

Multilane Automated Driving with Optimal Control and Mixed Integer Programming

R. Austin Dollar, *Student Member, IEEE*, and Ardalan Vahidi, *Senior Member, IEEE*

Abstract—Road vehicle lane changes often initiate traffic disturbances and can therefore impact road networks’ energy and time efficiency. Furthermore, unexpected changes in traffic conditions may also render lane changes counterproductive for the lane-changing vehicle. Vehicle-to-vehicle connectivity combined with anticipative control could address these challenges via improved lane change decisions by automated vehicles. In a move toward this objective, receding horizon control cast as a mixed integer quadratic program is used to plan lane changing and acceleration in a coupled optimization. A long-term pacing module, based on Pontryagin’s Minimum Principle from optimal control theory, sets terminal and input references for receding horizon control to target a user’s expected travel time. To remove nonlinear vehicle dynamics from the receding horizon controller, lane change commands are passed to a pure pursuit steering module whose response is approximated by a 2nd order linear model. Comparison against a rule-based reactive algorithm in arterial and highway scenarios shows an 8.9 % to 13.7 % reduction in energy consumption and a 5.2 % to 10.3 % reduction in travel time, along with navigational improvements.

I. INTRODUCTION

LESS-than-ideal traffic flow due to congestion and suboptimal driving practices saps the time and energy efficiency of today’s road networks. Small speed perturbations can grow, causing significant slowdowns upstream. Such disturbances may originate in the longitudinal direction only or result from lateral cut-ins [1]. Multi-lane roads allow vehicles to avoid conflicts in these cases, which could improve traffic flow. However, conventional vehicles may not take full advantage of this capability and their lane changes sometimes cause capacity drop [2]. Increasing prevalence of connected and automated vehicles (CAVs) could improve lane decisions via higher-quality information and optimal control.

Efficient motion control has been more thoroughly researched in longitudinal-only scenarios, perhaps because the problem naturally fits well-established control techniques. Within a single lane, the drivable region is convex and conventional model predictive control (MPC) using a standard quadratic program (QP) applies. In contrast, multi-lane vehicle guidance involves a non-convex drivable region that opens the possibility of local minima and multiple global optima.

The complexity of this problem has led some researchers to data-driven machine learning methods. In 2007, Ngai and

Yung [3] proposed reinforcement learning for vehicle overtaking by targeting maximum speed and heading toward the goal in the reward function. More recently, Wang and Chan [4] presented a reinforcement learning algorithm for highway merging. In the related field of off-road navigation, Pfeiffer et al. [5] experimentally applied end-to-end learning to avoid obstacles and reach a goal in a non-convex drivable region.

Model-based optimization has also been proposed to address lane changing. Some of these efforts have involved integer programming, which has also been applied to the related problem of scheduling vehicle arrivals at intersections [6]. Mukai and Kawabe [7] proposed an explicit hybrid MPC based on mixed integer linear programming (MILP) for two-lane roads. More recently, Du et al. [8] considered several lanes using a mixed logical dynamical (MLD) model where the vehicle was assumed to fully change lanes between each discretization step. Dollar and Vahidi [9] combined integer-valued lane decision variables with a continuous lateral motion model and quadratic objective to formulate a mixed integer quadratic program (MIQP). Energy results in a two-lane passing scenario showed an 8 % reduction in energy consumption and 6 % reduction in travel time compared to a rule-based algorithm. All of these approaches used a receding-horizon MPC alone, implying that the overall trip is not directly optimized.

Another line of research does optimize over the whole trip, that is, the process of moving the vehicle to the next stopping point. Pontryagin’s Minimum Principle (PMP) and Dynamic Programming (DP) are commonly used in such systems. For example, Hu et al. [10] compared various PMP approaches to hybrid vehicle driving and Abbas et al. [11] combined both PMP and DP to minimize energy consumption in electric vehicles. However, like most similar systems the impact of obstacles including traffic is not addressed. Mensing et al. [12] point this out and include traffic constraints using DP. However, only car-following is considered. Wang et al. [13] [14] proposed a hierarchical scheme where a finite state machine executes either optimal or rule-based transitions between maneuvers e.g. following or lane-changing.

The system described in this paper takes a distinct hierarchical approach featuring one unified mode. Variational optimal control provides a longitudinal-only trajectory ending at the next stopping point, assuming no conflicts with obstacles. In reality, surrounding traffic and stationary obstacles may render this solution infeasible. Therefore, it is not used as the final control input but rather as a reference for a receding horizon controller (RHC) that handles complex obstacle avoidance and lane decisions. The reference thus alleviates the RHC’s short-sightedness in speed planning.

The following section elaborates on the proposed architec-

Manuscript submitted June 14, 2019. This research was supported by the U.S. Dept. of Energy Vehicle Technologies Office under project no. DE-EE0008232.

R. A. Dollar is with the Dept. of Mechanical Engineering, Clemson University, Clemson, SC, 29634-0921 USA e-mail: rdollar@clemson.edu.

A. Vahidi is with the Dept. of Mechanical Engineering, Clemson University, Clemson, SC, 29634-0921 USA e-mail: avahidi@clemson.edu.

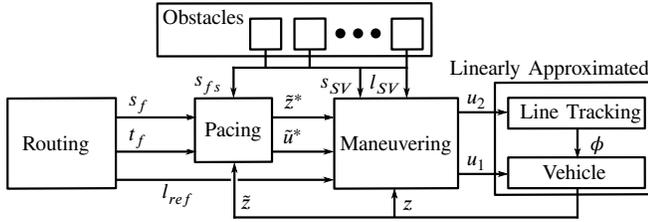


Fig. 1. Block diagram of the automated driving system proposed in this paper.

TABLE I
BLOCK DIAGRAM NOMENCLATURE.

Parameter	Definition
s_{SV}	Surrounding vehicle's longitudinal position
l_{SV}	Surrounding vehicle's lane
s_f	Ego vehicle's goal position for the current subtrip
t_f	Ego vehicle's goal time for the current subtrip
s_{fs}	Surrounding vehicle's goal position
l_{ref}	Constant reference lane for the current subtrip
z	State vector
\tilde{z}	Longitudinal state vector
\tilde{z}^*	Reference longitudinal state vector
\tilde{u}^*	Reference acceleration command
u_1	Scalar longitudinal acceleration command in the road frame
u_2	Lane command
ϕ	Steering angle

ture and the corresponding structure of the methods portion of the paper, that is, Sections III, IV, and V. Section III provides both nonlinear and approximate linear models for simulation and control, respectively. Section IV derives the long-term reference trajectory using PMP. Then, Section V shows how the reference is used in RHC and how obstacles are avoided in a multi-lane environment. Next, simulation scenarios based on real-world road segments are detailed in Section VI. The results including time, energy, and lane targeting benefits are presented in Section VII. Finally, Section VIII reviews the contributions and looks ahead to future research in algorithms and microsimulation.

II. ARCHITECTURE

Figure 1, with nomenclature in Table I, shows the relationships between several of the components involved in model based autonomous driving. The term “ego” in Table I refers to the individual controlled vehicle, i.e. the host. This paper addresses implementation of the pacing, maneuvering, and line tracking blocks as well as modeling of actuators and vehicle hardware. Input data from sensors and connectivity is also used. While it is not the subject of this paper, a routing module is assumed to choose road links and provide a goal position s_f , time t_f , and lane l_{ref} . These goals apply for the current subtrip, that is, the process of traveling from one stopping point to the next. For an example of a routing algorithm considering link times with traffic signal integration, see [15].

Trajectory planning involves two levels. The first level, called the *pacing module*, computes a reference for RHC by optimizing the ego's motion over the rest of the road link neglecting traffic. Its purpose is to enable RHC to target the acceleration that minimizes the objective not only during the receding horizon, but also beyond it. The pacing module does so using Pontryagin's Minimum Principle (PMP) in a

shrinking horizon [16] scheme as described in Section IV, consuming a goal position and time to generate position, velocity, and acceleration profiles.

While the pacing module's long horizon performs well in the absence of traffic, it does not attempt the constraint-intensive task of collision avoidance. This is left to the second trajectory planning component called the *maneuvering module*, which controls both lateral and longitudinal degrees of freedom to avoid obstacles. It follows the pacing module's references over a receding horizon and outputs acceleration and lane commands. In addition to the longitudinal references from the pacing module, it also consumes a fixed reference lane directly from the routing module. The implementation of the maneuvering module builds upon [9]. A new lane indicator formulation enables an arbitrary number of lanes, among other improvements detailed in Section V.

The acceleration command is sent to an acceleration controller that delivers it using the powertrain and brakes. Meanwhile, the commanded lane's centerline becomes the reference for the line tracking module, which controls the vehicle's steering angle. Section III details both nonlinear and linear approximation of the system consisting of the vehicle hardware, acceleration controller, and line tracking controller.

III. MODELING

This section describes both nonlinear modeling for the simulation testbed and linear approximation for online optimization. It moves from vehicle modeling to the steering controller and ends with linear approximation.

A. Vehicle Model

A standard kinematic bicycle model, shown in Fig. 2, is combined with a first-order approximation of powertrain and brake dynamics to simulate the vehicle. The inputs to this model are the vehicle-aligned acceleration command u_t and the steering angle ϕ . No-slip is assumed, enabling calculation of the distance R from the instant center of chassis yaw rotation to the rear wheel using the steering angle and wheelbase. All vehicles are 1.9 m wide with wheelbases $L = 4.52$ m.

$$R = \frac{L}{\tan \phi} \quad (1)$$

Next, the following system with time constant τ determines the tangential speed v_t , which is needed to find the normal acceleration a_n and the yaw rate $\dot{\psi}$.

$$\dot{v}_t = a_t, \quad \dot{a}_t = -\frac{1}{\tau} a_t + \frac{1}{\tau} u_t \quad (2)$$

$$a_n = \frac{v_t^2}{R} \quad (3)$$

$$\dot{\psi} = \frac{v_t}{L} \tan \phi \quad (4)$$

Coordinate transformations then yield velocity and acceleration in the road frame.

$$\dot{x}_a = v_t \cos \psi, \quad \dot{y}_a = v_t \sin \psi \quad (5)$$

$$a_x = a_t \cos \psi - a_n \sin \psi, \quad a_y = a_n \cos \psi + a_t \sin \psi \quad (6)$$

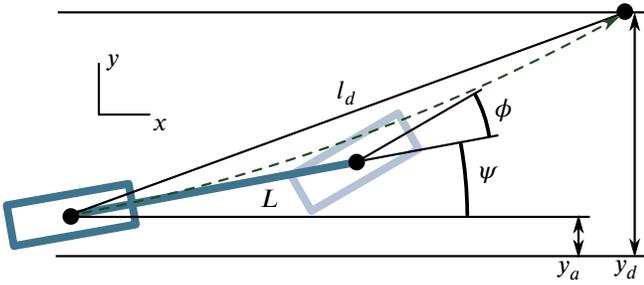


Fig. 2. Schematic of the bicycle model with pure pursuit parameters shown.

As shown in Fig. 1, the maneuvering module provides u_1 in the road frame; however, Eqn. (2) consumes u_t in the vehicle frame. The following converts u_1 into u_t .

$$u_t = \frac{1}{\cos \psi} \left(u_1 + \frac{v_r^2 \sin \psi}{R} \right) \quad (7)$$

It is also possible to obtain a dynamic bicycle model by using tire slip angles to obtain the forces that act on the chassis. These more complex models can capture the effects of limited grip between the tires and road surface. The kinematic model here cannot and is not intended to accomplish this; its role is to approximate a real dynamic vehicle to motion-planning consistency in human-like, sub-limit lane changes. The satisfaction of this assumption is discussed further in Section III-D.

B. Steering Control

As illustrated in Fig. 2, a pure pursuit controller [17] tracks the rear axle's position y_a to the maneuvering module's commanded lane after it is converted to the lateral reference position y_d . This controller calculates a steering angle based on the arc required to drive the vehicle's rear axle to the point lying a fixed distance away on the reference path, called the *lookahead* point. Although this paper assumes straight, parallel lanes to simplify location of the lookahead point, pure pursuit can also be applied to general curves by defining them as a set of discrete points [17]. Equation (8) determines the lookahead distance l_d , which is proportional to vehicle speed with gain $k_h = 1.5$.

$$l_d = \max \left\{ k_h v, \sqrt{(y_d - y_a)^2 + (1.5L)^2} \right\} \quad (8)$$

The value of k_h was set to obtain a human-like lane change duration based on data from [18]. At low speeds, l_d is saturated to prevent instability.

The pure pursuit control law for steering angle ϕ is given in Eqn. (9).

$$\phi = \arctan \frac{2L \sin \left(\arcsin \frac{y_d - y_a}{l_d} - \psi \right)}{l_d} \quad (9)$$

Finally, the steering rate is limited to $310 \frac{\text{deg}}{\text{s}}$ at the steering column to match actuator specifications from [19]. This results

in a limit of $\dot{\phi}_{\max} = 20.4 \frac{\text{deg}}{\text{s}}$ at the wheels, which the following limit imposes.

$$\phi(k) - \Delta t_h \dot{\phi}_{\max} \leq \phi(k+1) \leq \phi(k) + \Delta t_h \dot{\phi}_{\max} \quad (10)$$

$\Delta t_h = 0.08 \text{ s}$ is the sampling time used to simulate the steering controller and vehicle dynamics.

C. Linear Approximation for Lane Change Planning

The maneuvering module requires a linear model of the system that it controls. This system, outlined in Fig. 1, includes the vehicle model and steering controller described earlier in this section. The response from acceleration and lane commands u_1 and u_2 , respectively, to vehicle position is linearly approximated as in [9]. Position s , road-aligned velocity v , road-aligned acceleration a , lane position l , and lane position rate r_l make up the state vector of this approximated linear model. The latter two states use lane width as the distance unit, with integer values of l corresponding to lane centerlines. The lane position l is related to y_a by

$$l = \frac{y_a}{w_l} + 0.5 \quad (11)$$

where w_l denotes the lane width. For example, consider a road with two 4 m lanes. If y in Fig. 2 equals zero on the fog line separating the road from its shoulder, then a vehicle centered in the right lane has $y_a = 2 \text{ m}$ and $l = 1.0$. Exactly halfway through a lane change, $y_a = 4 \text{ m}$ and $l = 1.5$. In the remainder of this study, $w_l = 3.7 \text{ m}$. The first control input $u_1 \in \mathbf{R}$ is the desired acceleration and the second input $u_2 \in \mathbf{Z}$ is the integer-valued lane command.

An underdamped 2nd order response was selected to linearly approximate the lateral dynamics. Intuitively, such a model can capture the pure pursuit controller's small overshoot in addition to the zero initial yaw rate when steering angle is applied. Notice that despite the discrete u_2 , the lane state l is continuous. This feature helps enforce lane discipline while still comprehending intermediate positions during lane changes. A first-order system with time constant τ represents the lag in powertrain response. Equation (12) gives the overall model and Table II shows the parameters that were obtained by matching the approximate 2nd order response to the nonlinear response using classical system identification. The static gain K of 1 implies that u_2 and l have the same unit.

$$\frac{d}{dt} \begin{bmatrix} s \\ v \\ a \\ l \\ r_l \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega_n^2 & -2\xi\omega_n \end{bmatrix} \begin{bmatrix} s \\ v \\ a \\ l \\ r_l \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{\tau} & 0 \\ 0 & 0 \\ 0 & K\omega_n^2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (12)$$

D. Validity

The vehicle and steering controller were simulated in a sequence of left and right-hand lane change maneuvers at various speeds. Figure 3 shows the correlation in lane state l between the nonlinear and linear models. The most significant

TABLE II
LINEAR MODEL PARAMETERS.

Parameter	Definition	Unit	Value
τ	Acceleration time constant	sec	0.275
ξ	Lane change damping ratio	—	0.7077
ω_n	Lane change natural frequency	rad/s	0.9666
K	Lane change static gain	—	1

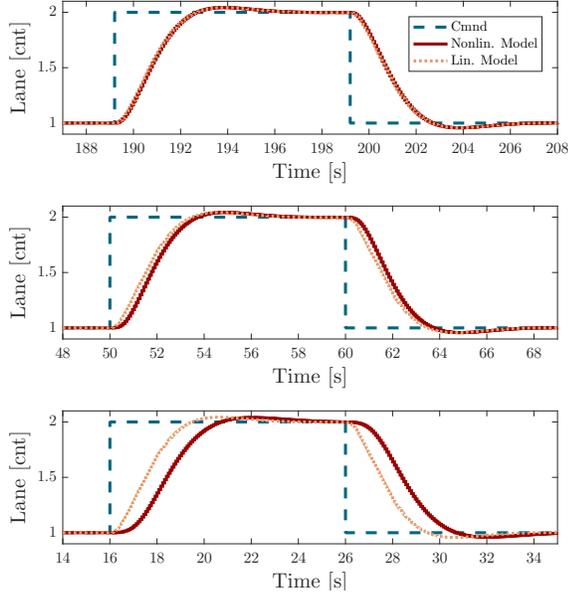


Fig. 3. Combined steering and controller and plant responses to lane command steps at (top to bottom) 24.3 m/s, 7.7 m/s, and 3.9 m/s.

cause of deviation was the steering rate limit, the effects of which diminished when high speed led to smaller steering angles¹. Correlation was sufficient for the receding horizon controller to perform as intended when lane changes were prohibited at low speeds. For example, constraints prevent the maneuver of Figure 3 (bottom) from occurring in closed-loop.

As previously noted, the kinematic model is inconsistent with real vehicle dynamics near the tire limit. Polack et al. [20] found that a lateral acceleration limit of $0.5\mu_r g$ should be observed when using a kinematic model, where μ_r is the road friction coefficient and g is the gravitational acceleration. The lane change responses in Fig. 3 did not exceed 3.3 m/s^2 , placing them below the $0.5\mu_r g$ limit for dry pavement where $\mu_r \approx 1$.

IV. PACING MODULE

This section explains the pacing block of Figure 1, which sets the traffic-free state and control references $\tilde{z}^*(i)$ and $\tilde{u}^*(i)$ to be consumed later by the RHC. The superscript $*$ here indicates that the value of the base variable solves the optimal control problem. The reference state \tilde{z} contains position s and speed v , while the reference control input \tilde{u} is physically equivalent to the acceleration a . It is possible to design a stand-alone maneuvering module as in [9] or set \tilde{z}^* according to a user-specified constant speed. This could be useful at lower autonomy levels where an operator might specify a

¹To understand why this occurs, refer to Eqns. (4, 5) and increase v_i while holding \dot{y}_a constant.

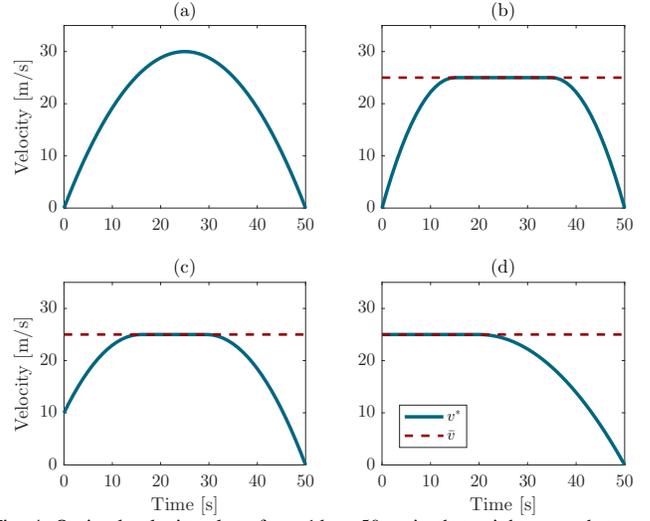


Fig. 4. Optimal velocity plans for a 1 km, 50 s trip that might occur between intersections. Various cases $\{a, b, c, d\}$ are shown, with $\bar{v} = \{\infty, 25, 25, 25\} \text{ m/s}$ and $v_0 = \{0, 0, 10, 25\} \text{ m/s}$.

target speed. However, if the trip involves acceleration and deceleration rather than simple cruising, the result of such receding horizon optimization is not optimal for the whole trip. Hence the goal of this section is to solve a simplified but longer-term optimization whose solution will guide the maneuvering module.

The pacing module is implemented in a shrinking horizon manner. Similarly to the popular receding horizon control, shrinking horizon control (SHC) operates in closed loop by re-solving its optimization at each step. Unlike RHC, the end of the horizon is fixed at the end of the trip and thus the horizon shrinks as real time and position advance.

The proposed controller uses the velocity profile of Figure 4, which consists of parabolic and constant phases. The remainder of this section will show that such a profile minimizes the square of the ego vehicle's acceleration \tilde{u} over the trip. Minimizing acceleration is expected to save energy, deliver a comfortable ride, and prevent traffic disturbances by reducing unneeded braking. Sciarretta and Vahidi [21] present more complex energy-optimal velocity profiles that could be substituted for a theoretical energy benefit.

The following optimal control problem minimizes the objective \tilde{J} subject to boundary conditions on the states and a pure state constraint that limits speed to \bar{v} .

$$\begin{aligned}
 \min \tilde{J} &= \int_{t_0}^{t_f} \tilde{u}^2 dt \\
 \text{s.t. } &s(t_0) = s_0, \quad s(t_f) = s_f \\
 &v(t_0) = v_0, \quad v(t_f) = 0 \\
 &\dot{s} = v, \quad \dot{v} = \tilde{u} \\
 &v \leq \bar{v}
 \end{aligned} \tag{13}$$

Following the direct adjoining approach from [22] with state vector $\tilde{z} = [s \quad v]^T$, the Hamiltonian \mathcal{H} and Lagrangian \mathcal{L} are thus formed with costates λ and Lagrange multiplier μ .

$$\mathcal{H} = \tilde{u}^2 + \lambda_1 v + \lambda_2 \tilde{u} \tag{14}$$

$$\mathcal{L} = \tilde{u}^2 + \lambda_1 v + \lambda_2 \tilde{u} + \mu(\bar{v} - v) \tag{15}$$

The following conditions are necessary for optimality [22].

$$\frac{\partial \mathcal{L}}{\partial \ddot{u}} = 0, \quad \frac{\partial \mathcal{L}}{\partial \ddot{u}} = 2\ddot{u} + \lambda_2 \implies \ddot{u}^* = -\frac{1}{2}\lambda_2^* \quad (16)$$

$$\dot{\lambda}_1^* = -\frac{\partial \mathcal{L}}{\partial s}, \quad \frac{\partial \mathcal{L}}{\partial s} = 0 \implies \dot{\lambda}_1^* = c_1 \quad (17)$$

$$\dot{\lambda}_2^* = -\frac{\partial \mathcal{L}}{\partial v}, \quad \frac{\partial \mathcal{L}}{\partial v} = \lambda_1 - \mu \implies \dot{\lambda}_2^* = -\lambda_1^* + \mu^* \quad (18)$$

$$\mu^* \geq 0, \quad \mu^* (\bar{v} - v^*) = 0 \quad (19)$$

Combining the above conditions yields

$$\ddot{u}^* = \frac{1}{2}c_1 - \frac{1}{2}\mu^* \quad (20)$$

Since \ddot{u}^* depends on μ^* , two cases are considered: Case I when the velocity constraint is inactive and Case II when it is active. As explained in [23], the solutions of optimal control problems like this one are not always described by continuous functions. Instead, they may involve several continuous phases joined together to form a piecewise solution. Cases I and II result in phases of the piecewise solution to this problem. First, the optimal control $\ddot{u}^*(t)$ is derived in Case I.

$$\mu^* (\bar{v} - v^*) = 0, \quad \bar{v} - v^* > 0 \implies \mu^* = 0 \quad (21)$$

$$\ddot{u}^* = \frac{1}{2}c_1 \implies \ddot{u}^* = \frac{1}{2}c_1 t - c_2 \quad (22)$$

In Case II, $v^* = \bar{v}$ gives $\ddot{u}^*(t) = 0$. The jump conditions [22] result in additional restrictions at corner times t_c . Let p and q index continuous phases. First, Eqn. (23) shows that c_1 is the same for all phases, where η is the non-negative jump parameter [22].

$$\lambda_1^*(t_c^-) = \lambda_1^*(t_c^+) + \eta(t_c) \frac{\partial}{\partial s} (\bar{v} - v) \quad (23a)$$

$$\lambda_1^*(t_c^-) = \lambda_1^*(t_c^+), \quad \lambda_1^* = c_1 \implies c_1^p = c_1^q \quad \forall p, q \quad (23b)$$

The second corner condition, which reduces to continuity of the Hamiltonian, shows that \ddot{u}^* is continuous and v^* is C^1 continuous. Although Eqn. (24) supposes t_c^- is in Case I, generality is not lost since $\frac{\partial}{\partial t} (\bar{v} - v) = 0$. Recall $\ddot{u} = 0$ in Case II and velocity must be continuous.

$$H^*(t_c^-) = H^*(t_c^+) - \eta(t_c) \frac{\partial}{\partial t} (\bar{v} - v) \quad (24a)$$

$$\ddot{u}^{*2}(t_c^-) + c_1 v^*(t_c^-) + \lambda_2^*(t_c^-) \ddot{u}^*(t_c^-) = \ddot{u}^{*2}(t_c^+) + c_1 v^*(t_c^+) + \lambda_2^*(t_c^+) \ddot{u}^*(t_c^+) \quad (24b)$$

$$\ddot{u}^{*2}(t_c^-) + \lambda_2^*(t_c^-) \ddot{u}^*(t_c^-) = 0 \quad (24c)$$

$$\ddot{u}^{*2}(t_c^-) - 2\ddot{u}^{*2}(t_c^-) = 0 \implies \ddot{u}^*(t_c^-) = 0 \quad (24d)$$

Since the vehicle is assumed to be static at the end of the trip, the following piecewise arrangements are possible depending on whether the constraint is active and whether the initial speed is less than, or equal to the maximum speed. In the case where the simple parabolic solution does not violate

\bar{v} , Eqn. (22) applies for all t . When $v_0 < \bar{v}$ and the constraint is active:

$$\ddot{u}^*(t) = \begin{cases} \frac{1}{2}c_1 t - c_2^I & ; \quad t < t_1 \\ 0 & ; \quad t_1 \leq t < t_2 \\ \frac{1}{2}c_1 t - c_2^{III} & ; \quad t_2 \leq t < t_f \end{cases} \quad (25)$$

When $v_0 \geq \bar{v}$:

$$\ddot{u}^*(t) = \begin{cases} 0 & ; \quad t \leq t_1 \\ \frac{1}{2}c_1 t - c_2 & ; \quad t_1 \leq t < t_f \end{cases} \quad (26)$$

The most complex case, Eqn. (25), requires six constants to determine $v^*(t)$: c_1 , c_2^I , c_2^{III} , t_1 , t_2 , and finally c_3 which arises from integrating acceleration for velocity. Two relations come from the boundary conditions $s(t_f) = s_f$ and $v(t_f) = 0$ after integrating Eqn. (25). Two additional relations come from the continuity of $v^*(t)$, which requires $v^*(t_1^-)$ and $v^*(t_2^+)$ to equal the constraint value \bar{v} . C^1 continuity of $v^*(t)$ at t_1 and t_2 complete the set of six equations to solve for the parameters. The two simpler cases use a reduced form of this strategy.

The unconstrained trajectory's coefficients are first computed explicitly. If a constraint violation results, the parameters of the appropriate piecewise solution are determined by solving the system described above.

Finally, the solution v^* for $v(t)$ is integrated forward to obtain the desired terminal position for the RHC. This is the ideal position of the ego vehicle on the last step of the RHC's horizon. The maneuvering module will plan to achieve this position if obstacles do not prevent it from doing so. N and Δt_l are the prediction steps and sampling time of the RHC, respectively.

$$s^*(N|k) = s(k) + \int_{t(k)}^{t(k)+N\Delta t_l} v^*(t) dt \quad (27)$$

V. MANEUVERING MODULE

With a plant model, steering controller, and pacing module in place, the maneuvering module plans acceleration and lane position trajectories over a receding 10s horizon using the linear model of Eqn. (12). This section presents the objective, continuous-valued constraints, and mixed integer constraints. The latter category helps handle discrete-valued lane selection and disjunctions in the collision avoidance constraints. For details on casting the problem as a standard-form MIQP, refer to the Appendix.

A. Objective

The maneuvering objective J is based on the authors' prior research [9]. Penalties on deviation from the desired acceleration and lane are retained with weights q_a and q_l , respectively. The function

$$J = z_e^T(N) P z_e(N) + \sum_{i=0}^{N-1} [z_e^T(i) Q z_e(i) + u_e^T(i) R u_e(i)] \quad (28a)$$

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_l & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad P = \begin{bmatrix} q_s & 0 & 0 & 0 & 0 \\ 0 & q_v & 0 & 0 & 0 \\ 0 & 0 & q_a & 0 & 0 \\ 0 & 0 & 0 & q_l & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (28b)$$

TABLE III
RECEDING HORIZON CONTROLLER PARAMETERS.

Parameter	Definition	Value
q_s	Terminal position weight	10
q_v	Terminal velocity weight	2
q_a	Acceleration weight	30
q_l	Lane weight	10
N	No. of prediction steps	25
Δt_l	Sampling time	0.4 s

$$R = \begin{bmatrix} q_a & 0 \\ 0 & q_l \end{bmatrix} \quad (28c)$$

is minimized, where $z_e(i) := z(i) - z_{ref}(i)$ and $u_e(i) := u(i) - u_{ref}(i)$. The state reference z_{ref} combines the longitudinal position and speed \tilde{z}^* with the acceleration \tilde{u}^* and the routing-based lane reference l_{ref} according to the following equation. Since l_{ref} is constant, the reference lane rate is 0.

$$z_{ref}(i) = [\tilde{z}^{*\top}(i) \quad \tilde{u}^*(i) \quad l_{ref} \quad 0]^\top \quad (29)$$

Notice that the higher-order maneuvering model's input u_1 differs from the pacing model's input \tilde{u} because of the former's first-order lag. This small difference is resolved by approximating the first-order lag as a one-step delay i.e. $a(i+1) \approx u_1(i)$. Since l_{ref} is constant and $u_2 = l_{ref}$ results in $l = l_{ref}$ at steady state, l_{ref} is directly used as the reference for u_2 .

$$u_{ref}(i) = [\tilde{u}^*(i+1) \quad l_{ref}]^\top \quad (30)$$

Thus, the pacing module's position, speed, and acceleration targets are combined with the constant lane reference and used in the maneuvering module. Table III lists the relevant parameter values. In the rest of this section, the prediction step argument i is dropped where it is equal for all terms.

B. Standard Constraints

A set of standard continuous-valued constraints is necessary to support basic motion planning. Many of these constraints involve state variables. In practice, factors including model mismatch, prediction mismatch, and measurement errors can alter the relationship between the control inputs, states, and constraint limits. This can render the optimal control problem infeasible, which is unacceptable in the field because it prevents the controller from returning a solution. Therefore, pure state constraints are softened using techniques described in [24]. The slack variables for constraint softening are denoted ϵ with subscripts to distinguish each particular slack variable from the others. All of these ϵ are non-negative.

$$\epsilon_j \geq 0 \quad \forall j \quad (31)$$

Velocity v is constrained to be non-negative and below a speed limit \bar{v} .

$$0 - \epsilon_2 \leq v \leq \bar{v} + \epsilon_3 \quad (32)$$

The RHC assumes a constant braking capacity \underline{u}_1 . In contrast, forward acceleration capacity varies as a function of speed. This results from increasing aerodynamic drag at higher speed and varying torque capacity as a function of engine speed. As [25] shows, the maximum acceleration limit is approximated as a piecewise function with two linear phases. The

slopes of these lines in (v, u_1) space are $m_1 = 0.285 s^{-1}$ and $m_2 = -0.1208 s^{-1}$, and the u_1 -intercepts are $b_1 = 2 m/s^2$ and $b_2 = 4.83 m/s^2$. Similar constraints on the acceleration state a prevent engine torque from momentarily exceeding capacity because of the lag in Eqn. (12).

$$u_1 \geq \underline{u}_1 \quad (33a)$$

$$u_1 \leq m_1 v + b_1, \quad u_1 \leq m_2 v + b_2 \quad (33b)$$

$$a \leq m_1 v + b_1 + \epsilon_6, \quad a \leq m_2 v + b_2 + \epsilon_6 \quad (33c)$$

Constant minimum and maximum constraints \underline{l} and \bar{l} on lane position l guarantee that the vehicle drives on the general road surface. In contrast to [9], δ is defined as the maximum deviation from a lane's centerline for which the vehicle remains at least partially in that lane.

$$\underline{l} - (1 - \delta) \leq u_2 \leq \bar{l} + (1 - \delta) \quad (34a)$$

$$\underline{l} - (1 - \delta) - \epsilon_4 \leq l \leq \bar{l} + (1 - \delta) + \epsilon_5 \quad (34b)$$

C. Mixed Integer Formulation for Arbitrary Lane Count

The Big M technique [26] is used throughout this section to model disjunctive constraints, or sets of constraints where at least one but not all must be satisfied. First, the constant M is set to at least the largest possible constraint violation. Then, M is multiplied by a binary variable and added to the constraints. Thus, the term containing M either drops to the yield the original constraint or guarantees constraint satisfaction depending on the binary's value.

Each lane, identified by the integer λ , is associated with two binary indicator variables $\mu_{\lambda a}$ and $\mu_{\lambda b}$. Labels are assigned λ values such that $l = \lambda$ at the centerline of lane λ . When part of a vehicle resides in a lane, the following constraints set both equal to 1.

$$-l - M(1 - \mu_{\lambda a}) \leq -\lambda + \delta, \quad l - M\mu_{\lambda a} \leq \lambda - \delta \quad (35a)$$

$$l - M(1 - \mu_{\lambda b}) \leq \lambda + \delta, \quad -l - M\mu_{\lambda b} \leq -\lambda - \delta \quad (35b)$$

Since the ego vehicle is always either in lane λ , in a lane greater than λ , or in a lane less than λ , at least 1 of $\mu_{\lambda a}$ and $\mu_{\lambda b}$ is always equal to 1. Therefore,

$$2 - \mu_{\lambda a} - \mu_{\lambda b} = \begin{cases} 0 & \lambda - \delta \leq l \leq \lambda + \delta \\ 1 & \text{otherwise} \end{cases} \quad (36)$$

The result in Eqn. (36) can be used to relax the collision avoidance constraints when the ego is outside lane λ . This leads to the position constraints

$$-s - M(2 - \mu_{\lambda a} - \mu_{\lambda b}) - M(1 - \beta_\zeta) \leq -s_{min}^\zeta - \underline{d} + \epsilon_1 \quad (37a)$$

$$s - M(2 - \mu_{\lambda a} - \mu_{\lambda b}) - M\beta_\zeta \leq s_{max}^\zeta - \underline{d} - l_v + \epsilon_1 \quad (37b)$$

where l_v denotes the ego vehicle's length and the binary variable β_ζ allows movement either in front of or behind obstacle ζ . Notice that $\beta_\zeta = 0$ only relaxes (37a) so that the ego vehicle must drive upstream of the obstacle and $\beta_\zeta = 1$ only relaxes (37b) so that the ego vehicle must drive downstream of the obstacle. The buffer distance \underline{d} separates the ego vehicle from the surrounding vehicle's (SV's) front and rear bumpers, located at s_{min}^ζ and s_{max}^ζ respectively. While

[27] briefly mentions Eqn. (37), it does not explain indicator setup or the other constraints that follow later in this section. The interested reader can instead refer to [27] for a study of how these softened collision avoidance constraints perform in an exceptionally hazardous traffic scenario.

Compared to smooth quadratic penalties on slack variables, non-smooth linear penalties have the advantage of yielding the exact hard-constrained solution as desired [24]. In earlier simulations, however, linear penalties led to harsh braking when model mismatch in dense traffic caused small constraint violations. To retain linear penalties' exactness in emergencies while first attempting to avoid undue proximity to SVs, a two-stage soft constraint strategy is adopted. In Eqn. (38), the quadratically-penalized ϵ_1 (the first stage) is upper-bound soft constrained to a maximum violation δ_v with linearly-penalized slack ϵ_7 (the second stage). Equation (39) gives the augmented cost.

$$\epsilon_1 \leq \delta_v + \epsilon_7 \quad (38)$$

$$J_a = J + \rho_1 \epsilon_1^2 + \sum_{m=2}^7 \rho_m \epsilon_m \quad (39)$$

The minimum distance between vehicle bumpers is set proportionally to the maximum speed for the road link. However, only the final 2 m before contact is penalized linearly with δ_v comprising the difference. The forms of the penalties along with the weight settings $\rho_1 = 1.5 \times 10^2$ and $\rho_7 = 1 \times 10^6$ cause the controller to trade off ϵ_1 with other objectives, but maintain $\epsilon_7 \approx 0$ when feasible.

On roads with more than two lanes, the RHC may output undesirable lane change commands if left without further constraints. Specifically, the RHC may step the lane command through multiple lanes in one loop. The linear model would predict a faster lane change rate in this case compared to a step of one lane. However, the linear approximation was not validated in such maneuvers, which are generally frowned upon among human drivers. Furthermore, the RHC may pulse the integer lane command to more precisely control the lane change rate. Therefore, the following constraints are introduced where the convergence threshold $\gamma = 0.05$ is just larger than the lane overshoot shown in Fig. 3.

$$u_2 - l \leq 1 + \gamma, \quad -u_2 + l \leq 1 + \gamma \quad (40)$$

Equation (40) prevents u_2 from moving more than $1 + \gamma$ beyond the current lane l . The effect is that when crossing multiple lanes, the RHC steps u_2 by one lane and waits for l to come within γ of u_2 before moving u_2 further.

To prevent u_2 from moving back to the previous lane before lane position has converged, the following constraints are introduced where $\Delta u_2(i) = u_2(i) - u_2(i-1)$.

$$\Delta u_2(i) + l(i) - u_2(i-1) \leq 1 + \gamma \quad (41a)$$

$$-\Delta u_2(i) + u_2(i-1) - l(i) \leq 1 + \gamma \quad (41b)$$

For example, consider an increasing-lane transition in progress where $l(i) < u_2(i)$. In order to revert to the previous lane i.e. set $\Delta u_2(i) = -1$, the remaining lane fraction $u_2(i-1) - l(i)$ must be less than γ to satisfy (41b).

While preventing the RHC from aborting lane changes reduces its ability to respond to disturbances, the RHC offsets this disadvantage in two ways. First, anticipative planning tends to obviate the abortion of lane changes, especially when chance constraints as in [27] are used for unconnected traffic. Second, the longitudinal acceleration u_1 remains free and is jointly optimized with the lane command.

Figure 3 shows that the linear approximation's validity deteriorates at low longitudinal speed. The RHC is therefore allowed to initiate lane changes only when $v \geq \underline{v}$. The following equations constrain the integer change over one timestep Δu_2 to a cone in $(v, \Delta u_2)$ space that is symmetric about the v axis. The numerical slack $\delta_n = 0.01$.

$$-m_l v + \Delta u_2 - m_l \epsilon_2 \leq \delta_n, \quad -m_l v - \Delta u_2 - m_l \epsilon_2 \leq \delta_n \quad (42a)$$

$$m_l = \underline{v}^{-1} \quad (42b)$$

Another set of constraints with a similar goal constrains the lane state l to remain converged to the command u_2 at low speeds. An indicator μ_v is first set to 0 for $v < \underline{v}$ and 1 for $v > \underline{v}$.

$$-v - M(1 - \mu_v) \leq -\underline{v}, \quad v - M\mu_v \leq \underline{v} \quad (43a)$$

Then, the difference between l and u_2 is constrained to be smaller than $1 - \delta$ when $v < \underline{v}$. The coefficient of μ_v is set just large enough in magnitude to relax the constraint when $\mu_v = 1$.

$$(l - u_2) + (-2 + \delta)\mu_v - \epsilon_2 \leq 1 - \delta \quad (44a)$$

$$-(l - u_2) + (-2 + \delta)\mu_v - \epsilon_2 \leq 1 - \delta \quad (44b)$$

VI. SIMULATION METHODS

A. Discretization

Two levels of discretization are used in simulation. The low-frequency event occurs every $\Delta t_l = 0.4$ s and includes maneuvering and pacing module calculations. The prediction model's discretization time is equal to Δt_l . Five high-frequency events with $\Delta t_h = 0.08$ s occur per low-frequency event. The steering controller and plant dynamics are simulated at high frequency to improve steering controller performance. This two-timestep idea also appears in the experimental research of [28], albeit with a 2-to-1 ratio between the step durations.

To focus on the core RHC formulation and integration with higher and lower-level control layers, computation delay is neglected and the agents operate synchronously. The vehicles compute their control plans sequentially and communicate them to other vehicles. Depending on the sequence, some information is Δt_l old when it is used by another CAV. These trajectories are advanced in time by Δt_l before incorporation into the constraints. More formally, if z_{SV} denotes the SV's state and the SV's trajectory is Δt_l old, then $z_{SV}(i) \leftarrow z_{SV}(i+1)$.

Although this paper focuses on a fully connected environment, it is sometimes necessary to predict SV motion. In the kinematic approach used here and in [27], a constant longitudinal acceleration of either 1.15 m/s^2 , 0 , or -2.94 m/s^2 is expected depending on the SV's proximity to the nearest static obstacle or intersection stop bar. Laterally, constant velocity

is expected until the SV reaches the next lane centerline. This prediction applies in two situations. The first is when the furthest-ahead SV state $z_{SV}(N)$ is unavailable because the SV's communication is old, as discussed previously. In this case, $z_{SV}(N)$ is predicted from $z_{SV}(N-1)$. The second case is the first simulation step when some SVs have not yet sent their intended trajectories. At this initial step only, trajectories are predicted for these SVs as if they were not connected.

Regarding the validity of this section's assumptions, [29] experimentally showed under 0.01 s communication latency using DSRC, well below the 0.08 s simulation timestep. Furthermore, the authors' earlier two-lane MATLAB implementation ran faster than real-time on average [9] and a more recent multi-lane C++ version computed faster than real-time in 95 % of cases. Although computation time is promising for a decentralized implementation and the simulations do include some delayed information, the sequential computation scheme does not model pairs of vehicles that use old information from one another. To address such cases, the authors have devoted [30] to the multi-agent topic including a parallel-computed variant of the algorithm presented here. Asynchronous computation between vehicles is left for future research.

B. Scenarios

The simulations are evaluated in three situations: two highway and one arterial. Figure 5 depicts the two road segments. The highway geometry is evaluated at two initialization rates to assess the controller's sensitivity to traffic density. These scenarios are either all-reactive or all-CAV, where the CAVs communicate their intended future trajectories to form the surrounding CAVs' constraints. In an unconnected environment, an AV must predict these trajectories. The performance of this paper's RHC and steering control layers in such an environment is examined in [27], where chance constraints help account for the added uncertainty.

The vehicles compute their solutions in a sequential, decentralized fashion with a fixed ordering, which is listed as part of each scenario's description below. Although the ordering affects collective performance, it is not studied here for the sake of brevity. However, the authors' related research in [30] does explore this topic by simulating various orderings and proposing a technique for dynamically constructing higher performing ones.

Lane changes are disallowed where solid markings, called *dividers*, appear in Fig. 5. This is accomplished using a rule-based algorithm that inserts a fictitious obstacle on the appropriate side of the divider depending on l_{ref} and the divider's location relative to the host. The advantage of this technique is that it uses only the constraints of Section V-C without adding binary variables; the limitation is that the controller cannot plan to traverse the off-target side of a short upcoming divider and return to the target lane later. Such a maneuver is not possible in this study regardless of algorithm.

1) *Highway Merge*: The highway scenario includes two main lanes of traffic flow, an on-ramp, and an exit ramp. The two ramps share a merging zone that models South Carolina Exit 48 on US Interstate 85. To reduce memory load, the

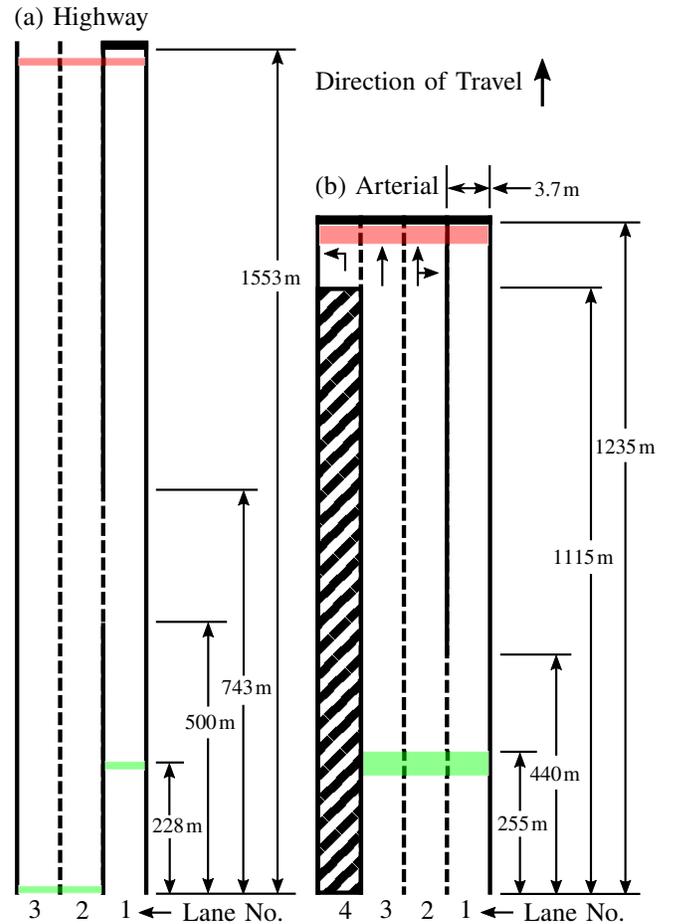


Fig. 5. The road geometry. Approximate vehicle initialization and deactivation zones are shaded in green and red, respectively. Lane changes are allowed across dashed markings and prohibited across solid markings. Longitudinal distances are to scale, but aspect ratio is distorted for visibility.

far left lane of the real highway is omitted from the model. Vehicles initialize on the main road's right-hand lane at 30 m/s with either 4 s or 2 s headways. Other vehicles initialize on the on-ramp with a time headway 3 times that of the main traffic flow. All of these entering vehicles have $l_{ref} = 2$, meaning that they target the right lane of the highway. Among vehicles that initialize on the highway, a randomly selected $\frac{1}{3}$ of them have $l_{ref} = 1$ and therefore attempt to exit the highway. This results in balanced flow rates of entering and exiting vehicles.

In this case, the vehicles enter and exit the simulation dynamically and do not reach their goal positions. The pacing module targets either the end of the following exit ramp in the case of highway-bound vehicles or the next intersection in the case of exiting vehicles. That intersection's stop bar is shown at the end of the right lane in Figure 5(a). Boundary effects are mitigated in post-processing by ignoring data at the extreme beginning and end of the road link. Furthermore, only vehicles that complete the entire link while it is full are analyzed. This time period begins when the first vehicle reaches the uppermost shaded area of Figure 5(a) and ends when the last vehicle is initialized. In this higher-speed scenario, the soft-constrained bumper-to-bumper distance is 12 m.

The computation sequence for the decentralized network solution in the highway scenario follows the temporal order

in which the vehicles are introduced at the upstream sources.

2) *Arterial Lane Positioning*: The arterial scenario models a block of US-123 in Clemson, South Carolina. This area is subject to changing and sometimes heavy congestion levels depending on time of year and local events. Vehicles begin at rest and travel to the stop bar of the subsequent intersection. A turning lane directs some traffic into a residential area on the right-hand side. At the next intersection, two lanes continue straight or turn right while one lane turns left.

Vehicles are initialized in an evenly spaced grid with 4 vehicles per lane. This is the maximum number of vehicles in the intersection's queue before the preceding intersection is blocked. The target lane of each vehicle is randomized such that 1 of the 12 vehicles always targets the right-hand turnoff, 3 vehicles target the left turn lane, and 4 target each remaining lane. The random lane reference causes the vehicles to rearrange their lane positions differently in each case, thereby affecting the results. Virtual obstacles are introduced downstream of the goal intersection's stop bar to prevent incursions into the intersection using Eqn. (37). In this lower-speed scenario, the soft-constrained bumper-to-bumper distance is 8 m.

The computation sequence for the decentralized network solution in the arterial scenario begins with downstream vehicles and ends with upstream vehicles. When vehicles are initialized side-by-side, right-hand vehicles compute first.

C. Baseline Rule-Based Algorithm

The intelligent driver model (IDM) [31] is augmented with a rule-based lane change algorithm and applied as a baseline. This combined system will be referred to as IDM-RB. Unlike the optimal control algorithm, it uses only current measured information about SVs rather than a connected preview. Pseudorandom IDM parameters that were derived in [25] based on experimentally measured histograms from [32] model variation in human preferences.

The full rule-based lane change algorithm is lengthy and can be found in [27]. In summary, it reacts to a 2 m/s slowdown in the desired lane by changing lanes if safe. Safety is determined based on the presence of neighboring vehicles within a velocity-based gap in the target lane. The logical flow of the lane change decision is thus similar to the classic Gipps model [33] without factors such as heavy vehicles that are not needed here. The rear time headway from [27] is omitted in this paper because doing so improved performance in the highway merge scenario.

A Note on Applicability to Humans: The longitudinal parameters in this study are empirically based to place model behavior within the range of human acceptability. However, comparability to human drivers is limited by the simple lane change model which, like all but a few lane decision models [34], has not been empirically validated on humans. Part of the complexity of human lane change modeling stems from anticipation and collaboration in human lane changing [35]. A specific limitation of IDM-RB is its lack of a gap relaxation phenomenon, which is an anticipative feature [35]. Nonetheless, IDM-RB does indicate the performance that

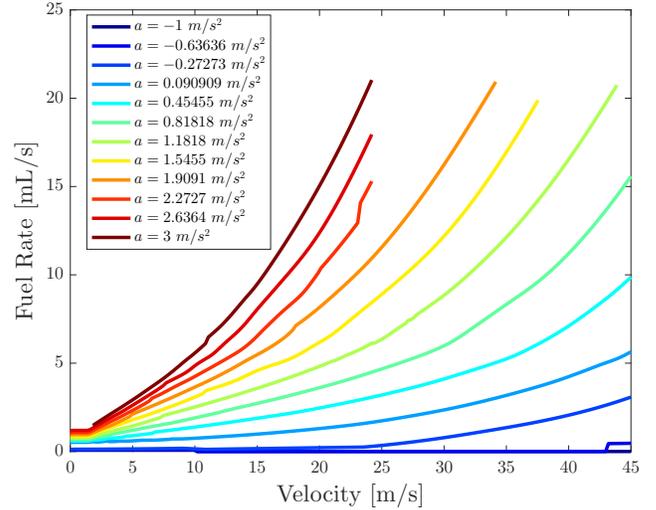


Fig. 6. Fuel consumption model output at various acceleration levels.

is possible using only direct, instantaneous measurements. Thus, the results demonstrate the benefits of anticipation and information-sharing in lane change combined with longitudinal eco-driving.

D. Energy Consumption

Energy consumption is calculated using the combustion-engine powertrain model first described in [36]. Additional parameter details and results can be found in [25]. With a level of detail between curve-fit models like the Willans approximation [37] and dynamic models like Autonomie [38], the powertrain model uses Newton's 2nd law and static maps to determine instantaneous fuel rate from velocity, current acceleration, and desired acceleration. The model works in four main steps.

- 1) Use Newton's law to compute the current and desired transmission output speed and torque based on the given operating point. Aerodynamic drag and rolling resistance are accounted for in this step.
- 2) Pass the desired operating condition from 1 to a lookup table to determine the optimum steady-state transmission gear. This lookup table is populated offline by computing the fuel consumption in each feasible gear and selecting the gear that minimizes it.
- 3) Using the gear ratio from 2 and the operating point from 1, compute the engine speed and torque.
- 4) Look up fuel consumption rate from a table computed from brake specific fuel consumption (BSFC) data.

Figure 6 illustrates the output as a function of vehicle speed and acceleration.

VII. RESULTS

As shown in Figure 7 and Table IV, CAVs generally resulted in both reduced energy use and quicker travel compared to the unconnected and reactive driver model. This indicates an efficiency improvement by eliminating unnecessary braking, since aerodynamic drag generally causes a trade-off between energy use and travel time. These benefits were greatest in the

TABLE IV
CHANGE RELATIVE TO IDM-RB.

Scenario	Fuel [L/veh]	Time [s/veh]	Lane Success [%]
Hwy. 4 s	-8.9	-5.2	+13.8
Hwy. 2 s	-10.0	-9.7	+99.6
Arterial	-13.7	-10.3	+12.1

arterial scenario with its high local density and frequent lane changes.

Regarding collision avoidance, no CAV-involved collisions occurred in either scenario. The baseline IDM-RB algorithm did experience a collision in one arterial case, which was removed from both the IDM-RB and CAV datasets. None of the vehicles that stopped at intersections intruded into those intersections. The more in-depth collision avoidance study in [27] evaluated the CAV algorithm in a denser unconnected environment with a perception fault and produced CAV-involved collisions in some cases.

The remainder of this section explains the calculation of performance metrics before more closely examining the highway and arterial scenarios.

A. Performance Metrics

The fuel and time metrics are expressed on a per-vehicle basis. Lane success rate is defined as the percentage of vehicles that ended the simulation in a lane position within 0.5 lanes of the integer-valued reference lane.

In the highway scenarios, three groups of vehicles were defined for statistical purposes: highway initialization with highway reference, highway initialization with exit ramp reference, and on-ramp initialization with highway reference. Without this grouping, the results distribution would have three peaks and would be poorly approximated as normal. The expectations and variances are calculated for the random variable T , which itself represents the weighted average of the random variables A , B , and C corresponding to each group.

$$T = aA + bB + cC, \quad a + b + c = 1 \quad (45)$$

The highway-to-highway fraction $a = 0.5$ and $b = c = 0.25$ based on the probabilities of any given vehicle belonging to each group. These coefficients are certain for the population because the groups are based on simulation inputs, e.g. reference lane, rather than outputs, e.g. actual final lane.

B. Highway Merge

Figure 8 demonstrates the optimal control system's planned trajectories during a highway merge with 2 s initial headways. The ego (vehicle 1) and vehicle 2 face a potential conflict, causing vehicle 2 to move into lane 3 to allow the ego to merge. The prediction horizon is long enough for vehicle 2 to plan a later return to its reference lane. Meanwhile, vehicle 3 plans to exit the highway by moving in front of vehicle 1 and vehicle 6 plans to merge just ahead of the exit ramp. Longitudinally, the ego's maneuvering module adjusts its acceleration plan relative to the pacing module's reference in order to merge smoothly and safely. Relative to the reference, the adjustment causes slower acceleration prior to the start of

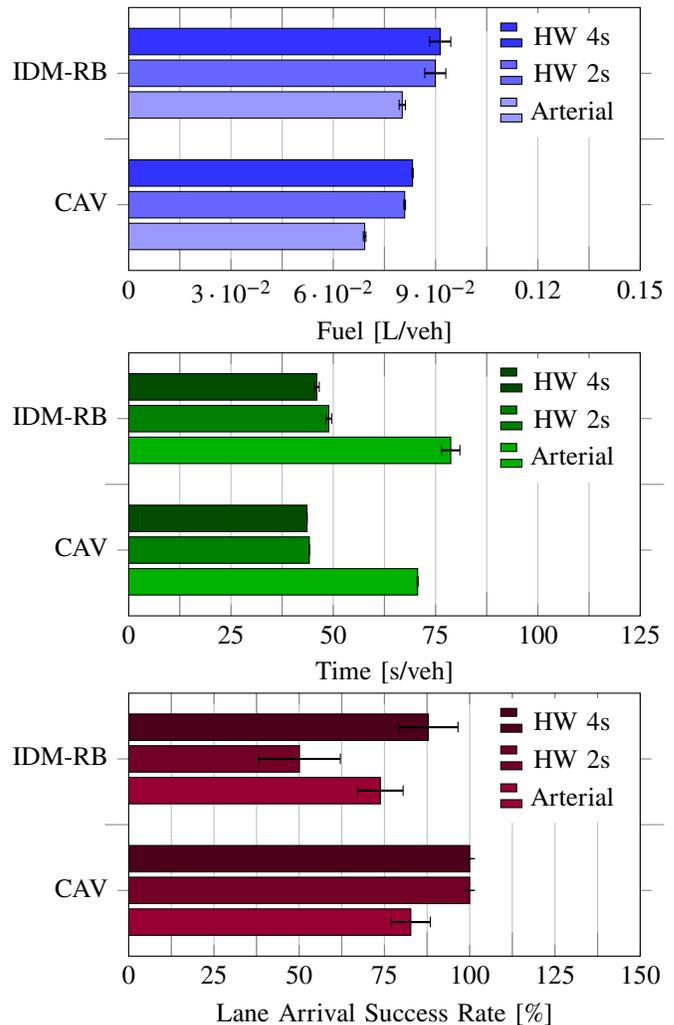


Fig. 7. Performance comparison between the combined Intelligent Driver Model + Rule Based (IDM-RB) and connected optimal control (CAV) algorithms. 95% confidence intervals are shown.

merging and more aggressive acceleration afterward, which matches the intuitive expectation of faster traffic on the main highway compared to the ramps.

The CAVs succeeded in attaining the target lane in all highway simulations as shown in Figure 7. IDM-RB also accomplished this for most vehicles in the lower-density scenario, but performance seriously degraded at higher density. In contrast, the density increase had little effect on CAV performance. Velocity-smoothing trends mirrored those of lane positioning as shown in Figure 9. As expected, a jam formed in the merging zone and dissipated downstream of the exit ramp. The MIQP-based lane change algorithm practically eliminated the jam under these conditions, resulting in reduced energy consumption and travel time compared to IDM-RB. Counterintuitively, the energy consumption results in Figure 7 decreased when density was increased. This is likely a result of lower average speed since travel time increased as expected.

C. Arterial Lane Positioning

The arterial scenario presented a greater lane positioning challenge for the optimal control algorithm. While all CAVs

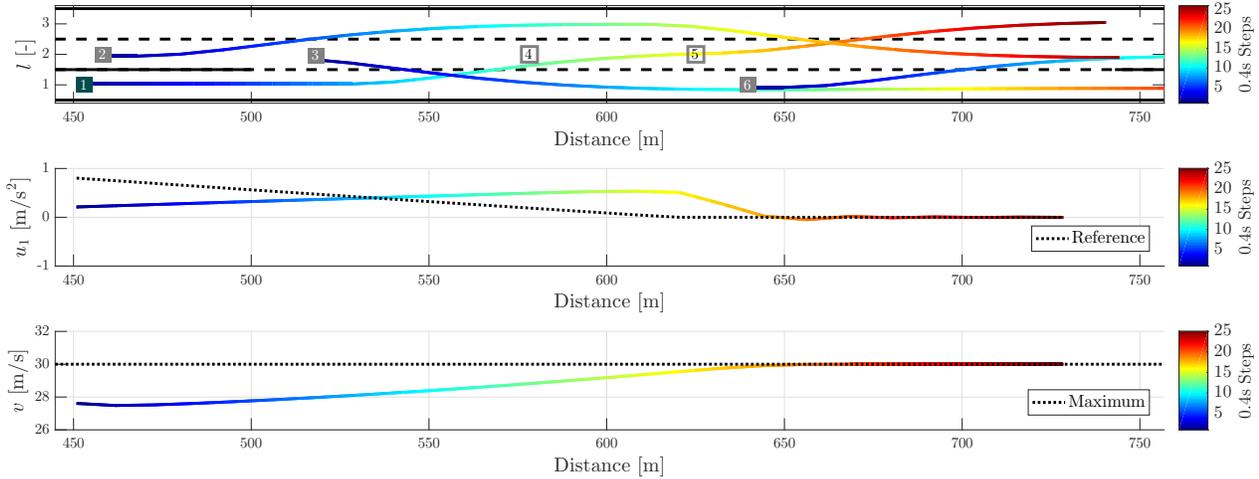


Fig. 8. Highway scenario position trajectories for the ego (green) and several surrounding vehicles (solid gray) along with planned velocity v and acceleration command u_1 for the ego vehicle. Other surrounding vehicles with omitted trajectories are shown as outlines. Color transitions from blue to red signify increasing prediction time.

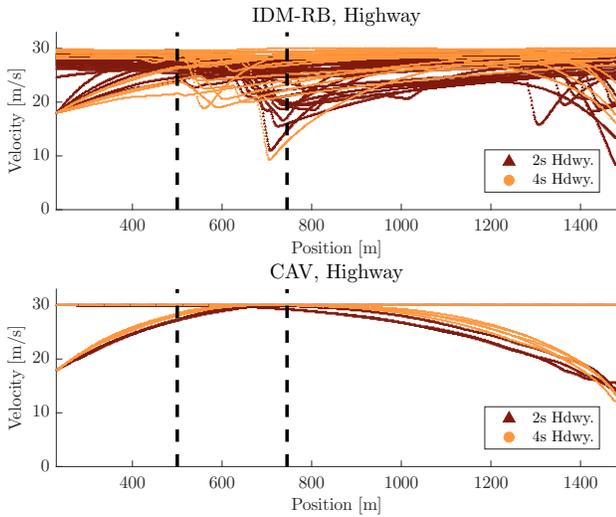


Fig. 9. Position-based speed profiles in the highway merge scenario. Dashed reference lines mark the merge zone boundaries.

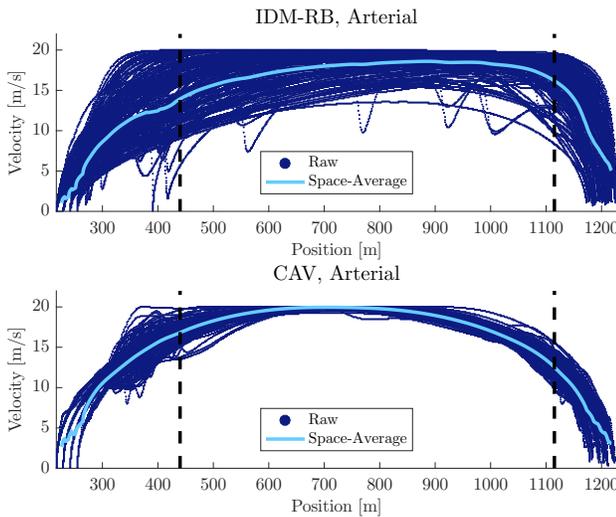


Fig. 10. Position-based speed profiles and 20m space-averages in the arterial lane positioning scenario. Dashed reference lines mark the separation of the right lane and the start of the left turning lane (see Figure 5).

did not reach their reference lanes before arrival at the next stoplight, success rate was still moderately improved compared

TABLE V
STANDARD DEVIATION.

Scenario	Fuel [L/veh]		Time [s/veh]		Lane Success [%]	
	IDM-RB	CAV	IDM-RB	CAV	IDM-RB	CAV
Hwy. 4 s	7.6E-3	4.4E-4	1.4	1.4E-1	12.3	0
Hwy. 2 s	8.2E-3	6.8E-4	1.9	2.7E-1	31.8	0
Arterial	6.0E-3	2.6E-3	14.8	9.1E-1	44.1	37.9

to IDM-RB (Figure 7). Combined with the lack of collisions, this result demonstrates that the optimal controller appropriately sacrifices road position when it cannot be attained safely.

With all vehicles starting and ending at rest, the arterial speed traces of Figure 10 offer insights into the benefits of long-term planning combined with precise short-term anticipation. IDM-RB tended toward longer, slower acceleration and more abrupt braking compared to the CAVs. It is shown in [31] that this behavior is associated with relatively large comfortable braking b_0 compared to the maximum acceleration a_0 used in the IDM. [32] showed that humans possess this characteristic. By accelerating more strongly and starting to decelerate earlier, the CAVs may have accessed more efficient engine operating points and spent more time in fuel cutoff. Furthermore, connected and anticipative lane change planning reduced the energy-wasting mid-trip braking events of Figure 10. These were typically driven by lane cut-ins, which have been mentioned as an opportunity for traffic improvement with intelligent driving. The above factors led to the time and energy benefits shown in Figure 7, whose error bars also indicate that the CAVs performed more consistently.

Table V compares the standard deviations of each performance metric between algorithms. The CAV results were more consistent in part because their randomness stemmed only from initial conditions, while IDM-RB also used random IDM parameters. However, optimal control also promoted consistency. For example, the CAVs' consistent travel times in the arterial scenario reflect the precise planning and closed-loop correction of the hierarchical system, even when the vehicle was obstructed by surrounding traffic. In contrast, IDM-RB does not recover lost time. If the vehicle encounters an obstruction, the trip is delayed. Even receding-horizon systems like [9] that track a constant velocity reference are

similar to IDM-RB in this respect.

VIII. CONCLUSION

Integrated algorithms for multi-lane automated driving were presented from plant modeling to trip-level longitudinal pacing. Pontryagin's Minimum Principle and mixed integer receding horizon control were combined to realize the benefits of each in long- and short-term planning, respectively. Simulations in highway merging and arterial road environments demonstrated the combined system's ability to approximate the simplified optimal control solution while satisfying the complicated constraints arising from traffic and vehicle capacities. Compared to the explicit intelligent driver model with rule-based lane-changing using current information only, the anticipative optimal control approach resulted in 8.9% to 13.7% less energy consumption. The travel time benefit of 5.2% to 10.3% shows that the energy savings come from more efficient rather than slower driving.

It is emphasized that the disturbances and subsequent efficiency gains reported here result from road geometry and vehicles' navigational goals. This contrasts with studies like [25] and [39] that test a controller's response to an imposed, drive cycle-based disturbance. It also offers an explanation of reduced energy benefit relative to [25], since all speed fluctuations in a drive cycle do not necessarily model phenomena that are avoidable with connectivity.

Even in this study, the results were strongly scenario-dependent. To further improve realism, the algorithms described here are currently being implemented in VISSIM. When fully developed, this environment will support integration with multiple links and traffic signals. It also includes a more mature human driver model to generate more accurate relative change results.

Along with more complex road networks, real-world implementations must contend with factors like curved roads and variable road surface conditions that influence the duration of safe lane changes. To help relax these simulations' dry-pavement and straight-lane assumptions, the authors are currently studying lane change responses from a commercial vehicle model that comprehends tire and suspension dynamics. Solutions under consideration involve variable model parameters ω_n and ξ or simplified nonlinear models.

APPENDIX

MIXED INTEGER QUADRATIC PROGRAM ASSEMBLY

This section describes the casting of the optimization as a standard-form MIQP. Commercial solvers including IBM CPLEX and Gurobi [40] [41] consume problem data formatted as the parameters of Eqn. (46).

$$\begin{aligned} \min \quad & \theta^T G_a \theta + f_a^T \theta \\ \text{s.t.} \quad & S_a \theta \leq \Xi_a \end{aligned} \quad (46)$$

In this section, θ denotes the vector of MIQP decision variables including control inputs u , slack variables ϵ , indicator variables μ , and Big M binaries β .

$$\theta := \begin{bmatrix} \mathbf{U}^T & \epsilon^T & \bar{\beta}^T \end{bmatrix}^T \quad (47)$$

$$\mathbf{U} := \begin{bmatrix} u(0)^T & u(1)^T & \cdots & u(N-1)^T \end{bmatrix}^T \quad (48)$$

$$\bar{\beta} := \begin{bmatrix} \mu(1)^T & \beta(1)^T & \cdots & \mu(N)^T & \beta(N)^T \end{bmatrix}^T \quad (49)$$

The subscript a indicates that a matrix has been augmented relative to its standard QP contents. Retained steps from standard QP-based MPC assembly are omitted here for brevity, but are described well by Maciejowski [42].

Since (46) does not include state variables z explicitly, (50) eliminates z from the formulation of Section V.

$$\mathbf{Z} = \Phi z(0) + \Gamma \mathbf{U} \quad (50)$$

$$\mathbf{Z} := \begin{bmatrix} z(1)^T & z(2)^T & \cdots & z(N)^T \end{bmatrix}^T \quad (51)$$

The matrices Φ and Γ follow from repeated application of the linear model (12) as in [42], except that move blocking of the integer-valued u_2 helps manage computation time [9].

The matrices G and f are constructed by standard QP procedure. Since μ and β do not enter the objective, ϵ_1 has a quadratic penalty, and ϵ_{2-7} have a linear penalties (Eqn. (39)), G is thus augmented to size the problem for the auxiliary variables. Let h denote the total number of control inputs, m the number of slack variables, and p the number of binaries.

$$G_a = \begin{bmatrix} G & \mathbf{0}_{h \times (m+p)} \\ \mathbf{0}_{1 \times h} & \rho_1 & \mathbf{0}_{1 \times (m-1+p)} \\ \mathbf{0}_{(m-1+p) \times h} & \mathbf{0}_{(m-1+p) \times (m+p)} \end{bmatrix} \quad (52)$$

The linear slack variable penalties are incorporated into f_a .

$$f_a = \begin{bmatrix} f^T & 0 & \rho_2 & \cdots & \rho_m & \mathbf{0}_{1 \times p} \end{bmatrix}^T \quad (53)$$

The constraint coefficient matrix S_a has 3 components: the standard QP-like constraint matrix S , the slack variable constraint matrix Υ , and the binary constraint matrix \mathcal{B} .

$$S_a = \begin{bmatrix} S & \Upsilon & \mathcal{B} \\ \mathbf{0} & & \mathbf{0} \end{bmatrix} \quad (54)$$

The right-hand side of the inequality constraints $\Xi = C + \mathcal{W}x_0$ contains binary-related terms via the vector C . These result from moving known left-hand side terms in Section V-C to the right-hand side. C is thus formed, where each b_i corresponds to a single prediction stage.

$$C = \begin{bmatrix} b_0^T & b_1^T & \cdots & b_N^T \end{bmatrix}^T \quad (55)$$

Each b_i is organized into three sections. The first section \bar{b}_i does not include content related to collision avoidance. It handles the standard Section V-B constraints, lane change completion constraints, and low-speed lane change disablement. The second section \hat{b}_i is used to set up the lane indicator variables. The final section \tilde{b}_i deals with constraints on position for collision avoidance.

$$b_i = \begin{bmatrix} \bar{b}_i^T & \hat{b}_i^T & \tilde{b}_i^T \end{bmatrix}^T \quad (56)$$

Since in this paper the road is modeled such that the number of lanes n_l is constant, \hat{b}_i does not depend on prediction stage. Therefore, its subscript is dropped and it is constructed once as follows. The symbol \otimes denotes the Kronecker tensor product.

$$\hat{b} = \mathbf{1}_{n_l \times 1} \otimes \begin{bmatrix} \delta + M \\ -\delta \\ \delta + M \\ -\delta \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ \vdots \\ n_l \end{bmatrix} \otimes \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} \quad (57)$$

TABLE VI
 NON-COLLISION CONSTRAINT ROWS.

Row	Description	u_2 Blocked
1-2	u_1 bounds	Present
3-4	u_2 bounds	Dropped
5-6	v bounds	Present
7-8	l bounds	Present
9-10	Low speed Δu disablement	Dropped
11-12	Low speed lane convergence	Dropped
13-14	Low speed indicator	Dropped
15-18	Lane convergence	Dropped
19-22	Max acceleration	Present

To facilitate the assembly of \tilde{b}_i , a vector \bar{s}_{ob} containing final position limits from obstacles is first constructed. Although the argument is dropped to reduce clutter, \bar{s}_{ob} depends on the prediction stage i . Let n_ζ denote the maximum number of reachable obstacles per lane. If an obstacle does not exist, $s_{min} \leftarrow -\infty$ and $s_{max} \leftarrow \infty$.

$$\bar{s}_{ob} = \left[s_{min}^{11} \quad s_{max}^{11} \quad \dots \quad s_{min}^{\zeta\lambda} \quad s_{max}^{\zeta\lambda} \quad \dots \quad s_{min}^{n_\zeta n_l} \quad s_{max}^{n_\zeta n_l} \right]^T \quad (58)$$

The component \tilde{b}_i is now constructed from \bar{s}_{ob} . The symbol \circ denotes the Hadamard product. After this computation, C and subsequently Ξ follow as discussed previously.

$$\tilde{b}_i = \left(\mathbf{1}_{n_l n_\zeta \times 1} \otimes \begin{bmatrix} -1 \\ 1 \end{bmatrix} \right) \circ \bar{s}_{ob} - \mathbf{1}_{n_l n_\zeta \times 1} \otimes \begin{bmatrix} d \\ d + l_v \end{bmatrix} + \mathbf{1}_{n_l n_\zeta \times 1} \otimes \begin{bmatrix} 3M \\ 2M \end{bmatrix} \quad (59)$$

Ξ_a is completed by augmenting with 0 to enforce $\epsilon > 0$ and δ_v to limit the quadratically-penalized slack.

$$\Xi_a = \left[\Xi^T \quad \mathbf{0}_{1 \times m} \quad \delta_v \right]^T \quad (60)$$

The slack variable matrix Υ is assembled from blocks v at each stage. At stages where u_2 is free:

$$\bar{v}_i = \begin{bmatrix} \mathbf{0}_{4 \times m} \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & -m_l & 0 & 0 & 0 & 0 \\ 0 & -m_l & 0 & 0 & 0 & 0 \\ \mathbf{0}_{10 \times m} \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ \mathbf{0}_{4n_l \times m} \end{bmatrix} \quad (61)$$

Refer to Table VI when interpreting the rows of Eqn. (61).

The number of obstacle constraints depends on the maximum number of obstacles $\zeta_{max}(i) = n_l n_\zeta$.

$$\tilde{v}_i = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & & \\ -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{2n_l \zeta_{max}(i) \times m} \quad (62)$$

Given \tilde{v}_i for obstacle constraints and \bar{v}_i for others, $v_i = \left[\bar{v}_i^T \quad \tilde{v}_i^T \right]^T$. The full-horizon matrix Υ is then assembled. The matrix \hat{v} establishes the linear and quadratic softening shown in Eqn. (39). [30] provides more detail on \hat{v} .

$$\Upsilon = \begin{bmatrix} v_0^T & v_1^T & \dots & v_N^T & \hat{v}^T \end{bmatrix}^T \quad (63)$$

Like Υ , the binary constraint block matrix B is composed of several matrices that correspond to prediction stages.

$$\tilde{B}_i = \begin{bmatrix} \mathbf{0}_{10 \times (1+2n_l)} & & & \\ -2 + \delta & 0 & \dots & 0 \\ -2 + \delta & 0 & \dots & 0 \\ M & 0 & \dots & 0 \\ -M & 0 & \dots & 0 \\ \mathbf{0}_{8 \times (1+2n_l)} & & & \\ \mathbf{0}_{4n_l \times 1} & & \mathbf{I}_{2n_l} \otimes \bar{M} & \end{bmatrix} \quad (64)$$

$$\bar{M} = \begin{bmatrix} M & -M \end{bmatrix}^T \quad (65)$$

When interpreting Eqn. (64), Table VI applies to the first 22 rows. The remaining rows accommodate the lane indicator variables. The first column enables Eqns. (43) and (44).

The matrix \tilde{B} incorporates front/rear binary variables β .

$$\tilde{B}_i = \mathbf{I}_{n_l \zeta_{max}} \otimes \bar{M} \quad (66)$$

\hat{B}_i contains the coefficients of $\mu_{\lambda a}$ and $\mu_{\lambda b}$.

$$\hat{B}_i = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{n_l} \otimes \dot{M} \end{bmatrix}, \quad \dot{M} = \begin{bmatrix} M & M \\ \vdots & \\ M & M \end{bmatrix}_{2\zeta_{max} \times 2} \quad (67)$$

$$B_i = \begin{bmatrix} \tilde{B} & \mathbf{0} \\ \hat{B} & \tilde{B} \end{bmatrix} \quad (68)$$

The full binary matrix is then assembled. This result is used in (54) to incorporate the binary variables in the constraints.

$$B = \begin{bmatrix} \mathbf{0}_{12 \times p} & & & \\ B_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & B_2 & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & & \ddots & & \vdots \\ \mathbf{0} & \dots & & \mathbf{0} & B_N \end{bmatrix} \quad (69)$$

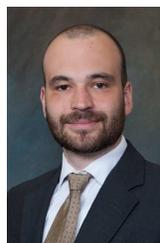
ACKNOWLEDGMENT

The authors thank Dr. Beshah Ayalew for proposing the project's lane change facet and Tyler Ard for his user feedback on the Appendix.

REFERENCES

- [1] Z. Zheng, S. Ahn, D. Chen, and J. Laval, "Freeway traffic oscillations: Microscopic analysis of formations and propagations using wavelet transform," *Procedia-Social Behavioral Sci.*, vol. 17, pp. 702–706, 2011.
- [2] M. J. Cassidy and J. Rudjanakanoknad, "Increasing the capacity of an isolated merge by metering its on-ramp," *Transp. Res. Part B: Methodological*, vol. 39, no. 10, pp. 896–913, 2005.
- [3] D. Ngai and N. Yung, "Automated vehicle overtaking based on a multiple-goal reinforcement learning framework," in *Intell. Transp. Syst. Conf. (ITSC) 2007*, 2007, pp. 818–823.
- [4] P. Wang and C. Chan, "Formulation of deep reinforcement learning architecture toward autonomous driving for on-ramp merge," in *Intell. Transp. Syst. (ITSC), 2017 IEEE 20th Int. Conf.*, 2017, pp. 1–6.
- [5] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *2017 IEEE Int. Conf. Robotics Automation (ICRA)*, 2017, pp. 1527–1533.
- [6] S. A. Fayazi and A. Vahidi, "Mixed-integer linear programming for optimal scheduling of autonomous vehicle intersection crossing," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 3, pp. 287–299, 2018.

- [7] M. Mukai and T. Kawabe, "Model predictive control for lane change decision assist system using hybrid system representation," in *SICE-ICASE, 2006. Int. Joint Conf.*, no. 4, 2006, pp. 5120–5125.
- [8] Y. Du, Y. Wang, and C.-Y. Chan, "Autonomous lane-change controller," in *Intell. Vehicles Symp. (IV), 2015 IEEE*, 2015, pp. 386–393.
- [9] R. A. Dollar and A. Vahidi, "Predictively coordinated vehicle acceleration and lane selection using mixed integer programming," in *ASME 2018 Dynamic Syst. Control Conf.*, 2018, pp. 1–9.
- [10] J. Hu et al., "Integrated optimal eco-driving on rolling terrain for hybrid electric vehicle with vehicle-infrastructure communication," *Transp. Res. Part C: Emerging Technol.*, vol. 68, pp. 228–244, 2016.
- [11] H. Abbas, Y. Kim, J. B. Siegel, and D. M. Rizzo, "Synthesis of Pontryagin's maximum principle analysis for speed profile optimization of all-electric vehicles," *ASME J. Dynamic Syst., Measurement, Control*, vol. 141, no. 7, pp. 071004–1 – 071004–10, 2019.
- [12] F. Mensing, E. Bideaux, and R. Trigui, "Trajectory optimization for eco-driving taking into account traffic constraints," *Transp. Res. Part D: Transp. Environment*, vol. 18, pp. 55–61, 2013.
- [13] T. Weiskircher, Q. Wang, and B. Ayalew, "Predictive guidance and control framework for (semi-) autonomous vehicles in public traffic," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 6, pp. 2034–2046, 2017.
- [14] Q. Wang, B. Ayalew, and T. Weiskircher, "Optimal assigner decisions in a hybrid predictive control of an autonomous vehicle in public traffic," in *2016 Amer. Control Conf.*, 2016, pp. 3468–3473.
- [15] R. K. Ahuja, J. B. Orlin, S. Pallottino, and M. Grazia Scutellà, "Minimum time and minimum cost-path problems in street networks with periodic traffic lights," *Transp. Sci.*, vol. 36, no. 3, pp. 326–336, 2002.
- [16] J. F. Paredes, G. P. Cazar, and M. Donkers, "A shrinking horizon approach to eco-driving for electric city buses: Implementation and experimental results," *IFAC-PapersOnLine*, vol. 52, no. 5, pp. 556–561, 2019.
- [17] R. Coulter, "Implementation of the pure pursuit path tracking algorithm," The Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, Tech. rep. CMU-RI-TR-92-01, January 1992. [Online]. Available: <http://www.dtic.mil/dtic/tr/fulltext/u2/a255524.pdf>
- [18] T. Toledo and D. Zohar, "Modeling duration of lane changes," *Transp. Res. Rec.*, vol. 1999, no. 1, pp. 71–78, 2007.
- [19] A. Ishihara, Y. Kuromaru, and M. Naka, "Steering actuator for autonomous driving and platooning," in *Proc. JSAE Annu. Congr.*, 2009, pp. 5–8.
- [20] P. Polack, F. Alché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *2017 IEEE Intell. Vehicles Symp. (IV)*. IEEE, 2017, pp. 812–818.
- [21] A. Sciarretta and A. Vahidi, *Energy-Efficient Driving of Road Vehicles*. Springer, 2020.
- [22] R. F. Hartl, S. P. Sethi, and R. G. Vickson, "A survey of the maximum principles for optimal control problems with state constraints," *SIAM review*, vol. 37, no. 2, pp. 181–218, 1995.
- [23] D. E. Kirk, *Optimal Control Theory: An Introduction*. Springer, 1970.
- [24] E. C. Kerrigan and J. M. Maciejowski, "Soft constraints and exact penalty functions in model predictive control," in *UKACC Int. Conf., Cambridge*, 2000. [Online]. Available: https://spiral.imperial.ac.uk/bitstream/10044/1/10241/6/cued_control_53.pdf
- [25] R. A. Dollar and A. Vahidi, "Efficient and collision-free anticipative cruise control in randomly mixed strings," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 4, pp. 439–452, 2018.
- [26] A. Vecchiotti, S. Lee, and I. E. Grossmann, "Modeling of discrete/continuous optimization problems: characterization and formulation of disjunctions and their relaxations," *Computers & chemical engineering*, vol. 27, no. 3, pp. 433–448, 2003.
- [27] R. A. Dollar and A. Vahidi, "Automated vehicles in hazardous merging traffic: A chance-constrained approach," *IFAC-PapersOnLine*, vol. 52, no. 5, pp. 218–223, 2019.
- [28] 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," *J. Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.
- [29] Z. Xu, X. Li, X. Zhao, M. H. Zhang, and Z. Wang, "DSRC versus 4G-LTE for connected vehicle applications: A study on field experiments of vehicular communication performance," *J. Advanced Transp.*, vol. 2017, 2017.
- [30] R. A. Dollar, A. Sciarretta, and A. Vahidi, "Multi-agent control of lane-switching automated vehicles for energy efficiency," in *2020 Amer. Control Conf. (ACC)*. IEEE, 2020, pp. 422–429.
- [31] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Rev. E*, vol. 62, no. 2, pp. 1805–1824, 2000.
- [32] M. Pourabdollah, E. Bjärkvik, F. Füller, B. Lindenberg, and K. Burgdorf, "Calibration and evaluation of car following models using real-world driving data," in *Intell. Transp. Syst. (ITSC), 2017 IEEE 20th Int. Conf. on*, 2017, pp. 1–6.
- [33] P. G. Gipps, "A model for the structure of lane-changing decisions," *Transp. Res. Part B: Methodological*, vol. 20, no. 5, pp. 403–414, 1986.
- [34] Z. Zheng, "Recent developments and research needs in modeling lane changing," *Transp. Res. Part B: Methodological*, vol. 60, pp. 16–32, 2014.
- [35] P. Hidas, "Modeling vehicle interactions in microscopic simulation of merging and weaving," *Transp. Res. Part C: Emerging Technol.*, vol. 13, no. 1, pp. 37–62, 2005.
- [36] R. A. Dollar and A. Vahidi, "Quantifying the impact of limited information and control robustness on connected automated platoons," in *IEEE Intell. Transp. Syst. Conf.*, 2017.
- [37] C. R. He, H. Maurer, and G. Orosz, "Fuel consumption optimization of heavy-duty vehicles with grade, wind, and traffic information," *ASME J. Comput. Nonlinear Dynamics*, vol. 11, no. 6, pp. 061011–1 – 061011–12, 2016.
- [38] Argonne National Laboratory. (2019, January) Autonomie. [Online]. Available: <https://www.autonomie.net/index.html>
- [39] T. Stanger and L. del Re, "A model predictive cooperative adaptive cruise control approach," in *Amer. Control Conf. (ACC), 2013*, no. 4. IEEE, 2013, pp. 1374–1379.
- [40] IBM ILOG CPLEX, "V12.7.0 documentation," 2017. [Online]. Available: <https://www.ibm.com>
- [41] Gurobi Optimization, LLC, "Gurobi optimizer reference manual," 2018. [Online]. Available: <http://www.gurobi.com>
- [42] J. M. Maciejowski, *Predictive control: with constraints*. Essex, England: Pearson education, 2002.



Robert Austin Dollar received the M.S. degree in mechanical engineering from Clemson University, South Carolina, USA in 2018 and the B.S. degree in mechanical engineering from Purdue University, West Lafayette, Indiana, USA, in 2014. He has visited IFP Energies nouvelles in France as a Chateaubriand Fellow while working toward a Ph.D. degree in mechanical engineering at Clemson University. Prior to joining Clemson University in 2016, he was a Gasoline Engine Calibration Specialist with General Motors. His current research focuses on optimal control of automated and semi-automated vehicles including connectivity and eco-driving.



Ardalan Vahidi is a Professor of Mechanical Engineering at Clemson University, South Carolina. He received his Ph.D. in mechanical engineering from the University of Michigan, Ann Arbor, in 2005, M.Sc. in transportation safety from George Washington University, Washington, DC, in 2002, and B.S. and M.Sc. in civil engineering from Sharif University, Tehran in 1996 and 1998, respectively. In 2012–2013 he was a Visiting Scholar at the University of California, Berkeley. He has also held scientific visiting positions at BMW Technology Office in California, and at IFP Energies nouvelles, in France. His research is at the intersection of energy, vehicular systems, and automatic control. His recent publications span topics in alternative vehicle powertrains, intelligent transportation systems, and connected and autonomous vehicle technologies.