MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

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

Ozatay, E.; Onori, S.; Wollaeger, J.; Ozguner, U.; Rizzoni, G.; Filev, D.; Michelini, J.; Di Cairano, S.

TR2014-120 May 2014

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.

IEEE Transactions on Intelligent Transportation Systems

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Copyright © Mitsubishi Electric Research Laboratories, Inc., 2014
201 Broadway, Cambridge, Massachusetts 02139

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

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

***Abstract*—Driving style, road geometry, and traffic conditions have significant impact on the vehicle 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 study, 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 which 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 significant effect on the travel time. Although the current research implements the speed advisory system (SAS) in a conventional vehicle, the solution is generic and it is applicable to any kind of powertrain structure.**

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

I. Introduction

As the number of vehicles on the road has increased worldwide, the importance of decreasing 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, pneumatic, mechanical, fuelcell, etc., 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 are 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 US, 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 *et al.* [11] have proposed a control algorithm that adapts the velocity profile to guarantee 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 sub-optimal velocity trajectory and implemented the algorithm in an existing cruise control system. A similar approach has been 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],

Manuscript received .....

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 is with the Automotive Engineering Department, Clemson University, Greenville, SC, 29607 USA, e-mail: *sonori@clemson.edu*

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

D. Filev and J. Michelini are with Powertrain Control R&A, Ford Motor Company, Dearborn, MI 48124, USA e-mail: (*dfilev@ford.com, jmichel1@ford.com*).

S. Di Cairano was with Powertrain Control R&A, Ford Motor Company, at the time of this work. He is now with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, e-mail: *dicairano@ieee.org*.

This paper is based on the work supported by Ford Motor Company, Deaborn, USA under the University Research Project (URP) Program and the National Science Foundation under the Cyber-Physical Systems (CPS) Program(ECCS-0931669).

[16] the authors have 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 on board 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. Also 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.

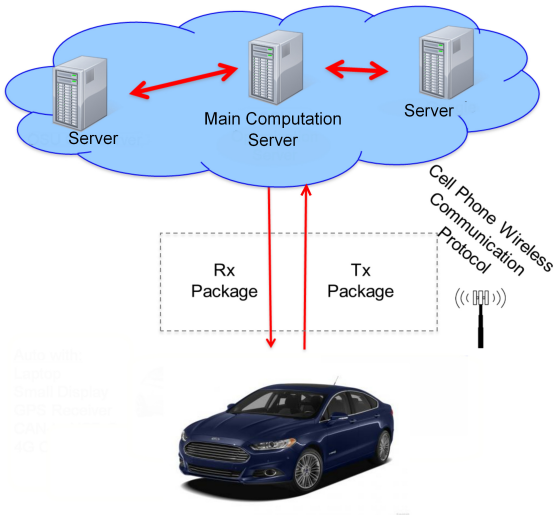


Fig. 1. Cloud-vehicle interaction schematic

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

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 which 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 limited number of automotive applications and preliminary analysis [25], [26]. Some examples are the Ford's *MyFordMobile* application [27] which uses an on board 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.

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 the paper by summarizing the overall system and presenting possible future developments.

II. VEHICLE MODELING

In this section we introduce the vehicle longitudinal and fuel consumption models and their use in a vehicle simulator. The simulator operates backwards 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 Tab. I.

The longitudinal dynamics of the vehicle is given by

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

where v is the vehicle speed, m_{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, F_{trac}, is the force supplied by the engine and transmitted to the tires by means of mechanical connections and its formulation is given by

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

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

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

where g is the gravitational constant, α is the road grade, 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_{aero}, is

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

where ρ 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_{aero} dominates the other resistive forces at high velocities. The road grade force, F_{grade}, is defined as

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

Finally, F_{brake} is brake force. By substituting (2), (3), (4) and (5) into (1) we obtain the vehicle longitudinal dynamics as

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

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, while 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

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

where C_0, C_1, C_2, C_3 are functions of the engine speed, determined experimentally at Ford Technical Center, Dearborn, MI and reported in Tab. 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

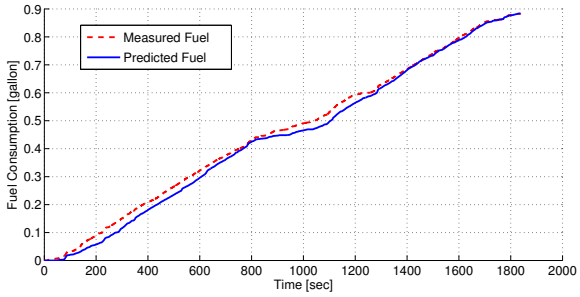


Fig. 2. Measured and model predicted cumulative fuel consumption curves for the same velocity profile from an actual experiment.

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 (α_{ave}) is executed by the vehicle backward simulator as depicted in Fig. 3. In the 1-sec sampling block, first, we determine the travel time and the average acceleration of the road segment based on p, q, l by

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

Then, we determine the velocity at each one second sample time, i.e, we calculate v_k for $k = 0, 1, \cdots, \lfloor t \rfloor$ where $v_k = p + k \cdot a_{ave}$ and $\lfloor \cdot \rfloor$ is the floor operator. We insert the one second interval average speed, $v_{ave,k} = \frac{v_k + v_{k+1}}{2}$ if $k \in \{0, 1, \cdots, \lfloor t \rfloor - 1\}$, and $v_{ave,k} = \frac{v_k + q}{2}$ if $k = \lfloor t \rfloor$, a_{ave} and α_{ave} into Eq. (6) to calculate the required torque at the wheels, $T_{w,k}$ at time k. Then we determine the gear number, ξ_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 $\omega_{e,k} = \frac{\gamma(\xi_k)\cdot\gamma_{fd}}{R_{wh}} \cdot v_k$. The mechanical limitations 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 points 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 Eq. (7) to obtain the instantaneous fuel consumption and sends it to the cumulative summation block which calculates the cumulative fuel consumption. We repeat the procedure until we process $\forall k \in \{0, 1, \cdots, \lfloor t \rfloor\}$ and obtain the total fuel consumption of

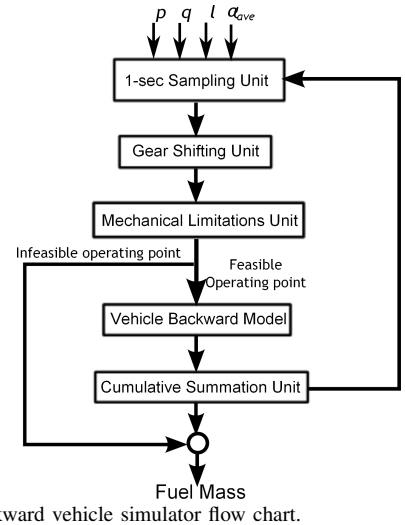


Fig. 3. Backward vehicle simulator flow chart.

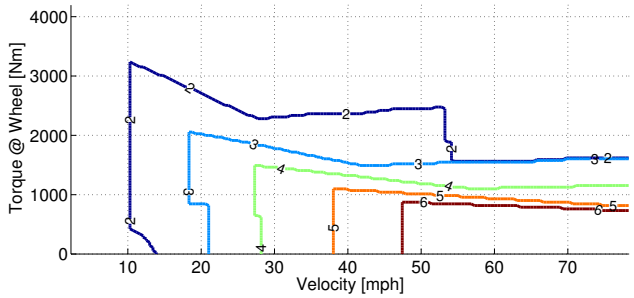


Fig. 4. The contour plot of the gear shifting strategy

the road segment, m_{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 spatial domain by means of the following transformation (which converts time domain equations into 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:

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

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

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

obtained from (6) through (9).

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

Control Constraint Set

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

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

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

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

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

State Constraint Set

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

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

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

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

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.

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 Eq. (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 $s = 1, 2, \cdots, m$), $\alpha(D)$, $v_{lim}^{min}(D)$ and $v_{lim}^{max}(D)$ are fixed parameters and it is straight forward to determine $\mathcal{U}_D$ and $\mathcal{X}_D$. Although, the boundary condition (17) seems to render the system state dynamic equation (11) undefined at the boundary points, this is not the case due to the inherent discrete

nature of the dynamic programming (DP) solution used to solve the spatial domain optimization, i.e., in 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 Eq. (11) is taken as $v = \frac{v(k)+v(k+1)}{2}$ and the condition $v = 0$ rendering the problem undefined never occurs.

IV. Dynamic Programming Algorithm

The section details the solution of the non-linear optimal control problem formulated in the spatial domain by the DP algorithm. A set of points identify the route specific data, and the route. Specifically the route consists of the points $P = \{p_0, p_1, p_2, \cdots, p_K\}$. Each point, $p_k \in P$ for $k = 0, 1, \cdots, K$, has its own characteristic parameters; $p_k = \left[lat_k, lon_k, d_k, h_k, \alpha_k, v_k^{max}, v_k^{min}\right]^T$ where lat_k is the latitude, lon_k is the longitude, d_k is the distance, h_k is the elevation, α_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.

Assumptions

- lat_k and lon_k for $k = 0, 1, \cdots, K$ are known.
- d_k is known and $d_{k+1} > d_k$ for $k = 0, 1, \cdots, K-1$.
- h_k and v_k^{max} for $k = 0, 1, \cdots, K$ are known and change linearly between p_k and p_{k+1}.

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

Manipulation of Set P

The manipulation of P aims at creating a new set $P^* = \{p_0^*, p_1^*, p_2^*, \cdots, p_N^*\}$ such that $S \subset P^*$ and the elements of P^* are dispersed with a regular pattern. The easiest way is to define a constant $d_c = d_k \quad \forall k \in \{0, 1, \cdots, N-1\}$ 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

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

Then we generate $p_k^* \in P^*$ such that $d_k^* = \Delta D_k$ for $k = 0, 1, \cdots, N-1$. 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.

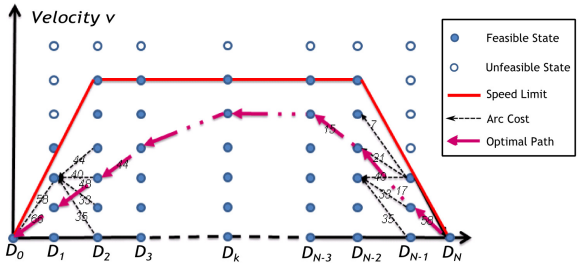


Fig. 5. Schematic Representation of DP Algorithm

The grade between any two consecutive points is calculated by $\bar{\alpha}_k^* = tan^{-1}\left(\frac{h_{k+1}^* - h_k^*}{d_k}\right)$. 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

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

The optimal control problem is cast in such a way that the fuel consumed is the only cost criterion (Eq. 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 study, we limit $v_k^{*\,min}$ to be 10 mph 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, Δv, where too small step values unnecessarily increase the calculation time and too large values reduce performance. In this paper we selected $\Delta v = 2$ mph previously determined [22] as a good compromise between the calculation time and the accuracy of the solution.

DP Formulation

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

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

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 Eq. (17), the terminal cost function, $g_N(v_N)$, is defined as

$$g_N(v_N) = \begin{cases} 0 & \textit{if} \quad v_N = 0 \\ \infty & \textit{if} \quad v_N \neq 0 \end{cases} \tag{21}$$

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

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

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 $V^* = \{v_0^*, v_1^*, v_2^*, \cdots, v_N^*\}$,

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

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

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

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

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 firstly, calculates the transition cost to the points at $(N-1)^{th}$ step. The cost of the transition to a feasible point (filled circle) is determined by the vehicle backward simulator, while the cost of the transition to an unfeasible point (unfilled circle) is infinity as defined by Eq. (22). After we determine the transition costs from step $k+1$ to k for $k = 0, 1, \cdots, N-1$ in backwards 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 onwards is adjusted to maintain future optimality.

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), at 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 way points 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 (API) and the Google Maps Server generates the route in the form of polylines defined by the latitude (lat_k) and longitude (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 (SOAP). The ArcGIS Server containing the digital elevation model(DEM) 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 above mentioned communication sequence the main server gathers h_k, lat_k, lon_k, and d_k information of p_k for $k = 0, 1, \cdots, K$ which constitutes P as described

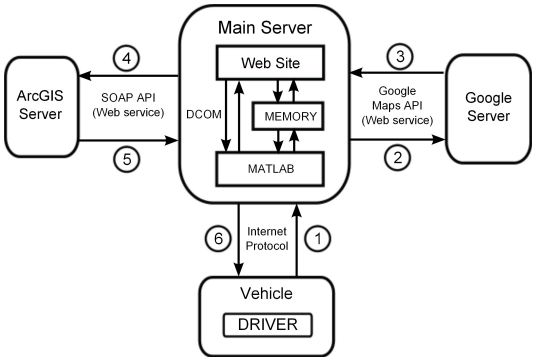


Fig. 6. Cloud communication sequence and protocol diagram.

in Section IV. Employing the Distributed Component Object Model (DCOM), 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^*, lat_k^*, and lon_k^* of $p_k^* \in P^*$ for $k = 0, 1, \cdots, N$, 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 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.

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.

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

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 ECU and laptop a parallel connection to the CAN data bus of the vehicle is established.

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.

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

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 (GUI) to show the advised velocity to the user and allows driver to interact with the cloud.

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 the waypoints and send a request to the cloud. In the cloud the calculations are performed, 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. Here after, 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

TABLE III
Highway Test Route Inforamtion.

	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

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 have 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 are determined, and the minimum speed limits are selected as 10 mph 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 which are 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, while the "Lead Foot" driving profile corresponds to 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

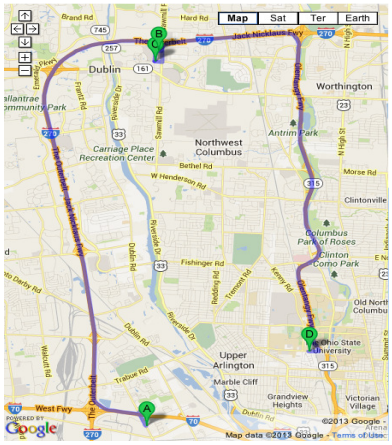


Fig. 8. The highway driving route

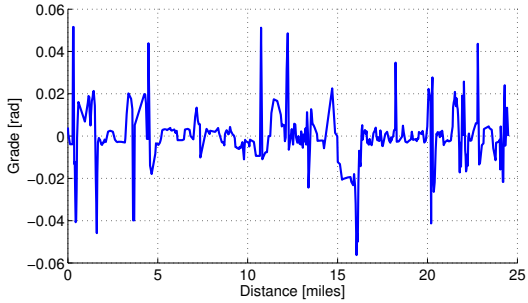


Fig. 9. The highway test route grade profile

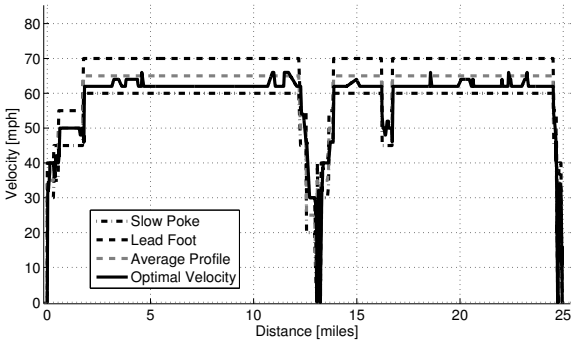


Fig. 10. Slow poke, foot lead and average velocity profiles for the highway route.

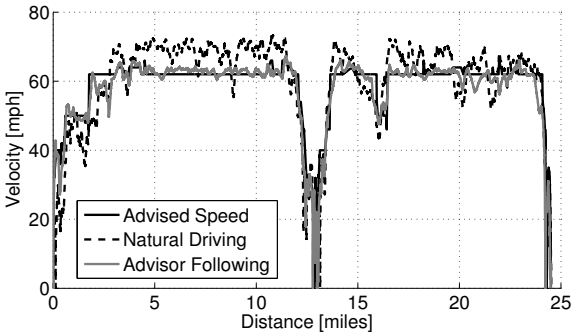


Fig. 11. The highway test route velocity profiles

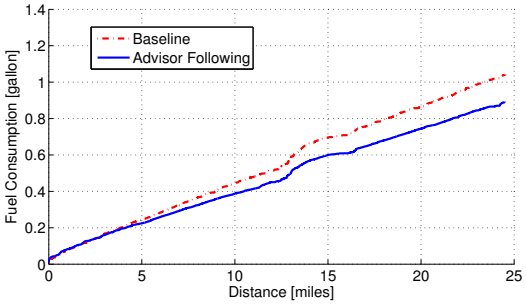


Fig. 12. The highway test route fuel consumption evolution for baseline and advisor following cases

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 minutes while it is 31.1 minutes 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 in average by 12.6% with 3.6% increase in travel time in the 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 case of perfect tracking of the advised speed, the fuel economy on the simulated velocity profiles are compared. In Table VI, the relative fuel economy of each driving profile with respect to the advised velocity profile are 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, in 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 in average by 14.1% with respect to the baseline driving.

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

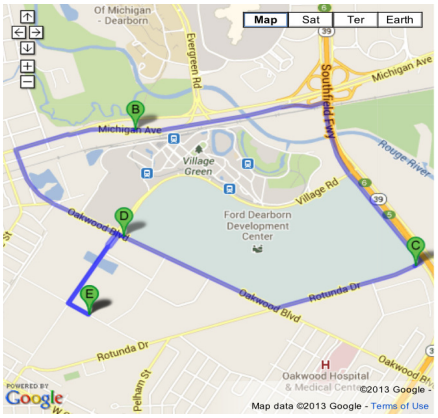


Fig. 13. Urban driving test route

B. Urban Driving Test Results

TABLE VII
URBAN TEST ROUTE INFORAMTION.

	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

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 miles 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 mph less than the maximum speed limits. The position of the stop signs are determined and included

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

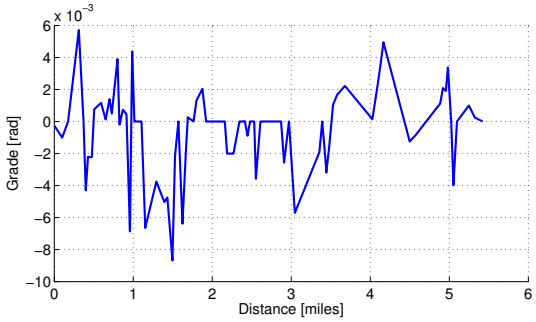


Fig. 14. Urban driving test route grade profile

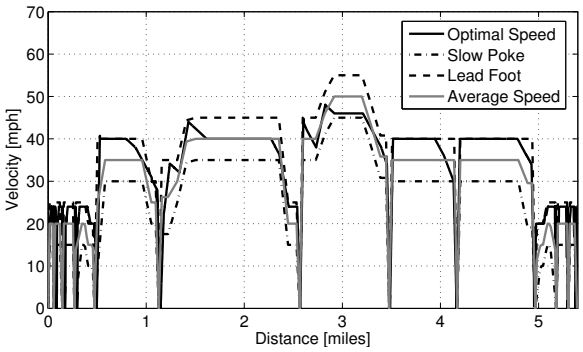


Fig. 15. Slow poke, foot lead and average velocity profiles for the urban route.

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 (like 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 re-acceleration phases are discarded from the logged data.

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, especially 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 drivers natural velocity is more significant, and more improvement in fuel economy can be expected. In Fig. 17, the cumulative fuel consumption curves

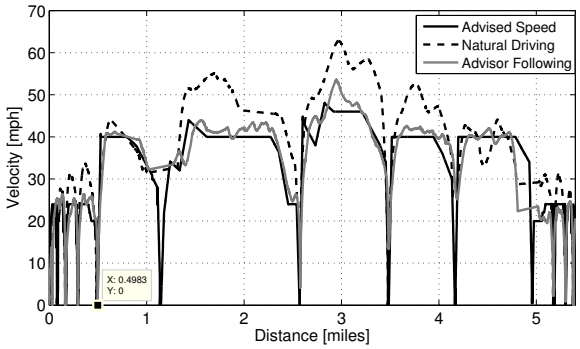


Fig. 16. The urban test route velocity profiles of the first driver (test-1).

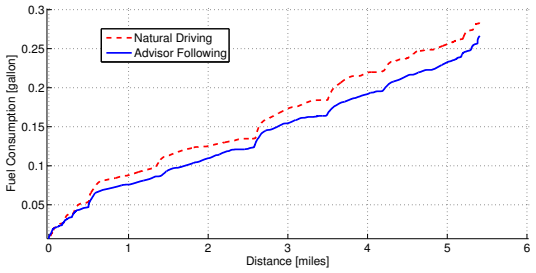


Fig. 17. The urban test route fuel consumption evolution of natural driving and advisor following cases for test-1

along the travel distance for the runs of the first driver are shown. In Table VIII, the comparison of the travel times and

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

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 seconds for a 13.3 minutes driving cycle.

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, 23.2%, while the lead foot profile has 5.4% relative fuel economy as given in Table X. In average the drivers could achieve 6.3% better fuel economy if they could

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

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

perfectly follow the advised velocity. In that case the fuel economy improvement of the drivers would be in 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], [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

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

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

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

The vector *W* defines a weighted average aggregating operator with exponentially decreasing weights that are parametrized by the forgetting factor β. Parameter β 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

In this section, we explore the technical details of the speed advisory system 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 Sec. V-B. LTE mobile wireless communication provides peak rates of 300Mb/s and 50Mb/s for download and upload, respectively, [33],[34]. The study [35] on the performance of 4G LTE Networks in the United States, however, shows that the average rates are 12.7 Mb/s and 5.6 Mb/s for download and upload. To assess the required communication bandwidth, we consider the average rates. The data sent from the vehicle to cloud is the origin, destination and way points of the desired route. Assuming maximum character length of $MCL = 50$ for an address, the desired route information requires

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

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

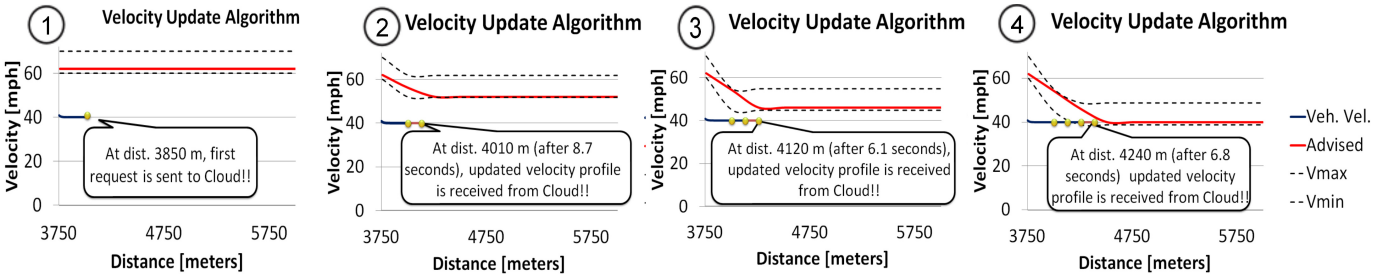


Fig. 18. Evolution of the advised speed obtained from the reinforcement learning algorithm.

characters, corresponding to $C_U = 1.6$ kbit and $C_U = 2$ kbit 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^*, lat_k^* and lon_k^*, $\forall p_k^* \in P^*$ as presented in Sec. V-A. Assuming the transferred data is in double precision floating point format (i.e., 8-byte) and $N+1$ is the number of points in P^*, the number of transferred data bits is

$$C_D = 32 \cdot BPB \cdot (N+1) \tag{29}$$

For the highway and urban driving routes, $N_H = 365$ and $N_U = 121$ corresponding to 93.4 kbit and 30.9 kbit 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 $C_U^{High.} < 0.4\ msec$, $C_U^{Urban} < 0.3\ msec$, and the download times are $C_D^{High.} = 7.4\ msec$, $C_D^{Urban} < 2.4\ msec$, respectively. The given values only refer to the data transfer time and does 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, Eq. (20), in the cloud where the computing unit is armed with powerful Inter Core i7 processor with 4 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 $k = 0, 1, \cdots, N-1$. 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-dimensional, namely distance-velocity, computation space which are quantized by ΔD_k and Δv. The number of intervals in distance dimension is N and the number of intervals in velocity dimension is determined by

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

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

Furthermore, the amount of computations for each transition cost calculation is not the same. For instance, the computation amount of feasible transition cost where the feasible states and control 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 which requires a significant number of computations denoted by S_{vbs}. On the other hand, the cost of an unfeasible state transition is infinity, as given in Eq. (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

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

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

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

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

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

As given in the preceding section, the highway and urban driving routes have $N_H = 365$ and $N_U = 121$, respectively. For both routes, $\|\mathbf{v}^{*\mathbf{max}}\|_\infty = 70$ mph, $\Delta v = 2$ mph, and $V_{fb} = 10$ mph and we assume that the backward simulator incurs an average 100 computations, i.e., $S_{vbs} = 100$. Based on these route specific parameters we report the number of computations of each route in Tab. XI. In the cloud, that many calculations take 3.2 sec for highway driving and 1.8 seconds 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

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

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 Tab. XI, it amounts to 3.8 MB 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 while the urban driving route requires 5.63 MB of total memory.

C. Assessment on the Implementation of the Advisory System

In this section 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 smaller size of memory than personal computers (PCs). In a typical car, the clock speed of MCUs are in the range of 40-180 MHz with 256 KB - 1 MB RAM memory and 1MB - 4MB 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 of MCUs used in automotive applications are approximately 20x slower than the processor used in the cloud which has 3.2 GHz clock speed and 4 cores in the current framework and, roughly, the computation time of DP algorithm in the vehicle would be 256 sec and 144 sec 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 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 addition of multiple computing processor units (CPUs) and even with graphical processor units (GPUs) 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

Besides 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 information sent to the cloud are the waypoints of the desired route and the received data carries 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 in average 12.6% fuel economy improvement is achieved while 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 significant reduction in fuel consumption. We believe that the increase in the number of agents (e.g., other vehicle on the network, pedestrians) and the infrastructures (e.g., traffic lights) communicating with cloud will render the approach even more powerful.

ACKNOWLEDGMENT

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

REFERENCES

[1] A. Sciarretta, L. Guzzella, *Control of Hybrid Electric Vehicles, Control Systems*, IEEE Control Systems Magazine, vol.27, no.2, pp.60-70, April 2007.

[2] G.E. Katsargyri, I. V. Kolmanovsky, J. Michelini,M. L. Kuang, A. M. Phillips, M. Rinehart, M. A. Dahleh, *Optimally Controlling Hybrid Electric Vehicles Using Path Forecasting*, American Control Conference, pp. 4613-4617, 10-12 June 2009.

[3] Y. Bin, Y. Li, Q. Gong, Z. Peng, *Multi-information Integrated Trip Specific Optimal Power Management for Plug-in Hybrid Electric Vehicles*, American Control Conference, pp.4607-4612, 10-12 June 2009.

[4] F. Gustafsson, *Automotive Safety Systems: Replacing Costly Sensors with Software Algorithms*, IEEE Signal Processing Magazine, pp. 32-47 ,July 2009.

[5] M. E. Russell, C. A. Drubin, A. S. Marinilli, W. G. Woodington, and M. J. DelCheccolo, *Integrated Automotive Sensors*, IEEE Trans. Microwave Theory Tech., vol. 50, no. 3, 2002.

[6] G. Leer, D. Hefferman, *Expanding Automotive Electronic Systems*, IEEE Computer, vol: 35, pp. 8893, Jan. 2002.

[7] E. Koenders, J. Vreeswijk, *Cooperative Infrastructure*, in Proc. IEEE Intelligent Vehicle Symp., pp 721-726, 2008.

[8] M. Maile, F. Ahmed-Zaid, L. Caminiti, J. Lundberg, P. Mudalige, C. Pall, *Cooperative Intersection Collision Avoidance System Limited to Stop Sign and Traffic Signal Violations*, NHTSA, Federal Highway Administration, Washington, DC, Tech. Rep. DOT HS 811 048, Oct., 2008.

[9] O. Servin, K. Boriboonsomsin, M. Barth, *An Energy and Emissions Impact Evaluation of Intelligent Speed Adaptation*, Intelligent Transportation Systems Conference, pp.1257-1262, 17-20 Sept. 2006.

[10] K. Boriboonsomsin, O. Servin, M. Barth, *Selection of Control Speeds in Dynamic Intelligent Speed Adaptation System: A Preliminary Analysis*, 14th International Conference - Road Safety on Four Continents, Bangkok, Thailand, 14-16 Nov. 2007.

[11] B. Asadi, A. Vahidi, *Predictive Cruise Control: Utilizing Upcoming Traffic Signal Information for Improving Fuel Economy and Reducing Trip Time*, IEEE Transactions on Control Systems Technology, pp. 1233-1238, Vol. 19, No. 3, May 2011.

[12] C. Raubitschek, N. Schutze, E. Kozlov, B. Baker, *Predictive Driving Strategies under Urban Conditions for Reducing Fuel Consumption based on Vehicle Environment Information*, 2011 IEEE Forum on Integrated and Sustainable Transportation Systems Vienna, Austria, June 29 - July 1, 2011.

[13] E. Ozatay, U. Ozguner, S. Onori, G. Rizzoni, *Analytical Solution to the Minimum Fuel Consumption Optimization Problem with the Existence of a Traffic Light*, Dynamic Systems and Control Conference (DSCC), Oct. 2012.

[14] W. Dib, A. Chasse, A. Sciarretta, P. Moulin, *Optimal Energy Management Compliant with Online Requirements for an Electric Vehicle in Eco-Driving Application*, IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling (ECOSM'12), 2012.

[15] M. Ivarson, J. Aslund, and L. Nielsen, *Look-ahead control consequences of a non-linear fuel map on truck fuel consumption*, Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, pp. 1223-1238, Oct., 2009.

[16] E. Hellstrom, M. Ivarson, J. Aslund, L. Nielsen, *Look-ahead control for heavy trucks to minimize trip time and fuel consumption*, Control Engineering Practice 17, pp. 245254, 209.

[17] H. Khayyam, S. Nahavandi, S. Davis, *Adaptive Cruise Control Look-ahead System for Energy Management of Vehicles*, Expert Systems with Applications,39 (3), pp. 3874-3885

[18] M. Kamal, M. Mukai, J. Murata,T. Kawabe, *Ecological driver assistance system using model-based anticipation of vehicle-road-traffic information*, IET Intell. Transp. Syst., 2010, Vol. 4, Iss. 4, pp. 244-251, 2010.

[19] S. Park, H.Rakha, K. Ahn, K. Moran, *Predictive Eco-Cruise Control: Algorithm and Potential Benefits*, IEEE Forum on Integrated and Sustainable Transportation Systems, Vienna, Austria, June 29 - July 1, 2011

[20] P. G. Howlett, I.P. Milroy, P.J. Pudney, *Energy-efficient train control*, Control Engineering Practice, Volume 2, Issue 2, Pages 193-200, ISSN 0967-0661, April 1994.

[21] P. G. Howlett, *Optimal strategies for the control of a train*, Automatica, Volume 32, Issue 4, Pages 519-532, ISSN 0005-1098, April 1996.

[22] J. Wollaeger, S.A. Kumar, S. Onori, D. Filev, U. Ozguner, G. Rizzoni, S. Di Cairano *Cloud-Computing Based Velocity Profile Generation for Minimum Fuel Consumption: A Dynamic Programming Based Solution*, IEEE American Control Conference (ACC), pp 2108 - 2113, June, 2012.

[23] B. Furht, A. Escalante, *Handbook of Cloud Computing*, Springer, 2011.

[24] P. Mell. T. Grance, *NIST Definition of Cloud Computing*, Special Publication 800-145, National Institute of Standards and Technology, U.S. Dept. of Commence, 2011.

[25] M. Aoyama, *Computing for the Next Generation Automobile*, IEEE Computer, Vol: 45, pp. 32-37, 2012.

[26] Z. Li, C. Chen, K. Wang, *Cloud Computing for Agent-Based Urban Transportation Systems*, IEEE Intelligent Systems, pp. 73-79, Jan., 2011.

[27] D. Filev, F. U. Syed, *Applied Intelligent Systems: Blending Fuzzy Logic with Conventional Control*, International Journal of General Systems, 39:4, pp. 395-414, 2010.

[28] J. B. Heywood, *Internal Combustion Engine Fundamentals*, McGraw-Hill Inc, 1988.

[29] L. Guzzella, C. H. Onder, *Modeling and Control of IC Engine Systems*, Springer, 2nd Edition, 2010.

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

[31] M. Bichi, G. Ripaccioli, S. Di Cairano, D. Bernardini, A. Bemporad, I. Kolmanovsky, *Stochastic Model Predictive Control with Driver Behavior Learning for Improved Powertrain Control*, 49th IEEE Conference on Decision and Control (CDC), pp. 6077–6082, 2010

[32] S. Di Cairano, D. Bernardini, A. Bemporad, I. Kolmanovsky, *Stochastic MPC with Learning for Driver-predictive Vehicle Control and its Application to HEV Energy Management*, IEEE Trans. Control System Technology, In press, 2013.

[33] D. Astely, E. Dahlman, A. Furuskar, Y. Jading, M. Lindstrom, S. Parkvall, *LTE: The Evolution of Mobile Broadband*, IEEE Communications Magazine, Vol. 47, pp. 44-51, April, 2009.

[34] T. Mshvidobadze, *Evolution Mobile Wireless Communication and LTE Networks*, Application of Information and Communication Technologies (AICT), , vol: 45, pp. 1-7, October, 2012.

[35] J. Huang, F. Qian, A. Gerber, Z. M. Mao , S. Sen, O. Spatscheck, *A Close Examination of Performance and Power Characteristics of 4G LTE Networks*, The Proceedings of 10^{th} International Conference on Mobile Systems, Applications and Services (MobiSys), pp. 225-238, June, 2012.

[36] E. Rogard, B. Vrignon, J. Shepherd, R. Moseley, E. Sicard, *Characterization and Modelling of Parasitic Emission of a 32-bit Automotive Microcontroller Mounted on 2 Types of BGA*, IEEE International Symposium on Electromagnetic Compatibility (EMC), Vol.6, pp 66-71, 2009.

[37] A. Mayer, F. Hellwig, *System Performance Optimization Methodology for Infineon's 32-Bit Automotive Microcontroller Architecture*, Design Automation and Test in Europe (DATE), pp 962-966, 2008.

[38] B. Fleming, *Microcontroller Units in Automobiles*, IEEE Vehicular Technology Magazine, Vol.6, pp 4-8, 2011.

Engin Ozatay received his B.S. degree in Mechanical Engineering in 2009 from Middle East Technical University (METU), Turkey and his M.S. degree also in Mechanical Engineering in 2011 from the Institute of Dynamic Systems and Control (IDSC) at Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, where he had been granted with the Excellence Scholarship & Opportunity Award.

He worked as an intern in 2007 at Ford-Otosan, Turkey, in 2008 at AVL Powertrain, United Kingdom, in 2009 at Roketsan, Turkey, in 2010-2011 at Hilti, Liechtenstein and in 2013 at Ford Motor Company, Dearborn, USA. Since 2011, he has been appointed as a Graduate Research Assistant and conducting his PhD studies with the Department of Electrical and Computer Engineering, Ohio State University. His research interests include constraint optimization, model predictive control, robust control, estimation, powertrain control systems, intelligent transportation systems and autonomous vehicles.

Simona Onori received her Laurea Degree, summa cum laude, (Computer Eng.) in 2003, her M.S. (Electrical and Computer Eng.) in 2004, her Ph.D. (Control Eng.) in 2007, from University of Rome Tor Vergata, University of New Mexico, Albuquerque, USA, and University of Rome Tor Vergata, respectively. Prior to joining the Department of Automotive Engineering, Clemson University in August 2013, she was a research scientist at the Center for Automotive Research the Ohio State University (CAR-OSU) and lecturer at the Mechanical and Aerospace Engineering Dept. at Ohio State University where she taught at undergraduate and graduate level. 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 drivetrains, energy storage systems, and fuel cell systems with 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, among others, by Ford, General Motors, Cummins, the National Science Foundation and the US Department of Energy. She is an IEEE, ASME and SAE member. She is an ASME-Associate Editor for the American Control Conference and Dynamic System Control Conference, and Associate Editor for SAE International Journal of Advanced Powertrains. She is the recipient of the 2012 Lumley Interdisciplinary Research Award by OSU College of Engineering and TechColumbus 2011 Outstanding Technology Team Award.

James Wollaeger (IEEE member since 2006) was born in Cleveland, OH USA. He attended Ohio Northern University for a Bachelor of Science degree in Electrical Engineering(2010). His study of Electrical Engineering continued at The Ohio State University for his Masters of Science degree(2012).

He worked as a summer intern for Parker Hannifin Corporation in Mayfield Heights, Ohio where he learned how to work with embedded control system software development tools(2007-2009). After earning his Masters degree, he moved on to his current position at Robert Bosch LLC in Plymouth, MI USA. He is currently a Systems Engineer in the Chassis Systems Control division working on vehicle dynamics modeling.

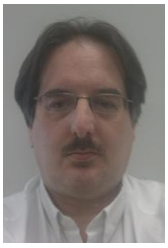
Umit Ozguner received his Ph.D. from the University of Illinois and held positions at I.B.M. Research Labs, University of Toronto and Istanbul Technical University. He has been with the Ohio State University since 1981 where he is a Professor of Electrical and Computer Engineering and a Senior Fellow of the Center for Automotive Research. He holds the TRC Inc. Chair on Intelligent Transportation Systems (ITS). Professor Ozguner is a Fellow of the Institute of Electrical and Electronic Engineers.

His areas of research interest are in ITS, decentralized control and autonomy in large systems and is the author or co-author of over 400 publications including a 2011 book on Autonomous Ground Vehicles. He has advised over 40 students in their MS and over 25 students in their PhD research.

He was the first President of the IEEE ITS Council in 1999 as it transformed into The IEEE ITS Society. He has also been the ITS Society VP for Conferences. Professor Ozguner has also served the IEEE Control Society in various positions. He participated in the organization of many conferences and been 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 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 TARDEC. Prof. Ozguners research has been supported by many industrial companies including Ford, GM, Honda and ASELSAN; and organizations like NSF, AFOSR, NASA and AFWAL.

Giorgio Rizzoni received his BS, MS and PhD in Electrical and Computer Engineering in 1980, 1982 and 1986, from the University of Michigan. Between 1986 and 1990 he was a post-doctoral fellow, and then a lecturer and assistant research scientist at the University of Michigan. In 1990 he joined the Department of Mechanical Engineering at Ohio State 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. Since 2002 he has been the Ford Motor Company Chair in Electromechanical Systems, and a Professor in the Department of Electrical and Computer Engineering. He also holds a courtesy appointment in the Department of Design. Prof. Rizzonis specialization is in dynamic systems and control, and his research activities are related to sustainable mobility. He is a Fellow of the Institute of Electrical and Electronic Engineers and Fellow of the Society of Automotive Engineers, and has 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 received his PhD. degree in Electrical Engineering from the Czech Technical University in Prague in 1979. Dr. Dimitar Filev is a Senior Technical Leader - Intelligent Control & Information Systems, Ford Research & Advanced Engineering. His research interests are in modeling and control of complex systems, intelligent control, fuzzy and neural systems, and their applications to automotive engineering. He is the recipient of the 2008 Norbert Wiener Award of the IEEE SMC Society and the 2007 IFSA Outstanding Industrial Applications Award. Dr. Filev is a Fellow of IEEE and IFSA.



John Michelini graduated from University of Detroit (BSME) and holds graduate degrees from University of Michigan (MSME) and Wayne State University (Engineering Management). He has worked at Powertrain R&A for over 10 years and Ford Motor Company for over 20 years. Mr. Michelini has worked on powertrain controls for various engine fuel economy technologies, Ti-VCT, and camless (EVA). He holds over 65 US patents and has co-authored 4 conference papers.

Stefano Di Cairano (M'08) received the Master (Laurea) in Computer Engineering in 2004, and the PhD in Information Engineering in 2008, both from the University of Siena, Italy. He was also granted the Int. Curr. Opt. of Doctoral Studies in Hybrid Control for Complex Distributed and Heterogeneous Embedded Systems. He was visiting student at the Technical University of Denmark, Lyngby, Denmark, in 2002–2003, and at the California Institute of Technology, Pasadena, CA, in 2006–2007. In 2008–2011, he was Senior Researcher and a Technical Expert with Powertrain Control R&A, Ford Research and Adv. Engineering, Dearborn, MI. Since 2011 he is a Principal Member of the Research Staff in Mechatronics, at the Mitsubishi Electric Research Labs, Cambridge, MA. His research is on advanced control strategies for complex mechatronic systems, in automotive, factory automation, and aerospace. His interests include model predictive control, constrained control, networked control systems, hybrid systems, optimization, automotive, aerospace, and factory automation. Since 2011, Dr. Di Cairano is Chair of the IEEE CSS Technical Committee on Automotive Controls, and member of the IEEE CSS Conference Editorial Board, and since 2013 he is an Associate Editor of IEEE Trans. Control Systems Technology.