# Smoothing Mixed Traffic with Robust Data-driven Predictive Control for Connected and Autonomous Vehicles

Xu Shang[1], Jiawei Wang[2], and Yang Zheng[1]

***Abstract*— The recently developed `DeeP-LCC` (Data-EnablEd Predictive Leading Cruise Control) method has shown promising performance for data-driven predictive control of Connected and Autonomous Vehicles (CAVs) in mixed traffic. However, its simplistic zero assumption of the future velocity errors for the head vehicle may pose safety concerns and limit its performance of smoothing traffic flow. In this paper, we propose a robust `DeeP-LCC` method to control CAVs in mixed traffic with enhanced safety performance. In particular, we first present a robust formulation that enforces a safety constraint for a range of potential velocity error trajectories, and then estimate all potential velocity errors based on the past data from the head vehicle. We also provide efficient computational approaches to solve the robust optimization for online predictive control. Nonlinear traffic simulations show that our robust `DeeP-LCC` can provide better traffic efficiency and stronger safety performance while requiring less offline data.**

## I. Introduction

In traffic flow, small perturbations of vehicle motion may propagate into large periodic speed fluctuations, leading to so-called stop-and-go traffic waves [1]. This phenomenon significantly reduces traffic efficiency and driving safety. It has been widely demonstrated that connected and autonomous vehicles (CAVs) equipped with advanced control technologies, such as Cooperative Adaptive Cruise Control (CACC), have great potential to mitigate traffic jams [2], [3]. Yet, these technologies require a fully CAV environment, and the near future will meet with a transition phase of mixed traffic where human-driven vehicles (HDVs) coexist with CAVs [4], [5]. Thus, it is important to consider the behavior of HDVs when designing control strategies for CAVs.

The existing methods for the control of mixed traffic are generally categorized into model-based and model-free techniques. Model-based approaches typically use classical car-following models for HDVs, *e.g.*, the Optimal Velocity Model (OVM) [6], to derive a parametric representation for mixed traffic. This parametric model is then utilized for CAV controller design, using methods such as optimal control [7], [8], $\mathcal{H}_\infty$ control [9], model predictive control (MPC) [10], [11], and barrier methods [12]. For these approaches, an accurate identification of the car-following models is non-trivial due to the complex and non-linear human driving behaviors. In contrast, model-free methods bypass system identification and directly design controllers for CAVs from data. For example, reinforcement learning [13] and adaptive dynamic programming [14] have been employed to learn wave-dampening CAV strategies. However, practical deployments of these methods are limited due to their computation burden and lack of interpretability and safety guarantees.

Alternatively, data-driven predictive control methods that combine learning techniques with MPC have shown promising results. In particular, the recent `DeeP-LCC` [15] exploits the Data-EnablEd Predictive Control (`DeePC`) [16], [17] technique for the Leading Cruise Control (LCC) [18] system in mixed traffic. This method directly utilizes the measured traffic data to design optimal control inputs for CAVs and explicitly incorporates input/output constraints in terms of limits on acceleration and car-following distance. Large-scale numerical simulations [15] and real-world experiments [19] have validated the capability of `DeeP-LCC` to smooth mixed traffic flow. However, the standard `DeeP-LCC` has an important zero velocity error assumption, *i.e.*, the future velocity of the head vehicle remains the same as the equilibrium velocity of traffic flow. This assumption facilitates the online computation of `DeeP-LCC`, but it will cause a mismatch between the real traffic behavior and its online prediction, which may compromise safety and control performance.

To address this issue, we develop a robust `DeeP-LCC` method to control CAVs in mixed traffic. Our key idea is to robustify `DeeP-LCC` by considering all potential velocity error trajectories and formulating a robust problem. In particular, our main contributions include: 1) We propose a robust `DeeP-LCC` to handle the unknown velocity errors from the head vehicle. Our predictive controller will predict a series of future outputs based on the disturbance set and requires all of them to satisfy the safety constraint, thus providing enhanced safety performance. 2) We introduce two disturbance estimation methods, the constant velocity model and the constant acceleration model, based on the past disturbance data of the head vehicle. Our methods are able to provide satisfactory estimations of the future velocity errors and improve the control performance of robust `DeeP-LCC`. 3) We further provide efficient computational approaches for solving the robust optimization problem. We analyze and compare the complexity of two different solving methods from the robust optimization literature [20], [21] and provide a down-sampling method, adapted from [22], to further decrease the computational complexity. Numerical experiments validate the enhanced performance of the robust `DeeP-LCC` in reducing fuel consumption and improving driving safety while requiring less pre-collected data. For example, our

The work of X. Shang and Y. Zheng is supported by NSF ECCS-2154650 and NSF CMMI-2320697.

[1]X. Shang and Y. Zheng are with the Department of Electrical and Computer Engineering, University of California San Diego, CA 92093, USA. (x3shang@ucsd.edu; zhengy@ucsd.edu),

[2]J. Wang is with the Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI 48109, USA. (jiawe@umich.edu).

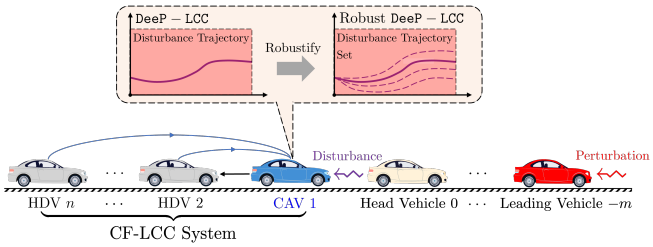



Fig. 1. Schematic of CF-LCC system. Original DeeP-LCC assumes one single trajectory for future disturbance, while the proposed robust DeeP-LCC explicitly addresses an estimated set of future disturbances. The notion $n$ and $m$ denote the number of HDVs behind the CAV and that ahead of the head vehicle, respectively.

robust DeeP-LCC only results in 4 and 0 emergencies out of 100 safety tests using small and large offline datasets, respectively; however, these numbers for standard DeeP-LCC [15] are 66 and 51 (which are unacceptably large).

The rest of the paper is organized as follows. Section II reviews the background on mixed traffic and DeeP-LCC. Section III presents our robust DeeP-LCC formulation. The disturbance set estimation methods and efficient computations are discussed in Section IV. Section V demonstrates our numerical results. We conclude the paper in Section VI.

## II. Data-driven Predictive Control in CF-LCC

In this section, we briefly review the DeeP-LCC [15] for a Car-Following LCC (CF-LCC) system [18]. As shown in Fig. 1, CF-LCC consists of one CAV, indexed as 1, and $n-1$ HDVs behind, indexed as $2,\dots,n$ from front to end. All these vehicles follow a head vehicle, indexed as 0, which is immediately ahead of the CAV. Such a CF-LCC system can be considered the smallest unit for general cascading mixed traffic systems [18], whose leading vehicle is indexed as $-m$. Note that our robust DeeP-LCC can be extended to general mixed traffic systems; see our extended report [23].

### A. Input/Output of CF-LCC System

For the $i$-th vehicle at time $t$, we denote its position, velocity and acceleration as $p_i(t)$, $v_i(t)$ and $a_i(t)$, $i=1,\dots,n$, respectively. We define the spacing between vehicle $i$ and its preceding vehicle as $s_i(t)=p_{i-1}(t)-p_i(t)$ and their relative velocity as $\dot{s}_i(t)=v_{i-1}(t)-v_i(t)$. In an equilibrium state, each vehicle moves at the same velocity $v^*$ with an equilibrium spacing $s_i^*$ that may vary from different vehicles.

In DeeP-LCC, we consider the error state of the traffic system. In particular, the velocity error and spacing error for each vehicle are defined as $\tilde{v}_i(t)=v_i(t)-v^*, \tilde{s}_i(t)=s_i(t)-s_i^*$. Then, we form the state $x\in\mathbb{R}^{2n}$ of the CF-LCC system by lumping the error states of all the vehicles

$$x(t)=[\tilde{s}_1(t),\tilde{v}_1(t),\tilde{s}_2(t),\tilde{v}_2(t),\dots,\tilde{s}_n(t),\tilde{v}_n(t)]^\mathsf{T}.$$

The spacing errors of HDVs are not directly measurable, since it is non-trivial to get the equilibrium spacing $s_i^*$ for HDVs due to the unknown car-following behaviors. By contrast, the equilibrium velocity $v^*$ can be estimated from the past velocity trajectory of the leading vehicle. Accordingly, the system output is formed by the velocity errors of all vehicles and the spacing error of the CAV, defined as

$$y(t)=[\tilde{v}_1(t),\tilde{v}_2(t),\dots,\tilde{v}_n(t),\tilde{s}_1(t)]^\mathsf{T}\in\mathbb{R}^{n+1}.$$

The input $u(t)\in\mathbb{R}$ of the system is defined as the acceleration of the CAV, as widely used in [4], [5]. Finally, the velocity error of the head vehicle 0 is regarded as an external disturbance signal $\epsilon=\tilde{v}_0(t)=v_0(t)-v^*\in\mathbb{R}$, and its past trajectory can be recorded, but its future trajectory is in general unknown. Based on the definitions of the system state, input, and output, after linearization and discretization, a state-space model of the CF-LCC system is in the form of

$$\begin{cases} x(k+1)=Ax(k)+Bu(k)+H\epsilon(k), \\ \quad y(k)=Cx(k), \end{cases} \tag{1}$$

where $k$ denotes the discrete time step. The details of the matrices $A,B,C,H$ can be found in [15, Section II-C].

Note that the parametric model (1) is non-trivial to accurately obtain due to the unknown HDVs' behavior. To address this issue, the recently proposed DeeP-LCC method directly uses the input/output trajectories for behavior prediction and controller design, thus bypassing the system identification process that is common in model-based methods.

### B. Data-Driven Representation of System Behavior

DeeP-LCC is an adaption of the standard DeePC [16] for mixed traffic control. It starts by forming a data-driven representation of the system with rich enough pre-collected offline data, which is further employed as a predictor to predict the dynamical behavior of CF-LCC (1). We recall a notion of persistent excitation [24] for offline data collection.

*Definition 1 (Persistently Exciting):* The sequence of signal $\omega=\mathrm{col}(\omega(1),\omega(2),\dots,\omega(T))$ with length $T$ ($T\in\mathbb{N}$) is persistently exciting of order $L$ ($L<T$) if its associated Hankel matrix with depth $L$ has full row rank:

$$\mathcal{H}_L(\omega)=\begin{bmatrix}\omega(1) & \omega(2) & \cdots & \omega(T-L+1)\\ \omega(2) & \omega(3) & \cdots & \omega(T-L+2)\\ \vdots & \vdots & \ddots & \vdots\\ \omega(L) & \omega(L+1) & \cdots & \omega(T)\end{bmatrix}.$$

We begin with collecting an input/output trajectory of length $T$ for the CF-LCC system offline:

$$\begin{aligned} u^\mathrm{d}&=\mathrm{col}(u^\mathrm{d}(1),u^\mathrm{d}(2),\dots,u^\mathrm{d}(T))\in\mathbb{R}^T,\\ \epsilon^\mathrm{d}&=\mathrm{col}(\epsilon^\mathrm{d}(1),\epsilon^\mathrm{d}(2),\dots,\epsilon^\mathrm{d}(T))\in\mathbb{R}^T,\\ y^\mathrm{d}&=\mathrm{col}(y^\mathrm{d}(1),y^\mathrm{d}(2),\dots,y^\mathrm{d}(T))\in\mathbb{R}^{(n+1)T}. \end{aligned}$$

We then use the offline collected data to form a Hankel matrix of order $L$, which is partitioned as follows:

$$\begin{bmatrix}U_\mathrm{P}\\ U_\mathrm{F}\end{bmatrix}:=\mathcal{H}_L(u^\mathrm{d}),\ \begin{bmatrix}E_\mathrm{P}\\ E_\mathrm{F}\end{bmatrix}:=\mathcal{H}_L(\epsilon^\mathrm{d}),\ \begin{bmatrix}Y_\mathrm{P}\\ Y_\mathrm{F}\end{bmatrix}:=\mathcal{H}_L(y^\mathrm{d}), \tag{2}$$

where $U_\mathrm{P}$ and $U_\mathrm{F}$ contains the first $T_\mathrm{ini}$ rows and the last $N$ rows of $\mathcal{H}_L(u^\mathrm{d})$, respectively (similarly for $E_\mathrm{P}$ and $E_\mathrm{F}$, $Y_\mathrm{P}$ and $Y_\mathrm{F}$). The Hankel matrices (2) can be used to construct the online behavior predictor for predictive control. We then have the following result.

*Proposition 1 ([15, Proposition 2]):* At time step $k$, we collect the most recent past input sequence $u_\mathrm{ini}$ with length $T_\mathrm{ini}$, and let the future input sequence $u$ with length $N$ as

$$\begin{aligned} u_\mathrm{ini}&=\mathrm{col}(u(k-T_\mathrm{ini}),u(k-T_\mathrm{ini}+1),\dots,u(k-1)),\\ u&=\mathrm{col}(u(k),u(k+1),\dots,u(k+N-1)). \end{aligned}$$

The notations $\epsilon_{\text{ini}}$, $\epsilon$, $y_{\text{ini}}$ and $y$ are denoted similarly. If the input trajectory $u^{\text{d}}$ is persistently exciting of order $L+2n$ (where $L = T_{\text{ini}} + N$), then the sequence $\text{col}(u_{\text{ini}}, \epsilon_{\text{ini}}, y_{\text{ini}}, u, \epsilon, y)$ is a valid trajectory with length $L$ of (1) if and only if there exists a vector $g \in \mathbb{R}^{T-L+1}$ such that

$$\text{col}(U_{\text{P}}, E_{\text{P}}, Y_{\text{P}}, U_{\text{F}}, E_{\text{F}}, Y_{\text{F}})g = \text{col}(u_{\text{ini}}, \epsilon_{\text{ini}}, y_{\text{ini}}, u, \epsilon, y). \quad (3)$$

If $T_{\text{ini}} \geq 2n$, then $y$ is unique for any $(u_{\text{ini}}, y_{\text{ini}}, u, \epsilon)$.

This proposition provides a data-driven representation (3) for the CF-LCC system: all valid trajectories can be constructed by a linear combination of rich enough pre-collected trajectories. Thus, we can predict the future output $y$ using trajectories $(u^{\text{d}}, \epsilon^{\text{d}}, y^{\text{d}})$, given the future input $u$, disturbance $\epsilon$ and initial condition $(u_{\text{ini}}, \epsilon_{\text{ini}}, y_{\text{ini}})$.

### C. DeeP-LCC Formulation

Using the data-driven representation (3), the DeeP-LCC in [15] solves an optimization problem at each time step:

$$\min_{g,\sigma_y,u,\epsilon,y} \quad V(u,y) + \lambda_g ||g||_2^2 + \lambda_y ||\sigma_y||_2^2 \quad (4a)$$

$$\text{subject to} \quad \begin{bmatrix} U_{\text{P}} \\ E_{\text{P}} \\ Y_{\text{P}} \\ U_{\text{F}} \\ E_{\text{F}} \\ Y_{\text{F}} \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ \epsilon_{\text{ini}} \\ y_{\text{ini}} \\ u \\ \epsilon \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \sigma_y \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (4b)$$

$$\tilde{s}_{\min} \leq G_1 y \leq \tilde{s}_{\max}, \quad (4c)$$

$$u_{\min} \leq u \leq u_{\max}, \quad (4d)$$

$$\epsilon = \epsilon_{\text{est}}, \quad (4e)$$

where $G_1 = I_N \otimes \begin{bmatrix} 0_{1\times n}, & 1 \end{bmatrix}$ selects the spacing error of the CAV from the output, $[\tilde{s}_{\min}, \tilde{s}_{\max}]$ is the safe spacing error range of CAV, $[u_{\min}, u_{\max}]$ is the physical limitation of the acceleration and $\epsilon_{\text{est}}$ is the estimation of the future velocity errors of the head vehicle 0.

For the cost function (4a), $V(u,y)$ penalizes the output deviation from equilibrium states and the energy of the input:

$$V(u,y) = ||u||_R^2 + ||y||_Q^2,$$

with $R \in \mathbb{S}_+^{N\times N}$ and $Q \in \mathbb{S}_+^{N(n+1)\times N(n+1)}$. There are two regularization terms $\|g\|_2^2$ and $\|\sigma_y\|_2^2$ in the cost function with weight coefficients $\lambda_g, \lambda_y$. Also, a slack variable $\sigma_y$ is added to the data-driven representation (4b). Note that the original data-driven behavior representation (3) is only applicable to linear systems with noise-free data. The regularization herein is commonly used for nonlinear systems with stochastic noises, and we refer interested readers to [15], [16] for detailed discussions.

*Remark 1 (Robustification):* In the standard DeeP-LCC formulation [15], it is assumed that *the future velocity error for the head vehicle remains zero*, *i.e.*, $\epsilon_{\text{est}} = 0$. However, this assumption hardly stands since in real-world traffic, strong oscillations may happen, particularly during the occurrence of traffic waves. An inaccurate estimation can cause a mismatch between the prediction and the real traffic behavior, which may not only degrade the control performance but also pose safety concerns (*e.g.*, rear-end collisions). □

**Algorithm 1** Robust DeeP-LCC

**Input:** Pre-collected offline data $(u^{\text{d}}, \epsilon^{\text{d}}, y^{\text{d}})$, initial time step $k_0$, terminal time step $k_f$;

1: Construct data Hankel matrices (2) for input, disturbance, and output as $U_{\text{P}}, U_{\text{F}}, E_{\text{P}}, E_{\text{F}}, Y_{\text{P}}, Y_{\text{F}}$;
2: Initialize the most recent past traffic data $(u_{\text{ini}}, \epsilon_{\text{ini}}, y_{\text{ini}})$ before the initial time $k_0$;
3: **while** $k_0 \leq k \leq k_f$ **do**
4: Estimate $\mathcal{W}$ from $\epsilon_{\text{ini}}$;
5: Solve (5) for optimal future control input sequence $u^* = \text{col}(u^*(k), u^*(k+1), \ldots, u^*(k+N-1))$;
6: Apply the input $u(k) \leftarrow u^*(k)$ to the CAV;
7: $k \leftarrow k+1$;
8: Update past traffic data $(u_{\text{ini}}, \epsilon_{\text{ini}}, y_{\text{ini}})$;
9: **end while**

## III. Tractable Robust DeeP-LCC Formulation

In this section, we present a new framework of robust DeeP-LCC to control the CAV in the CF-LCC system which can properly address unknown future velocity errors, leading to enhanced performance and safety.

### A. Robust DeeP-LCC Formulation

As shown in Fig. 1, instead of estimating one single disturbance trajectory $\epsilon = \epsilon_{\text{est}}$ in DeeP-LCC, we introduce a disturbance set $\mathcal{W}$ as the estimation, *i.e.*, $\epsilon \in \mathcal{W}$, which, by valid design (see details in Section IV), will contain the real trajectory with a much higher possibility.

Our key idea is to plan over the worst trajectory in $\mathcal{W}$ for predictive control, leading to a robust optimization problem

$$\begin{aligned} \min_{g,\sigma_y,u,y} \ \max_{\epsilon\in\mathcal{W}} \quad & V(u,y) + \lambda_g ||g||_2^2 + \lambda_y ||\sigma_y||_2^2 \\ \text{subject to} \quad & \text{(4b), (4c), (4d).} \end{aligned} \quad (5)$$

Compared with the original formulation in (4), the robust formulation (5) promises to provide better control performance and higher safety guarantees since the gap between online prediction and real implementation has been reduced. As a trade-off, note that the complexity of the optimization problem is increased, and we also need to estimate $\mathcal{W}$ properly. Both issues will be discussed in the sections below.

For implementation, the optimization problem (5) is solved in a receding horizon manner at each time step $k$ and $\mathcal{W}$ is re-estimated iteratively based on the updated velocity errors of the head vehicle (see Section IV). Algorithm 1 lists the overall procedure of robust DeeP-LCC.

### B. Reformulations of Min-max Optimization

The min-max optimization problem (5) is solved at each iteration of Algorithm 1, but standard solvers are not applicable with respect to its current form. We proceed to present a sequence of reformulation (and relaxations) for (5), which further allows for efficient computations in Section IV.

We first eliminate the equality constraint by expressing $g$ and $y$ in terms of $u$, $\sigma_y$ and $\epsilon$ to :

$$g = H_{\text{p}}^{\dagger} b + H_{\text{p}}^{\perp} z, \quad (6a)$$

$$y = Y_{\text{F}} g = Y_{\text{F}} H_{\text{p}}^{\dagger} b + Y_{\text{F}} H_{\text{p}}^{\perp} z, \quad (6b)$$

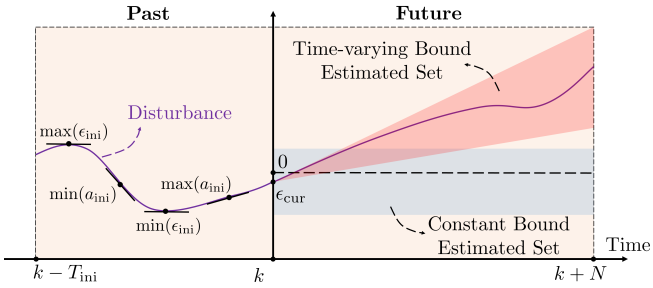



Fig. 2. Schematic of two disturbance estimation methods. The purple line represents the actual disturbance trajectory with known past part and unknown future part. In the past region, the black line segment denotes the information needed for estimation. In the future region, the black dashed line represents the zero estimation while the red and blue regions denote the time-varying bound and constant bound estimated sets, respectively.

where $H_{\text{p}} = \text{col}(U_{\text{p}}, E_{\text{p}}, Y_{\text{p}}, U_{\text{F}}, E_{\text{F}})$ with $H_{\text{p}}^{\dagger}$ denoting its pseudo-inverse, $H_{\text{p}}^{\perp} = I - H_{\text{p}}^{\dagger} H_{\text{p}}$, $b = \text{col}(u_{\text{ini}}, \epsilon_{\text{ini}}, y_{\text{ini}} + \sigma_y, u, \epsilon)$, and $z \in \mathbb{R}^{T-L+1}$. For simplicity, we set $z = 0$ in the following derivation, which decreases the complexity of the optimization problem but also reduces the feasible set. From the simulations in Section V, we note that this simplification already provides satisfactory control performance. Then, the min-max robust problem (5) becomes:

$$\min_{u,\sigma_y} \max_{\epsilon \in \mathcal{W}} \quad x^{\mathsf{T}} M x + d^{\mathsf{T}} x + c_0 \tag{7a}$$

$$\text{subject to} \quad \tilde{s}_{\min} \le P_1 x + c_1 \le \tilde{s}_{\max}, \tag{7b}$$

$$u_{\min} \le P_2 x \le u_{\max}, \tag{7c}$$

where $x = \text{col}(u, \sigma_y, \epsilon)$ denotes the decision variable and $M, d, c_0, P_1, P_2, c_1$ depend on problem data [23, Appx. A].

Without loss of generality, we eliminate the constant $c_0$. We finally consider $\epsilon$ as an uncertainty parameter, and transform problem (7) into its epi-graph form

$$\min_{x,t} \quad t$$

$$\text{subject to} \quad x^{\mathsf{T}} M x + d^{\mathsf{T}} x \le t, \quad \forall \epsilon \in \mathcal{W}, \tag{8a}$$

$$\tilde{s}_{\min} \le P_1 x + c_1 \le \tilde{s}_{\max}, \quad \forall \epsilon \in \mathcal{W}, \tag{8b}$$

$$u_{\min} \le P_2 x \le u_{\max}. \tag{8c}$$

Compared with (7), the formulation (8) requires its feasible solutions to satisfy the safety constraint for all disturbance trajectories in $\mathcal{W}$ which enhances the safety. On the other hand, the stricter safety constraint further increases the complexity, which will be addressed in Section IV-B.

*Remark 2 (Uncertainty Quantification):* We require an accurate and non-conservative estimation of $\mathcal{W}$ for velocity error trajectories. The actual disturbance trajectory should be inside or close to $\mathcal{W}$; otherwise, a gap between online prediction and real traffic behavior may still exist. A conservative estimation is not preferred either, which will shrink the feasible solution set and degrade the control performance. □

## IV. Disturbance Estimation and Efficient Computation

In this section, we first introduce two disturbance estimation methods based on different assumptions of human driving behaviors. We then present two solving methods of (8) and compare their complexity. Also, we provide a down-sampling method of low-dimensional approximation for the disturbance set $\mathcal{W}$ for real-time computation.

### A. Uncertainty Quantification

In our problem, the estimated disturbance set is considered an $N$-dimensional polytope that is

$$\mathcal{W} = \{\epsilon \in \mathbb{R}^N | A_\epsilon \epsilon \le b_\epsilon\}, \tag{9}$$

where $A_\epsilon = [I; -I]$, $b_\epsilon = [\epsilon_{\max}; -\epsilon_{\min}]$ and $\epsilon_{\max}, \epsilon_{\min}$ are the upper and lower bound vectors of $\epsilon$. The key part of estimating the disturbance set becomes estimating its (time-varying) bounds from the past velocity errors $\epsilon_{\text{ini}}$.

We propose two different estimation methods (see Fig. 2 for illustration) and analyze their performance:

*1) Constant disturbance bounds:* We assume that the disturbance (velocity error) of the head vehicle will not have a large deviation from its current value in a short time period based on the constant velocity model, and the disturbance variation for the future disturbance trajectory is close to its past trajectory. From the historical disturbance values $\epsilon_{\text{ini}}$, we can get the value of the current disturbance, *i.e.*, $\epsilon_{\text{ini}}(\text{end})$, and estimate the disturbance variation as $\Delta\epsilon_{\text{low}} = \min(\epsilon_{\text{ini}}) - \text{mean}(\epsilon_{\text{ini}})$ and $\Delta\epsilon_{\text{up}} = \max(\epsilon_{\text{ini}}) - \text{mean}(\epsilon_{\text{ini}})$. Then, the estimated bound of the future disturbance is given by

$$\epsilon_{\min} = \epsilon_{\text{cur}} + \Delta\epsilon_{\text{low}}, \ \ \epsilon_{\max} = \epsilon_{\text{cur}} + \Delta\epsilon_{\text{up}}.$$

*2) Time-varying disturbance bounds:* We can also assume the acceleration of the head vehicle will not deviate significantly from its current value based on the constant acceleration model, and its variation in the future is close to the variation in the past. We first get the past acceleration information from $\epsilon_{\text{ini}}$ as $a_{\text{ini}}(k) = \frac{\epsilon_{\text{ini}}(k+1) - \epsilon_{\text{ini}}(k)}{\Delta t}$ where $\Delta t$ is the sampling time period. Then, using a similar procedure as in the previous approach, the acceleration variation bound is estimated as $\Delta a_{\text{low}} = \min(a_{\text{ini}}) - \text{mean}(a_{\text{ini}})$ and $\Delta a_{\text{up}} = \max(a_{\text{ini}}) - \text{mean}(a_{\text{ini}})$. Thus, the future disturbance in an arbitrary time step $k$ is bounded by the following inequalities:

$$\epsilon_{\text{ini}}(\text{end}) + (a_{\text{cur}} + \Delta a_{\text{low}}) \cdot k\Delta t \le \epsilon(k) \le \epsilon_{\text{ini}}(\text{end}) + (a_{\text{cur}} + \Delta a_{\text{up}}) \cdot k\Delta t.$$

Fig. 2 illustrates the two disturbance estimation methods. It is clear that a large gap exists between the actual disturbance trajectory and the zero line. For the constant disturbance bounds, the actual disturbance trajectory stays in the estimated set in the short term but will deviate from the set over time. For the second method using time-varying disturbance bounds, it includes the actual trajectory in the estimated set in this case but with a relatively conservative bound at the end of the time period. In most of our numerical simulations, the time-varying disturbance bounds outperform the constant disturbance bounds because traffic waves usually have high amplitude with low frequency (see details in our report [23]).

### B. Efficient Computations

Upon estimating $\mathcal{W}$, the robust optimization problem (8) is well-defined. We here adapt standard robust optimization techniques to solve (8) and compare their complexity.

**M1: Vertex-based**. Our first method utilizes constraints evaluated at vertices of $\mathcal{W}$ to replace the robust constraints.

The compact polytope $\mathcal{W}$ can be represented as the convex hull of its extreme points as

$$\mathcal{W} = \text{conv}(\omega_1, \dots, \omega_{n_\text{v}}), \tag{10}$$

where $n_\text{v}$ denotes the number of extreme points, and its value is $2^N$ if no low-dimensional approximation is applied. Using this representation, we can rewrite problem (8) as

$$\begin{aligned} \min_{x,t} \quad & t \\ \text{subject to} \quad & x_j^\mathsf{T} M x_j + d^\mathsf{T} x_j \le t,\ j = 1, \dots, n_\text{v}, && \text{(11a)} \\ & \tilde{s}_{\min} \le P_1 x_j + c_1 \le \tilde{s}_{\max},\ j = 1, \dots, n_\text{v}, && \text{(11b)} \\ & u_{\min} \le P_2 x \le u_{\max}, && \text{(11c)} \end{aligned}$$

where $x_j$ represents the decision variable when the uncertainty parameter $\epsilon$ is fixed to one of the extreme points $w_j$ and the expression becomes $\text{col}(u, \sigma_y, w_j)$.

**M2: Duality-based**. The second method treats robust constraint (8a) the same as the first method, but forms (8b) as a sub-level optimization problem and then changes it into its dual problem to combine both levels. For example, the right hand inequality of (8b) can be reformulated as

$$\tilde{s}_{\max} \ge \max_{\epsilon \in \mathcal{W}} \ p_l^\mathsf{T} x + c_{1,l},\ l = 1, \dots, N, \tag{12}$$

where $p_l^\mathsf{T}$ and $c_{1,l}$ is the $l$-th row vector and element in $P_1$ and $c_{1,l}$, respectively. Given the origin representation $\mathcal{W}$ in (9), the right-hand side of (12) is a linear program (LP). Then, we can change them to their dual problems and the strong duality of LPs ensures the new formulation is equivalent to (12). The bi-level optimization problem becomes a min-min problem and we can combine both levels[1]. The original problem (8) can then be equivalently reformulated as

$$\begin{aligned} \min_{x_\text{d}, t, \lambda_1, \lambda_2} \quad & t \\ \text{subject to} \quad & p_{l,\text{d}}^\mathsf{T} x_\text{d} + b_\epsilon^\mathsf{T} \lambda_{l,1} + c_{1,l} \le \tilde{s}_{\max}, && \text{(13a)} \\ & A_\epsilon^\mathsf{T} \lambda_{l,1} - p_{l,\epsilon} = 0, && \text{(13b)} \\ & -p_{l,\text{d}}^\mathsf{T} x_\text{d} + b_\epsilon^\mathsf{T} \lambda_{l,2} - c_{1,l} \le -\tilde{s}_{\min}, && \text{(13c)} \\ & A_\epsilon^\mathsf{T} \lambda_{l,2} + p_{l,\epsilon} = 0, && \text{(13d)} \\ & \lambda_{l,1} \ge 0, \lambda_{l,2} \ge 0,\ l = 1, 2, \dots, N, && \text{(13e)} \\ & \text{(11a)}, \text{(11c)}, \end{aligned}$$

where $x_\text{d}$ is the decision variable $\text{col}(u, \sigma_y)$, and $\lambda_{l,1}, \lambda_{l,2} \in \mathbb{R}^{2N}$ are dual variables with $\lambda_1 = \text{col}(\lambda_{1,1}, \lambda_{2,1}, \dots, \lambda_{N,1})$, $\lambda_2 = \text{col}(\lambda_{1,2}, \lambda_{2,2}, \dots, \lambda_{N,2})$; parameters $c_{1,l}, p_l$ are the same as (12) and $p_{j,\text{d}}$ represents $\text{col}(p_{l,u}, p_{l,\sigma_y})$ with $p_l$ subdivided into $\text{col}(p_{l,u}, p_{l,\sigma_y}, p_{l,\epsilon})$ corresponding to $u, \sigma_y$ and $\epsilon$.

*Theorem 1:* Suppose (8) is feasible and its uncertainty set $\mathcal{W}$ is a polytope. Problems (8), (11) and (13) are equivalent.

The equivalence between (8) and (11) is relatively straightforward. It requires standard duality arguments to establish the equivalence between (8) and (13) (see details in [23]).

Both (11) and (13) are standard convex optimization problems, which can be solved using standard solvers (*e.g.*, Mosek [25]). We here discuss the complexity of the above two methods; see Table I. The main difference lies in the different formulations of (8b), *i.e.*, (11b) and (13a) - (13e).

[1]This operation is standard; we refer the interested reader to Section 2.1 of https://zhengy09.github.io/ECE285/lectures/L17.pdf.

TABLE I

COMPLEXITY COMPARISON BETWEEN (11) AND (13)

| | Decision Variables Number | Constraints Number |
|---|---|---|
| **M1** | $(n+1)T_\text{ini} + N + 1$ | $2^N + N \cdot 2^{N+1} + 2N$ |
| **M2** | $(n+1)T_\text{ini} + N + 1 + 4N^2$ | $2^N + 2N(3N+2)$ |
| **M1 (L)** | $(n+1)T_\text{ini} + N + 1$ | $2^{n_\epsilon} + N \cdot 2^{n_\epsilon+1} + 2N$ |
| **M2 (L)** | $(n+1)T_\text{ini} + N + 1 + 4Nn_\epsilon$ | $2^{n_\epsilon} + 2N(3n_\epsilon+2)$ |

[1] M1 and M2 represent the vertex-based and the duality-based method. The last two rows denote their complexities after down-sampling.

In **M1**, (11b) represents $N \cdot 2^{N+1}$ inequality constraints while (13a) - (13e) together represent $2N(3N+1)$ inequality constraints in **M2**. The value $N \cdot 2^{N+1}$ is much larger than $2N(3N+2)$ when the prediction horizon $N$ is large.

### C. Down-sampling Strategy

We here discuss a down-sampling strategy that is adapted from [22] to relieve the exponential growth of the number of constraints. It approximates the $N$-dimensional disturbance trajectory by choosing one point for every $T_\text{s}$ steps along it and performing linear interpolation. We denote the low-dimensional representation of the future disturbance trajectory as $\tilde{\epsilon} \in \mathbb{R}^{n_\epsilon}$ where $n_\epsilon = (\lfloor \frac{N-2}{T_\text{s}} \rfloor + 2)$. An approximated representation $\hat{\epsilon}$ of $\epsilon$ can be derived as

$$\hat{\epsilon}^{(k)} = \begin{cases} \tilde{\epsilon}^{(\bar{k}+1)} + ((k-1) \bmod T_\text{s}) \times \dfrac{\tilde{\epsilon}^{(\bar{k}+2)} - \tilde{\epsilon}^{(\bar{k}+1)}}{T_\text{s}}, \\ \qquad\qquad\qquad\qquad 1 \le k \le \tilde{k} \cdot T_s \\ \tilde{\epsilon}^{(\tilde{k}+1)} + (k - \tilde{k} \cdot T_\text{s} - 1) \times \dfrac{\tilde{\epsilon}^{(\tilde{k}+2)} - \tilde{\epsilon}^{(\tilde{k}+1)}}{N - \tilde{k} \cdot T_\text{s} - 1}, \\ \qquad\qquad\qquad\qquad \tilde{k} \cdot T_s < k \le N \end{cases}$$

where $\bar{k} = \lfloor \frac{k-1}{T_\text{s}} \rfloor$ and $\tilde{k} = \lfloor \frac{N-2}{T_\text{s}} \rfloor$. Then, we can use $\tilde{\epsilon} \in \mathbb{R}^{n_\epsilon}$ to represent $\epsilon \in \mathbb{R}^N$ as

$$\epsilon \approx \hat{\epsilon} = E_\epsilon \tilde{\epsilon}, \tag{14}$$

where $\tilde{\epsilon} \in \tilde{\mathcal{W}}$ and $\tilde{\mathcal{W}}$ can be estimated using the same methods we introduced before. Also, substituting (14) into our previous derivation will not affect its correctness.

The complexities of both methods (11) and (13) after using low-dimensional approximation are updated in the last two rows of Table I. Theoretically, with the same computational resource, the duality-based method allows us to choose a larger $n_\epsilon$ because the coefficient of its exponential growth term $2^{n_\epsilon}$ is 1 while it is $2N+1$ for the vertex-based method. In our implementation, $n_\epsilon$ is usually chosen as a small number to ensure real-time computational performance and these two methods might not have obvious differences.

## V. TRAFFIC SIMULATIONS

In this section, we carry out two nonlinear and non-deterministic traffic simulations to test the performance of robust `DeeP-LCC` with the time-varying disturbance bound in controlling the CF-LCC system in mixed traffic. We implement an automatic routine to transform (13) into standard conic programs[2], which are solved by Mosek [25]. More numerical results can be found in the extended version [23].

[2]Our open-source implementation is available at https://github.com/soc-ucsd/Decentralized-DeeP-LCC/.

### A. Experimental Setup

The car-following behaviors of HDVs are modeled by the nonlinear OVM model in [8], and a noise signal following the uniform distribution of $\mathbb{U}[-0.1, 0.1]\,\mathrm{m/s^2}$ is added to the acceleration for each HDV. For the CF-LCC system in mixed traffic, we consider the case where the CAV 1 is followed by 4 HDVs, and there are 3 vehicles in front of the head vehicle 0, *i.e.*, $n = 5$, $m = 3$ in Fig. 1. We use the following parameters in both `DeeP-LCC` and robust `DeeP-LCC`:

(1) *Offline data collection*: lengths of pre-collected data sets are $T = 500$ for a small data set and $T = 1500$ for a large data set with $\Delta t = 0.05$s. They are collected around the equilibrium state of the system with velocity $15\,\mathrm{m/s}$. Both $u^{\mathrm{d}}$ and $\epsilon^{\mathrm{d}}$ are generated by a uniform distributed signal of $\mathbb{U}[-1, 1]$ which satisfies the persistent excitation requirement in Proposition 1.

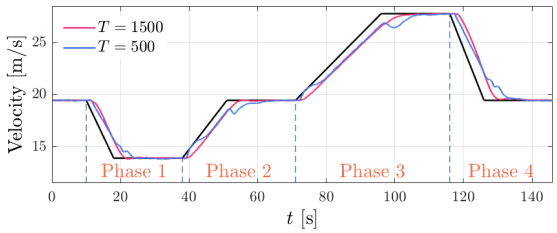
(2) *Online predictive control*: the initial signal sequence and the prediction horizon are set to $T_{\mathrm{ini}} = 20$ and $N = 50$. For the objective function in (5), we have $R = 0.1 I_N$ and $Q = I_N \otimes \mathrm{diag}(Q_v, w_s)$ where $Q_v = \mathrm{diag}(1, \ldots, 1) \in \mathbb{R}^n$ and $w_s = 0.5$. The regularized parameters are set to $\lambda_g = 100$ and $\lambda_y = 10000$. The spacing constraints for CAV are set as $s_{\max} = 40\,\mathrm{m}$, $s_{\min} = 5\,\mathrm{m}$, and the bound of the spacing error is updated in each iteration as $\tilde{s}_{\max} = s_{\max} - s^*$ and $\tilde{s}_{\min} = s_{\min} - s^*$. Note that $s^*$ is also updated in each time step according to the current equilibrium state estimated by the leading vehicle's past trajectory [15]. The acceleration limits are set as $a_{\max} = 2\ \mathrm{m/s^2}$ and $a_{\min} = -5\ \mathrm{m/s^2}$.

### B. Numerical Results

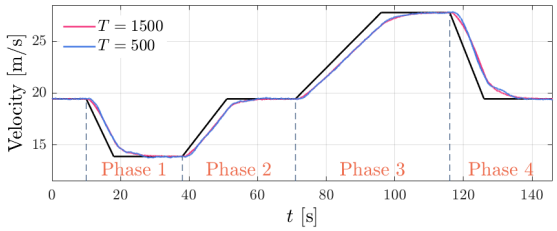
**Experiment A:** We first validate the control performance of robust `DeeP-LCC` in a comprehensive simulation scenario which is motivated by New European Driving Cycle (NEDC) [26]. We design the velocity trajectory of the leading vehicle as the black profile in Fig. 3 and calculate the fuel consumption of the 5 following vehicles in CF-LCC system using the numerical model in [27] for evaluation.

The velocity profiles of robust `DeeP-LCC` and original `DeeP-LCC` with different sizes of data sets are shown in Fig. 3. Both methods allow for the CAV to track the desired velocity when using a large data set (see red curves in Fig. 3). However, in the case of using a small data set, the degradation of control performance for `DeeP-LCC` is apparent, and there are some undesired oscillations (see blue curves in Fig. 3(a)), while robust `DeeP-LCC` remains a smooth velocity profile (see blue curves in Fig. 3(b)). Such performance degradation is highly related to the mismatch between the online prediction and real system behavior, caused by representation and estimation errors. The robust `DeeP-LCC` allows for a relatively small estimation error, and provides more margin for potential representation errors which is the main reason that the robust `DeeP-LCC` performs better than `DeeP-LCC` for a relatively small data set.

Table II lists fuel consumption results when using the large data set. Both robust `DeeP-LCC` and `DeeP-LCC` reduce fuel consumption compared with the case with all HDVs, and the improvement in the braking phase (Phases 1 and 4) is higher




(a) `DeeP-LCC`




(b) Robust `DeeP-LCC`

Fig. 3. Velocity profiles in Experiment A. The black profile denotes the leading vehicle. The red profile and the blue profile represent `DeeP-LCC` control with data sets of size $T = 1500$ and $T = 500$, respectively. (a) The CAV utilizes `DeeP-LCC`. (b) The CAV utilizes robust `DeeP-LCC`.

TABLE II

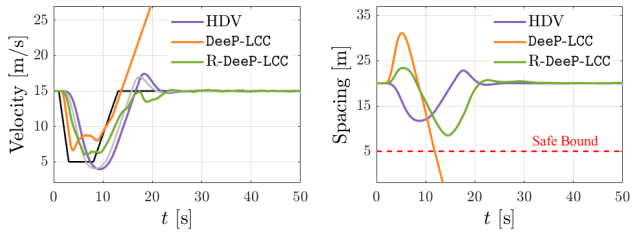
FUEL CONSUMPTION IN EXPERIMENT A (UNIT: mL)

| | All HDVs | `DeeP-LCC` | Robust `DeeP-LCC` |
|---|---|---|---|
| Phase 1 | 145.59 | 141.02 (↓ 3.14%) | 135.60 (↓ **6.86%**) |
| Phase 2 | 314.77 | 312.95 (↓ 0.58%) | 311.83 (↓ **0.94%**) |
| Phase 3 | 725.28 | 723.95 (↓ 0.18%) | 722.88 (↓ **0.33%**) |
| Phase 4 | 259.05 | 246.16 (↓ 4.97%) | 237.89 (↓ **8.17%**) |
| Total Process | 1530.15 | 1509.6 (↓ 1.54%) | 1493.6(↓ **2.39%**) |

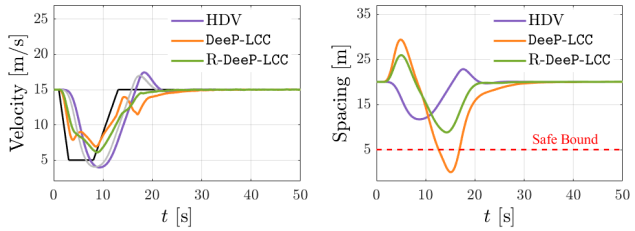
than the accelerating phases (Phases 2 and 3). Moreover, we note that robust `DeeP-LCC` achieves better fuel economy than `DeeP-LCC` in all phases.

**Experiment B:** We further validate the safety performance of robust `DeeP-LCC` in the braking scenario. In this experiment, the leading vehicle that moves at $15\,\mathrm{m/s}$ will brake with the maximum deceleration $-5\,\mathrm{m/s^2}$, stay at $5\,\mathrm{m/s}$ for a while, and then speed up back to $15\,\mathrm{m/s}$. We conduct 100 experiments each with randomly generated small and large datasets. We define "violation" and "emergency" as the case where the CAV's spacing deviates more than $1\,\mathrm{m}$ and $5\,\mathrm{m}$ from the safe range (set as $5\,\mathrm{m}$ to $40\,\mathrm{m}$). An "emergency" occurs when a rear-end collision happens or the spacing of the CAV becomes too large, reducing traffic capacity.

The results are presented in Table III, which clearly shows that `DeeP-LCC` has a much higher violation and emergency rate for both small and large data sets. On the other hand, the robust `DeeP-LCC` can provide a remarkably low violation and emergency rate using small data sets which are 5% and 4%, and both of them are decreased to 0% when using large data sets. Fig. 4 displays two typical examples. When using a large data set, both methods exhibit smaller velocity fluctuations compared with the case of all human drivers. It can be clearly observed that the CAV controlled by robust `DeeP-LCC` always stays inside the safety bound for both large and small data sets, while `DeeP-LCC` is likely to lead to a rear-end collision for the small data set, and still violate the safe bound even with the large data set.




(a) Small offline data set with $T = 500$




(b) Large offline data set with $T = 1500$

Fig. 4. Simulation results in Experiment B. The black and the gray profiles represent the leading vehicle and the head vehicle, respectively. The orange and the green profiles correspond to DeeP-LCC and robust DeeP-LCC, respectively, while the purple profile corresponds to the all HDV case. (a) and (b) show the velocity and spacing profiles at different sizes of data sets.

TABLE III

COLLISION AND SAFETY CONSTRAINT VIOLATION RATE

| | DeeP-LCC | | Robust DeeP-LCC | |
|---|---|---|---|---|
| | $T = 500$ | $T = 1500$ | $T = 500$ | $T = 1500$ |
| Violation Rate | 74% | 62% | **5%** | **0%** |
| Emergency Rate | 66% | 51% | **4%** | **0%** |

## VI. CONCLUSION

In this paper, we have proposed the robust DeeP-LCC for CAV control in mixed traffic. The robust formulation and disturbance set estimation methods together provide a strong safety guarantee, improve the control performance, and allow for the applicability of a smaller data set. Efficient computational methods are also provided for the real-time implementation. Extensive traffic simulations have validated the performance of robust DeeP-LCC in comprehensive and braking scenarios. Interesting future directions include learning-based estimation for future disturbances, incorporation of communication-delayed traffic data, and extension to large-scale mixed traffic scenarios.

## REFERENCES


[1] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S.-i. Tadaki, and S. Yukawa, "Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam," *New Journal of Physics*, vol. 10, no. 3, p. 033001, 2008.

[2] S. E. Li, Y. Zheng, K. Li, Y. Wu, J. K. Hedrick, F. Gao, and H. Zhang, "Dynamical modeling and distributed control of connected and automated vehicles: Challenges and opportunities," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 46–58, 2017.

[3] Y. Zheng, S. E. Li, J. Wang, D. Cao, and K. Li, "Stability and scalability of homogeneous vehicular platoon: Study on the influence of information flow topologies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 14–26, 2015.

[4] Y. Zheng, J. Wang, and K. Li, "Smoothing traffic flow via control of autonomous vehicles," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3882–3896, 2020.

[5] G. Orosz, "Connected cruise control: modelling, delay effects, and nonlinear behaviour," *Vehicle System Dynamics*, vol. 54, no. 8, pp. 1147–1176, 2016.

[6] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama, "Dynamical model of traffic congestion and numerical simulation," *Physical Review E*, vol. 51, no. 2, p. 1035, 1995.

[7] I. G. Jin and G. Orosz, "Optimal control of connected vehicle systems with communication delay and driver reaction time," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 2056–2070, 2016.

[8] J. Wang, Y. Zheng, Q. Xu, J. Wang, and K. Li, "Controllability analysis and optimal control of mixed traffic flow with human-driven and autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7445–7459, 2021.

[9] S. S. Mousavi, S. Bahrami, and A. Kouvelas, "Synthesis of output-feedback controllers for mixed traffic systems in presence of disturbances and uncertainties," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 6450–6462, 2023.

[10] S. Feng, Z. Song, Z. Li, Y. Zhang, and L. Li, "Robust platoon control in mixed traffic flow based on tube model predictive control," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 711–722, 2021.

[11] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, "Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 899–910, 2016.

[12] C. Zhao, H. Yu, and T. G. Molnar, "Safety-critical traffic control by connected automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 154, p. 104230, 2023.

[13] C. Wu, A. R. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: A modular learning framework for mixed autonomy traffic," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 1270–1286, 2021.

[14] M. Huang, Z.-P. Jiang, and K. Ozbay, "Learning-based adaptive optimal control for connected vehicles in mixed traffic: robustness to driver reaction time," *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 5267–5277, 2020.

[15] J. Wang, Y. Zheng, K. Li, and Q. Xu, "DeeP-LCC: Data-enabled predictive leading cruise control in mixed traffic flow," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 6, pp. 2760–2776, 2023.

[16] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the deepc," in *18th European Control Conference (ECC)*. IEEE, 2019, pp. 307–312.

[17] I. Markovsky and F. Dörfler, "Behavioral systems theory in data-driven analysis, signal processing, and control," *Annual Reviews in Control*, vol. 52, pp. 42–64, 2021.

[18] J. Wang, Y. Zheng, C. Chen, Q. Xu, and K. Li, "Leading cruise control in mixed traffic flow: System modeling, controllability, and string stability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 861–12 876, 2021.

[19] J. Wang, Y. Zheng, J. Dong, C. Chen, M. Cai, K. Li, and Q. Xu, "Implementation and experimental validation of data-driven predictive control for dissipating stop-and-go waves in mixed traffic," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 4570–4585, 2023.

[20] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM Review*, vol. 53, no. 3, pp. 464–501, 2011.

[21] J. Löfberg, "Automatic robust convex programming," *Optimization Methods and Software*, vol. 27, no. 1, pp. 115–129, 2012.

[22] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, "Decentralized data-enabled predictive control for power system oscillation damping," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 3, pp. 1065–1077, 2021.

[23] X. Shang, J. Wang, and Y. Zheng, "Decentralized robust data-driven predictive control for smoothing mixed traffic flow," *Technical report, available at arXiv preprint arXiv:2401.15826*, 2024.

[24] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[25] E. D. Andersen and K. D. Andersen, "The mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm," in *High performance optimization*. Springer, 2000, pp. 197–232.

[26] DieselNet, "Emission test cycles ece 15 + eudc/nedc," 2013. [Online]. Available: https://dieselnet.com/standards/cycles/ece_eudc.php

[27] D. P. Bowyer, R. Akçelik, and D. Biggs, *Guide to fuel consumption analyses for urban traffic management*. Vermont South, Australia: ARRB Transport Research Ltd, 1985, no. 32.