Reinforcement Learning vs. Backstepping Control of Stop-and-Go Traffic

Huan Yu\textsuperscript{1}, Sachong Park\textsuperscript{2}, Scott Moura\textsuperscript{2}, Alexandre Bayen\textsuperscript{2}, Miroslav Krstic\textsuperscript{1}

Abstract—This article develops a Reinforcement Learning (RL) boundary controller of stop-and-go traffic congestion on a freeway segment. The traffic dynamics are governed by a macroscopic Aw-Rascle-Zhang (ARZ) model, consisting of $2 \times 2$ nonlinear Partial Differential Equations (PDEs) for traffic density and velocity. The boundary actuation of traffic flow is implemented with ramp-metering, which is a common approach to freeway congestion management. We use a discretized ARZ PDE model to describe the macroscopic freeway traffic environment for a stretch of freeway, and apply deep RL to develop continuous boundary control on the outlet boundary. The control objective is to achieve $L^2$ norm regulation of the traffic state to a spatially uniform density and velocity. A recently developed neural network based policy gradient algorithm is employed, called proximal policy optimization. For comparison, we also consider an open-loop controller and a PDE backstepping approach. The backstepping controller is a model-based approach recently developed by the co-authors. Ultimately, we demonstrate that the RL approach nearly recovers the control performance of the model-based PDE backstepping approach, despite no a priori knowledge of the traffic flow dynamics.

I. INTRODUCTION

Stop-and-go traffic is a common phenomenon in congested freeways, causing increased consumption of fuel and unsafe driving conditions. Oscillations can be caused by delayed driver response. Traffic instabilities, also known as “jamiton”, \cite{1,2,3,4} can be modeled with the Aw-Rascle-Zhang (ARZ) model \cite{5,6}, which consists of second-order, nonlinear hyperbolic PDEs modeling traffic density and velocity.

To stabilize the oscillations of stop-and-go traffic, we propose boundary control strategies. Boundary control through ramp metering and varying speed limits are widely and effectively used in freeway traffic management. In developing boundary feedback control through ramp metering and varying speed limits, many recent efforts \cite{4,10,11} focus on the ARZ model, due to its simplicity and realism. Traffic dynamics are governed with Aw-Rascle-Zhang (ARZ) model, consisting of $2 \times 2$ nonlinear hyperbolic partial differential equations (PDEs). Boundary control of the ARZ PDE on a freeway segment is developed in \cite{7,8,9} with backstepping control design and $L^2$ norm stabilization of traffic oscillations in finite time.

One challenge with model-based control design is calibration of the PDE model parameters with field data. This can be challenging in practice. Furthermore, scalability is an issue, as control design becomes exceedingly complex for freeway networks. Finally, traffic flow dynamics are fundamentally nonlinear phenomena, and the vast majority of model-based PDE control theory is limited to linear systems. These challenges motivate investigation into model-free control methods.

Recent developments in Reinforcement Learning (RL) enable model-free control of high-dimensional continuous control systems, which further motivates the present study. Using model-free RL control, we do not assume any prior knowledge of the model structure and parameters, and it thus does not rely on model calibration. Instead, we leverage iterative interactions with a simulator of the traffic flow dynamics. The article in \cite{12} uses a multi-agent RL algorithm to control the traffic light around a traffic junction. The authors propose a framework where each agent is able to switch between independent and integrated modes. In the integrated mode, the agent solves the multi-agent RL problem using modular Q-learning. The article in \cite{13} designs a RL-based controller using policy gradient methods, such as REINFORCE, Trust Policy Optimization (TRPO), and the Truncated Natural Policy Gradient (TNPG) algorithm. The authors also propose a mutual weight regularization (MWR) algorithm which alleviates the curse of dimensionality associated with multi-agent control schemes by sharing experience between agents while giving each agent the opportunity to specialize its action policy. This work considers the Lighthill-Whitham-Richard (LWR) first-order PDE model. Lastly the work of \cite{14} attempts to solve the optimal ramp metering problem via Q-learning, motivated by the uncertain and stochastic nature of traffic dynamics.

The main contribution of this article is the very first result on RL control of the inhomogeneous ARZ model, to authors’ knowledge. We first formulate a state regulation control problem for the ARZ PDE model via boundary control. Then we develop a RL approach based on proximal policy optimization (PPO), recently developed in 2017, which falls within the class of policy gradient methods. PPO ultimately yields a state feedback boundary controller. However, the design is obtained from interactions with a simulation environment as opposed to direct synthesis from a mathematical model. Performance of PPO is compared with a PDE backstepping controller recently developed by the co-authors \cite{7}. Interestingly, PPO nearly recovers the control performance of the model-based PDE backstepping approach.

The outline of this article is as follows: Section 2 sum-

\textsuperscript{1}Huan Yu and Miroslav Krstic are with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, 9500 Gilman Dr, La Jolla, CA 92037 huy015@ucsd.edu
\textsuperscript{2}Sachong Park, Scott Moura and Alexandre Bayen are with the Department of Civil and Environmental Engineering, University of California, Berkeley, CA 94720 sspark@berkeley
marizes the ARZ traffic flow model. Section 3 details the reinforcement learning-based approach to boundary control. Section 4 provides numerical results, comparing a baseline, benchmark and RL controller. The conclusion summarizes the main results, and discusses future work.

II. ARZ PDE Traffic Model

We consider the ARZ PDE model to describe the traffic dynamics on a freeway segment. The state variables are traffic density $\rho(x,t)$ and traffic speed $v(x,t)$, defined on the domains $x \in [0,L]$, $t \in [0,T]:$

$$\rho_t + (\rho v)_x = 0, \quad (1)$$

$$v_t + (v - \rho p'(\rho)) v_x = \frac{V(\rho) - v}{\tau}, \quad (2)$$

where $(\cdot)_z$ is short-hand notation for the differential operator $\partial/\partial z$. Parameter $\tau$ is the relaxation time, and captures how quickly drivers adjust their velocity to the equilibrium. The variable $p(\rho)$ is defined as the traffic pressure, an increasing function of density

$$p(\rho) = \rho^\gamma, \quad (3)$$

and $\gamma \in \mathbb{R}_+$. The equilibrium velocity-density relationship $V(\rho)$ is given by the Greenshield model,

$$V(\rho) = v_f \left[ 1 - \left( \frac{\rho}{\rho_m} \right)^\gamma \right]. \quad (4)$$

Our control objective is to regulate the state around an equilibrium reference state $(\rho^\star, v^\star)$, where

$$v^\star = V(\rho^\star). \quad (5)$$

We choose the density $\rho^\star$ such that the reference system $(\rho^\star, v^\star)$ is in the congested regime, which can be characterized by the two characteristics of the linearized PDE model [7]

$$\lambda_1 = v^\star > 0, \quad (6)$$

$$\lambda_2 = v^\star + \rho^\star V'(\rho^\star) < 0. \quad (7)$$

The first characteristic is always greater than 0. When traffic is light, $\lambda_2 > 0$ is satisfied. When traffic is dense, then $\lambda_2 < 0$ and in this regime there can be upstream propagation of oscillations in the states. This can also be characterized with the Traffic Froude Number (TFN) in [13]. Consequently, we have hetero-directional propagation of oscillations in congested traffic, represented with blue and red arrows in Fig. 1. The density oscillations are carried downstream by vehicles while the velocity oscillations are transported upstream. Intuitively, drivers are mostly affected by vehicles driving in front of them. The stop-and-go traffic is characterized by oscillations, caused by delayed driver reaction to vehicles in front of them.

III. CONTROL OF ARZ MODEL WITH REINFORCEMENT LEARNING

In this subsection, we introduce a reinforcement learning approach to boundary control of the nonlinear ARZ traffic flow model. Although explicit knowledge of the differential equations are not required, we assume the traffic dynamics are governed by a Markov Decision process. In particular, we will use a policy gradient method since they are applicable to continuously valued control actions.

A. Control objectives

Our control objective is $L^2$ regularization of the density and velocity to uniform steady state values, via ramp-metering on the boundary. Without control, oscillations can occur due to delayed driver response. In order to reduce oscillations in congested traffic, we actuate traffic flow from the downstream outlet of the freeway segment. This can be realized with ramp metering (RM) so that outgoing flow is actuated, as shown in Fig. 1. Specifically, a traffic light located on the on-ramp manages incoming traffic flow via pulse width modulation. Alternatively, one can actuate velocity by installing a variable speed limit (VSL) sign. In this setup, velocity can be controlled at the freeway segment outlet, and upstream traffic can be stabilized.

B. Reinforcement Learning formulation

In this section, we briefly introduce the Markov Decision Process (MDP). MDP is a classical modeling paradigm for many reinforcement learning problems, including the one used in this paper. At each time step, the controller (a.k.a. “agent” in the language of RL researchers) performs an action which leads to two things: the state evolves, and then the controller receives a cost (or reward) from the system. The agent’s goal is to discover an optimal policy (a.k.a. state feedback controller in the language of controls scientists) such that it maximizes the total rewards received from the system in response to its actions. An MDP consists of a tuple of 5 elements:

1. $\mathcal{S}$: Set of states. At each time step the state of the environment is an element $s \in \mathcal{S}$.
2. $\mathcal{A}$: Set of actions. At each time step the agent chooses an action $a \in \mathcal{A}$ to perform.
3. $\mathcal{P}(s_{t+1}|s_t, a_t)$: Probabilistic state transition model that describes how the system’s state changes when the user performs an action $a$. Note the dynamics are conceptualized as a stochastic process in discrete time.

Fig. 1. Ramp metering located at the outlet of a freeway segment.
4. $\mathcal{R}(s_t, a_t)$: Reward model that describes the real-valued reward an agent receives from the system after performing an action. In MDP, the reward value depends on the current state and the action performed, i.e., $r(s, a)$.

5. $\gamma \in [0, 1)$: A discount factor that encodes the importance of future rewards.

The control policy (i.e., control law) is denoted by the symbol $\pi$. We consider randomized policies, so $\pi$ outputs the probability of applying action $a$, conditioned on being in state $s$:

$$\pi(a|s) : A \times S \to [0, 1].$$  \hspace{1cm} (8)

The agent’s goal is to find the policy that will maximize the total rewards received from the system. In this work, the agent is a ramp metering controller that actuates outgoing flow, and the system is the ARZ PDE model. The states $s_t$ are traffic density $\rho(x,t)$ and speed $v(x,t)$. The reward $r(s, a)$ is the $L^2$ norm of density and speed. The total discounted reward from time $t$ onward can be expressed as:

$$\mathcal{R}_t = r_t(s_t, a_t) + \gamma r_{t+1}(s_{t+1}, a_{t+1}) + \gamma^2 r_{t+2}(s_{t+2}, a_{t+2}) + \cdots = \sum_{t'=t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'}) .$$ \hspace{1cm} (9)

Note that $\gamma \in [0, 1)$ ensures that total discounted reward $\mathcal{R}_t$ remains finite over infinite time.

The state-action value function $Q^\pi$, the state value function $V^\pi$, and the advantage function $A^\pi$ for policy $\pi$ are defined as follows:

$$Q^\pi(s_t, a_t) = r_t(s_t, a_t) + \mathbb{E}_{s_{t+1} \sim \mathcal{P}(s_{t+1}|s_t, a_t)} [V^\pi(s_{t+1})] ,$$ \hspace{1cm} (10)

$$V^\pi(s_t) = \sum_{t'=t}^{T} \mathbb{E} [r(s_{t'}, a_{t'})] ,$$ \hspace{1cm} (11)

$$A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t) .$$ \hspace{1cm} (12)

These definitions will be used in the RL algorithm.

1) State and action space: We consider a discretized approximation of the ARZ PDE model using the second-order Lax-Wendroff scheme [19] with conservative state variables. The solution $\rho(x,t)$ and $v(x,t)$ to the ARZ PDE model is approximated by piecewise constant functions on discretized temporal and spatial domains. The solution domain is $[0, L] \times [0, T]$. The discretization resolution $\Delta t$ and $\Delta x$ are chosen such that the Courant-Friedricks-Lewy (CFL) condition is met $\Delta t \leq c\Delta x$, where $c$ is defined as the maximum characteristic speed of the nonlinear hyperbolic ARZ PDE model. This is further detailed in Section IV.

The action space consists of outgoing flow at discretized elements $\{0, \Delta t, \ldots, T - \Delta t, T\} \times \{L\}$, and belongs to a bounded domain $[0, q_c]$, where $q_c$ is the road capacity representing the maximum flow allowed by the road.

We can explicitly write the states and actions at discrete time $t$ in the MDP formulation as,

$$s_t = [\rho(0,t), \rho(\Delta x, t), \ldots, \rho(L,t)] ,$$
$$v(0,t), v(\Delta x, t), \ldots, v(L,t)]^T ,$$ \hspace{1cm} (13)

$$a_t = [q(L,t)]^T .$$ \hspace{1cm} (14)

Note, we have abused notation by allowing $t$ to represent both continuously and discretely-valued times. Nevertheless, the meaning will be clear from context.

2) Parameterized stochastic policies with deep neural networks: We construct our controller based on a neural network as follows:

$$a_t \sim N(\mu, \sigma^2) , \quad \text{where } [\mu, \sigma] = f_{\text{DNN}}(s_t; \theta) \hspace{1cm} (15)$$

That is, the control action is normally distributed with a mean $\mu$ and standard deviation $\sigma$ computed from a deep neural network (DNN). The DNN $f_{\text{DNN}}(s_t; \theta) : S \to \mathbb{R}^2$ and is parameterized by weight vector $\theta$. Our task is to optimize $\theta$ to maximize $\mathcal{R}_t$.

3) Proximal Policy Optimization (PPO): We adopt a policy gradient-based approach to obtain a continuous-valued stochastic control policy. Mathematically, the goal is to find:

$$\theta^* = \arg \max_{\theta} \mathbb{E}[\mathcal{R}_t]$$ \hspace{1cm} (16)

where the expectation is taken w.r.t. $\mathcal{P}(s_{t+1}|s_t, a_t)$ and $\pi_\theta(a_{t}|s_t)$, and $\theta$ parameterizes the control policy distribution. Policy gradient methods essentially solve [16] via gradient ascent. The key challenge is estimating the gradient, since it is computationally intractable to compute it exactly. One can re-formulate this optimization problem in terms of the expected reward of policy $\pi_\theta$ and the advantage of $\pi_{\theta_{\text{old}}}$ [15], [16]:

$$\max_{\theta} \mathbb{E}_t \left[ \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} A_t \right]$$ \hspace{1cm} (17)

where the hats on $\mathbb{E}_t$ signify a sample mean, and $\hat{A}_t$ indicates an estimate from simulations. In [16], the authors prove the expected reward corresponding to $\pi_\theta$ increases relative to $\pi_{\theta_{\text{old}}}$ if a distance measure between $\pi_\theta$ and $\pi_{\theta_{\text{old}}}$ is sufficiently bounded. This motivates the following trust region policy optimization algorithm:

$$\max_{\theta} \mathbb{E}_t \left[ \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} A_t \right]$$ \hspace{1cm} (18)

subject to $\mathbb{E}_t [KL(\pi_{\theta_{\text{old}}}(\cdot|s_t), \pi_\theta(\cdot|s_t))] \leq \delta$ \hspace{1cm} (19)

where $\theta_{\text{old}}$ is the vector of policy parameters before the update. KL-divergence measures the difference between the old policy and current policy. The constraint ensures that the new policy does not deviate from the old policy by $\delta$. Importantly, this guarantees monotonically increasing expected rewards as the policy updates.

In this work, we adopt the PPO reinforcement learning algorithm [17], which is based on trust region policy optimization (TRPO) [16]. The PPO algorithm similarly limits the new policy from being excessively far from the previous one. However, it does so with a modified objective.
that penalizes changes to the policy that move \( r_t(\theta) = \pi_{\theta}(a_t|s_t)/\pi_{\theta}\omega(a_t|s_t) \) away from 1. The key idea is to use probability clipping, as follows:

\[
\max_{\theta} \mathbb{E}_t \left[ \min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1 - \varepsilon, 1 + \varepsilon)A_t) \right]. \quad (20)
\]

The main idea of PPO is to modify the objective by clipping the probability ratio. This removes the incentive for moving \( r_t \) outside of the interval \([1 - \varepsilon, 1 + \varepsilon]\). With this clipping method, the lower bound of objective function is maximized. See [17] for more details. For implementation, we use Scalable Reinforcement Learning: RLlib framework [18].

4) Reward: Recall the reward function that is defined in [9]. Given the spatially discretized states and control action in the ARZ model, the immediate reward \( r_t \) is defined by the Euclidean norm of the states

\[
r_t(s_t, a_t) = -\left[ \sum_i \rho(i \cdot \Delta x, t) - \rho^* \right]^2 - \left[ \sum_i v(i \cdot \Delta x, t) - v^* \right]^2.
\]

(21)

The control objective is to achieve regulation of the traffic states to a spatially uniform density and velocity.

C. Baseline: Open-loop Control

For the baseline case, we consider a constant incoming flow and constant outgoing velocity. When there is no actuation at the outlet, the boundary conditions are

\[
\rho(0, t)v(0, t) = q^*, \quad \rho(L, t)v(L, t) = q^*.
\]

(22)

(23)

where \( q^* = \rho^*v^*, v^* = V(\rho^*) \).

D. Benchmark: PDE Backstepping Control

As a benchmark, we consider the full-state feedback control law developed in [7] for a ARZ model. Note that the backstepping control law is designed for the linearized system, but is applied to the original nonlinear ARZ model. For initial conditions near the reference, the system is locally exponentially stable under outlet actuation. We summarize the PDE backstepping controller next:

The boundary condition at the inlet is

\[
q(0, t) = q^*.
\]

(24)

The boundary controller at the outlet, implemented via ramp-metering, is given by:

\[
\tilde{v}(L, t) = \int_0^L M(L - \xi) (v(\xi, t) - v^*) d\xi.
\]

(25)

\[
+ \frac{\lambda_2}{L} \int_0^L K(L, \xi) \exp\left(\frac{\xi}{\tau v^*}\right) (v(\xi, t) - v^*) d\xi
\]

\[
+ \frac{\lambda_1 - \lambda_2}{\rho^*} \int_0^L K(L, \xi) \exp\left(\frac{\xi}{\tau v^*}\right) (\rho(\xi, t) - \rho^*) d\xi,
\]

where \( U_{\text{out}}(t) = \rho^* \tilde{v}(L, t) \) represents the controlled flux at the outlet-side on-ramp. The control kernels \( K(L, \xi), M(L - \xi) \) are obtained by solving the following equations

\[
\lambda_2 K_x + \lambda_1 K_\xi = -c(\xi)K(x - \xi, 0),
\]

(26)

\[
K(x, x) = -\frac{c(\xi)}{\lambda_1 - \lambda_2},
\]

(27)

\[
M(x) = -K(x, 0).
\]

(28)

where the kernel variables \( K(x, \xi) \) evolve in the triangular domain \( T = \{(x, \xi) : 0 \leq \xi \leq x \leq 1\} \) and the spatial function \( c(\xi) \) is defined as

\[
c(\xi) = -\frac{1}{\tau} \exp\left(\frac{-\xi}{\tau v^*}\right).
\]

(29)

The full-state feedback controller (25) designed with the linearized ARZ model guarantees that the state variables \( (\rho(x, t), v(x, t)) \) are regulated to the reference system \((\rho^*, v^*)\) in the spatial \( L^2 \) norm

\[
||\rho(x, t) - \rho^*|| \rightarrow 0,
\]

(30)

\[
||v(x, t) - v^*|| \rightarrow 0,
\]

(31)

where \( ||u(x, t)|| \) for \( x \in [0, L] \) is denoted as \( ||u(x, t)|| = \left(\int_0^L u^2(x, t) dx\right)^{1/2} \). The convergence to the reference system is reached in the finite-time \( t_f \), where

\[
t_f = \frac{L}{|\lambda_1|} + \frac{L}{|\lambda_2|}.
\]

(32)

The following conclusion can be drawn for the closed-loop system, and will be illustrated with simulation later.

Theorem 1 ([7]). Consider system (1)-(2) linearized around \([5]\) with characteristics \([6, 7]\) and the control law (25). The equilibrium \( \rho(x, t) = \rho^*, v(x, t) = v^* \) of the linearized system is exponentially stable in the \( L^2 \) sense and the equilibrium is reached in finite time \( t = t_f \) given in (32).

IV. SIMULATIONS AND COMPARATIVE ANALYSIS

In this section, we numerically test the control designs with simulation and compare the open-loop system, PDE backstepping controller, and RL control policy. First, we prepare the ARZ model for numerical implementation.

The in-homogeneous nonlinear ARZ model written in conservative form is given by

\[
\rho_{t} + (\rho v)_{x} = 0,
\]

(33)

\[
y_{t} + (y v)_{x} = -\frac{y}{\tau},
\]

(34)

where \( \rho \) and \( y \) are conservative variables, and \( y \) is defined as

\[
y = \rho(v - V(\rho)).
\]

(35)

We apply the Lax-wendroff scheme on the \((\rho, y)\) system with spatial step \( \Delta x \) and time step \( \Delta t \). The CFL condition requires us to select step sizes such that:

\[
c = \max|\lambda_{1, 2}| \leq \frac{\Delta x}{\Delta t}.
\]

(36)

Note that in simulation, we need to specify \( \rho(\cdot, t) \) and \( y(\cdot, t) \) at the respective boundaries, which depend on the direction of characteristics. We assume sinusoidal initial conditions:

\[
\rho(x, 0) = 0.1 \sin\left(\frac{3\pi x}{L}\right) \rho^* + \rho^*.
\]

(37)
In Fig. 2 the initial condition is highlighted in blue. The boundary control, located at the outlet, is highlighted in red. In other words, the red curves visualize the control inputs for the baseline, benchmark and RL controllers.

The evolution of density and velocity are shown for the baseline open-loop controller case in Fig. 2. Oscillations persist for 2 min, although they appear lightly damped. The benchmark backstepping controller is shown in Fig. 3. The density and velocity converge to steady states values after 75 s. In Fig. 4, the RL policy regulates the states to the references values after about 80 s. Since the actuated command is generated from a Gaussian distribution, we see high-frequency noise in the velocity state. This may be an issue for real-world implementation. However, the stochasticity can be reduced with low-pass filters, or by simply applying the mean of the distribution.

An interesting finding from comparing the RL and PDE backstepping control algorithms is that RL learns a policy which produces a control input (red line in Fig. 4) that closely replicates the backstepping control input (red line in Fig. 3). The RL policy is developed without explicit knowledge of the differential equations and parameters. Instead, it is trained iteratively on the nonlinear simulation model. In contrast, the PDE backstepping state feedback control law is obtained by rigorous theoretical control design assuming perfect knowledge of the model. Interestingly, both methods yield similar control input trajectories.

In Fig. 5, instantaneous rewards (21) for the baseline, benchmark and RL controller are shown. We can see that the benchmark PDE backstepping controller converges to zero after 75 s, as suggested by Theorem 1 derived in [7]. The reward of the baseline open-loop controller tends toward zero over time, but does not reach there within 120 sec. The RL algorithm converges to $-0.1$ in 120 sec, nearly recovering the backstepping controller performance. We come to the conclusion that RL outperforms the baseline, and nearly recovers the PDE backstepping method. Note that the RL
result shown here is trained through multiple simulation episodes, and the reward trajectory in Fig. 5 corresponds to the policy parameters computed after the cumulative reward has converged.

In Fig. 6 we plot the so-called “learning curve” of the RL policy, as a function of iterations. The performance is measured by the cumulative reward across each episode. The distribution of rewards is visualized by the shaded region, since the policy is stochastic. As we can see, the expected reward increases monotonically, as suggested by the theoretical analysis in [16]. After roughly 200 iterations of training, the RL policy is approximately matches the backstepping PDE controller.

V. Conclusion

In this work, we develop a proximal policy optimization (PPO) reinforcement learning (RL) algorithm to stabilize oscillations on a freeway segment via on-ramp metering control. The traffic dynamics are governed by the Aw-Rascle-Zhang (ARZ) PDE model. The control objective is to achieve $L^2$ stabilization of the traffic density and velocity to spatially uniform steady state values. The RL controller is compared against an open-loop “baseline” controller and a closed-loop PDE backstepping “benchmark” controller. Despite the complex nonlinear infinite-dimensional dynamics, the RL policy is able to nearly recover the PDE backstepping controller’s performance. In this case, we examined a single freeway segment with fixed parameter values. In future work, we are interested in examining freeway networks with uncertain parameters, using multi-agent RL control.

Acknowledgement

Huan Yu and Saehong Park contribute equally to this work. We would like to thank Eugene Vinitsky and Aboudy Kreidieh for their insights and help in establishing this work.

References

[1] Flynn, M. R., Kasimov, A. R., Nave, J. C., Rosales, R. R., Seibold, B. (2009). Self-sustained nonlinear waves in traffic flow. Physical Review E, 79(5), 056113.
[2] Seibold, B., Flynn, M. R., Kasimov, A. R., Rosales, R. R. (2012). Constructing set-valued fundamental diagrams from jamiton solutions in second order traffic models. arXiv preprint arXiv:1204.5510.
[3] Fan, S., Herty, M., & Seibold, B. (2013). Comparative model accuracy of a data-fitted generalized Aw-Rascle-Zhang model. arXiv preprint arXiv:1310.8219.
[4] Belletti, F., Hua, M., Litrico, X., & Bayen, A. M. (2015). Prediction of traffic convective instability with spectral analysis of the Aw-RascleZhang model. Physics Letters A, 379(38), 2319-2330.
[5] Aw, A., & Rascle, M. (2000). Resurrection of “second order” models of traffic flow. SIAM Journal on Applied Mathematics, 60(3), 916-938.
[6] Zhang, H. M. (2002). A non-equilibrium traffic model devoid of gas-like behavior. Transportation Research Part B: Methodological, 36(3), 275-290.
[7] Yu, H., & Krstic, M. (2019). Traffic congestion control for AwRascleZhang model. Automatica, 100, 38-51.
[8] Yu, H., & Krstic, M. (2018, November). Varying Speed Limit Control of Aw-Rascle-Zhang Traffic Model. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC) (pp. 1846-1851). IEEE.
[9] Yu, H., & Krstic, M. (2018, December). Traffic Congestion Control on Two-lane Aw-Rascle-Zhang Model. In 2018 IEEE Conference on Decision and Control (CDC) (pp. 2144-2149). IEEE.
[10] Zhang, L., & Prieur, C. (2017). Necessary and Sufficient Conditions on the Exponential Stability of Positive Hyperbolic Systems. IEEE Transactions on Automatic Control.
[11] Karafyllis, I., Bekiaris-Liberis, N., & Papageorgiou, M. (2017). Analysis and Control of a Non-Standard Hyperbolic PDE Traffic Flow Model. arXiv preprint arXiv:1707.02209.
[12] El-Tantawy, Samah, Baher Abdullhia, and Hossam Abdelgawad. "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown Toronto." IEEE Transactions on Intelligent Transportation Systems 14.3 (2013): 1140-1150.
[13] Belletti, Francois, et al. "Expert level control of ramp metering based on multi-task deep reinforcement learning." IEEE Transactions on Intelligent Transportation Systems 19.4 (2018): 1198-1207.
[14] Fares, Ahmed, and Walid Gomaa. "Freeway ramp-metering control based on reinforcement learning." Control & Automation (ICCA), 11th IEEE International Conference on. IEEE, 2014.
[15] Kakade, Sham and Langford, John. "Approximately optimal approximate reinforcement learning" In 2002 International Conference on Machine Learning (ICML) (pp. 267-274). IEEE.
[16] Schulman, John and Levine, Sergey and Abbeel, Pieter and Jordan, Michael I and Moritz, Philipp. "Trust Region Policy Optimization." In 2015 International Conference on Machine Learning (ICML) (pp. 1889-1897). IEEE.
[17] Schulman, John and Wolski, Filip and Dhariwal, Prafulla and Radford, Alec and Klimov, Oleg. "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347.
[18] Liang, Eric and Liu, Richard and Moritz, Philipp and Nishihara, Robert and Fox, Roy and Goldberg, Ken and Gonzalez, Joseph E and Jordan, Michael I and Stoica, Ion. "RLlib: Abstractions for distributed reinforcement learning." In 2018 International Conference on Machine Learning (ICML) (pp. 1711-1719). IEEE.
[19] Lax, Peter and Wendroff, Burton. (1960). "Systems of conservation laws". Communications on Pure and Applied Mathematics 13(2), (pp. 217–237).