Proceedings of the 43rd Chinese Control Conference
July 28-31, 2024, Kunming, China

Event-triggered Model Predictive Control with Reinforcement Learning for Highway Local Ramp Metering

Bo Jin[1,2], Jiayang Yu[1], Yaqiang He[1], Zuchen Que[1], Song Yang[1]

1. Zhejiang Scientific Research Institute of Transport, Hangzhou, China

2. Institute of Intelligent Transportation Systems, College of Civil Engineering and Architecture, Zhejiang University, Hangzhou, China
E-mail: jinbo0201@outlook.com, 13258332059@163.com, 873220830@qq.com, zjlsqzc@126.com, 36370548@qq.com

Abstract: In the backdrop of increasing pressure on highway traffic flow control, this paper proposed a reinforcement learning (RL) based event-triggered model predictive control (eMPC) method for the local ramp metering problem. Embed into the eMPC controller structure, the RL agent is responsible for providing triggering commands. The RL agent learns the event-triggered 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. By applying the novel RL-eMPC method, the traffic efficiency can be significantly improved with less computational costs. This is due to the combination of the optimal control sequences of MPC controller and the intelligent triggering rule of RL agent.

Key Words: active traffic control, Q-learning, learning-based model predictive control, data-driven control

1 Introduction

In recent decades, there has been a substantial escalation in the demand for traffic mobility, yielding advantageous implications for societal advancement while concurrently engendering many adverse outcomes for drivers and traffic controllers. The scholarly and industrial communities have extensively investigated traffic control methodologies with the aim of improving traffic safety and efficiency, which promotes the exploration of diverse strategies and methodologies. The domain of traffic control includes both urban and highway networks, this paper focuses specifically on highway traffic flow control.

Highway traffic flow control strategies mainly include ramp management, mainstream management and route guidance. Ramp management limits traffic flow entering the freeway to relieve congestion. Mainstream management regulates traffic flow already present in the mainstream, by using methods such as dynamic speed limits. Route guidance routes traffic flow on alternative paths of a network to disperse demand. In order to achieve safety and efficiency goals, the highway traffic control problem is constructed as an optimization model and the optimal strategies are obtained through solving the model. The model-based optimal control method MPC has been widely applied in the highway traffic flow control optimization problem [1–3]. 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 [4, 5]. Both MPC and RL method shave their advantages and disadvantages in dealing with optimal control problems. MPC method requires a significant amount of computing resources during real time control. Conversely, RL method exhibits an innate capacity to address intricate challenges with minimal online computational overhead. Nevertheless, the process of training a proficient RL is typically time-consuming, particularly for intricate systems [6].

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

To address the above challenges in MPC and RL methods for highway traffic flow control, this paper proposed an RL-eMPC method. In the proposed method, MPC is responsible for handling complex constraints and objectives. While, RL is responsible for outputting triggering commands, which is simpler compared to directly providing control strategies. Thanks to the optimized triggering commands, the MPC controller can complete the entire control cycle task with lower computational costs.

1.1 Related Work

In this section, an overview of related work that applies MPC, RL, and MPC with RL methods to solve highway traffic flow control problems is given.

MPC is a widely recognized method for real-time control of dynamic systems. It operates by predicting the system states over a finite time horizon and optimizing a suitable objective function. This is accomplished through an iterative solution of a Finite-Horizon Optimal Control Problem (FHOCP), which is updated using real-time system states. The macroscopic traffic flow models CTM and METANET [7] are frequently employed for the purpose of prediction in highway traffic control problems. In [1], the traffic control problem was solved by MPC, where the METANET model was used as the prediction model. In [2], MPC controllers with standard and modified CTM models were compared via simulation. Moreover, in order to reduce the computational cost for real-time application, an event-triggered control scheme is applied to avoid unnecessary calculations. In [3], a control scheme was proposed to reduce the computational load with a feedback MPC controller in which suitable triggering conditions were defined.

RL-based artificial intelligence has achieved remarkable progress by surpassing top human professionals in complex multiplayer games. These achievements serve as a compelling demonstration of the immense potential inherent in RL methods. Taking traffic flow simulations as training environments, Wang et al. [8] 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. In [4], a more effective RL method is developed for differential variable speed limit control, which was trained and tested under a microscopic simulator. In [5], the RL model was trained using a combination of historic data and synthetic data generated from a traffic flow model.

Combining RL and MPC can fully leverage the advantages of model-based and learning-based strategies in control problems. In [9], an RL model algorithm was developed to obtain the closed-loop optimal/suboptimal solutions, so that the computational costs were reduced in MPC controllers. By using the RL method to solve the optimization problem in the MPC framework, an accurate and highly efficient solution can be obtained [10]. In [11], an RL model was used to trigger MPC aiming to balance the closed-loop control performance and event frequency. A few studies have investigated this topic and applied RL-MPC methods in the field of highway traffic control. Sun et al. [6] 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. Airaldi et al. [12] utilized RL to adjust the parametrisation of MPC based on observed data, in which the accuracy of the METANET model was improved to enhance closed-loop performance.

1.2 Proposed Approach and Contributions

Few researchers have addressed the method of combining MPC and RL for the highway traffic control problem. The contribution of this paper is proposing an RL-based eMPC (RL-eMPC) 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 eMPC provides optimal triggering commands to avoid unnecessary calculations, which can reduce the computational cost of the controller.

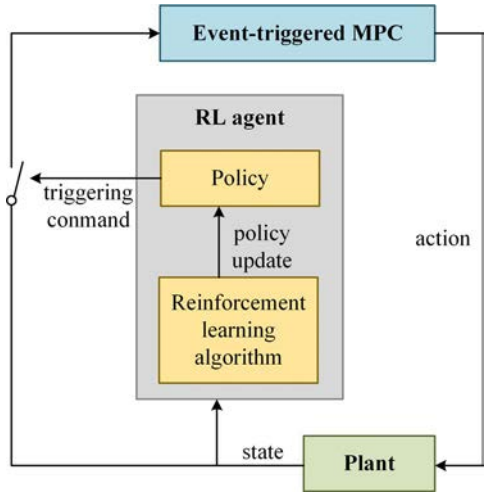

Fig. 1: The framework of RL-eMPC

The structure of this paper is organized as follows: Section 2 presents the METANET model and related MPC algorithm for the highway ramp metering problem. Section 3 provides details on the novel RL-based eMPC method that is proposed in this paper. Section 4 gives numerical case studies that implement the proposed RL-based eMPC method on a three-segment highway network. Finally, Section 5 concludes this paper and proposes topics for future work.

2 Problem Formulation

2.1 METANET modelling

In this paper, the macroscopic second-order METANET framework is used to formulate a discrete-time dynamical representation of the highway traffic under local ramp metering. In the METANET framework, the discrete time step is denoted by T. Each segment i, $i \in I_{all}$, at discrete time $t = kT$ is characterized by the following variables:

- Traffic density $\rho_i(k)$ (vel/km/lane) is the number of vehicles in segment i at time kT divided by length L_i and the number of lanes μ_i.
- Mean speed $v_i(k)$ (km/h) is the mean speed of the vehicles included in segment i at time kT.
- Traffic flow $q_i(k)$ (vel/h) is the number of vehicles leaving segment i during the time step $[kT, (k+1)T)$, divided by T.

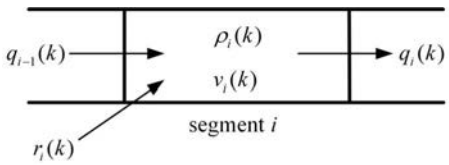

Fig. 2: Characteristic of segment i at time kT

The previously defined traffic variables, as shown in Fig. 2, are calculated for each segment i at each time step k by the following equations:

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

$$q_i(k) = \lambda_i \rho_i(k) v_i(k) \quad (2)$$

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

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

where, λ_i is the number of lanes in segment i; L_i is the length of segment i; τ, ν, κ, μ and α are model parameters; v_{free} is the free speed; ρ_{crit} is the critical density. In particular, $r_i(k)$ is the incoming flow generated by the on-ramp connected to segment i. 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:

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

where, $u_i(k)$ is the metering rate, $u_i(k) \in [0,1]$, which is regarded as the control action; ρ_{max} is maximum density; w_i denotes the queue length of on-ramp connected to segment i at time step k, which can be calculated as:

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

where, $d_i(k)$ is the demand flow of on-ramp connected to segment i at time step k, which acts as an uncontrollable external input.

2.2 MPC formulation

MPC is a widely recognized control framework that was initially employed in conjunction with the METANET framework, as documented in [1]. 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 N_p step 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.

The objectives of the traditional highway ramp metering are to minimize the total travel time and the penalty cost about the variability of control actions. The total travel time can be calculated as:

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

where, x_k is the state vector at time step k. The penalty cost about the variability of control actions can be calculated as:

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

where, u_k is the control action vector at time step k. Additionally, in order to avoid safety scenarios caused by the long queue length, a soft constraint is introduced [1]:

$$\begin{cases} w_i(k) - w_{\max} \le \sigma_i(k) \\ 0 \le \sigma_i(k) \end{cases}, i \in I_{\text{on}} \tag{9}$$

where, $w_{\max}$ is the maximum limit value for the queue length; $\sigma_i(k)$ is a slack variable, which represents the penalty factor for queuing over the limit. The penalty factor should be considered in the objective function, which is minimized to avoid long queue length at on-ramps:

$$L_W(\sigma_k) = \sum_{i\in I_{\text{on}}} \sigma_k \tag{10}$$

where, σ_k is the penalty factor vector at time step k.

To reduce the complexity of the model, the following two modifications are applied. Firstly, considering that Eq. (4) is highly nonlinear, it is difficult to solve the optimization model containing this kind of equation. Thus, it 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 [13]:

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

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, the min operator in Eq. (5) can cause the gradient to be zero over a vast region of the state-action space [12]. Thus, 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:

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

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.

Overall, the optimal control problem 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:

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

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

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

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 $j_c(j) = \lfloor j/M \rfloor$. As aforementioned, once the formulated optimal control problem is solved, the first optimal control action $r^*_{0|k_c}$ is applied from simulation time step Mk_c to $(m+1)K_c-1$, as per the receding horizon approach.

3 eMPC with RL-based Policy Learning

3.1 RL-eMPC framework

The framework of RL-eMPC is shown in Fig. 1. The RL agent learns the event-triggered policy π_θ by continuously interacting with the environment. For the problem studied in this paper, the environment consists of a plant and an eMPC controller. At each time step, the RL agent sends a triggering command to the environment based on the current system states $\hat{x}_k$, which can be described as:

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

where, $\hat{a}_k$ represents the triggering command at time step k; θ represents the parameters characterizing the event-triggered policy. Based on the triggering command, the eMPC will be triggered when $\hat{a}_k = 1$ and will not be triggered when $\hat{a}_k = 0$. Then, the plant moves to the next time step based on the updated ($\hat{a}_k = 1$) or un-updated ($\hat{a}_k = 0$) control sequence. The RL agent observes the system states and reward signals, then updates θ.

The complete RL-eMPC algorithm is shown in Algorithm 1. In this algorithm, E represents the total number of training episode; K represents the total time step of each episodes; U_{opt} represents the optimal control sequence obtained by solving the optimization problem in the MPC framework; i_u represents the index of control action in the optimal control sequence; $\hat{r}_k$ represents the reward. The RL agent interacts with the environment for E number of episodes. At each episode, the MPC controller is triggered to calculate 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 event-triggered policy π_θ.

3.2 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 every time step, and are defined as it 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 space is defined as $\{0, 1\}$, where 0 means no trigger event and 1 indicates a trigger event.

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.

Algorithm 1 RL-eMPC algorithm

Input: E, K, MPC controller
Output: π_θ
Initialize θ
for episode = 0 to $E-1$ **do**
  Initialize $\hat{s}_k$, U_{opt}, $i_u \leftarrow 0$
  for k = 0 to $K-1$ **do**
    Select action $\hat{a}_k \sim \pi_\theta(\hat{s}_k)$
    if $\hat{a}_k == 1$ **then** ▷ MPC controller is triggered
      $U_{\text{opt}} \leftarrow$ Solving the optimization problem
      $i_u = 0$
    else ▷ MPC controller is untriggered
      $i_u \leftarrow i_u + 1$
    end if
    $u_k \leftarrow U_{\text{opt}}(i_u)$
    $\hat{r}_k$, $\hat{s}_{k+1} \leftarrow$ 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

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 fewer events to reduce online computation [11].

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:

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

where, $Q(\hat{s}_k, \hat{a}_k)$ represents the Q-value of taking action $\hat{a}_k$ at state $\hat{s}_k$; φ is the learning rate representing the impact of new information on the Q-values; γ represents the discount factor that balances immediate rewards with future rewards.

4 Numerical Case Study

4.1 Settings

A simple highway network with three segments (see Fig. 3) 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 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 2000 vel/h, $C_3 = 2000$ vel/h. The density ρ_4 is used to simulate downstream congestion. The network parameters as found in [1] 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: $N_p = 4$, $M = 6$, $\xi = 1$, $\xi_R = 1$, $\xi_R = 1$, $w_{\max} = 50$ 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: $E = 500$, $K = 1000$, $\xi_a = 1$, $\varphi = 0.1$, $\gamma = 0.99$.

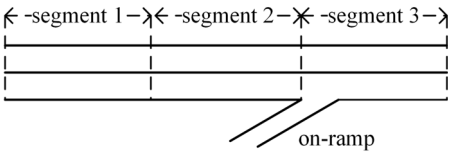

Fig. 3: Structure of the three-segment highway network

To validate the effectiveness of the proposed RL-eMPC method, it is compared with the traditional eMPC method, no control method. More details about these four methods are as follows:

- NC: No-control method, in which the control action is fixed as 1.
- eMPC: event-triggered method, in which the optimization is triggered based on the observed states, including the traffic density of segment 2 $\rho_2(k)$ and the queue length of on-ramp connected to segment 3 $w_3(k)$. The event-triggered policy of eMPC is defined as:

$$\hat{a}_k = \begin{cases} 1, \text{ if } \rho_2(k) > \rho_{\text{crit}} \text{ or } w_3(k) > w_{\max} \\ 0, \text{ else} \end{cases} \tag{24}$$

- RL-eMPC: RL-based event-triggered MPC method, in which the optimization is triggered based on the trained Q-values. Specially, the state space $\{\hat{\rho}, \hat{w}\}$ is definded as:

$$\hat{\rho} = \lfloor \rho_2(k)/10 \rfloor \tag{25}$$

$$\hat{w} = \begin{cases} \lfloor w_3(k)/10 \rfloor, \text{ if } \lfloor w_3(k)/10 \rfloor < 9 \\ 9, \text{ else} \end{cases} \tag{26}$$

4.2 Simulation results and analysis

The traffic density, queue length and triggering command for three different control methods are shown in Fig. 4, Fig. 5 and Fig. 6 respectively, and the performance is shown in Table 1. In Fig. 4, Fig. 5 and Fig. 6, 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 the three methods are compared in Table 1, which can be calculated based on the Eq. (7). The total cost in Table 1 means the sum of all objective costs in Eq. (17).

Table 1: Performance comparison between three methods

Type of method	TTT [h]	Total cost	Triggering times
NC	643.0	20190.4	0
eMPC	632.1	18969.2	64
RL-eMPC	618.7	17915.8	93

As shown in Fig. 4, during the whole control process, the triggering command is zero, which means that the MPC controller has never been triggered in NC method. On

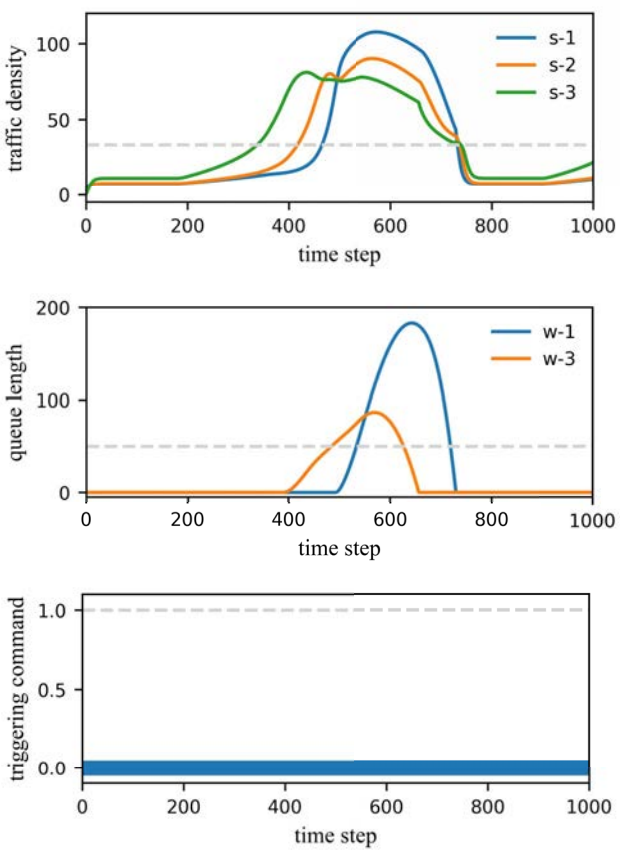

Fig. 4: Results for NC method

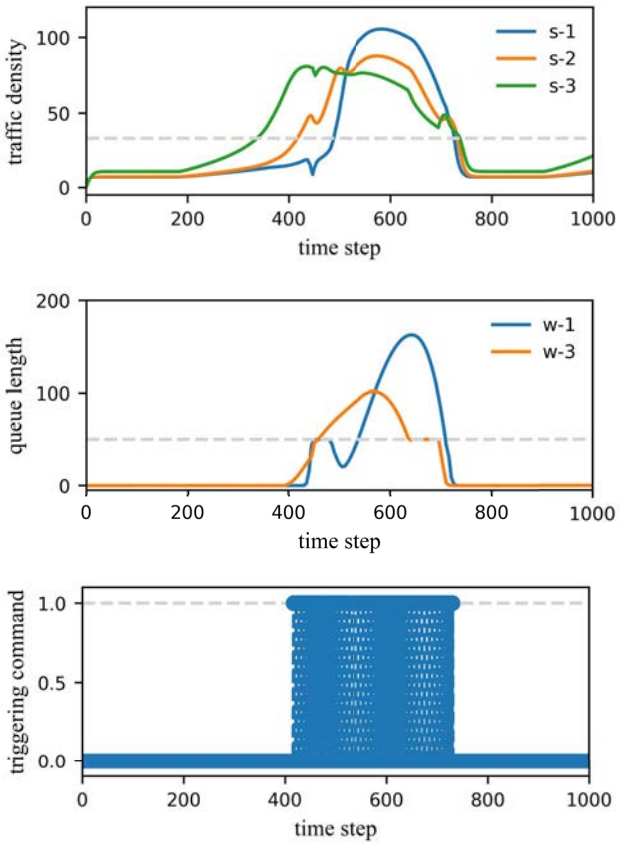

Fig. 5: Results for eMPC method

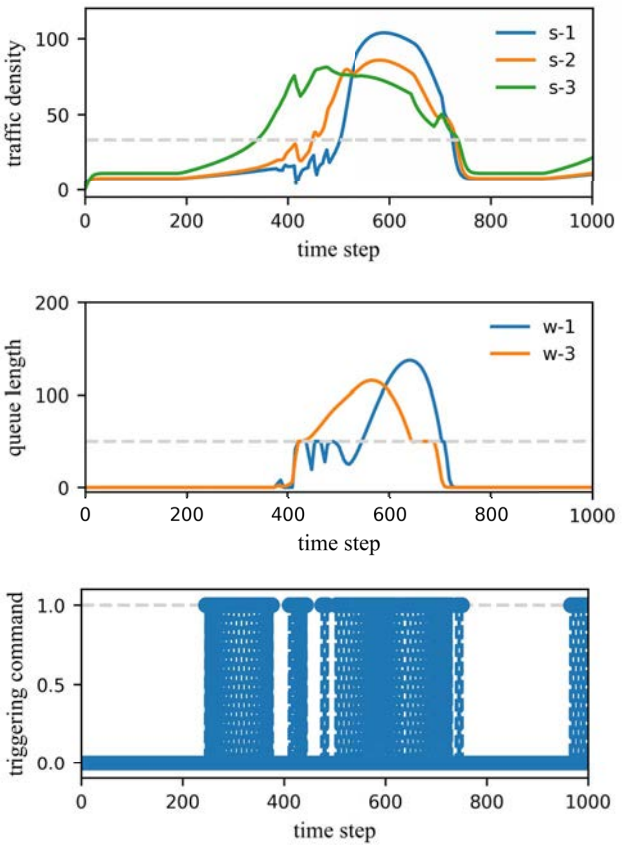

Fig. 6: Results for RL-eMPC method

the other hand, the MPC controller is triggered 64 times in eMPC method, and 93 times in RL-eMPC method. In eMPC method, the MPC controller is mainly triggered in time steps from 400 to 700, when $\rho_2(k) > 33.5$ veh/km/lane or $w_3(k) > 50$ vel. This is consistent with the triggering rules set in the Eq. (24). In RL-eMPC method, the triggering rule is more flexible, which is based on the trained Q-values. Different from eMPC method, the MPC controller is triggered in RL-eMPC method before the traffic density over the critical density or the queue length over the maximum limit value. Optimal control actions are applied in advance to adjust traffic flow.

Due to the flexibility and foresight, the RL-eMPC method demonstrates better performance in terms of improving traffic efficiency. As shown in Table 1, by applying RL-eMPC method, the TTT can be reduced around 3.8% and 2.1% in comparison to NC and eMPC methods respectively. The total cost (mainly including TTT and the penalty factor for queuing over the limit) can be reduced around 11.3% and 5.6% in comparison to NC and eMPC methods respectively. As shown in Fig. 4, Fig. 5 and Fig. 6, the queue length over the maximum limit is suppressed by applying eMPC and RL-eMPC methods. The suppression effect of RL-eMPC method is more pronounced in comparison to eMPC method.

5 Conclusion

In this paper, a novel learning-based and model-based approach to the highway local ramp metering problem that combines MPC and RL. By leveraging RL agent to adjust the triggering command based on observed states, the MPC controller is triggered reasonably to improve closed-loop performance while balancing computation cost. The results show that the proposed RL-eMPC method can significantly improve traffic efficiency, thanks to the optimal control sequence of MPC and the intelligent triggering rule of RL agent. Future work directions include: 1) the use of different RL algorithms to capture the nonlinear triggering rule; 2) the application of the proposed RL-eMPC framework to different highway traffic control strategies and larger scale highway networks.

References

[1] A. Hegyi, B. D. 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, pp. 185–209, 2005.

[2] L. Maggi, S. Sacone, and S. Siri, "Freeway traffic control considering capacity drop phenomena: Comparison of different mpc schemes," *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 457–462, 2015.

[3] A. Ferrara, S. Sacone, and S. Siri, "Design of networked freeway traffic controllers based on event-triggered control concepts," *International Journal of Robust and Nonlinear Control*, vol. 26, pp. 1162–1183, 2016.

[4] Y. Wu, H. Tan, L. Qin, and B. Ran, "Differential variable speed limits control for freeway recurrent bottlenecks via deep actor-critic algorithm," *Transportation Research Part C-emerging Technologies*, vol. 117, p. 102649, 2020.

[5] Y. Han, M. Wang, L. Li, C. Roncoli, J. Gao, and P. Liu, "A physics-informed reinforcement learning-based strategy for local and coordinated ramp metering," *Transportation Research Part C: Emerging Technologies*, vol. 137, p. 103584, 2022.

[6] 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*, 2024, early access.

[7] 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.

[8] 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, pp. 15 522–15 535, 2022.

[9] X. Xu, H. Chen, C. Lian, and D. Li, "Learning-based predictive control for discrete-time nonlinear systems with stochastic disturbances," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, pp. 6202–6213, 2018.

[10] L. Roveda, A. Testa, A. A. Shahid, F. Braghin, and D. Piga, "Q-learning-based model predictive variable impedance control for physical human-robot collaboration," *Artificial Intelligence*, vol. 312, p. 103771, 2022.

[11] J. Chen, X. Meng, and Z. Li, "Reinforcement learning-based event-triggered model predictive control for autonomous vehicle path following," *2022 American Control Conference (ACC)*, pp. 3342–3347, 2022.

[12] F. Airaldi, B. D. Schutter, and A. Dabiri, "Reinforcement learning with model predictive control for highway ramp metering," *ArXiv*, vol. abs/2311.08820, 2023.

[13] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, pp. 407–427, 1999.