Saxton Transportation Operations Laboratory

TO17: GlidePath Prototype Application

Final Report



**Contract Number: DTFH61-12-D-00020**

**Submitted: 10/26/2016**

**Version: 1**

Abstract

Together the Intelligent Transportation Systems (ITS) Joint Program Office (JPO) and Federal Highway Administration’s (FHWA’s) Turner Fairbank Highway Research Center (TFHRC) Office of Operations, Research and Development (HRDO) implemented and successfully demonstrated the GlidePath Prototype Application. The GlidePath Prototype Application demonstrated operation of a Level 1[[1]](#footnote-1) connected-automated vehicle communicating with an intelligent traffic signal using vehicle-to-infrastructure (V2I) communications to optimize the vehicle’s speed as it traverses a signalized intersection. Using dedicated short range communications (DSRC), the equipped vehicle was able to receive messages about the intersection geometry and the signal phase and timing (SPaT) from a roadside unit (RSU) that was connected to a traffic signal controller operating a traffic signal with a fixed timing plan at a single signalized intersection at TFHRC. Using the vehicle’s speed and position along with the SPaT and MAP messages received from the RSU, an algorithm executing on the vehicle was able to compute an optimum speed profile to minimize fuel consumption and emissions as the vehicle approached and departed from the intersection. Upon computing this speed profile, and activation by the driver, the GlidePath software takes control of the vehicle’s accelerator and brakes to safely and smoothly drive it through the intersection while respecting the traffic signal and local speed limit.

Field experiments were conducted to better understand the potential environmental benefits of the application. To do so, the vehicle’s environmental performance was characterized over a range of operating speeds and intersection entry times. Field experiments collected data from three scenarios: (i) the automated GlidePath Prototype Application, (ii) human driver receiving speed recommendations from a human machine interface (HMI), and (iii) uninformed drivers operating the vehicle without receiving any information on recommended speeds. Data collected in field experiments revealed that average fuel consumption was improved in vehicles equipped with the GlidePath Eco-Approach and Departure application. Results from August 2015 indicate that a driver with an HMI saw 7% fuel savings over un-informed drivers, while a driver with partial automation and the GlidePath application saw 22% fuel savings over the un-informed driver. These results show a 15% fuel improvement from a driver trying to follow an HMI speed recommendation to the partial automated GlidePath application. These improvements are due to minimizing the lag in speed changes to keep the optimal speed and approach.

Table of Contents

[1 Background 1](#_Toc465261727)

[2 Overview 2](#_Toc465261728)

[3 Eco-Approach and Departure at Signalized Intersections 5](#_Toc465261729)

[4 Eco-Approach and Departure Algorithm 8](#_Toc465261730)

[4.1 Architecture of Vehicle Trajectory Planning Algorithm (VTPA) 8](#_Toc465261731)

[4.2 Scenario Identifier 10](#_Toc465261732)

[4.3 Trajectory Generator 12](#_Toc465261733)

[5 The GlidePath Vehicle 16](#_Toc465261734)

[6 Testbed Infrastructure 17](#_Toc465261735)

[7 Engineering Process 19](#_Toc465261736)

[8 Speed Control Software 21](#_Toc465261737)

[9 Demonstrations 22](#_Toc465261738)

[10 Algorithm Performance Evaluation 23](#_Toc465261739)

[10.1 Data Collection 23](#_Toc465261740)

[10.1.1 Data Logged 26](#_Toc465261741)

[10.1.2 Field Study Matrix 27](#_Toc465261742)

[10.2 Results and Analysis 27](#_Toc465261743)

[10.2.1 Interval-Based Comparison 27](#_Toc465261744)

[10.2.2 Scenario-Based Comparison 35](#_Toc465261745)

[11 Conclusion and Next Steps 36](#_Toc465261746)

[12 Lessons Learned and Recommendations 37](#_Toc465261747)

[12.1 Speed Control and EAD Algorithm 37](#_Toc465261748)

[12.2 Vehicle Configuration 38](#_Toc465261749)

[12.3 Infrastructure and Communications 38](#_Toc465261750)

[Appendix A – References 39](#_Toc465261751)

[Appendix B – Acronyms 40](#_Toc465261752)

Table of Figures

[Figure 1 AERIS operational scenarios 5](#_Toc465276431)

[Figure 2 GlidePath system components 6](#_Toc465276432)

[Figure 3 Speed profile scenarios 8](#_Toc465276433)

[Figure 4 Block diagram of VTPA 9](#_Toc465276434)

[Figure 5 Sub-systems of VTPA 10](#_Toc465276435)

[Figure 6 Diagram of the decision maker sub-system 10](#_Toc465276436)

[Figure 7 Scenario identifier details showing flow of control through the logic 11](#_Toc465276437)

[Figure 8 Acceleration profile of the piecewise trigonometric-linear function 12](#_Toc465276438)

[Figure 9 Deceleration profile of the piecewise trigonometric-linear function 13](#_Toc465276439)

[Figure 10 Trajectory generator details, showing flow of control for each of the four scenario outputs in Figure 7 16](#_Toc465276440)

[Figure 11 Electronic equipment and driver-vehicle interface (DVI) 17](#_Toc465276441)

[Figure 12 Overhead view of the testbed, showing signal, cabinet and RSU locations 19](#_Toc465276442)

[Figure 13 Graphic interface for “manual-HMI-assisted” driving 25](#_Toc465276443)

[Figure 14 DVI graphic interface for “(partially) automated” driving 26](#_Toc465276444)

[Figure 15 An example of field study matrix for data collection and presentation 27](#_Toc465276445)

Table of Tables

[Table 1 Other project documentation 4](#_Toc441833883)

[Table 2 Median of Normalized Fuel Consumption (across All Drivers) for Each Test Cell (gram/mile) 29](#_Toc441833884)

[Table 3 Relative Improvement (%) between Stages with Respect to Median of Normalized Fuel Consumption 30](#_Toc441833885)

[Table 4 Median of Relative Improvement (%) between Stages (across All Drivers) for Normalized Fuel Consumption 30](#_Toc441833886)

[Table 5 Standard Deviation of Relative Improvement (%) between Stages (across All Drivers) for Normalized Fuel Consumption 31](#_Toc441833887)

[Table 6 Median of Normalized CO2 Emissions (across All Drivers) for Each Test Cell (gram/mile) 32](#_Toc441833888)

[Table 7 Median of Normalized CO Emissions (across All Drivers) for Each Test Cell (10-3 gram/mile) 32](#_Toc441833889)

[Table 8 Median of Normalized HC Emissions (across All Drivers) for Each Test Cell (10-3 gram/mile) 33](#_Toc441833890)

[Table 9 Median of Normalized NOx Emissions (across All Drivers) for Each Test Cell (10-3 gram/mile) 33](#_Toc441833891)

[Table 10 Median of Trip Time (across All Drivers) for Each Test Cell (second) 34](#_Toc441833892)

[Table 11 Relative Improvement (%) between Stages with Respect to Median of Normalized Fuel Consumption (Scenario-Based) 35](#_Toc441833893)

[Table 12 Relative Improvement (%) between Stages with Respect to Median of Trip Time (Scenario-Based) 36](#_Toc441833894)

# Background

Connected vehicles have the potential to transform travel as we know it by combining leading edge technologies—advanced wireless communications, on-board computer processing, advanced vehicle-sensors, Global Positioning System (GPS) navigation, smart infrastructure, and others—to address safety, mobility, and environmental challenges. At its foundation, connected vehicle technologies include a communications network that supports vehicle-to-vehicle (V2V) two-way communications, vehicle-to-infrastructure (V2I) one- and two-way communications to support cooperative system capability. The United States Department of Transportation’s (USDOT’s) connected vehicle research is establishing an information backbone for the surface transportation system that will support applications to enhance safety and mobility and, ultimately, an information-rich surface transportation system. Connected vehicle research also supports applications to enhance livable communities, environmental stewardship, and traveler convenience and choices.

The Applications for the Environment: Real-Time Information Synthesis (AERIS) Research Program is the environmental component of the USDOT’s connected vehicle research. The program has a vision of “Cleaner Air through Smarter Transportation”. A five-year research program, AERIS was established by the Intelligent Transportation Systems (ITS) Joint Program Office (JPO) to investigate whether it was possible and feasible to:

* Identify connected vehicle applications that could provide environmental impact reduction benefits via reduced fuel use and efficiency impacts on emissions;
* Facilitate and incentivize “green choices” by transportation service consumers and providers (i.e., system users, system operators, policy decision makers, etc.);
* Identify V2V, V2I, and vehicle-to-grid (V2G) data (and other) exchanges via wireless technologies of various types;
* Model and analyze connected vehicle applications to estimate the potential environmental impact reduction benefits; and
* Develop a prototype for one of the applications to test its efficacy and usefulness.

While the AERIS Research Program was not initially charged with developing prototypes, as research progressed, foundational activities including AERIS modeling proved to be promising. In particular, the Eco-Approach and Departure at Signalized Intersections application appeared to be a near-term application that had potential to yield significant environmental benefits.

As a result, in 2012 the AERIS team conducted a field experiment at Turner Fairbank Highway Research Center (TFHRC) for the Eco-Approach and Departure at Signalized Intersections application. As part of the experiment, the Eco-Approach and Departure at Signalized Intersections application was installed in a single vehicle. As the vehicle approached a signalized intersection, operating a fixed signal timing plan, a roadside unit (RSU) broadcasted signal phase and timing (SPaT) and geometric intersection design (GID) messages, hereinafter referred to as MAP messages, using dedicated short range communications (DSRC). Using the vehicle’s location and speed, along with the information included in the SPaT and MAP messages, an algorithm located in the vehicle determined the most energy efficient speed profile for the vehicle to approach and depart from the intersection. The objective of the field experiment was to reduce the fuel consumption and emissions of the vehicle by taking data input from vehicle sensors and roadside infrastructure to calculate an “eco-friendly” speed trajectory. Drivers were provided with speed recommendations displayed on a small human-machine interface (HMI) tablet in the forward view of the driver as a green bar on a speedometer, such that the driver could accept the speed guidance to improve environmental performance of the vehicle. Preliminary field tests that were conducted in 2012 demonstrated reduced fuel consumption of up to 18%.

The HMI-based test showed that there was value in pursuing additional research with the Eco-Approach and Departure at Signalized Intersections application. The initial 2012 field experiment, however, had two drawbacks. First, relying completely on the driver for manual control of the vehicle proved to be difficult for drivers to match the speed guidance produced by the algorithm. Second, the driver spent an inordinate amount of attention watching the HMI, taking attention away from external safety concerns. Therefore, the AERIS Research Program, in coordination with FHWA, decided to undertake the GlidePath prototype application project—a first of its kind prototype—which incorporated automated longitudinal control capabilities along with the eco-approach and departure algorithms. Initial hypotheses were that automated control of the vehicle would result in the vehicle following the algorithm’s guidance more closely and free up the driver to pay attention to the safety of the vehicle. Initial hypotheses also included additional reductions in fuel and emissions.

# Overview

This document serves as the final report for the GlidePath Prototype Application, developed under Saxton Lab Task Order 17 for the FHWA’s Office of Operations, Research and Development (HRDO). The purpose of Task Order 17 was to create a fully-integrated GlidePath prototype vehicle that incorporates the eco-approach and departure (EAD) algorithm, developed by researchers at the University of California at Riverside (UCR), with a vehicle equipped with automated longitudinal control capabilities, a DSRC device, computer processor, global positioning system (GPS), and an audio/visual display. The GlidePath prototype consisted of the EAD algorithm providing speed profile outputs to control the vehicle’s longitudinal movement (i.e., accelerator and brakes) while the driver maintained control of steering ensuring safe operation of the vehicle along the roadway. Task Order 17 also included provision for field testing and demonstration of the GlidePath prototype at TFHRC’s Intelligent Intersection.

Effort on this project was partitioned into the following areas, each of which is described in detail in the following sections.

* **The eco-approach and departure (EAD) algorithm** – During the GlidePath project the algorithm version that was in the 2012 AERIS field experiment was further simulated and slightly revised based on those simulation results. Changes were also made to smooth out the accelerations to account for vehicle response lag and somewhat uneven terrain.
* **The GlidePath prototype vehicle** – A 2010 Ford Escape was acquired that had been previously outfitted with a robotic controller as part of an unrelated research effort. The GlidePath project further enhanced this vehicle to add an electronics equipment rack in the rear, containing an onboard computer, communications hardware, and the necessary power equipment to operate these electronics. Communications antennas were also added to the roof.
* **The roadway infrastructure at TFHRC** – Infrastructure located at the Intelligent Intersection at TFHRC was enhanced to support the vehicle-to-infrastructure (V2I) communications required for this project. The existing traffic signal at the facility was reconfigured for a fixed-timing cycle that suited the experimental geography. A black box was also configured to forward signal timing information as SPaT messages from the signal controller, as well as to broadcast MAP messages describing the intersection geometry. New roadside units (RSUs) were mounted in the intersection vicinity to broadcast these messages coming from the black box. Network hardware was also added to enable these communication channels.
* **Engineering process** – A full systems engineering life cycle was executed to ensure the project’s success. This process resulted in several documents that are illustrated in Table 1. The project began with a review of the Eco-Signal Operations Concept of Operations, and developed a rigorous requirements specification for the GlidePath prototype. The design activities resulted in a System Design Document that specified how all elements, both hardware and software, were to be implemented. During system construction the components were unit tested as much as possible. Integration testing then verified correct connections among components, and system testing verified correct performance of the system as a whole, including vehicle hardware, software, and roadway infrastructure components. Once the informal system testing was completed formally documented acceptance testing was performed. Tests were traced to the agreed upon requirements specification to ensure completeness. The formal tests passed and results were accepted by FHWA.
* **Speed control software development** – Control software was built using Java to run on the vehicle’s on-board Linux computer. It reads data from the robotic controller, which pulls several vehicle parameters from the CAN bus, position data from the GPS subsystem, and infrastructure messages from the DSRC on-board unit (OBU) radio receiver. This software implements the EAD algorithm, which makes use of these various sensor inputs, and passes the resulting speed commands back to the robotic controller, which sets the throttle and brake positions as necessary.
* **Demonstration** – The prototype was demonstrated for several members of the USDOT and the White House Office of Science and Technology Policy, and was featured on a CBS News television report.
* **Algorithm performance evaluation** – Once the acceptance testing was completed, a tightly controlled series of evaluations were made, capturing the vehicle’s performance across the range of speed and signal timing scenarios allowable in the TFHRC infrastructure. Runs included automated (GlidePath) control, and for comparison purposes, identical runs were made with multiple uninformed drivers (i.e. drivers who were not provided any speed recommendations). Comparison runs were also made with human control using guidance from the 2012 HMI device. Detailed performance logs were captured for all runs and analyzed to understand the performance comparisons between the different methods of control.

This document serves two purposes. First, it provides a summary of activities and accomplishments in the GlidePath project; as such it presents only limited detail in many areas that have already been documented elsewhere. Second, it presents some new information regarding the EAD algorithm and performance analysis. Analysis here areas go into more detail than previously published, and covers more depth and consistency of raw data (additional drivers with additional runs on newer control software, version 1.6). The following table serves as a reference to the most recent set of GlidePath technical documentation, where additional details can be found.

Table . Other project documentation

|  |  |  |
| --- | --- | --- |
| Document Title | Version | Delivery Date |
| Final Requirements Document | 7.0 | 11/18/2015 |
| Preliminary System Design Document | 3.1 | 11/20/2015 |
| Vehicle Hardware Configuration Document | 1 | 1/19/2016 |
| Prototype Requirements Test Plan | 2.1 | 11/17/2015 |
| System Acceptance Test Report | 2.1 | 11/10/2015 |
| GlidePath Vehicle Basic User Manual | 1 | 6/29/2015 |

# Eco-Approach and Departure at Signalized Intersections

AERIS research activities focused on five Operational Scenarios or strategic bundles of applications that were defined early in the program. As depicted in Figure 1, AERIS Operational Scenarios include: Eco-Signal Operations, Eco-Lanes, Low Emissions Zones, Eco-Traveler Information, and Eco-Integrated Corridor Management (Eco-ICM). Each Operational Scenario encompassed a set of applications which individually achieved environmental benefits. However, by strategically bundling the applications, the AERIS Program saw that the Operational Scenarios could achieve additional environment benefits above those of the individual applications.



Figure AERIS operational scenarios

The Eco-Signal Operations Operational Scenario includes applications use connected vehicle technologies to decrease fuel consumption and emissions on arterials by reducing idling, reducing unnecessary stops, and improving traffic flow at signalized intersections. The Eco-Signal Operations Operational Scenario includes five applications: (1) Eco-Traffic Signal Timing, (2) Eco-Traffic Signal Priority, (3) Eco-Approach and Departure at Signalized Intersections, (4) Connected Eco-Driving, and (5) Wireless Inductive/Resonance Charging.

The Eco-Approach and Departure at Signalized Intersection application uses wireless data communications sent from a roadside equipment unit to connected vehicles to encourage “green” approaches to signalized intersections. The application, located in a vehicle, collects SPaT and MAP messages using V2I communications and data from nearby vehicles using V2V communications. Upon receiving these messages, the application performs calculations to determine the vehicle’s optimal speed to pass the next traffic signal on a green light or to decelerate to a stop in the most eco-friendly manner. Speed recommendations may be provided to the driver using a DVI or provided to the vehicle systems that support automated longitudinal control capabilities.

This GlidePath project is designed to demonstrate the effects of automating the EAD algorithm when passing through a single signalized intersection with a fixed timing plan and no other traffic involved. These simplistic constraints are appropriate for a first step in evaluating the economic value of automated EAD control. The components of the experiment are illustrated in Figure 2. As the vehicle approaches the intersection and comes within range of the DSRC radio, the vehicle will receive SPaT and MAP messages from the intersection describing its current configuration. It can then quickly determine what speed profile scenario is possible, and compute the shape of the most economical one.

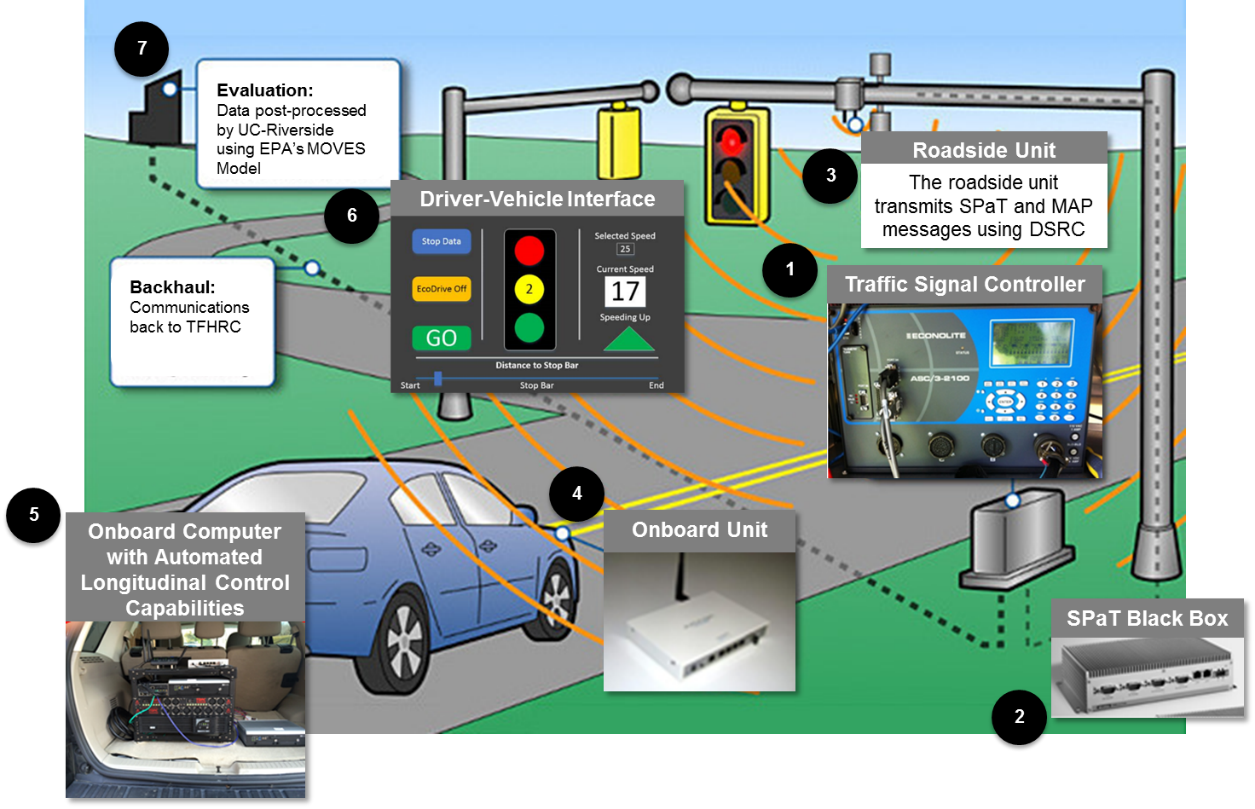


Figure GlidePath system components

As shown in Figure 2, the standard signal controller (1) drives the signal heads that drivers see and provides its current timing data to a computer in the intersection cabinet called the “black box” (2). The black box transposes the signal timing data into a standard SPaT message, which it forwards to the roadside unit (RSU) radio that is mounted near the intersection (3). The black box forwards SPaT messages 10 times per second (10 Hz) and also forwards a static MAP message, describing the intersection geometry, to the RSU once per second. The RSU broadcasts each of these messages as they are received. Within the test vehicle and onboard unit (OBU) radio (4) receives the DSRC messages from the RSU. It then forwards them as received to the control computer (5) that houses the longitudinal control software, also known as the GlidePath application. This application executes the EAD algorithm to compute the desired trajectory and sends control signals to the vehicle’s XGV robotic controller, which actuates the throttle and brake to achieve the commanded speed. In addition to receiving input from the OBU, the application software also receives current vehicle position data from the vehicle’s PinPoint navigation component. The PinPoint unit combines a two-antenna GPS solution with an inertial measurement unit to generate high accuracy location data. In practice we have observed that the measured location typically has an error less than 2 meters, which is sufficient for determining which lane the vehicle is in, since each lane is nearly 4 meters wide. As the application software is executing it displays a driver-vehicle interface (DVI) on a tablet computer mounted to the dashboard (6). The control computer is connected to this tablet via WiFi. The DVI shows the driver the current signal phase and timing information for the lane that the vehicle is in, and allows the driver to select an “operating speed”, which is the desired cruise speed for the experiment. It also allows the driver to signal when to begin automated driving (at the beginning of an experiment the vehicle will be at rest until the driver gives the Go command). Once underway, the DVI will continue to show the signal timing information, along with an indicator of the distance to the stop bar and the vehicle’s current speed. Throughout an experiment run the application generates a log file that contains lots of internal information about the software’s logic and intermediate computations, as well as columnar data of the vehicle state and input data suitable for plotting. These data are stored indefinitely on the control computer and can be downloaded to the laboratory for later analysis.

The vehicle can create a trajectory, using the UCR EAD algorithm, that fits into one of the four scenarios shown in Figure 3 below. Each time the vehicle approaches an intersection a new trajectory shape is computed on the fly, depending on the exact vehicle speed, distance to the intersection, and signal timing for the approach lane at the time of decision making. Scenario 1 represents the ideal situation. It will occur when the signal is green on the approach lane at the time the vehicle anticipates arriving at the intersection. Since speed is held constant throughout, fuel economy is maximized; travel time is also as if the intersection were absent. If the algorithm determines that a constant speed is not possible, then it tests for scenario 2, which requires a slight speed-up, while staying within the posted speed limit. This scenario applies if the signal would otherwise turn red just before a constant speed arrival. If neither of these scenarios is possible then the algorithm attempts to create a slow-down trajectory like scenario 4. Scenario 4 applies when a constant speed trajectory would approach near the end of the red phase. Slowing the vehicle as little as possible to allow the red phase to expire just before the vehicle arrives will maximize fuel economy without violating the signal constraint. There is a lower limit “crawling speed” below which the scenario 4 trajectory cannot go, due to limitations in the control hardware. If none of these scenarios is possible then scenario 3 applies, which brings the vehicle to a full stop using an eco-friendly trajectory. It will remain stopped at the intersection until the red phase expires, and then will continue to an automated, eco-friendly departure after the driver acknowledges that it is safe to resume motion.

In determining which scenario is appropriate, the algorithm treats the signal’s yellow phase as if it were red; that is, it will not plan to enter the intersection under a yellow light.

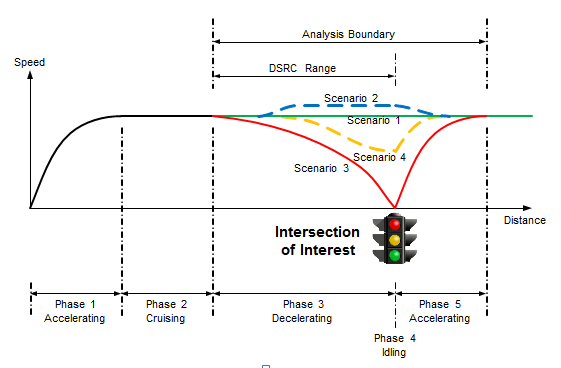


Figure Speed profile scenarios

# Eco-Approach and Departure Algorithm

## Architecture of Vehicle Trajectory Planning Algorithm (VTPA)

The core of the GlidePath prototype system is the Vehicle Trajectory Planning Algorithm (VTPA), a block diagram of which is shown in Figure4. Its external inputs include:

* Vehicle’s current state, such as GPS location (i.e., latitude, longitude and altitude) received from the PinPointTM system and instantaneous velocity obtained from the vehicle CAN Bus interface (see Figure 5);
* Message sets received from the DSRC device, which include SPaT and MAP messages; and
* System constraints and parameters, such as maximum acceleration and deceleration, maximum jerk (i.e., derivative of acceleration), and the roadway speed limit.



Figure Block diagram of VTPA

The output is the target velocity. A further insight into the VTPA system is illustrated in Figure 5, where there are five sub-systems:

1. *MAP Parser*: by following FHWA format, this sub-system can decode the MAP messages broadcasted by the road-side equipment (RSE) and extract the characteristics of key nodes at/around the intersection, such as the indices of approach and lane, latitude and longitude;
2. *Map Matcher*: based on the vehicle’s current location, stop-bar location, and locations of those key nodes in-between, this sub-system can estimate the vehicle’s distance to the stop-bar or distance-to-arrival (DTA) at each time step;
3. *SPaT Parser*: Also by following FHWA format, this sub-system can decode the SPaT messages broadcasted by the RSE and extract the current signal status (i.e., green/yellow/red and solid/arrow) of each movement as well as the range (the minimum and maximum) of count-down to the end of current status;
4. *Green Window Estimator*: this sub-system aims at estimating available green windows for the subject vehicle according to the vehicle’s desired movement, SPaT and signal controller’s type (e.g., fixed-time or traffic-responsive) and settings. For the fixed-time signal controller used in this project, the green window is known and no estimation is necessary. But for a traffic-response signal controller, more advanced statistical techniques need to be employed to obtain relatively more reliable estimation of green windows, depending on the degree of actuation [Hao et al., 2015]. The sub-system is incorporated here as a place-holder for the design of a general purpose algorithm.
5. *Decision Maker*: this sub-system takes into account the holistic information, including system constraints (e.g., maximum acceleration/deceleration, roadway speed limit), to identify the scenario that the subject vehicle faces and determine the appropriate target speed profile. As shown in Figure 6, this sub-system consists of two components: 1) Scenario Identifier; and 2) Trajectory Generator.



Figure Sub-systems of VTPA



Figure Diagram of the decision maker sub-system

## Scenario Identifier

As shown in Figure 7, the Scenario Identifier component selects the appropriate scenario for the trajectory, based on some key parameters (such as speed, SPaT, distance to intersection and other system constraints) at current time. For example, if the subject vehicle can cruise at the current velocity and pass the intersection at green, then the trajectory is categorized into Scenario 1 (cruise), and the cruise time, , is given as:

(1)

where is the route distance to the stop-bar and is the instantaneous speed at current time instance, . In addition, the available green window, , can be written as follows:

(2)

where denotes the end of current green window associated with the vehicle’s movement; and represent the start and end of next green window, respectively. Generally speaking, should be the set of all subsequent green windows after . But within the limited communication range of DSRC (300 meters, nominally), the time window up to the end of next green should be practically long enough to handle with most situations (except for some over-saturated traffic conditions).



Figure Scenario identifier details showing flow of control through the logic

If Scenario 1 is not guaranteed, then the earliest time to arrival, , will be calculated to determine whether the trajectory satisfies the condition of Scenario 2, i.e., speed-up (without breaking the speed limit) to pass through the signal without any stop. The calculation of largely depends on the proposed trajectory model, a piecewise trigonometric-linear function, which will be elaborated in the following section.

If it is determined that the subject vehicle will not be able to pass the intersection by moderate acceleration, then the vehicle has to decelerate to a full stop (Scenario 3) or to “glide” in an environmentally friendly manner (Scenario 4), depending on the latest time to arrival without any stop, . Again, the calculation of is model-dependent.

## Trajectory Generator

This component determines the time-of-arrival, (for Scenario 3, it is the time until departing from the stop-bar), and the target vehicle trajectory for the selected scenario, as shown in Figure 7. As mentioned in Section 2, the control logic for the target velocity tries to minimize the vehicle’s acceleration/deceleration before the intersection, so that the vehicle can pass the intersection with the target speed that is closest to its initial speed (assuming it is the free-flow speed). Therefore, after passing the intersection, the vehicle can get back to its initial speed with minimal fuel usage. As suggested in previous studies, there are numerous ways to accelerate or decelerate from one speed to another, such as the constant acceleration and deceleration rates, linear acceleration and deceleration rates, and constant power rates; the family of piecewise trigonometric-linear function is used here as the target velocity profiles (for both approach and departure portions), due to its mathematical tractability and smoothness [Barth et al., 2011].

Figures 8 and 9 depict general profiles for acceleration and deceleration, respectively. The acceleration and deceleration are designed to achieve the desired cruise speed in the shortest amount of time, while ensuring the driving comfort by limiting the jerk. To avoid unnecessary idling, the vehicle tries to reach the intersection during the green phase of the signal.



Figure Acceleration profile of the piecewise trigonometric-linear function



Figure Deceleration profile of the piecewise trigonometric-linear function

For Scenario 1, since the vehicle is able to cruise through the intersection, the time-to-arrival, , and the target velocity, , is simply the current velocity (at , without loss of generality), .

For Scenario 2, the approach portion takes the similar shape of acceleration profile in Figure 8. To reach back to after passing the signal, the departure portion is the mirror symmetry of the approach one for simplicity. More specifically, without compromising the travel time, the time-to-arrival is given as

(3)

The target velocity, , where

(4)

and (>0) is chosen as the maximum that satisfies:

(5)

and,

(6)

where ; ; ; ; ; and are the maximum acceleration and deceleration, respectively; is the maximum jerk whose value can be chosen as recommended in [Yi and Chung, 2001]; The parameters and define the family of trigonometric functions, whose values control the rate of change in acceleration and deceleration profiles. In addition, the parameters and are coupled in order to guarantee the smoothness of entire speed profile (especially at those break points) and the area under the curve being the distance to the stop-bar, .

According to the Equation Set 4, the earliest time-to-arrival, , can be calculated as

(7)

and

(8)

where, represents the upper limit (hard constraint) of the target velocity due to the subject vehicle’s ability or roadway enforcement.

As aforementioned, to determine if the speed profile belongs to Scenario 3 or Scenario 4, the latest time-to-arrival without any stop, , can be calculated as

(9)

and

(10)

where denotes an arbitrarily small coasting speed that is a user-defined parameter based on driving comfort.

For Scenario 3, since the vehicle needs to have a full stop at the stop-bar, the time-to-arrival (or time to leave from the stop-bar), and the target velocity, , where

(11)

and,

(12)

where ; ; and ;

For Scenario 4, the time-to-arrival is given as

(13)

And the target velocity, shares the same format of , but . Figure 10 summarizes how to determine the time-to-arrival, , the average speed, , and the target speed, , for each of the four scenarios.



Figure Trajectory generator details, showing flow of control for each of the four scenario outputs in Figure 7

# The GlidePath Vehicle

The GlidePath prototype vehicle is a 2010 Ford Escape Hybrid outfitted by TORC Robotics® with their ByWire XGV (experiment ground vehicle) robotic control system, enabling full-range longitudinal and lateral control. The on-board XGV control unit communicates with the vehicle’s low-level components via CAN bus, which allows it to read data on wheel speed, steering wheel angle, and many other pieces of information. It also allows it to issue commands to the throttle and brake, which allows for automatic longitudinal control. Under the GlidePath project the team integrated the following additional components into the test vehicle:

* Emergency stop and manual override controls;
* Arada DSRC on-board unit (OBU) and roof-top antenna;
* PinPoint advanced positioning system with twin roof-top antennas;
* Samsung tablet mounted to the dashboard as a driver-vehicle interface (DVI);
* Wi-Fi router;
* 3G cellular data modem for remote access to the vehicle network;
* On-board computer running Ubuntu Linux;
* Power converters to operate the added electronic equipment;
* Cabling to support the 2012 HMI device mounting to the dashboard.



Figure Electronic equipment and driver-vehicle interface (DVI)

Note that the DVI tablet, used for driver interaction during automated operation, is distinct from the HMI device, which was used for the manual-HMI-assisted driving. During automated driving the DVI tablet informs the driver of the signal timing situation, current vehicle speed and distance to stop bar. It gives no information about the algorithm’s computations, or even which scenario it has selected. During manual driving, on the other hand, the HMI also shows signal timing information, but its emphasis is on displaying the algorithm’s recommended speed at any given time, along with the actual speed, so that the driver can manually attempt to match the recommendation.

The project was structured from the outset to minimize the risk of injury or property damage inherent in operating an experimental automated vehicle. The vehicle had several safety features built into it, including external emergency stop buttons, a console manual override button, and a special brake pedal sensor that would disengage the XGV robotic control system. In designing the experimental concept and the control system we expended significant resources to develop a detailed state model for the vehicle system, communicating all of the possible state transitions and how those transitions could be enacted. This model was presented in the system design document, and used as the basis for the speed control software design.

# Testbed Infrastructure

The signalized intersection at TFHRC is the centerpiece of the GlidePath testbed. It is controlled by an Econolite ASC/3 traffic signal controller that is connected to the TFHRC network. On this network is the signal’s “black box” that disseminates signal information to other network users. The black box takes signal phasing information from the controller and reformats it into FHWA formatted SPAT messages [[2]](#footnote-2)and broadcasts them over the network. The black box also broadcasts MAP messages in FHWA format. For this project the controller was configured to provide a fixed timing plan for the signal phases, where each of the four approach routes experienced the same pattern: 27 sec green, 3 sec yellow, 30 sec red.

Figure 12 is an aerial view of the test route, showing the locations of the various hardware elements. An Arada roadside unit (RSU) DSRC device was added to the signal mast arm (point 1 in Figure 12). This RSU was connected to the infrastructure network and was configured to broadcast the SPAT and MAP messages, which could then be received by the vehicle’s OBU. System integration testing (described below, in Section 8) determined that the intersection geography limited the RSU transmission range, due to tree coverage and a hill in the road. The team therefore mounted a second RSU along the primary approach route (point 2 in Figure 12) to provide sufficient radio coverage throughout the experimental route. These two RSUs were connected so that they provide redundant, simultaneous broadcasts.

To ensure that the SPAT signals are interpreted correctly by the vehicle’s speed control software, it needs an accurate representation of the intersection geometry to determine which lane the vehicle is in and its distance to the stop bar. The team constructed aMAP message from detailed survey data to represent this geometry. It accounted for five approach lanes (one each from the west, north and east, and two lanes from the south) and two departure lanes (eastbound and westbound) throughout the extent of possible experimental maneuvers. The MAP message describes the centerlines of each of these lanes by way of a series of latitude/longitude nodes, including the stop bar location for each approach lane. It also describes the lane widths and maneuvers allowed (straight through, left turn only, etc). The MAP message also broadcasts a unique identifier for the intersection along with a version ID for the message so that receivers can compare it to previously received messages for possible changes in the content. Both the SPAT and MAP messages were broadcast at 10 Hz so that the approaching vehicle would be able to receive the information quickly as soon as it arrived within broadcast range of an RSU.

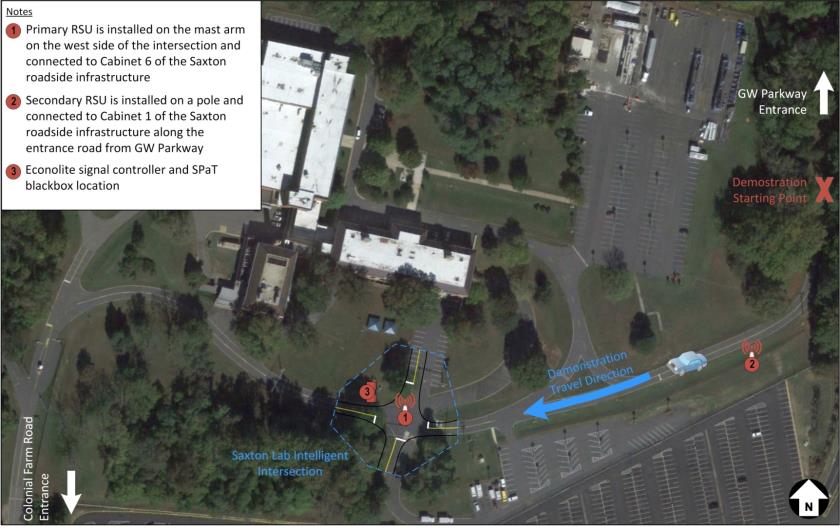


Figure Overhead view of the testbed, showing signal, cabinet and RSU locations

# Engineering Process

The engineering work on the GlidePath project followed standard Leidos systems engineering practices to ensure maximum quality and efficiency of work. These practices involved requirements management, software and deliverable configuration management, formal design work, peer reviews of deliverable documents and of all software components, and multiple levels of testing. Each deliverable document was submitted in draft for FHWA review and comment before a final version was delivered.

The requirements specification was developed based on the Eco-Signal Operations Concept of Operations provided by FHWA. Requirements were elicited from various stakeholders and reviewed with them before being finalized and approved by FHWA. Changes to the requirements after this approval were minimal, but were carefully documented, controlled and agreed to by FHWA, based on the understood project impacts.

Based on the accepted requirements, a system design was developed, covering all aspects of the vehicle, the roadway infrastructure, and the speed control software. Considerations were made to maximize reuse of existing components where practical. The design was documented in a System Design Document and delivered as a preliminary design, addressing the more abstract issues and decisions, then a final version that presented the information in more detail.

The speed control software was designed to the various interfaces provided by the vehicle components, including CANBus, XGV robotic controller, GPS, DSRC device, and Ethernet/Wi-Fi access to the DVI tablet. The speed control software is described in more detail below. During its development, the engineering process included source code version control in a private GitHub repository. Each new module and change was made in its own repository branch, and was subjected to a peer review before it was merged into the main integration branch. In parallel to development of the operational code, automated unit tests were created using the JUnit framework to exercise as much of the code as could be practically done in the development environment. These tests were executed each time the software was built, in order to continuously verify correct functionality at the lowest level.

The speed control software was tested at three levels of detail to verify that it met the design and stated functionality requirements. First was automated unit testing mentioned above. Unit test software has been delivered alongside the production code.

Once all of the software was in place and passing the set of unit tests, the code was tagged with a software version ID, which is printed at the top of each run log. The version IDs began with 1.0 for the first full integration testing.

System integration testing was performed by driving the vehicle through the testbed course. This testing was informal and evaluated the vehicle’s performance based on externally observable phenomena (e.g. trajectory performance) as well as detailed evaluation of the speed control software’s logs. These logs capture intermediate details of the data handling and trajectory computation activities, giving engineers plenty of insight into how the software, and system as a whole, is working throughout the route.

After completion of the integration testing, the formal customer acceptance tests were performed. An acceptance test plan was generated, which included cases specifically designed to exercise all of the testable requirements. This plan was reviewed and accepted by the FHWA prior to formal testing. The formal acceptance tests involved project staff operating the system, with FHWA representatives riding in the vehicle, observing that the test cases were performed according to the approved plan. As each case was successfully demonstrated, the FHWA representative signed off on that case. Acceptance testing was performed on software version 1.3. Several questions arose during testing, and minor red-line changes to the test plan were requested by FHWA during the testing. These issues were recorded in a comment resolution matrix, and all were addressed with the final test report and with updates to various other documents. Those that were significant enough are captured in the Lessons Learned section later in this document.

Once the system was accepted, a change request backlog was begun. This simple spreadsheet is used to record any ideas for changes, whether they are enhancements or descriptions of observed anomalous behavior. From time to time, during the demonstration phase after acceptance, and during the non-operational period (from approximately Oct 2015 to Sep 2016 due to intersection construction), software improvements were made by addressing various change requests recorded in this backlog.

# Speed Control Software

The speed control software was designed to the allocated system level requirements. It is a Java executable built on the Spring Boot framework. Spring Boot provides an embedded Tomcat web service, which allows the application to act as a complete web server packaged as a stand-alone jar file (no dependency on other system components). This software executes on the on-board Linux computer, and interacts with several other components on the vehicle’s network, including the Arada OBU, the PinPoint positioning system, the DVI tablet (via Wi-Fi), and the XGV controller.

The speed control software implements the rules of the EAD algorithm, running it at 10 Hz. Within each time step the software pulls data from the Arada, PinPoint and XGV sensors, and converts that raw data into usable information on vehicle speed, acceleration, position, and current signal phase situation. It then forwards this information to the EAD algorithm. The algorithm will then generate a new speed command for that time step.

Once the vehicle is reliably receiving SPAT and MAP messages from the intersection infrastructure, and is positioned within the boundaries defined by the MAP data, the driver can command the software to begin the experiment (via a button push on the DVI). The experiment begins with the vehicle sitting still in a defined approach lane. When the driver commands the experiment to begin, the software takes control of longitudinal motion, accelerating the vehicle, within configurable acceleration and jerk limits, up to the driver-selected operating speed. Upon reaching the operating speed, the software then checks the distance to stop bar; if it is farther away than the configurable experiment start distance, then the vehicle simply continues to cruise at the operating speed. Once the start distance is crossed (our experiments set this at 190 m in front of the stop bar), the EAD algorithm is invoked for the first time, whereupon its scenario identifier determines the best trajectory shape to be used for the situation at hand. At that point the software freezes the trajectory selection[[3]](#footnote-3) and all remaining time steps use the trajectory generator aspect of the algorithm to calculate a new speed command as the trajectory is traversed. All output speed commands are sent back to the XGV controller, which converts them into throttle and brake activation signals.

The speed control software includes a fail-safe algorithm that is independent of the EAD algorithm. The purpose of the fail-safe module is to ensure that the vehicle comes to a stop to avoid running a red light, regardless of what the EAD algorithm may be commanding. When it determines that a red light violation may be imminent it overrides the EAD output command with a more aggressive stopping command. A configuration flag allows the fail-safe code to be disabled if desired.

If the EAD algorithm (or the fail-safe logic) brings the vehicle to a stop at a red light, then the EAD algorithm will also command a smooth acceleration departure trajectory when the light turns green. However, the speed control software includes an additional safety feature, wherein it requires explicit driver input before resuming motion. This gives the driver a chance to ensure that the path of travel is clear of pedestrians or other traffic before proceeding through the intersection.

Once past the intersection, the speed control software will bring the vehicle back to the selected operating speed until a designated experiment end distance has been reached (our experiments set this at approximately 100 m beyond the stop bar). Once this distance is reached the software will shut down and return the vehicle to fully manual control, as if the emergency override button had been pressed.

The speed control software has a configurable capability to log varying degrees of detail regarding its operation. It can be set up to only report on critical error situations, or three additional levels of detail, up to full debugging capability, where lots of intermediate calculation results are routinely written to the log file. In addition to the information logged about the software’s operation, it also records experimental data at the end of each time step. This experimental data includes quantities about the vehicle state and the EAD algorithm inputs and outputs. These data are recorded in a format that makes them easy to import into Excel for plotting and trajectory analysis. Results of these analyses are reported in section 10.2.

# Demonstrations

After the system was formally accepted, it was used to provide several demonstrations of the system to a wide range of interested parties from across the USDOT and outside organizations, including the White House Office of Science and Technology Policy.

During this demonstration phase, which lasted for several months during 2015 and included numerous dry runs and informal evaluation runs, the project staff and FHWA project management continued to identify opportunities to enhance the system’s performance. These improvements were incorporated into updates of the speed control software, resulting in new software releases identified as versions 1.4, 1.5 and 1.6, each of which was deployed to the vehicle and used in one or more demonstrations.

During this period the demonstration process evolved, eventually including a rigorous script dictating preparation activities and demonstration scenarios, as well as keeping a log of each demonstration delivered.

In October 2015 a professional video production company was hired to produce several representative videos of the vehicle in action. These videos are available for public outreach, and one is even available on YouTube

(https://www.youtube.com/watch?v=I753gGLJAcg&feature=youtu.be).

# Algorithm Performance Evaluation

## Data Collection

After the demonstration phase was well underway and the speed control software was consistently performing as desired, the EAD algorithm performance evaluations began. All of the data collection runs described here were performed with speed control software version 1.6.

There were some general observations developed during these test runs and tuning of the system prior to doing these test runs. These observations are noted here:

* Acceleration limits: initial limits were set at 3.0 m/s2, which created a rough ride. The team eventually settled on values in the 1.8 to 2.0 m/s2 range as being comfortable, yet still providing reasonable performance.
* Jerk limits: the software has been configured to limit jerk to 10 m/s3, and none of the runs ever came close to that level. It is apparently not within the capability of the vehicle hardware to provide a jerk more than around 4 m/s3, which seems to be satisfactory for all drivers and passengers who have experienced it.
* The horizontal error of the PinPoint component was normally seen to be < 2 m, which is satisfactory for this experiment. This is enough to distinguish which lane the vehicle is travelling in. It is also marginally sufficient to guide the vehicle to the correct stop bar location, as it is a little less than half of the vehicle length. In this project it was acceptable for the vehicle to stop with its front bumper +/- 2m from the stop bar. However, a more exact location will be desirable in a more realistic environment, with other vehicles in the scene, multiple travel lanes, and possible pedestrians in crosswalks.
* A corollary to the vehicle location accuracy is the accuracy of the MAP message. The message used in this experiment was constructed from survey data using WGS-84 ellipsoid calculations to define offsets for the lane centerlines. The MAP was perfectly adequate for this experiment. However, it did not include speed limit information, which was given to the software directly through a configuration parameter.
* The 190 m experiment distance was chosen as the largest distance that would allow the experiment to be run on fairly level terrain (starting any farther away would have involved a significant hill).

The field experiment was designed to be comprehensive in that the test vehicle will approach the intersection at different times throughout the entire signal cycle (i.e., every 5 seconds in the 60-second cycle). Furthermore, the vehicle approached the intersection at different driving speeds (i.e., operating speeds), ranging from 20 mph to 25 mph. The limitations of the TFHRC facility roadway prevent use of higher operating speeds. Since the test vehicle was a gas/electric hybrid, it was not possible to directly measure fuel consumption as a normal gasoline powered car would experience. Thus the vehicle fuel economy and CO2 emissions were calculated by applying the Comprehensive Modal Emissions Model (CMEM)[[4]](#footnote-4) to the logged trajectories, and compared between the following stages:

* Stage I: **“manual-uninformed” driving**. At this stage, a driver approached and traveled through the intersection in a normal fashion without guidance or automation, stopping as needed without any automated vehicle control. Data collected at this stage establish a baseline that can be used as a point of comparison for the Stage II and III experiments.
* Stage II: **“manual-HMI-assisted” driving**. At this stage, a driver was provided an enhanced dashboard which presented a speed range band overlaid onto a speedometer for the driver to follow as guidance on how to approach and depart the intersection in an environmentally friendly manner while obeying the traffic signal (see Figure 13). This stage does not involve any automated vehicle control but the advisory speed trajectories were generated using the same core algorithm described in Vehicle Trajectory Planning Algorithm Section. For this stage, the recommended speed profile will be re-calculated if the subject cannot follow the recommendation well enough (i.e., the accumulative following error reaches the user-defined threshold). Such mechanism may trigger the change from one scenario to another en route.



Figure Graphic interface for “manual-HMI-assisted” driving

* Stage III: **“(partially) automated” driving**. At this stage, the developed GlidePath prototype system was responsible for longitudinal control of the vehicle allowing it to speed up or slow down while the driver steered for lateral control and monitored the application on the DVI (see Figure 14). At this stage, the vehicle automatically controlled the brake and throttle based on the output of the VTPA, which calculated an eco-friendly velocity profile according to the DSRC message sets and distance to the stop-bar.



Figure DVI graphic interface for “(partially) automated” driving

For the Stage I and Stage II experiments, four drivers who had not worked on the Eco-Approach and Departure concept were recruited to conduct test runs. For the Stage III experiments, the test vehicle was operated by a trained driver to maintain safety as a top priority. Because the GlidePath prototype system can automatically control the longitudinal motion of the vehicle, it is not necessary for novice drivers to operate the vehicle during the “(partially) automated” driving stage.

### Data Logged

As described in Background Section and Vehicle Trajectory Planning Algorithm Section, the data for trajectory planning and system performance evaluation were collected from two major sources: 1) roadside infrastructure, and 2) XGV (i.e., Ford Escape Hybrid).

On the infrastructure side, the logged data include the geographic information of the testing route as well as settings and dynamic data stream from the traffic signal controller (i.e., Econolite 2070). The route geometric information was used to determine the test vehicle’s distance to the stop-bar in real-time, based on the stop-bar location and route geometry encoded in the MAP messages (broadcasted by the RSU) and vehicle’s GPS coordinated (i.e., latitude, longitude) reported by the PinPointTM localization system. Regarding the traffic signal controller, both the signal timing settings (pre-timed) and dynamic signal information were sent to the DSRC on-board unit (OBU) and logged. The former includes intervals of “Green”, “Yellow”, “All-red-clearance” and “Red” phases, while the latter envelops the current phase and count-down time (i.e., time to the next phase) which should be decoded from the SPaT messages. The time to the next phase used in the algorithm was also derived from the pre-timed plan under fixed-time signal control.

On the vehicle side, the logged data are focused on the test vehicle’s dynamic states (at 10 Hz), such as vehicle location, actual speed, recommended speed, acceleration and jerk and distance to stop-bar which were used for estimating instantaneous energy consumption and pollutant emissions of the XGV by applying the CMEM model.

### Field Study Matrix

In order to cover every possible driving scenario (as mentioned in Section II), a field study matrix (see Figure 15) that varies the vehicle’s operating speed and signal timing start with respect to the overall cycle of the traffic signal, was developed for each driver at each stage (as used in [Wu et al., 2014]). This test matrix consists of the operating speed along the vertical axis, and the delay in the signal cycle across the horizontal access as well as the expected current phase of the traffic signal. In this matrix, there are a total of 12 (intervals) × 2 (speed levels) = 24 test cells. For the experiments, the drivers had to drive through the intersection at least once in order to fill out the field study matrix for each cell. Therefore, a total of 24 × 3 (stages) × 4 (drivers) = 288 test runs were conducted. For each data run, key data elements were logged at 10Hz and post-processed to determine energy consumption and other performance measures.

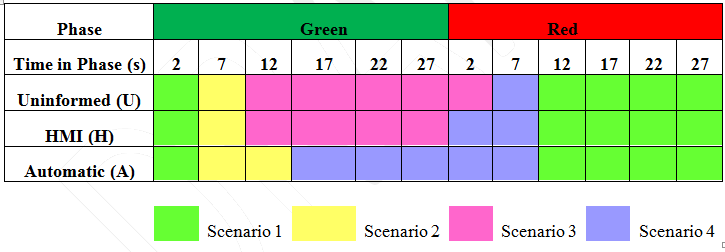


Figure An example of field study matrix for data collection and presentation

## Results and Analysis

Due to the applicability of the CMEM model, results in this Section are based on the second-by-second trajectories aggregated from higher resolution (e.g., 10 Hz) data. Comparative studies between different stages have been conducted on an interval basis and on a scenario basis, respectively.

### Interval-Based Comparison

As aforementioned, field study matrices have been created to facilitate the presentation of test results. Since there are multiple drivers involved in the data collection for Stage I and Stage II experiments, aggregate statistics are calculated for comparison. Median is more robust statistics than Mean [Huber, 1981], therefore, the medians of performance measures are evaluated in this study.

#### Fuel Consumption

Table 2 summarizes the results on fuel consumption per distance for different stages (i.e., “manual-uninformed” driving, “manual-HMI-assisted” driving, and “automated” driving) at different entry speeds (20 mph and 25 mph). As shown in the table, the “automated” driving significantly outperforms (on average) either “manual-uninformed” driving or “manual-HMI-assisted” driving in terms of fuel consumption at different entry speeds. The “manual-HMI-assisted” driving performs (on average) better than “manual-uninformed” driving when entry speed is 25 mph but there is trivial difference between these two stages when entry speed is 20 mph. For better interpretation, each cell is colored based on the scenario into which the test run sample is categorized. It can be observed that:

* Scenario 2 cases occurred for “automated” driving but neither for “manual-uninformed” driving nor “manual-HMI-assisted” driving. The hypothesis is that for Stage I there is no information on when the phase will change, while for Stage II the driver (“median” one) may be conservative or it may be nontrivial to consistently follow the recommended speed displayed on HMI such that the “speed-up and pass” opportunity is missed.
* Scenario 3 NEVER occurred for “automated” driving under the settings in this study, but it happened multiple times for “manual-uninformed” driving and “manual-HMI-assisted” driving. As mentioned in Data Collection Section, for “manual-HMI-assisted” driving, an en-route scenario change from 4 to 3 may occur if the driver cannot follow the recommendation very well.
* On the cells of “Green 7” (when entry speed is 20 mph), the “uninformed” driver (“median” one) barely passed through the intersection at yellow while “manual-HMI-assisted” driver (“median” one) did not get through the signal even with assistance. A potential explanation is the “manual-HMI-assisted” driver (“median” one) failed to closely follow the speed advice (which could be updated if the threshold of accumulative following error was reached) at the beginning, then the scenario changed from 2 to 3.

Table 2. Median of Normalized Fuel Consumption (across All Drivers) for Each Test Cell (gram/mile)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Spd, mph** | **Phase** | **Green** | | | | | | **Red** | | | | | | **Avg** |
| **TIPa** | **2** | **7** | **12** | **17** | **22** | **27** | **2** | **7** | **12** | **17** | **22** | **27** |
| **20** | **Ub** | 58.1 | 60.9 | 143.1 | 135.7 | 126.7 | 126.2 | 126.5 | 111.6 | 57.9 | 56.6 | 60.2 | 61.1 | 93.7 |
| **Hc** | 63.8 | 111.1 | 137.3 | 128.0 | 116.4 | 122.0 | 122.8 | 80.8 | 61.5 | 57.0 | 60.9 | 60.3 | 93.5 |
| **Ad** | 54.5 | 56.1 | 92.4 | 107.8 | 101.6 | 92.8 | 80.9 | 59.8 | 56.8 | 51.8 | 51.4 | 53.6 | 71.6 |
| **25** | **U** | 62.1 | 55.3 | 159.6 | 157.2 | 153.8 | 146.2 | 141.1 | 136.5 | 109.1 | 58.2 | 60.3 | 59.5 | 108.2 |
| **H** | 54.7 | 58.7 | 144.5 | 148.5 | 140.1 | 138.7 | 139.2 | 130.9 | 71.3 | 55.3 | 53.8 | 55.2 | 99.2 |
| **A** | 48.4 | 50.0 | 97.2 | 142.1 | 134.4 | 132.1 | 130.3 | 97.5 | 75.3 | 49.3 | 49.1 | 46.7 | 87.7 |

a Time-In-Phase, seconds, when the test vehicle entered the region (i.e., 190 meters to the stop-bar);

b “Manual-uninformed” driving stage (i.e., Stage I);

c “Manual-HMI-assisted” driving stage (i.e., Stage II);

d “Automated” driving stage (i.e., Stage III). In addition, for this stage, there is only one value in each cell.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Scenario 1 |  | Scenario 2 |  | Scenario 3 |  | Scenario 4 |

To get further insight into the improvement in fuel economy for the GlidePath prototype system, relative changes with respective to the “median” driver and the median of relative changes across all drivers have been calculated. The results are shown in Tables 3 and 4, respectively. As can been seen from the tables, results from both tables are quite consistent, and the difference between counterpart cells is slightly less than 5%. On average, the “automated” driving can consistently save about 18% to 20% fuel, compared to the “manual-uninformed” driving (the “median” case). The performance of “manual-HMI-assisted” driving (the “median” case) seems to be varying with entry speed. For example, when entry speed is 25 mph, the “manual-HMI-assisted” driving can consume about 8% fuel than the “manual-uninformed” driving. However, it works worse than the “uninformed” case for entry speed of 20 mph.

Table 3. Relative Improvement (%) between Stages with Respect to Median of Normalized Fuel Consumption

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Speed (mph)** | **Phase** | **Green** | | | | | | **Red** | | | | | | **On Avg.** |
| **TIPa** | **2** | **7** | **12** | **17** | **22** | **27** | **2** | **7** | **12** | **17** | **22** | **27** |
| **20** | **H vs. U** | -9.8 | -82.4 | 4.1 | 5.7 | 8.1 | 3.4 | 2.9 | 27.6 | -6.1 | -0.8 | -1.1 | 1.2 | -3.9 |
| **A vs. U** | 6.2 | 7.9 | 35.5 | 20.6 | 19.8 | 26.5 | 36.0 | 46.4 | 2.0 | 8.4 | 14.7 | 12.2 | 19.7 |
| **A vs. H** | 14.5 | 49.5 | 32.7 | 15.8 | 12.7 | 23.9 | 34.1 | 26.0 | 7.6 | 9.1 | 15.6 | 11.2 | 21.1 |
| **25** | **H vs. U** | 11.9 | -6.2 | 9.5 | 5.6 | 8.9 | 5.2 | 1.4 | 4.1 | 34.7 | 5.0 | 10.7 | 7.1 | 8.1 |
| **A vs. U** | 22.0 | 9.6 | 39.1 | 9.6 | 12.6 | 9.6 | 7.7 | 28.6 | 31.0 | 15.3 | 18.6 | 21.4 | 18.8 |
| **A vs. H** | 11.5 | 14.9 | 32.8 | 4.3 | 4.0 | 4.7 | 6.4 | 25.5 | -5.6 | 10.8 | 8.8 | 15.4 | 11.1 |

Table 4. Median of Relative Improvement (%) between Stages (across All Drivers) for Normalized Fuel Consumption

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Speed (mph)** | **Phase** | **Green** | | | | | | **Red** | | | | | | **On Avg.** |
| **TIPa** | **2** | **7** | **12** | **17** | **22** | **27** | **2** | **7** | **12** | **17** | **22** | **27** |
| **20** | **H vs. U** | -7.0 | -70.7 | 0.8 | 2.9 | 8.1 | 0.9 | -3.6 | 19.4 | -0.4 | 0.2 | -2.2 | 4.9 | -3.9 |
| **A vs. U** | 6.2 | 7.9 | 35.5 | 20.6 | 19.8 | 26.5 | 36.0 | 46.4 | 2.0 | 8.3 | 14.5 | 11.3 | 19.6 |
| **A vs. H** | 14.5 | 46.4 | 32.7 | 15.8 | 12.7 | 23.9 | 34.0 | 25.8 | 7.6 | 9.0 | 15.6 | 11.2 | 20.8 |
| **25** | **H vs. U** | 10.5 | -8.3 | 9.4 | 6.8 | 10.2 | 4.1 | 2.4 | 6.6 | 36.1 | 3.4 | 6.4 | 4.4 | 7.6 |
| **A vs. U** | 21.7 | 9.6 | 39.1 | 9.6 | 12.6 | 9.6 | 7.7 | 28.6 | 31.0 | 15.2 | 18.2 | 21.4 | 18.7 |
| **A vs. H** | 11.5 | 14.7 | 32.7 | 4.2 | 4.0 | 4.6 | 6.3 | 25.4 | -6.3 | 10.5 | 8.8 | 15.3 | 11.0 |

Table 5 presents the standard deviation of relative improvement on fuel consumption per distance across all drivers between stages. It can be observed from the table that compared to the “manual-HMI-assisted” stage, the “automated” driving performs much more robustly. In other words, the “automatic” driving can provide much higher fuel savings over the “manual-uninformed” stage and with much less variation. For example, as the entry speed changes from 20 mph to 25 mph, the standard deviation of relative fuel reductions (on average) offered by the “manual-HMI-assisted” driving vary from 15.2% to 18.5%, while the range provided by the “automatic” driving is only between 6.3% and 9.0%.

Table 5. Standard Deviation of Relative Improvement (%) between Stages (across All Drivers) for Normalized Fuel Consumption

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Speed (mph)** | **Phase** | **Green** | | | | | | **Red** | | | | | | **On Avg.** |
| **TIPa** | **2** | **7** | **12** | **17** | **22** | **27** | **2** | **7** | **12** | **17** | **22** | **27** |
| **20** | **H vs. U** | 4.00 | 59.58 | 5.35 | 12.16 | 2.97 | 6.96 | 17.58 | 13.15 | 10.76 | 20.19 | 10.65 | 19.40 | 15.23 |
| **A vs. U** | 7.28 | 5.77 | 2.69 | 6.89 | 3.29 | 5.09 | 0.98 | 6.16 | 8.79 | 10.02 | 7.15 | 11.47 | 6.30 |
| **A vs. H** | 6.29 | 20.88 | 3.38 | 5.30 | 6.13 | 4.06 | 11.33 | 11.73 | 5.52 | 9.72 | 2.49 | 9.25 | 8.01 |
| **25** | **H vs. U** | 8.81 | 21.81 | 83.62 | 7.42 | 3.87 | 8.84 | 6.63 | 15.05 | 42.55 | 5.31 | 6.83 | 10.71 | 18.45 |
| **A vs. U** | 9.53 | 8.18 | 45.38 | 2.51 | 2.57 | 3.96 | 3.73 | 3.22 | 5.07 | 10.07 | 6.64 | 7.19 | 9.00 |
| **A vs. H** | 3.13 | 14.54 | 55.99 | 5.54 | 4.90 | 5.61 | 6.17 | 16.80 | 31.63 | 7.69 | 9.02 | 9.37 | 14.20 |

#### Emissions

Results on emissions of CO2, and other criteria pollutants, including CO, HC and NOx, for different stages at different entry speeds are illustrated in Tables 6-9, respectively. For CO2, CO, hydrocarbon (HC) and NOx emissions (Tables 6-9), results follow similar trends as shown in the fuel consumption (see Table 2). Trend in the results for HC emissions (Table 8) is much less conclusive, which might result from much more complicated chemical processes and higher sensitivity on operating conditions of the internal combustion engine (ICE) and catalytic converter.

Table 6. Median of Normalized CO2 Emissions (across All Drivers) for Each Test Cell (gram/mile)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Speed (mph)** | **Phase** | **Green** | | | | | | **Red** | | | | | | **On Avg.** |
| **TIP** | **2** | **7** | **12** | **17** | **22** | **27** | **2** | **7** | **12** | **17** | **22** | **27** |
| **20** | **U** | 184 | 193 | 454 | 430 | 402 | 400 | 401 | 354 | 184 | 179 | 191 | 194 | 297 |
| **H** | 202 | 352 | 435 | 406 | 369 | 387 | 389 | 256 | 195 | 181 | 193 | 191 | 296 |
| **A** | 173 | 178 | 293 | 342 | 322 | 294 | 256 | 190 | 180 | 164 | 163 | 170 | 227 |
| **25** | **U** | 197 | 175 | 506 | 498 | 487 | 463 | 447 | 432 | 339 | 180 | 191 | 189 | 342 |
| **H** | 173 | 186 | 458 | 470 | 434 | 430 | 441 | 415 | 226 | 175 | 171 | 175 | 313 |
| **A** | 153 | 159 | 308 | 430 | 426 | 419 | 413 | 309 | 239 | 156 | 156 | 148 | 276 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Scenario 1 |  | Scenario 2 |  | Scenario 3 |  | Scenario 4 |

Table 7. Median of Normalized CO Emissions (across All Drivers) for Each Test Cell (10-3 gram/mile)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Speed (mph)** | **Phase** | **Green** | | | | | | **Red** | | | | | | **On Avg.** |
| **TIP** | **2** | **7** | **12** | **17** | **22** | **27** | **2** | **7** | **12** | **17** | **22** | **27** |
| **20** | **U** | 34 | 43 | 255 | 299 | 263 | 242 | 243 | 199 | 40 | 31 | 37 | 41 | 144 |
| **H** | 44 | 185 | 254 | 230 | 216 | 226 | 248 | 93 | 40 | 30 | 38 | 38 | 137 |
| **A** | 25 | 30 | 175 | 145 | 133 | 120 | 105 | 29 | 27 | 23 | 22 | 24 | 71 |
| **25** | **U** | 55 | 39 | 375 | 414 | 450 | 328 | 373 | 354 | 245 | 41 | 52 | 54 | 232 |
| **H** | 40 | 47 | 311 | 451 | 356 | 338 | 337 | 389 | 101 | 38 | 40 | 36 | 207 |
| **A** | 24 | 23 | 205 | 276 | 259 | 258 | 263 | 177 | 118 | 25 | 25 | 23 | 140 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Scenario 1 |  | Scenario 2 |  | Scenario 3 |  | Scenario 4 |

Table 8. Median of Normalized HC Emissions (across All Drivers) for Each Test Cell (10-3 gram/mile)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Speed (mph)** | **Phase** | **Green** | | | | | | **Red** | | | | | | **On Avg.** |
| **TIP** | **2** | **7** | **12** | **17** | **22** | **27** | **2** | **7** | **12** | **17** | **22** | **27** |
| **20** | **U** | 1.55 | 1.55 | 3.15 | 2.90 | 2.75 | 2.15 | 1.90 | 1.55 | 1.45 | 1.50 | 1.55 | 1.55 | 1.96 |
| **H** | 1.55 | 2.40 | 3.25 | 2.75 | 2.55 | 2.05 | 1.80 | 1.55 | 1.50 | 1.65 | 1.55 | 1.60 | 2.02 |
| **A** | 1.80 | 1.60 | 0.80 | 3.50 | 3.20 | 2.70 | 2.20 | 2.20 | 1.90 | 1.80 | 1.70 | 1.90 | 2.11 |
| **25** | **U** | 1.20 | 1.05 | 3.15 | 2.70 | 2.50 | 2.10 | 1.70 | 1.45 | 1.25 | 1.20 | 1.15 | 1.00 | 1.70 |
| **H** | 1.10 | 1.15 | 3.05 | 2.90 | 2.45 | 2.00 | 1.80 | 1.30 | 1.20 | 1.15 | 0.90 | 1.15 | 1.68 |
| **A** | 1.50 | 1.30 | 1.00 | 2.90 | 2.70 | 2.30 | 2.00 | 1.70 | 1.30 | 1.00 | 1.50 | 1.30 | 1.71 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Scenario 1 |  | Scenario 2 |  | Scenario 3 |  | Scenario 4 |

Table 9. Median of Normalized NOx Emissions (across All Drivers) for Each Test Cell (10-3 gram/mile)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Speed (mph)** | **Phase** | **Green** | | | | | | **Red** | | | | | | **On Avg.** |
| **TIP** | **2** | **7** | **12** | **17** | **22** | **27** | **2** | **7** | **12** | **17** | **22** | **27** |
| **20** | **U** | 3.5 | 5.0 | 14.3 | 15.4 | 13.3 | 12.9 | 14.7 | 10.9 | 4.3 | 4.3 | 4.1 | 4.6 | 8.9 |
| **H** | 4.9 | 11.5 | 13.9 | 13.3 | 11.4 | 12.8 | 13.5 | 7.1 | 4.5 | 3.6 | 4.5 | 4.5 | 8.8 |
| **A** | 2.8 | 3.6 | 9.7 | 10.1 | 7.7 | 7.8 | 6.7 | 3.0 | 2.9 | 3.1 | 3.2 | 2.8 | 5.3 |
| **25** | **U** | 5.3 | 4.5 | 20.4 | 20.1 | 23.7 | 18.1 | 19.2 | 17.9 | 12.8 | 4.9 | 5.2 | 5.0 | 13.1 |
| **H** | 4.1 | 5.1 | 15.9 | 20.8 | 17.9 | 17.9 | 17.0 | 20.2 | 7.5 | 4.0 | 4.3 | 4.3 | 11.6 |
| **A** | 2.9 | 2.5 | 12.4 | 15.6 | 15.3 | 14.9 | 16.4 | 11.1 | 7.6 | 3.7 | 3.3 | 2.7 | 9.0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Scenario 1 |  | Scenario 2 |  | Scenario 3 |  | Scenario 4 |

#### Mobility

In addition to the fuel consumption and pollutant emissions, mobility performance (in terms of trip time) is also compared across different stages at different entry speeds. As shown in Table 10, at different entry speeds, the average trip time of “automated” driving (the “median” case) is slightly less than that of “manual-uninformed” driving (the “median” case), while the “manual-HMI-assisted” driving (the “median” case) performs the worst, i.e., its average trip time is a bit longer than that of “manual-uninformed” driving. Most of the mobility benefits of the “automated” driving result from the cases where “speed-up and pass” scenarios occur, compared to the “full-stop” scenarios of “manual” driving.

Table 10. Median of Trip Time (across All Drivers) for Each Test Cell (second)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Speed (mph)** | **Phase** | **Green** | | | | | | **Red** | | | | | | **On Avg.** |
| **TIP** | **2** | **7** | **12** | **17** | **22** | **27** | **2** | **7** | **12** | **17** | **22** | **27** |
| **20** | **U** | 32.5 | 32.5 | 62.5 | 57.0 | 52.0 | 47.5 | 42.0 | 37.0 | 33.0 | 32.5 | 33.0 | 33.5 | 41.3 |
| **H** | 34.5 | 49.0 | 62.0 | 58.0 | 52.5 | 47.5 | 43.0 | 35.5 | 34.5 | 34.5 | 33.5 | 34.5 | 43.3 |
| **A** | 35.0 | 33.0 | 27.0 | 60.0 | 57.0 | 50.0 | 46.0 | 39.0 | 35.0 | 35.0 | 35.0 | 36.0 | 40.7 |
| **25** | **U** | 27.0 | 27.5 | 59.5 | 56.0 | 50.0 | 46.5 | 41.0 | 36.0 | 30.0 | 27.5 | 27.0 | 27.0 | 37.9 |
| **H** | 27.0 | 27.5 | 60.5 | 56.0 | 51.0 | 46.5 | 42.0 | 36.5 | 28.5 | 28.0 | 28.0 | 27.0 | 38.2 |
| **A** | 28.0 | 28.0 | 22.0 | 59.0 | 56.0 | 50.0 | 46.0 | 38.0 | 33.0 | 28.0 | 28.0 | 28.0 | 37.0 |

### Scenario-Based Comparison

As previously mentioned, the most drastic changes in performance measures occur at boundary cells between different scenarios. In addition, the majority of benefits (in terms of environment and mobility) of the “automated” driving stage lie in those scenarios that are different from “manual” driving stages. To make such difference stand out, scenario-based comparison is conducted in this section.

Tables 11 and 12 summarize the results for fuel consumption and trip time on a scenario basis (column-wise combination), where results of cells in Table 1 and Table 8 experiencing the associated scenarios are aggregated and the relative change (%) are then calculated. As can be observed from the tables, for the same set of time interval in the cycle, when “automated” driving (the “median” case) is experiencing Scenario 2 while “manual” driving (the “median” case) is experiencing Scenario 3, the improvements in fuel economy and trip time can be as high as 40% and 64%, respectively. If “automated” driving is experiencing Scenario 4 while “manual” driving is experiencing Scenario 3, then reduction in fuel consumption may range from 9% to 29% (depending on both stage and entry speed) with compromise of mobility. The smoother acceleration profile for departure contributes to the increase in trip time for “automated” driving stage.

Table 11. Relative Improvement (%) between Stages with Respect to Median of Normalized Fuel Consumption (Scenario-Based)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Speed (mph)** | **Stage** | **Scenario** | | | | | | |
| **1 vs. 1** | **2 vs. 1** | **2 vs. 3** | **3 vs. 1** | **3 vs. 3** | **4 vs. 3** | **4 vs. 4** |
| **20** | **H vs. U** | -3.3 | / | / | -82.4 | 4.8 | 27.6 | / |
| **A vs. U** | 8.8 | 7.9 | 35.5 | / | / | 29.3 | / |
| **A vs. H** | 11.7 | / | 40.2 | / | / | 21.7 | 26.0 |
| **25** | **H vs. U** | 5.9 | / | / | / | 5.9 | / | 34.7 |
| **A vs. U** | 17.5 | / | 39.1 | / | / | 13.4 | 31.0 |
| **A vs. H** | 12.3 | / | 32.8 | / | / | 8.7 | -5.6 |

Table 12. Relative Improvement (%) between Stages with Respect to Median of Trip Time (Scenario-Based)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Speed (mph)** | **Stage** | **Scenario** | | | | | | |
| **1 vs. 1** | **2 vs. 1** | **2 vs. 3** | **3 vs. 1** | **3 vs. 3** | **4 vs. 3** | **4 vs. 4** |
| **20** | **H vs. U** | -4.3 | / | / | -50.8 | -0.8 | 4.1 | / |
| **A vs. U** | -7.0 | -1.5 | 56.8 | / | / | -7.0 | / |
| **A vs. H** | -2.6 | / | 45.9 | / | / | -6.0 | -9.9 |
| **25** | **H vs. U** | -1.1 | / | / | / | -1.2 | / | 5.0 |
| **A vs. U** | -2.9 | / | 63.0 | / | / | -8.5 | -10.0 |
| **A vs. H** | -1.8 | / | 63.6 | / | / | -7.3 | -15.8 |

# Conclusion and Next Steps

In this project the GlidePath prototype system has been developed and its performance has been validated through extensive field experiments. By integrating connected vehicle technology with vehicle automation, the GlidePath prototype system has shown great potential in reducing the vehicle’s fuel consumption and emissions when traveling through the signalized intersection, compared to manual driving (either “uninformed” or “HMI-assisted”). The results show that such fuel savings may range from 11% to 21% on average, depending on the entry speed and availability of driving assistance. In addition, the GlidePath prototype system can offer the most efficient but least burdensome maneuver for urban driving at signalized intersections.

The comparative study also shows that the performance of “manual-HMI-assisted” driving is not as good as anticipated (especially at the entry speed of 20 mph), although it shares the same vehicle trajectory planning algorithm with the GlidePath prototype system. This may suggest that the information disseminated via HMI may not be suitable for human drivers (e.g., too much information in that it could be distracting). Regardless of how well the information is displayed to the driver, the speed recommendations tended to change abruptly, and it is impractical to expect a human driver to exercise rapid, fine control over the vehicle’s speed to keep up with the recommendations.

Further improvement on the vehicle trajectory planning algorithm may include:

* to accommodate the capabilities of vehicle controller(s) and inputs from other on-board sensors;
* to handle the eco-approach and departure at intersections controlled by actuated signals and with traffic queues at the intersection;
* to adapt to real-world traffic conditions, roadway topology and weather conditions;
* to extend for multiple signalized intersections or a signalized corridor;
* to cooperate with other equipped vehicles to achieve better operational efficiency, including in a Cooperative Adaptive Cruise Control (CACC) platoon.

# Lessons Learned and Recommendations

This section provides some lessons learned and recommendations for better executing similar task orders in the future.

## Speed Control and EAD Algorithm

* The trigonometric algorithm felt unnatural in a slow-down maneuver, and may be disruptive to following traffic, or cause the driver to distrust it and disable it. The team should consider making the slow-down more gradual, like a human driver taking the foot off the gas pedal.
* As configured today, a gradual, no-throttle slow-down is probably not possible with just an algorithm modification, as the XGV controller tends to use the brake for small downward speed adjustments. The team may want to change our XGV engagement mode to allow direct control of the throttle and brake to avoid unnecessary braking action.
* The XGV controller is designed to prefer a step input function projected well into the future. The EAD algorithm provides gradually changing speeds at each time step, which seems to make the XGV less responsive. The team was trying to do with the EAD algorithm what is already built into the XGV, but forcing a different speed profile. The team may want to investigate how to bypass the XGV’s control loop so that the EAD algorithm can fully specify the shape of the trajectory.
* During acceptance testing it was noted that, during a speed-up scenario, the vehicle began returning to the operating speed immediately upon crossing the stop bar. This was seen as a safety problem, because in traffic, following drivers will probably not be prepared for the sudden deceleration. Therefore, the algorithm has been changed to carry the extra speed through the intersection, then to decelerate much more gradually in the departure lane.

## Vehicle Configuration

* A good deal of effort was spent porting the DVI web content from one version of Android (and Chrome) to another as the team switched display devices or ran into tablet browser issues. It would have been more productive to settle on a specific platform configuration and version before starting development of this part of the software.
* The required audible alerts were difficult to achieve due to issues with the Android operating system on the DVI tablet. The speed control software achieved the required functionality, but depended upon the user to touch the screen after the vehicle’s motion began (and before the audible alert was issued). Such a demand on the user probably could be avoided with some redesign of the user interface so that all presentations are made on a single web page, where the driver would be touching the screen anyway, in order to begin the experiment.

## Infrastructure and Communications

* Our performance data analysis revealed a common occurrence of communication DSRC drop-outs and suspect position data in many of the experimental runs. The team may want to consider ways to make both of these signal inputs more robust and reliable.
* Energy consumption results were calculated due to the hybrid vehicle’s lack of consistent gasoline engine use. The team may want to find a way to get ground truth energy consumption to better understand the energy saving benefits.
* There were several mornings prior to demos when the team had to troubleshoot the black box and RSU setup because communications weren’t working. A simple auto-start of black box software on reboot would help minimize these problems. The primary RSU at the east end of the course was connected to a temporary extension cord that was susceptible to temporary failure in wet weather; upon completion of roadway construction, this RSU will have a permanent power connection. Once or twice, the signal controller needed to be rebooted. Debugging these geographically dispersed components is time-consuming. So a network monitoring application may be worth considering.
* The EAD algorithm was based on its knowledge of the duration of the current signal phase as well as the next two phases. Only the current phase and next phase was broadcast in the SPaT message, so the phase durations were input to the EAD algorithm as software configuration parameters. Related to this, the default signal controller behavior included an “all red” phase that ensured all four approach lanes saw a red signal simultaneously before a green signal was issued. Since this counted as a distinct phase in the SPaT messages, it further deteriorated the ability to understand the signal’s future sequencing. The signal controller was adjusted to eliminate the all red phase for this project. Future work should involve making an algorithm that is robust enough to deal with the more limited amount of data in the SPaT message alone, as it won’t have any other way to know about the signal it is approaching.

# Appendix A – References

M. Barth, F. An, T. Younglove, G. Scora, C. Levine, M. Ross, and T. Wenzel (2000), “The Development of a Comprehensive Modal Emissions Model”, NCHRP Web-Only Document 122, Contractor’s final report for NCHRP Project 25-11, National Cooperative Highway Research Program, April, 307 p.

M. Barth, G. Scora, and T. Younglove (2004), “Modal Emissions Model for Heavy-Duty Diesel Vehicles”, Transportation Research Record, Vol. 1880, Transportation Research Board, Washington, DC, pp. 10-20.

M. Barth, N. Davis, G. Scora, J. Collins, and J. Norbeck (2006), “Measuring and Modeling Emissions from Extremely Low Emitting Vehicles”, Transportation Research Record, Vol. 1987, Transportation Research Board, Washington, DC, pp. 21-31.

M. Barth, and K. Boriboonsomsin (2008), “Real-world carbon dioxide impacts of traffic congestion,” Transportation Research Record, Vol. 2058, pp. 163-171

P. Hao, G. Wu, K. Boriboonsomsin, and M. Barth (2015), “Developing a Framework for Eco-Approach and Departure Application for Actuated Signal Control”, IEEE Intelligent Vehicles Symposium, pp. 796 – 801

M. Barth, S. Mandava, K. Boriboonsomsin, and H. Xia (2011), “Dynamic ECO-driving for arterial corridors,” 2011 IEEE Forum on Integrated and Sustainable Transportation System (FISTS), pp.182-188

K. S. Yi and J. T. Chung (2001), “Nonlinear Brake Control for Vehicle CW/CA Systems,” IEEE/ASME Transactions on Mechatronics, Vol. 6, No. 1, pp. 17-25

G. Wu, K. Boriboonsomsin, H. Xia, and M. Barth (2014), “Supplementary Benefits from Partial Vehicle Automation in an Eco-Approach and Departure Application at Signalized Intersections,” Transportation Research Record, No. 2424, pp. 66 – 75

# Appendix B – Acronyms

AERIS Applications for the Environment: Real-time Information Synthesis

CACC Cooperative Adaptive Cruise Control

CMEM Comprehensive Modal Emissions Model; used by UCR to analyze fuel economy

DSRC Dedicated Short-Range Communications

DVI Driver-Vehicle Interface; the dash-mounted tablet used for interaction with the automation software

EAD Eco-Approach and Departure

FHWA Federal Highway Administration

GID Geometric Intersection Design

GPS Global Positioning System

HC Hydrocarbon

HMI Human-Machine Interface; the initial device that gave speed recommendations to a human driver

HRDO FHWA’s Office of Operations, Research and Development

ICE Internal Combustion Engine

ICM Integrated Corridor Management

ITS Intelligent Transportation Systems

OBU Onboard Unit; a DSRC radio device onboard a vehicle

RSU Roadside Unit; a DSRC radio device mounted near the intersection

SPaT Signal Phase and Timing

TFHRC Turner Fairbank Highway Research Center

UCR University of California at Riverside

V2V Vehicle-to-vehicle communications

V2I Vehicle-to-infrastructure communications (generally considered bi-directional)

VTPA Vehicle Trajectory Planning Algorithm

1. National Highway Transportation Safety Administration (NHTSA) defines five levels of vehicle automation: No automation (level 0), function-specific automation (level 1), combined function automation (level 2), limited self-driving automation (level 3), full self-driving automation (level 4). [↑](#footnote-ref-1)
2. SPAT and MAP message formats for this project are defined by the FHWA Office of Operations Research and Development Interface Control Document for the Signal Phase and Timing and Related Messages for V-I Applications, dated June 2013. This format was chosen due to its recent release and local familiarity among the project team. [↑](#footnote-ref-2)
3. Early in the integration testing the algorithm was allowed to continually update the trajectory shape throughout the route. Because it did not account for vehicle response lag it frequently recomputed a new trajectory, starting the trigonometric shape over at the vehicle’s current location. Since the early portion of these circular arcs in the speed profile change very slowly, the speed commands never departed significantly from the actual speed, so a slow-down or a speed-up maneuver was usually inadequate. Occasionally, the scenario identifier would even change the type of trajectory as a result, which would result in an overly jerky ride. Hence, the team decided to force the trajectory to be computed only once, at the beginning of the route. In this way speed errors would quickly become large enough to be driven out in time to meet the trajectory objectives. [↑](#footnote-ref-3)
4. http://www.cert.ucr.edu/cmem/ [↑](#footnote-ref-4)