Traffic Control via Connected and Automated Vehicles

AN OPEN-ROAD FIELD EXPERIMENT WITH 100 CAVS

JONATHAN W. LEE[*,1], HAN WANG[*,1], KATHY JANG[*,1], NATHAN LICHTLÉ[*,†,1], AMAURY HAYAT[†,1], MATTHEW BUNTING[‡,1],
ARWA ALANQARY[*,2], WILLIAM BARBOUR[‡,2], ZHE FU[*,2], XIAOQIAN GONG[§,2], GEORGE GUNTER[‡,2], SHARON HORNSTEIN[‖,2], ABDUL RAHMAN KREIDIEH[*,2], MATTHEW W. NICE[‡,2], WILLIAM A. RICHARDSON[‡,2], ADIT SHAH[*,2], EUGENE VINITSKY[*,2], FANGYU WU[*,2], SHENGQUAN XIANG[‖‖,2],
SULAIMAN ALMATRUDI[*,3], FAHD ALTHUKAIR[*,3], RAHUL BHADANI[‡,3], JOY CARPIO[*,3], RAPHAEL CHEKROUN[*,3], ERIC CHENG[*,3], MARIA TERESA CHIRI[¶,3], FANG-CHIEH CHOU[*,††,3], RYAN DELORENZO[‡‡,3], MARSALIS GIBSON[*,3], DEREK GLOUDEMANS[‡,3], ANISH GOLLAKOTA[*,3], JUNYI JI[‡,3], ALEXANDER KEIMER[*,3], NOUR KHOUDARI[**,3], MALAIKA MAHMOOD[‡‡,3], MIKAIL MAHMOOD[‡‡,3], HOSSEIN NICK ZINAT MATIN[*,3], SEAN MCQUADE[‡‡,3], RABIE RAMADAN[**,3], DANIEL URIELI[‖,3], XIA WANG[‡,3], YANBING WANG[‡,3], RITA XU[*,3], MENGSHA YAO[**,3], YILING YOU[*,3], GERGELY ZACHÁR[‡,3], YIBO ZHAO[*,3],
MOSTAFA AMELI[¶¶, *,4] MIRZA NAJAMUDDIN BAIG[††,4], SARAH BHASKARAN[*,4], KENNETH BUTTS[§§,4], MANASI GOWDA[*,4], CAROLINE JANSSEN[‡,4], JOHN LEE[*,4], LIAM PEDERSEN[††,4], RILEY WAGNER[‡,4], ZIMO ZHANG[*,4], CHANG ZHOU[*,4],
DANIEL B. WORK[‡], BENJAMIN SEIBOLD[**], JONATHAN SPRINKLE[‡], BENEDETTO PICCOLI[‡‡], MARIA LAURA DELLE MONACHE[*], AND ALEXANDRE M. BAYEN[*]

[*]–University of California, Berkeley
[†]–École des Ponts Paristech, Marne la Vallée
[‡]–Vanderbilt University
[§]–Arizona State University
[‖]–General Motors
[¶¶]–Université Gustave Eiffel, COSYS-GRETTIA
[‖‖]– Peking University
[¶]–Queen's University
[**]–Temple University
[††]–Nissan North America
[‡‡]–Rutgers University-Camden
[§§]–Toyota North America
[1]–These authors contributed equally
[2]–These authors contributed equally
[3]–These authors contributed equally
[4]–These authors contributed equally

The promise of generating societal benefit from autonomous vehicle technology has long been capturing the imagination of researchers, legislators, and popular culture. Its footprints can be seen in the earliest stages of modern AV research and development in the United States. AV technology advancements, spanning over three decades of work, are sometimes broken down into three generations:

Generation One. In the 1990's, the Federal Highway Administration (FHWA) established the National Automated Highway System Consortium (NAHSC) to demonstrate the potential of an automated vehicle and highway system for societal benefit [1]. The consortium partners included University of California, Berkeley, California Partners for Advanced Transit and Highways (PATH), and General Motors (GM), among several others. In the implementation of Demo '97 on Interstate 15 in San Diego [2], California, the consortium introduced and/or brought to realization concepts such as Adaptive Cruise Control (ACC), Vehicle-to-Vehicle (V2V) Communication, and Cooperative ACC (CACC) [3], [4], [5]. The demonstrations involved high-speed platooning, made possible by V2V and CACC, as well as infrastructure sensor integration. Moreover, it laid the foundation for modern automated highway system AHS architectures, emphasizing the importance of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications [6]. Some of the key challenges and barriers faced at the time included scalability due to infrastructure, reliable communication systems, and accurate vehicle positioning. Its eventual demise came in the late 1990's with tightening research budgets at USDOT.

Generation Two. In the 2000's, the Defense Advanced Research Projects Agency (DARPA) introduced the DARPA Grand Challenge. In the first Grand Challenge (2004), a $1 million prize was offered to any team whose autonomous vehicle could complete a 150-mile course in the Mojave Desert; no teams were successful. The second Grand Challenge (2005) doubled the prize money, and five teams were successful in completing the course [7]. The third Grand Challenge (2007) moved to an urban environment. The DARPA Grand Challenge competitions were notable for creating a renewed interest in autonomous vehicles and directly supported efforts to commercialize successful technologies [8] in the following decade.

Generation Three. In the 2010's, many major automotive manufacturers (such as GM, BMW, Audi, and Tesla) and technology companies (such as Google/Waymo, Uber, Lyft) began developing autonomous vehicle technology for commercialization purposes. The investments have generated an industrial ecosystem centered on hardware, software, and services designed to advance fully autonomous

driving. Today multiple companies are testing AVs in numerous public roads in the US, with Waymo most notably operating an autonomous ride-share service in Phoenix, San Francisco, and Los Angeles without a safety driver in the vehicle, and Cruise, which operates autonomous shuttles in San Francisco. At the same time, Level 1 and Level 2 automated vehicle technologies, including ACC, are now widely available in the commercial market. These commercial systems predominantly focus on safety and driver comfort [9], rather than improved transportation system efficiency [10].

Many experiments involved using CACC within platooning experiments for improving fuel efficiency on highways [11]. In 2009, the Safe Road Trains for the Environment (SARTRE) [12] three year project was funded by the European Commission to investigate the benefits of platooning (following previous European projects such as PROMOTE CHAUFFEUR I + II or Konvoi), where a leading bus or truck driven by a professional driver controls a following platoon of a few heavy or lights vehicles, and achieving up to 16% energy savings [13] on public highways. In other works, Energy ITS [14] deployed a platoon of three fully-automated trucks in Japan in 2008. In 2011, nine teams developed CACC controllers with V2V communication and compared them in urban and highway settings as part of the Grand Cooperative Driving Challenge (GCDC) [15], [16] held in the Netherlands, and later reiterated in a 2016 edition [17]. Between 2015 and 2017, the California PATH program has been involved in CACC truck platooning experiments on public highways [18]. More recently, the European ENSEMBLE project [19] investigated multi-brand truck platooning to integrate V2V technologies in all European brands of trucks.

Additional studies have experimentally demonstrated the benefits of vehicle platooning. In [20], [21], CACC platoons were developed and deployed on commercially available vehicles using linear control theory and string stability. These CACC platoons were shown to outperform ACC vehicles in terms of disturbance dissipation. [22] showed in simulation that a moderate penetration rate of CACC vehicles can significantly increase network capacity. The research in [23] demonstrated that CACC control allows for safe platooning at low following distances and high speeds, thus improving traffic throughput. Furthermore, [24] indicated that energy consumption in freight trucks could be reduced by driving at low following distances, due to decreased air resistance. Recent experiments in [25] built upon these prior CACC studies by employing model predictive control V2I communication, showing that vehicle platoons can enhance throughput on arterial roadways with traffic lights.

Our work brings full circle the nascent imaginings from the early 1990's. The Generation One promise of automating highways for societal benefit began to manifest

in the late 2010's within our team and in several groups around the world. In 2016, a group led by co-authors Delle Monache, Piccoli, Seibold, Sprinkle, and Work replicated the Sugiyama experiment [26] of string instability in a ring road; in what is known as the "Arizona ring experiment," they further demonstrated that phantom jams can be reduced using partially automated vehicle (AV) technologies and specially-designed algorithms [27]. This work was followed shortly by co-author Bayen's lab at UC Berkeley with the development of `Flow`, a software package to interface then-state-of-the-art microsimulation software with deep reinforcement learning (deep-RL) libraries [28], [29], [30], [31], [32], [33]. With `Flow`, the UC Berkeley team was able to independently train an AV controller to replicate the findings of the Arizona ring experiment [34] and generalize them to a variety of other settings, such as freeway and urban traffic [35]. (Note the distinction in terminology between "autonomous vehicle" and "automated vehicle," specifically because our past and present work focuses on automated longitudinal control as Lagrangian traffic controllers.)

In 2019, co-authors Bayen, Piccoli, Seibold, Sprinkle, and Work united to form the Congestion Impacts Reduction via CAV-in-the-loop Lagrangian Energy Smoothing (CIRCLES) Consortium (see https://circles-consortium. github.io/) [36], [37], [38], joined shortly thereafter by co-authors Lee and Delle Monache, each leading various aspects of the CIRCLES project. The CIRCLES project seeks to extend these prior research efforts to real traffic. To achieve our goals, our group designed a modular, hierarchical control framework, consisting of a centralized Speed Planner and decentralized Vehicle Controllers, and implemented it on 100 vehicles in a large-scale field operational test, dubbed the MegaVanderTest (MVT). Our aim is to deploy flow-smoothing ACC-enabled AVs that don't require explicit communication and are mixed within highway traffic in an unstructured way, contrary to many of the CACC truck platooning experiments previously mentioned. As many modern vehicles come equipped with ACC technology, the focus of this work is to modify the ACC algorithm to become significantly better at flow smoothing, without requiring additional hardware, road infrastructure, or V2V communication capabilities.

This article presents our control system design and subsequent analysis of field test results. Diverse candidates for each module of the framework are developed utilizing cross-disciplinary knowledge and tools, including ordinary differential equation/partial differential equation (ODE/PDE)-based flow control [39], [40], deep-RL [41], stabilization theory [42], functional analysis [43], optimal control on microscopic and macroscopic systems [44], [45], approximation theory [46], mean-field limits [47], non-entropic solutions to hyperbolic systems [48], model predictive control (MPC) via linearly constrained quadratic

programming (LCQP) [49], kernel smoothing [50], traffic flow theory [51], variable speed limit [52], and many other areas. The control system was then evaluated and tested in the open road as part of the MVT, the largest deployment of AVs designed to smooth traffic flow. In this test, we deployed 100 AVs on Interstate 24 (I-24) near Nashville, TN in November 2022. The experiment coincided with the debut of I-24 MOTION (https://i24motion.org) [53], [54], a four-mile section of I-24 near Nashville, TN to capture ultra-high resolution trajectory data of all vehicles.

The rest of the article is organized as follows: Section "Design of the Controller Architecture" introduces the MegaController, our controller architecture which consists of several components that operate together. Section "Design of Controller Components" then details the design and functioning of each of these individual components. Section "Candidate Controller Selection" explains how controllers are selected for the test, and finally Section "Open Road Field Operational Test" introduces the experimental design for the 100 AV deployment, as well as implementation details for the selected controllers, deployment on hardware and data collection procedures. Furthermore, several sidebars are spread throughout the paper, which can serve different roles: giving background about certain parts of the projects, diving into much more details about certain aspects of the project, or presenting interesting work that happened before or in parallel to the project, or that didn't make it into the final deployment. The sidebars are as follows, in order of appearance: "Adaptive Cruise Control Modeling", "Macroscopic ODE/PDE models", "Optimal control of measure PDEs", "MPC controller", "Optimized Vehicle Trajectory", "Vehicle Energy Models", "Trajectory Simulator", "Car-Following Models" and "Vehicle Interfacing".

Summary

The CIRCLES project aims to reduce instabilities in traffic flow, which are naturally occurring phenomena due to human driving behavior. Also called "phantom jams" or "stop-and-go waves," these instabilities are a significant source of wasted energy [55], [56]. Toward this goal, the CIRCLES project designed a control system referred to as the MegaController by the CIRCLES team, that could be deployed in real traffic. Our field experiment leveraged a heterogeneous fleet of 100 longitudinally-controlled vehicles as Lagrangian traffic actuators, each of which ran a controller with the architecture described in this paper. The MegaController is a hierarchical control architecture, which consists of two main layers. The upper layer is called Speed Planner, and is a centralized optimal control algorithm. It assigns speed targets to the vehicles, conveyed through the LTE cellular network. The lower layer is a control layer, running on each

This work presents a modular, hierarchical control framework, consisting of a centralized Speed Planner and decentralized Vehicle Controllers, as it was used during the MegaVanderTest.

vehicle. It performs local actuation by overriding the stock adaptive cruise controller, using the stock on-board sensors. The Speed Planner ingests live data feeds provided by third parties, as well as data from our own control vehicles, and uses both to perform the speed assignment. The architecture of the speed planner allows for modular use of standard control techniques, such as optimal control, model predictive control, kernel methods and others. The architecture of the local controller allows for flexible implementation of local controllers. Corresponding techniques include deep reinforcement learning, model predictive control and explicit controllers. Depending on the vehicle architecture, all on-board sensing data can be accessed by the local controllers, or only some. Likewise, control inputs vary across different automakers, with inputs ranging from torque or acceleration requests for some cars, and electronic selection of ACC set points in others. The proposed architecture technically allows for the combination of all possible settings proposed above, that is {Speed planner algorithms} × {local controller algorithms} × {full or partial sensing} × {torque or speed control}. Most configurations were tested throughout the ramp up to the MegaVandertest.

DESIGN OF THE CONTROLLER ARCHITECTURE

We introduce the **MegaController**, a control framework, depicted in Figure 1, for the mixed autonomy traffic flow problem. Mixed autonomy refers to the setting in which some vehicles are automated, while other vehicles are manually controlled (level 0 automation). The primary design paradigm of the proposed control framework is to achieve two goals: hierarchy for task allocation and modularity for control flexibility.

We designed a hierarchical structure to efficiently coordinate the control goals between macroscopic traffic flow optimization [57] and microscopic vehicle control [58], [59], [60], [61] and to efficiently solve the computational task allocation problem between the server and vehicle sides. There is an inherent interplay between the two components, as the Speed Planner informs the Vehicle Controller of downstream events, and the vehicles are instrumented to report observations to the server for data aggregation.

Use of modular design

A modular structure is used for two important purposes: (1) to facilitate a diversity of controller design approaches, and (2) to enable a heterogeneous vehicle fleet with different sensing and actuation capabilities.

As long as each controller utilizes the available interfaces, it is technically straightforward to exchange different controllers when using a modular design. This decision allows us to support a wide range of expertise from our team who may explore many different designs, each with a different technical focus or approach. While there may be subtleties for stability, convergence, and other properties if integration decisions are made strictly through type matching, our modular approach mitigates a key challenge: the dynamics of the roadway and our ability to control it may not be fully understood until testing begins. Since testing may only be possible at a scale when we can evaluate its impact, we will be very close to the full test deadline when we make final decisions. By embracing a modular approach, we can use data from tests that concluded less than 24 hours earlier to decide what controller to run during the following test.

The fleet of vehicles used in the MVT represented 3 different years, makes, and models. The fleet was composed of 2023 Nissan Rogue, 2022 Cadillac XT5, and 2020-2021 Toyota Rav4. Trade secrets and design differences between the systems on these vehicles mean that it is not possible to have a uniform system interface for sensing and control. For example, although each car we used has adaptive cruise control which uses forward facing radar for safety, the data from those sensors were not available to our system for all vehicle types. Designing a controller that can work on only one vehicle type may give greater controllability, but reduces the potential impact of the design when deployed at a societal scale.

Thus it is a design strength that our system can operate across vehicles with myriad different sensing and control modalities. While the interfaces to the Speed Planner remain the same, different vehicles will have different implementations that have completely unique characteristics compared to other vehicles, similar to textbook design patterns in software engineering for producer/consumer architectures. This allows a flexibility to adapt to a heterogeneous fleet, theoretically opening the door to many
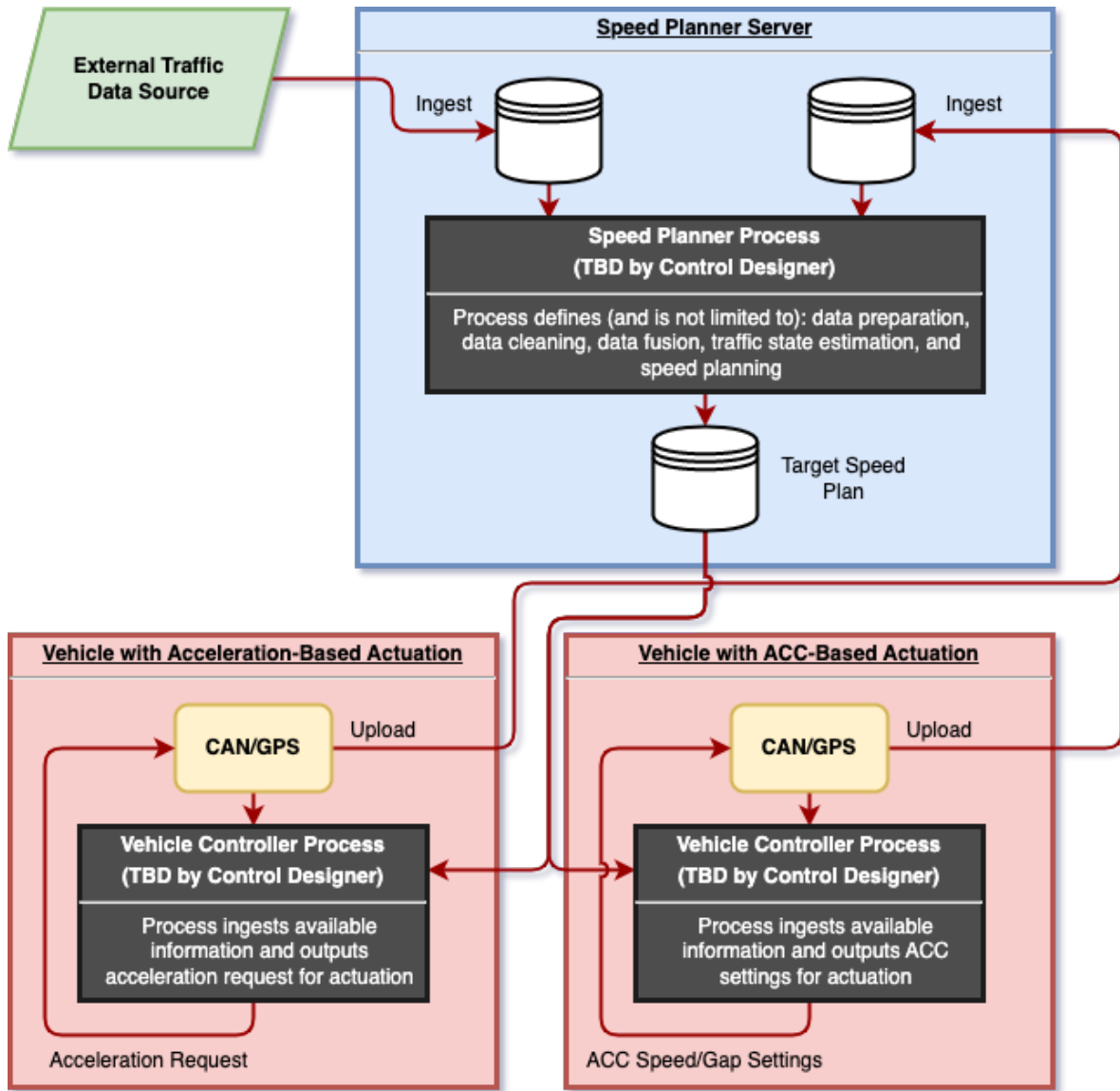
FIGURE 1 Architectural framework of the MegaController. The hierarchical and modular nature of the design allows for greater flexibility in design decisions and dealing with varied sensing and actuation capabilities of the heterogeneous fleet. The blue box represents the centralized Speed Planner unit, and the red boxes represent decentralized Vehicle Controllers, which are vehicle-dependent (that is, each vehicle has a different control architecture and thus requires a different control paradigm). The components work in concert to achieve higher level goals of flow smoothing.

other vehicle configurations and constraints.

In the MegaController design, there are two distinct components:

» A server-side **Speed Planner**, which is a centralized planner unit that provides high-level macroscopic speed suggestions based on periodic state updates from the distributed vehicles and external macroscopic data source, like INRIX employed in this study. The algorithms deployed on the server-side are designed to handle computationally-heavy data aggregation and macro-state tasks.

» A vehicle-side **Vehicle Controller**, which is a networked decentralized controller [62] that commands local actuation of the vehicle. The algorithms deployed here take into consideration the target speed suggested by the server-side planner unit, the latency of that information, and any observations from the vehicle's onboard sensors.

Borrowing conceptually from object-oriented programming, the interfaces are agreed upon *a priori*, allowing an abstraction of individual components for ease of design and testing.

> **The design paradigm of the control framework achieves two goals: hierarchy for task allocation, and modularity for control flexibility.**

Speed Planner

The objective of the Speed Planner is to optimize the overall traffic flow efficiency by providing macroscopic guiding speeds for AV fleets using realtime traffic data from multiple heterogeneous sources. While optimality for our deployment was defined by flow efficiency, our framework allows for arbitrary objective functions.

The data sources at our disposal are the set of recent vehicle ping messages from our AV fleet using our custom hardware and software stack (detailed later in Figure 11), as well as vehicular velocity data within distinct sections of highway through INRIX, a provider of live data for traffic, parking, and other transportation data. INRIX data has been studied in the literature for its applicability in traffic estimation, and validated by researchers for its accuracy using *post hoc* methods [63]. Utilizing these heterogeneous data sources presented some key challenges and considerations, namely unknown latency of external data sources and differences in spatial and temporal resolution of those data sources when fused with our vehicle message data. Although the specific aggregation algorithm is not publicly known, INRIX data latency is explored in [64], which describes delays between 3-5 minutes, and in some cases up to 10 minutes. A summary of data specifications can be found in Table 1.

Data Source:	INRIX	AV Ping
Period (seconds)	60	1
Latency (seconds)	~180	negligible
Segment Length (miles)	~0.5 (varied)	N/A
Lateral aggregation	Lane averaged	Lane specific

TABLE 1 A summary table of the data sources for traffic state estimation (and subsequent processes). INRIX aggregates data from a fleet of vehicles on the road. The AV Ping data come from our own fleet, where each car posts speeds to the server through an API that contains speed (from CAN) and positioning (from GPS), as well as timing and vehicle identity information.

Targeting the challenges, the two main functions of the Speed Planner are (1) Traffic State Estimation (TSE) enhancement and (2) the design of the Speed Plan, an ideal target speed profile. The TSE enhancement module is designed to eliminate the effects of inherent latency in the third-party data source and to improve the spatial and temporal resolution of the input traffic data. The target design module generates Speed Plans with the goal of reducing vehicle energy consumption and increasing the overall throughput of the traffic flow. Various approaches are utilized to develop the target design module, including kernel smoothing [50], ODE/PDE-based flow control [39] and vehicle trajectory optimization [58].

Vehicle Controller

The Vehicle Controller enables each car to use traffic information provided by the Speed Planner in context with local state, such as the distance to the vehicle in front, relative speed, etc. The Vehicle Controller aims to improve the flow of traffic while ensuring the safety of both the AV and other vehicles around it.

Input data for the Vehicle Controller are obtained from the in-vehicle network, and may include other inputs such as the target speeds from the Speed Planner. A subset of data from the CAN bus are made available to the vehicle controller, and vary depending on the vehicle make/model/year. Realizing control algorithms on vehicles required new computer hardware, electronics, and software development [65], [66], [67], [68], [69], [70] because of the heterogeneity in vehicle data and control interfaces.

Depending on the vehicle platform, we have explored options including speed-based control [71], [72], acceleration-based control [67], [73], and ACC-based control [41]. Speed-based control was initially explored, but then deprecated as acceleration based control best matched desired goals from algorithm designers. The acceleration-based and ACC-based approaches fit into the overall architectural design. Torque-based control (that is, applying the requisite torque directly to the gears) would be the most prescriptive control with tightest actuation, but it also requires a level of CAN bus access that prevents quick return of the vehicle to stock state. The majority of the vehicles used in the experiment were stock vehicles that were later returned to the manufacturer, so all changes to enable experimental control needed to be minimal and non-invasive changes to the vehicle. Therefore, for this deployment, our architecture included two types of control capabilities on our heterogeneous fleet, which are summarized in Table 2.

Acceleration-based Controller

While acceleration controllers don't have natural safety enforcement such as the one speed controllers have, we

Adaptive Cruise Control Modeling

by Fang-Chieh Chou, Jonathan W. Lee

*A*daptive Cruise Control (ACC) is an advanced driver assistance system that automatically operates vehicles at a set driving speed or keeps a set driving gap with respect to a leading vehicle. Operation of ACC requires a speed setting and a gap setting input by the driver. When there is no leading vehicle, ACC automatically adjusts the vehicle's driving speed to the speed setting preset by the driver. When there is a leading vehicle ahead, the system automatically regulates the gap with respect to the leading vehicle to a separation set by the driver, while not driving faster than the speed setting.

An ACC-controlled car therefore be modeled as a dynamical system composed of two control modes: speed-control mode and gap-control mode. The system switches between two modes depending on the proximity and speed of a leading vehicle. When there is no nearby leading vehicle, the system is in speed-control mode, in which the system tracks the speed setting. On the other hand, when the system is in gap-control mode, the system tracks both the leading vehicle speed and the gap setting.

To model an ACC-controlled car, we used parametric models for each control mode. Mathematically, the speed-control mode can be written as

$$a_e = f_p(v_e, v_{ref}), \tag{S1}$$

where a_e is the acceleration of the ego vehicle, v_e is the speed of the ego vehicle, and v_{ref} is the speed setting. $f_p : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a parametric model of parameters p.

Gap-control mode can be written as

$$a_e = f_q(v_e, v_l, d_{el}, g_{ref}), \tag{S2}$$

where v_l is speed of leading vehicle, d_{el} is the space gap between the ego vehicle and the leading vehicle, and g_{ref} is the gap setting. $f_q : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a parametric model of parameters q.

To fit parameters p and q for speed-control mode and gap-control mode, field experiments are carried out using a Nissan Leaf vehicle equipped for data collection. For the speed-control mode, data collection is done with a variety of initial speeds and speed settings so that dynamic responses with different speed changes can be collected. For the gap-control mode, highway driving data under different traffic conditions is collected so that dynamic response over a range of speeds can be covered in the data set. The collected data is firstly smoothed to reduce sensor noise. Some outliers of abnormal driving conditions (for example, temporally near cut-in and cut-out events) are removed before fitting the models. The model is validated by comparing the dynamic response of the model in simulation to the response of the real system in field experiments. While the models are fitted for the Nissan Leaf, they are assumed to be acceptable approximations for the ACC systems of other vehicles in our AV fleet.

		Toyota RAV4	Nissan Rogue, Cadillac XT5
Vehicle(s):			
Actuation Type:		Acceleration	ACC
Sensing	Ego Position	✓	✓
	Ego Speed	✓	✓
	Ego ACC Speed Setting	✓	✓
	Ego ACC Gap Setting	✓	✓
	Leader Space Gap	✓	✗
	Leader Minicar	✓	✓
	Leader Relative Speed	✓	✗

TABLE 2 A summary table of the sensing and actuation available to our system, for the heterogeneous vehicle fleet. Space gap refers to the distance between the ego vehicle's front bumper and the leader vehicle's rear bumper. The minicar refers to a boolean indication whether the ego vehicle's sensors detect the presence of a leader vehicle (an approximate 80-100 meter maximum distance). Relative speed is the leader vehicle's speed minus the ego vehicle's speed.

have found a way to add safety enforcements. Using new approaches in [74], we demonstrated that it was possible to constrain unsafe accelerations when they were passed to the vehicle. This opened the door to use of acceleration-based control on cars that support it. Acceleration-based control is a natural analog to how most of us drive: we press down on the accelerator pedal when we want to go faster, and we either let off the accelerator, or press the brake pedal, when we want to go slower.

When making control requests to the car, we provide a desired input over the vehicle's CAN, which is interpreted by the powertrain and ACC subsystems of the car. This desired input is not strictly a direct acceleration command: it is processed by the vehicle's ACC system which interprets the commanded acceleration and decides whether it will meet it (a) by actuating the brake, (b) by decreasing acceleration through reducing the throttle, (c) by keeping the throttle constant, or (d) by increasing the throttle. Examples of how these dynamics from step inputs have been observed can be found in [71]. The specific input selected by the car is a function of many different vehicle dynamics and perhaps even trade secrets for reducing wear-and-tear of parts, so there is no single mapping or predictive function that easily codifies the transfer function of the system once an acceleration is requested.

Using acceleration-based control *implicitly* adds safety

challenges, as the controller must ignore all existing acceleration commands from the car—including any acceleration commands that come from the safety subsystems on board. Safety wrappers such as [74] are needed to prevent rear-end collisions, and additional care must be taken to evaluate external sensors. Further, environmental challenges such as lane-changes by other vehicles require our system to ensure that our dynamics do not inadvertently cause their own stop-and-go waves. We decompose the behavior of this controller into two modes:

» A **base controller** prescribes a performance-based desired acceleration under "ideal" conditions. The base controller is designed to output an acceleration that satisfies a specific performance-based goal, for example, reducing energy footprint or minimizing the magnitude of accelerations. In general, care is taken during its design-phase to ensure that other aspects of the traffic flow are not negatively impacted. Note that the base controller should still, by design, be safe and collision-free, under the assumption that accelerations are actuated perfectly.

» A **lane-change recovery controller** modulates the base controller's desired acceleration in the event of discontinuities in the lead-vehicle space gap. The acceleration output of the base controller can change dramatically if there is a time-discontinuity in the input signals. Most commonly, this appears in the form of a lane-change by another vehicle in front of the ego vehicle. Since a lane-change is out of the ego vehicle's control, it can theoretically put the ego vehicle into an unsafe state if the objective were to control to a time gap with discontinuous input. This mode recognizes that discrete changes in the space gap are not controller failures, so the objective of this controller is to comfortably return the vehicle to a safe state where the base controller can resume actuation.

ACC-based Controller

Another option for controlling vehicles is to update vehicle's *Adaptive Cruise Control* (ACC) set points electronically. This mimics what the driver can do through buttons on the steering wheel, and has the benefit of keeping the safety features of the car's ACC systems in the loop. This approach is contrasted with the acceleration-based approach in a few ways, notably:

» The controllability of the system is affected and the bandwidth of the controller is likely reduced, since the stock ACC has its own gains and modes.

» The multi-model design features of the acceleration-based approach (namely the lane-change recovery controller and safety wrapper) are not needed here, as those features are built into the stock ACC algorithm.

To understand the changes in controllability, we explored the vehicle's ACC dynamics with a goal to make design decisions on whether rate limits or other constraints on the input signals would be required for stability or performance reasons. Every vehicle make and model equipped with ACC has its own unique mapping of state space (for example, ego speed, space gap, leader speed, etc.) and ACC inputs (speed setting and gap setting) to actuated acceleration. Architecturally, we take as given that this model is available for the design of the base ACC controller. For a more detailed discussion of ACC and our specific approach to obtaining a model for our algorithm design, see "Adaptive Cruise Control Modeling".

As a result, the only component of the ACC-based controller that requires specific designing is the base controller. Just as with the acceleration-based controller, the base controller is performance-based, striving for some performance goal. Similar design philosophies are applied as before. Uniquely different, as noted above, is that this base controller does not need to explicitly consider safety and vehicle-specific dynamics—the controller design can implicitly account for the vehicle-specific dynamics by utilizing the ACC model in a feedback loop.

DESIGN OF CONTROLLER COMPONENTS

Here, we describe in detail how each of the modular components was designed. The discussion is in a parallel structure to the previous section, with particular focus on controllers that were actually implemented for deployment. In the lifespan of the project, we explored many approaches for designing the controller. See "MPC controller" and "Optimized Vehicle Trajectory" for some notable research and designs that were not deployed but contributed key concepts that helped the ultimate implementation.

Speed Planner

Based on the hierarchical framework, Figure 5 indicates the implementation of the speed planner we tested in the MVT. As introduced in Table 1, INRIX and probe vehicle data are posted to the central database in different frequencies: each vehicle makes its post approximately every 1 second, and a server-side process inserts new data from INRIX approximately every 60 seconds. It is important to note that while the INRIX data provides a single speed across all lanes, the AV pings provide lane-level speed information.

The sequence of events of a Speed Plan publication can be summarized as follows:

1) Each new INRIX update is combined with historical INRIX data, and fed into the prediction module.

2) All vehicle observations from the previous 60 seconds are fetched, and fused with the INRIX prediction, to obtain a lane-level traffic state estimate.

3) The lane-level TSE is smoothed with the forward-kernel average.

4) Bottleneck identification is performed with the smoothed lane-level TSE.

5) If a standing bottleneck is identified in the lane, a deceleration region is prescribed as a buffer segment. For all other regions, the smoothed lane-level TSE is used as the lane-level Speed Plan.

6) Publish the Speed Plan for all lanes.

The subsequent sections delve deeper into the intricacies of the TSE Enhancement and Target Design, providing a comprehensive understanding of the steps outlined above. Wang et al. [39] provides in-depth methodology description of the Speed Planner.

TSE Enhancement

The TSE enhancement module comprises two main components: the INRIX prediction module and the data fusion module. The prediction module's primary role is to minimize the latency issues associated with INRIX realtime data, especially when applied to vehicular control, as opposed to its standard use for general traffic insights. The fusion module complements this by integrating realtime data from our system's probe vehicles, enabling a more detailed, lane-level TSE with enhanced time-space precision. A detailed mathematical explanation of the TSE enhancement module for MVT implementation is available in [39].

To achieve a lane-level TSE with superior spatial detail, we further segment the INRIX data into smaller units. After generating the INRIX prediction, it is merged with realtime data from the system's controlled vehicles. Given that each AV communicates with the server at a rate of 1Hz, we receive 60 ping records from each vehicle for every Speed Plan generation. These records help determine the average speed of each vehicle over the previous update period, which then updates the TSE for the respective sub-segment. Assuming that our drivers consistently stick to their designated lanes (a behavior largely confirmed through systematic data review), we can generate lane-specific TSE estimates by combining vehicle data with the broader INRIX speed data. When it comes to data fusion, we give precedence to data from our vehicles over INRIX data, considering the inherent characteristics of both sources. Our vehicles gather and relay perception data through a system that is both observable and controllable, with quantifiable error and latency as detailed in [79]. In contrast, the INRIX realtime API employs averaging

techniques to forecast over a given period. This method might introduce inaccuracies at specific points within that timeframe, which could impact our control execution. The INRIX system, in essence, operates as a somewhat opaque system, with its error and latency aspects largely inferred from provider descriptions. In this paper, we adopt the following notation to represent the discrete TSE:

$$\{(x_j, \bar{v}_j), j \in \mathcal{J}\}, \tag{15}$$

where j represents the index of road segments, x_j is the postmile coordination of the central of road segment j and $\bar{v}_j$ is the average speed of the corresponding road segment j. Wang et al. [39] details the procedure of data fusion applied in MVT.

Target Design

This section introduces the Target Design's main modules, namely the kernel-based smoothing and the learning-based buffer design. The kernel smoothing processes the enhanced TSE at each time step using a chosen kernel to improve the fuel consumption caused by the shockwave in a high-density traffic flow. The buffer design utilizes Reinforcement Learning (RL) to form a buffer area upstream of the standing bottleneck with the goal of improving throughput at the bottleneck. The target speed suggested by the RL is employed in a mathematical model of traffic, represented by a strongly coupled Partial and Ordinary Differential Equation (PDE-ODE). The outcome of this mathematical model is an identification of traffic density $(t, x) \mapsto \rho(t, x)$. The kernel smoothing then receives this information for learning the velocity of the next time step.

In kernels smooth module, we rely on enhanced TSE data to synchronize the driving speeds of automated vehicles. In particular, vehicles are assigned target speed profiles contingent on traffic state information, which is shared and common among all AVs.

At any fixed time step t, the desired speed profile $\mathbf{v} : \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}_+$ is extracted from enhanced TSE utilizing kernel methods. First, we preprocess the sparse TSE data by interpolating the discrete data pairs $(x_i, \bar{v}_i)$ to a continuous speed profile $(t, x) \in \mathbb{R}_+ \times \mathbb{R} \mapsto v(t, x)$, as an approximation of the average speed of a higher granularity traffic at a position x and at time t. Then, for any fixed time $t = t_\circ$ we obtain the desired speed by applying a kernel

Macroscopic ODE/PDE models

by Xiaoqian Gong

Here we model the traffic dynamics in the presence of M AVs and N human-driven vehicles on a single lane from the microscopic perspective using systems of ordinary differential equations. We assume that only AVs can be controlled and have a greater impact on the vehicle population than human-driven vehicles. Let $T > 0$ be a fixed time horizon, $I_M = \{1, \cdots, M\}$ and $I_N = \{1, \cdots, N\}$ the index sets of AVs and human-driving vehicles, respectively. Denote by $(x, v) \in \mathbb{R}^N \times \mathbb{R}^N_{\geq 0}$ and $(y, w) \in \mathbb{R}^M \times \mathbb{R}^M_{\geq 0}$ the position-velocity vectors of human-driven vehicles and AVs, respectively. To represent the positions and velocities of the M AVs during the time interval $[0, T]$, we define the time-dependent atomic probability measure on $\mathbb{R} \times \mathbb{R}_{\geq 0}$, also referred to as the *empirical measure*, as

$$\mu_M(t) = \frac{1}{M} \sum_{i=1}^{M} \delta_{(y_i(t), w_i(t))}, \quad t \in [0, T]. \quad \text{(S3)}$$

Alternatively, we can represent solutions as a measure supported on absolutely continuous trajectories $t \in [0, T] \mapsto (y_i(t), w_i(t)) \in \mathbb{R} \times \mathbb{R}_{\geq 0}, i \in I_M$. Similarly, we use the atomic measure

$$\mu_N(t) = \frac{1}{N} \sum_{j=1}^{N} \delta_{(x_j(t), v_j(t))}, \quad t \in [0, T] \quad \text{(S4)}$$

to track the positions and velocities of the N human-driven vehicles during the time interval $[0, T]$. The dynamics of the $M + N$ vehicles is given as follows:

$$\dot{y}_i = w_i,$$
$$\dot{w}_i = (H_1 *_1 (\mu_N + \mu_M) + H_2 * (\mu_N + \mu_M)) (y_i, w_i) + u_i, \ i \in I_M,$$
$$\dot{x}_j = v_j,$$
$$\dot{v}_j = (H_1 *_1 (\mu_N + \mu_M) + H_2 * (\mu_N + \mu_M)) (x_j, v_j), \ j \in I_N, \quad \text{(S5)}$$

where the convolution kernels $H_1 \colon \mathbb{R} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ and $H_2 \colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ represents a microscopic model, such as Optimal Velocity [75], Follow-the-Leader [76] or a combination of them [77]. Here $*_1$ is the convolution concerning the first variable, and $u_i \colon [0, T] \to \mathbb{R}$ are measurable controls for $i \in I_M$ influencing the time-evolution of AVs. Given initial data $(x(0), v(0), y(0), w(0)) = (x_0, v_0, y_0, w_0) \in \mathbb{R}^N \times \mathbb{R}^N_{\geq 0} \times \mathbb{R}^M \times \mathbb{R}^M_{\geq 0}$, the existence and uniqueness of solutions to system (S5) can be proved using Carathéodory theorem. This is a consequence of the fact that the two convolution kernels H_1 are locally Lipschitz with sub-linear growth. For more detailed discussions, we refer the readers to [S1].

Now we consider modeling mixed traffic dynamics when the number of human-driven vehicles is much greater than the number of AVs. This allows us to pass to the mean-field limit of the system (S5) with the number of human-driven vehicles formally going to infinity, that is $N \to \infty$. The mean-field limit of the system (S5 is given by a Vlasov-type PDE coupled with a system of controlled ODEs. The Vlasov-type PDE describes the evolution of the density of human-driven vehicles as a measure and the ODEs describe the controlled behavior of the M AVs. Specifically, the interaction between M AVs and the human-driven vehicles can be modeled using the following system:

$$\dot{y}_i = w_i,$$
$$\dot{w}_i = (H_1 *_1 (\mu + \mu_M) + H_2 * (\mu + \mu_M)) (y_i, w_i) + u_i, \ i \in I_M,$$
$$\partial_t \mu + v \partial_x \mu + \partial_v ((H_1 *_1 (\mu + \mu_M) + H_2 * (\mu + \mu_M)) \mu) = 0, \quad \text{(S6)}$$

where $(y, w) \colon t \in [0, T] \mapsto (y(t), w(t)) \in \mathbb{R}^M \times \mathbb{R}^M_{\geq 0}$ is the position-velocity vector of the M AVs, μ_M is defined as in (S3) tracking the position and velocity of the M AVs, $H_1 \colon \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R}$, $H_2 \colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ are locally Lipschitz convolution kernels with sub-linear growth, $*_1$ is the convolution with respect to the first variable and $\mu \in \mathcal{P}(\mathbb{R} \times \mathbb{R}_{\geq 0})$ is a measure on $\mathbb{R} \times \mathbb{R}_{\geq 0}$ representing the density distribution of the human-driven vehicles in position and velocity.

The rigorous limit process connecting the finite-dimensional system of ODEs (S5) to an infinite-dimensional system with coupled Vlasov-type partial differential equation (PDE) and ODEs (S6) was proved in [S2] using Wasserstein distance.

In addition, one can use finite-dimensional hybrid systems to model multi-lane, multi-class traffic dynamics with M AVs and N human-driven vehicles on an open stretch of the road with m lanes. The hybrid nature of the model is based on the vehicles' continuous dynamics and the discrete events due to the vehicle's lane-changing maneuvers. The mean-field limit of the finite-dimensional hybrid system is an infinite-dimensional hybrid system containing a Vlasov-type PDE with a source term, ODEs, and discrete events caused by the lane-changing behavior of the AVs. For the rigorous derivation of the mean-field limit of the finite-dimensional hybrid system, please see [S3].

REFERENCES

[S1] B. Piccoli and F. Rossi, "Transport Equation with Nonlocal Velocity in Wasserstein Spaces: Convergence of Numerical Schemes" Acta Applicandae Mathematicae volume 124, pages 73–105 (2013).

[S2] M. Fornasier, B. Piccoli, and F. Rossi, "Mean-field sparse optimal control," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 372, no. 2028, p. 20130400, 2014.

[S3] X. Gong, B. Piccoli and G. Visconti, "Mean-Field of Optimal Control Problems for Hybrid Model of Multilane Traffic," in IEEE Control Systems Letters, vol. 5, no. 6, pp. 1964-1969, Dec. 2021, doi: 10.1109/LC-SYS.2020.3046540.

Optimal control of measure PDEs

by Xiaoqian Gong

In this section, our goal is to investigate the optimal control problem (OCP) of multi-class traffic consisting of AVs and human-driven vehicles on a single lane. In applications, it aims to minimize congestion, energy consumption, or travel delays, by adding controls on AVs, rather than controlling all vehicles in the population.

Let us again assume that we have a small number of M AVs that have a large impact on the vehicle population and a fixed large number of N human-driven vehicles that have a small impact on the vehicle population. Let $T > 0$ be a fixed time horizon. We choose controls $u \in L^1((0, T), \mathcal{U})$, where $\mathcal{U}$ is a fixed nonempty compact subset of $\mathbb{R}^M$. We model the situation by solving the following finite-dimensional optimization problem:

$$\min_{u \in L^1((0,T),\mathcal{U})} F_N(u) = \int_0^T \left\{ L(y_N(t), w_N(t), \mu_N(t)) + \sum_{i=1}^{M} \frac{|u_i(t)|}{M} \right\} dt \,, \tag{S7}$$

where $L(\cdot)$ is a suitable continuous map in its arguments, μ_N is the atomic probability measure tracking the positions x and velocities v of the N human-driven vehicles as defined in (S4), and the position-velocity vectors (y_N, w_N) and (x, v) satisfies the dynamics (S5) with given initial datum $(x(0), v(0), y_N(0), w_N(0)) = (x_0, v_0, y_{N,0}, w_{N_0}) \in \mathbb{R}^N \times \mathbb{R}_{\geq 0}^N \times \mathbb{R}^M \times \mathbb{R}_{\geq 0}^M$ and control $u \in L^1((0, T), \mathcal{U})$. Note that we added the subscript N to the AVs' position-velocity vector (y, w) indicating the dependence of the AVs' positions and velocities on the number of human-driven vehicles N. The existence of optimal control for the finite-dimensional optimization problem (S7) was proved in [S1].

The mean-field limit of the finite-dimensional system (S5) was given by system (S6) coupled with Vlasov-type PDE and ODEs when the number of human-driven vehicle goes to infinity, that is, $N \rightarrow \infty$. Correspondingly, we introduce the following infinite-dimensional optimization problem:

$$\min_{u \in L^1((0,T),\mathcal{U})} F(u) = \int_0^T \left\{ L(y(t), w(t), \mu(t)) + \sum_{i=1}^{M} \frac{|u_i(t)|}{M} \right\} dt \,, \tag{S8}$$

where $L(\cdot)$ is a suitable continuous map in its arguments, (y, w, μ) is the unique solution to system (S6) with given initial condition $(y^0, w^0, \mu^0) \in \mathbb{R}^M \times \mathbb{R}_{\geq 0}^M \times \mathcal{P}(\mathbb{R} \times \mathbb{R}_{\geq 0})$ (μ^0 is compactly supported) and control $u \in L^1((0, T), \mathcal{U})$.

It turns out that the cost functional F_N in (S7) $\Gamma-$ converges to the functional F in (S8) as $N \rightarrow \infty$. This leads to the existence of optimal controls for the infinite-dimensional OCP (S8). In fact, the solutions of the finite-dimensional OCP (S7) converge to the optimal controls for the infinite-dimensional OCP (S8). For more details, please see [S1].

Furthermore, for multi-lane and multi-class traffic dynamics, we can study the mean-field limit of an OCP of a finite dimensional hybrid systems, which is given by an OCP of an infinite-dimensional hybrid system. The existence of optimal control for the OCP associated with the infinite-dimensional hybrid system is again due to a $\Gamma-$ convergence result. For more detailed discussion, please see [S2].

REFERENCES

[S1] M. Fornasier, B. Piccoli, and F. Rossi, "Mean-field sparse optimal control," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 372, no. 2028, p. 20130400, 2014.

[S2] X. Gong, B. Piccoli, and G. Visconti, "Mean-field limit of a hybrid system for multi-lane multi-class traffic," 2021. [Online] Available: https://arxiv.org/abs/2007.14655.

function $K(\cdot)$ at a position $x = x_\alpha$:

$$\mathbf{v}(t_\circ, x_\alpha) = \frac{\int_{x=x_\alpha}^{x_\alpha+w} K(x_\alpha, x) v(t_\circ, x) dx}{\int_{x=x_\alpha}^{x_\alpha+w} K(x_\alpha, x) dx}, \tag{16}$$

where w is the width of the estimation window. Many different kernel functions, such as Gaussian kernel, Triangular kernel, Quartic kernel, Uniform kernel etc., can be chosen. For the purposes of this paper, we consider a uniform kernel, the simplest of such mapping. The desired speed profile at a position x_α is accordingly defined as:

$$\mathbf{v}(t_\circ, x_\alpha) = \frac{\int_{x=x_\alpha}^{x_\alpha+w} v(t_\circ, x) dx}{w}. \tag{17}$$

When human drivers observe a gap between their vehicle and the one preceding, they tend to accelerate to close the distance. Our proposed desired speed profile aims to slow down *in advance*, although not excessively, to create a gap from the preceding vehicle. This approach takes into account the information provided by the TSE, which indicates the presence of congestion in the nearby downstream area. The proposed desired speed profile is adaptive to traffic states and offers relative robustness, as it only requires one parameter, w, to tune.

For buffer design, we consider the interval $\mathcal{I} \subset \mathbb{R}$ as the region of interest. In addition, we consider a subregion $\mathcal{I}_c \subset \mathcal{I}$ as a congested area. The idea is to determine the controlled vehicle target speed at a time step $t_\circ$, denoted by $\mathbf{u}(t_\circ, x)$, such that the density $\rho(t, x)$ for $x \in \mathcal{I}_c$ and $t \geq t_\circ$ is distributed uniformly through the region $\mathcal{I}$. Determining the controlled vehicle's target speed will be done in the following steps: (i) Designing a target speed $\mathbf{u}(t_\circ, x)$, given the input $\mathbf{v}(t_\circ, x)$ from kernel smoothing step, (ii) identifying the density $(t, x) \in \mathbb{R}_+ \times \mathcal{I} \mapsto \rho(t, x)$, given the target speed of the controlled vehicle, employing a strongly coupled PDE-ODE model of traffic flow, (iii)

MPC controller
by Fangyu Wu

Stop-and-go waves, characterized by periods of motion followed by abrupt halts, present significant challenges in traffic control and vehicular automation. To address this issue, we consider Model Predictive Control (MPC), a popular optimal control method [78]. The appeal of MPC lies in its ability to translate a typically discrete-time control task into a finite-dimensional optimization problem.

At each discrete time step i, the optimization problem obtains an estimate of the initial state of the plant. It then computes an optimal sequence of control over a specified planning horizon. Only the first control sequence is dispatched for actuation, with this cycle repeating until task completion.

We denote the state and input of the plant as $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ respectively. The discrete-time dynamics is expressed as $x_{i+1} = f(x_i, u_i)$, where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. The initial state, x_0, equals x_{init}. The control task imposes the following state and actuation constraints: $x \in \mathcal{X}, u \in \mathcal{U}$.

The objective of MPC is to minimize the cost function $l = \sum_{i=0}^{N-1} \ell(x_i, u_i) + \ell_f(x_N)$. This function measures the cumulative effect of state and control inputs, with $\ell, \ell_f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$. Analytically, the MPC entails to solving the following optimization at each time step:

$$\min_{x_i, u_i} \sum_{i=0}^{N-1} \ell(x_i, u_i) + \ell_f(x_N)$$
$$\text{s.t.} \quad x_0 = x_{\text{init}},$$
$$x_{i+1} = f(x_i, u_i), \quad i = 0, \dots, N-1, \quad \text{(S9)}$$
$$x_i \in \mathcal{X}, \quad i = 1, \dots, N,$$
$$u_i \in \mathcal{U}, \quad i = 0, \dots, N-1.$$

Upon each iteration, the MPC accepts x_{init} as input and generates u_0 as output. This feedback control process continues in a receding horizon manner until the task is terminated.

For the CIRCLES project, our team has developed an MPC that adopts a linearly constrained quadratic programming formulation, as demonstrated in [S1]. This control method primarily focuses on the longitudinal dynamics of the ego vehicle, defined as the vehicle controlled by the algorithm. The state comprises the vehicle's position and velocity, and the control input is its acceleration.

The imposed state constraints ensure that the ego vehicle neither collides with nor overtakes the predicted position of the lead vehicle. They also enforce a maximum road speed limit. Concurrently, input constraints set upper and lower bounds on acceleration. The MPC's objective is to minimize the sum of the L^2 norm on acceleration. This objective leads to a standard convex quadratic programming problem, which is solvable by widely available solvers.

The primary challenge associated with employing MPC for wave attenuation lies in accurately predicting the lead vehicle's longitudinal position across a substantial planning horizon. To optimally smooth traffic waves, it is imperative that this planning horizon aligns with both the spatial and temporal scales of the wave.

To overcome this challenge, our approach incorporates a realtime map service for long-term prediction and leverages instantaneous acceleration extrapolation for short-term forecasting. This dual predictive strategy serves two critical functions: the long-term predictions guide the ego vehicle to effectively dampen the traffic waves, while the short-term forecasts act as safeguards, preventing potential collisions with the lead vehicle.

REFERENCES
[S1] F. Wu & A. Bayen, "A Hierarchical MPC Approach to Car-Following via Linearly Constrained Quadratic Programming," *IEEE Control Systems Letters*, vol. **7**, pp. 532-537, 2022.

Evaluation step in which using the density, the speed profile will be updated by smoothing kernel.

Paper [39] provides details of the procedure introduced above.

Vehicle Controller
Recall in the architectural design of the controller, the Vehicle Controller layer features two distinct methods for actuating the vehicle: acceleration-based control and ACC-based control. Here, we will discuss component implementations prepared for both options.

Acceleration-Based Controller: Base Control
The acceleration-based controller requires a baseline controller to use under normal and emergency operating conditions, and a lane-change recovery controller to behave smoothly when cut-ins and cut-outs take place.

The base controller is an explicit, mathematically-defined controller designed to reach and keep an ideal target speed without being trapped in the stop-and-go wave. These two goals are somewhat antagonistic: a naive approach consisting in following the ideal target speed (unless it is unsafe to do so) may result in braking at the same time and with a comparable amplitude as the leading car when caught in the wave.

To address this, the base controller has an anticipation mechanism based on a paradigm that can be summarized as "act swiftly but slightly," which reduces variability in acceleration. It is comprised of three key components: *target*, *anticipation*, and *safety*. The target component infers an ideal target speed for each specific AV based on inputs from the Speed Planner and from the local data

Optimized Vehicle Trajectory

by Arwa AlAnqary

To gain more insights into the car-following task and to create a baseline for performance benchmark of control algorithms, we propose an optimal control formulation of the problem. We consider a mixed-autonomy platoon of vehicles driving on a single lane following a leader with pre-specified trajectory over a fixed interval $[0, T]$. The aim is to find the optimal control signal for all automated vehicles in the platoon to minimize an the platoon's energy consumption.

Optimal Control Problem

We consider a mixed autonomy platoon of M AVs, N human vehicles (HVs), and a leader vehicle. The AVs are controlled in their acceleration, the HVs' acceleration is governed by a *car-following model* (CFM) $A: \mathbb{R}^2 \times \mathbb{R}^2_{\geq 0} \to \mathbb{R}$, and the leader trajectory of the leader is specified by its position $x_\ell(t)$ and velocity $v_\ell(t)$. We index the vehicles in the platoon from front to back with $i = 0$ being the leader vehicle. Let $\mathbf{u}(t), \mathbf{x}(t), \mathbf{v}(t), \in \mathbb{R}^{N+M}$ be the control, position, and velocity vectors, respectively. Given initial value vectors $\mathbf{x}_0$ and $\mathbf{v}_0$, the platoon dynamics are governed by the following system of ODEs

$$
\begin{aligned}
\dot{\mathbf{x}}_i(t) &= \mathbf{v}_i(t), & i &\in I \setminus \{0\}, \\
\dot{\mathbf{v}}_i(t) &= A\big(\mathbf{x}_i(t), \mathbf{x}_{i-1}(t), \mathbf{v}_i(t), \mathbf{v}_{i-1}(t)\big) & i &\in I_h, \quad \text{(S10)} \\
\dot{\mathbf{v}}_i(t) &= \mathbf{u}_i(t) & i &\in I_a,
\end{aligned}
$$

where $I = \{0, 1, \ldots, M + N\}$, I_a and I_h are the sets of indices of the AVs and HVs, respectively.

We use Bando-follow-the-leader CFM [75] for the HVs in the platoon. The model describes the acceleration of the vehicle as a function of its space gap, velocity, and relative velocity. For parameters α, β, k, d, and car length l, we have

$$
A(x_\ell, x, v_\ell, v) = \alpha\big(V(x_\ell - x - l) - v\big) + \beta \frac{v_\ell - v}{(x\ell - x - l)^2}, \quad \text{(S11)}
$$

where

$$
V(h) = v_{\max} \frac{\tanh(kh - d) + \tanh(l + d)}{1 + \tanh(l + d)}. \quad \text{(S12)}
$$

Next, we define the set of admissible controllers as the set of functions $\mathbf{u} : [0, T] \mapsto \mathbf{R}$ that satisfy certain conditions: (1) the controllers can only be applied to the AVs, and they are bounded; (2) the AVs can not drive backwards; (3) the AVs remain within an allowable space gap envelope.

Lastly, we define the the objective functional of the *optimal control problem* (OCP) as the L^2 norm of the acceleration of all the vehicles in the platoon. We use this as a simple proxy of the fuel consumption of the vehicles.

Based on the above, we formulate the following OCP

$$
\inf_{u} \int_0^T \sum_{i \in I_a} \mathbf{u}_i^2(t) + \sum_{i \in I_h} \big(A(x_{i-1}(t), x_i(t), v_{i-1}(t), v_i(t))\big)^2 \mathrm{d}t, \quad \text{(S13)}
$$

where (x, v) satisfies Equation (S10) and $\forall i \in I_a$, $\forall t \in [0, T]$

$$
\begin{aligned}
h_{\min} v_i(t) + d_{\min} &\leq x_{i-1}(t) - x_i(t) - l, \\
h_{\max} v_i(t) + d_{\max} &\geq x_{i-1}(t) - x_i(t) - l, \quad \text{(S14)} \\
v_i(t) &\geq 0,
\end{aligned}
$$

where $h_{\min}$ and $h_{\max}$ are bounds on the allowable time gap, and $d_{\min}$ and $d_{\max}$ are the allowable minimum and maximum space gap at zero velocity.

To solve this optimization problem we parameterize the controls using piece-wise constant functions. This renders a finite dimensional optimization problem that we solve by means of gradient descent. We compute the analytical gradients of the problem using the adjoint formulation. For further details see [S1].

Numerical Simulation

We apply the proposed approach to a platoon with one AV following an empirical leader trajectory. This trajectory exhibits stop-and-go waves. We show the simulation results in Figure S1. By introducing one AV, we achieve up to 10% reduction in the fuel consumption compared to the baseline of full human-driven platoon. We apply the same approach sequentially to multiple AVs in a larger platoon. In Figure S2 we show the time-space diagrams for the baseline platoon compared to one with 8 AVs which achieves 24% reduction in the fuel consumption.
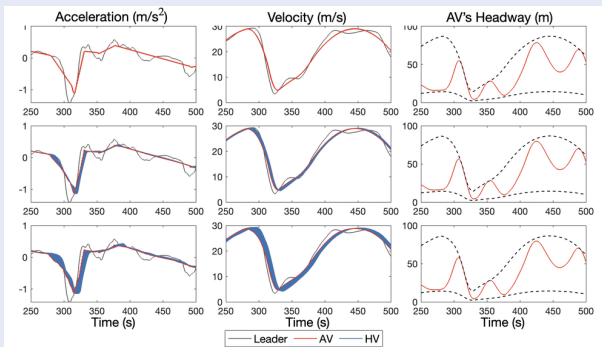

FIGURE S1 Trajectories of different platoons following the considered leader. Platoon size: 0 HVs (top row), 10 HVs (middle row), and 20 HVs (bottom row). The dashed lines in the space gap plots represent the feasible space gap profile. By introducing one AV behind the leader, we can achieve up to 10% reduction in the energy consumption compared to the fully human-driven platoon.
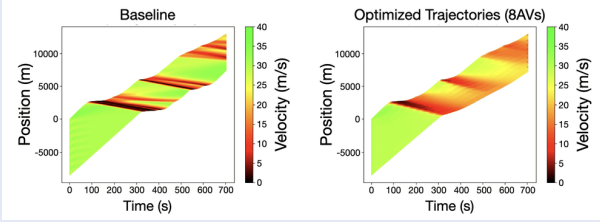
FIGURE S2 Time-space diagrams showing the effect of introducing 8 AVs with trajectories optimized using the proposed

approach (right) and comparing it with a baseline of a full human-driven platoon (left). The introduction of the AVs dampens the propagation of stop-and-go waves that appear in the leader's trajectory. It also achieves 24% reduction in the fuel consumption compared to the baseline.

REFERENCES
[S1] A. Alanqary, X. Gong, A. Keimer, B. Seibold, B. Piccoli, and A. Bayen, *Optimal control of autonomous vehicles for flow smoothing in mixed autonomy traffic*, To appear in: 2023 IEEE 62nd Conference on Decision and Control (CDC). IEEE, 2023.
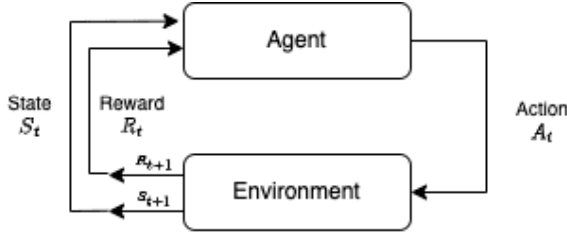
FIGURE 4 The Markov Decision Process that RL is based on. An agent exists in an environment and repeatedly chooses action based upon a state and receives rewards, which then informs the agent on the value of the state and action pair.

available. The anticipation component is an MPC module that aims to anticipate the leader's behavior based on its current acceleration. This enhances the controller's ability to follow the target speed effectively. The safety component is a safety module, that has priority over all the other components and ensures that the AV remains safe at all times. Since car-following models never represent traffic perfectly (see for instance [80], [81]), it is important that the safety module, at least, is independent of the mathematical model used to model vehicle behavior.

The mathematical expression of the commanded acceleration is given by

$$a_{\mathrm{cmd}} = \min(a_{\mathrm{safe}}, a_{\mathrm{target}}, a_{\mathrm{MPC}}) , \qquad (18)$$

where a_{safe}, a_{target}, and a_{MPC} correspond to the *safety*, *target*, and the MPC-based *anticipation* components. The safety component is given by

$$a_{\mathrm{safe}}(t) = -k(v(t) - v_{\mathrm{safe}}(t)) + \dot{v}_{\mathrm{safe}}(t) ,$$
$$v_{\mathrm{safe}}(t) = \sqrt{2|a_{\mathrm{min}}| \left(h(t) - s_0 + \frac{1}{2} \frac{v_{lead}^2(t)}{|a_{l,\mathrm{min}}|} \right)} , \qquad (19)$$

where k is a positive parameter, v is the velocity of the ego vehicle, s_0 is safety distance, h is the space gap, v_{lead} is the leader vehicle velocity and $a_{l,\mathrm{min}}$ is the minimal possible leader vehicle acceleration (so maximal deceleration), and a_{min} the minimal possible ego acceleration.

The target component is defined by

$$a_{\mathrm{target}}(t) = -k(v(t) - v_{\mathrm{target}}(t)) , \qquad (20)$$

where v_{target} is an inferred ideal speed obtained either from the speed planner when available or by integrating local measurements.

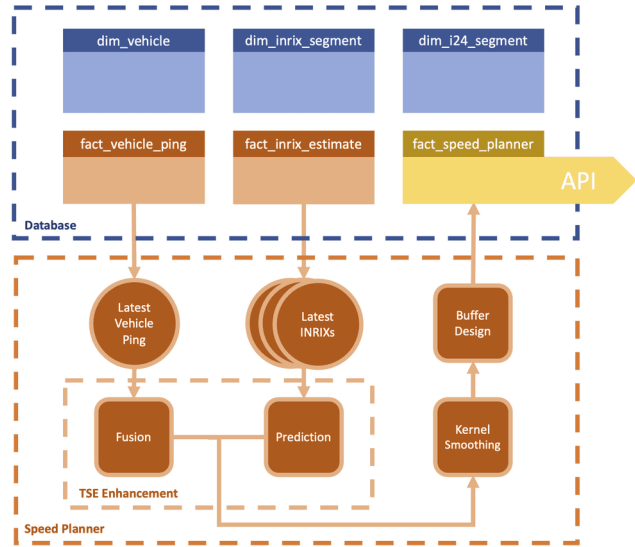The MPC-based anticipation component a_{MPC} is given

FIGURE 5 Data pipeline and major function modules of Speed Planner. At the beginning of each update, the Speed Planner extracts a combination of macroscopic TSE and vehicle observations from the corresponding factual tables (`fact_inrix_estimate`, `fact_vehicle_ping`) in the database to calculate the target speed profile. The raw TSE is used as the input of the prediction module, of which the output is fused with vehicle observations. The fusion is then smoothed and used in the buffer design module, of which the output is saved into the database (`fact_speed_planner`) and published as the target speed profile.

by

$$a_{\mathrm{MPC}}(t) = \begin{cases} a_{\text{min. brake}}(h(t),v(t),v_{lead}(t),a_{lead}(t)), \\ \qquad\qquad\text{if } P_1 > 0,\ a_{lead}(t) < 0 \\[4pt] a_{lead}v(t)/v_{lead}(t), \\ \qquad\qquad\text{if } P_1 \le 0 \text{ and } P_2 \ge 0,\ a_{lead}(t) < 0 \\[4pt] a_{lead} - \dfrac{(v(t) - v_{lead}(t))^2}{2(h(t) - s_0)}, \\ \qquad\qquad\text{if } P_1 \le 0 \text{ and } P_2 < 0,\ a_{lead}(t) < 0 \\[4pt] a_{lead} - \dfrac{(v(t) - v_{lead}(t))^2}{2(h(t) - s_0)}, \\ \qquad\qquad\text{if } P_2 < 0,\ a_{lead}(t) \ge 0, \\[4pt] \min(a_{\max}, a_{lead}(t)(1 + k_2(v_{\text{lead}}(t) - v(t)))), \\ \qquad\qquad\text{otherwise.} \end{cases} \tag{21}$$

where, as in (19), h is the space gap and s_0 the safety distance, and k_2 is a positive constant, a_{lead} is the leader vehicle acceleration, $a_{\max}$ is the maximal possible acceleration of the ego vehicle. The terms $a_{\text{min. brake}}$, P_1, and P_2 are given by

$$\begin{aligned} &a_{\text{min. brake}}(h(t),v(t),v_{lead}(t),a_{lead}(t)) \\ &= -\left(h(t) - s_0 + \frac{1}{2}\frac{v_{lead}^2(t)}{-a_{lead}(t)}\right)^{-1}\frac{(v(t))^2}{2}, \\ &P_1 = a_{\text{min. brake}}(h(t),v(t),v_{lead}(t),a_{lead}(t)) \\ &\quad - a_{lead}(t)v(t)/v_{lead}(t), \\ &P_2 = v_{lead}(t) - v(t). \end{aligned} \tag{22}$$

Further details of this controller can be found in [82].

Acceleration-Based Controller: Lane-Change
The design of Lane-Change Recovery Controller relies on global traffic information as well as local traffic states, such as the space gap to the leader vehicle and the relative velocity. The event of lane changing of the leader vehicle creates a discontinuity in the observed local state. Such discontinuities can translate to sudden, large jumps in the controller output, causing large jerk values. This sudden jump in the controller output might be necessary to avoid collision (for example, if a vehicle cuts-in in front of the AV and has low relative velocity). However, in many cases, these jumps are byproducts of the discontinuity in the observation and can be avoided without creating additional safety threats. To remedy this effect of lane-changes in such conditions, we designed a simple lane-change handling mechanism that treats the main acceleration-based controller as a black box and makes minimal assumptions

about it. Essentially, the mechanism works by detecting the event of lane changing, assessing the safety conditions created by the event, and (when appropriate) smoothing the output of the controller.

We assume discrete observations with fixed frequency. At a step k, we denote the ego velocity v_k, the relative velocity Δv_k, the space gap s_k, the output of the main controller a_k, and the actual acceleration input to the vehicle u_k. If the lane-change mechanism is not active, we have $u_k = a_k$.

In order to design this mechanism, we make a few natural assumptions about the behavior of the underlying controller. We assume that the controller is a continuous and non-decreasing function of the leader's states (that is, the space gap s_k and relative velocity Δv_k). We also assume that the main controller encodes its own safety measures as the lane-change handling mechanism does not provide any additional safety measures. Beyond these assumptions, the exact form of the main controller is treated as a black box.

We impose multiple criteria for the lane-change handling mechanism to take effect:

1) a lane-change event is detected—this is done by detecting a discontinuity in the space gap to the leader vehicle;
2) the lane-change event is significant—this is measured by the amount of jerk it causes;
3) the lane-change event does not cause safety threats—this is measured by the *time to collision* (TTC) (if $\Delta v_k < 0$) or time gap (if $\Delta v_k \ge 0$) at the lane-change event.

When the lane-change mechanism is activated, it smooths the acceleration by taking a convex combination of the main controller output at the current step and the actual acceleration command in the previous step:

$$u_k = a_k u_{k-1} + (1 - a_k)a_k. \tag{23}$$

Here $0 \le a_k < 1$ is a smoothing factor where larger values mean smoother acceleration change. Here we consider a time-varying a_k whose value is a non-decreasing function of the relative velocity and time headway at step k. The rationale for this modeling choice is to make the smoothing effect stronger the less critical the situation is (that is, larger space gap and relative velocity). The smoothing factor has the following functional form:

$$a_k = c \cdot \left(f_1\left(\frac{s_k}{v_k}\right)\right) + (1 - c) \cdot f_2(\Delta v_k, s_k). \tag{24}$$

It depends on the time headway through the function

$$f_1\left(\frac{s_k}{v_k}\right) = \tanh\left(t^\star \cdot \frac{s_k}{v_k}\right) \tag{25}$$

where the parameter $t^\star$ is chosen such then the function is close to 1 when the time headway $\frac{s_k}{\Delta v_k} \geq h_{\text{safe}}$, where h_{safe} is an acceptable safety time headway. The smoothing factor depends on the relative velocity through the function

$$f_2(\Delta v_k, s_k) = \begin{cases} \frac{1}{2}\tanh\left(\Delta v^\star \cdot \frac{s_k}{|\Delta v_k|}\right) + \frac{1}{2} & \Delta v_k < 0 \\ \frac{1}{2}\tanh\left(\Delta v_k\right) + \frac{1}{2} & \Delta v_k \geq 0 \end{cases} \tag{26}$$

where the parameter $\Delta v^\star$ is selected such that the function value is close to 0 when the time to collision $\frac{s_k}{|\Delta v_k|} \leq C_{\text{safe}}$, where C_{safe} is an acceptable safety time to collision. Concretely, we use $h_{\text{safe}} = 2s$, $C_{\text{safe}} = 4.5s$, $t^\star = 1.32$ and $\Delta v^\star = 10.3$. The parameters $c = 0.75$ is chosen by testing on multiple trajectories and lane-change scenarios.

Finally, the lane-change mechanism is deactivated when the main controller acceleration becomes close enough to the output acceleration

$$|a_k - u_k| \leq \epsilon. \tag{27}$$

An example of this mechanism is depicted in Figure 6. In the figure, the dashed red line indicates the lane-change event (cut-in) happening at around 14 seconds. The acceleration profile of the main controller (shown in gray in the bottom left panel) decelerates heavily in reaction to the lane-change. This event is detected and considered safe by the lane-change recovery controller which remained active for 2.9 seconds during which it smoothed the acceleration profile significantly and removed the unnecessary jerk cause by the lane-change.

ACC-Based Controller

The ACC-based controller is the version of the controller that was ultimately deployed on 97 of the 100 vehicles during the final MVT. The crucial distinction between the ACC-based controller and the acceleration-based controller introduced in the prior section is the controller's output. The ACC-based controller, rather than providing an acceleration to actuate, provides outputs setpoints for the AV's native ACC system, which controls the AV's longitudinal movements based on those setpoints. We use the stock ACC system's safety assurances and lane-change handling. Thus while this provides a less direct form of control, it is more robust in ensuring the safety and smoothness of the ride. For more information on how the ACC works, please refer to "Adaptive Cruise Control Modeling." For context on the simulator that was developed for training this algorithm, please refer to "Trajectory Simulator."

The ACC-based controller is an RL controller that is trained using Proximal Policy Optimization [83]. Elements of training the Markov Decision Process (MDP) problem [84] are described below:
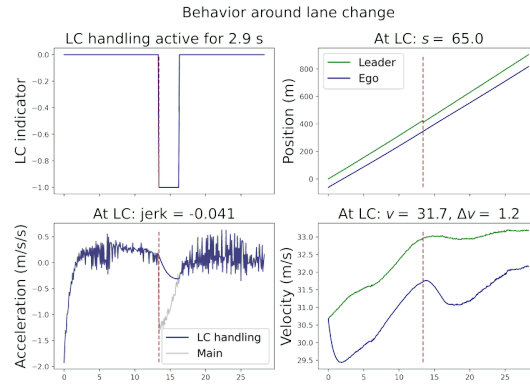
Observation Space

FIGURE 6 An example of the effect of the lane-change handling mechanism on a real-world trajectory in the event of a cut-in. *Top left*: lane-change event is detected and considered safe so the lane-change recovery controller was active for 2.9s. *Top right*: a car cuts-in in front of the ego vehicle at a headway of 65m. *Bottom left:* the main controller commanded acceleration drops sharply due to the lane-change event causing a large jerk value, but the lane-change controller smooths out this drop in the acceleration. Bottom right: The relative velocity is large enough at the lane-change allowing the controller to be active.

» v, velocity of the AV.
» v_s, target speed given by the Speed Planner.
» l, a "minicar" flag that indicates whether there is a leader vehicle detected, nominally within 80 meters (dubbed as such due to the miniature car icon that appears in the dashboard when the leader is detected).
» s, the current ACC speed setting
» g, the current ACC gap setting

Action Space

» The requested ACC speed setting, which our on-board computer will realize in a series of button presses (see Figure 7). The speed setting dictates the maximum speed at which the ACC can drive.
» The requested ACC gap setting, which the onboard computer will realize with priority over the speed setting. The gap setting takes on three bars between one and three, with each bar indicating a higher allowable gap. Each bar roughly corresponds to constant time gaps of 1.2, 1.5, and 2.0 seconds.

In addition, a clipping mechanism was used to ensure controller safety and social acceptability, particularly in the absence of leader state information. The *post facto* lower and upper bounds placed on the speed setting output are based on the average speed of the ego vehicle during the last one second (ten timesteps, with 0 mph and null

FIGURE 7 A photograph of a Nissan Rogue's steering wheel buttons that control the vehicle's ACC. The ACC system is turned on by manually pressing the blue icon on the far right. Through our vehicle interfacing efforts, we are able to electronically press the $+$ and $-$ buttons to toggle the ACC speed setting up 1 mph and down 1 mph, respectively, or hold them to increment by 5 mph. The three ACC gap settings are rotated through by pressing the button on the bottom right.

observations omitted):

$$v_{\text{lower}} = \frac{1}{10} \sum_{i=1}^{10} v_i - 15\text{mph} ,$$

$$v_{\text{upper}} = \frac{1}{10} \sum_{i=1}^{10} v_i + 5\text{mph} ,$$

$$(28)$$

where v_{lower} and v_{upper} are the lower and upper bounds of the clip, respectively. The final clip is executed as:

$$v_{\text{clip}} = \min(\max(v_{\text{action}}, v_{\text{lower}}), v_{\text{upper}})$$

$$v_{\text{final}} = \min(\max(v_{\text{clip}}, 20\text{mph}), 73\text{mph}) ,$$

$$(29)$$

where v_{final} is the final speed setting command and v_{action} is the un-normalized speed output, or action, from the neural net. The 73 mph maximum speed accounts for modest speeds in excess of the posted speed limit of 70 mph.

Reward Function

The reward function is:

$$r_t = 1 - c_1 a_t^2 - c_2 (v_t^{av} - v_t^{sp})^2 + ...$$

$$- \frac{c_3}{n} \sum_{i=1}^{n} E_t^i - c_4 \mathbb{1}_{h_t^{av} \le h_{min} \vee h_t^{av} \ge h_{max}} ,$$

$$(30)$$

where $c_1 - c_4$ are coefficients, a_t^2 is an acceleration penalty; $(v_t^{av} - v_t^{sp})^2$ is a squared penalty on the difference between the Speed Planner's suggested speed and the actual speed; $\frac{1}{n} \sum_{i=1}^{n} E_t^i$ is instantaneous fuel consumption according to the energy models described in "Vehicle Energy Models;" and the last indicator term is an intervention penalty that is invoked if the space gap is less than the minimum space gap or greater than the maximum space gap.

Further details on the development and design decisions for the state and reward representations and the intermediate and final policies are given in depth in [41].

CANDIDATE CONTROLLER SELECTION

In the months leading up to the MVT open road test, our team developed a multitude of candidate controllers, as seen in previous sections and sidebars. In order to pare down the list of candidate controllers to be considered for real-world deployment in November 2022, various (Speed Planner, ACC-based Vehicle Controller)-combinations were assessed over a range of simulation scenarios. The simulation and testing framework follows from [87] and features an updated simulation methodology described in "Trajectory Simulator."

Simulation scenarios included shockwaves, bottlenecks, and freeflow. Shockwave and freeflow scenarios follow precisely from the simulator described in "Trajectory Simulator" and only differ in the leader trajectory (one being stop-and-go, and the other being all high speed). The bottleneck scenario features a dynamically imposed speed limit in a spatial region of the domain. The speed limit is inversely related to the vehicle density of the region, and the scaling parameter is tuned to historically observed speeds in the I-24 region. The bottleneck region is meant to model a weaving area where a close proximity of on-ramps and off-ramps results in increased lane-change frequency.

The key performance indicators (KPIs) used to assess the controllers include fuel economy, throughput, and network speed. Fuel economy is the overall miles per gallon of all vehicles, and it is computed by applying the energy model (see "Vehicle Energy Models") and computing

$$\text{fuel economy} = \frac{\sum_{t=0}^{T} \sum_{i=1}^{n} E_t^i}{\sum_{i=1}^{n} x^i} , \qquad (32)$$

where the numerator is the total fuel consumed by all vehicles over all simulation time, and the denominator is the total distance traveled by all vehicles. The throughput is measured by counting the number of vehicles crossing various positions along the highway, normalized by time. For the shockwave and freeflow simulations, this is taken to be the straight average of the time-average of five equispaced measurement locations. For the bottleneck, this is taken to be the steady-state (or final value) at the measurement location just downstream of the bottleneck region. The network speed KPI is defined as the total distance traveled by all vehicles divided by the total driving time of all vehicles.

At the time of our self-imposed "controller freeze" (that is, the time at which substantial changes to controllers are no longer allowed), there were two Speed Planner variants and 12 Vehicle Controller variants. KPI performance for each of these controllers is shown in Figure 8. The Speed Planner variants are kernel smooth (just the first three steps

Vehicle Energy Models

by Nour Khoudari, Sulaiman Almatrudi, Rabie Ramadan, Joy Carpio, Mengsha Yao, Kenneth Butts, Jonathan W. Lee, Benjamin Seibold

The quantification of the energy demand of the vehicles on the road, given their trajectories, requires vehicle-specific energy models that take as an input the velocity profile $v(t)$ and the road grade profile $\theta(t)$ and output the resulting energy/fuel consumption rate $f(t)$. This project requires the quantification of the energy demand of traffic flow at large, composed of many vehicles; and also the use of reinforcement learning and optimization techniques that minimize (under certain constraints) the energy demand of traffic. Thus, the energy models used should accurately represent different vehicle types on the road and should average out any local non-convexity behavior due to gear switching, to avoid trapping the optimizer in local minima. For that purpose, we use energy models derived from a systematic model-reduction procedure to generate simple fitted models. The procedure starts from the fidelity software Autonomie [85], for a number of vehicles, each of which represents a typical average vehicle of a given class.

VEHICLE PORTFOLIO

To capture the diversity and prevalence of different vehicle types on US roads we select a representative group of vehicle classes on which we apply the model-reduction process to derive their corresponding simplified energy models. Those vehicle classes are divided into two categories: (1) light-duty vehicles: compact size sedan, midsize sedan, midsize SUV, and Pickup, and (2) heavy-duty vehicles: Class4PND (Pickup and Delivery) and Class8Tractor. Each vehicle model represents a class of vehicles that have comparable weight (with load assumed half full) and fuel consumption characteristics [S1].

AUTONOMIE AND VIRTUAL CHASSIS DYNAMOMETER (VCD)

We use the simulation software Autonomie Rev 16SP7 [85] with a library of energy models for several types of vehicles, including a detailed plant and controller model for its components in MATLAB and Simulink, where the blocks and files can be customized. To build our models, we use (i) physics-based vehicle parameters extracted directly from Autonomie, (ii) tuned parameters extracted in an automated fashion by running the Autonomie model on test cycles, and (iii) performance maps computed gear-by-gear on a complete velocity–load phase space of driving by running Autonomie's customized vehicle models on a VCD (those maps are vehicle speed to engine speed, vehicle speed and wheel force to engine torque, and engine speed and torque to fuel rate).

SEMI-PRINCIPLED ENERGY MODELS

We build an energy model that is semi-principled in that it has a physics-based part using Autonomie's extracted physics-based vehicle parameters, but it also relies on the maps obtained from the VCD. Gear scheduling in this model is based on choosing the feasible gear that yields the minimal fuel consumption, and the torque converter bypass clutch is assumed open in the first gear only. In contrast to Autonomie that considers hysteresis effects, this model yields the fuel consumption rate (and other outputs) as a direct function of instantaneous velocity v, acceleration a, and road grade θ.

SIMPLIFIED ENERGY MODELS

A further model-reduction step is conducted by fitting the semi-principled models into simplified models. Those models have a simple polynomial structure that can easily be integrated into optimization and control problems, yet they are highly accurate. The fuel consumption rate function is

$$f(v, a, \theta) = \max \left\{ \beta, \ C(v) + P(v)a + Q(v)a_+^2 + Z(v)\theta \right\}, \tag{S31}$$

where β is the minimum fuel rate set to be zero or a positive constant depending on the fuel cut criteria, $a_+ = \max(-\frac{P(v)}{2Q(v)}, a)$, $C(v) = c_0 + c_1 v + c_2 v^2 + c_3 v^3$, $P(v) = p_0 + p_1 v + p_2 v^2$, $Q(v) = q_0 + q_1 v$, and $Z(v) = z_0 + z_1 v + z_2 v^2$.

In the above functions, c_0 ensures that fuel is being consumed at idle, the c_1, c_2, and c_3 terms can be interpreted as fuel consumed due to friction and air-drag, the $P(v)$ term yields fuel demand due to non-zero accelerations, the $Q(v)$ term captures the super-linear trend of fuel rate with respect to a, and the $Z(v)$ term captures the fuel consumed due to road grade, with the z_1 term playing the role of the weight force exerted at θ. Both types of models are validated, for all different vehicle types, against Autonomie models as the ground truth on standard EPA drive cycles [S2] for flat roads and constant road grades drive cycles, and the results showed that the models are highly accurate (within 4% for zero road grades). See [86] for more details on the design of the energy models and corresponding results.

REFERENCES

[S1] Argonne National Laboratory (ANL). "Autonomie Compiled Vehicles," *A technical manual for training in Autonomie*, 2020.

[S2] United States Environmental Protection Agency (US EPA). "Dynamometer Drive Schedules, 2022.," *https://www.epa.gov/vehicle-and-fuel-emissions-testing/dynamometer-drive-schedules*.

described in Speed Planner) and kernel smooth with RL buffer (all steps described in Speed Planner). The Vehicle Controller variants include:

- » **Simple**: a hand-designed logic-based controller that largely adheres to the Speed Planner suggestion.
- » **MicroAccel**: the main controller described in Acceleration-Based Controller with modifications to output ACC set points. This controller itself has six variants with different parameter choices.
- » **RL**: the RL controller described in ACC-Based Controller. This controller has five variants with different training meta-parameter choices.

In the shockwave scenario, an ideal controller should maximize fuel economy improvements and remain neutral on throughput and network speed. In the bottleneck scenario, an ideal controller should maximize throughput improvements and optionally improve or stay neutral on the other two KPIs. In the freeflow scenario, an ideal controller will not worsen on any of the KPIs. Given these criteria, we determined that the HybridRL Vehicle Controller paired with the Kernel Smooth with RL Buffer Speed Planner presented the ideal combination for the ACC-Based vehicles.

OPEN ROAD FIELD OPERATIONAL TEST

Following the controller selection, this section introduces the implementation details of the selected controllers, experimental design for the 100 AV deployment, and contemporaneous data collection procedures.

Server-side implementation

As indicated in Figure 5, the server-side implementation includes the construction of the database, the API, and the Speed Planner algorithm scripts. The database schema is designed following the star schema data model, consisting of dimension tables (for storing static metadata) and fact tables (for storing quantitative data). The tables include:

- » `dim_vehicle` for storing the vehicle metadata, route and lane assignment;
- » `dim_inrix_segment` for storing INRIX road segment data for I-24;
- » `dim_i24_segment` for storing finer-grained road segment data, which supports TSE and Speed Plan profiles after data fusion;
- » `fact_vehicle_ping` for storing the realtime AV ping information;
- » `fact_vehicle_observation` for storing extracted vehicle observations from the AV ping table;
- » `fact_inrix_estimate` for storing realtime INRIX data;
- » `fact_speed_planner` for storing Speed Plan profiles.

The `fact_vehicle_ping` and `fact_inrix_estimate` tables are continually being populated with realtime updates at 1Hz. After processing these input data via the Python-based Speed Planner algorithm, the Speed Plan is written to the `fact_speed_planner` table, which is exposed to the internet via a PHP-based HTTP API.

Vehicle-side implementation

The vehicle-side implementation includes several software libraries and a mix of custom and off the shelf hardware components.

In the past decades , vehicles have been gradually increasing the implementation of the open Controller Area Network (CAN) protocol for in-vehicle networking communication between various electrical modules. The usage

SHOCKWAVE SCENARIO	Speed Planner					
	Kernel Smooth			Kernel Smooth with RL Buffer		
Vehicle Controller	Fuel Economy % Increase	Throughput % Increase	Speed % Increase	Fuel Economy % Increase	Throughput % Increase	Speed % Increase
Simple	17.33%	1.18%	-0.32%	24.24%	-0.04%	-5.86%
MicroAccel-a	9.33%	-0.08%	-0.37%	10.71%	1.18%	-0.75%
MicroAccel-b	4.56%	0.26%	7.68%	8.87%	2.27%	2.87%
MicroAccel-c	9.33%	-0.08%	-0.37%	10.71%	1.18%	-0.75%
MicroAccel-d	9.33%	-0.08%	-0.37%	10.71%	1.18%	-0.75%
MicroAccel-e	9.33%	-0.08%	-0.37%	10.71%	1.18%	-0.75%
MicroAccel-f	12.20%	1.02%	0.37%	14.94%	0.35%	0.45%
RL-a	14.31%	-1.11%	-1.72%	13.25%	-1.32%	-1.69%
RL-b	12.38%	-1.10%	-1.87%	11.63%	-1.45%	-1.87%
RL-c	-2.76%	0.82%	0.37%	-1.42%	0.36%	0.15%
RL-d	3.24%	0.26%	0.67%	-0.53%	0.11%	-0.27%
RL-e	23.18%	2.84%	-2.19%	18.47%	2.04%	-1.69%
HybridRL	22.96%	2.75%	-2.29%	17.49%	1.69%	-2.14%

BOTTLENECK SCENARIO	Speed Planner					
	Kernel Smooth			Kernel Smooth with RL Buffer		
Vehicle Controller	Fuel Economy % Increase	Throughput % Increase	Speed % Increase	Fuel Economy % Increase	Throughput % Increase	Speed % Increase
Simple	6.85%	13.57%	8.76%	8.44%	14.34%	12.63%
MicroAccel-a	4.90%	12.77%	4.21%	3.97%	13.16%	3.87%
MicroAccel-b	1.81%	13.55%	8.16%	-0.60%	13.16%	7.99%
MicroAccel-c	4.90%	12.77%	4.21%	3.97%	13.16%	3.87%
MicroAccel-d	4.90%	12.77%	4.21%	3.97%	13.16%	3.87%
MicroAccel-e	4.90%	12.77%	4.21%	3.97%	13.16%	3.87%
MicroAccel-f	1.26%	13.16%	3.35%	1.09%	12.39%	3.18%
RL-a	4.25%	9.30%	4.47%	3.45%	9.30%	4.64%
RL-b	3.83%	9.30%	4.55%	3.36%	9.30%	4.64%
RL-c	0.88%	13.16%	3.18%	0.88%	13.16%	3.18%
RL-d	0.91%	12.77%	3.78%	1.06%	8.82%	4.81%
RL-e	8.22%	14.34%	8.08%	6.72%	14.73%	8.68%
HybridRL	2.95%	14.73%	9.19%	2.92%	14.74%	9.19%

FREEFLOW SCENARIO	Speed Planner					
	Kernel Smooth			Kernel Smooth with RL Buffer		
Vehicle Controller	Fuel Economy % Increase	Throughput % Increase	Speed % Increase	Fuel Economy % Increase	Throughput % Increase	Speed % Increase
Simple	1.13%	-0.56%	-2.14%	-2.79%	-1.24%	-1.81%
MicroAccel-a	14.50%	5.13%	-11.63%	14.52%	5.21%	-11.53%
MicroAccel-b	13.78%	4.49%	-12.73%	13.80%	5.90%	-11.63%
MicroAccel-c	14.50%	5.13%	-11.63%	14.52%	5.21%	-11.53%
MicroAccel-d	14.50%	5.13%	-11.63%	14.52%	5.21%	-11.53%
MicroAccel-e	14.50%	5.13%	-11.63%	14.52%	5.21%	-11.53%
MicroAccel-f	15.68%	0.33%	-15.38%	15.81%	0.63%	-15.20%
RL-a	27.27%	2.50%	-25.19%	26.39%	2.65%	-22.38%
RL-b	32.76%	2.33%	-23.78%	25.27%	3.47%	-21.54%
RL-c	-0.19%	1.49%	0.43%	-0.03%	1.60%	0.26%
RL-d	0.05%	0.12%	-0.86%	0.05%	0.12%	-0.86%
RL-e	29.47%	1.41%	-34.28%	25.13%	2.35%	-27.52%
HybridRL	-0.19%	1.49%	0.43%	-0.03%	1.60%	0.26%

FIGURE 8 Summary of controller assessments prior to MVT deployment. Candidate controller components to the MegaController were simulated under three scenarios (top: shockwave; middle: bottleneck; bottom: freeflow) and evaluated against key performance indicators (fuel economy, throughput, network speed) relative to a baseline simulation of homogeneous human drivers. Speed Planner candidates are shown across the top, and Vehicle Controllers are shown in rows. Key performance indicators are shown in columns. Results are color-coded from improved (green) to neutral (white) to worsened (red).

Trajectory Simulator

by Nathan Lichtlé

Establishing a robust, representative, and reliable simulation environment is key for the successful implementation of our wave-smoothing controllers in autonomous vehicles, as it ensures their efficient operation across a wide range of traffic conditions. Our simulator [S1] is based on real human driver data collected on the highway[S2], and has been extensively used for designing, training and evaluating the different types of controls implemented in this work, assessing their safety, energy-reducing performances, robustness and smoothness. This sidebar describes the procedure by which highway trajectory data was gathered and a simulator created from it.

ACQUISITION OF DATA

The trajectory dataset [S2] for our study were recorded on a 14.5-kilometer segment of I-24, located southeast of Nashville, Tennessee. An example recorded speed trajectory profile can be seen in Figure S3. An instrumented vehicle is used to gather data which logs vehicle controller area network (CAN) data through `libpanda` [65] and GPS information from an in-built receiver. The CAN data collection comprises measurements like the speed of the vehicle under consideration (ego vehicle), the relative speed of the lead vehicle (the vehicle in front), the instantaneous acceleration, and the space gap (distance from bumper to bumper).
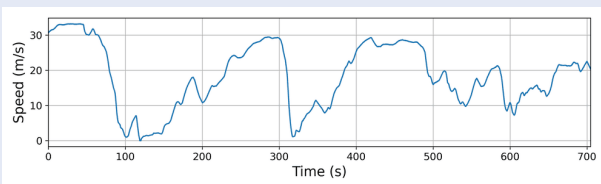
FIGURE S3 Speed vs. time for one of the dataset trajectories, exhibiting large acceleration and breaking patterns that can typically lead to stop-and-go waves.

REFINING RAW DATA

Each drive's raw data is stored in two separate files: a CAN data file and a GPS file. The relevant data are extracted from the CAN file and adjusted to match the GPS time, which is recorded at 10 Hz. The high-frequency CAN data are downsampled and interpolated linearly to align with GPS time, whereas low-frequency CAN data are subjected to linear interpolation to match the 10 Hz GPS time. GPS position data is used to calculate the distance traveled and direction. As westbound data usually demonstrate more consistent congestion, they are primarily used for training, comprising 60 trajectories, which translates to 8.8 hours and 772.3 kilometers of driving.

ANALYZING THE DATASET

The trajectory dataset encapsulates a wide variety of traffic conditions ranging from nearly stationary congested traffic to maximum speed free-flow traffic, including diverse acceleration and deceleration patterns associated with stop-and-go traffic. The example trajectory displayed in Figure S3 demonstrates the ego vehicle's quick transitions between low and high speeds.

While the primary interest lies in mitigating high-frequency waves that are common in congestion, Figure S4 indicates the tendency of speeds in the training dataset towards the higher end. Despite the possibility of simplifying the learning problem by filtering out high-speed data, congestion zones are often immediately followed by high-speed areas. To ensure our controller's competent behavior at high speeds and during transitions from high to low-speed zones, the training dataset retains both low and high-speed data.
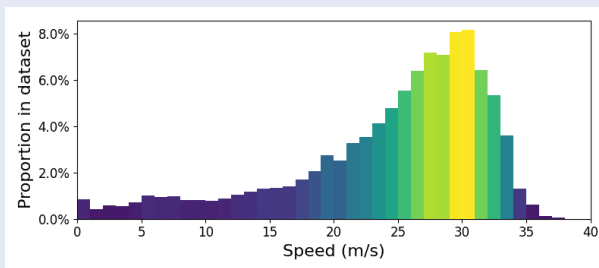

FIGURE S4 Distribution of speeds in the I-24 dataset.

DEVELOPING THE TRAINING AND EVALUATION FRAMEWORK

To exploit the gathered data, a single-lane training environment is designed where an AV follows the trajectory data recorded from human drivers. The human driver is simulated at the front of a vehicle platoon, followed by the AV, and then a number of vehicles operating according to the *Intelligent Driver Model* (see "Car-Following Models"). This setup guarantees the growth of waves in congestion, due to the string-unstable nature of the chosen IDM parameters. While a comprehensive micro-simulation of I-24 might allow training with more complex vehicle interactions, the proposed simulator focuses on realistic driving dynamics representative of the highway's wave types and drivers' reactions to wave formation. The simulator's efficiency is also commendable as it achieves 2000 steps per second, whereas a micro-simulation of the complete 14-kilometer section would be computationally expensive due to the thousands of vehicles in congestion.

REFERENCES

[S1] N. Lichtlé, E. Vinitsky, M. Nice, B. Seibold, D. Work and A.M. Bayen, *Deploying Traffic Smoothing Cruise Controllers Learned from Trajectory Data*, 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 2022, pp. 2884-2890, doi: 10.1109/ICRA46639.2022.9811912.

[S2] M. Nice, N. Lichtlé, G. Gumm, M. Roman, E. Vinitsky, S. Elmadani, M. Bunting, R. Bhadani, K. Jang, G. Gunter, M. Kumar, S. McQuade, C. Denaro, R. Delorenzo, B. Piccoli, D. Work, A. Bayen, J. Lee, J. Sprinkle, and B. Seibold, *The I-24 Trajectory Dataset (1.1.1)*, 2021. Zenodo. https://doi.org/10.5281/zenodo.6456348

Car-Following Models

by Nour Khoudari, Benjamin Seibold

C*ar-following models* (CFMs) [88], [51] are systems of ordinary differential equations (ODEs) describing the dynamics of each vehicle on the road, where drivers react to the changes in the relative positions of the vehicle ahead. The ODEs could describe the vehicle velocity only and those are first-order models, or the velocity and the acceleration and those are second-order models. Second-order CFMs are of the form

$$\dot{v}(t) = f(s(t), v(t), \Delta v(t)) , \qquad (S33)$$

where s is the gap to the vehicle ahead measured in meters, v is the velocity of the vehicle, commonly measured in m/s, and Δv is the velocity difference or the approaching rate to the vehicle ahead, commonly measured in m/s. An elegant CFM is the Optimal Velocity Model (OVM) [75]

$$f(s, v, \Delta v)_{OVM} = \alpha \left[V(s) - v \right] , \qquad (S34)$$

where the optimal velocity function, $V(s)$, determined by the gap s to the vehicle ahead, is a positive monotonically increasing function, with $s \to \infty$ asymptote at the speed limit. Some variations of this model were proposed to avoid car collisions, an example is the Optimal Velocity Follow the Leader Model (OVM-FtL):

$$f(s, v, \Delta v)_{OVM\text{-}FtL} = \alpha \left[V(s) - v \right] + \beta \left[\frac{\Delta v}{s^{\nu}} \right] , \qquad (S35)$$

where ν is a positive exponent, and β is a positive braking coefficient (measured in m^{ν}/s). Another example of second-order CFMs is the Intelligent Driver Model (IDM), introduced in [89] and suitably adapted and used in this work. The IDM acceleration function is

$$f(s, v, \Delta v)_{IDM} = a \left[1 - \left(\frac{v}{v_0} \right)^{\delta} - \left(\frac{s^*(v, \Delta v)}{s} \right)^2 \right] , \qquad (S36)$$

where $s^*(v, \Delta v) = s_0 + vT + \frac{\max\{0, v\Delta v\}}{2\sqrt{ab}}$. Here v_0 is the desired velocity on an empty road (measured in m/s), s_0 represents the minimum spacing between vehicles (measured in m), T is the minimum possible time gap to reach the vehicle ahead (measured in s), δ is an acceleration exponent, and a and b are the maximum vehicle acceleration and minimum desired comfortable deceleration, respectively (measured in m/s^2).
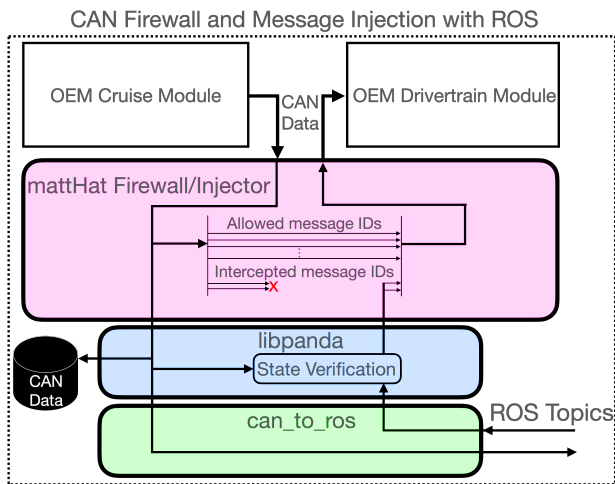

FIGURE 11 CAN firewall and message injection with ROS. A special CAN interface called the mattHat firewalls messages between OEM modules at the hardware level, and can replace blocked messages with third party messages from software with `libpanda`. `libpanda` provides direct CAN recording as well as verification of CAN data to be sent by checking OEM's CAN states. CAN messages can be read and sent through ROS topics provided by adapters in `can_to_ros`.

of the CAN bus lets automotive manufacturers use minimal wiring to design complex systems involving anything from engine diagnostics, infotainment, security, emissions, and (in more recent years) adaptive cruise control (ACC) and Lane Keep Assist (LKA). The ACC system on cars typically involves a sensor module to measure the leading vehicle dynamics and a separate controller module, which communicates with the vehicle's transmission and engine. Since these modules are physically located in different parts of the vehicle they communicate using CAN busses, where listening CAN analyzers can record the information. Figure 11 shows the method of tapping into a CAN bus for message reading.

While automotive manufacturers provide information on the structure of the in-vehicle network in the form of wiring diagrams, specific information being sent on the CAN bus is typically a closely guarded secret, likely due to both trade secrets and safety concerns. However, some companies (such as comma.ai [90] and Intrepid Control Systems [91]) are entering the space of custom vehicle autonomy by selling modules that intercept CAN messages between these modules and provide their own inputs based on custom controllers. This is only possible after spending time decoding messages and reverse engineering the protocols needed to inject custom messages. Compared to basic CAN reading, Figure 11 shows the different electrical architecture needed for CAN message interception and injection between OEM modules.

In this work our solution is based on a similar principle, however additional protocols on the 2023 Nissan Rogue (which comprise the vast majority of our control fleet) prevented the injection of commands, though CAN data could still be read for realtime controller inputs. Due

Vehicle Interfacing

by Matthew Bunting

The set of hardware involves both off-the-shelf and custom components assembled as a stack of Printed Circuit Boards (PCB). Figure S5 shows the relationship of hardware components installed in each car.

HARDWARE

A Raspberry Pi 4 served as the main computer to run controllers. This was chosen due to its wide open-source support, and its availability and low cost. The operating system was 64-bit Raspbian Lite. We chose Raspbian instead of other Linux distributions since it was the only actively fully supported version from the hardware manufacturer.

Attached to the Raspberry PI was an x728 battery backup Uninterrupted Power Supply (UPS). An experiment lifecycle began turning on the car and therefore providing power to the Raspberry Pi, then ended when the vehicle parked and shut off at our headquarters. The UPS would continue to provide power to the Pi, and signal that the Pi should stop processed of custom control and upload the experiment's data over WiFi.

To provide internet to the Pi during an experiment, needed for receiving control setpoints, a separate mobile hotspot was connected to the Pi's ethernet. Our chosen hotspot was an industrial grade Cradlepoint IBR900 and IBR600. This hardware setup was shown to be effective for societal-scale experiments [S1].

A uBlox M8 series USB GPS module provided the system with positional tracking. This sensor was also used to synchronize the Raspberry Pi's system clock to GPS time, to ensure that later data processing would involve minimal manual realignment.

The mattHAT served as the interface for the vehicles, with the majority being a 2023 Nissan Rogue. The only standard interface on vehicles is the *Onboard Diagnostics* (OBD) port, however this only provides useful-yet-minimal data like the *Vehicle Identification Number* (VIN) While modern vehicles have a rich set of sensors and actuators using standardized CAN bus for communication, they are minimally documented to prevent third parties from interpreting sensors and from sending actuation commands. This leads to a significant effort to decode CAN signals. While off-the-shelf CAN decoders exist, our solution had to be custom in order to send low-level electrical signals to operate the stock ACC unit. This was possible for the Nissan Rogue by applying an electrical resistance to spoof button presses, requiring a custom circuit. The mattHAT was designed with these three communication components in mind, to read the VIN over OBD, read vehicle sensors and state information over specific CAN busses, and to send low-level ACC control commands.
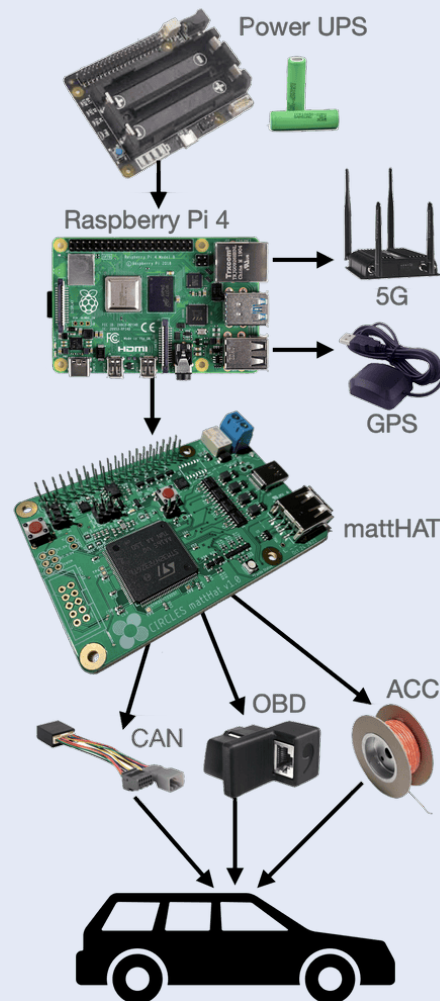

FIGURE S5 The set of hardware installed in each vehicle.

SOFTWARE

`libpanda` is a low-level C++ library to operate the mattHAT, written with a focus on data timeliness and low CPU usage [S2]. `libpanda` abstracted the vehicle interface in the form of an observer design pattern to easy write software for reading CAN and GPS data, and to send control commands wither through CAN message injection or through the mattHAT's ACC interface. In addition to low-level operation of the vehicle, `libpanda` also features a set of utilities to manage the Raspberry Pi for functions like automatic data upload, *Over-the-Air* (OTA) updates, and reporting system status.

Using `libpanda`, a ROS node was designed to translate the various vehicle functions into the form of ROS topics. A project named `can_to_ros` provided pre-written middleware so that controller designers could use tools like MATLAB's Simulink to generate code in the form of C++ to control the vehicle [S3]. `can_to_ros` was also expanded to handle a heterogeneous fleet, due to the mixture of models and make of vehicles [S4] [S5]. In conjunction with `can_to_ros` and `libpanda`, a data analysis tools named `bagpy` [S6] and `strym` [S7] were used to quickly analyze and decode recorded data. Data from the signal decoding and system characterization process for a 2020 Toyota RAV4 has also been made publicly available [S8].

REFERENCES

[S1] A. Richardson, M.W. Nice, M. Bunting, J. Lee, R. Bhadani, D. Work, and J. Sprinkle, "Analysis of a runtime data sharing architecture over lte for a heterogeneous cav fleet," in Proceedings of Cyber-Physical Systems and Internet of Things Week 2023, 2023, pp. 164–169.

[S2] M. Bunting, R. Bhadani, and J. Sprinkle, "Libpanda: A high performance library for vehicle data collection," in Proceedings of the Workshop on Data- Driven and Intelligent Cyber-Physical Systems, 2021, pp. 32–40.

[S3] S. Elmadani, M. Nice, M. Bunting, J. Sprinkle, and R. Bhadani, "From can to ros: A monitoring and data recording bridge," in Proceedings of the Workshop on Data-Driven and Intelligent Cyber-Physical Systems, 2021, pp. 17– 21.

[S4] M. Nice, M. Bunting, D. Work, and J. Sprinkle, "Middleware for a heterogeneous cav fleet," in Proceedings of Cyber-Physical Systems and Internet of Things Week 2023, 2023, pp. 86–91.

[S5] R. Bhadani, M. Bunting, M. Nice, F. Wu, A. Hayat, J. W. Lee, A. Bayen, B. Piccoli, B. Seibold, D. Work et al., "Approaches for synthesis and deployment of controller models on automated vehicles for car-following in mixed autonomy," in Proceedings of Cyber-Physical Systems and Internet of Things Week 2023, 2023, pp. 158–163.

[S6] R. Bhadani and J. Sprinkle, "Prototyping vehicle control applications using the cat vehicle simulator," arXiv preprint arXiv:2301.04574, 2023.

[S7] R. Bhadani, M. Bunting, M. Nice, N. M. Tran, S. Elmadani, D. Work, and J. Sprinkle, "Strym: A python package for real-time can data logging, analysis and visualization to work with usb-can interface," in 2022 2nd Workshop on Data-Driven and Intelligent Cyber-Physical Systems for Smart Cities Workshop (DI-CPS). IEEE, 2022, pp. 14–23

[S8] M. Bunting, R. Bhadani, M. Nice, S. Elmadani, and J. Sprinkle, "Data from the development evolution of a vehicle for custom control," in 2022 2nd Workshop on Data-Driven and Intelligent Cyber-Physical Systems for Smart Cities Workshop (DI-CPS). IEEE, 2022, pp. 40–46.
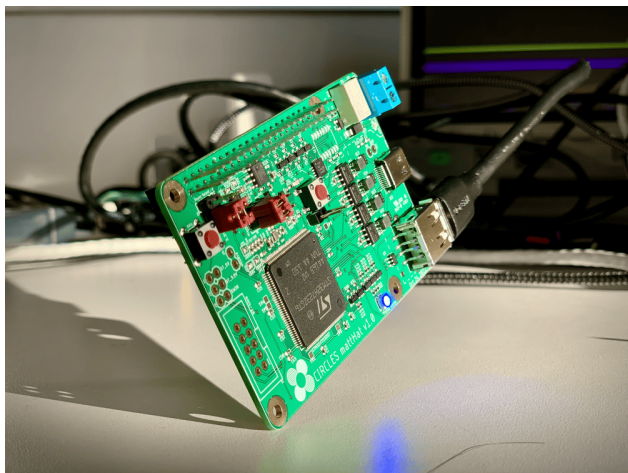
FIGURE 13 The mattHat. This custom designed circuit board was rapidly manufactured for both CAN interfacing and ACC button command spoofing.

primarily to supply chain issues and additionally to an inability to inject CAN messages, a custom circuit board was developed, coined mattHAT (Hardware Attached on Top). A mattHat can be seen in Figure 13. The mattHAT acted as a CAN interface and also provided methods of sending ACC button press commands using custom circuitry. The board was designed to plug into a Raspberry Pi 4. Custom wire harness cables were designed and built for vehicle installation across the 97 Nissan Rogue vehicles, targeting specific CAN busses related to the vehicle's ACC state.

A C++ based library named `libpanda` [65] was developed to operate the mattHAT's CAN interface and button spoofer as well as record raw CAN and GPS data. `libpanda` also supports a USB-based GPS module to both record the vehicle's position and to synchronize the system's time. A tool named Strym was also developed to decode and quickly analyze libpanda's recorded data for CAN signal decoding and classification [92], [93], [94]. To allow control designers to easily make use of `libpanda`, the Robot Operating System (ROS) was installed on the Raspberry Pi. A set of ROS nodes were designed in the software project named `can_to_ros` [66], [69], which abstracts various sensors and actuators into a set of ROS topics. A tool named `bagpy` was also developed to quickly process and plot recorded ROS data, in the format of bag-files [95]. With the infrastructure of ROS in place, control designers could use modeling software like MATLAB's Simulink to design controllers and generate code to greatly ease controller integration.

`libpanda` also features a set of auxiliary services to manage the vehicle at scale. A method to automatically perform over the air (OTA) updates was implemented so that vehicles would not need to be manually handled. An additional server integration named `piStatus` was built so that the system could regularly post its status to easily assess the system's health. Figure 14 shows headquarters personnel monitoring `piStatus`, ensuring vehicles pass status checks before deployment. `piStatus` would be continuously monitored throughout each experiment to note any hardware issues so that vehicles could undergo

FIGURE 14 `piStatus` web interface. Co-author Matthew Bunting monitors the `piStatus` page to clear vehicles for experiment deployment.

maintenance before the next deployment. A shutdown script also worked in tandem with an off-the-shelf battery-backup Pi HAT to perform automatic data uploads at the end of an experimental drive when the vehicle was shut off. Figure 15 shows a WiFi antenna installed in the center of the vehicle parking lot for granting each vehicle's embedded computer internet access for data upload and OTA updates. Without these set of services, managing and maintaining a project at this scale would have been impossible.

FIGURE 15 Parking lot WiFi antenna. A WiFi antenna is placed in the center of the vehicle parking lot, enabling OTA data uploads and software updates for each Raspberry Pi in each vehicle.

I-24 MOTION

The CIRCLES team deployed the vehicle controllers on I-24 southeast of Nashville, TN due to the recent creation of a new testbed, known as I-24 Mobility Technology Interstate Observation Network (MOTION) [53], [54]. I-24 MOTION is a four-mile section of I-24 designed to produce ultra-high resolution trajectory data of all vehicles on the roadway for the purposes of traffic science and experimentation on automated vehicles and traffic management. The system produces trajectory data (Figure 16) using 276 cameras (Figure 17) on fixed roadside poles between 110–135ft tall, to minimize visual occlusion. The debut of the I-24 MOTION testbed coincided with the live CIRCLES experiment, for which the testbed is uniquely suited to gather data on the traffic stream impacts of a fleet of test vehicles.

Processing of raw video into vehicle trajectories by I-24 MOTION happens in two stages. 1) A computer vision pipeline [97], [54] makes the initial vehicle detection and type classification from the raw video [97], including vehicle dimensions by using 3-D bounding boxes. Vehicle classes used by the object detection are: sedan, midsize, pickup, van, semi, truck, motorcycle. It tracks detected vehicles across adjacent camera views in the vehicle's direction of travel. The computer vision processing is distributed across ten servers with contiguous groups of cameras allocated to each. Vehicles are not tracked between servers, so a vehicle trajectory covering the length of the testbed consists of at least ten fragments; additional fragmentation may occur due to unavoidable occlusion (for example, by overpasses). The computer vision pipeline also converts image space coordinates (where vehicle 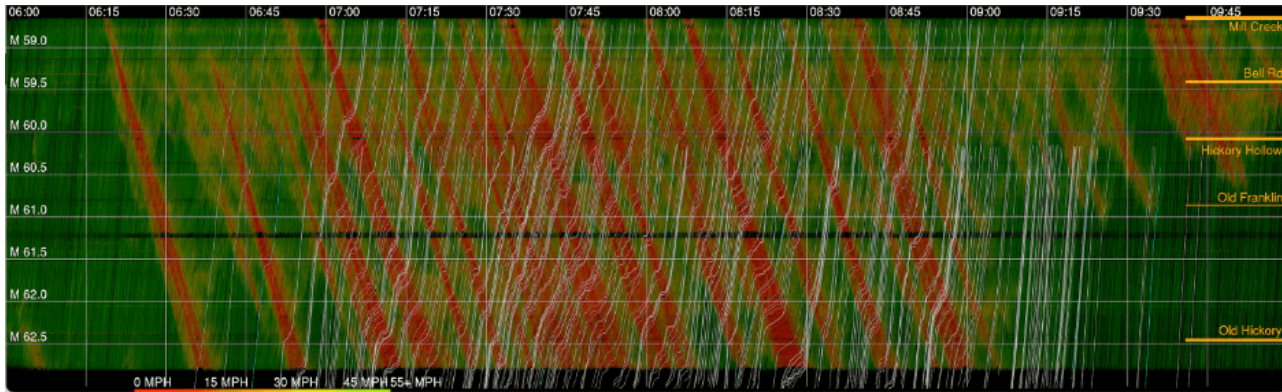detections occur) into a roadway coordinate system using a homography transformation calibrated every hour to each camera. 2) Post-processing algorithms [98] first stitch fragmented trajectories together using an online minimum cost network flow graph problem. Each trajectory is then subject to a reconciliation procedure to ensure feasible and smooth higher order dynamics (acceleration and jerk), formulated as a quadratic program.

Processing of raw video into vehicle trajectories by I-24 MOTION happens in two stages. 1) A computer vision pipeline [97], [99], [100] makes the initial vehicle detection and type classification from the raw video, including vehicle dimensions by using 3-D bounding boxes. Vehicle classes used by the object detection are: sedan, midsize, pickup, van, semi, truck, motorcycle. It tracks detected vehicles across adjacent camera views in the vehicle's direction of travel. The computer vision processing is distributed across ten servers with contiguous groups of cameras allocated to each. Vehicles are not tracked between servers, so a vehicle trajectory covering the length of the testbed consists of at least ten fragments; additional fragmentation may occur due to unavoidable occlusion (for example, by overpasses). The computer vision pipeline also converts image space coordinates (where vehicle detections occur) into a roadway coordinate system using

FIGURE 16 Time (horizontal axis) space (vertical axis) diagram generated by I-24 MOTION [53] and associated visualization library [96] during the MVT. Vehicle trajectories driving westbound (up) are colored based on the speed of vehicles (green: freeflow to red: congested). Experiment vehicle trajectories are overlaid in white.

a homography transformation calibrated every hour to each camera. 2) Post-processing algorithms first stitch fragmented trajectories together using an online minimum cost network flow graph problem [101]. Each trajectory is then subject to a reconciliation procedure [98] to ensure feasible and smooth higher order dynamics (acceleration and jerk), formulated as a quadratic program. A data visualization library [96] assists with interpreting the datasets generated by the system.

Due to the critical nature of data from the MVT, the I-24 MOTION system retained a secure backup of the experiment data and baseline traffic conditions. Raw I-24 MOTION imagery is not accessible outside of the I-24 MOTION administrators, per the testbed's privacy policy. However, this backup allows re-processing of the video and/or the raw vehicle detections from the computer vision pipeline as data generation pipeline advances. The current released version of the data has gone through re-processing rounds to address known errors and limitations, including using hourly re-calibrated homography transformations for camera perspective [100].

Experimental Design

The live traffic experiment took place on I-24 in November 2022. With 100 vehicles to deploy, over 150 drivers were hired from local colleges, security guards, delivery drivers, team members' relatives, and elsewhere. The drivers were trained to adhere to specific assigned routes and lanes and to activate our custom ACC system whenever driving in their assigned lanes. Safety features were implemented such that the controller would default to stock ACC behavior if engaged off of the highway.

The specific routes driven by the AVs were chosen to maximize the penetration rate where the I-24 MOTION system [53] would capture the effect on the surrounding commuters (bulk traffic). Initially, a single-loop route that

would have the AVs circulating from 7:30am to 9:30am was considered. It was determined that this would likely cause increased congestion on the Exit 57 off-ramp, leading to extended queuing and potential spillback onto I-24. For this reason, we divided our AVs into two groups with the two different routes partially overlapped on I-24. The portion of the highway where the experiment happened is displayed in Figure 18.

The AVs are released at 6:00am in order for all 100 to be deployed by the target time 7:30am. The drivers are instructed to repeat their driving routes for 2 hours, or sooner if they wanted a break. On a return trip, they exit the highway at Exit 60 (Hickory Hollow), to return to the Field Headquarters parking lot and return their car keys at a desk before leaving the lot. As AVs return to the lot for a break, other drivers (already on break) head down to a queue to come out to the lot when needed, following instructions from our "airline reservation board" (Figure 19). With this smoothly running rotation system, we were able to keep the 100 AVs on the routes during peak hours, while keeping track of their locations in real time (Figure 20). For more details about the routes, and logistical choices made for the experiment, driver training, the daily schedule, or the penetration rate estimates, see the article [102].

Data Release

As a part of this publication, we are releasing to the public a dataset from the MVT. This data release will include data collected by the AVs as well as the I-24 MOTION [54] system. See also [97], [100], [99] for other datasets generated by I-24 MOTION.

The data contains GPS and CAN data collected by the 100 AVs from all testing days. The longitudinal position along the highway is modified *post hoc* with an offset to account for systematic positioning error (for example,
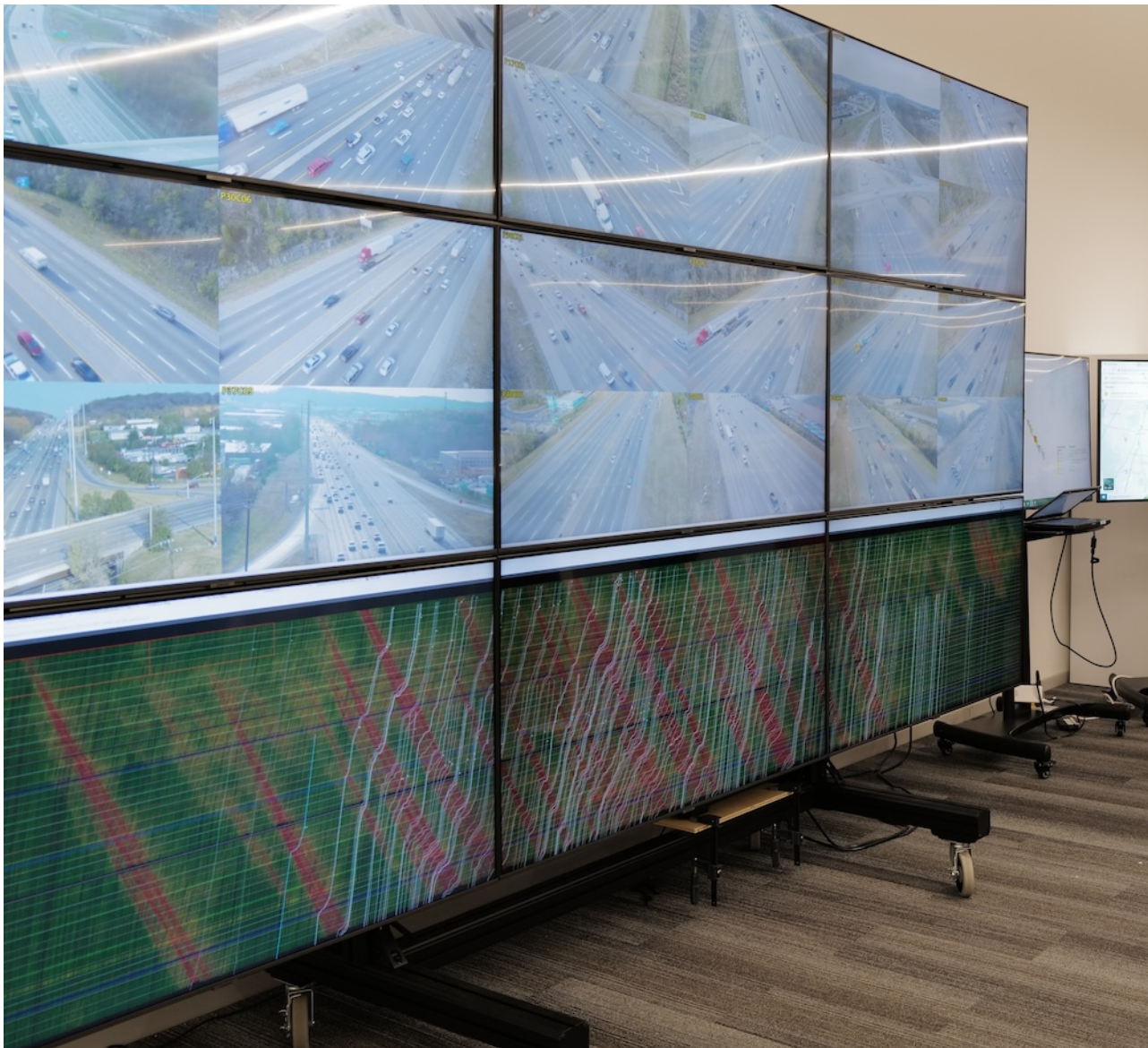
FIGURE 17 Visualization wall in the operation center. The display shows live video feeds from the I-24 MOTION cameras installed along the highway, as well as a time-space diagram generated from the I-24 MOTION and AV data (see Figure 16 for more details).

caused by placement of the GPS unit within the vehicle). The offset is determined by finding all likely matches in the I-24 MOTION dataset, and computing the median delta. All other data is provided unaltered except for interpolating onto a resampled, fixed 10Hz time grid. Lane identification is not provided; rather, the driver's assigned lane can reasonably be assumed to be the true lane when the controller is engaged.

The raw I-24 MOTION trajectory information is processed [98], [101] and then is appended with additional information. Speed and acceleration are inferred from the longitudinal position (which, as described previously, is processed to be sufficiently smooth for second-order derivatives). Road grade information is added via a parametric road grade map model. Fuel information, including various fuel rates and those for reference trajectories, are appended through the application of fuel rate models (see "Vehicle Energy Models" and [86]). Fuel rate models are selected with a one-to-one mapping of vehicle classes to fuel model classes. For each I-24 MOTION trajectory segment, a reference trajectory is constructed that matched the initial and terminal positions and speeds and minimize the functional $\max_t a(t) - \min_t a(t)$ subject to the additional constraint $v(t) \geq 0$. This functional is a simple vehicle-
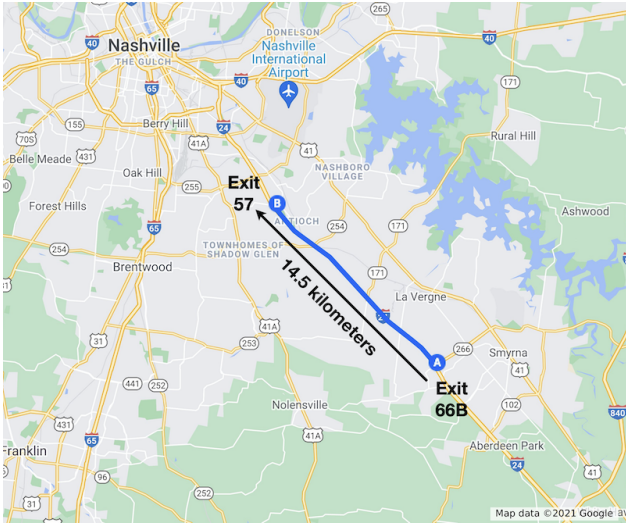
FIGURE 18 Portion of the Interstate 24 (I-24) highway where the experiment happened, near Nashville, Tennessee, US. The 14.5 kilometers (9 miles) portion approximately spans between Exit 57 and Exit 66B.

independent proxy for the more complicated energy models constructed in [86], motivated by the following arguments: (a) due to the super-linear growth of fuel rate with increasing acceleration, peak accelerations should be minimized; and (b) unnecessarily heavy braking results in wasted fuel, hence peak braking (which is $-\min_t a(t)$ for typical trajectories) should also be minimized. By Pontryagin's maximum principle, this generally results in piecewise-constant acceleration profiles with a jump at the trajectory's midpoint in time. Lastly, using the implicit lane and longitudinal matching with the AV dataset, we also supply relative distances and vehicle IDs for nearest upstream/downstream, engaged-or-not AVs to assist in subsequent analyses.

Further information on the structure of the data will be provided in the data documentation released with the data.

RESULTS

The experimental, observational, data collection, and data processing framework described above generated a large amount of data, capturing every single vehicle on a highway during the deployment of 100 controlled vehicles. Analogous to previous seminal traffic data sets, such as NGSIM [103], the new data are expected to inspire and enable many subsequent findings. To highlight this potential, we here present some key first findings and insights, based on an analysis of the data with a macroscopic perspective.

We construct macroscopically meaningful fields in time–space, most prominently a field that shows the energy (in)efficiency of traffic at large on the I-24 highway

segment. This is achieved by applying Edie's method [104] on boxes of size $h_t \times h_x$, where $h_t = 10s$ and $h_x = 200m$, to the I-24 MOTION trajectories to construct the following fields:

- » vehicle density $\rho(t,x)$, as the total vehicle time spent in each box, divided by the size of the box, $h_t \cdot h_x$;
- » flow rate $q(t,x)$, as the total distance traveled in each box, divided by $h_t \cdot h_x$; and
- » fuel rate density $f(t,x)$, as the total fuel consumed in each box, divided by $h_t \cdot h_x$.

From these fields, other meaningful fields are obtained, such as the bulk velocity field $u(t,x) = q(t,x)/\rho(t,x)$, the bulk fuel rate $\phi(t,x) = f(t,x)/\rho(t,x)$, and the bulk fuel consumption $\psi(t,x) = f(t,x)/q(t,x)$. The latter quantity $\psi(t,x)$, measurable for instance in grams per meter, represents the fuel demand per distance traveled of all vehicles in the $h_t \times h_x$ vicinity of the position (t,x). Figure 21 shows $\psi(t,x)$ for two experimental days: Wednesday November 16 and Thursday November 17, 2022. Each plot is overlaid with the trajectories of all control vehicles. This represents the first time that a complete time–space diagram of the energy inefficiency of traffic, based on accurate trajectories of *all* vehicles on the roadway, has been provided.

In the same spirit as the purely microscopic Figure 16, one can, for both days shown in Figure 21, clearly see the traffic waves traveling backwards along the highway, as well as the increased fuel consumption incurred in these waves. The figure also shows the increased fuel demand in the uphill segment between $x = 5km$ and $x = 6km$, and the reduced fuel demand in the downhill segment thereafter. The AV trajectories are colored red when the automated controller was activated, and white when the vehicle was under human control. The two shown test days were quite different in terms of the engagement rates of the controllers: on 11/16, the controllers were engaged 38% of all times, while on 11/17, the engagement rate went up to 78%. This difference was caused by a combination of increased driver comfort with the automation and a more reliable communication of traffic information to the vehicles on 11/17.

Given the low penetration rate of the control vehicles on the highway, it was not expected that they would completely smooth out all traffic waves—and the plots in Figure 21 confirm that expectation. However, the AVs may still have had some positive contribution on the energy efficiency of the flow at large. Whether that was in fact the case, we first note that the macroscopic plots like in Figure 21 allows for a targeted inspection of different regions of interest in time-space. For instance, on 11/16, there is a distinct region of high fuel inefficiency, around time 08:25:00 and location 5.0km–6.5km; and notably, this high-fuel region coincides with all AV controller being inactive. In contrast, on 11/17 such clusters of inactive AVs did not occur—and the fuel consumption map does not

FIGURE 19 Driver Assignment Status board. The display is located in the drivers' break room and lets drivers know when they should get ready to drive, and when they should go to the operation center lobby.

FIGURE 20 Vehicle Tracker System (middle screen). Co-author Jonathan Lee introduces the display to US DOE stakeholder, Heather Croteau. Located in the operation center, the realtime location, speed, direction, and control status of all deployed vehicles on the map.

exhibit similarly large high-fuel regions. Another notable anecdote occurs on 11/16 in the wave that goes through $x = 2.3$km at time 07:00:00. First two active AVs notably dampen the wave; then the wave keeps on growing while four inactive AVs run through it (at $x = 3$km); followed by several active AVs (around $x = 4$km) notably dampening the wave again.

Many more quantitative results and plots are available in the publications of the different teams that contributed and deployed a controller during the MegaVanderTest [82], [39], [41]. Since each team deployed their controllers in different manners, on different vehicles, or at different scales, the results are not easily comparable—which leads to interesting analyses from different points of views. Each team has detailed the process they went through to clean, parse, and extract relevant information from the data, in order to estimate their controller's impact in terms of energy savings and traffic smoothing. Continued effort to quantify our vehicles' overall impact is underway and planned for future publication(s).

CONCLUSION

This work describes the control architectures and implementations of a 100 automated vehicle deployment to improve traffic efficiency on a freeway using a small fraction of automated vehicles. It is the largest field experiment to use CAVs to regulate the overall traffic flow, and the deployment strategy enabled algorithms from diverse fields spanning model based control to reinforcement learning. The control strategy presented in this work was a hierarchical control approach in which the upper level speed planner provided target velocities to a lower level control law responsible for performance and safety. These algorithms were deployed in the largest field experiment

of its kind, on a heterogeneous vehicle fleet using low cost vehicular instrumentation. The data from the vehicles were combined with datasets generated from I-24 MOTION [53], [54], providing a large data resource for further study on the interaction of control vehicles on bulk traffic flow.

AUTHORS

The CIRCLES Consortium is led by:

Jonathan W. Lee (jonny5@berkeley.edu) Jonathan Lee received the B.S. degree in engineering physics from the University of California, Berkeley and M.S. and Ph.D. degrees in mechanical engineering from Rice University. At Sandia National Laboratories (2011–13), he completed his postdoctoral appointment studying electrical and material
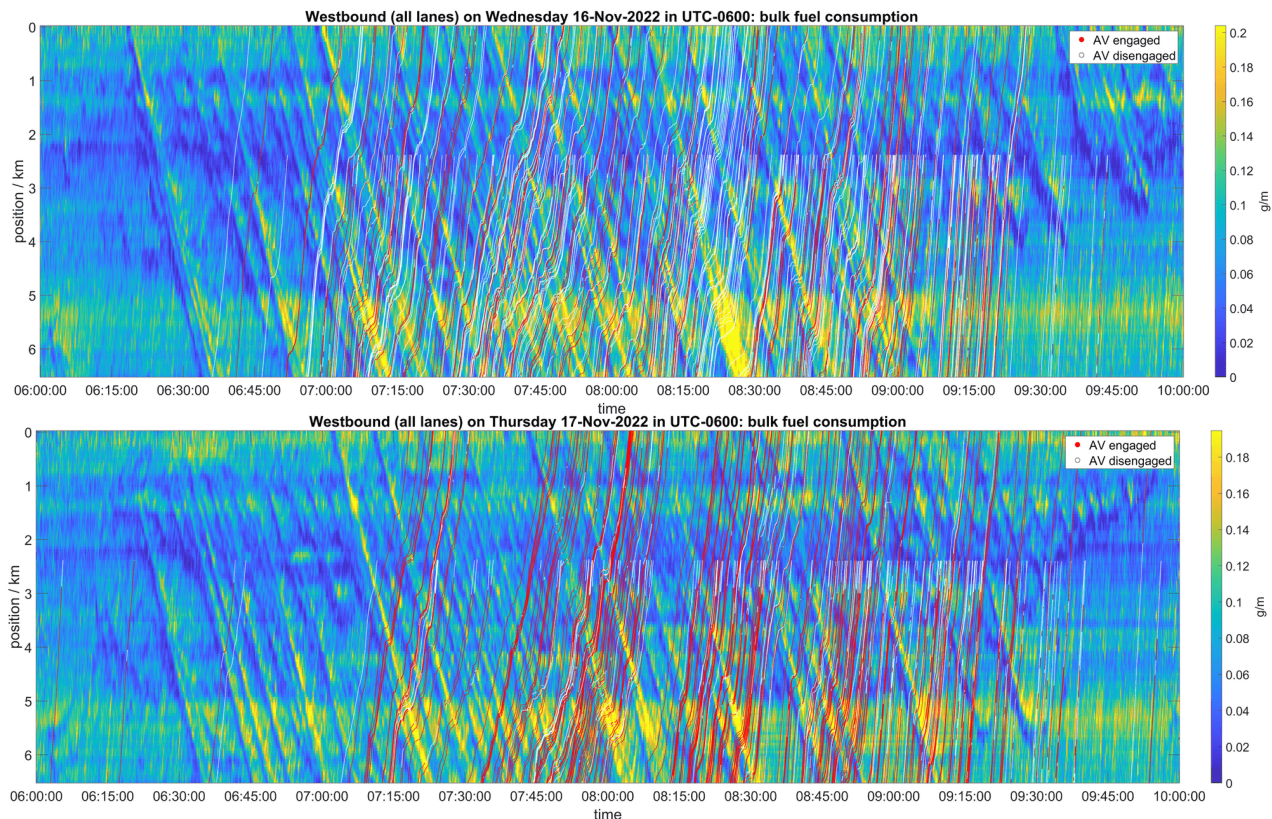
FIGURE 21 Bulk fuel consumption heatmap in time (horizontal axis) and space (vertical axis), based on I-24 MOTION [53] of all vehicles (aggregated over all lanes) driving in the Westbound direction with AV trajectories overlaid (white: controller not engaged; red: controller engaged). Shown are November 16, 2022 (top) and November 17, 2022 (bottom).

properties via molecular dynamics simulations. He subsequently served as a senior data scientist and product manager on various teams at Uber Technologies, Inc. (2014-2019). Since 2019, he has served as an engineering manager at the University of California, Berkeley and the program manager and Chief Engineer of CIRCLES.

Daniel B. Work (dan.work@vanderbilt.edu) Daniel Work is a Professor of Civil and Environmental Engineering and the Institute for Software Integrated Systems, at Vanderbilt University. His interests are in transportation cyber-physical systems.

Benjamin Seibold (seibold@temple.edu) Benjamin Seibold is a Professor of Mathematics and Physics, and the Director of the Center for Computational Mathematics and Modeling, at Temple University. His research areas, funded by NSF, DOE, DAC, USACE, USDA, and PDA, are computational mathematics (high-order methods for differential equations, CFD, molecular dynamics) and applied mathematics and modeling (traffic flow, invasive species, many-agent systems, radiative transfer).

Jonathan Sprinkle (jonathan.sprinkle@vanderbilt.edu) Jonathan Sprinkle is a Professor of Computer Science at Vanderbilt University since 2021. Prior to joining Vanderbilt he was the Litton Industries John M. Leonis Distinguished Associate Professor of Electrical and Computer Engineering at the University of Arizona, and the Interim Director of the Transportation Research Institute. From 2017-2019 he served as a Program Director in Cyber-Physical Systems and Smart & Connected Communities at the National Science Foundation in the CISE Directorate.

Benedetto Piccoli (piccoli@camden.rutgers.edu) Benedetto Piccoli is University Professor and the Joseph and Loretta Lopez Chair Professor of Mathematics at Rutgers University - Camden. He also served as Vice Chancellor for Research. His research interests span various areas of applied mathematics, including control theory, traffic flow on networks, crowd dynamics, math finance and application to autonomous driving, population health and bio-medical systems. He is author of more than 300 research papers and 7 books and is the founding editor of Networks and Heterogeneous Media. Piccoli is the recipient of the 2009 Fubini Prize, Plenary speaker at ICIAM 2011, and 2012 inaugural Fellow of American Mathematical Society.

Maria Laura Delle Monache (mldellemonache@berkeley.edu) Maria Laura Delle Monache is an Assistant Professor in the Department of Civil and Environmental Engineering at the University of

California, Berkeley. Dr. Delle Monache's research lies at the intersection of transportation engineering, mathematics, and control and focuses on modeling and control of mixed autonomy large-scale traffic systems.

Alexandre M. Bayen (bayen@berkeley.edu) Alexandre Bayen is the Associate Provost for Moffett Field Program Development at UC Berkeley, and the Liao-Cho Professor of Engineering at UC Berkeley. He is a Professor of Electrical Engineering and Computer Science and of Civil and Environmental Engineering (courtesy). He is a visiting Professor at Google. He is also a Faculty Scientist in Mechanical Engineering, at the Lawrence Berkeley National Laboratory (LBNL). From 2014 - 2021, he served as the Director of the Institute of Transportation Studies at UC Berkeley (ITS).

REFERENCES

[1] W. B. Stevens, "The automated highway system program: A progress report," *IFAC Proceedings Volumes*, vol. 29, no. 1, pp. 8180–8188, 1996.

[2] H.-S. Tan, R. Rajamani, and W.-B. Zhang, "Demonstration of an automated highway platoon system," in *Proceedings of the 1998 American control conference. ACC (IEEE Cat. No. 98CH36207)*, vol. 3. IEEE, 1998, pp. 1823–1827.

[3] S. E. Shladover, C. Nowakowski, X.-Y. Lu, and R. Ferlis, "Cooperative adaptive cruise control: Definitions and operating concepts," *Transportation Research Record*, vol. 2489, no. 1, pp. 145–152, 2015.

[4] D. Swaroop, J. K. Hedrick, C. Chien, and P. Ioannou, "A comparision of spacing and headway control laws for automatically controlled vehicles1," *Vehicle system dynamics*, vol. 23, no. 1, pp. 597–625, 1994.

[5] P. A. Ioannou and C.-C. Chien, "Autonomous intelligent cruise control," *IEEE Transactions on Vehicular technology*, vol. 42, no. 4, pp. 657–672, 1993.

[6] A. Girault, "A hybrid controller for autonomous vehicles driving on automated highways," *Transportation Research Part C: Emerging Technologies*, vol. 12, no. 6, pp. 421–452, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X04000208

[7] M. Buehler, K. Iagnemma, and S. Singh, *The 2005 DARPA grand challenge: the great robot race*. springer, 2007, vol. 36.

[8] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[9] L. Xiao and F. Gao, "A comprehensive review of the development of adaptive cruise control systems," *Vehicle system dynamics*, vol. 48, no. 10, pp. 1167–1192, 2010.

[10] G. Gunter, D. Gloudemans, R. E. Stern, S. McQuade, R. Bhadani, M. Bunting, M. L. Delle Monache, R. Lysecky, B. Seibold, J. Sprinkle *et al.*, "Are commercially implemented adaptive cruise control systems string stable?" *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 6992–7003, 2020.

[11] S. E. Shladover, C. Nowakowski, X.-Y. Lu, and R. Hoogendoorn, "Using cooperative adaptive cruise control (cacc) to form high-performance vehicle streams," 2014.

[12] A. Dávila and M. Nombela, *Sartre-safe road trains for the environment reducing fuel consumption through lower aerodynamic drag coefficient*. SAE International Warrendale, PA, USA, 2011.

[13] P. Jootel, "Safe road trains for the environment (sartre) final report," *tech. rep., Technical Report for European Commission under the Framework 7 Programme Project 233683)*, 2012.

[14] S. Tsugawa, S. Kato, and K. Aoki, "An automated truck platoon for energy saving," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 4109–4114.

[15] J. Ploeg, S. Shladover, H. Nijmeijer, and N. van de Wouw, "Introduction to the special issue on the 2011 grand cooperative driving challenge," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 989–993, 2012.

[16] R. Kianfar, B. Augusto, A. Ebadighajari, U. Hakeem, J. Nilsson, A. Raza, R. S. Tabar, N. V. Irukulapati, C. Englund, P. Falcone *et al.*, "Design and experimental validation of a cooperative driving system in the grand cooperative driving challenge," *IEEE transactions on intelligent transportation systems*, vol. 13, no. 3, pp. 994–1007, 2012.

[17] C. Englund, L. Chen, J. Ploeg, E. Semsar-Kazerooni, A. Voronov, H. H. Bengtsson, and J. Didoff, "The grand cooperative driving challenge 2016: boosting the introduction of cooperative automated vehicles," *IEEE Wireless Communications*, vol. 23, no. 4, pp. 146–152, 2016.

[18] S. Shladover, X.-Y. Lu, S. Yang, H. Ramezani, J. Spring, C. Nowakowski, and D. Nelson, "Cooperative adaptive cruise control (cacc) for partially automated truck platooning," 2018.

[19] D. M. Willemsen, A. J. Schmeitz, and E. Mascalchi, "Eu ensemble project: Specification of an interoperable solution for a support function for platooning," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[20] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Transactions on intelligent transportation systems*, vol. 15, no. 1, pp. 296–305, 2013.

[21] V. Milanés and S. E. Shladover, "Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 285–300, 2014.

[22] S. E. Shladover, D. Su, and X.-Y. Lu, "Impacts of cooperative adaptive cruise control on freeway traffic flow," *Transportation Research Record*, vol. 2324, no. 1, pp. 63–70, 2012.

[23] C. Nowakowski, J. O'Connell, S. E. Shladover, and D. Cody, "Cooperative adaptive cruise control: Driver acceptance of following gap settings less than one second," in *Proceedings of the human factors and ergonomics society annual meeting*, vol. 54, no. 24. SAGE Publications Sage CA: Los Angeles, CA, 2010, pp. 2033–2037.

[24] A. Al Alam, A. Gattami, and K. H. Johansson, "An experimental study on the fuel reduction potential of heavy duty vehicle platooning," in *13th international IEEE conference on intelligent transportation systems*. IEEE, 2010, pp. 306–311.

[25] S. W. Smith, Y. Kim, J. Guanetti, R. Li, R. Firoozi, B. Wootton, A. A. Kurzhanskiy, F. Borrelli, R. Horowitz, and M. Arcak, "Improving urban traffic throughput with vehicle platooning: Theory and experiments," *IEEE Access*, vol. 8, pp. 141 208–141 223, 2020.

[26] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S. 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, mar 2008. [Online]. Available: https://dx.doi.org/10.1088/1367-2630/10/3/033001

[27] R. E. Stern, S. Cui, M. L. D. Monache, R. Bhadani, M. Bunting, M. Churchill, N. Hamilton, H. Pohlmann, F. Wu, B. Piccoli, B. Seibold, J. Sprinkle, and D. B. Work, "Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments," *Transportation Research Part C*, vol. 89, 04/2018 2018.

[28] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: Architecture and benchmarking for reinforcement learning in traffic control," *arXiv preprint arXiv:1710.05465*, p. 10, 2017.

[29] C. Wu, A. Kreidieh, E. Vinitsky, and A. M. Bayen, "Emergent behaviors in mixed-autonomy traffic," in *Conference on Robot Learning (CoRL)*, 2017. [Online]. Available: http://proceedings.mlr.press/v78/wu17a.html

[30] E. Vinitsky, A. Kreidieh, L. Le Flem, N. Kheterpal, K. Jang, C. Wu, F. Wu, R. Liaw, E. Liang, and A. M. Bayen, "Benchmarks for reinforcement learning in mixed-autonomy traffic," in *Conference on robot learning*. PMLR, 2018, pp. 399–409. [Online]. Available: https://proceedings.mlr.press/v87/vinitsky18a.html

[31] K. Jang, E. Vinitsky, B. Chalaki, B. Remer, L. Beaver, A. A. Malikopoulos, and A. Bayen, "Simulation to scaled city: zero-shot policy transfer for traffic control via autonomous vehicles," in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, 2019, pp. 291–300.

[32] J. Cui, W. Macke, H. Yedidsion, A. Goyal, D. Urieli, and P. Stone, "Scalable multiagent driving policies for reducing traffic congestion," in *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, May 2021.

[33] Y. Zhang, W. Macke, J. Cui, D. Urieli, and P. Stone, "Learning a robust

multiagent driving policy for traffic congestion reduction," 2023.

[34] C. Wu, A. M. .Bayen, and A. Mehta, "Stabilizing traffic with autonomous vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6012–6018.

[35] E. Vinitsky, A. Kreidieh, L. L. Flem, N. Kheterpal, K. Jang, C. Wu, F. Wu, R. Liaw, E. Liang, and A. M. Bayen, "Benchmarks for reinforcement learning in mixed-autonomy traffic," in *Proceedings of The 2nd Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Billard, A. Dragan, J. Peters, and J. Morimoto, Eds., vol. 87. PMLR, 29–31 Oct 2018, pp. 399–409. [Online]. Available: https://proceedings.mlr.press/v87/vinitsky18a.html

[36] A. M. Bayen, J. W. Lee, B. Piccoli, B. Seibold, J. M. Sprinkle, and D. B. Work, "CIRCLES: Congestion Impacts Reduction via CAV-in-the-loop Lagrangian Energy Smoothing." Presented at the 2020 Vehicle Technologies Office Annual Merit Review, Washington, DC (virtual), 2020.

[37] ——, "CIRCLES: Congestion Impacts Reduction via CAV-in-the-loop Lagrangian Energy Smoothing." Presented at the 2021 Vehicle Technologies Office Annual Merit Review, Washington, DC (virtual), 2021.

[38] J. M. Sprinkle, A. M. Bayen, J. W. Lee, B. Piccoli, B. Seibold, and D. B. Work, "CIRCLES: Congestion Impacts Reduction via CAV-in-the-loop Lagrangian Energy Smoothing." Presented at the 2022 Vehicle Technologies Office Annual Merit Review, Washington, DC, 2022.

[39] H. Wang, Z. Fu, J. Lee, A. Alanqary, H. N. Zinat Matin, D. Urieli, S. Hornstein, A. R. Kreidieh, R. Chekroun, W. A. Richardson, D. Work, B. Piccoli, B. Seibold, J. Sprinkle, A. M. Bayen, and M. L. Delle Monache, "Centralized av speed planner: Hierarchical framework for lagrangian variable speed limit in mixed autonomy traffic," *IEEE Control Systems Magazine*, 2024.

[40] N. Kardous, A. Hayat, S. T. McQuade, X. Gong, S. Truong, T. Mezair, P. Arnold, R. Delorenzo, A. Bayen, and B. Piccoli, "A rigorous multi-population multi-lane hybrid traffic model for dissipation of waves via autonomous vehicles," *The European Physical Journal Special Topics*, vol. 231, no. 9, pp. 1689–1700, 2022. [Online]. Available: https://doi.org/10.1140/epjs/s11734-022-00580-z

[41] K. Jang, N. Lichtlé, E. Vinitsky, A. Shah, M. Bunting, M. Nice, B. Piccoli, B. Seibold, D. B. Work, M. L. Delle Monache, J. Sprinkle, J. W. Lee, and A. M. Bayen, "Large-scale wave dampening via single-agent reinforcement learning algorithms: Controlling and deploying 100 vehicles on a highway with learning-based algorithms," *IEEE Control Systems Magazine*, 2024.

[42] A. Bacciotti and L. Rosier, *Liapunov functions and stability in control theory*. Springer Science & Business Media, 2005.

[43] K. Yosida, *Functional analysis*. Springer Science & Business Media, 2012.

[44] A. Bressan and B. Piccoli, *Introduction to the mathematical theory of control*. American institute of mathematical sciences Springfield, 2007, vol. 1.

[45] E. Casas and M. Mateos, "Optimal control of partial differential equations," *Computational Mathematics, Numerical Analysis and Applications: Lecture Notes of the XVII'Jacques-Louis Lions' Spanish-French School*, pp. 3–59, 2017.

[46] L. N. Trefethen, *Approximation Theory and Approximation Practice, Extended Edition*. SIAM, 2019.

[47] F. Golse, "On the dynamics of large particle systems in the mean field limit," *Macroscopic and large scale phenomena: coarse graining, mean field limits and ergodicity*, pp. 1–144, 2016.

[48] P. Baiti, P. G. LeFloch, and B. Piccoli, "Uniqueness of classical and non-classical solutions for nonlinear hyperbolic systems," *Journal of Differential Equations*, vol. 172, no. 1, pp. 59–82, 2001.

[49] F. Wu and A. M. Bayen, "A hierarchical mpc approach to car-following via linearly constrained quadratic programming," *IEEE Control Systems Letters*, vol. 7, pp. 532–537, 2023.

[50] Z. Fu, A. R. Kreidieh, H. Wang, J. W. Lee, M. L. Delle Monache, and A. M. Bayen, "Cooperative driving for speed harmonization in mixed-traffic environments," in *2023 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2023, pp. 1–8.

[51] M. Garavello, K. Han, B. Piccoli *et al.*, *Models for vehicular traffic on networks*. American Institute of Mathematical Sciences (AIMS), Springfield, MO, 2016, vol. 9.

[52] X.-Y. Lu and S. E. Shladover, "Review of variable speed limits and advisories: Theory, algorithms, and practice," *Transportation research record*, vol. 2423, no. 1, pp. 15–23, 2014.

[53] D. Gloudemans, Y. Wang, J. Ji, G. Zachar, W. Barbour, E. Hall, M. Cebelak, L. Smith, and D. B. Work, "I-24 MOTION: An instrument for freeway traffic science," *Transportation Research Part C: Emerging Technologies*, vol. 155, p. 104311, 2023.

[54] D. Gloudemans, W. Barbour, N. Gloudemans, M. Neuendorf, B. Freeze, S. ElSaid, and D. B. Work, "Interstate-24 motion: Closing the loop on smart mobility," in *2020 IEEE Workshop on Design Automation for CPS and IoT (DESTION)*. IEEE, 2020, pp. 49–55.

[55] M. R. Flynn, A. R. Kasimov, J.-C. Nave, R. R. Rosales, and B. Seibold, "Self-sustained nonlinear waves in traffic flow," *Phys. Rev. E*, vol. 79, no. 5, p. 056113, 2009.

[56] V. Giammarino, S. Baldi, P. Frasca, and M. L. Delle Monache, "Traffic flow on a ring with a single autonomous vehicle: An interconnected stability perspective," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4998–5008, 2020.

[57] A. Bayen, M. L. Delle Monache, M. Garavello, P. Goatin, and B. Piccoli, *Control problems for conservation laws with traffic applications: modeling, analysis, and numerical methods*. Springer Nature, 2022.

[58] A. Alanqary, X. Gong, A. Keimer, B. Seibold, B. Piccoli, and A. Bayen, "Optimal control of autonomous vehicles for flow smoothing in mixed autonomy traffic," in *(to appear) 2023 IEEE 62nd Conference on Decision and Control (CDC)*. IEEE, 2023.

[59] R. Stern, D. Work, S. Cui, H. Pohlmann, B. Seibold, M. L. D. Monache, B. Piccoli, and J. Sprinkle, "Stabilizing traffic with a single autonomous vehicle," in *7th International Conference on Cyber-Physical Systems (ICCPS)*. ACM/IEEE, 2016.

[60] *Dissipation of Emergent Traffic Waves in Stop-and-Go Traffic Using a Supervisory Controller*, vol. 57. Fontainbleau, Miami Beach, USA: IEEE, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8619700

[61] A. Hayat, X. Gong, J. Lee, S. Truong, S. McQuade, N. Kardous, A. Keimer, Y. You, S. Albeaik, E. Vinitsky, P. Arnold, M. L. Delle Monache, A. Bayen, B. Seibold, J. Sprinkle, D. Work, and B. Piccoli, *A Holistic Approach to the Energy-Efficient Smoothing of Traffic via Autonomous Vehicles*. Cham: Springer International Publishing, 2022, pp. 285–316.

[62] J. Baillieul and P. J. Antsaklis, "Control and communication challenges in networked real-time systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 9–28, 2007.

[63] S. K. Vesal Ahsani, Mostafa Amin-Naseri and A. Sharma, "Quantitative analysis of probe data characteristics: Coverage, speed bias and congestion detection precision," *Journal of Intelligent Transportation Systems*, vol. 23, no. 2, pp. 103–119, 2019. [Online]. Available: https://doi.org/10.1080/15472450.2018.1502667

[64] S. Kim and B. Coifman, "Comparing inrix speed data against concurrent loop detector stations over several months," *Transportation Research Part C: Emerging Technologies*, vol. 49, pp. 59–72, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X14002940

[65] M. Bunting, R. Bhadani, and J. Sprinkle, "Libpanda: A high performance library for vehicle data collection," in *Proceedings of the Workshop on Data-Driven and Intelligent Cyber-Physical Systems*, 2021, pp. 32–40. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3459609.3460529

[66] S. Elmadani, M. Nice, M. Bunting, J. Sprinkle, and R. Bhadani, "From CAN to ROS: A monitoring and data recording bridge," in *Proceedings of the Workshop on Data-Driven and Intelligent Cyber-Physical Systems*, 2021, pp. 17–21. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3459609.3460531

[67] N. Lichtlé, E. Vinitsky, M. Nice, B. Seibold, D. Work, and A. M. Bayen, "Deploying traffic smoothing cruise controllers learned from trajectory data," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2884–2890. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9811912

[68] R. Bhadani, M. Bunting, M. Nice, F. Wu, A. Hayat, J. W. Lee, A. Bayen, B. Piccoli, B. Seibold, D. Work *et al.*, "Approaches for synthesis and deployment of controller models on automated vehicles for car-following in mixed autonomy," in *Proceedings of Cyber-Physical Systems and Internet of Things Week 2023*, 2023, pp. 158–163.

[69] M. Nice, M. Bunting, D. Work, and J. Sprinkle, "Middleware for a heterogeneous CAV fleet," in *Proceedings of Cyber-Physical Systems and Internet of Things Week 2023*, 2023, pp. 86–91. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3459609.3460531

[70] M. Nice, M. Bunting, G. Zachar, R. Bhadani, P. Ngo, J. Lee, A. Bayen, D. Work, and J. Sprinkle, "Parameter estimation for decoding sensor signals," in *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, 2023, pp. 253–255.

[71] M. Bunting, R. Bhadani, M. Nice, S. Elmadani, and J. Sprinkle, "Data from the development evolution of a vehicle for custom control," in *2022 2nd Workshop on Data-Driven and Intelligent Cyber-Physical Systems for Smart Cities Workshop (DI-CPS)*. IEEE, 2022, pp. 40–46.

[72] F.-C. Chou, M. Gibson, R. Bhadani, A. M. Bayen, and J. Sprinkle, "Reachability analysis for followerstopper: Safety analysis and experimental results," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8607–8613.

[73] N. Lichtlé, K. Jang, A. Shah, E. Vinitsky, J. W. Lee, and A. M. Bayen, "Traffic smoothing controllers for autonomous vehicles using deep reinforcement learning and real-world trajectory data," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2023, pp. 4346–4351.

[74] G. Gunter, M. Nice, M. Bunting, J. Sprinkle, and D. B. Work, "Experimental testing of a control barrier function on an automated vehicle in live multi-lane traffic," in *2022 2nd Workshop on Data-Driven and Intelligent Cyber-Physical Systems for Smart Cities Workshop (DI-CPS)*. IEEE, 2022, pp. 31–35.

[75] 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.

[76] D. C. Gazis, R. Herman, and R. W. Rothery, "Nonlinear follow-the-leader models of traffic flow," *Operations research*, vol. 9, no. 4, pp. 545–567, 1961.

[77] A. Hayat, B. Piccoli, and S. Truong, "Dissipation of traffic jams using a single autonomous vehicle on a ring road," *SIAM Journal on Applied Mathematics*, vol. 83, no. 3, pp. 909–937, 2023.

[78] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[79] A. Richardson, M. W. Nice, M. Bunting, J. Lee, R. Bhadani, D. Work, and J. Sprinkle, "Analysis of a runtime data sharing architecture over lte for a heterogeneous cav fleet," in *Proceedings of Cyber-Physical Systems and Internet of Things Week 2023*, 2023, pp. 164–169.

[80] S. Albeaik, A. Bayen, M. T. Chiri, X. Gong, A. Hayat, N. Kardous, A. Keimer, S. T. McQuade, B. Piccoli, and Y. You, "Limitations and improvements of the intelligent driver model (idm)," *SIAM Journal on Applied Dynamical Systems*, vol. 21, no. 3, pp. 1862–1892, 2022.

[81] X. Gong and A. Keimer, "On the well-posedness of the "bando-follow the leader" car following model and a time-delayed version," *Networks and Heterogeneous Media*, vol. 18, no. 2, pp. 775–798, 2023. [Online]. Available: https://doi.org/10.3934/nhm.2023033

[82] A. a. Hayat, "Traffic smoothing with a single vehicle: Dissipating stop-and-go waves with a single controlled vehicle in dense traffic: Experimental evidence," *IEEE Control Systems Magazine*, 2024.

[83] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[84] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[85] A. N. Laboratory, "Autonomie," https://www.anl.gov/taps/autonomie-vehicle-system-simulation-tool, 2023.

[86] N. Khoudari, S. Almatrudi, R. Ramadan, J. Carpio, M. Yao, K. Butts, A. M. Bayen, J. W. Lee, and B. Seibold, "A systematic model reduction pipeline from detailed vehicle energy dynamics to simple models with desirable physics-like properties," *IEEE Control Systems Magazine*, 2024.

[87] J. W. Lee, G. Gunter, R. Ramadan, S. Almatrudi, P. Arnold, J. Aquino, W. Barbour, R. Bhadani, J. Carpio, F.-C. Chou, M. Gibson, X. Gong, A. Hayat, N. Khoudari, A. R. Kreidieh, M. Kumar, N. Lichtlé, S. McQuade, B. Nguyen, M. Ross, S. Truong, E. Vinitsky, Y. Zhao, J. Sprinkle, B. Piccoli, A. M. Bayen, D. B. Work, and B. Seibold, "Integrated framework of vehicle dynamics, instabilities, energy models, and sparse flow smoothing controllers," in *Proceedings of the Workshop on Data-Driven and Intelligent Cyber-Physical Systems*, ser. DI-CPS'21. New York, NY, USA: Association for Computing Machinery, 2021, p. 41–47. [Online]. Available: https://doi.org/10.1145/3459609.3460530

[88] M. Brackstone and M. McDonald, "Car-following: a historical review," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 2, no. 4, pp. 181–196, 1999.

[89] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[90] [Online]. Available: https://comma.ai/

[91] [Online]. Available: https://intrepidcs.com

[92] R. Bhadani, M. Bunting, M. Nice, N. M. Tran, S. Elmadani, D. Work, and J. Sprinkle, "Strym: A python package for real-time can data logging, analysis and visualization to work with usb-can interface," in *2022 2nd Workshop on Data-Driven and Intelligent Cyber-Physical Systems for Smart Cities Workshop (DI-CPS)*. IEEE, 2022, pp. 14–23. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9805366

[93] P. Ngo and J. Sprinkle, "Lightweight lstm for can signal decoding," in *Proceedings of the Workshop on Data-Driven and Intelligent Cyber-Physical Systems*, 2021, pp. 27–31.

[94] P. Ngo, J. Sprinkle, and R. Bhadani, "Canclassify: Automated decoding and labeling of can bus signals," *Berkeley Master's Thesis*, vol. 1, no. 1, pp. 1–8, 2022.

[95] R. Bhadani and J. Sprinkle, "Prototyping vehicle control applications using the cat vehicle simulator," *arXiv preprint arXiv:2301.04574*, 2023.

[96] G. Zachár, "Visualization of large-scale trajectory datasets," ser. CPS-IoT Week '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 152–157. [Online]. Available: https://doi.org/10.1145/3576914.3587710

[97] D. Gloudemans and D. B. Work, "Vehicle tracking with crop-based detection," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2021, pp. 312–319.

[98] Y. Wang, D. Gloudemans, Z. N. Teoh, L. Liu, G. Zachár, W. Barbour, and D. Work, "Automatic vehicle trajectory data reconstruction at scale," *arXiv preprint arXiv:2212.07907*, 2022.

[99] D. Gloudemans, Y. Wang, G. Gumm, W. Barbour, and D. B. Work, "The interstate-24 3d dataset: a new benchmark for 3d multi-camera vehicle tracking," *arXiv preprint arXiv:2308.14833*, 2023.

[100] D. Gloudemans, G. Zachár, Y. Wang, J. Ji, M. Nice, M. Bunting, W. Barbour, J. Sprinkle, B. Piccoli, M. L. Monache, A. Bayen, B. Seibold, and D. B. Work, "So you think you can track?" *arXiv preprint arXiv:2309.07268*, 2023.

[101] Y. Wang, J. Ji, W. Barbour, and D. B. Work, "Online min cost circulation for multi-object-tracking on fragments," in *2023 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2023, to appear.

[102] M. Ameli, S. McQuade, J. W. Lee, R. DeLorenzo, M. L. Delle Monache, S. Hornstein, M. Mahmood, M. Mahmood, M. Nice, M. Bunting, D. Timsit, R. Wagner, R. Xu, B. Seibold, D. B. Work, J. Sprinkle, B. Piccoli, and A. M. Bayen, "Live traffic control experiment via automated vehicles," *IEEE Control Systems Magazine*, 2024.

[103] US Department of Transportation Federal Highway Administration, "Next Generation Simulation (NGSIM)," Website, http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm.

[104] L. C. Edie, "Car-following and steady-state theory for noncongested traffic," *Operations research*, vol. 9, no. 1, pp. 66–76, 1961.