Real-time Ecological Velocity Planning for Plug-in Hybrid Vehicles with Partial Communication to Traffic Lights

Sangjae Bae[1], Yongkeun Choi[2], Yeojun Kim[2], Jacopo Guanetti[2], Francesco Borrelli[2], and Scott Moura[1]

***Abstract*— This paper presents the design of an ecological adaptive cruise controller (ECO-ACC) for a plug-in hybrid vehicle (PHEV) which exploits automated driving and connectivity. Most existing papers for ECO-ACC focus on a short-sighted control scheme. A two-level control framework for long-sighted ECO-ACC was only recently introduced [1]. However, that work is based on a deterministic traffic signal phase and timing (SPaT) over the entire route. In practice, connectivity with traffic lights may be limited by communication range, e.g. just one upcoming traffic light. We propose a two-level receding-horizon control framework for long-sighted ECO-ACC that exploits deterministic SPaT for the upcoming traffic light, and utilizes historical SPaT for other traffic lights within a receding control horizon. We also incorporate a powertrain control mechanism to enhance PHEV energy prediction accuracy. Hardware-in-the-loop simulation results validate the energy savings of the receding-horizon control framework in various traffic scenarios.**

I. Introduction

Advanced Driving Assistance Systems (ADAS) are driving automation technologies that seek to improve driver comport and safety. Adaptive Cruise Control (ACC), autonomous emergency braking, and lane keeping assistance are examples of widely deployed functions in today's vehicles with so-called Level 2 automation [2]. Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) connectivity further advance innovative ADAS technologies, particularly including energy-efficient driving [3].

To reduce energy consumption, many studies have focused on longitudinal control and proposed ecological ACC designs within short, immediate surroundings [4], [5]. In that regard, given a fixed route, finding an optimal velocity trajectory, or "Eco-driving", has been studied from the perspective of optimal control [3], [6], [7]. In the presence of signalized intersections, eco-driving yields significant energy savings by utilizing SPaT information from traffic lights [1], [3], [8].

Our previous work [8] focused on the Eco-driving problem through signalized intersections with *uncertain* effective red light duration. The uncertainty is addressed by formulating chance constraints on passing through the intersections during green lights. Simulations showed potential fuel savings of up to 40%, compared to a modified intelligent driver model [9]. That said, the proposed algorithm did not consider surrounding traffic in non-free flow conditions. Also, like most literature on safe and ecological driving, the algorithm was not validated through experiments in real-word traffic conditions. Therefore, in our recent paper [1], we extended our previous work to incorporate ACC into the Eco-driving controller. This work balances energy efficiency with collision avoidance and traffic signal compliance. The combined Eco-driving and ACC controller is called Ecological Adaptive Cruise Controller (ECO-ACC). Vehicle-in-the-loop experiments were performed based on a recently introduced test setup for Connected and Automated Vehicles (CAVs) in real-world traffic [1], [10].

While there is an extensive literature on ECO-ACC, the integration of (*short-sighted*, but aware of immediate traffic changes) ACC and (often *long-sighted*, but only aware of slowly changing traffic information) Eco-driving mostly ends up being a *short-sighted* energy efficient ACC. Therefore, we proposed a *long-sighted* energy efficient ACC in a two-level control framework [1] that enables both long-term velocity planning and short-term collision avoidance. However, we assumed persistent connectivity with all traffic signals. This allows the ECO-ACC to receive deterministic SPaT information, and an optimal policy can be computed offline. This is hard to implement in practice, due to limited communication range between traffic lights and vehicles. Therefore, one requires an *online* algorithm to recursively update the *long-sighted* velocity trajectory as SPaT information becomes available from approaching intersections.

Since our goal is energy consumption reduction, we are keenly interested in the powertrain dynamics in addition to longitudinal vehicle dynamics control. In that regard, most existing literature on Eco-driving incorporates a simple powertrain model into its velocity planning [7], [8], [11]. These existing methods focus on either electric vehicles or gasoline vehicles. Plug-in hybrid vehicles present an additional challenge, since there are two power generating sources which introduces an additional degree of freedom. In particular, it is challenging to incorporate a PHEV powertrain model into the Eco-driving optimization problem since it increases the state space and control input space size [12]. To alleviate this issue, an appropriate method needs to be designed to incorporate PHEV powertrain dynamics into the Eco-driving optimization problem, while still retaining sufficient computational simplicity to enable online computations.

The main contributions of this paper address two practical problems. (i) We propose a two-level (velocity planning and safety control) receding horizon control framework that systematically balances energy consumption, travel time, and safety, given limited traffic signal information. (ii) We incorporate a PHEV powertrain model into the velocity

[1]Department of Civil and Environmental Engineering, University of California, Berkeley, USA. Corresponding author: sangjae.bae@berkeley.edu

[2]Department of Mechanical Engineering, University of California, Berkeley, USA.

planning layer that accurately captures fuel and electricity use for a given powertrain controller, without increasing the state or control input space. This ensures the optimal control problem can be solved in *realtime*.

The remainder of this paper is organized in the following manner. Section II details the two-level control framework and mathematical formulations of ECO-ACC. Section III presents simulation results and discuss limitations of the proposed framework and future works. Section IV concludes the paper with a summary.

II. Receding Horizon Eco-driving Controller Design

We proposed a two-level control framework of ECO-ACC in our previous paper [1]. The control architecture is depicted in Fig. 1. Eco-driving control, which computes a reference velocity trajectory across a long horizon in space, seeks to minimize energy based on both the probabilistic real-time SPaT and its empirical statistics. Adaptive Cruise Control (ACC), which computes the wheel torque to follow the reference velocity, guarantees safety (i.e., collision avoidance and traffic signal compliance) against uncertain road traffic.

In this work we focus on the development of a new Eco-driving control approach, while using the same ACC approach from our previous work [1]. Readers are encouraged to read [1] for details about our ACC design. The novelty of the new proposed Eco-driving control is two-fold: (i) It is executed in a receding horizon control framework with approximated terminal cost and the real-time traffic information. (ii) A PHEV powertrain model is incorporated without additional computational costs.

A. Vehicle Dynamics and Power Train Model

In our previous paper [1], we only considered longitudinal vehicle dynamics when planning the velocity profile. Consequently, the velocity trajectory is optimized to minimize wheel energy. Although wheel energy is a proxy of fuel and battery energy at a vehicle dynamics level, it does not capture the powertrain dynamics nor inefficiencies. Therefore, in this paper, we include a powertrain model in both the planning algorithm and HIL simulations to more accurately predict and minimize fuel and battery energy. In this section, we describe the powertrain dynamics as well as vehicle dynamics models used in this work.

1) Vehicle Dynamics: Consider the longitudinal vehicle dynamics where the longitudinal acceleration at step k, $a(k)$ is expressed as

$$a(k) = \frac{T_w(k)}{m} - g\left(cos(\theta(k))C_r - sin(\theta(k))\right) - \frac{\rho A C_d}{2m} v(k)^2, \tag{1}$$

where the input is a wheel torque $T_w(k)$ and the model parameters are vehicle mass m, product of the gearbox and final drive ratio r_{gb}, shaft torque before the gearbox T_{sft}, the wheel rolling radius R_w, road grade θ, air density ρ, the front cross-sectional area A, rolling resistance coefficient C_r, and air drag coefficient C_d. With the velocity $v(k)$ and travel time $t(k)$ as states at position $k\Delta s$, the system dynamics are

$$\underbrace{\begin{bmatrix} v(k+1) \\ t(k+1) \end{bmatrix}}_{x(k+1)} = \underbrace{\begin{bmatrix} v(k) \\ t(k) \end{bmatrix}}_{x(k)} + \begin{bmatrix} \frac{a(k)\Delta s}{v(k)} \\ \frac{\Delta s}{v(k)+\frac{a(k)\Delta s}{v(k)}} \end{bmatrix} \tag{2}$$

for $k \in \{0, \cdots, N-1\}$, with the position step size Δs[1]. We denote $v(k+1) = f_v(k)$ and $t(k+1) = f_t(k)$ for convenience.

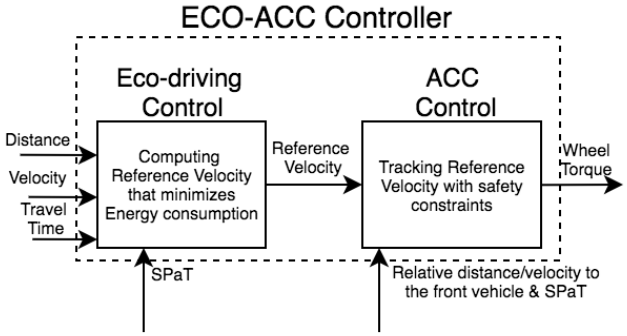

Fig. 1. The conceptual diagram of ECO-ACC controller which is composed of the Eco-driving controller and the ACC Controller in separate layers

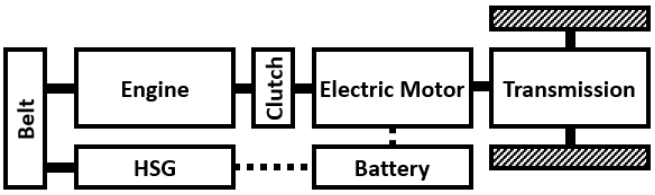

Fig. 2. PHEV powertrain architecture

2) Power Train Model: The powertrain architecture is a pre-transmission parallel hybrid as shown in Figure 2. The input wheel torque $T_w(k)$ in the longitudinal vehicle dynamics can be expressed as

$$T_w(k) = \frac{r_{gb}(v) T_{\text{sft}} - T_{\text{brk}}}{R_w}, \tag{3}$$

where

$$T_{\text{sft}}(k) = T_m(k) + e_{\text{on}}(k)\eta_c T_e(k) \tag{4}$$

where $T_m(k)$ denotes the torque produced by the electric motor at step k, $T_e(k)$ is the torque produced by the internal combustion engine at step k, $e_{on} \in \{0, 1\}$ is the engine on/off status, and η_c is clutch efficiency.

The electric motor power P_m can be represented as

$$P_m = \frac{T_m \omega_m}{\eta_m(T_m, \omega_m)} \tag{5}$$

where ω_m is an electric motor speed and η_m is an electric motor efficiency, which is a nonlinear function of motor torque T_m and electric motor speed ω_m. The electric motor speed can be computed by $\omega_m = \frac{v}{R_w r_{gb}}$. The hybrid starter

[1]Throughout the paper, the position step size is 1 meter, i.e., $\Delta s = 1$.

generator (HSG) power P_{HSG} and fuel power P_f are also computed in the same way as the electric motor power, i.e.,

$$P_{\text{HSG}} = \frac{T_{\text{HSG}}\omega_{\text{HSG}}}{\eta_{\text{HSG}}(T_{\text{HSG}}, \omega_{\text{HSG}})}, P_f = \frac{T_f \omega_f}{\eta_f(T_f, \omega_f)}. \quad (6)$$

Finally, the battery state-of-charge (SOC) dynamics model can be expressed as

$$\dot{SOC} = -\frac{V_{\text{oc}} - \sqrt{V_{\text{oc}}^2 - 4R_b P_b}}{2R_b Q_b} \quad (7)$$

where P_b is a terminal battery power, which is written

$$P_b = P_m + P_{\text{HSG}} + P_{\text{aux}}, \quad (8)$$

P_{aux} is an auxiliary power, V_{oc} is an open-circuit voltage, R_b is an internal resistance, and Q_b is the battery pack capacity. Details on PHEV powertrain models can be found in [13].

B. Cost function for Optimization

Similar to [1], the objective is to minimize a convex combination of energy consumption and travel time. However, the differences are the following. First, the control horizon is *limited and receding* and optimal solutions are found *in realtime*. Second, the energy consumption is evaluated based on the powertrain model. Third, SPaT is *uncertain*, except for the upcoming traffic light, which is detailed in the succeeding section.

1) PHEV Powertrain Cost Function: We use a cost function that is the total power cost from the battery and the liquid fuel, mathematically written as

$$g(v(k), T_w(k), T_m(k), SOC(k)) = P_f(v(k), T_w(k) - T_m(k)) + s \cdot P_{\text{elec}}(v(k), T_m(k), SOC(k)) \quad (9)$$

where s is a tuning parameter, and P_{elec} is an electrochemical battery power which is computed as $P_{\text{elec}} = V_{\text{oc}} \cdot I_b$. Note that the tuning parameter compromises the electric power cost with the fuel power cost.

In our simulation studies (details in Section III-A), we assume that the classic ECMS controller [14], [15] represents the production powertrain controller. Therefore, we need to estimate an output power cost of the ECMS controller in our Eco-driving controller. We further assume that the ECMS parameter is given and fixed at each SOC level, i.e., the tuning parameter s in (9) is deterministic.

Note that to evaluate the powertrain cost, we additionally need the battery SOC and motor torque, as well as the vehicle dynamics (2). Increasing problem dimensions, however, is challenging as we adopt Dynamic Progamming (DP) to run the Eco-driving controller *online*. To remedy the curse of dimensionality, two approximations are made. First, we treat SOC as a fixed parameter over the receding horizon. Our reasoning is the following: (i) the SOC does not change significantly over a short distance (like our receding horizon); and (ii) the optimization finds new solutions every few seconds, during which the SOC is reset to its measured value. Second, we approximate powertrain dynamics by a static relation and pre-optimize cost function (9) for all possible SOC grids with the grid step size of 0.01 using the tuning parameter, s. More specifically, we minimize g from all possible T_w and T_m combinations. As a result, the empirical model, g^* that maps (v, T_w, SOC) to a single numeric cost value is obtained.

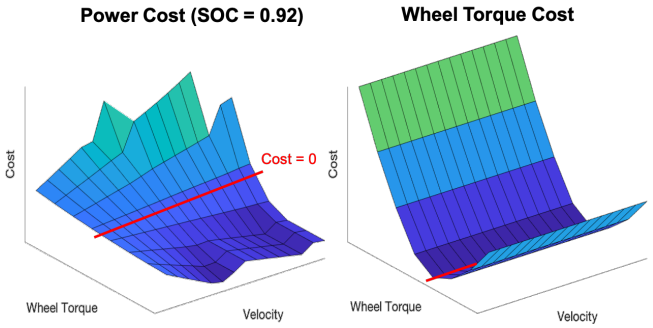

Fig. 3. Power map surface. The vertical and horizontal scales of both plots have been omitted for confidentiality reasons.

Figure 3 shows an example of the obtained empirical cost map (on the left) for the SOC level, 0.92. There are two highlights in the power cost map. (i) Unlike the wheel torque cost map, the power cost map can be negative because of regenerative braking. (ii) Certain wheel torque and velocity combinations result in particularly large power costs. This makes sense because turning the engine on is expensive in terms of energy when the SOC level is high (i.e., when the PHEV is in charge-depleting mode).

To summarize, at the current location d (distance from the origin) and using the empirical powertrain cost map g^*, the objective function over a receding horizon d_H is given by

$$J_{[d,d_H]} = \sum_{k=d}^{d+d_H} \underbrace{\left(g^*(v(k), T_w(k); SOC(d)) + \lambda \left(\frac{\Delta s}{v(k)} \right) \right)}_{=h(k)}, \quad (10)$$

with the weight λ. The instantaneous cost at each distance step k is denoted by $h(k)$ for convenience. The control variable is the wheel torque $T_w(k)$[2]. Note that the second term in $h(k)$ indicates the travel time cost. Also, now the

2) Approximate Terminal Cost: The objective function of the cost-minimizing problem from the current location d to destination d_f can be written

$$J_{[d,d_f]} = \sum_{i=d}^{d+d_H} h(i) + \sum_{j=d+d_H+1}^{d_f} h(j), \quad (11)$$

where $h(i)$ denotes an instantaneous cost function at space step i in (10). Note that if the information of traffic signal schedules are deterministic and known from the current location to the destination, the controller finds globally optimal solutions. The receding horizon controller can only evaluate the first term in (11). That is, the receding horizon controller will be myopic without a terminal cost that captures cost

[2]Note that we do not consider the allocation problem of engine and motor torque. [14], [15]

from the end of the horizon to the destination. That being said, in our problem the traffic signal phase and timing is uncertain and dynamic. There, is no "reference" SPaT that represents all traffic schedule and flow scenarios. We therefore approximate the expected cost ahead the current receding horizon, i.e., $\sum_{j=d+d_H+1}^{d_f} \mathbb{E}[h(j)]$, as a sample mean over different scenarios.

The objective function is then

$$\min \sum_{i=d}^{d+d_H} h(i) + J_{[d_H,d_f]}(x(d+d_H+1)), \tag{12}$$

where $J_{[d_H,d_f]}(\cdot)$ is the expected cost from space step d_H to final destination d_f. Specifically, $J_{[d_H,d_f]} = \min_{T_w} \sum_{j=d+d_H+1}^{d_f} \mathbb{E}[h(j)]$. Here, the expectation is taken w.r.t. uncertain SPaT. We approximate $J_{[d_H,d_f]}$ using a sample mean over scenarios generated from Monte Carlo simulations of SPaT information, generated offline. Mathematically

$$\hat{J}_{[d_H,d_f]} = \frac{1}{M} \sum_{j=1}^{M} \min \sum_{i=d}^{d+d_H} h(i;\sigma_j), \tag{13}$$

where σ_j represents a randomly generated SPaT scenario. This enables us to construct a terminal cost function that accounts for cost-to-go beyond the control horizon, while accounting for uncertain SPaT.

3) Soft terminal constraint: In addition to the approximated terminal cost, we leverage soft constraints to penalize deviations from a reference travel time. The soft terminal constraint in each receding horizon d_H is written

$$\underbrace{\left[t_f^D - (t+\tau_H)\right]}_{\text{Remaining time}} \hat{v}_{\text{avg}} \geq \underbrace{\left[d_f - (d+d_H)\right]}_{\text{Remaining distance}} - \gamma \tag{14}$$

where τ_H is a total travel time within the spatial horizon, t_f^D is the desired arrival time, $\hat{v}_{\text{avg}}$ is an empirical average speed over the remaining route, and γ is a slack variable. Note that we use the slack variable γ to relieve the constraint because each traffic scenario is randomly generated and therefore a hard constraint can result in infeasible solutions. The slack variable is evaluated in the terminal cost as

$$\gamma = \left[d_f - (d+d_H)\right] - \left[t_f^D - (t+\tau_H)\right]\hat{v}_{\text{avg}}. \tag{15}$$

Without this substitution, we require an additional control variable for the slack variable, which correspondingly increases the problem dimensions.

C. Constraints Setup

The constraints are set to

$$T_w^{\min} \leq T_w(k) \leq T_w^{\max}, \tag{16}$$

$$a^{\min} \leq a(k) \leq a^{\max}, \tag{17}$$

$$v^{\min} \leq v(k) \leq v^{\max}(k), \tag{18}$$

$$t^{\min}(k) \leq t(k) \leq t^{\max}(k), \tag{19}$$

for all $k \in \{d, \cdots, d+d_H\}$, where the inequality constraints (16), (17), and (18) ensure the wheel torque, acceleration, and velocity, respectively, are lower and upper bounded by appropriate values, which are given and known. Particularly, the wheel torque T_w is lower-bounded by the maximum braking torque which is the sum of the maximum mechanical friction and regenerative braking torques. Both the maximum wheel torque and regenerative braking torque are governed by characteristics of the electric motor, and their values depend upon the shaft speed, i.e., the vehicle speed. The maximum acceleration $a^{\max}$ is set to a physically feasible limit, and the maximum velocity $v^{\max}$ is set to the maximum speed limit on the road. The inequality constraint (19) ensures that the travel time is bounded by minimum and maximum travel time boundaries ($t^{\min}(k)$ and $t^{\max}(k)$, respectively).

1) Dynamic constraints for traffic lights: We utilize the SPaT information in the form of constraints. We assume that we know the current signal phase and timing of the first upcoming traffic light. We only know historical SPaT for the other traffic lights. This assumption takes into account of the limited range of V2I communication, and also reduces communication requirements for practical purposes.

Given SPaT information, we evaluate "infeasible" cases of driving behavior. The main intuition of the infeasible cases at each intersection is that the vehicle cannot pass through (or, "infeasible" to pass through) the intersections during the red light phase (i) in the current cycle; and (ii) in the next cycles. When the current signal phase is yellow, the controller is set to be conservative so that the car does not pass through the intersection, for safety reason.

Mathematically, we find the "infeasible" set (denoted by **IS**) of states $x(k) = [v(k), t(k)]^\top$, for distance step $k \in \{d, ..., d+d_H\}$, that satisfy:

$$\begin{cases} (f_t(k) \leq s_t) \cup \{(f_t(k) > s_t) \\ \cap \left(\mathbf{R}\left(f_t(k) - s_t, \ell_c^{(n)}\right) \geq \ell_c^{(n)} - \hat{\ell}_r^{(n)}\right)\} & \text{if } s_p = \text{red} \\ \hline (f_t(k) > s_t) \\ \cap(\mathbf{R}(f_t(k) - s_t, \ell_c^{(n)}) \leq \hat{\ell}_r^{(n)}) & \text{if } s_p = \text{green} \\ \hline (f_t(k) \leq s_t) \cup \{(f_t(k) > s_t) \\ \cap(\mathbf{R}(f_t(k) - s_t, \ell_c^{(n)}) \leq \hat{\ell}_r^{(n)})\} & \text{if } s_p = \text{yellow} \end{cases}$$

if an intersection is located at distance step $k+1$ and it is the first upcoming intersection from the current location d, and

$$\mathbf{R}(f_t(k) + \ell_{c,O}^{(n)}, \ell_c^{(n)}) \leq \hat{\ell}_r^{(n)}, \tag{20}$$

if an intersection is located at distance step $k+1$ and it is not the first upcoming intersection, where s_p is a current signal phase, s_t is a remaining time of the signal phase, $\ell_c^{(n)}$ is a signal cycle length at intersection n, $\hat{\ell}_r^{(n)}$ is an estimated red light duration, $\ell_{c,O}^{(n)}$ is a time shift of the signal cycle initiation, and $\mathbf{R}(\cdot)$ is the modulo operator. Remind that $f_t(k)$ is the travel time at distance step $k+1$ given the states and input pair $(x(k), T_w(k))$ at distance step k.

The inequalities with the modulo operator $\mathbf{R}(\cdot)$ indicate the following. At each intersection, given current states (velocity and travel time at distance step k), it is infeasible if a travel time at distance step $k+1$ is within a red light period of a traffic signal cycle. Take the inequality (20) as an

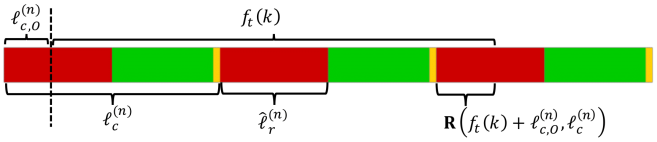

Fig. 4. Graphical demonstration of a cycle clock time $\mathbf{R}(f_t(k) + \ell_{c,O}^{(n)}, \ell_c^{(n)})$. Each colored block represents a traffic light, i.e., red, green, and yellow from left.

example. The left hand side represents the remainder of "the sum of cumulative travel time and shifted cycle initiation time" divided by "the signal cycle length". The remainder is equivalent to a clock time within the signal cycle, as depicted in Figure 4. If the cycle clock time is less than the estimated red light duration, i.e., less than $\hat{\ell}_r^{(n)}$, then the state and input pair $(x(k), T_w(k))$ is set to be infeasible. Similarly, we find infeasible states at the upcoming intersections with deterministic SPaT.

The estimated red light duration $\hat{\ell}_r^{(n)}$ is determined as a score at η^{th} percentile of the conditional probability density function (PDF) of red light durations. The optimal wheel torques at the infeasible states are forced to be minimum, i.e., maximum braking, to ensure the vehicle does not pass the intersection on red light.

D. Optimization Problem Formulation in Receding Horizon Control Scheme

The complete optimization problem is summarized as

$$\min_{T_w,v,t} \quad J_{[d,d_H]} + \hat{J}_{[d_H,d_f]} + \beta\gamma^2 \tag{21}$$

subject to

vehicle dynamics (2)
constraints (16)-(19)
feasible states $x; x \notin \mathbf{IS}$.

We apply DP to solve the above optimization problem, given the current states and SPaT information from the upcoming traffic light. Algorithm 1 demonstrates the process of computing the optimal policy map of wheel torques T_w^* at each iteration. The complete algorithm is the following. First, the Eco-driving controller receives the current information (states, SPaT). Second, the aforementioned optimization problem is solved via Algorithm 1, which takes a few seconds, and the optimal policy map T_w^* is updated with the recent policy map. It is important to note that while the optimization is being solved with the recently measured states and SPaT, the Eco-driving controller sends the ACC controller a velocity reference associated to the current states, i.e., $T_w^*(d, v, t)$, at every 0.2 seconds. The Eco-driving controller repeats solving the optimization problem until the vehicle arrives at the destination, i.e., $d \leq d_f$.

In Algorithm 1, d_I is a set of traffic light locations, $(n_v \times n_t)$ is the grid size of (v,t), the operator $(\cdot)_+$ takes positive elements in $(\cdot)$, and β is a weight of the slack variable γ defined in Section II-B.

Algorithm 1: Computing optimal wheel torques

Input : $d, t, d_I, d_H, \ell_c, \hat{\ell}_r, s_p, s_t$
Output: $T_w^* \in \mathrm{R}^{(d_H \times n_v \times n_t)}$
Init : Compute relative distance to traffic lights within the distance horizon
$\tilde{d}_I = (d_I - d)_+ \in [0, d_H]$

Set the terminal cost
$\hat{J}_{[d_H,d_f]} + \beta\gamma^2$

for $k = d_H - 1 \to 0$ **do**
  Solve Bellman's equation for all feasible states
  $\forall (v_i, t_j) \in \{(v(k), t(k)) \,|\, (v(k), t(k)) \notin \mathbf{IS}\}$,
  $V_k(v_i, t_j) = \min_{T_w}\{g(v_i, t_j) + V_{k+1}(f(v_i, t_j, T_w))\}$
  Get the minimizers
  $[T_w^*]_{i,j} \leftarrow$ minimizer of $V_k(v_i, t_j)$
end

III. Hardware-in-the-loop Simulation Result

A. Hardware-in-the-loop Simulation setup

The hardware-in-the-loop simulation (HILS) is identical to our previous work [1], except that the real PHEV is replaced by a mathematical model (this includes the ECMS controller, vehicle and powertrain dynamics). Therefore the HILS in this paper consists of a desktop (for traffic simulation and for mathematical vehicle model), Matrix embedded PC-Adlink (for Eco-driving control), and dSPace MicroAutoBox (for ACC control). Note that the ACC updates its torque control every 0.2 seconds to ensure the driver safety from immediate traffic changes. In contrast, the Eco-driving control updates its solution every 4 seconds to find a locally optimal velocity profile over a few hundred meters[3] ahead. While the Eco-driving controller is updating a new solution with recent traffic information, the optimal velocity at current distance and travel time is found from the most recent solution and is sent to the ACC at every recalculation of optimal torque control, i.e., 0.2 seconds. We consider the Live Oak corridor in Arcadia, CA, with a total of eight intersections, and SPaT identical to our previous paper [1] to which interested readers are referred for more details.

B. Simulation Result

In this section, we present simulation results to illustrate the effectiveness of the proposed control framework.

We first compare optimization of wheel energy versus equivalent fuel. Specifically, we investigate energy savings of the Eco-driving controller using the power cost map compared to the wheel torque map. In this case, we only compare the deterministic scenario of SPaT to evaluate ideal energy savings. Figure 5 illustrates velocity and SOC trajectories for the two controllers. Only SOC profiles are shown because the engine remained off in both cases. The controller using power cost map benefits from the negative

[3]One can adjust the receding control horizon for the Eco-driving controller to compromise the computation time with the optimality of solution.

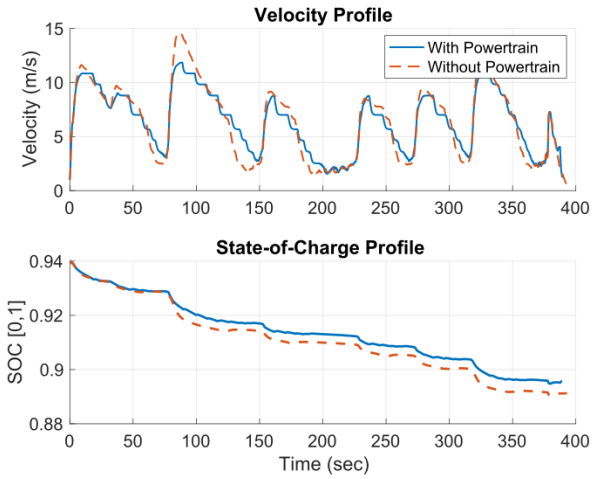

Fig. 5. Comparisons in Velocity and State-of-Charge of ECO-ACC with powertrain and without powertrain.

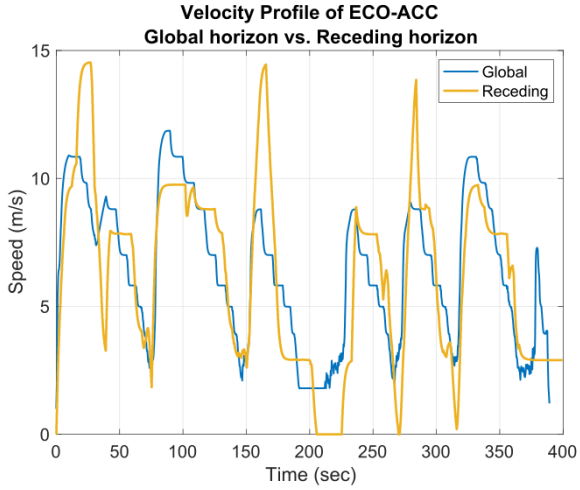

Fig. 6. Comparison in actual velocity profile

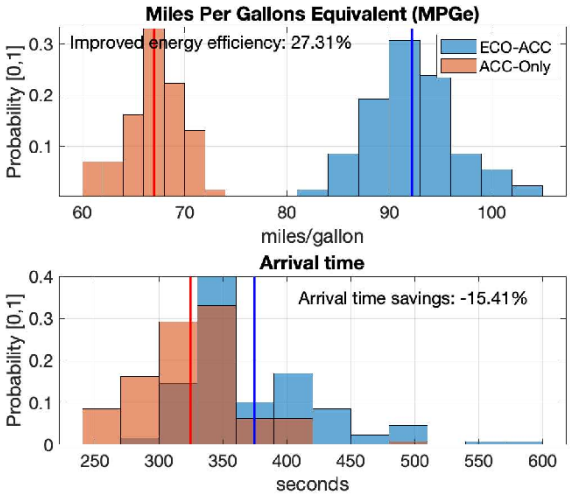

Fig. 7. The probability distributions of MPGe (top) and arrival times (bottom) obtained by Hardware-in-the-loop Simulations. The distributions are consist of a total of 130 traffic scenarios in each ECO-ACC and ACC-Only cases. We calculate that 33.7 kilowatt-hours of electricity is equivalent to one gallon of gas [16].

power cost, meaning it strategically uses regenerative braking to maximize recharging of the battery as shown in Fig. 5. As a result, the controller with the power cost map spends less battery energy, improving in energy performance by 9% in MPGe.

We also examine the energy consumption of the receding horizon ECO-ACC compared to that of the global horizon ECO-ACC. Note that the global horizon ECO-ACC has perfect information of the traffic signal schedules, and therefore DP finds a globally optimal velocity profile over the entire spatial horizon, i.e. from origin to destination, 2500 meters. In contrast, the receding horizon ECO-ACC finds a locally optimal velocity profile over the receding spatial horizon, which is practically set due to limited communication range with the intersections. We set the receding horizon to 400 meters so that at least one intersection is considered at each DP computation. Figure 6 shows the velocity profiles from the origin to destination for ECO-ACC with the global control and receding horizon. The velocity profile of the global horizon ECO-ACC illustrates that, in the ideal case, the vehicle does not have to speed up to its maximum and also it does not have to stop at the intersections in the middle of the route. Yet, it secures the same travel time. However, we found that the velocity profile of the receding horizon ECO-ACC is more volatile, approaching the maximum speed limit and zero speed. This is mainly because the receding horizon controller looks only a few hundred meters ahead in its velocity planning, and only the upcoming SPaT is deterministic. Consequently, the energy efficiency of the trip, measured by Miles-per-Gallon equivalent (MPGe), is 15% lower in the receding horizon control compared to the global horizon control. Despite this loss of optimality, this is already a significant improvement compared to ACC with a constant velocity profile. The details are presented later in this section.

Finally, we validate the energy savings of the receding horizon ECO-ACC in various traffic scenarios, based on Monte Carlo simulations. At each simulation, traffic schedules (e.g., red light duration, time shift of cycle initiation at each intersection) are randomly sampled from empirical PDFs conditioned to the hour of the day. To compute the conditional PDF we use a month of SPaT collected by Sensys Networks over the Live Oak route in Arcadia. In Fig. 7, it is clearly seen that ECO-ACC is significantly more energy efficient (27.31%) compared to ACC only, in any traffic scenarios at any hour of the day. That said, ECO-ACC results in longer travel time (15.41%) compared to ACC only, which is aligned with the intuition from our previous experimental results with the global horizon ECO-ACC in [1]. One can further investigate with different penalties on travel time in the objective function (10) to trade off energy consumption with travel time.

C. Limitations and Future Work

One limitation of the receding horizon control framework is that the Eco-driving controller can possibly update a DP solution *immediately before* passing the upcoming intersection. Consequently, the velocity planning will only rely on the historical SPaT at the next neighboring intersection, which might result in poor performance if the historical SPaT does not represent the actual SPaT well. If the estimated SPaT of the next intersection has a large offset with the actual SPaT, then ECO-ACC can unnecessarily waste its energy. For example, suppose that the optimal velocity plan is to keep a current velocity until the next intersection. With a poorly estimated SPaT, the vehicle might instead reduce its velocity until the DP is recalculated with the true SPaT information. Consequently, the vehicle needs to spend additional energy to catch up with the optimal velocity trajectory. These offsets are demonstrated as damping points around 100 or 255 seconds in Fig. 6. Therefore, a robust design for velocity trajectories remains for future work.

Another limitation is that we only considered a charge-depleting powertrain mode in the simulation (SOC level starts high). Due to this assumption, liquid fuel usage was minimized, and total energy consumption heavily depends on the battery usage. That being said, it is straightforward to extend the proposed framework to adopt charge sustaining mode by considering heterogeneous powertrain cost maps by SOC, which also remains for future work.

IV. Conclusion

This paper proposes a receding horizon control framework for an online Ecological Adaptive Cruise (ECO-ACC) control, with considerations for limited vehicle-to-infrastructure communication range and energy consumption behavior of Plug-in Hybrid Vehicles (PHEV). The overall objective is to minimize energy consumption while avoiding collisions and complying with traffic signals. The framework is based on a two-layer structure, where the upper layer corresponds to the velocity planning algorithm and the lower layer corresponds to collision avoidance and traffic signal compliance. This paper focuses on the velocity planning algorithm in the upper layer, which is adaptive to dynamically updated traffic signals within a receding control horizon. The receding control scheme is designed for hardware implementation and experimentation. Therefore several practical issues must be addressed, including efficient computations and limited traffic signal information *realtime*. Our control design is experimentally validated through a recently developed hardware-in-the-loop simulation.

Acknowledgement

The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number de-ar0000791. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

The authors would also like to thank Hyundai America Technical Center, Inc. for providing us with the vehicle and a testing facility, and Sensys Networks for granting access to the traffic data used in this paper.

References

[1] Sangjae Bae, Yeojun Kim, Jacopo Guanetti, Francesco Borrelli, and Scott Moura. Design and implementation of ecological adaptive cruise control for autonomous driving with communication to traffic lights. *arXiv preprint arXiv:1810.12442*, 2018.

[2] Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems, 2014.

[3] Jacopo Guanetti, Yeojun Kim, and Francesco Borrelli. Control of Connected and Automated Vehicles: State of the Art and Future Challenges. *Annual Reviews in Control*, 45:18–40, 2018.

[4] Roman Schmied, Harald Waschl, and Luigi Del Re. Extension and experimental validation of fuel efficient predictive adaptive cruise control. In *American Control Conference*, volume 2015-July, pages 4753–4758, 2015.

[5] Valerio Turri, Yeojun Kim, Jacopo Guanetti, Karl H Johansson, and Francesco Borrelli. A model predictive controller for non-cooperative eco-platooning. In *American Control Conference (ACC), 2017*, pages 2309–2314. IEEE, 2017.

[6] Antonio Sciarretta, Giovanni De Nunzio, and Luis Leon Ojeda. Optimal Ecodriving Control: Energy-Efficient Driving of Road Vehicles as an Optimal Control Problem. *IEEE Control Systems Magazine*, 35(5):71–90, 2015.

[7] Engin Ozatay, Simona Onori, James Wollaeger, Umit Ozguner, Giorgio Rizzoni, Dimitar Filev, John Michelini, and Stefano Di Cairano. Cloud-based velocity profile optimization for everyday driving: A dynamic-programming-based solution. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):2491–2505, 2014.

[8] Chao Sun, Jacopo Guanetti, Francesco Borrelli, and Scott Moura. Robust eco-driving control of autonomous vehicles connected to traffic lights. *arXiv preprint arXiv:1802.05815*, 2018.

[9] Arne Kesting, Martin Treiber, and Dirk Helbing. Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 368(1928):4585–4605, 2010.

[10] Yeojun Kim, Samuel Tay, Jacopo Guanetti, and Francesco Borrelli. Hardware-in-the-loop for connected automated vehicles testing in real traffic. In *14th International Symposium on Advanced Vehicle Control (to appear)*, 2018.

[11] DIB Wissam, Alexandre Chasse, Antonio Sciarretta, and Philippe Moulin. Optimal energy management compliant with online requirements for an electric vehicle in eco-driving applications. *IFAC Proceedings Volumes*, 45(30):334–340, 2012.

[12] Antonio Sciarretta and Lino Guzzella. Control of hybrid electric vehicles. *IEEE Control Systems Magazine*, 27(2):60–70, 2007.

[13] Lino Guzzella, Antonio Sciarretta, et al. *Vehicle propulsion systems*, volume 1. Springer, 2007.

[14] Stephanie Stockar, Vincenzo Marano, Marcello Canova, Giorgio Rizzoni, and Lino Guzzella. Energy-optimal control of plug-in hybrid electric vehicles for real-world driving cycles. *IEEE Transactions on Vehicular Technology*, 60(7):2949–2962, 2011.

[15] Cristian Musardo, Giorgio Rizzoni, Yann Guezennec, and Benedetto Staccia. A-ecms: An adaptive algorithm for hybrid electric vehicle energy management. *European Journal of Control*, 11(4-5):509–524, 2005.

[16] Peter D Friedman and Phil Grossweiler. An analysis of us federal mileage ratings for plug-in hybrid electric vehicles. *Energy Policy*, 74:697–702, 2014.