VEHICLE VELOCITY TRAJECTORY OPTIMIZATION IN NON-DETERMINISTIC ENVIRONMENTS

by

John Borek

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Mechanical Engineering

Charlotte

2021

Approved by:

Dr. Christopher Vermillion

Dr. Scott Kelly

Dr. Srinivas Pulugurtha

Dr. Amirhossein Ghasemi

Dr. Jim Conrad

ABSTRACT

JOHN BOREK. VEHICLE VELOCITY TRAJECTORY OPTIMIZATION IN
NON-DETERMINISTIC ENVIRONMENTS. (Under the direction of DR. CHRISTOPHER
VERMILLION)

Optimally modulating a vehicle's speed profile through highway, urban, and suburban environments can result in pronounced fuel savings, particularly with heavy-duty vehicles. However, existing strategies for speed profile optimization traditionally rely on aspirational and often deterministic assumptions, which cease to be accurate in the presence of real-world features such as non-deterministic traffic and actuated signalized intersections. The research in this dissertation establishes a hierarchical Green-Light Approach Speed (h-GLAS) strategy for controlling vehicles traveling through non-deterministic highway, urban, and suburban environments. The h-GLAS strategy utilizes vehicle-to-infrastructure (V2I) communication to receive information about the route's topology, speed limits, and signal phase and timing (SPaT). For suburban environments that employ semi- and fully-actuated signalized intersections, past signal timing information is used to forecast the future values of the signal phase lengths. This information is used to construct a desired velocity profile to be tracked by a semi-economic model predictive controller (MPC), which computes the optimal wheel force command for the vehicle. When traveling through highway environments, the desired velocity profile represents a globally optimal dynamic program solution, computed offline before the beginning of a trip. For urban and suburban environments, however, the velocity profile is constructed to allow the vehicle to arrive at upcoming intersections when the probability of a green signal is maximized. The semi-economic MPC minimizes a quadratic objective function, which reflects a trade-off between minimizing mechanical energy expenditure and braking effort, with tracking the supplied desired velocity profile. This MPC is unable to maintain vehicle following constraints without changing its convex nature, so a command governor (CG) is located downstream to maintain vehicle-following constraints efficiently. The CG modifies the MPC's control action by the minimal amount necessary to maintain safe vehicle following. Using the simulation packages from PTV VISSIM, the h-GLAS is validated against real signal timing algorithms within a stochastic traffic environment parameterized by real-world data. Simulation results show that the h-GLAS controller is capable of significantly reducing a vehicle's fuel consumption by 16%-26% when compared to a baseline control strategy traveling through the same suburban environment.

ACKNOWLEDGMENTS

Over the past five years, I have had the great privilege to study and learn under my advisor, Dr. Chris Vermillion. He has provided significant guidance and support throughout my final year as an undergraduate student and throughout my studies and work as a PhD student. Thank you for giving me the opportunity as an inexperienced undergraduate to conduct meaningful research in the area of optimal vehicle control. I would also like to thank my committee members, Dr. Scott Kelly, Dr. Srinivas Pulugurtha, and Dr. Amirhossein Ghasemi for their support and guidance throughout this process.

I would also like to thank the City of Concord, NC's DOT for providing me the necessary information to simulate signalized intersections and traffic within VISSIM and PreScan. Specifically Victoria Forrester for her assistance in the proper modeling of signal controllers.

Lastly, I would like to extend my thanks to Volvo Group Trucks in Hagerstown, Md for providing the medium-fidelity Simulink model used for simulations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Autonomous vehicle technology has improved significantly over the past decade, with most of the major automotive manufacturers actively exploring autonomous vehicle programs [2]. A major motivation for the push to develop autonomous vehicle technology is the potential for improved fuel economy, reduced emissions, and increased safety. A study by the National Renewable Energy Laboratory on behalf of Volvo showed a 5% to 7% decrease in fuel consumption in cars driving with adaptive cruise control (ACC) as compared with human drivers [3]. Here, ACC, one of the simplest forms of autonomous vehicle technology beyond basic cruise control, uses a (typically) closed-form control law to maintain a prescribed minimum following distance relative to a lead vehicle. This raises an important question: if such a simple technology can achieve such improvements, how much *additional* improvement can be achieved through more sophisticated, optimization-based technologies?

In evaluating the possibilities of optimization-based control methodologies on fuel economy, the work within this dissertation focuses on a heavy-duty truck application. Commercial trucks account for 13.4% of all registered vehicles in the U.S., with a total of 3.91 million being class-8 trucks [4]. In 2019, these vehicles traveled over 300 billion miles, and with fuel costs accounting for 30% of the truck's lifetime cost [4], the trucking industry can significantly benefit from these technology advancements. Furthermore, since heavy-duty vehicles are much larger than passenger vehicles, they hold a greater potential for improving fuel economy. In particular, the kinetic energy of a fully-loaded heavy-duty truck, traveling at 35 mph, is approximately equal to the amount of work required to travel just one mile on flat terrain, due to the larger mass. Therefore, by maintaining its kinetic energy (by avoiding braking maneuvers), a heavy-duty vehicle has the potential to save a significant amount of fuel. This research evaluates control algorithms on a class-8 heavy-duty truck simulation model of an actual Volvo truck and trailer driving on real-world routes.

1.1 Background - Velocity trajectory optimization

In general, optimal vehicle control strategies can be categorized as either highway–based or city–based. Highway–based control strategies tend to be simpler, as there are few traffic control devices. In particular, the optimal control strategy involves careful consideration of the terrain, along with appropriate consideration of traffic. City-based control strategies have all the same features as the highway–based strategies (including for terrain and traffic), but they also contain traffic control devices like signalized intersections, pedestrian crosswalks, and railway crossings. A large subset of autonomous vehicle technology, focused on improving fuel economy, is velocity trajectory optimization (VTO) [5–35]. Other technologies include autonomous steering control [36–41], optimal powertrain control [42–44], and optimal thermal system and engine management [45, 46]. The work presented in this dissertation focuses on longitudinal vehicle control, which falls into the VTO category.

The goal of VTO is to determine the optimal velocity profile for the vehicle to travel at, and the corresponding control command (or vice-versa), based on some performance metric (i.e. minimizing fuel consumption or trip time) and a dynamic model of the vehicle. While such an optimization can be performed offline [5–7, 11], disturbances such as traffic and signalized intersections necessitate an online feedback mechanism to ensure robustness [8–10, 12–35]. It is imperative that this online feedback mechanism supports the goal of fuel efficiency, while ensuring robust constraint satisfaction when it comes to maintaining vehicle following and obeying signalized intersections. Consequently, online constrained optimal control approaches represent attractive solutions for addressing this goal.

The most popular constrained optimal control technique for VTO is model predictive control (MPC) [21–35]. Because MPC typically involves the optimization of a finite-dimensional control sequence over a receding horizon, the use of MPC for VTO involves discretization of the route in either the time or position domain. At each time or position step, MPC repeatedly finds the optimal control sequence for the vehicle via minimization (or maximization) of an objective (cost) function over a finite horizon length (time or position). The first decision variable in the control sequence is then implemented at each step, and the optimization is repeated at the subsequent step (multi-step implementations exist, but are less common). MPC re-optimizes an N-step control sequence at each step, making it capable of reacting to external disturbances and mismatches between the dynamic model used for optimization and the actual system it is controlling. MPC-based optimal control strategies for VTO are able to produce optimal velocity profiles accounting for road topology, vehicle state and control constraints, and environmental disturbances such as traffic and signalized intersections.

1.1.1 Highway Velocity Trajectory Optimization

From a velocity trajectory optimization standpoint, the highway environment is the simplest of driving environments. Variation in the terrain and speed limits allows predictive control strategies to optimize the vehicle's velocity trajectory subject to safe vehicle following [5–10, 30–35, 47]. Prior knowledge of the upcoming terrain allows the vehicle to slow down along an uphill such that no braking will be required along the downhill. This saves the extra energy that would have been required to maintain its speed over the entire uphill. Additionally, prior knowledge of changing speed limits can also be beneficial in regard to fuel economy. If the vehicle controller knows that the speed limit is going to decrease in the near future, it can make use of natural deceleration forces (aerodynamic, gravitational, and rolling resistances) to slow down without braking, before reaching the speed change. Otherwise, the vehicle will maintain its velocity until being required to brake for the speed change, which is typically observed with human driving (this work requires all speed limits be obeyed at all times by all control strategies). Lastly, by equipping vehicles with radar/lidar systems, inter-vehicular distances can be measured, allowing the MPCs to maintain vehicle-following constraints while optimizing the vehicle's velocity trajectory.

Because heavy-duty trucks typically follow pre-defined routes, an offline optimal control solution can be useful. The benefit of an offline optimal solution, is that it can be a global optimal solution as computation time becomes irrelevant. Therefore, techniques like exhaustive searches and dynamic program (DP) optimizations are commonly used to obtain globally optimal solutions for the vehicle's velocity trajectory [5–7]. The main drawback with offline control, however, is the inability to react to unforeseen disturbances and mismatches between the dynamic model referenced and the actual system dynamics. Therefore, coupling an offline globally optimal solution with an online controller can be very advantageous [34, 35]. Shown in Figure 1, the h-GLAS strategy presented in this research downloads an offline DP solution, through vehicle-to-infrastructure (V2I) communication, to act as a supervisory reference for an online MPC. The online MPC refines/adjusts the DP solution according to real-time inputs like surrounding traffic to ultimately drive the vehicle.

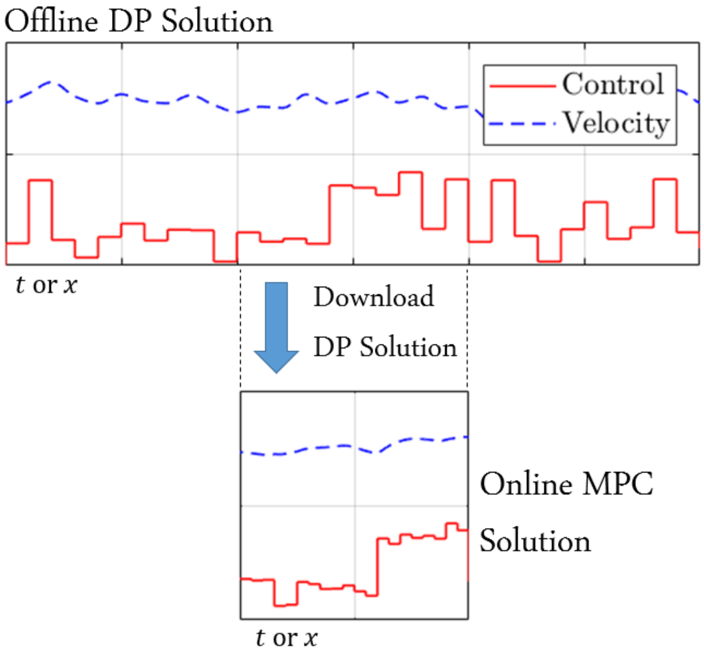


Figure 1: Offline DP + online MPC refinement.

1.1.2 Urban/Suburban Velocity Trajectory Optimization

Unlike highway environments, urban and suburban environments contain signalized intersections that need to be accounted for. When the environment contains signalized intersections, the optimal control problem is often referred to as optimal arrival/departure at intersections. The main objective of optimal arrival/departure control is to determine the optimal vehicle trajectory through a sequence of signalized intersections and surrounding traffic that satisfies some performance objective. With vehicle-to-infrastructure (V2I) communication, the ego vehicle (vehicle being controlled) can receive signal phase and timing (SPaT) and basic traffic information for upcoming intersections, which can be used by the optimal controller [12–29]. The received SPaT and traffic information can include (but is not limited to):

- Intersection locations;

- The current signal phase and cycle clock time (how far into the current phase);
- Programmed signal phase and cycle lengths under *unactuated* operation;
- Past signal phase lengths and start times under *actuated* operation;
- Average traffic flow, density, and speed;

Signalized intersections fall into two main types of operation: fixed or pre-timed, and semi- or fully-actuated operation. With the former, the signal phase lengths for each direction of travel are constant, and therefore, future SPaT information is deterministic. In the presence of deterministic SPaT information and an absence of non-deterministic behavior from other vehicles, a vehicle can readily use that information (available through V2I communication) in a deterministic online optimization in order to maintain its kinetic energy through an urban corridor [12–15, 21–25]. However, this type of intersection signal control is typically only used in tight urban areas and central business districts with closely spaced intersections [1]. Outside of these areas, signalized intersections operate under semi- or fully-actuated signal control. In this dissertation, urban environments are assumed to use fixed-time signal operation, and suburban environments are assumed to use actuated signal operation. Traditionally with signalized intersections, there is a designated major approach and minor approach (i.e. East/West being major and North/South being a minor approach). For a signalized intersection to be considered actuated, it requires detectors, either in the road or implemented via cameras, to identify vehicles approaching the intersection and to make decisions on how to actuate the signal phases accordingly. Intersections with detection only occurring on a subset of all approaches are typically semi-actuated, whereas if detection can occur on all approaches, the intersections can be fully-actuated. However, some fully-actuated signalized intersections will remain semi-actuated when coordinated with other intersections during peak traffic hours [48]. In summary, signal actuation results in signal phases that vary from one cycle to the next. With the ability to adapt to traffic flow, future SPaT information for semi- or fully-actuated intersections is non-deterministic. Even pre-timed signals that operate off a fixed timing plan are subject to some uncertainty, as their signal clocks can drift over a 24-hour period [17]. Figure 2 shows the green and red signal phase lengths for a specific lane of travel at an intersection in Concord, NC. Both the green and red signal phases are independent and vary from cycle to cycle. In the literature, prior work by other researchers [16, 17, 19, 20, 26–28] and myself [29] have developed optimal arrival/departure controllers that navigate through signalized intersections with partially and fully uncertain SPaT information.

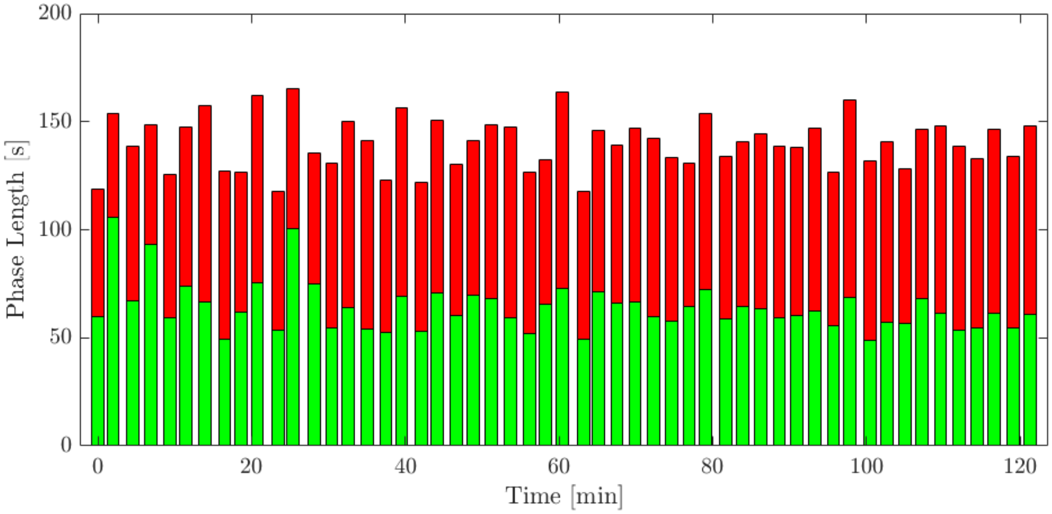


Figure 2: Example green and red signal phase lengths over a two-hour period for a specific lane of travel.

The authors in [16, 17] consider uncertainty within the SPaT information by treating the phase lengths as varying entities, while keeping the total cycle length fixed from cycle to cycle. The authors then assume that the average phase lengths are known through historic data and that the current phase color is received through V2I communication. Based on the current phase color, average phase lengths, and assumed cyclic behavior (fixed cycle lengths), a closed-form expression for the conditional probability of the signal being green can be obtained. This probability is then implemented as a barrier term within a dynamic programming (DP) formulation's objective function. Therefore, as the probability of a green signal approaches 0, the value of the DP's objective function goes to infinity and the optimization avoids those scenarios. While accounting for uncertainty in the future signal timing, this approach has two significant limitations: 1) the signal cycle length is assumed to be fixed, which is untrue for actuated traffic signals; and 2) the optimal control strategy relies on a DP optimization, which can be problematic for real-time implementation.

The authors in [26–28] also consider the SPaT information to have some uncertainty and model each signalized intersection as having a constant cycle length with a nominal red-phase duration plus a stochastic time delay, referred to as the *effective red-light duration* (ERD). The ERD describes the stochastic feasible passing time of vehicles at upcoming signalized intersections. The delay term is there to account for signal variation and vehicle congestion. The time delay is assumed to be distributed over the green-phase duration, where the distribution function is obtained via real data. The authors developed a robust chance-constrained approach where chance constraints within a cloud-based dynamic programming (DP) optimization are used to generate an optimal velocity profile. The resulting velocity profile allows the vehicle to pass through an upcoming intersection only when the probability of a green signal phase is greater than a user-defined threshold. This velocity profile is then tracked by a lower-level MPC to minimize control effort while enforcing vehicle following (safety) constraints. While addressing the uncertainty in future signal phase lengths, the same two limitations from [16, 17] remain: 1) the signal cycle length is assumed to be fixed, which is not the case for actuated traffic signals; and 2) the optimal velocity profile is found through a DP formulation, which can be computationally intensive and limiting for real-time implementation.

The only literature that treats signalized intersections as truly actuated (outside of my work in [29]) are [19] and [20]. The algorithm developed in [19] uses the programmed minimum/maximum green values to calculate minimum/maximum times-to-change for the signal. The vehicle approaching the intersection is provided a trajectory to follow based off the times-to-change, which becomes more reliable as the vehicle gets closer to the intersection. However, without any prediction of future timing, the work in [19] relies on vehicle actuation before updating the times-to-change, which limits the algorithm's application and reduces fuel savings. The authors in [20] focused on semi-actuated signalized intersections, where only the minor approach lanes are actuated. They relied on a detection system that consists of two detectors approaching the intersection. Using known geometry of the intersection and measured vehicle speeds over the detectors, the *time-to-green* for the major approaches is approximated. For cases where the ego vehicle would arrive at the intersection while it is green (under its current speed) or there is room for acceleration in order to meet a green signal, the algorithm developed in [20] advises the ego vehicle to do just that. For the other case, where the ego vehicle would have to slow down in order to meet a green signal, an exhaustive search optimization is performed off-line to develop a lookup library. The decision variable in the exhaustive search is the deceleration of the ego vehicle on the approach and acceleration on the departure. The lookup library is based on the ego vehicle's velocity, distance to the intersection, vehicle queue length, and the time-to-green value at each timestep. Simulation results using the traffic simulator PTV-VISSIM demonstrated 29% fuel consumption reduction for vehicles under this control algorithm. One major limitation in [20] is the requirement of two detectors on the intersection approach. This would require an infrastructure overhaul, which limits where this control strategy can be implemented. Lastly, the simulation study in [20] only evaluated a single intersection, and only the case where without control, the vehicles were timed to arrive at the intersection while the signal was red.

1.2 Problem statement and intended contributions

The current research gap, within the existing body of literature, is the lack of a comprehensive longitudinal vehicle controller that can be applied to realistic, non-deterministic highway and suburban environments. The research in this dissertation will present a hierarchical Green-Light Approach Speed (h-GLAS) strategy capable of safely and efficiently navigating through non-deterministic highway, urban, and suburban environments, closing the aforementioned research gap. The h-GLAS strategy will utilize V2I communication to receive information on the route's topology, speed limits, and signal timing, which is illustrated in Figure 3. A SPaT forecast model utilizes past signal phase lengths, received through V2I communication, to forecast future signal phase lengths with some uncertainty. This feature of the h-GLAS controller is bypassed for fixed-time signal operation in urban environments. A desired velocity profile is then constructed,

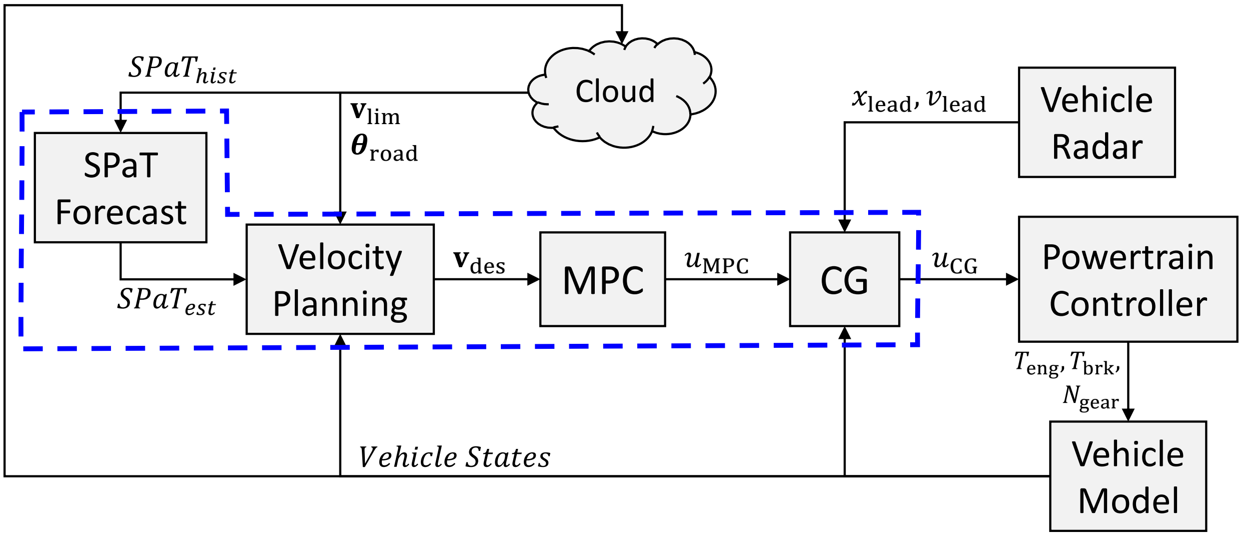


Figure 3: The h-GLAS strategy (inside the blue dashed box) receives historical SPaT and route information through V2I communication, the current lead vehicle states from an on-board radar, and the current vehicle states to determine the optimal wheel force command u_{CG}.

based on the current environment (highway/urban/suburban), to be referenced by a model predictive controller. Lastly, a command governor is implemented downstream from the MPC to enforce vehicle following constraints for safety. This control formulation will be fully detailed throughout Chapters 3 and 4.

The contributions of this research to the field of longitudinal vehicle control are as follows:

1. A comprehensive hierarchical longitudinal vehicle controller developed for traversing non-deterministic highway, urban, and suburban environments;
2. A signal timing forecast model that utilizes past SPaT information to predict future signal behavior;
3. A velocity planning algorithm that considers both fixed-time and actuated signal operation;
4. An MPC strategy that trades off minimization of energy expenditure with tight tracking of a desired kinetic energy profile.

This dissertation is organized as follows. Chapter 2 details the dynamic models used by the proposed control formulation and the various environmental models used for simulation. Chapter 3 details the highway and urban controllers that serve as a foundation to the suburban controller, including the dynamic program optimization, desired velocity profile construction, and the model predictive control formulation. In Chapter 4, the adaptations for a suburban environment and corresponding simulation results are described, including the SPaT forecast, desired velocity profile construction considering uncertainty, and the command governor. Finally, areas where this work can be extended will be discussed in Chapter 5.

CHAPTER 2: MODELING

The controllers developed within this research have all been validated through simulation, thus requiring sufficiently accurate models to capture the behavior of the vehicle and its environment. Simulation of the vehicle dynamics and corresponding fuel consumption were performed on a proprietary medium-fidelity Simulink model of a class-8 heavy-duty truck, provided by Volvo Group Trucks. This Simulink model captures the dynamics of the majority of systems present on a heavy-duty diesel truck, including the diesel engine, mechanical and electrical auxiliary systems, the alternator, clutch, transmission, driveline, wheel assemblies, chassis dynamics, the battery, and the junction box. With a total of 16 states being modeled, this dynamic model is much too complex to use for the design of model-based controllers that rely on online optimizations. Therefore, two simplified control-oriented vehicle models are considered, only capturing the longitudinal dynamics of the vehicle, which is all that is required for the model-based controllers developed in this work. Lastly, the environments require modeling, including surrounding traffic and signalized intersections. The development of vehicle-following controllers require the simulation of lead vehicles, so a *stochastic* vehicle model was developed, using collected on-road radar data, to model lead vehicles. This is sufficient for emulating traffic in a highway and urban environment, but for the suburban environment, a more sophisticated traffic model is required, as traffic and actuated signalized intersections are dependant on one another. Therefore, the software VISSIM was used in producing the results of this dissertation for suburban environments, as VISSIM can properly model actuated signalized intersections.

2.1 Vehicle Models

This section will detail the vehicle models used in the development and validation of the controllers presented in later chapters. This includes the two control-oriented dynamic models:

1. A second-order nonlinear time-based longitudinal vehicle dynamic model, where the states are position and velocity and the control input is tire force on the road over a single time step;
2. A first-order affine distance-based longitudinal vehicle energy dynamic model, where the single state is the vehicle's kinetic energy and the control inputs are traction and braking work over a single distance step.

2.1.1 Time-based, Nonlinear, Control-Oriented Dynamic Model

Any *time-based* predictive control or constraint enforcement algorithm for a vehicle will need to work with a time-domain dynamic model to predict the future states of the vehicle, which is the case for the dynamic program optimization and model predictive controllers, used for highway driving, and the command governor, used for vehicle following. The longitudinal continuous-time dynamic equations of motion are expressed as

$$\dot{x} = v, \tag{1}$$

$$\dot{v} = \frac{1}{M_{\text{eff}}}\big[u - F_{\text{drag}} - F_{\text{grav}} - F_{\text{road}}\big], \tag{2}$$

where x is the truck's longitudinal position, v is its velocity, and u is the propulsive force applied to the road through the wheel. The term M_{eff} is the effective mass of the vehicle, accounting for wheel/axle inertia and reflected engine inertia. The variables F_{drag}, F_{grav}, and F_{road} represent the forces due to aerodynamic drag, gravity, and rolling resistance, respectively, where

$$F_{\text{drag}} = \frac{1}{2}\rho C_d A_{\text{ref}} v^2, \tag{3}$$

$$F_{\text{grav}} = mg\sin\theta(x), \tag{4}$$

$$F_{\text{road}} = C_{\text{rr}} mg\cos\theta(x). \tag{5}$$

The parameters ρ, C_d, A, m, g, and C_{rr} refer to the air density, drag coefficient, reference area, vehicle mass, gravitational acceleration, and rolling resistance coefficient, respectively. The variable $\theta(x)$ represents the road grade, which depends on each position along the route. The velocity state, v, and control variable, u (the applied propulsive force), are related to the wheel speed (ω_{wh}) and torque (T_{wh}) through the wheel radius (r_{wh}):

$$\omega_{\text{wh}} = v/r_{\text{wh}}, \qquad T_{\text{wh}} = r_{\text{wh}} u. \tag{6}$$

The DP and CG formulations are discrete, therefore requiring Equations (1) and (2) to be discretized in time. A trapezoidal approximation for position and a forward Euler approximation for velocity are used to accomplish this. These approximations are reasonable, as the time steps used by the DP and CG are significantly smaller than the time constants associated with the longitudinal dynamics of the vehicle. The discrete-time dynamic equations are expressed as

$$x(i+1) = x(i) + \frac{v(i+1)+v(i)}{2}\Delta t, \tag{7}$$

$$v(i+1) = v(i) + \frac{\Delta t}{M_{\text{eff}}}\big[u(i) - F_{\text{drag}}(i) - F_{\text{grav}}(i) - F_{\text{road}}(i)\big], \tag{8}$$

where i and Δt represent the index corresponding to the current time step and its length, respectively. To avoid confusion, the variables i and k will represent time and distance steps, respectively.

The simulation model provided by Volvo contains a brake-specific fuel consumption (BSFC) map for calculating fuel consumption. The BSFC map relates engine power (P_{eng}) to the fuel rate (R_{fuel}) through

$$BSFC = \frac{R_{\text{fuel}}(i)}{P_{\text{eng}}(i)} = \frac{R_{\text{fuel}}(i)}{\omega_{\text{eng}}(i) T_{\text{eng}}(i)}, \tag{9}$$

where P_{eng} is the product of engine speed (ω_{eng}) and torque (T_{eng}). In order to calculate BSFC, both u and v need to be related to T_{eng} and ω_{eng}, respectively. To accomplish this task, a static shift map is implemented, which selects the gear ratio (τ_g) that minimizes BSFC for every combination of ω_{wh} and T_{wh}:

$$BSFC(\omega_{\text{wh}}(i), T_{\text{wh}}(i)) = \min_{\tau_g} \; BSFC(\omega_{\text{eng}}(i), T_{\text{eng}}(i)), \tag{10}$$

where ω_{eng} and T_{eng} are related to ω_{wh} and T_{wh} through

$$\omega_{\text{eng}}(i) = \omega_{\text{wh}}(i)\tau_g(i)\tau_f, \qquad\qquad T_{\text{eng}}(i) = \frac{T_{\text{wh}}(i)}{\tau_g(i)\tau_f\eta_{\text{dl}}}. \tag{11}$$

Here, τ_f and η_{dl} represent the final drive ratio and driveline efficiency, respectively.

2.1.2 Distance-based, Affine, Control-Oriented Dynamic Model

One of the model predictive control strategies in this work requires the dynamic vehicle model to be linear or affine in order for the optimal control problem to be convex. A linearization of the previous model could work. However, that would only provide a sufficiently accurate model over a limited domain of the states. To achieve a model that is suitable for a convex optimization formulation, the longitudinal dynamics of a heavy-duty truck are considered in the terms of work and energy, where discretization is done with respect to position rather than time. This provides an affine, distance-based model that ultimately leads to a convex MPC formulation. This energy-based formulation has been adopted from [47]; it provides the benefits of simplicity and linearity (with the exception of constant terms that make the dynamics affine), at the expense of not explicitly modeling fuel consumption. This model treats the kinetic energy of the vehicle as the single state variable, $\mathbf{x} = [E_\mathrm{k}]^T$, which allows the vehicle's aerodynamic energy loss to appear linearly (as a function of E_k) in the state equation. The control variables for these dynamics are the traction and braking work done by the wheels, $\mathbf{u} = [E_\mathrm{t} \;\; E_\mathrm{b}]^T$, which are related to the traction and braking forces applied to the road through

$$F_\mathrm{wh,pos} = E_\mathrm{t}/\Delta x, \qquad F_\mathrm{wh,neg} = E_\mathrm{b}/\Delta x, \tag{12}$$

where Δx represents the distance step. Both $F_\mathrm{wh,pos}$ and $F_\mathrm{wh,neg}$ are related to the control input u from the nonlinear time-domain model through

$$u = F_\mathrm{wh,pos} - F_\mathrm{wh,neg}. \tag{13}$$

The discrete-position dynamic equation for this model, where each index (k and $k+1$ below) refers to a *position* along the route and not *time*, is given by

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k) + F(k), \tag{14}$$

where A, B, and F represents the state matrix, input matrix, and the affine environmental resistance, respectively. These matrices are calculated as

$$A = 1 - \frac{1}{m}\rho C_d A_\mathrm{ref}, \qquad B = [1 \;\; -1], \tag{15}$$

where the second term in A contains the aerodynamic terms, which when multiplied by $\mathbf{x}$ becomes the energy lost through aerodynamic drag. Lastly, the environmental resistance F reflects the amount of mechanical work that must be done in order to traverse the distance step of length Δx, at step k. This term is comprised of gravitational (potential energy) and rolling resistance energy losses:

$$F(k) = F_{\text{grav}}(k)\Delta x + F_{\text{road}}(k)\Delta x. \tag{16}$$

It is important to note that because each position along a discretized distance horizon is fixed, the forces $F_{\text{grav}}(k)$ and $F_{\text{road}}(k)$ are also fixed for every position step k along the prediction horizon. Therefore, the overall dynamics are affine.

2.2 Environment Models

Three different environments are considered and emulated within this research, namely highway, urban, and suburban environments. For highway environments, the only non-deterministic feature that requires modeling is traffic behavior. Therefore, collected on-road radar data, which provides sufficient information regarding lead vehicle behavior, is used to develop a *stochastic* lead vehicle model as a traffic model. Urban and suburban environments, on the other hand, contain signalized intersections, which also need to be modeled. In this work, the difference between urban and suburban environments is simply defined by the type of signal controller being used for the signalized intersections. Specifically, urban environments are assumed to use fixed-time signal operation and suburban environments are assumed to use actuated signal operation. The urban environment simulations can use the same stochastic lead vehicle model as the highway environment, but the suburban environment needs its own solution. For suburban environments, the software VISSIM is considered for this purpose, as it can model actuated signalized intersections as well as traffic. This is important, as traffic behavior dictates the behavior of actuated signalized intersections.

2.2.1 Stochastic Lead Vehicle Model

Traffic modeling for highway and urban environments is done using a stochastic lead vehicle model to define the lead vehicle's behavior. Calibrated using actual traffic data from an on-board radar, this model characterizes the lead vehicle's entry point (x_{enter}), its duration on the road (t_f), initial vehicle spacing (δ_0), and the velocity profile that the lead vehicles follow $(v_{\text{lead}}(1,\dots,N))$, between X_0 and X_f. These quantities are formulated as:

$$x_{\text{enter}} = r_{\text{enter}}, \qquad t_f = r_{\text{exit}}, \qquad \delta_0 = r_{\text{gap}}, \qquad v_{\text{lead}}(i) = r_{\text{vel}}(i),\ \forall\, i = 1,\dots,N. \tag{17}$$

where r_{enter}, r_{exit}, r_{gap}, and r_{vel} represent Pearson random variables. Using collected radar data, values for each variable's mean, standard deviation, skewness, and kurtosis were found and are shown in Table 1. The value of x_{enter} encodes how much distance lies between the previous vehicle's exit and the next vehicle's entry. Note that the mean value for velocity needs to be adjusted to account for the current speed limit along the route. The value of v_{mean} is set to one standard deviation below the speed limit.

Table 1: Stochastic parameters used for modeling traffic.

Traffic Parameters				
Variable	**Mean**	**Std**	**Skew**	**Kurt**
r_{enter} **[m]**	6000	8975	2.62	9.63
r_{exit} **[s]**	94	64	2.47	9.4
r_{gap} **[m]**	63	35	-0.04	1.43
r_{vel} **[m/s]**	v_{mean}	1.88	-0.72	2.71

The values of velocity profile, $v_{\text{lead}}(1,\dots,N)$, provided by the random number generator will be temporally uncorrelated, but there will most certainly be a significant correlation between closely (temporally) spaced velocities of the lead vehicle, making temporal correlation necessary. We have addressed this issue by performing a fast Fourier transform (FFT) analysis on the collected velocity data to identify the critical frequencies in the velocity signal to select a cutoff frequency (and corresponding filter time constant) for a low-pass filter that introduces this temporal correlation. Figure 4 overlays the frequency spectrum of the collected velocity data with that of the generated velocity profile from the stochastic model. Lastly, to adjust from a highway to urban setting, the value for the space between lead vehicle exit-entry is reduced. This increases the total duration over which a lead vehicle is present throughout the route.

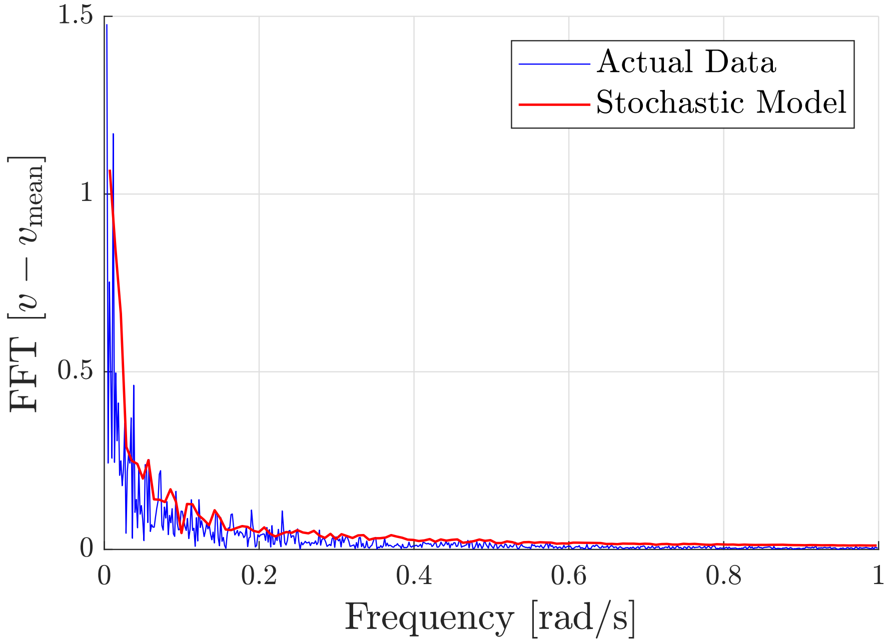


Figure 4: Frequency spectrum of collected traffic data and stochastic velocity profiles.

2.2.2 Signalized Intersection Models

With a core contribution of this work involving the control of vehicles through sequences of signalized intersections, modeling of the signal controllers is required. Prior work in vehicle control through urban environments assumes fixed-time signal operation, where suburban environments assume actuated signal operation, which are both detailed in the following subsections. In this work, the traffic signal cycle is defined to begin when the signal phase changes from red to green and to end after the subsequent red signal phase ends.

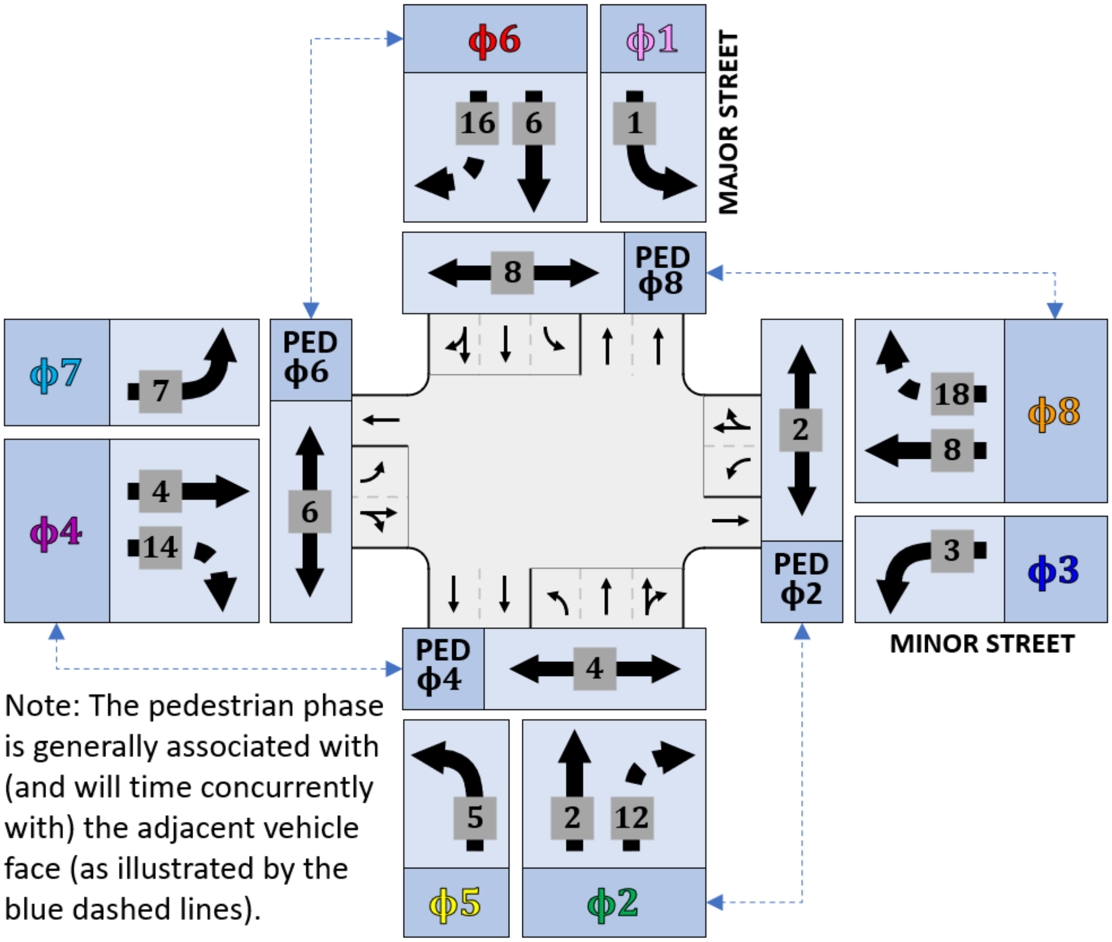
2.2.2.1 Fixed-time Signal Operation

Fixed-time signal controllers use pre-programmed signal phase lengths to time signal displays, which do not change unless programmed to do so at different times throughout the day (rush hour begins/ends). This type of signal operation is most commonly found in tight urban districts where the traffic is so consistent that actuating the signals has no positive effect on throughput. Therefore, simulations involving fixed-time signal operation only required the intersection locations (x_{L}^{m}), cycle lengths (T_{L}^{m}), and green signal phase lengths (ϕ_{g}^{m}) be defined for each simulation (the superscript m represents the intersection number along a sequence of intersections). The intersection locations are fixed and therefore do not change from one simulation to another. However, to evaluate different timing scenarios, the cycle and green signal phase lengths do need to vary from simulation to simulation. These values are populated using a uniform number generator between specific bounds, rounded to the nearest 5 seconds, which were obtained from [49]:

$$T_{\mathrm{L}}^{m} \in [50, 120], \qquad \phi_{\mathrm{g}}^{m} \in [20, T_{\mathrm{L}}^{m} - 20], \qquad \forall\, m = 1, \ldots, M. \tag{18}$$

2.2.2.2 Actuated Signal Operation

Moving outside of the tight urban districts and into more suburban areas, actuated signalized intersections become the prominent signal control method [1]. Signalized intersections have become extremely sophisticated, with the ability to collect information about traffic flow and strategically adjust signal phase lengths in an effort to maximize vehicle throughput. Additionally, modern signalized intersections can be networked such that they can be coordinated to further improve progression throughout the corridor. With these features and the advancements of 5G wireless communication technology, optimal vehicle controllers have access to a plethora of information, through V2I communication, to help make better-informed control decisions. For actuated signalized intersections, this information includes (but is not limited to) past signal phase timings, detector calls, traffic flow rate, and traffic density [1].



Note: The pedestrian phase is generally associated with (and will time concurrently with) the adjacent vehicle face (as illustrated by the blue dashed lines).

Figure 5: General four-way intersection with protected left turns.

In the U.S., the standard actuated signalized intersection contains three main features: vehicle detectors, a signal controller box, and indicator displays. To complete the system, a fourth component is the user (vehicle) who interacts with the signalized intersection. A user approaching an intersection has several movements as options (i.e. left turn/right turn/go straight) that are categorized into phases. Signalized intersections cycle through a sequence of phases, where each phase (ϕ) represents a green indicator for a specific set of movements [1]. Figure 5 shows a four-way, fully-actuated intersection with protected left turns, where the major direction of travel is north/south. The diagram is labeled with the traditional vehicle movements and phase numbering 1–8, along with pedestrian movements. The standard convention for phase numbers is as follows:

- *Even* phase numbers are associated with *through* movements.
- *Odd* phase numbers are associated with *left turn* movements.
- Phases 2, 6, 1, and 5 represent movements along the major street.
- Phases 4, 8, 3, and 7 represent movements along the minor street.

Figure 5 also shows phase numbers for right turn and pedestrian movements as well. Unless the right–turn movements require their own phase (protected right turn) in operation, they share the same phase number with the through movements. The pedestrian phase numbers are also associated with the adjacent vehicle phases for through movements [1].

A signal controller is ultimately responsible for organizing the signal phase sequences and durations. The organization of the signal phases is accomplished through *rings* and *barriers*, illustrated in Figure 6, where a ring shows a sequence of conflicting phases. Dual ring operations allow compatible phases to operate concurrently with phases in another ring. A barrier is the point at which the phases in both rings must end simultaneously, and typically separate major and minor street phases [1]. Phases between two barriers are known as *compatible* phases and are subject to the following two main rules:

1. Any phase in ring 1 can time with any phase in ring 2. For example, Figure 6 shows that

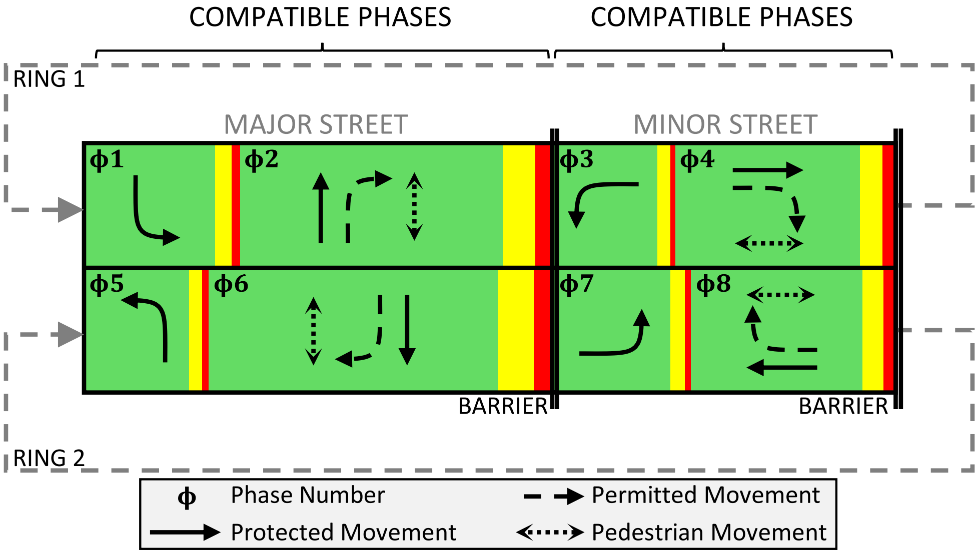


Figure 6: Ring and Barrier Diagram.

Phases 1, 2, 5, and 6 are compatible, which implies that Phases 1 or 2 can time with Phases 5 or 6 (similarly for the phases in between the other barriers).

2. Any phase in a ring can be skipped and/or give unused time to a following phase in that ring. For example, Phase 1 can give time to Phase 2, or Phase 3 can be skipped, allowing Phase 4 to start early if there is no demand for westbound left turns (referring back to Figure 5).

Lastly, it is possible for the left-turn phases to either lead or lag the through phases, or both (for example, Phase 1 can lead Phase 2 where Phase 5 can lag behind Phase 6). In Figure 6, both left–turn phases lead the through phases.

Several basic timing parameters are typically programmed for each phase at a signalized intersection, which all help determine the phase length. These parameters include, but are not limited to:

- Yellow change - Interval to warn users that there is about to be a change in the right-of-way assignment at the intersection,

- Red clearance - Interval after the yellow change interval, during which the phase has a red signal display before the display of green for the following phase. Its purpose is to allow the clearance of vehicles from the intersection that may have entered late,
- Minimum green - Parameter that represents the least amount of time that a green signal indication will be displayed for a given phase,
- Maximum green - Parameter that represents the maximum amount of time that a green signal indication can be displayed in the presence of a call on a conflicting phase (i.e., a vehicle approached the intersection on a conflicting lane of travel),
- Passage (Gap) time - A parameter that can be used to terminate the current phase when a large enough gap in traffic is identified,
- Pedestrian intervals – Interval to allow pedestrians to cross the intersection.

The yellow change and red clearance intervals are both static intervals. Since vehicles are lawfully allowed to pass through the intersection under a yellow indicator, that interval length will be considered part of the green signal phase and the red clearance interval will be considered part of the red signal phase, for the purposes of control. The minimum green parameter provides a lower bound on what the signal phase length can be, which will be utilized by the h-GLAS strategy. The maximum green parameter does not provide a precise upper bound on the signal phase length, as the maximum green timer will not start until there is a call for a phase change. However, since calls for phase changes can be recorded, the h-GLAS strategy can be notified when a call occurs, which then sets the maximum green parameter as the upper bound on the phase length. Actuated signalized intersections ultimately rely on detectors to notify the signal controller when vehicles arrive at/pass through an intersection. For a given green signal phase, every time a vehicle passes through an intersection, a passage timer (with length equal to the passage time parameter) is started/reset so that gaps in traffic can be identified.

For intersections that are spaced relatively close together (typically a mile or less), they may be coordinated with the goal that a vehicle can pass through all intersections without stopping, illustrated in Figure 7. This requires the local clock for each intersection to be referenced to a master clock with some offset parameter. The start of a signal cycle clock is defined to be the beginning of the signal phases for the major street of travel, which can be either the protected left turn or through movements. The top and bottom intersections, in Figure 7, have an offset of 50 s compared to the middle intersection with no offset. It is common practice to have the offset reference point be the beginning of the first coordinated phase yellow change interval [1]. Traffic control engineers try to program signal controllers in an effort to maximize vehicle progression, quantified through vehicle bandwidths. Vehicle bandwidths can be identified through vehicle trajectories overlaid upon a time-space diagram like in Figure 7. There are two vehicle trajectories of interest when identifying the bandwidth:

1. Beginning band: the vehicle trajectory that passes through each intersection at the moment the signal phase turns green.
2. Ending band: the vehicle trajectory that passes through each intersection at the moment the signal phase turns red.

The wider the bandwidth, the more effective the coordination.

The ring barrier diagrams, as well as coordination of signalized intersections, all imply fixed cycle time and phase lengths, which are actually programmed into the signal controllers. However, the signal controllers can perform an action called a *gap out*, which causes the phase lengths and ultimately the cycle times to vary. A gap out refers to the early termination of a signal phase, which only occurs if all of the following conditions are met:

1. The minimum green timer has expired.

2. A call is waiting for service on a conflicting phase.

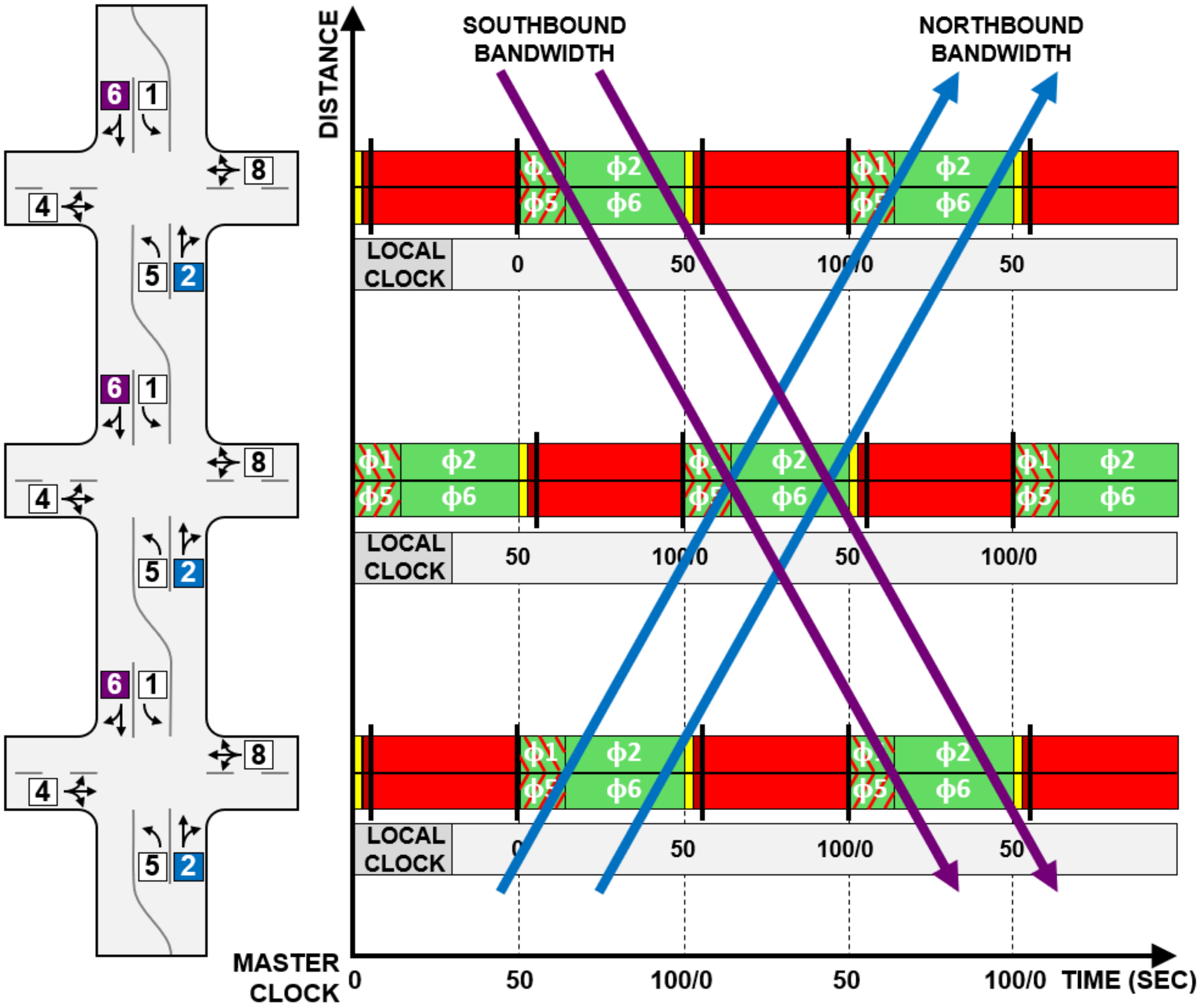


Figure 7: Basic Time–Space diagram (adopted from [1]) showing 3 intersections where the major through movements are coordinated (Phases 2 and 6). Focusing only on the major street of travel, the ring-barrier diagrams omit the phases for the side streets (illustrated by the red phases). The goal of coordination is to maximize the vehicle trajectory bandwidths, denoted by the diagonal purple and blue arrows.

3. The passage timer has expired.

The minimum green timer begins as soon as a signal phase begins. A call refers to the detector for that signal phase being triggered by a vehicle. When a vehicle arrives at the intersection for a conflicting phase (its phase is red), a call will go to the signal controller so that it knows there is a demand for a phase change. Lastly, the passage timer begins/resets every time a vehicle passes over a detector for a given active signal phase. Therefore, a given signal phase length can range anywhere from the minimum green length to the maximum green length.

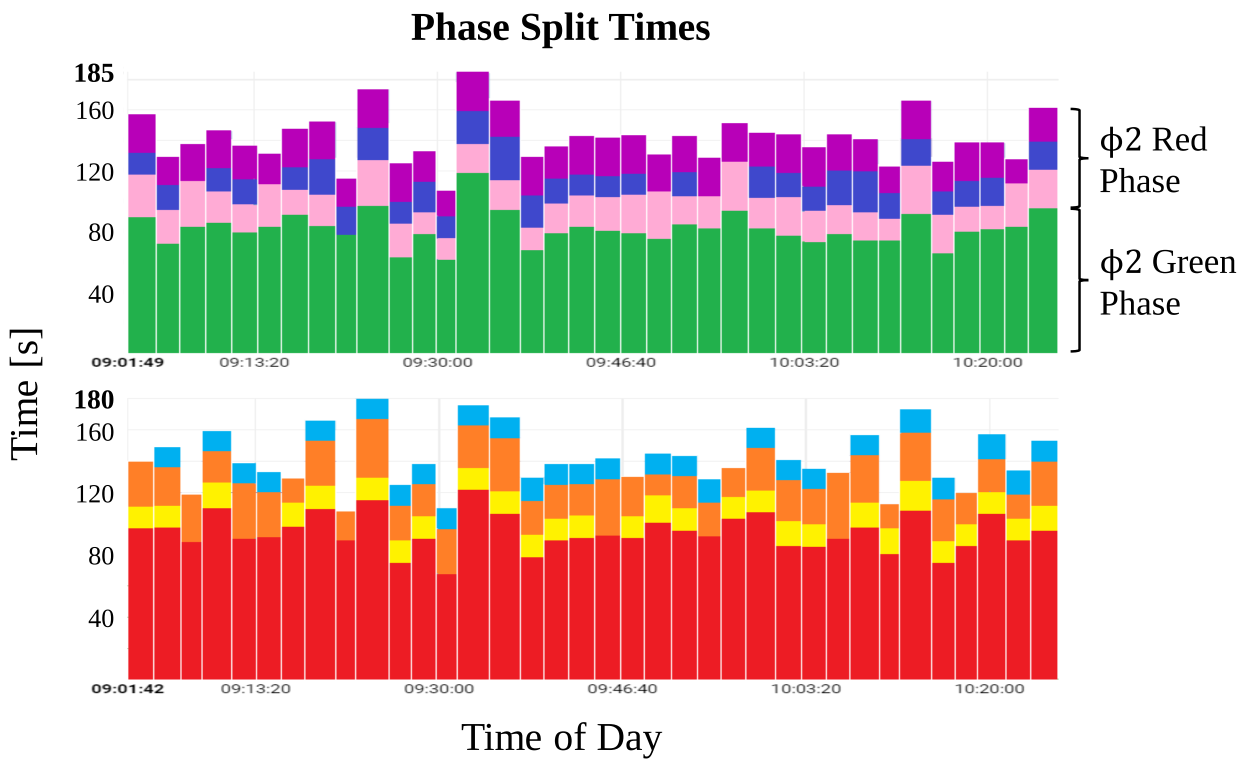


Figure 8: Signal phase split times for a four-way intersection with 8 individuals signal phases. The top plot shows the split times for ring 1 and the bottom shows a split times for ring 2 from the ring-barrier diagram. Note that the red signal phase length is the sum of the conflicting phases within its ring.

The phase split times for the 36 cycles pictured in Figure 8 represent the length of the green phase for each respective signal phase. The red phase length for a given movement is the summation of the phase lengths for the conflicting phases within its ring. For example, the red phase length for Phase 2 is the sum of Phases 1, 3, and 4. Since this research focuses on a single lane of travel, the h–GLAS strategy only requires the green and red phase lengths (ϕ_{green} and ϕ_{red}) for the movement that the ego vehicle will perform. This is achieved by receiving the start and end times for the signal phase corresponding to said phase.

Since signal phases are subject to early termination, the ego vehicle can only receive the signal phase lengths up to the current phase (the current phase length is unknown). Fortunately, there are potentially two points within each signal phase that can identify the time at which the phase will end. If protected pedestrian crosswalks exist, the first and most useful point is when the pedestrian crosswalk timer begins if the yellow change and red clearance intervals immediately follow, which is common practice [1]. Without the presence of pedestrian crosswalks, the point at which the end of the single phase is known is the beginning of the yellow change interval. Therefore, the current signal phase length is unknown until the time at which one of the two aforementioned points occur.

Two simulation studies involving actuated signalized intersections are presented later in this work, but are implemented in different ways. The simulation study that only evaluates the effect of actuated signal timing on fuel consumption (ignoring traffic) does not need to model the signal controllers online and rather relies on collected signal phase lengths. Therefore, using simulation time in a lookup table, the simulation identifies the current cycle number and supplies the vehicle controller with SPaT only up to that cycle number. If the simulation time falls during a yellow signal, the vehicle controller is also supplied with that phase's end time. For the simulation study that evaluates the combined effect that signalized intersections and traffic play on fuel consumption, the signal controllers *will* need to be modeled in the simulations, as the signal timing *depends* on traffic movements. For this, the commercial traffic simulator VISSIM from PTV is used to simulate both signal timing and traffic behavior. For either study, however, the SPaT information that is ultimately sent to the ego vehicle includes the following:

- The signal controller parameters (programmed timing intervals),
- The intersection locations,
- The current signal phase color and local clock time,
- Beginning and end timestamps for the signal phases over a desired time interval to infer past green and red phase lengths.

2.2.3 Suburban Environment Modeling

Simulation of the suburban environment is accomplished through PTV VISSIM, which is the standard traffic simulator in over 2500 cities worldwide [50]. VISSIM was also selected for its ability to interface with Matlab through a COM interface. This is required since all simulation and control design was performed in Matlab/Simulink. VISSIM is excellent at modeling traffic networks, including RBC signal controllers.

The network being simulated in VISSIM is US Hwy 29 through Concord, NC, which is a two-lane suburban highway. The simulated route contains 7 actuated signalized intersections along a 13.2 km route, which were programmed using controller parameters provided by Concord, NC's department of traffic (DOT). Only a subset of intersections was considered in this work, as the DOT was unable to share three timing plans along this portion of US Hwy 29. In order to simplify the network for simulation, the route was straightened out to remove all curves. This was done without loss of accuracy, because the simulations focus only on a single direction of travel. Simulating a curving route would require a transformation from longitude and latitude coordinates to position, which is not necessary for validating the controllers developed in this research. The intersection locations and speed limits were found using Google Maps, where in this work, an intersection's location is defined as the *stop line* location for the desired lane of travel. The roads that cross through US 29 are modeled only 100 m before and after the intersection.

In order to build an actuated intersection in VISSIM, several additional pieces of information are needed for each intersection:

- Vehicle detector locations, sizes, and operation parameters (i.e., delays, operation mode, corresponding phases, etc.). Note that in Figure 9, the induction-loop detectors, for the east-west direction, are located well before the intersection. This is common for higher speed approaches like US 29;
- Timing intervals (i.e., minimum green, maximum green, yellow change, etc.);
- RBC phase sequences;
- Coordination parameters (i.e., split times, cycle length, offset, etc.).

As ring-barrier control is the default actuated signalized intersection control method, RBC signal control is one of the default signal control options, only requiring the above parameters to be defined.

The last major feature that needs to be programmed is traffic behavior. VISSIM has three default road-vehicle classes that can be simulated in this network, namely passenger vehicles, heavy-duty vehicles, and buses. At the beginning of each road/lane modeled in VISSIM, *vehicle input* parameters must be defined for vehicles to actually enter the network, which encode the types and number of vehicle classes entering the network at the given lane of interest. Once a simulation begins, vehicles are randomly inserted according to a *relation* value on vehicle class. For example, relation values of 0.8, 0.1, and 0.1 for passenger vehicles, heavy-duty vehicles, and buses, respectively, mean 80% of vehicles entering the network at that location will be passenger vehicles, 10% will be heavy-duty vehicles and 10% will be buses. Once vehicles are in the network, they travel straight until passing over a *routing decision* that decides which movement to perform at the next intersection (U-turn, left/right turn, or go straight) based on its own *relation* parameters. These relation parameters operate exactly like vehicle inputs, defining the ratio of vehicles turning versus going straight. In regard to right turn-on-red (and other yielding movements), VISSIM requires the definition of movement priorities, which are shown in Figure 9 as the green and red objects.

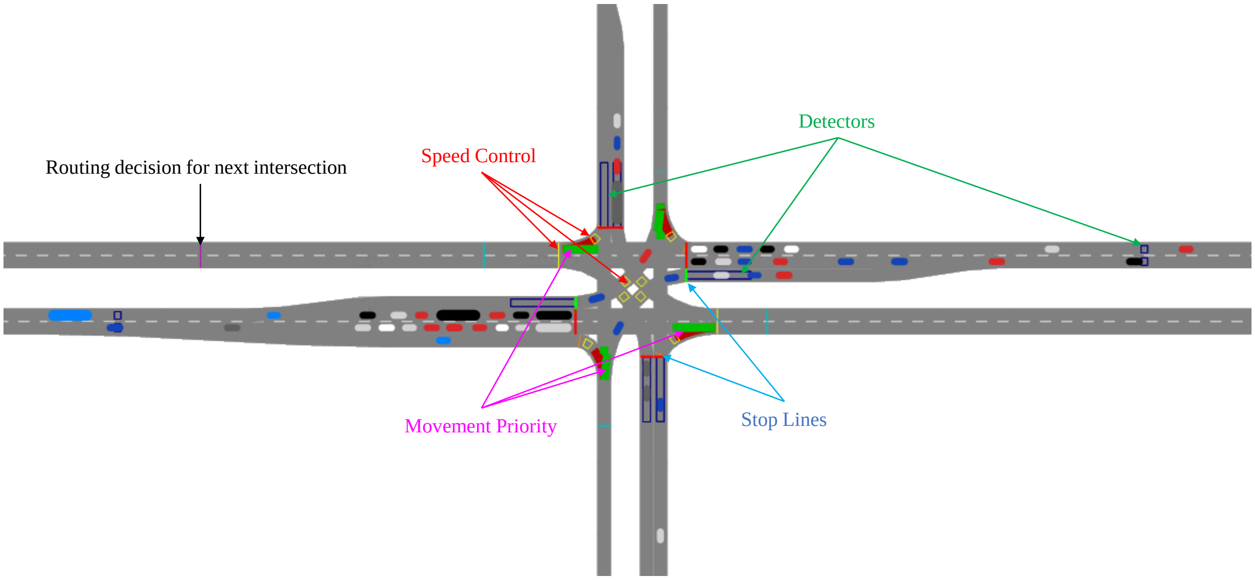


Figure 9: Intersection of US 29 (east-west) and Pitts School Rd. (north-south).

With these behaviors defined alone, vehicles can navigate throughout the network. However, additional behavior is still required to capture the proper behavior of traffic. Each vehicle class receives its own desired speed distribution that defines the velocity behavior for each vehicle. For example, passenger vehicles will have larger speed distributions than heavy-duty vehicles and buses on average, as they typically travel at faster speeds. In VISSIM, desired speed distributions are defined at the vehicle input locations, but additional definition points need to be included along the route. These locations correspond to speed limit changes and every intersection exit. A vehicle that makes a turn at an intersection will maintain its desired speed distribution from before a turn, which is why it requires a new speed distribution be defined at the intersection exit. The final speed behavior to be defined is a *speed reduction* zone for turning at intersections. Speed reductions merely define what turning speed each vehicle class will follow. If speed reduction zones were omitted, vehicles would turn at full speed, which is unrealistic.

One of the signalized intersections is displayed in Figure 9. Vehicle detectors are shown as blue rectangles on each road. All speed control objects are shown as yellow lines for desired speed distribution decisions or yellow boxes for speed reductions. Movement priorities are shown as the green and red objects, with the green lane of travel having right-of-way. The stop lines double as signal heads during simulation. Shown in Figure 9, the left turns for US 29 currently have green signals. Lastly, the routing decision for an intersection to the left is shown as a purple line on the road.

2.2.4 Simulation Routes

A total of six routes were used to validate the control strategies developed in this research, including four highway, one urban, and one suburban route. The four highway routes represent four real-world trucking routes. Their terrain profiles and speed limits are displayed in Figure 10.

The four highway routes are designated as HWY 1 (top left), HWY 2 (bottom left), HWY 3 (top right), and HWY 4 (bottom right). HWYs 1 and 4 have more variable terrain than the other two routes. Therefore, one can expect the fuel savings to be greater on HWYs 1 and 4.

The single urban route simulated in this research, also provided by Volvo Group Trucks, contains

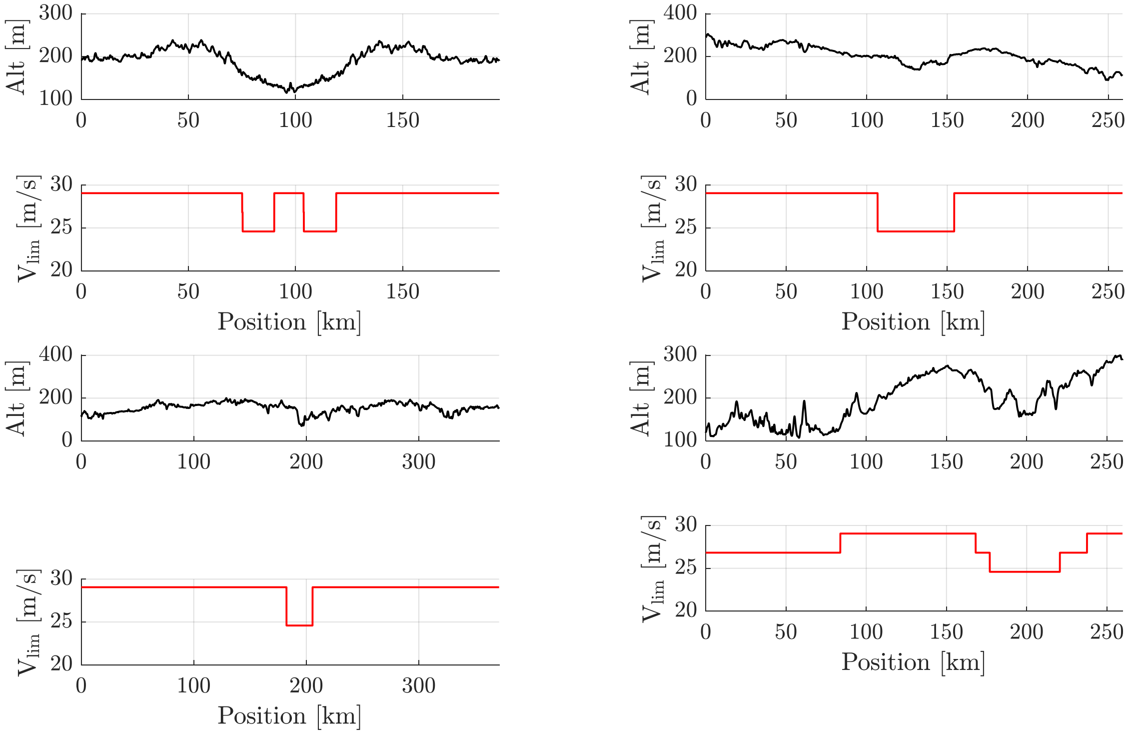


Figure 10: Terrain profiles and speed limits for the four highway routes considered in this research.

the intersection locations (27 in total), terrain profile, and speed limits. Before simulation, the SPaT values are selected through equation (18). The SPaT shown in the top plot of Figure 11 is just one example of the timing from (18). One should note that the timing parameters *do* reflect realistic values, but since they are randomly assigned, they will not reflect actual programmed values seen in real life.

For the suburban route, US Hwy 29 through Concord, NC is the selected candidate for consideration in this dissertation. Shown in Figure 12, there are 10 intersections that are more spread out than in the urban route, and the speed limit is greater than the urban route on average. The signal timing shown represents actual signal timing collected over 500 seconds by the DOT. The variation in signal phase lengths can easily be seen in multiple intersections. A Google map of the suburban route is shown in Figure 13 with the intersection locations overlaid on top.

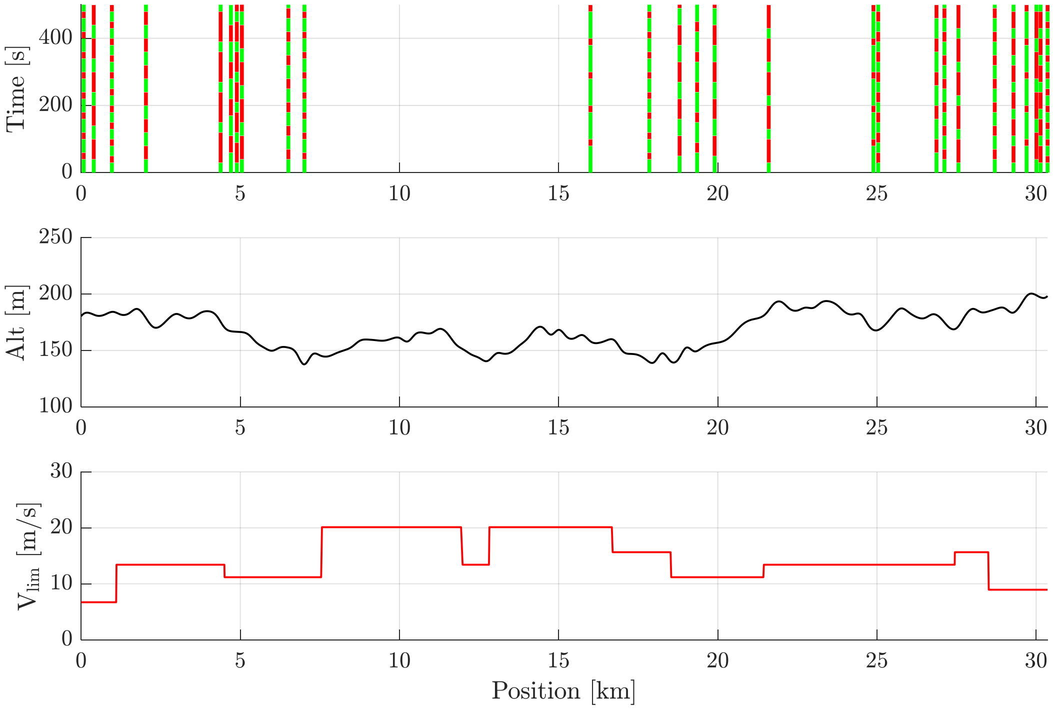


Figure 11: Single urban route intersection locations, terrain profile, and speed limit.

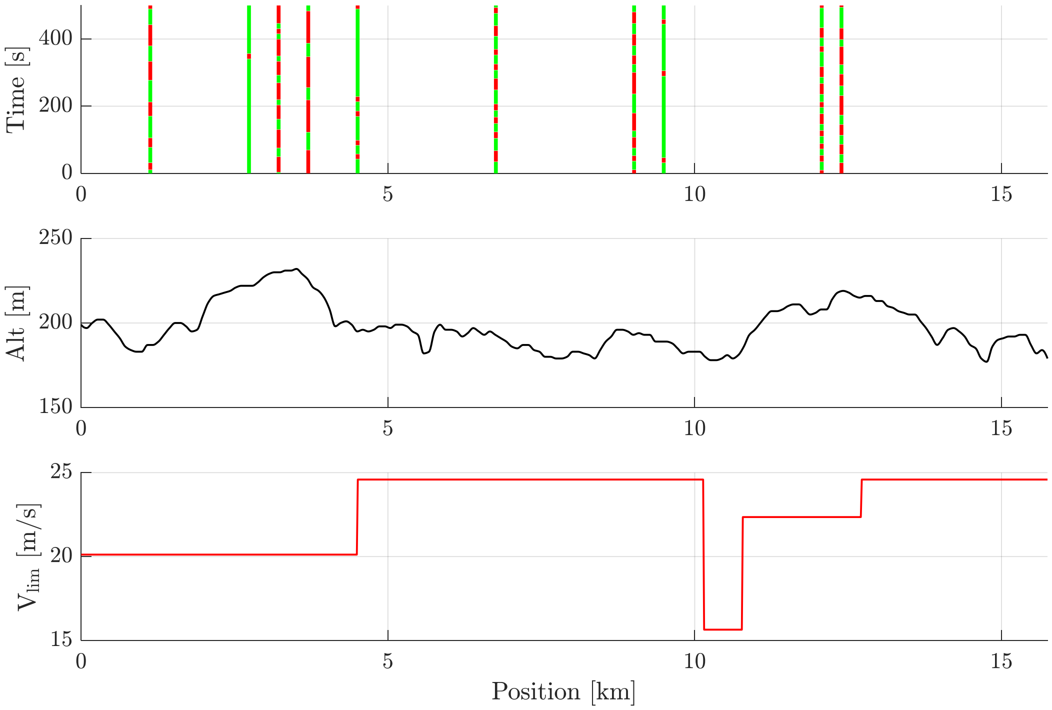


Figure 12: Single suburban route (US Hwy 29) intersection locations, terrain profile, and speed limit.

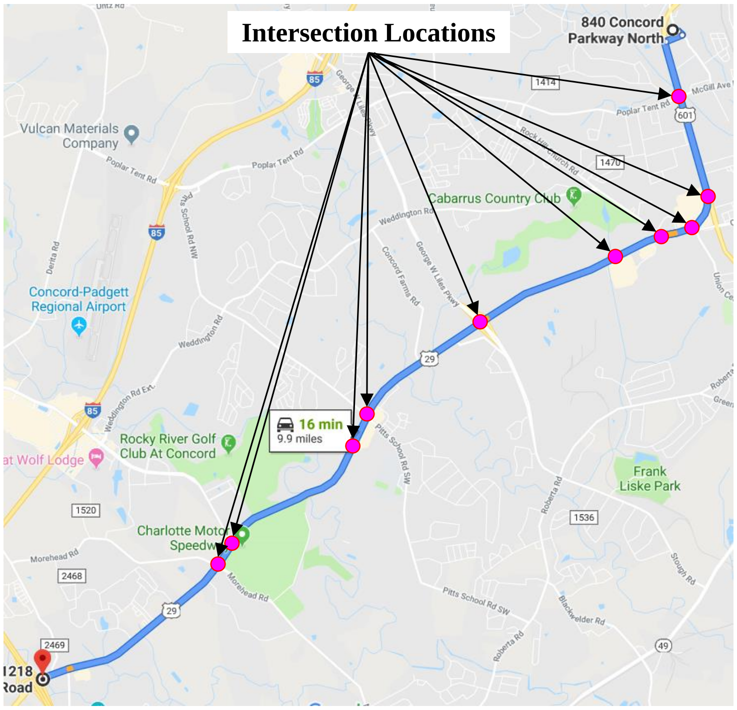


Figure 13: Portion of HWY route 29 used for simulating a suburban environment. Credit: Google Maps.

CHAPTER 3: FOUNDATIONAL CONTROLLERS

Early work in this research focused on highway environments, as they account for the majority of a heavy-duty truck's domain. As a matter of fact, highway environments represent just a subset of urban/suburban environments, with the only difference (in regard to control) being the presence of intersections. Therefore, following a model predictive controller (MPC) comparative study in a highway environment, the h-GLAS controller is adapted for urban environments with fixed-time signalized intersections. All control strategies developed in this dissertation are validated against a *baseline* control strategy, designed to emulate human driving behavior and based on a combination of a PI controller and the well-established Gipps car following model [51].

3.1 Baseline Control Strategy

In order to evaluate the efficacy of each control strategy developed in this work, a baseline control strategy needs to be defined. The baseline control strategy used in these studies consists of a simple PI speed controller that tracks a velocity setpoint (v_{sp}). In an open road scenario (no intersections or traffic), v_{sp} is set to the speed limit. This is because typical human driving behavior is to drive at or near the speed limit. To emulate human following behavior in the presence of traffic, a Gipps car following model is used to adjust v_{sp} based on the lead vehicle's position and velocity [51]. Lastly, in the presence of signalized intersections, the values for v_{sp} were adopted from [52] and [53], which utilizes a stopping velocity profile from a deceleration curve extracted from collected vehicle stopping data. This stopping velocity profile was only referenced when the vehicle controlled by the baseline controller was destined to arrive at the upcoming intersection while the traffic signal was yellow or red. Once the signal turned green, v_{sp} was reset back to its original value.

3.2 Highway Environment

To solve the problem of optimally traversing a highway environment, three separate MPC strategies were developed in [35] to discover the best option, considering both performance and implementation feasibility:

1. *Direct Fuel Consumption Minimization (MPC 1):* In this approach, a sequential quadratic program (SQP)-based MPC is used with the nonlinear time-based dynamic model from Chapter 2.1.1 to minimize predicted fuel consumption directly, subject to terminal penalties based on deviation from a DP-optimized velocity trajectory.

2. *Braking Penalty (MPC 2):* In this approach, the direct penalty on fuel consumption from the first approach is replaced with a penalty on braking force (negative applied wheel force), recognizing that (1) this is the single largest contributor to wasted energy and (2) the replacement of fuel consumption with a braking penalty greatly simplifies the computational burden.

3. *Convex Energy Approximation (MPC 3):* In this approach, a convex quadratic program (QP)-based MPC is used with the affine distance-based dynamic model from Chapter 2.1.2, which trades off energy consumption over the horizon with adherence to the velocity trajectory prescribed by the DP optimization.

The three strategies trade off fidelity of the underlying optimization with computational complexity. In particular, MPC 1 considers an objective function that is directly related to the true economic objective of the system at hand, but also results in the most complex optimization of the three strategies. MPC 3 utilizes an objective function and dynamic model that are farthest from the economic and physical "truth" of the system at hand, but also carries the lowest computational burden. MPC 2 falls between the first and third strategies in terms of fidelity and computational complexity; it incorporates an optimization that is non-convex but nonetheless simpler than that of the direct fuel consumption minimization, while using an objective function that acts only as a surrogate for the true economic objective at hand. The block diagram, shared by all three strategies, is displayed in Figure 14, where the only difference is the math within the MPC block.

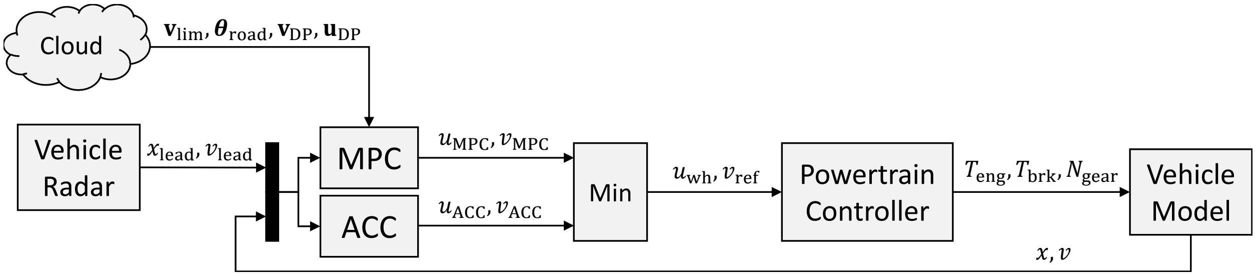


Figure 14: Highway controllers' block diagram.

The control strategies presented in this work rely on vehicle-to-infrastructure (V2I) communication. Through this connectivity, the MPC receives the route speed limits ($\mathbf{v}_{\text{lim}}$) and grade (θ_{road}) profiles with respect to position along the route. Additionally, all three MPC strategies reference a globally optimal solution from a dynamic programming (DP) optimization, which is performed offline before the vehicle departs on its trip. This profile is referenced to obtain terminal state constraints and to serve as an *initial solution* for the SQP-based MPCs and a reference trajectory for the QP-based MPC. Therefore, in addition to $\mathbf{v}_{\text{lim}}$ and θ_{road}, the DP solution is also provided ($\mathbf{v}_{\text{DP}}$ and $\mathbf{u}_{\text{DP}}$) to the MPC. Using the information provided by the cloud, radar data on the lead vehicle (x_{lead} and v_{lead}), and its own states (x and v), the MPC computes the optimal wheel force

(u_{MPC}) for the vehicle to apply, and its reference velocity (v_{MPC}). The value of v_{MPC} is simply the velocity that the MPC predicts the vehicle will be traveling at after applying u_{MPC}, which is used to select optimal gear. Due to the discrete-time nature of the MPC, implemented in parallel is a continuous vehicle-following controller (ACC block), which is present to ensure safe vehicle following in-between MPC time steps. Lastly, the minimum of u_{MPC} and u_{ACC} is sent to the powertrain controller (traditional adaptive cruise control setup), which computes the appropriate engine and brake torque to drive the vehicle model.

3.2.1 Offline Dynamic Programming Optimization

The first component of the control strategy for highway environments involves the generation of an offline globally optimal solution for each candidate route. This is a reasonable feature to include as heavy-duty trucking routes are typically pre-planned trips. For each route, the DP process optimizes the sequence of propulsive forces, denoted by $\mathbf{u} = [u(0) \ \ldots \ u(N_{\mathrm{DP}} - 1)]^T$, over the course of an entire route (N_{DP} is the total number of time steps). Starting at time step N_{DP} and a final desired position state ($x(N_{\mathrm{DP}})$), DP works backwards, one time step at a time, finding the optimal control input between each time step until an optimal sequence has been formed from any initial state to $x(N_{\mathrm{DP}})$. The optimization problem is formulated as

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimize}} \quad & J(\mathbf{u};\mathbf{x},\mathbf{v}) = \sum_{i=0}^{N_{\mathrm{DP}}-1} R_{\mathrm{fuel}}(i)\Delta t_{\mathrm{DP}}, \qquad (19) \\ s.t. \quad & u_{\min} \le u(i) \le u_{\max}, \\ & v_{\min} \le v(i) \le v_{\max}, \quad \forall\, i = 0 \ldots N_{\mathrm{DP}} - 1, \\ & x(N_{\mathrm{DP}}) = x_{\mathrm{target}}, \\ & \text{Equations (7) and (8)}, \end{aligned}$$

Table 2: DP Discretization Parameters. These represent the discretization steps of trip time, position along the route, vehicle velocity, and the control input (wheel force).

DP Parameters	
Time Step [s]	10
Distance Step [m]	20
Velocity Step [m/s]	0.2
Control Step [N]	200

where Δt_{DP} is the time step used by the DP optimization and R_{fuel} is the rate of fuel consumed, which is calculated through equation (9). Both the control force and velocity are subject to saturation constraints ($u_{\min}$, $u_{\max}$, $v_{\min}$, and $v_{\max}$). Ultimately, the DP optimization results not only in an optimal control sequence, $\mathbf{u}^*$, but it also provides the optimal state sequences for position (x) and velocity (v) as a byproduct. These optimized sequences ($\mathbf{x}_{\mathrm{DP}}$ and $\mathbf{v}_{\mathrm{DP}}$) are denoted by

$$\mathbf{x}_{\mathrm{DP}} = \begin{bmatrix} x_{\mathrm{DP}}(0) & \dots & x_{\mathrm{DP}}(N_{\mathrm{DP}}) \end{bmatrix}^T, \tag{20}$$

$$\mathbf{v}_{\mathrm{DP}} = \begin{bmatrix} v_{\mathrm{DP}}(0) & \dots & v_{\mathrm{DP}}(N_{\mathrm{DP}}) \end{bmatrix}^T. \tag{21}$$

These sequences are referenced by each of the MPC strategies to form terminal state constraints and initial trajectories for the SQP-based MPCs and a reference trajectory for the QP-based MPC. The main parameters for the DP optimization are listed in Table 2. These were selected to create as fine of a discretization as possible while considering practical computational limitations. On average, it takes approximately 24 hours to compute the DP solution for a 100 km route using these values.

3.2.2 Highway Model Predictive Controllers

The goal of all three MPC strategies presented here is to minimize fuel consumption without compromising the total trip time from a baseline control strategy. For the SQP-based MPC strategies, the DP solution is used to formulate penalties on the terminal states ($x_{\text{final,des}}$ and $v_{\text{final,des}}$) of the MPC optimization, as well as the initial guess for the SQP solver. Specifically, the values of the terminal states represent the position and velocity values at the end of the MPC horizon that would be realized if the DP solution were implemented exactly as originally optimized. These terminal penalties ensure that (1) the MPC solution reaches its final position in a fixed time (maintain trip time) and that (2) the vehicle does not release its built up kinetic energy at the end of the horizon, as the end of the MPC horizon is not the end of the trip. The terminal penalties do not constitute hard constraints and are instead implemented as soft constraints in the objective function, penalizing any violations. This is required, as $x_{\text{final,des}}$ and $v_{\text{final,des}}$ can (and will) be violated, especially for instances where the presence of lead vehicles necessitates control to avoid a collision (violates $x_{\text{final,des}}$ in this example). The QP-based MPC cannot enforce terminal states, as the dynamic model only characterizes the vehicle's kinetic energy. Therefore, the velocity state sequence from the DP solution is converted into a desired kinetic energy sequence to be tracked within the MPC's objective function. Lastly, the time-based and distance-based MPC strategies optimize N_t and N_d horizon lengths, respectively.

3.2.2.1 Vehicle Following Strategy

Before detailing the MPC formulations, the vehicle following policy needs to be defined. For vehicle-following control, the ego vehicle needs to receive information regarding the lead vehicle's current position and velocity ($x_{\text{lead}}(I)$ and $v_{\text{lead}}(I)$), where I represents the current time step along the route. This is accomplished through an on-board radar/lidar system at the front of the vehicle to measure the inter-vehicular distance. Over the MPC horizon, v_{lead} is assumed to be constant, and x_{lead} evolves according to a forward Euler approximation:

$$x_{\text{lead}}(i+1|I) = x_{\text{lead}}(i|I) + v_{\text{lead}}(I)\Delta t. \tag{22}$$

From here, the following distance $(\delta(i|I))$ is calculated as the difference between the lead and ego vehicle's position:

$$\delta(i|I) = x_{\text{lead}}(i|I) - x(i|I). \tag{23}$$

In the equations above, and throughout this work, the notation $(i|I)$ denotes the predicted value of a variable at step i along a horizon corresponding to the optimization (and corresponding prediction) performed at step I along the route.

Most of the literature on MPC with car following [30–32] utilizes a single following constraint in their optimization that acts as a *hard* constraint:

$$\delta_{\text{min}} \leq \delta(i|I),\ \forall\, i = I, \ldots, I + N_t - 1, \tag{24}$$

where δ_{min} represents the minimum following distance between the lead and ego vehicles. This accomplishes safe car following, but can be inefficient if the lead vehicle travels slower than the MPC solution on average. This results in the ego vehicle operating at nearly *exactly* the safe following distance, forcing the ego vehicle to brake any time that the lead vehicle brakes. Therefore, MPCs 1 and 2 incorporate an increased *soft* following distance constraint (i.e., a penalty) within the objective function, while also including the minimum following distance (δ_{hard}) as a *hard* constraint to be enforced, shown in Figure 15. Incorporation of this two-constraint policy ensures that the ego vehicle never travels at an unsafe following distance (through he hard constraint) but also incentivizes the ego vehicle to maintain *additional* following distance (through the soft constraint), so to be able to decelerate with minimal braking and still always maintain the minimum safe following distance (as dictated by the hard constraint). Both δ_{hard} and δ_{soft} are characterized by a corresponding following time $(t_{\text{hard}}$ and $t_{\text{soft}})$ and the ego vehicle's current velocity:

$$\delta_{\text{hard}}(i|I) = \min(v(I)t_{\text{hard}}, \delta(I) + ri\Delta t), \tag{25}$$

$$\delta_{\text{soft}}(i|I) = v(I)t_{\text{soft}}, \forall\, i = I, \ldots, I + N_t - 1. \tag{26}$$

The second term in the $\min$ function of (25) accounts for vehicle cut-ins that violate the hard constraint at step I. This term sets $\delta_{\text{min}}(i|I)$ to the initial following distance $(\delta(I))$ and ramps $\delta_{\text{min}}(i|I)$ at each time step until it reaches its predefined value $(v(I)t_{\text{hard}})$ at a rate determined by r, where $t_{\text{hard}} < t_{\text{soft}}$. This ensures that the SQP optimization always converges to a feasible solution. From an interview with a long-haul truck driver, Chuck Watson, a following time of $1\,\text{s}$ is recommended for t_{hard} and a value of $2\,\text{s}$ for t_{soft}.

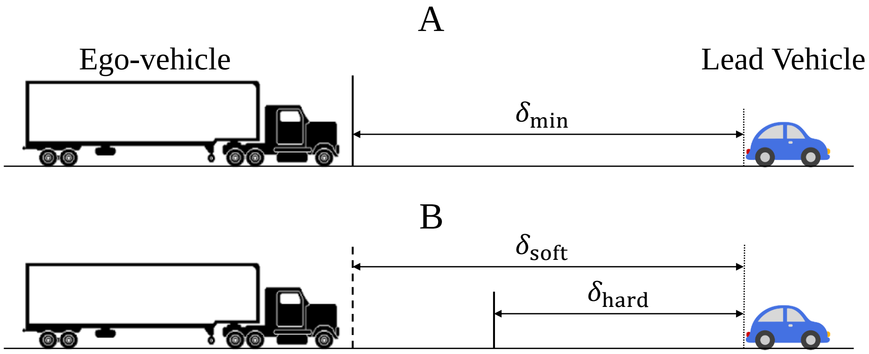


Figure 15: MPC Following Distance Schematic: (A) Traditional MPC hard following constraint. (B) Implemented soft/hard following constraint combination.

3.2.2.2 Direct Fuel Consumption Minimization Strategy

The first MPC formulation minimizes the predicted fuel consumed over the entire prediction horizon, as well as violations on terminal conditions derived from the DP optimization (where "terminal" in this context refers to the end of the MPC horizon, as opposed to the end of the route), along with the vehicle following penalty. The prediction horizon for this strategy is discretized into N_t time steps of length Δt_{MPC}. The optimization problem for this strategy is formulated as follows:

$$
\begin{aligned}
\underset{\mathbf{u}}{\text{minimize}}\quad J(\mathbf{u}(I); x(I), v(I)) = \sum_{i=I}^{I+N_t-1} \Big[& R_{\text{fuel}}(i|I)\Delta t_{\text{MPC}} \\
& + a_1[\min(0, x(I+N_t|I) - x_{\text{final,des}})]^2 \\
& + a_2[\min(0, v(I+N_t|I) - v_{\text{final,des}})]^2 \\
& + a_3[\min(0, \delta(i|I) - \delta_{\text{soft}})]^2 \Big]
\end{aligned}
\tag{27}
$$

$$
\begin{aligned}
s.t.\quad & u_{\min} \le u(i|I) \le u_{\max}(I), \\
& v_{\min}(i|I) \le v(i|I) \le v_{\max}(i|I), \\
& \delta_{\text{hard}} \le \delta(i|I),\ \forall\, i = I, \ldots, I+N_t-1, \\
& \text{Equations (7) and (8)},
\end{aligned}
$$

where a_1, a_2, and a_3 are tunable weights that dictate the importance placed on each penalty term in the objective function. The maximum control value is dependent on the current gear at step I, and therefore can change from one time step to another. However, the MPC assumes the current gear will remain constant over the prediction horizon. The value of $v_{\max}$ is the speed limit along the route, and the value of $v_{\min}$ is 70% of $v_{\max}$. This allows sufficient room for the MPC to vary the vehicle's velocity to improve fuel economy.

3.2.2.3 Braking Penalty Strategy

Because of the relatively complex nonlinear dynamics and static mappings that ultimately relate R_{fuel} to x, v, and u, computational complexities arise from the approach of directly penalizing fuel consumption, as is done in the aforementioned strategy. The braking penalty strategy uses braking force as a surrogate measure for wasted energy, recognizing that substantial braking effort is highly correlated with poor fuel economy. The braking reduction strategy disincentivizes negative control inputs instead of directly minimizing R_{fuel}. The MPC optimization problem for this strategy is as

$$
\begin{aligned}
\underset{\mathbf{u}(I)}{\text{minimize}} \quad & J(\mathbf{u}(I); x(I), v(I)) = \sum_{i=I}^{I+N_t-1} \Big[\min(0, u(i|I))^2 \\
& \qquad + b_1[\min(0, x(I+N_t|I) - x_{\text{final,des}})]^2 \\
& \qquad + b_2[\min(0, v(I+N_t|I) - v_{\text{final,des}})]^2 \\
& \qquad + b_3[\min(0, \delta(i|I) - \delta_{\text{soft}})]^2 \Big] \\
s.t. \quad & u_{\min} \le u(i|I) \le u_{\max}(I), \\
& v_{\min}(i|I) \le v(i|I) \le v_{\max}(i|I), \\
& \delta_{\text{hard}} \le \delta(i|I), \ \forall\, i = I, \dots, I+N_t-1, \\
& \text{Equations (7) and (8)},
\end{aligned}
\tag{28}
$$

where $x_{\text{final,des}}$ and $v_{\text{final,des}}$ are found using the same process used for the direct fuel consumption minimization strategy, and b_1, b_2, and b_3 are again tunable weights on each penalty of the objective function. When there is no lead vehicle present, the prediction and control horizon for MPC strategies 1 and 2 were selected to be 120 and 80 seconds respectively, with a time step of 5 seconds. Using a control horizon of 80 seconds vs. 120 seconds drops the number of decision variables to be optimized by 33%, reducing the overall computational burden. When the control horizon is smaller than the prediction horizon, the control value at the end of the control horizon is assumed to be held constant over the remaining steps of the prediction horizon. A prediction horizon of 120 seconds equates to a distance around 3 km under normal highway velocities, which allows for the most efficient driving over the rolling terrain [35]. However, a 5 second time step is not reasonable to account for vehicle following, so in the presence of a lead vehicle, the time step is reduced to 1 second, with prediction and control horizons of 24 and 16 seconds, respectively. The number of control and prediction steps are kept the same for each scenario to simplify the implementation. Note that all controller parameters for the three strategies were obtained through heuristic tuning.

3.2.2.4 Convex Energy Minimization Strategy

While the aforementioned braking reduction strategy eliminates some of the complexities of the direct fuel consumption minimization strategy, it still involves a nonlinear model, which leads to a non-convex optimization problem. In an effort to further simplify the optimization, a third MPC formulation relies on the affine distance-based energy model of (14) and a quadratic objective function, which leads to the following convex quadratic program (QP) problem. The implementation for this formulation will be more detailed than the previous two, as it is the strategy used throughout the remainder of this research. With the goal of developing a control formulation that can be deployed to target hardware for hardware-in-the-loop, engine-in-the-loop, and in-vehicle testing, this MPC candidate represents the best option. Further quantitative justification will be seen when discussing the highway simulation results.

Minimize:

$$J(\mathbf{u};\mathbf{x}_0) = \sum_{k=K}^{K+N_d-1} \Big[\big[\mathbf{x}(k+1) - \mathbf{x}_d(k+1)\big]^T Q \big[\mathbf{x}(k+1) - \mathbf{x}_d(k+1)\big] + \mathbf{u}^T(k) R \mathbf{u}(k) \Big], \quad (29)$$

subject to:

$$\mathbf{u}_{\min} \leq \mathbf{u}(k) \leq \mathbf{u}_{\max}(K), \tag{30}$$

$$\mathbf{x}_{\min}(k+1) \leq \mathbf{x}(k+1) \leq \mathbf{x}_{\max}(k+1), \;\; \forall\, k = K, \ldots, K + N_d - 1, \tag{31}$$

The variable $\mathbf{x}_d$ represents a state setpoint to be tracked from the DP solution, serving as a mobility term to replace the terminal state penalties from MPC 1 and 2. Referring back to Chapter 2.1.2, this system consists of a single state, $\mathbf{x} = [E_k]$, and two control variables, $\mathbf{u} = [E_m \;\; E_b]^T$, representing kinetic energy and traction and braking work, respectively (even though there is only one state, the vector notation $(\mathbf{x})$ will be maintained). The variables Q and R represent weighting matrices to dictate the importance of minimizing deviations from the reference trajectory vs minimizing control effort. This optimization is distance-based; therefore, k represents a distance step along the horizon at distance step K along the route.

A QP formulation requires (29) to be represented in terms of only $\mathbf{u}$ and known quantities. Therefore, a lifted representation of this system needs to be constructed, which is accomplished by expanding (14) out a few steps to obtain a recurring relationship between $\mathbf{u}$ and $\mathbf{x}_0$ for $\mathbf{x}(k)$.

$$\begin{aligned}
\mathbf{x}(1) &= A\mathbf{x}_0 + B\mathbf{u}(0) + F(0), \\
\mathbf{x}(2) &= A\mathbf{x}(1) + B\mathbf{u}(1) + F(1) = A^2\mathbf{x}_0 + AB\mathbf{u}(0) + B\mathbf{u}(1) + AF(0) + F(1), \\
\mathbf{x}(3) &= A\mathbf{x}(2) + B\mathbf{u}(2) + F(2) = A^3\mathbf{x}_0 + A^2B\mathbf{u}(0) + AB\mathbf{u}(1) + B\mathbf{u}(2) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad + A^2F(0) + AF(1) + F(2).
\end{aligned}$$

After three steps, the following relationship can be seen:

$$\mathbf{x}(k) = \sum_{k=K}^{K+N_d-1} \left[A^{n-1-k}B\mathbf{u}(k) + A^{n-1-k}F(k) + A^n\mathbf{x}_0, \;\; \forall\, k = K, \ldots, K + 1 + N_d, \right. \tag{32}$$

which can be expressed in matrix form $\mathbf{x} = H\mathbf{u} + G\mathbf{F} + A^{*}\mathbf{x}_0$:

$$\begin{bmatrix} \mathbf{x}(1) \\ \mathbf{x}(2) \\ \mathbf{x}(3) \\ \vdots \\ \mathbf{x}(N_d) \end{bmatrix}_{N_d\times 1} = \begin{bmatrix} B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ A^2B & AB & B & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{N_d-1}B & A^{N_d-2}B & A^{N_d-3}B & \cdots B & \end{bmatrix}_{N_d\times 2N_d} \begin{bmatrix} \mathbf{u}(0) \\ \mathbf{u}(1) \\ \mathbf{u}(2) \\ \vdots \\ \mathbf{u}(N_d-1) \end{bmatrix}_{2N_d\times 1}$$

$$+ \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ A & 1 & 0 & \cdots & 0 \\ A^2 & A & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{N_d-1} & A^{N_d-2} & A^{N_d-3} & \cdots 1 & \end{bmatrix}_{N_d\times N_d} \begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ \vdots \\ F(N_d-1) \end{bmatrix}_{N_d\times 1} + \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^{N_d} \end{bmatrix} \mathbf{x}_0, \tag{33}$$

An expression for $\mathbf{x}(k)$ now exists, which is given by

$$\mathbf{x}(k) = h(k)\mathbf{u} + g(k)F + A^k\mathbf{x}_0,\ \ \forall\, k = K, \ldots, K+1+N_d, \tag{34}$$

where $h(k)$ and $g(k)$ represent the k^{th} row of the matrices being multiplied by $\mathbf{u}$ and $\mathbf{F}$ from equation (33), respectively. The sizes of H and G are $N_d \times 2N_d$ and $N_d \times N_d$, respectively. Inserting (34) into (29), a cost function without the state variable present is obtained, albeit a messy one:

$$J = \sum_{k=K}^{K+N_d-1} \Big[\big[h_{k+1}\mathbf{u} + g_{k+1}F + A^{k+1}\mathbf{x}_0 - \mathbf{x}_{d,k+1}\big]^T Q \big[h_{k+1}\mathbf{u} + g_{k+1}F + A^{k+1}\mathbf{x}_0 - \mathbf{x}_{d,k+1}\big] + \mathbf{u}_k^T R \mathbf{u}_k \Big]. \tag{35}$$

To clean it up, the expression around Q needs to be expanded. As there are four terms being expanded, 16 new terms now exist:

$$
\begin{aligned}
&J_{1,1} = \mathbf{u}^T h_{k+1}^T Q h_{k+1} \mathbf{u}, && J_{1,2} = \mathbf{u}^T h_{k+1}^T Q g_{k+1} F, && J_{1,3} = \mathbf{u}^T h_{k+1}^T Q A^{k+1} \mathbf{x}_0, \\
&J_{1,4} = -\mathbf{u}^T h_{k+1}^T Q \mathbf{x}_{d,k+1}, && J_{2,1} = F^T g_{k+1}^T Q h_{k+1} \mathbf{u}, && J_{2,2} = F^T g_{k+1}^T Q g_{k+1} F, \\
&J_{2,3} = F^T g_{k+1}^T Q A^{k+1} \mathbf{x}_0, && J_{2,4} = -F^T g_{k+1}^T Q \mathbf{x}_{d,k+1}, && J_{3,1} = \mathbf{x}_0^T \left(A^{k+1}\right)^T Q h_{k+1} \mathbf{u}, \\
&J_{3,2} = \mathbf{x}_0^T \left(A^{k+1}\right)^T Q g_{k+1} F, && J_{3,3} = \mathbf{x}_0^T \left(A^{k+1}\right)^T Q A^{k+1} \mathbf{x}_0, && J_{3,4} = -\mathbf{x}_0^T \left(A^{k+1}\right)^T Q \mathbf{x}_{d,k+1}, \\
&J_{4,1} = -\mathbf{x}_{d,k+1}^T Q h_{k+1} \mathbf{u}, && J_{4,2} = -\mathbf{x}_{d,k+1}^T Q g_{k+1} F, && J_{4,3} = -\mathbf{x}_{d,k+1}^T Q A^{k+1} \mathbf{x}_0, \\
&J_{4,4} = \mathbf{x}_{d,k+1}^T Q \mathbf{x}_{d,k+1}, && \forall\, k = K, \ldots, K+1+N_d.
\end{aligned}
$$

Out of the 16 new components and the last term from (35), the coefficients associated with the quadratic terms can be consolidated into a new variable,

$$\bar{Q} \triangleq \sum_{k=K}^{k+N_d-1} \left[h_{k+1}^T Q h_{k+1} + R I_{N_d \times N_d}\right]. \tag{36}$$

Since Q is a scalar, $Q = Q^T$ and $F^T g_{k+1}^T Q h_{k+1} \mathbf{u} = \mathbf{u}^T h_{k+1}^T Q g_{k+1} F$. The coefficients associated with the linear terms from (35) can now be consolidated into a new variable,

$$\bar{r} \triangleq \sum_{k=K}^{K+N_d-1} 2\left[F^T g_{k+1}^T Q h_{k+1} + \mathbf{x}_0^T \left(A^{k+1}\right)^T Q h_{k+1} - \mathbf{x}_{d,k+1}^T Q h_{k+1}\right]. \tag{37}$$

Lastly, the constant terms can also be consolidated into a single variable,

$$
\begin{aligned}
\bar{k} \triangleq \sum_{k=K}^{K+N_d-1} \Big[& F^T g_{k+1}^T Q g_{k+1} F + \mathbf{x}_0^T \left(A^{k+1}\right)^T Q A^{k+1} \mathbf{x}_0 + \mathbf{x}_{d,k+1}^T Q \mathbf{x}_{d,k+1} \\
& + 2\left[F^T g_{k+1}^T Q A^{k+1} \mathbf{x}_0 - F^T g_{k+1}^T Q \mathbf{x}_{d,k+1} - \mathbf{x}_0^T \left(A^{k+1}\right)^T Q \mathbf{x}_{d,k+1}\right]\Big].
\end{aligned} \tag{38}
$$

Re-combining (36), (37), and (38) into a single cost function results in a compact representation,

$$J(\mathbf{u};\mathbf{x}_0) = \mathbf{u}^T\bar{Q}\mathbf{u} + \bar{r}\mathbf{u} + \bar{k}. \tag{39}$$

Now that the objective function is free from state terms, the state constraints also need to be reformatted without $\mathbf{x}(k)$, which also requires the lifted system matrix H, producing

$$\mathbf{x}_{\min}(k+1) \le h(k+1)\mathbf{u}(k) + g(k+1)F(k) + A^{k+1}\mathbf{x}_0 \le \mathbf{x}_{\max}(k+1), \tag{40}$$

$$\forall\, k = K, \ldots, K + N_d - 1.$$

For this strategy, a distance horizon of $2000\,\mathrm{m}$ with a distance step of $25\,\mathrm{m}$ were selected. Now this formulation has $2N_d$ decision variables, which is an order of magnitude more than MPC 1 and 2. This will be important when evaluating performance.

3.2.3 Continuous Vehicle-Following Controller

Referring back to Figure 14, there is an additional vehicle-following controller implemented in parallel with the MPC. This controller is present for MPC 1 and 2 to maintain safe vehicle following in-between time steps. However, MPC 3 cannot enforce vehicle following constraints as there is no expression for time in the distance-based dynamics. Therefore, the ACC block serves as the only mechanism for vehicle following for this strategy. The block diagram for this controller is shown in Figure 16. The following error $(e(t))$ is calculated as the difference between the current following distance $(\delta(t))$ and the minimum following constraint $(\delta_{\min}(t))$ before entering a filtered

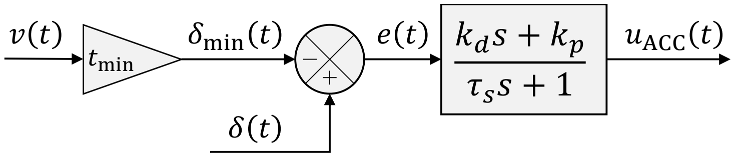


Figure 16: Continuous vehicle-following controller block diagram

PD controller to calculate the required force (u_{ACC}) that will maintain the appropriate vehicle spacing. This controller is tuned such that $u_{\text{MPC}} > u_{\text{ACC}}$ unless the minimum following distance *will* be violated.

3.2.4 Highway Controller Evaluation

The first simulation studies performed within this research investigated the efficacy of the three MPC strategies in a highway environment. Volvo Group Trucks provided data on four real-world trucking routes to use in simulations, including the terrain and speed limit profiles (detailed in Chapter 2.2.4). In addition to analyzing fuel consumption and trip times from each simulation, all results were also broken down into four critical components of energy expenditure to better understand where the fuel savings originate:

1. Mechanical/traction energy expended $(E_{\text{tract}}^{\text{tot}})$, representing positive work;
2. Energy dissipated through braking $(E_{\text{brk}}^{\text{tot}})$, representing negative work;
3. Energy lost by the vehicle due to aerodynamic drag $(E_{\text{aero}}^{\text{tot}})$;
4. Engine efficiency, quantified by the average brake-specific fuel consumption $(BSFC_{\text{avg}})$ over the entire route.

These quantities are obtained through

$$E_{\text{tract}}^{\text{tot}} = \sum_{i=1}^{N} \left[\max(F_{\text{wh}}(i), 0)\Delta x\right], \qquad E_{\text{brk}}^{\text{tot}} = \sum_{i=1}^{N} \left[\min(F_{\text{wh}}(i), 0)\Delta x\right], \tag{41}$$

$$E_{\text{aero}}^{\text{tot}} = \sum_{i=1}^{N} \left[F_{\text{aero}}(i)\Delta x\right], \qquad BSFC_{\text{avg}} = \frac{R_{\text{fuel,avg}}}{P_{\text{eng,avg}}}, \tag{42}$$

where N represents the number of simulation steps. The computational efficiency of each strategy was also analyzed by dividing the total simulation time by the number of simulated time steps (N), which provides an approximation of the average controller execution times.

The first study analyzed the MPC's performance in an open-highway setting (no traffic). The simulation results for the three MPC strategies are displayed in Figure 17, which represent reductions from the baseline simulation results (i.e., positive values are desired). In terms of fuel consumed, MPC 1 outperformed the other two strategies, followed by MPC 2 and then MPC 3. This is expected, as the objective of MPC 1 is direct fuel consumption minimization. However, when looking at the reduction in traction energy expended, MPC 3 outperformed the other two on average. This result is due to the focus of MPC 3 objective function, which minimizes the traction energy directly. Lastly, MPC 2 outperformed the other two in regards to braking effort avoided for the same reason as the first two observations, the focus of the objective function. Even though MPC 3 reduced a larger amount of work than the other two strategies, the average BSFC

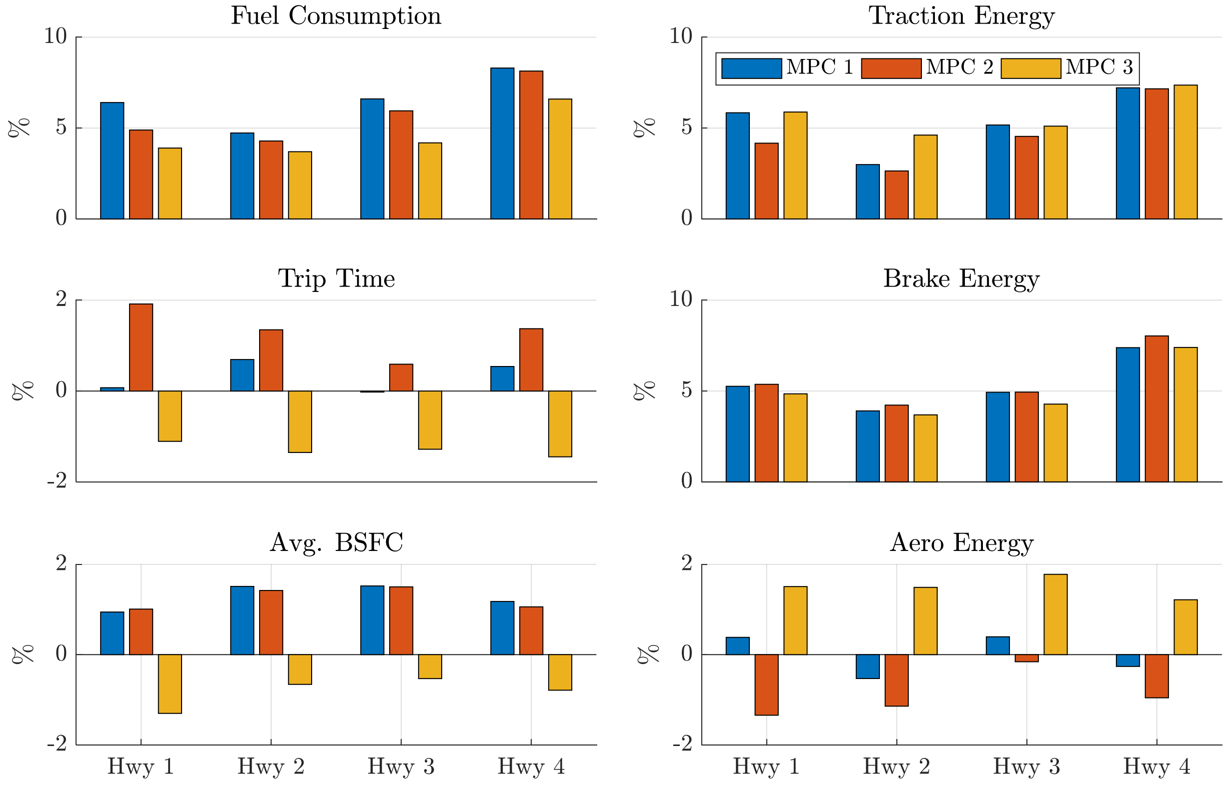


Figure 17: Open highway simulation results for three MPC strategies.

statistic shows that the engine was operating with less efficient engine torque and speed combinations, which caused greater fuel consumption. The final component of energy loss to examine is aerodynamic losses, which are tied to the trip time statistic. In order to finish the trips faster than the baseline strategy, the ego vehicle is required to travel at greater velocities on average, incurring higher aerodynamic drag loads. Therefore, further fuel savings can be realized by sacrificing trip time, if desired.

To better understand how the controllers operate, Figure 18 shows the velocity and braking torque profiles, along with the route's topology. All three MPC strategies varied the vehicle's velocity according to the terrain such that the vehicle would avoid braking along downhills, whereas the baseline maintained a consistent speed. In order to analyze simulations with similar trip times

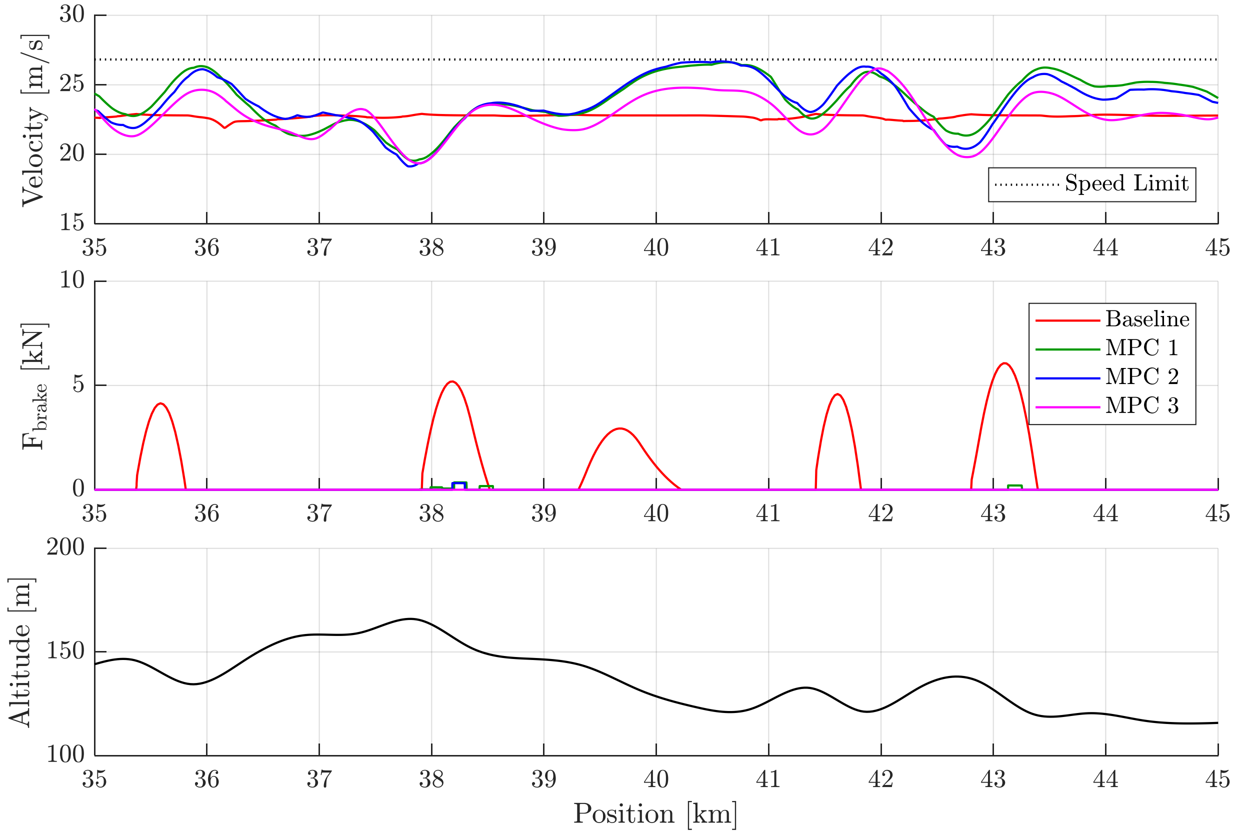


Figure 18: Open-highway (HWY 4) velocity trajectories with corresponding braking force over the terrain.

(longer trips equate to less fuel consumption), the baseline strategy's velocity setpoint is set to the fraction of the speed limit such that its and the MPC strategies' trip times match up as closely as possible.

To evaluate the impact of traffic, the first two MPC strategies were simulated over the same four routes with the stochastic lead vehicle model implemented. MPC 3 was omitted from this comparison, as the focus was on the dual-following constraint evaluation. Therefore, MPC 2 was modified to remove the penalty on vehicle following from its objective function (MPC 2*). For each route, 10 different traffic scenarios were simulated, where the fuel consumption, trip time, and energy reductions were averaged over the different traffic scenarios, which are displayed in Figure 19. The total fuel savings for both strategies improved as the inclusion of traffic clearly caused the baseline control strategy to brake more often. Similarly to the open-highway simulation results,

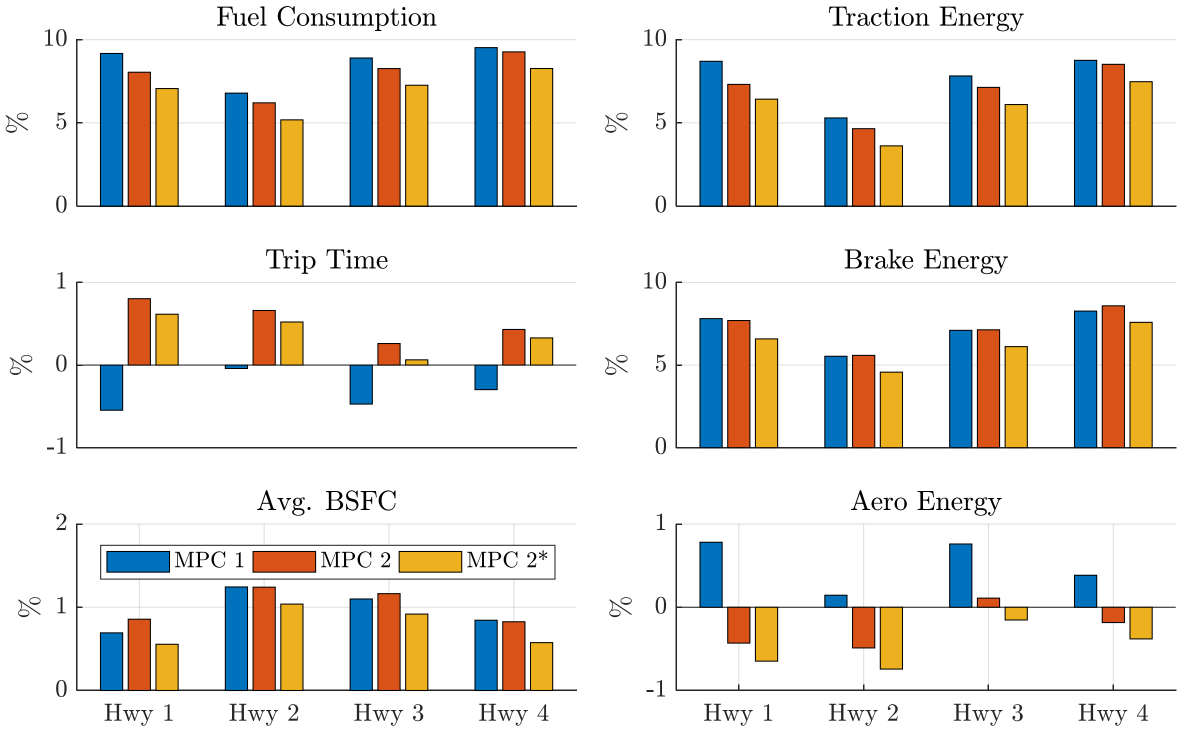


Figure 19: Highway with traffic simulation results for three MPC strategies.

the objective function that directly penalized fuel consumption did outperform the braking penalty strategy. Looking at the results where the penalty on vehicle following was removed from MPC 2's objective function, the dual-following constraints did have a positive effect. Over all four routes, an additional 1% improvement was realized. Shown in Figure 20, by allowing the ego vehicle to violate δ_{hard} for a brief moment in time, the ego vehicle can avoid unnecessary braking events. As the baseline and MPC 2*'s following distances approach δ_{soft}, they were both required to brake, wasting unnecessary energy. Over the course of a trip, these small braking events added up to that additional 1% extra in fuel savings. Any time a vehicle is required to brake, the prior acceleration event that brought the vehicle to the current speed is wasted, and ultimately needs to be redone after the braking event.

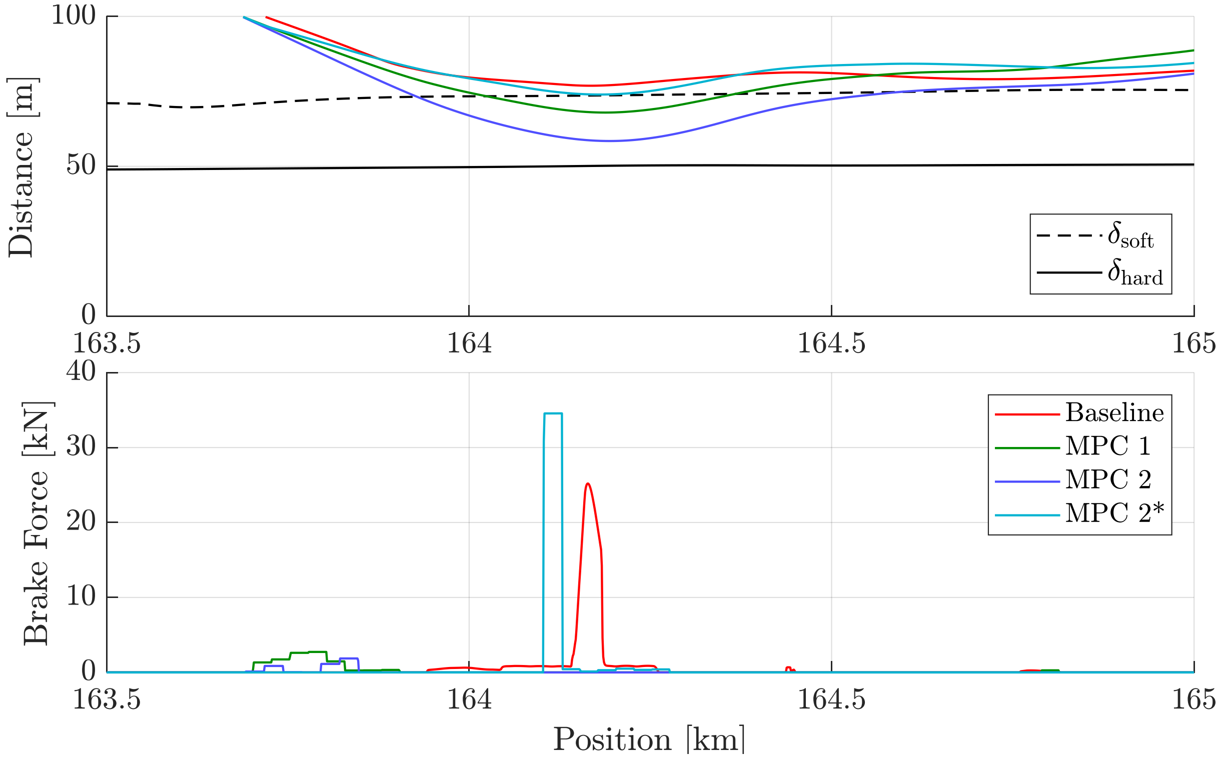


Figure 20: Inter-vehicular distance to a lead vehicle and corresponding braking force.

The last statistic used to evaluate the efficacy of all three MPC strategies is execution time. It is worth noting that these values are conservative, as they also include computations that live outside the controller. All simulations were run in Simulink on an Intel Xeon E5-2667v3 processor with 16 Gb of RAM. Table 3 shows the average execution times for the open-highway and highway with traffic simulations. With an average execution time of 410 milliseconds, MPC 1 could be feasible for an open-highway scenario where the MPC time step is 5 seconds. However, it takes almost half the MPC's time step to solve the MPC's optimization when the ego vehicle is in the presence of traffic. The other two strategies are fairly competitive, but the additional fuel savings from MPC 2 are not enough to warrant implementing an SQP-based optimal control problem for embedded purposes. Quadratic programs are much easier to implement, and solutions to do so are more available. SQP optimization solutions on the other hand are more uncommon and typically require commercial solver licenses to use. Therefore, the MPC candidate used throughout the remainder of this work is the affine distance-based MPC.

Table 3: MPC Execution Times

Open-highway		
Strategy	**Avg [ms]**	**MPC Δt [ms]**
MPC 1	410	5000
MPC 2	15	5000
MPC 3	2.4	$\approx$ 1000
Highway with Traffic		
Strategy	**Avg [ms]**	**MPC Δt [ms]**
MPC 1	417	1000
MPC 2	84	1000

3.3 Urban Environment

Realizing that heavy-duty trucks do travel through urban and suburban environments as well, the highway controller was adapted into the h-GLAS control strategy. For urban environments, signalized intersections are assumed to use fixed-time signal operation. The signal phase and timing (SPaT) information provided through vehicle-to-infrastructure (V2I) communication includes the intersection locations $(\mathbf{x}_L)$, cycle lengths $(\mathbf{T}_L)$, and green signal phase lengths (ϕ_g). Additionally, the current signal cycle start time is also provided so that the controller can extrapolate future SPaT from, since it is deterministic. In order to adapt the highway controller into the h-GLAS controller, the offline DP solution is replaced with an online *velocity planning* algorithm, where the control block diagram is displayed in Figure 21. Both the desired velocity profile $(\mathbf{v}_{\text{des}})$ and the environmental resistance profile $(\mathbf{F}_{\text{env}})$ are sent to the MPC from the velocity planning block. The only changes to the QP-based MPC for urban environments are the horizon length and distance step, which are now 500 and 10 meters, respectively. For vehicle following, the continuous vehicle-following controller from Chapter 3.2.3 is implemented in parallel with the MPC.

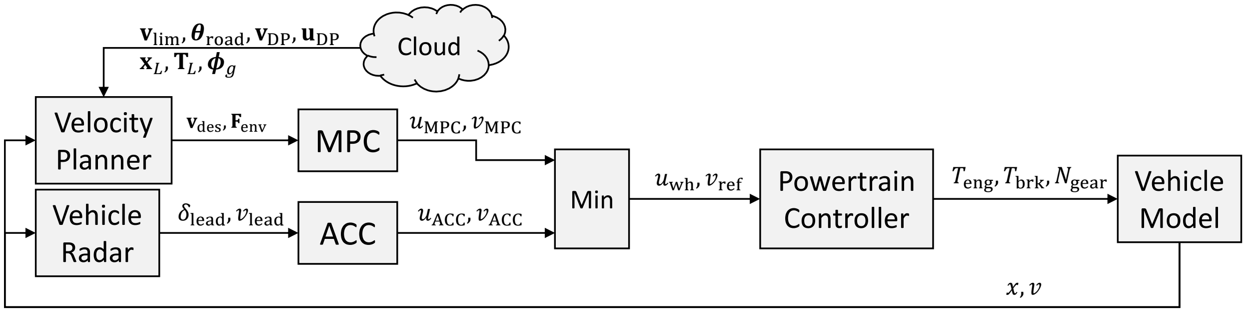


Figure 21: Urban controller block diagram.

3.3.1 Desired Velocity Construction

The objective of the desired velocity planning algorithm (Velocity Planning block) depends on the vehicle's proximity to upcoming intersections. For the scenario where there are no upcoming intersections within the MPC horizon length, the desired velocity is simply set to the speed limit $(\mathbf{v}_{\text{lim}})$ over the MPC horizon, understanding that the MPC will make alterations to account for the terrain. For the other scenario (intersections within MPC horizon), the objective is to use the provided SPaT information to construct $\mathbf{v}_{\text{des}}$ such that the vehicle will arrive at each intersection at the time that allows the vehicle to best maintain its kinetic energy (arrive at a green signal). This was originally motivated after seeing the correlation between fuel savings and braking work avoided in the highway simulation studies. At each step along the route, the velocity planning algorithm implements a *time-optimal* solution as it selects the first feasible green window of each intersection to pass through, within the MPC horizon.

Through V2I communication, the position of each upcoming signalized intersection (x_L^m) is known. Therefore, the distance between the vehicle and the first intersection (δ_L^1) is known, as well as the distances between all other intersections $(\delta_L^m,\ \forall\, m = 2, \ldots, M)$ within the MPC horizon. The superscript m represents the intersection number in a sequence of M upcoming intersections. Fixed-time operation of signalized intersections allows the controller to also know when future green signal phases will start (g_n^m) and end (r_n^m). The subscript n represents the cycle number for the given intersection. As this is a time-optimal solution, the candidate crossing time $\big(t_*^m(n)\big)$ for each green window is set equal to g_n^m.

The desired velocity profile to be constructed is position-based and consists of N_d elements, which are indexed by k (representing positions along the horizon) and separated by Δx. Construction of the desired velocity profile begins with the first upcoming intersection within the MPC horizon, then moves on to the next intersection, and so on. For the first upcoming intersection, the velocity to reach each $t_*^1(n)$ is calculated as

$$v_*^1(n) = \frac{\delta_L^1}{t_*^1(n) - t(k)}, \qquad \forall\, n = 1, \ldots, N_f. \tag{43}$$

where N_f represents the number of future green windows to consider. Each $v_*^1(n)$ (beginning at $n = 1$) is compared to a threshold speed limit, $v_{\text{lim}}^*(k)$, where the first $v_*^1(n)$ satisfying $v_*^1(n) \leq$

$v_{\mathrm{lim}}^{*}(k)$ is selected as the corresponding target velocity, $v_{\mathrm{des}}(k)$, for all $k \leq x_L^1$. For the remaining intersections within the MPC horizon, the same search method is applied; however, the calculation of the velocity to reach each $t_*^m(n)$ relies on $t_*^{m-1}(n)$ corresponding to the selected $v_*^{m-1}(n)$:

$$v_*^m(n) = \frac{\delta_L^m}{t_*^m(n) - t_*^{m-1}}, \qquad \forall\, n = 1, \ldots, N_f, \text{ and } \forall\, m = 2, \ldots, M. \tag{44}$$

Each $v_*^m(n)$ is then compared to $v_{\mathrm{lim}}^{*}(x_L^{m-1})$ to obtain the values of $v_{\mathrm{des}}(k)$, for all $x_L^{m-1} < k \leq x_L^m$. This is repeated until $m = M$. The scenario where $v_*^m(n) > v_{\mathrm{lim}} \cap v_r^m(n) \leq v_{\mathrm{lim}}$ can exist; therefore, the desired velocity is set to $v_*^m(n) = v_{\mathrm{lim}}$ for these cases.

A graphical representation for the green window selection over two intersections is shown in Figure 22. Beginning at the current position and time ($x(K)$ and $t(K)$), desired velocity candidates for the first intersection are calculated ($v_*^1(n)$) and compared to the speed limit (v_{lim}). The candidate desired velocity value of $v_*^1(3)$ is the first feasible velocity, which is set to v_{des} up until the first intersection. The desired velocity profile from the first intersection to the next begins with the position and time of the first intersection's crossing (x_L^1 and $t_*^1(3)$). Comparing the second set of velocity candidates to the speed limit shows only one feasible velocity candidate, $v_*^2(5)$. However, the prior green window can still be reached under the speed limit, so the value of v_{des} is set equal to v_{lim} for this portion of the horizon.

Once $\mathbf{v}_{\mathrm{des}}$ has been constructed, itself, the environmental resistance profile ($\mathbf{F}_{\mathrm{env}}$), and weighting matrices (Q and R) are sent to the QP-based MPC. The QP objective function weighting matrices are not static values, and are adjusted according to the ego vehicle's proximity to an intersection. Referring back to Chapter 3.2.2.4, Q is a scalar, for the single state variable, and R is a 2×2 diagonal matrix, for the two control inputs. Their values are selected according to the

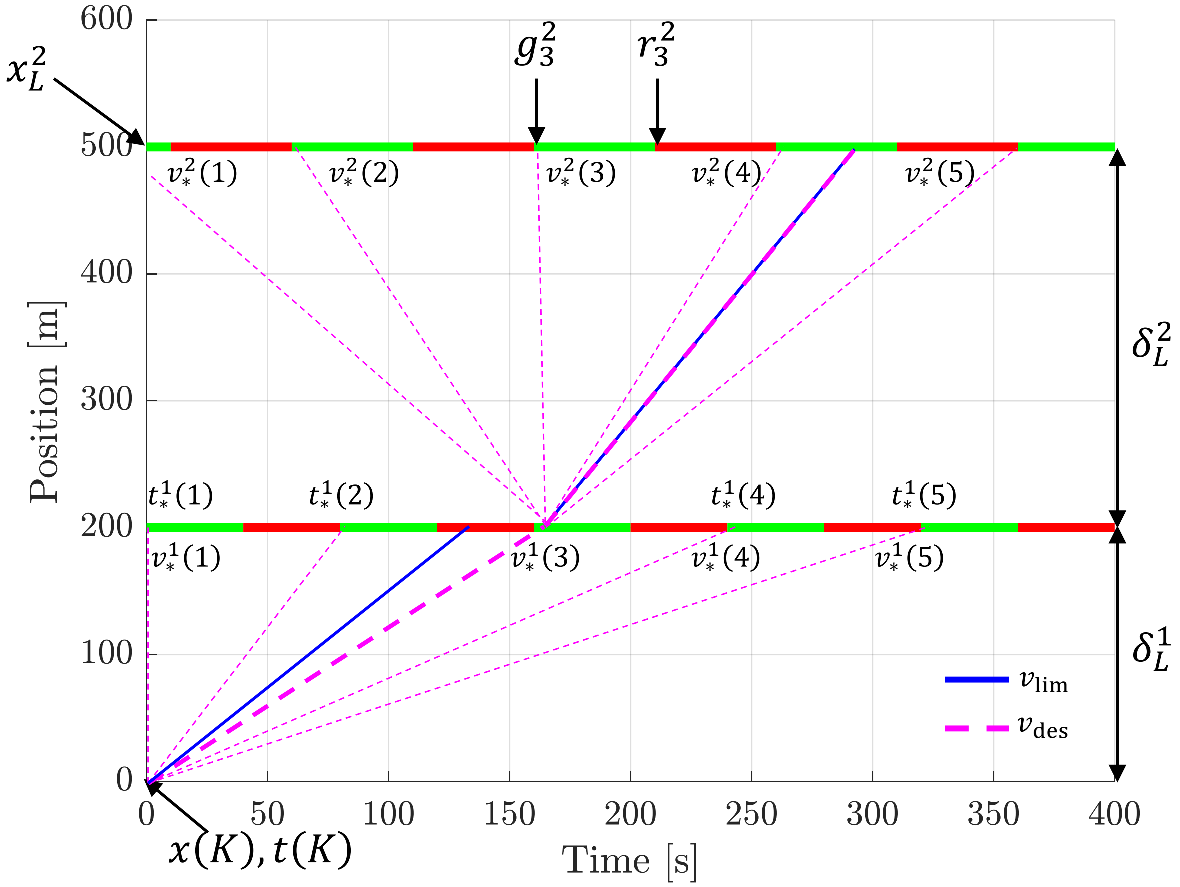


Figure 22: Desired velocity profile construction over two intersections.

following piece-wise functions:

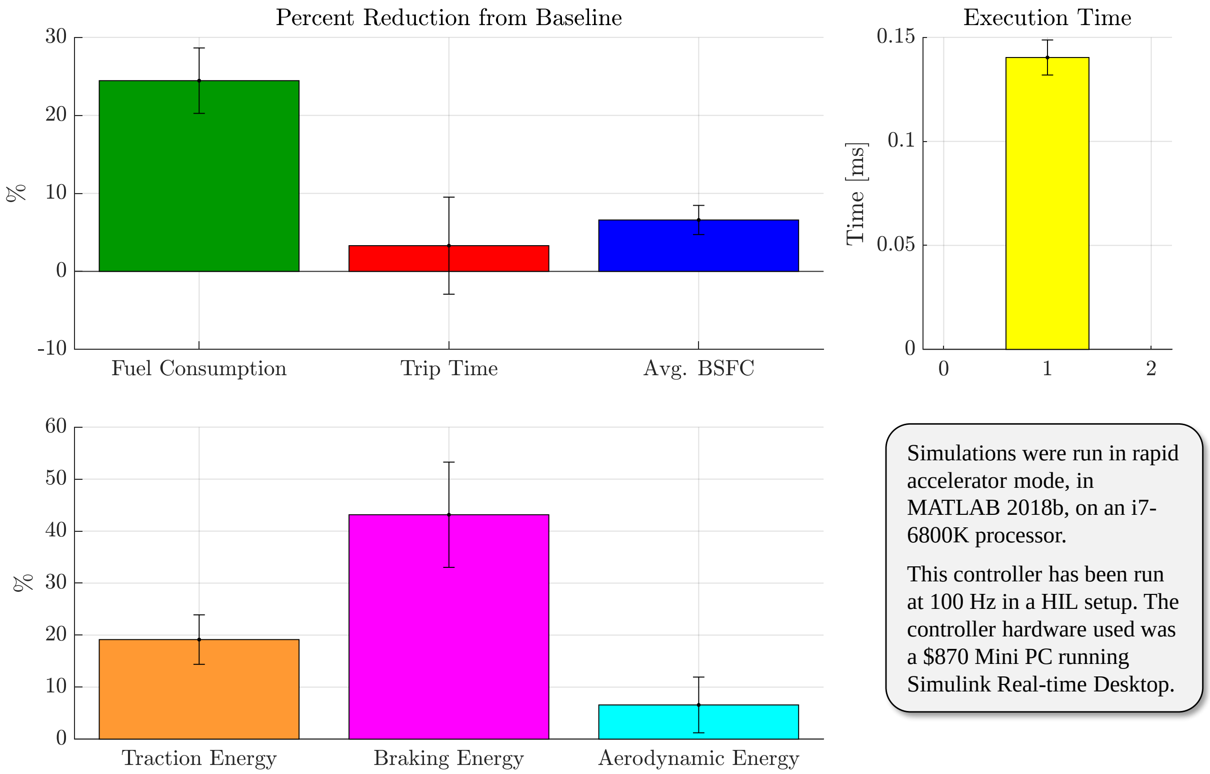
$$Q = \begin{cases} 10, & \delta_L^1 < 150\,\mathrm{m} \\ 1, & 150\,\mathrm{m} \leq \delta_L^1 \leq 250\,\mathrm{m} \\ 0.1, & \delta_L^1 > 250\,\mathrm{m} \end{cases}, \tag{45}$$

$$R_{1,1} = \begin{cases} 1, & \delta_L^1 < 150\,\mathrm{m} \\ 1, & 150\,\mathrm{m} \leq \delta_L^1 \leq 250\,\mathrm{m} \\ 1, & \delta_L^1 > 250\,\mathrm{m} \end{cases}, \qquad R_{2,2} = \begin{cases} 10, & \delta_L^1 < 150\,\mathrm{m} \\ 100, & 150\,\mathrm{m} \leq \delta_L^1 \leq 250\,\mathrm{m} \\ 1000, & \delta_L^1 > 250\,\mathrm{m} \end{cases}. \tag{46}$$

As the vehicle approaches an intersection, the importance of following $\mathbf{v}_{\text{des}}$ (encoded by Q) relative to minimizing braking energy expenditure (encoded by $R_{2,2}$) is reversed. The weight on minimizing traction energy expenditure (encoded by $R_{1,1}$) is left at a value of 1, regardless of the ego vehicle's proximity to an intersection. By incorporating variable weighting matrices, the MPC has more flexibility to vary the vehicle's velocity, in an effort to minimize energy expenditure, when tracking $\mathbf{v}_{\text{des}}$ is not as critical.

3.3.2 Urban Controller Evaluation

The route used to validate the urban h-GLAS controller was provided by Volvo Group Trucks, and consists of 27 fixed-time signalized intersections over a 30 km stretch. Using the signal timing model from Chapter 2.2.2.1, 20 different timing combinations of the cycle and green signal phase lengths were generated to test the robustness of the urban controller. Additionally, each of the 20 timing scenarios were simulated 5 times, where the lead vehicle trajectories were also varied from one simulation to another, providing 100 simulations in total. Similarly to the highway simulation studies, along with the fuel consumption and trip time comparison, the results were broken down into the four critical components of energy expenditure, which are displayed in Figure 23. After simulating the 100 cases, fuel economy improvements of 24.45% were observed on average, ranging between 20 and 30%. Because the signal timing information is deterministic, and the velocity planning algorithm constructs a time-optimal trajectory for the MPC to reference,



Simulations were run in rapid accelerator mode, in MATLAB 2018b, on an i7-6800K processor.

This controller has been run at 100 Hz in a HIL setup. The controller hardware used was a $870 Mini PC running Simulink Real-time Desktop.

Figure 23: Urban environment simulation results. The error bars reflect a 95% confidence interval.

the h-GLAS strategy finished the route faster than the baseline by 3.29% on average. However, the h-GLAS strategy incurred less aerodynamic resistance overall, which is the opposite from what was observed in the highway simulation studies. This is a direct result of the presence of signalized intersections. The baseline strategy would travel faster than the h-GLAS strategy on average, but red signals would require the baseline strategy to come to a stop, shown in Figure 24. Since aerodynamic energy is computed by integrating F_{aero} over distance, the baseline incurred more aerodynamic resistance. If it weren't for the intersections, the baseline strategy would have finished far ahead of the h-GLAS strategy. The h-GLAS strategy was also able to outperform the baseline strategy in regard to average brake-specific-fuel-consumption experienced over the whole route. Improvements in average BSFC basically come down to better engine torque and speed combinations used to realize the appropriate wheel force desired by the controllers. The baseline

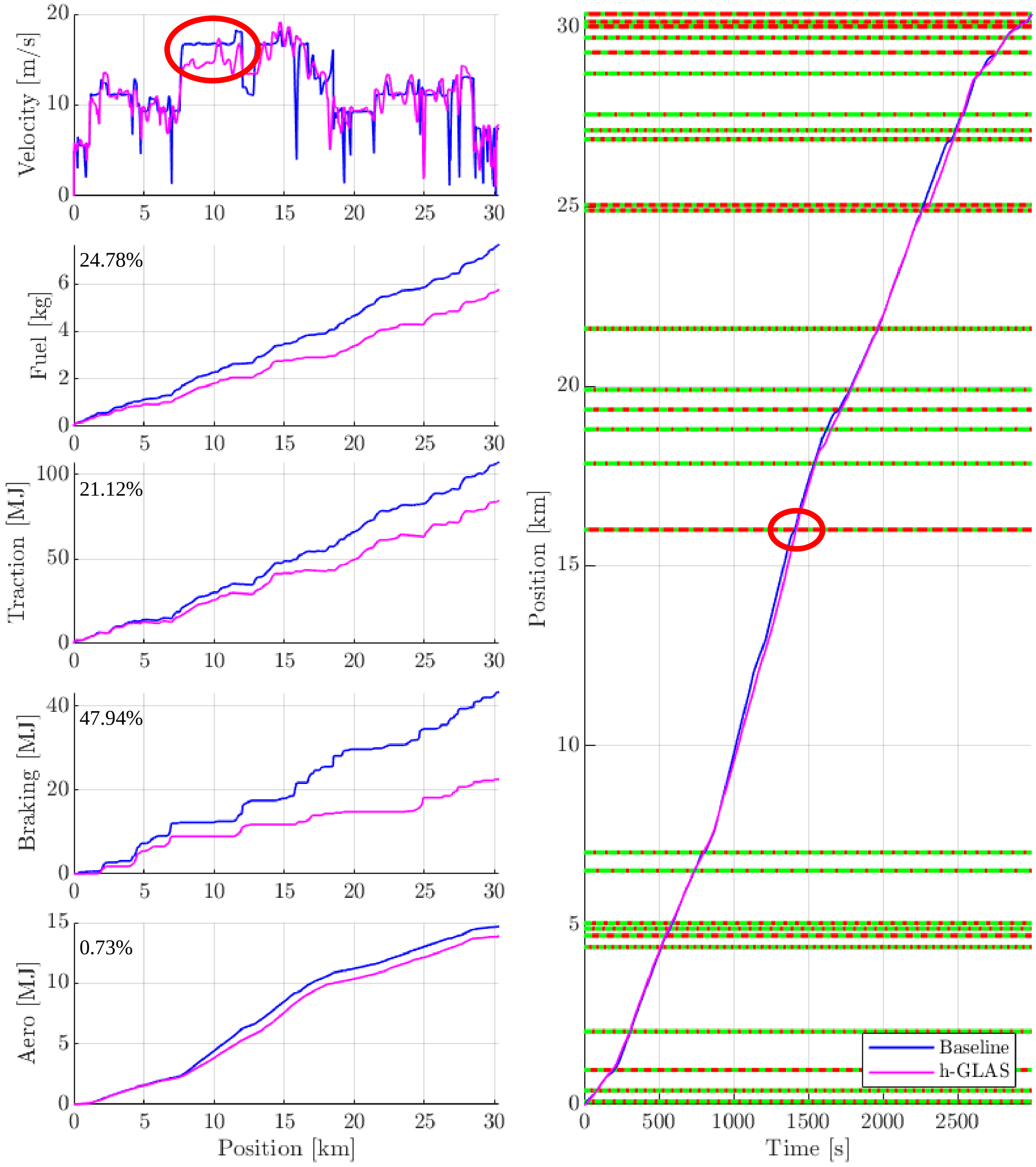


Figure 24: Simulation trajectories and cumulative energy plots.

strategy experienced many more idling events than the h-GLAS strategy, which occurred due to the required stops for red traffic signals.

Looking at the h-GLAS strategy's average execution time of 0.14 ms, real-time feasibility can easily be achieved. As a part of the ARPA-E Nextcar project, the h-GLAS strategy was one component of a more complex hierarchical control formulation. Hardware-in-the-loop and Engine-in-the-loop testing of this control formulation on a Mini PC, running Simulink Real-time Desktop, demonstrated the h-GLAS strategy's ability to successfully run at 100 Hz on dedicated hardware [54].

CHAPTER 4: H-GLAS CONTROL THROUGH SUBURBAN ENVIRONMENTS

This chapter focuses on a core contribution of this research, which is the h-GLAS controller's adaptation for suburban environments and the evaluation thereof. This includes the SPaT forecast model, adaptations to the velocity planning algorithm to account for signal timing uncertainty, and the command governor from [55], implemented for efficient vehicle following. Lastly, two simulation studies were performed to: 1) understand how actuated signalized intersections impact fuel consumption performance alone and 2) understand how the inclusion of traffic changes performance.

Like the other two environments, communication between the vehicle and infrastructure is still a fundamental component of this control formulation; therefore the ego vehicle is assumed to be equipped with the necessary equipment to request/receive SPaT and route information (DSRC or 5G). Shown in Figure 25, the current vehicle states $(x$ and $v)$ are sent to the cloud and in return, the h–GLAS controller (inside the blue dashed lines) receives road properties and signal timing information. Specifically, these include the speed limit and road grade $(\mathbf{v}_{\text{lim}}$ and $\theta_{\text{road}})$ as functions of position and past SPaT information $(SPaT_{\text{hist}})$ for all upcoming intersections within the MPC's prediction horizon. The *SPaT Forecast* block contains a forecast model that uses $SPAT_{\text{hist}}$ to predict the future SPaT information $(SPaT_{\text{est}})$. The *Velocity Planning* block uses $\mathbf{v}_{\text{lim}}$, θ_{road}, and $SPaT_{\text{est}}$ along with the vehicle states $(x$ and $v)$ to construct a desired velocity profile $(\mathbf{v}_{\text{des}})$ that navigates through upcoming intersections in a safe and efficient manner, corresponding to the time at which the probability of a green signal is maximized. The MPC uses $\mathbf{v}_{\text{des}}$ and the environmental resistance energy profile $(\mathbf{E}_{\text{env}})$ to compute the next optimal wheel force command (u_{MPC}) and its corresponding reference velocity (v_{MPC}) to be used downstream by the CG. To deal with vehicle following constraints, the CG uses u_{MPC}, v_{MPC}, x, v, and the traffic radar information $(x_{\text{lead}}$ and $v_{\text{lead}})$ to determine if the safe vehicle following constraints will be violated. If so, u_{MPC} is adjusted to u_{CG} to ensure safe vehicle following; otherwise $u_{\text{CG}} = u_{\text{MPC}}$. Finally, the powertrain controller takes u_{CG} and its corresponding reference velocity (v_{CG}) and finds the optimal gear ratio using equation (10) (the same shifting strategy is used by the baseline controller). This optimal gear ratio is then used to compute the necessary engine and brake torque $(T_{\text{eng}}$ and $T_{\text{brk}})$ from u_{CG} to drive the vehicle model.

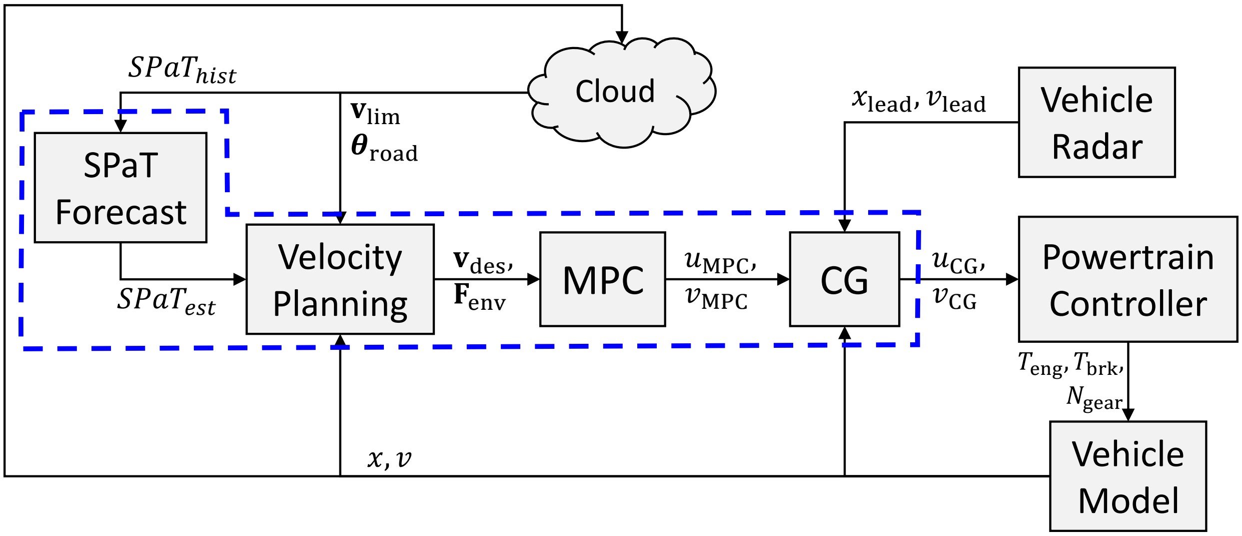


Figure 25: Control System Block Diagram.

The upcoming sections will detail the SPaT information forecast (GP model), the desired velocity profile construction, and the MPC and CG formulations, followed be the evaluation of the suburban controller.

4.1 Signal Phase and Timing Forecast

The SPaT forecast requires historical SPaT information to predict future SPaT information. The following SPaT information, for each upcoming intersection within the ego vehicle's MPC horizon length, is assumed to be received through V2I communication:

- The signal controller parameters,
- The intersection locations,
- The current signal phase color and how much time has passed since the phase started,
- Beginning and end timestamps for the signal phase over a desired time interval to infer green and red phase lengths from.

Because future SPaT information is unknown, a mechanism for predicting the future signal timing is required. To accomplish this task, a Gaussian Process (GP) model is considered first, as it is a popular choice for non-parametric regression in machine learning. The goal of a GP model is to obtain an approximation of a complex map from an input vector (generically given by $\mathbf{n}$) to output vector (generically given by $\phi(\mathbf{n})$), which is assumed to have a joint Gaussian distribution [56].

For this application, the GP framework will be used to predict the future green and red signal phase lengths $\big(\phi_{\text{green}}(n)$ and $\phi_{\text{red}}(n)\big)$ for future cycle numbers $(n \in \mathbf{n}_f)$, based on a set of N_p past observations $\mathbf{n}_p = \{1 : N_p\}$. To simplify the notation for the presentation of the GP model formulation, the green/red subscripts from ϕ will be dropped, as the GP model is structured identically for each variable. Because GP models in general assume the output variable to have zero mean [56], a shifted output variable is defined by subtracting the observed mean $(\mu_{\mathbf{n}_p})$ from each value:

$$\bar{\phi}(n) = \phi(n) - \mu_{\mathbf{n}_p}, \;\; \forall\, n \in \mathbf{n}_p. \tag{47}$$

After the GP prediction, $\mu_{\mathbf{n}_p}$ is then added to the predicted mean values of $\bar{\phi}(n)$ to obtain the actual predicted future signal phase lengths.

The shifted signal phase length for future cycle numbers $\big(\bar{\phi}(\mathbf{n}_f)\big)$ is modeled to follow a multivariate Gaussian distribution [56]:

$$\begin{bmatrix} \bar{\phi}(\mathbf{n}_p) \\ \bar{\phi}(\mathbf{n}_f) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\mathbf{n}_p, \mathbf{n}_p) & K(\mathbf{n}_p, \mathbf{n}_f) \\ K(\mathbf{n}_f, \mathbf{n}_p) & K(\mathbf{n}_f, \mathbf{n}_f) \end{bmatrix}\right). \tag{48}$$

In equation above, the $K(\cdot,\cdot)$ terms represent the covariance matrices for past/past, past/future, and future/future cycle number pairs (the subscripts on $\mathbf{n}$ differentiate past vs future cycles). Individual elements of the covariance matrices $\big($i.e. $k(n_i, n_j)\big)$ encode the correlation between two different cycle numbers and are computed through a *covariance kernel*. In this work, the squared exponential (SE) kernel function is used to represent the elements of the covariance matrix. For two cycle numbers, n_i and n_j, the SE kernel is defined as

$$k(n_i, n_j) = \sigma_0^2 \mathrm{exp}\left(-\frac{1}{2}\lambda^{-2}(n_i - n_j)^2\right) + \beta\delta_{ij}. \tag{49}$$

where δ_{ij} represents the Kronecker delta operator:

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases} \tag{50}$$

The SE covariance kernel reflects the reality that the signal timing properties of two different cycle numbers are more closely correlated when the cycle numbers are close to each other. The quantitative correlation between different data points is characterized by *hyperparameters*, which are denoted by $\theta = \{\sigma_0, \lambda, \beta\}$ and represent the signal variance, length scale, and noise variance, respectively. The signal variance determines variation of the function value from its mean. The length scale describes how smooth a function is by describing how quickly it tends to vary with respect to the independent variable. With small length scales, function values can change quickly and for large length scales, function values will change slowly. Lastly, the noise variance is formally not part of the SE kernel itself, but is added to allow for noise in the training data [57]. For numerical reasons, the noise variance term is usually included even in the absence of noise within the training data. The eigenvalues of the covariance matrix can decay very rapidly, and without the noise variance term acting as a stabilizing factor, the covariance matrix cannot be inverted [56]. In this work, the value of σ_0 has been set to the standard deviation of $\bar{\phi}(\mathbf{n}_p)$, and the values of λ and β have been identified as 1 and 0.5, respectively, based on available training data. This implies that the signal phase lengths are not highly correlated from cycle to cycle and can change quickly. Ultimately, the GP model provides a straightforward mechanism for characterizing prediction mean and prediction variance (the conditional variance of modeling error, which acts as a measure of uncertainty), which are given by:

$$\bar{\phi}_{\mathbf{n}_f} = K(\mathbf{n}_f, \mathbf{n}_p)K(\mathbf{n}_p, \mathbf{n}_p)^{-1}\bar{\phi}(\mathbf{n}_p), \tag{51}$$

$$\sigma^2_{\mathbf{n}_f} = K(\mathbf{n}_f, \mathbf{n}_f) - K(\mathbf{n}_f, \mathbf{n}_p)K(\mathbf{n}_p, \mathbf{n}_p)^{-1}K(\mathbf{n}_p, \mathbf{n}_f), \tag{52}$$

where the diagonal elements of $\sigma^2_{\mathbf{n}_f}$ represent the variance for each element of $\bar{\phi}(\mathbf{n}_c)$. Finally, by adding $\mu_{\mathbf{n_p}}$ to $\bar{\phi}_{\mathbf{n}_f}$, the actual predicted future signal phase lengths $\big(\phi_{\text{green}}(\mathbf{n}_f)$ and $\phi_{\text{red}}(\mathbf{n}_f)\big)$ are obtained.

To evaluate the value of a GP-based forecast model, a simpler model is also considered. In this forecast model, the predicted mean is set equal to the mean value of the past signal phases ($\bar{\phi}_{\mathbf{n}_f} = \mu_{\mathbf{n}_p}$), and the predicted variance is set equal to the variance of the past signal phases ($\sigma^2_{\mathbf{n}_f}$). This forecast model will be referred to as *Average-based.*

4.2 Velocity Planning Adaptation for Actuated Signal Operation

The forecasted SPaT information includes timing uncertainty that needs to be considered within the velocity planning algorithm. Therefore, knowledge of the *expected* time at which the signal phase will be green is required. The same velocity profile construction algorithm from Chapter 3.3.1 will be implemented, so this section will just cover the expected green signal phase calculation used to obtain $t^m_*(n)$.

The SPaT forecast model ultimately produces four vectors of length N_f (number of future predictions) for each upcoming intersection:

- Predicted mean green and red signal phase lengths (ϕ_{green} and ϕ_{red});
- Corresponding prediction variances (σ^2_{green} and σ^2_{red}) for each predicted mean.

Through V2I communication, it is assumed that the start time for the current cycle $\big(t_g(1)\big)$ is also received, along with the past SPaT information. At the current time corresponding to the MPC step $\big(t(k)\big)$, if the time left in the current signal phase (green or red) is less than the yellow phase length, it is assumed that the current phase length is known and therefore does not need to be predicted. This holds true in practice as yellow phase lengths are fixed; therefore, once the yellow signal is displayed, a countdown to the next signal phase has begun. Similarly to the urban controller, each index k refers to a particular vehicle position, as the MPC is position-based.

The predicted green signal phase start time at a given cycle $\big(t_g(n)\big)$ is calculated as the sum of the predicted signal phases lengths $\big(\phi_{\text{green}}(i)$ and $\phi_{\text{red}}(i)\big)$ up to cycle n. Therefore, the predicted start time for each *green* phase, at any subsequent time step $n+1$, can be calculated as:

$$t_g(n+1) = t_g(n) + \phi_{\text{green}}(n) + \phi_{\text{red}}(n), \qquad \forall\, n = 1, \dots, N_f - 1. \tag{53}$$

Because of the very small temporal correlation between successive signal phase timings, as found when identifying hyper-parameters for the GP model, the prediction variance at any subsequent time step $n+1$ can be closely approximated through a direct summation, neglecting covariance terms that have been observed to be small:

$$\sigma_g^2(n+1) = \sigma_g^2(n) + \sigma_{\text{green}}^2(n) + \sigma_{\text{red}}^2(n), \qquad \forall\, n = 1, \dots, N_f - 1. \tag{54}$$

The predicted start times for each *red* signal phase (t_r) are obtained by adding ϕ_{green} to each t_g. The values for t_r and its corresponding variance (σ_r^2) are then calculated as

$$t_r(n) = t_g(n) + \phi_{\text{green}}(n), \qquad \forall\, n = 1, \dots, N_f, \tag{55}$$

$$\sigma_r^2(n) = \sigma_g^2(n) + \sigma_{\text{green}}^2(n), \qquad \forall\, n = 1, \dots, N_f. \tag{56}$$

Even though the random variables, ϕ_{green} and ϕ_{red}, are loosely correlated, by assuming they are independent, the values of σ_g^2 and σ_r^2 will represent close approximations for the variance of the start times of each future signal phase.

Once the predicted start times for each signal phase (green and red) and the corresponding variances are obtained, the probability distribution for each future signal green phase (beginning and end) is known. Because $t_g(n)$ and $t_r(n)$ are predicted values that include some uncertainty, they are adjusted according to a 95% confidence interval (CI), which is approximated closely by the predicted mean plus/minus two standard deviations:

$$\bar{t}_g(n) = t_g(n) + 2\sigma_g(n), \qquad \forall\, n = 1, \ldots, N_f, \tag{57}$$

$$\bar{t}_r(n) = t_r(n) - 2\sigma_r(n), \qquad \forall\, n = 1, \ldots, N_f. \tag{58}$$

There exists the possibility that $\bar{t}_r(n) \leq \bar{t}_g(n)$, which implies that the predicted probability of a green signal phase over the entirety of signal cycle n is less than 0.95. For these cases, one must look to the cumulative distribution functions (CDFs) for $t_g(n)$ and $t_r(n)$ to identify the time at which the probability of a green signal phase is maximized for each cycle where $\bar{t}_r(n) \leq \bar{t}_g(n)$. The CDF for $t_g(n)$ represents the probability of the signal *turning* green, where the CDF of $t_r(n)$ represents the probability of the signal *remaining* green. Illustrated in Figure 26, the CDF for each $t_g(n)$ and $t_r(n)$ is characterized through $\sigma_g(n)$ and $\sigma_r(n)$, respectively. By centering the CDFs on their respective mean values, the point at which they intersect represents the point at which the probability of a green signal is maximized. Note that the CDFs for t_r are flipped in Figure 26, as we are interested in the probability of the green signal remaining green and not the probability of when the signal will turn red. These intersection points (during a predicted green window) are assigned to $\bar{t}_g(n)$ and $\bar{t}_r(n)$.

Now that the times at which the signal phase will be green with some level of confidence have been identified, the h-GLAS strategy selects $\bar{t}_g(n)$ as the target intersection crossing time, $\big(t_*^m(n)\big)$, for each future signal cycle n and intersection m within the MPC horizon, similarly to the strategy for fixed-time signal operation.

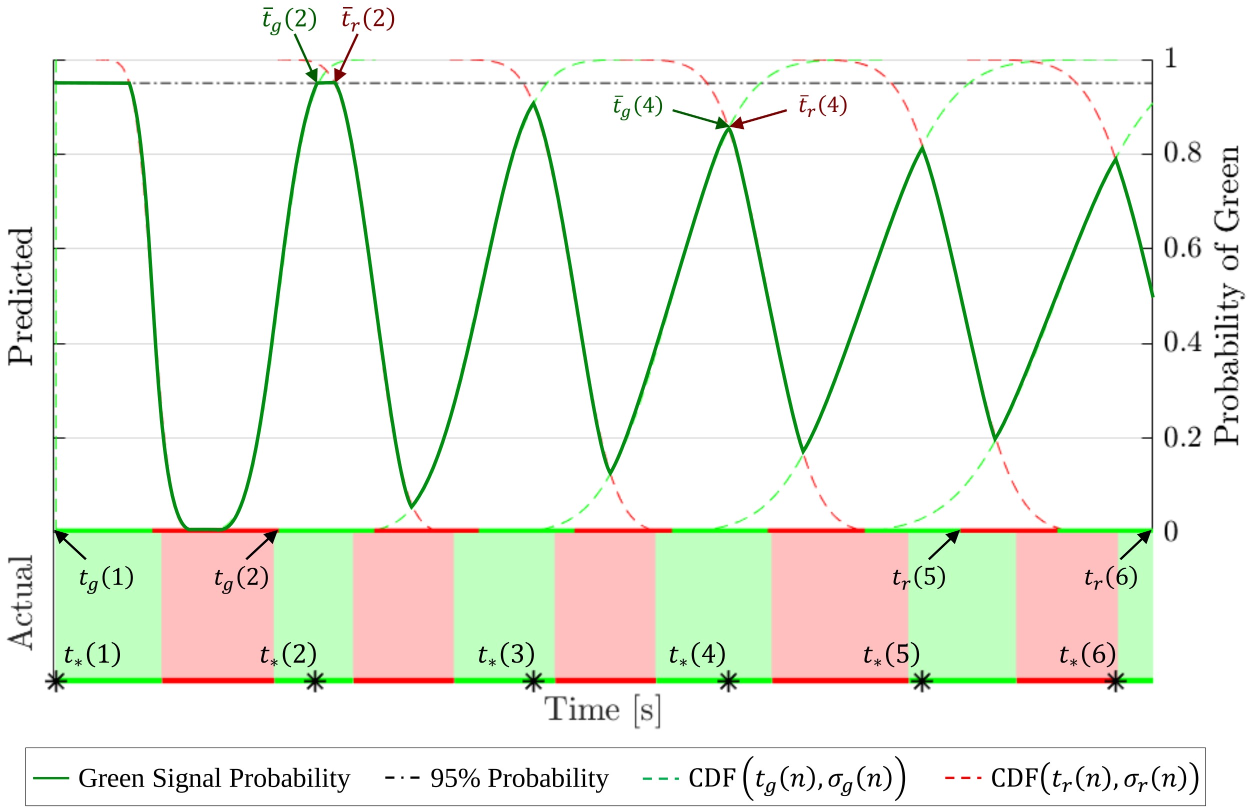


Figure 26: Predicted future signal phase timing. The start of each signal phase ($t_g(n)$ and $t_r(n)$) has its own probability distribution characterized through a cumulative distribution function (CDF) *centered* on $t_g(n)$ and $t_r(n)$ respectively. Here we show the CDF for each $t_g(n)$ and $t_r(n)$ as dashed green and red lines, where $n = 1, \ldots, 6$. Note that the CDF for $t_r(n)$ is flipped (goes from 1 to 0), as this represents the probability distribution for the start of a red signal phase. Therefore, the probability of the signal phase remaining green will be falling. Combining all of the CDF curves, along with the 95% probability line, we obtain the probability of a green signal curve over all future cycles, illustrated by the solid green curve. It's important to note that for each future cycle, the maximum probability that the signal will be green decreases. For this example, the probability of a green signal for cycles 3-6 is less than 95% over the entire green window. The actual future signal phase timing is shown at the bottom with the solid shaded regions extending up to the predicted timing to show the efficacy of the forecast model. Lastly, the target crossing times ($t_*(n)$) are superimposed over the actual timings along the bottom.

4.3 Command Governor Formulation

Since the MPC cannot enforce vehicle-following constraints (or ensure red signal adherence) and the vehicle-following controller from Chapter 3.2.3 is not optimization-based, the h-GLAS strategy implements the CG from [55] to maintain safe vehicle following and ensure that all red signals are obeyed. The goal here is to implement an *efficient* vehicle-following controller in regard to fuel consumption, which is accomplished with the CG.

Traditionally, CGs are used to enforce constraints by modifying the reference input to an already stable system. In particular, CGs modify the reference to a value that, if held constant, would satisfy all constraints over all future time. The set of all originating states and constant inputs that satisfy constraints is commonly referred to as the *maximal output admissible set*. In this work, the control strategy places the CG downstream from the MPC for enforcing vehicle-following and red signal constraints, as originally proposed in [55], over a finite time-based horizon. The CG implements a truncated analogy of the maximal output admissible set, denoted as O_T, which is the set of all originating states $\big(\mathbf{x}(i)\big)$ and inputs (u_{CG}) such that if the input is held constant over a finite prediction horizon, the constraints are satisfied. The finite horizon is chosen to reasonably reflect the future horizon over which the lead vehicle's velocity profile can be predicted. The CG receives the MPC commanded wheel force (u_{MPC}) as an input and outputs an adjusted wheel force (u_{CG}) whenever an adjustment is needed to avoid a constraint violation. The CG strategy is formulated as follows:

$$u_{\text{CG}}(i) = \arg\min_{u_{\text{CG}}} \quad \|u_{\text{CG}} - u_{\text{MPC}}(i)\|^2, \tag{59a}$$

$$\text{subj. to:} \quad \{\mathbf{x}(i), u_{\text{CG}}\} \in O_T, \ \ \forall\, i = I, \ldots, I + N_{\text{CG}} - 1, \tag{59b}$$

where N_{CG} represents the number of time steps within the CG horizon. The CG adjusts $u_{\text{MPC}}(i)$ only when necessary, and only by the smallest amount required for safe vehicle following. If no constraint violation occurs, $u_{\text{CG}}(i) = u_{\text{MPC}}(i)$. If a constraint violation occurs over the CG horizon, the minimization in equation (59) is performed via bisection search. The maximum correction the CG can apply is $u_{\text{CG}} = u_{\text{max,brk}}$, where $u_{\text{max,brk}}$ is the maximum *negative* force that can be applied to the road through the wheel via braking. The constraints are checked by marching equations (7) and (8) over the CG horizon and checking the following distance at each time step.

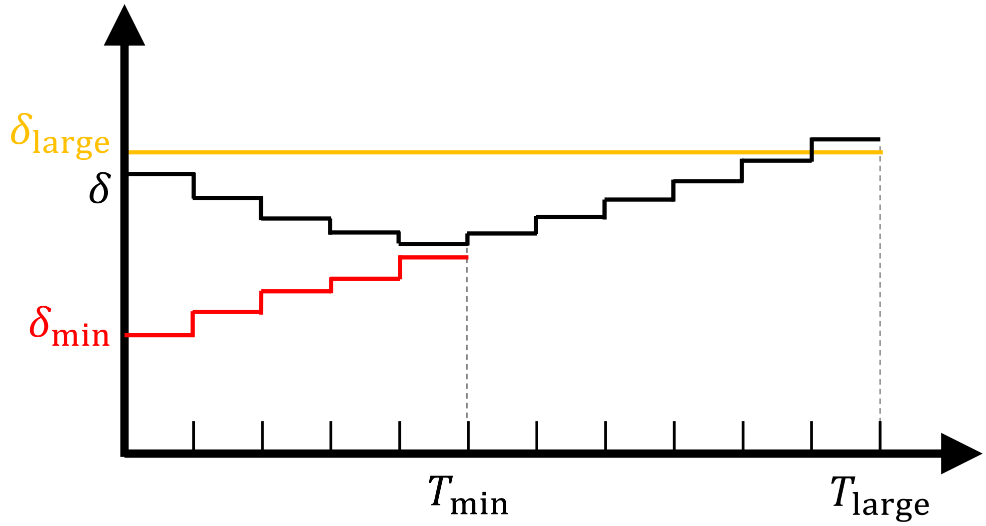


Figure 27: CG following distance illustration. The following distance (δ) must be greater than δ_{min} up to T_{min}. The value of δ must be greater than δ_{large} only at T_{large}.

The CG is formulated to enforce two minimum following distance constraints, similar to the vehicle-following strategy in Chapter 3:

1. A *desired* minimum following distance (δ_{large}) at the final time step of the CG horizon (T_{large}).
2. A minimum *safe* following distance (δ_{min}) that must be enforced at each time step along a beginning portion of the CG horizon (T_{min}).

Two following distance constraints are implemented in the CG formulation to account for the fact that the lead vehicle's velocity profile will often differ from what the CG assumes. Specifically, while the actual velocity profile of the lead vehicle is unknown, its velocity is assumed by the CG to be constant over the prediction horizon. If only δ_{min} were enforced over the prediction horizon, the CG would allow the vehicle to travel at a following distance near that minimum, under the assumption of a constant-speed lead vehicle. If the lead vehicle slows down unexpectedly instead of remaining at the assumed constant speed, then the CG would have no choice but to apply a large braking force at the subsequent time step. To mitigate this issue, the control strategy introduces an additional terminal constraint at the end of the CG horizon, which requires a deceleration profile that forces the vehicle to follow at a larger following distance, δ_{large}. As illustrated in Figure 27, this results in a buffer between the vehicle's actual following distance and the absolute minimum allowable following distance, thereby allowing the ego vehicle to avoid large braking events upon every unexpected deceleration of the lead vehicle.

In order to obey all red signals, the CG requires the red signal be characterized as a lead vehicle. Therefore, if the signal is red and there is no lead vehicle between the ego vehicle and the intersection, a *phantom* lead vehicle is simply placed at the stop line with zero velocity. This allows the CG to also maintain intersection constraints.

For the simulations performed in this work, the values of T_{large} and T_{min} were taken as 20 and 6 seconds, respectively, with a time step of 0.2 seconds. The horizon length values were obtained through an exhaustive search of simulations, selecting the pair with the greatest performance (least braking overall). The following distances δ_{large} and δ_{min} are obtained by multiplying the lead vehicle's velocity by time headways of 3 and 2 seconds, respectively. However, a static headway of 5 m is implemented for stopped or very slow traffic.

4.4 Suburban Controller Evaluation

This section will detail the two main simulation studies used to validate the h–GLAS strategy for suburban environments. One has been used to validate the SPaT forecast model, the velocity planning algorithm, and the MPC in isolation of traffic, and the other has been used to validate the full h–GLAS strategy in a comprehensive traffic simulator. The first simulation study only requires a dynamic simulation model of the vehicle and the recorded SPaT information over the given route of interest and time period, whereas the second study will require a more advanced simulation setup in order to capture the correct behavior between traffic and actuated signalized intersections. Similarly to the other simulation studies, the dynamic simulation model provided by Volvo Group Trucks was used to assess overall fuel consumption (m_{fuel}) in both studies.

4.5 h-GLAS Strategy Validation Study 1

The city of Concord, NC's DOT supplied SPaT information for 10 actuated signalized intersections along HWY route 29. A Google map with the signalized intersections overlaid is displayed in Figure 28. This route was selected, as it represents a non-highway route that sees a lot of heavy-duty traffic and consists of actuated signalized intersections. Interestingly, every signalized intersection in the city of Concord, NC is either semi-or fully-actuated. The supplied SPaT information for this study includes the green and red start times for the signal phases corresponding to the aforementioned route, collected on August 15, 2019 from 6 A.M. to 6 P.M. The interval between the green and red start times represent the green signal phase. However, this green interval includes the yellow change and red clearance times. Therefore, the red clearance time will need to be subtracted from the green signal phase and added to the red signal phase intervals. Depending on the signal state, the duration for the current signal phase (green or red) is assumed to be known only when the yellow signal phase has begun, illustrated by Figure 29. Since this is only a single route, in order for the vehicle to observe different signal timings from one simulation to another, multiple departure times were simulated. This allows the vehicle to encounter each intersection at different times throughout the day, changing the observed SPaT.

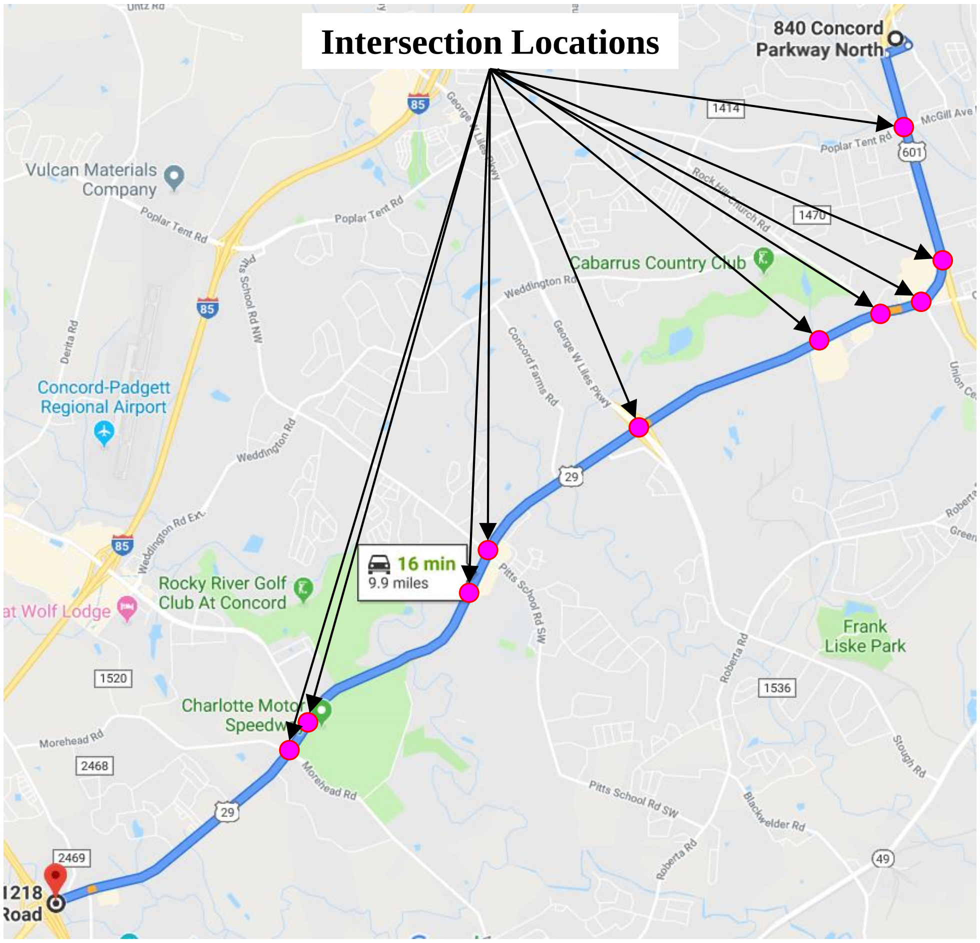


Figure 28: Portion of HWY route 29 used for simulation. Credit: Google Maps.

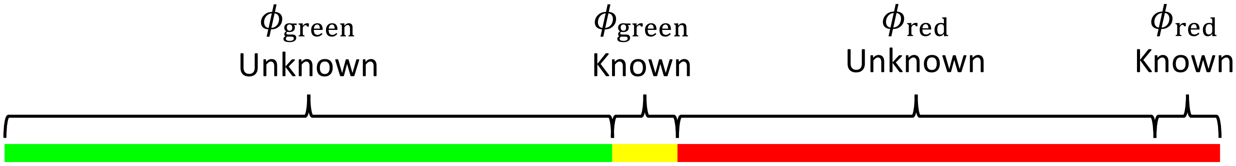


Figure 29: Situations where the current signal phase (green or red) length is known.

4.5.1 Summary of results

This simulation study evaluates the efficacy of both SPaT forecast models (GP- and Average-based) against an omniscient case. A total of 20 different departure times were simulated in this study, where the overall simulation results are displayed in Figure 30.

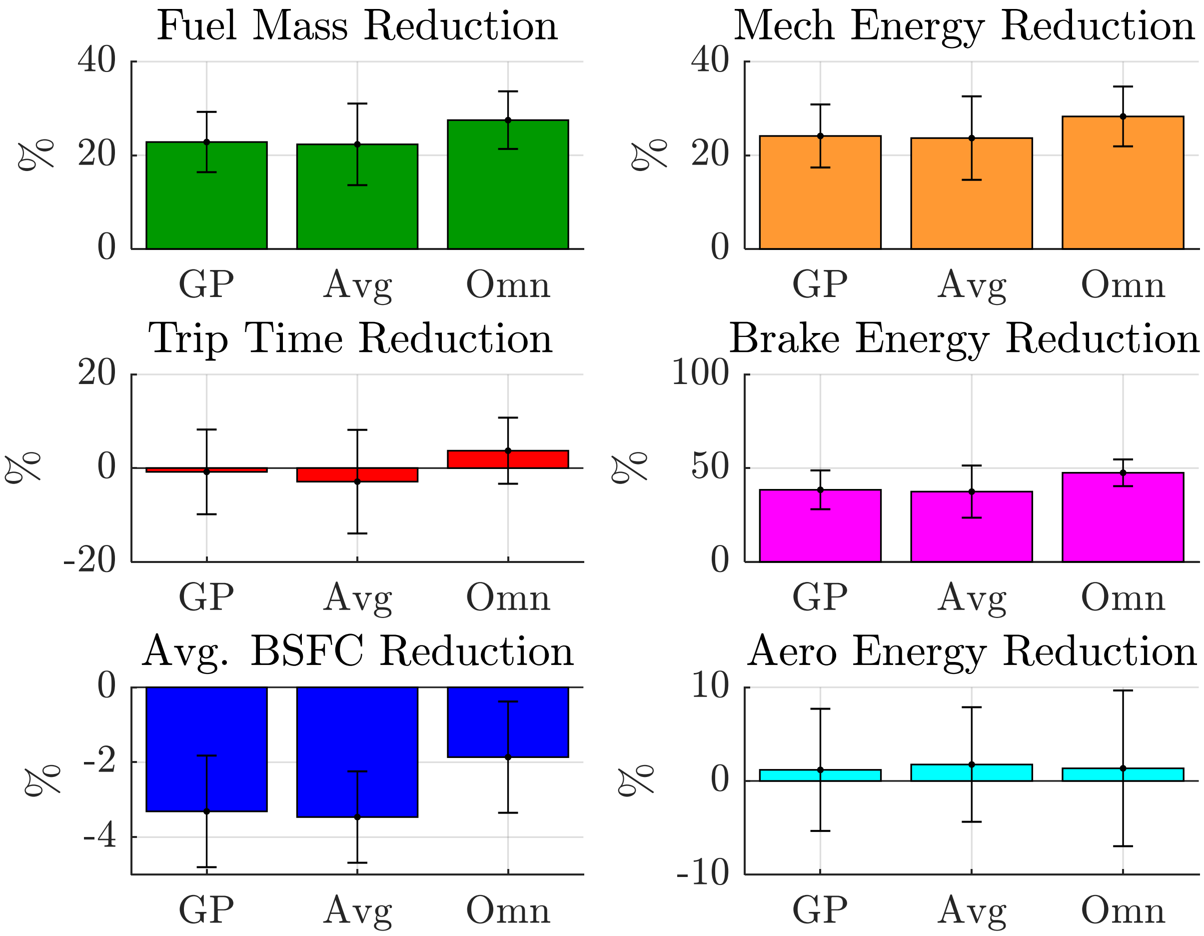


Figure 30: Signalized Intersection simulation results. The bar plots show the average percent *reduction* from the baseline simulation and the error bars correspond to a 95% confidence interval.

In all simulations, the h-GLAS strategy significantly outperformed the baseline control strategy, with average fuel consumption improvements of 22.8% and 22.3% for the GP- and average-based forecast models, respectively. As observed with the highway and urban studies, the major source of these improvements was found in the reduction of braking effort, where the h-GLAS strategy was able to reduce the average braking energy expenditure by 38.4% and 37.5% for the GP- and average-based forecast models, respectively. Looking into the simulations where the forecast models were replaced with omniscient SPaT, only an additional 5% reduction in fuel consumption was realized, which implies that the forecast models presented in this research can be improved, but are very effective nonetheless. However, the most interesting result lies between the two forecast models. With only a 0.5% difference in fuel consumption improvement, the more complex GP forecast model is not required to realize a significant reduction in fuel consumption.

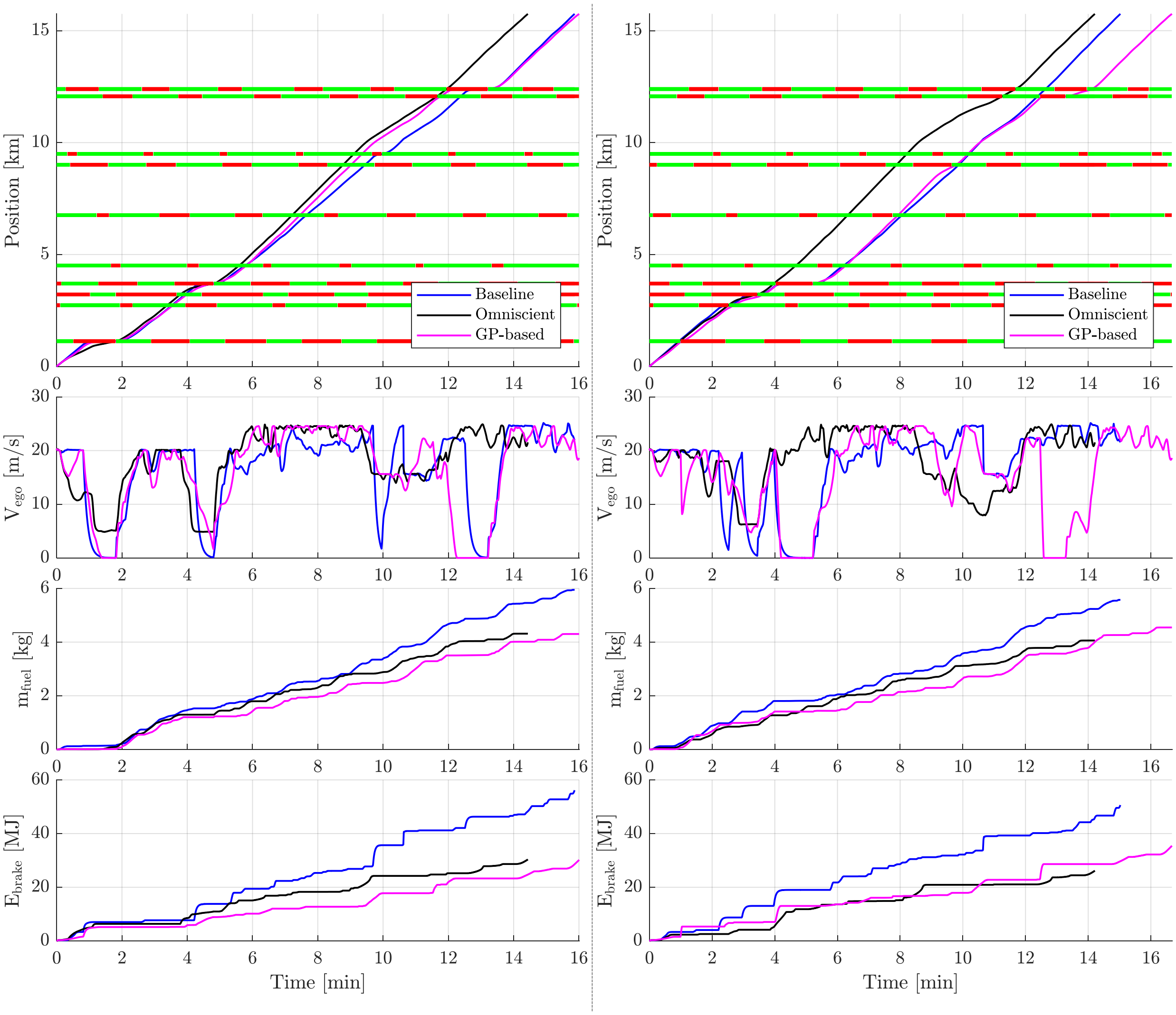


Figure 31: Best and worst simulation results of the baseline and optimal control strategies traveling through the signalized intersections. The top plots show the trajectories of the baseline, omniscient, and GP-based h-GLAS simulations. The other three plots display the ego vehicle's velocity, fuel consumed, and braking energy, respectively. The average-based trajectories were omitted as they were almost identical to the GP-based results.

Figure 31 displays the results for two simulation cases, namely the simulation that produced the most fuel consumption improvement (best result) and the simulation where the baseline's trip time was significantly less than the optimal controller's trip time (worst result). For the simulation with the the most fuel consumption improvement (left set of plots), there were four intersections where the baseline was forced to stop, whereas the h-GLAS strategy only stopped for three. Even though the h-GLAS strategy also had to come to a few stops, it used its preview knowledge of the route to do so much sooner than the baseline. The baseline strategy kept moving at its desired speed until being forced to stop, which resulted in increased fuel consumption. For the set of plots on the right in Figure 31, both the baseline and GP-based h-GLAS strategies had similar trip times until the second to last intersection, where the h-GLAS strategy was forced to stop. While approaching that intersection, the h-GLAS strategy was traveling slightly slower than the baseline, which resulted in an encounter with a red signal. As shown in the velocity and E_{brake} plots in Figure 31, the GP-based h-GLAS strategy was required to slow down very abruptly, expending about 5 MJ in braking energy. The increased fuel consumption for the GP-based h-GLAS strategy right before the 12 minute mark could have been avoided if the controller had known when the green signal phase would end, which implies that there is room for improvement in the forecast models.

4.6 h-GLAS Strategy Validation Study 2

While the previous simulation study allowed for the evaluation of the forecast models and the impact of actuated signalized intersections on fuel consumption, it did not test the resilience of the h-GLAS controller, as the simulations were not true representations of a suburban environment. To accomplish the task of truly simulating a suburban environment, the same route was constructed in PTV's VISSIM, using the exact signal controller parameters implemented in the field. However, in order for the simulation to be complete, traffic needs to be simulated, as their interaction with the actuated intersections drive the signal timing.

Referring back to the VISSIM model description in Chapter 2.2.3, the vehicle input density (vehicles per hour) for each road needs to be defined. Fortunately, the same data collection used to record the SPaT information for the previous simulation study also records average vehicle density observed at the intersections. Therefore, those values were selected for the vehicle input densities. Simulating traffic in VISSIM also requires routing decisions be defined for each intersection approach. This information is unfortunately unavailable, requiring these values be defined based off of my experience driving this route for the past 10 years. The values implemented were verified to be reasonable by a staff engineer, Victoria Forrester, at Concord, NC's DOT. Lastly, three intersections were unable to be modeled, as the DOT was unable to share the signal controller parameters.

Through a component object model (COM) interface, VISSIM can interact directly with the Matlab environment [58]. This includes acquiring signal and vehicle states from each intersection and vehicle in the network and stepping through the simulation. Unfortunately, however, Simulink cannot interact through a COM interface, which is where the vehicle dynamics are simulated to assess performance. Therefore, the following procedure is implemented in order to simulate VISSIM and Simulink:

1. In a Matlab for loop, begin simulating the VISSIM network, one step at a time, *until* it is saturated with vehicles. This is required, as vehicles only enter the network at the beginning of each road.

2. Insert a single ego vehicle at the beginning of the desired route with the flag to ignore VISSIM objects. This is required to avoid the default vehicle-following and adherence to signal-

ized intersection controllers defined for each vehicle in the VISSIM network, such that the ego vehicle can be controlled independently.

3. Acquire the signal, ego, and preceding vehicle's states at the current time step.
4. Run Simulink for the duration of *one* VISSIM time step (0.1 seconds).
5. Store the values of the integrator stares from Simulink to be used for the initialization of the Simulink model at the following time step.
6. Use the final velocity from Simulink to set the ego vehicle's velocity in VISSIM through the COM interface.
7. Run VISSIM for one time step.
8. Repeat steps 3-7 until the ego vehicle reaches the end of the route.

The interactions between VISSIM, Matlab, and Simulink are pictured in Figure 32. At each time step of the VISSIM simulation (length 0.1 seconds), the signal and traffic states are formatted in the "sensors" block for the controller in Simulink to use. Using the formatted SPaT and traffic information, along with the ego vehicle's states, Simulink is run for 0.1 seconds at a sample rate of 100 Hz (vehicle dynamics require this sample rate), making this setup a multi-rate simulation.

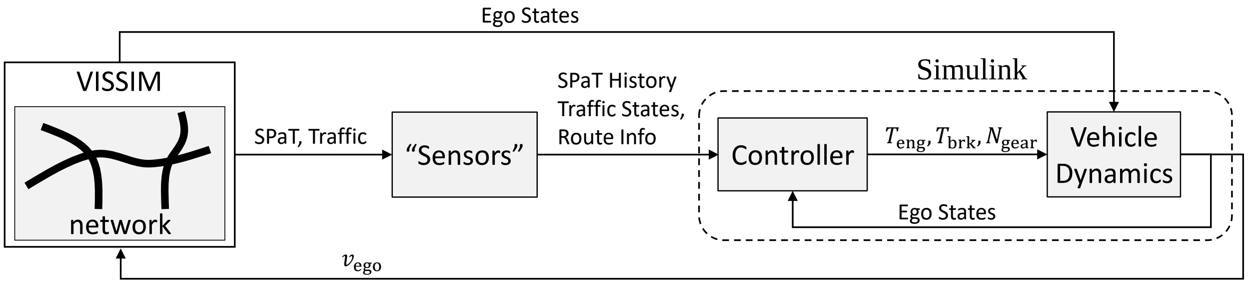


Figure 32: Block diagram for the VISSIM simulation study.

Because there are 10 Simulink time steps for each VISSIM time step, the lead vehicle is defined to follow its desired acceleration in Simulink, which is another property available through the COM interface. The final velocity value from the Simulink simulation is then defined as the ego vehicle's velocity in VISSIM, to be followed over the next VISSIM time step. After running VISSIM for another time step, the process is then repeated until the route is finished. Any states that require continuous simulation in Simulink are stored from one simulation to another. This way, the repeated 0.1 second simulations behave as one continuous simulation. Similarly to the previous simulation studies with signalized intersections, various departure times are considered to allow the vehicles to encounter different signal timings. Lastly, because the baseline and h-GLAS strategies will have different trajectories throughout the route, surrounding traffic will differ slightly, which ultimately results in slightly different signal timing.

A total of 7 departure times were simulated to produce the results summarized in Figure 33. The h-GLAS control strategy was able to reduce fuel consumption by 23.2% on average, similarly to the results from the previous study. However, there is one significant difference between the two studies, namely the trip times. On average, the h-GLAS strategy's trip time was 19.6% longer than the baseline, which equates to about 3 minutes on average.

The difference in trip times appears to be a result of the forecast model being too conservative. One difference between the two studies is the knowledge of the red signal end time. If the current signal phase is red, due to the complexity of how the signal states are stored in VISSIM, the end of the red signal phase is unknown until it changes. Therefore, the forecast model is required to predict the end time until it actually ends. As shown in Figure 34, the h-GLAS strategy is very conservative when approaching a red signal, which is very apparent for the intersection around 6.5 km. Looking at the zoomed-in plot, the h-GLAS strategy continues at a slower velocity (shallower

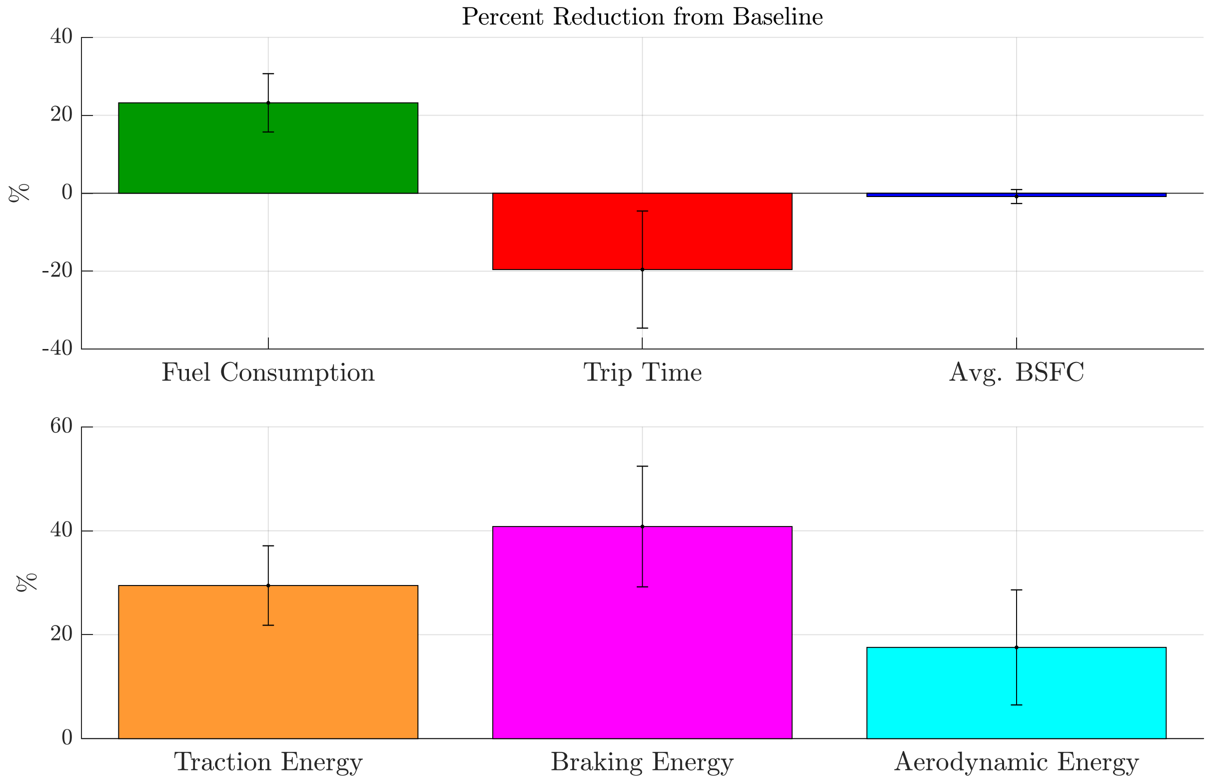


Figure 33: Urban environment simulation results. The error bars reflect a 95% confidence interval.

slope) until the signal turns green, at which the vehicle begins to accelerate. However, the delayed response to the signal changing caused the vehicle to reach the following intersection while the signal was red. If the h-GLAS controller was provided with the end time for the first intersection, it could have crossed the intersection closer to the beginning of the green window, allowing the vehicle to reach the following intersection while it was green. This is possible, but will require a more sophisticated “sensor” block to implement it.

A new question now exists: what would the aggregate fuel consumption savings be if the suburban route was followed by a highway trip, where the time lost in the suburban route was required to be made back up? To answer this question, one needs to quantify the relationship between deviations in trip time and fuel consumed for a highway environment. To accomplish this, the baseline strategy was again simulated over the four highway routes, where the velocity setpoint (percentage

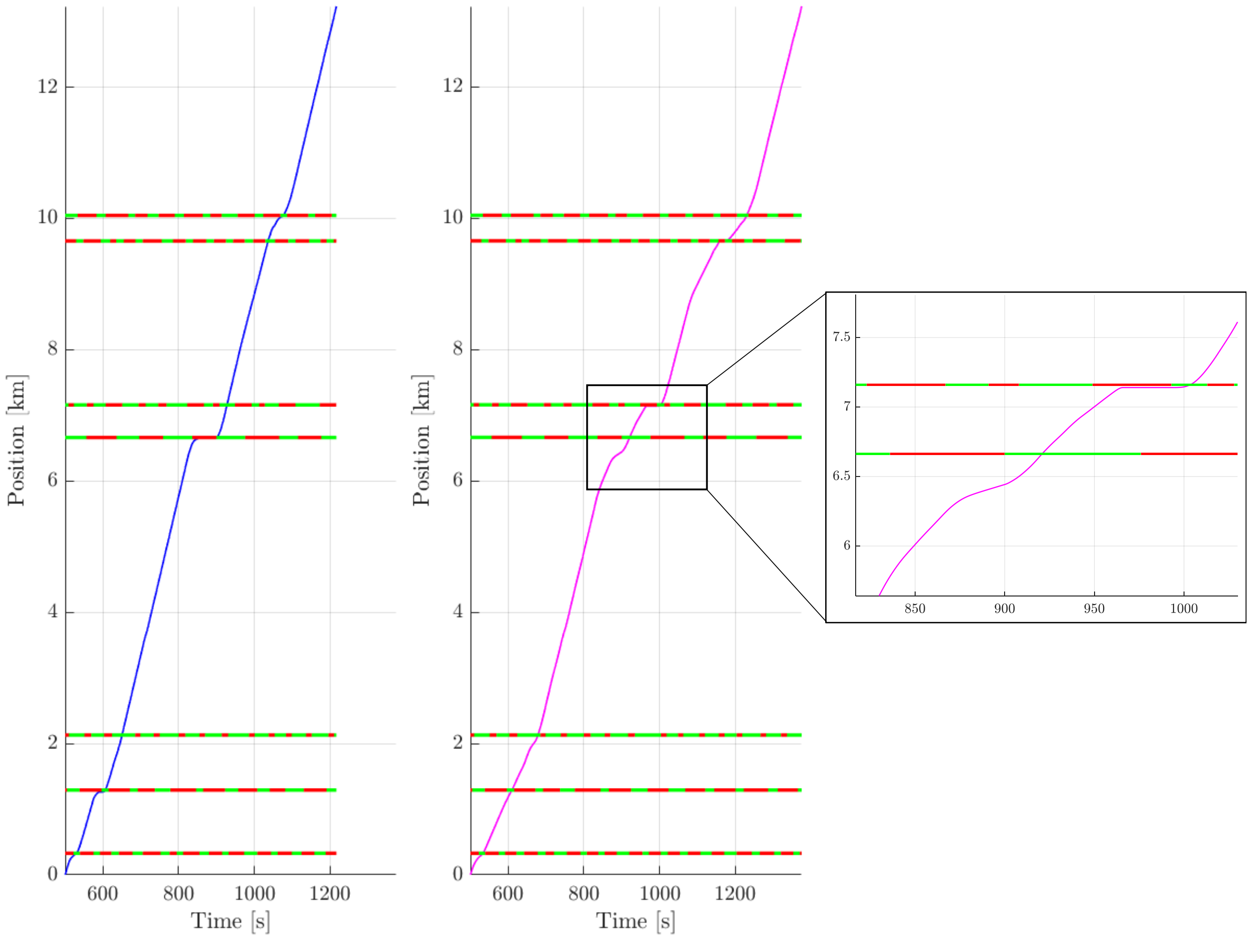


Figure 34: Vehicle trajectories for the baseline (left) and the h-GLAS (right) strategies.

of the speed limit) was varied from one simulation to another. This provides a variation in trip time from a *nominal* simulation used in the highway simulation study, which can be linked to variations in fuel consumption. Figure 35 shows this relationship for HWY 1, which can be used for a lookup table. To assess the total fuel consumption for a suburban + highway trip, the difference in trip time from the suburban simulation was then used to look up the fuel consumption required to make up that difference. Table 4 shows the total fuel consumption savings for each suburban run followed by one of the highway routes. Even though the vehicle was forced to expend additional fuel over the highway portion of the route in order to make up for the trip time difference, there

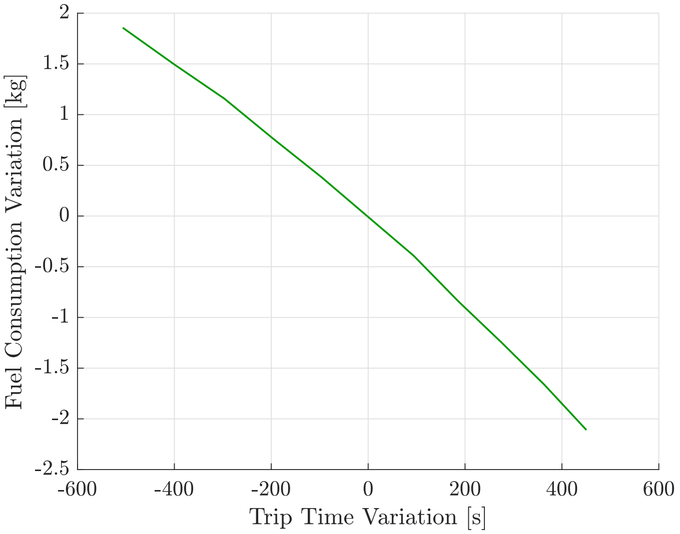


Figure 35: Differences in fuel consumption vs differences in trip time for Hwy 1.

	Sub 1	Sub 2	Sub 3	Sub 4	Sub 5	Sub 6	Sub 7	**Avg.**
Hwy 1	0.60 kg	0.63 kg	0.90 kg	1.10 kg	0.24 kg	1.16 kg	0.56 kg	**0.74 kg**
Hwy 2	0.53 kg	0.55 kg	0.83 kg	1.09 kg	0.18 kg	1.13 kg	0.48 kg	**0.68 kg**
Hwy 3	0.59 kg	0.62 kg	0.89 kg	1.11 kg	0.23 kg	1.16 kg	0.55 kg	**0.73 kg**
Hwy 4	0.67 kg	0.70 kg	0.97 kg	1.14 kg	0.30 kg	1.21 kg	0.64 kg	**0.80 kg**

Table 4: Total fuel consumption savings from a trip consisting of a suburban route (Sub 1-7) followed by a highway route (Hwy 1-4), where trip times are required to be equal.

were still enough savings for the total differences to remain positive. Note that a value of 0 kg in Table 4 refers to a situation where *only* the fuel savings realized in the suburban portion were cancelled out, and only the savings from the highway portion remain. The average fuel savings in the suburban environment *alone* were 1.31 kg in total. Therefore, by averaging the last column in Table 4, the 23.2% reduction, shown in Figure 33, becomes 13.2% instead. However, if the SPaT information can include the end times for red signal phases, the trip time discrepancy will lessen, providing additional fuel savings.

CHAPTER 5: CONCLUSIONS AND FUTURE PROSPECTS

This chapter summarizes the key achievements of this work and also proposes possible avenues of investigation for future researchers.

5.1 Conclusions

The overarching goal of this research has been to develop a comprehensive longitudinal control formulation to minimize overall fuel consumption for vehicles traveling through highway, urban, and suburban environments with non-deterministic elements, specifically focusing on actuated signalized intersections. In order to accomplish this goal, I have developed a hierarchical Green-Light Approach Speed (h-GLAS) controller that 1) uses past signal timing information to forecast its future behavior, and 2) efficiently and safely navigates through a suburban corridor with actuated signalized intersections.

This work introduced three novel control features, within a hierarchical control formulation, for a vehicle traveling through a non-deterministic suburban environment. These include:

1. a signal timing forecast model, which uses past signal phase length information in order to forecast predicted future signal phase lengths,

2. a velocity planning algorithm, which constructs a desired velocity profile to allow the ego vehicle to arrive at upcoming intersections when the probability of a green signal is maximized, and

3. a model predictive controller, which trades off minimizing traction and braking energy ex-

penditure with tracking of a desired kinetic energy profile.

This dissertation then showed the effectiveness of the hierarchical controller in a comprehensive simulation study, consisting of non-deterministic traffic behavior with actuated signalized intersections programmed with the actual signal controller parameters used in the field.

5.2 Future Prospects

The work presented in this dissertation contains several novel contributions. However, there exist several opportunities for further investigation into the problem of efficiently and safely traversing through suburban environments with actuated signalized intersections. These include at least three clear opportunities:

- An investigation into the improvement of the signal phase and timing forecast model.
- An investigation into the development of a velocity planning algorithm that focuses more on the objective of minimizing predicted fuel consumption.
- The combined coordination of signalized intersections and V2I capable vehicles.

The h-GLAS control strategy, when provided with omniscient signal timing information, was able to realize an additional 5% in fuel savings over when the h-GLAS strategy relied on a forecast model. Along with past signal phase length information, the information collected at signalized intersections also includes information about vehicles passing over detectors, which can aid in the prediction of when signal phases will end. Furthermore, actuated signalized intersections can be networked, as is the case for the city of Concord, NC, which implies that the vehicle can receive information about every intersection at any time. Therefore, can detected vehicle movements from each intersection be used to extrapolate future signal timing behavior?

The second item in the above list proposes improvements for the velocity planning algorithm. The algorithm proposed in this work provides a *time-optimal* velocity profile to be referenced by the model protective controller. How much additional fuel consumption improvement can be realized if the velocity planning algorithm was focused on minimizing predicted fuel consumption or any other fuel consumption surrogate like traction or braking energy expenditure? Additionally, how much fuel consumption improvement can be realized if the velocity planning algorithm considers traffic behavior when constructing the desired velocity profile? The h-GLAS control strategy was separated into a hierarchical structure, as the computational burden of the "whole" was less than the "sum of its parts". However, if one can begin to recombine the individual components, without sacrificing computational performance, how much more effective of a control formulation can exist?

Finally, having demonstrated significant fuel economy improvement for a single vehicle traveling through a suburban environment, it seems appropriate to consider a more saturated connected and autonomous vehicle network. We are reaching a point where autonomous vehicle and wireless communication technology are both sophisticated enough, such that signal controllers can receive the predicted future behavior of connected and autonomous vehicles to be used in the determination of signal timing. If signal controllers can receive the predicted trajectories for a majority of vehicles on the road, can signal controllers also provide predicted signal timing back to said vehicles? If so, how much additional fuel economy improvement can be realized, and would this eliminate the need for a SPaT forecast model? I see a future where traffic control devices and vehicles are coordinated together to optimize vehicle throughput and fuel economy simultaneously.

REFERENCES

[1] Transportation Research Board and National Academies of Sciences, Engineering, and Medicine. *Signal Timing Manual - Second Edition*. The National Academies Press, Washington, DC, 2015.

[2] MIT Technology Review Insights. Self-driving cars take the wheel, 2019. https://www.technologyreview.com/s/612754/self-driving-cars-take-the-wheel/.

[3] L. Zhu, J. Gonder, E. Bjarkvik, M. Pourabdollah, and B. Lindenberg. An automated vehicle fuel economy benefits evaluation framework using real-world travel and traffic data. *IEEE Intelligent Transportation Systems Magazine*, 11(3):29–41, Fall 2019.

[4] American Trucking Association. Economics and industry data, 2020. https://www.trucking.org/economics-and-industry-data.

[5] J.N. Hooker. Optimal driving for single-vehicle fuel economy. *Transportation Research Part A: General*, 22(3):183 – 201, 1988.

[6] R. Wang and S. M. Lukic. Dynamic programming technique in hybrid electric vehicle optimization. In *IEEE International Electric Vehicle Conference*, Greenville, SC, 2012.

[7] G. Heppeler, M. Sonntag, and O. Sawodny. Fuel efficiency analysis for simultaneous optimization of the velocity trajectory and the energy management in hybrid electric vehicles. *IFAC Proceedings Volumes*, 47(3):6612 – 6617, 2014.

[8] A. Schwarskopf and R. Leipnik. Control of highway vehicles for minimum fuel consumption over varying terrain. *Transportation Research*, 11(4):279–286, 1977.

[9] E. Ozatay, U. Ozguner, J. Michelini, and D. Filev. Analytical solution to the minimum energy consumption based velocity profile optimization problem with variable road grade. *IFAC Proceedings Volumes*, 47(3):7541 – 7546, 2014.

[10] M. Wahba and S. Brennan. Mpc-based energy management of a parallel hybrid electric vehicle using terrain information. In *ASME Dynamic Systems and Control Conference*, Columbus, OH, 2015.

[11] M. Rodriguez and H. Fathy. Speed trajectory optimization for a heavy-duty truck traversing multiple signalized intersections: A dynamic programming study. In *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1454–1459, Aug 2018.

[12] B. Asadi and A. Vahidi. Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. *IEEE Transactions on Control Systems Technology*, 19(3):707–714, 2011.

[13] S. E. Li, S. Xu, X. Huang, B. Cheng, and H. Peng. Eco-departure of connected vehicles with v2x communication at signalized intersections. *IEEE Transactions on Vehicular Technology*, 64:5439–5449, Dec 2015.

[14] V. A. Butakov and P. Ioannou. Personalized driver assistance for signalized intersections using v2i communication. *IEEE Transactions on Intelligent Transportation Systems*, 17:1910–1919, July 2016.

[15] X. Huang and H. Peng. Speed trajectory planning at signalized intersections using sequential convex optimization. In *American Control Conference*, Seattle, WA, 2017.

[16] G. Mahler and A. Vahidi. Reducing idling at red lights based on probabilistic prediction of traffic signal timings. In *American Control Conference*, Montreal, QC, Canada, 2012.

[17] G. Mahler and A. Vahidi. An optimal velocity-planning scheme for vehicle energy efficiency through probabilistic prediction of traffic-signal timing. *IEEE Transactions on Intelligent Transportation Systems*, 15:2516–2523, Dec 2014.

[18] C. Sun, X. Shen, and S. Moura. Robust optimal eco-driving control with uncertain traffic signal timing. In *American Control Conference*, Milwaukee, WI, 2018.

[19] P. Hau, G. Wu, K. Boriboonsomsin, and M. Barth. Eco-approach and departure (ead) application for actuated signals in real-world traffic. *IEEE Transactions on Intelligent Transportation Systems*, 20(1):30–40, 2019.

[20] S. Mousa, S. Ishak, R. Mousa, and J. Codjoe. Developing an eco-driving application for semi-actuated signalized intersections and modeling the market penetration rates of eco-driving. *Transportation Research Record*, 2673(5):466–477, 2019.

[21] L. Makarem and D. Gillet. Model predictive coordination of autonomous vehicles crossing intersections. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, Netherlands, 2013.

[22] B. HomChaudhuri, A. Vahidi, and P. Pisu. A fuel economic model predictive control strategy for a group of connected vehicles in urban roads. In *American Control Conference*, Chicago, IL, 2015.

[23] B. HomChaudhuri, A. Vahidi, and P. Pisu. Fast model predictive control-based fuel efficient control strategy for a group of connected vehicles in urban road conditions. *IEEE Transactions on Control Systems Technology*, 25:760–767, March 2017.

[24] A. Pozzi, S. Bae, Y. Choi, F. Borrelli, D. M. Raimondo, and S. Moura. Ecological velocity planning through signalized intersections: A deep reinforcement learning approach. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 245–252, Dec 2020.

[25] J. Borek, B. Groelke, C. Earnhardt, and C. Vermillion. Optimal control of heavy-duty trucks in urban environments through fused model predictive control and adaptive cruise control. In *American Control Conference*, Philadelphia, PA, 2019.

[26] S. Bae, Y. Kim, J. Guanetti, F. Borrelli, and S. Moura. Design and implementation of ecological adaptive cruise control for autonomous driving with communication to traffic lights. In *American Control Conference*, Philadelphia, PA, 2019.

[27] S. Bae, Y. Choi, Y. Kim, J. Guanetti, F. Borrelli, and S. Moura. Real-time ecological velocity planning for plug-in hybrid vehicles with partial communication to traffic lights. In *Conference on Decision and Control*, Nice, France, 2019.

[28] C. Sun, J. Guanetti, F. Borrelli, and S. J. Moura. Optimal eco-driving control of connected and autonomous vehicles through signalized intersections. *IEEE Internet of Things Journal*, 7(5):3759–3773, May 2020.

[29] J. Borek, B. Groelke, C. Earnhardt, and C. Vermillion. Optimal control of heavy-duty trucks through signalized intersections with non-deterministic signal timing. *Under review - IEEE Transactions on Intelligent Transportation Systems*, pages 1–10, 2019.

[30] L. Li-hua, L. Hong, L. Ping, and W. Hui. Model predictive control for adaptive cruise control with multi-objectives: comfort, fuel-economy, safety and car-following. *Journal of Zhejiang University SCIENCE A*, 11(3):191–201, 2010.

[31] T. Stanger and L. del Re. A model predictive cooperative adaptive cruise control approach. In *American Control Conference*, Washington DC, 2013.

[32] L. Bertoni, J. Guanetti, M. Basso, M. Masoero, S. Cetinkunt, and F. Borrelli. An adaptive cruise control for connected energy-saving electric vehicles. *IFAC-PapersOnLine*, 50(1):2359 – 2364, 2017.

[33] M. Vajedi and N. L. Azad. Ecological adaptive cruise controller for plug-in hybrid electric vehicles using nonlinear model predictive control. *IEEE Transactions on Intelligent Transportation Systems*, 17(1):113–122, 2016.

[34] B. Groelke, J. Borek, C. Earnhardt, J. Li, S Geyer, and C. Vermillion. A comparative assessment of economic model predictive control strategies for fuel optimization for heavy-duty trucks. In *American Control Conference*, Milwaukee, WI, 2018.

[35] J. Borek, B. Groelke, C. Earnhardt, and C. Vermillion. Economic optimal control for minimizing fuel consumption of heavy-duty trucks in a highway environment. *IEEE Transactions on Control Systems Technology*, pages 1–13, 2019.

[36] S. Xu and H. Peng. Design, analysis, and experiments of preview path tracking control for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–11, 2019.

[37] M. Song, N. Wang, T. Gordon, and J. Wang. Flow-field guided steering control for rigid autonomous ground vehicles in low-speed manoeuvring. *Vehicle System Dynamics*, 57(8):1090–1107, 2019.

[38] X. He, Y. Liu, C. Lv, X. Ji, and Y. Liu. Emergency steering control of autonomous vehicle for collision avoidance and stabilisation. *Vehicle System Dynamics*, 57(8):1163–1187, 2019.

[39] J. Lee and H. Chang. Multi-parametric model predictive control for autonomous steering using an electric power steering system. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 233(13):3391–3402, 2019.

[40] H. Nam, W. Choi, and C. Ahn. Model predictive control for evasive steering of an autonomous vehicle. *International Journal of Automotive Technology*, 20(5):1033–1042, Oct 2019.

[41] K. Chen, X. Pei, D. Sun, Z. Chen, X. Guo, and K. Guo. Active steering control for autonomous vehicles based on a driver-in-the-loop platform: A case study of collision avoidance. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 233(10):1422–1437, 2019.

[42] C. Vermillion, K. Butts, and K. Reidy. Model predictive engine torque control with real-time driver-in-the-loop simulation results. In *American Control Conference*, Baltimore, MD, 2010.

[43] C. Xu, A. Al-Mamun, S. Geyer, and H. K. Fathy. A dynamic programming-based real-time predictive optimal gear shift strategy for conventional heavy-duty vehicles. In *American Control Conference*, pages 5528–5535, June 2018.

[44] C. Xu, S. Geyer, and H. K. Fathy. Formulation and comparison of two real-time predictive gear shift algorithms for connected/automated heavy-duty vehicles. *IEEE Transactions on Vehicular Technology*, 68(8):7498–7510, Aug 2019.

[45] B. Block, B. Huynh, S. Boyle, S. Stockar, S. Geyer, J. Li, and J. Huber. Analysis of the effect of vehicle platooning on the optimal control of a heavy duty engine thermal system. In *SAE Technical Paper*, 04 2019.

[46] S. Boyle and S. Stockar. Comparison of input shaping and predictive reference generator techniques for ic engine setpoints commands. *9th IFAC Symposium on Advances in Automotive Control*, 52(5):279 – 284, 2019.

[47] M. Henzler, M. Buchholz, and K. Dietmayer. Online velocity trajectory planning for manual energy efficient driving of heavy duty vehicles using model predictive control. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, China, 2014.

[48] R. Gordon and W. Tighe. *Traffic Control Systems Handbook*. Dunn Engineering Associates, Westhampton Beach, NY, 2005.

[49] Transportation Research Board, National Academies of Sciences, Engineering, and Medicine. *Signal Timing Manual - Second Edition*. The National Academies Press, Washington, DC, 2015.

[50] PTV Group. Ptv vissim, 2020. https://www.ptvgroup.com/en/solutions/products/ptv-vissim/.

[51] P.G. Gipps. A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, 15(2):105 – 111, 1981.

[52] H. Berndt, S. Wender, and K. Dietmayer. Driver braking behavior during intersection approaches and implications for warning strategies for driver assistant systems. In *IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, 2007.

[53] M.A.S. Kamal, M. Mukai, J. Murata, and T. Kawabe. Ecological driving based on preceding vehicle prediction using mpc. *IFAC Proceedings Volumes*, 44(1):3843 – 3848, 2011.

[54] C. Xu, B. Groelke, M. Alvarez, C. Earnhardt, J. Borek, E. Pelletier, S. Boyle, B. Huynh, M. Wahba, S. Geyer, C. Graham, M. Magee, K. Palmeter, M. Naghnaeian, S. Brennan, S. Stockar, C. Vermillion, and H. Fathy. Engine-in-the-loop study of a hierarchical predictive online controller for connected and automated heavy-duty vehicles. In *WCX SAE World Congress Experience*. SAE International, Apr 2020.

[55] B. Groelke, J. Borek, C. Earnhardt, and C. Vermillion. Design and performance analysis of a cascaded model predictive controller and command governor for fuel-efficient control of heavy-duty trucks. *IEEE Transactions on Control Systems Technology - Under Review*, 2019.

[56] C. Rasmussen and C. Williams. *Gaussian Process for Machine Learning*. The MIT Press, Cambridge, MA, 2006.

[57] Evelina Gabasova. Covariance functions, 2014. http://evelinag.com/Ariadne/covarianceFunctions.html.

[58] PTV Group. *PTV VISSIM 11 User Manual*. PTV Group, Karlsruhe, Germany, 2018.