C2SMART
CONNECTED CITIES WITH SMART TRANSPORTATION
A USDOT University Transportation Center

New York University

Rutgers University

University of Washington

The University of Texas at El Paso

City College of New York

Learning to Drive Autonomously

February 2020

Learning to Drive Autonomously

Zhong-Ping Jiang
New York University
0000-0002-4868-9359

Kaan Ozbay
New York University
0000-0001-7909-6532

Leilei Cui
New York University
0000-0001-8031-7638

C2SMART Center is a USDOT Tier 1 University Transportation Center taking on some of today's most pressing urban mobility challenges. Some of the areas C2SMART focuses on include:

Urban Mobility and Connected Citizens

Urban Analytics for Smart Cities

Resilient, Smart, & Secure Infrastructure

Disruptive Technologies and their impacts on transportation systems. Our aim is to develop innovative solutions to accelerate technology transfer from the research phase to the real world.

Unconventional Big Data Applications from field tests and non-traditional sensing technologies for decision-makers to address a wide range of urban mobility problems with the best information available.

Impactful Engagement overcoming institutional barriers to innovation to hear and meet the needs of city and state stakeholders, including government agencies, policy makers, the private sector, non-profit organizations, and entrepreneurs.

Forward-thinking Training and Development dedicated to training the workforce of tomorrow to deal with new mobility problems in ways that are not covered in existing transportation curricula.

Led by New York University's Tandon School of Engineering, **C2SMART** is a consortium of leading research universities, including Rutgers University, University of Washington, the University of Texas at El Paso, and The City College of NY.

Visit c2smart.engineering.nyu.edu to learn more

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

Acknowledgements

This project is supported in part by the USDOT Tier-1 C2SMART Center grant, in part by an NYU SOE PhD Fellowship awarded to the student, and in part by the US NSF.

Executive Summary

Autonomous vehicles (AV) and connected vehicles (CV) technologies have been much of the focus of transportation industry lately. In this project, to reduce congestion and improve network performance and safety, we have combined AV and CV technologies for connected and autonomous vehicles (CAVs) and developed innovative learning-based optimal control algorithms using reinforcement learning and adaptive dynamic programming techniques.

Firstly, we have developed (data-driven) adaptive learning algorithms to tackle cooperative adaptive cruise control (CACC) and combined longitudinal and lateral control of CAVs. Thanks to these advanced adaptive learning algorithms, we arrive at overcoming the limitations of existing model-based CACC for CAVs, by achieving both theoretically provable performance guarantees and learning-based adaptive responsiveness to uncertain and non-stationary environments. To explicitly address energy savings, this proposal generalizes previous results on CACC for CAVs by solving an optimal control problem that minimizes an integral quadratic constraint on the inter-vehicle distances and control inputs for a string of vehicles traveling in close proximity.

Secondly, instead of considering only purely autonomous vehicles, we have also studied the CACC problem for connected vehicles comprised of autonomous vehicles and human-driven vehicles. We have developed a learning-based adaptive optimal controller design framework that considers the human-vehicle interaction and heterogeneous driver behavior, without assuming the precise knowledge of the vehicle dynamics or the parameters of the driver model in the platoon. We have generalized our previous work in data-driven CACC to the realistic situation where the driver reaction time is not negligible and the vehicle nonlinearities are not approximated by linear functions. This is one of the first results in learning-based optimal control for mixed-traffic CAVs for tackling problems like the lane change and path following subject to human reaction time-delay. Research in this direction is important yet technically challenging because the full vehicle model is strongly nonlinear and involves couplings between the vehicle dynamics and the tire model as well as between the vehicle lateral dynamics of the platoon.

Last but not least, the adaptive learning control framework has been validated by means of SUMO and MATLAB based computer simulations for various driving conditions.

Table of Contents

List of Figures

Section 1 Learning-Based Adaptive Optimal Control for CVs

Subsection 1.1 Background and Contribution

Capitalizing on vehicle-to-vehicle (V2V) communication, such as dedicated short-range communications (DSRCs), CACC has been proposed as a longitudinal control strategy for a platoon of autonomous vehicles to achieve small headways and to attenuate the disturbance from leading vehicles [1]–[6]. However, given the currently low penetration rate of autonomous vehicles, we need to consider a more practical situation for the near future, where CAVs share the road with human-driven vehicles (HDVs). In addition, several field experiments have demonstrated that HDVs will undermine the performance of CACC design if HDVs are not taken into consideration for the control design [7]–[9]. Consequently, the control design for a platoon mixed with HDVs and CAVs is required. In [10], such a mixed traffic scenario is called connected cruise control, where a CAV receives motional data from its preceding vehicles and adjusts its speed while HDVs remain controlled solely by human drivers. In [10], a linear quadratic regulator (LQR) is designed to minimize velocity and headway fluctuation in the platoon, assuming that the drivers' feedback gains and reaction time of the adjacent HDVs are homogeneous. Accordingly, a classical model-based LQR approach lacks adaptivity and robustness to deal with the heterogeneous driver car-following models. To overcome the uncertainties caused by this heterogeneity of state-of-the-art driver control models, robust control methods have been proposed in the literature, e.g., [11] and [12]. Based on the prior knowledge of nominal driver parameters, stability/robustness is ensured by these robust controllers. But the transient performance of the platoon is not optimized as an objective in the control design. Hence, the optimality and the adaptivity have not been addressed simultaneously by these existing model-based methods. Because of these observations, we believe that adaptive optimal control (AOC) is more desirable for practical implementation, which can continually handle the model uncertainty introduced by the unknown driver-dependent parameters and simultaneously optimize the transient performance of the platoon.

This report adopts ideas from reinforcement learning [13] and adaptive dynamic programming (ADP) [14] to develop an intelligent and safe AOC algorithm for CAVs in the mixed traffic scenario. By systematic use of control theory, ADP has proven to be a powerful method to learn safe and stable controllers by using real-time data collected along the trajectory of the controlled system. One major advantage of ADP, as opposed to traditional reinforcement learning [13], lies in the fact that the closed-loop stability of the dynamic system is established when the learned control policy is implemented. Meanwhile, the stability/robustness of the CAV controller characterizes the convergence of the platoon toward a desired equilibrium state (headway and velocity). As a result of the closed-loop stability, it ensures that the state is bounded around the equilibrium and, thus, the safety can be ensured at all times. In [15], an ADP-based control design is proposed for the CAVs under a mixed traffic-flow environment. In our prior work [16], we developed a learning-based AOC algorithm to tackle the input-delay issue resulting from the vehicle's engine lags. Nonetheless, the impact of the human driver reaction time in the platoon has not been fully investigated. The reaction time constitutes the delays in the state of the human–vehicle platooning system, which can notably affect stability and control performance and even cause congestion and possibly crashes [17]. Therefore, a learning-based AOC algorithm to properly deal with the effect of the human driver reaction delay is advantageous to ensure the safety of mixed platoons and the optimal performance of the vehicular network.

In this report, we model the system of a platoon mixed with multiple connected HDVs and a CAV as a set of differential difference equations (DDEs), by taking into account drivers' heterogeneous feedback gains and reaction delays as well as the actuator (engine) delay. (See Fig. 1 for the communication topology.) Then, we follow an approximate discretization procedure to rewrite DDEs into a sampled-data linear system with approximation error. This discretization step transforms the problem of controlling DDEs to the control of an augmented linear system without delay and simplifies the control design procedure. Next, we incorporate a value-iteration (VI)-based ADP method with sampled-data control theory and propose a learning-based AOC design for the CAV without the exact knowledge of the human drivers in the platoon. We show the effectiveness of our proposed method through a SUMO-based simulation [35], which is an open-source microscopic traffic simulation platform. In addition, the proposed algorithm is validated using the well-known next-generation simulation (NGSIM) dataset. The main contributions of this report are summarized as follows.

1) The proposed learning-based controller for the CAV can adapt to different platoon dynamics caused by heterogeneous driver behavior.

2) The performance of the controlled platoon is optimized according to a linear quadratic criterion such that the velocity and headway fluctuations between vehicles are minimized and abrupt accelerations/decelerations that can cause unsafe situations are avoided.

3) To address safety concerns during real-time data collection, the presented algorithm can employ historical data and real-time data at the same time, which aims to speed up the learning process.

Subsection 1.2 Modeling of Mixed Traffic Flow

In this section, we briefly introduce the mathematical modeling of car-following behaviors of the HDV and the CAV. In our study, a mixed traffic platoon of n HDVs and one CAV is examined, as shown in Fig. 1. hi is defined as the bumper-to- bumper distance between vehicle i and its preceding vehicle i−1, vi as the velocity of vehicle i. With this simplified topology of the communication network, each CAV only receives motion data from the preceding connected HDVs. This setting does not cause a loss of generality for multiple CAVs, where every platoon can be considered to be separated by a CAV [12].

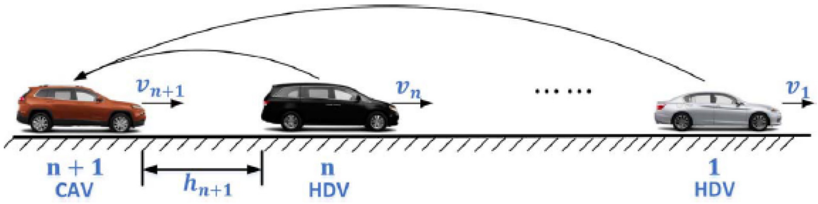


Figure 1. Platoon of Connected Human-driven and CAVs

Here, we consider the well-known optimal velocity model (OVM) with reaction delays [18], which is a nonlinear system. After linearizing it around the equilibrium, the OVM model can be written as DDEs

$$\dot{\tilde{h}}_i(t) = \tilde{v}_{i-1}(t) - \tilde{v}_i(t)$$

$$\dot{\tilde{v}}_i(t) = \alpha_i \left(N^* \tilde{h}_i(t - L_i) - \tilde{v}(t - L_i) \right) + \beta_i \big(\tilde{v}_{i-1}(t - L_i) - \tilde{v}_i(t - L_i) \big)$$

where $\tilde{h}_{n+1}(t) = h_{n+1} - h^*_{n+1}$, $\tilde{v}_{n+1}(t) = v_{n+1} - v^*$, and *u* is the designed control input, that is, the acceleration of the vehicle *n*+1 and η ≥ 0 is the input time delay.

Combining the aforementioned error dynamics, the statespace of the mixed traffic flow can be formulated. Suppose, for such a platoon of $n + 1$ vehicles, there exist p distinct drivers' reaction delays, where $p \leq n$. Assuming that there is a fictitious leading vehicle 0 traveling at constant velocity with $v_0(t) \equiv v^*$, $h_0(t) \equiv h^*$. Then, we focus on the controller design for the CAV in the platoon, whose state space representation is as follows:

$$\dot{x}(t) = A_0 x(t) + \sum_{i=1}^{p} A_i x(t - L_i) + Bu(t - \eta)$$

where the state vector $x(t) = [\tilde{h}_1, \tilde{v}_1, \ldots, \tilde{h}_{n+1}, \tilde{v}_{n+1}]^T \in \mathbb{R}_{nx}$, $nx = 2(n+1)$ and the input vector $u \in \mathbb{R}$. The designed controller u is only for the CAV at the tail of the platoon, while HDVs are not directly controlled in any form. Note that since HDVs are assumed to be stable, that is, each driver of the vehicle can achieve the traffic-flow equilibrium, the whole platooning system (5) is stabilizable in this setting, which is equivalent to

$$rank\left(sI - A_0 - \sum_{i=1}^{p} A_i e^{-sL_i} \quad B\right) = n_x$$

for s ∈ $\mathbb{C}^+$.

Subsection 1.3 Discretization of DDEs

To achieve a digital implementation, a sampled-data system with sampling interval T is considered. Integrating the linear model of the platoon over a sampling interval $[kT, kT + T]$ gives

$$x(kT+T) = e^{A_0 T} x(kT) + \int_0^T e^{A_0(T-r)} \left(\sum_{i=1}^{p} A_i x(kT + r - L_i)\right) dr + \int_0^T e^{A_0(T-r)} \left(Bu(kT + r - \eta)\right) dr$$

Here, we briefly review the procedure of sampling a set of DDEs. First, the drivers' reaction delays and the engine lag can be rewritten into

$$L_i = (N_i - 1 + a_i)T$$
$$\eta = (M - 1 + b)T$$

where integers $N_i \geq 1$, $M \geq 1$, and $a_i \in [0, 1)$, $b \in [0, 1)$. Suppose the maximum value of N_i is known as $N_{\max}$ and M is known as well. Then, the system in Figure 1 is discretized based on the following rules: 1) during each sampling interval, the delayed control signal $u(t-\eta)$ is piecewise constant and 2) the delayed state variable $x(t - L_i)$ is estimated by the interpolation method. Figure 2 illustrates the discretization procedure.

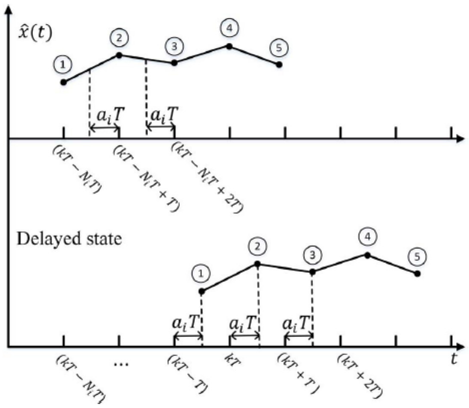


Figure 2. Interpolation-based Estimation for Delayed State Variables

The expression of the discretized input *u* can be expressed as

$$u(kT + r - \eta) = \begin{cases} u(kT - MT), if\ r \in [0, bT) \\ u(kT - MT + T), if\ r \in [bT, T) \end{cases}.$$

Similarly, for $r \in [0, a_iT)$, the interpolated value $\hat{x}(kT+r-L_i)$ is given by

$$\hat{x}(kT + r - L_i) = \left(a_i - \frac{r}{T}\right) x(kT - N_iT) + \left(1 - a_i + \frac{r}{T}\right) x(kT - N_iT + T)$$

and for $r \in [a_iT, T]$, $\hat{x}\ (kT + r - L_i)$ is described by

$$\hat{x}(kT + r - L_i) = \left(1 - \frac{r}{T} + a_i\right) x(kT - N_iT + T) + \left(\frac{r}{T} - a_i\right) x(kT - N_iT + 2T)$$

For notational simplicity, we define $x_k = x(kT)$ and $u_k = u(kT)$ for $k \in \mathbb{Z}_+$. Then according to the aforementioned equations, the dynamics of the state *x* can be expressed as

$$x_{k+1} = F_0 x_k + G_M u_{k-M} + G_{M-1} u_{k-M+1} + w_k + \sum_{i=1}^{p} (F_{N_i} x_{k-N_i} + F_{N_i-1} x_{k-N_i+1} + F_{N_i-2} x_{k-N_i+2})$$

By defining an augmented state $\bar{z}_k = [x_k^T, x_{k-1}^T, \dots, x_{k-Nmax}^T, u_k^T, u_{k-1}^T, \dots, x_{k-M}^T]^T \in \mathbb{R}^{n_z}$, where $n_z = nx(N\text{max} + 1) + M$. we have the following system representation

$$\bar{z}_{k+1} = \mathcal{A}\bar{z}_k + \mathcal{B}u_k + \mathcal{D}w_k$$

Subsection 1.4 Data-Driven ADP Design

Subsection 1.4.1 Model-based VI

In order to attenuate the disturbance for the mixed traffic, and considering the aforementioned discretized linear system without the approximation error *w*, the following LQR problem is formulated

$$\min_u \sum_{k=0}^{\infty} (\bar{z}_k{}^T Q \bar{z}_k + \bar{r} u_k^2)$$

Subject to $\bar{z}_{k+1} = \mathcal{A}\bar{z}_k + \mathcal{B}u_k$.

If the accurate dynamic model of the platoon is known, we can solve the LQR problem by the following VI approach

$$P_{j+1} = \mathcal{A}^T P_j \mathcal{A} + Q - \mathcal{A}^T P_j \mathcal{B}(\bar{r} + \mathcal{B}^T P_j \mathcal{B})^{-1} \mathcal{B}^T P_j \mathcal{A}$$

$$K_{j+1} = (\bar{r} + \mathcal{B}^T P_j \mathcal{B})^{-1} \mathcal{B}^T P_{j+1} \mathcal{A}$$

Subsection 1.4.2 Data-driven VI

In this section, a data-driven ADP learning algorithm is proposed to solve the optimal control problem without accurate knowledge of the human driver's feedback gains and reaction time.

First, denote

$$H_j = \begin{bmatrix} H_j^{11} & H_j^{12} \\ (H_j^{12})^T & H_j^{22} \end{bmatrix} = \begin{bmatrix} \mathcal{B}^T P_j \mathcal{B} & \mathcal{B}^T P_j \mathcal{A} \\ \mathcal{A}^T P_j \mathcal{B} & \mathcal{A}^T P_j \mathcal{A} \end{bmatrix}$$

where j is a non-negative integer and H_0 is zero matrix of appropriate dimension. Then, the augmented system derived in the previous section can be rewritten into

$$\bar{z}_{k+1} = \mathcal{A}_j \bar{z}_k + \mathcal{B}\big(K_j \bar{z}_k + u_k\big) + \mathcal{D}w_k$$

where $A_j = A\text{-}BK_j$. Then according to the model based value iteration and the aforementioned equation, we have

When the dynamics is unknown to us, based on ADP technique, we propose a data-driven method to solve the aforementioned equations,

$$\begin{aligned}\overline{z}_{k+1}^{T}\mathrm{Q}\overline{z}_{k+1} &= -\overline{z}_{k+1}^{T}\mathcal{F}\left(P_j\right)\overline{z}_{k+1} + \overline{z}_{k+1}^{T}P_{j+1}\overline{z}_{k+1} \\ &= -\overline{z}_{k+1}^{T}\left[H_j^{22} - (H_j^{12})^T\left(r + H_j^{11}\right)^{-1}H_j^{12}\right]\overline{z}_{k+1} + \left[vecv\left(\begin{bmatrix} u_k \\ \overline{z}_k \end{bmatrix}\right)\right]^T vecs\left(H_{j+1}\right) + \xi_k^j \\ &= -\phi_{k+1}^j + \psi_k^T vecs\left(H_{j+1}\right) + \xi_k^j\end{aligned}$$

where

$$\mathcal{F}\left(P_j\right) = \mathcal{A}^T P_j \mathcal{A} - \mathcal{A}^T P_j \mathcal{B}\left(\overline{r} + \mathcal{B}^T P_j \mathcal{B}\right)^{-1}\mathcal{B}^T P_j \mathcal{A}$$

$$\phi_{k+1}^j = \overline{z}_{k+1}^T\left[H_j^{22} - (H_j^{12})^T\left(r + H_j^{11}\right)^{-1}H_j^{12}\right]\overline{z}_{k+1}$$

$$\psi_k = vecv\left(\begin{bmatrix} u_k \\ \overline{z}_k \end{bmatrix}\right).$$

Also, we have $\xi_k^j = 2w_k^T\mathcal{D}^T P_{j+1}\mathcal{A}\overline{z}_k + 2w_k^T\mathcal{D}^T P_{j+1}\mathcal{B}u_k + w_k^T\mathcal{D}^T P_{j+1}w_k$.

In order to determine H_{j+1} from (23), measurable data, that is, $.z$ and u, are collected at multiple time instants $k_0 < k_1 < \cdot\cdot\cdot < k_{.s}$ where $\overline{s}$ is a sufficiently large positive integer. In particular, we can define

$$\Psi = [\psi_{k_0}, \psi_{k_1}, \dots, \psi_{k_p}]^T$$

$$\Phi_j = [\overline{z}_{k_0+1}^T Q\overline{z}_{k_0+1} + \phi_{k_0+1}^j, \dots, \overline{z}_{k_0+1}^T Q\overline{z}_{k_s+1} + \phi_{k_s+1}^j]^T$$

Then (23) can be expressed in the following matrix form:

$$\Psi\mathrm{U}_{j+1} + \Xi_j = \Phi_j, \mathrm{U}_{j+1} = vecs\left(\mathrm{H}_{j+1}\right), \Xi_j = [\xi_{k_0}^j, \dots, \xi_{k_s}^j]^T$$

Then U_{j+1} and K_{j+1} can be updated by

$$\mathrm{U}_{j+1} = (\Psi^T\Psi)^{-1}\Psi^T\left(\Phi_j - \Xi_j\right), \mathrm{K}_{j+1} = \left(r + H_j^{11}\right)^{-1}H_j^{12}$$

However, the approximation error w_k is not measurable in general, which implies that U_{j+1} cannot be obtained as the aforementioned method, so as K_{j+1}. Next, an approximate solution is proposed.

We define the matrix $\widehat{H}_j$ as the approximate solution to H_j

$$\widehat{H}_j = \begin{bmatrix} \widehat{H}_j^{11} & \widehat{H}_j^{12} \\ (\widehat{H}_j^{12})^T & \widehat{H}_j^{22} \end{bmatrix}$$

Where $\widehat{H}_0 = H_0$. Similar to the definition of ϕ_{k+1}^j, we construct

$$\widehat{\phi}_{k+1}^j = \bar{z}_{k+1}^T \left[\widehat{H}_j^{22} - (\widehat{H}_j^{12})^T \left(\bar{r} + \widehat{H}_j^{11}\right)^{-1} \widehat{H}_j^{12}\right] \bar{z}_{k+1}$$

which defines $\widehat{\Phi}_j = [\bar{z}_{k0+1}^T Q \bar{z}_{k0+1} + \widehat{\phi}_{k0+1}^j, \ldots, \bar{z}_{ks+1}^T Q \bar{z}_{ks+1} + \widehat{\phi}_{ks+1}^j]^T$, with $\widehat{\Phi}_0 = \Phi_0$. Furthermore, let the approximate solution$\widehat{U}_j = \mathrm{vecs}(\widehat{H}_j)$, which is solved by

$$\widehat{\mathrm{U}}_{j+1} = (\Psi^T \Psi)^{-1} \Psi^T \widehat{\Phi}_j,$$

Then, the controller can be updated by the approximated solution $\widehat{U}_j$

$$\mathrm{K}_{j+1} = \left(\bar{r} + \widehat{H}_j^{11}\right)^{-1} \widehat{H}_j^{12}.$$

Algorithm 1 shows the detailed VI-Based ADP algorithm.

Algorithm 1. VI-Based ADP Algorithm

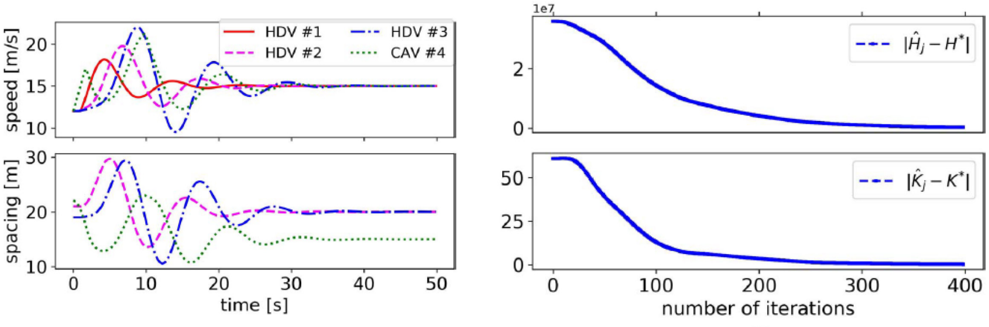
```
Begin
Select a sufficiently small threshold $\eta_h > 0$. $H_0 \leftarrow 0$.
Apply an initial controller, e.g. adaptive cruise controller with exploration noise on the time interval $[k_0, k_{\bar{s}}]$ to collect real time data. Compute $\widehat{\Phi}_0$ and $\Psi$. Let $j \leftarrow 0$.
while the rank condition is NOT satisfied
    Draw an experience $e$ from historical data set $\mathcal{H}$, and insert it into $\widehat{\Phi}_0$ and $\Psi$;
end while
repeat
    Determine $\widehat{H}_{j+1}$ and $\widehat{K}_{j+1}$.
    $j \leftarrow j + 1$;
until $|\widehat{H}_j - \widehat{H}_{j-1}| < \eta_h$
$j^* \leftarrow j$
Update controller $u_k = -\widehat{K}_{j^*} \bar{z}_k$.
```

Subsection 1.5 Validation Using Sumo Simulations and NGSIM Data

In this section, we first present the simulation results using SUMO to demonstrate the efficacy of the proposed data-driven ADP method. Here, the sampling period is set as T = 0.2 [s].

The platoon consists of three HDV followed by a CAV. The weighting matrices are set as $Q = 10^{-2}I$ and $\bar{r} = 1$. The initial control policies for the CAV are adaptive cruise control, which does not employ the data exchange from the HDVs. We collect the real-time data from 0 [s] to 10 [s], which generates 50 data points, and then obtain the rest from the historical data points. At 10 [s], the controller is updated following the proposed algorithm (Algorithm 1). The time trajectories of the platooning vehicles are depicted in Fig. 3. The convergence results of the algorithm are shown in Fig. 3(b). As analyzed in Theorems 1 and 2, the difference between the learned controller and the optimal one is affected by the quality of the data, including the sampling interval. It is observed that after 400 iterations the differences are close to zero. The computation time of the proposed algorithm for all iterations in this simulation is 2.2 [s] using Intel Core i7- 4720HQ CPU 2.60 GHz and 16.0-GB memory.

Robustness Evaluation Compared with the Model-Based Approach: We compare our proposed learning-based control algorithm with a model-based optimal control design method, that is, LQR, where the design is based on the nominal driver-dependent parameters. As a result, the mismatch between the nominal and the actual values of the driver-dependent parameters causes the nonoptimal performance of the model-based control design method. In the simulation, the velocity of the fictitious leading vehicle 0 follows $v_0(t) = v^* + 4\sin(t)$. We implement our learning-based controller and the model-based design for the CAV. The result is shown in Figure 4, where the ADP-based controller produces smaller magnitude of oscillation in terms of velocity and headway. As a consequence, the learning-based design can lead to a better disturbance attenuation performance compared to the model-based optimal control design.



(a) (b)

Figure 3. Time Trajectories of the Four-vehicle Platooning System. (a) Speed and Spacing Trajectory of Vehicles #1–#4. (b) Convergence Results of the Proposed Algorithm

In addition, the energy efficiency improvement is computed for the CAVs' trajectories in Figure 4, where the net energy is defined as

$$E_w = \int_0^{t_f} F_w(t)v(t)dt$$

where tractive or braking force at the wheels $F_w(t) = ma(t) + mgC_r + (1/2)\rho_a A_f C_D v2(t)$, m is the mass of the vehicle, C_r is the coefficient of rolling resistance, ρ_a is the air density, A_f is vehicle front area, C_D is the aerodynamic drag coefficient, and g is the gravitational acceleration. $a(t)$ and $v(t)$ are the acceleration and the velocity of the CAV, respectively.It shows that the ADP-based control algorithm can reduce the total energy consumption by 7.12%, compared to the model-based optimal control design using the homogeneous nominal driver-dependent parameters.

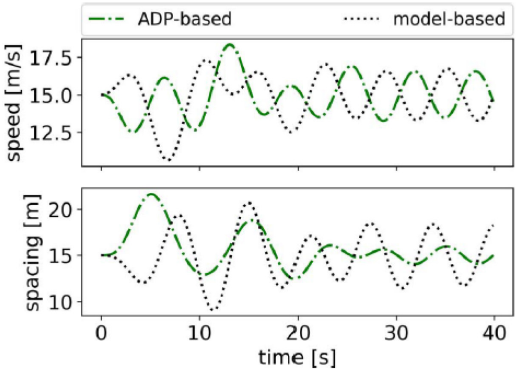


Figure 4. Speed and Spacing Trajectory of the CAV using the Proposed ADP-based Controller and the Model-based Optimal Controller

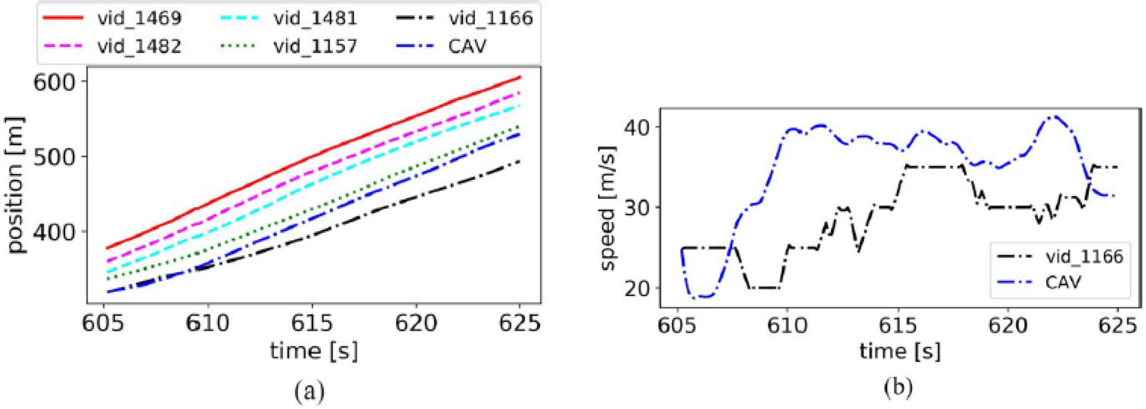


Figure 5. Time Trajectories of a Five-vehicle Platoon in NGSIM. (a) Space–time and Velocity Profile of all Vehicles. (b) Velocity Profile of Vehicle 1166 and the CAV

Then, we validate our proposed ADP algorithm using the vehicle trajectories from the NGSIM dataset. In particular, The U.S. Highway 101 (US 101) dataset is adopted, and the investigated lane is lane 4 in the five mainline lanes. The platoon trajectories of vehicles 1469, 1482, 1481, 1157, and 1166 are collected and the detailed illustration is shown in Figure 5(a). In the simulation, the vehicle 1166 (black dotted line) is replaced by a CAV equipped with our ADP controller and starts with the same initial condition as shown by the blue dotted line in Fig. 5(a). When the platoon is formed, we note that the CAV can safely keep a smaller steady-state headway (60 [m]) with respect to its immediately preceding vehicle, compared to the one between vehicles 1166 and 1157 This can potentially increase the traffic throughput. In addition, the velocity profile of the CAV is smoother than the one of vehicle 1166, which can lead to more passenger comfort and better energy efficiency.

Subsection 1.6 Conclusion

In this article, we have studied a general learning-based AOC problem for a platoon mixed with a CAV and multiple HDVs (equipped with V2V communication technology) subjected to heterogeneous drivers' behavior. By integrating the sampled-data control theory and an ADP method, an approximate optimal controller is designed for the CAV at the tail of the platoon, using a data-driven approach and without the exact knowledge of the driver behavior model in the platoon. The novel reinforcement-learning-based control method employs both the historical data and the data collected in real time, due to the off-policy property of our proposed ADP algorithm. This significantly reduces learning duration and thus provides additional safety improvements. We have validated our proposed approach through SUMO simulations and the NGSIM dataset.

In this study, the range policy in the OVM is assumed to be homogeneous for drivers in the platoon. We aim to relax the assumption in our future work. In addition, our future work will also include the AOC of connected vehicles for improved safety performance, energy efficiency, and more complex maneuvers, such as lane changing. The AOC design with different communication topologies of CAVs will be considered as well as highly nonlinear vehicle models.

Section 2. Combined Longitudinal and Lateral Control of AVs based on Reinforcement Learning

Subsection 2.1 Background and Contribution

The increase in the number of vehicles challenges the capability of the existing transportation infrastructure. To solve the congestion and safety problems caused by increased transportation demands, one way is to optimize the transportation infrastructure, including the highway design and traffic signals. The other way is to reduce the distance between vehicles to increase road capability. ACC is developed to reduce the inter-vehicle distance with the feedback of the inter-vehicle distance and the relative velocity. Then with the V2V, ACC is extended to CACC, which can not only reduce the distance between vehicles but also attenuate the disturbance along the platoon [19]. For ACC and CACC, most existing methods emphasis on the longitudinal control of autonomous vehicles, which assumes that the vehicles move along a straight road. However, in many cases, roads are curved. In this case, vehicles should not only maintain a desired inter-vehicle distance, but also stay in lane. Therefore, combined longitudinal and lateral control of autonomous vehicles is a significant research topic.

To achieve the combined longitudinal and lateral control for autonomous vehicles, one method is to decompose it into two independent subsystems: longitudinal subsystem and lateral subsystem [20]. For longitudinal control, ACC or CACC can be applied. For instance, in [21], the authors propose a data-driven adaptive optimal control approach to solve the CACC problem considering the input delay and the disturbance. For lateral control, a lane keeping controller design method should be applied. For instance, in [22], camera is applied to detect the lane and a data-driven optimal control approach is proposed to achieve the lateral control. Ploeg [23] propose a look-ahead approach to follow the preceding vehicle. However, cutting corner phenomenon will happen when the following vehicle follows the preceding vehicle directly. To overcome the cutting-corner limitation, an extended look-ahead approach is proposed in [24]. However, when considering the nonlinear dynamics of the vehicle, the physical parameters, especially the tire cornering stiffness, are hard to measure. Besides, in these methods, the performance of the designed controller cannot be guaranteed optimal.

ADP is an effective data-driven approach to find the optimal controller without requiring the precise knowledge of the system dynamics. ADP is developed based on the dynamic programming and reinforcement learning. The data along the trajectories of the control system, including the states and the control inputs, are collected, and then these data are applied iteratively to find the optimal controller. It is theoretically shown that at each iteration a sub-optimal controller with improved performance can be obtained, and with the iteration of the learning algorithm, these obtained sub-optimal controllers can converge to the optimal one. Guaranteed stability with learning-based controllers for the closed-loop system is an advantage of ADP over traditional reinforcement learning algorithm. Therefore, ADP attracts considerable attention in the transportation field of which safety is a top priority to be considered. For instance, some researchers propose an ADP-based CACC for an exclusive bus line. Some presented an ADP-based control strategy to achieve lateral stability of the autonomous vehicle. Someone proposed a shared framework of the driver and the autonomous vehicle to achieve lateral control. In these papers, ADP is applied to linear systems. However, for the combined longitudinal and lateral control of the autonomous vehicle, the vehicle dynamics is nonlinear. Clearly, using a linearized model can only guarantee the local stability of the closed-loop system.

In this report, to solve the data-driven combined longitudinal and lateral optimal control problem, considering the nonlinear dynamics of the vehicle, the ADP based output regulation approach [25] is adopted. Firstly, the nonlinear dynamics of the following vehicle is established. Based on the extended look-ahead approach [24] and the dynamics of the following vehicle, the error states of the system are defined, and the dynamics of the corresponding error system is derived. Then, based on the dynamics of the error system, the HJB equation is applied to solve the corresponding optimal control problem and a model-based policy iteration algorithm is proposed to solve the HJB equation. Finally, based on the ADP approach, a two-phase data-driven policy iteration algorithm is proposed. The first phase is to obtain the desired driving inputs when the error states are zero by solving the corresponding regulator equation. The second phase is to iteratively solve the HJB equation by the collected data.

In this report, to solve the data-driven combined longitudinal and lateral optimal control problem, considering the nonlinear dynamics of the vehicle, the ADP based output regulation approach is adopted. Firstly, the nonlinear dynamics of the following vehicle is established. Based on the extended look-ahead approach and the dynamics of the following vehicle, the error states of the system are defined and the dynamics of the corresponding error system is derived. Then, based on the dynamics of the error system, the HJB equation is applied to solve the corresponding optimal control problem and a model based policy iteration algorithm is proposed to solve the HJB equation. Finally, based on the ADP approach, a two-phase data-driven policy iteration algorithm is proposed. The first phase is to obtain the desired driving inputs when the error states are zero by solving the corresponding regulator equation. The second phase is to iteratively solve the HJB equation by the collected data. Compared with previous works of others, this paper proposes a data-driven solution to the combined longitudinal and lateral control when the nonlinear dynamics of the following vehicle is considered. In particular, using reinforcement learning techniques, an optimal controller is learned from real-time data without assuming the precise knowledge of the vehicle model.

Subsection 2.2 Dynamic Modelling

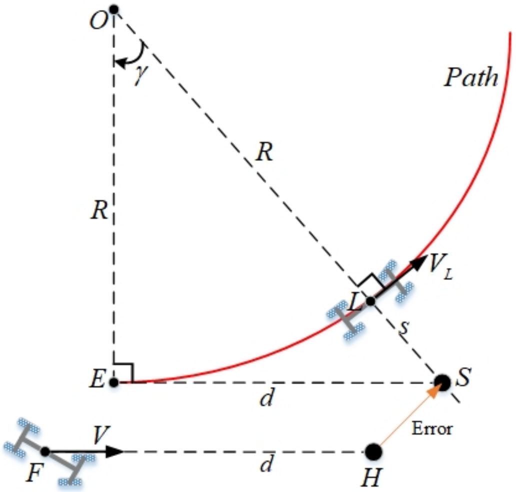


Figure 6. The Leading and the Following Vehicle

Let $X_L = [x_L, y_L, \varphi_L, V_L, \omega_L]^T$ denote the state of the leading vehicle, The kinematics of the leading vehicle can be expressed as

$$\dot{\mathrm{x}}_L = V_L \cos\varphi_L, \dot{\mathrm{y}}_L = V_L \sin\varphi_L$$

$$\dot{\varphi}_L = \omega_L, \dot{V}_L = a_L, \dot{\omega}_L = \Omega_L$$

Therefore, $\dot{\mathrm{X}}_{\mathrm{L}} = \mathrm{f}_L(\mathrm{X}_L; \psi)$. Then the problem solved in this section can be formulated as: In the absence of the precise knowledge of the vehicle dynamics (*F*), calculate an optimal controller which can regulate the signal of the motor installed at each wheel of the following vehicle, such that the vehicle *F* can keep a desired distance from the vehicle L and move along the path.

Let x, y, and φ denote the position and yaw angle of the vehicle *F*. *V*x, *V*y and *w* denote the longitudinal, lateral and yaw angular velocity of the vehicle *F*. Then according to the kinematics of the vehicle F, one can obtain the following equation

$$\dot{x} = V_x \cos\varphi - V_y \sin\varphi$$
$$\dot{y} = V_x \sin\varphi + V_y \cos\varphi$$
$$\dot{\varphi} = \omega$$

Figure 7 shows the dynamic model of the following vehicle. Considering the tire model, the dynamics of the following vehicle can be expressed as

$$\dot{V}_x = V_y\omega - \frac{C_a}{M}V_x^2 + \frac{2k}{M}u_1 + \frac{2k}{M}u_3$$

$$\dot{V}_y = -V_x\omega - \frac{C_f + C_r}{M}\frac{V_y}{V_x} + \frac{C_r l_r - C_f l_f}{M}\frac{\omega}{V_x} + \frac{C_f}{M}u_2$$

$$\dot{\omega} = \frac{C_r l_r - C_f l_f}{M}\frac{V_y}{V_x} - \frac{C_r l_r^2 + C_f l_f^2}{I_z}\frac{\omega}{V_x} - \frac{2l_s k}{I_z}u_1 + \frac{C_f l_f}{I_z}u_2 + \frac{2l_s k}{I_z}u_3$$

Therefore

$$\dot{X}_V = [\dot{V}_x, \dot{V}_y, \dot{\omega}]^T = f(X_V) + gu$$

where u = [u_1,u_2,u_3].

In Fig.6, d denotes a constant look-ahead distance. H is the look-ahead point and it is in the direction of V. If the vehicle *F* follows the vehicle *L* directly, it will cut the corner of the path. In order to avoid the cutting-corner problem, a virtual point S is defined which is in the direction of OL, and the vehicle F follows the virtual point S instead of L. ES=d is a tangent to the path. Therefore, in order to follow the vehicle L and move along the path, the desired position of vehicle F is E, and if this happens, S coincides with H. Therefore, OES forms a right triangle, and s = LS is determined by

$$\mathrm{s} = \begin{cases} 0, \kappa = 0 \\ \dfrac{-1 + \sqrt{1 + \kappa^2 d^2}}{\kappa}, \kappa \neq 0 \end{cases}$$

γ can be determined by

$$\gamma = \tan^{-1} \kappa d$$

The position of the virtual point is

$$\mathrm{S} = \begin{bmatrix} x_L \\ y_L \end{bmatrix} + s\begin{bmatrix} \sin\varphi_L \\ -\cos\varphi_L \end{bmatrix} = \begin{bmatrix} x_L \\ y_L \end{bmatrix} + \begin{bmatrix} s_x \\ s_y \end{bmatrix}$$

In order to let H converge to S, according to Fig. 6, the following variables are defined.

$$\begin{gathered} e_1 = x_L + s_x - x - d\cos\varphi \\ e_2 = y_L + s_y - y - d\sin\varphi \\ e_3 = \varphi_L - (\varphi + \gamma) \\ e_4 = V_L - V_x \\ e_5 = -V_y \\ e_6 = \omega_L - \omega \end{gathered}$$

Besides, the following variables are defined.

$$z_1 = \cos(\varphi + \gamma)e_1 + \sin(\varphi + \gamma)e_2$$

$$z_2 = -\sin(\varphi + \gamma)e_1 + \cos(\varphi + \gamma)e_2$$

Then the error states of the system are $\mathrm{e} = [z_1, z_2, \mathrm{e}_3, \dots, \mathrm{e}_6]^T$. The error dynamics can be expressed as

$$\dot{e} = f_e(\kappa, X_L, e) + g_e u_e$$

where

$$f_e = \begin{bmatrix} z_2\omega + (V_L + s\omega_L)\cos e_3 - V_x \cos\gamma - (V_y + d\omega)\sin\gamma \\ -z_1\omega + (V_L + s\omega_L)\sin e_3 + V_x \sin\gamma - (V_y + d\omega)\cos\gamma \\ e_6 \\ -V_y\omega + \dfrac{C_a}{M}(V_x^2 - V_L^2) \\ V_x\omega - V_L\omega_L + \dfrac{C_f + C_r}{M}\dfrac{V_y}{V_x} + \dfrac{C_f l_f^2 + C_r l_r^2}{I_z}\left(\dfrac{\omega}{V_x} - \dfrac{\omega_L}{V_L}\right) \\ -\dfrac{C_r l_r - C_f l_f}{I_z}\dfrac{V_y}{V_x} + \dfrac{C_f l_f^2 + C_r l_r^2}{I_z}\left(\dfrac{\omega}{V_x} - \dfrac{\omega_L}{V_L}\right) \end{bmatrix}$$

$$g_e = \begin{bmatrix} & 0_{3\times3} & \\ -\dfrac{2k}{M} & 0 & -\dfrac{2k}{M} \\ 0 & -\dfrac{C_f}{M} & 0 \\ \dfrac{2l_s k}{I_z} & -\dfrac{2l_f C_f}{I_z} & -\dfrac{2l_s k}{I_z} \end{bmatrix}$$

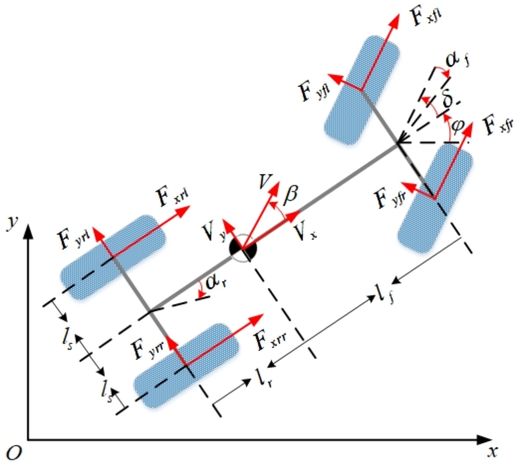


Figure 7. The Dynamic Model of the Following Vehicle

In the aforementioned error system, $u_e = u - u_d$ and u_d is the solution to the following regulator equation

$$[a_L, 0, \Omega_L]^T - f(V_L, 0, \omega_L) - g u_d = 0$$

Subsection 2.3 Two-phase Data-driven Policy Iteration

Subsection 2.3.1 Model-based Policy Iteration

In this section, to reduce the error states and the energy consumption, the cost criterion is defined as

$$\mathrm{J}(\kappa, \psi_0, \mathrm{X}_{L0}, e_0, u_e) = \int_0^\infty e^T Q e + u_e{}^T R u_e$$

where the weighting matrices Q and R are positive definite. The HJB equation for the corresponding optimal control problem can be expressed as

$$\frac{\partial V^{*T}}{\partial \psi}\mathrm{E}\psi + \frac{\partial V^{*T}}{\partial X_L} f_L + \frac{\partial V^{*T}}{\partial X_L} f_e - \frac{1}{4}(\frac{\partial V^{*T}}{\partial e} g_e) R^{-1} (\frac{\partial V^{*T}}{\partial e} g_e)^T + e^T Q e = 0$$

where $V^*(\kappa, \psi, X_L, e)$ is the value function. The optimal controller is

$$u_e^* = -\frac{1}{2} R^{-1} g_e^T \frac{\partial V^*}{\partial e}$$

Here, a model-based policy iteration algorithm is proposed to minimize the aforementioned cost criterion,

$$\frac{\partial V_i{}^T}{\partial \psi}\mathrm{E}\psi + \frac{\partial V_i{}^T}{\partial X_L} f_L + \frac{\partial V_i{}^T}{\partial X_L}\left(f_e + g_e u_{e,i}\right) + e^T Q e + u_{e,i}^T R u_{e,i} = 0$$

$$u_{e,i+1} = -\frac{1}{2} R^{-1} g_e^T \frac{\partial V_i}{\partial e}$$

At each iteration, we can obtain an improved controller $u_{e,i+1}$ and it converges to the optimal controller.

Subsection 2.3.2 Data-driven Policy Iteration

According to the previous sections, to obtain the optimal controller$u^* = u_e^* + u_d$, the regulator equation should be solved. Besides, when calculating the model-based policy iteration, the dynamic information of the system is required. However, it is laborious to establish an accurate dynamic model in practice. In addition, even if the dynamics of the vehicle is given, solving model-based policy iteration is still nontrivial. In this section, to obtain u^* without knowing the dynamics of the following vehicle *F*, a two-phase data-driven policy iteration algorithm is developed. The first phase is to solve the regulator equation, and the second phase is to solve the HJB equation with the collected data of the system based on the policy iteration.

The first phase of the data-driven policy iteration is to solve the regulator equation. To solve the regulator equation, the key to the problem is to find the approximations of f and g. In (4), f is a nonlinear function and g is a constant matrix. According to the approximation theory, f can by approximated by $\hat{\mathrm{f}} = \sum_{j=1}^{N_f} \hat{\sigma}_j \, \phi_j^f(X_V)$, where $\hat{\sigma}_j \in \mathbb{R}^3$ and $\{\phi_j^f(X_V)\}_{j=1}^{\infty}$ are linearly independent basis functions. $\hat{\mathrm{g}} \in \mathbb{R}^{3\times 3}$ approximates g. Therefore, according to the dynamics of the following vehicle, we have

$$\frac{1}{2} X_V^T X_V|_{t_k}^{t_{k+1}} = \int_{t_k}^{t_{k+1}} X_V^T[\hat{f}(X_V) + \hat{g}u]d\tau + \varepsilon_k$$

Then, the approximation error can be written as

$$\begin{aligned}\varepsilon_k &= \frac{1}{2} X_V^T X_V|_{t_k}^{t_{k+1}} - \int_{t_k}^{t_{k+1}} X_V^T[\hat{f}(X_V) + \hat{g}u]d\tau = \frac{1}{2} X_V^T X_V|_{t_k}^{t_{k+1}} - \int_{t_k}^{t_{k+1}} X_V^T\left[\hat{\sigma}\Phi^f(X_V) + \hat{g}u\right]d\tau \\ &= \frac{1}{2} X_V^T X_V|_{t_k}^{t_{k+1}} - \int_{t_k}^{t_{k+1}} \Phi^{fT}(X_V) \otimes X_V^T d\tau vec(\hat{\sigma}) - \int_{t_k}^{t_{k+1}} u^T \otimes X_V^T d\tau vec(\hat{g})\end{aligned}$$

where $\hat{\sigma} = [\hat{\sigma}_1, \dots, \hat{\sigma}_{N_f}]$ and $\Phi^f = [\phi_1^f, \dots, \phi_{N_f}^f]$. Define $\varepsilon = [\varepsilon_1, \dots, \varepsilon_{l_\varepsilon}]$. Then

$$\varepsilon = \Xi_\varepsilon - \Lambda_\varepsilon[vec^T(\hat{\sigma}), vec^T(\hat{g})]^T$$

where

$$\Xi_\varepsilon = [\frac{1}{2} X_V^T X_V|_{t_1}^{t_2}, \dots, \frac{1}{2} X_V^T X_V|_{t_l}^{t_{l+1}}]^T$$

$$\Lambda_\varepsilon = \begin{bmatrix} \int_{t_1}^{t_2} \Phi^{fT}(X_V) \otimes X_V^T d\tau & \int_{t_1}^{t_2} u^T \otimes X_V^T d\tau \\ \dots & \dots \\ \int_{t_l}^{t_{l+1}} \Phi^{fT}(X_V) \otimes X_V^T d\tau & \int_{t_l}^{t_{l+1}} u^T \otimes X_V^T d\tau \end{bmatrix}$$

Then $\hat{\mathrm{f}}$ and $\hat{\mathrm{g}}$ can be approximated by

$$[vec^T(\hat{\sigma}), vec^T(\hat{g})]^T = (\Lambda_\varepsilon^T \Lambda_\varepsilon)^{-1} \Lambda_\varepsilon^T \Xi_\varepsilon$$

where both Λ_ε and Ξ_ε can be constructed by the data collected along the trajectory of the system. The regulator equation can be solved by

$$u_d = \hat{g}^{-1}([a_L, 0, \Omega_L]^T - \hat{f}(V_L, 0, \omega_L))$$

Once the regulator equation (11) is solved with the data from the system, u_d can be obtained, and consequently, HJB equation can be solved by the policy iteration approach. Define

$$\upsilon_i = u - u_d - u_{e,i}$$

then the error system can be rewritten as

$$\dot{e} = f_e(\kappa, \psi, X_L, e) + g_e u_{e,i} + g_e \upsilon_i$$

Then along the trajectory generated by the aforementioned equation, with the driving input u, the derivative of Vi can be expressed as

$$\dot{V}_i = \frac{\partial V_i^T}{\partial \psi} \mathrm{E}\psi + \frac{\partial V_i^T}{\partial X_L} f_L + \frac{\partial V_i^T}{\partial e}\left(f_e + g_e u_{e,i} + g_e \upsilon_i\right)$$

Then with the model-based policy iteration, the following equation can be obtained

$$\dot{V}_i = -e^T Q e - u_{u_{e,i}}^T R u_{e,i} + \frac{\partial V_i^T}{\partial e} g_e \upsilon_i = -e^T Q e - u_{u_{e,i}}^T R u_{e,i} - 2u_{u_{e,i+1}}^T \upsilon_i.$$

Therefore,

$$\mathrm{V}_i(t_{k+1}) - \mathrm{V}_i(t_k) + 2\int_{t_k}^{t_{k+1}} u_{u_{e,i+1}}^T R\, \hat{\upsilon}_i d\tau = \int_{t_k}^{t_{k+1}} -e^T Q e - u_{u_{e,i}}^T R u_{e,i}\, d\tau + \Delta_{i,k}$$

$$\hat{\upsilon}_i = u - \hat{u}_d - u_{e,i}, \Delta_{i,k} = 2\int_{t_k}^{t_{k+1}} -u_{u_{e,i+1}}^T R(\mathrm{u} - \hat{u}_d)\, d\tau$$

For V_i and $u_{e,i+1}$, they can be approximated by linear combinations of a set of linearly independent basis functions,

$$\hat{V}_i(\kappa, \psi, X_L, e) = \sum_{j=1}^{N_V} \hat{\rho}_{i,j} \phi_{i,j}^V(\kappa, \psi, X_L, e)$$

$$\hat{u}_{e,i+1}(\kappa,\psi,X_L,e) = \sum_{j=1}^{N_u} \hat{\mu}_{i,j}\phi_{i+1,j}^{u}(\kappa,\psi,X_L,e)$$

Then, the following equation can be derived.

$$\hat{\rho}_i^T\Phi_i^V(t_{k+1}) - \hat{\rho}_i^T\Phi_i^V(t_k) + 2\int_{t_k}^{t_{k+1}} \Phi_{i+1}^{uT}\,\hat{\mu}_i{}^T R\bar{\upsilon}_i d\tau = \int_{t_k}^{t_{k+1}} -e^T Qe - \hat{u}_{u_{e,i}}^T R\hat{u}_{e,i}\,d\tau + \xi_{i,k}$$

with

$$\bar{\upsilon}_i = u - \hat{u}_d - \hat{u}_{e,i},\ \Phi_i^V = [\phi_{i,1}^V,\ldots,\ \phi_{i,j}^V,\ldots,\phi_{i,N_V}^V]^T$$

$$\Phi_i^u = [\phi_{i,1}^u,\ldots,\ \phi_{i,j}^u,\ldots,\phi_{i,N_u}^u]^T,\ \hat{\rho}_i = [\hat{\rho}_{i,1},\ldots,\hat{\rho}_{i,j},\ldots,\hat{\rho}_{i,N_V}]^T$$

$$\hat{\mu}_i = [\hat{\mu}_{i,1},\ldots,\hat{\mu}_{i,j},\ldots,\hat{\mu}_{i,N_u}]^T$$

Therefore, the approximation error is

$$\xi_{i,k} = [\Phi_i^V|_{t_k}^{t_{k+1}}]^T\hat{\rho}_i + 2\int_{t_k}^{t_{k+1}} \overline{(\upsilon}_i^T R)\otimes\Phi_{i+1}^{uT}\,d\tau vec(\hat{\mu}_{i+1}^T)$$

Define $\xi_i = [\xi_{i,1},\ldots,\xi_{i,k},\ldots,\xi_{i,l}]^T$, then the following equation holds

$$\xi_i = \Lambda_{\xi,i}\begin{bmatrix}\hat{\rho}_i\\ vec(\hat{\mu}_{i+1}^T)\end{bmatrix} + \Xi_{\xi,i}$$

$$\Xi_{\xi,i} = \int_{t_1}^{t_2} e^T Qe + \hat{u}_{u_{e,i}}^T R\hat{u}_{e,i}\,d\tau,\ldots,\int_{t_l}^{t_{l+1}} e^T Qe + \hat{u}_{u_{e,i}}^T R\hat{u}_{e,i}\,d\tau$$

$$\Lambda_{\xi,i} = \begin{bmatrix}[\Phi_i^V|_{t_1}^{t_2}]^T & 2\int_{t_1}^{t_2}\overline{(\upsilon}_i^T R)\otimes\Phi_{i+1}^{uT}\,d\tau\\ \cdots & \cdots\\ [\Phi_i^V|_{t_l}^{t_{l+1}}]^T & 2\int_{t_l}^{t_{l+1}}\overline{(\upsilon}_i^T R)\otimes\Phi_{i+1}^{uT}\,d\tau\end{bmatrix}$$

According to least square method, the unknown parameters of the approximations can be calculated by

$$\begin{bmatrix}\hat{\rho}_i\\ vec(\hat{\mu}_{i+1}^T)\end{bmatrix} = -(\Lambda_{\xi,i}^T\Lambda_{\xi,i})^{-1}\Lambda_{\xi,i}^T\Xi_{\xi,i}$$

The detailed algorithm is shown in Algorithm 2.

Algorithm 2. Two-Phase Data-Driven Policy Iteration

```
Begin
    Phase-one: solving the regulator equation.
    Choose Driving inputs $u_{x,1}$ to explore the system.
    Collect data and construct the matrix $\Xi_\varepsilon$ and $\Lambda_\varepsilon$
    Estimate $\hat{f}$ and $\hat{g}$.
    Calculate $\hat{u}_d$
    Phase-two: solving the HJB equation.
    Repeat
        Choose Driving inputs $u_{x,2}$ to explore the system.
        Collect data and construct the matrix $\Xi_{\xi,i}$ and $\Lambda_{\xi,i}$
        Calculate $\hat{V}_i$ and $\hat{u}_{e,i+1}$
        $i \leftarrow i + 1$;
    until $i > M$
    update controller $\hat{u}_i(t) = \hat{u}_{e,i} + \hat{u}_d$.
```

Subsection 2.4 Numerical Simulation

In this section, simulations are conducted to show the efficiency of the two-phase data-driven policy iteration approach. Q and R are set as the identity matrix. As shown in Figure 8, initially, the cost is $J = 967.87$. After 12 iterations, the cost converges to $J = 20.08$. With the generated by the Algorithm 2, the error states' trajectory is shown in Figure 9. From these two figures, we can conclude that the learned controller can achieve the combined longitudinal and lateral control and minimize the cost at the same time.

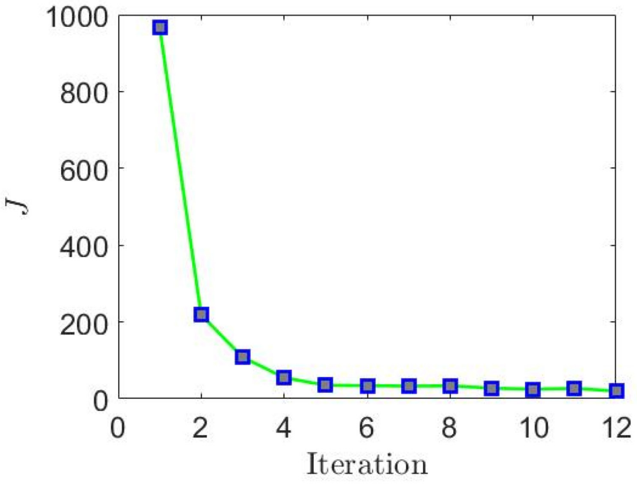


Figure 8. Cost for each Iteration

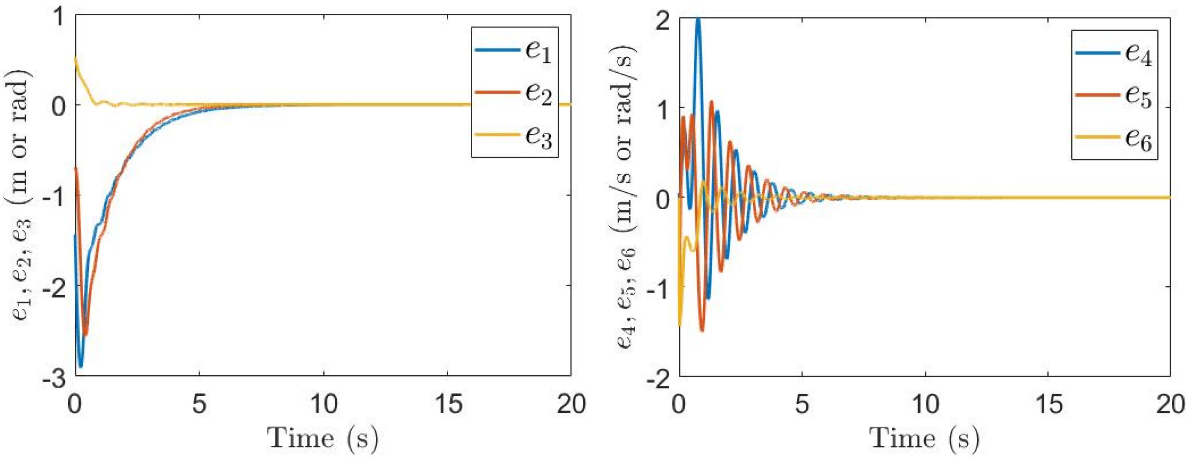


Figure 9. Error States of the Leading-following Vehicles

Subsection 2.5 Conclusion

In this paper, to achieve the combined longitudinal and lateral optimal control of an autonomous vehicle in the absence of the full knowledge of the following vehicle's dynamic model, we propose a two-phase data-driven policy iteration algorithm. Compared with the traditional model-based control approaches, the proposed algorithm avoids the need to build a mathematical model for the following vehicle and identify the physical parameters of the model. Compared with existing conventional reinforcement learning approaches, the proposed learning algorithm generates a sequence of stable sub-optimal controllers that converge to the optimal controller. In addition, its efficacy has been validated by using numerical simulations.

Section 3. References

1. E. van Nunen, J. Reinders, E. Semsar-Kazerooni, and N. van de Wouw, "String stable model predictive cooperative adaptive cruise control for heterogeneous platoons," *IEEE Trans. Intell. Veh.*, vol. 4, no. 2, pp. 186–196, Jun. 2019.
2. B. Besselink and K. H. Johansson, "String stability and a delay-based spacing policy for vehicle platoons subject to disturbances," *IEEE Trans. Autom. Control*, vol. 62, no. 9, pp. 4376–4391, Sep. 2017.
3. Y. Zhu, D. Zhao, and Z. Zhong, "Adaptive optimal control of heterogeneous CACC system with uncertain dynamics," *IEEE Transactions Control Syst. Technol.*, vol. 27, no. 4, pp. 1772–1779, Jul. 2019.
4. A. Petrillo, A. Pescapé, and S. Santini, "A secure adaptive control for cooperative driving of autonomous connected vehicles in the presence of heterogeneous communication delays and cyberattacks," *IEEE Trans. Cybern.*, early access, Jan. 23, 2020.
5. Z. Peng, D. Wang, T. Li, and M. Han, "Output-feedback cooperative formation maneuvering of autonomous surface vehicles with connectivity preservation and collision avoidance," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2527–2535, Jun. 2020.
6. Y. Zheng, J. Wang, and K. Li, "Smoothing traffic flow via control of autonomous vehicles," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3882–3896, May 2020.
7. V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 296–305, Feb. 2014.
8. M. Wang, W. Daamen, S. P. Hoogendoorn, and B. van Arem, "Cooperative car-following control: Distributed algorithm and impact on moving jam features," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 5, pp. 1459–1471, May 2016.
9. R. E. Stern *et al.*, "Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments," *Transp. Res. C, Emerg. Technol.*, vol. 89, pp. 205–221, Apr. 2018.
10. J. I. Ge and G. Orosz, "Optimal control of connected vehicle systems with communication delay and driver reaction time," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 2056–2070, Aug. 2017.
11. D. Hajdu, J. I. Ge, T. Insperger, and G. Orosz, "Robust design of connected cruise control among human-driven vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 749–761, Feb. 2020.

12. M. Di Vaio, G. Fiengo, A. Petrillo, A. Salvi, S. Santini, and M. Tufo, “Cooperative shock waves mitigation in mixed traffic flow environment,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4339–4353, Dec. 2019.
13. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*.
14. Y. Jiang and Z. P. Jiang, *Robust Adaptive Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2017.Cambridge, MA, USA: MIT Press, 2018.
15. W. Gao, Z. P. Jiang, and K. Ozbay, “Data-driven adaptive optimal control of connected vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1122–1133, May 2017.
16. M. Huang, W. Gao, and Z. P. Jiang, “Connected cruise control with delayed feedback and disturbance: An adaptive dynamic programming approach,” *Int. J. Adapt. Control Signal Process.*, vol. 33, pp. 356–370, 2019.
17. R. Sipahi and S.-I. Niculescu, “Deterministic time-delayed traffic flow models: A survey,” in *Complex Time-Delay Systems: Theory and Applications*. Heidelberg, Germany: Springer, 2010, pp. 297–322.
18. M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama, “Dynamical model of traffic congestion and numerical simulation,” *Phys. Rev. E*, vol. 51, pp. 1035–1042, Feb. 1995.
19. A. Vahidi and A. Sciarretta, “Energy saving potentials of connected and automated vehicles,” *Transportation Research Part C: Emerging Technologies*, vol. 95, Sept. 2018.
20. R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, “Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons,” *IEEE Trans. on Control Systems Technology*, vol. 8, no. 4, pp. 695–708, 2000.
21. M. Huang, W. Gao, and Z. P. Jiang, “Connected cruise control with delayed feedback and disturbance: An adaptive dynamic programming approach,” *International Journal of Adaptive Control and Signal Processing*, vol. 33, Oct. 2017.
22. M. Huang, M. Zhao, P. Parikh, Y. Wang, K. Ozbay, and Z. P. Jiang, “Reinforcement learning for vision-based lateral control of a selfdriving car,” *in Proc. 2019 IEEE 15th International Conference on Control and Automation*, pp. 1126–1131, July 2019, Edinburgh, UK.
23. J. Ploeg, N. van de Wouw, and H. Nijmeijer, “lp string stability of cascaded systems: Application to vehicle platooning,” *IEEE Trans. On Control Systems Technology*, vol. 22, no. 2, pp. 786–793, 2014.
24. A. Bayuwindra, J. Ploeg, E. Lefeber, and H. Nijmeijer, “Combined longitudinal and lateral control of car-like vehicle platooning with extended look-ahead,” *IEEE Trans. on Control Systems Technology*, vol. 28, no. 3, pp. 790–803, 2020.

25. W. Gao and Z. P. Jiang, “Learning-based adaptive optimal tracking control of strict-feedback nonlinear systems,” *IEEE Trans. on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2614–2624, 2018.