2024 International Annual Conference on Complex Systems and Intelligent Science (CSIS-IAC)
September 20~22, 2024, Guangzhou, China

Adaptive Horizon Model Predictive Control with Reinforcement Learning for Highway Local Ramp Metering

Bo Jin
College of Civil Engineering and Architecture
Zhejiang University
Hangzhou, China
jinbo0201@outlook.com

Jiayang Yu
Zhejiang Scientific Research Institute of Transport
Hangzhou, China
13258332059@163.com

Xinliang Fu
Zhejiang Scientific Research Institute of Transport
Hangzhou, China
xinliangf@163.com

Song Yang
Zhejiang Scientific Research Institute of Transport
Hangzhou, China
36370548@qq.com

2024 International Annual Conference on Complex Systems and Intelligent Science (CSIS-IAC) | 979-8-3315-0475-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/CSIS-IAC63491.2024.10919369

***Abstract*—Model predictive control is a widely used approach to the problem of highway local ramp metering problem. The main parameter that affects the computational complexity is the prediction horizon. Considering the performance sensitivity to the prediction horizon length, this paper proposes a reinforcement learning (RL) based adaptive horizon model predictive control (AHMPC). In the RL-AHMPC, the prediction horizon is adjusted based on an RL agent according to the system states. The RL agent learns the adaptive horizon policy by continuously interacting with the environment. Meanwhile, an effective MPC controller is designed to handle the constraints and objectives in the ramp metering problem, and provide optimal control sequences. Simulation results show that the novel RL-AHMPC method can improve traffic efficiency and keep the computational cost at a low level. This is due to the combination of the optimal control sequences from the MPC controller and the intelligent adaptive horizon generated by the RL agent.**

***Index Terms*—active traffic control, data-driven control, learning-based control, reinforcement learning**

I. Introduction

In recent years, there has been a significant escalation in the demand for transport mobility, with beneficial implications for societal progress, but also with many adverse consequences for drivers and traffic controllers. The academic and industrial communities have extensively investigated traffic control methodologies with the aim of improving traffic safety and efficiency. While the field of traffic control encompasses both urban and highway networks, this paper focuses specifically on highway traffic flow control.

To achieve safety and efficiency goals, the highway traffic control problem is constructed as an optimization model, and the optimal strategies are obtained by solving the model. The model-based optimal control method, MPC, has been widely applied to the highway traffic flow control optimization problem [1]. Meanwhile, the learning-based optimal control method, RL, is a recent technique that has shown its success and potential in the field of control, including highway traffic control [2]. Both MPC and RL methods have their advantages and disadvantages when dealing with optimal control problems. The MPC method requires a significant amount of computational resources during real-time control. Conversely, the RL method has an innate ability to handle complicated challenges with minimal online computational overhead. Nevertheless, the process of training a proficient RL agent is typically time consuming, especially for complex systems.

Addressing the above challenges in MPC and RL methods for highway traffic flow control, this paper proposed an RL-based MPC method. In the proposed method, MPC is responsible for handling complex constraints and objectives. While, RL is responsible for outputting dynamic prediction horizons. Thanks to the optimized prediction horizons, the MPC controller can complete the entire control cycle task with lower computational costs.

A. Related Work

This section gives an overview of related work that applies MPC, RL, and MPC with RL methods to solve highway traffic flow control problems.

MPC is a widely recognised method for real-time control of dynamic systems. It works by predicting system states over a finite time horizon and optimising an appropriate objective function. This is achieved by iteratively solving a Finite-Horizon Optimal Control Problem that is updated with real-time system states. Groot et al. [3] used MPC to solve a highway traffic flow control optimization problem integrated the METANET model. A piecewise-affine (PWA) approximation

This work is supported by the Independent Research Project of Zhejiang Scientific Research Institute of Transport under grant no. ZK202411, the Science and Technology Plan Projects of Zhejiang Provincial Department of Transport under grant no. 2023013.

of the nonlinear METANET model was proposed to facilitate real-time implementation. Paula et al. [4] applied MPC to highway traffic networks, where the goal was reducing the time spent by the drivers through a dynamic setting of variable speed limit (VSL) and ramp metering (RM). Todorovic et al. [5] proposed a distributed MPC algorithm to coordinated control of discrete VSLs and continuous RM. The proposed algorithm used a distributed control architecture and an alternating optimization scheme to balance the computational complexity and system performance.

RL-based AI has made remarkable progress in outperforming top human professionals in complex multiplayer games. These achievements are a powerful demonstration of the immense potential of RL methods. Taking traffic flow simulations as training environments, Wang et al. [6] introduced actor-critic-based RL methods to learn actions, with the reward function taking into account the waiting time, average speed and on-ramp queuing limit. Zheng et al. [7] introduced a multi-agent RL-based VSL approach to enhance collaboration among VSL controllers. The proposed approach used centralised training with a decentralised execution structure to achieve a joint optimal solution for a set of VSL controllers. To improve the traffic safety and efficiency of freeway tunnels, Jin et al. [8] proposed a novel VSL control strategy based on the RL framework. The VSL control agent was trained using a deep dyna-Q method.

Combining RL and MPC can fully leverage the advantages of model-based and learning-based strategies in control problems. Chen et al. [9] proposed a stochastic MPC method based on RL for energy management of plug-in hybrid electric vehicles. The RL controller was embedded into the stochastic MPC controller to determine the optimal strategy at each step. Flessner et al. [10] utilized RL to determine the triggering mechanism of the MPC controller, thereby balancing computational complexity and control effectiveness. Bøhn et al. [11] proposed to learn the optimal prediction horizon as a function of the state using RL. The results showed that clear improvements over the fixed horizon MPC scheme with less training time. There have been a few studies on this topic and the application of RL-MPC methods in the field of highway traffic control. Airaldi et al. [12] used RL to adjust the parameterisation of the MPC based on observed data, improving the accuracy of the METANET model to improve the closed-loop performance. Sun et al. [13] proposed a hierarchical structure combining RL and MPC, in which a high-level MPC component provided a baseline control input, while a low-level RL component modified the output generated by MPC.

B. Proposed Approach and Contributions

The method of combining MPC and RL for the problem of motorway traffic control has been studied by only a few researchers. The contribution of this paper is to propose an RL-based AHMPC (RL-AHMPC) for the highway local ramp metering, as shown in Fig. 1, in order to incorporate the advantages of both MPC and RL methods. In particular, an efficient MPC controller is designed to handle the constraints and objectives in the ramp metering problem, and provide optimal control sequences. Meanwhile, an RL agent in the control structure of AHMPC provides an optimal prediction horizon to balance the computational cost and control effectiveness.

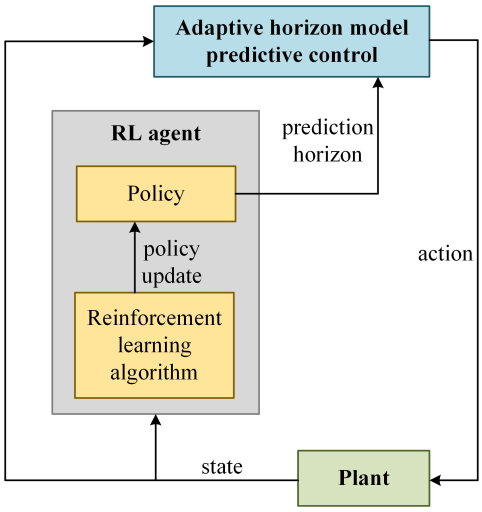

Fig. 1: The framework of RL-AHMPC

The structure of this paper is as follows: Section II presents the METANET model and the associated MPC algorithm for the local ramp metering problem. Section III discusses the details of the novel RL-AHMPC method. Section IV gives numerical case studies implementing the proposed RL-AHMPC method on a highway section. Finally, Section V concludes this paper and suggests topics for future work.

II. Problem Formulation

A. Defnition of Symbols

For a better understanding of this paper, we defne the necessary notations and parameters in Table I.

B. METANET modelling

We adopt the macroscopic second-order METANET framework to formulate a discrete-time dynamical representation of the highway traffic under local ramp metering [14]. For the METANET framework, each segment i, $i \in I_{all}$, at discrete time $t = kT$ is characterized by three state variables $\rho_i(k)$, $v_i(k)$ and $q_i(k)$. These traffic variables can be calculated by the following equations:

$$\rho_i(k+1) = \rho_i(k) + \frac{T}{\lambda_i L_i}(q_{i-1}(k) - q_i(k) + r_i(k)) \tag{1}$$

$$q_i(k) = \lambda_i \rho_i(k) v_i(k) \tag{2}$$

$$\begin{aligned} v_i(k+1) = v_i(k) &+ \frac{T}{\tau}(V(\rho_i(k)) - v_i(k)) \\ &+ \frac{T}{L_i} v_i(k)(v_{i-1}(k) - v_i(k)) \\ &- \frac{\nu T}{\tau L_i} \frac{\rho_{i+1}(k) - \rho_i(k)}{\rho_i(k) + \kappa} \\ &- \frac{\mu T}{L_i \lambda_i} \frac{v_i(k)}{\rho_i(k) + \kappa} r_i(k) \end{aligned} \tag{3}$$

TABLE I: Notations and parameters

Index	Description
i	Index of segment
k	Index of time step
k_c	Index of control time step
Parameters	**Description**
T	Length of time step
L_i	Length of segemnt i
λ_i	Number of lanes in segment i
τ, ν, κ, μ, α	Parameters of METANET
v_{free}	Free speed
ρ_{crit}	Critical density
C_i	Capacity of on-ramp connected to segment i
ρ_{max}	Maximum traffic density
$w_{\max}$	Maximum limit value for the queue length of on-ramp
θ	Parameters characterizing the adaptive horizon policy
E	Total number of training episode
K	Total time step of each episodes
φ	Learning rate representing the impact of new information on the Q-values
γ	Discount factor that balances immediate rewards with future rewards
State variables	**Description**
$\rho_i(k)$	Traffic density in segment i at time kT
$v_i(k)$	Mean speed of the vehicles included in segment i at time kT
$q_i(k)$	Traffic flow leaving segment i during the time step $[kT,(k+1)T)$
$w_i(k)$	Queue length of on-ramp connected to segment i at time kT
$d_i(k)$	Demand flow of on-ramp connected to segment i at time kT
$\hat{a}_k$	Updated horizon at time step k, action in the RL algorithm
U_{opt}	Optimal control sequence obtained by solving the optimization problem in the MPC framework
$\hat{r}_k$	Reward in the RL algorithm
$\hat{s}_k$	State in the RL algorithm
$Q(\hat{s}_k,\hat{a}_k)$	Q-value of taking action $\hat{a}_k$ at state $\hat{s}_k$
Decision variables	**Description**
$u_i(k)$	Metering rate of on-ramp connected to segment i at time kT
$r_i(k)$	Incoming flow generated by the on-ramp connected to segment i at time kT
P_{kc}	Prediction horizon at control time step k_c

$$V(\rho_i(k)) = v_{\text{free}} \exp(-\frac{1}{\alpha}(\frac{\rho_i(k)}{\rho_{\text{crit}}})^{\alpha}) \tag{4}$$

Let I_{on} denotes the set of segments with on-ramp connections. If the segment is connected with a ramp, i.e. $i \in I_{\text{on}}$, $r_i(k)$ can be calculated based on the relation between the queue length $w_i(k)$, capacity of on-ramp C_i and traffic density ρ_i; if none is connected, i.e. $i \notin I_{\text{on}}$, ρ_i is equal to zero, which can be described as:

$$r_i(k) = \begin{cases} u_i(k)\min\{d_i(k) + \frac{w_i(k)}{T}, C_i, C_i(\frac{\rho_{\max}-\rho_i(k)}{\rho_{\max}-\rho_{\text{crit}}})\}, \\ \qquad\qquad\qquad\qquad i \in I_{\text{on}} \\ 0, \qquad\qquad\qquad\qquad i \notin I_{\text{on}} \end{cases} \tag{5}$$

where, $u_i(k) \in [0,1]$, which is regarded as the control action. Queue length w_i can be calculated as:

$$w_i(k+1) = w_i(k) + T(d_i(k) - r_i(k)), i \in I_{\text{on}} \tag{6}$$

C. MPC formulation

In MPC, optimal control actions are implemented repeatedly in a rolling horizon manner. Let M relates the control time step k_c and simulation time step k as $k = Mk_c$. At each control time step k_c, an optimal control problem is solved based on the measured states at step k_c over a P_{kc} prediction horizon, and a set of optimal control sequences can be obtained. Then, only the first control action of the optimal control sequence is applied to the system. At the next control time step k_c+1, the optimal control problem is solved again based on the newly updated system states at step k_c+1, and also only the first control action is applied to the system, and repeat. Specially, the prediction horizon P_{kc} can be changed dynamically based on the updated states.

Traditionally, the objectives of local ramp metering are to minimize the total travel time, the penalty cost about the variability of control actions, and the penalty factor for queuing over the limit, which can be described respectively as:

$$L_T(x_k) = T\left(\sum_{i\in I_{\text{all}}} L_i\lambda_i\rho_i(k) + \sum_{i\in I_{\text{on}}} w_i(k)\right) \tag{7}$$

$$L_U(u_k) = \sum_{i\in I_{\text{on}}} (u_i(k) - u_i(k-1))^2 \tag{8}$$

$$L_W(\sigma_k) = \sum_{i\in I_{\text{on}}} \sigma_k \tag{9}$$

where, x_k is the state vector at time step k; u_k is the control action vector at time step k; σ_k is the penalty factor vector at time step k, which can be calculated as:

$$\begin{cases} w_i(k) - w_{\max} \le \sigma_i(k) \\ 0 \le \sigma_i(k) \end{cases}, i \in I_{\text{on}} \tag{10}$$

On the other hand, two modifications are applied to reduce the complexity of METANET model. Firstly, the highly nonlinear Eq. (4) is approximated by the piecewise approximation (PWA) method as:

$$V_{\text{PWA}}(\rho_i(k)) = \begin{cases} \alpha_1\rho_i(k) + \beta_1, 0 \le \rho_i(k) \le \rho_{\text{mid}} \\ \alpha_2\rho_i(k) + \beta_2, \rho_{\text{mid}} < \rho_i(k) \le \rho_{\max} \end{cases} \tag{11}$$

where, ρ_{mid} represents a parameter generated in approximation, the coefficients α_1, α_2, β_1 and β_2 can be generated in approximation. Then, a binary variable $\delta_i(k)$ is introduced to describe the logical conditions, defined as:

$$[\rho_i(k) \le \rho_{\text{mid}}] \leftrightarrow [\delta_i(k) = 1] \tag{12}$$

The implication of binary variable $\delta_i(k)$ can be modeled by the following linear constraints:

$$\begin{cases} \rho_i(k) - \rho_{\text{mid}} \le \rho_M(1 - \delta_i(k)) \\ \rho_i(k) - \rho_{\text{mid}} \ge \varepsilon + (\rho_m - \varepsilon)\delta_i(k) \end{cases} \tag{13}$$

where, ε is a small tolerance, typically the machine precision; $\rho_M = \rho_{\max} - \rho_{\text{mid}}$; $\rho_m = -\rho_{\text{mid}}$. With the binary variable $\delta_i(k)$, Eq. (11) can be transformed to a simpler form:

$$\begin{aligned} V_{\text{PWA}}(\rho_i(k)) = {} & \delta_i(k)(\alpha_1\rho_i(k) + \beta_1) \\ & + (1 - \delta_i(k))(\alpha_2\rho_i(k) + \beta_2) \end{aligned} \tag{14}$$

Secondly, over a large region of the state-action space, the min operator in eq. (5) will cause the gradient to be zero. Then, the control action is adjusted from metering rate $u_i(k)$ to on-ramp flow $r_i(k)$. According to Eq. (5), the following constraints should be considered to make sure that the new control action $r_i(k)$ is feasible:

$$\begin{cases} r_i(k) \leq d_i(k) + \frac{w_i(k)}{T} \\ r_i(k) \leq C_i \\ r_i(k) \leq C_i(\frac{\rho_{\max} - \rho_i(k)}{\rho_{\max} - \rho_{\text{crit}}}) \end{cases}, i \in I_{\text{on}} \tag{15}$$

Meanwhile, the cost term Eq. (8) is updated to:

$$L_R(r_k) = \sum_{i \in I_{\text{on}}} \left(\frac{r_i(k) - r_i(k-1)}{C_i} \right)^2 \tag{16}$$

where, r_k is the on-ramp flow vector at time step k.

Considering the above adjustment, the optimization model with METANET modelling for the MPC formulation is given by:

$$\begin{aligned} \min \sum_{j=1}^{MN_p} L_T(x_{j|k_c}) + \xi_R \sum_{j=0}^{N_p-1} L_R(r_{j_c(j)|k_c}) \\ + \xi_W \sum_{j=1}^{MN_p} L_W(\sigma_{j|k_c}) \end{aligned} \tag{17}$$

subject to:

$$x_{0|k_c} = x_{k_c} \tag{18}$$

$$x_{j+1|k_c} = f(x_{j|k_c}, r_{j_c(j)|k_c}, \delta_{j|k_c}), j = 0, ..., MN_p - 1 \tag{19}$$

$$g(x_{j|k_c}, r_{j_c(j)|k_c}, \delta_{j|k_c}, \sigma_{j|k_c}) \leq 0, j = 0, ..., MN_p \tag{20}$$

where, δ_k is the binary variable vector at time step k; ξ_R and ξ_W are the weight coefficients; the definition of $j_c(j)$ entails that the control action is kept constant for a complete control time step (including M simulation time steps), which is defined as $j_c(j) = \lfloor j/M \rfloor$. Once the above optimization model is solved, the first optimal control action $r^*_{0|k_c}$ is applied from the simulation time step Mk_c to $(m+1)K_c - 1$, as per the receding horizon approach.

III. AHMPC with RL-based Policy Learning

A. RL-AHMPC framework

The framework of RL-AHMPC is shown in Fig. 1. The RL agent learns the optimal horizon policy π_θ by continuously interacting with the environment. In addition, the environment consists of a plant and an MPC controller. At each time step, the RL agent sends a horizon to the environment based on the current system states $\hat{x}_k$, which can be described as:

$$\hat{a}_k \sim \pi_\theta(\hat{s}_k) \tag{21}$$

The optimization problem with the updated horizon will be solved at each step. Then, the plant moves to the next time step based on the optimal control sequence in the MPC framework. The RL agent observes the system states and reward signals, then updates θ.

The complete RL-AHMPC algorithm is shown in Algorithm 1. The RL agent interacts with the environment for E number of episodes. At each episode, the MPC controller calculates the optimal control sequence U_{opt}, and control action u_k is obtained based on U_{opt} to update system dynamics. In addition, the policy parameter θ is updated using observed states, reward and action $\{\hat{s}_k, \hat{a}_k, \hat{r}_k, \hat{s}_{k+1}\}$. After each episode, the environment is reset for the next episode. When finishing E number of episodes, the algorithm outputs the adaptive horizon policy π_θ.

Algorithm 1 RL-AHMPC algorithm

Input: E, K, MPC controller
Output: π_θ
Initialize θ
for episode = 0 to $E-1$ **do**
  Initialize $\hat{s}_k$, U_{opt}
  for k = 0 to $K-1$ **do**
    Select horizon $\hat{a}_k \sim \pi_\theta(\hat{s}_k)$
    $U_{\text{opt}}(k) \leftarrow$ Solving the optimization problem with horizon $\hat{a}_k$ in the MPC framework
    $u_k \leftarrow U_{\text{opt}}(1)$
    $\hat{r}_k$, $\hat{s}_{k+1} \leftarrow$ Simulate system dynamics using u_k
    Update θ based on $\{\hat{s}_k, \hat{a}_k, \hat{r}_k, \hat{s}_{k+1}\}$
    $k \leftarrow k+1$
  end for
end for

B. RL algorithm

In this paper, Q-learning is investigated to update the policy in the RL agent. Q-learning is a model-free method that works well on discrete action and state spaces. In the training process, the action, state and reward of Q-learning are updated at every time step, and are defined as follows.

Action $\hat{a}_k$: The action space in Q-learning refers to the set of all possible actions that the agent can take in a given state of the environment. For the focused problem, the action represents the adaptive horizon.

State $\hat{s}_k$: The state space in Q-learning refers to the set of all possible states that the environment can be in. For the focused problem, the state space is defined as $\{\hat{\rho}, \hat{w}\}$, where $\hat{\rho}$ is the observed traffic density and $\hat{w}$ is the observed queue length.

Reward $\hat{r}_k$: In Q-learning, the agent receives a reward signal from the environment after taking an action. For the focused problem, the reward is defined as:

$$\hat{r}_k \triangleq \frac{1}{L_T} + \frac{1}{\xi_R L_R} + \frac{1}{\xi_W L_W} - \xi_a \hat{a}_k \tag{22}$$

where, the first three elements measure the closed-loop control performance corresponding to the MPC controller and the last element encourages optimal prediction horizons to reduce online computation.

The Q-learning algorithm updates its Q-values $Q(\hat{s}_k, \hat{a}_k)$ based on the Bellman equation, which is a recursive equation

that expresses the value of a state-action pair in terms of the immediate reward and the estimated value of the next state. The policy π_θ in the algorithm 1 specifically refers to the Q-values $Q(\hat{s}_k, \hat{a}_k)$ in Q-learning algorithm. The update equation for Q-learning is given by:

$$Q(\hat{s}_k, \hat{a}_k) \leftarrow Q(\hat{s}_k, \hat{s}_k) + \varphi \left[\hat{r}_k + \gamma \max_{\hat{a}_{k+1}} Q(\hat{s}_{k+1}, \hat{a}_{k+1}) - Q(\hat{s}_k, \hat{a}_k) \right] \tag{23}$$

IV. Numerical Case Study

A. Settings

A simple highway network with three segments (see Fig. 2) is considered in numerical case studies, and each segment with 1 km length consists of two lanes, $L_i = 1$ km, $\lambda_i = 2$. Segment 1 is supplied by the uncontrolled mainstream original demand d_0, and is characterized by a capacity of 3500 vel/h, $C_0 = 3500$ vel/h. Segment 3 is additionally supplied by the uncontrolled on-ramp demand d_3, and is characterized by a capacity of 2000 vel/h, $C_3 = 2000$ vel/h. The density ρ_4 is used to simulate downstream congestion. The network parameters as found in [15] are used: $T = 10$ s, $\tau = 18$ s, $\nu = 60$ km^2/h, $\kappa = 40$ vel/km/lane, $\mu = 0.0122$, $\rho_{\max} = 180$ vel/km/lane, $\rho_{\text{crit}} = 33.5$ vel/km/lane, $v_{\text{free}} = 102$ km/h, $\alpha = 1.867$. The parameters in the MPC framework are used: $M = 6$, $\xi = 1$, $\xi_R = 1$, $\xi_R = 1$, $w_{\max} = 50$ vel. The parameters in the PWA function are used: $\rho_{\text{mid}} = 75.98$ vel/km/lane, $\alpha_1 = -1.3$, $\alpha_2 = -0.031$, $\beta_1 = 102$, $\beta_2 = 5.58$. The parameters in the Q-learning algorithm are used: $E = 500$, $K = 1000$, $\xi_a = 1$, $\varphi = 0.1$, $\gamma = 0.99$. The action space is defined as $\{1, 2, 3, 4, 5, 6\}$, which means that the prediction horizon is adjusted between 1 and 6. Specially, the state space $\{\hat{\rho}, \hat{w}\}$ is definded as:

$$\hat{\rho} = \lfloor \rho_2(k)/10 \rfloor \tag{24}$$

$$\hat{w} = \begin{cases} \lfloor w_3(k)/10 \rfloor, & \text{if } \lfloor w_3(k)/10 \rfloor < 9 \\ 9, & \text{else} \end{cases} \tag{25}$$

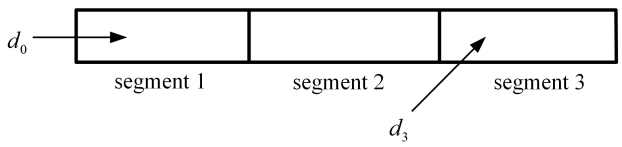

Fig. 2: Structure of the three-segment highway network

B. Simulation results and analysis

To validate the effectiveness of the proposed RL-AHMPC method, it is compared with the no control (NC) method and the traditional MPC methods with different prediction horizons. The traffic densities and queue lengths for the NC method are shown in Fig. 3. The traffic densities, queue lengths and dynamic prediction horizons for the RL-AHMPC method are shown in Fig. 4. And, the performances of different control methods are shown in Table II. In Fig. 3 and 4, the traffic density changes in three segments (s-1, s-2, s-3) and the queue length connected to segment 1 and 3 (w-1, w-3) are described. Meanwhile, the total travel times (TTT) of different control methods are compared in Table II, which can be calculated based on the Eq. (7). The total cost means the sum of all objective costs in Eq. (17). The total computation time (TCT) means the computational cost of the MPC controller with different prediction horizon among the whole simulation process.

TABLE II: Performance comparison between different methods

Type of MPC	TTT [h]	Total cost	TCT [s]
NC	643.97	20190.42	/
MPC-1	615.05	18094.90	30.40
MPC-2	607.09	16613.48	47.51
MPC-3	600.43	15166.97	75.53
MPC-4	596.03	14192.13	105.20
MPC-5	598.20	15056.40	125.32
RL-AHMPC	613.84	17913.17	39.84

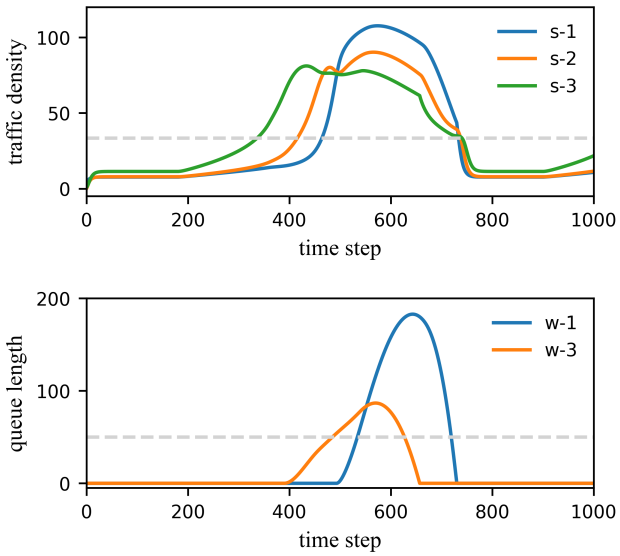

Fig. 3: Results for NC method

As shown in Table II, the MPC methods can effectively reduce the TTT and total cost. By applying the MPC-5, the TTT and total cost can be reduced by around 7.11% and 25.43% respectively in comparison to the NC method. Meanwhile, by applying the AHMPC, the TTT and total cost can be reduced by around 7.11% and 25.43% respectively in comparison to the NC method.4.68% and 11.28% respectively in comparison to the NC method. More specifically, as shown in Fig. 3 and 4, there are relatively large differences in the traffic details and queue lengths due to the application of MPC. In the RL-AHMPC method, the queue length is kept at a lower level compared to the NC method.

A comparison of the different MPC methods shows that there are some differences in performance due to the different prediction horizons. As the prediction horizon increases from

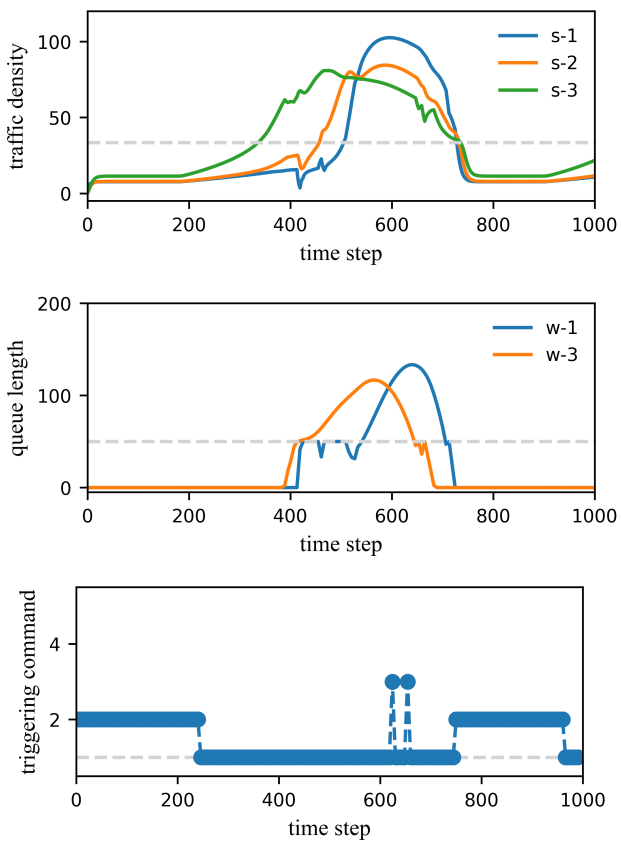

Fig. 4: Results for RL-AHMPC method

1 to 5, the computation time increases from 30.40 s to 125.32 s. The longer the prediction horizon in the MPC method, the more computation time is required. On the other hand, the MPC method with a prediction horizon of 4 (MPC-4) shows the best performance in terms of improving traffic efficiency. This suggests that it is not the case that the larger the prediction horizon, the more effective the MPC method is. In the RL-AHMPC method, the prediction horizon is adjusted based on the Q-values considering the system states. As shown in Fig. 4, when the traffic density is low, the prediction horizon is kept at 2; when the traffic density is high, the prediction horizon is kept at 1. Thanks to the adaptive prediction horizon, the RL-AHMPC method can effectively improve the traffic efficiency and keep the computational cost at a low level.

V. Conclusion

This paper presents a novel learning and model-based approach to the local ramp metering problem on highways, which combines MPC and RL. By leveraging the RL agent to adjust the prediction horizon based on observed states, the optimization problem in the MPC framework is optimized with a dynamic prediction horizon to improve the closed-loop performance while balancing the computational cost. The results show that the proposed RL-AHMPC method can significantly improve traffic efficiency, thanks to the optimal control sequence of the MPC controller and the intelligent horizon of the RL agent. Future work directions include: 1) the use of different RL algorithms to capture the adaptive prediction horizon; 2) the application of the proposed RL-AHMPC framework to different highway traffic control strategies and larger scale highway networks.

References

[1] S. Siri, C. Pasquale, S. Sacone, and A. Ferrara, "Freeway traffic control: A survey," *Automatica*, vol. 130, p. 109655, 2021.

[2] Y. Han, M. Wang, and L. Leclercq, "Leveraging reinforcement learning for dynamic traffic control: A survey and challenges for field implementation," *Communications in Transportation Research*, vol. 3, p. 100104, 2023.

[3] N. Groot, B. De Schutter, and H. Hellendoorn, "Integrated Model Predictive Traffic and Emission Control Using a Piecewise-Affine Approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 587–598, 2013.

[4] P. Chanfreut, J. M. Maestre, and E. F. Camacho, "Coalitional Model Predictive Control on Freeways Traffic Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 6772–6783, 2021.

[5] U. Todorovic, J. R. D. Frejo, and B. D. Schutter, "Distributed MPC for Large Freeway Networks Using Alternating Optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 1875–1884, 2022.

[6] C. Wang, Y. Xu, J. Zhang, and B. Ran, "Integrated Traffic Control for Freeway Recurrent Bottleneck Based on Deep Reinforcement Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 522–15 535, 2022.

[7] S. Zheng, M. Li, Z. Ke, and Z. Li, "Coordinated Variable Speed Limit Control for Consecutive Bottlenecks on Freeways Using Multiagent Reinforcement Learning," *Journal of Advanced Transportation*, vol. 2023, pp. 1–19, 2023.

[8] J. Jin, Y. Li, H. Huang, Y. Dong, and P. Liu, "A variable speed limit control approach for freeway tunnels based on the model-based reinforcement learning framework with safety perception," *Accident Analysis & Prevention*, vol. 201, p. 107570, 2024.

[9] Z. Chen, H. Hu, Y. Wu, Y. Zhang, G. Li, and Y. Liu, "Stochastic model predictive control for energy management of power-split plug-in hybrid electric vehicles based on reinforcement learning," *Energy*, vol. 211, p. 118931, 2020.

[10] D. Flessner, J. Chen, and G. Xiong, "Reinforcement Learning-Based Event-Triggered Active-Battery-Cell-Balancing Control for Electric Vehicle Range Extension," *Electronics*, vol. 13, no. 5, p. 990, 2024.

[11] E. Bøhn, S. Gros, S. Moe, and T. A. Johansen, "Reinforcement Learning of the Prediction Horizon in Model Predictive Control," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 314–320, 2021.

[12] F. Airaldi, B. De Schutter, and A. Dabiri, "Reinforcement Learning with Model Predictive Control for Highway Ramp Metering," 2023.

[13] D. Sun, A. Jamshidnejad, and B. D. Schutter, "A Novel Framework Combining MPC and Deep Reinforcement Learning With Application to Freeway Traffic Control," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2024.

[14] Y. Wang, X. Yu, J. Guo, I. Papamichail, M. Papageorgiou, L. Zhang, S. Hu, Y. Li, and J. Sun, "Macroscopic traffic flow modelling of large-scale freeway networks with field data verification: State-of-the-art review, benchmarking framework, and case studies using METANET," *Transportation Research Part C: Emerging Technologies*, vol. 145, p. 103904, 2022.

[15] A. Hegyi, B. De Schutter, and H. Hellendoorn, "Model predictive control for optimal coordination of ramp metering and variable speed limits," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 3, pp. 185–209, 2005.