Title

Chapter 2 - Data-Driven Energy Efficient Driving Control in Connected Vehicle Environment

Authors
Qi, Xuewei
Boriboonsomsin, Kanok
Barth, Matthew J

Publication Date
2019

Peer reviewed

Chapter 2

Data-Driven Energy Efficient Driving Control in Connected Vehicle Environment

Xuewei Qi[*,†], Guoyuan Wu[†], Kanok Boriboonsomsin[†] and Matthew J. Barth[*,†]

[*] *Department of Electrical and Computer Engineering, University of California, Riverside, CA, United States,* [†] *College of Engineering-Centre for Environmental Research and Technology (CE-CERT), University of California, Riverside, CA, United States*

Chapter Outline

Data-Driven Solutions to Transportation Problems. https://doi.org/10.1016/B978-0-12-817026-7.00002-3

At the heart of Plug-in hybrid electric vehicles (PHEV) technologies, the energy management system (EMS) whose functionality is to control the power streams from both the internal combustion engine (ICE) and the battery pack based on vehicle and engine operating conditions have been studied extensively. In the past decade, a large variety of EMS implementations have been developed for HEVs and PHEVs, whose control strategies may be well categorized into two major classes:

(a) Rule-based strategies rely on a set of simple rules without a priori knowledge of driving conditions. Such strategies make control decisions based on instant conditions only and are easily implemented, but their solutions are often far from optimal due to the lack of consideration of variations in trip characteristics and prevailing traffic conditions.

(b) Optimization-based strategies are aimed at optimizing some predefined cost function according to the driving conditions and vehicle's dynamics. The selected cost function is usually related to the fuel consumption or tailpipe emissions.

Based on how the optimization is implemented, such strategies can be further divided into two groups: (1) off-line optimization which requires a full knowledge of the entire trip to achieve the global optimal solution; and (2) short-term prediction-based optimization, which takes into account the predicted driving conditions in the near future and achieves local optimal solutions segment by segment within an entire trip. However, major drawbacks of these strategies include heavy dependence on the knowledge of future driving conditions and high computational costs that are difficult to implement in real-time.

To address the aforementioned issues, we propose two data-driven on-line energy management strategies for PHEV energy efficient driving control in connected vehicle environment:

- Data-driven evolutionary algorithm-based self-adaptive EMS, which utilizes the rolling horizon technique to update the prediction of propulsion load as well as the power-split control. There are two major advantages over the existing strategies: (a) computationally competitive. There is no need to initiate a complete process for optimization while the algorithm keeps evolving and converging to obtain an optimal solution; (b) no a priori knowledge about the trip duration required.
- Data-driven reinforcement learning-based EMS, which is capable of simultaneously controlling and learning the optimal power-split operations in real-time from the historical driving data. There are three major features:

(1) this model can be implemented in real-time without any prediction efforts, since the control decisions are made only upon the current system state. The control decisions also considered for the entire trip information by learning the optimal or near-optimal control decisions from historical driving behavior. Therefore, this model achieves a good balance between real-time performance and energy saving optimality; (2) the proposed model is a data-driven model which does not need any PHEV model information once it is well trained, since all the decision variables can be observed and are not calculated using any vehicle powertrain models (these details are described in the following sections); and (3) compared to existing RL-based EMS implementations, the proposed strategy considers charging opportunities along the way (a key distinguishing feature of PHEVs as compared with HEVs).

The validation over real-world driving data has indicated that the proposed data-driving EMS strategies are very promising in terms of achieving a good balance between real-time performance and fuel savings when compared with some existing strategies, such as binary mode EMS and dynamic programming-based EMS. In addition, there is no requirement for the (predicted) information on the entire route.

2.1 INTRODUCTION

Air pollution and climate change impacts associated with the use of fossil fuels have motivated the electrification of transportation systems. In the realm of powertrain electrification, groundbreaking changes have been witnessed in the past decade in terms of research and development of hybrid electric vehicles (HEVs) and electric vehicles (EVs) [1]. As a combination of HEVs and EVs, PHEVs can be plugged into the electrical grid to charge their batteries, thus increasing the use of electricity and achieving even higher overall fuel efficiency, while retaining the ICE that can be called upon when needed [2].

In comparison to conventional HEVs, the EMS in PHEVs are significantly more complex due to their extended electric-only propulsion (or extended all-electric range capability) and battery chargeability via external electric power sources. Numerous efforts have been made in developing a variety of EMS for PHEVs [3, 4]. From the control perspective, existing EMS can be roughly classified as rule-based [5] and optimization-based [6]. This is discussed in more detail in Section 2.2.

In spite of all these efforts, most of the existing PHEV's EMS have one or more of the following limitations:

- Lack of adaptability to real-time information, such as traffic and road grade. This applies to rule-based EMS (either deterministic or using fuzzy logic) whose parameters or criteria have been pretuned to favor certain conditions (e.g., specific driving cycles and route elevation profiles) [3]. In addition, most EMS that are based on global optimization off-line assume that the future driving condition is known [2]. Thus far, only a few studies have focused on the development of on-line EMS for PHEVs [7].

- Dependence on accurate (or predicted) trip information that is usually unknown in advance. Many of the existing EMS require at a minimum the trip duration as known or predicted information prior to the trip [8]. Furthermore, it is reported that the performance of EMS is largely dependent on the time span of the trip [8]. Very few studies analyze the impacts of trip duration on the performance of EMS for PHEVs.
- Emphasis on a single trip level optimization without considering opportunistic charging between trips. The most critical feature that differentiates PHEVs from conventional HEVs is that PHEVs' batteries can be charged by plugging into an electrical outlet. Most of the existing EMS are designed to work on a trip-by-trip basis. However, taking into account inter-trip charging information can significantly improve the fuel economy of PHEVs [2].

2.2 BACKGROUND AND STATE OF THE ART

2.2.1 PHEV Modeling

Typically, there are three major types of PHEV powertrain architectures: (a) series, (b) parallel, and (c) power-split (series-parallel). This chapter focuses on the power-split architecture where the ICE and electric motors can power the vehicle, either alone or together, while the battery pack may be charged simultaneously through the ICE. Different approaches with various levels of complexity have been proposed for modeling PHEV powertrains [9]. However, a complex PHEV model with a large number of states may not be suitable for the optimization of PHEV energy control. A simplified but sufficiently detailed power-split powertrain model has been developed in MATLAB and used in this chapter. For more details, please refer to [2].

2.2.2 Operation Mode and SOC Profile

During the operation of a PHEV, the state-of-charge (SOC) may vary with time, depending on how the energy sources work together to provide the propulsion power at each instant. The SOC profile can serve as an indicator of the "PHEV" operating modes, i.e., charge sustaining (CS), pure electric vehicle (EV), and charge depleting (CD) modes [3], as shown in Fig. 2.1.

The CS mode occurs when the SOC is maintained at a certain level (usually the lower bound of SOC) by jointly using power from both the battery pack and the ICE. The pure EV mode is when the vehicle is powered by electricity only. The CD mode represents the state when the vehicle is operated using power primarily from the battery pack with supplemental power from the ICE as necessary. In the CD mode, the ICE is turned on if the electric motor is not able to

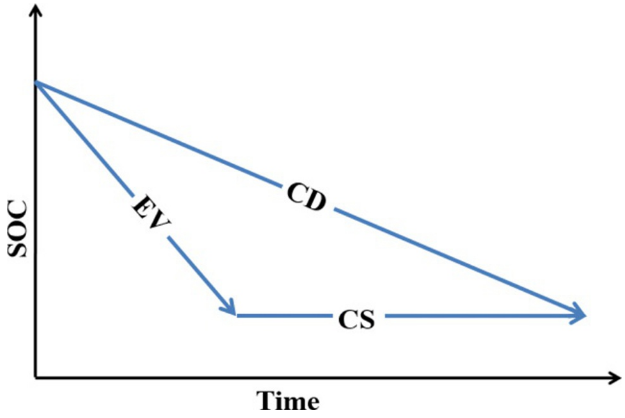

FIG. 2.1 Basic operation modes for PHEV.

provide enough propulsion power or the battery pack is being charged (even when the SOC is much higher than the lower bound) in order to achieve better fuel economy.

2.2.3 EMS for PHEVs

The goal of the EMS in a PHEV is to satisfy the propulsion power requirements while maintaining the vehicle's performance in an optimal way. A variety of strategies have been proposed and evaluated in many previous studies [4]. A detailed literature review on EMS for PHEVs is provided in this section. Broadly speaking, the existing EMS for PHEVs can be divided into two major categories:

(1) Rule-based EMS are fundamental control schemes operating on a set of predefined rules without prior knowledge of the trip. The control decisions are made according to the current vehicle states and power demand only. Such strategies are easily implemented, but the resultant operations may be far from being optimal due to not considering future traffic conditions.

(2) Optimization-based EMS aim at optimizing a predefined cost function according to the driving conditions and behaviors. The cost function may include a variety of vehicle performance metrics, such as fuel consumption and tailpipe emissions.

For rule-based EMS, deterministic and fuzzy control strategies (e.g., binary control) have been well investigated. For optimization-based EMS, the strategies can be further divided into three subgroups based on how the optimizations are implemented:

(1) Off-line strategy which requires a full knowledge of the entire trip beforehand to achieve the global optimal solution;

(2) Prediction-based strategy or so-called real-time control strategy which takes into account predicted future driving conditions (in a rolling horizon manner) and achieves local optimal solutions segment-by-segment. This group of strategies is quite promising due to the rapid advancement and massive deployment of sensing and communication technologies (e.g., GPS) in transportation systems that facilitate the traffic state prediction; and.

(3) Learning-based strategy which is recently emerging owing to the research progress in machine learning techniques. In such a data-driven strategy, a dynamic model is no longer required. Based on massive historical and real-time information, trip characteristics can be learned and the corresponding optimal control decisions can be made through advanced data mining schemes. This strategy fits very well for commute trips.

Fig. 2.2 presents a classification tree of EMS for PHEVs and the typical strategies in each category, based on most existing studies.

In addition to the classification above, Table 2.1 highlights several important features which help differentiate the aforementioned strategies. Example references are also included in Table 2.1.

2.2.4 PHEVs' SOC Control

For a power-split PHEV, the optimal energy control is, in principle, equivalent to the optimal SOC control. Most of the existing EMS for PHEVs implicitly integrate SOC into the dynamic model and regard it as a key control variable [25], while only a few studies have explicitly described their SOC control strategies. A SOC reference control strategy is proposed in [20] where a supervisory SOC

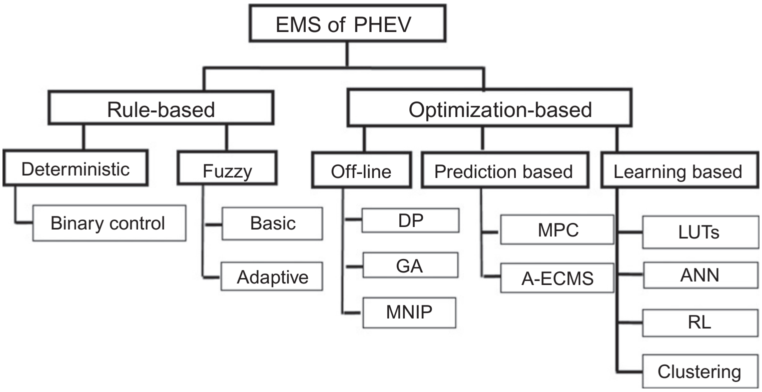

FIG. 2.2 Basic classification of EMS for PHEV. *Note*: PMP, Pontraysgin's minimum principle; MNIP, mixed nonlinear integer programming; DP, dynamic programming; QP, quadratic programming; RL, reinforcement learning; ANN, artificial neural network; LUTs, look-up-tables; MPC, model predictive control; AECMS, adaptive equivalent consumption minimization strategy.

TABLE 2.1 Classification of Current Literature

	Rule-Based	Off-Line Optimization	Prediction-Based	Learning-Based
Optimality	Local	Global	Local	Local
Real-time	Yes	No	Yes	Yes
SOC control	No	Yes	Yes	No
Need trip duration	No	Yes	Yes	Yes
Example references	[7,10–12]	[2, 6, 13–17]	[8, 18–23]	[9, 18, 19, 24–26]

planning method is designed to precalculate an optimal SOC reference curve. The proposed EMS then tries to follow this curve during the trip to achieve the best fuel economy. Another SOC control strategy is proposed in [8], where a probabilistic distribution of trip duration is considered. More recently, machine learning-based SOC control strategies (e.g., [9]) have emerged, where the optimal SOC curves are precalculated using historical data and stored in the form of look-up tables for real-time implementation. A common drawback for all these strategies is that accurate trip duration information is required in an either deterministic or probabilistic way. In reality, however, such information is hard to know ahead of time or may vary significantly due to the uncertainties in traffic conditions. To ensure the practicality of our proposed EMS for PHEVs, we employ a self-adaptive SOC control strategy in this chapter that does not require any information about the trip duration (or length).

2.3 PROBLEM FORMULATION

2.3.1 Data-Driven On-Line EMS Framework for PHEVs

In this chapter, we propose an on-line EMS framework for PHEVs, using the receding horizon control structure (see Fig. 2.3). The proposed EMS framework consists of information acquisition (from external sources), prediction, optimization, and power-split control. With the receding horizon control, the entire trip is divided into segments or time horizons. As shown in Fig. 2.4, the prediction horizon (N sampling time steps) needs to be longer than the control horizon (M sampling time steps). Both horizons keep moving forward (in a rolling horizon style) while the system is operating. More specifically, the prediction model is used to predict the power demand at each sampling step (i.e., each second) in the prediction horizon. Then, the optimal ICE power supply for each second during the prediction horizon is calculated with this predicted information.

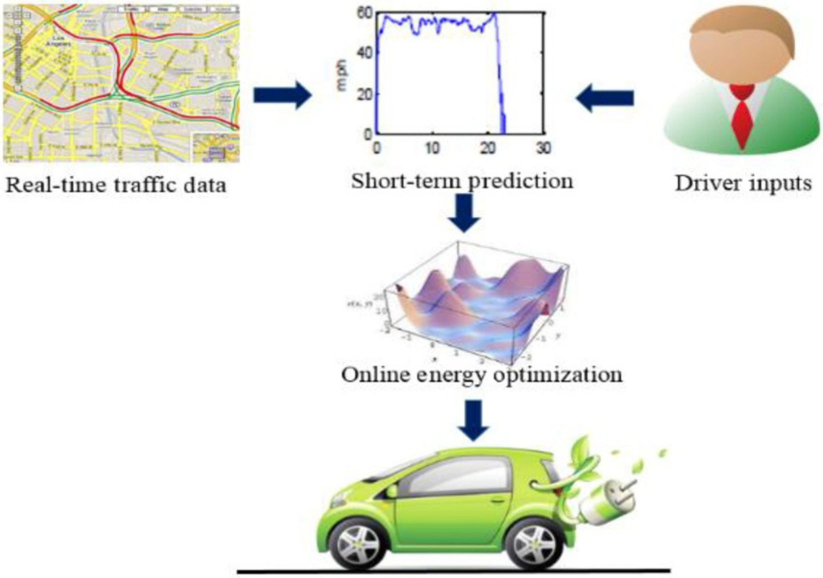

FIG. 2.3 Flow chart of the proposed on-line EMS.

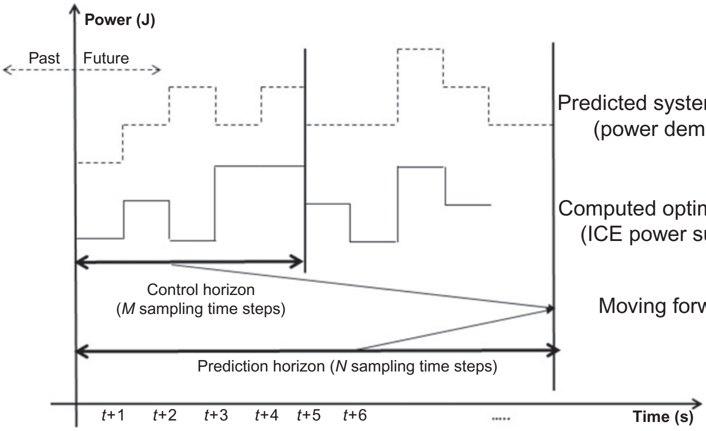
Power (J)
Past
Future
Predicted system states (power demand)
Computed optimal input (ICE power supply)
Control horizon (*M* sampling time steps)
Moving forward
Prediction horizon (*N* sampling time steps)
t+1
t+2
t+3
t+4
t+5
t+6
.....
Time (s)

FIG. 2.4 Time horizons of prediction and control.

In each control horizon, the precalculated optimal control decisions are inputted into the powertrain control system (e.g., electronic control unit, or ECU) at the required sampling frequency. In this chapter, we focus on the on-line energy optimization, assuming that the short-term prediction model is available (which is one of our future research topics).

2.3.2 Optimal Power-Split Control Formulation

Mathematically, the optimal (in terms of fuel economy) energy management for PHEVs can be formulated as a nonlinear constrained optimization problem. The objective is to minimize the total fuel consumption by ICE along the entire trip.

$$\begin{cases} \min\left\{\int_0^T h(\omega_e, q_e, t)dt\right\} \\ \text{subject to:} \\ \dot{\text{SOC}} = f(\text{SOC}, \omega_{\text{MG1}}, q_{\text{MG1}}, \omega_{\text{MG2}}, q_{\text{MG2}}) \\ (\omega_e, q_e) = g(\omega_{\text{MG1}}, q_{\text{MG1}}, \omega_{\text{MG2}}, q_{\text{MG2}}) \\ \text{SOC}_{\min} \le \text{SOC} \le \text{SOC}_{\max} \\ \omega_{\min} \le \omega_e \le \omega_{\max} \\ q_{\min} \le q_e \le q_{\max} \end{cases} \tag{2.1}$$

where T is the trip duration, ω_e, q_e are the engine's angular velocity and engine's torque, respectively, $h(\omega_e, Tq_e)$ is ICE fuel consumption model, ω_{MG1}, q_{MG1} are the first motor/generator's angular velocity and torque, respectively, ω_{MG2}, q_{MG2} are the second motor/generator's angular velocity and torque, respectively, and $f(\text{SOC}, \omega_{\text{MG1}}, q_{\text{MG1}}, \omega_{\text{MG2}}, q_{\text{MG2}})$ is the battery power consumption model. For more details about the model derivations and equations, please refer to [2].

Such a formulation is quite suitable for traditional mathematical optimization methods [13] with high computational complexity. In order to facilitate on-line optimization, we herein discretize the engine power and reformulate the optimization problem represented by Eq. (2.1) as follows:

$$\min \sum\nolimits_{k=1}^{T} \sum\nolimits_{i=1}^{N} x(k,i) P_i^{\text{eng}} / \eta_i^{\text{eng}} \tag{2.2}$$

subject to

$$\sum\nolimits_{k=1}^{j} f\left(P_k - \sum\nolimits_{i=1}^{N} x(k,i) P_i^{\text{eng}}\right) \le C \quad \forall j = 1, \ldots, T \tag{2.3}$$

$$\sum\nolimits_{i=1}^{N} x(k,i) = 1 \quad \forall k \tag{2.4}$$

$$x(k,i) = \{0,1\} \quad \forall k, i \tag{2.5}$$

where N is the number of discretized power level for the engine, k is the time step index, i is the engine power level index, C is the gap of the battery pack's SOC between the initial and the minimum, P_i^{eng} is the ith discretized level for the engine power and η_i^{eng} is the associated engine efficiency, and P_k is the driving power demand at time step k.

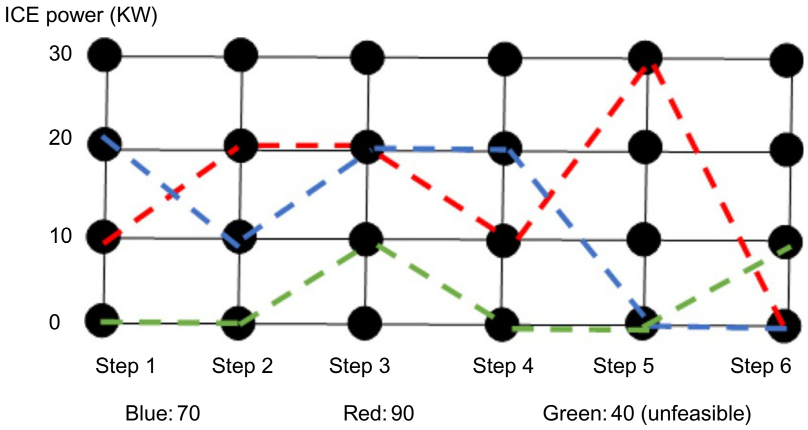

FIG. 2.5 Example solutions of power-split control.

Furthermore, if the change in SOC (ΔSOC) for each possible engine power level at each time step is pre-calculated given the (predicted) power demand, then constraint (2.3) can be replaced by

$$\mathrm{SOC}^{\mathrm{ini}} - \mathrm{SOC}^{\mathrm{max}} \le \sum\nolimits_{k=1}^{j} x(k,i)\Delta\mathrm{SOC}(k,i) \le \mathrm{SOC}^{\mathrm{ini}} - \mathrm{SOC}^{\mathrm{min}} \quad \forall j = 1,\ldots,T \tag{2.6}$$

where $\mathrm{SOC}^{\mathrm{ini}}$ is the initial SOC, and $\mathrm{SOC}^{\mathrm{min}}$ and $\mathrm{SOC}^{\mathrm{max}}$ are the minimum and maximum SOC, respectively. Therefore, the problem is turned into a combinatory optimization problem whose objective is to select the optimal ICE power level for each time step given the predicted information in order to achieve the highest fuel efficiency for the entire trip. Fig. 2.5 gives three example ICE power output solutions. The solution represented by the blue line (starting from 20 KW) has a lower total ICE power consumption (i.e., 40 units) than the red line (starting from 10 KW) (i.e., 90 units), while the green line (starting from 0 KW) represents an infeasible solution due to the SOC constraint.

2.4 DATA-DRIVEN EVOLUTIONARY ALGORITHM (EA) BASED SELF-ADAPTIVE ON-LINE OPTIMIZATION

The motivations for applying EA are:

(1) compared to the traditional derivative or gradient-based optimization methods, EAs are easier to implement and require less complex mathematical models;

(2) EAs are very good at solving nonconvex optimization problems where there are multiple local optima; and

(3) it is very flexible to address multiobjective optimization problems using EAs.

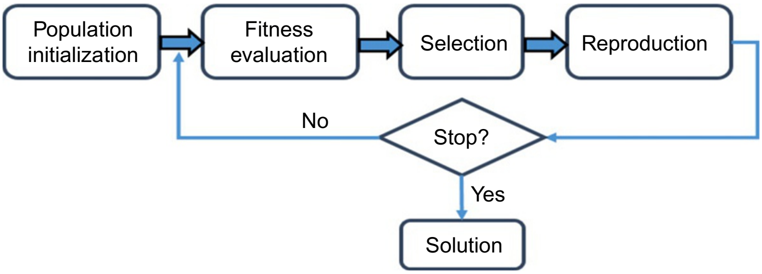

FIG. 2.6 Estimation and sampling process of EA.

Theoretically, in the proposed framework, any EAs can be used to solve the optimization problem for each prediction horizon described in Fig. 2.4. A typical EA is a population-based and iterative algorithm that starts searching for the optimal solution with a random initial population. Then, the initial population undergoes an iterative process that includes multiple operations, such as fitness evaluation, selection, and reproduction, until certain stopping criteria are satisfied. The flow chart of an EA is provided in Fig. 2.6.

Among many EAs, the estimation distribution algorithm (EDA) is very powerful in solving high-dimensional optimization problems and has been applied successfully to many different engineering domains [27]. In this chapter, we choose EDA as the major EA kernel in the proposed framework due to the high-dimensionality nature of the PHEV energy management problem. This selection is justified by experimental results in the following sections.

In the problem representation of EDA, each individual (encoded as a row vector) of the population defined in the algorithm is a candidate solution. For the PHEV energy management problem, the size of the individual (vector) is the number of time steps within the trip segment. The value of the *i*th element of the vector is the ICE power level chosen for that time step. In the example individual in Table 2.2, the ICE power level is 3 (or 3 kW) for the first time step, 0 kW (i.e., only battery pack supplies power) for the second time step, 1 for the third time step, and so forth.

It is very flexible to define a fitness function for EAs. Since the objective is to minimize fuel consumption, the fitness function herein can be defined as the summation of total ICE fuel consumption for the trip segment defined by Eq. (2.5) and a penalty term

$$f(s) = C_{\text{fuel}} + P \tag{2.7}$$

where s is a candidate solution, C_{fuel} is fuel consumption, and P is the imposed penalty that is the largest possible amount of energy that can be consumed in this trip segment. The penalty is introduced to guarantee the feasibility of the solution, satisfying constraint (2.3), which means that the SOC should always

TABLE 2.2 Representation of One Example Individual

Time	1 s	2 s	3 s	4 s	....	$n-3$	$n-2$	$n-1$	n
Individual	3	0	1	4	....	1	2	0	5

fall within the required range at each time step. Then, all the individuals in the population are evaluated by the fitness function and ranked by their fitness values in an ascending order since this is a minimization problem. A good evaluation and ranking process is crucial in guiding the evolution towards good solutions until the global optima (or near optima) is located.

Furthermore, EDA assumes that the value of each element in a good individual of the population follows a univariate Gaussian distribution. This assumption has been proven to be effective in many engineering applications [28], although there could be other options [29]. For each generation, the top individuals (candidate solutions) with least fuel consumption values are selected as the parents for producing the next generation by an estimation and sampling process [30].

The flow chart of the proposed EDA-based on-line EMS is presented in Fig. 2.7. t_0 is the current time, N is the length of the prediction time horizon,

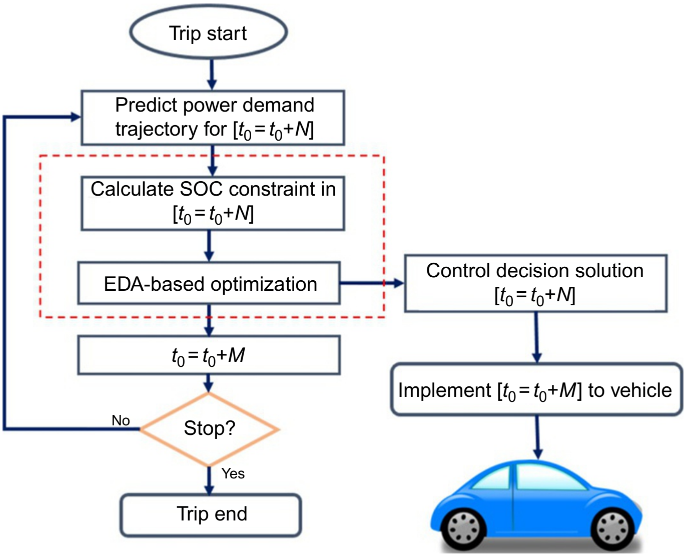
Trip start
Predict power demand trajectory for [$t_0 = t_0+N$]
Calculate SOC constraint in [$t_0 = t_0+N$]
EDA-based optimization
Control decision solution [$t_0 = t_0+N$]
$t_0 = t_0+M$
Implement [$t_0 = t_0+M$] to vehicle
No
Stop?
Yes
Trip end

FIG. 2.7 EDA-based on-line energy management system.

and M is length of the control time horizon. The block highlighted by the dashed box is the core component of the system, and more details about this block is given in Section 2.4.

2.4.1 Optimality and Complexity

Evolutionary algorithms (EA) are stochastic search algorithms that do not guarantee to find the global optima. Hence, in the proposed on-line EMS, the optimal power control for each trip segment is not guaranteed to be found. Moreover, EAs are also population-based iterative algorithms that are usually criticized due to their heavy computational loads [31], especially for real-time applications. Theoretically, time complexity of EAs is worse than $\theta(m^2 * \log(m))$ where m is the size of the problem [32]. However, we apply the receding horizon control technique in this chapter, where the entire trip is divided into small segments. Therefore, the computational load can be significantly reduced since the EA-based optimization is applied only for each small segment rather than the entire trip. In this sense, the proposed framework can be implemented in "real-time," as long as the optimization for the next prediction horizon can be completed in the current control horizon (see Fig. 2.4). As previously discussed, the rule-based EMS can run in real-time but the results may be far from optimal while most of the optimization-based EMS have to operate off-line. Therefore, the proposed on-line EMS would be a well-balanced solution between the real-time performance and optimality.

2.4.2 SOC Control Strategies

An appropriate SOC control strategy is critical in achieving the optimal fuel economy for PHEVs [33]. In the previously presented problem formulation, the major constraint for SOC is defined by Eq. (2.6), which means that at any time step, the SOC should be within the predefined range (e.g., between 0.2 and 0.8) to avoid damage to the battery pack. However, this constraint only may not be enough to accelerate the search for the optimal solution. Hence, additional constraint(s) on battery use (e.g., reference bound of SOC) should be introduced to improve the on-line EMS. To investigate the effectiveness of different SOC control strategies within the proposed framework, two types of SOC control strategies—reference control and self-adaptive control—are designed and evaluated in this chapter.

2.4.2.1 SOC Reference Control (Known Trip Duration)

When the trip duration is known, a SOC curve can be pre-calculated and used as a reference to control the use of battery power along the trip to achieve optimal fuel consumption. We propose three heuristic SOC references (i.e., lower

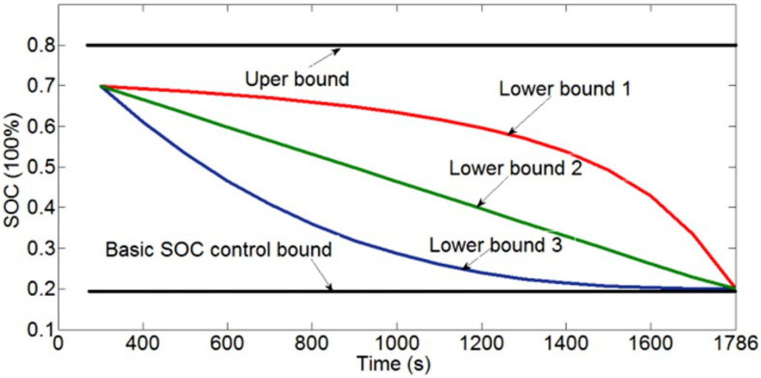

FIG. 2.8 SOC reference control bound examples.

bounds) in this chapter (see Fig. 2.8 for example): (1) concave downward; (2) straight line; and (3) concave upward. These SOC minimum bounds are generated based on the given trip duration information by the following equations, respectively:

- Concave downward control (lower bound 1):

$$\mathrm{SOC}_i^{\mathrm{min}} = \frac{\left(\mathrm{SOC}_{\mathrm{init}} - \mathrm{SOC}^{\mathrm{min}}\right)}{T - (i^*M)} {}^*N + \mathrm{SOC}^{\mathrm{init}} \tag{2.8}$$

- Straight line control (lower bound 2):

$$\mathrm{SOC}_i^{\mathrm{min}} = \frac{-\left(\mathrm{SOC}_i^{\mathrm{min}} - \mathrm{SOC}^{\mathrm{min}}\right)}{T} \cdot ((i-1) \cdot M + N) + \mathrm{SOC}^{\mathrm{init}} \tag{2.9}$$

- Concave upward control (lower bound 3):

$$\mathrm{SOC}_i^{\mathrm{min}} = \frac{-\left(\mathrm{SOC}_{i-1}^{\mathrm{end}} - \mathrm{SOC}^{\mathrm{min}}\right)}{T - (i^*M)} {}^*N + \mathrm{SOC}_{i-1}^{\mathrm{end}} \tag{2.10}$$

where i is the segment index; $\mathrm{SOC}_i{}^{\mathrm{min}}$ is the minimum SOC at the end of ith segment; and $\mathrm{SOC}_{i-1}{}^{\mathrm{end}}$ is the SOC at the end of last control horizon. It is self-evident that the concave downward bound (i.e., lower bound 1) is much more restrictive than a concave upward bound (i.e., lower bound 3) in terms of battery energy use at the beginning of the trip.

A major drawback for these reference control strategies is that they assume that the trip duration (i.e., T) is given, or at least can be well estimated beforehand. As mentioned earlier, this assumption may not hold true for many real-world applications. Therefore, a new SOC control strategy without relying on the knowledge of trip duration would be more attractive.

TABLE 2.3 Example Fitness Evaluation by Different Fitness Functions

Indiv. Index	Fuel Con.		SOC Decrease	R_{fuel}	R_{soc}	Rank by Eq. (2.7)	Rank by Eq. (2.11)
1	0.001		0.005(P)	5	35	98	140
2	0.010		0.002	25	14	33	39
3	0.007		0.003	19	23	24	42
4	0.002		0.004(P)	7	32	99	139
....	......		....	....		....	....

2.4.2.2 SOC Self-Adaptive Control (Unknown Trip Duration)

In this chapter, we also propose a novel self-adaptive SOC control strategy for real-time optimal charge-depleting control, where trip duration information is not required. Unlike those SOC reference control strategies that control the use of battery by explicit reference curves, the self-adaptive control strategy controls the battery power utilization implicitly by adopting a new fitness function in place of the one in Eq. (2.7):

$$f(s) = R_{\text{fuel}} + R_{\text{soc}} + P' \tag{2.11}$$

where R_{fuel} and R_{soc} are the ranks (in an ascending order) of ICE fuel consumption and SOC decrease, respectively, of an individual candidate solution s in the current population; and $P'$is the added penalty when the individual s violates the constraints given in Eq. (2.6). The penalty value is selected to be greater than the population size in order to guarantee that an infeasible solution always has a lower rank (i.e., larger fitness value) than a feasible solution in the ascending order by fitness value. Compared to the fitness function adopted for SOC reference control (see Eq. (2.7)), this new fitness function tries to achieve a good balance between two conflicting objectives: least fuel consumption and least SOC decrease. For a better understanding of the differences between these two fitness functions, Table 2.3 provides an example of fitness evaluation of the same population. In this case, the population size is 100. As we can see in the table, Individual 2, who has a better balance between fuel consumption, and SOC decrease, is more favorable than Individual 3 in the ranking by Eq. (2.11) than that by Eq. (2.7).

2.4.3 EDA-Based On-Line EMS Algorithm With SOC Control

Details of the proposed EDA-based on-line EMS algorithm with SOC control are summarized in Algorithm 1. This algorithm is implemented on each prediction horizon (*N* time steps) within the framework presented in Fig. 2.8 (see the box with dashed line).

Algorithm 1: EDA-based on-line EMS with SOC control

```
1: Initialize a random output solution $I_{best}$($N$ time steps)
2: $P_{current}$ <= Generate initial population randomly
3: While iteration_number $\leq$ Max_iterations, do
4:     For each individual s in $P_{current}$
5:         Calculate fuel consume $C_{fuel}$ using Eq. (2.1).
6:         Calculate SOC decrease using Eq. (2.5)
7:         Obtain the rank index of s: $R_{fuel}$
8:         Obtain the rank index of s: $R_{soc}$
9:           If SOC reference control is adopted
10.               Calculate the lower bound using Eqs. (2.8)–(2.10)
11:                If individual $s$ violates Eq. (2.6)
12:                  $P = P_0$;//largest fuel consumption in $N$ steps
13:             Else
14:                 $P = 0$;
15:             End If
16:               Calculate the fitness value for s using Eq. (2.7)
17:          Else If SOC self-adaptive control is adopted
18:                 If individual $s$ violates Eq. (2.6)
19:                   $P'=S$
20:             Else
21:                 $P'=0$;
22:             End If
23:               Calculate the fitness value for s using Eq. (2.11)
24:          End If
25:     End For
26:     Rank $P_{current}$ in ascending order based on fitness
27:     $P_{top}$ <= Select top $\alpha$% individuals from $P_{current}$
28:     E < = Estimate a new distribution from $P_{top}$
29:     $P_{new}$ <= Sample $N$ individuals from built model E
30:     Evaluate each individual in $P_{new}$ using line 5–14
31:     Mix $P_{current}$ and $P_{new}$ to form 2$N$ individuals
32:     Rank 2$N$ individuals in ascending order by fitness
33:     $P_{current}$<= Select top $N$ individuals
34:     Update $I_{best}$ if a better one is identified.
35:     Iteration_number ++
36: End While
37: Output $I_{best}$
```

In the following section, we compare the performance of the proposed self-adaptive SOC control with other SOC control strategies. For convenience, we list the abbreviations of all the involved strategies in Table 2.4.

TABLE 2.4 Abbreviations of Different SOC Control Strategies Compared in This Chapter

SOC Control Strategies	Abbreviations
Binary control	B-I
Basic SOC control	B-A
Concave downward	C-D
Straight line	S-L
Concave upward	C-U
Self-adaptive SOC control	S-A

2.4.4 Synthesized Trip Information

To validate the proposed EMS for PHEVs, we use real-world data collected on January 17, 2012, along I-210 between I-605 and Day Creek Blvd in San Bernardino, California, as a case study (see Fig. 2.9). Please refer to [2] for more detailed description of data collection and specifications of the power-split PHEV model if interested.

Based on the collected traffic data along with road grade information, second-by-second vehicle velocity trajectory and power demand have been synthesized as described in [2]. As pointed out earlier, it is impractical to have a priori knowledge of the exact vehicle velocity trajectory. In this chapter, we focus on the development of the optimal power-split control, assuming perfect

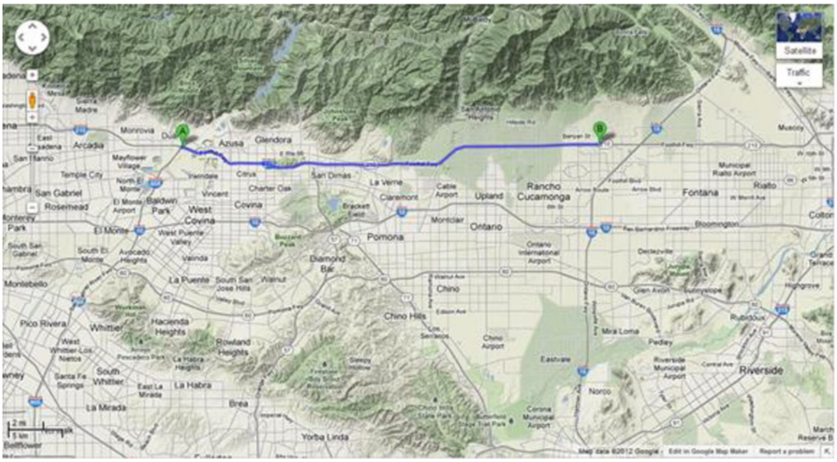

FIG. 2.9 Example trip along I-210 in southern California used for evaluation.

prediction of vehicle velocity trajectory. Research on improving the prediction of vehicle velocity trajectory in real-time is part of our future work.

2.4.5 Off-Line Optimization for Validation

To justify the selection of EDA as the kernel of the proposed framework, we first test EDA on the full-trip off-line optimization. The results are compared with those obtained from two other popular EA: genetic algorithm (GA) and particle swarm optimization (PSO). The fitness (i.e., total ICE energy consumption) of EDA-based off-line optimization obtains better fuel economy (0.346 gal) than the other two (0.364 gal for GA and 0.377 for PSO, respectively), at the same computational expense (i.e., same population size and same number of iterations) [30]. In addition, the result from EDA is much closer to the global optimum (0.345 gal in this case), with the difference being $<1\%$.

2.4.6 Real-Time Performance Analysis and Parameter Tuning

As aforementioned, a necessary condition for on-line implementation of the proposed EMS is that the optimization for the next prediction horizon has to be finished within the current control horizon (see Fig. 2.4). In our study, for example, the optimization for a prediction horizon of 50 s can be completed within 1.1 s (with Intel Core i7, 3.4 GHz, RAM 4G, and 64bit-Matlab 2012). In addition, one of our previous works [30] has shown that the lengths of prediction horizon and control horizon may significantly affect the algorithm performance. The best combination of these two parameters is found to be $N = 250$ and $M = 10$ in this case.

Unlike the conventional MPC, whose optimization has to be implemented along each prediction horizon, our proposed EA-based online EMS (see Fig. 2.7) can take advantage of the optimal results from previous prediction horizons, which avoids a new optimization starting from scratch and therefore saves a lot of computational overheads. As can be seen in Fig. 2.10, part of the optimal decisions from previous prediction optimization horizon are adopted as the seed for initial population of current prediction horizon optimization. For example, when the control horizon is 3 s and prediction/optimization horizon

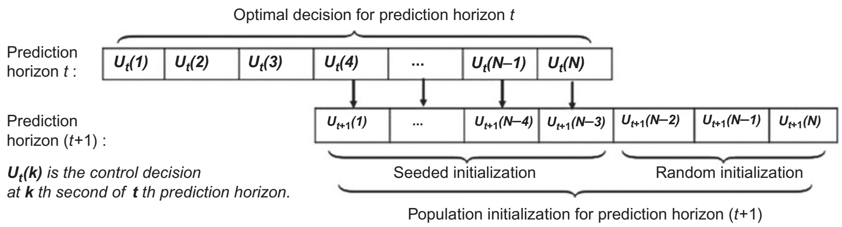
Optimal decision for prediction horizon t
Prediction horizon t :
$U_t(1)$
$U_t(2)$
$U_t(3)$
$U_t(4)$
...
$U_t(N-1)$
$U_t(N)$
Prediction horizon $(t+1)$:
$U_{t+1}(1)$
...
$U_{t+1}(N-4)$
$U_{t+1}(N-3)$
$U_{t+1}(N-2)$
$U_{t+1}(N-1)$
$U_{t+1}(N)$
$U_t(k)$ is the control decision at k th second of t th prediction horizon.
Seeded initialization
Random initialization
Population initialization for prediction horizon $(t+1)$

FIG. 2.10 Population initialization from the second prediction horizon (i.e., $t \geq 2$).

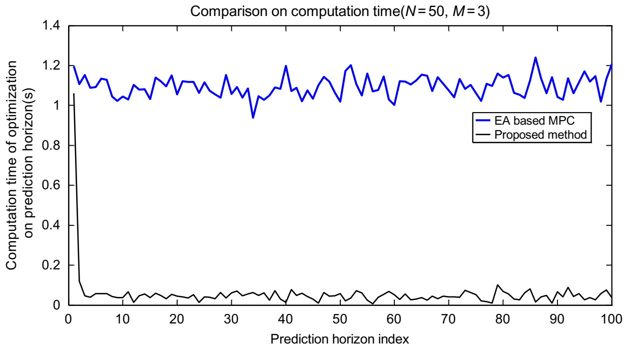

FIG. 2.11 Comparison of computation time.

is N, only 3 control decisions need to be randomly initialized and optimized in the second prediction/optimization horizon. This allows the optimization or search to be much more efficient compared to the same process over entire prediction horizon. To validate further this computational performance, we designed an EA-based MPC (EAMPC) that activates a complete new optimization for each prediction/optimization horizon, and compared it with our proposed model. The computation time track in Fig. 2.11 shows that for a 50-s prediction horizon, the conventional MPC takes around 1.1 s for each optimization horizon, but our proposed model can take <0.1 s to finish the optimization from the second prediction horizon.

2.4.7 On-Line Optimization Performance Comparison

To evaluate fully the performance of the proposed on-line EMS strategies, we compare them to the conventional binary control (implementable in real-time) strategy as well as the off-line global optimal control strategy (with the use of dynamic programming (DP) [11]). The comparisons are carried out on both the single trip scenario and multiple trips scenario.

When tested on a single (westbound) trip, the fuel consumption and SOC profiles by different strategies are illustrated in Fig. 2.12. It is shown that the proposed S-A algorithm achieves the lowest fuel consumption (0.3515 gal) which is only 1.56% worse than that of global optima obtained by the off-line optimization (0.3460 gal). These results can be explained by the shape of the resultant SOC profiles. For instance, SOC decreases very quickly in the B-I strategy, and reaches the lower bound (i.e., 0.2) at around 1200 s because the use of battery power is always prioritized whenever available. Therefore, ICE has to supply most of the demanded power after 1200 s. This is very similar to the cases of the B-A and C-U strategies where the battery power is also consumed aggressively at the beginning of the trip with very loose constraints. On the other hand, the S-L and C-D strategies perform better since their battery

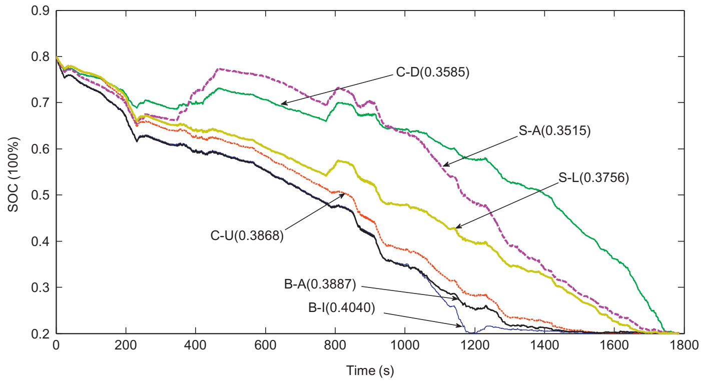

FIG. 2.12 SOC trajectories resulted from different control strategies.

power is used more cautiously along the trip. These findings are consistent with the conclusions of many other studies [8, 33] in that a smoother distribution of battery power usage along the trip would result in higher fuel efficiency.

In order to understand the statistical significance of the different EMS strategies, we test them on 30 randomly selected trip profile data extracted from the same road segment on 12 different days. The results are also compared to the binary control and D-P strategies. For the purpose of comparison, we set the fuel consumption obtained by the binary control strategy as the baseline and calculate the percentage of fuel savings achieved by the other EMS strategies. As we can see in Fig. 2.13, the D-P strategy achieves the best fuel savings with an average of 19.4% and the least variance simply because it is an off-line optimization strategy. The proposed S-A strategy achieves an average of 10.7% fuel savings, which is higher than all other on-line strategies and consistent with the result of

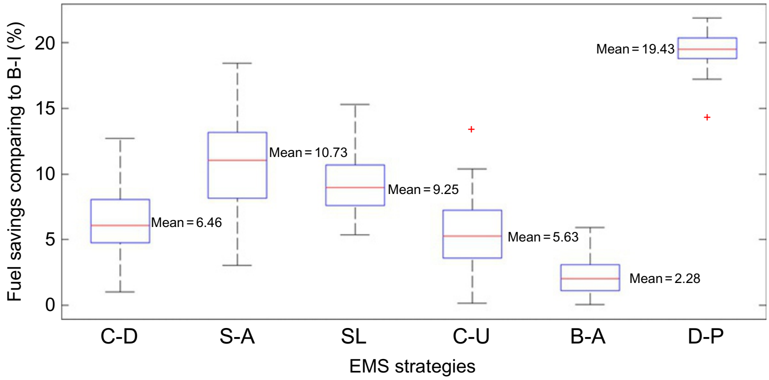

FIG. 2.13 Box-plot of fuel savings on 30 trips.

TABLE 2.5 Comparisons With Existing Models

EMS Model	Year	STP[a]	Trip Distance	FEI[b]	Consider Charging
This work	2016	Yes	Unknown	10.7%	Yes
EAMPC	2016	Yes	Unknown	7.9%	Yes
MPC [21]	2014	Yes	Known	8.5%	No
MPC [20]	2015	Yes	Known	6.7%	No
A-ECMS [21]	2014	Yes	Known	10.2%	No
A-ECMS [18]	2015	Yes	Known	7.6%	No
DP [22]	2015	Yes	Known	5.8%	No
SDP[c] [23]	2011	Yes	Known	7.7%	No

[a] *Short-term prediction.*
[b] *Fuel economy improvement comparing to binary control.*
[c] *Stochastic dynamic programming.*

the single trip test. An interesting observation is that the S-L strategy has better average fuel savings (i.e., 9.3%) than the C-D and C-U strategies, which is not consistent with the test result of the single trip test. A possible reason is that the C-D strategy performs better on some trips in which the power demand is higher in later stages of the trip, but the C-U strategy performs better on the trips in which the power demand is higher in earlier stages. On the other hand, the S-L strategy balances the SOC control between these two types of trip pattern, and therefore has better average performance.

For further validation, the proposed S-A strategy with the best performance is compared with other existing PHEV EMS strategies that employ short-term prediction. Although these strategies were proposed to handle powertrain models with different fidelity as well as different data set for validation, they all used the binary control strategy as a benchmark (the same as used in this chapter). This provides us with a chance to compare all models in a relatively fair manner. The comparison results are listed in Table 2.5, which proves that our model achieves the largest improvement of fuel efficiency (with regard to the binary control strategy) but requires less trip information.

2.4.8 Analysis of Trip Duration

In this section, we analyze and compare the effectiveness of the proposed on-line EMS for longer trips. These longer trips are constructed by concatenating multiple trip profiles, and the results are shown in Fig. 2.14. As can be

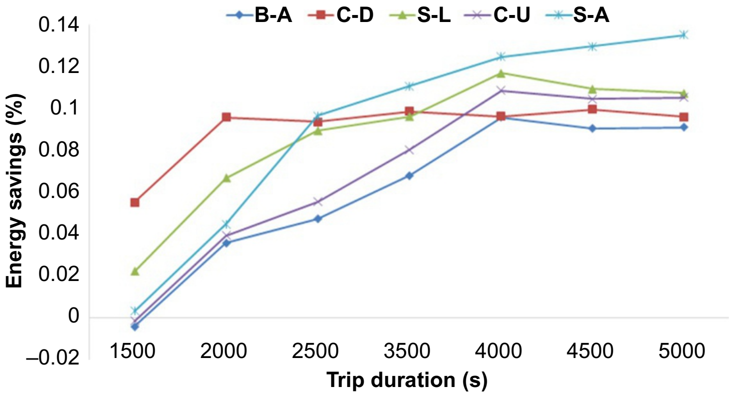

FIG. 2.14 Fuel savings for trips with different duration, compared to B-I.

observed, the B-I strategy has the best fuel economy when the trip duration is shorter than 1500 s. For these short trips, the PHEV can mostly rely on battery energy. However, as the trip duration becomes longer, especially when longer than 2500 s, the S-A strategy outperforms all the others.

To explain this finding further, the resultant fuel consumption and the corresponding SOC profiles for the longest trip (5000 s) are provided in Fig. 2.15. According to the figure, the S-A strategy has the lowest fuel consumption and its SOC profile is a combination of the CD mode (defined in Fig. 2.1) before 2000 s and the CS mode after 2000 s. This contradicts with most of the existing studies, which report that an optimal fuel economy for the trip can be achieved by operating solely in the CD mode [8]. Here, we present evidence that it is not always the case, and that the CD + CS operation can result in optimal fuel

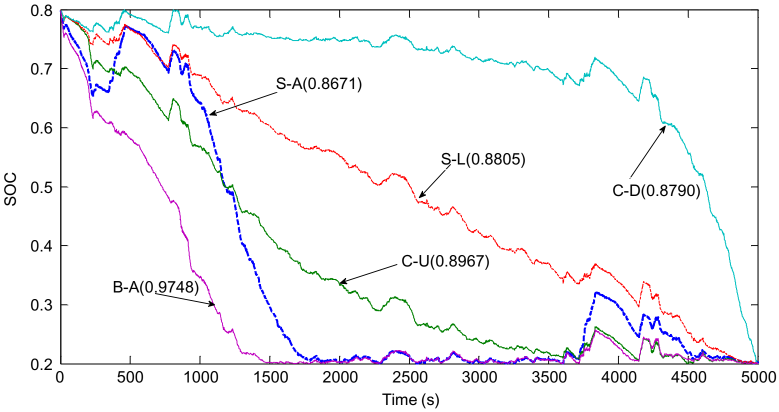

FIG. 2.15 Resultant SOC curve when trip duration is 5000 s.

efficiency for long trips. Furthermore, this finding also implies the potential for the proposed S-A strategy to adapt to different trip durations.

2.4.9 Performance With Charging Opportunity

Considering the plug-in capability of PHEVs, we evaluate the performance of the proposed strategies at the tour level. More specifically, we consider the commute trips of the case study as a tour and assume that there is a charging opportunity (to a full charge) between the end of the westbound trip and the beginning of the eastbound trip. We then compare the different SOC control strategies under the following two scenarios:

(1) Scenario I: The proposed EMS with a priori knowledge of the charging opportunity;

(2) Scenario II: The proposed EMS without a priori knowledge of the charging opportunity. In this case, a conservative strategy is applied by assuming that there is no charging station available in between the trips.

The results are illustrated in Fig. 2.16. They show that the knowledge of the charging opportunity information has great influence on the resultant SOC profiles for the deterministic SOC reference control strategies but no influence on the SOC self-adaptive control strategy. Table 2.6 presents the increased fuel consumption due to the lack of knowledge of the charging opportunity prior to the tour. As shown in the table, the C-D, S-L, and C-U strategies all have 13% or more increase in fuel consumption if the charging opportunity information is unknown, while the B-I and S-A strategies are not affected because the trip duration is not considered in their decision-making process. These findings further emphasize the advantage of the proposed SOC self-adaptive control strategy in terms of robustness to the level of knowledge about charging availability.

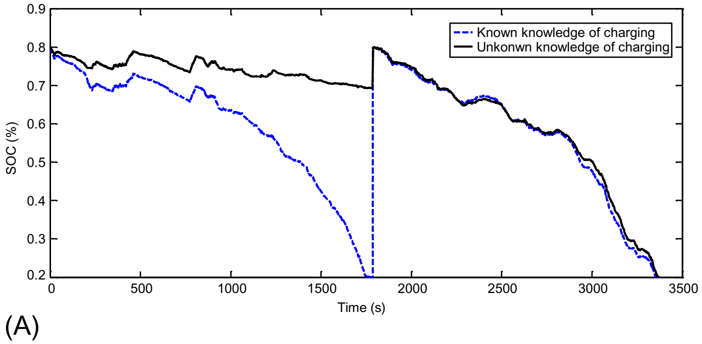

FIG. 2.16 SOC track with known or unknown charging opportunity. (A) C-D.

(Continued)

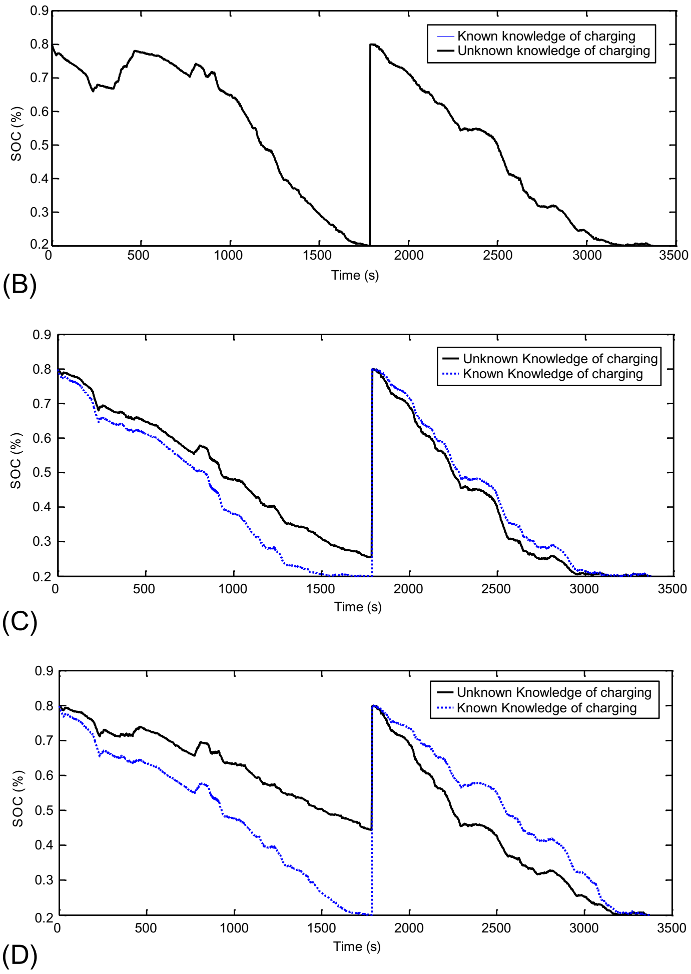

FIG. 2.16, cont'd (B) S-A. (C) C. (D) S-L.

2.5 DATA-DRIVEN REINFORCEMENT LEARNING-BASED REAL-TIME EMS

2.5.1 Introduction

As mentioned in the previous section, the EMS is at the heart of PHEV fuel economy, whose functionality is to control the power streams from both the ICE and the battery pack, based on vehicle and engine operating conditions. In the past decade, a large variety of EMS implementations have been developed for PHEVs, whose control strategies may be well categorized into two major classes, as shown in Fig. 2.17:

TABLE 2.6 Increased Fuel Consumption

Control Strategy	Known (gal)	Unknown (gal)	Increased Fuel Consumption (%)
B-I	0.9748	0.9748	00.0
B-A	0.7109	0.7543	06.1
C-D	0.6729	0.8439	25.1
S-L	0.6809	0.7853	15.0
C-U	0.7066	0.8034	13.0
S-A	0.6681	0.6681	00.0

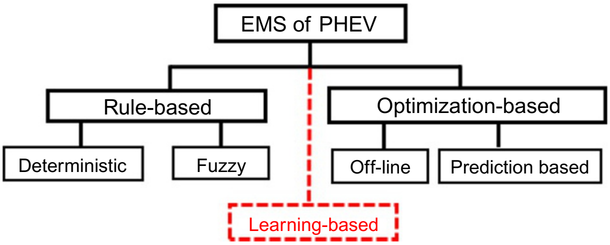

FIG. 2.17 Taxonomy of current EMS.

(a) Rule-based strategies rely on a set of simple rules without a priori knowledge of driving conditions. Such strategies make control decisions based on instant conditions only and are easily implemented, but their solutions are often far from being optimal due to the lack of consideration of variations in trip characteristics and prevailing traffic conditions; and.

(b) Optimization-based strategies are aimed at optimizing some predefined cost function according to the driving conditions and vehicle's dynamics.

The selected cost function is usually related to the fuel consumption or tailpipe emissions. Based on how the optimization is implemented, such strategies can be further divided into two groups: (1) off-line optimization which requires a full knowledge of the entire trip to achieve the global optimal solution; and (2) short-term prediction-based optimization which takes into account the predicted driving conditions in the near future and achieves local optimal solutions segment by segment within an entire trip. However, major drawbacks of these strategies include heavy dependence on the a priori knowledge of future driving conditions and high computational costs that are difficult to implement in real time.

As discussed above, there is a trade-off between the real-time performance and optimality in the energy management for PHEVs. More specifically, rule-based methods can be easily implemented in real-time but are far from being optimal, while optimization-based methods are able to achieve optimal solutions but are difficult to implement in real-time. To achieve a good balance in between, reinforcement learning (RL) has recently attracted researchers' attention. Liu et al. proposed the first and only existing RL-based EMS for PHEVs, which outperforms the rule-based controller with respect to the defined reward function but is worse in terms of fuel consumption without considering charging opportunity in the model.

In this chapter, a novel model-free RL-based real-time EMS of PHEVs is proposed and evaluated, which is capable of simultaneously controlling and learning the optimal power-split operations in real time. The proposed model is theoretically derived from DP formulations and compared to DP in the computational complexity perspective. Three major features distinguish it from existing methods:

(1) The proposed model can be implemented in real-time without any prediction efforts, since the control decisions are made only upon the current system state. The control decisions also considered for the entire trip information by learning the optimal or near-optimal control decisions from historical driving behavior. Therefore, it achieves a good balance between real-time performance and energy saving optimality.

(2) The proposed model is a data-driven model that does not need any PHEV model information once it is well trained, since all the decision variables can be observed and are not calculated using any vehicle powertrain models (these details are described in the following sections).\

(3) Compared to existing RL-based EMS implementations, the proposed strategy considers charging opportunities along the way (a key distinguishing feature of PHEVs as compared with HEVs). This proposed method represents a new class of models that could be a good supplement to the current methodology taxonomy as shown in Fig. 2.17.

2.5.2 Dynamic Programming

The above optimization problem can be solved by DP, since it satisfies Bellman's Principle of Optimality. Let s ϵ S be the state vector of the system, and a ϵ A the decision variable. The optimization problem can be converted into the following single equation given the initial state s_0 and the decisions a_t for each time step t.

$$\min_{a_t \epsilon A} E\left\{\sum_{t=0}^{T-1} \beta^t g(s_t, s_{t+1}) | \, s_0 = s\right\} \tag{2.12}$$

where β is a discount factor and β ϵ (0,1). It can be solved by recursively calculating:

$$J(s_t) = \min_{a_t \epsilon A} E\left\{ \sum_{t=0}^{T-1} g(s_t, s_{t+1}) + \beta J(s_{t+1}) |\ s_t = s \right\}, \text{for}\ t = T-1, T-2, \ldots, 0. \tag{2.13}$$

where T is the time duration, $g(.)$ is a one-step cost function, and $J(s)$ is the true value function associated with states. Eq. (2.13) is also often referred to as Bellman's equation. In the case of PHEV energy management, s_t can be defined as a combination of vehicle states, such as the current SOC level and the remaining time to the destination, which is discussed in the following sections. a_t can be defined as the ICE power supply at each time step.

It is well known that the high computational cost of Eq. (2.13) is always the barrier that impedes its real-world application, although it is a very simple and descriptive definition. It could be computationally intractable even for a small-scale problem (in terms of state space and time span). The major reason is that the algorithm has to loop over the entire state space to evaluate the optimal decision for every single step. Another obvious drawback in the real-world application of DP is that it requires the availability of the full information of the optimization problem. In our case, it means the power demand along the entire trip should be known prior to the trip, which is always impossible in practice.

2.5.3 Approximate Dynamic Programming and Reinforcement Learning

To address the above issues, approximate dynamic programming (ADP) has been proposed (Eq. 2.13). The major contribution of ADP is that it significantly reduces the state space by introducing an approximate value function $\hat{J}(s_t, p_t)$ where p_t is a parameter vector. By replacing this approximate value function, Eq. (2.13) can be reformulated as

$$\hat{J}(s_t) = \min_{a_t \epsilon A} E\left\{ \sum_{t=0}^{T-1} g(s_t, s_{t+1}) + \beta \hat{J}(s_{t+1}, p_t) \right\}, \text{for}\ t = 0, 1, \ldots, T-1 \tag{2.14}$$

Now the optimal decision can be calculated at each time step t by

$$a_t = \arg\min_{a_t \epsilon A} E\left\{ \sum_{t=0}^{T-1} g(s_t, s_{t+1}) + \beta \hat{J}(s_{t+1}, p_t) \right\}, \tag{2.15}$$

The calculation of Eq. (2.15) now only relies on the current system state s_t, which substantially reduces the computational requirement of Eq. (2.13) to polynomial time in terms of the number of the state variables, rather than being exponential to the size of state space. In addition, the value iteration that solves the DP problem becomes forward into time, rather than being backward in Eq. (2.13). In the case of PHEV energy management, this is actually a bonus since the predicted state (e.g., power demand) at the end of the time horizon is much less reliable compared to that at the beginning of the time horizon.

In principle, the value approximate function can be learned by tuning and updating the parameter vector p_t upon the addition of each observation on state transitions. RL is an effective tool for this purpose. The specific learning technique employed in this chapter is temporal-difference learning (TD-Learning), which was originally proposed by Sutton to approximate the long-term future cost as a function of current states. The details about the implementation of the algorithm are presented in the following sections.

2.5.4 Reinforcement Learning-Based EMS

In this chapter, a TD-learning strategy is adopted for the RL problem. An action-value function: Q (s, a) is defined as the expected total reward for the future receipt starting from that state. This function is to estimate "how good" it is to perform a given action in a given state in terms of the expected return. More specifically, we define $Q^{\pi}(s,a)$ as the value of taking action a in state s under a control policy π (i.e., a map that maps the optimal action to a system state), which is also the expected return starting from s, taking the action a, and thereafter following policy π:

$$Q^{\pi}(s,a)=E_{\pi}\left\{\sum_{k=1}^{\infty}\gamma^{k}*r(s_{t+k},a_{t+k})\middle|s_t=s,a_t=a\right\} \tag{2.16}$$

where s_t is the state at time step t; γ is a discount factor in (0, 1) to guarantee the convergence; $r(s_{t+k}, a_{t+k})$ is the immediate reward based on the state s and action a at a given time step ($t + k$). The ultimate goal of RL is to identify the optimal control policy that maximizes the above action-value function for all the state-action pairs.

Comparing to the formulations defined by Eqs. (2.13) and (2.14), the policy π is the ultimate decision for each time step along the entire time horizon. The reward function $r(s_{t+k},a_{t+k})$ here is $g(.)$ in Eq. (2.13). The action-value function (i.e., $Q(s, a)$) is actually an instantiation of the approximate value function $\hat{J}(s_t)$. Thus, it is not difficult to understand that the DP formulas are the basis for a RL problem.

Conceptually, a RL system consists of two basic components: a learning agent and an environment. The learning agent interacts continuously with the environment in the following manner: at each time step, the learning agent receives an observation on the environment state. The learning agent then chooses an action that is subsequently input to the environment. The environment then moves to a new state due to the action, and the reward associated with the transition is calculated and fed back to the learning agent. Along with each state transition, the agent receives an immediate reward and these rewards are used to form a control policy that maps the current state to the best control action

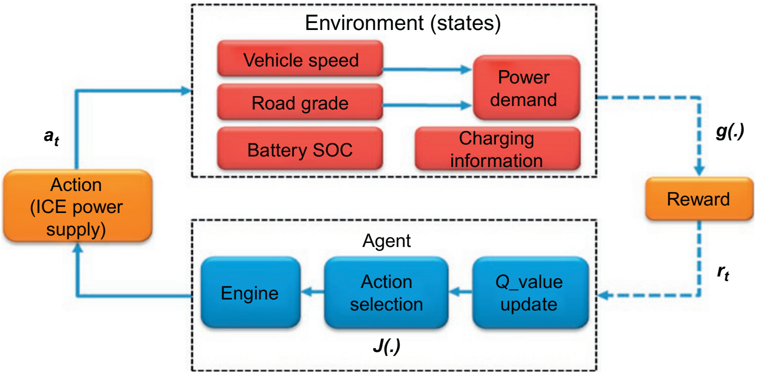

FIG. 2.18 Graphical illustration of reinforcement learning system.

upon that state. At each time step, the agent makes the decision based on its control policy. Ultimately, the optimal policy can guide the learning agent to take the best series of actions in order to maximize the cumulated reward over time that can be learned after sufficient training. Fig. 2.18 gives a graphical illustration of the learning system. The definition of the environmental states, actions, and reward are provided as follows.

2.5.5 Action and Environmental States

In this chapter, we define the discretized ICE power supply level (i.e., P_i^{eng}) as the only action the learning agent can take. The environment states include any other system parameters that could influence the decision of engine power supply. Herein we define a 5-dimensional state space for the environment, including the vehicle velocity (v_{veh}), road grade (g_{road}), percentage of remaining time to destination (t_{togo}), the battery pack's SOC (b_{soc}), and the available charging gain (c_g) of the selected charging station:

$$S = \left\{ s = \left[v_{\mathrm{veh}}, g_{\mathrm{road}}, t_{\mathrm{togo}}, b_{\mathrm{soc}}, c_g \right]^T \mid v_{\mathrm{veh}} \epsilon V_{\mathrm{veh}}, g_{\mathrm{road}} \epsilon G_{\mathrm{road}}, t_{\mathrm{togo}} \epsilon T_{\mathrm{togo}}, \right.$$
$$\left. b_{\mathrm{soc}} \epsilon B_{\mathrm{soc}}, c_c \epsilon C_g \right\}$$

where V_{veh} is the set of discretized vehicle speed level, G_{road} is the set of discretized road grade levels, and P_{brk} is the discretized level of power collected from regenerative braking (note: this power is negative compared to power demand). The minimum and maximum value of vehicle velocity, road grade, and regenerative braking power can be estimated from the historical data of commuting trips, which will be elaborated in the following section. B_{soc} is the set of battery SOC levels between the lower bound (e.g., 20%) and upper bound (e.g., 80%); T_{togo} is the percentage (10%–90%) of remaining time out of the entire trip duration, which is calculated based on the remaining distance

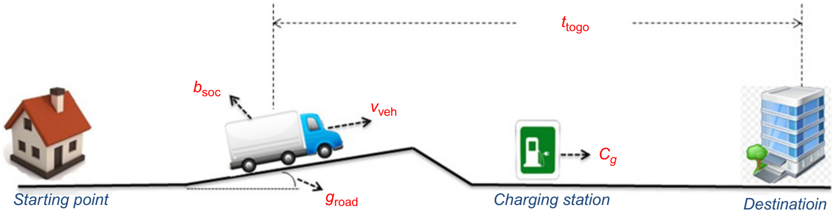

FIG. 2.19 Illustration of environment states along a trip.

to destination. C_g is the set of discretized charging gain (e.g., 30%, 60%) of the selected charger. This charging gain represents the availability of the charger, which may be a function of the charging time and charging rate, and is assumed to be known beforehand. It is noteworthy that all the states can be measured and updated in real time as the vehicle is running. Fig. 2.19 shows all the real-time environmental states.

2.5.6 Reward Initialization (With Optimal Results From Simulation)

The definition of reward is dependent upon the control objective, which is to minimize the fuel cost while satisfying the power demand requirement. Hence, we define the reciprocal of the resultant ICE power consumption at each time step as the immediate reward. A penalty term is also included to penalize the situation where the SOC is beyond the predefined SOC boundaries. The immediate reward is calculated by the following equations:

$$r_{ss'}^{a} = \begin{cases} \dfrac{1}{P_{\mathrm{ICE}}} \ \text{if}\ P_{\mathrm{ICE}} \neq 0 \ \text{and}\ 0.2 \leq \mathrm{SOC} \leq 0.8 \\ \dfrac{1}{P_{\mathrm{ICE}} + P} \ \text{if}\ P_{\mathrm{ICE}} \neq 0 \ \text{and}\,(\mathrm{SOC} \leq 0.2 \ \text{or}\ \mathrm{SOC} \geq 0.8) \\ \dfrac{2}{Min_{P_{\mathrm{ICE}}}} \ \text{if}\ P_{\mathrm{ICE}} = 0 \ \text{and}\ 0.2 \leq \mathrm{SOC} \leq 0.8 \\ \dfrac{1}{2{*}P} \ \text{if}\ P_{\mathrm{ICE}} = 0 \ \text{and}\ (\mathrm{SOC} \leq 0.2 \ \text{or}\ \mathrm{SOC} \geq 0.8) \end{cases} \tag{2.17}$$

where $r_{ss'}{}^{a}$ is the immediate reward when state changes from s to s' by taking action a, P_{ICE} is the ICE power supply, P is the penalty value and is set as the maximum power supply from ICE in this chapter, and Min_P_{ICE} is the minimum nonzero value of ICE power supply. This definition guarantees that the minimum ICE power supply (action) which satisfies the power demand as well as SOC constraints can have the largest numerical reward. A good initialization of reward is also critical for the quick convergence of the proposed algorithm. In this case, the optimal or near-optimal results of typical trips obtained from simulation are used as the initial seeds. These optimal or near-optimal results are deemed as the control decisions made by "good drivers" from historical driving. In order to obtain a large number of such good results for algorithm training, an EA is adopted for the off-line full-trip optimization, since EA can provide multiple solutions for one single run. These solutions are of different quality, which may well represent different levels of driving proficiency in a real-world situation.

2.5.7 Q-Value Update and Action Selection

In the algorithm, a *Q* value, denoted by *Q* (*s*, *a*), is associated with each possible state-action pair (*s*, *a*). Hence there is a *Q* table which is updated during the learning process and can be interpreted as the optimal control policy that the learning agent is trying to learn. At each time step, the action is selected upon this table after it is updated. The details of the algorithmic process are given in the following pseudo code:

Algorithm: RL based PHEV EMS algorithm

Inputs: Initialization 6-D Q (s, a) table; discount factor $\gamma = 0.5$; learning rate $\alpha = 0.5$; exploration probability $\varepsilon \in (0,1)$; vehicle power demand profile P_d (N time steps)

Outputs: Q (s, a) array; control decisions P_d (T time steps)

```
Initialize $Q$ $(s, a)$ arbitrarily
For each time step $t = 1$: $T$
    Observe current $s_t$ ($v_{\text{veh}}$, $g_{\text{road}}$, $t_{\text{togo}}$, $b_{\text{soc}}$, $C_{\text{g}}$)
    Choose action $a_t$ for the current state $s_t$:
            temp = random (0,1);
            if temp $<= \varepsilon$
                $a_t = \arg\max_{a \epsilon A} \{ Q(s_t, a)\}$
            else
                $a_t =$ randomly choose an action;
            end
        Take action $a_t$, observe next state $s_{t+1}$ ($P_{t+1}$, $SOC_{t+1}$)
    if $SOC_{t+1} < 0.2$
        Switch into charging-sustaining mode;
        Give big penalty to $r_t$ according to Eq. (2.10)
    else
        Calculate reward $r_t$ according to Eq. (2.10)
    end
    Update $Q$ $(s_t, a_t)$ with following value:
      $Q(s_t, a_t) + \alpha\{r_t + \gamma * \max_{a_{t+1}} \{Q(s_{t+1}, a_{t+1})\} - Q(s_t, a_t)\}$
end
```

2.5.8 Validation and Testing

The proposed model is then evaluated with real-world data in two different scenarios: one without charging opportunities and the other with charging opportunities.

To obtain a series of real trip data (second-by-second velocity trajectories), we apply the trajectory synthesis technique proposed in our previous work to the inductive loops detector (ILD) data archived in the California Freeway Performance Measurement System (PeMS).

The trajectory synthesis is a two-step process: (1) estimating average velocity by applying 2-dimensional interpolation method to real-world traffic data (e.g., volumes and occupancy) collected from ILDs; and (2) generating random velocity disturbance based on representative driving cycles from the MOVES (MOtor Vehicle Emission Simulator) database. Real traffic data were collected at the I-210 freeway segment between I-605 and Day Creek Blvd in southern California, starting at 8:00 a.m. in the morning (westbound) and returning at 4:00 p.m. in the afternoon every weekday during the period between January 9, 2012 and January 17, 2012. Twelve trips (including eastbound and westbound) were generated in total. The road grade information was also synchronized with the trip data to estimate the second-by-second power demands. For more detailed information on the trajectory synthesis and power demand profile generation, please refer to [15].

2.5.9 Model Without Charging Opportunity (Trip Level)

To validate the proposed strategy, the model without considering charging opportunity is first trained and tested with trips where there is no charging opportunity within the trip. Data for multiple westbound trips described in [15] are used for training. Although it has been proven that Q-learning is guaranteed to converge mathematically, an experimental analysis of convergence is conducted in this chapter. In the experiment, the trip data for the first 6 days are concatenated one by one to form a single training cycle. The proposed model is trained with repeated training cycles. At the end of each training cycle, the trained model is tested with the 7th day trip, and the fuel consumption is recorded in Fig. 2.20. In addition, the training with or without good initialization using simulated optimal or near-optimal solution are also compared. As we can see in the figure, there is a clear convergence in fuel consumption for both cases. However, the initialization with simulated optimal or near-optimal solutions help achieve a faster convergence.

As previously described, the selected state space is 5-dimensional and the action space has 1 dimension. Therefore the Q (s, a) table is 6-dimensional. Fig. 2.21 shows the 4-D slice diagram of the learned Q (s, a) table in which different color grids represent different numerical reward values (e.g., darker color means lower values) and three slices on the (ICE power supply, power demand)

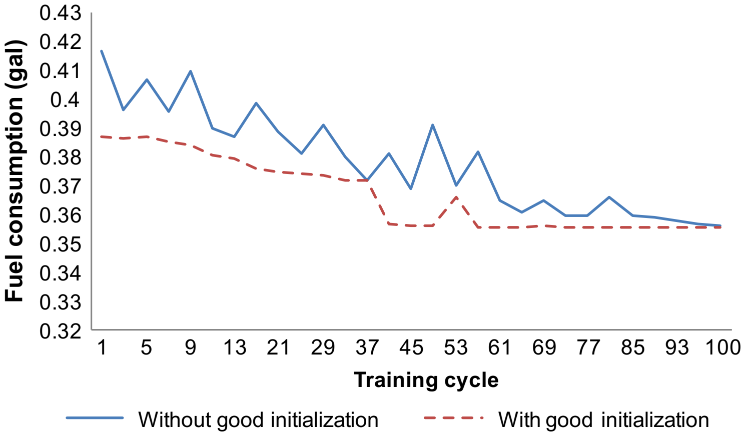

FIG. 2.20 Convergence analysis (ε =0.7; $\gamma = 0.5$; $\alpha = 0.5$).

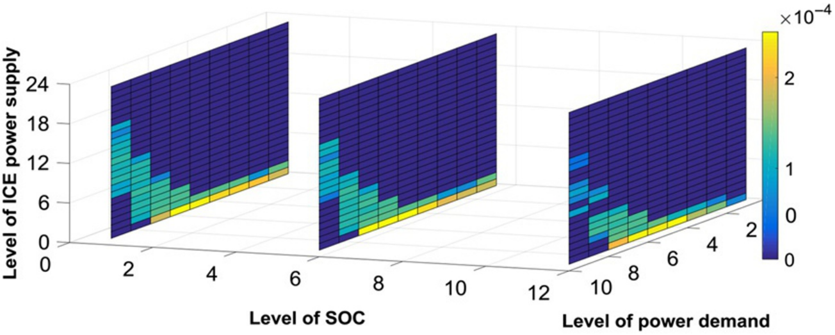

FIG. 2.21 4-D slice diagram of the learned Q table.

space are given at three different SOC levels: 1, 6, and 12 (i.e., 20%, 50%, and 80%). Please note that the road grade and vehicle speed are implicitly aggregated into power demand. The dimension of remaining time is not indicated in the figure. As can be observed in each slice, when the power demand is not so high (e.g., below level 5), action level 1 or 2 is usually the most appropriate because the least ICE power is consumed. When the power demand becomes higher, the range of the feasible action levels gets wider also. In such cases, lower levels of ICE power supply may not be enough to satisfy the power demand and the resultant SOC level could be lower than 0.2, resulting in a penalty defined in Eq. (2.17). It is also noted that when SOC level is high, it is less likely that the higher ICE power supply level would be chosen to satisfy the same power demand. This is because when the vehicle battery SOC is high, the ICE power is not likely to be used aggressively.

As discussed in the previous sections, an exploration-exploitation strategy is adopted for the action selection process to avoid premature convergence. The action with the biggest Q value has a probability of $1 - \varepsilon$ to be selected. Hence the value of ε may significantly affect the performance of the proposed method. To evaluate such impacts, a sensitively analysis of ε is carried out and illustrated in Fig. 2.8. It can be observed that both the fuel consumption and the resultant SOC curves are very close to those of the binary mode control if the value of ε is small. A possible explanation is that a small ε value indicates a large probability to select the most aggressive action with the biggest Q value (or the lowest levels of ICE power supply). Therefore, the battery power is consumed drastically as it is with the binary mode control. However, if the value of ε is too large (e.g., >0.8), the battery power is utilized too conservatively where the final SOC is far away from the lower bound, resulting in much greater fuel consumption. It is found that the best value of ε in this chapter is around 0.7, which ensures the SOC curve is quite close to the global optimal solution obtained by the off-line DP strategy. With this best ε value, the fuel consumption is 0.3559 gal, which is 11.9% less than that of the binary mode control and only 2.86% more than that of DP strategy as shown in Fig. 2.8. This also implies that an adaptive strategy for determining exploration rate along the trip could be useful. Fig. 2.9A shows a linearly decreasing control of ε along the trip. A smaller ε is preferred at the later stage of the trip because SOC is low and the battery power should be consumed more conservatively. With this adaptive strategy for ε, the proposed mode could also achieve a good solution with 0.3570 gal of fuel consumption, which is 11.7% less than the binary control shown in Fig. 2.22.

2.5.10 Model With Charging Opportunity (Tour Level)

The most distinctive characteristics of PHEVs from HEVs is that PHEV can be externally charged whenever a charging opportunity is available. To evaluate further the impacts due to charging availability, we include this information

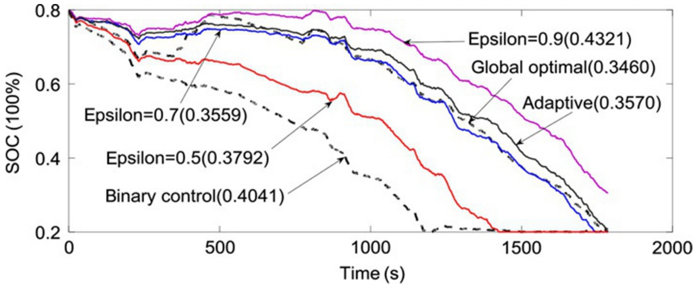

FIG. 2.22 Fuel consumption in gallon *(bracketed values)* and SOC curves by different exploration probabilities.

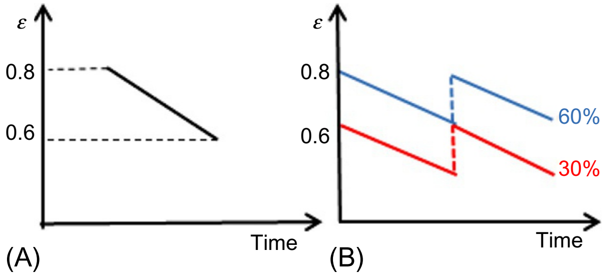

FIG. 2.23 (A) Linear adaptive control of ε; (B) linear adaptive control of ε with charging opportunity.

in the proposed model as a decision variable. For simplicity, the charging opportunity is quantified by the gain in the battery's SOC, which may be a function of available charging time and charging rate. Data for a typical tour are constructed by combining a round trip between the origin and destination. We assume there is a charger in the working place (west-most point in the map) and the available charging gain has only two levels: 30%–60%. In this case, a corresponding adaptive strategy of ε is also used as shown in Fig. 2.23B. The rationale behind this adaptive strategy is that battery power should be used less conservatively (i.e., higher ε value) after charging, and/or when C_g is higher.

The obtained optimal results are shown in Figs. 2.24 and 2.25. As we can see in both figures, the resultant SOC curves are much closer to the global optimal solutions obtained by DP than the binary control. To obtain a statistical significance of the performance, the proposed model is tested with 30 different trips

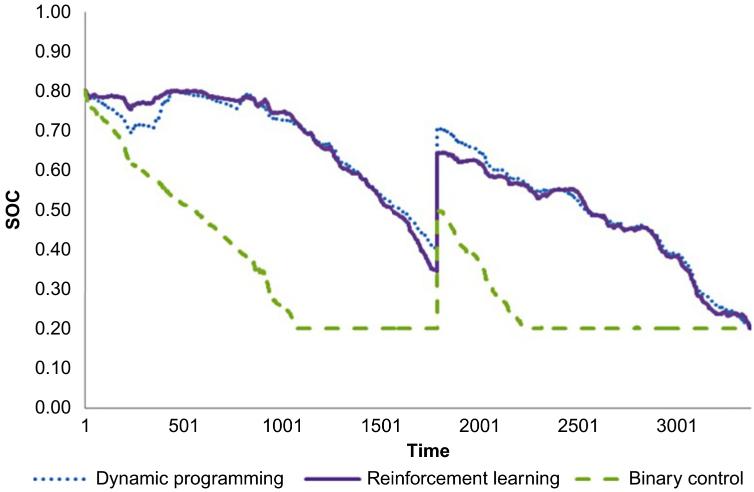

FIG. 2.24 Optimal results when available charging gain is 0.3 ($C_g = 0.3$).

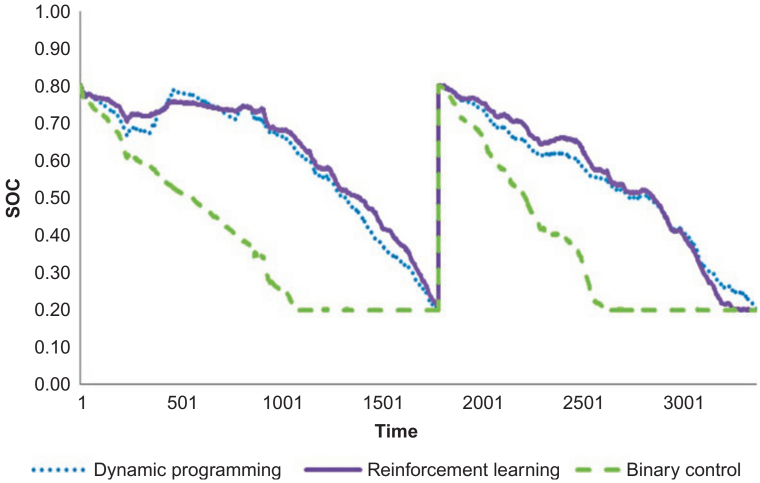

FIG. 2.25 Optimal results when available charging gain is 0.6 ($C_g = 0.6$).

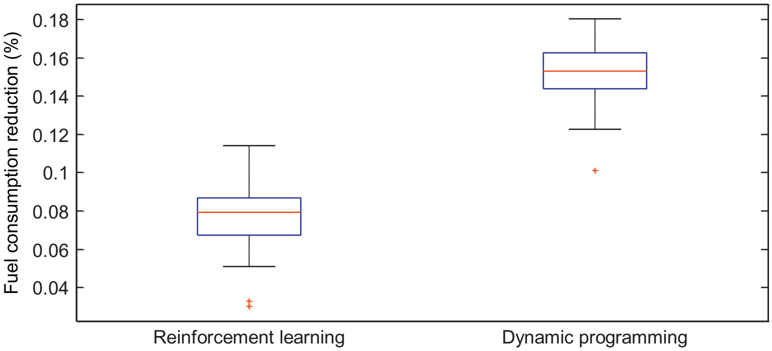

FIG. 2.26 Fuel consumption reduction compared to binary control.

by randomly combining two trips and a charging station is assumed in between with a random C_g (randomly choose from 30% to 60%). By taking the binary control as baseline, the reduced fuel consumption is given in Fig. 2.26. As we can see in the figure, RL model achieves an average of 7.9% fuel savings. It seems that having more information results in lower fuel savings, which is counterintuitive. The reason is that the inclusion of additional information or state variable to the model exponentially increases the search space of the problem, which thereby increases the difficulty of learning the optimal solution. In addition, more uncertainty is introduced to the learning process due to the errors within the added information, which degrades the quality of the best solution the model can achieve.

2.6 CONCLUSIONS

In this chapter, two different data-driven on-line EMS for PHEVs, (i.e., EA based EMS and RL based EMS) are presented with both system architecture and real-world validation.

For the EA-based EMS, the proposed framework applies the self-adaptive strategy to the control of the vehicle's SOC in a rolling horizon manner for the purpose of real-time implementation. The control of the vehicle's SOC is formulated as a combinatory optimization problem that can be efficiently solved by the EDA. The proposed EMS is evaluated comprehensively using a number of trip profiles extracted from real-world traffic data. The results show that the self-adaptive control strategy used in the proposed system statistically outperforms the conventional binary control strategy with an average of 10.7% fuel savings. The sensitivity analysis reveals that the optimal prediction horizon window of the proposed EMS is 250 s, which requires 5.8 s of computation time in our study case. This amount of time is much less than the optimal control horizon window of 10 s, which confirms the feasibility of real-time implementation. Another important advantage of the proposed EMS is that, unlike other existing systems, it does not require a priori knowledge about the trip duration. This allows the proposed system to be robust against real-world uncertainties, such as unexpected traffic congestion that increases the trip duration significantly, and changes in intertrip charging availability.

For RL-based EMS, it is capable of simultaneously controlling and learning the optimal power-split operation. The proposed EMS model is tested with trip data (i.e., multiple speed profiles) synthesized from real-world traffic measurements. Numerical analyses show that a near-optimal solution can be obtained in real-time when the model is well trained with historical driving cycles. For the study cases, the proposed EMS model can achieve better fuel economy than the binary mode strategy by about 12%–8% at the trip level and tour level (with charging opportunity), respectively. The possible topics for future work are: (1) propose a self-adaptive tuning strategy for exploration-exploitation (ε); (2) test the proposed model with more real-world trip data which could include other environmental states, such as the position of charging stations; and (3) conduct a robustness analysis to evaluate the performance of the proposed EMS model when there is error present in the measurement of environment states.

REFERENCES

[1] U.S. Department of Transportation, http://www.its.dot.gov/research/vehicle_electrification_smartgrid.htm. (Accessed 5 January 2015).

[2] G. Wu, K. Boriboonsomsin, M. Barth, Development and evaluation of an intelligent energy-management strategy for plug-in hybrid electric vehicles, IEEE Trans. Intell. Transp. Syst. 15 (3) (2014) 1091–1100.

[3] S.G. Wirasingha, A. Emadi, Classification and review of control strategies for plug-in hybrid electric vehicles, IEEE Trans. Veh. Technol. 60 (1) (2011) 111–122.

[4] A. Panday, H.O. Bansal, A review of optimal energy management strategies for hybrid electric vehicle, Int. J. Veh. Technol. (2014) 19.

[5] H. Banvait, S. Sohel, Y. Chen, in: A rule-based energy management strategy for plug-in hybrid electric vehicle (PHEV), Proceedings of American Control Conference, St. Louis, MO, June 2009, 2009, pp. 3938–3943.

[6] Q. Gong, Y. Li, in: Trip based optimal power management of plug-in hybrid electric vehicles using gas-kinetic traffic flow model, Proceedings of American Control Conference, Seattle, WA, June 2008, 2008, pp. 3225–3230.

[7] L. Tribiloli, M. Barbielri, R. Capata, E. Sciubba, E. Jannelli, G. Bella, A real time energy management strategy for plug-in hybrid electric vehicles based on optimal control theory, Energy Procedia 45 (2014) 949–958.

[8] C. Hou, L. Xu, H. Wang, M. Ouyang, H. Peng, Energy management of plug-in hybrid electric vehicles with unknown trip length, J. Frankl. Inst. 352 (2) (2015) 500–518.

[9] M. Vajedi, M. Chehrehsaz, N.L. Azad, Intelligent power management of plug-in hybrid electric vehicles, part I: real-time optimum SOC trajectory builder, Int. J. Electr. Hybrid Veh. 6 (1) (2014) 46–67.

[10] N. Denis, M.R. Dubois, A. Desrochers, Fuzzy-based blended control for the energy management of a parallel plug-in hybrid electric vehicle, IET Intell. Transp. Syst. 9 (1) (2015) 30–37.

[11] X. Wang, H. He, F. Sun, X. Sun, H. Tang, Comparative study on different energy management strategies for plug-in hybrid electric vehicles, Energies 6 (2013) 5656–5675.

[12] W. Jian, in: Fuzzy energy management strategy for plug-in hev based on driving cycle modeling, Control Conference (CCC), 2014 33rd Chinese, 28–30 July, 2014, pp. 4472–4476.

[13] L. Tribioli, S. Onori, in: Analysis of energy management strategies in plug-in hybrid electric vehicles: application to the GM Chevrolet volt, American Control Conference (ACC), 17–19 June 2013, 2013, pp. 5966–5971.

[14] H. Yu, M. Kuang, R. McGee, Trip-oriented energy management control strategy for plug-in hybrid electric vehicles, IEEE Trans. Control Syst. Technol. 22 (4) (2014) 1323–1336.

[15] Q. Gong, Y. Li, Z.-R. Peng, in: Trip based optimal power management of plug-in hybrid electric vehicles using gas-kinetic traffic flow model, American Control Conference, 11–13 June 2008, 2008, pp. 3225–3230.

[16] M.P. O'Keefe, T. Markel, Dynamic Programming Applied to Investigate Energy Management Strategies for a Plug-In HEV, National Renewable Energy Laboratory, Golden, CO, 2006. Report No. NREL/CP-540-40376.

[17] Z. Chen, C.C. Mi, R. Xiong, X. Jun, C. You, Energy management of a power-split plug-in hybrid electric vehicle based on genetic algorithm and quadratic programming, J. Power Sources 248 (15) (2014) 416–426.

[18] T. Feng, L. Yang, Q. Gu, Y. Hu, T. Yan, B. Yan, A supervisory control strategy for plug-in hybrid electric vehicles based on energy demand prediction and route preview, IEEE Trans. Veh. Technol. PP (99) (2015) 1.

[19] V. Larsson, L. Johannesson Mårdh, B. Egardt, S. Karlsson, Commuter route optimized energy management of hybrid electric vehicles, IEEE Trans. Intell. Transp. Syst. 15 (3) (2014) 1145–1154.

[20] C. Sun, S.J. Moura, X. Hu, J.K. Hedrick, F. Sun, Dynamic traffic feedback data enabled energy management in plug-in hybrid electric vehicles, IEEE Trans. Control Syst. Technol. 23 (3) (2015) 1075–1086.

[21] M. Vajedi, A. Taghavipour, N.L. Azad, J. McPhee, in: A comparative analysis of route-based power management strategies for real-time application in plug-in hybrid electric vehicles, American Control Conference (ACC), 2014, Portland, OR, 2014, pp. 2612–2617.

[22] Z. Chen, W. Liu, Y. Yang, W. Chen, Online energy management of plug-in hybrid electric vehicles for prolongation of all-electric range based on dynamic programming, Math. Probl. Eng. 2015 (2015). 11 pages.

[23] S.J. Moura, H.K. Fathy, D.S. Callaway, J.L. Stein, A stochastic optimal control approach for power management in plug-in hybrid electric vehicles, IEEE Trans. Control Syst. Technol. 19 (3) (2011) 545–555.

[24] C. Liu, Y.L. Murphey, in: Power management for plug-in hybrid electric vehicles using reinforcement learning with trip information, Transportation Electrification Conference and Expo (ITEC), 2014 IEEE, 15–18 June, 2014, pp. 1–6.

[25] X. Lin, H. Banvait, S. Anwar, Y. Chen, in: Optimal energy management for a plug-in hybrid electric vehicle: real-time controller, American Control Conference (ACC), June 30, 2010–July 2, 2010, 2010, pp. 5037–5042.

[26] X. Qi, G. Wu, K. Boriboonsomsin, M.J. Barth, J. Gonder, Data-driven reinforcement learning-based real-time energy management system for plug-in hybrid electric vehicles, Transp. Res. Rec.: J. Transp. Res. Board 2572 (2016) 1, https://doi.org/10.3141/2572-01.

[27] M. Hauschile, M. Pelican, An Introduction and Survey of Estimation of Distribution Algorithms, MEDAL Report No. 2011004, University of Missouri-St. Louis, 2011.

[28] X. Qi, K. Rasheed, K. Li, W.D. Potter, in: A fast parameter setting strategy for particle swarm optimization and its application in urban water distribution network optimal design, The 2013 Int'l. Conf. on Genetic and Evolutionary Methods (GEM), 2013.

[29] X. Qi, Swmarm Intelligence Inspired Engineering Optimization: Concepts, Modeling and Evaluation, Lambert Academic Publishing House, 2014. ISBN: 978-3-659-35681-0.

[30] X. Qi, G. Wu, K. Boriboonsomsin, M.J. Barth, in: An on-line energy management strategy for plug-in hybrid electric vehicles using an estimation distribution algorithm, IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), 2014, pp. 2480–2485. 8–11 October 2014.

[31] A.E. Eiben, Introduction to Evolutionary Computing, Springer, 2007.

[32] P.S. Oliveto, J. He, X. Yao, Time complexity of evolutionary algorithms for combinatorial optimization: a decade of results, Int. J. Autom. Comput. 4 (2007) 281–293.

[33] D. Kum, Modeling and Optimal Control of Parallel HEVs and Plug-In HEVs for Multiple Objectives (Ph.D. Dissertation), University of Michigan, 2010.