Di Chen
Department of Mechanical Engineering,
University of Michigan,
Ann Arbor, MI 48109
e-mail: dichencd@umich.edu

Mike Huang
Department of Mechanical Engineering,
University of Michigan,
Ann Arbor, MI 48109
e-mail: huxuli@umich.edu

Anna Stefanopoulou
Department of Mechanical Engineering,
University of Michigan,
Ann Arbor, MI 48109
e-mail: annastef@umich.edu

Youngki Kim
Department of Mechanical Engineering,
University of Michigan-Dearborn,
Dearborn, MI 48128
e-mail: youngki@umich.edu

A Receding-Horizon Framework for Co-Optimizing the Velocity and Power-Split of Automated Plug-In Hybrid Electric Vehicles

This paper presents a control framework to co-optimize the velocity and power-split operation of a plug-in hybrid vehicle (PHEV) online in the presence of traffic constraints. The principal challenge in its online implementation lies in the conflict between the long control horizon required for global optimality and limits in available computational power. To resolve the conflict between the length of horizon and its computation complexity, we propose a receding-horizon strategy where co-states are used to approximate the future cost, helping to shorten the prediction horizon. In particular, we update the co-state using a nominal trajectory and the temporal-difference (TD) error based on co-state dynamics. Our simulation results demonstrate a 12% fuel economy improvement over the sequential/layered control strategy for a given driving scenario. Moreover, its real-time practicality is evidenced by a computation time per model predictive controller (MPC) step on average of around 80 ms within a 10 s prediction horizon.

[DOI: 10.1115/1.4050191]

Keywords: automotive systems, iterative learning control, model predictive control, optimal controls, predictive control for nonlinear systems

1 Introduction

The technical maturity of advanced driving assistance systems (ADAS) encourages research into designing a longitudinal velocity controller systematically from an optimal control perspective. Advancements in ADAS, together with a powertrain-level controller, can achieve excellent overall system-wide efficiency. Among various types of vehicles with different energy sources, a plug-in hybrid vehicle (PHEV) is of particular interest. It is because, for a PHEV with two energy sources (fuel and electricity), energy-efficient driving can further benefit fuel economy beyond what can be achieved with an optimal power-split on ordinary human driving.

Over the years, there have been extensive efforts toward increasing the efficiency of (P)HEVs. The book of Sciarretta and Vahidi [1] presents comprehensive discussions that incorporate almost every aspect of connected and automated vehicles using a variety of energy sources. The work of Bae et al. [2] presents an ecological adaptive cruise controller (ECO-ACC) with a two-level framework for a PHEV to minimize energy consumption while avoiding collisions and complying with traffic signals. However, its focus was based solely on velocity planning and safety control rather than jointly optimizing vehicle-following and powertrain dynamics. The work of Heppeler et al. [3] introduced a layered approach for the predictive control of the vehicle and powertrain dynamics of an HEV and demonstrated a 3–10% fuel consumption reduction compared to the desired velocity obtained respecting speed limits followed by the equivalent consumption minimization strategy. However, their proposed controller did not account for car-following, i.e., it did not consider a lead vehicle (LV). Therefore, it is unclear how the controller would perform in terms of fuel economy and safety guarantee when traffic constraints induced by the LV need to be considered in the short-horizon but are ignored in the long-horizon SOC planning.

Most of the existing literature on energy-efficient control of a PHEV employs a layered control framework, including (1) a velocity planning (eco-driving) layer, sometimes with an additional ACC layer to guarantee safety in the presence of an LV, albeit the information of the powertrain dynamics can be partially included to better estimate the energy consumption [3] and (2) a powertrain-control layer to coordinate the operation between the engine and electric motors to minimize fuel and guarantee a desired battery state-of-charge (SOC) level at the end of the trip with the velocity determined by a velocity planning layer. Unlike the existing eco-driving approaches based on the layered control framework that indirectly aims to minimize fuel consumption, in this study, it is proposed to co-optimize the velocity and powertrain operations to explicitly achieve the maximum system-wide efficiency while guaranteeing the safety and the desired terminal SOC. Moreover, the proposed solution strategy is potentially implementable in real-time.

For PHEVs, the predominant roadblock to adopt a co-Optimization control framework online originates from its numerical implementation. To be more concrete, to optimize the battery charge-depletion rate and the velocity for a PHEV, one needs to solve a trajectory optimization problem (TOP) for the entire trip with a specified SOC to be satisfied at the end of the trip. In the online implementation, the TOP is posed as an economic model predictive controller (EMPC) problem. To the authors' knowledge, the work of Huang et al. [4] is the first that succeeded in solving the TOP directly in real-time rather than tracking a reference explicitly. However, the work in Ref. [4] only considered the powertrain dynamics with a single battery SOC. For the co-optimization problem, due to additional controls and states, it is unclear of their real-time implementability when the EMPC aims to cover the entire trip. Besides, as shown later in the paper, co-optimization requires predicting a LV's driving trace. Such a prediction will become less accurate as the prediction horizon increases.

The length of the prediction horizon is generally constrained to reduce the computational complexity associated with the TOP, which makes it difficult to achieve global optimality. To resolve this issue, we propose a receding-horizon strategy where the

Manuscript received August 23, 2020; final manuscript received February 9, 2021; published online March 11, 2021. Assoc. Editor: Loucas Louca.

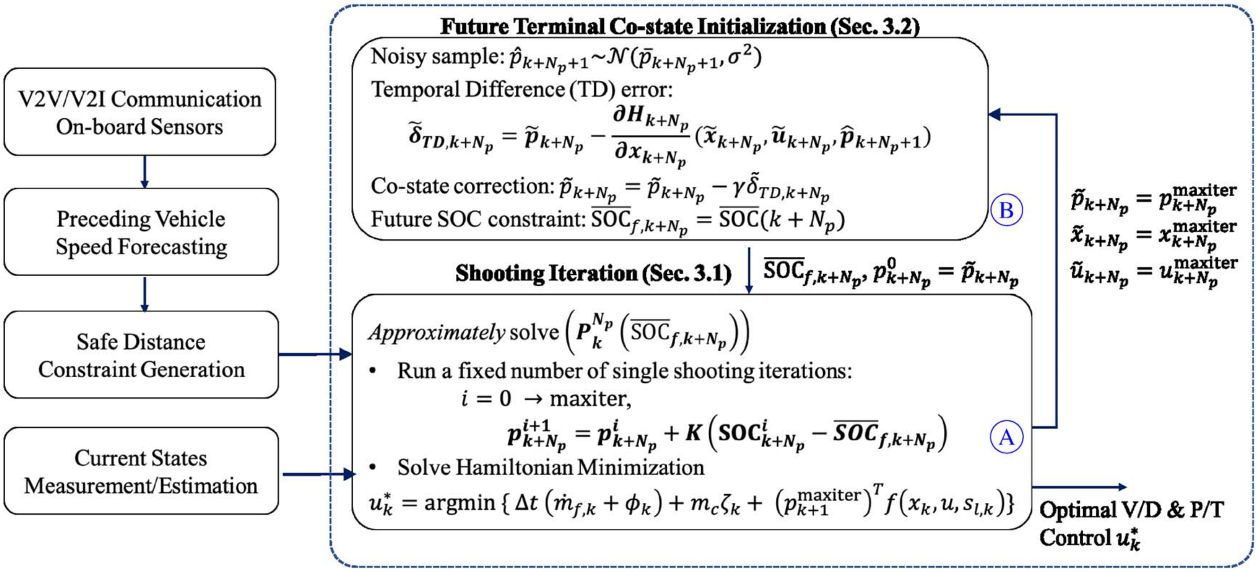
Future Terminal Co-state Initialization (Sec. 3.2)
Noisy sample: $\hat{p}_{k+N_p+1} \sim \mathcal{N}(\bar{p}_{k+N_p+1}, \sigma^2)$
Temporal Difference (TD) error:
$\tilde{\delta}_{TD,k+N_p} = \tilde{p}_{k+N_p} - \frac{\partial H_{k+N_p}}{\partial x_{k+N_p}}(\tilde{x}_{k+N_p}, \tilde{u}_{k+N_p}, \hat{p}_{k+N_p+1})$
Co-state correction: $\bar{p}_{k+N_p} = \tilde{p}_{k+N_p} - \gamma \tilde{\delta}_{TD,k+N_p}$
Future SOC constraint: $\overline{\mathrm{SOC}}_{f,k+N_p} = \overline{\mathrm{SOC}}(k+N_p)$
B
V2V/V2I Communication On-board Sensors
Preceding Vehicle Speed Forecasting
Safe Distance Constraint Generation
Current States Measurement/Estimation
$\tilde{p}_{k+N_p} = p^{\mathrm{maxiter}}_{k+N_p}$
$\tilde{x}_{k+N_p} = x^{\mathrm{maxiter}}_{k+N_p}$
$\tilde{u}_{k+N_p} = u^{\mathrm{maxiter}}_{k+N_p}$
Shooting Iteration (Sec. 3.1)
$\overline{\mathrm{SOC}}_{f,k+N_p}, p^0_{k+N_p} = \bar{p}_{k+N_p}$
Approximately solve $\left(\boldsymbol{P}_k^{N_p}\left(\overline{\mathrm{SOC}}_{f,k+N_p}\right)\right)$
• Run a fixed number of single shooting iterations:
$i = 0 \rightarrow$ maxiter,
$p^{i+1}_{k+N_p} = p^i_{k+N_p} + K\left(\mathrm{SOC}^i_{k+N_p} - \overline{SOC}_{f,k+N_p}\right)$
A
• Solve Hamiltonian Minimization
$u_k^* = \mathrm{argmin}\{\Delta t\left(\dot{m}_{f,k} + \phi_k\right) + m_c\zeta_k + \left(p^{\mathrm{maxiter}}_{k+1}\right)^T f(x_k, u, s_{l,k})\}$
Optimal V/D & P/T Control u_k^*

Fig. 1 A receding-horizon control strategy to co-optimize vehicle longitudinal and powertrain dynamics for energy-efficient driving. The conventional notation $t+k|t$ in MPC is replaced by k here for simplicity.

co-states are used to approximate the future cost. In particular, the co-state is updated using a nominal trajectory and the temporal-difference (TD) error based on co-state dynamics. The proposed receding-horizon control framework with co-state correction based on TD error is shown in Fig. 1, which will be detailed in Sec. 3. In the proposed strategy, distance constraints are generated from the prediction of the LV's trajectory. A reference SOC from a nominal trajectory obtained offline is used to softly update the co-state at the end of each prediction horizon, which is then rolled out backward-in-time to get the control at time $t=k$. This is the so-called shooting iteration block and will be discussed in Sec. 3.1. A TD error is used to correct the co-state value sampled from a noisy nominal trajectory to warm-start the single shooting iteration based on its co-state dynamics. This is the future terminal co-state initialization block and detailed in Sec. 3.2.

The remainder of the paper is organized in the following manner: Sec. 2 briefly summarizes the control-oriented model and the centralized control framework as an EMPC problem with related constraints. Then, an innovative strategy to solve the centralized control problem is detailed in Sec. 3. Section 4 presents the simulation results with their comparison against the offline sequential velocity smoothing and power-split optimization. Section 5 concludes the paper with a summary and future works.

2 Receding-Horizon Fuel-Efficient Controller Design

This section describes a control framework to co-optimize the velocity and powertrain operation of the PHEV online as an EMPC problem. A detailed discussion of the novel online implementation strategy based on a TD Bellman equation error will be presented in the next section.

2.1 Control-Oriented Model. Two subsystems are considered for co-optimizing the ego PHEV's velocity and powertrain operation to achieve minimum fuel consumption in the presence of a LV [5]: (i) the vehicle-following subsystem and (ii) the hybrid powertrain. Detailed descriptions of the dynamics and their corresponding constraints are omitted for space consideration. These two subsystems are connected through the vehicle (longitudinal) velocity v_k and the driver-demanded torque tp_k. In this work, the ego vehicle is assumed to drive on a single lane without grades and considers only its longitudinal dynamics.

To summarize, the overall system consisting of the vehicle-following and powertrain subsystems can be expressed with state-space representation as

$$x_{k+1} = f(x_k, u_k, w_k) \tag{1}$$

where the states are $x_k = [s_k, v_k, \mathrm{SOC}_k, e_k]^T$, denoting the position, velocity, battery SOC, and the normalized engine cranking state of the considered ego PHEV, respectively. The control inputs are $u_k = [r_k, ne_k, te_k, E_k]^T$, denoting the normalized distance gap regularizer in an ACC-type feedback controller, engine speed, engine torque, and engine on/off command, respectively. The external known disturbance $w_k = s_{l,k}$ is the position of the LV.

2.2 Economic Model Predictive Controller Formulation for Co-Optimization. The objective of the EMPC design for our problem is not to penalize deviation from a pre-defined equilibrium [6] but rather to directly minimize the total fuel consumption (the economic cost) of the PHEV on a given trip by simultaneously optimizing its velocity and its powertrain operation. Despite the lack of explicit reference tracking, the battery SOC is required to be depleted to a specified low level at the end of the entire trip. In this study, the vehicle is assumed to be in traffic and hence is subject to tight time-varying upper and lower bounds on its position depending on its LV.

In detail, the minimum fuel consumption problem is formulated directly in discrete-time in a receding-horizon manner, with the vehicle level constraints augmented to the stage cost via smooth exterior penalties, and this is denoted by $\mathbf{P}_k^{N_f}(\mathbf{SOC}_f)$ with N_f large enough to cover the whole trip.

$$\mathbf{P}_k^{N_f}(\mathbf{SOC}_f)\min J_k = \sum_{i=k}^{k+N_f}\{\Delta t(\dot{m}_{f,i|k} + \phi_{i|k}) + m_c \max(e_{i+1|k} - e_{i|k}, 0)\} \tag{2a}$$

where m_c is grams of fuel per cranking. Note that an absolute value can be used instead of the max-term in (2a). The absolute value would penalize both engine on and off events, whereas the max penalizes only the engine on event (consistent with the high-fidelity evaluation model).

The penalty term in (2a) is defined as

$$\begin{aligned}\phi_{i|k} = \gamma_1[&\max(s_{i|k} - s_{i|k}^{\max}, 0)^2 + \max(s_{i|k}^{\min} - s_{i|k}, 0)^2] \\ &+ \gamma_2[\max(a_{i|k} - a_{i|k}^{\max}, 0)^2 + \max(a_{i|k}^{\min} - a_{i|k}, 0)^2] \\ &+ \gamma_3[\max(v_{i|k} - v_{i|k}^{\max}, 0)^2 + \max(v_{i|k}^{\min} - v_{i|k}, 0)^2]\end{aligned} \tag{2b}$$

where γ_1, γ_2, and γ_3 are adjustable weights on the constraints violation. $s_{i|k}^{\max}$, $s_{i|k}^{\min}$, $v_{i|k}^{\max}$, $v_{i|k}^{\min}$, $a_{i|k}^{\max}$, and $a_{i|k}^{\min}$ are the maximum and minimum (possibly time-varying) position, velocity, and acceleration constraints, respectively (see Sec. 4). Moreover, the problem is subject to the state dynamics

$$x(i+1|k) = f(x(i|k), u(i|k), w(i|k)) \tag{2c}$$

the control constraints

$$r_{i|k} \in [-1, 1], \quad E_{i|k} \in \{0, 1\}, \quad [te_{i|k}, ne_{i|k}] \in [0, 0] \cup \Omega_k^{\mathrm{HV}} \tag{2d}$$

and the desired terminal SOC

$$\mathrm{SOC}(N_f + k|k) = \mathrm{SOC}_f \tag{2e}$$

As seen from $\mathbf{P}_k^{N_f}(\mathbf{SOC}_f)$, there is no explicit reference trajectory for tracking. When a proper SOC reference trajectory is not utilized, the entire trip needs to be considered to minimize the total fuel consumption through an SOC charge-blending strategy, which has been shown for the case of a human-driven PHEV in Ref. [4]. In Ref. [4], the fuel-efficient control is cast as a TOP for the entire trip and then solved in a receding-horizon manner. However, Ref. [4] considers only a single state of the battery SOC with two control inputs. Solving $\mathbf{P}_k^{N_f}(\mathbf{SOC}_f)$ directly is computationally demanding due to the increase in the problem's dimensions: four states and four control inputs. In the following section, an innovative approach is presented to incorporate the information of the entire trip in the original EMPC formulation $\mathbf{P}_k^{N_f}(\mathbf{SOC}_f)$ with a relatively short prediction horizon.

3 Approximation Strategy for Solving Economic Model Predictive Controller

In the original EMPC formulation ($\mathbf{P}_k^{N_f}(\mathrm{SOC}_f)$) presented in the previous section, the problem horizon needs to cover the entire trip. However, as mentioned in the previous section, this formulation could be too computationally demanding to be solved in real-time without a specialized numerical strategy like the one proposed in Ref. [4]. Inspired by the TD update rule [7] in reinforcement learning, an online implementation strategy is proposed where only a short prediction horizon N_p instead of the entire trip horizon N_f can be considered. In the proposed strategy, intermediate SOC and its associated co-state values from nominal trajectories obtained offline are used to warm-start the computation. Note that the nominal SOC trajectory is used only to softly update the co-state, and hence no explicit reference tracking formulation is needed in the proposed implementation strategy. Figure 1 presents the receding-horizon implementation strategy to *approximately* solve $\mathbf{P}_k^{N_f}(\mathbf{SOC}_f)$ by *approximately* solving $\mathbf{P}_k^{N_p}(\mathbf{SOC}_{f,k+N_p})$. Among all are two critical blocks: (1) shooting iteration Ⓐ and (2) final terminal co-state initialization Ⓑ, will be explained in detail in this section.

3.1 Shooting Iteration. The numerical algorithm in solving the associated mixed-integer nonlinear optimal control problem for the entire trip offline $\mathbf{P}_0^{N_f}(\mathbf{SOC}_f)$ with single shooting is a modified version of the algorithm proposed in Ref. [8]. Since the algorithm for solving $\mathbf{P}_0^{N_f}(\mathbf{SOC}_f)$ is not the focus of this work, in the sequel, $\mathbf{P}_0^{N_f}(\mathbf{SOC}_f)$ is considered solvable. Note that unlike the general single shooting based on the discretization of the continuous-time Pontryagin's minimum principle, single shooting is applied with co-state backward-in-time propagation. To be concrete, at time $t = k\Delta t$, it requires

(1) initial guesses of the state and control trajectories within the prediction horizon N_p, $X_{k+N_p}^0$ and $U_{k+N_p}^0$, where $X_{k+N_p}^0 = [x_k^0, \ldots, x_{k+N_p}^0]$ and $U_{k+N_p}^0 = [u_k^0, \ldots, u_{k+N_p}^0]$, with x_k and u_k defined in Sec. 2.1 and
(2) an initial guess of the terminal co-state associated with the SOC at the end of the prediction horizon, $p_{k+N_p}^0$

to warm-start the numerical computation. This is the starting point of Ⓐ in Fig. 1. Note that since no equality constraints are enforced for s_k, v_k, and e_k over the entire trip, their corresponding terminal co-states at the end of prediction horizon $k + N_p$ can always be set to 0 (no terminal cost is considered in this problem). On the other hand, the co-state associated with the SOC needs to be properly updated at the end of the prediction horizon because of the terminal SOC constraint at the end of the entire trip. Based on Bellman's principle of optimality, ideally, the SOC at the end of each prediction horizon should be optimal with respect to the remaining trip.

In reality, however, the exact trace of the entire trip of the ego PHEV's LV is not known in advance, making it impossible to accurately predict the exact SOC value at the end of each prediction horizon. Nevertheless, the traffic monitoring system, on-board GPS, and mobile apps enable the possibility of recording the velocity trace of the same route. The repeated traces can be averaged and used as the nominal LV's velocity trace to form the position constraints for the ego PHEV. Then, the minimum fuel consumption problem $\mathbf{P}_0^{N_f}(\mathbf{SOC}_f)$ can be (approximately) solved to obtain the nominal SOC and its corresponding co-state trajectories. As will be detailed next, the nominal SOC $\overline{\mathrm{SOC}}$ and its corresponding co-state $\bar{p}$ trajectory are used in Ⓐ and Ⓑ in Fig. 1. As is shown in Fig. 1 Ⓑ, at each time $t = k\Delta t$, from the nominal SOC trajectory, the nominal SOC value at the end of the prediction horizon $\overline{\mathrm{SOC}}_{f,k+N_p}$ is obtained; then, this value is used in Ⓐ to solve $\mathbf{P}_k^{N_p}(\overline{\mathbf{SOC}}_{f,k+N_p})$. However, due to the uncertainty in the future driving condition, strict enforcement of the terminal SOC at $\overline{\mathrm{SOC}}_{f,k+N_p}$ can result in infeasibility. To avoid this infeasibility issue, single shooting is performed only with a fixed number of iterations, and $\mathbf{P}_k^{N_p}(\overline{\mathbf{SOC}}_{f,k+N_p})$ is solved *approximately*.

3.2 Future Terminal Co-State Initialization. As mentioned previously, to solve $\mathbf{P}_k^{N_p}(\overline{\mathbf{SOC}}_{f,k+N_p})$ numerically, an initial guess of the state and control trajectories as well as the initial guess of the terminal co-state $p_{k+N_p}^0$ at the end of the prediction horizon are required. The initial guesses of the state and control trajectories are obtained from the shifted trajectories of the previous model predictive controller (MPC) step. The most critical part in the short-horizon implementation strategy is the proper initialization of the terminal co-state $p_{k+N_p}^0$. Although it is possible to warm-start the terminal co-state $p_{k+N_p}^0$ using the value $\bar{p}_{k+N_p}$ from the nominal co-state trajectory $\bar{p}$, it is observed that (1) this simple interpolation would induce a systematic bias to the terminal SOC at the end of the trip when errors exist in the position constraints within the prediction horizon (induced by the prediction error of the LV's position) and (2) if the terminal co-state from the previous MPC step is used to warm-start the next MPC step; large oscillations will be induced in the resulting co-state trajectory.

To alleviate the above issues, a correction step, inspired by the TD error in reinforcement learning, is added to the terminal co-state update. More specifically, the co-state dynamics require the following equation to be satisfied:

$$p_{k+N_p} = \frac{\partial H_{k+N_p}(x_{k+N_p}, u_{k+N_p}, p_{k+1+N_p})}{\partial x_{k+N_p}} \tag{3a}$$

namely, the TD error $\delta_{\mathrm{TD},k+N_p} = 0$, where

$$\delta_{\mathrm{TD},k+N_p} = p_{k+N_p} - \frac{\partial H_{k+N_p}(x_{k+N_p}, u_{k+N_p}, p_{k+N_p+1})}{\partial x_{k+N_p}} \tag{3b}$$

At time $t = k\Delta t$, by solving $\mathbf{P}_k^{N_p}(\overline{\mathbf{SOC}}_{f,k+N_p})$ approximately, the corresponding terminal co-state $\tilde{p}_{k+N_p}$, state $\tilde{x}_{k+N_p}$, and control $\tilde{u}_{k+N_p}$ can be obtained. Afterward, $\hat{p}_{k+N_p+1}$ is obtained as a noisy sampled version from the nominal co-state trajectory $\hat{p}_{k+N_p+1} \sim \mathcal{N}(\bar{p}_{k+N_p+1}, \sigma^2)$. Then, the TD error is calculated as

$$\tilde{\delta}_{\mathrm{TD},k+N_p} = \tilde{p}_{k+N_p} - \frac{\partial H_{k+N_p}(\tilde{x}_{k+N_p}, \tilde{u}_{k+N_p}, \hat{p}_{k+N_p+1})}{\partial x_{k+N_p}} \tag{4}$$

and the co-state is updated as

$$\tilde{p}_{k+N_p} = \tilde{p}_{k+N_p} - \gamma \tilde{\delta}_{\mathrm{TD},k+N_p} \tag{5}$$

with a learning rate of γ, which is then used to warm-start $\mathbf{P}_{k+1}^{N_p}(\overline{\mathbf{SOC}}_{f,k+N_p+1})$.

Remark 1. Since the battery SOC dynamics are slow, the dynamics of its corresponding co-state are also slow [4]. It means that the updated $\tilde{p}_{k+N_p}$ can be used as the warm-start of the next MPC iteration, $\tilde{p}^0_{k+N_p+1}$.

4 Simulation Results and Discussions

This section presents and discusses the results obtained from PHEV simulation with the proposed centralized EMPC-based strategy and the previously developed sequential optimization strategy. In simulation, the learning rate γ in (5) is set to be 0.1. The time-step is $\Delta t = 1s$, and the prediction step is $N_p = 10$, indicating the total prediction horizon is 10 s. To reduce the computation time, we replace the exact power-split optimization with a selection among several engine operations points close to the potential local minimizers identified through offline power-split optimization. The standard deviation considered in co-state sampling is set to be $\sigma = 10$. Given that the prediction of the LV's trajectory is not the focus of this work, to incorporate potential prediction errors, Gaussian white noise is added to the actual LV's acceleration with a signal-to-noise ratio = 10 and integrated to compute its trajectory in prediction. The LV is assumed to have the same trajectory as the one shown in the first subplot in Fig. 3(*a*), representing a trip of approximately 2 h long that exceeds the total battery range of the considered PHEV. It is a combination of standard driving cycles, including US06 (high acceleration aggressive driving), Urban Dynamometer Driving Schedule (UDDS, city driving), and the Highway Fuel Economy Driving Schedule (HWFET, highway driving). The trip we have considered is found through our offline simulations, among other combinations, to provide the most significant fuel economy by optimizing powertrain operation due to the high power demands at the beginning of the trip. To leverage a driver's ride comfort, a surrogate cost $\tilde{\phi}$ is used to replace ϕ in (2*a*), where

$$\tilde{\phi}_{i|k} = \phi_{i|k} + \rho \cdot a^2_{i|k} \tag{6}$$

In simulation, each MPC step with prediction horizon N_p is performed with 10 single shooting iterations (Ⓐ as shown in Fig. 1). The computation time per MPC step is around 80 ms on average.[1] For comparison, a single shooting iteration for the entire trip offline is 17.7 s for this particular 2-h trip.

Previously, we demonstrated the effectiveness of velocity smoothing followed by a power-split optimization (sequential optimization) in fuel economy improvement [9]. Minimizing the acceleration of the velocity profile induces a smoothed driving trace in line with some of the work in eco-driving [10,11]. However, the sequential type of optimization requires a layered implementation, where each layer has its own objective function. Consequently, such an implementation is in essence *decentralized*. By comparison, the direct fuel-efficient MPC strategy in this work adopts a *centralized* control framework. Since one of the objectives is to compare the performance of the centralized and decentralized control framework, in this work, the result obtained by offline sequential optimization is used as the baseline for comparison.

The time-domain response of the battery SOC and fuel consumption of the considered 2-h trip are shown in Fig. 2. The initial battery SOC is $\mathrm{SOC}_0 = 0.85$, and the desired terminal SOC is $\mathrm{SOC}_f = 0.14$. The first subplot presents the battery SOC trajectories. The second subplot presents the cumulative fuel consumption trajectories (including cranking fuel). The third subplot presents cumulative cranking fuel consumption trajectories. Note that the SOC at the end of the trip is not guaranteed to strictly match the desired SOC level in online implementation. For better comparison, we correct the total fuel consumption considering the deviation of the actual terminal SOC ($\mathrm{SOC}(N_f)$) from the desired terminal SOC (SOC_f) and present the fuel consumption results with and without SOC correction, which is computed by

$$fc_{\mathrm{cor}} = fc + 100 \cdot \Delta f(\mathrm{SOC}_f - \mathrm{SOC}(N_f)) \tag{7}$$

where fc denotes the actual (uncorrected) total fuel consumption from simulation, and fc_{cor} denotes the corrected total fuel consumption based on terminal SOC difference. Δf represents the difference in fuel consumption corresponding to a one-percent difference in SOC, which is chosen to be 16 g in this work based on simulation results. The results are summarized in Table 1. Detailed views of the resulting velocity and acceleration are illustrated in Fig. 3(*a*). The actual distance gaps between the ego PHEV and its LV corresponding to these time periods are shown in Fig. 3(*b*).

4.1 Position Constraints Under Uncertainty in Prediction.

When implementing the receding horizon, the minimum and maximum allowable distances, $s^{\min}_{i|k}$ and $s^{\max}_{i|k}$, are functions of the position of the LV and are defined as

$$s^{\min}_{i|k} = s_{l,i-3/\Delta t|k} - 4 \quad \text{and} \quad s^{\max}_{i|k} = s_{l,i-1/\Delta t|k} - 1 \tag{8}$$

in this work. Note that at $t = k$, the position of the LV $s_{l,j|k}, \forall j \le k, j \in \mathbb{Z}^+$, is known (despite that it might be subject to measurement noise). As a result, assuming accurate measurement, in (8) the position upper bound $s^{\max}_{i|k}$ is known exactly for $i = k, \ldots, k + 1/\Delta t$, and the position lower bound $s^{\min}_{i|k}$ is known for $i = k, \ldots, k + 3/\Delta t$. The position constraints otherwise depend on the prediction of that LV's trajectory. As discussed above, the design of the position constraints (8) guarantees their accuracy for the first several future time-steps, albeit the predicted LV's future trajectory is always inaccurate. As a result, as is illustrated in Fig. 3(*b*), in all the online co-optimization simulations, the ego vehicle is always able to keep a desired distance from its LV. Note that in simulation, some mild constraint violation can still be observed due to limited shooting iterations and the fact that the constraints are handled softly by a smooth exterior penalty method, as shown in (2*b*). Strict enforcement of the position constraints will be investigated in future work.

4.2 Influence of the Penalty on Acceleration.

This section investigates the tradeoff between a driver's ride comfort and fuel economy. As can be seen from the dot dash curves in Fig. 3(*a*) (also observed in the entire driving cycle when the speed is not too high), the resulting acceleration obtained from direct fuel-efficient EMPC is of a bang-bang type and may not be acceptable for abrupt changes in acceleration. The penalty on acceleration ρ as in (6) is set to be [0, 0.01, 0.1], and the MPC results with these penalty values are presented in Fig. 2 as compared to those obtained

[1] The computations are done on a Mac OS X with an Intel® Core i5 2.7 GHz processor and 8GB RAM.

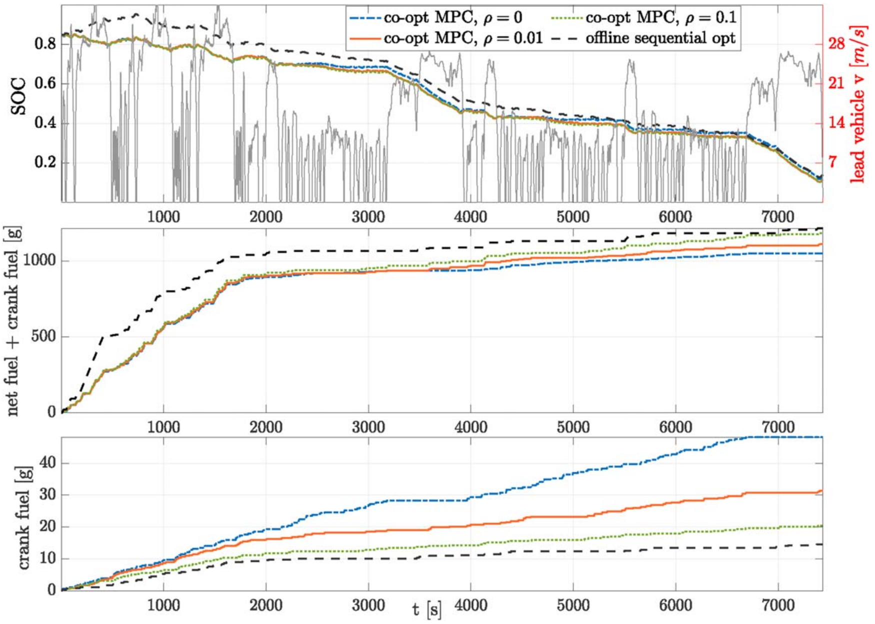

Fig. 2 Battery SOC and fuel consumption comparisons among MPC with different weights ρ on acceleration as in (6) and offline sequential optimization on a 2-h trip

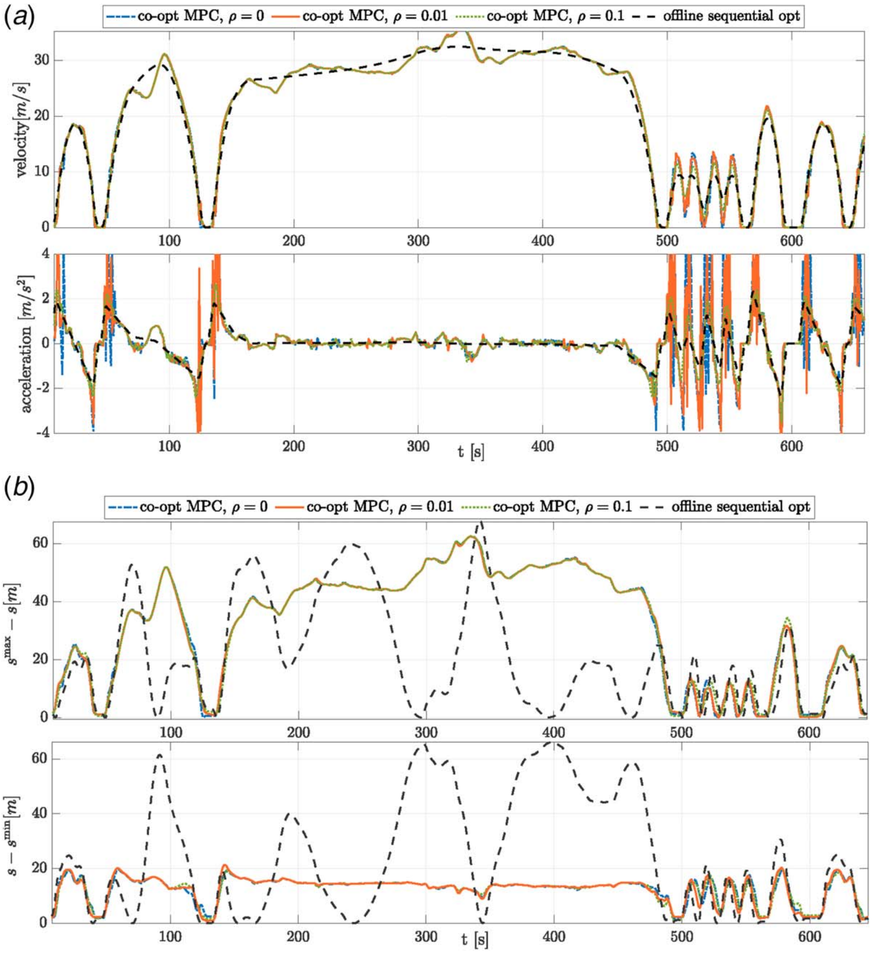

Fig. 3 Detailed views of resulting velocity, acceleration, and distance gap between 0 and 700 s. (*a*) Comparisons of velocity and acceleration and (*b*) Comparisons of distance gaps.

Table 1 Performance comparison among different controllers: the initial and terminal SOC values are set to be 0.85 and 0.14, respectively

Test name	Actual terminal SOC	Corrected total fuel (g)	Difference (%)
Offline seq. opt	0.14	1213	0
Co-opt $\rho = 0$	0.13	1062	−12.4
Co-opt $\rho = 0.01$	0.12	1146	−5.5
Co-opt $\rho = 0.1$	0.12	1219	−0.5

from offline sequential optimization. As seen from Fig. 3(*a*), the resulting acceleration becomes smoother with an increase in the penalty ρ upon acceleration. However, the total fuel consumption increases, as shown in the second subplot in Fig. 2 and Table 1, meaning that a smoothed driving trace is not the most fuel-efficient one for a PHEV.

5 Conclusion

This paper proposes a receding-horizon control framework to determine the powertrain operation and velocity simultaneously for fuel-efficient car-following of a PHEV. To resolve the conflict between the horizon length and the resulting computation complexity, we propose approximating the future cost with the co-state. The co-state based on the data from a nominal trajectory is adjusted with a TD error for preventing oscillations and systematic bias in the co-state estimation. The proposed control strategy demonstrates an additional 12% fuel economy benefit in its online implementation compared to what can be achieved by the offline solution of a typical layered approach, velocity smoothing followed by power-split optimization. Additionally, to accommodate drivability, a penalty on acceleration is considered, and the degradation in fuel economy is quantified.

Acknowledgment

The work was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000837, also known as NEXTCAR. The authors would like to thank Southwest Research Institute and Toyota Motor North America Research & Development for continuous feedback and support.

Conflict of Interest

There are no conflicts of interest.

Nomenclature

K = the rate of adjustment of co-state based on the difference in the SOC, used in the single shooting

$\hat{p}$ = co-state sampled from the nominal trajectory $\overline{p}$

$\tilde{p}$ = co-state corrected by TD error

$\mathbf{P}_k^{N_f(N_p)}(\mathbf{SOC}_f)$ = the MPC problem at time $t = k\Delta t$ with a horizon length N_f that equals to the length of the entire trip (prediction horizon N_p). The resulting terminal state of charge (SOC) at the end of the trip should be equal (or very close) to the desired value SOC_f ($\text{SOC}_{f,k+N_p}$)

$\overline{\text{SOC}}$, $\overline{p}$ = nominal SOC and its corresponding co-state trajectories obtained as an average of offline simulation results

References

[1] Sciarretta, A., and Vahidi, A., 2020, Energy-Efficient Speed Profiles (Eco-Driving), Springer, Cham, pp. 131–178.

[2] Bae, S., Choi, Y., Kim, Y., Guanetti, J., Borrelli, F., and Moura, S., 2019, “Real-Time Ecological Velocity Planning for Plug-In Hybrid Vehicles With Partial Communication to Traffic Lights,” 2019 IEEE 58th Conference on Decision and Control (CDC), Nice, France, Dec. 11–13, pp. 1279–1285.

[3] Heppeler, G., Sonntag, M., Wohlhaupter, U., and Sawodny, O., 2017, “Predictive Planning of Optimal Velocity and State of Charge Trajectories for Hybrid Electric Vehicles,” Control Eng. Pract., **61**, pp. 229–243.

[4] Huang, M., Zhang, S., and Shibaike, Y., 2019, “Real-Time Long Horizon Nodel Predictive Control of a Plug-Pn Hybrid Vehicle Power-Split Utilizing Trip Preview,” Technical Report, SAE Technical Paper.

[5] Chen, D., Kim, Y., Huang, M., and Stefanopoulou, A., 2020, “An Iterative and Hierarchical Approach to Co-Optimizing the Velocity Profile and Power-Split of Plug-In Hybrid Electric Vehicles,” 2020 American Control Conference (ACC), IEEE, pp. 3059–3064.

[6] Grüne, L., and Pannek, J., 2017, Nonlinear Model Predictive Control, Springer, London, pp. 45–69.

[7] Lian, C., Xu, X., Chen, H., and He, H., 2015, “Near-Optimal Tracking Control of Mobile Robots Via Receding-Horizon Dual Heuristic Programming,” IEEE Trans. Cybernet., **46**(11), pp. 2484–2496.

[8] Alamir, M., and Attia, S.-A., 2004, “On Solving Optimal Control Problems for Switched Hybrid Nonlinear Systems by Strong Variations Algorithms,” Proceedings of 6th IFAC Symposium on Nonlinear Control Systems, pp. 558–563.

[9] Chen, D., Prakash, N., Stefanopoulou, A. G., Huang, M., Kim, Y., and Hotz, S. R., 2018, “Sequential Optimization of Velocity and Charge Depletion in a Plug-In Hybrid Electric Vehicle,” 14th International Symposium on Advanced Vehicle Control, Beijing, China, July 16–20, pp. 558–563.

[10] Hyeon, E., Kim, Y., Prakash, N., and Stefanopoulou, A. G., 2019, “Influence of Speed Forecasting on the Performance of Ecological Adaptive Cruise Control,” ASME 2019 Dynamic Systems and Control Conference, Park City, UT, Oct. 8– 11.

[11] Prakash, N., Cimini, G., Stefanopoulou, A. G., and Brusstar, M. J., 2016, “Assessing Fuel Economy From Automated Driving: Influence of Preview and Velocity Constraints,” ASME 2016 Dynamic Systems and Control Conference, Minneapolis, MN, Oct. 12–14.