May 2021

# Efficient Automated Driving Strategies Leveraging Anticipation and Optimal Control 

Robert Austin Dollar<br>Clemson University, adollarnc@hotmail.com

Follow this and additional works at: https://tigerprints.clemson.edu/all_dissertations

## Recommended Citation <br> Dollar, Robert Austin, "Efficient Automated Driving Strategies Leveraging Anticipation and Optimal Control" (2021). All Dissertations. 2804. <br> https://tigerprints.clemson.edu/all_dissertations/2804

This Dissertation is brought to you for free and open access by the Dissertations at TigerPrints. It has been accepted for inclusion in All Dissertations by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.

# EFFICIENT AUTOMATED DRIVING STRATEGIES LEVERAGING ANTICIPATION AND OPTIMAL CONTROL 

| A Dissertation Presented to the Graduate School of Clemson University |
| :---: |
| In Partial Fulfillment of the Requirements for the Degree <br> Doctor of Philosophy <br> Mechanical Engineering |
| by Robert Austin Dollar May 2021 |
| Accepted by: <br> Dr. Ardalan Vahidi, Committee Chair <br> Dr. John Wagner <br> Dr. Phanindra Tallapragada <br> Dr. Yunyi Jia <br> Dr. Antonio Sciarretta (Guest) |


#### Abstract

Automated vehicles and advanced driver assistance systems bring computation, sensing, and communication technologies that exceed human abilities in some ways. For example, automated vehicles may sense a panorama all at once, do not suffer from human impairments and distractions, and could wirelessly communicate precise data with neighboring vehicles. Prototype and commercial deployments have demonstrated the capability to relieve human operators of some driving tasks up to and including fully autonomous taxi rides in some areas. The ultimate impact of this technology's large-scale market penetration on energy efficiency remains unclear, with potential negative factors like road use by empty vehicles competing with positive ones like automatic eco-driving. Fundamentally enabled by historic and look-ahead data, this dissertation addresses the use of automated driving and driver assistance to optimize vehicle motion for energy efficiency.

Facets of this problem include car following, co-optimized acceleration and lane change planning, and collaborative multi-agent guidance. Optimal control, especially model predictive control, is used extensively to improve energy efficiency while maintaining safe and timely driving via constraints. Techniques including chance constraints and mixed integer programming help overcome uncertainty and non-convexity challenges. Extensions of these techniques to tractor trailers on sloping roads are provided by making use of linear parameter-varying models. To approach the wheel-input energy eco-driving problem over generally shaped sloping roads with the computational potential for closed-loop implementation, a linear programming formulation is constructed. Distributed and collaborative techniques that enable connected and automated vehicles to accommodate their neighbors in traffic are also explored and compared to centralized control. Using simulations and vehicle-in-the-loop car following experiments, the proposed algorithms are benchmarked against others that do not make use of look-ahead information.


## Dedication

For my grandfather Robert Dollar, who laid the foundation.

## Acknowledgments

Much of this research would not have been possible without the cohesive efforts and support of a larger team. In particular, thank you to Tyler Ard, Ali Fayazi, Longxiang Guo, Nathan Goulet who made the vehicle-in-the-loop experiments possible with their multitude of moving parts. Our collaborator Beshah Ayalew also deserves special acknowledgment for his proposal of the lane change research that led to several of the publications summarized in this dissertation. The project that brought the team together was supported by the U.S. Department of Energy Vehicle Technologies Office (Project No. DE-EE0008232).

Chapter 6 refers to two conference papers that are based upon research supported by the Chateaubriand Fellowship of the Office for Science \& Technology of the Embassy of France in the United States. As supervisor, Antonio Sciarretta added advanced optimal-control insight and precision to these efforts. IFP Energies nouvelles graciously hosted the author, who would have been rather lost without the team's support in adjusting to life in France.

At a workshop held at IFP, Ardalan Vahidi, Gábor Orosz, and the author began to discuss a collaboration on use of information from multiple leading vehicles in cruise control. This research resulted in an accepted conference paper that is summarized in Chapter 3. Contributions and helpful critique from Orosz and Tamás Molnár at University of Michigan were indispensable to that research.

The tractor-trailer research in Chapters 4 and 5 sprang out of a collaboration with Cummins. In addition to technical contributions like vehicle parameter values, the input of Ali Borhan, Bibin Pattel, and Jingxuan Liu steered the project in an industrially relevant direction overall.

Finally, Ardalan Vahidi served as an exemplary advisor who fostered a collegial laboratory environment that helped convince the author to pursue the doctorate. He has been co-author on all of the publications that this dissertation draws from and contributed not only technical and communication ideas, but also encouragement and confidence.

## Table of Contents

Page
Title Page ..... i
Abstract ..... ii
Dedication ..... iii
Acknowledgments ..... iv
List of Tables ..... vii
List of Figures ..... viii
1 Introduction ..... 1
1.1 State of the Art ..... 2
1.2 Technical Background ..... 9
1.3 Contributions ..... 15
2 Single-Predecessor Car Following ..... 18
2.1 Base Control ..... 19
2.2 Non-Convex Constraints ..... 21
2.3 Probability Modeling ..... 21
2.4 Results of Optimal Car-Following ..... 25
3 Multi-Predecessor Car Following ..... 30
3.1 Scenario and Assumptions ..... 31
3.2 Modeling ..... 31
3.3 Optimal Controller ..... 33
3.4 Performance Assessment ..... 36
3.5 Results ..... 37
4 Optimal Lane-Change Decision Making ..... 40
4.1 Modeling ..... 41
4.2 Hierarchical Control Using Pontryagin's Minimum Principle ..... 48
4.3 Receding Horizon Control Using Mixed Integer Programming ..... 51
4.4 Extension to Tractor-Trailers ..... 53
4.5 Chance Constraints ..... 55
4.6 Results ..... 56
5 Advanced Long-Term Speed Planning ..... 61
5.1 Performance Requirements ..... 62
5.2 Architecture ..... 62
5.3 Core Formulation ..... 62
5.4 Speed Constraints ..... 68
5.5 Initialization and Iteration ..... 70
5.6 Simulation Methods ..... 73
5.7 Results ..... 79
5.8 Discussion ..... 85
6 Collaborative Guidance ..... 90
6.1 Car Following ..... 91
6.2 Multi-Lane Guidance ..... 96
7 Summary ..... 103
7.1 Accomplishments and Contributions ..... 103
7.2 Limitations ..... 105
7.3 Recommendations for Further Research ..... 105
A Energy Modeling ..... 107
A. 1 Internal Combustion Engine Vehicles ..... 107
A. 2 Electric Vehicles ..... 109
Bibliography ..... 111

## List of Tables

Table Page
2.1 As-calibrated computation time for car following. ..... 27
2.2 Experimental performance comparison, VIL (Mazda, combustion) [6]. ..... 29
2.3 Experimental performance comparison, VIL (Nissan, electric) [6]. ..... 29
3.1 IDM parameters in multi-predecessor car following. ..... 33
3.2 Wheel-input energy and headway results. ..... 37
4.1 Linear model parameters. ..... 43
4.2 Block diagram nomenclature ..... 49
4.3 Change relative to IDM-RB in multi-lane guidance. ..... 60
4.4 Change relative to IDM-RB in the arterial scenario at various CAV fractions. ..... 60
5.1 Performance targets. ..... 62
5.2 Truck parameter sets. ..... 74
5.3 IDM parameters in the tractor-trailer simulations. ..... 78
5.4 Speed planner assembly time under various settings ..... 82
5.5 Speed planner solution time using various settings. ..... 82
5.6 Speed planner solution time for the Lanesville-Siberia route with 50 m steps ..... 82
6.1 Acronyms for single-lane multi-agent algorithms. ..... 94
6.2 Modified IDM and MOBIL Parameters ..... 101
6.3 MATLAB computation time for collaborative algorithms. ..... 102
A. 1 Powertrain model results. ..... 108
A. 2 Powertrain model constants. ..... 109

## List of Figures

Figure Page
2.1 Mean PV u residuals of the prediction algorithm after one cycle. ©2018 IEEE. ..... 22
2.2 The sample road segment for the Tiger Commute prediction study. ..... 23
2.3 Block diagram of the Markov chain applied to the Tiger Commute data. ..... 23
2.4 Accuracy results of Markov chain and constant effective velocity predictions. ..... 24
2.5 Markov chain and constant velocity RMSE at various prediction horizons. ..... 24
2.6 Speed histogram used in the Markov chain, with discretization boundaries. ..... 25
2.7 Markov chain RMSE benefit vs. constant speed across the velocity levels marked in Fig. 2.6. ..... 25
2.8 Fuel economy at various string compositions. ©2018 IEEE. ..... 26
2.9 Space utilization at various string compositions. ©2018 IEEE. ..... 26
2.10 VISSIM fuel savings. Figure by Tyler Ard [5], used with permission. ..... 27
2.11 Schematic of a VIL system. The results shown here used one real vehicle. ..... 28
2.12 The Nissan Leaf test vehicle (a) and its pedal actuators (b). ..... 29
3.1 The three predecessors and the CAV to be controlled. ©2021 IEEE. ..... 31
3.2 Block diagram of the multi-predecessor car following controller. ..... 34
3.3 The time horizons used for multi-predecessor detection and prediction. ©2021 IEEE. ..... 34
3.4 The experimental trajectories of the human-driven predecessors. ©2021 IEEE. ..... 36
3.5 Simulated trajectories of each algorithm in headway-speed space. ©2021 IEEE ..... 38
3.6 The identified number of hidden vehicles over time. ©2021 IEEE. ..... 39
4.1 Lane step responses at (top to bottom) $24.3 \mathrm{~m} / \mathrm{s}, 7.7 \mathrm{~m} / \mathrm{s}$, and $3.9 \mathrm{~m} / \mathrm{s}$. ©2021 IEEE. ..... 43
4.2 Various models of lateral tractor motion. ..... 44
4.3 Diagram of the trailer including the hitch point $H$ and rear point $R$. ..... 45
4.4 Models of lateral trailer motion. ..... 46
4.5 Least-squares road grade approximation for receding horizon control. ..... 47
4.6 Block diagram of the hierarchical automated driving system. ..... 49
4.7 An example of the online-linearized truck acceleration constraint. ..... 54
4.8 Safe distance as a function of prediction step and robustness [32]. ..... 56
4.9 Comparison of aggregate rule-based and MPC results to the ideal free-flow case [31]. ..... 57
4.10 Mean computation time during a sample reference velocity case [31]. ..... 57
4.11 Approximate road geometry with the AV (solid) and surrounding vehicles (hatched) [32]. ..... 58
4.12 Frequency of AV-involved collisions with $95 \%$ confidence intervals [32]. ..... 59
4.13 Position plans including the ego's (green), with the ego's longitudinal plan. ©2021 IEEE ..... 59
4.14 Speed profiles and 20 m space-averages in the arterial scenario. ©2021 IEEE. ..... 60
5.1 Block diagram of the LP speed planner's integration with obstacle avoidance. ..... 63
5.2 Block diagram of the multi-predecessor car following controller. ..... 65
5.3 The expected maximum weaving speed, shown as functions of headway and density. ..... 70
5.4 The first 5 LP solutions in a closed-loop simulation. ..... 71
5.5 The shortened 6 km scenario (a) and the full Lanesville-Siberia route (b). ..... 75
5.6 The simulation window in a sparse traffic scene during a CAV lane change. ..... 76
5.7 Trajectories for MPC and the baseline MOBIL in the 6 km scenario. ..... 80
5.8 Fuel consumption and speed in the 6 km scenario with $90 \%$ confidence intervals. ..... 81
5.9 Example comparison of the microscopic traffic conditions to the macroscopic targets. ..... 81
5.10 Closed-loop trajectories over the Lanesville-Siberia scenario using parameter set A. ..... 83
5.11 Closed-loop trajectories over the Lanesville-Siberia scenario using parameter set B. ..... 84
5.12 Energy, fuel consumption, and speed, Lanesville-Siberia, parameter set A, $95 \%$ confidence. ..... 85
5.13 Energy, fuel consumption, and speed, Lanesville-Siberia, parameter set B, $95 \%$ confidence. ..... 85
5.14 Energy loss breakdown for the two truck parameter sets. ..... 88
6.1 The drive cycles for single-lane multi-agent simulations [26]. ..... 93
6.2 Energy use and string length in WLTC High and Low cycles [26]. ..... 94
6.3 Mean absolute acceleration at various string indices [26] ..... 95
6.4 Velocity trajectories in the WLTC Low Cycle [26]. ..... 95
6.5 Collaborative vehicles navigate the case study's scenario. ©2020 IEEE. ..... 100
6.6 Velocity trajectories resulting from the three presented algorithms. ©2020 IEEE ..... 101
6.7 The DSP priority order's evolution. ©2020 IEEE. ..... 102
6.8 MOBIL, decentralized, collaborative, and centralized performance. ©2020 IEEE. ..... 102
A. 1 Passenger vehicle gear map from Dollar and Vahidi [30]. ©2018 IEEE. ..... 109
A. 2 Heavy vehicle gear map from Dollar and Vahidi [30]. ©2018 IEEE. ..... 109
A. 3 Battery output power from Dollar et al. [27]. ©2020 IEEE ..... 110

## Chapter 1

## Introduction

Driving style is widely acknowledged as a strong factor in multiple transport outcomes, for example, energy efficiency, emissions, safety, comfort, and timeliness. This dissertation seeks to answer the question: how can automated vehicles use their computation, sensing, and communication capabilities to improve transport performance by anticipating a traffic scene's future evolution? Driven by the social need to accomplish freedom of movement while conserving limited environmental resources and promoting energy independence, special attention is paid to energetic and ecological outcomes. Transport performance in this context not only refers to the individual vehicle but also considers indirect impacts via micro- and macroscopic traffic interactions.

The motivation is clear in terms of benefits for humanity. Automated driving is projected to become increasingly common [125], which could provide individual transport to people like some elderly who cannot currently enjoy manual driving [18]. Unfortunately, the performance of current road transport could struggle as new users enter the roadway and cost is reduced for existing users [130]. A 2019 study found that in 2017, U.S. road users typically lost 54 hours per year in traffic [108]. These traffic jams also involve wasted energy, contributing to increasing petroleum use [15] despite increasingly stringent vehicle fuel economy standards [90]. While the economic, environmental, and political benefits of reducing energy consumption in internal combustion engines attract ample attention, electric vehicles (EVs) can also benefit from reduced energy consumption. Range is currently a perceived disadvantage of EVs compared to combustion engine-powered vehicles due to both battery and charging infrastructure limitations [97]. Lower-energy transport systems could soften this drawback and encourage greater adoption of EVs. The future impact of automated driving on energy consumption, however, is still uncertain. Wadud et al. [130] predicted that energy consumption could
be approximately doubled or halved depending on the future implementation and use practices of automated vehicles. Currently, several commercial adaptive cruise control systems have been shown to be string unstable [42], a harmful property for energy efficiency in traffic. The control techniques explored in this dissertation seek a more favorable outcome by maximizing energy benefits from congestion mitigation and ecological driving.

Of course, the costs of transport shortcomings are not limited to energy. Collisions cost the U.S. economy an estimated $\$ 242$ billion per year according to a 2010 NHTSA report revised in 2015 (the latest) [12] and over $90 \%$ of them are caused by driver-related factors ranging from perception error to car control to drowsiness [112]. More importantly, 36,560 people in the U.S. died in motor vehicle collisions in 2018 [58]. An early study of prototype automated vehicles [37] found that they were rear-ended about twice as often as their human-driven counterparts. Since then, several severe and sometimes fatal collisions involving automated or partially-automated vehicles have captured the public's attention via the lay media [41] [94] [115]. Therefore, this research pays close attention to collision avoidance even under uncertainty.

The following section will first review existing automated driving technologies. Then, more detailed technical background is provided as necessary to support this dissertation research. The chapter closes with a summary of the contributions.

### 1.1 State of the Art

Automated driving is a highly active research topic in several disciplines and includes a multitude of subtopics either closely or tangentially related to this research. This review will outline some of the main categories and delve into more detail about particularly relevant ones.

### 1.1.1 Control Paradigms

Diverse overall paradigms have been used in automated driving and each involves a different set of subsystems. The three categories considered here are end-to-end, modular machine learning, and modelbased.

### 1.1.1.1 End-to-End Approaches

End-to-end learning seeks to control the vehicle based on sensory data directly [117]. This typically involves training a deep neural network to map data from camera, lidar, etc. to steering, acceleration and
braking commands. Advantages of end-to-end learning include simplicity of application and adaptability. As more data becomes available, driving performance can improve without additional algorithm development. The most oft-cited drawback of end-to-end learning is that it lacks physically-interpretable intermediate results [19] and performance of complex neural network models is difficult to explain [138] and guarantee [86]. The large initial data requirement is another disadvantage [21], at least early in development. This factor would have increased the difficulty of using end-to-end learning here. Moreover, most end-to-end systems in the reinforcement learning category seek movement toward a goal and collision avoidance while those in the imitation learning category seek human-like driving [117]. This dissertation seeks energy efficiency, putting it in a less-explored area if end-to-end learning were used. It should be noted that some examples of reinforcement learning for energy minimization do exist in the literature; see [104] on the related problem of hybrid vehicle energy management and [74] on a model-based reinforcement learning approach to eco-driving.

### 1.1.1.2 Modular Machine Learning

Other machine learning approaches may still utilize data-driven techniques like neural networks, but do so in several layers that produce more interpretable results [117]. For example, a convolutional neural network could help identify obstacles from sensor data [84] and a separate neural network could then combine all surrounding obstacles with the ego vehicle's state to arrive at control inputs [35]. While the performance of the individual modules still depends on the available data, separation of tasks into modules helps pinpoint the source of malfunctions. The complexity of development can exceed that of end-to-end learning, however [19]. It is also possible to mix machine learning and model-based elements. For example, [137] uses deep reinforcement learning to track waypoints that are set by a model-based planner.

### 1.1.1.3 Model-Based Approaches

In contrast with machine learning, model-based approaches primarily rely on prior physical knowledge of the vehicle and/or the surroundings to accomplish automated driving. This reduces the amount of data that the system initially requires to operate. It also enables the controller to compute a solution that is optimal in some physical sense (e.g. minimum-acceleration, minimum energy) from the outset rather than gradually improving performance through a trial-and-error process. Hence, this energy-focused research takes a modelbased approach. The main drawback is lack of flexibility to operation in new domains that the control engineer may not have considered, which may limit this approach to SAE Level 4 [106] and below without an artificial intelligence-based supervisor. Examples include the trajectory planning research in [134] and the modular
system successfully deployed in the DARPA Grand Challenge [119].

### 1.1.2 Perception

Automated vehicles and those equipped with advanced driver assistance systems (ADAS) use several sensing technologies to measure the host vehicle and its environment [126]. This research will depend on several of these measurements, so a brief review is appropriate. The host vehicle's speed is obtained from speed sensors in the wheel or drivetrain and its acceleration is measured using an inertial measurement unit (IMU). A global positioning system (GPS) can roughly measure absolute position with meter-level accuracy, although real-time kinematic (RTK) GPS can be used to obtain centimeter-level accuracy in a test track environment.

Particularly for adaptive cruise control (ACC) systems, the position and speed of surrounding vehicles can be measured using camera and radar systems. Higher levels of automation often use lidar to provide a more detailed, 360-degree view of the surroundings. Multiple sensors are often used together to produce more reliable estimates via sensor fusion [72]. The field of simultaneous localization and mapping (SLAM) [17] is devoted to finding the position of the host vehicle among measured obstacles in the absence of high-accuracy GPS.

It is worth mentioning that the autonomous-vehicle perception technology is highly complex, safetycritical, and subject to random failures depending on environmental factors like weather conditions [99]. Although this dissertation focuses on motion planning and control including safety considerations, this emphasis should not be taken to imply that all autonomous vehicle collisions are caused by motion planning and control decisions. For example, the well-known collision in [118] was caused by a perception fault.

### 1.1.3 Motion Control and Planning

This topic refers to processing involved in driving a vehicle along a predetermined route. Various approaches from existing literature dealing with both lateral and longitudinal control will be reviewed. The broad categories of these approaches include classical control, analytical optimal control, numerical optimal control, and reinforcement learning.

### 1.1.3.1 Classical Control

A class of motion control techniques are deemed "classical" because of their reliance on current state feedback and explicit, model-based calculations. A few such techniques are reviewed here.

Adaptive cruise control (ACC) has been implemented using velocity and gap feedback. Ntousakis et al. [93] provide the following typical example.

$$
\begin{gather*}
v_{e}=v-v_{d}  \tag{1.1a}\\
a_{s c}=\max \left\{\min \left\{-0.4 v_{e}, 2\right\},-2\right\}  \tag{1.1b}\\
d_{e}=d-d_{d}, \quad d_{d}=T v  \tag{1.1c}\\
u_{c}= \begin{cases}a_{s c} & \text { speed control } \\
\max \left\{\min \left\{\dot{d}+0.25 d_{e}, a_{s c}\right\},-2\right\} & \text { gap control } \\
u=\min \left\{u_{c}, \bar{u}(v)\right\}\end{cases} \tag{1.1d}
\end{gather*}
$$

The velocity error $v_{e}$, equal to the difference between the desired velocity $v_{d}$ and the ego vehicle's velocity $v$, is multiplied by a gain to compute the free-flow acceleration $a_{s c}$. In gap control mode, however, the gap $d$ between the ego and preceding vehicle ( PV ) also plays a role. The deviation $d_{e}$ from the desired gap $d_{d}$, is multiplied by a gain and used to compute the control move $u_{c}$. The desired gap itself depends on speed to target a time headway $T$. Since minimum and maximum constraints, including potentially complex mechanical limits denoted $\bar{u}(v)$, are not explicitly included in the controller, saturation is needed afterward. Controllers of this general type can be practical; [114] showed damping of traffic waves in a ring road experiment by introducing just one automated vehicle.

Connected cruise control can also be implemented using classical control techniques. Milanés et al. [87] obtain the preceding vehicle's control action via connectivity and use it as a feedforward to improve control performance. He et al. [50] use connectivity to track the speeds of multiple vehicles ahead, a topic explored further in Chapter 3.

Another longitudinal control law called the Intelligent Driver Model (IDM) [120] was proposed for human driver modeling. Required parameters not shared with the ACC above include minimum distance $d_{0}$,
maximum acceleration $a_{0}$, and comfortable deceleration $b_{0}$.

$$
\begin{gather*}
d_{d e s}=d_{0}+\max \left(0, T v+\frac{v \dot{d}}{\sqrt{4 a_{0} b_{0}}}\right)  \tag{1.2}\\
a_{\text {des }}=a_{0}\left[1-\left(\frac{v}{v_{d}}\right)^{\delta}-\left(\frac{d_{d e s}(v, \dot{d})}{d}\right)^{2}\right] \tag{1.3}
\end{gather*}
$$

Classical state-feedback techniques can also be applied to lateral motion control to achieve line tracking. A popular example is the pure pursuit controller [22], which geometrically computes the steering angle necessary to drive the center of the rear axle to a lookahead point. This lookahead point lies a distance $l_{d}$ away from the center of the rear axle, where $l_{d}$ can be chosen as a linear function of vehicle speed i.e. $l_{d}=k v$. Dollar and Vahidi [33] implemented the pure pursuit controller as

$$
\begin{equation*}
\phi=\arctan \frac{2 L \sin \left(\arcsin \frac{y_{d}-y_{a}}{l_{d}}-\psi\right)}{l_{d}} \tag{1.4}
\end{equation*}
$$

where the lookahead distance is

$$
\begin{equation*}
l_{d}=\max \left\{k v, \sqrt{\left(y_{d}-y_{a}\right)^{2}+(1.5 L)^{2}}\right\} \tag{1.5}
\end{equation*}
$$

and $L, y_{d}, y_{a}, \psi$, and $\phi$ denote the vehicle length, lateral position of the lookahead point, actual lateral position of the rear axle, yaw angle, and steering angle of the controlled vehicle, respectively.

In a structured environment with discrete lanes, rule-based algorithms can help extend explicit, statefeedback car following controllers to include lane change decisions. One such algorithm is MOBIL [69], which compares the current accelerations of the ego and nearby vehicles to their potential values if the ego vehicle changes lanes. A lane change is executed if

$$
\begin{equation*}
\tilde{a}_{c}-a_{c}+p\left(\tilde{a}_{n}-a_{n}+\tilde{a}_{o}-a_{o}\right)>\Delta a_{\mathrm{th}} \tag{1.6}
\end{equation*}
$$

where $a_{c}$ is the ego vehicle's acceleration, $a_{n}$ is the acceleration of the ego's follower after the potential lane change, $a_{o}$ is the acceleration of the ego's current follower, $\Delta a_{\mathrm{th}}$ is a hysteresis threshold to prevent toggling, and the accent tilde indicates the potential value after a lane change by the ego. An acceleration bias can be added to the right-hand side to induce a keep-right or keep-left preference. The politeness factor $p$ controls
how much the controller prioritizes surrounding vehicles compared to itself. Other rule-based lane-change algorithms also exist. For an early example, see the more complex decision process proposed by Gipps [40].

### 1.1.3.2 Analytical Optimal Control

In some cases, an optimal control can be determined analytically. This is desirable because it yields optimal performance with minimal computation time, especially when compared to numerical techniques. It also provides clear insight into what behavior results in good control, which can be used by manual operators or simpler rule-based controllers. Pontryagin's Minimum Principle is an important tool for this type of optimal control and has been used in related applications like hybrid vehicle energy management [70]. Sciarretta and Vahidi [109] and Han et al. [48] describe analytical solutions to several eco-driving problems. One elegant example shows that the energy-optimal speed profile for an electric vehicle is parabolic. A more complex solution considering acceleration and position constraints is also provided, along with feasibility analysis. The preceding vehicle (PV) acceleration is assumed constant at $a_{p}$. When the constraint is active, the ego approaches the preceding vehicle (PV) with parabolic speed until the contact time $\tau_{1}$ as formalized in Eqn. (1.7). Let $v_{i}, v_{p, 0}, s_{p, 0}$, and $\tau$ denote the ego's initial speed, PV's initial speed, PV's initial position relative to the ego, and time relative to the present, respectively.

$$
\begin{align*}
v(\tau)= & v_{i}+\left(a_{p}+\frac{4\left(v_{p, 0}-v_{i}\right)}{\tau_{1}}+\frac{6 s_{p, 0}}{\tau_{1}^{2}}\right) \tau  \tag{1.7}\\
& -\left(\frac{6 s_{p, 0}}{\tau_{1}^{3}}+\frac{3\left(v_{p, 0}-v_{i}\right)}{\tau_{1}^{2}}\right) \tau^{2}, \quad \tau \in\left[0, \tau_{1}\right)
\end{align*}
$$

The contact time $\tau_{1}$ solves the following equation, where $\tau_{f}, s_{f}$, and $v_{f}$ are the final time, position, and velocity, respectively.

$$
\begin{align*}
\left(v_{i}\right. & \left.-v_{f}+a_{p} \tau_{f}\right) \tau_{1}^{3} \\
& +\left(4 v_{p, 0} \tau_{f}+v_{f} \tau_{f}-2 v_{i} \tau_{f}+\frac{a_{p} \tau_{f}^{2}}{2}-3 s_{f}\right) \tau_{1}^{2}  \tag{1.8}\\
& +\left(6 s_{p, 0} \tau_{f}+v_{i} \tau_{f}^{2}-v_{p, 0} \tau_{f}^{2}\right) \tau_{1}-3 s_{p, 0} \tau_{f}^{2}=0
\end{align*}
$$

Then, the ego vehicle follows the standard parabolic profile until the end of the trip. Reference [47] proposes an MPC using a similar concept and considering both speed and position constraints. Malikopoulos and Zhao [83] provide another example of position-constrained closed-form optimal control using a Hamiltonian
analysis.
It is also possible to use analytical optimal control techniques to describe the solution, then solve those conditions numerically. For example, Hu et al. [56] applied PMP to efficient control of hybrid vehicles considering variable road grade using such a technique.

### 1.1.3.3 Numerical Optimal Control

For greater flexibility, optimal control problems can be solved numerically. Dynamic programming (DP) is a highly general but computation-intensive technique for doing so. It is typically used to find open-loop solutions to complex problems. For example, [2] applies DP to optimize an EV speed profile over varying road grade after using PMP to restrict the possible operating modes. Reference [85] uses DP for combustion engine eco-driving with car-following constraints. DP infamously suffers from a "curse of dimensionality"[71] where it scales poorly and becomes computationally intractable as the number of states increases. A diverse field of approximate dynamic programming techniques has emerged to combat this problem [102] with some using linear programming [24] or randomization [105].

Other numerical techniques exist to solve optimal control problems over a complete mission. For example, [95] uses sequential quadratic programming (SQP) to eco-drive, enabling a shrinking-horizon implementation where the problem is repeatedly solved from the current time to a fixed end point as real time advances. GPOPS-II is a notable tool that solves optimal control problems using a collocation approach originally developed to find finite-element solutions to partial differential equations [96].

When computational resources are limited and distant-future references and disturbances are either unimportant or uncertain, optimal control is often implemented in a receding horizon manner. This model predictive control (MPC) technique, the fundamentals of which are the topic of Section 1.2.1, has found extensive application in vehicle motion planning and control. Cruise control is one example, where [65] used MPC to attenuate traffic jams in simulation and [107] evaluated an MPC car-following controller's impact on emissions using a real engine. Truck platooning presents an opportunity to reduce aerodynamic drag losses using MPC, and [123] takes a coordinated approach to this task where a platoon coordinator supervises the vehicles. Compared to the previously discussed classical ACC systems, MPC can readily consume preview information and explicitly considers constraints, including collision avoidance. However, the shortsighted receding horizon does not directly guarantee actual collision avoidance and stability guarantees are more challenging compared to ACC. Dunbar and Caveney [36] provide an example of a string stability proof for receding horizon control and [82] combines MPC with a second controller for which collision avoidance is
formally guaranteed.
MPC has also been applied to lateral vehicle guidance. This can come in the form of continuous nonlinear programs commanding yaw rate as in [133] or discrete formulations where the lane change decision is the degree of freedom. Mukai and Kawabe [89] cast the latter problem as a mixed-integer linear program (MILP) and Du et al. [34] describe it as a mixed logical dynamical (MLD) model.

### 1.1.3.4 Machine Learning

Machine learning has also been applied to motion control, especially when lateral control among obstacles is considered. [98] used end-to-end learning in an off-road context that involves the same fundamental source of non-convexity as the lane changing problem addressed in Chapter 4. Ngai and Yung [92] used speed and heading in the reward function of a reinforcement learning controller for road vehicle overtaking. Later, [132] proposed reinforcement learning for automated merging.

Machine learning has also been applied to longitudinal eco-driving. Ma et al. [80] did so in an ecocoaching setting, where human drivers are shown how to improve their efficiency. Shi et al. [110] minimized $\mathrm{CO}_{2}$ emissions during connected intersection approach using reinforcement learning. Using Q-learning and deep deterministic policy gradients, [43] combined longitudinal eco-driving and lane selection. In general, these machine learning approaches tend to handle more complex and realistic scenarios but do not result in as clear physical insight compared to analytical optimal control methods.

### 1.2 Technical Background

This section introduces certain technical concepts that appear throughout. These include model predictive control, mathematical programming (optimization), and Pontryagin's Minimum Principle.

### 1.2.1 Model Predictive Control

Model predictive control (MPC) is a central concept to both this dissertation and much of the state-of-the-art that precedes it. The concept involves using a model, typically in state-space form, to predict the future evolution of the controlled dynamic system. Conventional notation denotes the states, outputs, and
manipulated inputs as $x, y$, and $u$ respectively.

$$
\begin{align*}
\dot{x} & =f(x, u)  \tag{1.9}\\
y & =g(x, u)
\end{align*}
$$

A specified objective $J$ that is a function of the system states and control inputs can then be optimized (usually minimized) with respect to the vector $U$ of manipulated inputs over time. Another key feature of MPC is the application of only the current time's control input and repetition of the process at the next sampling time while rolling the optimization horizon forward [9]. This achieves closed-loop control.

Where other control techniques like proportional-integral-derivative (PID) and linear quadratic regular (LQR) require saturation for input constraints and do not readily accommodate output constraints, MPC explicitly includes input constraints and can also include output constraints if feasibility is guaranteed. The canonical model predictive control problem thus takes the following form.

$$
\begin{array}{ll}
\min _{U} & J(x, u) \\
\text { s.t. } & c(x, u) \leq 0  \tag{1.10}\\
& \dot{x}=f(x, u) \\
& y=g(x, u)
\end{array}
$$

From this point forward, the presence of the plant model in the optimal control problem (OCP) is implied and the last two constraints in (1.10) may be dropped from the notation. Along with the number of decision variables, the form of the general nonlinear functions in (1.10) and the presence of integer decision variables affect the OCP's numerical tractability. For more detail on a variety of problem types and solution methods in model predictive control, see [13]. A commonly-used form is introduced next.

### 1.2.1.1 Linear-Quadratic Model Predictive Control

The solution of OCP (1.10) can either be numerically solved or have its solution explicitly calculated with the help of offline computations. The former method is called implicit MPC and the latter is called explicit MPC [73]. This dissertation will focus on implicit MPC, where computation time is often a limiting factor in control design. If the plant model i.e. functions $f$ and $g$ in (1.10) are linear and the cost is quadratic, (1.10) can be transformed into a computationally inexpensive quadratic program ( QP ) using the process described in [81]. A QP-based MPC can run sufficiently quickly for embedded control in commerical applications [10]
[11]. Such an MPC solves the following problem, which is used extensively in this research. Let $i$ denote the prediction step and $N$ denote the number of steps in the prediction horizon. Similar to $U, X$ is the combined vector of states over time. Equation (1.11) introduces the canonical linear-model form with matrices $A, B, C$, and $D$. The matrices $P, Q$, and $R$ contain weights.

$$
\begin{array}{ll}
\min _{U} & x^{\mathrm{T}}(N) P x(N)+\sum_{i=1}^{N} x^{\mathrm{T}}(i) Q x(i)+u^{\mathrm{T}}(i) R u(i) \\
\text { s.t. } & K\left[\begin{array}{ll}
X^{\mathrm{T}} & U^{\mathrm{T}}
\end{array}\right]^{\mathrm{T}} \leq w  \tag{1.11}\\
& \dot{x}=A x+B u \\
& y=C x+D u
\end{array}
$$

### 1.2.1.2 State Constraint Softening for Feasibility

In experiments or simulations involving model mismatch, the MPC OCP can be infeasible if hard output constraints are used. This causes the solver to exit without a solution, which is practically unacceptable. This dissertation addresses this problem by softening pure state constraints. The form of a single softened constraint, shown below, can be applied to as many constraints as needed. A slack variable $\epsilon$ is introduced as a non-negative decision variable that can relax a constraint in the following fashion.

$$
\begin{equation*}
c(x, y, u)-\epsilon \leq 0, \quad \epsilon \geq 0 \tag{1.12}
\end{equation*}
$$

The augmented objective $J_{a}$ then penalizes $\epsilon$ so that the constraints still influence the solution as intended, where the slack variables are collected into the vector $\bar{\epsilon}$. In Eqn. (1.13), the diag $(\mathbf{q})$ transformation arranges the elements of the $n$-vector $\mathbf{q}$ along the diagonal of an $n \times n$ matrix whose elements are zero otherwise.

$$
\begin{equation*}
J_{a}=J+\bar{\rho}_{l}^{\mathrm{T}} \bar{\epsilon}+\bar{\epsilon}^{\mathrm{T}} \operatorname{diag}\left(\bar{\rho}_{q}\right) \bar{\epsilon} \tag{1.13}
\end{equation*}
$$

As explained in [68], linear MPC can use linear or quadratic penalty functions on the slack variables $\epsilon$. Equation (1.13) includes both linear and quadratic penalties with weight vectors $\bar{\rho}_{l}$ and $\bar{\rho}_{q}$, respectively. While they render the cost non-smooth and can cause abrupt control action, linear penalty functions theoretically yield the exact hard-constrained solution to hard-constrained feasible problems for sufficiently high penalty weight. Quadratic penalties can provide smoother control performance but cannot deliver the exact hard-constrained solution.

### 1.2.1.3 Chance Constraints

When constraint satisfaction is especially critical and uncertainty especially great, constraints can include a safety margin. As applied in [131], probability can help convert a random variable's uncertainty into such a deterministic safety margin. Constraints using this technique are called chance constraints. The concept involves adjusting (usually tightening) the constraint so that the base constraint is satisfied with a specified probability $\alpha$ when the chance constraint is active. In general terms,

$$
\begin{equation*}
E C(x, u)+d \leq 0, \tag{1.14}
\end{equation*}
$$

with $d$ chosen such that

$$
\begin{equation*}
E C+d=0 \Longrightarrow F_{C}(x, u)(0)=\alpha \tag{1.15}
\end{equation*}
$$

where $F$ is the cumulative distribution function (CDF) of the random variable $C$ and $E$ is the expectation operator. The distribution of $C$ is translated by $d$ to influence its value for zero input. A more in-depth view of chance-constrained MPC can be found in [76]. An application of this idea to a practical problem by assuming the multivariate normal distribution appears in Section 4.5.

### 1.2.2 Mathematical Programming

Since several of the algorithms involved in this dissertation rely on numerical optimization, a brief overview of relevant mathematical programming is in order. This will cover linear programming (LP), quadratic programming (QP), and mixed integer programming (MIP) from an engineering viewpoint.

### 1.2.2.1 Linear and Quadratic Programming

A quadratic program is an optimization problem having a quadratic function of the decision variables as an objective and affine constraints. These constraints may contain equalities and inequalities. Such a problem is often expressed in the following canonical form. Here the symbol $x$, which usually denotes the
state vector, is the vector of decision variables in the optimization.

$$
\begin{array}{ll}
\min _{x} & \frac{1}{2} x^{\mathrm{T}} Q x+f^{\mathrm{T}} x \\
\text { s.t. } & A x \leq b  \tag{1.16}\\
& A_{e q} x=b_{e q} \\
& x \in \mathbf{R}^{n}
\end{array}
$$

A linear program can be described as a special case of optimal control problem (1.16) where $Q=\mathbf{0}$.
The computational ease of LP and QP results from their beneficial mathematical properties. First, the objective is a convex function and the constraint-admissible set is a convex set. A convex function [8] is defined as a function $f(x)$ where

$$
\begin{equation*}
f\left(\lambda x_{1}+(1-\lambda) x_{2}\right) \leq \lambda f\left(x_{1}\right)+(1-\lambda) f\left(x_{2}\right), \quad \lambda \in[0,1] \tag{1.17}
\end{equation*}
$$

and a convex set [8] is a set $S$ where

$$
\begin{equation*}
x_{1}, x_{2} \in S \Longrightarrow \lambda x_{1}+(1-\lambda) x_{2} \in S, \quad \lambda \in[0,1] \tag{1.18}
\end{equation*}
$$

A local optimum of an optimization with convex objectives and convex constraints is also a global optimum. This makes gradient search methods well-suited. Perhaps the best-known LP algorithm is the simplex method due to Dantzig [23], which travels along edges of the constraint-admissible set until it reaches the optimum. Quadratic programs also have the property that their Hessians are constant and specified, eliminating the need for costly finite differences calculations. Polynomial time algorithms exist for QPs, and some LP algorithms can be extended to solve them [91].

### 1.2.2.2 Mixed Integer Programming and Its Application

Mixed integer programs (MIPs) are a class of optimization problems involving both discrete and continuous decision variables. Notice that the constraint-admissible set of an MIP is inherently non-convex since the integers are not a convex set. The non-convexity of the integers can be leveraged to encode other non-convexity in an MIP.

For example, some problems including those over non-convex sets contain disjunctions, or pairs of constraints where either one $O R$ the other must be satisfied. Such a problem with a quadratic objective cannot
be cast as a QP, but it can be cast as a mixed-integer quadratic program (MIQP). One such method called Big $\mathrm{M}[128]$ involves introducing binary decision variables $\beta \in\{0,1\}$ and a sufficiently large scalar $M$. Say the problem contains the following disjunctive constraints.

$$
\begin{equation*}
a_{1}^{\mathrm{T}} x \leq b_{1} \quad \text { OR } \quad a_{2}^{\mathrm{T}} x \leq b_{2} \tag{1.19}
\end{equation*}
$$

Then for $\beta$ and $M$ as described previously, the following formulation allows one of (1.19) to be violated, but not both.

$$
\begin{equation*}
a_{1}^{\mathrm{T}} x-M \beta \leq b_{1} \quad A N D \quad a_{2}^{\mathrm{T}} x-M(1-\beta) \leq b_{2} \tag{1.20}
\end{equation*}
$$

This dissertation will also use some binary variables, denoted $\mu$, as indicator variables. Such decision variables are constrained to take a value of true or false when a certain event occurs. They can then be used with Big M to relax constraints depending on that event. See [38] and Section 4.3 of this dissertation for example applications.

### 1.2.3 Pontryagin's Minimum Principle

Analytical tools exist to solve or at least implicitly describe the solutions of optimal control problems (OCPs). This dissertation makes use of Pontryagin's Minimum Principle (PMP) to solve certain OCPs with only a few states and constraints. [49] uses the following notation, which is adopted here. This research will mainly focus on the case where the initial time $t_{0}$ and the final time $t_{f}$ are fixed.

$$
\begin{array}{ll}
\min _{u(t)} & J=\int_{t_{0}}^{t_{f}} F(x(t) u(t)) \\
\text { s.t. } & \dot{x}=f(x, u)  \tag{1.21}\\
& g(x(t) u(t)) \geq 0 \\
& h(x(t)) \geq 0
\end{array}
$$

In PMP, the optimal control $u^{*}(t)$ is found by minimizing a quantity called the Hamiltonian at all times. The Hamiltonian $\mathcal{H}$ with Lagrange multipliers $\bar{\lambda}$ is formed according to Eqn. (1.22).

$$
\begin{equation*}
\mathcal{H}=F(x(t) u(t))+\bar{\lambda}^{\mathrm{T}} f(x, u) \tag{1.22}
\end{equation*}
$$

When the problem involves state inequality constraints, the direct adjoining method can be used. This requires
the following Lagrangian $\mathcal{L}$ with additional Lagrange multipliers $\mu$.

$$
\mathcal{L}=F(x(t) u(t))+\bar{\lambda}^{\mathrm{T}} f(x, u)+\bar{\mu}^{\mathrm{T}}\left[\begin{array}{ll}
g(x(t), u(t))^{\mathrm{T}} & h(x(t))^{\mathrm{T}} \tag{1.23}
\end{array}\right]^{\mathrm{T}}
$$

The following conditions are then necessary for optimality.

$$
\begin{equation*}
\frac{\partial \mathcal{L}}{\partial u}=0, \quad \dot{\lambda}=-\frac{\partial \mathcal{L}}{\partial x}, \quad \mu \geq 0, \quad \mu g(x, u)=0, \quad \mu h(x)=0 \tag{1.24}
\end{equation*}
$$

The solution need not be continuous but can instead be a piecewise function of time [71]. In such cases, the jump conditions with jump parameter $\eta$ describe the solution at the contact time $t_{c}$ where two phases of the solution meet.

$$
\begin{equation*}
\lambda\left(t_{c}^{-}\right)=\lambda\left(t_{c}^{+}\right)+\eta \frac{\partial h}{\partial x}, \quad \mathcal{H}\left(t_{c}^{-}\right)=\mathcal{H}\left(t_{c}^{+}\right)-\eta \frac{\partial h}{\partial t}, \quad \eta \geq 0, \quad \eta h\left(t_{c}\right)=0 \tag{1.25}
\end{equation*}
$$

These necessary conditions describe the form of the solution, which can then be solved with the help of boundary conditions on the states. In general, the result is a two-point boundary value problem. In some cases, however, the result is a system of algebraic equations that can be solved quickly online.

### 1.3 Contributions

The overall contribution of this dissertation is the integration of look-ahead information in automated driving through methods that require minimal data as input and are optimal in some known sense. Novel solutions are presented to some challenges that arise when facing realistic applications, including non-convexity and computational constraints. A strength of this work is that it encompasses a range of operational domains and vehicles, including car-following with either one or multiple predecessors, multi-lane driving among arbitrary obstacles, and varying road slope.

Chapter 2 deals with car following, where a relatively large body of research exists. The main algorithmic contributions are a suite of controller variants enabling model predictive control in mixed strings of automated and conventional vehicles, and the consideration of a speed-varying maximum acceleration constraint where a constant constraint is typically used [88] [131] [141]. This latter contribution was deepened by applying mixed-integer programming to accommodate vehicles with highly non-convex powertrain operating spaces. Such non-convexity occurs in heavy diesel trucks with high aerodynamic drag. The consideration
of such heterogeneous strings consisting of both passenger, heavy, connected, and unconnected vehicles with realistically varying parameters differentiates Chapter 2 from other speed smoothing studies like [139] and [65]. The energy savings of the passenger vehicle car-following controllers that were developed in simulation were validated on real vehicles running in vehicle-in-the-loop experiments. The author's main individual contributions in car following were vehicle motion algorithm development including the MPC formulation, probability models, and chance constraints, as well as the MATLAB simulation tools and prototype simulation studies that used them. Other team members' contributions included but were not limited to VISSIM [103] simulations, acceleration control via the test vehicles' pedals, and connectivity implementation.

Where the algorithms in Chapter 2 plan motion considering the immediate predecessor, Chapter 3 uses information from multiple vehicles ahead that could be communicated using vehicle-to-vehicle (V2V) connectivity [111]. Inspired by the more classical controller proposed in [140], Chapter 3 explores how such information might assist MPC. A low-automation, medium-connectivity environment is assumed where surrounding vehicles are not automated but may be connected. Most car-following controllers, including the ones reviewed so far in this dissertation, only consider the immediate predecessor. Notably, [64] does consider more distant connected vehicles in order to derive a road speed profile. The approach in Chapter 3 is more microscopic in the sense that individual occluded unconnected vehicles are identified in order to predict transient gap adjustments. In addition to the algorithm itself, Chapter 3 contributes a comparison of several algorithms to isolate the benefits of considering multiple predecessors, identifying occluded vehicles, and using MPC.

The car following research is then expanded to multi-lane roads in Chapter 4. A novel model predictive controller is proposed for coupled acceleration and lane change planning using mixed integer quadratic programming. Because it combines a discrete lane selection input with continuous states, lane discipline is built-in without approximating lane changes as one-step jumps as in [34]. This approach fully comprehends that a vehicle engaged in a lane change obstructs two lanes at once. It can also be used to plan multiple lane changes in a prediction horizon, differentiating it from [66]. In a second contribution, the receding horizon planner is combined with references from a PMP-based shrinking horizon controller to enable eco-driving while avoiding general dynamic obstacles. This addresses a gap pointed out by [85] where eco-driving optimizers do not typically include traffic and goes a step further by considering lane changes. It also mitigates the well-known shortsightedness of receding horizon control. The technical work presented in Chapter 4, including algorithm design and simulation, was part of the author's individual contribution to a larger project on improving the energy efficiency of heterogeneous fleets. Extensions to tractor-trailers on sloping roads are
also presented using linear parameter varying (LPV) MPC.
Varying road slope and the desire to minimize energy in the objective motivated enhancements beyond the PMP trip planner of Chapter 4. Since the computational burden of DP was too great for closed-loop operation in the manner of Chapter 4, a novel formulation is developed in Chapter 5 to cast the wheel input energy eco-driving problem over generally varying road slope as an easier-to-solve linear program (LP). DP is typically used for this type of problem but is solved in open-loop, although [61] implemented an eco-driving controller on a vehicle using a power-based formulation cast as a MILP. A MILP is inherently more complex to solve than a similarly-sized LP since MILP involves multiple LP subproblems. Held [51] approaches a similar problem to the one in Chapter 5 and, like this research, uses a position-discretized formulation. However, the PMP-based solution involves a shooting method for boundary value problems. Because of the availability of fast commercial and open-source simplex method solvers for LP, the formulation of Chapter 5 could be a practical new alternative for online optimization in eco-driving over slopes. Physical insights are also gained from the results and discussed.

Chapter 6 adds contributions in the more advanced topic of multi-agent control of automated road vehicles. Centralized control is implemented and evaluated, but the main contributions are distributed collaborative algorithms. In car following, a considerate algorithm is proposed that includes neighboring vehicles in each vehicle's objective function. The distributed computation and constrained MPC distinguish this work from the centralized LQR-based platooning controller in [63]. String simulations show that such an approach can improve traffic compactness relative to strictly decentralized optimal control with a mild improvement in energy efficiency. In a multi-lane context, a different approach is proposed where the network's agents are ordered such that the distributed solution is improved through sequential decentralized optimization. Unlike the multi-agent consensus algorithms in [62] and [75] where the number of iterations may vary, the collaborative algorithm here requires exactly two solutions for each control move computed. The proposed approach also requires no roadside infrastructure or predefined conflict zones, distinguishing it from [60] and [79], respectively.

## Chapter 2

## Single-Predecessor Car Following

An anticipative cruise controller was developed to smooth traffic in a string of vehicles. It uses model predictive control to consume the preceding vehicle's (PV's) intended motion and generate its own future intentions. Four algorithm variants are proposed whose intended applications depend on the convexity of the ego's maximum acceleration constraints and the connectivity of the PV. MPC-C and MPC-U apply to ego vehicles with convex acceleration constraints that follow connected and unconnected PVs, respectively. Similarly, ego vehicles with non-convex acceleration constraints use MIPC-C or MIPC-U for connected or unconnected PVs. MIPC abbreviates Mixed Integer Predictive Control, named for its use of integer decision variables. The following sections will describe the base control with its objective and constraints, followed by probabilistic techniques used to model unconnected preceding vehicles. The non-convex acceleration constraints are explained next, followed by the probability modeling. Finally, results are shown from simulation and vehicle-in-the-loop (VIL) experiments.

This chapter contains algorithm development research that was published by Dollar and Vahidi in $[29]^{1}$ and [30] ${ }^{2}$. The VISSIM results were published in Ard et al. [5] ${ }^{3}$ and the VIL results are under review in Ard et al. [6] ${ }^{4}$, both of which list this dissertation's author as a co-author. This research was supported in part by an award from the U.S. Department of Energy Vehicle Technologies Office (Project No. DE-EE0008232).

[^0]
### 2.1 Base Control

The model predictive controller uses the following linear plant model, a double integrator with firstorder lag between the acceleration command $u$ and actual acceleration $a$. The position $s$ and velocity $v$ complete the state vector.

$$
\left[\begin{array}{l}
\dot{s}  \tag{2.1}\\
\dot{v} \\
\dot{a}
\end{array}\right]=\left[\begin{array}{ccc}
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & -\frac{1}{\tau}
\end{array}\right]\left[\begin{array}{l}
s \\
v \\
a
\end{array}\right]+\left[\begin{array}{l}
0 \\
0 \\
\frac{1}{\tau}
\end{array}\right] u
$$

The time constant $\tau$ varies depending on whether the model represents a passenger or heavy vehicle. In the simulation testbed, $\tau$ is varied depending on whether the powertrain or brake system is used. Powertrain $\tau$ is either 0.45 s or 0.9 s and braking $\tau$ is either 0.10 s or 0.25 s , with passenger vehicles taking the smaller value and heavy vehicles taking the larger one. The mean of the powertrain and braking $\tau$ is always used in the MPC prediction model.

Together with the linear model, the controller uses a quadratic cost and linear constraints to produce a quadratic program (QP). The following objective balances acceleration minimization with following the preceding vehicle $(\mathrm{PV})$. Let $r_{a}$ denote the anticipated PV position, $l_{v}$ the vehicle length, and $d_{r e f}$ the reference distance.

$$
\begin{align*}
J= & q_{g}\left(s(N)-s_{r e f}(N)\right)^{2}+q_{a} a^{2}(N) \\
& +\sum_{i=0}^{N-1}\left[q_{g}\left(s(i)-s_{r e f}(i)\right)^{2}+q_{a}\left(a^{2}(i)+u^{2}(i)\right)\right] \tag{2.2}
\end{align*}
$$

$$
\begin{equation*}
s_{r e f}=r_{a}-l_{v}-d_{r e f} \tag{2.3}
\end{equation*}
$$

The constants $q_{g}$ and $q_{a}$ are tuning weights and $N$ is the prediction horizon. These are set based on a parameter sensitivity study in [30]. The index $i$ denotes the prediction step.

The car following problem involves several constraints for safety, legal, and mechanical reasons. Minimum and maximum speeds $\underline{v}$ and $\bar{v}$ are imposed along with a braking acceleration limit $\underline{u}$.

$$
\begin{equation*}
\underline{u} \leq u, \quad \underline{v}-\epsilon_{3} \leq v \leq \bar{v}+\epsilon_{2} \tag{2.4}
\end{equation*}
$$

The slack variables $\epsilon_{2}$ and $\epsilon_{3}$ are for constraint softening as mentioned in Section 1.2.1.2. The maximum
acceleration is a piecewise linear function of speed defined by the coefficients $m_{1}, m_{2}, b_{1}$, and $b_{2}$.

$$
\begin{array}{cc}
-m_{1} v+u \leq b_{1} & -m_{2} v+u \leq b_{2} \\
-m_{1} v+a \leq b_{1}+\epsilon_{4} & -m_{2} v+a \leq b_{2}+\epsilon_{4} \tag{2.6}
\end{array}
$$

The constraints on the acceleration state are also softened. The penalty on soft constraints and its role in the objective are

$$
\begin{equation*}
J_{\epsilon}=\sum_{j=1}^{4} \rho_{j}\left\|\epsilon_{j}\right\|_{\infty}, \quad J_{a}=J+J_{\epsilon} \tag{2.7}
\end{equation*}
$$

where $\rho_{j}$ is the penalty weight on slack variable $j$.
An important constraint limits the gap between the ego and the PV to a safe minimum $\underline{d}$.

$$
\begin{equation*}
\underline{d}-\epsilon_{1} \leq r_{c}-s \tag{2.8}
\end{equation*}
$$

In the more ideal connected case where a preview of the PV position is available, that anticipated PV position $r_{a}$ is used directly as $r_{c}$. When the PV is not connected, the position constraint must add additional safety margin. A worst-case constraint was used to generate the results shown in Section 2.4. In worst-case constraints, a safe PV position $r_{w c}$ is calculated by assuming that the PV maximally brakes and passing that input sequence through linear model (3.1). This guarantees collision avoidance as long as the PV's maximum braking capability is weaker than or equal to the assumed value ${ }^{5}$. In summary,

$$
r_{c}=\left\{\begin{array}{lll}
r_{a} & : & \text { MPC-C, MIPC-C }  \tag{2.9}\\
r_{w c} & : & \text { MPC-U, MIPC-U }
\end{array}\right.
$$

Finally, a constraint on the terminal position and speed guarantees that the receding-horizon solution will not only prevent collisions in the horizon, but also result in a safe situation at the end of the horizon. The parameters $m_{3}$ and $\xi$ are recomputed at each step as described in [30].

$$
\begin{equation*}
-m_{3} v(N)+s(N) \leq \xi \tag{2.10}
\end{equation*}
$$

[^1]
### 2.2 Non-Convex Constraints

Constraints of the form exemplified by (2.5) apply well to vehicles with approximately convex powertrain operating spaces as shown in Fig. A.1. In contrast, other vehicles like the Class 8 diesel truck modeled in Section A. 1 have highly non-convex operating spaces as in Fig. A.2. Use of (2.5) and (2.6) severely limits the low-speed acceleration of heavy trucks, even beyond their restrictive hardware limits. To overcome the nonconvexity, the truck's operating space is approximated as two disjunctive, rather than conjunctive, constraints that are placed in canonical form using the Big M method. This results in a mixed integer quadratic program (MIQP) instead of a quadratic program. Equations (2.11, 2.12) show the disjunctive constraints, where $\beta \in\{0,1\}$ and the constant $M$ is set to the maximum difference between the two acceleration constraints within the feasible speed range.

$$
\begin{equation*}
u-m_{1} v-M \beta \leq b_{1}, \quad u-m_{2} v-M(1-\beta) \leq b_{2} \tag{2.11}
\end{equation*}
$$

$$
\begin{equation*}
a-m_{1} v-M \beta \leq b_{1}, \quad a-m_{2} v-M(1-\beta) \leq b_{2} \tag{2.12}
\end{equation*}
$$

### 2.3 Probability Modeling

As mentioned in Section 2.1, the ego vehicle relies on vehicle-to-vehicle (V2V) connectivity to obtain the expected PV position $r_{a}$ when the PV is connected. When the PV is not connected, the ego must predict $r_{a}$ by other means. In [30], the author proposed a Markov-like technique with the PV's current speed $v$ (discretized as $\nu$ ) and brake light state $\lambda$ as inputs and the PV's acceleration at a sequence of future steps as output. The term Markov-like is used because although the algorithm involves a probabilistic transition matrix similar to a Markov chain, the acceleration multiple steps in the future depends on the current state and therefore lacks the Markov property.

The probability model's function is briefly reviewed. At each step, the previous acceleration command is estimated since it is not directly measured by radar. Then, the transition matrix is updated according to the following process. Let $h$ denote the index of the discrete acceleration bin, $l$ denote the timestep, and $P$ denote the transition matrix composed of probabilities. The strength $M$ of an input tuple is the number of samples collected for that tuple.

1. Reconstruct the number of samples for the current transition: $n_{\lambda \nu h l}=P_{\lambda v h l} M_{\lambda v l}$
2. Increment $n_{\lambda v h l}$
3. Increment the strength $M_{\lambda v l}$
4. Compute the new transition probability: $P_{\lambda v h l}=\frac{n_{\lambda v h l}}{M_{\lambda v l}}$

After updating the transition matrix, the input $u_{a}$ can be predicted for a prediction step $l$ using a weighted average of the $n_{b i n, t}$ acceleration bins' values $u_{h}$.

$$
\begin{equation*}
u_{a}(l)=\sum_{h=1}^{n_{b i n, t}} u_{h} P_{j h l} \tag{2.13}
\end{equation*}
$$

Despite only consuming the PV's speed and brake light state without traffic context, the model was able to improve prediction compared to the free response assumption for 5 s into the future as shown in Fig. 2.1. The PV in this evaluation used the Intelligent Driver Model [120] to follow an open-loop leader that followed the EPA US06 cycle.


Figure 2.1: Mean PV $u$ residuals of the prediction algorithm after one cycle. ©2018 IEEE.

### 2.3.1 Application to Real-World Bus Data

To assess the applicability of similar probability modeling techniques to real-world vehicles, a prediction model was developed and evaluated on GPS data obtained from Clemson University's Tiger Commute bus system. The highway trip down both directions of the segment of Interstate 85 shown in Fig. 2.2 was extracted for the study. This provided about 149,000 data points from 18 different drivers, sampled at 5 Hz . Only the speed and position of the bus was available, not the states of surrounding vehicles. This situation posed a similar challenge to the car following problem described in the previous section, albeit at a lower sampling frequency. Department of Energy project milestones set the accuracy target of 10 m maximum error for 5 s predictions.


Figure 2.2: The sample road segment for the Tiger Commute prediction study.

Because of the longer sample time, a speed-based approach was adopted rather than one based on acceleration. This algorithm also used a true Markov chain where the state probabilities at each step only depend on the states at the previous step. Figure 2.3 shows the algorithm as a block diagram where the inputs to the transition matrices are the direction of travel $\mu$, position $s$, time $t$, instantaneous speed $v$, and the difference between the instantaneous speed $v$ and the effective speed $\tilde{v}$ that is obtained by dividing the measured distance traveled during the last step by the step's time duration. Separate transition matrices predict the next instantaneous and effective speeds. The subsequent inputs can then be recursively computed by maintaining constant direction, advancing time by the timestep, and computing the new position as the product of the timestep and effective speed.


Figure 2.3: Block diagram of the Markov chain applied to the Tiger Commute data.

The transition matrices were trained using $80 \%$ of the Tiger Commute GPS data and the system was validated using the remaining $20 \%$. In addition to comparing against the milestone target, the Markov chain approach was also compared to simple kinematic predictions using constant instantaneous or effective speed. The error results in Fig. 2.4 show that the Markov chain system exceeded its performance targets by predicting position within $10 \mathrm{~m}, 10 \mathrm{~s}$ in advance. It should be mentioned that assuming constant effective speed also met the target. Fig. 2.5 shows that constant instantaneous speed performed better than constant


Figure 2.4: Accuracy results of Markov chain and constant effective velocity predictions.


Figure 2.5: Markov chain and constant velocity RMSE at various prediction horizons.
effective speed, and the advantage of the Markov chain over these approaches appears in longer predictions of 15 s or more. Figures 2.6 and 2.7 offer further insight into where the Markov chain had an advantage and why. The speed histogram shows that most data was collected at typical highway cruising speeds, but some points at lower speeds were also observed. The RMSE curve in Fig. 2.7 then shows that constant speed performed
similarly to the Markov chain in the cruising-speed points, but the Markov chain performed better at lower speed points. Since slowdowns are typically fleeting compared to normal cruising, the Markov chain would have been able to accurately predict transitions back to cruising speed. On the other hand, the constant-speed algorithm would have predicted a continuation of low-speed driving, deviating from the data.


Figure 2.6: Speed histogram used in the Markov chain, with discretization boundaries.


Figure 2.7: Markov chain RMSE benefit vs. constant speed across the velocity levels marked in Fig. 2.6.

### 2.4 Results of Optimal Car-Following

This car following algorithm was initially simulated in a custom MATLAB environment. Written by the author of this dissertation, this MATLAB simulator uses object-oriented programming to achieve multiagent microsimulation where information is appropriately restricted between vehicles. Later, the author's colleague Tyler Ard implemented this algorithm in $\mathrm{C}++$ and evaluated a modified version in a customized VISSIM setup. This C++ version was also evaluated in vehicle-in-the-loop (VIL) experiments involving
several colleagues. The author's role in these later efforts was to provide algorithm support, with colleagues leading the VISSIM simulation and VIL sides of the project. This section focuses mainly on the author's individual results, although the subsequent group efforts are mentioned for the sake of completeness.

### 2.4.1 MATLAB Microsimulations

The algorithm was evaluated in randomly mixed 8 -vehicle strings. An open-loop lead vehicle followed the EPA US06 cycle exactly and the 8 following vehicles consisted of a random mix of heavy and passenger vehicles with automated and modeled human drivers. The various vehicle and driver combinations were simulated in 2224 different arrangements to arrive at the average results. Figures 2.8 and 2.9 show the fuel economy and space utilization results as functions of the concentration of CAVs in the string, where space utilization is defined as the mean distance from vehicle 1 's front bumper to vehicle 8 's rear bumper. Fuel economy steadily improved with the addition of CAVs, although the presence of heavy vehicles with limited acceleration reduced the benefit of CAVs by introducing their own damping effect. Overall fuel economy benefits of up to $19 \%$ were observed in simulation. Figure 2.9 shows that while the $100 \%$ CAV strings were more compact than the $0 \%$ CAV ones, the addition of CAVs increased string length at low CAV concentrations. This phenomenon results from the conservative worst-case constraints used in MPC-U and MIPC-U, which are deployed more frequently when CAVs are rarer and more likely to follow a conventional vehicle.


Figure 2.8: Fuel economy at various string compositions. ©2018 IEEE.


Figure 2.9: Space utilization at various string compositions. ©2018 IEEE.

Computation time, a major consideration in the controller's design, was also measured. Table 2.1 shows the results, where MPC denotes the convex QP-based formulation, MIPC denotes the mixed-integer formulation, and the suffix specifies the connected (C) or unconnected $(\mathrm{U})$ variant. Ctrl. Time is the total
time needed to compute one vehicle's control move at each step and Opt. Time is the time needed to solve the mathematical program. For comparison, the later experimental implementation computed a new move every 0.1 s . Hence these results were considered promising for real-time implementation, especially considering the improvement realized by transitioning from MATLAB to $\mathrm{C}++$.

Table 2.1: As-calibrated computation time for car following.

| Algorithm | Mean Ctrl. Time [s] | Max Ctrl. Time [s] | Mean Opt. Time [s] | Max Opt. Time [s] |
| :--- | :--- | :--- | :--- | :--- |
| MPC-C | 0.0337 | 0.0561 | 0.0108 | 0.0444 |
| MPC-U | 0.0757 | 0.1134 | 0.0110 | 0.0892 |
| MIPC-C | 0.0435 | 0.0789 | 0.0148 | 0.0504 |
| MIPC-U | 0.0571 | 0.0919 | 0.0069 | 0.0425 |

### 2.4.2 VISSIM Microsimulations

As documented in [5], the author's colleague Tyler Ard implemented the proposed optimal car following algorithm in $\mathrm{C}++$ with the addition of a time headway term to the objective. In the unconnected PV case, the worst-case constraints were replaced with chance constraints similar to Section 4.5 to improve string compactness. This version was simulated in VISSIM [103], which uses the partially stochastic Wiedemann human driver model and replicates traffic shockwaves without the need for an imposed drive cycle disturbance. Figure 2.4.2 shows the reduction in fuel consumption from CAVs at a range of traffic volumes using the author's internal combustion engine vehicle (ICEV) energy model that was described in Section A.1.


Volume per Hour
CAV Fleet Penetration [\%]

Figure 2.10: VISSIM fuel savings. Figure by Tyler Ard [5], used with permission.

Reference [5] also provides energy results using the Autonomie vehicle and powertrain model [46]. That analysis showed up to 25 \% fleet improvement from CAVs where the average CAV used $6.6 \%$ to $22.1 \%$ less fuel than the average conventional vehicle depending on traffic volume.

### 2.4.3 Vehicle-in-the-Loop Experiments

The same version of the algorithm that was used in the VISSIM study was implemented in Vehicle-in-the-Loop (VIL) experiments. VIL enables real vehicles to interact with simulated ones by passing measurements of the simulated vehicles' states to the real vehicle(s) and vice versa. Thus VIL experiments can include real vehicle dynamics and energy flows with reduced cost and improved safety compared to trackonly or real-world testing. Figure 2.4.3 illustrates the VIL architecture, using two real vehicles as an example. While the team did conduct some testing with two real vehicles, the results shown in this section used one real vehicle. A Nissan Leaf EV (Fig. 2.4.3(a)) and Mazda CX7 ICEV were equipped with steering and pedal


Figure 2.11: Schematic of a VIL system. The results shown here used one real vehicle.
actuators (Fig. 2.4.3(b)) along with RTK GPS and IMUs. A group effort involving numerous colleagues assessed the energy benefit of CAVs equipped with MPC over simulated human drivers. The author's main technical contributions were optimal car following algorithm development and support, along with modeling ICEV fuel consumption from on-board diagnostics (OBD) data. The publication in review [6] lists the author's contributions as methodology, data curation, formal analysis and writing - reviewing and editing.

The results in Tables 2.2 and 2.3 show an energy benefit of $12 \%$ or $23 \%$ for the ICEV and $7.9 \%$ or 20.6 \% for the EV, depending on whether or not the lead vehicle was connected. This was accomplished without increasing travel time or mean time headway, which is important for maintaining road throughput. These experimental results were calculated from a model that used OBD mass airflow and commanded equivalence ratio as input and was fine-tuned based on historic fuel trim data and chassis dynamometer fuel volume measurements. The Intelligent Driver Model (IDM) [120] and Wiedemann (WIE) [129] [135] driver model were used as baselines.

The outcome of the VIL results is interesting from an algorithm development process perspective. The controllers were designed and initially evaluated in simulation without any vehicle hardware; energy


Figure 2.12: The Nissan Leaf test vehicle (a) and its pedal actuators (b).

Table 2.2: Experimental performance comparison, VIL (Mazda, combustion) [6].

| Metric | WIE | IDM | MPC-U | MPC-C |
| :--- | :--- | :--- | :--- | :--- |
| Travel Time (change) [min:s] | $24: 01$ | $23: 45(-1.1 \%)$ | $24: 00(0 \%)$ | $23: 34(-1.9 \%)$ |
| Mean Headway (change) [s] | 3.47 | $5.73(+65.1 \%)$ | $3.32(-4.3 \%)$ | $2.75(-20.7 \%)$ |
| Net Fuel (change) [L] | 2.556 | $2.174(-15 \%)$ | $2.241(-12 \%)$ | $1.978(-23 \%)$ |

Table 2.3: Experimental performance comparison, VIL (Nissan, electric) [6].

| Metric | WIE | IDM | MPC-U | MPC-C |
| :--- | :--- | :--- | :--- | :--- |
| Travel Time (change) | $23: 49$ | $23: 49(0 \%)$ | $23: 40(-0.9 \%)$ | $23: 36(-1.9 \%)$ |
| Mean Headway (change) [s] | 3.96 | $5.81(+46.7 \%)$ | $2.93(-26.0 \%)$ | $2.82(-28.8 \%)$ |
| Net Energy (change) [kwh] | 4.090 | $3.730(-8.8 \%)$ | $3.766(-7.9 \%)$ | $3.247(-20.6 \%)$ |

use was modeled using dynamometer data from [116] and physical vehicle parameters. Nonetheless, the magnitude of the benefits predicted in the MATLAB simulations of Section 2.4.1 is similar to that observed in VIL. Table 2.2 for the ICEV reports a 12-23 \% benefit from anticipative control and the MATLAB singlevehicle results reported in [30] show about a 13-20 \% benefit. Implementation was not without its challenges. Specifically, variable latency in cellular communication and mismatch between the commanded and actual acceleration were two of the most significant issues encountered at the test track. Even after improvement in the low-level controller throughout the project, some jerks persisted as MPC brought the vehicle to a stop. The VIL experiments enabled human operators to ride along in the vehicles, exposing this sort of comfort issue more clearly than simulations do.

## Chapter 3

## Multi-Predecessor Car Following

The previous chapter addressed model predictive control for car following. Although prediction was explored including worst-case and probabilistic approaches, only one predecessor was considered. In reality, each vehicle's motion in traffic results from interactions with other traffic elements, particularly their own predecessors in single-lane driving. Therefore, this chapter hypothesizes that information about vehicles ahead of the immediate predecessor can improve prediction and ultimately control performance.

Assuming low automation and medium connectivity, this hypothesis is evaluated by designing algorithms to predict predecessor motion using connected predecessors' measurements. Since not all vehicles are connected, an important part of this process is detection of unconnected vehicles that may lie between connected ones in a vehicular string. The prediction is then applied to the previous chapter's optimal controller. The result is evaluated using experimental data from real human drivers as input. A classical controller using multiple predecessors is benchmarked.

This chapter draws from research [25] ${ }^{1}$ that has been accepted to the 2021 American Control Conference at the time of writing. The study was a collaborative effort with Gábor Orosz and Tamás G. Molnár, who worked at the University of Michigan. The Michigan team provided the experimental human dataset and the benchmark classical controller. The dissertation author developed the detection, prediction, and optimal control algorithms and simulated their performance.

[^2]
### 3.1 Scenario and Assumptions

To focus on a realistic near-term scenario, only the ego vehicle is assumed to be automated. It senses its immediate predecessor using state-of-the-art autonomous sensing, but further predecessors are assumed occluded. The second predecessor is not connected and the connected automated vehicle (CAV) has no direct information about its state. However, the third predecessor is connected and communicates its current position and speed to the CAV. Figure 3.1 depicts the scenario, where Human-driven Vehicles are abbreviated HV and Connected Human-driven Vehicles are abbreviated CHV.


Figure 3.1: The three predecessors and the CAV to be controlled. ©2021 IEEE.

### 3.2 Modeling

To capture communication lag, the ego CAV is modeled using a double integrator with constant delay $\sigma=0.6 \mathrm{~s}$.

$$
\begin{align*}
& \dot{s}(t)=v(t) \\
& \dot{v}(t)=a(t)  \tag{3.1}\\
& a(t)=\min \{\max \{u(t-\sigma),-\underline{u}\}, \bar{u}(v(t))\} .
\end{align*}
$$

As in the previous chapter, $s, v, a$, and $u$ denote the position, speed, acceleration, and acceleration command, respectively. Time is denoted as $t$. The braking limit $\underline{u}$ and acceleration limit $\bar{u}(v)$ are incorporated into the model.

The delay is captured exactly in the model predictive controller by adding delay states to the discrete-
time model

$$
\begin{align*}
s(k+1) & =s(k)+v(k) \Delta t+a(k) \Delta t^{2} / 2 \\
v(k+1) & =v(k)+a(k) \Delta t \\
a(k) & =\min \left\{\max \left\{\delta_{q}(k),-\underline{u}\right\}, \bar{u}(v(k))\right\} \\
\delta_{q}(k+1) & =\delta_{q-1}(k)  \tag{3.2}\\
& \vdots \\
\delta_{1}(k+1) & =\delta_{0}(k) \\
\delta_{0}(k) & =u(k)
\end{align*}
$$

where $k$ is the simulation step. In this case, the number $q$ of delay states $\delta$ is 3 since the timestep $\Delta t=0.2 \mathrm{~s}$ and the delay is 0.6 s . The delay is neglected when predicting the motion of human-driven predecessors.

A human driver model is needed to predict each predecessor's motion based on its own predecessor. Although the proposed algorithm is modular in the sense that general obstacle-aware longitudinal driver models can be used, this study uses the Intelligent Driver Model (IDM) [120] that has been described earlier in Section 1.1.3.1. It is important for this algorithm that the driver model parameters accurately describe the encountered human drivers. Therefore, the following optimization was solved for the experimental dataset to find the best IDM parameters for this group of drivers. The parameters were not individualized; one set was used for the whole string.

$$
\begin{array}{ll}
\min _{x_{\mathrm{p}}} & J_{\mathrm{p}}=\frac{1}{N_{v}} \sum_{n=1}^{N_{v}} \sqrt{\frac{1}{N_{s}} \sum_{k=1}^{N_{s}}\left(\hat{d}_{n}(k)-d_{n}(k)\right)^{2}} \\
\text { s.t. } & 0.1 \mathrm{~m} / \mathrm{s}^{2} \leq a_{0} \leq 4 \mathrm{~m} / \mathrm{s}^{2} \\
& 0.1 \mathrm{~m} / \mathrm{s}^{2} \leq b_{0} \leq 8.5 \mathrm{~m} / \mathrm{s}^{2} \\
& 0.1 \mathrm{~m} \leq d_{\mathrm{st}} \leq 10 \mathrm{~m}  \tag{3.3}\\
& 0.1 \mathrm{~s} \leq \tau_{\mathrm{h}} \leq 4 \mathrm{~s} \\
& 0 \leq v_{\max } \leq 36 \mathrm{~m} / \mathrm{s} \\
& 1 \leq \delta \leq 10 \\
& x_{\mathrm{p}}=\left[\begin{array}{lllll}
a_{0} & b_{0} & d_{\mathrm{st}} & \tau_{\mathrm{h}} & v_{\max }
\end{array} \quad \delta\right]
\end{array}
$$

The squared error between the IDM-based simulated gap $\hat{d}_{n}$ and the actual gap $d_{n}$ was minimized over all $N_{v}$ vehicles and $N_{s}$ steps. The degrees of freedom $x_{\mathrm{p}}$ are the standard IDM parameters from Section 1.1.3.1. Table 3.1 lists the identified parameters.

Table 3.1: IDM parameters in multi-predecessor car following.

| Parameter | Description | Result |
| :--- | :---: | :---: |
| $a_{0}$ | Maximum acceleration | $2.43 \mathrm{~m} / \mathrm{s}^{2}$ |
| $b_{0}$ | Deceleration coefficient | $8.5 \mathrm{~m} / \mathrm{s}^{2}$ |
| $d_{\mathrm{st}}$ | Stopped distance | 3.3 m |
| $\tau_{\mathrm{h}}$ | Time headway | 0.76 s |
| $v_{\max }$ | Speed limit | $36 \mathrm{~m} / \mathrm{s}$ |
| $\delta$ | Exponent | 6.13 |

Finally, an energy model is used for benchmarking. The energy input to the wheels $E$ is calculated using the formula

$$
\begin{equation*}
E=\int_{0}^{t_{f}} \max \left\{\left(a(t)+a_{\mathrm{r}}+c_{\mathrm{r}} v^{2}(t)\right), 0\right\} v(t) \mathrm{d} t \tag{3.4}
\end{equation*}
$$

where $t_{f}$ denotes the final time. The resistance coefficients $a_{\mathrm{r}}=1.47 \times 10^{-1} \mathrm{~m} / \mathrm{s}^{2}$ and $c_{\mathrm{r}}=2.75 \times 10^{-4} \mathrm{~m}^{-1}$ were derived from the Ford Escape energy model of Appendix A.

### 3.3 Optimal Controller

The optimal controller follows the block diagram in Fig. 3.2. This section will describe each of the three main processes: detection, prediction, and optimization. Figure 3.3 shows the past and future time horizons involved in the process and defines the number of steps $K, M$, and $N$.

### 3.3.1 Detecting Hidden Vehicles

Hidden vehicles that may exist between connected and/or sensed ones are detected by simulating $n_{\mathrm{h}} \in \mathbf{Z}$ hypothetical hidden vehicles between a known leader and follower. The value $n_{\mathrm{h}^{*}}$ that best explains the follower's recorded past motion according to the following optimization problem is selected.

$$
\begin{align*}
n_{\mathrm{h}}^{*}(k)= & \underset{n_{\mathrm{h}}(k)}{\arg \min } \frac{k_{\mathrm{s}}}{K} \sum_{j=M-K+1}^{M}\left(\hat{s}_{1}\left(n_{\mathrm{h}}(k), j\right)-s_{1}(j)\right)^{2}  \tag{3.5}\\
& \text { s.t. } \quad 0 \leq n_{\mathrm{h}} \leq \min _{j} \frac{s_{l}(j)-s_{1}(j)-\ell-d_{\mathrm{st}}}{\ell+d_{\mathrm{st}}}
\end{align*}
$$



Figure 3.2: Block diagram of the multi-predecessor car following controller.


Figure 3.3: The time horizons used for multi-predecessor detection and prediction. ©2021 IEEE.
where

$$
k_{\mathrm{s}}= \begin{cases}1 & \text { if } \quad n_{\mathrm{h}}(k)=n_{\mathrm{h}}^{*}(k-1)  \tag{3.6}\\ 1.5 \quad \text { otherwise }\end{cases}
$$

The estimated position $\hat{s}_{1}$ of the immediate predecessor, which trails the possible hidden vehicles, is compared to its actual recorded position $s_{1}$. The factor $k_{s}$ penalizes the act of switching the number of hidden vehicles, which can ultimately cause jerk in the CAV's control move. The number of hidden vehicles must be nonnegative and small enough for each vehicle of length $\ell$ to physically fit with the IDM's minimum stopped gap $d_{\mathrm{st}}$ between them. The initial conditions for vehicle $p$ in these ID simulations are

$$
\begin{equation*}
s_{p}(0)=s_{1}(0)+p \frac{s_{l}(0)-s_{1}(0)}{n_{\mathrm{h}}+1}, \tag{3.7}
\end{equation*}
$$

and

$$
\begin{equation*}
v_{p}(0)=\frac{v_{l}(0)+v_{1}(0)}{2}, \tag{3.8}
\end{equation*}
$$

where $s_{l}$ and $v_{l}$ are the position and speed of the lead vehicle and $s_{1}$ and $v_{1}$ are the position and speed of vehicle 1 in Fig. 3.1. The purpose of simulating the 20 s before the data points used for comparison is to reduce sensitivity to these initial conditions.

### 3.3.2 Prediction

With the number of hidden vehicles identified, the preceding string's motion can be simulated into the future. Simulation proceeds sequentially from front to rear using the driver model. The terminal states from the optimal ID simulation serve as initial conditions for prediction. Once the CAV's immediate predecessor has been simulated, a usable predecessor trajectory is available for MPC.

Two other types of preview appear in the benchmark results. The term full preview means that MPC was provided with it's predecessor's actual experimental trajectory. This is impossible in reality because it assumes exact information about future events. The MPC was also benchmarked using only information from its immediate predecessor in a probability model as in the unconnected case from Chapter 2.

### 3.3.3 Optimization

The core MPC is based on that of the previous chapter, tracking a time headway $T$ behind the predecessor. The maximum acceleration $\bar{u}(v)$ is carried over from the passenger car model in Section 2.1, illustrated in Fig. A.1. The collision avoidance constraint is modified here to maintain a minimum time headway $\underline{T}$. This preserves the requirement that the baseline classical controller was developed to meet, enabling a more controlled comparison. The weights $q_{g}$ and $q_{a}$ were balanced to observe a maximum time headway target.

$$
\begin{array}{ll}
\min _{u(i)} J= & q_{\mathrm{g}}\left(s_{1}(N)-s(N)-\ell-d_{\mathrm{st}}-T v(N)\right)^{2} \\
& +\sum_{i=0}^{N-1}\left[q_{\mathrm{g}}\left(s_{1}(i)-s(i)-\ell-d_{\mathrm{st}}-T v(i)\right)^{2}+q_{\mathrm{a}}\left(u^{2}(i)+a^{2}(i)\right)\right] \\
\text { s.t. } & 0 \leq v \leq v_{\max } \\
\quad 0 \leq s_{1}-s-\ell-\underline{T} v-\underline{d}-d_{\mathrm{c}}  \tag{3.9}\\
\quad-\underline{u} \leq u \leq \bar{u}(v) \\
\quad-\underline{u} \leq a \leq \bar{u}(v) \\
\bar{u}(v)=\min \left\{m_{1} v+b_{1}, m_{2} v+b_{2}\right\}
\end{array}
$$

### 3.4 Performance Assessment

The experimental dataset shown in Fig. 3.4 was used to assess performance in simulation. In addition to energy, headway and detection accuracy were also examined. Several different controllers were benchmarked and are now reviewed.


Figure 3.4: The experimental trajectories of the human-driven predecessors. ©2021 IEEE.

### 3.4.1 Benchmark Controllers

In addition to the proposed MPC with multi-predecessor prediction and hidden vehicle identification, an MPC using only a single predecessor for preview was simulated as a basis for comparison. Other simulated MPCs included a high-performing benchmark that was supplied with perfect preview, the proposed system omitting hidden vehicle identification, and the proposed system assuming perfect hidden vehicle identification. These help to assess the necessity and quality of identification and the opportunity to improve performance in future research.

A classical controller proposed in [140] that uses information from multiple predecessors was also benchmarked. It explicitly calculates the CAV's acceleration command as

$$
\begin{equation*}
u=\alpha(V(d)-v)+\sum_{n=1}^{y} \beta_{n}\left(W\left(v_{n}\right)-v\right) \tag{3.10}
\end{equation*}
$$

where

$$
\begin{equation*}
V(d)=\min \left\{\max \left\{0,\left(d-d_{\mathrm{st}}\right) / T\right\}, v_{\max }\right\} \tag{3.11}
\end{equation*}
$$

and

$$
\begin{equation*}
W\left(v_{n}\right)=\min \left\{v_{n}, v_{\max }\right\} . \tag{3.12}
\end{equation*}
$$

The effect of the target speed $V(d)$ is to drive the ego vehicle toward a target time headway $T$ or, if the predecessor is far away, a maximum speed $v_{\max }$. The remaining terms involving $W\left(v_{n}\right)$ drive the CAV to match speed with its predecessors, so long as they do not exceed the speed limit. The gains are set to $\alpha=0.4 \mathrm{~s}^{-1}, \beta_{1}=0.2 \mathrm{~s}^{-1}$, and $\beta_{2}=0.0 \mathrm{~s}^{-1}$, and $\beta_{3}=0.6 \mathrm{~s}^{-1}$. The setting $\beta_{2}=0.0 \mathrm{~s}^{-1}$ implies that information about the hidden vehicle is not used for control, which enables direct comparison with the proposed MPC.

### 3.5 Results

Table 3.2 shows the overall energy performance of each algorithm. Energy generally trades off with gap in car following, so Table 3.2 also shows the corresponding average gaps. Each algorithm's trajectories in headway-speed space are shown in Fig. 3.5 to enable comparison with the target, minimum, and maximum headways.

Table 3.2: Wheel-input energy and headway results.

| Controller | Energy [J/kg] | Mean Gap [m] | Gap RMSE [m] |
| :--- | :---: | :---: | :---: |
| MPC 1 Pred. | 6180 | 50.34 | 14.29 |
| Classical | 5868 | 42.45 | 6.95 |
| MPC ID Off | 5948 | 34.81 | 13.77 |
| MPC ID On | 5505 | 47.32 | 12.98 |
| MPC Ideal ID | 5395 | 48.29 | 12.29 |
| MPC Full Prev. | 4698 | 43.84 | 8.29 |

MPC with full preview performed the best overall, as expected. The classical controller with multipredecessor preview performed better than the MPC that only used a single predecessor; however, adding information from vehicle 3 to MPC enabled it to save energy compared to the classical controller. Detection of the hidden vehicle was necessary to realize this benefit, as demonstrated by the significantly worse performance with ID off compared to ID on. Neglecting the hidden vehicle not only increased energy use but also led to poor headway tracking and violation of the minimum headway constraint, see Fig. 3.5(c). The consistent bias toward shorter headways results from the following process. If vehicle 2 is omitted from the
———Headway Specification - - Headway Reference _—_ Vehicle Trajectory
(a) MPC 1 Pred.


(c) ID Off
(d) ID On


(e) Ideal ID
(f) MPC Full Prev.



Figure 3.5: Simulated trajectories of each algorithm in headway-speed space. ©2021 IEEE.
prediction simulation, the simulated vehicle 1 observes an oversized gap ahead. Vehicle 1 is then predicted to accelerate harder or brake more softly to close this gap. The CAV then moves closer to its predecessor in anticipation of exaggerated forward motion that never manifests.

Figure 3.6 assesses the accuracy of hidden vehicle detection (a) and its impact on the acceleration command (b). The single vehicle was correctly identified $90.6 \%$ of the time, even when the first 23 s were spent accumulating data to run the detection algorithm. This accuracy explains the close performance of MPC with ID on to MPC with Ideal ID, where the hidden vehicle was treated as connected. However, this performance required identification of the IDM parameters based on the group of human drivers involved in the experiments. To make this algorithm practical, further development is needed to identify human driver characteristics online and integrate the results with the proposed controller.


Figure 3.6: The identified number of hidden vehicles over time. ©2021 IEEE.

## Chapter 4

## Optimal Lane-Change Decision Making

Where the previous two chapters have only considered longitudinal motion control on a single lane, this chapter adds lane-change decisions as a degree of freedom. The possibility of changing lanes alters the problem substantially. In obstacle avoidance, it is no longer known which obstacles will be ahead of or behind the ego vehicle at a given point in the prediction horizon. Without a fixed car to follow, the automated vehicle needs to be capable of determining its own independent speed trajectory to complete its mission.

This chapter addresses these problems in the following fashion. Modeling is discussed first, including the nonlinear simulation testbed, control-oriented approximations, and extensions to tractor-trailers. The hierarchical architecture that introduces long-term speed planning is described next, followed by the receding horizon obstacle avoidance module and its extensions to tractor-trailers. Chance constraints for handling mixed traffic are presented before proceeding to results in both homogeneous and heterogeneous traffic.

This section draws on published research by Dollar and Vahidi $[31]^{1}[32]^{2}[33]^{3}$, and was supported in part by an award from the U.S. Department of Energy Vehicle Technologies Office (Project No. DEEE0008232). The material in Section 4.4 was developed in collaboration with Cummins.

[^3]
### 4.1 Modeling

Modeling for the multilane guidance system involves nonlinear models for simulation and linear approximations for control. The discussion begins with the nonlinear model.

### 4.1.1 Base Vehicle Model

The multi-agent simulations use a kinematic bicycle model for all vehicles. This model uses a path coordinate frame based on normal and tangential unit vectors, where velocity, acceleration, and yaw angle are states and tangential acceleration command and steering angle command are manipulated inputs. A rate limit of $310 \mathrm{deg} / \mathrm{s}$ is applied to the steering angle $\phi$ based on [59]. First, the longitudinal model from the car-following simulations determines the tangential speed and acceleration $v_{t}$ and $a_{t}$ from the tangential acceleration command $u_{t}$.

$$
\begin{equation*}
\dot{v}_{t}=a_{t}, \quad \dot{a}_{t}=-\frac{1}{\tau} a_{t}+\frac{1}{\tau} u_{t} \tag{4.1}
\end{equation*}
$$

Then, the normal acceleration $a_{n}$ is calculated from the radius of curvature $R$, which is itself a algebraic function of the steering angle and vehicle wheelbase $L$.

$$
\begin{equation*}
a_{n}=\frac{v_{t}^{2}}{R}, \quad R=\frac{L}{\tan \phi} \tag{4.2}
\end{equation*}
$$

The yaw rate $\dot{\psi}$ can also be updated using the steering angle and tangential velocity.

$$
\begin{equation*}
\dot{\psi}=\frac{v_{t}}{L} \tan \phi \tag{4.3}
\end{equation*}
$$

A final coordinate transformation then yields the change in position and new accelerations in the global coordinate frame, where $x$ is the longitudinal position and $y$ is the lateral position.

$$
\begin{gather*}
\dot{x}=v_{t} \cos \psi, \quad \dot{y}=v_{t} \sin \psi  \tag{4.4}\\
a_{x}=a_{t} \cos \psi-a_{n} \sin \psi, \quad a_{y}=a_{n} \cos \psi+a_{t} \sin \psi \tag{4.5}
\end{gather*}
$$

To minimize the complexity that the optimal controller must handle, a lower-level pure pursuit controller as described in Section 1.1.3.1 is implemented to track a target lane. This controller generates a line
from the commanded lane center and computes the steering angle $\phi$ required to track the line.

### 4.1.2 Linear Approximation for Passenger Vehicles

Although the lane change optimization contains integer variables, a linear control model permits the subproblems of the mixed integer program to be QPs. Therefore, the lumped lateral response of the kinematic bicycle model and pure pursuit steering controller is approximated as $2^{\text {nd }}$ order linear. The longitudinal model is retained from car following to form a $5^{\text {th }}$ order model with position $s$, velocity $v$, acceleration $a$, lane $l$, and lane rate $r_{l}$ as states. The lane state $l$ is the lateral position normalized by the lane width such that $l=1$ implies the ego vehicle is centered in the right lane, $l=n_{l}$ implies the ego vehicle is centered in the far left lane, and integer $l$ always corresponds to a lane center, where $n_{l}$ is the number of lanes. The actual lateral position $y_{a}$ is related to $l$ by the formula

$$
\begin{equation*}
l=\frac{y_{a}}{w_{l}}+0.5 \tag{4.6}
\end{equation*}
$$

where $w_{l}$ is the lane width. The inputs to the linear model are the acceleration command $u_{1}$ and the lane command $u_{2}$. Unlike the real-valued lane state, the lane command is integer-valued. This helps build lane discipline into the controller.

$$
\begin{align*}
& \frac{d}{d t}\left[\begin{array}{l}
s \\
v \\
a \\
l \\
r_{l}
\end{array}\right]=\left[\begin{array}{ccccc}
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & -\frac{1}{\tau} & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & -\omega_{n}^{2} & -2 \xi \omega_{n}
\end{array}\right]\left[\begin{array}{l}
s \\
v \\
a \\
l \\
r_{l}
\end{array}\right]+\left[\begin{array}{ll}
0 & 0 \\
0 & 0 \\
\frac{1}{\tau} & 0 \\
0 & 0 \\
0 & K \omega_{n}^{2}
\end{array}\right]\left[\begin{array}{l}
u_{1} \\
u_{2}
\end{array}\right]  \tag{4.7a}\\
& {\left[\begin{array}{lllll}
s & v & a & l & r_{l}
\end{array}\right]^{\mathrm{T}} \in \mathbf{R}^{5}, \quad u_{1} \in \mathbf{R}, \quad u_{2} \in \mathbf{Z} } \tag{4.7b}
\end{align*}
$$

First, a series of unit $u_{2}$ steps were simulated at various speeds. Then, the parameters listed in Table 4.1 were identified to fit the nonlinear response using classical system identification. Figure 4.1 shows the model fit, which was better at higher speeds.

Table 4.1: Linear model parameters.

| Parameter | Definition | Unit | Value |
| :---: | :--- | :--- | :--- |
| $\tau$ | Acceleration time constant | sec | 0.275 |
| $\xi$ | Lane change damping ratio | - | 0.7077 |
| $\omega_{n}$ | Lane change natural frequency | $\mathrm{rad} / \mathrm{s}$ | 0.9666 |
| $K$ | Lane change static gain | - | 1 |





Figure 4.1: Lane step responses at (top to bottom) $24.3 \mathrm{~m} / \mathrm{s}, 7.7 \mathrm{~m} / \mathrm{s}$, and $3.9 \mathrm{~m} / \mathrm{s}$. ©2021 IEEE.

### 4.1.3 Lateral Tractor-Trailer Dynamics

Tractor-trailers are attractive for application of anticipative control, and their lateral dynamics require special modeling attention. Using the commercial dynamic model Truckmaker [122] as ground truth, this section develops a fast-running nonlinear simulation model before approximating it as linear for control purposes. First, lane steps at various speeds were simulated in Truckmaker. Examples of these responses are shown in Fig. 4.2. In addition to the longer lane change time, the symmetry between the head and tail of the lane change differs from the nonlinear passenger car model used to generate Fig. 4.1. This difference combined with the near-constant lane rate during the lane change led to the conjecture that Truckmaker tracks a spacially linear path between the current and target lane.

Based on this conjecture, the pure pursuit controller was deployed to track such a trajectory. The pure pursuit control law is unchanged; only the the desired lateral position $y_{d}$ is now located on the linear target
path. Since the lookahead point is also located on the circle around the ego vehicle's rear axle with radius equal to the lookahead distance $l_{d}$, the roots of the following quadratic equation are candidates for $y_{d}$.

$$
\begin{equation*}
\left(\left(\frac{v t_{l}}{w_{l}}\right)^{2}+1\right) y_{d}^{2}+2\left(\operatorname{sgn} \Delta u_{2} \frac{v t_{l}}{w_{l}}\left(s_{1}-\operatorname{sgn} \Delta u_{2} \frac{v t_{l}}{w_{l}} y_{1}-s\right)-y_{a}\right) y_{d}+\left(s_{1}-\operatorname{sgn} \Delta u \frac{v t_{l}}{w_{l}} y_{1}-s\right)^{2}+y_{a}^{2}-k^{2} v_{t}^{2}=0 \tag{4.8}
\end{equation*}
$$

In Eqn. (4.8), $\operatorname{sgn} \Delta u_{2}$ denotes the sign of the change in lane command that triggered the lane change, that is, leftward lane changes are positive and rightward lane changes are negative. The pure pursuit parameters $k$ and $t_{l}$ have the same meaning as in the passenger car application. The lane width and actual lateral position are denoted $w_{l}$ and $y_{a}$, respectively. The longitudinal position $s_{1}$ is the location where the lane change path's ramp begins, which must be stored in memory until the lane change is complete.

Figure 4.2 compares the nonlinear model obtained from the kinematic bicycle model, pure pursuit controller, and ramp path to the Truckmaker response. The two are nearly identical, so this simplified nonlinear model was used as the plant in subsequent control simulations. An advantage of the ramp trajectory is that it removes the initial acceleration spike caused by the stepped path.


Figure 4.2: Various models of lateral tractor motion.

A trailer model is also needed for collision avoidance simulations. Since the tractor dynamics were replicated with sufficient accuracy using a kinematic model, one was also used for the trailer. See Fig. 4.3 for a schematic of the trailer model. The no-slip condition on the trailer's rear axle yields the following model for the trailer's yaw rate, where $L_{t}$ denotes the length of the trailer.


Figure 4.3: Diagram of the trailer including the hitch point $H$ and rear point $R$.

$$
\begin{equation*}
\dot{\gamma}=\frac{w_{l} r_{l} \cos \gamma-v_{H x} \sin \gamma}{L_{t}} \tag{4.9}
\end{equation*}
$$

Since the trailer is rigid and attached to the tractor at point $H$, the angle $\gamma$ and the tractor position suffice to fix the state of the trailer. The trailer's lag behind the tractor in a lane change varies with speed as shown in Fig. 4.4.

Equation (4.9) gives a nonlinear state-space description of the trailer state $\gamma$. For control, a linearized form along with linearized output equations for the longitudinal and lateral positions of the trailer's rear edge are needed. Applying the small angle approximations $\sin \theta=\theta, \cos \theta=1, \tan \theta=\theta$, and $\theta^{2}=0$ to Eqn. (4.9) yields

$$
\begin{equation*}
\dot{\gamma}=\frac{w_{l}}{L_{t}} r_{l}-\frac{1}{L_{t}} \gamma v \tag{4.10}
\end{equation*}
$$

The linearization is still incomplete since the second term includes the product of $\gamma$ and $v$. Applying the first-order Taylor expansion about $\gamma=\gamma_{0}$ and $v=v_{0}$ resolves this issue.

$$
\begin{equation*}
\dot{\gamma}=\frac{w_{l}}{L_{t}} r_{l}-\frac{1}{L_{t}} \gamma_{0} v-\frac{1}{L_{t}} v_{0} \gamma+\frac{2}{L_{t}} \gamma_{0} v_{0} \tag{4.11}
\end{equation*}
$$

A linearization point must now be chosen that will apply to the entire prediction horizon. Since the vehicle is expected to remain in the vicinity of $\gamma=0$ with small deviations during lane changes, $\gamma_{0}=0$ is selected. The same linearization point would be unsuitable for speed since long cruising intervals are expected. Therefore, the model is linearized at each step around $v_{0}=v(k)$. Equation (4.12) describes the linear parameter varying


Figure 4.4: Models of lateral trailer motion.
(LPV) model for trailer yaw rate.

$$
\begin{equation*}
\dot{\gamma}=\frac{w_{l}}{L_{t}} r_{l}-\frac{1}{L_{t}} v_{0} \gamma \tag{4.12}
\end{equation*}
$$

The output equations for the trailer longitudinal and lateral positions $s_{t}$ and $l_{t}$, where $s_{t}$ has units of distance and $l_{t}$ has units of lanes, are

$$
\begin{equation*}
s_{t}=s-L_{t}, \quad l_{t}=l-\frac{L_{t}}{w_{l}} \gamma \tag{4.13}
\end{equation*}
$$

which make use of the small angle approximation. Comparison of this LPV model to the nonlinear one is provided in Fig. 4.4.

### 4.1.4 Linearized Modeling of Road Grade and Aerodynamic Drag

With an eye toward tractor-trailer applications where acceleration constraints are especially restrictive, a longitudinal LPV model is developed using tractive acceleration rather than body acceleration as a manipulated input. This model is inspired by [4] and adds the effect of road grade $\theta$, which is approximated
as a linear function of position $s$.

$$
\begin{equation*}
\theta(s) \approx b_{0}+b_{1} s \tag{4.14}
\end{equation*}
$$

The coefficients $b_{0}$ and $b_{1}$ are set using least-squares fitting such that $\theta\left(s_{0}\right)=\theta_{0}=b_{0}+b_{1} s_{0}$ exactly and $b_{1}$ best fits the road grade data over the expected position horizon. This position horizon is obtained from the trip planner described in Section 4.2. Figure 4.5 provides an example of such a fit.


Figure 4.5: Least-squares road grade approximation for receding horizon control.
Derivation of the linearized model begins with the state space equation for $\dot{v}$ that results from dividing the longitudinal force balance by the mass. Let $g, \mu, m, \rho_{a}, C_{d}$, and $A$ denote the gravitational acceleration, vehicle mass, air density, drag coefficient, and frontal area, respectively.

$$
\begin{equation*}
\dot{v}=a_{t}-g \mu \cos \theta-g \sin \theta-\frac{1}{2 m} \rho_{a} C_{d} A v^{2} \tag{4.15}
\end{equation*}
$$

Substituting Eqn. (4.14) yields

$$
\begin{equation*}
\dot{v} \approx a_{t}-g \mu \cos \left(b_{0}+b_{1} s\right)-g \sin \left(b_{0}+b_{1} s\right)-\frac{1}{2 m} \rho_{a} C_{d} A v^{2} \tag{4.16}
\end{equation*}
$$

which, after the first-order expansion, becomes

$$
\begin{align*}
\dot{v} \approx & a_{t}+g b_{1}\left(\mu \sin \theta_{0}-\cos \theta_{0}\right) s-\frac{1}{m} \rho_{a} C_{d} A v_{0} v  \tag{4.17}\\
& -g b_{1}\left(\mu \sin \theta_{0}-\cos \theta_{0}\right) s_{0}-g\left(\mu \cos \theta_{0}+\sin \theta_{0}\right)+\frac{1}{2 m} \rho_{a} C_{d} A v_{0}^{2}
\end{align*}
$$

Equation (4.17) can be more compactly written in terms of the constants $c_{0}, c_{1}$, and $c_{2}$.

$$
\begin{equation*}
\dot{v} \approx c_{1} s+c_{2} v+a_{t}+c_{0} \tag{4.18}
\end{equation*}
$$

where

$$
\begin{gather*}
c_{0}:=-g b_{1}\left(\mu \sin \theta_{0}-\cos \theta_{0}\right) s_{0}-g\left(\mu \cos \theta_{0}+\sin \theta_{0}\right)+\frac{1}{2 m} \rho_{a} C_{d} A v_{0}^{2}  \tag{4.19a}\\
c_{1}:=g b_{1}\left(\mu \sin \theta_{0}-\cos \theta_{0}\right)  \tag{4.19b}\\
c_{2}:=-\frac{1}{m} \rho_{a} C_{d} A v_{0} \tag{4.19c}
\end{gather*}
$$

As in the body acceleration model, tractive acceleration is related to the commanded tractive acceleration by a first-order lag.

$$
\begin{equation*}
\dot{a}_{t}=-\frac{1}{\tau} a_{t}+\frac{1}{\tau} u_{1} \tag{4.20}
\end{equation*}
$$

Equation (4.21) provides the combined linear model used for obstacle avoidance in tractor-trailer applications. The $2^{\text {nd }}$ order lane change model parameters $\omega_{n}$ and $\xi$ maintain their meanings as the natural frequency and damping ratio of the step response to $u_{2}$.

$$
\frac{d}{d t}\left[\begin{array}{l}
s  \tag{4.21}\\
v \\
a \\
1 \\
l \\
r_{l} \\
\gamma
\end{array}\right]=\left[\begin{array}{ccccccc}
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
c_{1} & c_{2} & 1 & c_{0} & 0 & 0 & 0 \\
0 & 0 & -\frac{1}{\tau} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & -\omega_{n}^{2} & -2 \xi \omega_{n} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{w_{l}}{L_{t}} & -\frac{1}{L_{t}} v_{0}
\end{array}\right]\left[\begin{array}{l}
s \\
v \\
a \\
1 \\
l \\
r_{l} \\
\gamma
\end{array}\right]+\left[\begin{array}{cc}
0 & 0 \\
0 & 0 \\
\frac{1}{\tau} & 0 \\
0 & 0 \\
0 & 0 \\
0 & \omega_{n}^{2} \\
0 & 0
\end{array}\right]\left[\begin{array}{l}
u_{1} \\
u_{2}
\end{array}\right]
$$

### 4.2 Hierarchical Control Using Pontryagin's Minimum Principle

Pontryagin's Minimum Principle (PMP) has its advantages, especially in problems where it can be solved algebraically. In such cases, it delivers a solution considering the entire time span of the optimal control problem with minimal computation time. It can also include nonlinearities in the objective and constraints.

However, PMP can quickly become intractable as constraints are added. In contrast, receding horizon model predictive control handles constraints well, even when they become numerous. Its main disadvantage is the finite horizon. In [33], the author proposed the following hierarchical architecture with nomenclature listed in Table 4.2 to combine the benefits of these two techniques.


Figure 4.6: Block diagram of the hierarchical automated driving system.

Table 4.2: Block diagram nomenclature.

| Symbol | Definition |
| :---: | :--- |
| $s_{S V}$ | Surrounding vehicle's longitudinal position |
| $l_{S V}$ | Surrounding vehicle's lane |
| $s_{f}$ | Ego vehicle's goal position |
| $t_{f}$ | Ego vehicle's goal time |
| $s_{f s}$ | Surrounding vehicle's goal position |
| $l_{r e f}$ | Reference lane |
| $z$ | State vector |
| $\tilde{z}$ | Longitudinal state vector |
| $\tilde{z}^{*}$ | Reference longitudinal state vector |
| $\tilde{u}^{*}$ | Reference acceleration command |
| $u_{1}$ | Acceleration command in the road frame |
| $u_{2}$ | Lane command |
| $\phi$ | Steering angle |

This architecture involved a pacing module that uses PMP to provide a reference trajectory to an MPC-based maneuvering module. The PMP analysis uses the reduced-order double integrator model shown in the optimal control problem (4.23) and its state and control input results $\tilde{z}^{*}$ and $\tilde{u}^{*}$ become the MPC's references according to the relations

$$
\begin{gather*}
z_{\text {ref }}(i)=\left[\begin{array}{lll}
\tilde{z}^{* T}(i) & \tilde{u}^{*}(i) & l_{\text {ref }} \\
0
\end{array}\right]^{\mathrm{T}},  \tag{4.22a}\\
u_{r e f}(i)=\left[\begin{array}{ll}
\tilde{u}^{*}(i+1) & l_{r e f}
\end{array}\right]^{\mathrm{T}} \tag{4.22b}
\end{gather*}
$$

where the one-step time shift applied to $\tilde{u}^{*}$ approximates the first-order lag in the MPC model. The function
of these references is to cause the MPC to track the long-term optimal trajectory as closely as possible. If no obstacles are present, the MPC tracks the shrinking-horizon reference. To understand the advantage of such an architecture, consider the performance of an acceleration-minimizing MPC without the pacing module, starting from rest in the reference lane. The optimal trajectory is to remain stationary with zero acceleration. A speed reference can be added to the MPC to cause the vehicle to move; in fact, the author did so in [31]. However, the form of the acceleration to the target speed is not generally optimal in the long run.

The implementation of the PMP-based pacing module is flexible, but the integrated controller previously developed by the author minimized the integral of the square of acceleration. In a collaboration with Antonio Sciarretta, Laurent Thibault, and Mohamed Laraki at IFP Energies nouvelles, the author also solved the PMP problem for diesel-engine nitrogen oxide $\mathrm{NO}_{\mathrm{x}}$ minimization with results in review [28].

Acceleration minimization is the objective of the pacing module in this chapter. The design reasoning is twofold. On an individual vehicle level, acceleration minimization can reduce unnecessary braking relative to human drivers or naïve algorithms. On a collective level, acceleration minimization can also realize secondary energy benefits through traffic smoothing. The corresponding OCP is given below, where response lag is neglected to reduce model order and the control input $\tilde{u}$ is the exact acceleration. The boundary conditions $s_{f}$ and $t_{f}$ until the next stop need to be defined as inputs. It is envisioned that this would be accomplished by a routing algorithm like [3] that considers travel times over links.

$$
\begin{align*}
& \min \tilde{J}=\int_{t_{0}}^{t_{f}} \tilde{u}^{2} d t \\
& \text { s.t. } s\left(t_{0}\right)=s_{0}, s\left(t_{f}\right)=s_{f} \\
& \quad v\left(t_{0}\right)=v_{0}, v\left(t_{f}\right)=0  \tag{4.23}\\
& \quad \dot{s}=v, \dot{v}=\tilde{u} \\
& \quad v \leq \bar{v}
\end{align*}
$$

The solution, derived in [33], is piecewise and consists of parabolic velocity phases, possibly adjacent to a constant maximum velocity phase. If a purely parabolic velocity profile satisfies the maximum speed constraint, that profile is the solution. Otherwise, the following system is solved by using the boundary conditions on speed and cumulative distance to find the coefficients $c_{1}, c_{2}$ and times $t_{1}$ and $t_{2}$. When $v_{0}<\bar{v}$ and the constraint
is active:

$$
\tilde{u}^{*}(t)=\left\{\begin{array}{llr}
\frac{1}{2} c_{1} t-c_{2}^{\mathrm{I}} & ; & t<t_{1}  \tag{4.24}\\
0 & ; & t_{1} \leq t<t_{2} \\
\frac{1}{2} c_{1} t-c_{2}^{\mathrm{III}} & ; & t_{2} \leq t<t_{f}
\end{array}\right.
$$

When $v_{0} \geq \bar{v}$ :

$$
\tilde{u}^{*}(t)=\left\{\begin{array}{llr}
0 & ; & t \leq t_{1}  \tag{4.25}\\
\frac{1}{2} c_{1} t-c_{2} & ; & t_{1} \leq t<t_{f}
\end{array}\right.
$$

Details and a plot of this solution can be found in [33]. Refernece [109] shows that this velocity profile is also energy-optimal for EVs. This might be expected since physically, EVs experience resistance losses proportional to the square of motor torque.

### 4.3 Receding Horizon Control Using Mixed Integer Programming

Position constraints to avoid collisions are omitted from OCP (4.23) to keep the problem analytically tractable for a general number of obstacles. As illustrated in Fig. 4.6, a numerical receding horizon controller handles such constraints. This model-based controller uses the combined longitudinal and lateral model of Section 4.1.2 with the following objective function where $z_{e}=z-z_{r e f}$ is the combined lateral and longitudinal state deviation from the reference $z_{r e f}$ and, similarly, $u_{e}$ is the deviation from the reference control input. Recall that these references come from the pacing module according to Eqn. (4.22).

$$
\begin{gather*}
J=z_{e}^{\mathrm{T}}(N) P z_{e}(N)+\sum_{i=0}^{N-1}\left[z_{e}^{\mathrm{T}}(i) Q z_{e}(i)+u_{e}^{\mathrm{T}}(i) R u_{e}(i)\right]  \tag{4.26a}\\
Q=\left[\begin{array}{lllll}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & q_{l} & 0 \\
0 & 0 & 0 & 0 & 0
\end{array}\right], \quad P=\left[\begin{array}{ccccc}
q_{s} & 0 & 0 & 0 & 0 \\
0 & q_{v} & 0 & 0 & 0 \\
0 & 0 & q_{a} & 0 & 0 \\
0 & 0 & 0 & q_{l} & 0 \\
0 & 0 & 0 & 0 & 0
\end{array}\right]  \tag{4.26b}\\
R=\left[\begin{array}{ll}
q_{a} & 0 \\
0 & q_{l}
\end{array}\right] \tag{4.26c}
\end{gather*}
$$

Longitudinally, this quadratic objective penalizes the deviation from the reference acceleration command and terminal state with weights $q_{a}, q_{v}$, and $q_{s}$. Laterally, a constant target lane is tracked with weight $q_{l}$ to accommodate keep-right rules and navigational needs. The penalty weights are tuned such that the ego vehicle will leave its reference lane to avoid slowing down for a slow-moving vehicle ahead.

The main challenge in the receding horizon lane optimization is the collision avoidance constraints. Since the ego vehicle can permissibly drive either ahead of or behind any obstacle and can pass beside the obstacle on a different and unspecified lane, the constraints involve logical disjunctions. Hence the problem is cast as a mixed integer quadratic program (MIQP). Equation (4.27) introduces the collision avoidance constraints for safe gap $\underline{d}$, which are repeated for all lanes $\lambda$ and obstacles $\zeta$. The positions $s_{\text {min }}^{\zeta}$ and $s_{\text {max }}^{\zeta}$ are the downstream and upstream edges of obstacle $\zeta$, respectively. The collision avoidance constraints are softened with slack $\epsilon_{1}$. Small violations are penalized quadratically, but a second linear penalty is imposed before a collision would occur.

$$
\begin{gather*}
-s-M\left(2-\mu_{\lambda a}-\mu_{\lambda b}\right)-M\left(1-\beta_{\zeta}\right) \leq-s_{\min }^{\zeta}-\underline{d}+\epsilon_{1}  \tag{4.27a}\\
s-M\left(2-\mu_{\lambda a}-\mu_{\lambda b}\right)-M \beta_{\zeta} \leq s_{\max }^{\zeta}-\underline{d}-l_{v}+\epsilon_{1} \tag{4.27b}
\end{gather*}
$$

Equation (4.27) relies on binary variables $\beta_{\zeta}, \mu_{\lambda a}$, and $\mu_{\lambda b}$. The front-rear binary $\beta_{\zeta}$ is a standard Big M variable as described in Section 1.2.2.2. The term $2-\mu_{\lambda a}-\mu_{\lambda b}$ acts as an indicator that equals 0 if the ego vehicle is in lane $\lambda$ and 1 otherwise, that is,

$$
2-\mu_{\lambda a}-\mu_{\lambda b}=\left\{\begin{array}{cr}
0 & \lambda-\delta \leq l \leq \lambda+\delta  \tag{4.28}\\
1 & \text { otherwise }
\end{array}\right.
$$

where $\delta$ is the maximum deviation from a lane's centerline when a part of the vehicle still resides in that lane. The following constraints on $l$ guarantee proper function of $\mu_{\lambda a}$ and $\mu_{\lambda b}$.

$$
\begin{equation*}
-l-M\left(1-\mu_{\lambda a}\right) \leq-\lambda+\delta, \quad l-M \mu_{\lambda a} \leq \lambda-\delta \tag{4.29a}
\end{equation*}
$$

$$
\begin{equation*}
l-M\left(1-\mu_{\lambda b}\right) \leq \lambda+\delta, \quad-l-M \mu_{\lambda b} \leq-\lambda-\delta \tag{4.29b}
\end{equation*}
$$

### 4.4 Extension to Tractor-Trailers

Tractor-trailers require special attention because the rear of the trailer deviates laterally from the tractor during lane changes. This impacts the occupied lanes and therefore the indicator variable setup. Rather than introducing additional binary variables for the trailer, the constraints are modified to make use of the existing two lane indicators $\mu_{\lambda a}$ and $\mu_{\lambda_{b}}$ for each lane.

The following indicator setup equations are retained from the passenger car formulation.

$$
\begin{equation*}
l-M \mu_{\lambda a} \leq \lambda-\delta, \quad-l-M \mu_{\lambda b} \leq-\lambda-\delta \tag{4.30}
\end{equation*}
$$

The remaining passenger vehicle indicator setup constraints from Eqn. (4.29) are omitted. While they strengthen the formulation by excluding some solutions, the constraints above are sufficient to guarantee that $\mu_{\lambda a}$ and $\mu_{\lambda b}$ are equal to 1 when needed. Otherwise, the solver automatically selects 0 if it is possible and necessary to relax the position constraint in an optimal solution. The constraints below guarantee that the indicators also reflect the trailer's lateral position.

$$
\begin{equation*}
l_{t}-M \mu_{\lambda a} \leq \lambda-\delta, \quad-l_{t}-M \mu_{\lambda b} \leq-\lambda-\delta \tag{4.31}
\end{equation*}
$$

Substituting the small-angle approximation for the trailer's lateral position yields

$$
\begin{equation*}
l-\frac{L_{t}}{w_{l}} \gamma-M \mu_{\lambda a} \leq \lambda-\delta, \quad-l+\frac{L_{t}}{w_{l}} \gamma-M \mu_{\lambda b} \leq-\lambda-\delta \tag{4.32}
\end{equation*}
$$

Notice that the constraints above are effective even in the extreme case where the trailer straddles lane $\lambda$ but neither the tractor nor the trailer have their rear edges located in lane $\lambda$. The component (i.e. tractor or trailer) with the larger lateral position will force $\mu_{\lambda a}=1$ and the other with the lower lateral position will force $\mu_{\lambda b}=1$. This imposes an identical tightening of the position constraint to what would occur if the entire truck were in lane $\lambda$. However, the validity of the small angle approximation used to linearly model $\gamma$ should be further investigated in this case.

Longitudinally, the small angle approximation $\cos \gamma \approx 1$ results in the following modified minimum position constraint. The passenger car maximum constraint still applies by letting the vehicle length equal the tractor length.

$$
\begin{equation*}
-s-M\left(2-\mu_{\lambda a}-\mu_{\lambda b}\right)-M\left(1-\beta_{\zeta}\right) \leq-s_{\min }^{\zeta}-\underline{d}-L_{t}+\epsilon_{1} \tag{4.33a}
\end{equation*}
$$

$$
\begin{equation*}
s-M\left(2-\mu_{\lambda a}-\mu_{\lambda b}\right)-M \beta_{\zeta} \leq s_{\max }^{\zeta}-\underline{d}-l_{v}+\epsilon_{1} \tag{4.33b}
\end{equation*}
$$

The issue of highly non-convex powertrain operating spaces observed in Section 2.2 applies here as well. Since an LPV approach is already needed to capture road grade and trailer dynamics, a more accurate model of maximum acceleration's dependence on speed is developed. The model is based on constant maximum tractive acceleration in each gear, with gear changing as a function of speed. The maximum tractive acceleration $\bar{u}_{t}\left(v_{0}\right)$ at the current speed $v_{0}$ is thus looked up and used as the linearization point. The linearized constraint has the structure

$$
\begin{equation*}
u_{t} \leq \bar{u}_{t}\left(v_{0}\right)+m_{u}\left(v-v_{0}\right) \tag{4.34}
\end{equation*}
$$

so that planned changes in speed are approximately accounted for in the future acceleration constraint.
One of two methods is used to determine the coefficient $m_{u}$ depending on speed. Normally, $m_{u}$ is obtained by linearizing a constant-power approximation, which leads to the formula

$$
\begin{equation*}
\tilde{m}_{u}=-\frac{\bar{u}_{t}\left(v_{0}\right)}{v_{0}} . \tag{4.35}
\end{equation*}
$$

At lower speeds, this tangent-line approach can prevent the controller from planning to reach maximum speed. Therefore, the slope $m_{u}$ is limited according to

$$
\begin{equation*}
m_{u}=\max \left\{\tilde{m}_{u},-\frac{\bar{u}_{t}\left(v_{0}\right)}{\bar{v}-v_{0}}\right\} . \tag{4.36}
\end{equation*}
$$

Figure 4.7 illustrates the piecewise-constant maximum acceleration function and its linearized approximation.


Figure 4.7: An example of the online-linearized truck acceleration constraint.

### 4.5 Chance Constraints

Additional safety margin for collision avoidance is needed when dealing with unconnected surrounding vehicles whose future intentions are not known. However, worst-case constraints can result in excessive following distances and reduced road throughput. Instead, chance constraints seek to avoid collisions assuming reasonable surrounding vehicle (SV) behavior. In this context, the collision avoidance constraints are expressed in terms of the expected SV edges $E S_{\text {min }}^{\zeta}$ and $E S_{\text {max }}^{\zeta}$ where $S$ is a random variable. The goal of the chance constraint calculation is to find the safe distance $d_{r}$.

$$
\begin{gather*}
-s-M\left(2-\mu_{\lambda a}-\mu_{\lambda b}\right)-M \beta_{\zeta}^{\mathrm{C}} \leq-E S_{\min }^{\zeta}-d_{r}+\epsilon_{1}  \tag{4.37a}\\
s-M\left(2-\mu_{\lambda a}-\mu_{\lambda b}\right)-M \beta_{\zeta} \leq E S_{\max }^{\zeta}-d_{r}+\epsilon_{1} \tag{4.37b}
\end{gather*}
$$

$$
\beta_{\zeta}, \mu_{\lambda a}, \mu_{\lambda b} \in\{0,1\}
$$

This section will present a sketch of the chance constraint derivation. For a more complete version, see [5]. First, the SV's position and speed are assumed to be measured with certainty at the current time and the uncertainty arises from the SV's acceleration. These assumptions are formalized in the following initial covariance $\Lambda_{0}$.

$$
\Lambda_{0}=\left[\begin{array}{lll}
0 & 0 & 0  \tag{4.38}\\
0 & 0 & 0 \\
0 & 0 & \sigma_{a}^{2}
\end{array}\right]
$$

The following formula for the SV state covariance at future stage $i$ is derived from properties of the multivariate normal distribution under linear transformation [45]. The upper-left element of $\Lambda_{i}$ is the variance $\sigma_{s}^{2}$ of the SV position.

$$
\begin{equation*}
\Lambda_{i}=A^{i} \Lambda_{0}\left(A^{i}\right)^{T} \tag{4.39}
\end{equation*}
$$

Thus the SV position $S \sim \mathcal{N}\left(E S, \sigma_{s}^{2}\right)$. The probabilistic goal of the chance constraints is formally restated.

$$
\begin{equation*}
\operatorname{Pr}\left(S_{\max } \geq s \mid \beta_{\zeta}=0 \cap\left(2-\mu_{\lambda a}-\mu_{\lambda b}\right)=0\right) \geq \alpha \tag{4.40a}
\end{equation*}
$$

$$
\begin{equation*}
\operatorname{Pr}\left(S_{\min } \leq s \mid \beta_{\zeta}=1 \cap\left(2-\mu_{\lambda a}-\mu_{\lambda b}\right)=0\right) \geq \alpha \tag{4.40b}
\end{equation*}
$$

Then, the derivation in [5] is followed to obtain the following formula for $d_{r}$ where $F_{s}^{-1}(\alpha)$ is the inverse cumulative distribution function of the SV position evaluated at the desired safe probability $\alpha$.

$$
\begin{equation*}
d_{r}=F_{S}^{-1}(\alpha)-E S \tag{4.41}
\end{equation*}
$$

Selection of $\alpha$ is a design choice. Since the closed-loop controller has greater opportunity to correct for prediction errors that occur farther in the future, $\alpha$ is specified as a linear function of prediction step such that $\alpha(i=1)=0.99999$ and $\alpha(i=N)=0.5$. The latter condition causes $d_{r}=0$ for $i=N$. Figure 4.8, reproduced from Dollar and Vahidi [32], shows the function $F_{s}^{-1}$ at various $i$ and $\alpha$ along with the results of Eqn. (4.42). This implementation uses a 0.4 s timestep with a prediction horizon of $N=25$.

$$
\begin{equation*}
\alpha(i)=-0.020833 i+1.020823 \tag{4.42}
\end{equation*}
$$



Figure 4.8: Safe distance as a function of prediction step and robustness [32].

### 4.6 Results

The decentralized mixed-integer programming lane change algorithm was evaluated in phases as features were added. The results of the initial two-lane formulation were published in [31]. Then, chance constraints were added and assessed in mixed traffic in [32]. Reference [33] presents the results of the multilane formulation in two fully connected environments. The same algorithm was also evaluated in mixed traffic and the results were submitted in various U.S. Department of Energy project reports including [124].

### 4.6.1 Two Lanes with Full Connectivity

This preliminary assessment used only the maneuvering module and omitted the nonlinear simulation model. It involved 4 CAVs with target speeds of $\{35,32,29,26\} \mathrm{m} / \mathrm{s}$, assigned full-factorial, passing a slowmoving vehicle that traveled at approximately $4.5 \mathrm{~m} / \mathrm{s}$. Performance was compared to a reactive model that combined the Intelligent Driver Model with gap-based lane change rules. The optimal algorithm reduced fuel consumption by $8.4 \%$. Further comparison to ideal constant-velocity travel revealed that the fuel saved amounted to $80 \%$ of the congestion-induced fuel consumption. Moreover, travel time was also reduced by $6.2 \%$ as shown in Fig. 4.9, reproduced from Dollar and Vahidi [31].


Figure 4.9: Comparison of aggregate rule-based and MPC results to the ideal free-flow case [31].

Computation time was measured for the MATLAB implementation. The mean times shown in Fig. 4.10, reproduced from [31], are promising for real-time implementation, especially when move blocking is used to reduce the size of the MIQP. This does not, however, exclude overruns in rare cases.


Figure 4.10: Mean computation time during a sample reference velocity case [31].

### 4.6.2 Collision Avoidance in Merging

The state of California publishes police reports of automated vehicle collisions online. When Apple's prototype AV experienced its first collision, the author analyzed the police report and a map of the relevant intersection to reproduce the scenario in simulation [32]. A human driver rear-ended the AV while the AV was nearly stopped as it waited for a safe merging opportunity. By applying exceptionally dangerous human driver model parameters and introducing a perception fault in the rear-ending vehicle, a hazardous scenario was simulated to stress-test the chance-constrained algorithm. Figure 4.11, reproduced from [32], depicts the scenario, which also serves as an example of the custom MATLAB simulator's capability.


Figure 4.11: Approximate road geometry with the AV (solid) and surrounding vehicles (hatched) [32].
The three algorithm variants Yield Except Following (YEF), Yield To All (YTA), and Yield Except Following with Avoidance (YEFa) differ in the set of surrounding vehicles that they include in the constraints. YEF omits the following vehicle in the same lane from the constraints similarly to car following algorithms that only consider the lead vehicle. YTA simply includes all surrounding vehicles in the constraints. YEFa typically ignores the following vehicle, but adds it to the constraints when an imminent collision is detected. In such an event, the chance constraints are also relaxed to $\alpha=0.5$ so that they ego vehicle drives more aggressively when attempting to avoid a collision.

Since YEF does not attempt to avoid the following vehicle even under a perception fault, it is expected that it incurred the most collisions. YTA reduced the number of collisions compared to YEF, but YEFa had the fewest AV-involved collisions overall. Figure 4.12, reproduced from [32], compares these results. Looking deeper, most YEF collisions involved the AV being rear-ended as in the real Apple collision. The AV would not have been at-fault in such incidents.


Figure 4.12: Frequency of AV-involved collisions with $95 \%$ confidence intervals [32].

### 4.6.3 Multiple Lanes Including Partial Connectivity

With multi-lane capability, chance constraints for mixed traffic, and the pacing module implemented, the lane selection algorithm was evaluated in multi-lane environments. Highway merge and arterial scenarios were evaluated with homogeneous traffic. Then, the percentage of traffic made up of CAVs was varied in the arterial scenario to obtain mixed traffic results.

Figure 4.13 provides a detailed view of several vehicles' optimal lane change plans during the highway merge. The color gradients show the progression of prediction time. Longitudinal acceleration $u_{1}$ and velocity $v$ are also shown for the ego vehicle, whose index is 1 .


Figure 4.13: Position plans including the ego's (green), with the ego's longitudinal plan. ©2021 IEEE.

Figure 4.14 demonstrates the impact of the planning module on collective control performance. Dashed lines mark the separation of the right lane and the start of the left turning lane. An orderly longterm speed profile is obtained on average despite the trip's length exceeding that of the receding horizon. This is possible because the RHC's reference results from the full-length optimal control problem.

Table 4.3 lists the quantitative improvements over the baseline algorithm with $100 \%$ CAVs. As in the two-lane preliminary results, both travel time and energy benefits were realized.


Figure 4.14: Speed profiles and 20 m space-averages in the arterial scenario. ©2021 IEEE.

Table 4.3: Change relative to IDM-RB in multi-lane guidance.

| 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 |

In the mixed traffic simulations, travel time was held approximately constant to isolate the energy savings shown in Table 4.4. Under these conditions, $16 \%$ energy savings was attained at $100 \%$ CAVs with steady improvements as CAVs were added.

Table 4.4: Change relative to IDM-RB in the arterial scenario at various CAV fractions.

| CAV Fraction | $1 / 3$ | $2 / 3$ | 1 |
| :---: | :--- | :--- | :--- |
| Energy Chng. [\%] | -3.4 | -9.4 | -16.2 |
| Time Chng. [\%] | 0.0 | +0.8 | -0.7 |

## Chapter 5

## Advanced Long-Term Speed Planning

This chapter will enhance the existing shrinking horizon planning module by relaxing the flat road assumption, directly minimizing wheel-to-distance energy, and introducing position-varying speed limits. An additional anticipative speed constraint based on macroscopic traffic conditions is also explored, although it did not improve performance in simulation and was not needed to meet the proposed targets. These innovations will be evaluated in a Class 8 truck application using microscopic simulations designed to target specified macroscopic traffic conditions.

The planning module described in Section 4.2 adds a shrinking horizon reference to the receding horizon controller. This allows the receding horizon controller to follow an optimal eco-driving trajectory until the next stopping point, even if the time until that stopping point exceeds the finite horizon. However, Section 4.2's planning module is limited to a single maximum speed limit. This means it cannot account for regulatory speed limit changes or physical speed limits around curves, nor can it comprehend bulk traffic speed changes like those caused by congestion. Section 4.2 also neglected road slope, which is a consideration for all vehicles but especially important for heavy tractor-trailers with more restrictive maximum acceleration constraints.

Reference [7] proposed a dynamic programming (DP) approach to long-term planning. However, closed-loop operation in traffic demands that the optimization be repeated online. This presents a problem for DP because of its sensitivity to discretization and high computation times that increase exponentially with the number of states [71]. The method proposed in this section reformulates the problem as a linear program (LP) for greatly improved computability and scalability. LP can be solved in polynomial time [67] and several commercial and open source solvers exist [39] [44] [77].

This research was conducted in collaboration with Ali Borhan, Bibin Pattel, and Jingxuan Liu at Cummins. The author's contribution was the LP formulation and MATLAB simulation environment.

### 5.1 Performance Requirements

Table 5.1 summarizes the requirements stated in the research proposal. The lateral model deviation target was set between known tolerable and intolerable deviations at low speed in past studies; see Fig. 4.1. The fuel economy target was set based on improvements realized in Chapter 4, which addressed a similar problem with more assumptions. Anticipating that chattering would reduce the acceptability of a given acceleration level, the acceleration chattering magnitude requirement was set to under half of the subjective "just noticeable" threshold listed in [52]. Although it was not a specific requirement in the proposal, near real-time computation times are desirable. Some increase over real time is accepted because of the addition of detail and features compared to Chapter 4 and the use of MATLAB in algorithm prototyping.

Table 5.1: Performance targets.

| Metric | Target |
| :--- | :---: |
| Lateral model deviation | 0.1 lanes |
| Acceleration chattering magnitude | $0.2 \mathrm{~m} / \mathrm{s}^{2}$ |
| Fuel economy improvement | $10 \%$ |

### 5.2 Architecture

The LP speed planner is implemented as part of the hierarchical lane change algorithm described in Section 4.2 as shown in Fig. 5.1. The new or updated components are highlighted and include the dynamic model, macroscopic traffic prediction, planning module core, and maneuvering module core. The signals exchanged between the modules match Chapter 4 except that tractive acceleration is used in place of body acceleration. In Fig. 5.1, the state is denoted $x$ instead of $z$ to reserve $z$ for another purpose in this chapter.

### 5.3 Core Formulation

The core formulation includes the dynamic modeling, objective, constraints, and approximation techniques that enable the eco-driving problem over general sloping roads to be cast as an LP.


Figure 5.1: Block diagram of the LP speed planner's integration with obstacle avoidance.

### 5.3.1 Dynamic Model

A work-energy approach is used to model the vehicle's longitudinal dynamics with position as the independent variable. The position step is denoted $i$. The basic equation is the first law of thermodynamics applied to one step, where $\Delta K E$ and $\Delta P E$ are the changes in kinetic and potential energy, respectively, $W_{f}$ is the frictional work excluding actuators, $E_{e}$ is the energy added to the vehicle by the powertrain, and $E_{b}$ is the energy removed from the vehicle as heat by friction braking.

$$
\begin{equation*}
\Delta K E+\Delta P E+W_{f}=E_{e}-E_{b} \tag{5.1}
\end{equation*}
$$

The terms of Eqn. (5.1) are now developed in more detail. Beginning with the kinetic and potential energies,

$$
\begin{equation*}
\Delta K E=\frac{1}{2} m v^{2}(i+1)-\frac{1}{2} m v^{2}(i), \quad \Delta P E=m g \Delta h=m g \Delta s \sin \theta \tag{5.2}
\end{equation*}
$$

where $m$ denotes the vehicle mass, $v$ the speed, $g$ the gravitational acceleration, $\Delta h$ the elevation change, $\Delta s$ the position step tangent to the road, and $\theta$ the road slope. The unactuated frictional work is computed using the following model.

$$
\begin{equation*}
F_{f}=\mu_{r} m g \cos \theta+\frac{1}{2} \rho_{a} C_{d} A_{f} v^{2}, \quad W_{f}=F_{f} \Delta s \tag{5.3}
\end{equation*}
$$

Substituting Eqns. (5.2) and (5.3) into Eqn. (5.1) yields the dynamic model

$$
\begin{equation*}
\frac{1}{2} m v^{2}(i+1)-\frac{1}{2} m v^{2}(i)+m g \Delta s \sin \theta(i)+\mu_{r} m g \Delta s \cos \theta(i)+\frac{1}{2} \rho_{a} C_{d} A_{f} \Delta s v^{2}(i)=E_{e}(i)-E_{b}(i) . \tag{5.4}
\end{equation*}
$$

Notice that all terms in Eqn. (5.4) are either known since the road grade profile is a known function of position, or vary with the square of speed. To take advantage of this fact, the state $z:=v^{2}$ is defined to render the dynamic model linear.

$$
\begin{equation*}
\frac{1}{2} m z(i+1)-\frac{1}{2} m z(i)+m g \Delta s \sin \theta(i)+\mu_{r} m g \Delta s \cos \theta(i)+\frac{1}{2} \rho_{a} C_{d} A_{f} \Delta s z(i)=E_{e}(i)-E_{b}(i) \tag{5.5}
\end{equation*}
$$

While Eqn. (5.5) handles the system's energy dynamics exactly, it results in elapsed time as a nonlinear function of $z$. The next section will address the approximation of time.

### 5.3.2 Piecewise-Linear Approximation of Time

As a result of aerodynamic drag loss increasing quadratically with speed, wheel-input energy minimization favors slower average speeds unless arrival time is either penalized in the objective or constrained. This formulation will constrain the final time after $N$ steps according to

$$
\begin{equation*}
t(N+1) \leq t_{f}, \quad t(N+1)=\sum_{i=0}^{N} \Delta t(i) \tag{5.6}
\end{equation*}
$$

The change in time $\Delta t$ from one position step to the next after dropping the argument $i$ is

$$
\begin{equation*}
\Delta t=\frac{\Delta s}{v}=\frac{\Delta s}{z^{\frac{1}{2}}} \tag{5.7}
\end{equation*}
$$

Given a nominal trajectory $z_{0}(i)$, it is possible to linearize Eqn. (5.7) using the first-order Taylor expansion. However, this approach yields the approximation demonstrated in Fig. 5.2 where elapsed time is underestimated at the extreme upper and lower bounds on $z$. Since final time is upper-bound constrained, this model deviation has the especially harmful effect of giving a fictitious time advantage to extreme speeds. The model then predicts that if extreme speeds are used, slower speeds can be selected overall to reduce drag. In reality, switching between extreme speeds increases aerodynamic drag loss compared to driving at constant speed, and the spurious time benefit does not manifest in reality. Another negative effect of the first-order expansion approach is that time is underestimated, resulting in consistent delays.

To improve the result, a piecewise-linear approximation inspired by [100] is used instead. Elapsed time is approximated as

$$
\begin{equation*}
\Delta t=t(i+1)-t(i) \approx \Delta t_{0}+p_{1} \tilde{z}_{u}-p_{2} \tilde{z}_{l}, \quad z=z_{0}+\tilde{z}_{u}-\tilde{z}_{l}, \quad \tilde{z}_{u}, \quad \tilde{z}_{l} \geq 0 \tag{5.8}
\end{equation*}
$$



Figure 5.2: Block diagram of the multi-predecessor car following controller.
where $\tilde{z}_{u}$ and $\tilde{z}_{l}$ represent the deviations from the nominal square of speed $z_{0}$ and $\Delta t_{0}$ is the elapsed time at $z_{0}$. The solver decides $\tilde{z}_{u}$ and $\tilde{z}_{l}$ subject to the constraints in Eqn. (5.8). The coefficients $p_{1}$ and $p_{2}$ are computed as

$$
\begin{equation*}
p_{1}=\frac{\Delta \bar{t}-\Delta t_{0}}{\bar{z}-z_{0}}, \quad p_{2}=\frac{\Delta t_{0}-\Delta \underline{t}}{z_{0}-\underline{z}} \tag{5.9}
\end{equation*}
$$

where $\bar{z}, \underline{z}, \Delta \bar{t}$, and $\Delta \underline{t}$ denote the maximum and minimum $z$ values and their corresponding elapsed times, respectively. This produces a secant-line approximation that intersects the true elapsed time at the nominal and extreme speeds. These extreme speeds are computed according to

$$
\begin{equation*}
\underline{z}=\max \left\{\delta_{\ell},\left(1-\delta_{z}\right) z_{0}\right\}, \quad \bar{z}=\min \left\{\left(1+\delta_{z}\right) z_{0}, \bar{z}_{e}\right\} \tag{5.10}
\end{equation*}
$$

where $\bar{z}_{e}$ is the environmental maximum speed described later in Section 5.4. The constants are set to $\delta_{\ell}=$ $0.1 \mathrm{~m} / \mathrm{s}$ and $\delta_{z}=0.2$ for the remainder of the chapter, although Fig. 5.2 uses $\delta_{z}=0.4$ for illustrative purposes.

While the introduction of $\tilde{z}_{u}$ and $\tilde{z}_{l}$ adds a degree of freedom, the optimizer finds an appropriate solution for the following reason. The desired piecewise function shown in Fig. 5.2 assumes that at least one of $\tilde{z}_{u}$ and $\tilde{z}_{l}$ is equal to 0 . For $z<z_{0}$, the time estimate increases as $\tilde{z}_{l}$ increases according to the slope $p_{2}<0$. For $z>z_{0}$, the time estimate increases as $\tilde{z}_{u}$ increases according to the slope $p_{1}<0$. Notice that the true function $\Delta t(z)$ is strictly convex for $z>0$, so $p_{1}>p_{2}$. A malfunction might occur if the solver could find $\tilde{z}_{u}$ and $\tilde{z}_{l}$ such that the estimated $\Delta t$, denoted $\Delta \hat{t}$, was less than the one shown in Fig. 5.2. Then, the optimization would predict satisfaction of the final constraint when it would be violated according to the intended approximation. The opposite malfunction could occur if some constraint forced the solver to select $\tilde{z}_{u}$ and $\tilde{z}_{l}$ so that the estimated time was greater than the intended function, resulting in unneeded selection of a faster speed trajectory. Therefore, we need to show that $\Delta \hat{t}$ is minimized by letting at least one of $\tilde{z}_{u}$ and $\tilde{z}_{l}$
equal 0 , given a candidate $z$ trajectory ${ }^{1}$. For each stage, this minimization is the LP

$$
\begin{align*}
\min _{z_{u}}, \tilde{z}_{l} & J_{t}=\Delta t_{0}+p_{1} \tilde{z}_{u}-p_{2} \tilde{z}_{l} \\
\text { s.t. } & \tilde{z}_{u}-\tilde{z}_{l}=z-z_{0}  \tag{5.11}\\
& \tilde{z}_{u} \geq 0 \\
& \tilde{z}_{l} \geq 0
\end{align*}
$$

which, after dropping the fixed $\Delta t_{0}$ from the objective, exploiting the equality constraint to eliminate $\tilde{z}_{u}$, and subsequently dropping the fixed $z$ and $z_{0}$ from the new objective, becomes

$$
\begin{align*}
\min _{z_{u}}, \tilde{z}_{l} & J_{t}^{\prime}=\left(p_{1}-p_{2}\right) \tilde{z}_{l} \\
\text { s.t. } & \tilde{z}_{l} \geq z_{0}-z  \tag{5.12}\\
& \tilde{z}_{l} \geq 0
\end{align*}
$$

Consider the case where $z \geq z_{0}$. Then the constraints of (5.12) reduce to $\tilde{z}_{l} \geq 0$. Since $p_{1}>p_{2}$ and hence the coefficient of $\tilde{z}_{l}$ is positive, $J_{t}^{\prime}$ is minimized when $\tilde{z}_{l}=0$. Now, consider the case where $z<z_{0}$. Then, the first constraint in (5.12) becomes active and the minimizer is $\tilde{z}_{l}=z_{0}-z$. Recovering $\tilde{z}_{u}$ from $\tilde{z}_{l}$ using the equality constraint of (5.11) reveals that $\tilde{z}_{u}=0$. Therefore, the assumption that at least one of $\tilde{z}_{u}$ and $\tilde{z}_{l}$ is equal to 0 holds. We emphasize that the convexity of Eqn. (5.7) over the relevant interval was critical to this reasoning.

The piecewise-linear approximation is compared to the first-order expansion in Fig. 5.2. In addition to improving overall accuracy, the piecewise technique imposes zero deviation from the true elapsed time at the nominal and extreme speeds. This resolves the motivating problem where the tangential approximation rewarded extreme speeds with incorrectly low elapsed time, according to the model. All other deviations of the piecewise approximation from the actual time are positive, which causes the model to slightly overestimate total time over all steps. When the modeled final time is equal to the boundary condition $t_{f}$, this biased model mismatch causes solutions to arrive slightly early rather than late. More formally,

$$
\begin{equation*}
\Delta \hat{t}(i) \geq \Delta t(i) \forall i, \quad \sum_{i=0}^{N} \Delta \hat{t}(i) \leq t_{f} \Longrightarrow \sum_{i=0}^{N} \Delta t(i) \leq t_{f} \tag{5.13}
\end{equation*}
$$

In contrast, the tangential approximation consistently underestimates travel time, biasing the result toward

[^4]lateness. This risk-aversion could offer a relative advantage to some users.

### 5.3.3 Optimal Control Problem

Since powertrain and braking energies are used as the inputs in (5.5), minimization of the total energy transferred from the powertrain to the wheels results in a linear objective function. This energy is called wheel-to-distance energy. The effect of this objective choice is that the tradeoff between the efficient practices of minimizing braking and minimizing speed variation is understood in the optimal control problem (OCP), but changes in engine efficiency are not. A second term, lightly weighted by $q_{z}$, helps to reduce chattering in closed-loop operation by penalizing the upper and lower speed deviations $\tilde{z}_{u}$ and $\tilde{z}_{l}$ from the initial guess. Since the previous solution is used as the initial guess, this second term promotes consistency. As in previous chapters, $N$ is the prediction horizon. Since a shrinking-horizon scheme is used in closed-loop, $N$ decreases as the vehicle moves forward. The final time $t_{f}$ is an input to the planner.

$$
\begin{array}{ll}
\min & J=\sum_{i=1}^{N} E_{e}(i)+\sum_{i=1}^{N+1} q_{z}\left(\tilde{z}_{u}(i)+\tilde{z}_{l}(i)\right) \\
\text { s.t. } & \underline{z}(i) \leq z(i) \leq \bar{z}(i) \\
& 0 \leq E_{e}(i) \leq \bar{E}_{e}(i)  \tag{5.14}\\
& 0 \leq E_{b}(i) \leq \bar{E}_{b} \\
& t(N+1) \leq t_{f}
\end{array}
$$

The minimum and maximum constraints $\underline{z}$ and $\bar{z}$ on the speed-related state $z$ depend on stage not only because of practical needs for position-varying speed limits, but also because the approximation of time is only locally valid.

Although the slack variables are omitted from OCP (5.14) to reduce clutter, the minimum speed and final time constraints are softened using linear penalties to guarantee feasibility. Specifically, final time may be infeasible if maximum speed and/or acceleration capacity is too low to meet the goal in time. The minimum speed constraint can also be infeasible in the presence of a steep uphill grade.

Notice that the maximum input energy $\bar{E}_{e}$ is a function of stage $i$. Typically, $\bar{E}_{e}(i)$ is conservatively approximated as constant using the maximum acceleration at maximum speed. However, a tighter upper bound can be obtained for the solution's first several stages by simulating speed under a true maximal acceleration and recovering the energy input during this process as a function of position. This takes advantage of the
fact that the vehicle may begin at a sub-maximal speed and cannot reach the maximum speed instantaneously. Therefore, there is no need to restrict it to the maximum speed's acceleration constraint immediately.

### 5.4 Speed Constraints

Two types of speed constraints are considered: regulatory and traffic. The regulatory speed limit is known precisely as a function of position. The same algorithm could also handle known speed limits on upcoming curves, although no curves were simulated in this study. Traffic-based constraints were also explored as a way to account for expected future congestion in the speed planner. This system was not used in the full-length simulations because early evaluation did not show a fuel economy benefit. It is, however, documented here to support the results and possible future development.

The regulatory speed limit $\bar{v}_{r}(i)$ is assumed to be known. In the absence of traffic, this directly establishes the environmental speed limit $\bar{z}_{e}(i)=\bar{v}_{r}^{2}(i)$. The nominal trajectory $z_{0}$ is also saturated to $\bar{z}_{e}(i)$, then rate limited based on the maximum and minimum accelerations to prevent impossibly quick transitions between speed limits from appearing in the nominal trajectory.

Traffic constraints are more complex. First, the traffic speed $v_{t}^{\prime}(s, t)$ depends on both position and time. Furthermore, slow-moving but sparse traffic may not present an obstacle to longitudinal motion at all if passing is consistently possible. In this design, the nominal speed trajectory is used to establish a position-time relationship for the traffic speed constraint. This approach enables anticipation of congestion that predictably occurs at a certain position, such as what might occur near a busy interchange or lane bottleneck. However, it does not attempt to optimize the position-time relationship to, for example, wait until a temporary jam has dissipated before arriving at its location. Such optimization is left for future research. The algorithm's equations are

$$
\begin{equation*}
t_{0}(i)=t(0)+\sum_{j=0}^{i-1} \frac{\Delta s}{v_{0}(j)}, \quad v_{t}(i)=v_{t}^{\prime}\left(s(i), t_{0}(i)\right) \tag{5.15}
\end{equation*}
$$

The expected maximum speed $\bar{v}$ of the lane-changing ego vehicle as a function of the regulatory speed $\bar{v}_{r}$ and the traffic speed $v_{t}$ is now derived using probability. A moving reference frame of constant speed equal to the traffic speed is used. As a first approximation, the following assumptions are adopted.

1. All surrounding vehicles travel at constant speed.
2. All surrounding vehicles travel at the same speed, equal to the expected traffic speed.
3. Density in both lanes is equal.
4. Vehicles are evenly spaced within a lane.
5. Phasing of vehicular strings is uniformly distributed between lanes.

Let $l_{v}, l_{s}, \Delta t_{c}$, and $v_{w}$, denote the ego vehicle length, surrounding vehicle length, time needed to change lanes, and the maximum weaving speed of the ego vehicle, respectively. Then the gap required to change lanes is

$$
\begin{equation*}
\Delta s_{w}=v_{w} \Delta t_{c}+l_{v} \Longrightarrow v_{w}=\frac{1}{\Delta t_{c}} \Delta s_{w}-\frac{1}{\Delta t_{c}} l_{v} \tag{5.16}
\end{equation*}
$$

This same gap can be expressed in terms of the minimum safe gap $\underline{d}$ and the preceding and following vehicles’ positions $s_{B}$ and $s_{A}$.

$$
\begin{equation*}
\Delta s_{w}=s_{A}-s_{B}-2 \underline{d}-l_{s} \tag{5.17}
\end{equation*}
$$

The phase difference $\phi$ between the two lanes is defined as

$$
\begin{equation*}
\phi:=\frac{s_{A}-s_{B}}{h} \tag{5.18}
\end{equation*}
$$

The maximum relative weaving speed can now be found be combining Eqns. (5.16), (5.17), and (5.18). The relative weaving speed cannot exceed the regulatory speed, nor can it fall below the speed of traffic since remaining in a single lane is always an option. Hence $v_{w}(\phi)$ is piecewise. Let $\tilde{v}_{r}=\bar{v}_{r}-v_{t}$ denote the regulatory speed limit relative to the traffic speed.

$$
v_{w}(\phi)=\left\{\begin{array}{l}
0 ; \quad \frac{h}{\Delta t_{c}} \phi-\frac{1}{\Delta t_{c}}\left(2 \underline{d}+l_{s}+l_{v}\right)<0  \tag{5.19}\\
\frac{h}{\Delta t_{c}} \phi-\frac{1}{\Delta t_{c}}\left(2 \underline{d}+l_{s}+l_{v}\right) ; \quad \text { otherwise } \\
\tilde{v}_{r} ; \quad \frac{h}{\Delta t_{c}} \phi-\frac{1}{\Delta t_{c}}\left(2 \underline{d}+l_{s}+l_{v}\right)>\tilde{v}_{r}
\end{array}\right.
$$

The expected weaving speed relative to traffic is now computed, assuming the phase $\phi$ is uniformly distributed. Integrating the product of the probability density function and the random variable yields the expectation. Taking the smallest gap between the staggered traffic strings results in symmetry that is exploited to simplify the integration.

$$
\begin{equation*}
\bar{v}=v_{t}+E v_{w}=v_{t}+\int_{0}^{1} v_{w}(\phi) d \phi=v_{t}+2 \int_{0}^{\frac{1}{2}} v_{w}(\phi) d \phi \tag{5.20}
\end{equation*}
$$

When traffic preview is used, $\bar{z}_{e}(s)=\bar{v}^{2}(s)$ from Eqn. (5.20).
Figure 5.3 shows the results of this integration in a sample case where $\bar{v}_{r}=31 \mathrm{~m} / \mathrm{s}$ and $v_{t}=25 \mathrm{~m} / \mathrm{s}$. As expected, maximum speed is equal to the traffic speed at high density and increases to approach the regulatory speed as density decreases.


Figure 5.3: The expected maximum weaving speed, shown as functions of headway and density.

### 5.5 Initialization and Iteration

This section addresses the generation of the nominal trajectory $v_{0}(s)$ that is needed to approximate time in the LP. An initial trajectory is generated by explicitly solving for a trajectory combining constant acceleration, constant speed, coasting, and braking. After an LP solution is obtained, it is used as an interpolant to generate the subsequent nominal trajectory in the process demonstrated in Fig. 5.4. In this way, the LP solution is steadily improved as real time advances and only one LP is needed per step.

On the first step when an old LP solution is unavailable, the following procedure is used to generate an initial guess. First, the transient from the initial speed $v(0)$ to the maximum speed $\bar{v}$ at the true maximum acceleration $\bar{a}(v)$ is simulated. This results in the position $\bar{s}_{a}$ where the maximum speed is reached, as well as a function $\bar{E}_{a}(v)$ that is used in the LP itself to improve constraint accuracy in the shortest-term steps. A heuristic constant acceleration for the initialization process is calculated as

$$
\begin{equation*}
\tilde{a}=\alpha_{a}\left(\bar{a}\left(v_{0}\right)\right)+\left(1-\alpha_{a}\right)\left(\frac{\bar{v}^{2}-v_{0}^{2}}{2 \bar{s}_{a}}+a_{r}+c_{r} \bar{v}^{2}\right) \tag{5.21}
\end{equation*}
$$

where $a_{r}$ and $c_{r}$ are the coastdown acceleration coefficients and $\alpha_{a}=0.7$. The vehicle's maximum braking acceleration is taken as input $\underset{\sim}{a}$. Recall that the final time $t_{f}$ and position $s_{f}$ are known. These are used to compute an approximate coastdown acceleration $r_{0}=a_{r}+c_{r}\left(\frac{s_{f}}{t_{f}}\right)^{2}$.

-     -         - Linearization ......... Minimum .......... Maximum __ LP Soln.
(a) Step 1

(b) Step 2

(c) Step 3

(d) Step 4

(e) Step 5

(f) Road Grade


Figure 5.4: The first 5 LP solutions in a closed-loop simulation.

With these approximate capacities, several possible forms are considered for the initial trajectory. The goal is to select a relatively low-energy trajectory from the ones that are feasible. Under certain boundary conditions, zero input energy may be possible. In the first such case considered, the CAV coasts until it reaches speed $v_{1}$ and then brakes. The degrees of freedom are $v_{1}$ and the braking acceleration $a_{b}$, and the equations are the total time and position boundary conditions

$$
\begin{equation*}
t_{f}=\frac{v_{1}-v_{0}}{-r_{0}}+\frac{v_{f}-v_{1}}{a_{b}} \tag{5.22}
\end{equation*}
$$

and

$$
\begin{equation*}
s_{f}=\frac{v_{1}^{2}-v_{0}^{2}}{-2 r_{0}}+\frac{v_{f}^{2}-v_{1}^{2}}{2 a_{b}} \tag{5.23}
\end{equation*}
$$

If a feasible solution to the above system is found, that trajectory is taken as the initial guess. If not, a lowerspeed zero energy case is attempted where braking occurs before coasting, resulting in the equations

$$
\begin{equation*}
t_{f}=\frac{v_{1}-v_{0}}{a_{b}}+\frac{v_{f}-v_{1}}{-r_{0}} \tag{5.24}
\end{equation*}
$$

and

$$
\begin{equation*}
s_{f}=\frac{v_{1}^{2}-v_{0}^{2}}{2 a_{b}}+\frac{v_{f}^{2}-v_{1}^{2}}{-2 r_{0}} \tag{5.25}
\end{equation*}
$$

where the degrees of freedom are again $v_{1}$ and $a_{b}$.
If neither of the previously described solutions are feasible, some energy must be expended. A family of trajectories can be derived from the equations

$$
\begin{equation*}
t_{f}=\frac{v_{c}-v_{0}}{a_{1}}+\Delta t_{c}+\frac{v_{f}-v_{c}}{a_{2}} \tag{5.26}
\end{equation*}
$$

and

$$
\begin{equation*}
s_{f}=\frac{v_{c}^{2}-v_{0}^{2}}{2 a_{1}}+v_{c} \Delta t_{c}+\frac{v_{f}^{2}-v_{c}^{2}}{2 a_{2}} \tag{5.27}
\end{equation*}
$$

where the constant speed $v_{c}$ and the time $t_{c}$ spent at constant speed are the degrees of freedom. Different trajectories are tried by setting $\left(a_{1}, a_{2}\right)$ to one of the possible orderings

$$
\begin{equation*}
\mathcal{A}=\left\{\left(\tilde{a}-r_{0}, \underline{a}\right),\left(\tilde{a}-r_{0},-r_{0}\right),\left(-r_{0}, \tilde{a}-r_{0}\right),\left(\underline{a}, \tilde{a}-r_{0}\right),\left(-r_{0},-r_{0}\right),\left(-r_{0}, \underline{a}\right),\left(\underline{a},-r_{0}\right),(\underline{a}, \underline{a})\right\} \tag{5.28}
\end{equation*}
$$

which represent various permutations of acceleration, coasting, and braking.
One additional possibility is considered where the maximum speed is reached and acceleration, coasting, and braking intervals are all present. In this case, the equations are

$$
\begin{equation*}
t_{f}=\frac{v_{c}-v_{c}}{\tilde{a}-r_{0}}+\Delta t_{c}+\frac{v_{3}-v_{c}}{-r_{0}}+\frac{v_{f}-v_{3}}{\underline{a}} \tag{5.29}
\end{equation*}
$$

and

$$
\begin{equation*}
s_{f}=\frac{v_{c}^{2}-v_{0}^{2}}{2\left(\tilde{a}-r_{0}\right)}+v_{c} \Delta t_{c}+\frac{v_{3}^{2}-v_{c}^{2}}{-2 r_{0}}+\frac{v_{f}^{2}-v_{3}^{2}}{2 \underline{a}} \tag{5.30}
\end{equation*}
$$

and the degrees of freedom are the time at maximum speed $\Delta t_{c}$ and the transition speed from coasting to braking $v_{3}$. The constant speed $v_{c}$ is not a degree of freedom since it is fixed at $\bar{v}$.

The energy input in candidate solutions is evaluated as

$$
\begin{equation*}
W_{e}=\int_{0}^{s_{f}} m u_{t} d s \tag{5.31}
\end{equation*}
$$

which has a nonzero integrand under acceleration and constant speed. In the latter case, $u_{t}=a_{r}+c_{r} v_{c}^{2}$. If multiple trajectories are feasible, the one that minimizes $W_{e}$ is selected as the initial guess. Finally, the initial speed trajectory is saturated to any position-dependent speed limits and rate limited to $\tilde{a}$ and $\underline{a}$ in acceleration and braking, respectively.

It is emphasized that despite the relatively lengthy description, this kinematic system is only used on the first step to initially linearize before solving the LP. Although the interval types chosen were selected from the possible optimal ones in combustion-engine eco-driving, there is no theoretical guarantee that the solution is energy-optimal. Besides, road grade is ignored and the boundary conditions may not be met if the trajectory needs to be saturated. It is therefore important that the LP be used for subsequent solutions and linearization trajectories in closed-loop operation.

### 5.6 Simulation Methods

The proposed algorithms introduce several features that derive their information and benefits from different aspects of the problem. The pacing module takes advantage of road grade and speed limit information over a whole trip, while the maneuvering module is concerned with microscopic obstacle avoidance. The simulation testbed should include the effects of both scales. Simulation of a realistically long highway route
is desired to show the trip planner's benefits, and the project's industry partner Cummins provided road grade information over a suitable 69.7 km route along Interstate 64 from Lanesville, IN to Siberia, IN. Simulations of this length present challenges, including long run times that can slow early prototype assessment and the need to compute and store data from many surrounding vehicles.

This section describes the various ways in which these challenges are overcome. It begins with the plant model used to simulate the vehicles. Then, the different scenarios used to select prototypes and assess overall performance are introduced. Techniques for simulating microscopic obstacles within computational resource constraints while targeting given macroscopic conditions are described. The section closes with a review of the lateral and longitudinal driver model used for the baseline and surrounding vehicles.

### 5.6.1 Plant Model

The simulation testbed is the nonlinear form of the kinematic bicycle model and pure pursuit steering controller with the ramped lane changing path described in Section 4.1.3. The lane change time $t_{l}$ was 5 s for the tractor-trailer and 2 s for the passenger vehicles. The tractor and surrounding vehicles were 5 m long. A 15 m trailer was added to the ego CAV using the nonlinear kinematic model in Section 4.1.3.

The longitudinal parameters from the heavy vehicle in Appendix A were used in earlier prototyping. They were modified slightly to move friction and accessory loss into the vehicle-level resistance model before the full-length Lanesville-Siberia simulations. Cummins also provided parameters obtained by either physical measurement or, in the case of the resistance coefficients, fuel economy model correlation. These parameters were also used in the Lanesville-Siberia simulations, enabling a comparison of two different trucks. Table 5.2 summarizes these parameter sets.

Table 5.2: Truck parameter sets.

| Parameter | Value, Set A | Value, Set B |
| :--- | :---: | :---: |
| Mass | 19400 kg | 30391 kg |
| Drag coefficient | 0.5489 | 0.5 |
| Frontal area | $10.8 \mathrm{~m}^{2}$ | $10.22 \mathrm{~m}^{2}$ |
| Rolling resistance coefficient | 0.0162 | 0.0051 |
| Maximum Torque | 1425 Nm | 2208 Nm |
| Air density | $1.184 \mathrm{~kg} / \mathrm{m}^{3}$ | $1.184 \mathrm{~kg} / \mathrm{m}^{3}$ |

### 5.6.2 Scenarios

The main purpose of the simulations was to assess the functionality of the optimal control algorithm's new features. Therefore, variations in road grade and regulatory speed limit were present and the optimal controller was provided with preview of both. Traffic was also introduced and a preview of its macroscopic speed and density characteristics was provided to MPC when traffic preview features were enabled. Long simulation run times motivated a preliminary study using a 6 km segment of the Lanesville-Siberia route to check functionality and decide which features were beneficial. Figure 5.5(a) depicts this shortened scenario, contrasted with the full-length one in Fig. 5.5(b). The longer scenario was subdivided into 13 sequentially-run time segments to help manage data storage.


Figure 5.5: The shortened 6 km scenario (a) and the full Lanesville-Siberia route (b).
Both scenarios used a default speed limit of $31 \mathrm{~m} / \mathrm{s}$, intended to match common interstate highway speeds. A reduced speed limit of $25 \mathrm{~m} / \mathrm{s}$, which might realistically occur in a construction zone, was imposed over a segment of each scenario. The boundary condition on the final time $t_{f}$ targeted an average speed of $25 \mathrm{~m} / \mathrm{s}$, anticipating that traffic-induced slowdowns and grade fluctuation would drive speeds in excess of $25 \mathrm{~m} / \mathrm{s}$ elsewhere, speed limits permitting. Initial and final speed boundary conditions were both set at $26 \mathrm{~m} / \mathrm{s}$.

To simulate a traffic jam that formed behind a obstruction that was eventually cleared, traffic was released from rest at a fixed position and time. The positions from which traffic was released are marked in green in Fig. 5.5, and the release times were 65 s in the 6 km scenario and 660 s in the full-length scenario. Targets for macroscopic speed, speed variance, and density of the surrounding traffic were precomputed by

Tyler Ard using methods from [54] and [55]. The input traffic data was calculated for a ring road, which traps vehicles on a fixed length of road. In reality, vehicles with different desired speeds will eventually dissipate given enough space. Therefore, density was ramped down starting at the positions $s_{\rho}$ marked in red in Fig. 5.5 to simulate such dissipation. The target density after the ramp is

$$
\begin{equation*}
\rho_{t}(s, t)=\max \left\{\bar{\rho}_{t}(s, t)+m_{\rho}\left(s-s_{\rho}\right), \underline{\rho}_{t}\right\} \tag{5.32}
\end{equation*}
$$

where $\bar{\rho}_{t}$, and $\rho_{t}$ are the raw ring-road traffic density and final target density, respectively. In the 6 km scenario, the density rate $m_{\rho}$ was $-0.004 \mathrm{veh} / \mathrm{km}$ per m and the minimum density target $\underline{\rho}_{t}$ was $2 \mathrm{veh} / \mathrm{km}$. In the fulllength scenario, the density rate was $-0.002 \mathrm{veh} / \mathrm{km}$ per m and the minimum density target was $5 \mathrm{veh} / \mathrm{km}$. The density reduction's realism is bolstered by the method used to target it, since vehicles are only removed from the simulation if their relative speed naturally increases their distance from the ego. The following section elaborates on this technique for microscopically realizing the macroscopic traffic targets.

### 5.6.3 Targeting Macroscopic Conditions in a Sliding Window

Running a 70 km highway simulation including all vehicle simultaneously would be computationally prohibitive and generate an excessive amount of log data. Moreover, long range radar (LRR) can only sense up to about a 150 m range [113] and so inclusion of more distant unconnected vehicles in obstacle avoidance is not generally realistic. Therefore, the specified macroscopic traffic conditions are approximately imposed on a smaller frame of width $w_{f}=0.34 \mathrm{~km}$. The frame translates so that it is always centered around the ego vehicle. An example simulation frame with the ego truck and surrounding vehicles is shown in Fig. 5.6.


Figure 5.6: The simulation window in a sparse traffic scene during a CAV lane change.

Surrounding vehicles are generated near the edges of the window. The criterion for generating a new surrounding vehicle is

$$
\begin{equation*}
n_{v}<\rho_{t} n_{\ell} w_{f} \tag{5.33}
\end{equation*}
$$

where $n_{v}$ is the number of vehicles in the frame, $\rho_{t}$ is the target density per lane, and $n_{\ell}$ is the number of lanes. Vehicles are removed from the simulation when they reach the edge of the reference frame. Once vehicle generation is triggered, the new vehicle's lane and speed must still be determined. First, a vehicle speed is randomly drawn from the predicted traffic distribution. If the speed is greater than the ego vehicle's, the new vehicle is inserted behind the ego. Otherwise, it is inserted ahead of the ego. Vehicles initialize in the lowerdensity lane unless that lane's initialization zone is occupied. These zones $S_{\mathrm{us}}$ and $S_{\mathrm{ds}}$ are defined relative to the edges of frame according to

$$
\begin{equation*}
S_{\mathrm{us}}=\left\{x \left\lvert\, x<s-\frac{1}{2} w_{f}+\underline{d}_{g}+T_{g} v\right.\right\}, \quad S_{\mathrm{ds}}=\left\{x \left\lvert\, x>s+\frac{1}{2} w_{f}-\underline{d}_{g}-T_{g} v\right.\right\} \tag{5.34}
\end{equation*}
$$

where $T_{g}$ and $\underline{d}_{g}$ are time headway and fixed distance constants.

### 5.6.4 Baseline and Surrounding Vehicle Algorithm

The lane-changing model MOBIL after [121] coupled with the Intelligent Driver Model (IDM) [120] is used as a baseline. MOBIL works by using a longitudinal driver model to compare the potential accelerations of the ego vehicle and its neighbors in the current and candidate lanes. The IDM equations are listed in Section 1.1.3.1 and the MOBIL lane change criteria are provided in Section 1.1.3.1. The parameters used for the surrounding vehicles are provided in Table 5.3. When each vehicle is generated, it is assigned a reference speed $v_{\text {ref }}=v_{t}+c_{t}$, where the random part of speed $c_{t}$ is drawn from $C_{t} \sim \mathcal{N}\left(0, \sigma_{t}^{2}\right)$, the normal distribution with the predicted macroscopic traffic variance. Surrounding vehicles used a reduced time headway parameter $\tau_{\mathrm{h}}$ relative to [30] in order to reproduce the target traffic conditions. As in [30], a log-normal distribution with parameters $\mu_{\tau}$ and $\sigma_{\tau}$ was used to randomly vary $\tau_{\mathrm{h}}$. The mean of a log-normally distributed variable $X$ as provided in [45] is

$$
\begin{equation*}
E X=e^{\mu+\frac{1}{2} \sigma^{2}} \tag{5.35}
\end{equation*}
$$

and the variance is

$$
\begin{equation*}
\operatorname{Var} X=e^{2 \mu}\left(e^{2 \sigma^{2}}-e^{\sigma^{2}}\right) \tag{5.36}
\end{equation*}
$$

resulting in a mean of 0.5125 s and a variance of $0.07610 \mathrm{~s}^{2}$.

Table 5.3: IDM parameters in the tractor-trailer simulations.

| Parameter | Description | Value |
| :--- | :---: | :---: |
| $a_{0}$ | Maximum acceleration | $1.0 \mathrm{~m} / \mathrm{s}^{2}$ |
| $b_{0}$ | Deceleration coefficient | $1.5 \mathrm{~m} / \mathrm{s}^{2}$ |
| $d_{\text {st }}$ | Stopped distance | 4 m |
| $\mu_{\tau}$ | Time headway $\mu$ | -0.7956 |
| $\sigma_{\tau}$ | Time headway $\sigma$ | 0.5044 |
| $\delta$ | Exponent | 4 |

One drawback of the sliding window scheme is that it does not simulate some downstream vehicles' predecessors, causing them to accelerate in a way that would not be possible in true traffic of the target density. To mitigate this problem while still simulating drivers' tendency to acceleration in the absence of obstacles, the IDM desired speed $v_{0}$ is computed as a blend of the vehicle's reference speed $v_{\text {ref }}$ and the maximum speed $\bar{v}$.

$$
\begin{equation*}
v_{0}=\alpha\left(\bar{v}+c_{t}\right)+(1-\alpha) v_{\mathrm{ref}} \tag{5.37}
\end{equation*}
$$

Although $v_{\text {ref }}$ is updated for each surrounding vehicle on each step, only $v_{t}$ is modified. The random part $c_{t}$ is usually maintained and only updated rarely to break unrealistic equilibria. The expected time until such an update is 5 min .

Some parameters differ where the IDM is used as part of the baseline algorithm to control the ego truck. In this case, a longer time headway of $\tau_{\mathrm{h}}=1.4 \mathrm{~s}$ is used to promote smoother driving. The target speed $v_{0}$ is tuned to match MPC's average speed in each scenario and is saturated to the local speed limit as needed.

### 5.6.5 Sampling and Statistics

For both scenarios, three runs were performed using each candidate algorithm. In the 6 km scenario, the first two candidates used optimal control with and without traffic preview and one used MOBIL. Because of the results of the 6 km study, the full Lanesville-Siberia study only compared optimal control without traffic preview and MOBIL. Multiple runs are needed because the surrounding vehicles' reference speeds and time headway parameters are pseudorandomly generated as described in the previous section, and these parameters affect the speed adjustments that the ego must perform for obstacle avoidance. On the other hand, the full Lanesville-Siberia route is costly to simulate in terms of computation time and storage. Hence, it was necessary to limit the number of samples while tracking the statistical confidence in any energy benefits.

Because of the small sample size, Student's $t$-distribution was used with the sample standard deviation to determine the confidence intervals.

### 5.7 Results

Results of the 6 km scenario are provided first, followed by those obtained over the full LanesvilleSiberia route. Discussion of these results will follow in Section 5.8. Separate from the closed-loop results, an open-loop computation time test matrix was also run for the LP formulation and solution. These results are provided in Section 5.7.3.

### 5.7.1 Initial Study on a Shortened Trip

Figure 5.7 provides the resulting trajectories from the 6 km scenario and Fig. 5.8 shows the performance metrics for each algorithm. Traffic preview did not significantly improve fuel economy, although it did improve consistency of fuel economy. Both optimal control variants improved fuel economy compared to MOBIL by $15 \%$ on average, based on a small sample of 3 runs each. Results for wheel-input energy, which was optimized in the LP, were similar to the fuel results. Traffic preview reduced MPC's mean energy consumption slightly in this sample, but statistical confidence in the difference is low at this sample size. MOBIL's $v_{0}$ parameter was tuned to deliver similar average speed to MPC, and the difference in average speed between the algorithms was not significant.

### 5.7.2 The Lanesville-Siberia Route

The full-length Lanesville-Siberia results begin by verifying the effectiveness of the traffic simulator. Figure 5.9 provides an example of how the target speed and density supplied to the traffic simulator compared with the actual speed and density achieved in the simulation window. It also shows which portions of the trip had the greatest impact from traffic. Figure 5.9 was taken from the case where MPC was used with parameter set B.

Two different truck parameter sets were simulated. Set B, which was provided by Cummins, had high mass and lower rolling resistance compared to set A. Figure 5.10 shows the trajectories obtained from model predictive control (MPC) coupled with the LP speed planner and MOBIL for parameter set A and Fig. 5.11 shows the same for set B. Figure 5.12 shows the overall performance in energy, fuel economy, and average speed for parameter set A and Fig. 5.13 shows the same for parameter set B. Error bars represent the


Figure 5.7: Trajectories for MPC and the baseline MOBIL in the 6 km scenario.


Figure 5.8: Fuel consumption and speed in the 6 km scenario with $90 \%$ confidence intervals.


Figure 5.9: Example comparison of the microscopic traffic conditions to the macroscopic targets.
$95 \%$ confidence intervals based on the sample standard deviation and $t$-distribution. MPC did not use traffic preview in this case.

### 5.7.3 Computation Time

Tables 5.4 and 5.5 provide computation times for various prediction horizons and discretization steps. Assembly time is the amount of time needed to build the matrices that define the LP and solution time is the amount of time needed to solve the LP after assembly. Table 5.6 compares computation times for three LP solution methods. All LPs were assembled using custom MATLAB code and solved using Gurobi [44]. The primal simplex method was used to generate Table 5.5. Blank table cells indicate that MATLAB memory limits were reached due to large problem sizes.

Although the exact computation time of the industry partner's dynamic programming tool for this purpose is kept confidential, these results show that the LP can be solved orders of magnitude faster. This substantial difference enables repeated solution of the LP in a closed-loop setting, a use case that is not practical with DP.

Table 5.4: Speed planner assembly time under various settings.

| Position Step | Time, $1 \mathrm{~km}[\mathrm{~s}]$ | Time, $10 \mathrm{~km}[\mathrm{~s}]$ | Time, Lanesville-Siberia $[\mathrm{s}]$ |
| :--- | :---: | :---: | :---: |
| 10 m | 0.044 | 0.582 | - |
| 20 m | 0.038 | 0.195 | - |
| 50 m | 0.037 | 0.062 | 1.094 |

Table 5.5: Speed planner solution time using various settings.

| Position Step | Time, $1 \mathrm{~km}[\mathrm{~s}]$ | Time, $10 \mathrm{~km}[\mathrm{~s}]$ | Time, Lanesville-Siberia [s] |
| :--- | :---: | :---: | :---: |
| 10 m | 0.019 | 0.313 | - |
| 20 m | 0.014 | 0.103 | - |
| 50 m | 0.012 | 0.032 | 0.522 |

Table 5.6: Speed planner solution time for the Lanesville-Siberia route with 50 m steps.

| Algorithm | Solution Time [s] |
| :--- | :---: |
| Primal Simplex | 0.522 |
| Dual Simplex | 0.530 |
| Barrier | 0.684 |




Figure 5.10: Closed-loop trajectories over the Lanesville-Siberia scenario using parameter set A.


Figure 5.11: Closed-loop trajectories over the Lanesville-Siberia scenario using parameter set B.


Figure 5.12: Energy, fuel consumption, and speed, Lanesville-Siberia, parameter set A, $95 \%$ confidence.


Figure 5.13: Energy, fuel consumption, and speed, Lanesville-Siberia, parameter set B, $95 \%$ confidence.

### 5.8 Discussion

This section reviews the impact of and seeks to explain the results shown in Section 5.7.

### 5.8.1 Conclusions from the Shortened Route

Based on the results in Section 5.7.1, traffic preview was switched off during the full-length simulations. While some consistency was lost, the optimal controller was still able to meet the project's targets and improve consistency compared to MOBIL.

The observed fuel economy improvement over MOBIL in the 6 km route was much greater than the
one in the Lanesville-Siberia route for a similar parameter set. This could be explained by the greater relative impact of traffic in the 6 km route, which necessitated a higher target speed in MOBIL order to match average speed with the optimal controller. This higher target speed required energy input to accelerate early, and that energy was dissipated in braking when the CAV encountered traffic.

### 5.8.2 Explanation for Parameter Sensitivity

Wheel-to-distance energy savings were highly sensitive to the truck's physical parameters, as shown by the $<1 \%$ and $17 \%$ improvements in parameter sets A and B. A simple analysis based on an energy balance explains this difference and could help predict which parameter sets will enjoy significant benefits from energy optimization when driving over a given road slope profile. The energy balance when driving from point 0 to point 1 reads

$$
\begin{equation*}
E_{e}=\Delta K E+\Delta P E+W_{f}+E_{b} . \tag{5.38}
\end{equation*}
$$

Since the initial and final elevations $h_{0}$ and $h_{1}$ are fixed and $\Delta P E=m g\left(h_{1}-h_{0}\right)$, the change in potential energy $\Delta P E$ is fixed. We respect boundary conditions on the speeds $v_{0}$ and $v_{1}$, so $\Delta K E=\frac{1}{2} m\left(v_{1}^{2}-v_{0}^{2}\right)$ is also fixed. The friction work $W_{f}$ is composed of the rolling resistance loss

$$
\begin{equation*}
W_{r}=\int_{0}^{s_{f}} \mu m g \cos \theta(s) d s \tag{5.39}
\end{equation*}
$$

which depends only on the road slope $\theta(s)$ and is therefore fixed, and the aerodynamic loss

$$
\begin{equation*}
W_{a}=\int_{0}^{t_{f}} \frac{1}{2} \rho_{a} C_{d} A v^{3}(t) d t \tag{5.40}
\end{equation*}
$$

which can be optimized. The braking energy $E_{b}$ is a manipulated input, so it can also be optimized. Thus, energy input minimization can be reduced to finding the optimal tradeoff between braking and aerodynamic drag. Aerodynamic drag losses subject to a fixed average speed are minimized by a constant-speed policy, as proven here. After dropping the positive coefficient $\frac{1}{2} \rho_{a} C_{d} A$, the functional to minimize becomes

$$
\begin{equation*}
F_{a}(v(t))=\int_{0}^{t_{f}} v^{3}(t) d t \tag{5.41}
\end{equation*}
$$

which can be expressed as

$$
\begin{equation*}
F_{a}(v(t))=\int_{0}^{t_{f}}(\tilde{v}(t)+\bar{v})^{3} d t \tag{5.42}
\end{equation*}
$$

where $\bar{v}$ is the average speed and $\tilde{v}$ is the deviation from that speed. The constraint on average speed results in

$$
\begin{equation*}
\bar{v}=\frac{1}{t_{f}} \int_{0}^{t_{f}}(\tilde{v}(t)+\bar{v}) d t \Longrightarrow \bar{v}=\frac{1}{t_{f}} \int_{0}^{t_{f}} \tilde{v}(t) d t+\bar{v} \Longrightarrow 0=\int_{0}^{t_{f}} \tilde{v}(t) d t \tag{5.43}
\end{equation*}
$$

Expanding Eqn. (5.42) yields

$$
\begin{equation*}
F_{a}(v(t))=\int_{0}^{t_{f}}\left(\tilde{v}^{3}+3 \bar{v} \tilde{v}^{2}+3 \bar{v}^{2} \tilde{v}+\bar{v}^{3}\right) d t \tag{5.44}
\end{equation*}
$$

which, after exploiting the conclusion of Eqn. (5.43), becomes

$$
\begin{equation*}
F_{a}(v(t))=\int_{0}^{t_{f}}\left(\tilde{v}^{3}+3 \bar{v} \tilde{v}^{2}+\bar{v}^{3}\right) d t \tag{5.45}
\end{equation*}
$$

The $\bar{v}^{3}$ term can be dropped since it is fixed, leaving

$$
\begin{equation*}
F_{a}^{\prime}(v(t))=\int_{0}^{t_{f}}\left(\tilde{v}^{3}+3 \bar{v} \tilde{v}^{2}\right) d t=\int_{0}^{t_{f}} \tilde{v}^{2}(\tilde{v}+3 \bar{v}) d t \tag{5.46}
\end{equation*}
$$

Clearly, $F_{a}^{\prime}(v(t))$ can be made equal to 0 by letting $\tilde{v}=0$ for all $t$. This is the constant-speed policy. To improve on that solution, the integrand of Eqn. (5.46) must be less than 0 for some $t$. Since $\tilde{v}^{2} \geq 0$ for any $\tilde{v}$, the only way to accomplish this is for

$$
\begin{equation*}
\tilde{v}+3 \bar{v}<0 \Longrightarrow \tilde{v}<-3 \bar{v} \tag{5.47}
\end{equation*}
$$

Speed must be non-negative, so

$$
\begin{equation*}
\tilde{v}+\bar{v} \geq 0 \Longrightarrow \tilde{v} \geq-\bar{v} \tag{5.48}
\end{equation*}
$$

Since $\bar{v} \geq 0$, conditions (5.47) and (5.48) are contradictory. Therefore, constant speed minimizes aerodynamic drag loss.

The braking term $E_{b}$ can take no better value than 0 . This leads directly to a fundamental conclusion about wheel-to-distance energy minimization over varying road grade: if braking is never required to maintain constant speed, then constant speed is optimal. This explains why some parameter sets on some roads may not experience a benefit from wheel-to-distance energy minimization when compared to a constant-speed policy like the baseline of this study.

The LP reduces braking energy and increases aerodynamic drag relative to constant speed, so the
amount of braking energy places an upper bound on the potential gain from optimization. Fig. 5.14 shows the energy loss pathways for parameter sets A and B traveling over the Lanesville-Siberia Route at $25 \mathrm{~m} / \mathrm{s}$. Parameter set A loses $3.2 \%$ of its input energy to braking, while parameter set B loses $32.7 \%^{2}$. This difference explains the large gap in fuel economy improvement during closed-loop simulation. Since braking occurs at constant speed when

$$
\begin{equation*}
-m g \sin \theta>\mu m g \cos \theta+\frac{1}{2} \rho_{a} C_{d} A v^{2} \tag{5.49}
\end{equation*}
$$

more massive vehicles dealing with steeper slopes with lower rolling resistance, lower speed, and lower drag properties are likely to gain more from this type of optimization.


Figure 5.14: Energy loss breakdown for the two truck parameter sets.

### 5.8.3 Eco-Driving Guidelines

Figure 5.11 shows general behavior over rolling hills that could serve as a guideline for human drivers. Even the optimal controller must brake if the maximum speed constraint is encountered while coasting down a hill, so it uses its preview capability to cease engine power and slow down before reaching the hill's crest. This causes it to reduce engine power sooner than MOBIL does at various points. The slowdown allows more margin for speed to rise while coasting down the hill before braking is needed.

Therefore, it is recommended that drivers follow the following guidelines to reduce energy use on hilly roads. If the slopes are mild enough that no braking is expected, constant speed should be maintained. If

[^5]braking is expected after the crest of a hill, it is recommended to lift the accelerator pedal before peak elevation to allow the upward slope to slow the vehicle, then coast down the hill until the maximum speed is reached. Of course, the magnitude of this slowdown must depend on the minimum safe speed for the traffic conditions.

## Chapter 6

## Collaborative Guidance

The previously summarized algorithms used a fully decentralized and ego-centric architecture, meaning that each vehicle computes its own control move and does so considering only its own states in the objective. While such an informed but greedy approach can realize improvements over reactive algorithms, groups of vehicles can further benefit from collaboration. For example, vehicles on a highway may adjust their speeds to allow a merging vehicle to enter, reducing the overall disturbance to traffic flow.

Centralization has been proposed to realize such control performance. Centralized algorithms delegate control move determination to one agent that commands the others. This agent can be a lead vehicle in a string or a roadside server. Ideally, a centralized system can find the global optimum for the network and thereby achieve the best control performance. However, such systems can be practically limited by computational cost, delays, and communication volume. They also transfer control of the vehicle to a remote supervisor, reducing the user benefit relative to mass transit. Therefore, the main goal of this section is to propose distributed collaborative approaches where vehicles maintain autonomy but work toward a group objective. Centralized control is also implemented as a high-performing benchmark. Unlike the preceding studies, these focus on electric vehicles and therefore the energy results account for regenerative braking.

The collaborative guidance research is divided into car following and multi-lane algorithms, each with accompanying simulation methods and results. These two sections summarize the author's prior research in [26] ${ }^{1}$ and [27] ${ }^{2}$, respectively, and more detail can be found in those references. This research was

[^6]conducted under the supervision of Antonio Sciarretta at IFP Energies nouvelles, and was supported by the Chateaubriand Fellowship of the Office for Science \& Technology of the Embassy of France in the United States and the U.S. Department of Energy Vehicle Technologies Office (Project No. DE-EE0008232).

### 6.1 Car Following

This section will focus on collaborative guidance in a single lane. Algorithms are presented first, followed by simulation methods and finally results.

### 6.1.1 Algorithms

This section's decentralized car following controller is a single-lane form of the hierarchical lane change algorithm in Section 4.2. Equation (6.1) summarizes the optimal control problem (OCP), where $x_{e}$ and $u_{e}$ are the state and control input deviations from the reference. As before, these references are based on the parabolic velocity profile in Section 4.2. The matrices $P, Q$, and $R$ are weights. Other constraints follow Chapter 2's notation. The maximum acceleration constraint defined by the bound $\bar{a}$ and mixed constraint coefficients $m_{c}$ and $b_{c}$ is set up to capture the electric vehicle's (EV's) maximum acceleration limit as a function of speed. A notable difference between OCP (6.1) and Eqn. (2.2) is that this problem lacks a gap tracking term. This means that the ego vehicle will follow the energy-optimal parabolic trajectory even if the preceding vehicle ( PV ) is moving fast enough to pull away. A constraint relative to the preceding vehicle's position $s_{p}$ is retained for collision avoidance.

$$
\begin{align*}
\min J= & x_{e}^{\mathrm{T}}(N) P x_{e}(N)+\sum_{i=0}^{N-1}\left[x_{e}^{\mathrm{T}}(i) Q x_{e}(i)+u_{e}^{\mathrm{T}}(i) R u_{e}(i)\right] \\
\text { s.t. } & \underline{u} \leq u(i) \leq \bar{u} \\
& u(i)-m_{c} v(i) \leq b_{c}, \quad i \in\{0,1, \cdots N-1\} \\
& v(i) \geq 0  \tag{6.1}\\
& v(i) \leq \bar{v} \\
& a(i) \leq \bar{a} \\
& a(i)-m_{c} v(i) \leq b_{c} \\
& s(i) \leq s_{p}(i)-\underline{d}, \quad i \in\{1, \cdots N\}
\end{align*}
$$

In contrast, the centralized OCP version considers not only the ego vehicle's $x_{e}$ and $u_{e}$, but the deviations of the entire network from their individual references. Therefore, the centralized OCP in Eqn. (6.2a) deals with the group state and control deviations $\mathcal{X}_{e}$ and $\mathcal{V}_{e}$. The group weighting matrices $\mathcal{P}, \mathcal{Q}$, and $\mathcal{R}$ are formed by arranging each vehicle's weighting matrices block-diagonally. Where the decentralized OCP only needed a single position constraint to prevent unsafe proximity to the PV, (6.2a) uses several constraints that keep each of the $m$ vehicles of index $j$ a safe distance from its leader with index $j-1$. Other individualvehicle constraints like acceleration and speed limits are retained and encapsulated in the constraint-admissible set $\Xi$. The symbol $\otimes$ denotes the Kronecker product [127].

$$
\begin{array}{ll}
\min \mathcal{J}= & \mathcal{X}_{e}^{\mathrm{T}}(N) \mathcal{P} \mathcal{X}_{e}(N)+\sum_{i=0}^{N-1}\left[\mathcal{X}_{e}^{\mathrm{T}}(i) \mathcal{Q} \mathcal{X}_{e}(i)+\mathcal{V}_{e}^{\mathrm{T}}(i) \mathcal{R} \mathcal{V}_{e}(i)\right] \\
\text { s.t. } & s_{j}(i) \leq s_{j-1}(i)-\underline{d} \\
& \left\{x_{j}, u_{j}\right\} \in \Xi \\
& i \in\{1, \cdots N\}, j \in\{1, \cdots m\} \\
& \mathcal{P}=\mathrm{I}_{m} \otimes P, \quad \mathcal{Q}=\mathrm{I}_{m} \otimes Q, \quad \mathcal{R}=\mathrm{I}_{m} \otimes R \tag{6.2b}
\end{array}
$$

The distributed and cooperative version of the single-lane algorithm uses OCP (6.2a) with network and architectural differences. Instead of a single vehicle optimizing over the whole network and dictating control moves to the other vehicles, each cooperative vehicle solves its own OCP that involves itself and its immediate neighbors. This means that at most 3 vehicles are included in each OCP , which promotes scalability. More formally, the constraints for vehicle $q$ become

$$
\begin{equation*}
\mathcal{I}=\{j \mid j \in\{q-1, q, q+1\} \cap\{1, \cdots m\}\} \tag{6.3a}
\end{equation*}
$$

$$
\begin{align*}
& \left\{x_{j}, u_{j}\right\} \in \Xi \\
& s_{j}(i) \leq s_{j-1}(i)-\underline{d}  \tag{6.3b}\\
& i \in\{1, \cdots N\}, j \in \mathcal{I}
\end{align*}
$$

where $\mathcal{I}$ is the set of vehicle indices whose trajectories are optimized. The trajectory of vehicle $j=\max \{0, q-2\}$, received from the previous step's solution via connectivity, is considered fixed. The objective of OCP (6.2a)
is retained, except that $\mathcal{X}_{e}$ and $\mathcal{V}_{e}$ are only composed of vehicles whose indices belong to $\mathcal{I}$. Only the ego vehicle's control input is applied, but it is decided considering potential responses of neighboring vehicles.

### 6.1.2 Simulation Methods

These three algorithms were simulated in 8 -vehicle strings following a lead vehicle. This leader tracked either the Worldwide Light-duty Test Cycle (WLTC) High or Low [136], which is shown in Fig. 6.1, reproduced from [26]. These two cycles exposed the controllers to different speeds. The WLTC Low involves complete stops. Since some stops like those at stop signs cannot be avoided using anticipative driving, the WLTC Low cycle was divided into five separate simulations. These segments of the cycle corresponding to each simulation are demarcated using arrows in Fig. 6.1. Classical ACC (Section 1.1.3.1) and a position-


Figure 6.1: The drive cycles for single-lane multi-agent simulations [26].
constrained shrinking horizon controller (PCSHC) from [109] served as bases for comparison. Two ACC time headways were used: 0.8 s to represent a string unstable controller and 1.5 s to represent a string stable one.

### 6.1.3 Results

This section provides the results for net energy from the EV's battery, mean string length, acceleration over the string, and speed trajectories. Acronyms are used for each algorithm throughout and their meanings are listed in Table 6.1, which is adapted from [26].

The two different ACC time headways outline the tradeoff between string length and energy efficiency when preview is not used (Figure 6.2, reproduced from [26]). While PCSHC outperformed ACC for a single vehicle [26], its string instability worsened the combined performance over 8 vehicles. The hier-

Table 6.1: Acronyms for single-lane multi-agent algorithms.

| Controller | Acronym |
| :---: | :---: |
| Adaptive Cruise, 0.8 s Headway | ACC, 0.8 s |
| Adaptive Cruise, 1.5 s Headway | ACC, 1.5 s |
| Position-Constrained Shrinking Horizon | PCSHC |
| Decentralized Hierarchical | DHC |
| Centralized Hierarchical | CHC |
| Cooperative Hierarchical | CoHC |

archical algorithms all realized similar energy improvements relative to ACC. However, the centralized and cooperative algorithms managed to do so in a more compact way, pushing them toward the higher-performing bottom-left corner of Fig. 6.2. Overall, the cooperative controller performed similarly well or, in the case of string length, better than the centralized approach.


Figure 6.2: Energy use and string length in WLTC High and Low cycles [26].

String stability is examined to help explain the results in Fig. 6.2. Since not all algorithms tracked a target headway, changes in mean acceleration from the front (lower index) to the rear (higher index) of the string were used to indicate string stability. If a controller is string stable, accelerations are expected to fall as vehicle index increases. The results, shown in Fig. 6.3 which is reproduced from [26], verify that ACC's string stability depends on time headway. The hierarchical algorithms all returned similar performance where the first CAV achieved relatively low average acceleration, which did not change significantly throughout the rest of the string. The analytical position-constrained shrinking-horizon controller (PCSHC) performed similarly to the hierarchical MPC algorithms for the first vehicle, but was string unstable. The latter property was responsible for PCSHC's higher string energy use.


Figure 6.3: Mean absolute acceleration at various string indices [26].

The velocity trajectories shown in Fig. 6.4, reproduced from [26], further demonstrate the phenomena that led to the performance differences in Fig. 6.2. PCSHC's trajectories are smoother than the cycle near the front of the string but amplify toward the rear of the string. On the other hand, the hierarchical controllers are consistently smooth in their interactions with PVs.


Figure 6.4: Velocity trajectories in the WLTC Low Cycle [26].

Perhaps the most surprising result in this single-lane multi-agent study is that collaboration or even centralization of the hierarchical MPC made less than a $2 \%$ impact on energy use. Instead, collaborative guid-
ance mainly reduced the road space occupied by the string. This result suggests that, given similar eco-driving controller designs and vehicle hardware, decentralized control is sufficient under sparse traffic conditions. Collaborative guidance would become more critical when closer spacing is required to support elevated demand.

### 6.2 Multi-Lane Guidance

As in car following, three lane-change algorithms are presented: ego-centric decentralized, centralized, and distributed cooperative. After these variants on Chapter 4's hierarchical MPC are described, a 3-lane collaboration-intensive obstacle avoidance scenario is presented for use in evaluation. The chapter closes with results on the effectiveness of collaborative guidance in this multi-lane situation.

### 6.2.1 Algorithms

From a multi-agent control perspective, multi-lane guidance differs from car following in the highly time-varying spacial arrangement of vehicles in the network. For example, a vehicle may lead the ego at one time and, several seconds later, move into a different lane and expose the ego to a different leader. This presents a problem for position constraints like the one in (6.1) that only consider vehicles that are known to be neighbors a priori. Therefore, a centralized multi-lane guidance formulation must consider position constraints between arbitrary vehicles $p$ and $q$ in the network as in Eqn. (6.4). Notation is borrowed from Chapter 4 with the addition of subscripts to distinguish different controlled vehicles. Binary variables $\beta$ with subscript $I$ indicate whether vehicle $p$ or $q$ is ahead.

$$
\begin{align*}
& s_{p}-s_{q}-M \beta_{I j}-M\left(2-\mu_{\lambda p}^{a}-\mu_{\lambda p}^{b}\right)-M\left(2-\mu_{\lambda q}^{a}-\mu_{\lambda q}^{b}\right)-\epsilon_{1 j} \leq-\underline{d}  \tag{6.4a}\\
& -s_{p}+s_{q}-M \beta_{I j}^{\mathrm{C}}-M\left(2-\mu_{\lambda p}^{a}-\mu_{\lambda p}^{b}\right)-M\left(2-\mu_{\lambda q}^{a}-\mu_{\lambda q}^{b}\right)-\epsilon_{1 j} \leq-\underline{d} \tag{6.4b}
\end{align*}
$$

Collisions with obstacles that are outside the network are handled using the method from Chapter 4.
As in car following, the centralized multi-lane objective considers the group state vector deviation $\mathcal{X}_{e}$ and the group control vector deviation $\mathcal{V}_{e}$ relative to nominal trajectories that the vehicles compute individually. The affine constraints include the position constraints above along with each vehicle's speed limits, acceleration limits, etc. described previously in Section 4.3. These constraints are encapsulated in $S_{a}$ and $\Xi$
for brevity. The decision variables for all vehicles in the network compose the vector $\overline{\mathcal{V}}$.

$$
\begin{align*}
\min _{\bar{\imath}} \mathcal{J}= & \mathcal{X}_{e}^{\mathrm{T}}(N) \mathcal{P} \mathcal{X}_{e}(N)+\sum_{i=0}^{N-1}\left[\mathcal{X}_{e}^{\mathrm{T}}(i) \mathcal{Q} \mathcal{X}_{e}(i)+\mathcal{V}_{e}^{\mathrm{T}}(i) \mathcal{R} \mathcal{V}_{e}(i)\right]+\mathcal{J}_{\bar{\epsilon}}(\bar{\epsilon})  \tag{6.5}\\
\text { s.t. } & S_{a} \overline{\mathcal{V}} \leq \Xi
\end{align*}
$$

This centralized optimal controller is abbreviated CO.
The decentralized algorithm was described earlier in Chapter 4. That algorithm, here termed Decentralized Ad-Hoc (DAH), is first-come-first-served in the sense that each vehicle computes its control move subject to currently available SV trajectories and claims a certain spatiotemporal zone. Thus the network solution depends on the computation order and it is not guaranteed that the order will be a good one for the group objective.

Therefore, the cooperative multi-lane algorithm seeks an improvement in the group objective compared to DAH by finding a good ordering dynamically. It does so by establishing a priority measure $\mathbf{P}_{p}$ for each vehicle, then ordering control move computation by decreasing priority. The priority is chosen as the optimal objective gradient of a nominal problem assuming that other vehicles yield to the ego.

$$
\begin{equation*}
\mathbf{P}_{p}=\left\|\nabla_{U} J_{p}\right\|=\left\|\frac{1}{2}\left(G_{p}+G_{p}^{\mathrm{T}}\right) U_{p}^{\star}+f_{p}\right\| \tag{6.6}
\end{equation*}
$$

The intuitive idea behind Eqn. (6.6) is now explained. In the absence of other CAVs in the network, each vehicle faces a different situation with respect to fixed obstacles and vehicles that do not belong to the collaborative network. Each vehicle thus has its own ideal trajectory that it would follow if it were first in the computation order. If other vehicles compute first instead, more constraints are present that may cause a deviation from this ideal solution at an excess cost. However, this order-dependent excess cost is not equal for all vehicles. Therefore, the multi-agent objective can be reduced by prioritizing those vehicles that suffer the most from the imposition of additional constraints. This expression of the total cost $\mathcal{J}$ in terms of each vehicle's ideal cost $J_{q}^{\star}$ and the excess cost $\Delta J_{q}$ for $m$ vehicles is formalized as

$$
\begin{equation*}
\mathcal{J}=\sum_{q=1}^{m}\left(J_{q}^{\star}+\Delta J_{q}\right) . \tag{6.7}
\end{equation*}
$$

The variable part of the total cost is approximately

$$
\begin{equation*}
\sum_{q=1}^{m} \Delta J_{q} \approx \sum_{q=1}^{m} \nabla_{\bar{U}} J_{q} \cdot \mathbf{d}_{q} \tag{6.8}
\end{equation*}
$$

for local deviations $\mathbf{d}_{q}$ from the nominal optimum. Placing a vehicle later in the priority order adds to $\mathbf{d}_{q}$, although the precise impact on its magnitude and direction is unknown in advance. The objective gradient $\nabla_{\bar{U}} J_{q}$, on the other hand, can be readily obtained by solving only the ideal problem where vehicle $q$ is unimpeded by other in-network vehicles. Using this gradient as the priority measure in Eqn. (6.6) seeks to pair larger gradients with smaller deviation vectors $\mathbf{d}_{q}$, thereby reducing the overall cost.

The network solution is computed in a sequence of steps at each control loop. In the first step, each vehicle solves such a nominal problem to obtain a solution $U_{p}^{\star}$ along with MIQP coefficients $G_{p}$ and $f_{p}$. Then, each vehicle $p$ computes its priority $\mathbf{P}_{p}$ and communicates it to the other vehicles. All vehicles can now determine the ordering and assemble their MIQPs such that position constraints with lower-priority vehicles are relaxed. Finally, each vehicle solves its MIQP and applies the resulting optimal control. Algorithm 1 from [27] captures this process. The decentralized controller with sensitivity-based prioritization is abbreviated DSP.

### 6.2.2 Simulation Methods

Figure 6.5 illustrates the collaboration-intensive simulation scenario used to evaluate the proposed algorithms. Three vehicle begin and end the simulation at rest and a constant final time boundary condition is used for all of the MPC variants. The side-by-side initial vehicle placement and severe bottleneck induced by the obstacles demand that the vehicles adjust their speeds away from their individually optimal trajectories to avoid a collision. This emphasizes collaboration. The obstacles are shown in black in Fig. 6.5 and the vehicles are in color. A segment of road exists beyond the frame shown in Fig. 6.5, but no obstacles are present there. The total distance of the simulations can be found in Fig. 6.6.

All 6 possible DAH orderings were simulated to avoid focusing on just one possible computation order. The MOBIL lane change model [121] was introduced as a baseline, with a few modifications to enable MPC-like travel times. One of these changes was the addition of an enlarged buffer distance when approaching static obstacles, based on [32]. Existing MOBIL parameters were also altered as shown in Table 6.2, reproduced from [27]. An acceleration $\Delta a_{\text {bias }}$ was applied to the right-hand side of inequality (1.6) to mimic each MPC vehicle's lane preferences.

```
Algorithm 1 Communication and optimization in collaborative decentralized control.
    procedure Compute Network Control Moves
        \(\mathcal{L} \leftarrow\) Set of cooperative agents
        \(\mathcal{E} \leftarrow\) Set of other obstacles
        \(\backslash \backslash x_{p q}(i)\) denotes agent \(p\) 's stored state of object \(q\)
            for prediction step \(i \in \mathbf{Z}\).
        \(\backslash \backslash \mathbf{P}_{p q}\) denotes agent \(p\) 's stored priority of object \(q\).
        \(\backslash\) Superscript \(*\) indicates an optimal solution.
        for \(p \in \mathcal{L}\) do
            \(x_{p(\mathcal{L} \cup \mathcal{E})}(0) \leftarrow \operatorname{sense}(\mathcal{L} \cup \mathcal{E})\)
        for \(p \in \mathcal{L}\) do
            \(\backslash \backslash\) Solve the nominal problem, unimpeded by cooperative vehicles.
            \(\left\{\bar{U}_{p}^{\star}, G_{p}, f_{p}\right\} \leftarrow\) solveNomOCP \(\left(x_{p(\mathcal{L} \cup \mathcal{E})}(0)\right)\)
            \(\backslash \backslash\) Evaluate Eqn. (6.6).
            \(\mathbf{P}_{p p} \leftarrow\) computePriority \(\left(\bar{U}_{p}^{\star}, G_{p}, f_{p}\right)\)
            send \(\left(\mathbf{P}_{p p}\right)\)
        for \(p \in \mathcal{L}\) do
            \(\left\{\mathbf{P}_{p(\mathcal{L} \backslash p)}, x_{p(\mathcal{L} \backslash p)}(i)\right\} \leftarrow \operatorname{get} \operatorname{Comm}, i \in[1, N]\)
        for \(p \in \mathcal{L}\) do
            \(\backslash \backslash\) Solve the decentralized OCP with all constraints.
            \(\left\{\bar{U}_{p}^{*}, x_{p}^{*}(i)\right\}=\operatorname{solveFinalOCP}\left(\mathbf{P}_{p \mathcal{L}}, x_{p(\mathcal{L} \backslash p)}(i), x_{p \mathcal{E}}(0)\right)\)
            setControl \(\left(u^{*}(0)\right) \quad \backslash \backslash \bar{U}_{p}^{*}\) contains \(u^{*}(0)\).
            \(\operatorname{send}\left(x_{p}(i)\right), i \in[0, N]\)
```



Figure 6.5: Collaborative vehicles navigate the case study's scenario. ©2020 IEEE.

### 6.2.3 Results

To observe the operation of the proposed prioritization approach, see Fig. 6.7. Since the right-hand vehicle must change lanes twice to avoid the obstacle, it is highly sensitive to the other vehicles' decisions and therefore has a higher initial priority. The other agents allow it to reserve its trajectory first. Once vehicle 3's situation improves, the priority order changes to allow other vehicles to compute first as appropriate.

The multi-agent system's initial sensitivity to the computation order causes variation in the DAH results, both qualitatively in Fig. 6.6 and in the error bars of Fig. 6.2.3. These error bars mark the best- and worst-case results. DSP consumed less energy than even the best DAH ordering. This is possible because DAH maintains a fixed ordering at all times while DSP dynamically reevaluates the priorities to find the best

Table 6.2: Modified IDM and MOBIL Parameters

|  | $v_{0}(\mathrm{~m} / \mathrm{s})$ | $a_{0}\left(\mathrm{~m} / \mathrm{s}^{2}\right)$ | $b_{0}\left(\mathrm{~m} / \mathrm{s}^{2}\right)$ | $\Delta a_{\text {bias }}\left(\mathrm{m} / \mathrm{s}^{2}\right)$ |
| :--- | :---: | :---: | :---: | :---: |
| Standard | 15 | 1.0 | 1.5 | 0.3 |
| Modified | 33.3 | 2.8 | 4.2 | 1.0 |



Figure 6.6: Velocity trajectories resulting from the three presented algorithms. ©2020 IEEE.
one. The centralized optimal controller $(\mathrm{CO})$ performed best as expected given the idealized computation and communication, reducing energy consumption by $8.6 \%$ relative to DAH. The distributed sensitivity-based DSP, which used 6.7 \% less energy than DAH, realized most of CO's benefit. Its computation time was also much more promising as shown in Table 6.3. Network computation times in Table 6.3 account for stacked sequential computation in DAH and parallel computation in DSP.


Figure 6.7: The DSP priority order's evolution. ©2020 IEEE.


Figure 6.8: MOBIL, decentralized, collaborative, and centralized performance. ©2020 IEEE.

Table 6.3: MATLAB computation time for collaborative algorithms.

|  | MOBIL | DAH | DSP | CO |
| :--- | :---: | :---: | :---: | :---: |
| Vehicle Mean (s) | 0.0011 | 0.4552 | 0.6919 | N/A |
| Network Mean (s) | 0.0017 | 1.3553 | 0.9247 | 7.4093 |

## Chapter 7

## Summary

This dissertation has addressed the use of anticipation for safe and ecological control of automated vehicles with emphasis on non-convexity and uncertainty issues. It consists of four technical chapters: car following, lane selection, collaborative guidance, and nonlinear motion planning. The completed research on car following, lane change decisions, and collaborative guidance was summarized drawing on the author's publications [29], [30] (car following), [25], [31], [32], [33] (lane change), [26], and [27] (collaborative guidance). Chapters 3 and 6 document research collaborations with Gábor Orosz and Tamás G. Molnár at the University of Michigan and Antonio Sciarretta at IFP Energies nouvelles, France. Motivated by a project with Cummins, Chapter 5 went beyond the published research to focus on long-term speed planning over varying road grade. The same project also led to additional techniques for extending the lane change algorithms to tractor-trailers on sloping roads. This concluding chapter will summarize the accomplishments, state the limitations, and envision future research.

### 7.1 Accomplishments and Contributions

Optimal control algorithms have been designed and evaluated for several applications related to anticipative automated driving. This first of these was a car-following algorithm for heterogeneous strings using model predictive control. Probability models were developed to predict the predecessor's motion and one such model was applied to real-world Tiger Commute bus data. In vehicle-in-the-loop experiments, car following with model predictive control yielded $8 \%$ to $23 \%$ reduction in energy use relative to the Wiedemann human driver model depending on scenario and powertrain type. This result was in-line with MATLAB and VISSIM
simulations. Car following with multiple predecessors was also explored, with a combination of MPC and multiple-predecessor information improving energy performance compared to using either alone. That study used measurements from a string of real human drivers as input to simulate the research controllers' responses.

The car following research expanded naturally into non-convex lane change planning. A higherlevel planner based on Pontryagin's Minimum Principle enabled a mixed integer receding horizon controller to follow a more optimal overall velocity profile. A systematic constraint formulation enabled anticipative avoidance of multiple dynamic obstacles. The energy performance of this algorithm was compared to a rulebased lane change approach in both homogeneous and mixed-traffic situations, resulting in improvements of $3.4 \%$ fleet energy when one in three vehicles were connected and automated up to $16.2 \%$ when all vehicles were automated. Chance constraints were introduced to help avoid collisions in dense traffic, and design variants were evaluated using an exceptionally hazardous scenario based on a real AV collision.

This lane change algorithm was extended to tractor-trailers on sloping roads. The mixed-integer programming formulation was modified to include the effects of tractor-to-trailer angle change and the model was linearized online to approximately comprehend speed and road grade effects. The resulting MPC was coupled with an improved long-term speed planner that minimized wheel input energy including the effects of road grade and aerodynamic drag. A novel LP formulation enabled this optimization to be solved online in closed-loop. This system was evaluated over the real road slope profile from Lanesville, IN to Siberia, IN. A target fuel consumption reduction of $10 \%$ was set based on experience with optimal passenger-vehicle lane changing in [31] [33], and the system was compared to a commonly used rule-based baseline algorithm. Depending on the truck's parameters such as mass and rolling resistance, fuel consumption benefits were either insignificant or exceeded the target at $13.7 \%$. Failure to meet the $10 \%$ target with one of the parameter sets motivated an investigation of the fundamental energy pathways, optimal strategies, and possibilities for energy savings. The results were ultimately explained by first showing that constant speed is energy-optimal if it does not require braking, and then noticing that the second parameter set required several times as much braking energy as the first did to maintain constant speed.

Collaborative guidance algorithms were proposed and evaluated for both car following and lane changing, showing that in some scenarios, traffic compactness and/or energy benefits are possible over optimal decentralized algorithms. The multilane algorithm assigned each vehicle a priority based on its objective gradient, enabling vehicles to help others that may be in more difficult situations. Where centralized optimization was able to reduce EV energy use by $8.6 \%$ relative to decentralized MPC, the prioritization approach was able to realize most of this benefit with $6.7 \%$ savings.

### 7.2 Limitations

The algorithms presented here have handled surrounding obstacles, lane count, speed limits, and road slope features generally. Nonetheless, the model-based approach used relies on some prior knowledge of the environment's structure and therefore cannot generalize to all possible situations. Therefore, these algorithms only apply through SAE autonomy level 4. Various situations encountered over the course of a real vehicle's life, off-road use and non-standard event parking for example, would require intervention by either a human or a well-trained machine learning system.

This also raises the question of how to recognize when an optimal algorithm for a given situation is available and select the appropriate one. Real-world implementation would require a solution to this problem that has not been presented here. A similar situation exists with other surrounding software components including sensing and routing.

The use of kinematic models for control throughout also presents limitations. Kinematic models approximate wheel motion on the pavement using a no-slip lateral condition. In reality, a difference between wheel direction and actual motion exists and corresponds to the forces that move the vehicle body. More complex dynamic models include these physics [78]. Polack et al. [101] found that kinematic models only match dynamic models up to about 0.5 g on dry pavement, meaning that these algorithms are not suitable for limit-handling situations.

As pointed out in [53], a limitation of most optimal-control, model-based vehicle guidance algorithms is the assumption that surrounding vehicle trajectories are fixed. That is, the future impact of the ego's decisions on the decisions of surrounding vehicle is neglected. This limitation applies to the base carfollowing and lane-change algorithms presented here. Although the collaborative algorithms in Chapter 6 do consider coupling between multiple agents, they only enable collaboration with other CAVs in their present form. Collaboration and even competition with human drivers is especially important in near-term practical applications and was not addressed here.

### 7.3 Recommendations for Further Research

Most of the limitations above could inspire future research. For example, one can imagine how the sensitivity-based collaboration algorithm in Section 6.2 could be extended to detect when a human driver faced a difficult situation, then accommodate that human with cooperative driving. Game-theoretic approaches may
be more appropriate when dealing with uncooperative human drivers and preventing platoon cut-ins.
Advanced human driver modeling is critical for such research. While several longitudinal models exist and have been correlated based on macroscopic metrics [14], correlated multi-lane human driver models are more sparse. More research is also needed to determine human driver models' accuracy at the sub-microsopic level. In other words, a model may predict traffic flow down a road segment while missing the specific path the vehicles follow between lanes or the initial reaction to a change in lateral position by a surrounding vehicle. The validity of popular human driver models for this type of prediction needs to be understood.

Assuming that the technical challenges above can be solved and automated vehicles can cooperate with other AVs and humans, each vehicle owner and operator should benefit from using such features. For example, a currency-based system might help compensate vehicles that slightly increase their own energy use to more significantly reduce overall energy use. This topic is inherently multidisciplinary and could benefit from expertise in controls, economics, and psychology, to name a few broad fields.

A less futuristic research direction could seek eco-coaching and driver assistance applications of the LP speed planner presented in Chapter 5. While this dissertation integrated the energy-minimizing LP with an automatic controller, it could also provide a human driver with instructions based on preview of upcoming road slopes. This guidance could instruct a driver to follow a specific speed. An even simpler interface might advise the driver that the downward slope following an upcoming hill crest is steep enough that braking will be needed to maintain the speed limit. It could then coach the driver to lift the accelerator pedal and allow the vehicle to slow down before the crest. Even without an online optimization, insights from the LP could be reduced to simple rules on when to maintain constant speed and when to coast while driving on a hilly road. Development of such rules and quantification of their benefits is interesting from an energy perspective since they could be widely deployed in driver education without new technology or hardware.

From the necessity of final-time boundary conditions in eco-driving solutions to the dependence of energy benefits on baseline final time in Section 4.6, this dissertation has frequently encountered the tradeoff between travel time and energy. While some trips may face hard constraints on travel time like those generally assumed here, some users may be willing to extend travel time on certain trips to save energy. On the other hand, using more energy to travel more quickly on one trip could enable a slower subsequent trip for an overall energy benefit. Accounting for the present value of uncertain future benefits and accurately informing users of the potential immediate benefit of a later arrival time could be addressed in a rigorous thermoeconomic analysis.

## Appendix A

## Energy Modeling

All energy models used to evaluate the proposed algorithms must consume vehicle-level kinematic information like velocity and acceleration and output instantaneous energy consumption. This involves a Newton's law calculation of the tractive force $F_{t}$ applied by either the engine or brake actuators, which are assumed to be applied at strictly separate times. Resistive forces are modeled with the coefficients $a_{r}$ and $c_{r}$, $m_{v}$ is the vehicle mass, and $g$ is the acceleration due to gravity. The model accounts for road grade $\theta$, although $\theta=0$ until Chapter 5.

$$
\begin{equation*}
F_{t}=a_{r}+c_{r} v^{2}+m_{v} g \sin \theta+m_{v} a \tag{A.1}
\end{equation*}
$$

The torque at the wheel $T_{w}$ is then determined using the tire radius $r_{t}$.

$$
\begin{equation*}
T_{w}=F_{t} r_{t} \tag{A.2}
\end{equation*}
$$

From here, the energy computation diverges depending on powertrain type.

## A. 1 Internal Combustion Engine Vehicles

In internal combustion engine vehicles (ICEVs), energy use takes the form of fuel flow $\dot{m}_{f}$ out of the tank. To a quasi-static approximation, the fuel flow rate is a function of engine speed $n_{e}$ and engine torque $T_{e}$. This operating point can be calculated from the speed $v$ and acceleration $a$ of the vehicle using the drivetrain gear ratio, which is itself a function of vehicle operating point.

Hence, the first step in the ICEV modeling workflow is to develop a gear lookup table $i_{g}\left(n_{t r_{o}}, T_{t r_{o}}\right)$
where the arguments are transmission output speed $n_{t r, o}$ and transmission output torque $T_{t r, o}$. Because of engine mechanical limits, not all gears are feasible for a given vehicle operating point. The fuel consumption is computed for each feasible gear and the fuel-minimizing gear is placed in the gear map. The following process results in the fuel consumption given a gear index $i_{g}$.

First, the engine speed $n_{e}$ and torque $T_{e}$ are computed from the given transmission gear's ratio.

$$
\begin{equation*}
n_{e}=n_{t r, o} r_{t r} \quad T_{e}=\frac{T_{t r, o}}{r_{t r}} \tag{A.3}
\end{equation*}
$$

Then, the mass fuel flow from the engine is looked up based on this engine operating point. $T_{e, c r k}$ denotes the crankshaft engine torque i.e. the sum of $T_{e}$ and accessory torque. Brake specific fuel consumption (BSFC) data from [116] is used to include quasi-static nonlinear phenomena in the fuel calculation. Speeds $n$ have dimensions of revolutions per unit time.

$$
\begin{equation*}
\dot{m}_{f}=2 \pi n_{e} T_{e, c r k} B S F C\left(n_{e}, T_{e, c r k}\right) \tag{A.4}
\end{equation*}
$$

With the gear lookup table established, the model can be run on a sequence of vehicle operating points. The transmission output conditions are computed at each point according to Eqn. (A.5). Once these are determined, the fuel consumption is calculated from Eqns. (A.3, A.4) where $r_{f}$ denotes the final drive ratio.

$$
\begin{equation*}
T_{t r, o}=F_{t} \frac{r_{t}}{r_{f}}, \quad n_{t r, o}=v \frac{r_{f}}{2 \pi r_{t}} \tag{A.5}
\end{equation*}
$$

Fuel consumption for the modeled 1.6 L turbocharged Ford Escape was computed for the U.S. Environmental Protection Agency (EPA) fuel economy drive cycles, enabling comparison against the window sticker fuel economy after the proper calculation [1]. Table A. 1 lists the results.

Table A.1: Powertrain model results.

| Cycle | EPA Label [MPG] | Model Result [MPG] |
| :--- | :---: | :---: |
| City | 23 | 22.9 |
| Highway | 31 | 31.3 |

By substituting the model's constants and BSFC map, a Class 8 line haul truck was also modeled. Table A. 2 lists the differences in parameters between the passenger and heavy vehicles used in the simulation studies. Figures A. 1 and A. 2 show the resulting maximum acceleration limits and gear maps (color contours), along with sample operating point traces. The operating point traces were obtained by following a vehicle that exactly tracked the EPA US06 cycle.

Table A.2: Powertrain model constants.

| Symbol | Definition | Light-Duty | Heavy-Duty |
| :--- | :--- | :--- | :--- |
| $m$ | mass | 1671 kg | 19400 kg |
| $m_{e f f}$ | effective mass | 1706.9 kg | 19616 kg |
| $l_{v e h}$ | overall length | 4.52 m | 22 m |
| $C_{d}$ | drag coefficient | 0.29 | 0.544 |
| $A_{v}$ | frontal area | $2.733 \mathrm{~m}^{2}$ | $10.8 \mathrm{~m}^{2}$ |
| $\mu$ | friction coefficient | 0.0150 | 0.0150 |
| $r_{f}$ | final drive ratio | 3.21 | 4.88 |
| $r_{t}$ | tire radius | 0.3454 m | 0.60 m |
| $a_{\min }$ | braking capacity | $-8.5 \mathrm{~m} / \mathrm{s}^{2}$ | $-6.0 \mathrm{~m} / \mathrm{s}^{2}$ |



Figure A.1: Passenger vehicle gear map from Dollar and Vahidi [30]. ©2018 IEEE.


Figure A.2: Heavy vehicle gear map from Dollar and Vahidi [30]. ©2018 IEEE.

## A. 2 Electric Vehicles

An important feature of electric vehicles (EVs) compared to ICEVs is their ability to recover kinetic energy through regenerative braking. However, not all energy can be recaptured in this way. The Nissan Leaf that was modeled is front wheel drive and as such cannot recover braking energy dissipated at the rear wheels. The rear wheels must provide some braking effort to maintain safe vehicle dynamics. Based on guidelines from [20], Eqn. (A.6) introduces a brake split model to apportion traction force between the front and rear axles where $\underline{F}_{t}$ denotes the maximum brake force.

$$
F_{f}= \begin{cases}F_{t} & \frac{F_{t}}{\underline{F}_{t}(v)} \leq 0.04  \tag{A.6}\\ 0.73 F_{t}+0.0108 \underline{F}_{t}(v) & \frac{F_{t}}{\underline{F}_{t}(v)}>0.04\end{cases}
$$



Figure A.3: Battery output power from Dollar et al. [27]. ©2020 IEEE.

Another limit on regenerative braking comes from the battery itself. The maximum charging rate was obtained from [57] as a function of state-of-charge (SOC).

The vehicle's power electrical circuit is modeled as two parallel power sinks: one for the motor power $P_{m}$ and another for auxiliary loads $P_{a}$. These make up the total power $P_{l}=P_{a}+P_{m}$, where $P_{m}$ is computed from the demanded motor speed and torque while accounting for motor and inverter efficiency using maps from [16]. The current $i_{b}$ and the total power $P_{T}$ then follow from a circuit analysis. $R_{b}$ and $V_{0}$ denote the electrical resistance and battery voltage, respectively. In Eqn. (A.7a), the sign is used which yields battery output power within the charge and discharge limits. This sign is unique throughout the operating space.

$$
\begin{gather*}
i_{b}=\frac{V_{0}(S O C) \pm \sqrt{V_{0}^{2}(S O C)-4 R_{b} P_{l}}}{2 R_{b}}  \tag{A.7a}\\
P_{T}=P_{l}+i_{b}^{2} R_{b}=V_{0} i_{b} \tag{A.7b}
\end{gather*}
$$

Figure A. 3 shows the battery output power as a function of vehicle speed and acceleration on a flat road with $S O C=60 \%$.

## Bibliography

[1] Vehicle-specific 5-cycle fuel economy and carbon-related exhaust emissions calculations. §600.11412.
[2] Hadi Abbas, Youngki Kim, Jason B. Siegel, and Denise M. Rizzo. Synthesis of Pontryagin's maximum principle analysis for speed profile optimization of all-electric vehicles. Journal of Dynamic Systems, Measurement, and Control, 141(7), 2019.
[3] Ravindra K. Ahuja, James B. Orlin, Stefano Pallottino, and Maria Grazia Scutellà. Minimum time and minimum cost-path problems in street networks with periodic traffic lights. Transportation Science, 36(3):326-336, 2002.
[4] Tyler Ard, Faraz Ashtiani, Ardalan Vahidi, and Hoseinali Borhan. Optimizing gap tracking subject to dynamic losses via connected and anticipative MPC in truck platooning. In 2020 American Control Conference (ACC), pages 2300-2305. IEEE, 2020.
[5] Tyler Ard, R. Austin Dollar, Ardalan Vahidi, Yaozhong Zhang, and Dominik Karbowski. Microsimulation of energy and flow effects from optimal automated driving in mixed traffic. Transportation Research Part C: Emerging Technologies, 120:102806, 2020.
[6] Tyler Ard, Longxiang Guo, R. Austin Dollar, Alireza Fayazi, Nathan Goulet, Yunyi Jia, Beshah Ayalew, and Ardalan Vahidi. Energy and flow effects of optimal automated driving in mixed traffic: Vehicle-in-the-loop experimental results. arXiv preprint arXiv:2009.07872, 2020.
[7] Behrang Asadi, Chen Zhang, and Ardalan Vahidi. The role of traffic flow preview for planning fuel optimal vehicle velocity. In ASME 2010 Dynamic Systems and Control Conference, pages 813-819. American Society of Mechanical Engineers Digital Collection, 2010.
[8] Mokhtar S. Bazaraa, John J. Jarvis, and Hanis D. Sherali. Linear programming and network flows. John Wiley \& Sons, 2008.
[9] Alberto Bemporad. Model predictive control design: New trends and tools. In Proceedings of the 45th IEEE Conference on Decision and Control, pages 6678-6683. IEEE, 2006.
[10] Alberto Bemporad, Daniele Bernardini, Michael Livshiz, and Bharath Pattipati. Supervisory model predictive control of a powertrain with a continuously variable transmission. In WCX World Congress Experience, pages 1-12. SAE International, 2018.
[11] Alberto Bemporad, Daniele Bernardini, Ruixing Long, and Julian Verdejo. Model predictive control of turbocharged gasoline engines for mass production. In WCX World Congress Experience, pages 1-10. SAE International, 2018.
[12] Lawrence Blincoe, Ted R. Miller, Eduard Zaloshnja, and Bruce A. Lawrence. The economic and societal impact of motor vehicle crashes, 2010 (revised). NHTSA tech. rep. DOT HS 812 013, NHTSA, Washington, DC, 2015.
[13] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. Predictive control for linear and hybrid systems. Cambridge University Press, 2017.
[14] Elmar Brockfeld, Reinhart D. Kühne, Alexander Skabardonis, and Peter Wagner. Toward benchmarking of microscopic traffic flow models. Transportation Research Record, 1852(1):124-129, 2003.
[15] Bureau of Transportation Statistics. Overview of U.S. petroleum prod., imports, exports, and consumption, July 2016. URL https://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/national _transportation_statistics/html/table_04_01.html.
[16] Tim Burress. Benchmarking state-of-the-art technologies. In Oak Ridge National Laboratory, 2013 US DOE Hydrogen Fuel Cells Program Vehicle Technologies Program Annual Merit Review Peer Evaluation Meeting, 2013.
[17] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on Robotics, 32(6):1309-1332, 2016.
[18] Ching-Yao Chan. Advancements, prospects, and impacts of automated driving systems. International Journal of Transportation Science and Technology, 6(3):208-216, 2017.
[19] Jianyu Chen, Shengbo Eben Li, and Masayoshi Tomizuka. Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning. arXiv preprint arXiv:2001.08726, 2020.
[20] Liang Chu, Liang Yao, Jian Chen, Libo Chao, Jianhua Guo, Yongsheng Zhang, and Minghui Liu. Integrative braking control system for electric vehicles. In 2011 IEEE Vehicle Power and Propulsion Conference, pages 1-5. IEEE, 2011.
[21] Felipe Codevilla, Eder Santana, Antonio M. López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9329-9338, 2019.
[22] R.C. Coulter. Implementation of the pure pursuit path tracking algorithm. Tech. rep. CMU-RI-TR-92-01, The Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, January 1992. See also URL http://www.dtic.mil/dtic/tr/fulltext/u2/a255524.pdf.
[23] George B. Dantzig. Origins of the simplex method. In A History of Scientific Computing, pages 141151. 1990.
[24] Daniela Pucci De Farias and Benjamin Van Roy. The linear programming approach to approximate dynamic programming. Operations Research, 51(6):850-865, 2003.
[25] R. Austin Dollar, Tamás G. Molnár, Ardalan Vahidi, and Gábor Orosz. MPC-based connected cruise control with multiple human predecessors. In 2021 American Control Conference (ACC). IEEE, 2021. To appear.
[26] R. Austin Dollar, Antonio Sciarretta, and Ardalan Vahidi. Information and collaboration levels in vehicular strings: A comparative study. In $21^{s t}$ IFAC World Congress. International Federation of Automatic Control (IFAC), 2020. To appear.
[27] R. Austin Dollar, Antonio Sciarretta, and Ardalan Vahidi. Multi-agent control of lane-switching automated vehicles for energy efficiency. In 2020 American Control Conference (ACC), pages 422-429. IEEE, 2020.
[28] R. Austin Dollar, Laurent Thibault, Mohamed Laraki, and Antonio Sciarretta. Eco-driving to minimize emissions of nitrogen oxides. 2020. In review.
[29] R. Austin Dollar and Ardalan Vahidi. Quantifying the impact of limited information and control robustness on connected automated platoons. In 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pages 1-7. IEEE, 2017.
[30] R. Austin Dollar and Ardalan Vahidi. Efficient and collision-free anticipative cruise control in randomly mixed strings. IEEE Transactions on Intelligent Vehicles, 3(4):439-452, 2018.
[31] R. Austin Dollar and Ardalan Vahidi. Predictively coordinated vehicle acceleration and lane selection using mixed integer programming. In ASME 2018 Dynamic Systems and Control Conference. American Society of Mechanical Engineers Digital Collection, 2018.
[32] R. Austin Dollar and Ardalan Vahidi. Automated vehicles in hazardous merging traffic: A chanceconstrained approach. IFAC-PapersOnLine, 52(5):218-223, 2019.
[33] R. Austin Dollar and Ardalan Vahidi. Multilane automated driving with optimal control and mixedinteger programming. IEEE Transactions on Control Systems Technology, 2021.
[34] Yaoqiong Du, Yizhou Wang, and Ching-Yao Chan. Autonomous lane-change controller. In 2015 IEEE Intelligent Vehicles Symposium (IV), pages 386-393. IEEE, 2015.
[35] Mihai Duguleana and Gheorghe Mogan. Neural networks based reinforcement learning for mobile robots obstacle avoidance. Expert Systems with Applications, 62:104-115, 2016.
[36] William B. Dunbar and Derek S. Caveney. Distributed receding horizon control of vehicle platoons: Stability and string stability. IEEE Transactions on Automatic Control, 57(3):620-633, 2011.
[37] Francesca M. Favarò, Nazanin Nader, Sky O. Eurich, Michelle Tripp, and Naresh Varadaraju. Examining accident reports involving autonomous vehicles in california. PLoS ONE, 12(9), 2017.
[38] S. Alireza Fayazi, Ardalan Vahidi, and Andre Luckow. Optimal scheduling of autonomous vehicle arrivals at intelligent intersections via MILP. In 2017 American Control Conference (ACC), pages 4920-4925. IEEE, 2017.
[39] Jared L. Gearhart, Kristin L. Adair, Richard J. Detry, Justin D. Durfee, Katherine A. Jones, and Nathaniel Martin. Comparison of open-source linear programming solvers. Sandia National Laboratories, SAND2013-8847, 2013.
[40] Peter G. Gipps. A model for the structure of lane-changing decisions. Transportation Research Part B: Methodological, 20(5):403-414, 1986.
[41] T. Griggs and D. Wakabayashi. How a self-driving Uber killed a pedestrian in Arizona. On the WWW, March 2018. URL https://www.nytimes.com/interactive/2018/03/20/us/self-driving-uber-pedestriankilled.html.
[42] George Gunter, Derek Gloudemans, Raphael E Stern, Sean McQuade, Rahul Bhadani, Matt Bunting, Maria Laura Delle Monache, Roman Lysecky, Benjamin Seibold, Jonathan Sprinkle, et al. Are commercially implemented adaptive cruise control systems string stable? IEEE Transactions on Intelligent Transportation Systems, 2020.
[43] Qiangqiang Guo, Ohay Angah, Zhijun Liu, and Xuegang Jeff Ban. Hybrid deep reinforcement learning based eco-driving for low-level connected and automated vehicles along signalized corridors. Transportation Research Part C: Emerging Technologies, 124:102980, 2021.
[44] Gurobi Optimization, LLC. Gurobi optimizer reference manual, 2021. URL http://www.gurobi.com.
[45] Allan Gut. An Intermediate Course in Probability. Springer, 2009.
[46] Shane Halbach, Phillip Sharer, Sylvain Pagerit, Aymeric P. Rousseau, and Charles Folkerts. Model architecture, methods, and interfaces for efficient math-based design and simulation of automotive control systems. In SAE 2010 World Congress \& Exhibition, 2010.
[47] Jihun Han, Antonio Sciarretta, Luis Leon Ojeda, Giovanni De Nunzio, and Laurent Thibault. Safe-and eco-driving control for connected and automated electric vehicles using analytical state-constrained optimal solution. IEEE Transactions on Intelligent Vehicles, 3(2):163-172, 2018.
[48] Jihun Han, Ardalan Vahidi, and Antonio Sciarretta. Fundamentals of energy efficient driving for combustion engine and electric vehicles: An optimal control perspective. Automatica, 103:558-572, 2019.
[49] Richard F. Hartl, Suresh P. Sethi, and Raymond G. Vickson. A survey of the maximum principles for optimal control problems with state constraints. SIAM Review, 37(2):181-218, 1995.
[50] Chaozhe R. He, I. Ge Jin, and Gábor Orosz. Fuel efficient connected cruise control for heavy-duty trucks in real traffic. IEEE Transactions on Control Systems Technology, 28(6):2474-2481, 2019.
[51] Manne Held. Fuel-efficient look-ahead control for heavy-duty vehicles with varying velocity demands. PhD thesis, KTH Royal Institute of Technology, 2020.
[52] L. L. Hoberock. A Survey of Longitudinal Acceleration Comfort Studies in Ground Transportation Vehicles. Journal of Dynamic Systems, Measurement, and Control, 99(2):76-84, 061977.
[53] Carl-Johan Hoel, Katherine Driggs-Campbell, Krister Wolff, Leo Laine, and Mykel J. Kochenderfer. Combining planning and deep reinforcement learning in tactical decision making for autonomous driving. IEEE Transactions on Intelligent Vehicles, 5(2):294-305, 2019.
[54] Serge P. Hoogendoorn. Multiclass continuum modelling of multilane traffic flow. PhD thesis, Delft University of Technology, September 1999.
[55] Serge P. Hoogendoorn and Piet H.L. Bovy. Continuum modeling of multiclass traffic flow. Transportation Research Part B: Methodological, 34(2):123-146, 2000.
[56] Jia Hu, Yunli Shao, Zongxuan Sun, Meng Wang, Joe Bared, and Peter Huang. Integrated optimal eco-driving on rolling terrain for hybrid electric vehicle with vehicle-infrastructure communication. Transportation Research Part C: Emerging Technologies, 68:228-244, 2016.
[57] Idaho National Laboratory. BEV battery testing results. Technical Report INL/MIS-14-31587, Idaho Falls, ID, July 2014.
[58] Insurance Institute for Highway Safety. Fatality facts 2018 yearly snapshot, 2019. URL https://www.iihs.org/topics/fatality-statistics/detail/yearly-snapshot.
[59] A. Ishihara, Y. Kuromaru, and M. Naka. Steering actuator for autonomous driving and platooning. In Proceedings of the JSAE Annual Congress, pages 5-8, 2009.
[60] Yuji Ito, Md. Abdus Samad Kamal, Takayoshi Yoshimura, and Shun-Ichi Azuma. Multi-vehicle coordination on merging roads based on pseudo-perturbation-based broadcast control. In 2018 Annual American Control Conference (ACC), pages 4008-4013. IEEE, 2018.
[61] Qiu Jin, Guoyuan Wu, Kanok Boriboonsomsin, and Matthew J. Barth. Power-based optimal longitudinal control for a connected eco-driving system. IEEE Transactions on Intelligent Transportation Systems, 17(10):2900-2910, 2016.
[62] K. Jones, A. Cortinovis, M. Mercangoez, and H.J. Ferreau. Distributed model predictive control of centrifugal compressor systems. IFAC-PapersOnLine, 50(1):10796-10801, 2017.
[63] Anan Kaku, Masakazu Mukai, and Taketoshi Kawabe. A centralized control system for ecological vehicle platooning using linear quadratic regulator theory. Artificial Life and Robotics, 17(1):70-74, Oct 2012.
[64] Md. Abdus Samad Kamal, Tomohisa Hayakawa, and Jun-ichi Imura. Road-speed profile for enhanced perception of traffic conditions in a partially connected vehicle environment. IEEE Transactions on Vehicular Technology, 67(8):6824-6837, 2018.
[65] Md. Abdus Samad Kamal, Jun-ichi Imura, Tomohisa Hayakawa, Akira Ohata, and Kazuyuki Aihara. Smart driving of a vehicle using model predictive control for improving traffic flow. IEEE Transactions on Intelligent Transportation Systems, 15(2):878-888, 2014.
[66] Md. Abdus Samad Kamal, Shun Taguchi, and Takayoshi Yoshimura. Efficient driving on multilane roads under a connected vehicle environment. IEEE Transactions on Intelligent Transportation Systems, 17(9):2541-2551, 2016.
[67] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing, pages 302-311, 1984.
[68] Eric C. Kerrigan and Jan M. Maciejowski. Soft constraints and exact penalty functions in model predictive control. In UKACC International Conference (Control 2000), 2000.
[69] Arne Kesting, Martin Treiber, and Dirk Helbing. General lane-changing model mobil for car-following models. Transportation Research Record, 1999(1):86-94, 2007.
[70] Namwook Kim, Sukwon Cha, and Huei Peng. Optimal control of hybrid electric vehicles based on Pontryagin's minimum principle. IEEE Transactions on Control Systems Technology, 19(5):12791287, 2010.
[71] Donald E. Kirk. Optimal control theory: an introduction. Dover Publications Inc., 2004.
[72] Jelena Kocić, Nenad Jovičić, and Vujo Drndarević. Sensors and sensor fusion in autonomous vehicles. In 2018 26th Telecommunications Forum (TELFOR), pages 420-425. IEEE, 2018.
[73] Michal Kvasnica. Implicit vs explicit MPC—similarities, differences, and a path owards a unified method. In 2016 European Control Conference (ECC), pages 603-603. IEEE, 2016.
[74] Heeyun Lee, Namwook Kim, and Suk Won Cha. Model-based reinforcement learning for eco-driving control of electric vehicles. IEEE Access, 8:202886-202896, 2020.
[75] Aaron Lelouvier, Jacopo Guanetti, and Francesco Borrelli. Eco-platooning of autonomous electrical vehicles using distributed model predictive control. In 20th International Conference on Intelligent Transportation Systems, 2017.
[76] Pu Li, Harvey Arellano-Garcia, and Günter Wozny. Chance constrained programming approach to process optimization under uncertainty. Computers \& Chemical Engineering, 32(1-2):25-45, 2008.
[77] Ricardo Lima and EWO Seminar. IBM ILOG CPLEX-what is inside of the box. In Proceedings of the 2010 EWO Seminar, pages 1-72, 2010.
[78] David J.N. Limebeer and Matteo Massaro. Dynamics and optimal control of road vehicles. Oxford University Press, 2018.
[79] Changliu Liu, Chung-Wei Lin, Shinichi Shiraishi, and Masayoshi Tomizuka. Distributed conflict resolution for connected autonomous vehicles. IEEE Transactions on Intelligent Vehicles, 3(1):18-29, 2017.
[80] Hongjie Ma, Hui Xie, and David Brown. Eco-driving assistance system for a manual transmission bus based on machine learning. IEEE Transactions on Intelligent Transportation Systems, 19(2):572-581, 2017.
[81] Jan Marian Maciejowski. Predictive control: with constraints. Pearson education, 2002.
[82] Silvia Magdici and Matthias Althoff. Adaptive cruise control with safety guarantees for autonomous vehicles. IFAC-PapersOnLine, 50(1):5774-5781, 2017.
[83] Andreas A. Malikopoulos and Liuhui Zhao. A closed-form analytical solution for optimal coordination of connected and automated vehicles. In 2019 American Control Conference (ACC), pages 3599-3604. IEEE, 2019.
[84] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 922-928. IEEE, 2015.
[85] Felicitas Mensing, Eric Bideaux, Rochdi Trigui, and Helene Tattegrain. Trajectory optimization for eco-driving taking into account traffic constraints. Transportation Research Part D: Transport and Environment, 18:55-61, 2013.
[86] Rhiannon Michelmore, Matthew Wicker, Luca Laurenti, Luca Cardelli, Yarin Gal, and Marta Kwiatkowska. Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 73447350. IEEE, 2020.
[87] Vicente Milanés, Steven E. Shladover, John Spring, Christopher Nowakowski, Hiroshi Kawazoe, and Masahide Nakamura. Cooperative adaptive cruise control in real traffic situations. IEEE Transactions on Intelligent Transportation Systems, 15(1):296-305, 2013.
[88] Dominik Moser, Roman Schmied, Harald Waschl, and Luigi del Re. Flexible spacing adaptive cruise control using stochastic model predictive control. IEEE Transactions on Control Systems Technology, 26(1):114-127, 2017.
[89] Masakazu Mukai and Taketoshi Kawabe. Model predictive control for lane change decision assist system using hybrid system representation. In 2006 SICE-ICASE International Joint Conference, pages 5120-5125. IEEE, 2006.
[90] National Highway Traffic Safety Administration. Corporate average fuel economy, March 2020. URL https://www.nhtsa.gov/laws-regulations/corporate-average-fuel-economy\#corporate-average-fuel-economy-fuel-economy-and-environment-label.
[91] Yurii Nesterov and Arkadii Nemirovskii. Interior-point polynomial algorithms in convex programming. SIAM, 1994.
[92] Daniel C.K. Ngai and Nelson H.C. Yung. Automated vehicle overtaking based on a multiple-goal reinforcement learning framework. In 2007 IEEE Intelligent Transportation Systems Conference, pages 818-823. IEEE, 2007.
[93] Ioannis A. Ntousakis, Ioannis K. Nikolos, and Markos Papageorgiou. On microscopic modelling of adaptive cruise control systems. In 4th International Symposium of Transport Simulation, pages 111127. Transportation Research Procedia, 2015.
[94] Sean O'Kane. Tesla hit with another lawsuit over a fatal autopilot crash. On the WWW, August 2019. URL https://www.theverge.com/2019/8/1/20750715/tesla-autopilot-crash-lawsuit-wrongful-death.
[95] J.C. Flores Paredes, G.P. Padilla Cazar, and M.C.F. Donkers. A shrinking horizon approach to eco-driving for electric city buses: Implementation and experimental results. IFAC-PapersOnLine, 52(5):556-561, 2019.
[96] Michael A. Patterson and Anil V. Rao. Gpops-II: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. ACM Transactions on Mathematical Software (TOMS), 41(1):1-37, 2014.
[97] Dario Pevec, Jurica Babic, Arthur Carvalho, Yashar Ghiassi-Farrokhfal, Wolfgang Ketter, and Vedran Podobnik. A survey-based assessment of how existing and potential electric vehicle owners perceive range anxiety. Journal of Cleaner Production, 276:122779, 2020.
[98] Mark Pfeiffer, Michael Schaeuble, Juan Nieto, Roland Siegwart, and Cesar Cadena. From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 1527-1533. IEEE, 2017.
[99] Juan Pimentel and Jennifer Bastiaan. Characterizing the safety of self-driving vehicles: A fault containment protocol for functionality involving vehicle detection. In 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), pages 1-7. IEEE, 2018.
[100] N.N. Pisaruk. Mixed integer programming: Models and methods. 2019.
[101] Philip Polack, Florent Altché, Brigitte d'Andréa Novel, and Arnaud de La Fortelle. The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In 2017 IEEE Intelligent Vehicles Symposium (IV), pages 812-818. IEEE, 2017.
[102] Warren B. Powell. What you should know about approximate dynamic programming. Naval Research Logistics (NRL), 56(3):239-249, 2009.
[103] PTV AG. PTV VISSIM 10 user manual. PTV AG: Karlsruhe, Germany, 2018.
[104] Xuewei Qi, Yadan Luo, Guoyuan Wu, Kanok Boriboonsomsin, and Matthew J. Barth. Deep reinforcement learning-based vehicle energy efficiency autonomous learning system. In 2017 IEEE Intelligent Vehicles Symposium (IV), pages 1228-1233. IEEE, 2017.
[105] John Rust. Using randomization to break the curse of dimensionality. Econometrica: Journal of the Econometric Society, pages 487-516, 1997.
[106] SAE on-road automated vehicle standards committee. Surface vehicle recommended practice J3016. SAE standard, rev. 2018, SAE, Warrendale, PA, 2018.
[107] Roman Schmied, Harald Waschl, Rien Quirynen, Moritz Diehl, and Luigi del Re. Nonlinear MPC for emission efficient cooperative adaptive cruise control. IFAC-PapersOnline, 48(23):160-165, 2015.
[108] David Schrank, Bill Eisele, and Tim Lomax. 2019 urban mobility report. Technical report, Texas A\&M Transportation Institute and INRIX, College Station, TX, August 2019. See also URL https://static.tti.tamu.edu/tti.tamu.edu/documents/mobility-report-2019.pdf.
[109] Antonio Sciarretta and Ardalan Vahidi. Energy-Efficient Driving of Road Vehicles. Springer, 2020.
[110] Junqing Shi, Fengxiang Qiao, Qing Li, Lei Yu, and Yongju Hu. Application and evaluation of the reinforcement learning approach to eco-driving at intersections under infrastructure-to-vehicle communications. Transportation Research Record, 2672(25):89-98, 2018.
[111] Joshua E. Siegel, Dylan C. Erb, and Sanjay E. Sarma. A survey of the connected vehicle land-scape-architectures, enabling technologies, applications, and development areas. IEEE Transactions on Intelligent Transportation Systems, 19(8):2391-2406, 2017.
[112] Santokh Singh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. NHTSA tech. rep. DOT HS 812 115, 2015.
[113] Andrzej Stateczny, Witold Kazimierski, Daria Gronska-Sledz, and Weronika Motyl. The empirical application of automotive 3d radar sensor for target detection for an autonomous surface vehicle's navigation. Remote Sensing, 11(10):1156, 2019.
[114] Raphael E. Stern, Shumo Cui, Maria Laura Delle Monache, Rahul Bhadani, Matt Bunting, Miles Churchill, Nathaniel Hamilton, Hannah Pohlmann, Fangyu Wu, Benedetto Piccoli, et al. Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. Transportation Research Part C: Emerging Technologies, 89:205-221, 2018.
[115] Jack Stewart. Tesla's autopilot was involved in another deadly car crash. On the WWW, March 2018. URL https://www.wired.com/story/tesla-autopilot-self-driving-crash-california/.
[116] Mark Stuhldreher, Charles Schenk, Jessica Brakora, David Hawkins, Andrew Moskalik, and Paul DeKraker. Downsized boosted engine benchmarking and results. In SAE 2015 World Congress \& Exhibition, 2015.
[117] Ardi Tampuu, Tambet Matiisen, Maksym Semikin, Dmytro Fishman, and Naveed Muhammad. A survey of end-to-end driving: Architectures and training methods. IEEE Transactions on Neural Networks and Learning Systems, 2020.
[118] The Tesla Team. A tragic loss, 2016. URL https://www.tesla.com/blog/tragic-loss.
[119] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al. Stanley: The robot that won the DARPA grand challenge. Journal of Field Robotics, 23(9):661-692, 2006.
[120] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. Physical Review E, 62(2):1805, 2000.
[121] Martin Treiber and Arne Kesting. Traffic Flow Dynamics. Springer-Verlag Berlin Heidelberg, 2013.
[122] IPG TruckMaker. Reference manual version 4.5. Simulation Solutions, Test Systems, Engineering Services, IPG Automotive, Karlsruhe, Germany, 2014.
[123] Valerio Turri, Bart Besselink, and Karl H. Johansson. Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning. IEEE Transactions on Control Systems Technology, 25(1):1228, 2016.
[124] Ardalan Vahidi. Boosting energy efficiency of heterogeneous connected and automated vehicle (CAV) fleets via anticipative and cooperative vehicle guidance. In Clemson University, 2019 US DOE Vehicle Technologies Office Annual Merit Review, 2019.
[125] Ardalan Vahidi and Antonio Sciarretta. Energy saving potentials of connected and automated vehicles. Transportation Research Part C: Emerging Technologies, 95:822-843, 2018.
[126] Jessica Van Brummelen, Marie O’Brien, Dominique Gruyer, and Homayoun Najjaran. Autonomous vehicle perception: The technology of today and tomorrow. Transportation Research Part C: Emerging Technologies, 89:384-406, 2018.
[127] Charles F. Van Loan. The ubiquitous Kronecker product. Journal of Computational and Applied Mathematics, 123(1-2):85-100, 2000.
[128] Aldo Vecchietti, Sangbum Lee, and Ignacio E. Grossmann. Modeling of discrete/continuous optimization problems: characterization and formulation of disjunctions and their relaxations. Computers \& Chemical Engineering, 27(3):433-448, 2003.
[129] Peter Vortisch. Where I can find the mathematical formulation of Wiedemann 99 car following model?, 2015. URL https://www.researchgate.net/post/Where_I_can_find _the_mathematical_formulation_of _Wiedemann_99_car_following_model.
[130] Zia Wadud, Don MacKenzie, and Paul Leiby. Help or hindrance? the travel, energy and carbon impacts of highly automated vehicles. Transportation Research Part A: Policy and Practice, 86:1-18, 2016.
[131] Nianfeng Wan, Chen Zhang, and Ardalan Vahidi. Probabilistic anticipation and control in autonomous car following. IEEE Transactions on Control Systems Technology, 27(1):30-38, 2017.
[132] Pin Wang and Ching-Yao Chan. Formulation of deep reinforcement learning architecture toward autonomous driving for on-ramp merge. In 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pages 1-6. IEEE, 2017.
[133] Thomas Weiskircher, Qian Wang, and Beshah Ayalew. Predictive guidance and control framework for (semi-) autonomous vehicles in public traffic. IEEE Transactions on Control Systems Technology, 25(6):2034-2046, 2017.
[134] Moritz Werling, Julius Ziegler, Sören Kammel, and Sebastian Thrun. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In 2010 IEEE International Conference on Robotics and Automation, pages 987-993. IEEE, 2010.
[135] Rainer Wiedemann. Simulation des strassenverkehrsflusses. Institut fur Verkehrswesen der Universitat Karlsruhe, 1974.
[136] World Forum for Harmonization of Vehicle Regulations. Proposal for amendments to global technical regulation no. 15 on worldwide harmonized light vehicles test procedure. Technical Report ECE/TRANS/WP.29/GRPE/2016/3, United Nations, Geneva, November 2015.
[137] Ekim Yurtsever, Linda Capito, Keith Redmill, and Umit Ozguner. Integrating deep reinforcement learning with model-based path planners for automated driving. In 2020 IEEE Intelligent Vehicles Symposium (IV), pages 1311-1316. IEEE, 2002.
[138] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In European Conference on Computer Vision, pages 818-833. Springer, 2014.
[139] Chen Zhang and Ardalan Vahidi. Predictive cruise control with probabilistic constraints for eco driving. In Dynamic Systems and Control Conference, pages 233-238. ASME, 2011.
[140] Linjun Zhang and Gábor Orosz. Motif-based design for connected vehicle systems in presence of heterogeneous connectivity structures and time delays. IEEE Transactions on Intelligent Transportation Systems, 17(6):1638-1651, 2016.
[141] Yang Zhou, Meng Wang, and Soyoung Ahn. Distributed model predictive control approach for cooperative car-following with guaranteed local string stability. Transportation Research Part B: Methodological, 128:69-86, 2019.


[^0]:    ${ }^{1}$ R. Austin Dollar and Ardalan Vahidi. Quantifying the impact of limited information and control robustness on connected automated platoons. In 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pages 1-7, IEEE, 2017.
    ${ }^{2}$ R. Austin Dollar and Ardalan Vahidi. Efficient and collision-free anticipative cruise control in randomly mixed strings. IEEE Transactions on Intelligent Vehicles, 3(4):439-452, 2018.
    ${ }^{3}$ Tyler Ard, R. Austin Dollar, Ardalan Vahidi, Yaozhong Zhang, and Dominik Karbowski. Microsimulation of energy and flow effects from optimal automated driving in mixed traffic. Transportation Research Part C: Emerging Technologies, 120:102806, 2020.
    ${ }^{4}$ Tyler Ard, Longxiang Guo, R. Austin Dollar, Alireza Fayazi, Nathan Goulet, Yunyi Jia, Beshah Ayalew, and Ardalan Vahidi. Energy and flow effects of optimal automated driving in mixed traffic: Vehicle-in-the-loop experimental results. arXiv preprint arXiv:2009.07872, 2020.

[^1]:    ${ }^{5}$ A less conservative alternative to the worst-case constraint is presented in Section 4.5.

[^2]:    ${ }^{1}$ R. Austin Dollar, Tamás G. Molnár, Ardalan Vahidi, and Gábor Orosz. MPC-based connected cruise control with multiple human predecessors. In 2021 American Control Conference (ACC). IEEE, 2021. To appear.

[^3]:    ${ }^{1}$ R. Austin Dollar and Ardalan Vahidi. Predictively coordinated vehicle acceleration and lane selection using mixed integer programming. ASME 2018 Dynamic Systems and Control Conference. American Society of Mechanical Engineers Digital Collection, 2018.
    ${ }^{2}$ R. Austin Dollar and Ardalan Vahidi. Automated vehicles in hazardous merging traffic: A chance-constrained approach. IFACPapersOnLine, 52(5):218-223, 2019.
    ${ }^{3}$ R. Austin Dollar and Ardalan Vahidi. Multilane automated driving with optimal control and mixed-integer programming. IEEE Transactions on Control Systems Technology, 2021

[^4]:    ${ }^{1}$ This reasoning focuses on the case where the maximum final time constraint is active. If it is inactive, then the elapsed time estimate has no effect on the solution because $t(i)$ only appears in the final time constraint.

[^5]:    ${ }^{2}$ These percentages based on input energy differ from the loss-based annotations in Fig. 5.14 because the change in potential energy is nonzero.

[^6]:    ${ }^{1}$ R. Austin Dollar, Antonio Sciarretta, and Ardalan Vahidi. Information and collaboration levels in vehiclar strings: A comparative study. In 21st IFAC World Congress. International Federation of Automatic Control (IFAC), 2020. To appear.
    ${ }^{2}$ R. Austin Dollar, Antonio Sciarretta, and Ardalan Vahidi. Multi-agent control of lane-switching automated vehicles for energy efficiency. In 2020 American Control Conference (ACC), pages 422-429. IEEE, 2020.

