**Proceedings of the ASME 2018 Dynamic Systems and Control Conference**
**DSCC2018**
**September 30-October 3, 2018, Atlanta, Georgia, USA**

# DSCC2018-9177

# PREDICTIVELY COORDINATED VEHICLE ACCELERATION AND LANE SELECTION USING MIXED INTEGER PROGRAMMING

**R. Austin Dollar** *

Department of Mechanical Engineering
Clemson University
Clemson, South Carolina 29634
Email: rdollar@clemson.edu

**Ardalan Vahidi**

Department of Mechanical Engineering
Clemson University
Clemson, South Carolina, 29634
avahidi@clemson.edu

## ABSTRACT

*Autonomous vehicle technology provides the means to optimize motion planning beyond human capacity. In particular, the problem of navigating multi-lane traffic optimally for trip time, energy efficiency, and collision avoidance presents challenges beyond those of single-lane roadways. For example, the host vehicle must simultaneously track multiple obstacles, the drivable region is non-convex, and automated vehicles must obey social expectations. Furthermore, reactive decision-making may result in becoming stuck in an undesirable traffic position. This paper presents a fundamental approach to these problems using model predictive control with a mixed integer quadratic program at its core. Lateral and longitudinal movements are coordinated to avoid collisions, track a velocity and lane, and minimize acceleration. Vehicle-to-vehicle connectivity provides a preview of surrounding vehicles' motion. Simulation results show a 79% reduction in congestion-induced travel time and an 80% decrease in congestion-induced fuel consumption compared to a rule-based approach.*

## NOMENCLATURE

- $s$ Longitudinal position of the ego vehicle along the road.
- $v$ Longitudinal velocity of the ego vehicle along the road.
- $a$ Actual longitudinal acceleration of the ego vehicle along the road.
- $l$ Actual lane of the ego vehicle. Fractional values indicate off-centerline positioning.
- $r_l$ Lane change rate of the ego vehicle over time.
- $u_1$ Commanded longitudinal acceleration of the ego vehicle.
- $u_2$ Commanded lane of the ego vehicle.
- $x$ Vehicle state vector.
- $\tau$ Time constant between commanded and actual longitudinal acceleration.
- $\xi$ Damping ratio between commanded and actual lane.
- $\omega_n$ Natural frequency between commanded and actual lane.
- $l_{veh}$ Vehicle length.
- $w_{veh}$ Vehicle width.
- $w_l$ Lane width.
- $J$ Objective to minimize.
- $\varepsilon_j$ Slack variable number $j$ for soft constraints.
- $\rho_j$ Penalty on slack variable number $j$ for soft constraints.
- $N$ Prediction horizon length in timesteps.
- $m_j$ Slope of maximum acceleration constraint $j$ in acceleration-velocity space.
- $b_j$ Acceleration offset of maximum acceleration constraint $j$.
- $\delta$ The maximum deviation, in units of lane widths, for a vehicle to remain wholly in a lane.
- $M$ A large value used in the Big M method to render constraints inactive.
- $\mu_w$ Binary variable indicating whether the ego vehicle occupies lane $w$.
- $\beta$ Binary variable for disjunctive constraints.
- $k_b$ Move blocking factor.

*Address all correspondence to this author.

## INTRODUCTION

The topic of automated ground vehicle motion planning stands poised to alter the day-to-day lives of transportation users. In 2015, the typical U.S. commuter lost 42 hours delayed in traffic according to the Texas A&M Transportation Institute's Urban Mobility Scorecard [1]. Rather than replicating this performance, automated driving could instead strive for superhuman results through optimization. The problem of optimizing longitudinal motion to alleviate traffic jams and/or reduce fuel use through drag reduction is an important topic that has received a great deal of research attention [2,3,4], but many roads including the majority of the U.S. Interstate Highway System have at least 2 lanes. In fact, the site of the recent pedestrian fatality involving an experimental self-driving vehicle [5] occurred on such a roadway. Optimal use of the lateral dimension provides an additional avenue to reduce speed fluctuations and avoid obstacles, leading to improved efficiency, safety, and travel time.

The technology to track a given path was famously demonstrated in events like the DARPA Grand Challenge [6]. Commercial lane-keeping [7, 8, 9] and lane-changing systems [10] have proved the commercial viability of relevant motion control. Such commercial systems typically rely on sensor data to deliver an immediately collision-free trajectory. However, the development of data-driven prediction models and the use of far-reaching information from connectivity could enable a more anticipative approach to 2D motion planning. In the off-road space, anticipative optimal control has been applied to path planning to not only avoid obstacles, but also to accomplish more advanced long-term objectives such as avoiding geographic traps [11]. Researchers have recently applied model predictive control (MPC) to multi-lane road driving. Weiskircher et al. [12] avoided obstacles without directly selecting a lane by approximating them as elliptic constraints. Kamal et al. [13] proposed a lane controller that reduced computation time by only considering one lane change during prediction.

The main contribution of this paper is a method of anticipatively coordinating longitudinal control and lane switching while considering the possibility of multiple future lane changes by the host. To comply with socially-accepted traffic rules, the proposed controller only issues discrete lane commands. Mixed integer quadratic programming (MIQP) is employed to handle the multi-input multi-output (MIMO) control problem with indicator variables and disjunctive constraints. Schouwenaars et al. [14] used mixed integer linear programming (MILP) for off-road path planning. Mukai et al. [15] applied similar MIP techniques to an on-road obstacle avoidance problem, but the system was simulated using a single obstacle and does not explicitly consider discrete lane expectations. Furthermore, the Mukai et al. study did not consider the possibility of a variable goal velocity. In addition to addressing these issues, this paper will also discuss a technique to reduce computation time and compare results with a rule-based algorithm.

Subsequent content is organized as follows. First, the architecture of the proposed system is introduced. The modeling approach is then presented before control design is discussed in detail. Control-related topics include the objective function, standard constraints, implementation of lane indicator variables, disjunctive collision-avoidance constraints, computation time reduction, and parameter selection. Simulation results will show performance in a multi-vehicle scenario with vehicle-to-vehicle (V2V) connectivity and one slow-moving obstacle. Finally, the paper will conclude by summarizing findings and anticipating future research on the topic.

## ARCHITECTURE

The proposed system is designed for a connected and automated car-like vehicle with control of its axle torque $TQ$ via powertrain and brakes and control of its orientation through the front wheel steering angle $\phi$. It envisions a hierarchical scheme as shown in Figure 1 where a routing algorithm determines the desired lane $l_{ref}$ and velocity $v_{ref}$ while sensors and vehicle-to-everything (V2X) connectivity provide the ego vehicle state $x_0$ and surrounding vehicle intentions $x^*$. An MPC module plans the vehicle's commanded longitudinal acceleration $u_1$ and lane $u_2$ over a finite future horizon. The latter module is the focus of this paper. It handles the tasks of smoothing the vehicle's trajectory, maintaining the desired velocity, placing the vehicle in the required lane for navigation, and avoiding collisions.

After the MPC determines the optimal trajectory, it would pass its commanded acceleration and lane to low-level motion controllers. One of these controllers would use the powertrain and brakes to deliver the commanded acceleration and the other would manipulate steering angle to track the lane. It is possible to use standard path tracking techniques, for example, pure pursuit [16], to accomplish this. Because such issues are already considered in the MPC module, the motion controller would not need to consider collision avoidance or other higher-level tasks.

## MODELING

Because of the geometric feasible region's non-convexity and the expected use of integer-valued decision variables, computation time is a potential challenge for this controller. The hierarchical approach of the previous section sacrifices the ability to optimize lane change rate to simplify the model used for MPC. Because the controller is designed to perform comfortable lane changes at higher speeds, the following linear model is used where the longitudinal and lateral dynamics are decoupled.

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

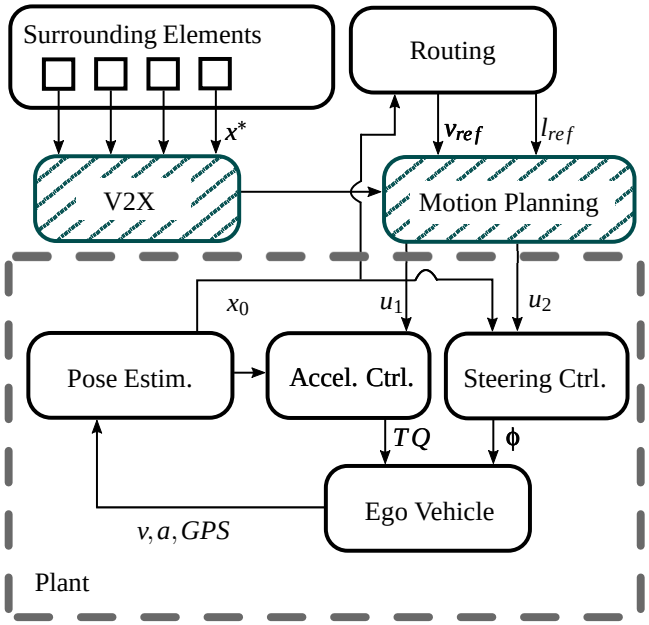


**FIGURE 1**. SCHEMATIC OF THE PROPOSED SYSTEM WITH HATCHED REGIONS HIGHLIGHTING THE FOCUS AREA.

**TABLE 1**. CONTROL MODEL PARAMETERS.

| Parameter | Unit | Value |
|---|---|---|
| $\tau$ | sec | 0.275 |
| $\xi$ | — | 1 |
| $\omega_n$ | rad/s | 1.091 |
| $K$ | — | 1 |
| $l_{veh}$ | m | 4.52 |
| $w_{veh}$ | m | 1.9 |
| $w_l$ | m | 3.7 |

Equation (1) considers the vehicle's longitudinal motion as a double integrator with time constant $\tau$ between the commanded and actual acceleration. The lateral position $l$, expressed in units of lane widths, is modeled as a critically-damped second-order system where the actual lane follows the commanded lane with a static gain $K$ of 1. This is meant to model a naturalistic lane change trajectory. The parameters of Table 1 are used in this paper. Figure 2 depicts the response of $a$ to a unit step in $u_1$ and that of $l$ to a unit step in $u_2$.

The road area that each vehicle occupies is approximated as a rectangle of fixed length $l_{veh}$ and width $w_{veh}$. Each lane also has a fixed width $w_l$. Table 1 lists these geometric parameters.

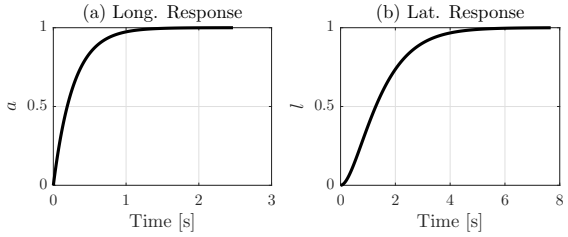


**FIGURE 2**. LONGITUDINAL ACCELERATION AND LANE POSITION STEP RESPONSES TO CONTROL INPUTS $u_1$ AND $u_2$, RESPECTIVELY.

## OPTIMAL CONTROL

The motion planner must balance the objectives of smooth driving, maintaining a target velocity, and seeking the desired lane for navigation or etiquette. Furthermore, it must trade off these goals while avoiding collisions and operating within mechanical and legal limits. In order to take advantage of any available preview information from V2X or prediction models, an MPC framework with both continuous and integer-valued decision variables is used. This section will discuss the objective function and standard constraints first. The subsections that follow concern the use of integers to indicate lane occupation and express *OR* logic for collision avoidance. Move blocking helps reduce computation time and the blocking scheme is outlined before listing the parameter values for simulation.

### Objective

The quadratic cost of Eqn. (2) is adapted from the authors' prior car-following work in [17]. As in [17], the commanded and actual accelerations are penalized after multiplication by the tuning parameter $q_a$. Unlike car-following, multi-lane planning generally involves switching between following and free-flow states during the $N$-step prediction horizon. Therefore, Eqn. (2) penalizes velocity tracking error in lieu of following distance tracking error. With an appropriately set velocity reference, this term encourages short travel times. Finally, the actual and commanded lane tracking errors are penalized. Like acceleration, velocity and lane error terms are weighted by the tuning parameters $q_v$ and $q_l$ respectively. The index $i$ in Eqn. 2 denotes the prediction timestep.

$$\begin{aligned} J = {} & q_v\left(v(N)-v_{ref}(N)\right)^2 + q_a a^2(N) + q_l\left(l(N)-l_{ref}(N)\right)^2 \\ & + \sum_{i=0}^{N-1}\Big[q_v\left(v(i)-v_{ref}(i)\right)^2 + q_a\left(u_1^2(i)+a^2(i)\right) \\ & \qquad + q_l\left(\left(u_2(i)-l_{ref}(i)\right)^2 + \left(l(i)-l_{ref}(i)\right)^2\right)\Big] \end{aligned} \tag{2}$$

Soft constraints on states ensure feasibility. The resulting slack variables add to the final cost $J_a$ as shown in Eqn. (3). This cost is minimized in the MIQP.

$$J_a = J + \sum_{j=1}^{6} \rho_j \varepsilon_j \tag{3}$$

### Standard Constraints

Several of the required constraints fit within a standard linear MPC framework with continuous variables. Equations (4a), (4b), (4c), (4d), (4e), and (4f) relate to longitudinal motion. Only forward velocity beneath a constant maximum speed limit $v_{max}$ is permitted. Acceleration and its command obey a constant minimum $u_{1,min}$ (maximum braking) and a piecewise-linear maximum as a function of velocity. Further detail on these constraints can be found in [17].

$$0 - \varepsilon_2 \leq v \leq v_{max} + \varepsilon_3 \tag{4a}$$

$$u_1 \geq u_{1,min} \tag{4b}$$

$$u_1 \leq m_1 v + b_1 \tag{4c}$$

$$u_1 \leq m_2 v + b_2 \tag{4d}$$

$$a \leq m_1 v + b_1 + \varepsilon_6 \tag{4e}$$

$$a \leq m_2 v + b_2 + \varepsilon_6 \tag{4f}$$

$$l_{min} - \delta \leq u_2 \leq l_{max} + \delta \tag{4g}$$

$$l_{min} - \delta - \varepsilon_4 \leq l \leq l_{max} + \delta + \varepsilon_5 \tag{4h}$$

$$\varepsilon_j \geq 0 \quad \forall j \tag{4i}$$

When combined with an integrality constraint on $u_2$, Eqn. (4g) limits the lane command to the roadway's usable lanes $l_{min}$ through $l_{max}$. Eqn. (4h) does the same for the actual lane position.

### Lane Indicators

Focusing on a roadway with two lanes per direction, this section develops indicator variables to flag whether or not a lane is occupied. These binary-valued variables will later enter the collision avoidance constraints to render them inactive if the ego vehicle does not occupy the relevant lane. The indicators should have the specified values in the scenarios listed in Table (2).

The following constraints result in the desired lane indicator values. Using the Big M method [18], the indicators introduce slack in lane position constraints that are otherwise infeasible under certain circumstances. For example, consider Eqn. (5a) when $\mu_1 = 0$. If the ego vehicle resides in lane 1, that is, $l < 2 - \delta$, the inequality is violated. However, if the decision variable $\mu_1 = 1$ then the inequality is satisfied for sufficiently large $M$. Thus $\mu_1$ is constrained to equal 1 when the ego vehicle resides in lane 1, as specified in Table 2.

**TABLE 2**. REQUIRED PERFORMANCE OF LANE INDICATOR VARIABLES.

| Scenario | $\mu_1$ Value | $\mu_2$ Value |
|---|---|---|
| Ego vehicle wholly resides in lane 1. | 1 | 0 |
| Ego vehicle is between lanes 1 and 2. | 1 | 1 |
| Ego vehicle wholly resides in lane 2. | 0 | 1 |

$$-l - M\mu_1 \leq -(2 - \delta) \tag{5a}$$

$$l + M\mu_1 \leq M + 2 - \delta \tag{5b}$$

$$l - M\mu_2 \leq 1 + \delta \tag{6a}$$

$$-l + M\mu_2 \leq -1 - \delta + M \tag{6b}$$

$$\mu_1, \mu_2 \in \{0, 1\} \tag{7}$$

### Collision Avoidance

Consider a one-dimensional path with a single obstacle. The drivable region in this case consists of the positions that are either in front of *OR* behind the obstacle, making it fundamentally non-convex. This constraint can be formulated for integer programming as follows.

$$s \geq s_{min} - M(1 - \beta) \tag{8a}$$

$$s \leq s_{max} + M\beta \tag{8b}$$

$$\beta \in \{0, 1\} \tag{8c}$$

In a multi-lane scenario, the constraints of Eqn. (8) should only be active if the ego vehicle occupies the lane in question. The lane indicator variables $\mu$ are introduced to deactivate the constraints when appropriate.

$$s \geq s_{min} - M(1 - \beta) - M(1 - \mu) \tag{9a}$$

$$s \leq s_{max} + M\beta + M(1-\mu) \tag{9b}$$

Applying Eqn. (9) to lanes 1 and 2, adding a slack variable $\varepsilon$, and converting to maximum form results in the following constraints for lane $\lambda$ and obstacle $\zeta$.

$$-s + M\beta_\zeta + M\mu_\lambda \leq 2M - s_{min}^{\lambda\zeta} + \varepsilon_1 \tag{10a}$$

$$s - M\beta_\zeta + M\mu_\lambda \leq s_{max}^{\lambda\zeta} + M + \varepsilon_1 \tag{10b}$$

Each potential surrounding obstacle that is reachable by the ego vehicle at a given prediction step is allocated an element for each lane in the constraint matrices. If the surrounding vehicle will occupy the lane during that step, $s_{min}^{\lambda\zeta}$ and $s_{max}^{\lambda\zeta}$ are finite. Otherwise, they are passed to the solver as infinite to render the constraint inactive.

### Move Blocking

Mixed integer quadratic programs are NP-hard and the decision form is NP-complete [19]. Therefore, care must be taken that the problem resulting from the proposed formulation can be solved sufficiently quickly in practice. A later section will show that computation time increases severely with respect to the length of prediction horizon and therefore the number of decision variables. However, a long horizon on the order of 10 s is needed to predict through multiple lane changes and a short timestep is desirable to respond quickly to disturbances. Move blocking offers a useful means of reducing the number of degrees of freedom in the optimization while maintaining the controller's predictive capability.

Move blocking constrains blocks of consecutive control inputs to be equal to one another. Standard input blocking does so in a fixed pattern. An example of this might allow a move at stage 0, but at stage 1 the move is held and a change is only allowed at every other stage. As Cagienard et al. note in [20], this strategy can significantly degrade closed-loop performance because the input sequence at $k+1$ cannot be produced by shifting the input sequence at $k$. Similarly to Cagienard et al., the proposed controller solves this problem using a time-varying blocking scheme. Figure 3 depicts the move arrangement used here. The effect is that $u_1$ and the prediction model operate on a short timestep $\Delta t$ and $u_2$ operates on a longer timestep equal to $k_b \Delta t$ where $k_b$ is an integer. This allows the controller to react quickly to a disturbance using the brakes or powertrain while reducing the number of integer-valued degrees of freedom, all within the same coordinated optimization.

### Parameter Values

Proper parameter selection is important to achieve the desired closed-loop performance. The time required to complete approximately 3 lane change maneuvers according to the plant model guides the prediction time. Then, the timestep $\Delta t$, blocking factor $k_b$, and prediction horizon $N$ are balanced to allow generous computation time margin on average (Figure 8). With the velocity weight $q_v$ fixed, the acceleration weight $q_a$ is set to obtain an occupant-acceptable peak free-flow acceleration. Finally, the lane tracking weight $q_l$ is used to reduce lane change busyness while allowing passing maneuvers. Table 3 lists the selected values.

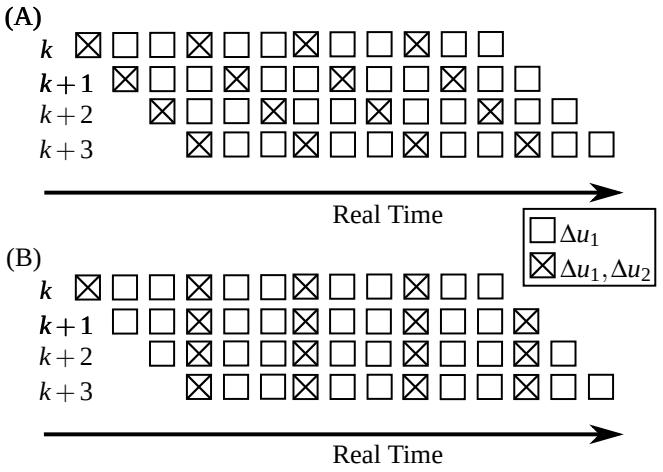


**FIGURE 3**. COMPARISON OF (A) STANDARD INPUT BLOCKING WITH (B) THE MULTI-TIMESCALE BLOCKING SCHEME USED IN THE LANE SELECTION ALGORITHM. $\Delta u$ INDICATES AN ALLOWED CHANGE IN THE CONTROL INPUT.

**TABLE 3**. CONTROLLER PARAMETERS FOR MICROSIMULATION.

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $N$ | 25 | $M$ | $10^4$ |
| $q_a$ | 300 | $\delta$ | 0.1081 |
| $q_v$ | 10 | $v_{max}$ | 36 m/s |
| $q_l$ | 10 | $u_{1,min}$ | -8.5 m/s$^2$ |
| $k_b$ | 3 | $m_1$ | 0.285 s$^{-1}$ |
| $\Delta t$ | 0.4 s | $b_1$ | 2.0 m/s$^2$ |
| $\rho_1$ | $10^7$ | $m_2$ | -0.1208 s$^{-1}$ |
| $\rho_{2\text{–}6}$ | $10^6$ | $b_2$ | 4.83 m/s$^2$ |

## BASELINE RULE-BASED ALGORITHM

A reactive rule-based algorithm assuming no connectivity is presented as a baseline. This controller uses the Intelligent Driver Model (IDM) [21] for longitudinal control. The IDM computes

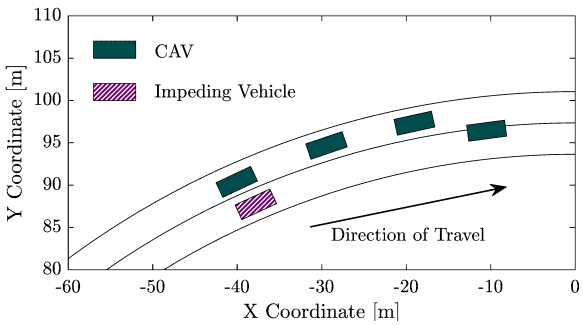


**FIGURE 4**. TRAFFIC FLOW DURING A SAMPLE SIMULATION CASE.

a desired following distance $d_{des}$ that results in an acceleration command using the ego vehicle's states, the preceding vehicle's position, and the preceding vehicle's relative velocity $\Delta v$. The safe time headway $\tau_h$, minimum distance $d_0$, acceleration exponent $\delta_a$, maximum acceleration $a_0$, and desired deceleration $b_0$ from [21] are used in the equations below to determine $u_1$.

$$d_{des} = d_0 + max\left(0, \tau_h v + \frac{v\,\Delta v}{\sqrt{4a_0 b_0}}\right) \tag{11}$$

$$u_1 = a_0\left[1 - \left(\frac{v}{v_{ref}}\right)^{\delta_a} - \left(\frac{d_{des}(v,\,\Delta v)}{d}\right)^2\right] \tag{12}$$

The lane change command is calculated using Algorithm 1. If during the course of car-following the ego vehicle's velocity drops from its reference by a small amount, a lane change out of the reference lane is triggered. The ego vehicle then monitors the *target vehicle*, that is, the nearest vehicle ahead in the desired lane. It will only return to the desired lane once the target vehicle either has an acceptable speed, vacates the desired lane, or is passed.

**Algorithm 1** RULE-BASED LANE SELECTION.

**procedure** DETERMINE LANE COMMAND
  $ltol \leftarrow 0.1$
  $vtol \leftarrow 3\,\mathrm{m/s}$
  $d_s \leftarrow 6\,\mathrm{m}$
  **if** $s_{tv} > s - l_v - d_s$ *AND* $s_{tv} < s + l_v + d_s$ **then**
    $vacant \leftarrow FALSE$
  **else** $vacant \leftarrow TRUE$

  **if** $|l-1| < ltol$ *OR* $|l-2| < ltol$ **then**
    **if** $|l - l_{ref}| > ltol$ **then**
      **if** *no target vehicle OR* $v_{tv} > v - vtol$ **then**
        **if** *vacant* **then**
          **if** $u_2(k-1) = 1$ **then return** 2
          **else return** 1
        **else return** $u_2(k-1)$
      **else return** $u_2(k-1)$
    **else**
      **if** $v < v_{ref} - vtol$ *AND* $u_1 \leq 0$ **then**
        **if** *vacant* **then**
          **if** $u_2(k-1) = 1$ **then return** 2
          **else return** 1
        **else return** $u_2(k-1)$
      **else return** $u_2(k-1)$
  **else return** $u_2(k-1)$

## MICROSIMULATION

### Scenario

Control performance is evaluated on a road with two lanes that travel in the same direction. Four CAVs begin the simulation in lane 1. Their reference lanes are always 1, but their reference velocity is either 35, 32, 29, or 26 m/s according to a full-factorial simulation plan without repetition. This results in $4! = 24$ total cases. A slow-moving open-loop vehicle that always remains in lane 1 precedes the CAVs and moves at a velocity of approximately 4.5 m/s. Real traffic elements that this approximates include disabled vehicles, construction machinery, and farm equipment. An animation visualizes the simulated vehicles' motion [22]. Figure 4 shows a sample frame of one such animation during a passing maneuver.

Results were analyzed over a distance interval $D = 2300\,\mathrm{m}$, which each of the $n_v$ vehicles running either algorithm type completed in all cases. Assuming that all vehicles travel at their preferred velocities for the duration of the trip results in an ideal congestion-free travel time $\bar{t}^*$ as shown in Eqn. (13).

$$\bar{t}^* = \frac{1}{n_v}\sum_{p=1}^{n_v}\frac{D}{v_{ref,p}} \tag{13}$$

The congestion-free travel time serves as a benchmark by showing the minimum realizable travel time subject to velocity preferences. Similarly, the fuel consumed during such constant-speed operation is also calculated and compared to each algorithm's performance in traffic.

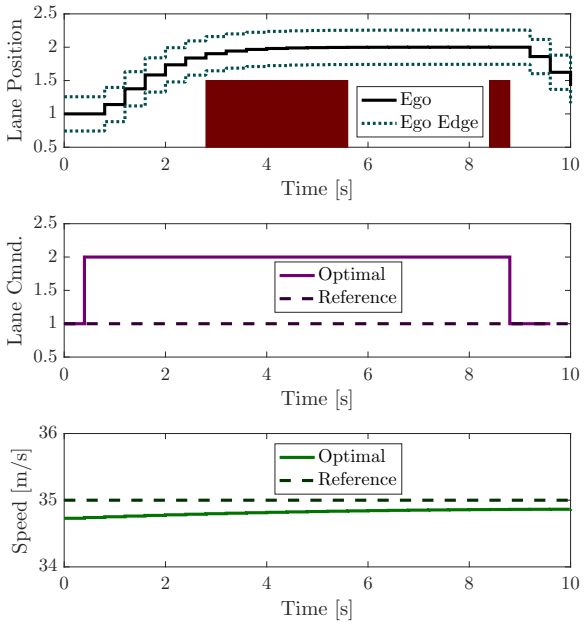


**FIGURE 5**. AN OPTIMAL PLAN TO PASS TWO VEHICLES MOVING AT DIFFERENT SPEEDS. SHADED REGIONS INDICATE OBSTRUCTED SPACE GIVEN THE OPTIMAL VELOCITY TRAJECTORY.

## Control Performance

A detailed example of the predictive algorithm's performance during a multi-vehicle interaction is shown in Figure 5. This data is taken from the fastest-moving CAV in the case where the CAVs' reference velocities were ordered 29, 35, 26, 32 m/s. Consequently, the host vehicle must pass the slow-moving impeding vehicle and a faster-moving CAV. The prediction horizon is sufficiently long to plan a collision-free path around both vehicles. The controller comprehends the necessity to remain in the passing lane longer, but still does so in order to maintain a near-ideal velocity. Notice that the first obstructed region in Figure 5 is larger than the second. This difference stems from the higher velocity and therefore longer pass duration for the CAV compared to the impeding vehicle. Although both controllers completed all 24 cases collision-free, the MPC algorithm enabled by V2V connectivity generally outperformed the rule-based one. As shown in Figure 7, the model predictive controller simultaneously resulted in reduced travel time i.e. greater average velocity *and* reduced energy consumption. This indicates a strong efficiency improvement because the fuel consumption model [17] accounts for increased aerodynamic drag as vehicle speed increases. The efficiency improvement was associated with smoother longitudinal motion, although the MPC algorithm changed lanes more actively (Figure 6). It also spent more time in the desired lane compared to the reactive algorithm, which brings advantages in navigation and potential for platoon formation.

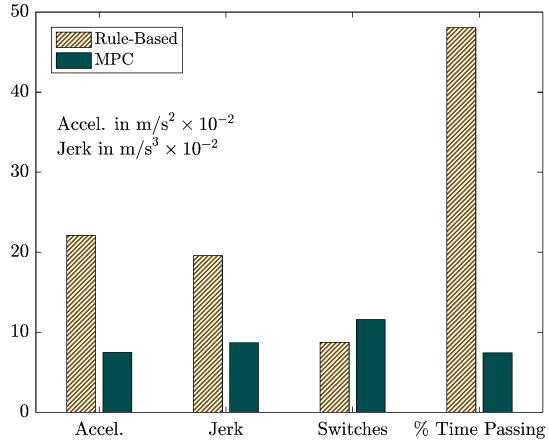


**FIGURE 6**. COMPARISON OF AGGREGATE RULE-BASED AND MPC RESULTS.

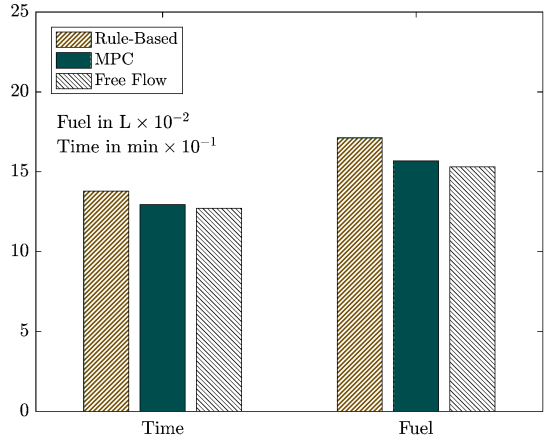


**FIGURE 7**. COMPARISON OF AGGREGATE RULE-BASED AND MPC RESULTS TO THE IDEAL FREE-FLOW CASE.

In Figure 6, *% Time Passing* indicates the percentage of time when a vehicle was outside of its desired lane while *Switches* indicates the number of lane changes. The *Fuel* metric in Figure 7 is expressed as the total fuel consumed by the average vehicle over the course of one case.

Quantitatively, the MPC algorithm reduced fuel consumption by 8.4 % and travel time by 6.2 %. Excess travel time, defined as the increase in travel time over the congestion-free value from Eqn. (13), decreased by 79 % relative to the reactive algorithm. Correspondingly, excess fuel consumption was reduced by 80 % compared to a baseline of 18.1 mL for the average vehicle. The ideal travel time $\overline{t^*}$ was 76.34 s with the MPC managing a mean excess travel time of only 1.34 s due to congestion.

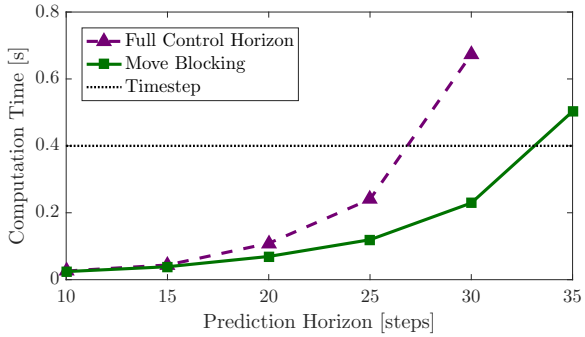


**FIGURE 8**. MEAN COMPUTATION TIME DURING A SAMPLE REFERENCE VELOCITY CASE.

### Computation Time

To prepare for real-time implementation, the impact of prediction horizon and move blocking on computation time is presented. As in standard continuous-variable MPC, increasing the prediction horizon monotonically increases mean computation time. Move blocking on the integer-valued second control input was found to effectively reduce computation time and enable a longer prediction horizon for a given sampling time (Figure 8).

The simulation case where target velocities increased from front to rear in the initial vehicle formation was used to measure computation time. The hardware platform was a laptop PC equipped with a 4 core, 2.7 GHz CPU and 16 GB RAM.

## CONCLUSION AND FUTURE RESEARCH

Using mixed integer quadratic programming to handle the necessary disjunctions and indicator variables, the algorithm proposed herein coordinates longitudinal vehicular motion with optimal lane planning. Preview information from vehicle-to-vehicle connectivity enables each host vehicle to optimize its trajectory around moving obstacles to minimize perturbation from the reference velocity. Computation time and a technique for its reduction were discussed. In simulated comparisons to a reactive rule-based algorithm, the model predictive controller resulted in more efficient travel times while simultaneously reducing fuel consumption. Longitudinal motion was also significantly smoother with a moderate increase in the number of lane changes.

The fundamental techniques described here open expansive opportunities for future research. Further developments will generalize the system to accommodate roadways with an arbitrary number of lanes and enable safe interaction with unconnected vehicles. Probability modeling with online adaptation presents an alternative to directly communicated position trajectories in the latter case. Error in the resulting previews could motivate the use of chance constraints in MPC.

In addition to generalizing the algorithm, opportunity exists to improve the simulation platform as well. While performance results are promising when the controller's internal plant model is used for simulation, the model approximates the performance of nonlinear mechanical systems and low-level controllers. Therefore, future simulations should include these controllers interacting with a nonlinear kinematic model for steering, such as the popular bicycle model. The authors' group is currently integrating the predictive algorithm with the traffic simulation software VISSIM. The resulting package will allow evaluation of large-scale scenarios with hundreds of vehicles, thereby improving confidence in the realism of fuel and time savings.

## ACKNOWLEDGMENT

This research was supported by an award from the U.S. Department of Energy Vehicle Technologies Office (Project No. DE-EE0008232). The authors also thank Prof. Beshah Ayalew of Clemson University's Department of Automotive Engineering for his role in inspiring this portion of the project.

## REFERENCES

[1] Schrank, D., Eisele, B., Lomax, T., and Bak, J., 2015. 2015 urban mobility scorecard. Technical report, Texas A&M Transportation Institute and INRIX, Kirkland, WA and College Station, TX, August. See also URL `https://static.tti.tamu.edu/tti.tamu.edu/documents/mobility-scorecard-2015.pdf`.

[2] Kamal, M. A. S., Imura, J., Hayakawa, T., Ohata, A., and Aihara, K., 2014. "Smart driving of a vehicle using model predictive control for improving traffic flow". *IEEE Trans. Intelligent Transportation Syst.*, ***15***(2), pp. 878–888.

[3] Alam, A. A., Gattami, A., and Johansson, K. H., 2010. "An experimental study on the fuel reduction potential of heavy duty vehicle platooning". In 13th International Conf. on Intelligent Transportation Syst., IEEE, pp. 306–311.

[4] Stern, R. E., Cui, S., Monache, M. L. D., Bhadani, R., Bunting, M., Churchill, M., Hamilton, N., Haulcy, R., Pohlmann, H., Wu, F., Piccoli, B., Seibold, B., Sprinkle, J., and Work, D. B., 2017. "Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments". *arXiv preprint arXiv:1705.01693*, ***15***(2), pp. 878–888.

[5] Griggs, T., and Wakabayashi, D., 2018. How a self-driving Uber killed a pedestrian in Arizona. On the WWW, March. URL `https://www.nytimes.com/interactive//2018/03/20/us/self-driving-uber-pedestrian-killed.html`.

[6] S. Thrun et al., 2006. "Stanley: The robot that won the DARPA Grand Challenge". *J. of Field Robotics*, ***23***(9), pp. 661–692.

[7] Mercedes-Benz, 2018. Active lane keeping assist. On the WWW. URL `https://www.mbusa.com/mercedes/technology/videos/detail/title-safety/`

videoId-e84b9423c67a7410VgnVCM100000ccec1e35RCRD.

[8] Toyota, 2018. Lane keeping assist. On the WWW. URL http://www.toyota-global.com/innovation/safety_technology/safety_technology/technology_file/active/lka.html.

[9] Acura, 2018. Lane keeping assist system. On the WWW. URL https://www.acura.com/mdx/modals/lane-keeping-assist-system.

[10] Bradley, R., 2016. Tesla Autopilot. On the WWW. URL https://www.technologyreview.com/s/600772/10-breakthrough-technologies-2016-tesla-autopilot/.

[11] Goswami, A., 2017. "Hierarchical Off-Road Path Planning and its Validation using a Scaled Autonomous Car". MS Thesis, Clemson University, Clemson , SC, December. See also URL https://tigerprints.clemson.edu/cgi/viewcontent.cgi?article=3800&context=\all_theses.

[12] Weiskircher, T., Wang, Q., and Ayalew, B., 2017. "Predictive guidance and control framework for (semi-) autonomous vehicles in public traffic". *IEEE Trans. on Control Syst. Techn.,* ***25***(6), pp. 2034–2046.

[13] Kamal, M. A. S., Taguchi, S., and Yoshimura, T., 2016. "Efficient driving on multilane roads under a connected vehicle environment". *IEEE Trans. on Intelligent Transportation Syst.,* ***17***(9), pp. 2541–2551.

[14] Schouwenaars, T., Moor, B. D., Feron, E., and How, J., 2001. "Mixed integer programming for multi-vehicle path planning". In European Control Conference (ECC), IEEE, pp. 2603–2608.

[15] Mukai, M., Murata, J., Kawabe, T., Nishira, H., Takagi, Y., and Deguchi, Y., 2008. "Optimal path generation for automotive collision avoidance using mixed integer programming". *SICE J. of Control, Measurement, and Syst. Integration,* ***1***(3), pp. 222–226.

[16] Coulter, R. C., 1992. Implementation of the pure pursuit path tracking algorithm. Technical report, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, January. See also URL http://www.dtic.mil/dtic/tr/fulltext/u2/a255524.pdf.

[17] Dollar, R. A., and Vahidi, A., 2017. "Quantifying the impact of limited information and control robustness on connected automated platoons". In 20th International Conf. on Intelligent Transportation Syst., IEEE, pp. 1–7.

[18] Vecchietti, A., Lee, S., and Grossmann, I. E., 2003. "Modeling of discrete/continuous optimization problems: characterization and formulation of disjunctions and their relaxations". *Computers & chemical engineering,* ***27***(3), pp. 433–448.

[19] Pia, A. D., Dey, S. S., and Molinaro, M., 2017. "Mixed integer quadratic programming is in NP". *Mathematical Programming,* ***162***(1-2), pp. 225–240.

[20] Cagienard, R., Grieder, P., Kerrigan, E. C., and Morari, M., 2007. "Move blocking strategies in receding horizon control". *J. of Process Control,* ***17***(6), pp. 563–570.

[21] Treiber, M., Hennecke, A., and Helbing, D., 2000. "Congested traffic states in empirical observations and microscopic simulations". *Physical Review E,* ***62***(2), p. 1805.

[22] Dollar, R. A., 2018. MPC lane change animation. On the WWW. URL https://youtu.be/3qgmSQ9kkxk.