IEEE Open Journal of Control Systems

Received 12 March 2025; revised 23 June 2025 and 4 August 2025; accepted 9 August 2025. Date of publication 20 August 2025; date of current version 10 September 2025. Recommended by Guest Editor Peter Seiler.

Digital Object Identifier 10.1109/OJCSYS.2025.3600925

MDP-Based High-Level Decision-Making for Combining Safety and Optimality: Autonomous Overtaking

XUE-FANG WANG [1] (Member, IEEE), JINGJING JIANG [2] (Member, IEEE), AND WEN-HUA CHEN [2,3] (Fellow, IEEE)

(Safe Motion Planning and Control for Autonomous Driving Under Multi-Source Uncertainty)

[1]School of Engineering, University of Leicester, LE1 7RH Leicester, U.K
[2]Department of Aeronautical and Automotive Engineering, Loughborough University, LE11 3TU Loughborough, U.K.
[3]Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong

CORRESPONDING AUTHOR: WEN-HUA CHEN (e-mails: w.chen@lboro.ac.uk; wenhua.chen@polyu.edu.hk).

This work was supported by U.K. Engineering and Physical Sciences Research Council (EPSRC) Established Career Fellowship under Grant EP/T005734/1.

ABSTRACT This paper presents a novel solution for optimal high-level decision-making in autonomous overtaking on two-lane roads, considering both opposite-direction and same-direction traffic. The proposed solutionaccounts for key factors such as safety and optimality, while also ensuring recursive feasibility and stability.To safely complete overtaking maneuvers, the solution is built on a constrained Markov decision process (MDP) that generates optimal decisions for path planners. By combining MDP with model predictive control (MPC), the approach guarantees recursive feasibility and stability through a baseline control policy that calculates the terminal cost and is incorporated into a constructed Lyapunov function. The proposed solution is validated through five simulated driving scenarios, demonstrating its robustness in handling diverse interactions within dynamic and complex traffic conditions.

INDEX TERMS Markov decision process, decision making under uncertain environments, autonomous overtaking, model predictive control.

I. INTRODUCTION

Autonomous overtaking on two-lane roads, whether involving oncoming traffic or vehicles traveling in the same direction, is a common yet highly challenging driving task. It plays a crucial role in improving trip efficiency by overtaking slower or stationary vehicles ahead (see, for example, [4], [16], [35], [36], [39], [10]). The core difficulty lies in guaranteeing safety across various overtaking scenarios, both during the decision-making phase and the execution of the maneuver. To perform safe overtaking on two-lane roads, several critical factors must be taken into account, including the presence of oncoming vehicles, interactions with vehicles in adjacent lanes (either in the same or opposite direction), relative distances and speeds, and diverse environmental or traffic-related conditions (e.g., road layout, weather, and visibility). These complexities require decision-making frameworks that operate effectively under uncertainty while ensuring both safety and efficiency.

To execute an overtaking maneuver safely and effectively, the ego vehicle must evaluate the surrounding traffic conditions, including available space in adjacent lanes. This assessment is critical not only when overtaking slower vehicles by temporarily entering an oncoming lane, where the risk of encountering opposing traffic is high, but also when changing lanes in the same direction, where fast-approaching vehicles from behind must be considered. In both cases, the maneuver typically occurs at high speeds and requires precise judgment to ensure sufficient clearance and maintain safety. Furthermore, the ego vehicle's field of view is often partially obstructed by the preceding vehicle, especially when approaching in preparation for overtaking (see Fig. 1). Ensuring safety during overtaking requires the satisfaction of

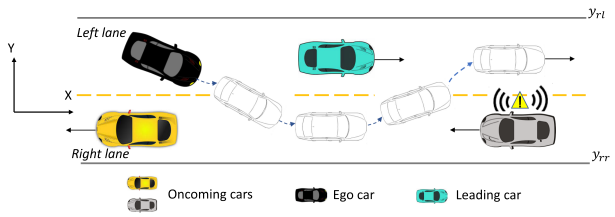

FIGURE 1. Autonomous overtaking with the oncoming traffic.

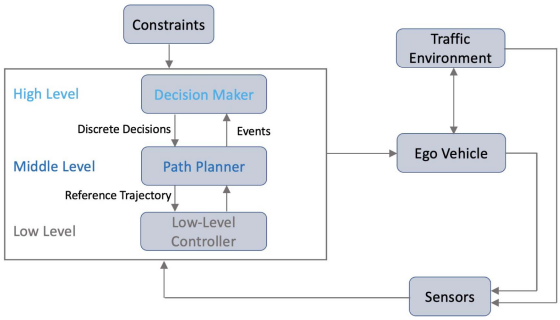

FIGURE 2. Hierarchical framework with three layers.

multiple constraints, while the high-level decision maker must handle rapidly changing and uncertain scenarios, including unexpected oncoming traffic. These factors present significant challenges in completing the maneuver safely and successfully, particularly when interacting with opposing traffic, as considered in this paper, which is widely regarded as one of the most challenging and hazardous maneuvers for human drivers. Therefore, given the inherent challenges of overtaking, it is crucial to make a series of well-informed, high-level decisions to perform the maneuver in a safe and timely manner, ultimately improving trip efficiency while meeting safety requirements.

In such scenarios, decision-making plays a central role in the ego vehicle, acting as the brain that bridges environment perception and motion control (see, for example, [6], [7], [8], [17], [29], [31], [35], [38], [28]). A hierarchical control structure is often employed to manage this task effectively [37]. The perception module first gathers and processes environmental data from sensor readings, providing the essential information needed for decision-making. Based on these perception outputs, the decision maker chooses a suitable action from a predefined set of possible choices, such as {"*maintain*", "*abandon*", "*initialize*", "*prepare*", "*recover*"}. The high-level decision is forwarded to the path planner, which generates a trajectory accordingly. The low-level controller then follows this trajectory, ensuring the overtaking maneuver is carried out smoothly and safely (see Fig. 2).

Existing methods for decision-maker design often lack the necessary assurance of safety and optimality of performance in complex environments. For example, current reinforcement learning approaches typically do not provide guarantees of stability or constraint satisfaction (see, for example, [2], [5], [11], [13], [22], [33], [41]). This limitation poses significant challenges for safety-critical applications such as autonomous overtaking, where verifying safe behavior is particularly difficult due to the wide variability in driving scenarios [11], [13], [22], [41]. To address this concern, Zhang et al. [41] studied a reinforcement learning approach with safety constraints, where a safe policy is learned through optimization techniques. Nevertheless, their framework does not explicitly consider high-level decision-making under dynamic and uncertain traffic conditions, which are essential for real-world deployment. Rule-based methods are also commonly used in designing high-level decision-makers for autonomous driving, but they come with several drawbacks. These methods can be error-prone, and correct behavior can only be ensured through exhaustive testing [23]. As demonstrated in [42], even carefully designed rule-based systems struggle to handle dynamic scenarios (e.g., merging vehicles with sudden acceleration) resulting in collision rates. Given the unpredictable nature of driving and road conditions, it is impossible to anticipate and account for every possible scenario during the design stage. Any oversight or omission, such as the failure to handle real-time constraint violations in [42], can lead to potentially dangerous consequences. Kim and Langari [21] employed a two-player game-theoretic framework to determine the optimal timing and necessity of lane-changing decisions. While this approach can yield intelligent decisions, it does not incorporate safety considerations into the decision-making process.

MDP-based frameworks provide a principled approach for modelling and solving sequential decision-making problems under uncertainty [14], [24], [26], [33]. However, while current work using the MDP approach can increase the probability of safe actions, it does not provide absolute guarantees, as there remains a risk of unsafe actions occurring [3], [15], [43], due to two key factors: the probabilistic nature of state transitions and the absence of strict constraint enforcement.

In addition, [35] integrates rule-based safety checks with MDP decision-making to improve overall safety. However, the method has two key limitations: 1) Safety rules are invoked only after the MDP has proposed an action, which may lead to violations of real-time constraints; 2) The tree-search-based MDP solver does not guarantee recursive feasibility in the presence of dynamic traffic environments.

Inspired by the above insights and considering that substantial progress has already been made in path planning and low-level vehicle control (see, for example, [9], [12], [18], [19], [20], [25], [30], [34], [40]), this work shifts its focus toward high-level decision-making, a critical yet less explored layer in autonomous vehicle control systems. Different from [35], our method integrates hard safety constraints directly into the MDP-MPC optimization, enabling proactive and provably safe decisions. The principal contributions are outlined as follows:

1) To facilitate autonomous decision-making in complex environments where interactions with other road users are captured, a safety-constrained MDP framework is

TABLE 1. Symbol Definitions and Physical Meanings

Symbol	Definition
y_{rl}	Left boundary of the *left lane*
y_{rr}	Right boundary of the *right lane*
y_{lc}	Lateral position of the centre line of *left lane*
y_{rc}	Lateral position of the centre line of *right lane*
v_c	Cruising speed for the original lane
v_{rc}	Cruising speed for the adjacent lane
v_s	User-defined speed (less than cruising speed)
Δt	Sampling time
Δx_j	Longitudinal safety margin (Vehicle j)
Δy_j	Lateral safety margin (Vehicle j)

proposed. This formulation allows us to formulate the generation of safe and optimal decisions as a control problem for MDP.

2) The proposed solution effectively handles dynamic scenarios through a safety-constrained MDP approach. This approach enables the specification of safety constraints, such as maintaining a minimum safe margin (i.e., elliptical collision boundaries), that must be satisfied at every time step between the ego vehicle and other road users, even when their behavior is partially unpredictable. To solve the safety-constrained MDP, an MPC scheme with constraint enforcement is designed. Compared with [42], our MDP-MPC framework dynamically adapts to uncertainty through online optimization while guaranteeing safety via hard constraints.
3) Recursive feasibility and stability of the new design are guaranteed under mild conditions without the need for terminal constraints. This is achieved by carefully designing a baseline control policy and reformulating the MDP problem with an associated cost.

The rest of the paper is structured as follows. Section II provides some preliminaries, and Section III presents the formulation of high-level decision-making for overtaking using MDP. Section IV proposes an MPC-based solution to the safety-constrained optimal MDP problem. Section V analyzes the proposed MDP-based approach, with a focus on recursive feasibility and stability. Section VI presents five different driving scenarios to evaluate the power of our new design. We also evaluate our method against one state-of-the-art baseline given in [42]. Section VII further analyzes the safety performance and computational efficiency of the algorithm. Finally, the main conclusions of the study are summarized in Section VIII.

II. PRELIMINARIES

A. NOTATION

Let $\mathbb{I}$ denote the set of integers, and $\mathbb{R}$ denote the set of real numbers. These symbols may include subscripts or superscripts for clarity when context demands. Given a set $\mathcal{D} \subset \mathbb{R}^n$, we define its complement as $\mathcal{D}^c = \mathbb{R}^n \setminus \mathcal{D}$.

B. MARKOV DECISION PROCESS

The MDP is a fundamental formulation for a hybrid system [1], [32]. A finite-horizon MDP $\mathcal{M}$ is a tuple $(\mathcal{S}, \mathcal{A}, f, J, \gamma)$, where $\mathcal{S} \in \mathbb{R}^{|\mathcal{S}|}$ represents the state space, $\mathcal{A} \in \mathbb{R}^{|\mathcal{A}|}$ denotes the action space, $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the state transition function that maps a given state-action pair to the subsequent state, $J(s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ defines the scalar cost associated with performing action a in state s and $\gamma \in (0, 1]$ is the discount rate that gives more weight to short-term reward. In this work, we assume complete knowledge of all elements in $\mathcal{M}$, and the state transitions follow the *Markovian*.

III. HIGH-LEVEL DECISION MODELLING VIA MDP

In autonomous driving, the ego vehicle operates as an intelligent agent that must make decisions while navigating complex and ever-changing environments. A key challenge in this process is to find an effective way to abstract both the behaviour of the ego vehicle and that of surrounding traffic participants (see, for example [27]). While the decisions themselves, such as lane following, waiting, and overtaking, are inherently discrete, they are governed by the vehicle's continuous dynamics and must respond to the continuously evolving context of the environment. Therefore, the abstraction must support integration into a decision-making framework that accounts for both the discrete nature of high-level choices and the continuous evolution of the underlying physical system. To address this challenge, integrating MDP into the decision-making framework of autonomous driving offers great promise, as it enables the system to simultaneously gather environmental information and exploit learned knowledge to optimize decisions. It is worth noting that the British transport rules are adopted in this paper, and the results can be easily converted to other traffic rules.

A. MODELING AUTONOMOUS OVERTAKING WITH MDP

To demonstrate how to use MDP modelling for autonomous decision-making process, we consider a specific case: Autonomous overtaking, depicted in Fig. 1.

To successfully and safely execute an overtaking maneuver, the ego vehicle must navigate through multiple discrete event states, including *lane following*, *waiting*, and *overtaking*. To ensure safety during autonomous overtaking maneuvers in complex traffic environments, we propose an MDP-based decision maker, as illustrated in Fig. 3. This innovative approach integrates safety constraints directly into the decision-making process (details will be given in Section IV), effectively addressing the challenges associated with autonomous overtaking.

Specifically, the components of $\mathcal{M}$ are defined as follows:

1) *State Space $\mathcal{S}$:* As shown in Fig. 3, three MDP states are considered. Hence, $\mathcal{S} := \{Lanefollowing(S_1), Waiting(S_2), Overtaking(S_3)\}$. Specifically, in state S_1, the ego vehicle should position itself on the *left lane* with a cruising speed; in state S_2, the ego vehicle should slow down or completely stop on the *left lane*, i.e., the velocity is less than the cruising speed; in state S_3, the ego vehicle positions itself on the *right lane* and aims

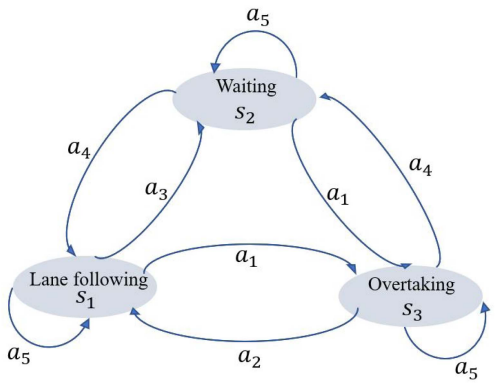

FIGURE 3. MDP state transition.

TABLE 2. State Transition Undertaken Actions

$1(a_k)$	S_1	S_2	S_3
S_1'	$1(a_5)$	$1(a_4)$	$1(a_2)$
S_2'	$1(a_3)$	$1(a_5)$	$1(a_4)$
S_3'	$1(a_1)$	$1(a_1)$	$1(a_5)$

to overtake the slower or stationary leading vehicle as quick as possible.

2) *Action Space $\mathcal{A}$ and Policy π:* To change system state from S_i to S_j where $i, j \in \{1, 2, 3\}$, proper action needs to be taken. In the overtaking problem, five actions are considered: *initialize (a_1), recover (a_2), prepare (a_3), abandon (a_4)* and *maintain (a_5)*. Therefore, the action space $\mathcal{A}$ is defined as $\mathcal{A} := \{a_1, a_2, a_3, a_4, a_5\}$. A policy π is selected to minimize the cumulative cost defined in Section IV-B. Under deterministic conditions, the policy π serves as a function that maps each state to a corresponding action, i.e., $\pi : \mathcal{S} \to \mathcal{A}$.

3) *States Transition Matrix:* The notation $S_i \xrightarrow{1(a_k)} S_j'$ represents a transition from state S_i to the resulting state S_j' when action a_k is applied, where $i, j \in \{1, 2, 3\}$, $k \in \{1, \ldots, 5\}$, and $1(a_k)$ is a function defined as 1 when action a_k is taken and 0, otherwise. The system transition matrix is defined in Table 2 in accordance to Fig. 3. Based on the state transition matrix, the relationships illustrated in Fig. 3 can be expressed by transition function:

$$s(k+1) = f(s(k), a(k)). \tag{1}$$

Remark 1: In Fig. 3, action *abandon* (a_4) indicates that the model can revise its decisions even after overtaking or lane transitions have been initiated. In this case, the ego vehicle should position itself in the left lane at an appropriate speed.

B. MODELING EGO VEHICLE AND SURROUNDING VEHICLES

To safely carry out an overtaking maneuver, it is important to effectively model the behaviors of the ego vehicle along with those of surrounding vehicles. Additionally, modelling the road circuit/layout is essential. All this information should be carefully integrated into the decision-making process as constraints. These include a simplified vehicle model, physical and information about oncoming vehicles, based on which physical and safety constraints of the ego vehicle can be formulated.

Below are the dynamic models representing the ego vehicle and the vehicles around it:

Ego Vehicle e :

$$\begin{cases} x_e(k+1) = x_e(k) + v_e(k)\Delta t \\ \begin{bmatrix} v_e(k+1) \\ y_e(k+1) \end{bmatrix} = \begin{cases} \begin{bmatrix} v_c \\ y_{lc} \end{bmatrix}, s(k+1) = S_1 \\ \begin{bmatrix} v_s \\ y_{lc} \end{bmatrix}, \begin{array}{l} s(k+1) = S_2 \text{ and} \\ x_{l_d}(k) - x_e(k) > d_{safe} \end{array} \\ \begin{bmatrix} 0 \\ y_{lc} \end{bmatrix}, \begin{array}{l} s(k+1) = S_2 \text{ and} \\ x_{l_d}(k) - x_e(k) \le d_{safe} \end{array} \\ \begin{bmatrix} v_{rc} \\ y_{rc} \end{bmatrix}, s(k+1) = S_3, \end{cases} \end{cases} \tag{2a}$$

Surrounding Vehicle j :

$$\begin{cases} x_j(k+1) = x_j(k) + v_j(k)\Delta t + \frac{1}{2}a_j(k)(\Delta t)^2 \\ v_j(k+1) = v_j(k) + a_j(k)\Delta t \\ y_j(k+1) = y_j(k), j \in \{l_d, O_r\} \end{cases} \tag{2b}$$

where $x.(k)$, $y.(k)$, $v.(k)$ and $a.(k)$ denote the longitudinal position, the lateral position, longitudinal speed and acceleration at time step k, respectively. v_c, v_{rc}, v_s, y_{lc}, y_{rc} and Δt are defined in Table 1. l_d denotes the leading vehicle and O_r denotes vehicle $r \in O$ in the adjacent lane. Moreover, the following assumption is needed throughout the paper.

Assumption 1: O is a finite set with indices $\{1, 2, \ldots, m\}$.

Remark 2: In the MDP model, if the decision "S_1 (*Lane following*)" is made, then the intermediate-level path planner will guide the ego vehicle to track the reference signal (v_c, y_{lc}) of the *left lane*. If the decision "S_2 (*Waiting*)" is made and the gap between the ego vehicle and the vehicle ahead exceeds a constant threshold d_{safe}, the ego vehicle will initially decelerate to speed v_s to assess the traffic situation. However, if the distance between the ego vehicle and the leading vehicle falls below the selected threshold d_{safe}, the ego vehicle will come to a complete stop, reducing its speed to zero. Without loss of generality, considering both opposite-direction and same-direction traffic on two-lane roads, we stipulate that the ego vehicle can only wait for a safe gap to initiate a change lane in the *left lane*. Hence, the y reference is set to y_{lc}, the same as when $s(k+1) = S_1$. Finally, if the decision is "S_3(*Overtaking*)", the path planner generates smooth paths to ensure that the ego vehicle

tracks the centre reference signal y_{rc} of the *right lane* while maintaining the cruising speed.

Remark 3: This study centers on high-level decision-making and analyzes its performance. Furthermore, a hierarchical control framework is implemented for autonomous vehicles. Hence, it is acceptable to use a relatively simple model (2a) to represent the vehicle's dynamics in the high-level decision-making, as minor inaccuracies will be compensated by the path planning layer and the low-level controller. Note that we assume the low-level controller tracks the planned path perfectly in this paper.

C. OPTIMAL MDP PROBLEM FORMULATION

Minimizing the cumulative cost (rather than the instant cost of the current state) is the ultimate aim of the ego vehicle over time. That is, the ego vehicle aims to optimize its decision-making process over a horizon of time, taking into account the future consequences of its actions. This allows the vehicle to anticipate potential risks, consider changing traffic conditions, and make proactive decisions that prioritize safety and adherence to traffic rules. It is important to design a cost function to induce the ego vehicle to generate desirable and safe behaviour, regardless of whether the surrounding traffic is in the same or opposite direction.

According to different MDP states, here we define costs associated with each state as:

$$J(s(k), a(k)) = \begin{cases} 0, & s(k) = S_1 \\ r_1, & s(k) = S_2 \\ r_2, & s(k) = S_3, \end{cases} \tag{3}$$

where r_1 and r_2 are positive constants with $r_1 > r_2 > 0$. Note that the cost values $r_1 > r_2 > 0$ are designed to enforce the following priorities:

1) Avoid prolonged *Waiting* (S_2): A high cost r_1 penalizes unnecessary stops (e.g., blocking traffic or causing rear-end collisions), ensuring the ego vehicle actively seeks opportunities to complete the overtaking maneuver.
2) Encourage timely *Overtaking* (S_3): A moderate cost r_2 promotes efficient overtaking while still allowing temporary lane changes when safe.
3) Default to *Lane following* (S_1): Zero cost for S_1 reflects the nominal desired state with no additional risk or inefficiency.

To further ensure safe and optimal decision-making in the high-level layer for autonomous overtaking in highly dynamic and uncertain environments, safety constraints play a crucial role in shaping the decision-making process. Therefore, these constraints (i.e., (4d) and (4e)) must be carefully formulated and effectively addressed, which will be incorporated into the following optimization problem.

Finally, the high-level decision-making is represented as the optimization problem below:

$$\min_{a(\cdot)} \sum_{k=0}^{\infty} \gamma^k J(s(k), a(k)) \tag{4a}$$

$$\text{s.t.} \quad State\,transition\,function\,(1) \tag{4b}$$

$$Vehicles'\,dynamics\,(2) \tag{4c}$$

$$\left(\frac{x_e(k) - x_j(k)}{\Delta x_j}\right)^2 + \left(\frac{y_e(k) - y_j(k)}{\Delta y_j}\right)^2 \geq 1 \tag{4d}$$

$$0 < x_{l_d}(k) - x_e(k) < (v_e(k) - v_{l_d}(k))t_{thd}$$

$$\Downarrow\Downarrow$$

$$s(k+1) \neq S_1 \tag{4e}$$

where the discount factor $\gamma \in (0, 1]$ balances the weight between immediate and future costs, and Δx_j, Δy_j are defined in Table 1 and can be calculated by incorporating the geometry (length and width) of Vehicle j and the ego vehicle. These are assumed to be available from sensing. The double arrow implies that constraint (4e) enforces a safety-critical implication: if the leading vehicle is within the time-to-collision threshold t_{thd}, the ego vehicle must abandon "S_1 (*Lane following*)" and switch to "S_3 (*Overtaking*)" or "S_2 (*Waiting*)".

From the perspective of accomplishing overtaking maneuvers, the ego vehicle strives to advance along its designated lane, where the instantaneous cost is minimised (as indicated in (4a)). Simultaneously, constraint (4d) enforces a minimum safe distance between the ego vehicle and surrounding vehicles, modeled as an elliptical boundary. This ensures collision avoidance. However, when a slower (or stationary) leading vehicle appears ahead of the ego vehicle, adhering to safety constraints (4d) and (4e), the ego vehicle is not always permitted to adopt the "S_1 (*Lane following*)" decision. From (3) and (4a) we know that the ego vehicle cannot always take "S_2 (*Waiting*)" state as this will result in larger cost, i.e., contrary to our goal. Consequently, after a certain period, the ego vehicle must complete the overtaking maneuver (as long as its speed surpasses that of the leading vehicle) and resume its trajectory on the road.

This new formulation (4) allows us to initiate, execute, hold or even abandon an overtaking maneuver. Next, an MPC-based framework will be developed to address the optimal MDP problem, enabling the design of high-level control while efficiently managing constraints.

IV. SOLUTION TO SAFETY-CONSTRAINED OPTIMAL MDP PROBLEM

A. MPC-BASED DECISION-MAKING ALGORITHM

The optimal decision-making problem formulated in (4) presents two key challenges. First, autonomous decision-makers must ensure both safety and optimality while operating in highly dynamic and uncertain environments. This necessitates continuous updates of the system and environmental states, along with corresponding adjustments to decisions. Second, solving an infinite-horizon optimization problem in real time is computationally intractable. To overcome these challenges, we employ the receding horizon approach in MPC. This method transforms the infinite-horizon optimization problem into a finite-horizon one, making it

TABLE 3. Mapping Between MDP Actions and MPC Control Objectives

MDP Action	High-Level Decision	MPC Control Objective	Terminal State Reference
a_1 (Initialize)	Start overtaking	Track right lane center y_{rc} with cruising speed v_{rc}; Enforce safety constraints (4d)–(4e).	S_3 (Overtaking)
a_2 (Recover)	Return to the original lane	Track left lane center y_{lc} with speed v_c; Ensure safe distance d_{safe} from the leading vehicle.	S_1 (Lane Following)
a_3 (Prepare)	Slow down/wait	Decelerate to v_s or stop ($v_e = 0$) if $d \le d_{\text{safe}}$; Maintain y_{lc}.	S_2 (Waiting)
a_4 (Abandon)	Abort overtaking	Revert to y_{lc} and adjust speed to avoid collisions; Prioritize (4d) over optimality.	S_1 or S_2 (Context-dependent)
a_5 (Maintain)	Continue current state	Hold current lane (y_{lc} or y_{rc}); Maintain speed (v_c or v_{rc}).	Unchanged (S_1, S_2, or S_3)

more manageable. The optimization problem is then solved repeatedly with regular updates to the system state and environmental information. Table 3 is given to clarify the connection between MDP actions and MPC's control objectives. Moreover, the control algorithm is outlined in Algorithm 1. Unlike standard MDP solvers that use dynamic programming, Algorithm 1 selects actions through online constrained optimization. This avoids the curse of dimensionality while enforcing hard safety guarantees, which is critical for autonomous driving.

In Algorithm 1, $\mathbb{Y}_e := [y_{\min}, y_{\max}]$, $\mathbb{V}_e := [v_{\min}, v_{\max}]$, $t_{thd} > 0$ is to be determined later.

Remark 4: Algorithm 1 captures the essence of Fig. 3 from an optimal control perspective, aligning with the core idea behind MPC-based methods. The MPC formulation approximates the infinite-horizon MDP by solving a finite-horizon optimization at each step. The terminal cost $J_f(s(N))$ is derived from a baseline policy (Section V-A) to guarantee stability without explicit terminal constraints. The ego vehicle aims to minimize the cumulative cost in (4a) by selecting optimal actions, while accounting for MDP state transitions (1), vehicle and environment dynamics (2), and safety constraints (4d)–(4e). In short, our safety-constrained MDP framework ensures that safety requirements are met during decision-making. However, it is important to note that autonomous driving behavior depends heavily on environmental conditions. These can be addressed by adjusting key design parameters in the algorithm. For instance, in rainy or snowy weather, the safety distance parameter d_{safe} should be increased to allow for longer stopping distances.

V. RECURSIVE FEASIBILITY AND STABILITY OF MDP

To analyze the stability of the proposed novel MDP solution, we will first design a baseline control policy π to calculate the terminal cost $J_f(s(N))$ which covers the cost-to-go.

A. BASELINE CONTROL POLICY DESIGN

For simplicity, the following sets are defined as collections of state tuples satisfying specific conditions:

$\mathcal{D}_r$*:* The ego vehicle is approaching the adjacent lane vehicles and the time to collision (TTC) is no greater than a predefined threshold t_{thdr}, i.e.,

$$\mathcal{D}_r := \{(v_e(i;k) - |v_{o_r}(i;k)|)t_{thdr} \ge |x_{o_r}(i;k) - x_e(i;k)|\}. \tag{8}$$

$\mathcal{D}_l$*:* When the ego vehicle is approaching a leading vehicle and the corresponding TTC is no more than t_{thd}, it starts to evaluate whether to change its action from a_5 to alternative ones, i.e.,

$$\mathcal{D}_l := \{x_{l_d}(i;k) - x_e(i;k) \le (v_e(i;k) - v_{l_d}(i;k))t_{thd}\}. \tag{9}$$

$\mathcal{D}_{sel}(\mathcal{D}_{sle})$*:* When the ego vehicle is on the original lane, its distance to the leading vehicle should always be larger than a predefined safe distance d_{safe}, i.e.,

$$\mathcal{D}_{sel} := \{x_e(i;k) - x_{l_d}(i;k) > d_{safe}\}, \tag{10}$$

$$\mathcal{D}_{sle} := \{x_{l_d}(i;k) - x_e(i;k) > d_{safe}\}, \tag{11}$$

where $i \ge N$.

The above sets act as Boolean propositions in the baseline policy designed in the following Tables 4–6.

Since the terminal state of MDP can fall into one of the three states, we design a baseline control policy for each of them.

- Case 1: $s(N;k) = S_1$ (*Lane following*), $y_e(N;k) = y_{lc}$, $(x_e(N;k), v_e(N;k)) \in \mathcal{D}_{sel} \| \mathcal{D}_{sle}$
- Case 2: $s(N;k) = S_2$ (*Waiting*), $y_e(N;k) = y_{lc}$, $(x_e(N;k), v_e(N;k)) \in \mathcal{D}_{sle}$
- Case 3: $s(N;k) = S_3$ (*Overtaking*)

The baseline policies used to calculate the terminal cost are given in Tables 4–6.

Then based on the rule-based policies given in Tables 4–6 and Assumption 1, after a finite time jump we have that

$$\begin{aligned} s(N;k) &= S_1 \to \cdots \to S_3 \to S_1 \\ s(N;k) &= S_2 \to \cdots \to S_3 \to S_1 \\ s(N;k) &= S_3 \to \cdots \to S_3 \to S_1. \end{aligned} \tag{12}$$

Remark 5: From Assumption 1, it is established that the worst case entails the ego vehicle waiting for a finite period, allowing all oncoming vehicles on the opposite lane to pass by. Subsequently, the ego vehicle initiates the overtaking maneuver and eventually returns to its original lane to continue along

Algorithm 1: Optimal MDP Problem Solved via MPC-Based Method.

1: Given initial states $s(0)$, $(x_e(0), y_e(0), v_e(0))$ and $(x_j(0), y_j(0), v_j(0))$.
2: Determine the the optimal action $a^*(i;k)$ for the following optimal problem

$$\min_{a(k)} \sum_{i=0}^{N-1} \gamma^i J(s(i;k), a(i;k)) + J_f(s(N)) \tag{5}$$

$$\text{s.t.} \qquad \text{s(i + 1; k)} = \text{f(s(i; k), a(i; k))},$$

$$\begin{cases} x_e(i+1;k) = x_e(i;k) + v_e(i;k)\Delta t \\ \begin{bmatrix} v_e(i+1;k) \\ y_e(i+1;k) \end{bmatrix} = \begin{cases} \begin{bmatrix} v_c \\ y_{lc} \end{bmatrix}, & s(i+1;k) = S_1 \\ \begin{bmatrix} v_s \\ y_{lc} \end{bmatrix}, & s(i+1;k) = S_2 \text{ and } x_{l_d}(i;k) - x_e(i;k) > d_{safe} \\ \begin{bmatrix} 0 \\ y_{lc} \end{bmatrix}, & s(i+1;k) = S_2 \text{ and } x_{l_d}(i;k) - x_e(i;k) \le d_{safe} \\ \begin{bmatrix} v_{rc} \\ y_{rc} \end{bmatrix}, & s(i+1;k) = S_3, \end{cases} \end{cases}$$

$$y_e(i;k) \in \mathbb{Y}_e,\ v_e(i;k) \in \mathbb{V}_e,$$

$$\begin{cases} x_j(i+1;k) = x_j(i;k) + v_j(i;k)\Delta t + \frac{1}{2}a_j(i;k)(\Delta t)^2 \\ v_j(i+1;k) = v_j(i;k) + a_j(i;k)\Delta t \\ y_j(i+1;k) = y_j(i;k),\ j \in \{l_d, O_r\}, \end{cases}$$

$$\left(\frac{x_e(i;k) - x_j(i;k)}{\Delta x_j}\right)^2 + \left(\frac{y_e(i;k) - y_j(i;k)}{\Delta y_j}\right)^2 \ge 1, \quad j \in \{l_d, O_r\} \tag{6}$$

$$0 < x_{l_d}(i;k) - x_e(i;k) < (v_e(i;k) - v_{l_d}(i;k))t_{thd} \Rightarrow s(i+1;k) \neq S_1. \tag{7}$$

3: Apply the first action $a^*(0;k)$ from the optimal sequence
4: $k \leftarrow k+1$ and go to step 2).

TABLE 4. The Logic Conditions of the Choice of Policy π When the Terminal State Starts From $s(N;k) = S_1$

policy π	logic conditions $(i;k), i \ge N$
$a_5(i;k)$	$\mathcal{D}_{sel} \parallel \mathcal{D}_l^c$
$a_3(i;k)$	$\mathcal{D}_{sle}$ & $\mathcal{D}_l$ & $\mathcal{D}_r$
$a_1(i;k)$	$\mathcal{D}_{sle}$ & $\mathcal{D}_l$ & $\mathcal{D}_r^c$

TABLE 5. The Logic Conditions of the Choice of Policy π When the Terminal State Starts From $s(N;k) = S_2$

policy π	logic conditions $(i;k), i \ge N$
$a_1(i;k)$	$\mathcal{D}_r^c$
$a_5(i;k)$	$\mathcal{D}_r$ & $\mathcal{D}_{sle}$ & $\mathcal{D}_l$
$a_4(i;k)$	$\mathcal{D}_r$ & $\mathcal{D}_l^c$

TABLE 6. The Logic Conditions of the Choice of Policy π When the Terminal State Starts From $s(N;k) = S_3$

policy π	logic conditions $(i;k), i \ge N$
$a_2(i;k)$	$\mathcal{D}_{sel}$
$a_5(i;k)$	$\mathcal{D}_{sel}^c$ & $\mathcal{D}_r^c$
$a_4(i;k)$	$\mathcal{D}_{sel}^c$ & $\mathcal{D}_r$

the road. Consequently, it can be inferred that the occurrence of (12) is a natural outcome of this process.

Next, we will calculate the terminal cost based on the aforementioned baseline control policy. According to (3) and (12), we can use stage cost $J(\cdot)$ to calculate the immediate cost for each step in (12), and then we have

$$J_f(s(N)) = \sum_{i=N}^{\infty} \gamma^i J(s(i;k), a(i;k)). \tag{13}$$

It follows from Assumption 1 that $J_f(s(N))$ in (13) is bounded. This means that there exists a constant $\bar{J}_f > 0$ such that $J_f(s(N)) \le \bar{J}_f$.

Before presenting the recursive feasibility of the proposed MDP solution, we will first give Lemma 1 to show the feasibility of the aforementioned baseline control policy in Tables 4–6. In this lemma, we introduce β as a buffer distance (e.g., braking distance). This guarantees the ego vehicle can stop without collision even if the leading vehicle suddenly brakes. This implies that we have the minimum safe distance $d_{safe} \ge \Delta x_j + \beta$. Additionally, t_{thd}, t_{thdr} represent the maximum allowed time to collision (TTC).

Lemma 1: Suppose that Assumption 1 holds. If t_{thd}, t_{thdr} and d_{safe} satisfy the following inequalities for all $i \ge N$

$$d_{safe} \ge \Delta x_j + \beta,\ j \in \{l_d, O_r\},\ \beta > 0 \tag{14a}$$

$$t_{thd} \ge \frac{d_{safe}}{|v_e(i;k) - v_{l_d}(i;k)|} \tag{14b}$$

$$t_{thdr} \ge \frac{d_{safe}}{|v_e(i;k) - |v_{o_r}(i;k)||}, \ r \in O, \tag{14c}$$

then, the baseline control policy is feasible for each case of the terminal state $s(N;k)$.

Proof: According to Tables 4–6, we will prove the conclusion from the following three cases.

In Case 1, where the ego vehicle is on the *left lane*, we verify the safety constraint based on Table 4 and conditions (14a)–(14c). We established that if the logic condition is ($\mathcal{D}_{sel} \parallel \mathcal{D}_l^c$), inequality (6) holds for the leading vehicle (i.e., $j = l_d$)

and $i = N$. Similarly, suppose the logic condition is ($\mathcal{D}_{sle}$ & $\mathcal{D}_l$ & $\mathcal{D}_r$). In this situation, the safety constraint (6) is satisfied for both the ego vehicle and surrounding traffic, especially for oncoming vehicles, since the ego vehicle remains in its designated lane. Additionally, when the logic condition is ($\mathcal{D}_{sle}$ & $\mathcal{D}_l$ & $\mathcal{D}_r^c$), it is straightforward to verify that (6) holds for all vehicles ($j \in \{l_d, O_r\}$).

In Case 2, it is easy to verify that safety constraint (6) holds for all $j \in \{l_d, O_r\}$ based on Table 5 and conditions (14a)–(14c).

In Case 3, where the ego vehicle is on the *right lane*, we analyze the logic conditions given in Table 6. If logic condition is $\mathcal{D}_{sel}$ or ($\mathcal{D}_{sel}^c$ & $\mathcal{D}_r^c$) and according to (14c), we conclude that (6) holds for all $j \in \{l_d, O_r\}$. Furthermore, in Case 3, we address the logic condition ($\mathcal{D}_{sel}^c$ & $\mathcal{D}_r$). From $s(N; k) = S_3$ we can deduce that at the predicted time step $(N-1; k)$ the following inequality

$$\begin{aligned}(v_e(N-1;k)-v_{o_j}(N-1;k))t_{thdr} < x_{o_j}(N-1;k)\\ -x_e(N-1;k)\end{aligned} \tag{15}$$

holds no matter $s(N-1;k) = S_1$, $s(N-1;k) = S_2$ or $s(N-1;k) = S_3$ (refer to Tables 4–6).

According to (2) we have

$$\begin{aligned}x_{o_j}(N;k) - x_e(N;k) = x_{o_j}(N-1;k) - x_e(N-1;k)\\ + (v_{o_j}(N-1;k) - v_e(N-1;k))\Delta t\\ + \frac{1}{2}a_{o_j}(N-1;k)(\Delta t)^2.\end{aligned} \tag{16}$$

Thus, it follows from (14a), (14c), (15) and (16) that there exists a large enough $\beta > 0$ such that

$$\begin{aligned}x_{o_j}(N;k) - x_e(N;k) &\geq d_{safe} - \beta\\ &\geq \Delta x_j.\end{aligned}$$

This implies that the safety constraint (6) holds. Therefore, we can conclude that the baseline control policy is feasible. The proof is completed. ■

B. RECURSIVE FEASIBILITY

This subsection shows that Algorithm 1 returns a feasible control action at all times if it is feasible at the beginning.

Theorem 1: Provided Assumption 1 and inequalities (14) are valid, feasibility of Algorithm 1 at time $k = 0$ ensures feasibility for all subsequent time steps $k > 0$.

Proof: At time k, the following optimal action and corresponding nominal state are given as:

$$\begin{aligned}a^*(k) &:= (a^*(0;k), a^*(1;k), \ldots, a^*(N-1;k))\\ s^*(k) &:= (s^*(0;k), s^*(1;k), \ldots, s^*(N-1;k), s^*(N;k)).\end{aligned} \tag{17}$$

It follows from the conclusion of Lemma 1 that the following safety constraint holds at prediction horizon N for each case of terminal state:

$$\left(\frac{x_e(N;k) - x_j(N;k)}{\Delta x_j}\right)^2 + \left(\frac{y_e(N;k) - j(N;k)}{\Delta y_j}\right)^2 \geq 1. \tag{18}$$

According to the baseline control policy given in Tables 4–6, at time k, we can figure out the action $a^*(N;k)$ and state $s^*(N+1;k)$.

Correspondingly, the following sequences are constructed and will be proven feasible for time $k+1$:

$$\begin{aligned}a(k+1) &:= (a^*(1;k), \ldots, a^*(N-1;k), a^*(N;k))\\ s(k+1) &:= (s^*(1;k), \ldots, s^*(N-1;k), s^*(N;k), s^*(N+1;k)).\end{aligned} \tag{19}$$

Similarly, from Lemma 1 we have

$$\left(\frac{x_e(N+1;k)-x_j(N+1;k)}{\Delta x_j}\right)^2+\left(\frac{y_e(N+1;k)-y_j(N+1;k)}{\Delta y_j}\right)^2 \geq 1.$$

Thus, we can conclude that (19) is a feasible solution to Algorithm 1 at time $k+1$. The proof is completed. ■

C. STABILITY ANALYSIS

Ensuring the stability of the algorithm is essential when employing MDP to model autonomous overtaking, as it guarantees reliable and accurate model outputs and enhances the practical applicability of the proposed method. Consequently, developing a stable MDP-based control algorithm is vital for effectively managing the overtaking maneuvers of ego vehicles and ensuring their successful execution.

Definition 1: MDP is said to be stable if $s(k) = S_1$ for all time $k \geq k_T$ where k_T is a positive integer.

Theorem 2: Assuming Assumption 1 and conditions (14) are satisfied, Algorithm 1 yields a solution that stabilizes the MDP.

Proof: To demonstrate stability, we first examine the value function of V_N at step k concerning the cost incurred by feasible sequences at step $k+1$. According to (5), at time k, the value function is given by:

$$\begin{aligned}V_N^*(k) &= \sum_{i=0}^{N-1} \gamma^i J(s^*(i;k), a^*(i;k)) + J_f(s^*(N))\\ &= J(s^*(0;k), a^*(0;k)) + \sum_{i=1}^{N-1} \gamma^i J(s^*(i;k), a^*(i;k))\\ &\quad + J_f(s^*(N)),\end{aligned} \tag{20}$$

where $J_f(\cdot)$ is defined in (13) and $J_f(s^*(N)) \in \{J_{f_1}(s^*(N)), J_{f_2}(s^*(N)), J_{f_3}(s^*(N))\}$ according to three choices of terminal sate $s^*(N;k) \in \{S_1, S_2, S_3\}$.

At time step $k+1$, the cost function with the feasible sequences is expressed as:

$$V_N(k+1) = \sum_{i=1}^{N-1} \gamma^i J(s^*(i;k), a^*(i;k)) + \tilde{J}_f(s(N)), \quad (21)$$

where $\tilde{J}_f(s(N))$ can be calculated by combining (13) and (12) with the first term of (13) being removed at time step k. That means the following inequality holds

$$\tilde{J}_f(s(N)) \leq J_f(s^*(N)). \quad (22)$$

Then it follows from (20)–(22) that

$$\begin{aligned} V_N^*(k+1) - V_N^*(k) &\leq V_N(k+1) - V_N^*(k) \\ &\leq -J(s^*(0;k), a^*(0;k)). \end{aligned} \quad (23)$$

Therefore, we can conclude that $V_N^*(k+1) - V_N^*(k) \leq 0$ since all cost J are non-negative. In addition, $V_N^*(k+1) - V_N^*(k) = 0$ if and only if $s(k) = S_1$ (i.e., steady state) which $s(k)$ can achieve after a finite time k_T. Hence, from Definition 1 we conclude that the MDP chain is stable. The proof is completed. ■

Remark 6: This section demonstrates that the proposed MDP-based MPC algorithm for high-level decision-making can successfully and safely accomplish autonomous overtaking maneuvers. The establishment of the formal properties like recursive feasibility and the stability of the proposed autonomous overtaking algorithm have threefold implications: i) guarantee the safety and effectiveness of the autonomous overtaking process; ii) make the new formulation and its solution easier to be applied to real-world autonomous driving scenarios; iii) improve computation efficiency which facilitates real-time applications.

VI. DRIVING SCENARIOS TESTING

In this section, we validate the effectiveness of the proposed MDP-based solution through simulations of two-lane traffic scenarios, including bidirectional country road and single-directional road. For all driving scenarios, the testing environments are created using MATLAB's Driving Scenario Designer.

Remark 7: To bridge the gap between high-level decision-making and actual vehicle control, an MPC-based method is employed in the path-planning layer to generate dynamically feasible trajectories based on the high-level decisions. This hierarchical structure reflects the separation of responsibilities across different time scales: the high-level decision-making module operates at a lower frequency (on the order of seconds), modeling the ego vehicle's behavior using the dynamic model in (2); meanwhile, the MPC-based path-planning layer runs at a higher frequency (on the order of milliseconds), utilizing the widely used kinematic model presented in the system (1) of [36] to ensure real-time responsiveness and smooth execution.

TABLE 7. Parameter Specification

Lane	Values(m)	Parameters	Values	Vehicle	Values(m)
Wide	3.6	$v_c(m/s)$	26	Wide	1.9
y_{rl}	3.1	$v_s(m/s)$	16	Δx_j	4
y_{rr}	-4.1	$d_{safe}(m)$	17	Δy_j	1.6
y_{lc}	1.3	$t_{thd}(s)$	5	l	3
y_{rc}	-2.3	$t_{thdr}(s)$	10	β	13

A. SIMULATION PARAMETERS

To clarify all variables and units used in the simulation, we first define vectors $\boldsymbol{x}_e = (x_e, y_e, \theta_e, v_e)$ with heading angle θ_e and $\boldsymbol{x}_j = (x_j, v_j, y_j)$ with $j \in \{l_d, O_r\}$, where the unit of $\boldsymbol{x}_e$ is $(m, m, rad, m/s)$ and the unit of $\boldsymbol{x}_j$ is $(m, m/s, m)$.

1) The initial state of the ego vehicle is $\boldsymbol{x}_e(0) = [25\,\text{m}, 1.3\,\text{m}, 0\,\text{rd}, 26\,\text{m/s}]$.
2) The MDP states and actions are represented numerically as follows
   - $S_1 := 1, S_2 := 2, S_3 := 3$
   - $a_1 := 4, a_2 := 5, a_3 := 6, a_4 := 7, a_5 := 8$.
3) Prediction horizon and sampling time
   - For the high-level decision-making layer, a prediction horizon of $N = 7$ with a time step of $\Delta t = 1\,\text{s}$ is used.
   - For the path-planning layer, the horizon is set to $N_p = 3$ with a prediction interval of $T_p = 0.25\,\text{s}$.

   In other words, high-level decisions are updated once every four path-planning steps.
4) The instant cost function is defined as

$$J(s(k), a(k)) = \begin{cases} 0, & s(k) = S_1 \\ 10, & s(k) = S_2 \\ 2, & s(k) = S_3. \end{cases} \quad (24)$$

5) Some other parameters are given in Table 7.
6) To avoid division by zero in (14b) and (14c), we add a minimum threshold $\epsilon_v = 0.01\,\text{m/s}$ in our implementation.

Based on the simulation parameters described above, we first examine four driving scenarios involving interactions with oncoming vehicles. In these scenarios, the ego vehicle must safely overtake a slower leading vehicle by temporarily entering the opposite lane without causing a collision. Subsequently, to show the generality of the proposed framework, we consider an additional scenario in which vehicles in both lanes travel in the same direction.

B. SCENARIO 1: NO ONCOMING VEHICLES

In this scenario, we consider the simplest case where the sensors of the ego vehicle detect the presence of a stationary leading vehicle in the same lane and a stationary oncoming vehicle in the opposite lane. The initial state of leading vehicle is $\boldsymbol{x}_{l_d}(0) = [100\,\text{m}, 0\,\text{m/s}, 1.3\,\text{m}]$ with $a_{l_d} = 0\,\text{m/s}^2$ and $v_{rc} = 26\,\text{m/s}$.

The total simulation duration is set to 8 seconds, with the outcomes illustrated in Figs. 4–7.

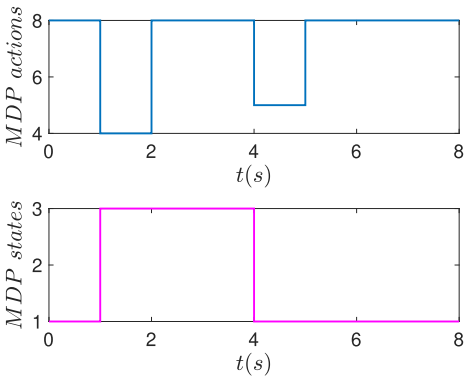

FIGURE 4. MDP state transition in Scenario 1. 1 := *Lane following*, 3 := *Overtaking*, 4 := *abandon*, 5 := *recover*, 8 := *maintain*.

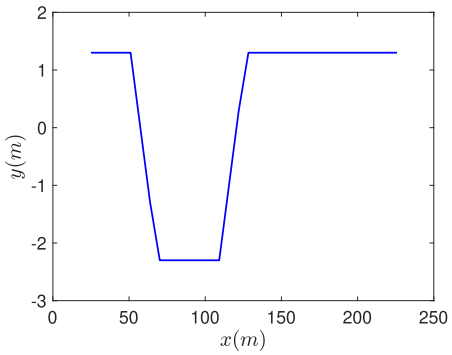

FIGURE 5. The longitudinal and lateral (*x* and *y*) positions of the ego vehicle in Scenario 1.

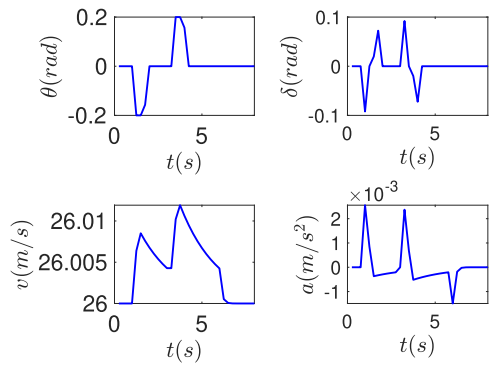

FIGURE 6. The states and control inputs of the ego vehicle in Scenario 1.

Fig. 4 illustrates that, at the beginning, the ego vehicle is following lane (i.e., $a(t = 0s) = 8(maintain)$, $s(t = 0s) = 1(Lane\ following)$). After 1(s), the ego vehicle decides to start to change to the opposite lane to overtake the parked leading vehicle (i.e., $a(t = 1s) = 4(initialize)$, $s(t = 1\text{s}) = 3(\text{Overtaking})$) since the sensors detect that the opposite lane is free and no oncoming vehicles (see Fig. 7). From Fig. 4 we know that the "maintain" action maintains for 2s. At $t = 4s$, the distance between the leading vehicle and the ego vehicle is greater than d_{safe}. Therefore, the high-level

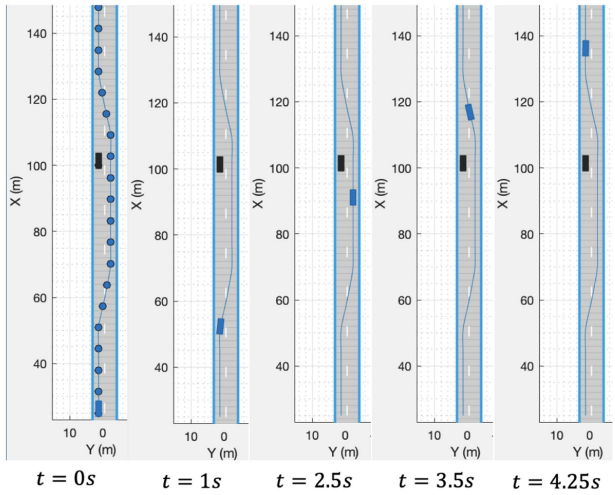

FIGURE 7. A series of screenshots for the overtaking process in Scenario 1. The leading and the ego vehicle are represented by the black and the blue rectangular, respectively.

controller updates its decision, directing the ego vehicle to return to its original lane (i.e., $a(t = 4\text{s}) = 5(\text{recover})$, $s(t = 4\text{s}) = 1(\text{Lane following})$), and then track the reference signal (v_c, y_{lc}) of the *left lane*. A series of screenshots of the overtaking process can be seen in Fig. 7. A video demonstration of this case is available at https://youtu.be/eiZ5rHkkzI8. Overall, the whole process can be summarized as:

$$1 \xrightarrow{8} \cdots \to 1 \xrightarrow{4} 3 \xrightarrow{8} \cdots \to 3 \xrightarrow{5} 1 \xrightarrow{8} \cdots \to 1.$$

C. SCENARIO 2: MANOEUVRING THROUGH PATIENCE WITH STOP AND WAIT STRATEGY

In contrast to Scenario 1, this scenario involves two oncoming vehicles moving at a constant speed on the opposite lane. The initial states of oncoming vehicles are $\boldsymbol{x}_{O_1}(0) = [174\,\text{m}, -24\,\text{m/s}, -2.3\,\text{m}]$ and $\boldsymbol{x}_{O_2}(0) = [155\,\text{m}, -24\,\text{m/s}, -2.3\,\text{m}]$. In this case, the ego vehicle not only needs to slow down, but also has to come to a complete stop and wait until both oncoming vehicles pass by.

The overall simulation time is 8 s, and the simulation results are shown in Figs. 8–11.

From Fig. 8 we observe that at the beginning the ego vehicle is following lane. After 1s, the sensors detect that the opposite lane has been occupied by two oncoming vehicles which are close to the ego vehicle (see Fig. 11). Thus, due to the safety constraint (6) the high-level generates an optimal command to ego vehicle to slow down (i.e., $a(t = 1s) = 6(prepare)$, $s(t = 1s) = 2(Waiting)$) and wait for future gaps. Also, from Fig. 10 we can see that the speed of the ego vehicle v smoothly decreases to zero. The ego vehicle remains in the "Waiting" state (which includes slowing down or stopping) for 3s. Once the opposite lane becomes clear, the high-level controller updates its decision, allowing the ego vehicle to initiate the overtaking maneuver (i.e., $a(t = 4\text{s}) = 4(\text{initialize})$, $s(t = 4\text{s}) = 3(\text{Overtaking})$), and the ego vehicle accelerates,

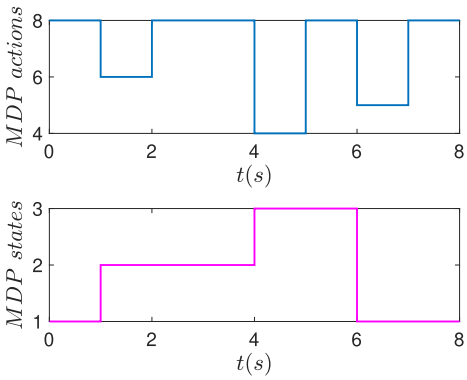

FIGURE 8. MDP state transition in Scenario 2. 1 := *Lane following*, 2 := *Waiting*, 3 := *Overtaking*, 4 := *abandon*, 5 := *recover*, 6 := *prepare*, 8 := *maintain*.

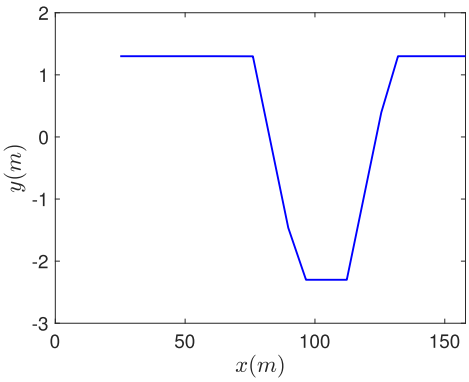

FIGURE 9. The longitudinal and lateral (*x* and *y*) positions of the ego vehicle in Scenario 2.

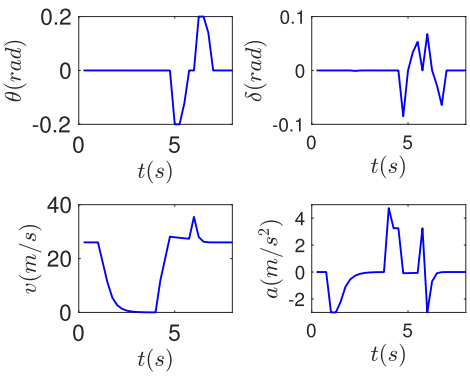

FIGURE 10. The states and control inputs of the ego vehicle in Scenario 2.

overtakes the stationary leading vehicle, seen Fig. 10. The overtaking process takes 2*s*. Then because the distance between the leading vehicle and the ego vehicle is greater than d_{safe}, the decision-maker generates a new command directing the ego vehicle to return to its initial lane (i.e., $a(t = 6\text{s}) = 5(\text{recover})$, $s(t = 6\text{s}) = 1(\text{Lane following})$). A series of screenshots of the overtaking process can be seen in Fig. 11. A video demonstration of this scenario is available at

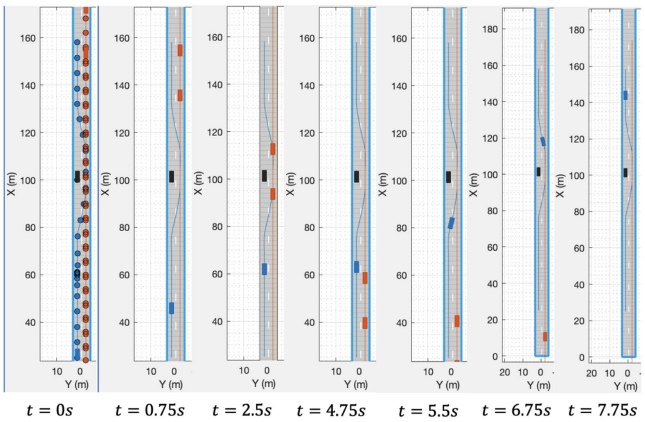

FIGURE 11. A series of screenshots for the overtaking process in Scenario 2. The leading, the oncoming vehicles and the ego vehicle are represented by the black, the orange and the blue rectangular, respectively.

https://youtu.be/7AKmpvPjEek. Overall, the whole process in Scenario 2 can be summarized as:

$$1 \xrightarrow{8} \cdots \to 1 \xrightarrow{6} 2 \xrightarrow{8} \cdots \to 2 \xrightarrow{4} 3 \xrightarrow{8} \cdots \to 3 \xrightarrow{5} 1 \xrightarrow{8} \cdots \to 1.$$

D. SCENARIO 3: FOLLOWING THE LEADING VEHICLE AS A PRIMARY APPROACH IN RESTRICTED LANE

Different from the above two scenarios, this case involves a slowly moving leading vehicle travelling at a constant speed. Additionally, three oncoming vehicles are involved in the opposite lane at a constant speed. The initial state of leading vehicle is $\boldsymbol{x}_{l_d}(0) = [100\,\text{m}, 15\,\text{m/s}, 1.3\,\text{m}]$, and the initial states of oncoming vehicles are $\boldsymbol{x}_{O_1}(0) = [300\,\text{m}, -24\,\text{m/s}, -2.3\,\text{m}]$, $\boldsymbol{x}_{O_2}(0) = [350\,\text{m}, -24\,\text{m/s}, -2.3\,\text{m}]$ and $\boldsymbol{x}_{O_3}(0) = [400\,\text{m}, -24\,\text{m/s}, -2.3\,\text{m}]$. This scenario illustrates the ego vehicle's ability to autonomously slow down and follow a slower leading vehicle when the opposite lane is blocked, patiently waiting for a safe opportunity to perform the overtaking maneuver.

The total simulation time is 17 s, and the simulation results are presented in Figs. 12–14. From Figs. 12 and 14, one can see that the ego vehicle is driving on the *left lane*, but its sensors detect the presence of a slowly moving leading vehicle ahead of it, while the opposite lane is unoccupied. Therefore, the ego vehicle decides to initialize overtaking (i.e., $a(t = 1\,\text{s}) = 4(\text{initialize})$, $s(t = 1\text{s}) = 3(\text{Overtaking})$). However, 3 s later, the sensors of the ego vehicle detect that there are oncoming vehicles in front of it and they are getting closer to it. Thus, to avoid collision (i.e., to satisfy safety constraint (6)), a new decision is made at the high level, instructing the ego vehicle to merge back into its original lane and wait for future gaps at $t = 4$ s. Also, from Fig. 15 we can see that the ego vehicle slows down and follows the speed of the leading vehicle for 5 s until the opposite lane is free when it restarts overtaking at $t = 9$s. Then, when the safety distance between the leading vehicle and the ego vehicle is satisfied, it returns to the original

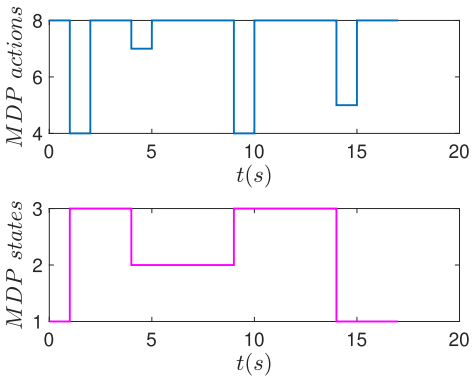

FIGURE 12. MDP state transition in Scenario 3. 1 := *Lane following*, 2 := *Waiting*, 3 := *Overtaking*, 4 := *abandon*, 5 := *recover*, 7 := *abandon*, 8 := *maintain*.

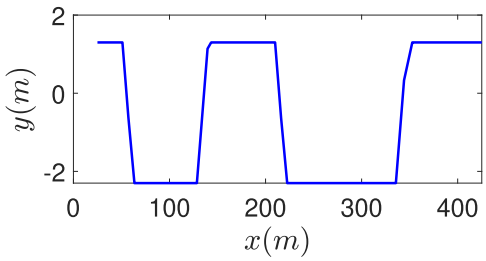

FIGURE 13. The longitudinal and lateral (*x* and *y*) positions of the ego vehicle in Scenario 3.

lane at $t = 14$s. A series of screenshots of the overtaking process can be seen in Fig. 14. A video demonstration of this case is available at https://youtu.be/M8aTbIHTHVs.

Overall, the whole process in Scenario 3 can be summarized as:

$$1 \xrightarrow{8} \cdots \to 1 \xrightarrow{4} 3 \xrightarrow{8} \cdots \to 3 \xrightarrow{7} 2 \xrightarrow{8} \cdots \to 2 \xrightarrow{4} 3 \xrightarrow{8} \cdots \to 3 \xrightarrow{5} 1 \xrightarrow{8} \cdots \to 1.$$

E. SCENARIO 4: DYNAMIC ENVIRONMENT CHALLENGES WITH SUDDEN ONCOMING VEHICLES ACCELERATION

In contrast to Scenario 3, this scenario involves the first two oncoming vehicles moving at a constant speed, while the third oncoming vehicle accelerates at a variable rate. The initial states of the leading vehicle and the first two oncoming vehicles are the same as $\boldsymbol{x}_{l_d}(0), \boldsymbol{x}_{O_1}(0), \boldsymbol{x}_{O_2}(0)$ given in Scenario 3. The settings for the third oncoming vehicle are given as $\boldsymbol{x}_{O_3}(0) = [528\text{m}, -4\text{m/s}, -2.3\text{m}]$. It first proceeds at constant speed -4m/s for 10s, followed by an acceleration of -1.5m/s^2 for 1.5s. Then it speeds up further with $a = -2\text{m/s}^2$ for 1s. After finishing this acceleration process, it moves forward at a constant speed of -24m/s. The changes in the speed for the whole process are shown in Fig. 16.

This scenario is designed to demonstrate that the novel MDP solution can handle emergency and unexpected situations effectively in dynamic and uncertain traffic conditions. The overall simulation time is 20s, and the simulation results are shown in Figs. 17–20.

In the initial 4s, the situation is the same as Scenario 3 (see Figs. 17 and 20). From Fig. 19 we can see that the ego vehicle first starts slowing down and then follows the leading vehicle for about 5s. That means 5s later, the opposite lane is available to overtake even though there is a slow oncoming vehicle, which it is far from the ego vehicle, so the ego vehicle restarts overtaking at $t = 9$s. Unexpectedly, the oncoming vehicle suddenly accelerates while the ego vehicle is overtaking on the opposite lane, creating a safety risk. Hence, the ego vehicle decides to change back to its original lane and continue following the leading vehicle. When $t = 12$s, the opposite lane is re-available and the decision is to re-initiate overtaking. Fig. 19 shows that the ego vehicle accelerates to finish overtaking quickly, and then returns back to the original lane at $t = 17$s. A series of screenshots of the overtaking process can be seen in Fig. 20. A video demonstration of this scenario is available at https://youtu.be/meX6Vym3TzM. The whole process in Scenario 4 can be summarized as:

$$1 \xrightarrow{8} \cdots \to 1 \xrightarrow{4} 3 \xrightarrow{8} \cdots \to 3 \xrightarrow{7} 2 \xrightarrow{8} \cdots \to 2 \xrightarrow{4} 3 \xrightarrow{7} 2 \xrightarrow{8} 2 \xrightarrow{4} 3 \xrightarrow{8} \cdots \to 3 \xrightarrow{5} 1 \xrightarrow{8} \cdots \to 1.$$

Remark 8: The computational time of our algorithm scales linearly with the prediction horizon N, as the Algorithm 1 requires solving N sequential constrained optimizations. For $N \in \{5, 7, 10\}$, the maximum computational time for optimal decision-making at each time step on an Intel i5-118G7 CPU is $\{0.4, 0.7, 1\}$ seconds, below the high-level control system's sampling time $T_s = 1$s. The complexity is $O\left(N \cdot n^3\right)$ where n is the state dimension, dominated by QP solves in MPC. This computational duration, being significantly less than the allocated sampling interval, convincingly demonstrates the real-time feasibility and practical applicability of our proposed algorithm in time-critical control scenarios.

To further highlight the advantages of the proposed MDP-based framework, we next consider a scenario in which vehicles in both lanes travel in the same direction. Comparisons with the rule-based decision-making framework presented in [42] will also be provided.

F. SCENARIO 5: OVERTAKING IN SAME DIRECTION

We consider the scenario illustrated in Fig. 21. In this scenario, a slow-moving leading vehicle (black), travelling at 15m/s, is positioned 40m ahead of the ego vehicle (blue). An orange vehicle in the adjacent lane is located 5m behind the ego vehicle, travelling at a speed of 30m/s, while the ego vehicle is moving at 26m/s. Additionally, a grey vehicle starts from the position $(-15, -60)$m and attempts to merge onto the main road from a side road at an angle of 29 degrees. We assume that its speed profile is given in Fig. 22.

This scenario is designed to demonstrate how different decision-making frameworks generate varying decisions for

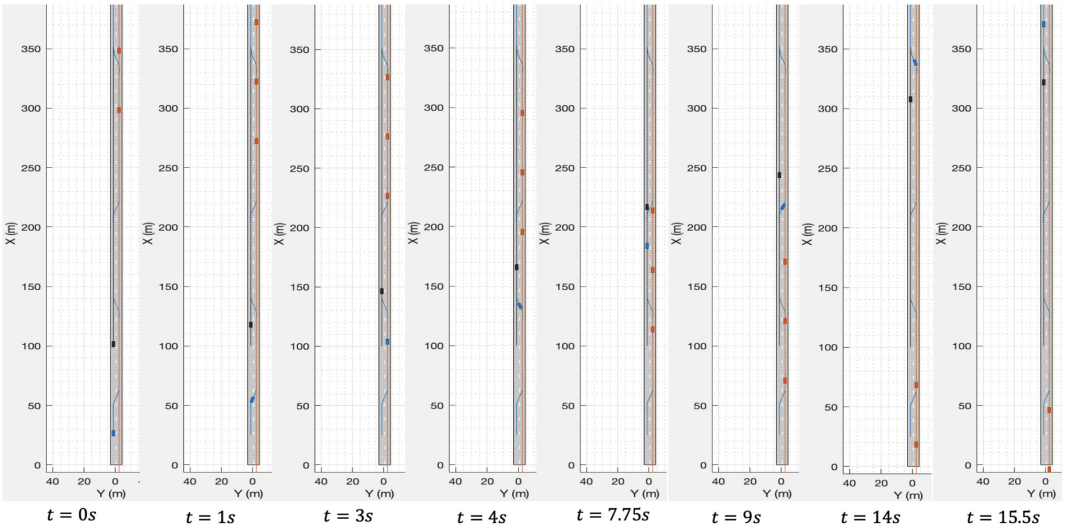

FIGURE 14. A series of screenshots for the overtaking process in Scenario 3. The leading, the oncoming vehicles and the ego vehicle are represented by the black, the orange and the blue rectangular, respectively.

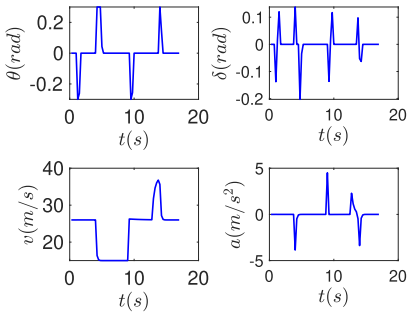

FIGURE 15. The states and control inputs of the ego vehicle in Scenario 3.

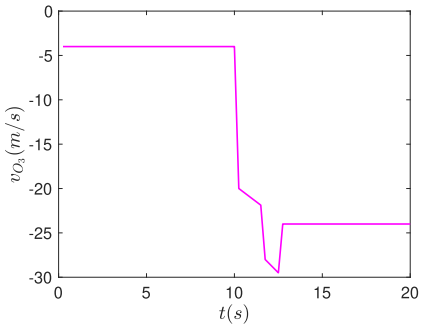

FIGURE 16. The changes in the speed of the third oncoming vehicle.

the same case, ensuring safety. The total simulation time is 11s, and the results are presented in Figs. 23–26.

The sensors of the ego vehicle detect a slow-moving vehicle ahead, while the adjacent lane is currently unavailable for a lane-change maneuver, as shown in Fig. 21. From speed changes shown in Fig. 24, it can be observed that the ego vehicle initially decelerates, then follows the leading vehicle while waiting for a sufficient safety gap to initiate a lane

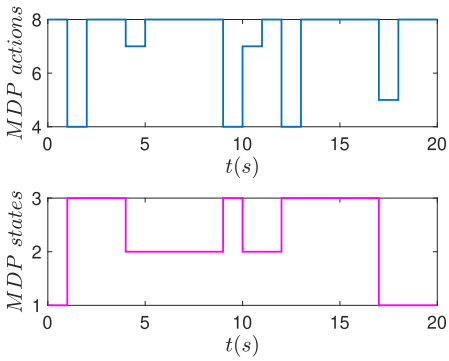

FIGURE 17. MDP state transition in Scenario 4. 1 := *Lane following*, 2 := *Waiting*, 3 := *Overtaking*, 4 := *abandon*, 5 := *recover*, 7 := *abandon*, 8 := *maintain*.

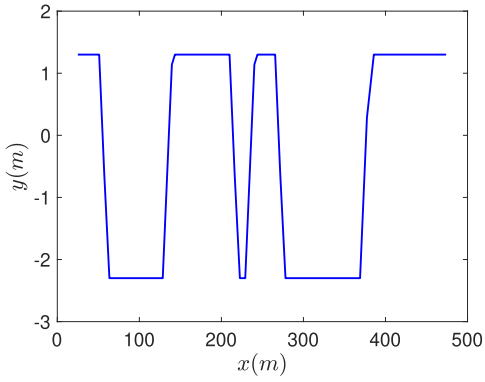

FIGURE 18. The longitudinal and lateral (x and y) positions of the ego vehicle in Scenario 4.

change. In contrast, when using the decision-making logic diagram presented in Fig. 5 of [42], the ego vehicle first attempts to create a safe gap by accelerating if the adjacent lane is not immediately available. Once the required safety

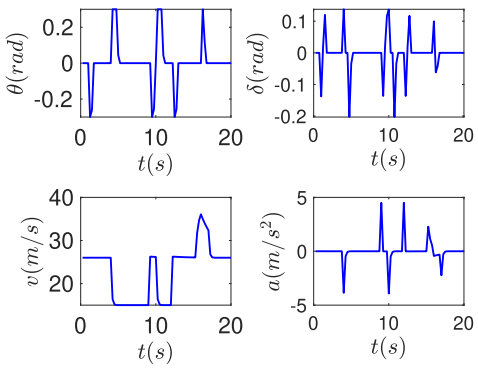

FIGURE 19. The states and control inputs of the ego vehicle in Scenario 4.

spacing is achieved, it promptly initiates a lane change. If the safety condition is not met, the vehicle continues to follow the leading vehicle in compliance with safety constraints. Then, based on the changes in the y-direction shown in Fig. 24, it can be observed that approximately 2 s later, the ego vehicle performs a lane change to initiate an overtaking maneuver.

Unexpectedly, around $t = 3s$, the sensors detect a faster vehicle merging from the side road. To ensure safety, the proposed MDP-based framework generates a "Waiting" command, prompting the ego vehicle to take an "abandon" action and "return" to its original lane and continuing to follow the leading vehicle. In contrast, as shown in Figs. 24 and 25, the rule-based framework generates an "Accelerating" command, causing the ego vehicle to "remain" in its current lane and speed up in an attempt to create a safe gap with the merging vehicle. However, the merging vehicle is moving too quickly, forcing the ego vehicle to accelerate to avoid a collision, as depicted in Figs. 24 and 25. Even though the rule-based method generates an "Accelerating" command, compared the time to collision (TTC) under these two frameworks, Fig. 26 demonstrates that the TTC obtained from the MDP-based approach is significantly larger than that obtained from the rule-based method. From Fig. 26 we can see that the minimum TTC under the rule-based framework is 0.04 s while it is 5 s under the MDP-based framework. Such an increase in minimum TTC significantly enhances safety in real-world implementations.

Additionally, in terms of passenger comfort, according to the following equation

$$Improvement = \frac{a_{ms}^{Rule} - a_{ms}^{MDP}}{a_{ms}^{Rule}} \times 100\%$$

where a_{ms} denotes the mean square of the acceleration profile, the proposed MDP-based approach demonstrates a 16% improvement.

A video demonstration of the MDP-based approach for this scenario is available at {https://youtu.be/vssat0SvHR0}, and the corresponding rule-based demonstration can be found at {https://youtu.be/Y2yB2vpseac}.

VII. DISCUSSIONS

Compared with the existing methods such as [3], [15], [35], [43], our proposed solution, based on safety-constrained MDP (see safety hard constraints (6)–(7) given in Algorithm 1), can effectively ensure safety requirements in highly dynamic and uncertain environments on two lanes in opposite directions (see *Scenario 4*). This approach enables the specification of constraints that ensure a minimum safe distance between the ego vehicle and other road users, thereby improving safety margins. In addition, compared with the rule-based framework presented in [42], Scenario 5 demonstrates that the proposed MDP-based framework not only effectively ensures safety but also enhances passenger comfort on two lanes in the same direction.

The decisions generated by MDP at the high level can be used to guide the objectives of MPC at the low level, thereby decomposing the computational task into smaller, more manageable sub-tasks. This combination allows for leveraging the fast response capability of MPC, reducing the computation time required for each decision-making process. Moreover, in an MPC-based solution to the MDP problem, using a simplified model instead of a complex physical model can reduce the computational burden of each optimization while maintaining sufficient accuracy. Additionally, Assumption 1 restricts the analysis to finite nearby vehicles, but this is consistent with the limited perception range and computational horizon of practical systems. Extensions to unbounded vehicle sets would require probabilistic safety bounds, which we leave for future work.

While we assume perfect knowledge of surrounding vehicles' states for theoretical analysis (similar to [15], [33]), real-world implementations would integrate sensor fusion (e.g., LiDAR, camera, V2X) with noise filtering. Our safety constraints (4d)–(4e) inherently provide robustness to bounded uncertainties, as the elliptical safe margin can be enlarged to account for perception errors. For example, if the position error of surrounding vehicles is within ϵ, the safety constraint (4d) can be modified to $(\frac{x_e(k)-x_j(k)}{\Delta x_j+\epsilon})^2 + (\frac{y_e(k)-y_j(k)}{\Delta y_j+\epsilon})^2 \geq 1$. Moreover, the perfect tracking assumption simplifies the high-level analysis, but in practice, tracking errors can be handled by robust low-level control designs (e.g., MPC, PID) [25], [30].

Our current action set (e.g., initialize, recover, prepare) is specifically designed to model the key decision states involved in overtaking maneuvers, rather than general driving behaviour. These high-level actions capture the key stages of overtaking, including initiation, preparation, execution, and recovery, and are sufficient for managing the complexity of this task within our hierarchical control framework. Although the action space may appear simplistic for full real-world driving, it is intentionally scoped to overtaking scenarios. More task-general action spaces could certainly be explored in future work, especially when expanding to broader driving behaviors (e.g., merging, intersection handling).

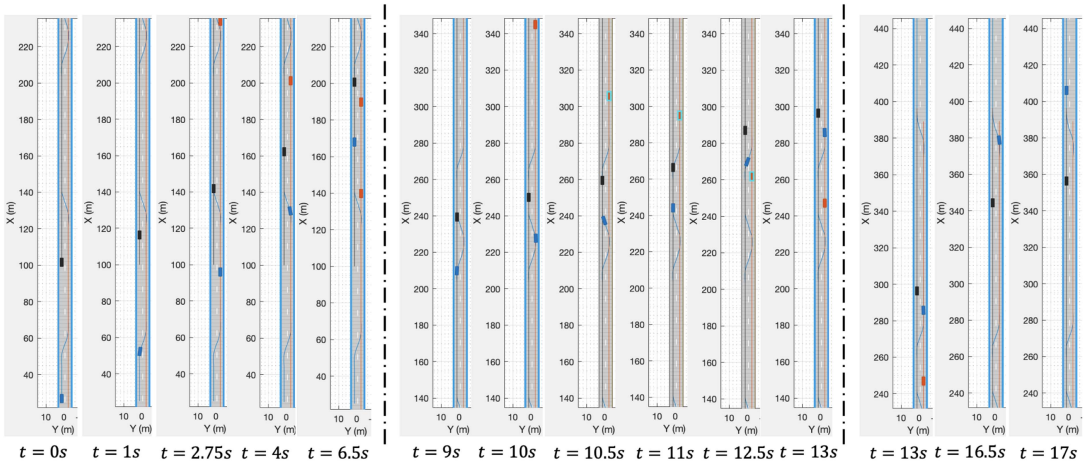

FIGURE 20. A series of screenshots for the overtaking process in Scenario 4. The leading, the oncoming vehicles and the ego vehicle are represented by the black, the orange and the blue rectangular, respectively.

FIGURE 21. Vehicles in two lanes travel in the same direction. The speed limit is 70 mph. The desired speed for the left lane is $v_c = 26m/s$ and $v_{rc} = 30m/s$ for the right lane. .

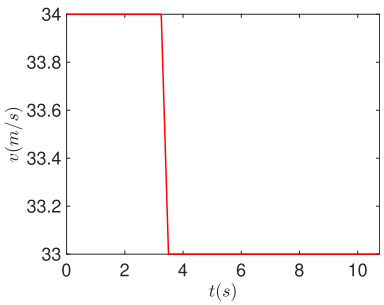

FIGURE 22. Speed of merging vehicle from side road.

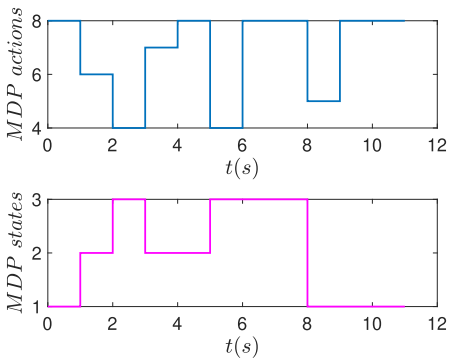

FIGURE 23. MDP state transition in Scenario 5. 1 := *Lane following*, 2 := *Waiting*, 3 := *Overtaking*, 4 := *abandon*, 5 := *recover*, 6 := *prepare*, 7 := *abandon*, 8 := *maintain*.

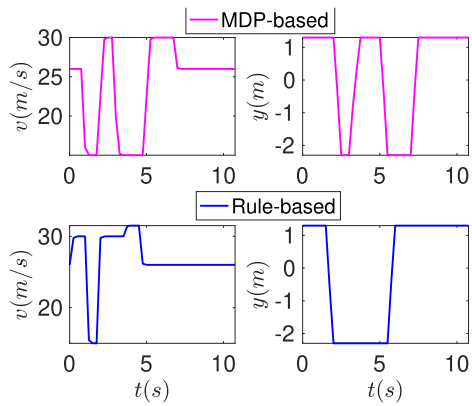

FIGURE 24. Comparisons of speeds and lane change under MDP-based and rule-based decision-making frameworks.

The four preset speed levels $(v_c, v_s, v_{rc}, 0)$ in our high-level MDP are deliberately designed to balance optimality and real-time safety. Within the hierarchical control framework, the MDP governs macro-level decisions (e.g., lane changes or overtaking initiation), where coarse speed discretization is sufficient to encode safe maneuver choices. Fine-grained, continuous speed adjustments are delegated to the low-level controller through dynamic tracking and constraint enforcement. This separation ensures that speed discretization does not compromise system-level optimality, as the low-level controller compensates for any quantization errors. Moreover, these speed levels are intrinsically aligned with safety constraints: v_c and v_{rc} comply with traffic rules (e.g., speed limits), whereas v_s and 0 serve as emergency fallbacks to ensure safety (i.e., d_{safe}) in critical situations (see (4d)–(4e)).

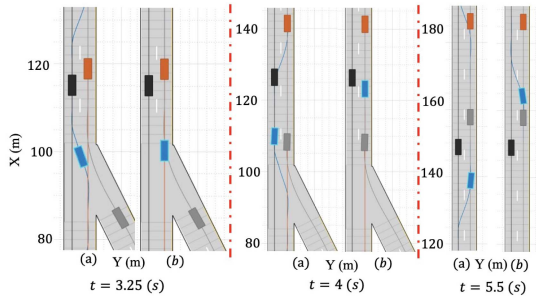

FIGURE 25. Screenshots for overtaking process under two frameworks. (a) the proposed MDP-based method, and (b) the rule-based method in [42].

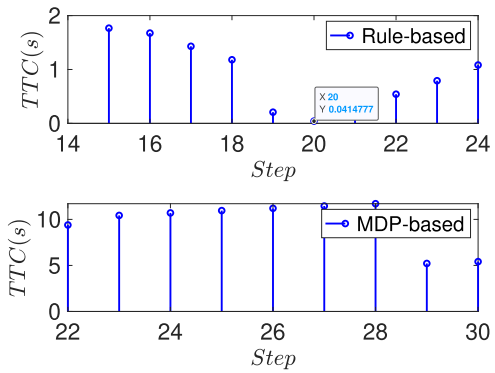

FIGURE 26. Time to collision (TTC) under the proposed MDP-based method, and the rule-based method in [42]. $TTC = \frac{Relative\ Distance}{Relative\ Speed}$.

This discretization reflects a principled trade-off—simplifying high-level decision-making without compromising safety or practical feasibility, as evidenced by the collision-free results across all tested scenarios (Section VI).

VIII. CONCLUSION

This paper presents a novel high-level MDP solution for making safe and optimal decisions in autonomous overtaking scenarios, particularly in dynamic environments with oncoming vehicles. By carefully formulating the top-level decision-making process within a hierarchical control structure as an MDP control problem, we have designed an MPC scheme that optimises overtaking decisions while ensuring safety. The link between high-level decision-making and lower-level behaviours and the status of the ego vehicle is crucial for ensuring accurate decision-making. We abstract the behaviour of the ego vehicle and surrounding vehicles, integrating these elements into the MDP-based decision-making framework. To ensure the recursive feasibility and stability of the proposed solution, we introduced a feasible baseline control policy to calculate the terminal cost that is involved in online solving the optimal MDP problem.

The performance of the new design is evaluated on the MATLAB platform with Yalmip solver using five driving scenarios on two lanes no matter of opposite directions or in the same direction. The testing results indicate that the developed MDP solution is able to make rational decisions under dynamic and unknown driving environments. While this work focuses on overtaking actions, the proposed framework can be extended to other driving scenarios by augmenting the action set. For instance, intersection navigation: Add actions for traffic light compliance (e.g., a_6: stop at the red light). The safety constraints (4d)–(4e) would similarly apply to these new actions, ensuring unified risk-aware decision-making. In future work, we plan to enhance robustness under imperfect perception, including techniques for sensor fusion, noise filtering, and uncertainty propagation within the decision layer.

REFERENCES

[1] R. Bellman, "A Markovian decision process," *J. Math. Mechan.*, vol. 6, no. 5, pp. 679–684, 1957.
[2] D. Bertsekas, *Reinforcement Learning and Optimal Control*. Belmont, MA, USA: Athena Scientific, 2019.
[3] L. Buşoniu, T. D. Bruin, D. Tolić, J. Kober, and I. Palunko, "Reinforcement learning for control: Performance, stability, and deep approximators," *Annu. Rev. Control*, vol. 46, pp. 8–28, 2018.
[4] R. Chai, A. Tsourdos, S. Chai, Y. Xia, A. Savvaris, and C. L. P. Chen, "Multiphase overtaking maneuver planning for autonomous ground vehicles via a desensitized trajectory optimization approach," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 74–87, Jan. 2023.
[5] W. H. Chen, "Perspective view of autonomous control in unknown environment: Dual control for exploitation and exploration vs reinforcement learning," *Neurocomputing*, vol. 497, pp. 50–63, 2022.
[6] W. H. Chen, "Goal-oriented Control Systems (GOCS): From how to what," *IEEE/CAA J. Automatica Sinica*, vol. 11, no. 4, pp. 816–819, Apr. 2024.
[7] Y. Chen, S. Li, X. Tang, K. Yang, D. Cao, and X. Lin, "Interaction-aware decision making for autonomous vehicles," *IEEE Trans. Transport. Electrific.*, vol. 9, no. 3, pp. 4704–4715, Sep. 2023.
[8] S. Coskun, Q. Zhang, and R. Langari, "Receding horizon Markov game autonomous driving strategy," in *Proc. Amer. Control Conf.*, 2019, pp. 1367–1374.
[9] S. Dixit et al., "Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects," *Annu. Rev. Control*, vol. 45, pp. 76–86, 2018.
[10] E. Dogan, E. Yousfi, T. Bellet, C. Tijus, and A. Guillaume, "Manual takeover after highly automated driving: Influence of budget time and lane change assist on takeover performance," in *Proc. 32nd Eur. Conf. Cogn. Ergonom.*, 2021, pp. 1–6 .
[11] B. B. Elallid, N. Benamar, A. S. Hafid, T. Rachidi, and N. Mrani, "A comprehensive survey on the application of deep and reinforcement learning approaches in autonomous driving," *J. King Saud University-Computer Inform. Sci.*, vol. 34, no. 9, pp. 7366–7390, 2022.
[12] B. Fan, H. Yuan, Y. Dong, Z. Zhu, and H. Liu, "Bidirectional agent-map interaction feature learning leveraged by map-related tasks for trajectory prediction in autonomous driving," *IEEE Trans. Automat. Sci. Eng.*, vol. 22, pp. 10801–10813, 2025.
[13] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 1437–1480, 2015.
[14] C. Gehring and D. Precup, "Smart exploration in reinforcement learning using absolute temporal difference errors," in *Proc. Int. Conf. Auton. Agents Multi-agent Syst.*, 2013, pp. 1037–1044 .
[15] Y. Guan, S. E. Li, J. Duan, W. Wang, and B. Cheng, "Markov probabilistic decision making of self-driving cars in highway with random traffic flow: A simulation study," *J. Intell. Connected Veh.*, vol. 1, no. 2, pp. 77–84, 2018.
[16] H. Andersen et al., "Trajectory optimization for autonomous overtaking with visibility maximization," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 1–8.
[17] P. Hang, C. Lv, Y. Xing, C. Huang, and Z. Hu, "Human-like decision making for autonomous driving: A noncooperative game theoretic approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2076–2087, Apr. 2021.

[18] X. Huang, W. Zhang, and P. Li, "A path planning method for vehicle overtaking maneuver using sigmoid functions," *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 422–427, 2019.
[19] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 952–964, Feb. 2017.
[20] Y. Ji, L. Ni, C. Zhao, C. Lei, Y. Du, and W. Wang, "Tripfield: A 3 d potential field model and its applications to local path planning of autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3541–3554, Mar. 2023.
[21] K. Kim and P. R. Kumar, "An MPC-based approach to provable system-wide safety and liveness of autonomous ground traffic," *IEEE Trans. Autom. Control*, vol. 59, no. 12, pp. 3341–3356, Dec. 2014.
[22] B. R. Kiran et al., "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.
[23] T. Korssen, V. Dolk, J. V. D. Mortel-Fronczak, M. Reniers, and M. Heemels, "Systematic model-based design and implementation of supervisors for advanced driver assistance systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 533–544, Feb. 2018.
[24] E. L. Law, "Risk-directed exploration in reinforcement learning," McGill Univ., 2005.
[25] Z. Li, J. Jiang, W. H. Chen, and L. Sun, "Autonomous lateral maneuvers for self-driving vehicles in complex traffic environment," *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1900–1910, Feb. 2023.
[26] W. Liu, S. W. Kim, S. Pendleton, and M. H. Ang, "Situation-aware decision making for autonomous driving on urban road using online POMDP," in *Proc. IEEE Intell. Veh. Symp.*, 2015, pp. 1126–1133.
[27] Y. Liu, A. Zhou, Y. Wang, and S. Peeta, "Proactive longitudinal control of connected and autonomous vehicles with lane-change assistance for human-driven vehicles," in *Proc. IEEE Int. Intell. Transp. Syst. Conf.*, 2021, pp. 776–781.
[28] H. Muslim, "Design and evaluation of lane-change collision avoidance systems in semi-automated driving," *IEEE Trans. Veh. Technol.*, vol. 72, no. 6, pp. 7082–7094, Jun. 2023.
[29] M. S. Rais, R. Boudour, K. Zouaidia, and L. Bougueroua, "Decision making for autonomous vehicles in highway scenarios using harmonic sk deep sarsa," *Appl. Intell.*, vol. 53, no. 3, pp. 2488–2505, 2023.
[30] M. Samuel, M. Mohamad, M. Hussein, and S. M. Saad, "Lane keeping maneuvers using proportional integral derivative (PID) and model predictive control (MPC)," *J. Robot. Control*, vol. 2, no. 2, pp. 78–82, 2021.
[31] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 1, pp. 187–210, 2018.
[32] R. S. Sutton et al. *Introduction to Reinforcement Learning*, vol. 135. Cambridge, MA, USA: MIT Press, 1998.
[33] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Syst. Mag.*, vol. 12, no. 2, pp. 19–22, Apr. 1992.
[34] H. Taghavifar, C. Hu, C. Wei, A. Mohammadzadeh, and C. Zhang, "Behaviorally-aware multi-agent RL with dynamic optimization for autonomous driving," *IEEE Trans. Automat. Sci. Eng.*, vol. 22, pp. 10672–10683, 2025.
[35] J. J. Verbakel, M. Fusco, D. M. C. Willemsen, J. M. V. D. Mortel-Fronczak, and W. P. M. H. Heemels, "Decision making for autonomous vehicles: Combining safety and optimality," *IFAC-PapersOnLine*, vol. 53,no. 2, pp. 15380–15387, 2020.
[36] X. F. Wang, W. H. Chen, J. Jiang, and Y. Yan, "High-level decision-making for autonomous overtaking: An MPC-based switching control approach," *IET Intell. Transport Syst.*, vol. 18, no. 7, pp. 1259–1271, 2024.
[37] X. F. Wang, J. Jiang, and W. H. Chen, "High-level decision making in a hierarchical control framework: Integrating HMDP and MPC for autonomous systems," *IEEE Trans. Cybern.*, vol. 55, no. 4, pp. 1903–1916, Apr. 2025.
[38] J. Wu, H. Yang, L. Yang, Y. Huang, X. He, and C. Lv, "Human-guided deep reinforcement learning for optimal decision making of autonomous vehicles," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 54, no. 11, pp. 6595–6609, Nov. 2024.
[39] Y. Yamada, A. S. M. Bakibillah, K. Hashikura, M. A. S. Kamal, and K. Yamada, "Autonomous vehicle overtaking: Modeling and an optimal trajectory generation scheme," *Sustainability*, vol. 14, no. 3, pp. 1–14, 2022.
[40] Y. Yan, X. F. Wang, B. J. Marshall, C. Liu, J. Yang, and W. H. Chen, "Surviving disturbances: A predictive control framework with guaranteed safety," *Automatica*, vol. 158, 2023, Art. no. 111238.
[41] L. Zhang, R. Zhang, T. Wu, R. Weng, M. Han, and Y. Zhao, "Safe reinforcement learning with stability guarantee for motion planning of autonomous vehicles," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5435–5444, Dec. 2021.
[42] Z. Zhang, L. Zhang, C. Wang, M. Wang, D. Cao, and Z. Wang, "Integrated decision making and motion control for autonomous emergency avoidance based on driving primitives transition," *IEEE Trans. Veh. Technol.*, vol. 72, no. 4, pp. 4207–4221, Apr. 2023.
[43] S. Zhou, Y. Wang, M. Zheng, and M. Tomizuka, "A hierarchical planning and control framework for structured highway driving," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9101–9107, 2017.

XUE-FANG WANG (Member, IEEE) received the B.S. degree from the Ocean university of China, Qingdao college, Qingdao, China, in 2013, and the Ph.D. degree in control theory and control engineering from the Dalian University of Technology, Dalian, China, in 2019. From 2017 to 2019, she was a Visiting Scholar to work with Prof. Andrew R. Teel with the University of California, Santa Barbara, Santa Barbara, CA, USA. From 2020 to 2023, she was a Research Associate with the Dalian University of Technology, and Loughborough University, Loughborough, U.K., respectively. She joined the School of Engineering, University of Leicester, Leicester, U.K., as a Lecturer in Control Engineering in 2023. Her main research interests include multiagent systems, hybrid systems, distributed optimisation problems, autonomous vehicle control design, and model predictive control.

JINGJING JIANG (Member, IEEE) received the B.E. degrees in electrical and electronic engineering from the University of Birmingham, Birmingham, U.K., and the Harbin Institute of Technology, Harbin, China, in 2010, and the M.Sc. degree in control systems and the Ph.D. degree from Imperial College London, London, U.K., in 2011 and 2016, respectively. She carried out research as part of the Control and Power Group, Imperial College and joined Loughborough University, Loughborough, U.K., as a Lecturer in 2018. She is currently a Senior Lecturer in Intelligent Mobility and Autonomous Vehicles with the Department of Aeronautical and Automotive Engineering, Loughborough University. Her research interests include driver assistance control and autonomous vehicle control design, control design of systems with constraints, and human-in-the-loop.

WEN-HUA CHEN (Fellow, IEEE) is currently the Chair of autonomous vehicles with the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, U.K. He is the Founder and Head of the Loughborough University Centre of Autonomous Systems. He has authored or coauthored nearly 300 papers and two books. His research interests include control, signal processing, and artificial intelligence and their applications in robots, aerospace, and automotive systems. He is also with U.K. Engineering and Physical Sciences Research Council Established Career Fellowship in developing new control theory for robotics and autonomous systems. He is a Fellow of Institution of Mechanical Engineers and Institution of Engineering and Technology, Stevenage, U.K.