Deep Reinforcement Learning for Autonomous Vehicles: Improving Traffic Flow in Mixed-Autonomy

Nathan Lichtle

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-66
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-66.html

May 9, 2024

Acknowledgement

I want to express my sincere gratitude to my advisor, Professor Alex Bayen, and my primary co-author, Eugene Vinitsky, for their unwavering support and mentorship across this and many other projects. I also deeply appreciate my friends and collaborators in our research lab, who have made my time at Berkeley truly memorable. Special thanks go to George Gunter and Akash Velu for their valuable contributions to this work, part of which was previously published at the IEEE International Conference on Intelligent Transportation Systems (IEEE ITSC). I'm grateful to the Savio institutional cluster at UC Berkeley for providing the computational resources that were essential to this research.

University of California, Berkeley

Spring 2024

Deep Reinforcement Learning for Autonomous Vehicles: Improving Traffic Flow in Mixed-Autonomy

Nathan Lichtlé

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of Master of Science, Plan II.

Approval for the Report and Comprehensive Examination:

Committee

Professor Alexandre Bayen
Research Advisor

May 7, 2024
Date

Signature

Professor Maria Laura Delle Monache
Second Reader

5/9/2024
Date

Signature

Abstract

In this work, we optimize fuel consumption in a large, calibrated traffic model of a portion of the Ventury Freeway (Interstate 210, near Los Angeles, California) by leveraging a low proportion of autonomous vehicles controlled by reinforcement learning algorithms. We specifically target stop-and-go waves, a phenomenon characterized by alternating acceleration and braking, which is widespread on real-world highways and significantly detrimental to fuel efficiency. In order to simulate these dynamics accurately, we introduce waves into the network using a string-unstable car-following model, as well as a ghost cell to enable wave propagation beyond the network boundary. Using multi-agent reinforcement learning, we develop a decentralized controller that effectively mitigates instabilities and partially dampens these waves, resulting in a significant 25% reduction in fuel consumption with only a 10% penetration rate of autonomous vehicles. We then investigate the designed controller's robustness by testing it under various conditions. Our results show that it maintains equilibrium speeds across a wide range of wave speeds and penetration rates far outside of the training regime, demonstrating its generalization and robustness.

Contents

Acknowledgment

I want to express my sincere gratitude to my advisor, Professor Alex Bayen, and my primary co-author, Eugene Vinitsky, for their unwavering support and mentorship across this and many other projects. I also deeply appreciate my friends and collaborators in our research lab, who have made my time at Berkeley truly memorable. Special thanks go to George Gunter and Akash Velu for their valuable contributions to this work, part of which was previously published at the *IEEE International Conference on Intelligent Transportation Systems* (IEEE ITSC) [1]. I'm grateful to the Savio institutional cluster at UC Berkeley for providing the computational resources that were essential to this research.

Chapter 1

Introduction

Adaptive, hands-free cruise control, in which a partially automated vehicle (AV) keeps a safe distance from a lead car is increasingly ubiquitous as cheap radars and advanced computer vision technology combine to make such systems inexpensive and safe to deploy. While not fully autonomous, these level-two systems (autonomous distance and lane-keeping), have low reaction times and can be programmed to achieve many of the potential gains associated with full-autonomy such as vehicle platooning [2] or close-following to minimize air resistance [3]. In particular, recent work has demonstrated that there are significant gains in traffic flow and energy efficiency to be had even in the low penetration rate regime e.g. 0-10% [4, 5], a regime often referred to as *mixed-autonomy traffic*. Given the widespread deployment of these automated driving systems, there is an opportunity to design and deploy cruise controllers that improve the energy impact of our transportation systems.

However, designing controllers and analyzing their energy impact is difficult due to the complexity of traffic: non-linear driving dynamics, lane changes, merges, etc. Hence, controllers are often designed and analyzed in simple settings whose relationship to actual highway networks is not entirely clear. For example, a significant fraction of recent traffic smoothing controllers are designed and analyzed with respect to a closed circular ring of dense traffic, a setting in which energy-consuming waves form spontaneously and persist throughout the network [6]. While this network is amenable to analysis and can model a single lane of traffic as it becomes infinitely long, the simplicity of the network makes it unclear how controllers designed in these settings will perform as complexity increases. Furthermore, these simple systems often have pernicious optimal solutions like slowing to a stop and gradually accelerating up to the equilibrium speed of the ring.

In this work, we focus on developing robust, traffic smoothing controllers for a system containing both traffic waves as well as lane changes. We build a multi-lane model (shown in Fig. 1.1) of a section of the Ventura Freeway in Los Angeles containing both on-ramps and lane drops. This system contains approximately one thousand vehicles and stretches about one mile, allowing us to see any possible long-range interactions between AVs, waves, and lane changing behavior. Using on-policy multi-agent reinforcement learning, we design traffic smoothing controllers that create a sharp increase in the energy efficiency of the traffic flow; these controllers also outperform a variety of available baseline controllers. The state input to our controller is easily implementable using radar or cameras, making it an easy add on to existing cruise controllers.

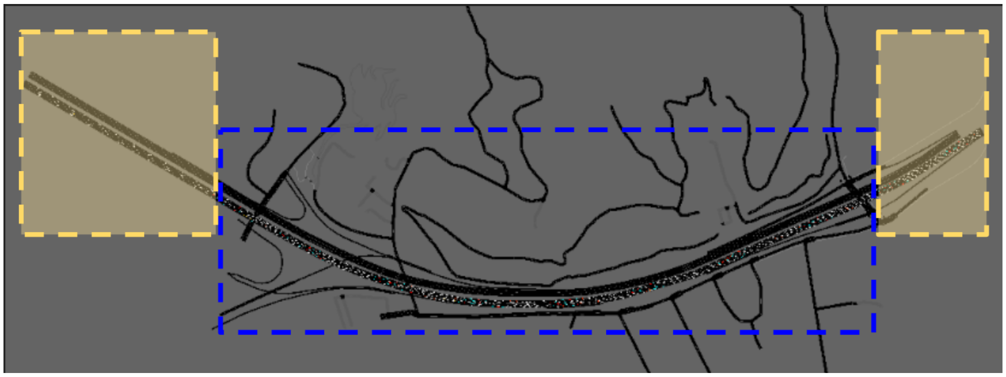

Figure 1.1: The I-210 network simulated within SUMO. Yellow areas represent the uncontrolled ghost cells, while the blue rectangle shows where control is applied and where metrics are computed.

Since it is highly likely that our simulator is not accurate in a variety of ways (imperfect model of human driving dynamics, vehicle dynamics, etc.), we demonstrate that our controller is a good candidate for deployment by performing a set of robustness tests. We sweep over a wide variety of the parameters that define driving behavior and system dynamics in our simulator and show that our controller maintains good performance under these changes. As we demonstrate, our controller appears robust to all these axes of variation.

The contributions we include in this work are as follows:

- We build and release a new, large-scale traffic network for investigating the effect and potential of traffic-smoothing autonomous vehicles.
- We use multi-agent reinforcement learning to construct controllers that sharply improve the energy efficiency of highway traffic. We demonstrate that our controllers generalize outside their training distribution and act like *controllers that know the equilibrium speed of the system*.
- We perform a variety of robustness checks and demonstrate that our controller is robust to a wide range of potential driving conditions.

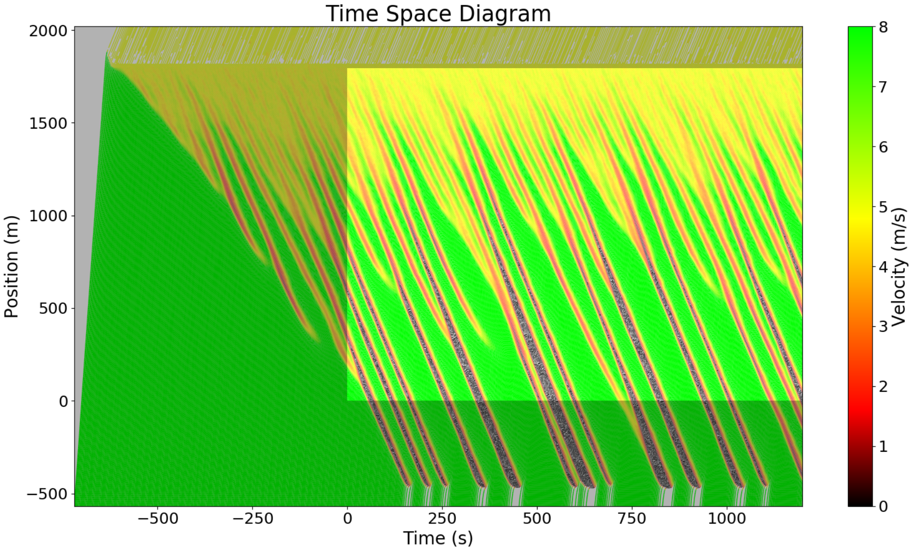

Figure 1.2: Time-space diagram of one lane of the I-210, showing the average velocity of the network as a function of time and position. The shaded areas correspond to the warm-up period and the ghost cells, and represent times and positions that are not considered in control or evaluation. Waves are visible as the downwards-sloping black lines.

Chapter 2

Related Work

In the seminal work of [6] it was experimentally shown for the first time that traffic streams can exhibit what are known as 'phantom-jams' in which a moving traffic jam can form without any outside prompts, such as lane-reductions or accidents. Following work in [5] empirically showed that under the same setup the phantom-jams that form could be effectively dissipated using a single automated vehicle running a control algorithm. In addition to increasing the average speed, and reducing the speed variance of the system, a significant increase in fuel efficiency for the vehicles was also found [7, 8]. These result established the potential of mixed-autonomy control to improve throughput and energy efficiency in relatively simplified settings.

Extensive work has been done to find new mechanisms by which mixed-autonomy can be used to improve transportation systems. [9, 10], consider the use of platoons of autonomous vehicles operating as moving bottlenecks to both dampen stop-and-go waves and minimize the effects of capacity drop. Other works consider the potential of vehicle-to-infrastructure coordination as a tool for eco-driving, a concept in which a controlled vehicle modifies its speed and acceleration profile to realize energy gains. [11] demonstrates the deployment from simulation to the roadway of coordination between a vehicle and a signalized intersection and shows marked improvements in energy efficiency albeit at the cost of travel time. [12] demonstrates in a physical experiment with ghost cars that a CAV using prediction of the lead vehicle trajectory or communicating with AVs further ahead in the string can sharply improve the energy efficacy of a drive.

Recently, many controllers for mixed autonomy settings have been generated using techniques from Reinforcement Learning. Reinforcement learning has been used in [4] to demonstrate that a vehicle equipped with memory could equilibrate the system for a wide range of ring densities. Other works have focused on the potential of reinforcement learning to improve traffic at scale. In [13, 14, 15, 16], multi-agent RL was used to optimize merges in a fully decentralized fashion. Both [17] and [18] concurrently used decentralized multi-agent RL for optimizing a scaled model of the San Francisco-Oakland Bay Bridge. Reinforcement learning has also seen significant use in traffic light pattern optimization [19, 20] as well as to develop traffic light controllers that could quickly adapt to new settings using Meta-RL [21].

Chapter 3

Problem Formulation: Smoothing in Multi-Lane Systems

Our goal is to study traffic smoothing in a setting with large numbers of vehicles, ubiquitous lane changes and multiple possible sources of onset mechanisms for wave formation. We choose a segment of the Ventura Freeway, or Interstate 210 (I-210), in California. This segment is approximately one mile long and can hold up to 2000 vehicles. It varies between five and six lanes over its length and has an on-ramp that can serve as a possible source of congestion formation; however, this on-ramp is disabled in this work since we use a different mechanism to generate congestion, as explained in Sec. 3.1. Due to the combination of the multi-lane nature and its high capacity, this network serves as an effective testbed for the complexity of realistic wave smoothing.

The challenge in this network is to improve the energy efficiency by eliminating traffic shockwaves that occur along this system, which we will refer to as *phantom jams*. These shockwaves are known to appear in real systems [22] and decrease the energy efficiency of travel by leading to patterns of braking and acceleration. By eliminating the phantom jams, we improve the energy efficiency of the system. As we will demonstrate, an interesting feature of these phantom jams is that they can be removed with minimal effect on the traffic flow: decreasing the fuel consumption of the roadway is something that can be achieved without any trade-offs on the system throughput.

3.1 I-210 model with phantom jams

The I-210 network has been imported from Open Street Maps into the microscopic traffic simulator SUMO [23]. The network is shown in Fig. 1.1. Traditionally, traffic congestion is hypothesized to be caused by 'bottlenecks' in which a road network cannot support as much flow through a downstream section as is being sent from the upstream. This discrepancy in capacity to receive compared to amount sent subsequently creates traffic congestion in which vehicles are forced to drive closer together and at a lower speed than they would otherwise.

One of the benefits of the ring-road as a well-posed traffic simulation environment is that congested regimes can be set directly by choosing a number of vehicles for a given ring length (i.e. the density is set directly). However, the ring lacks crucial components of realistic traffic such as lane-changing and routing choice. In order to allow for such traffic maneuvers the multi-lane, multi-edge, network present in the I-210 network is used. In addition, a subsequent downstream flow condition is imposed directly in the form of a decreased speed limit along a small portion of the end of the network. We refer to this speed limit as the *downstream speed*. By doing so, the congested regime for the traffic can be set in a very similar manner to the ring road. This downstream condition can then be varied to allow more or less flow through the end of the network, which allows for testing the proposed control framework across a number of traffic regimes.

3.2 Human controllers

An important component of micro-simulation is the *car-following* and *lane-changing* logic that individual vehicles in the simulation adhere to. Car-following refers to how vehicles manage their longitudinal motion within a lane as opposed to their lateral, lane-changing behavior.

For the lane-changing logic, we use the default model provided in SUMO [23], the traffic micro-simulator that we use. The dominant cause of lane-changing in this model mostly consists of a vehicle lane-changing for speed gain, i.e. it will lane-change if it can drive faster in the other lane.

As for the car-following logic, it is generally modeled as *ordinary differential equations* that dictate an ego vehicle's motion based on the state of the vehicle ahead of it. In this work a first-order discretization of the *Intelligent Driver Model* (IDM) [24] is used, which dictates a vehicle's longitudinal acceleration, and is of the form:

$$\begin{aligned} v_{t+1} &= v_t + \Delta t \times a \left[1 - \left(\frac{v_t}{v_0} \right)^{\delta} - \left(\frac{s^*\left(v_t, \Delta v_t\right)}{s_t} \right)^2 \right] \\ &\quad + \sqrt{\Delta t}\, \mathcal{N}\left(0, \sigma\right) \end{aligned} \tag{3.1}$$

with

$$\begin{aligned} s^*\left(v_t, \Delta v_t\right) &= s_0 + v_t T + \frac{\max\{0, v_t \Delta v_t\}}{2\sqrt{ab}} \\ s_{t+1} &= s_t + \Delta t \Delta v_t \end{aligned} \tag{3.2}$$

where v_t is the ego vehicle speed at time t, Δv_t is the difference between the leading vehicle's speed and the ego vehicle's speed, s_t is the distance to the lead vehicle, and a, b, s_0, T, v_0 and δ are all parameters of the model. t indexes the time-step and Δt refers to the size of the simulation step. $\mathcal{N}\left(0, \sigma\right)$ is zero-mean Gaussian noise used to perturb the accelerations at each time-step and is intended to represent both aleatoric and epistemic noise.

For this model, we select values of a, b such that the resultant dynamics are *string-unstable*. When a car-following model is string-unstable [25], small disturbances can grow in magnitude into large disturbances that propagate along a string of vehicles, allowing the *phantom jam* to propagate rather than dissipate. In this work, the criteria by which a and b are set is drawn from [26] due to their simplicity of the polynomial condition therein for determining string instability. All other parameters as chosen as being the default IDM values specified in [24].

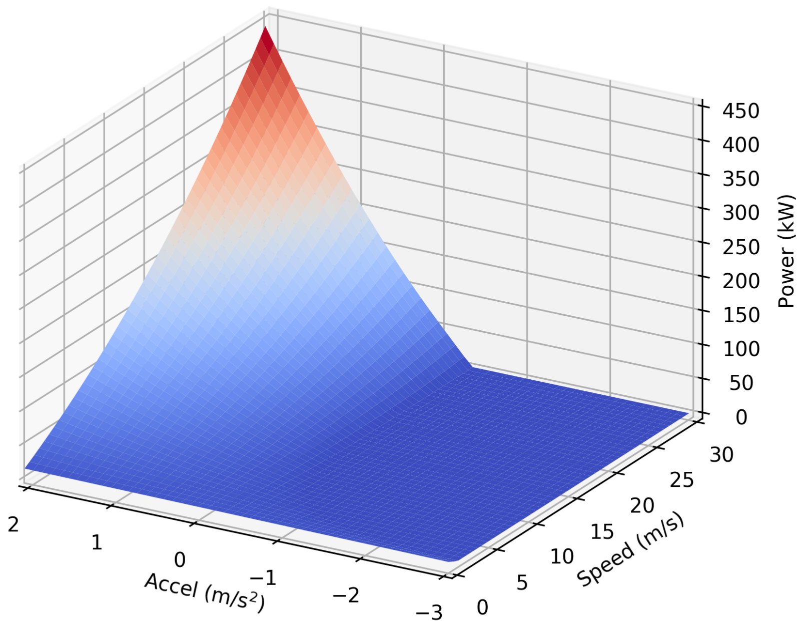

Figure 3.1: Polynomial fit of power consumption as a function of velocity and instantaenous acceleration from a Midsize Sedan model provided by Toyota. This model was made for a vehicle of mass 1743kg, and we assume a constant road grade of 0. We use the conversion 1 gallon/hour = 33430 watt.

Finally, since discretized car-following models can lead to collisions, we clip the output acceleration values such that collisions are not possible. The condition we use for safety is maximally conservative: an acceleration is unsafe if the lead car, braking at its maximum deceleration, will unavoidably collide with the ego vehicle. The exact implementation of this condition can be found in [23]. On top of that safe velocity failsafe, we also clip the output acceleration to respect the road speed limit and the acceleration bounds of the vehicle.

3.3 Energy model

As our calibrated energy model, we use a polynomial fit to a black box model of a Midsize Sedan model provided by Toyota. The calibrated model is shown in Fig. 3.1, and is a function of the instantaneous speed and acceleration. This model assumes a constant vehicle mass of 1743 kg.

The model shown here is fitted with a third-order polynomial, effectively smoothing out the effects of gear shifting that might otherwise be present. We do not attempt to fit this as it would require excessively large polynomial coefficients and the particular positions of the jumps due to gear shifting will vary sharply from vehicle to vehicle. For the derivation, coefficients of the polynomials, and full model details see [27].

However, it is not obvious that optimizing the energy model we use will translate the heterogeneous traffic. We argue that minimizing the energy model is equivalent to regulating around an optimal speed. While that optimal speed will vary from engine model to engine model, as long as that optimal speed is greater than the downstream speed, the optimal behavior is irrespective of engine type. For the Midsize Sedan model, the optimal operating speed is around 16.7m/s, which is well above any congestion speeds that might occur. A survey of the energy models available in Autonomie [28] suggests that the optimal speed for most engines is above this value. Since the downstream speed sets a system speed limit, the optimal solution for most engines will consequently be elimination of the waves and so we expect our results to hold generally across different energy models.

3.4 Multi-agent reinforcement learning

In this section, we discuss the notation and describe in brief the key ideas used in reinforcement learning. The system described in this article solves tasks which conform to the standard structure of a finite-horizon, discounted, decentralized multi-agent POMDP (Dec-POMDP) [29], an abstraction in which groups of agents with partial access to the true world state seek to optimize a discounted reward function across time. The Dec-POMDP is defined by the tuple $(\mathcal{S}_0, \mathcal{A}_0, \mathcal{O}_0, r_0, \rho_0, \gamma_0, T_0) \times \cdots \times (\mathcal{S}_n, \mathcal{A}_n, \mathcal{O}_n, r_n, \rho_n, \gamma_n, T_n) \times \times P \times \mathcal{Z}$, where n is the number of agents, $\mathcal{S}_i$ is a (possibly infinite) set of states for agent i, $\mathcal{A}_i$ is a set of actions for agent i, $\mathcal{Z} : (\mathcal{S}_0 \times \mathcal{A}_0) \times \cdots \times (\mathcal{S}_n \times \mathcal{A}_n) \to (\mathcal{O}_0, \ldots, \mathcal{O}_n)$ is a function describing how the world state is mapped into the observations of the POMDP, $P : (\mathcal{S}_0 \times \mathcal{A}_0 \times \mathcal{S}_0) \times \cdots \times (\mathcal{S}_n \times \mathcal{A}_n \times \mathcal{S}_n) \to \mathbb{R}_{\geq 0}$ is the transition probability distribution for moving from one set of agent states s to the next set of states s' given the set of actions $(a_0, \ldots, a_n)$, $r_i : (\mathcal{S}_0 \times \mathcal{A}_0) \times \cdots \times (\mathcal{S}_n \times \mathcal{A}_n) \to \mathbb{R}$ is the reward function for agent i, $\rho_i : \mathcal{S}_i \to \mathbb{R}_{\geq 0}$ is the initial state distribution for agent i, $\gamma_i \in (0, 1]$ is the discount factor for agent i, and T_i is the horizon for agent i.

The goal for a given agent i is to find a controller π_i that optimizes

$$J^{\pi_i} = \mathbb{E}_{\rho_0,\ p(s_{t+1}|s_t, a_t)} \left[\sum_{t=0}^{T} \gamma^t r_t \mid \pi(a_t|s_t) \right] \tag{3.3}$$

where r_t^i is the reward of agent i at time t and the expectation is over the start state distribution, the probabilistic dynamics, and the probabilistic controller π.

Chapter 4

Controller design

4.1 Optimization criterion

Our goal is to reduce the average energy consumption of the system. However, the energy minimizing solution is for all vehicles to come to a full stop. To avoid this degenerate solution, we will impose the constraint that all vehicles exit the system. In this section, we describe how this constraint is converted into a reward function so that our desired optimized quantity can be used in a standard reinforcement learning procedure.

Let L be the length of the controlled portion of the network and $E(v_t, a_t)$ the instantaneous energy consumption at time-step t, v_t and a_t being the velocity and acceleration respectively. For notation simplicity, we will only consider the trajectory of one AV, as the reward for each AV is computed independently of the others, and we assume that the trajectory starts at time $t = 0$ and ends at time $t = H$.

Ideally, we would like to maximize the cumulative miles per gallon value for each AV

$$\frac{L}{\sum_{t=0}^{H} E\left(v_t, a_t\right)} \tag{4.1}$$

Unfortunately, that quantity cannot be computed until the end of a trajectory, making the reward sparse. Sparse rewards are generally difficult to optimize so we propose a simple heuristic that approximates this quantity.

We attempt to turn the sparse cumulative miles per gallon reward into a per-step reward by noticing that since L, is a constant, maximizing Eq. 4.1 is equivalent to maximizing $\sum_{t=0}^{H} -E(v_t, a_t)$ as long as energy consumption is positive. We can thus give the agent a reward $r(s_t, a_t) = -E(v_t, a_t)$ at time-step t.

However, the issue that the optimum consists in coming to a full stop will still persist here. Amongst options considered, we observed that giving the agent a semi-sparse reward for making forwards progress achieved the largest improvement in fuel efficiency.

$$r(s_t, a_t) = \begin{cases} -E(v_t, a_t) & \text{if } c_t < M \\ -E(v_t, a_t) + B & \text{if } c_t \geq M \end{cases} \tag{4.2}$$

Here c_t is a counter of the total distance that we have travelled since receiving the last bonus and B is a bonus for completing M meters. c_t is reset back to zero every M meters.

Essentially, every time the vehicle completes M meters, it receives a bonus for doing so. We can think of this as approximately distributing a penalty for failing to exit the network across the spatial extent of the network but we note that the exact equivalence to the cumulative miles per gallon objective (Eq. 4.1) is now lost.

Finally, since the goal is still to optimize the energy consumption for the whole system, we also add the energy consumption of the N vehicles following the AV to its reward function, which are the vehicles that it has the most impact on.

$$r(s_t, a_t) = -E(v_t^0, a_t^0) - \sum_{i=1}^{N} E(v_t^i, a_t^i) + B_t \tag{4.3}$$

with

$$B_t = \begin{cases} 0 & \text{if } c_t < M \\ B & \text{if } c_t >= M \end{cases}$$

where we index the velocities and accelerations of the vehicles, 0 being the AV, 1 the vehicle following it, and N its n^{th} follower. Although this requires non-local information at training time, the reward is not part of the controller state and thus the controller will still only rely on local information.

4.2 Dec-POMDP design

We focused on picking controller inputs that could tractably and easily be placed onto a vehicle equipped with standard level-2 technology such as forwards facing radar, cameras, and GPS. The careful design of the state space is essential as the state space choice will have strong consequences for the generalization capabilities of the agents. As an example, consider an agent that has GPS coordinates as part of its input. This agent now has two potential generalization failure modes: 1) it may use the GPS position to block the network entrance and artificially reduce the inflow 2) it will adjust its behavior to perfectly optimize the particular network architecture that the agent is trained in and may be less likely work for different road network architectures.

Based on the criteria of maximizing likely generalization, we adopt the following Dec-POMDP:

- State space / Observation function: $[v, h, v_{\text{lead}}, c]$ where v is the ego speed, h is the distance to the leader, v_{lead} is the speed of the vehicle directly in front of the AV, and c is the distance travelled which is reset every m meters. This state space can be used in arbitrary networks and allows us to easily transfer learnt controllers between different network architectures. It is also easily implemented with radar and cameras.

- Action space: accelerations bounded between $[-2.6, 4.5]$. We do not allow the AVs to lane change.

- The reward function is described in Sec. 4.1.

4.3 Algorithm / Controller

As our training algorithm, we use Independent Proximal Policy Optimization [30], a ubiquitous policy gradient algorithm. All agents are homogeneous, that is, there is one controller that is duplicated across all agents although actions are still computed locally. The controller is a two layer fully connected neural network with 64 hidden units at each layer and a hyperbolic tangent non-linearity.

We make one small modification to the standard PPO algorithm and provide the total distance traveled by the agent at time t as an input to the value function. The value function is used exclusively during training for variance reduction (see [30] for details) and so non-local information can be used. The value V^π function estimates the reward-to-go from a given state s_t

$$V^\pi(s_t) = \mathbb{E}\left[\sum_{j=t}^{T} \gamma^i r(s_j, a_j) | s_j\right] \tag{4.4}$$

and since the reward-to-go strongly depends on the total distance remaining to the exit, it is difficult to estimate without this information.

4.4 Experimental setup

We ran the reinforcement learning training using the PPO implementation provided in RLlib [31][1] version 2.0.0.dev0, a distributed deep RL library. We use a learning rate of $3 \cdot 10^{-4}$, a training batch size and SGD minibatch size both equal to 500000, a number of SGD iterations of 5 and run the simulations for 220 iterations, each with 19 workers running in parallel with a horizon of 500 environment steps. Importantly, we set `multiagent/count_steps_by` to `agent_steps` so that steps are counted by agent step and not environment step. We also set `batch_mode` to `complete_episodes`, `gamma` to 0.995, `lambda` to 0.97 and `kl_target` to 0.02. The other RLlib and PPO parameters are left to their default value. Training for 220 iterations took about 2 days, running on a machine with 20 Intel Xeon E5-2670 v2 CPUs. The evolution of the reward function during training can be seen in Fig. 4.1.

Both the controller (policy network) and the value function network are feedforward neural networks (MLP) with two fully-connected hidden layers of size 64, and tanh activations. For our reward function, we used parameters $M = 50$m, $N = 5$ vehicles and $B = 2.5$.

We use the traffic micro-simulator SUMO [23] for running our simulations. To populate the simulation fully with vehicles, we allow a warmup period of 720 seconds during which the experiment runs uncontrolled after which 10% of the vehicles are turned into AVs. We keep a fixed inflow of 2050 vehicles per hour over the whole horizon. 90% of these vehicles are humans with an IDM controller, and the remaining 10% are AVs. The downstream speed limit is fixed to 5m/s. The IDM controller is used with parameters $a = 1.3$, $b = 2$, $v_0 = 30$, $T = 1$, $\delta = 4$ and $s_0 = 2$. Finally, taking note that the standard benchmark for ATARI games repeats each action four times [32], agent actions are actually sampled once for every 3 time-steps and the same action is applied for all 3 time-steps. We use an individual time step

[1] https://github.com/ray-project/ray/python/ray/rllib

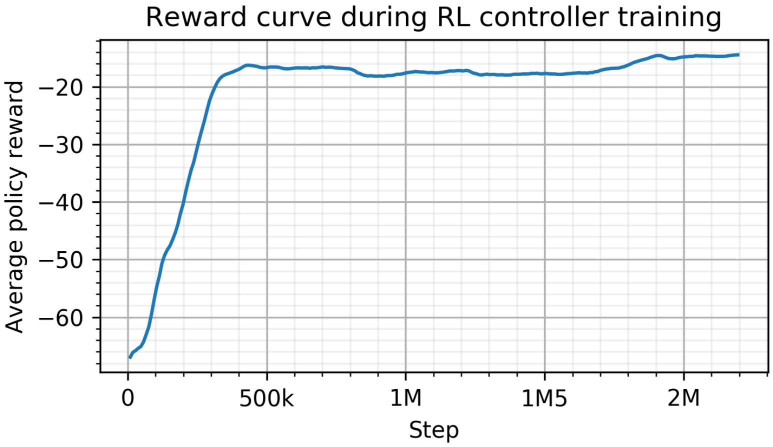

Figure 4.1: Evolution of the reward for an agent over two days of training; the reward depicted here is averaged across all the agents. The policy is close to converged approximately 25% of the way into training. Note that this represents the average sum of all rewards received by an agent, and not the discounted return.

size of 0.4s. This means that a horizon of 500 environment steps will run for 10 simulated minutes.

Chapter 5

Results

5.1 Evaluation procedure

In this section we evaluate the performance of our trained controller and perform sweeps around its training distribution in order to assess its generalization capabilities. For each sweep value, we run from 30 to 60 simulations using that sweep value and compute the mean and standard deviation of the results obtained during each rollout, both of which are shown in the plots below. Metrics are computed from averaging data collected over all vehicles (or AVs) post warm-up time, except those located in the ghost cells (see Fig. 1.1).

We benchmark our RL controller against the FollowerStopper (FS) [5], a control algorithm that achieved wave smoothing in a physical experiment. FS aims to drive at exactly a desired velocity v_{des} whenever safe (*i.e.*, as in a standard cruise controller), but will command a suitable lower velocity $v_{\text{cmd}} < v_{\text{des}}$ whenever safety requires. Importantly, it attempts to smoothly transition between those objectives. We keep the desired speed constant over each simulation, and use the same hyperparameters for other portions of the controller as in [5].

We compare the results obtained by our RL controller to the uncontrolled human baseline where all vehicles are IDM, to the FS controller (with different desired speeds v_{des}), and to a variant of the FS that has v_{des} set to the downstream speed. We refer to the latter as *cheating* FS as the downstream speed is non-local information that would not be available using on-board sensors; external infrastructure would be needed to observe the downstream speed.

5.2 Controller performance and robustness

Figure 5.1 shows the effect of the introduction of the RL controller on the time-space diagram of the system. Without control, at speeds of both 3 and 5 m/s, waves are visible as dark lines slopping from top-left to bottom-right. When RL control is introduced, the waves become markedly lessin number and occasionally completely dissipate. Gaps formed by the RL agents can be seen as white lines sloping from bottom-left to top-right. These gaps terminate near the boundaries of waves as they dissipate the wave and are consumed in the process of doing so.

Figure 5.2 examines the effects of the wave reduction on the average fuel efficiency (in

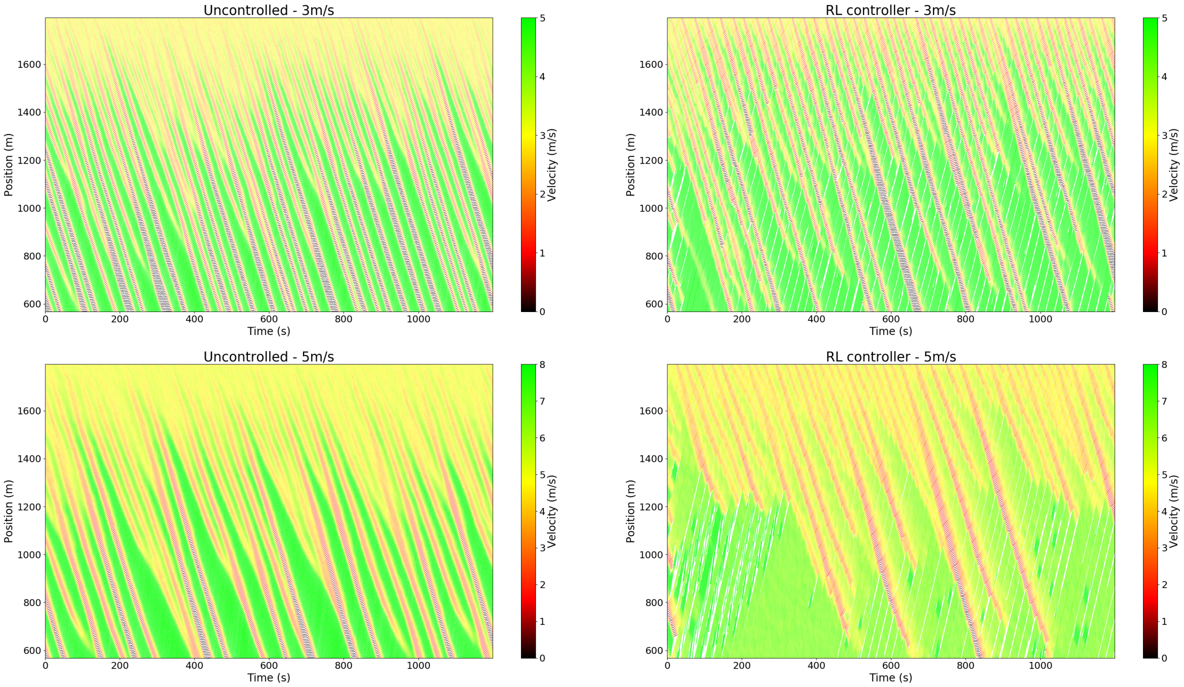

Figure 5.1: **Left:** time-space diagram of the I-210 simulation with no control applied, for a downstream speed of 3m/s (top) and 5m/s (bottom). **Right:** time-space diagram of the I-210 simulation when 10% of vehicles are AVs using our RL controller, for a downstream speed of 3m/s (top) and 5m/s (bottom). Time-space diagrams show the (average) vehicles velocities as a function of their position on the highway and simulation time. Ghost edges and warm-up time are not shown in these graphs.

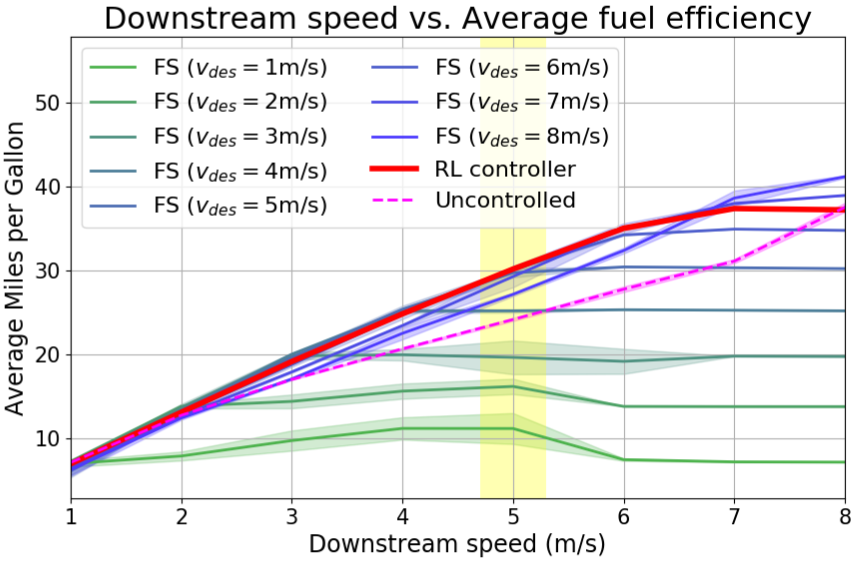

Figure 5.2: Average fuel efficiency of the RL controller on its training downstream speed of 5m/s (highlighted in yellow), and generalization to speeds outside that range. Miles per gallon fuel consumption is also shown for the FS controller with desired speed ranging from 1m/s to 8m/s and for the uncontrolled human baseline, as a function of the downstream speed. All plots are computed using a fixed penetration rate of 10% and using the energy model presented in Sec. 3.3.

Miles per Gallon i.e. mpg) of the system. For a fair comparison, we sweep the desired velocity of the FS controller over all possible values of the downstream speed. As can be observed, the RL controller improves markedly on the fuel efficiency of the system and achieves the best performance of each of the FS controllers up until 7 m/s. Essentially, up until 7 m/s, the RL controller acts almost as effectively as a controller that *knows what the downstream speed is*. Figure 5.2 shows these same results as a percentage improvement over the uncontrolled baseline. Here the cheating FS is added as an additional baseline, showing the close match in performance between the RL and the non-local controller.

Finally, we investigate potential robustness issues with our controller. Figure 5.4 provides a sanity check that the improvement in fuel consumption does not come at the cost of reduced outflow up until 7 m/s. As this reduction in outflow is potentially undesirable, the controller could be switched off around this boundary. Additionally, in Figure 5.5, we investigate the effects of changing penetration rate on the controller. Our controller is trained at a fixed penetration of 10%, shown as the yellow line in the Figure. Since at any time, randomness could cause the penetration rate to vary from this value, it is important that controller performance be preserved away from the training regime. The RL controller performance, shown in th red, indicates that performance improvements are maintained outside of the training distribution, with values close to 10% performing almost identically. Additionally, there is generalization outside of this value and energy improvements are seen at all penetrations.

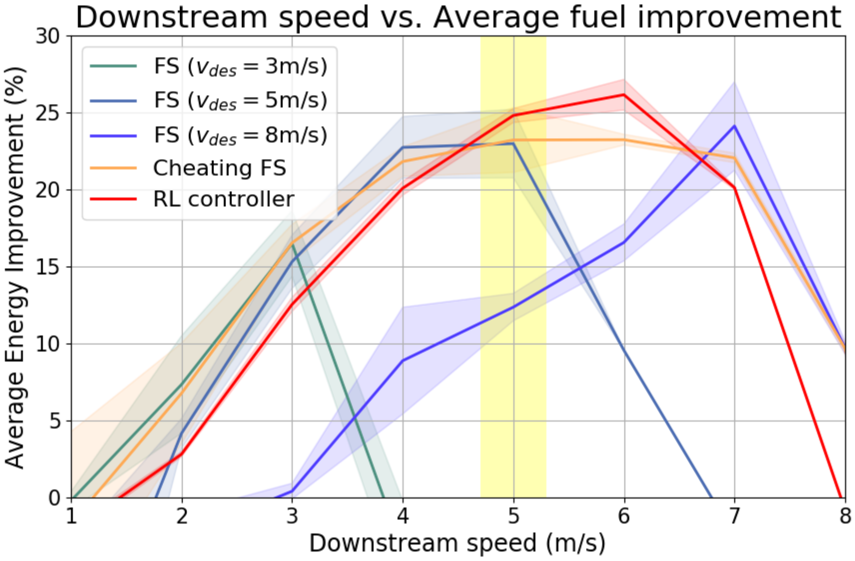

Figure 5.3: Fuel efficiency improvement of the RL controller on its training downstream speed of 5m/s (highlighted in yellow) over the uncontrolled human baseline, and generalization to speeds outside that range. Fuel improvement is also shown for the FS controller with a desired speed of 3m/s, 5m/s and 8m/s as well as for the cheating FS. All plots are computed using a fixed penetration rate of 10% and using the energy model presented in Sec. 3.3.

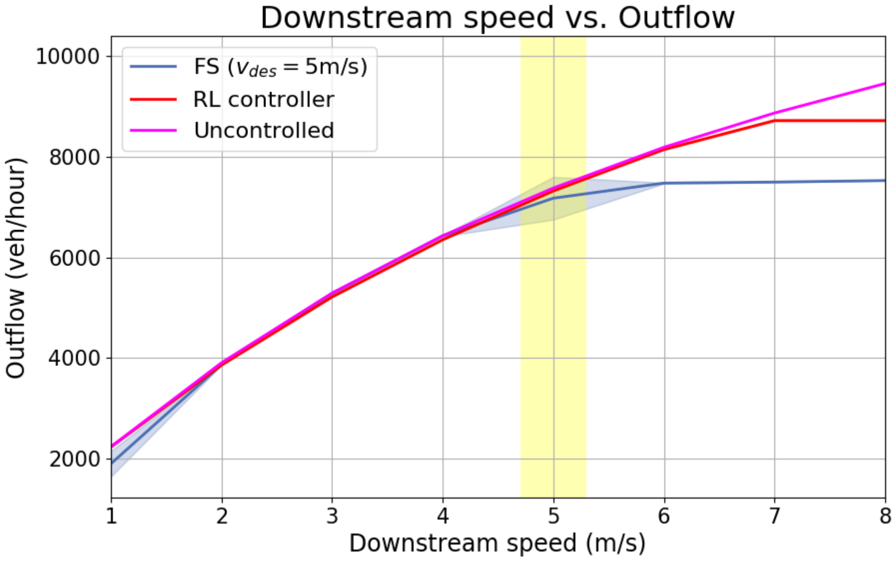

Figure 5.4: System outflow as a function of the downstream speed, shown for the RL controller, the FS controller with desired speed equal to 5m/s, and the uncontrolled human baseline. The yellow area highlights the downstream speed which the RL controller was trained on, outside of which it is acting in complete generalization. All plots are computed using a fixed penetration rate of 10%.

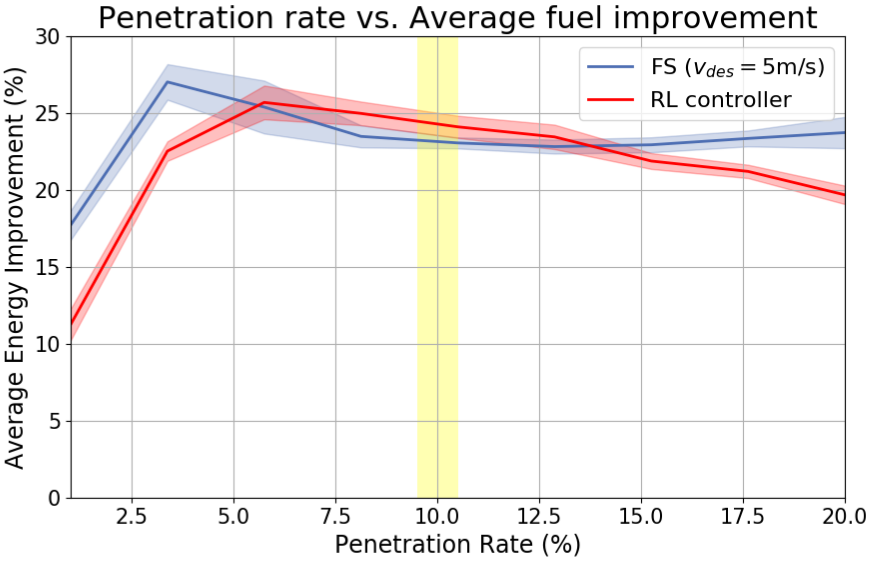

Figure 5.5: Fuel efficiency improvement of the RL controller on its training penetration rate of 10% (highlighted in yellow) over the uncontrolled human baseline, and generalization to penetration rates outside that range. Fuel improvement is also shown for the FS controller with a desired speed of 5m/s. Both plots are computed using a fixed downstream speed of 5m/s and using the energy model presented in Sec. 3.3.

5.3 Behavior analysis

In this section, we provide both qualitative and quantitative analysis to explain the energy improvements induced by the RL controller. In Figure 5.6, we plot the acceleration profile of the different controllers as a function of the speed of the lead vehicle and the space gap i.e. distance to the lead car. Since the acceleration profile is 3-dimensional, we show slices of the acceleration at 3, 5, and 7 m/s for the speed of the controller car. Note that these acceleration profiles are the output after post-processing of the desired output with the safety controller discussed in Sec. 3.2. As can be observed in the lower-half of the plots, the RL controller has a wide region where it accelerates at an almost fixed acceleration rate, and a vanishingly small region where it brakes. The RL controller is slowly accelerating at a fixed rate, with the magnitude of positive acceleration decreasing as the AV speed passes from 3 to 7 m/s. Above 7 m/s, the RL controller only brakes, which explains the reduction in outflow at downstream speeds above 7 m/s observed in Figure 5.4. Essentially, the RL controller is accelerating most of the time and then relying on the safety controller to brake sharply at the appropriate moment.

Finally, Figure 5.7, examines the net acceleration of the controllers. As can be seen, the RL controller has a consistently higher amount of acceleration than the cheating FS but is able to outperform it in MPG at both 5 and 6 m/s despite the higher accelerations at those values. This is possible due to an asymmetry in the energy function; braking incurs zero energy cost while the energy cost increases super-linearly with increasing acceleration. By maintaining low accelerations and braking sharply at the last possible second, the RL

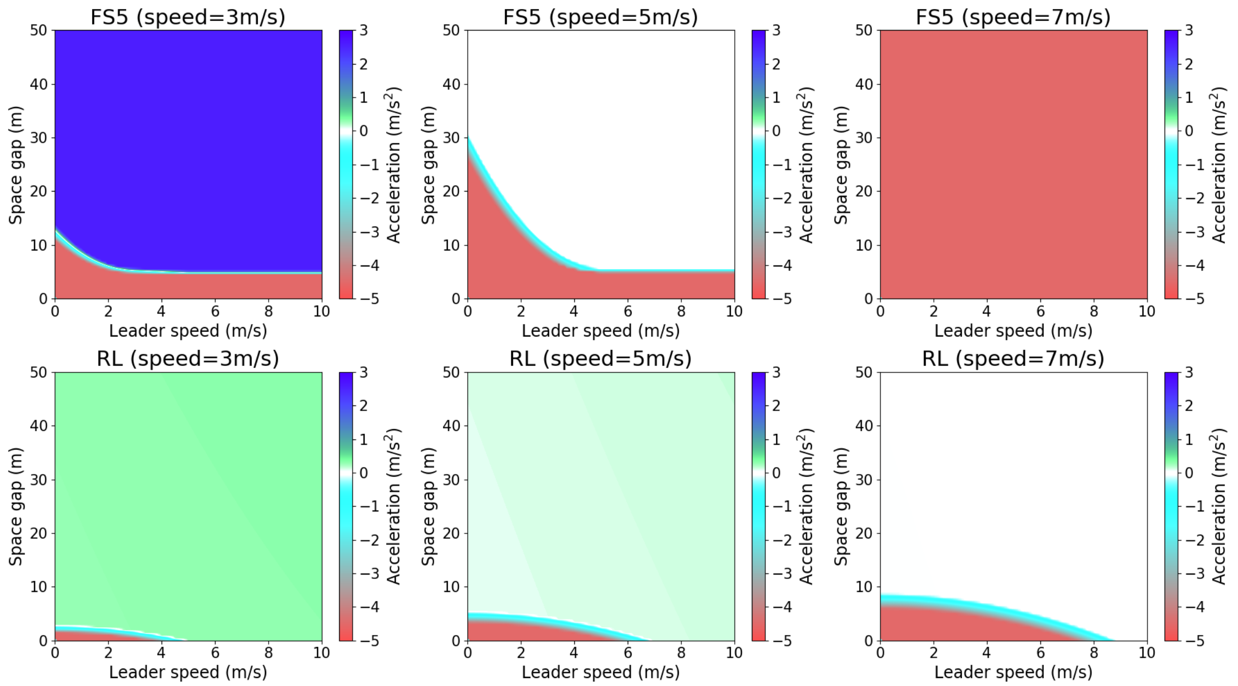

Figure 5.6: This figure demonstrates the difference in acceleration profile between our RL controller (bottom) and the FS controller set with a desired speed of 5m/s (top). The instantaneous acceleration output of both controllers is plotted as a function of the AV speed (left: 3m/s; middle: 5m/s; right: 7m/s), the leader speed and the space gap to the leader.

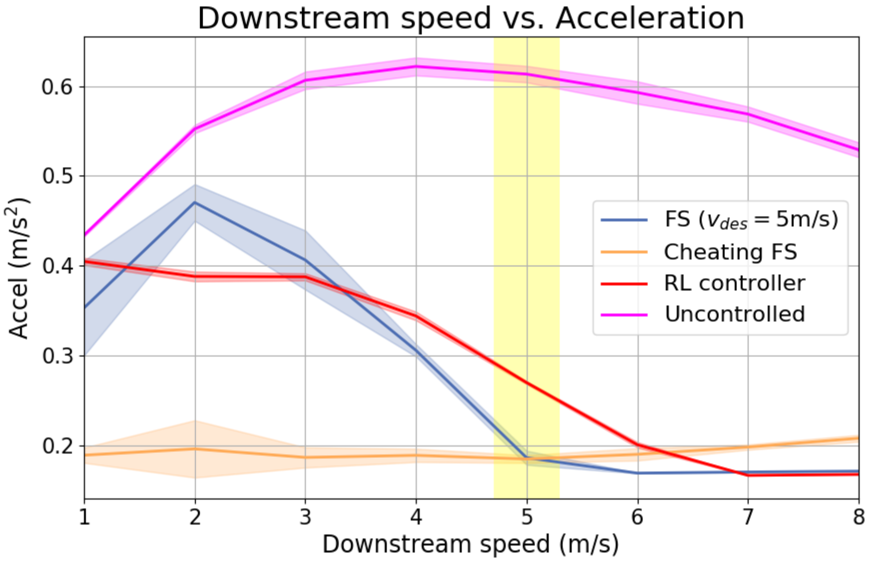

Figure 5.7: Average AV acceleration (absolute value of the instantaneous acceleration) using the RL controller at its training downstream speed of 5m/s (highlighted in yellow) and generalization to speeds outside that range. It is also plotted for the uncontrolled human baseline, the FS controller with a desired speed of 5m/s, and the cheating FS controller. All plots are computed using a fixed penetration rate of 10%.

controller is able to reduce energy expenditure while maintaining reasonable speeds.

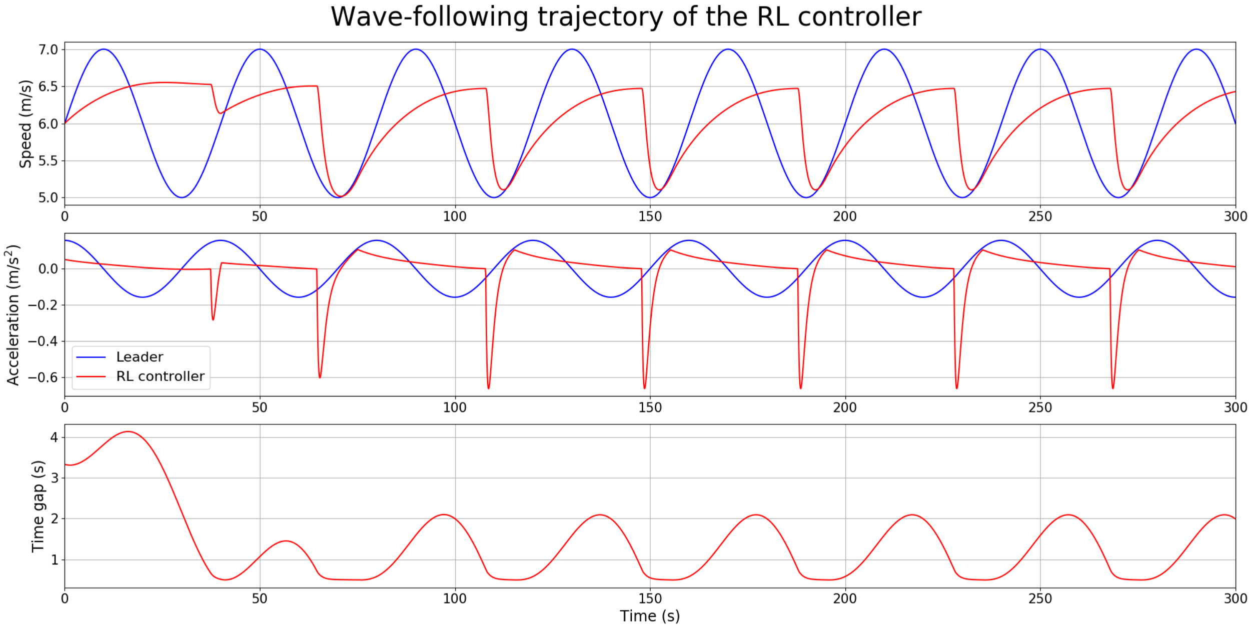

Figure 5.8: Acceleration, velocity and time gap profiles of an AV using the RL controller, following a leader trajectory that has a sinusoidal velocity centered around 6m/s with an amplitude of 1m/s and a period of 40s, whose velocity and acceleration profiles are also plotted.

Conclusion

In this work we set forth a challenging new network for phantom jam smoothing and demonstrate that multi-agent reinforcement learning can be used to design effective controllers for optimizing energy over the whole network. We find that controllers designed in this way are remarkably robust and, despite having no memory with which to perform system-identification, have the same efficacy as controllers that know the system equilibrium across a wide range of potential wave-inducing conditions. We qualitatively analyze the characteristics of these controllers relative to a standard baseline and additionally demonstrate that our controller functions effectively across varied penetration rates. Future work will investigate how well these controllers transfer to new networks as well as their robustness to a larger range of potential human driving dynamics.

Bibliography

[1] Nathan Lichtlé, Eugene Vinitsky, George Gunter, Akash Velu, and Alexandre M. Bayen. Fuel consumption reduction of multi-lane road networks using decentralized mixed-autonomy control. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2068–2073, 2021.

[2] Darbha Swaroop. String stability of interconnected systems: An application to platooning in automated highway systems. 1997.

[3] Assad Al Alam, Ather Gattami, and Karl Henrik Johansson. An experimental study on the fuel reduction potential of heavy duty vehicle platooning. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 306–311. IEEE, 2010.

[4] Cathy Wu, Aboudy Kreidieh, Kanaad Parvate, Eugene Vinitsky, and Alexandre M Bayen. Flow: Architecture and benchmarking for reinforcement learning in traffic control. *arXiv preprint arXiv:1710.05465*, page 10, 2017.

[5] Raphael E Stern, Shumo Cui, Maria Laura Delle Monache, Rahul Bhadani, Matt Bunting, Miles Churchill, Nathaniel Hamilton, Hannah Pohlmann, Fangyu Wu, Benedetto Piccoli, et al. Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transportation Research Part C: Emerging Technologies*, 89:205–221, 2018.

[6] Yuki Sugiyama, Minoru Fukui, Macoto Kikuchi, Katsuya Hasebe, Akihiro Nakayama, Katsuhiro Nishinari, Shin-ichi Tadaki, and Satoshi Yukawa. Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam. *New journal of physics*, 10(3):033001, 2008.

[7] Raphael E Stern, Yuche Chen, Miles Churchill, Fangyu Wu, Maria Laura Delle Monache, Benedetto Piccoli, Benjamin Seibold, Jonathan Sprinkle, and Daniel B Work. Quantifying air quality benefits resulting from few autonomous vehicles stabilizing traffic. *Transportation Research Part D: Transport and Environment*, 67:351–365, 2019.

[8] Fangyu Wu, Raphael E Stern, Shumo Cui, Maria Laura Delle Monache, Rahul Bhadani, Matt Bunting, Miles Churchill, Nathaniel Hamilton, Benedetto Piccoli, Benjamin Seibold, et al. Tracking vehicle trajectories and fuel rates in phantom traffic jams: Methodology and data. *Transportation Research Part C: Emerging Technologies*, 99:82–109, 2019.

[9] Li Jin and Karl Henrik Johansson. Coordinating vehicle platoons for highway bottleneck decongestion and throughput improvement. *arXiv preprint arXiv:1907.13049*, 2019.

[10] Mladen Čičić and Karl Henrik Johansson. Stop-and-go wave dissipation using accumulated controlled moving bottlenecks in multi-class ctm framework. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 3146–3151. IEEE, 2019.

[11] Sangjae Bae, Yeojun Kim, Yongkeun Eric Choi, Jacopo Guanetti, Preet Gill, Francesco Borrelli, and Scott Moura. Ecological adaptive cruise control of plug-in hybrid electric vehicle with connected infrastructure and on-road experiments. *Journal of Dynamic Systems, Measurement, and Control*, 2021.

[12] Tyler Ard, Longxiang Guo, Robert Austin Dollar, Alireza Fayazi, Nathan Goulet, Yunyi Jia, Beshah Ayalew, and Ardalan Vahidi. Energy and flow effects of optimal automated driving in mixed traffic: Vehicle-in-the-loop experimental results. *Transportation Research Part C: Emerging Technologies*, 130:103168, 2021.

[13] Jiaxun Cui, William Macke, Harel Yedidsion, Aastha Goyal, Daniel Urielli, and Peter Stone. Scalable multiagent driving policies for reducing traffic congestion. *arXiv preprint arXiv:2103.00058*, 2021.

[14] Yulin Zhang, William Macke, Jiaxun Cui, Daniel Urieli, and Peter Stone. Learning a robust multiagent driving policy for traffic congestion reduction. *arXiv preprint arXiv:2112.03759*, 2021.

[15] Sai Krishna Sumanth Nakka, Behdad Chalaki, and Andreas Malikopoulos. A multi-agent deep reinforcement learning coordination framework for connected and automated vehicles at merging roadways. *arXiv preprint arXiv:2109.11672*, 2021.

[16] Behrad Toghi, Rodolfo Valiente, Dorsa Sadigh, Ramtin Pedarsani, and Yaser P Fallah. Altruistic maneuver planning for cooperative autonomous vehicles using multi-agent advantage actor-critic. *arXiv preprint arXiv:2107.05664*, 2021.

[17] Paul Young Joun Ha, Sikai Chen, Jiqian Dong, Runjia Du, Yujie Li, and Samuel Labi. Leveraging the capabilities of connected and autonomous vehicles and multi-agent reinforcement learning to mitigate highway bottleneck congestion. *arXiv preprint arXiv:2010.05436*, 2020.

[18] Eugene Vinitsky, Nathan Lichtle, Kanaad Parvate, and Alexandre Bayen. Optimizing mixed autonomy traffic flow with decentralized autonomous vehicles and multi-agent rl. *arXiv preprint arXiv:2011.00120*, 2020.

[19] Huichu Zhang, Siyuan Feng, Chang Liu, Yaoyao Ding, Yichen Zhu, Zihan Zhou, Weinan Zhang, Yong Yu, Haiming Jin, and Zhenhui Li. Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The World Wide Web Conference*, pages 3620–3624, 2019.

[20] Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui Li. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3414–3421, 2020.

[21] Xinshi Zang, Huaxiu Yao, Guanjie Zheng, Nan Xu, Kai Xu, and Zhenhui Li. Metalight: Value-based meta-reinforcement learning for traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1153–1160, 2020.

[22] Martin Schönhof and Dirk Helbing. Empirical features of congested traffic states and their implications for traffic modeling. *Transportation Science*, 41(2):135–166, 2007.

[23] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

[24] Arne Kesting, Martin Treiber, and Dirk Helbing. Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1928):4585–4605, 2010.

[25] Darbha Swaroop and J Karl Hedrick. String stability of interconnected systems. *IEEE transactions on automatic control*, 41(3):349–357, 1996.

[26] Shumo Cui, Benjamin Seibold, Raphael Stern, and Daniel B Work. Stabilizing traffic flow via a single autonomous vehicle: Possibilities and limitations. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1336–1341. IEEE, 2017.

[27] Jonathan W. Lee, George Gunter, Rabie Ramadan, Sulaiman Almatrudi, Paige Arnold, John Aquino, William Barbour, Rahul Bhadani, Joy Carpio, Fang-Chieh Chou, Marsalis Gibson, Xiaoqian Gong, Amaury Hayat, Nour Khoudari, Abdul Rahman Kreidieh, Maya Kumar, Nathan Lichtlé, Sean McQuade, Brian Nguyen, Megan Ross, Sydney Truong, Eugene Vinitsky, Yibo Zhao, Jonathan Sprinkle, Benedetto Piccoli, Alexandre M. Bayen, Daniel B. Work, and Benjamin Seibold. Integrated framework of dynamics, instabilities, energy models, and sparse flow controllers. In *Proceedings of the Workshop on CPS Data for Transportation and Smart cities with Human-in-the-loop*, 2021.

[28] Argonne National Laboratory (ANL). Autonomie compiled vehicles, 2020. A technical manual for training in Autonomie.

[29] Frans A Oliehoek. Decentralized pomdps. In *Reinforcement Learning*, pages 471–503. Springer, 2012.

[30] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[31] Eric Liang, Richard Liaw, Philipp Moritz, Robert Nishihara, Roy Fox, Ken Goldberg, Joseph E. Gonzalez, Michael I. Jordan, and Ion Stoica. Rllib: Abstractions for distributed reinforcement learning, 2017.

[32] Marc G Bellemare, Georg Ostrovski, Arthur Guez, Philip Thomas, and Rémi Munos. Increasing the action gap: New operators for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.