, 2016.

[33] Nguyen Quang Hieu, Dinh Thai Hoang, Dusit Niyato, Ping Wang, Dong In Kim, Chau Yuen, Transferable deep reinforcement learning framework for autonomous vehicles with joint radar-data communications, IEEE Trans. Commun. 70 (8) (2022) 5164–5180.

[34] Yi Zhang, Ping Sun, Yuhan Yin, Lin Lin, Xuesong Wang, Human-like autonomous vehicle speed control by deep reinforcement learning with double Q-learning, in: Proceedings of the 2018 IEEE intelligent vehicles symposium (IV), IEEE, 2018, pp. 1251–1256.

[35] Faezeh Jamshidi, Lei Zhang, Fahimeh Nezhadalinaei, Autonomous Driving Systems: developing an Approach based on A* and Double Q-Learning, in: Proceedings of the 2021 7th International Conference on Web Research (ICWR), IEEE, 2021.

[36] Serena Raju, Riya Mol Raji, Sherin Shibu, Joel Thomas, Pragmatic implementation of reinforcement algorithms for path finding on raspberry Pi, in: Proceedings of the 2022 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), IEEE, 2022, pp. 1–6.

[37] Teng Liu, Yuyou Yang, Wenxuan Xiao, Xiaolin Tang, Mingzhu Yin, A comparative analysis of deep reinforcement learning-enabled freeway decision-making for automated vehicles, IEEe Access. (2024).

[38] Yuchuan Qiang, Xiaolan Wang, Xintian Liu, Yansong Wang, Weiwei Zhang, Edge-enhanced graph attention network for driving decision-making of autonomous vehicles via deep reinforcement learning, Proc. Inst. Mech. Eng., Part D (2024), 09544070231217762.

[39] Lixiang Zhang, Chen Yang, Yan Yan, Ze Cai, Yaoguang Hu, Automated guided vehicle dispatching and routing integration via digital twin with deep reinforcement learning, J. Manuf. Syst. 72 (2024) 492–503.

[40] Joshua Achiam, David Held, Aviv Tamar, Pieter Abbeel, Constrained policy optimization, in: Proceedings of the International conference on machine learning, PMLR, 2017, pp. 22–31.

[41] Yuchuan Fu, Changle Li, Fei Richard Yu, Tom H. Luan, Yao Zhang, An autonomous lane-changing system with knowledge accumulation and transfer assisted by vehicular blockchain, IEEe IoT J. 7 (11) (2020) 11123–11136.

[42] Zhu Gongsheng, Pei Chunmei, Ding Jiang, Shi Junfeng, Deep deterministic policy gradient algorithm based lateral and longitudinal control for autonomous driving, in: Proceedings of the 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), IEEE, 2020, pp. 740–745.

[43] Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms." arXiv preprint (2017).

[44] Attard, Daniel, and Josef Bajada. "Autonomous Navigation of Tractor-Trailer Vehicles through Roundabout Intersections." arXiv preprint (2024).

[45] Bilkan Ince, Hyo-Sang Shin, Antonios Tsourdos, Adversarial proximal policy optimisation for robust reinforcement learning, in: Proceedings of the AIAA SCITECH 2024 Forum, 2024, p. 1697.

[46] Abhinav Sinha, Devin White, Yongcan Cao, Deep reinforcement learning-based optimal time-constrained intercept guidance, in: Proceedings of the AIAA SCITECH 2024 Forum, 2024, p. 2206.

[47] Zarrouki, Baha, Marios Spanakakis, and Johannes Betz. "A Safe Reinforcement Learning driven Weights-varying Model Predictive Control for Autonomous Vehicle Motion Control." arXiv preprint (2024).

[48] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, Philipp Moritz, Trust region policy optimization, in: Proceedings of the International conference on machine learning, PMLR, 2015, pp. 1889–1897.

[49] Jia Liu, Yunduan Cui, Jianghua Duan, Zhengmin Jiang, Zhongming Pan, Kun Xu, Huiyun Li, Reinforcement learning-based high-speed path following control for autonomous vehicles, IEEe Trans. Veh. Technol. (2024).

[50] Xuesong Wang, Jiazhi Zhang, Diyuan Hou, Yuhu Cheng, Autonomous driving based on approximate safe action, IEEE Trans. Intell. Transport. Syst. (2023).

[51] Xiaohui Zhang, Jie Sun, Yunpeng Wang, Jian Sun, A hierarchical framework for multi-lane autonomous driving based on reinforcement learning, IEEE Open J.f Intell. Transport. Syst. (2023).

[52] Avantika Sivakumar, Shruthi Gopalakrishnan, Aditya Ramachandran, S. Chitrakala, GRAND: graph laplacian-based reinforcement learning for autonomous navigation decisions, in: Proceedings of the 2023 IEEE World Conference on Applied Intelligence and Computing (AIC), IEEE, 2023, pp. 204–209.

[53] Xu Wang, Erke Shang, Bin Dai, Yiming Nie, Qiguang Miao, Deep reinforcement learning-based off-road path planning via low-dimensional simulation, IEEE Trans. Intell. Veh. (2023).

[54] Kaiser, Lukasz, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan et al. "Model-based reinforcement learning for atari." arXiv preprint (2019).

[55] Thomas M. Moerland, Joost Broekens, Aske Plaat, Catholijn M. Jonker, Model-based reinforcement learning: a survey, Foundations Trends® Machine Learning 16 (1) (2023) 1–118.

[56] Shuping Chen, Huiyan Chen, MPC-based path tracking with PID speed control for autonomous vehicles, in: Proceedings of the IOP Conference Series: Materials Science and Engineering 892, IOP Publishing, 2020 012034.

[57] Jianyu Chen, Zining Wang, Masayoshi Tomizuka, Deep hierarchical reinforcement learning for autonomous driving with distinct behaviors, in: Proceedings of the 2018 IEEE intelligent vehicles symposium (IV), IEEE, 2018.

[58] Mario Zanon, Janick V. Frasch, Milan Vukov, Sebastian Sager, Moritz Diehl, Model predictive control of autonomous vehicles. Optimization and Optimal Control in Automotive Systems, Springer International Publishing, Cham, 2014, pp. 41–57.

[59] Fengying Dang, Dong Chen, Jun Chen, Zhaojian Li, Event-triggered model predictive control with deep reinforcement learning for autonomous driving, IEEE Trans. Intell. Veh. (2023).

[60] Flavia Sofia Acerbo, Jan Swevers, Tinne Tuytelaars, Tong Duy Son, Evaluation of MPC-based imitation learning for human-like autonomous driving, IFAC-PapersOnLine 56 (2) (2023) 4871–4876.

[61] Lei Du, Bolin Sun, Xujiang Huang, Xiaoyi Wang, Pu Li, A learning-based nonlinear model predictive control approach for autonomous driving, IFAC-PapersOnLine 56 (2) (2023) 2792–2797.

[62] Mohammad Al-Sharman, Rowan Dempster, Mohamed A. Daoud, Mahmoud Nasr, Derek Rayside, William Melek, Self-Learned autonomous driving at unsignalized intersections: a hierarchical reinforced learning approach for feasible decision-making, IEEE Trans. Intell. Transport. Syst. (2023).

[63] Kang Yuan, Yanjun Huang, Shuo Yang, Mingzhi Wu, Dongpu Cao, Qijun Chen, Hong Chen, Evolutionary decision-making and planning for autonomous driving: a hybrid augmented intelligence framework, IEEE Trans. Intell. Transport. Syst. (2024).

[64] Xiaolin Tang, Kai Yang, Hong Wang, Jiahang Wu, Yechen Qin, Wenhao Yu, Dongpu Cao, Prediction-uncertainty-aware decision-making for autonomous vehicles, IEEE Trans. Intell. Veh. 7 (4) (2022) 849–862.

[65] Yujing Hu, Weixun Wang, Hangtian Jia, Yixiang Wang, Yingfeng Chen, Jianye Hao, Feng Wu, Changjie Fan, Learning to utilize shaping rewards: a new approach of reward shaping, Adv. Neural Inf. Process Syst. 33 (2020) 15931–15941.

[66] Sabrina M. Hegner, Ardion D. Beldad, Gary J. Brunswick, In automatic we trust: investigating the impact of trust, control, personality characteristics, and extrinsic and intrinsic motivations on the acceptance of autonomous vehicles, Int. J. Human–Comput. Interaction 35 (19) (2019) 1769–1780.

[67] Zarrouki, Baha, Marios Spanakakis, and Johannes Betz. "A Safe Reinforcement Learning driven Weights-varying Model Predictive Control for Autonomous Vehicle Motion Control." arXiv preprint (2024).

[68] Christopher Diehl, Timo Sebastian Sievernich, Martin Krüger, Frank Hoffmann, Torsten Bertram, Uncertainty-aware model-based offline reinforcement learning for automated driving, IEEe Robot. Autom. Lett. 8 (2) (2023) 1167–1174.

[69] Carl-Johan Hoel, Krister Wolff, Leo Laine, Ensemble quantile networks: uncertainty-aware reinforcement learning with applications in autonomous driving, IEEE Trans. Intell. Transport. Syst. (2023).

[70] Tommaso Benciolini, Dirk Wollherr, Marion Leibold, Non-Conservative trajectory planning for automated vehicles by estimating intentions of dynamic obstacles, IEEE Trans. Intell. Veh. (2023).

[71] Yifan Hu, Junjie Fu, Guanghui Wen, Safe reinforcement learning for model-reference trajectory tracking of uncertain autonomous vehicles with model-based acceleration, IEEE Trans. Intell. Veh. (2023).

[72] Zarrouki, Baha, Chenyang Wang, and Johannes Betz. "A stochastic nonlinear model predictive control with an uncertainty propagation horizon for autonomous vehicle motion control." arXiv preprint (2023).

[73] Sampo Kuutti, Richard Bowden, Yaochu Jin, Phil Barber, Saber Fallah, A survey of deep learning applications to autonomous vehicle control, IEEE Trans. Intell. Transport. Syst. 22 (2) (2020) 712–733.

[74] Yang Zhou, Baihua Li, Jiangtao Wang, Emanuele Rocco, Qinggang Meng, Discovering unknowns: context-enhanced anomaly detection for curiosity-driven autonomous underwater exploration, Pattern Recogn. 131 (2022) 108860.

[75] Yair Wiseman, Ilan Grinberg, Circumspectly crash of autonomous vehicles, in: Proceedings of the 2016 IEEE International Conference on Electro Information Technology (EIT), IEEE, 2016.

[76] Chun Kai Ling, Kian Hsiang Low, Patrick Jaillet, Gaussian process planning with Lipschitz continuous reward functions: towards unifying Bayesian optimization, active learning, and beyond, in: Proceedings of the AAAI Conference on Artificial Intelligence 30, 2016.

[77] Nesma M. Ashraf, Reham R. Mostafa, Rasha H. Sakr, M.Z. Rashad, Optimizing hyperparameters of deep reinforcement learning for autonomous driving based on whale optimization algorithm, PLoS ONE 16 (6) (2021) e0252754.

[78] Javier García, Fernando Fernández, A comprehensive survey on safe reinforcement learning, J. Machine Learn. Res. 16 (1) (2015) 1437–1480.

[79] Shehryar Malik, Usman Anwar, Alireza Aghasi, Ali Ahmed, Inverse constrained reinforcement learning, in: Proceedings of the International conference on machine learning, PMLR, 2021, pp. 7390–7399.

[80] Lu Wen, Jingliang Duan, Shengbo Eben Li, Shaobing Xu, Huei Peng, Safe reinforcement learning for autonomous vehicles through parallel constrained policy optimization, in: Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2020, pp. 1–7.

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR

[81] Dongsheng Ding, Kaiqing Zhang, Tamer Basar, Mihailo Jovanovic, Natural policy gradient primal-dual method for constrained markov decision processes, Adv. Neural Inf. Process Syst. 33 (2020) 8378–8390.

[82] Long Wang, Fangmin Ye, Yibing Wang, Jingqiu Guo, Ioannis Papamichail, Markos Papageorgiou, Simon Hu, Lihui Zhang, A Q-learning foresighted approach to ego-efficient lane changes of connected and automated vehicles on freeways, in: Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE, 2019, pp. 1385–1392.

[83] Changxi You, Jianbo Lu, Dimitar Filev, Panagiotis Tsiotras, Highway traffic modeling and decision making for autonomous vehicle using reinforcement learning, in: Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 1227–1232.

[84] Junho Lee, Jun Won Choi, May i cut into your lane?: a policy network to learn interactive lane change behavior for autonomous driving, in: Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE, 2019.

[85] Wei Yuan, Ming Yang, Yuesheng He, Chunxiang Wang, Bing Wang, Multi-reward architecture-based reinforcement learning for highway driving policies, in: Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE, 2019, pp. 3810–3815.

[86] H. Friji, H. Ghazzai, H. Besbes, Y. Massoud, A DQN-based autonomous car-following framework using RGB-D frames, in: Proceedings of the 2020 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT), Dubai, United Arab Emirates, 2020, pp. 1–6, https://doi.org/10.1109/GCAIoT51063.2020.9345899.

[87] Li, Changjian, and Krzysztof Czarnecki. "Urban driving with multi-objective deep reinforcement learning." arXiv preprint (2018).

[88] Soumyajit Chakraborty, Subhadeep Kumar, Nirav Bhatt, Ramkrishna Pasumarthy, End-to-end autonomous driving in heterogeneous traffic scenario using deep reinforcement learning, in: Proceedings of the 2023 European Control Conference (ECC), IEEE, 2023, pp. 1–6.

[89] Carl-Johan Hoel, Krister Wolff, Leo Laine, Automated speed and lane change decision making using deep reinforcement learning, in: Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2018.

[90] Lei Zeng, Wen Hu, Bangji Zhang, Yang Wu, Dongpu Cao, Risk-aware deep reinforcement learning for decision-making and planning of autonomous vehicles, in: Proceedings of the 2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI), IEEE, 2022, pp. 1–6.

[91] Julian Bernhard, Robert Gieselmann, Klemens Esterle, Alois Knoll, Experience-based heuristic search: robust motion planning with deep Q-learning, in: Proceedings of the 2018 21st International conference on intelligent transportation systems (ITSC), IEEE, 2018, pp. 3175–3182.

[92] Martin Buechel, Alois Knoll, Deep reinforcement learning for predictive longitudinal control of automated vehicles, in: Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2018.

[93] Chris Paxton, Vasumathi Raman, Gregory D. Hager, Marin Kobilarov, Combining neural networks and tree search for task and motion planning in challenging environments, in: Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2017, pp. 6059–6066.

[94] Alexander Amini, Igor Gilitschenski, Jacob Phillips, Julia Moseyko, Rohan Banerjee, Sertac Karaman, Daniela Rus, Learning robust control policies for end-to-end autonomous driving from data-driven simulation, IEEe Robot. Autom. Lett. 5 (2) (2020) 1143–1150.

[95] Adith Boloor, Xin He, Christopher Gill, Yevgeniy Vorobeychik, Xuan Zhang, Simple physical adversarial examples against end-to-end autonomous driving models, in: Proceedings of the 2019 IEEE International Conference on Embedded Software and Systems (ICESS), IEEE, 2019, pp. 1–7.

[96] Peter AM Ruijten, Jacques MB Terken, Sanjeev N. Chandramouli, Enhancing trust in autonomous vehicles through intelligent user interfaces that mimic human behavior, Multimodal. Technol. Interact. 2 (4) (2018) 62.

[97] Jamie Craig, Mehrdad Nojoumian, Should self-driving cars mimic human driving behaviors?, in: Proceedings of the International Conference on Human-Computer Interaction, Cham Springer International Publishing, 2021.

[98] Joris Dinneweth, Abderrahmane Boubezoul, René Mandiau, Stéphane Espié, Multi-agent reinforcement learning for autonomous vehicles: a survey, Autonomous Intell. Syst. 2 (1) (2022) 27.

[99] Lukas M. Schmidt, Johanna Brosig, Axel Plinge, Bjoern M. Eskofier, Christopher Mutschler, An introduction to multi-agent reinforcement learning and review of its application to autonomous mobility, in: Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2022, pp. 1342–1349.

[100] Peter. Wagner, Traffic control and traffic management in a transportation system with autonomous vehicles, Autonomous Driving (2016) 301–316.

[101] Tamás Tettamanti, Arash Mohammadi, Houshyar Asadi, István Varga, A two-level urban traffic control for autonomous vehicles to improve network-wide performance, Transport. Res. Proc. 27 (2017) 913–920.

[102] Shukai Chen, Hua Wang, Qiang Meng, An optimal dynamic lane reversal and traffic control strategy for autonomous vehicles, IEEE Trans. Intell. Transport. Syst. 23 (4) (2021) 3804–3815.

[103] Ranjeev Mittu, Donald Sofge, Alan Wagner, William Frere Lawless (Eds.), Robust Intelligence and Trust in Autonomous Systems, Springer US, 2016.

[104] Jorge I. Poveda, Mouhacine Benosman, Andrew R. Teel, Ricardo G. Sanfelice, Robust coordinated hybrid source seeking with obstacle avoidance in multivehicle autonomous systems, IEEE Trans. Automat. Contr. 67 (2) (2021) 706–721.

[105] Andreia Martinho, Nils Herber, Maarten Kroesen, Caspar Chorus, Ethical issues in focus by the autonomous vehicles industry, Transp. Rev. 41 (5) (2021) 556–577.

[106] Jean-Fran¸cois Bonnefon, Azim Shariff, Iyad Rahwan, The social dilemma of autonomous vehicles, Science (1979) 352 (6293) (2016) 1573–1576.

[107] Alexander Hevelke, Julian Nida-Rümelin, Responsibility for crashes of autonomous vehicles: an ethical analysis, Sci. Eng. Ethics 21 (2015) 619–630.

[108] Hong Wang, Amir Khajepour, Dongpu Cao, Teng Liu, Ethical decision making in autonomous vehicles: challenges and research progress, IEEE Intell. Transport. Syst. Magazine 14 (1) (2020) 6–17.

[109] Matthew O'Kelly, Aman Sinha, Hongseok Namkoong, Russ Tedrake, John C. Duchi, Scalable end-to-end autonomous vehicle testing via rare-event simulation, in: Adv. Neural Inf. Process Syst., 31, 2018.

[110] Mark Colley, Marcel Walch, Enrico Rukzio, Unveiling the lack of scalability in research on external communication of autonomous vehicles, in: Proceedings of the Extended abstracts of the 2020 chi conference on human factors in computing systems, 2020.

[111] Bernardin Jr, L. Vincent, Theodore Mansfield, Benjamin Swanson, Hadi Sadrsadat, Sumit Bindra, Scenario modeling of autonomous vehicles with trip-based models, Transp. Res. Rec. 2673 (10) (2019) 261–270.

[112] Kevin K. Gifford, Robin R. Murphy, Incorporating terrain uncertainties in autonomous vehicle path planning, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96 3, IEEE, 1996.

[113] Eduardo Felipe Zambom Santana, Gustavo Covas, Fábio Duarte, Paolo Santi, Carlo Ratti, Fabio Kon, Transitioning to a driverless city: evaluating a hybrid system for autonomous and non-autonomous vehicles, Simul. Model. Pract. Theory. 107 (2021) 102210.

[114] Wymann, Bernhard, Eric Espié, Christophe Guionneau, Christos Dimitrakakis, Rémi Coulom, and Andrew Sumner. "Torcs, the open racing car simulator." Software available at http://torcs.sourceforge.net 4, no. 6 (2000): 2.

[115] Alexander Buyval, Aidar Gabdullin, Ruslan Mustafin, Ilya Shimchik, Realtime vehicle and pedestrian tracking for didi udacity self-driving car challenge, in: Proceedings of the 2018 IEEE international conference on robotics and automation (ICRA), IEEE, 2018, pp. 2064–2069.

[116] W. Schwarting, J. Alonso-Mora, D. Rus, Planning and decision-making for autonomous vehicles, Annu Rev. Control Robot. Auton. Syst. 1 (2018) 187–210. May 28.

[117] Jason Kong, Mark Pfeiffer, Georg Schildbach, Francesco Borrelli, Kinematic and dynamic vehicle models for autonomous driving control design, in: Proceedings of the 2015 IEEE intelligent vehicles symposium (IV), IEEE, 2015, pp. 1094–1099.

[118] Philip Polack, Florent Altché, Brigitte d'Andréa-Novel, Arnaud de La Fortelle, The kinematic bicycle model: a consistent model for planning feasible trajectories for autonomous vehicles?, in: Proceedings of the 2017 IEEE intelligent vehicles symposium (IV) IEEE, 2017, pp. 812–818.

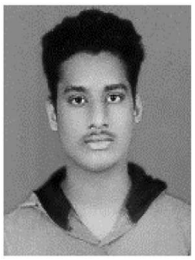
[119] R. Rajamani, Vehicle Dynamics and Control, Springer Science & Business Media, 2011.

[120] Owoyemi, J. (2018, April 26). Kalman Filter: predict, Measure, Update, Repeat. - Joshua Owoyemi - Medium. Retrieved from https://tjosh.medium.com.

Rohan Inamdar is presently pursuing a Bachelor of Technology in Electronics and Computer Engineering at Vellore Institute of Technology, Chennai. He has keen interest in computer vision and robotics.

S. Kavin Sundarr is presently pursuing a Bachelor of Technology in Electronics and Computer Electronics at Vellore Institute of Technology, Chennai. His area of interest are robotics, deep learning, and computer vision.

192.168.2.173 thienlu 2025-02-07 11:04 BOS SEMICONDUCTOR



Deepen Khandelwal is presently pursuing a Bachelor of Technology in Electronics and Computer Engineering at Vellore Institute of Technology, Chennai. He has passion for Machine learning, Deep learning and Robotics and Automation.

Varun Dev Sahu is presently pursuing a Bachelor of Technology in Electronics and Computer Engineering at Vellore Institute of Technology, Chennai. He is having interest in computer vision, Robotics and automation.

Nitish Katal is currently working as Assistant Professor (Senior Grade) with Vellore Institute of Technology, Chennai. He has obtained his PhD Degree in Robust Control Systems from Punjab Engineering College (Deemed to be University) Chandigarh. His areas of research are robust and optimal control systems, data-driven control systems, soft computing and deep learning.

BOS SEMICONDUCTOR