# Cloud-Based Velocity Profile Optimization for Everyday Driving: A Dynamic-Programming-Based Solution

Engin Ozatay, Simona Onori, *Member, IEEE*, James Wollaeger, *Member, IEEE*, Umit Ozguner, Giorgio Rizzoni, Dimitar Filev, *Fellow, IEEE*, John Michelini, and Stefano Di Cairano, *Member, IEEE*

***Abstract*—Driving style, road geometry, and traffic conditions have a significant impact on vehicles' fuel economy. In general, drivers are not aware of the optimal velocity profile for a given route. Indeed, the global optimal velocity trajectory depends on many factors, and its calculation requires intensive computations. In this paper, we discuss the optimization of the speed trajectory to minimize fuel consumption and communicate it to the driver. With this information the driver can adjust his/her speed profile to reduce the overall fuel consumption. We propose to perform the computation-intensive calculations on a distinct computing platform called the "cloud." In our approach, the driver sends the information of the intended travel destination to the cloud. In the cloud, the server generates a route, collects the associated traffic and geographical information, and solves the optimization problem by a spatial domain dynamic programming (DP) algorithm that utilizes accurate vehicle and fuel consumption models to determine the optimal speed trajectory along the route. Then, the server sends the speed trajectory to the vehicle where it is communicated to the driver. We tested the approach on a prototype vehicle equipped with a visual interface mounted on the dash of a test vehicle. The test results show 5%–15% improvement in fuel economy depending on the driver and route without a significant effect on the travel time. Although this paper implements the speed advisory system in a conventional vehicle, the solution is generic, and it is applicable to any kind of powertrain structure.**

***Index Terms*—Cloud computing, dynamic programming (DP), fuel economy, intelligent transportation systems (ITS), optimal control.**

Manuscript received August 15, 2013; revised December 10, 2013 and February 13, 2014; accepted April 10, 2014. Date of publication May 19, 2014; date of current version December 1, 2014. This work was supported in part by Ford Motor Company, Dearborn, MI, USA, under the University Research Project Program and in part by the National Science Foundation under the Cyber-Physical Systems Program (ECCS-0931669). The Associate Editor for this paper was W. Fan.

E. Ozatay, U. Ozguner, and G. Rizzoni are with the Center for Automotive Research, The Ohio State University, Columbus, OH 43202 USA (e-mail: ozatay.1@osu.edu; umit@ece.osu.edu; rizzoni.1@osu.edu).

S. Onori was with The Ohio State University, Columbus, OH 43210 USA. She is now with the Department of Automotive Engineering, Clemson University, Greenville, SC 29607 USA (e-mail: sonori@clemson.edu).

J. Wollaeger is with Robert Bosch LLC, Plymouth, MI 48170 USA (e-mail: james.Wollaeger@us.bosch.com).

D. Filev and J. Michelini are with the Powertrain Control Research and Advanced Engineering, Ford Motor Company, Dearborn, MI 48124 USA (e-mail: dfilev@ford.com; jmichel1@ford.com).

S. Di Cairano was with the Powertrain Control Research and Advanced Engineering, Ford Motor Company, Dearborn, MI 48124 USA. He is now with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA (e-mail: dicairano@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2014.2319812

## I. Introduction

As the number of vehicles on the road has increased worldwide, the importance of decreasing the overall vehicle energy consumption has grown. Increased environmental pollution and the limited petroleum supply, still the main source of energy in today's vehicles, compels society, academia, and industry to seek more efficient vehicles. Significant effort has been put forth in finding new powertrain with less energy consumption. This work resulted in breakthroughs allowing modern hybrid vehicles. Although hybrid vehicles take many forms (e.g., pneumatic, mechanical, and fuel cell), hybrid electric vehicles (HEVs) drew the most attention to date, and many studies focused on energy management of HEVs [1]–[3].

Another approach to reduce energy consumption is in the area of driving velocity profile optimization. However, the traffic and geographical information of the road networks require large storage units, and the search algorithms for global optimization may require high computation power, which is not available on current vehicle computing units [4]–[6]. As technology develops, however, cheaper and better communication systems emerge, more accurate sensors become available, and in-vehicle computation units become more powerful. Recently in Europe, some of the local public transportation vehicles have started to communicate with a certain number of traffic lights [7]. In the USA, industry and academia are conducting experiments in broadcasting red light timings for security warning systems [8]. These advances in communication systems, sensor technology, and high performance computation sources enable further work in driving profile optimization, an approach which still holds a great potential for energy consumption reduction of road vehicles [9], [10] at a very limited cost.

Recently, many algorithms have been proposed for speed trajectory optimization. Asadi and Vahidi [11] proposed a control algorithm that adapts the velocity profile to guarantee that a vehicle approaches a traffic light at green, whenever possible. The authors used a short-range radar and traffic signal information to predictively schedule a suboptimal velocity trajectory and implemented the algorithm in an existing cruise control system. A similar approach was proposed by Raubitschek *et al.* [12], where the authors divided the velocity profile into a number of modes and generated a velocity profile combined with these modes to ensure arrival at a green traffic light. In [13], we developed an analytical solution to generate an optimal velocity profile to minimize energy consumption on a given route with the

existence of a single traffic light. In our analysis, we assumed the availability of real-time traffic light information. Similarly in [14], a closed-form solution is proposed for the generation of optimal energy management in electric vehicles for a given route. In [15] and [16], the authors proposed a velocity profile optimizing algorithm for a certain look-ahead distance; however, their algorithm may lead to suboptimal solutions for the entire trip distance. In [17]–[19], algorithms based on traffic and topographic information of the road for energy consumption reduction have been proposed. The studies discussed so far, in general, require relatively expensive onboard computation resources and sensors, and their real-time applications have been limited. Another application of energy-efficient velocity optimization is conducted by Howlett *et al.* [20], [21], where the authors deal with the speed control problem within the context of train operation, although there is much less need for cloud computation because the route is repetitive and there are far fewer disturbances. In addition, the time scheduling in trains is the dominant factor because of the shared rail resource.

In our preliminary work [22], we have introduced the cloud framework, and in this paper, we extend the developed ideas with real-time implementation of the speed advisory system (SAS) to generate a global optimal velocity profile by incorporating the available geographical and traffic information, propose a solution by means of cloud computing [23], present more details on the algorithms, and their implementation, and present experimental results on multiple routes and for multiple drivers.

### A. Cloud Computing for Vehicle Applications

Cloud computing, as defined in [24], is a system for enabling on-demand network access to a shared pool of configurable computing resources that have "virtually unlimited" storage space and computational power. Resources can be rapidly acquired and released with minimum management effort. The recent penetration of the mobile wireless Internet access renders cloud computing possible for in-vehicle applications. Currently, cloud computing has a limited number of automotive applications and preliminary analysis [25], [26]. Some examples are the Ford's *MyFordMobile* application [27], which uses an onboard wireless Internet connection module to communicate with cloud computing services for infotainment and telematics features and the Progressive Insurance Company's MyRate driving monitoring device. The MyRate is the first automotive application monitoring the driving profile.

In this paper, we extend the utilization of cloud computing in automotive applications by providing a driving assistance system. The system aims at advising the driver of an optimal velocity profile to reduce the overall fuel consumption. For this purpose, we established a two-way communication system between the vehicle and the cloud, as shown in Fig. 1. The vehicle sends the intended trip information to the cloud. The associated traffic and geographical information is retrieved, and a route is generated via cloud computing. A dynamic programming (DP) algorithm is executed to calculate the optimal velocity trajectory and sent back to the vehicle. Then, the optimal speed is advised to the driver in real time by a visual interface.

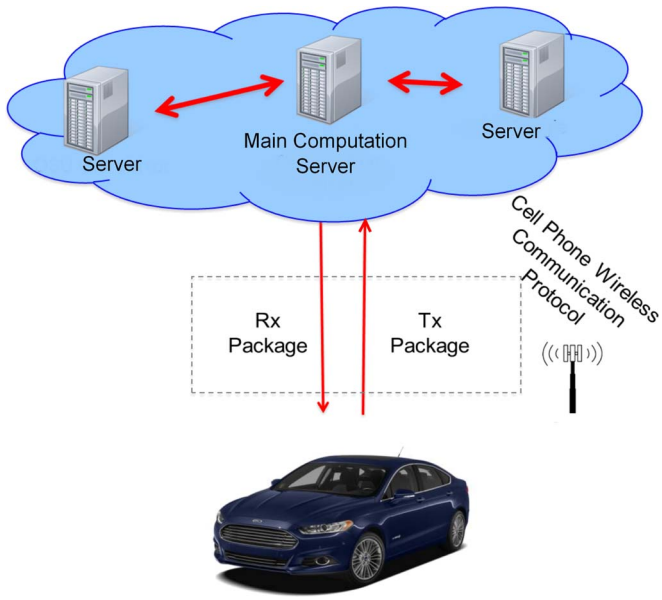



Fig. 1. Cloud–vehicle interaction schematic.

This paper is structured as follows. In Section II, the vehicle dynamics and fuel consumption models are described and the vehicle backward simulator is developed. In Section III, we formulate the spatial domain optimal control problem and solve it using a DP algorithm, as presented in Section IV. The test setup is explained in Section V, followed by the description of the test procedure and the test results in Section VI. In Section VII, we discuss the advisory system requirements in term of communication bandwidth, computation power, and memory size. Finally, in Section VIII, we conclude this paper by summarizing the overall system and presenting possible future developments.

## II. Vehicle Modeling

Here, we introduce the vehicle longitudinal and fuel consumption models and their use in a vehicle simulator. The simulator operates backward from the vehicle speed trajectory, through the powertrain, to determine the fuel consumption. Although the simulator uses quasi-steady-state equations, the studies in [29] and the other references there have shown that the backward simulator predicts the fuel consumption accurately and outperforms the forward simulator in terms of computation time. This work utilizes the backward simulator in the optimization because of the calculation time advantage.

### A. Vehicle Dynamics

We have developed a general backward simulation model that is used for fuel consumption optimization, and we have used as parameters the values for the prototype vehicle used for testing (Lincoln MKS) that are reported in Table I.

The longitudinal dynamics of the vehicle is given by

$$m_{\mathrm{eq}} \cdot \frac{dv}{dt} = F_{\mathrm{trac}} - F_{\mathrm{roll}} - F_{\mathrm{aero}} - F_{\mathrm{grade}} - F_{\mathrm{brake}} \qquad (1)$$

where $v$ is the vehicle speed, and $m_{\mathrm{eq}}$ is the equivalent mass of the vehicle, which is the sum of the curb weight of the vehicle $m$ and the inertia of all the rotating parts. The traction force

TABLE I
SPECIFICATIONS FOR LINCOLN MKS USED FOR TESTING

| Vehicle | | Engine | |
|---|---|---|---|
| Specification | Value | Specification | Value |
| Mass[kg] | 1954 | # of Cyl.[-] | 6 |
| Frontal Area [$m^2$] | 2.77 | Size [L] | 3.7 |
| Drag Coeff. [-] | 0.29 | Max. Trq [Nm] | 360 (4250 rpm) |
| Tire Radius [m] | 0.363 | Max. Speed [rpm] | 6500 |

$F_{\text{trac}}$ is the force supplied by the engine and transmitted to the tires by means of mechanical connections, and its formulation is given by

$$F_{\text{trac}} = \frac{\eta \cdot \gamma \cdot \gamma_{\text{fd}}}{R_{\text{wh}}} \cdot T_e \tag{2}$$

where $\eta$ is the efficiency of the transmission unit, $\gamma$ is the gear ratio of the selected gear, $\gamma_{\text{fd}}$ is the gear ratio of the final drive, $R_{\text{wh}}$ is the radius of the tires, and $T_e$ is the engine torque. The rolling resistance $F_{\text{roll}}$ is the friction force acting on the tires and is given by

$$F_{\text{roll}} = m \cdot g \cdot \cos(\alpha) \cdot (r_0 + r_1 \cdot v) \tag{3}$$

where $g$ is the gravitational constant, $\alpha$ is the road grade, and $r_0$ and $r_1$ are constants specific to the selected tires and wheels and may vary depending on the pressure, temperature, and the condition of the tires. The aerodynamic resistance $F_{\text{aero}}$ is

$$F_{\text{aero}} = \frac{1}{2}\rho A_f C_d v^2 \tag{4}$$

where $\rho$ is the air density, $A_f$ is the frontal area, and $C_d$ is the drag coefficient of the vehicle. Due to the proportionality to $v^2$, $F_{\text{aero}}$ dominates the other resistive forces at high velocities. The road grade force $F_{\text{grade}}$ is defined as

$$F_{\text{grade}} = m \cdot g \cdot \sin(\alpha). \tag{5}$$

Finally, $F_{\text{brake}}$ is the brake force. By substituting (2)–(5) into (1), we obtain the vehicle longitudinal dynamics as

$$\frac{dv}{dt} = \frac{1}{m_{\text{eq}}}\left(\frac{\eta\gamma\gamma_{\text{fd}}}{R_{\text{wh}}}T_e - mg\cos(\alpha)(r_0 + r_1 v) - \frac{1}{2}\rho A_f C_d v^2 - mg\sin(\alpha) - F_{\text{brake}}\right). \tag{6}$$

### B. Fuel Consumption Model

Developing an accurate fuel consumption model is crucial for addressing energy consumption optimization problems. In the literature, a number of fuel consumption models have been developed [28], [29]. Models based on the Willans line approximation suffer from accuracy over the entire range of the engine speed and engine torque, whereas the empirical models are, in general, developed for a particular class of engines.

In this paper, we use a fuel consumption model consisting of a polynomial function up to the third order of engine torque, i.e.,

$$\dot{m}_{\text{fuel}} = C_3(\omega_e)\cdot T_e^3 + C_2(\omega_e)\cdot T_e^2 + C_1(\omega_e)\cdot T_e + C_0(\omega_e) \tag{7}$$

where $C_0$, $C_1$, $C_2$, $C_3$ are functions of the engine speed, determined experimentally at Ford Technical Center, Dearborn, MI, USA, and reported in Table II. Several experimental validations of the model were conducted, and due to space limitations, we report only one comparison plot of the measured and predicted fuel consumption amounts in Fig. 2. Despite having small regional differences, the two cumulative fuel consumption curves are consistent.

TABLE II
FUEL CONSUMPTION MODEL PARAMETER VARIATION

| Engine Speed$[rpm]$ | $C_0$ $\left[10^{-4}\ \frac{gr}{s}\right]$ | $C_1$ $\left[10^{-5}\ \frac{gr}{s.N.m}\right]$ | $C_2$ $\left[10^{-8}\ \frac{gr}{s.N^2.m^2}\right]$ | $C_3$ $\left[10^{-10}\ \frac{gr}{s.N^3.m^3}\right]$ |
|---|---|---|---|---|
| 1000 | 2.8 | 0.47 | 0.11 | 0.14 |
| 2000 | 5.5 | 1.11 | -0.33 | 0.15 |
| 3000 | 9.5 | 1.95 | -2.62 | 0.57 |
| 4000 | 14.2 | 2.65 | -4.15 | 0.95 |
| 5000 | 18.9 | 2.84 | -1.01 | 0.77 |
| 6000 | 24.5 | 3.68 | -2.02 | 1.53 |

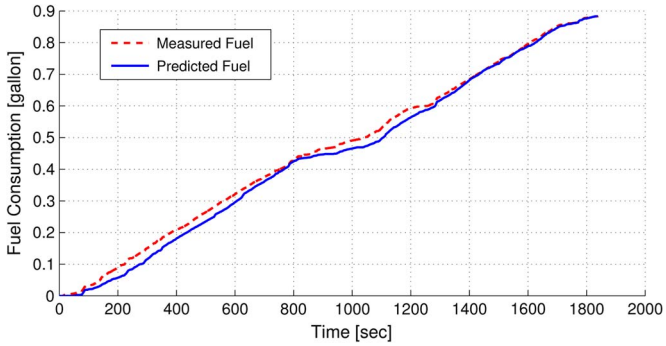



Fig. 2. Measured and model predicted cumulative fuel consumption curves for the same velocity profile from an actual experiment.

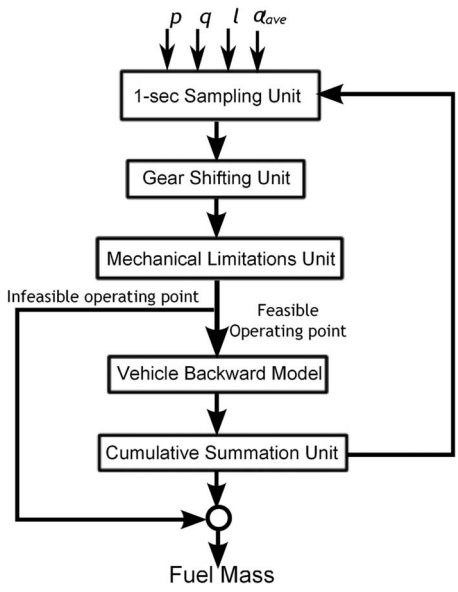



Fig. 3. Backward vehicle simulator flowchart.

### C. Vehicle Backward Simulator

A transition cost used in Section IV for DP calculations from an initial speed $p$ to a terminal speed $q$ for a given interval distance $l$ and the average road grade $\alpha_{\text{ave}}$ is executed by the vehicle backward simulator, as depicted in Fig. 3. In the 1-s

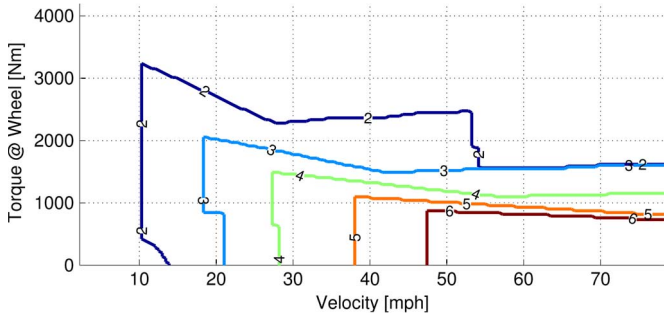



Fig. 4. Contour plot of the gear shifting strategy.

sampling block, first, we determine the travel time and the average acceleration of the road segment based on $p$, $q$, and $l$ by

$$t = \frac{2 \cdot l}{p+q} \qquad a_{\text{ave}} = \frac{q^2 - p^2}{2 \cdot l}. \tag{8}$$

Then, we determine the velocity at each 1-s sample time, i.e., we calculate $v_k$ for $k = 0, 1, \ldots, \lfloor t \rfloor$, where $v_k = p + k \cdot a_{\text{ave}}$ and $\lfloor \cdot \rfloor$ is the floor operator. We insert the 1-s interval average speed, i.e., $v_{\text{ave},k} = (v_k + v_{k+1})/2$ if $k \in \{0, 1, \ldots, \lfloor t \rfloor - 1\}$ and $v_{\text{ave},k} = (v_k + q)/2$ if $k = \lfloor t \rfloor$, $a_{\text{ave}}$, and $\alpha_{\text{ave}}$ into (6) to calculate the required torque at the wheels, i.e., $T_{w,k}$, at time $k$. Then we determine the gear number $\xi_k$ in the gear shifting block based on the gear shifting map shown in Fig. 4. In the figure, the contour plots indicate that an operating point between any two curves labeled by $j$ and $j+1$ is at the $j$th gear. An operating point outside of the most outer curve labeled by 2 is at the first gear. The engine speed at time $k$ is calculated by $\omega_{e,k} = (\gamma(\xi_k) \cdot \gamma_{\text{fd}}/R_{\text{wh}}) \cdot v_k$. The mechanical limitation block consisting of the vehicle's torque converter model checks the feasibility of the operating point in terms of the limits of the engine speed and torque at the selected gear. The block assigns an infinite cost if the operating point is infeasible and terminates any further calculations. On the other hand, for a feasible operating point, it sends the results to the vehicle backward model block.

The vehicle backward model block receives the feasible $\omega_{e,k}$ and $T_{e,k}$ as inputs, inserts the values in (7) to obtain the instantaneous fuel consumption, and sends it to the cumulative summation block that calculates the cumulative fuel consumption. We repeat the procedure until we process $\forall k \in \{0, 1, \ldots, \lfloor t \rfloor\}$ and obtain the total fuel consumption of the road segment, i.e., $m_{\text{fuel}}$, as the final output of the backward simulator.

## III. Optimal Control Problem Formulation

The objective of the optimal control problem is to find the optimal velocity profile that minimizes the fuel consumption over a travel distance $D_f$. For this purpose, the optimization is conducted in the spatial domain by means of the following transformation (which converts time-domain equations into the spatial domain) [16], [22]:

$$\dot{v} = \frac{dv}{dt} = \frac{dv}{dD} \cdot \frac{dD}{dt} = v \cdot \frac{dv}{dD} \tag{9}$$

where the traveled distance $D$ is the independent variable.

The cost function to be minimized is

$$J_D = \int_0^{D_f} \frac{\dot{m}_{\text{fuel}}\left(T_e(D), \omega_e(D)\right)}{v(D)} \cdot dD \tag{10}$$

where $D_f$ is the total travel distance, and the admissible control is $u(D) = T_e(D)$. The minimization of (10) is subject to the dynamic constraint

$$\frac{dv}{dD} = \frac{1}{m_{\text{eq}} \cdot v}\left(\frac{\eta\gamma}{R_{\text{wh}}} T_e - mg(r_0 + r_1 v) - \frac{1}{2}\rho A_f C_d v^2 - m \cdot g \cdot sin(\alpha) - F_{\text{brake}}\right) \tag{11}$$

obtained from (6)–(9).

In addition to the dynamic constraint (11), constraints are imposed on the control input and state during the optimization.

### A. Control Constraint Set

The control input $T_e$ is bounded by the maximum and minimum engine torques. The maximum engine torque of the target vehicle, i.e., the Lincoln MKS, is given in Table I, and the minimum engine torque is taken as zero; thus, the first constraint set is

$$\mathcal{U}_{D,1} := \{T_e(D), F_{\text{brake}}(D) : 0 \le T_e(D) \le T_e^{\max} \quad \text{and} \quad F_{\text{brake}}(D) = 0\}. \tag{12}$$

Similarly, a limitation is also enforced for the maximum braking force

$$\mathcal{U}_{D,2} := \{T_e(D), F_{\text{brake}}(D) : 0 \le F_{\text{brake}}(D) \le F_{\text{brake}}^{\max} \quad \text{and} \quad T_e(D) = 0\}. \tag{13}$$

Then, the control constraint set becomes $\mathcal{U}_D = \mathcal{U}_{D,1} \cup \mathcal{U}_{D,2}$.

### B. State Constraint Set

To guarantee vehicle operation within the legal speed limits, we define the constraint set

$$\mathcal{X}_{D,1} := \left\{v(D) : v_{\text{lim}}^{\min}(D) \le v(D) \le v_{\text{lim}}^{\max}(D)\right\}. \tag{14}$$

For driver comfort and safety issues, limitations are imposed on the vehicle acceleration

$$\mathcal{X}_{D,2} := \left\{v(D) : \left(\frac{dv}{dD}\right)^{\min} \le \frac{dv}{dD} \le \left(\frac{dv}{dD}\right)^{\max}\right\}. \tag{15}$$

Furthermore, the stop signs on the route impose a set of interior-point constraints defined by

$$\mathcal{X}_{D,3} := \{v(D) : v(D_s) = 0 \quad \text{for} \quad s = 1, 2, \ldots, m\}. \tag{16}$$

where $D_s$ is the location of the $s$th stop sign, and $m$ is the total number of the stop signs on the route. The state constraint set then becomes $\mathcal{X}_D = \mathcal{X}_{D,1} \cap \mathcal{X}_{D,2} \cap \mathcal{X}_{D,3}$.

In general, the traffic congestion further restricts $\mathcal{X}_{D,1}$ and $\mathcal{X}_{D,2}$ for real-time applications; however, in this application, we are not directly incorporating these effects into the state constraint sets. Instead, we accordingly update $\mathcal{X}_{D,1}$ (described in detail in Section VI-C), based on the driver's optimal velocity profile following characteristic such that the optimal speed trajectory tracking error is decremented.

### C. Boundary Conditions

In our calculations, we assume that a trip starts and ends at a standing position, i.e., the boundary conditions are

$$v(0) = v(D_f) = 0. \tag{17}$$

In the spatial domain framework, the optimization control problem aims at minimizing (10) by manipulating $u(D)$, subject to the constraints (11), $u(D) \in \mathcal{U}_D$ and $x(D) \in \mathcal{X}_D$ with boundary condition (17). Then for a given route $D_f$, $D_s$ (for $s = 1, 2, \ldots, m$), $\alpha(D)$, $v_{\lim}^{\min}(D)$, and $v_{\lim}^{\max}(D)$ are fixed parameters, and it is straightforward to determine $\mathcal{U}_D$ and $\mathcal{X}_D$. Although the boundary condition (17) seems to render the system state dynamic (11) undefined at the boundary points, this is not the case due to the inherent discrete nature of the DP solution used to solve the spatial domain optimization, i.e., in the DP solution, a speed transition from $v(k) = 0$ to $v(k+1) = 0$ is forbidden, where $v(k)$ and $v(k+1)$ is the speed at discrete times $k$ and $k+1$. Moreover, $v$ in (11) is taken as $v = (v(k) + v(k+1))/2$, and the condition $v = 0$ rendering the problem undefined never occurs.

## IV. DP Algorithm

This section details the solution of the nonlinear optimal control problem formulated in the spatial domain by the DP algorithm. A set of points identifies the route specific data and the route. Specifically, the route consists of the points $P = \{p_0, p_1, p_2, \ldots, p_K\}$. Each point $p_k \in P$ for $k = 0, 1, \ldots, K$ has its own characteristic parameters; $p_k = [\text{lat}_k, \text{lon}_k, d_k, h_k, \alpha_k, v_k^{\max}, v_k^{\min}]^T$, where $\text{lat}_k$ is the latitude, $\text{lon}_k$ is the longitude, $d_k$ is the distance, $h_k$ is the elevation, $\alpha_k$ is the grade, $v_k^{\max}$ is the maximum speed, and $v_k^{\min}$ is the minimum speed along the interval between $p_k$ and $p_{k+1}$. To define the DP algorithm, we require full information of the intended route. Some of these data are normally not available in the vehicle but are easily obtained in the cloud. In what follows, we describe the assumptions for $P$.

*A) Assumptions:*

1) $\text{lat}_k$ and $\text{lon}_k$ for $k = 0, 1, \ldots, K$ are known.
2) $d_k$ is known and $d_{k+1} < d_k$ for $k = 0, 1, \ldots, K-1$.
3) $h_k$ and $v_k^{\max}$ for $k = 0, 1, \ldots, K$ are known and linearly change between $p_k$ and $p_{k+1}$.

With the aforementioned assumptions, we calculate the other unknowns, namely, $\alpha_k$ and $v_k^{\min}$, for $k = 0, 1, \ldots, K$, as described in the next section. We associate the stop signs represented by the set $S = \{s_1, s_2, \ldots, s_m\}$ by the points $p_k$ with $v_k^{\max} = 0$.

The points in $P$ are not necessarily evenly spaced; however, DP requires regularity between the elements of $P$ for smooth transition of the optimal velocity profile. In the next section, we describe how we satisfy the regularity between the points.

### A. Manipulation of Set $P$

The manipulation of $P$ aims at creating a new set $P^* = \{p_0^*, p_1^*, p_2^*, \ldots, p_N^*\}$ such that $S \subset P^*$ and the elements of $P^*$ are dispersed with a regular pattern. The easiest way is to define a constant $d_c = d_k \; \forall k \in \{0, 1, \ldots, N-1\}$ and then insert the stop signs if they are not already in $P^*$. However, using a constant $d_c$ results in undesired behaviors on the optimal velocity profile, e.g., a relatively large equidistant value results in slow acceleration in the low-speed region. On the other hand, the selection of a small interval distance causes an unnecessary increase in the calculation time. As a compromise between the two situations, we define variable distance segments based on the regional maximum speed limit. We select the variable quantization interval as

$$\Delta D_k = \begin{cases} 50 \text{ m}, & \text{if } v_k^{\max} \leq 30 \text{ mi/h} \\ 150 \text{ m}, & \text{if } v_k^{\max} > 30 \text{ mi/h}. \end{cases} \tag{18}$$

Then, we generate $p_k^* \in P^*$ such that $d_k^* = \Delta D_k$ for $k = 0, 1, \ldots, N-1$. Moreover, we determine $h_k^*$ and $v_k^{*\max}$ by interpolating the corresponding values at $p_k \in P$. Finally, the stop sign and the boundary points of the route are inserted into $P^*$ by the monotonicity.

The grade between any two consecutive points is calculated by $\bar{\alpha}_k^* = tan^{-1}(h_{k+1}^* - h_k^*)/d_k$. However, to reduce the DP calculation time, we quantize the grade in intervals of 0.5°, i.e., $\alpha_k^* = \sum_{i=-2\alpha_{\max}}^{2\alpha_{\max}} r(i, \bar{\alpha}_k^*)$, where $\alpha_{\max} \in \mathbb{N}$ is the maximum absolute grade and

$$r\left(i, \bar{\alpha}_k^*\right) = \begin{cases} i/2, & \text{if } i/2 - 0.25 \leq \bar{\alpha}_k^* < i/2 + 0.25, \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

The optimal control problem is cast in such a way that the fuel consumed is the only cost criterion (10). However, for driver satisfaction, shorter travel time is also crucial. A reasonable speed bandwidth should be selected to satisfy shorter travel times. In this paper, we limit $v_k^{*\min}$ to be 10 mi/h less than $v_k^{*\max}$ to keep the travel time at an acceptable range.

The last parameter affecting the performance of DP is the quantization interval of the velocity, i.e., $\Delta v$, where too small step values unnecessarily increase the calculation time and too large values reduce performance. In this paper, we selected $\Delta v = 2$ mi/h previously determined [22] as a good compromise between the calculation time and the accuracy of the solution.

### B. DP Formulation

The DP algorithm [30], which proceeds backward in time from time step $N$ to 0, is defined as

$$J_k(v_k) = \min_{u_k \in \mathcal{U}_D} \left\{ g_k(v_k, u_k, \Delta D_k) + J_{k+1}\left(f(v_k, u_k)\right) \right\} \tag{20}$$

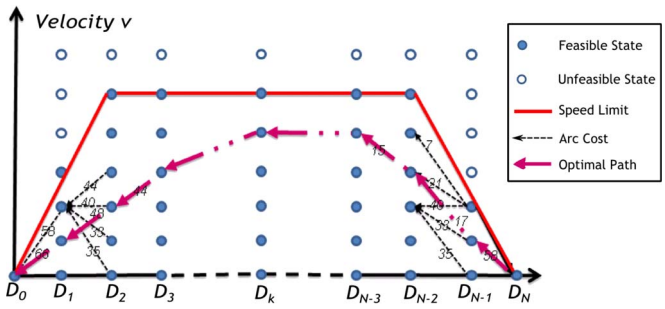



Fig. 5. Schematic representation of the DP algorithm.

where $J_k(v_k)$ is the cost-to-go function from step $k$ to $N$ starting from $v_k$ with terminal cost $J_N(v_N) = g_N(v_N)$. $u_k \in \mathcal{U}_D$ is the control input determining the velocity at the next step. $\mathcal{U}_D$ is the input constraint set defined in Section III. Since the boundary condition at the end of the travel is fixed by (17), the terminal cost function $g_N(v_N)$ is defined as

$$g_N(v_N) = \begin{cases} 0, & \text{if } v_N = 0 \\ \infty, & \text{if } v_N \neq 0. \end{cases} \tag{21}$$

Similarly, the transition cost function from step $k$ to $k+1$ for $k = 0, 1, \ldots, N-1$ is defined by

$$g_k(v_k, u_k, \Delta D_k) = \begin{cases} y_k, & \text{if } f(v_k, u_k) \in \mathcal{V}_{k+1}, \\ & \quad v_k \in \mathcal{V}_k, u_k \in \mathcal{U}_D, \\ \infty, & \text{otherwise} \end{cases} \tag{22}$$

where $y_k$ is the output of the vehicle backward simulator described in Section II-C, and $\mathcal{V}_k$ is the velocity bandwidth bounded by $v_k^{*\min}$ and $v_k^{*\max}$.

To retrieve the optimal path, i.e., the optimal velocity trajectory $V^* = \{v_0^*, v_1^*, v_2^*, \ldots, v_N^*\}$

$$\phi_k(v_k) = \underset{u_k \in \mathcal{U}_D}{\arg\min} \{g_k(v_k, u_k, \Delta D_k) + J_{k+1}(f(v_k, u_k))\} \tag{23}$$

for $k = 0, 1, \ldots N-1$. Then, the optimal control strategy $\mu^* = \{\mu_0^*, \mu_1^*, \ldots, \mu_N^*\}$ is obtained by the backtracking algorithm

$$\mu_k^* = \phi_k(v_k^*), \quad \text{where} \quad v_0^* = v(0) \tag{24}$$

$$v_{k+1}^* = f(v_k^*, \mu_k^*). \tag{25}$$

Fig. 5 shows a graphical representation of the DP algorithm, where the feasible and unfeasible points are represented with filled and unfilled circles, respectively. The DP algorithm starts from the last step $N$ and first calculates the transition cost to the points at the $(N-1)$th step. The cost of the transition to a feasible point (filled circle) is determined by the vehicle backward simulator, whereas the cost of the transition to an unfeasible point (unfilled circle) is infinity, as defined by (22). After we determine the transition costs from step $k+1$ to $k$ for $k = 0, 1, \ldots, N-1$ in the backward direction, we generate the optimal path by the backtracking algorithm.

The DP algorithm computes the entire feedback law $\mu_k = \phi_k(v_k)$. That is, if a disturbance occurs, the optimal control profile from the current time instant onward is adjusted to maintain future optimality.

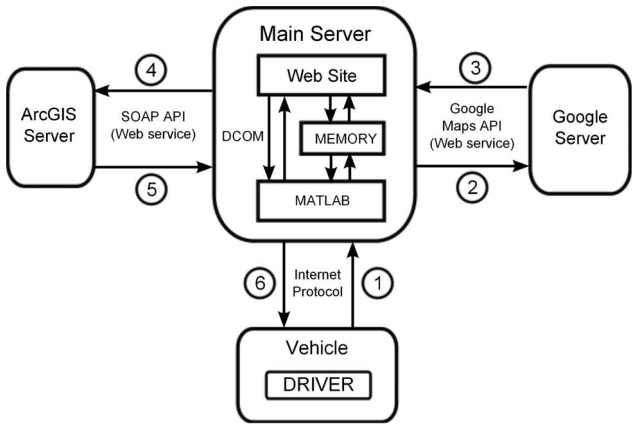



Fig. 6. Cloud communication sequence and protocol diagram.

## V. Description of the Test Setup

This section describes the cloud architecture and the setup developed in the vehicle.

### A. Cloud Architecture

Three servers comprise the cloud used in this project: the main server (optimization server), the ArcGIS Server, and the Google Maps Server. The main server resides at the Center for Automotive Research (CAR), The Ohio State University, and manages the communications between the servers and the vehicle. Fig. 6 presents a schematic of the communication sequence. The driver sends the origin, destination, and the waypoints of the desired route to the main server in the cloud through a webpage. The main server sends the desired trip information to the Google Maps Server using Google Maps Application Programming Interface, and the Google Maps Server generates the route in the form of polylines defined by the latitude $(\text{lat}_k)$ and longitude $(\text{lon}_k)$ of the edge points $(p_k, p_{k+1})$, which are utilized to calculate the distance $(d_k)$ between each point and then transferred to the ArcGIS Server through Single Object Access Protocol. The ArcGIS Server containing the digital elevation model of the states of Ohio and Michigan determines the elevation data $(h_k$ and sends it back to the main server. At the end of the aforementioned communication sequence, the main server gathers $h_k$, $\text{lat}_k$, $\text{lon}_k$, and $d_k$ information of $p_k$ for $k = 0, 1, \ldots, K$, which constitutes $P$ as described in Section IV. Employing the Distributed Component Object Model, the main server executes the optimization program that includes the DP algorithm, which is currently implemented in MATLAB. The result of the DP algorithm is the optimal velocity profile, and it is transferred back to the vehicle through the Internet connection. The data package that is sent contains $d_k^*$, $v_k^*$, $\text{lat}_k^*$, and $\text{lon}_k^*$ of $p_k^* \in P^*$ for $k = 0, 1, \ldots, N$, where $v_k^*$ is the optimal velocity at $p_k^*$.

An extension to our approach would be to determine a number of competing routes from the source point to the destination point, evaluate the energy consumption of each route, and pick the route having the least energy consumption amount. Furthermore, similar to the discussions for velocity profile update described in Section VI-C, we could also apply a reinforcement


Fig. 7. Lincoln MKS dashboard: the display screen and the GPS receiver installed for in-vehicle testing.

TABLE III
HIGHWAY TEST ROUTE INFORMATION

| | Address |
|---|---|
| Origin | 850 N Wilson Rd, Columbus, OH 43204, USA |
| Destination | 930 Kinnear Rd, Columbus, OH 43212, USA |
| Waypoint 1 | 6874 Dublin Center Dr, Dublin, OH 43017, USA |
| Waypoint 2 | 6688 Dublin Center Dr, Dublin, OH 43017, USA |

learning from the historical data and the most recent traffic and road information to update the route for minimum energy consumption along the trip. However, we leave the dynamic routing as a future work, and in this particular application of the cloud, without loss of generality, we only consider the fuel economy achievement based on velocity profile adaptation for a given route.

### B. Vehicle Instrumentation

The vehicle uses five main hardware components, namely, the display screen, GPS receiver, CAN Interface, 4G LTE-Capable USB Modem, and the vehicle laptop. Their functionalities are described next.

*1) Display Screen:* The screen displays the advised velocity to the driver. The background color of the display changes to warn the driver depending on the deviation amount of the vehicle speed with respect to the advised speed. A picture of the screen mounted on the dash of the test vehicle is shown in Fig. 7.

*2) GPS Receiver:* The GPS receiver and some other vehicle specific data are fused for vehicle localization.

*3) CAN Interface:* The advisory system requires the real time vehicle speed and the odometer information in order to update the advised velocity. To transfer the data between the electronic control unit (ECU) and the laptop, a parallel connection to the CAN data bus of the vehicle is established.

*4) LTE 4G USB Modem:* The communication from the driver to the cloud and vice versa is obtained through a mobile Internet connection. The test setup uses a 4G LTE USB Modem.

*5) Laptop:* The computing unit in the vehicle receives and logs the information received from the ECU, GPS, and the cloud server and runs the algorithm synchronizing the advised velocity with the vehicle position. It also runs the graphical user interface to show the advised velocity to the user and allows driver to interact with the cloud.

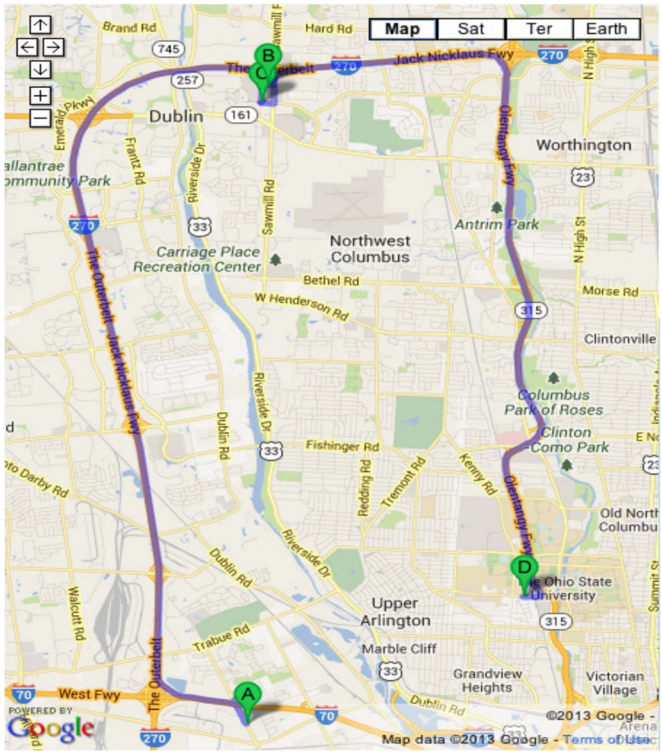



Fig. 8. Highway driving route.

## VI. TEST PROCEDURE AND RESULTS

The tests have been performed in a highway and an urban driving route. For both cases, we determine the origin, destination points, and waypoints and send a request to the cloud. In the cloud, the calculations are performed, and the optimal velocity profiles are generated and sent back to the vehicle. Then the driver drives along the generated route.

For each experiment, two test runs are performed. In the first run, the driver drives by his normal driving style without considering the advisory. Hereafter, the velocity profiles obtained from the first test are referred to as "natural driving" or "baseline driving," interchangeably. In the second run, the drivers follow the advised velocity profile. The second test run is referred to as "advisor following." In order to capture the average benefit obtained by the method, the same routes are tested by several drivers. In the following sections, we introduce the highway and the urban routes and present the test results.

### A. Highway Driving Test Results

The first set of experiments is conducted in highway driving. A route mainly consisting of highway and freeway segments is selected near The Ohio State University, Columbus, Ohio, USA. Only a small portion of the route is in urban area. The origin, destination and the waypoints information of the trip is presented in Table III. The route shown in Fig. 8 is generated by Google Maps Server. Based on the latitude and longitude values, the elevation information has been gathered from the GIS server. The elevation information is utilized to generate the road grade profile, as shown in Fig. 9. For the given route, the maximum speed limits and the location of stop signs

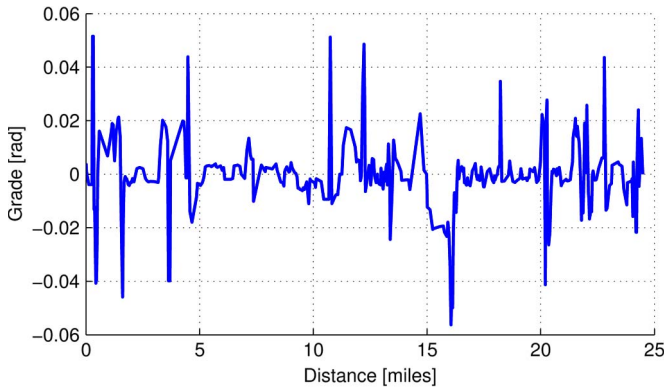



Fig. 9. Highway test route grade profile.

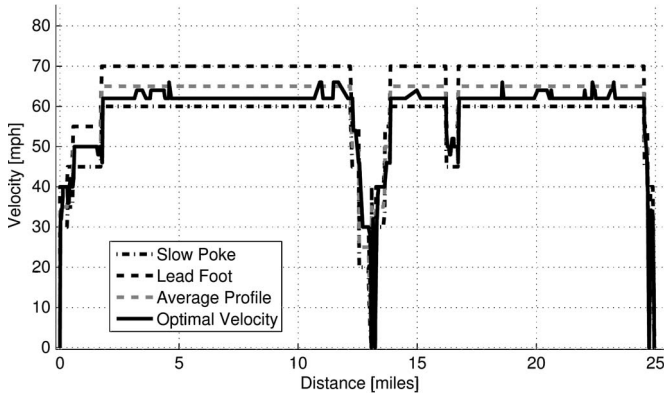



Fig. 10. Slow poke, foot lead, and average velocity profiles for the highway route.

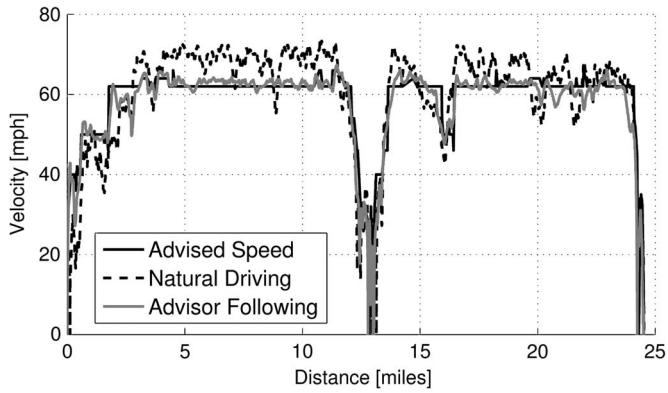



Fig. 11. Highway test route velocity profiles.

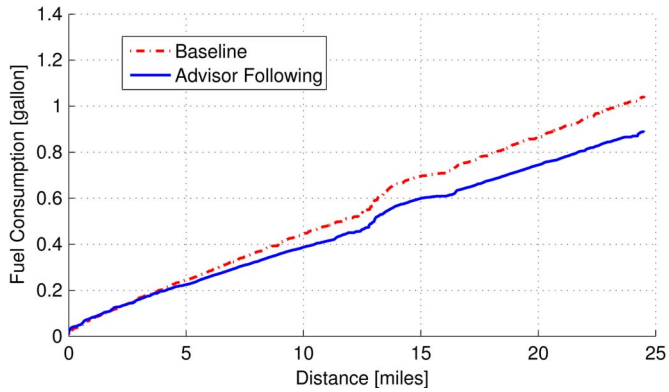



Fig. 12. Highway test route fuel consumption evolution for baseline and advisor following cases.

are determined, and the minimum speed limits are selected as 10 mi/h less than the maximum speed limits. Then, the optimization problem is solved, and the optimal velocity profile minimizing the total fuel consumption is calculated, as shown in Fig. 10. Apart from the optimal velocity trajectory, some other velocity trajectories, namely, slow poke, lead foot, and average speed profiles, are generated. The "slow poke" driving profile operates at the minimum velocity limit and reaches the destination after the longest time, whereas the "lead foot" driving profile corresponds to the legally permitted maximum speed profile and arrives at the destination point in the shortest time. The "average profile" is the average of the two previous scenarios. For the test route, the slow poke, lead foot, and average velocity trajectories are generated, as shown in Fig. 10. Three different drivers performed the tests. The velocity profiles from the test results of the first driver are presented in Fig. 11, indicating that in the baseline (natural) driving, the driver tends to drive faster than the optimal velocity trajectory suggested for the highway route. Indeed, the driver sometimes exceeds the speed limits. On the other hand, in the advisor following test, the driving pattern of the driver is smoother, and the average speed is lower. The travel time for the baseline is 30.6 min, whereas it is 31.1 min for the advisor following case.

Fig. 12 presents the cumulative fuel consumption of baseline and advisor following drivings of the first driver, and Table IV shows the trip time and the fuel economy of each test. The results show that the SAS improves the fuel economy, on average, by 12.6% with a 3.6% increase in travel time in the

TABLE IV
COMPARISON OF HIGHWAY TESTS

| Feature | Test 1 | Test 2 | Test 3 | Average |
|---|---|---|---|---|
| Baseline FE [mpg] | 23.7 | 23.7 | 23.7 | 23.7 |
| Adv. Fol. FE [mpg] | 27.2 | 27.7 | 26.5 | 27.1 |
| Improvement [%] | 12.9 | 14.4 | 10.6 | 12.6 |
| Baseline Trip Time [min] | 30.6 | 30.6 | 30.6 | 30.6 |
| Adv. Fol. Trip Time [min] | 31.1 | 31.7 | 31.4 | 31.4 |
| Improvement [%] | -1.6 | -3.6 | -2.6 | -2.6 |

TABLE V
FUEL ECONOMY OF VARIOUS HIGHWAY DRIVING PROFILES

| | Fuel Economy [mpg] | Trip Time [min] |
|---|---|---|
| Optimal Profile | 27.6 | 28.5 |
| Slow Poke | 27.0 | 36.1 |
| Average Profile | 26.3 | 29..8 |
| Lead Foot | 24.8 | 25.6 |

worst case test for highway driving. Despite the drivers' efforts, the velocity tracking is not perfect. The degradation in fuel economy in the case of imperfect tracking is thus assessed. First, we predict the fuel consumption of the slow poke, lead foot, average, and optimal velocity trajectories by employing the fuel consumption model of the vehicle introduced and verified in Section II-B. In Table V, the fuel consumption of that velocity trajectories and the corresponding travel times are presented. To assess the potential improvement in fuel economy in the case of perfect tracking of the advised speed, the fuel economy on the simulated velocity profiles is compared. In

TABLE VI
RELATIVE FUEL ECONOMY OF HIGHWAY TESTS

| | | | | | Test-1 | | Test-2 | | Test-3 | | Average Driver | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Optimal | Slow P. | Average | Lead F. | Adv. Fol. | Base. | Adv. Fol. | Base. | Adv. Fol. | Base. | Adv. Fol. | Base. |
| Relative FE [%] | 0.0 | 2.0 | 4.6 | 10.2 | 1.5 | 14.1 | 0.4 | 14.1 | 4.0 | 14.1 | 1.8 | 14.1 |

TABLE VII
URBAN TEST ROUTE INFORMATION

| | Address |
|---|---|
| Origin | S Military, Dearborn, MI, USA |
| Destination | S Military, Dearborn, MI, USA |
| Waypoint 1 | 20061 Michigan Ave, Dearborn, MI, USA |
| Waypoint 2 | 18125 Rotunda Dr, Dearborn, MI, USA |
| Waypoint 3 | 20800 Oakwood Blvd, Dearborn, MI, USA |

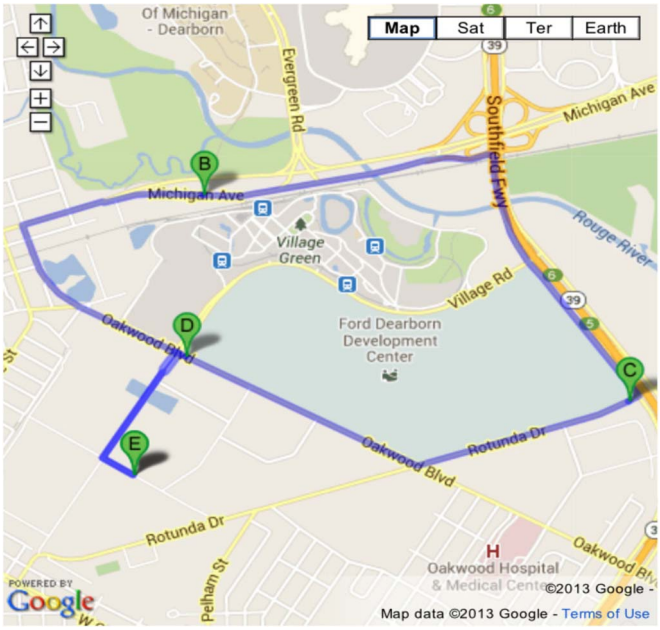



Fig. 13. Urban driving test route.

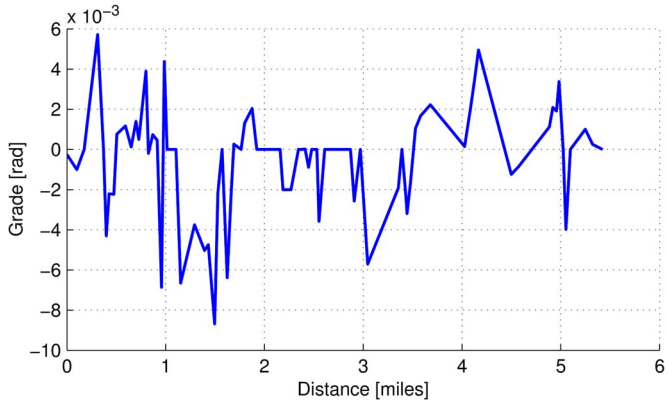



Fig. 14. Urban driving test route grade profile.

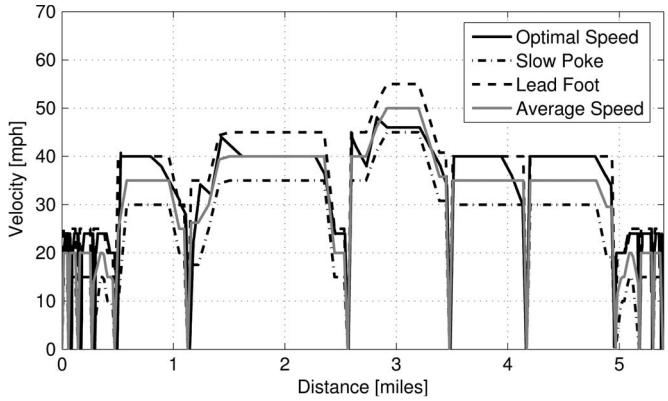



Fig. 15. Slow poke, foot lead, and average velocity profiles for the urban route.

Table VI, the relative fuel economy of each driving profile with respect to the advised velocity profile is given. Since in this particular test route the advised velocity profile is close to the lower speed limit, the relative fuel economy of the slow poke velocity profile is limited to 2%. On the other hand, the relative fuel economy of the lead foot velocity profile is 10.2%. Furthermore, when the drivers follow the advised velocity, on average, they have 1.8% relative fuel economy due to imperfect following of the advised velocity profile, as shown in Table VI. Finally, if the drivers could perfectly track the advised velocity, they could improve the fuel economy, on average, by 14.1% with respect to the baseline driving.

### *B. Urban Driving Test Results*

The second set of tests have been performed in an urban driving route. The route is in Dearborn, MI, USA, around the headquarters of Ford Motor Company. The origin destination and waypoints of the route are given in Table VII. Based on the trip information, the route is generated by the Google Maps Server. The generated route is 5.4 mi long and contains a number of traffic lights and stop signs, as presented in Fig. 13.

Based on the latitude and longitude values of the route, the elevation information is collected from the GIS server, and in the main server, the road grade profile is generated (see Fig. 14). Similar to highway tests, the minimum speed limits are selected to be 10 mi/h less than the maximum speed limits. The positions of the stop signs are determined and included as constraints in the optimization problem. As the operation sequence of the traffic lights are unknown, we simply consider the traffic lights as stop signs but suggest the driver to ignore the advised optimal velocity profile in the case of green at a traffic light. The calculated optimal velocity profile, the slow poke, the lead foot, and the average speed profiles are shown in Fig. 15. Seven different drivers performed two test runs: one for natural driving and one for advisor following. In the urban route, more drivers are used since the urban tests are more prone to external disturbances (such as traffic lights or variation in traffic flow). By increasing the number of drivers and averaging the results, more accurate judgements could be made. To compensate for traffic light disturbance, in the analysis, if a driver had to stop at a traffic light while it did not in another run, the stopping and reacceleration phases are discarded from the logged data.

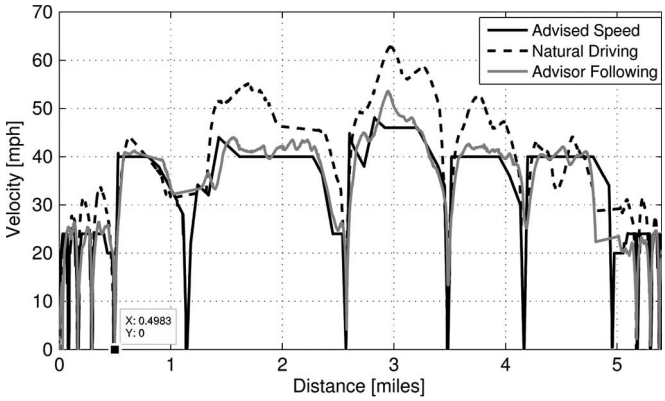



Fig. 16. Urban test route velocity profiles of the first driver (test-1).

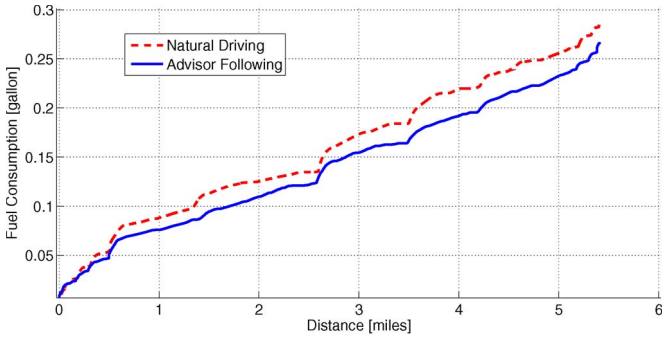



Fig. 17. Urban test route fuel consumption evolution of natural driving and advisor following cases for test-1.

TABLE VIII
COMPARISON OF URBAN TESTS

| Test No | #1 | #2 | #3 | #4 | #5 | #6 | #7 | Aver |
|---|---|---|---|---|---|---|---|---|
| Baseline FE [mpg] | 19.4 | 19.8 | 19.7 | 20.2 | 20.3 | 20.7 | 17.8 | 19.7 |
| Adv. Fol. FE [mpg] | 20.8 | 21.0 | 20.8 | 21.5 | 21.5 | 22.8 | 19.6 | 21.1 |
| Impr. [%] | 7.2 | 5.8 | 6.2 | 6.3 | 5.8 | 10.0 | 10.2 | 7.4 |
| Baseline TT [min] | 14.1 | 14.1 | 13.3 | 13.3 | 11.9 | 12.9 | 13.6 | 13.3 |
| Adv. Fol. TT [min] | 13.7 | 13.4 | 13.4 | 13.2 | 13.9 | 13.5 | 13.7 | 13.5 |
| Impr. [%] | 2.8 | 4.9 | -0.8 | 0.8 | -16.9 | -4.7 | -0.7 | -2.1 |

In Fig. 16, the natural driving and advisor following velocity profiles of the first driver are compared with the advised velocity profile. It is clear that the driver tends to drive around the maximum speed limit. In some sections, the maximum speed limit coincides with the advised (optimal) velocity, particularly in those regions where no significant improvement in fuel economy is expected. However, similar to highway driving, at the higher maximum speed limit region, the deviation between the optimal velocity and the driver's natural velocity is more significant, and more improvement in fuel economy can be expected. In Fig. 17, the cumulative fuel consumption curves along the travel distance for the runs of the first driver are shown. In Table VIII, the comparison of the travel times and fuel economy for both test runs of each driver and average values are presented. Table VIII shows that compared to highway tests, the fuel economy improvement is reduced, but it is still in the range of 5%–10%. An average fuel economy improvement of 7.4% is obtained, and the increase in the travel time is only 12 s for a 13.3-min driving cycle.

TABLE IX
FUEL ECONOMY OF VARIOUS URBAN DRIVING PROFILES

| | Fuel Economy [mpg] | Trip Time [min] |
|---|---|---|
| Optimal Velocity | 22.5 | 12.6 |
| Slow Poke | 17.3 | 21.2 |
| Average Profile | 21.0 | 14.2 |
| Lead Foot | 21.3 | 11.7 |

The predicted fuel economy and trip times of the slow poke, lead foot, average, and optimal velocity profiles are given in Table IX. Contrary to the highway driving, the slow poke velocity profile has much worse fuel economy relative to the advised velocity, i.e., 23.2%, whereas the lead foot profile has 5.4% relative fuel economy, as given in Table X. On average, the drivers could achieve 6.3% better fuel economy if they could perfectly follow the advised velocity. In that case, the fuel economy improvement of the drivers would be, on average, 12.5% compared to their average baseline driving profiles, which is similar to what was achieved in the highway tests. This clearly indicates that in urban driving, it is harder to follow the advised velocity profile. As we should expect, the generation of speed profiles that are easier to follow for the driver is an interesting research direction for future work and discussed next from a reinforcement learning point of view.

### C. Advisory System Adaptation by Reinforcement Learning

As discussed in the preceding sections, fuel economy degrades if the drivers do not follow the recommended speed profile. The driver may not follow the recommendation for several reasons, e.g., because he is not comfortable with the recommendation, or current traffic/safety conditions do not allow following the recommended profile. Different types of drivers may perceive the optimal velocity advice differently. It seems unrealistic to estimate the speed that the driver would be comfortable to follow unless we learn what speed is acceptable. A solution to this problem is the use of reinforcement learning of the driver's tendency; some applications of which are reported in [31] and [32]. In this paper, we propose a simplified form of reinforcement learning in which the driver's tendency to follow the recommended profile is continuously evaluated and the limits are adjusted accordingly. Therefore, an adaptive algorithm that can learn the driver perception of the recommended speed by monitoring his/her behavior with respect to the recommendation of the optimization algorithm is applied.

The adaptation algorithm uses the estimated driver characterization to dynamically adapt the speed limits to the specific driver pattern and improve its effectiveness. The adaptation increases the likelihood that the driver would follow the recommended speed profile and consequently increases the effectiveness of the advisory system.

The driver acceptance of the recommendations provided by the advisory system can be quantified through the frequency at acceptance [27]. The process of recursive calculation of the weighted frequency of rejection (with higher weights corresponding to the recent observations) is implemented by a low-pass filter with exponential smoothing, i.e.,

$$R(k)=\begin{cases}(1-\beta)R(k-1)+a, & \text{if } v_{\min}\le v(k)\le v_{\max}\\ (1-\beta)R(k-1), & \text{if } v(k)<v_{\min} \text{ or } v(k)>v_{\max}\end{cases} \quad (26)$$

TABLE X
RELATIVE FUEL ECONOMY OF URBAN TESTS

| | | | | | Test-1 | | Test-2 | | Test-3 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Optimal | Slow Poke | Average | Lead Foot | Adv. Fol. | Baseline | Adv. Fol. | Baseline | Adv. Fol. | Baseline |
| Relative FE [%] | 0.0 | 23.2 | 6.7 | 5.4 | 7.6 | 13.8 | 6.7 | 12.1 | 7.6 | 12.5 |
| | Test-4 | | Test-5 | | Test-6 | | Test-7 | | Average Driver | |
| | Adv. Fol. | Baseline | Adv. Fol. | Baseline | Adv. Fol. | Baseline | Adv. Fol. | Baseline | Adv. Fol. | Baseline |
| Relative FE [%] | 4.5 | 10.3 | 4.5 | 9.8 | 1.3 | 8.1 | 12.9 | 20.9 | 6.3 | 12.5 |

where $R$ is the rejection rate of the advised speed, and $\beta$ is a constant forgetting factor, $0 \leq \beta \leq 1$, controlling the rate of updating the weighted mean $R$. For a constant forgetting factor $\beta$, we obtain a vector of positive weights with unit sum by

$$W = \begin{bmatrix} \beta^n(1-\beta) & \beta^{n-1}(1-\beta) & \cdots & (1-\beta) \end{bmatrix}. \quad (27)$$

The vector $W$ defines a weighted average aggregating operator with exponentially decreasing weights that are parameterized by the forgetting factor $\beta$. Parameter $\beta$ defines the memory depth (the length of the moving window) of the weighted averaging aggregating operator. It can be shown that the memory depth $K_a$ is approximately related to the forgetting factor by $K_a = 1/(1-\beta)$. The operation of reinforcement-learning-based adaptation of the speed limits is illustrated in Fig. 18.

## VII. ADVISORY SYSTEM REQUIREMENTS

Here, we explore the technical details of the SAS in terms of the communication bandwidth, computation and memory requirements. At the end we also present a discussion of the system implementation in the vehicle and in the cloud.

### A. Communication Bandwidth Requirements

In the vehicle, we implemented a 4G LTE USB Modem to communicate with the cloud, as stated in Section V-B. LTE mobile wireless communication provides peak rates of 300 and 50 Mb/s for download and upload, respectively, [33], [34]. The study [35] on the performance of 4G LTE networks in the USA, however, shows that the average rates are 12.7 and 5.6 Mb/s for download and upload, respectively. To assess the required communication bandwidth, we consider the average rates. The data sent from the vehicle to the cloud are the origin, destination, and waypoints of the desired route. Assuming a maximum character length of MCL = 50 for an address, the desired route information requires

$$C_U = (M+2) \cdot \text{MCL} \cdot \text{BPB} \qquad M \in \mathbb{N} \quad (28)$$

number of bits, where $M$ is the number of waypoints (excluding initial and final positions) on the route, $\mathbb{N}$ is the set of natural numbers, and BPB is the number of bits per byte. For the highway and urban driving routes with two and three waypoints, we require to maximally send 200 and 250 characters, corresponding to $C_U = 1.6$ kb and $C_U = 2$ kb of data from the vehicle to the cloud, respectively.

On the other hand, the received data from the cloud to the vehicle consists of $d_k^*$, $v_k^*$, $\text{lat}_k^*$, and $\text{lon}_k^*$, $\forall p_k^* \in P^*$, as presented in Section V-A. Assuming that the transferred data are in double-precision floating-point format (i.e., 8 bytes) and $N+1$ is the number of points in $P^*$, the number of transferred data bits is

$$C_D = 32 \cdot \text{BPB} \cdot (N+1). \quad (29)$$

For the highway and urban driving routes, $N_H = 365$ and $N_U = 121$ corresponding to 93.4 and 30.9 kb of received data, respectively. Based on the average speeds of LTE mobile networks, for the tested highway and urban driving routes, the upload times are $C_U^{\text{High.}} < 0.4$ ms, $C_U^{\text{Urban}} < 0.3$ ms, and the download times are $C_D^{\text{High.}} = 7.4$ ms, $C_D^{\text{Urban}} < 2.4$ ms, respectively. The given values only refer to the data transfer time and do not include the latency in the communication protocol. The data transfer times show that an Internet connection with a moderate bandwidth in the vehicle is adequate for the cloud-based advisory system implementation in terms of communication load.

### B. Computation and Memory Requirements

In the proposed advisory system, we solve the DP algorithm (20) in the cloud where the computing unit is armed with a powerful Inter Core i7 processor with four cores and clock speed of 3.2 GHz and 16 GB RAM of memory. In the following, we present an estimate number of computations performed and present the total memory requirement in the cloud.

As presented in Fig. 5, the DP algorithm requires a recursive computation of the state transition cost $g_k(v_k, u_k, \Delta D_k)$, from $v_k$ to $v_{k+1}$ for $k = 0, 1, \ldots, N-1$. These recursive computations are the major source of computation load in the cloud. To determine the total number of state transition cost computations, first, we determine the total number of state transitions.

To estimate the number of state transitions, we utilize the 2-D, namely distance–velocity, computation space, which are quantized by $\Delta D_k$ and $\Delta v$. The number of intervals in distance dimension is $N$, and the number of intervals in velocity dimension is determined by

$$L = \left\lceil \frac{\|\mathbf{v}^{*\max}\|_\infty}{\Delta v} \right\rceil \quad (30)$$

where $\mathbf{v}^{*\max} = [v_1^{*\max}, v_2^{*\max}, \ldots, v_N^{*\max}]^T$, and the operator $\lceil \cdot \rceil$ denotes the ceiling function. Then the discrete velocity space becomes $\mathcal{L}_k = \mathcal{L} = \{v^0, v^1, \ldots v^L\}$ at $D_k$ for $k = 0, 1, \ldots, N$. For a fixed $k$ and $i \in \{0, 1, \ldots, L\}$, the number of possible transitions from $v_k^i$ to $\mathcal{L}_{k+1}$ is $(L+1)$. By considering all transitions from each element of $\mathcal{L}_k$ to $\mathcal{L}_{k+1}$ $\forall k \in \{0, 1, \ldots N-1\}$, we determine the total number of state transitions as $S_g = N \cdot (L+1)^2$.

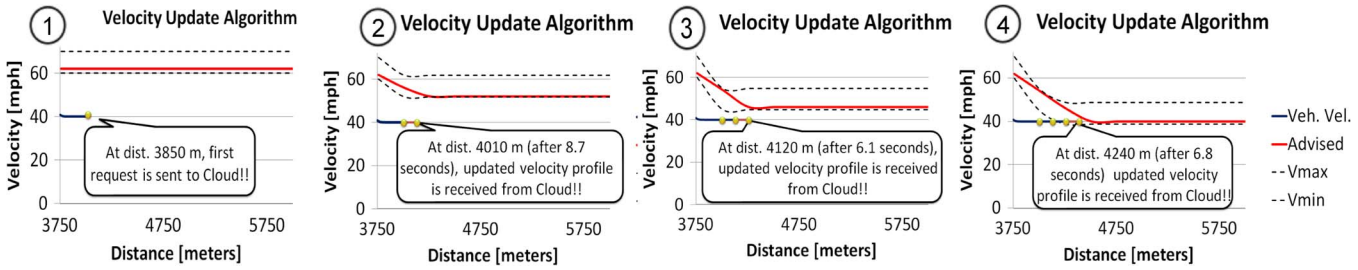



Fig. 18. Evolution of the advised speed obtained from the reinforcement learning algorithm.

TABLE XI
COMPUTATION NUMBERS AND CALCULATION TIME OF THE TESTS

| | Highway Driving | Urban Driving |
|---|---|---|
| $S_g$ $[-]$ | 473040 | 156816 |
| $S_{fg}$ $[-]$ | 13141 | 4356 |
| $S_{vbs}$ $[-]$ | 100 | 100 |
| $S_{tt}$ $[-]$ | 1773999 | 588060 |
| $S_T$ $[-]$ | 1787140 | 592416 |
| Calc. Time $[sec]$ | 3.2 | 1.8 |

Furthermore, the amount of computations for each transition cost calculation is not the same. For instance, the computation amount of a feasible transition cost, where the feasible states and control are characterized by $v_k \in \mathcal{V}_k \subset \mathcal{L}_k$, $v_{k+1} \in \mathcal{V}_{k+1} \subset \mathcal{L}_{k+1}$, and $u_k \in \mathcal{U}_D$, is different from the computation amount of an unfeasible transition cost. A feasible transition cost is the output of the vehicle backward simulator that requires a significant number of computations denoted by $S_{\rm vbs}$. On the other hand, the cost of an unfeasible state transition is infinity, as given in (22), thereby requiring one computation. To distinguish the feasible and unfeasible transitions, we define an average velocity range in which the state transitions are feasible as

$$V_{\rm fb} = \frac{1}{N}\sum_{i=0}^{N} v_i^{*\max} - v_i^{*\min} \tag{31}$$

Then, the number of intervals $R$ in $V_{\rm fb}$ is $R = \lceil V_{\rm fb}/\Delta v \rceil$, and the number of feasible state transitions are $S_{\rm fg} = N \cdot (R+1)^2$. After distinguishing the feasible and unfeasible transitions, we determine the total number of state transition cost computations as

$$S_{\rm tt} = S_{\rm fg} \cdot S_{\rm vbs} + (S_g - S_{\rm fg}). \tag{32}$$

Another source of computation load is due to the calculations of $\phi_k(v_k)$ [see (23)], for which the number of computations is equal to the number of feasible state transitions $S_\phi = S_{\rm fg}$. Ignoring the $P$ set manipulation and backtracking computations, the total number of computations in the cloud is then approximated by

$$S_T \approx S_{\rm tt} + S_\phi. \tag{33}$$

As given in the preceding section, the highway and urban driving routes have $N_H = 365$ and $N_U = 121$, respectively. For both routes, $\|\mathbf{v}^{*\max}\|_\infty = 70$ mi/h, $\Delta v = 2$ mi/h, and $V_{\rm fb} = 10$ mi/h and we assume that the backward simulator incurs an average of 100 computations, i.e., $S_{\rm vbs} = 100$. Based on these route specific parameters, we report the number of computations of each route in Table XI. In the cloud, that many calculations take 3.2 s for highway driving and 1.8 s for urban driving. In addition to the computation requirement, the DP algorithm also sets a certain memory size requirement on the computing unit. When we consider only the size of state transition cost information, the computation unit is required to store $S_g$ number of values, i.e., in double-precision floating-point format (8 bytes), we require $M_g = 8 \cdot S_g$ bytes of memory space. For the highway and urban driving routes with the values in Table XI, it amounts to 3.8 and 1.3 MB memory space, respectively. To determine the total memory consumed by the DP algorithm, we have utilized the Microsoft's task manager software and observed that the highway route uses 9.72 MB, whereas the urban driving route requires 5.63 MB of total memory.

### C. Assessment on the Implementation of the Advisory System

Here, we assess the advantages of the advisory system implementation in the cloud rather than in the vehicle. The microcontoller units (MCUs) are, in general, armed with much less powerful processors and with a smaller size of memory than personal computers. In a typical car, the clock speed values of MCUs are in the range of 40–180 MHz with 256 kB to 1 MB RAM memory and 1–4 MB Flash memory [36], [37]. However, as shown in the preceding section, DP requires 9.72 MB of free memory space for highway driving, i.e., for long trips, the memory size of the MCUs would be insufficient. Even if the MCUs would have enough memory, the clock speed values of MCUs used in automotive applications are approximately 20× slower than the processor used in the cloud, which has 3.2 GHz clock speed and four cores in the current framework, and roughly, the computation time of the DP algorithm in the vehicle would be 256 and 144 s for the urban and highway driving routes, respectively. That much latency is too large for real-time implementations of the advisory system and unacceptable with the reinforcement learning algorithm.

In addition, the cloud provides flexibility in the construction of the computing resource, and it is independent of the vehicle, i.e., we can extend the computing resource in the cloud as much as we require with the addition of multiple computing processor units and even with graphical processor units to perform general-purpose parallel computations without any change in the vehicle. On the other hand, the number of MCUs in the vehicle is rather limited [38].

Another advantage of the cloud framework is the low implementation cost. Although for a single vehicle the implementation cost of the system in the cloud and in the car is comparable, as the number of vehicles increases, the implementation cost of the system in the cloud would be significantly cheaper since the cloud can handle the computations of multiple cars simultaneously.

## VIII. Conclusion

In addition to the mechanical design, the smart utilization of information can significantly reduce vehicle energy consumption. The usage of cloud computing for vehicle applications rendered the real-time computation-intensive driving profile optimization possible. Although having a complex structure within itself, the cloud has a simple interaction with the vehicle; indeed, the only data sent to the cloud are the waypoints of the desired route, and the received data carry the velocity information of the points along the route. The tests have been executed in highway and urban drivings and performed by several drivers. The baseline and advisor following driving characteristics are averaged, which leads to more accurate assessment of the test results. The test results have shown that for highway driving, on average, 12.6% fuel economy improvement is achieved, whereas the improvement is 7.4% for urban driving. Compared to the highway tests, in urban driving, it is harder to follow the advised velocity profile due to external disturbances (other vehicles on the traffic, traffic lights, etc.).

The demonstrated application of the cloud computing for velocity profile optimization is a novel approach, and the preliminary results promise a significant reduction in fuel consumption. We believe that the increase in the number of agents (e.g., other vehicle on the network and pedestrians) and the infrastructures (e.g., traffic lights) communicating with the cloud will render the approach even more powerful.

## Acknowledgment

The authors would like to thank S. A. Kumar, a former graduate student, for his initial contributions to this paper.

## References

[1] A. Sciarretta and L. Guzzella, "Control of hybrid electric vehicles, control systems," *IEEE Control Syst. Mag.*, vol. 27, no. 2, pp. 60–70, Apr. 2007.

[2] G. E. Katsargyri *et al.*, "Optimally controlling hybrid electric vehicles using path forecasting," in *Proc. Amer. Control Conf.*, Jun. 10–12, 2009, pp. 4613–4617.

[3] Y. Bin, Y. Li, Q. Gong, and Z. Peng, "Multi-information integrated trip specific optimal power management for plug-in hybrid electric vehicles," in *Proc. Amer. Control Conf.*, Jun. 10–12, 2009, pp. 4607–4612.

[4] F. Gustafsson, "Automotive safety systems: Replacing costly sensors with software algorithms," *IEEE Signal Process. Mag.*, vol. 26, no. 4, pp. 32–47, Jul. 2009.

[5] M. E. Russell, C. A. Drubin, A. S. Marinilli, W. G. Woodington, and M. J. DelCheccolo, "Integrated automotive sensors," *IEEE Trans. Microw. Theory Tech.*, vol. 50, no. 3, pp. 674–677, 2002.

[6] G. Leer and D. Hefferman, "Expanding automotive electronic systems," *IEEE Computer*, vol. 35, no. 1, pp. 88–93, Jan. 2002.

[7] E. Koenders and J. Vreeswijk, "Cooperative infrastructure," in *Proc. IEEE Intell. Veh. Symp.*, 2008, pp. 721–726.

[8] M. Maile *et al.*, "Cooperative intersection collision avoidance system limited to stop sign and traffic signal violations," NHTSA, Federal Highway Admin., Washington, DC, USA, Tech. Rep. DOT HS 811 048, Oct. 2008.

[9] O. Servin, K. Boriboonsomsin, and M. Barth, "An energy and emissions impact evaluation of intelligent speed adaptation," in *Proc. Intell. Transp. Syst. Conf.*, Sep. 17–20, 2006, pp. 1257–1262.

[10] K. Boriboonsomsin, O. Servin, and M. Barth, "Selection of control speeds in dynamic intelligent speed adaptation system: A preliminary analysis," in *Proc. 14th Int. Con.—Road Safety Four Continents*, Bangkok, Thailand, Nov. 14–16, 2007, pp. 1–11.

[11] B. Asadi and A. Vahidi, "Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 3, pp. 1233–1238, May 2011.

[12] C. Raubitschek, N. Schutze, E. Kozlov, and B. Baker, "Predictive driving strategies under urban conditions for reducing fuel consumption based on vehicle environment information," in *Proc. IEEE Forum Integr. Sustainable Transp. Syst.*, Vienna, Austria, Jun. 29–Jul. 1, 2011, pp. 13–19.

[13] E. Ozatay, U. Ozguner, S. Onori, and G. Rizzoni, "Analytical solution to the minimum fuel consumption optimization problem with the existence of a traffic light," in *Proc. DSCC*, Oct. 2012, pp. 837–846.

[14] W. Dib, A. Chasse, A. Sciarretta, and P. Moulin, "Optimal energy management compliant with online requirements for an electric vehicle in eco-driving application," in *Proc. IFAC Workshop ECOSM Powertrain*, 2012, pp. 334–340.

[15] M. Ivarson, J. Aslund, and L. Nielsen, "Look-ahead control consequences of a non-linear fuel map on truck fuel consumption," *Proc. Inst. Mech. Eng. D, J. Autom. Eng.*, vol. 223, no. 10, pp. 1223–1238, Oct. 2009.

[16] E. Hellstrom, M. Ivarson, J. Aslund, and L. Nielsen, "Look-ahead control for heavy trucks to minimize trip time and fuel consumption," *Control Eng. Pract.*, vol. 17, no. 2, pp. 245–254, Feb. 2009.

[17] H. Khayyam, S. Nahavandi, and S. Davis, "Adaptive cruise control look-ahead system for energy management of vehicles," *Exp. Syst. Appl.*, vol. 39, no. 3, pp. 3874–3885, Feb. 2012.

[18] M. Kamal, M. Mukai, J. Murata, and T. Kawabe, "Ecological driver assistance system using model-based anticipation of vehicle-road-traffic information," *IET Intell. Transp. Syst.*, vol. 4, no. 4, pp. 244–251, Dec. 2010.

[19] S. Park, H. Rakha, K. Ahn, and K. Moran, "Predictive eco-cruise control: Algorithm and potential benefits," in *Proc. IEEE Forum Integr. Sustainable Transp. Syst.*, Vienna, Austria, Jun. 29–Jul. 1 2011, pp. 394–399.

[20] P. G. Howlett, I. P. Milroy, and P. J. Pudney, "Energy-efficient train control," *Control Eng. Pract.*, vol. 2, no. 2, pp. 193–200, Apr. 1994.

[21] P. G. Howlett, "Optimal strategies for the control of a train," *Automatica*, vol. 32, no. 4, pp. 519–532, Apr. 1996.

[22] J. Wollaeger *et al.*, "Cloud-computing based velocity profile generation for minimum fuel consumption: A dynamic programming based solution," in *Proc. ACC*, Jun. 2012, pp. 2108–2113.

[23] B. Furht and A. Escalante, *Handbook of Cloud Computing*. Berlin, Germany: Springer-Verlag, 2011.

[24] P. Mell and T. Grance, *NIST Definition of Cloud Computing*. Washington, DC, USA: National Institute of Standards and Technology, US Dept. of Commerce, 2011, Special Publication 800-145.

[25] M. Aoyama, "Computing for the next generation automobile," *Computer*, vol. 45, no. 6, pp. 32–37, Nov. 2012.

[26] Z. Li, C. Chen, and K. Wang, "Cloud computing for agent-based urban transportation systems," *IEEE Intell. Syst.*, vol. 26, no. 1, pp. 73–79, Jan./Feb. 2011.

[27] D. Filev and F. U. Syed, "Applied intelligent systems: Blending fuzzy logic with conventional control," *Int. J. Gen. Syst.*, vol. 39, no. 4, pp. 395–414, May 2010.

[28] J. B. Heywood, *Internal Combustion Engine Fundamentals*. New York, NY, USA: McGraw-Hill, 1988.

[29] L. Guzzella and C. H. Onder, *Modeling and Control of IC Engine Systems*, 2nd ed. New York, NY, USA: Springer-Verlag, 2010.

[30] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA, USA: Athena Scientific, 2005.

[31] M. Bichi *et al.*, "Stochastic model predictive control with driver behavior learning for improved powertrain control," in *Proc. 49th IEEE CDC*, 2010, pp. 6077–6082.

[32] S. Di Cairano, D. Bernardini, A. Bemporad, and I. Kolmanovsky, "Stochastic MPC with learning for driver-predictive vehicle control and its application to HEV energy management," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 3, pp. 1018–1031, May 2014.

[33] D. Astely *et al.*, "LTE: The evolution of mobile broadband," *IEEE Commun. Mag.*, vol. 47, no. 4, pp. 44–51, Apr. 2009.

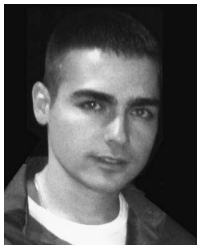
[34] T. Mshvidobadze, "Evolution mobile wireless communication and LTE networks," in *Proc. AICT*, Oct. 2012, vol. 45, pp. 1–7.

[35] J. Huang *et al.*, "A close examination of performance and power characteristics of 4 G LTE networks," in *Proc. 10th Int. Conf. MobiSys, Appl. Serv.*, Jun. 2012, pp. 225–238.

[36] E. Rogard, B. Vrignon, J. Shepherd, R. Moseley, and E. Sicard, "Characterization and modelling of parasitic emission of a 32-bit automotive microcontroller mounted on 2 types of BGA," in *Proc. IEEE Int. Symp. EMC*, 2009, vol. 6, pp. 66–71.

[37] A. Mayer and F. Hellwig, "System performance optimization methodology for Infineon's 32-bit automotive microcontroller architecture," in *Proc. DATE*, 2008, pp. 962–966.

[38] B. Fleming, "Microcontroller units in automobiles," *IEEE Veh. Technol. Mag.*, vol. 6, no. 3, pp. 4–8, Sep. 2011.



**Engin Ozatay** received the B.S. degree from Middle East Technical University, Ankara, Turkey, in 2009 and the M.S. degree from Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, in 2011, both in mechanical engineering. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH, USA.

He was an Intern in 2007 with Ford Otosan, Turkey, in 2008 with AVL Powertrain, Coventry, U.K., in 2009 with Roketsan, Turkey, in 2010–2011 with Hilti, Liechtenstein, and in 2013 with Ford Motor Company, Dearborn, MI, USA. Since 2011 he has been a Graduate Research Assistant with the Department of Electrical and Computer Engineering, The Ohio State University. His research interests include constraint optimization, model predictive control, robust control, estimation, powertrain control systems, intelligent transportation systems and autonomous vehicles.

Mr. Ozatay received an Excellence Scholarship and Opportunity Award from ETH.

**Simona Onori** (M'12) received the Laurea degree (*summa cum laude*) in computer engineering from University of Rome Tor Vergata, Rome, Italy, in 2003; the M.S. degree in electrical and computer engineering from University of New Mexico, Albuquerque, NM, USA, in 2004; and the Ph.D. degree in control engineering from the University of Rome Tor Vergata, in 2007.

Prior to joining the Department of Automotive Engineering, Clemson University, Clemson, SC, USA, in August 2013, she was a Research Scientist with the Center for Automotive Research, The Ohio State University (OSU), Columbus, OH, USA, and a Lecturer at the Mechanical and Aerospace Engineering Department, OSU, where she taught at undergraduate and graduate levels. She also previously worked with IBM (2000–2002) and Thales-Alenia Space (2007) in Rome, Italy. Her background is in control system theory, and her current research interests are in ground vehicle propulsion systems, including electric and hybrid electric drive trains, energy storage systems, and fuel cell systems, with an emphasis on modeling, simulation, optimization, and feedback control design for energy management in HEVs and PHEVs, fault diagnosis and prognosis for automotive system applications, and aging, characterization, modeling, and identification of advanced batteries. Her research has been funded by, among others, by Ford, General Motors, Cummins, the National Science Foundation, and the U.S. Department of Energy.

Dr. Onori is a member of the American Society of Mechanical Engineers (ASME) and the Society of Automotive Engineers (SAE). She is an ASME Associate Editor for the American Control Conference and Dynamic System Control Conference and an Associate Editor for *SAE International Journal of Advanced Powertrains*. She received the 2012 Lumley Interdisciplinary Research Award from the OSU College of Engineering and the TechColumbus 2011 Outstanding Technology Team Award.

**James Wollaeger** (M'06) was born in Cleveland, OH, USA. He received the B.S. degree in electrical engineering from Ohio Northern University, Ada, OH, USA, in 2010 and the M.S. degree in electrical engineering from The Ohio State University, Columbus, OH, 2012.

He was a Summer Intern for Parker Hannifin Corporation, Mayfield Heights, OH, where he learned how to work with embedded control system software development tools (2007–2009). After receiving the M.S. degree, he moved on to his current position at Robert Bosch LLC, Plymouth, MI, USA, where he is currently a Systems Engineer in the Chassis Systems Control Division, working on vehicle dynamics modeling.

**Umit Ozguner** received the Ph.D. degree from University of Illinois at Urbana-Champaign, Urbana, IL, USA.

He held positions at IBM Research Laboratories; University of Toronto, Toronto, ON, Canada; and Istanbul Technical University, Istanbul, Turkey. He has been with The Ohio State University, Columbus, OH, USA, since 1981, where he is a Professor of electrical and computer engineering and a Senior Fellow of the Center for Automotive Research. He is the TRC Inc. Chair on Intelligent Transportation Systems (ITS). He is the author or coauthor of more than 400 publications, including the book *Autonomous Ground Vehicles* (Artech House, 2011). He has advised more than 40 students in their M.S. research and more than 25 students in their Ph.D. research. His areas of research interest are in ITS, decentralized control, and autonomy in large systems. His research has been supported by many industrial companies, including Ford, GM, Honda, and ASELSAN, and organizations such as the National Science Foundation, Air Force Office of Scientific Research, National Aeronautics and Space Administration, and Air Force Weapons Analysis Laboratory.

Dr. Ozguner was the first President of the IEEE ITS Council in 1999 as it transformed into The IEEE ITS Society. He has been also the ITS Society Vice President for Conferences. He has also served the IEEE Control Society in various positions. He participated in the organization of many conferences and was the Program Chair of the first IEEE ITS Conference and the General Chair of the IEEE Control Systems Society 2002 CDC, ITS Society IV 2003, and ICVES 2008. Teams he coordinated participated successfully in the 1997 Automated Highway System Technology Demonstration, the Defense Advanced Research Projects Agency (DARPA) 2004 and 2005 Grand Challenges, the DARPA 2007 Urban Challenge, and the 2010 Multi Autonomous Ground-robotic International Challenge (MAGIC 2010) co-sponsored by the U.S. Army Tank Automotive Research, Development and Engineering Center.

**Giorgio Rizzoni** received the B.S., M.S., and Ph.D. degrees from University of Michigan, Ann Arbor, MI, USA, in 1980, 1982, and 1986, respectively, all in electrical and computer engineering.

Between 1986 and 1990 he was a Postdoctoral Fellow and then a Lecturer and Assistant Research Scientist with University of Michigan. In 1990 he joined the Department of Mechanical Engineering, The Ohio State University, Columbus, OH, USA, as an Assistant Professor. He was promoted to Associate Professor of mechanical engineering in 1995 and to Professor in 2000. In 1999 he was appointed Director of the Center for Automotive Research. He has been the Ford Motor Company Chair in Electromechanical Systems and a Professor with the Department of Electrical and Computer Engineering since 2002. He also holds a courtesy appointment in the Department of Design. His specialization is in dynamic systems and control, and his research activities are related to sustainable mobility.

Prof. Rizzoni is a Fellow of the Society of Automotive Engineers. He received numerous teaching and research awards, including the Stanley Harrison Award for Excellence in Engineering Education and the NSF Presidential Young Investigator Award.

**Dimitar Filev** (F'08) received the Ph.D. degree in electrical engineering from Czech Technical University, Prague, Czech Republic, in 1979.

He is a Senior Technical Leader in Intelligent Control and Information Systems with the Powertrain Control Research and Advanced Engineering division, Ford Motor Company, Dearborn, MI, USA. His research interests are in modeling and control of complex systems, intelligent control, fuzzy and neural systems, and their applications to automotive engineering.

Dr. Filev is a Fellow of the International Fuzzy Systems Association. He received the 2008 Norbert Wiener Award from the IEEE Systems, Man, and Cybernetics Society and the 2007 IFSA Outstanding Industrial Applications Award.

**John Michelini** received the B.S. degree in mechanical engineering from University of Detroit, Detroit, MI, USA; the M.S. degree in mechanical engineering from University of Michigan, Ann Arbor, MI; and the degree in engineering management from Wayne State University, Detroit.

He has been with Powertrain Control Research and Advanced Engineering for more than 10 years and with Ford Motor Company, Dearborn, MI, for more than 20 years. He is the coauthor of four conference papers and he holds more than 65 U.S. patents. He has worked on powertrain controls for various engine fuel economy technologies, Ti-VCT, and camless (EVA).

**Stefano Di Cairano** (M'08) received the Master (Laurea) and Ph.D. degrees in information engineering from University of Siena, Siena, Italy, in 2004 and 2008, respectively, and the International Curriculum Option for Doctoral Studies in Hybrid Control Systems.

He was a Visiting Student with the Technical University of Denmark, Lyngby, Denmark, in 2002–2003 and at the California Institute of Technology, Pasadena, CA, USA, in 2006–2007. In 2008–2011, he was with the Powertrain Control Research and Advanced Engineering, Ford Motor Company, Dearborn, MI, USA. Since 2011 he has been with the Mechatronics Group, Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, where he is now a Team Leader and a Senior Principal Member of Research Staff. His research is on advanced control strategies for complex mechatronic systems in automotive, factory automation, and aerospace. His research interests include model predictive control, constrained control, networked control systems, hybrid systems, and optimization.

Dr. Di Cairano is the Chair of the IEEE Control Systems Society (CSS) Technical Committee on Automotive Controls, a member of the IEEE CSS Conference Editorial Board, and an Associate Editor of IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY.