A Data-Driven Reinforcement Learning-Based Real-Time Energy Management System for Plug-in Hybrid Electric Vehicles with Charging Opportunities

Xuewei Qi, Ph.D. Candidate
Department of Electrical & Computer Engineering and CE-CERT
University of California at Riverside
1084 Columbia Ave, Riverside, CA 92507, USA
Tel: (951) 781-5620, Fax: (951) 781-5790
E-mail: qixuewei@ece.ucr.edu

Guoyuan Wu, Ph.D.
CE-CERT, University of California at Riverside
1084 Columbia Ave, Riverside, CA 92507, USA
Tel: (951) 781-5630, Fax: (951) 781-5790
E-mail: gywu@cert.ucr.edu

Kanok Boriboonsomsin, Ph.D., P.E.
CE-CERT, University of California at Riverside
1084 Columbia Ave, Riverside, CA 92507, USA
Tel: (951) 781-5792, Fax: (951) 781-5790
E-mail: kanok@cert.ucr.edu

Matthew. J. Barth, Ph.D.
Department of Electrical & Computer Engineering and CE-CERT
University of California at Riverside
1084 Columbia Ave, Riverside, CA 92507, USA
Tel: (951) 781-5782, Fax: (951) 781-5790
E-mail: barth@ece.ucr.edu

Jeffrey Gonder
National Renewable Energy Laboratory
15013 Denver West Parkway, Golden, CO 80401,USA
Tel: (303)-275-4462, Fax: (303)-275-3765
E-mail: Jeff.gonder@nrel.gov

Submitted to 95th Annual Meeting of
Transportation Research Board
Washington, D.C.
January 2016

Total word count: 4950(text) + 2500 (10 figures) =7450
Submitted on August 1st, 2015

Abstract

Plug-in hybrid electric vehicles (PHEVs) show great promise in reducing transportation-related fossil fuel consumption and greenhouse gas (GHG) emissions. Designing an efficient energy management system (EMS) for PHEVs to achieve better fuel economy has been an active research topic for decades. Most of the advanced systems rely on either *a priori* knowledge of future driving conditions to achieve the optimal but not real-time solution (e.g. using a dynamic programming strategy), or only the current driving situation to achieve a real-time but non-optimal solution (e.g. rule-based strategy). Towards this end, this paper proposes a reinforcement learning (RL) based real-time EMS for PHEVs to address the trade-off between real-time performance and optimal energy savings. The proposed model can optimize the power-split control in real time while learning the optimal decisions from historical driving cycles. A case study on a real world commute trip shows that about 12% fuel saving can be achieved without considering charging opportunities; further, a 8% fuel saving can be achieved when considering the charging opportunities, compared to the standard binary mode control strategy.

Key words:

Plug-in Hybrid Electric Vehicle (PHEV), Energy Management System (EMS), Approximate Dynamic Programming, Reinforcement Learning (RL).

1. INTRODUCTION

Reducing transportation-related energy consumption and greenhouse gas (GHG) emissions have been a common goal of public agencies and research institutes for years. In 2013, the total energy consumed by the transportation sector in the United States was as high as 24.90 Quadrillion BTU (1). U.S. Environmental Protection Agency (EPA) reported that nearly 27 % GHG emissions resulted from fossil fuel combustion for transportation activities in 2013 (2). From a vehicle perspective, innovative powertrain technologies, such as hybrid electric vehicles (HEVs), are very promising in improving fossil fuel efficiency and reducing exhaust emissions. Plug-in hybrid electric vehicles (PHEVs) attracted most of the attention due to their ability to also use energy off of the electricity grid, through charging their batteries, thereby achieving even higher overall energy efficiency (3).

The energy management system (EMS) is at the heart of PHEV fuel economy, 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. 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 Figure 1: a) *rule-based strategies* which rely on a set of simple rules without *a priori* knowledge of driving conditions (4 – 7). 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* which are aimed at optimizing some predefined cost function according to the driving conditions and vehicle's dynamics (3, 8 – 18). 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: 1) heavy dependence on the *a priori* knowledge of future driving conditions; and 2) high computational costs that are difficult to implement in real-time.

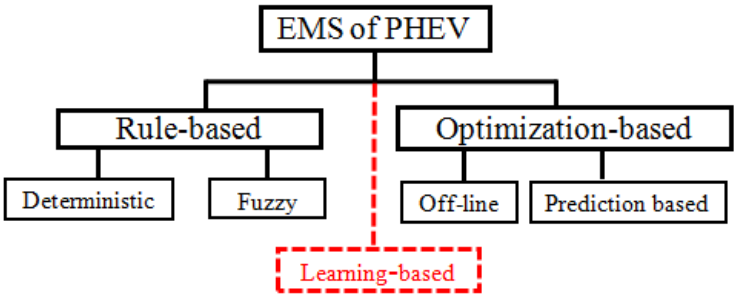

Fig. 1. Taxonomy of current EMS.

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. (20) 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 study, 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 dynamic programming (DP) formulations and compared to DP in the computational complexity perspective. There are three major features which 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 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 (20), 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 Figure 1.

2. BACKGROUND

2.1 PHEV Powertrain and Optimal Energy Management Formulation

There are three types of PHEV powertrain architectures: a) series, b) parallel, and c) power-split (series-parallel) (1). We focus on the power-split architecture in this study. The decision making on the power-split ratio between internal combustion engine (ICE) and battery pack is called power-split control problem (21). Mathematically, the optimal energy management (i.e., power-split control) for PHEVs can be defined as a nonlinear constrained optimization problem (21). In this study, we discretize ICE power supply into different levels and the optimal PHEV power-split control problem therefore can be formulated as follows:

$$min \sum_{t=1}^{M} \sum_{i=1}^{N} x(t,i)\, P_i^{eng} / \eta_i^{eng} \quad (1)$$

subject to:

$$\sum_{t=1}^{j} f\left(P_t - \sum_{i=1}^{N} x(t,i) P_i^{eng}\right) \le C \quad \forall j = 1, \dots, T \quad (2)$$

$$\sum_{i=1}^{N} x(t,i) = 1 \qquad \forall t \quad (3)$$

$$x(t,i) = \{0,1\} \qquad \forall t, i \quad (4)$$

where M is the time span of the entire trip; N is the number of discretized power level for the engine; t is the time step index; i is the ICE power level index; C is the gap of the battery pack's state of charge (SOC) between the initial and the minimum; P_i^{eng} is the i-th discretized level for the engine power and η_i^{eng} is the associated engine efficiency; and P_t is the driving demand power at time step t. The objective of the energy management problem is to find the optimal action (i.e. selection of the optimal ICE power level) for each time step to achieve the best fuel

efficiency along the entire trip.

2.2 Dynamic Programming

The above optimization problem can be solved by dynamic programming (DP), since it satisfies the *Bellman's Principle of Optimality* (22). Let $s \in S$ be the state vector of the system, and $a \in A$ the decision variable. The optimization problem represented by Eq. (1) – (4) can be converted into the following single equation given the initial state s_0 and the decisions a_t for each time step *t*.

$$\min_{a_t \epsilon A} E\left\{\sum_{t=0}^{T-1} \beta^t g(s_t, s_{t+1}) | s_0 = s\right\} \quad (5)$$

where β is a discount factor and $\beta \in (0,1)$. And it can be solved by recursively calculating:

$$J(s_t) = \min_{a_t \epsilon A} E\left\{\sum_{t=0}^{T-1} g(s_t, s_{t+1}) + \beta J(s_{t+1}) | s_t = s\right\}, for\ t = T-1, T-2, \dots, 0. \quad (6)$$

Where T is the time duration; *g(.)* is a one-step cost function; *J(s)* is the true value function associated with state s . Eq. (6) is also often noted as the *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. (6) 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.

2.3 Approximate Dynamic Programming and Reinforcement Learning

To address the above issues, approximate dynamic programming (ADP) has been proposed (23). 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. (6) can be reformulated as:

$$\hat{J}(s_t) = \min_{a_t \epsilon A} E\left\{\sum_{t=0}^{T-1} g(s_t, s_{t+1}) + \beta \hat{J}(s_{t+1}, p_t)\right\}, for\ t = 0,1, \dots, T-1 \quad (7)$$

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

$$a_t = arg \min_{a_t \epsilon A} E\left\{\sum_{t=0}^{T-1} g(s_t, s_{t+1}) + \beta \hat{J}(s_{t+1}, p_t)\right\}, \quad (8)$$

The calculation of Eq. (8) now only relies on the current system state s_t, which substantially reduces the computational requirement of Eq. (6) to polynomial time in terms of the number of

the state variables, rather than being exponential to the size of state space (24). In addition, the value iteration that solves the DP problem becomes forward into time, rather than being backward in Eq. (6). 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 (24). The Reinforcement Learning (RL) is an effective tool for this purpose. The specific learning technique employed in this study is *temporal-difference* learning (TD-Learning), which is originally proposed by Sutton (25) 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.

3. REINFORCEMENT LEARNING BASED EMS

In this study, a TD-learning strategy is adopted for the reinforcement learning 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:

$$Q^{\pi}(s,a)=E_{\pi}\{\sum_{k=1}^{\infty}\gamma^{k} * r(s_{t+k}, a_{t+k})|s_t = s, a_t = a\} \quad (9)$$

where s_t is the state at time step t; γ is a discount factor in (0, 1) to guarantee the convergence(26); $r(s_{t+k}, a_{t+k})$ is the immediate reward based on the state s and action a at a given time step ($t+k$). The ultimate goal of reinforcement learning 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 eq (6) and (7), 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 (6). The action-value function (i.e., Q(s,a)) is actually an instantiation of the approximate value function $\hat{f}(s_t)$. So, it is not difficult to understand that the DP formulas are the basis for a reinforcement learning 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 which 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 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. A graphical illustration of the learning system is given in Figure 2. The definition of the environmental states, actions and reward are provided as

following:

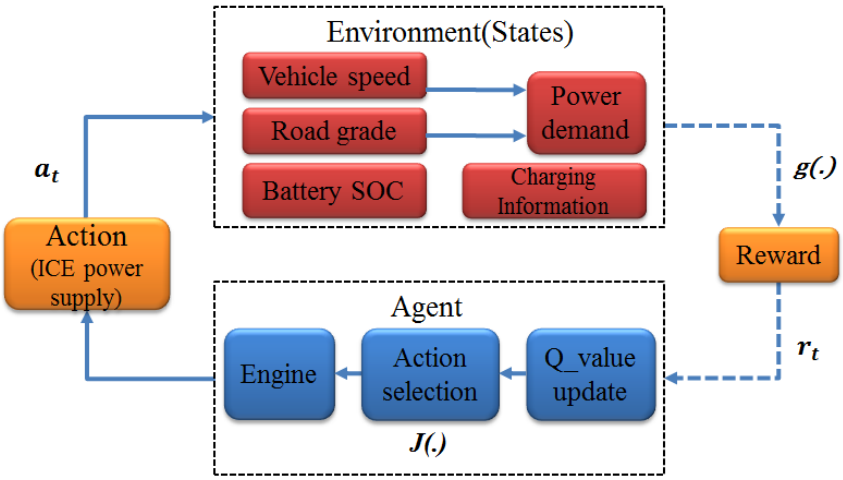

Figure 2. Graphical illustration of reinforcement learning system.

3.1 Action & Environmental States

In this study, we define the discretized ICE power supply level (i.e. P_i^{eng} in Eq. (1)) 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 state-of-charge (b_{soc}), the available charging gain (c_g) of the selected charging station:

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

where V_{veh} is the set of discretized vehicle speed level; G_{road} is the set of discretized road grade levels; 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 state-of-charge (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 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. Figure 3 shows all the real-time *environmental states*.

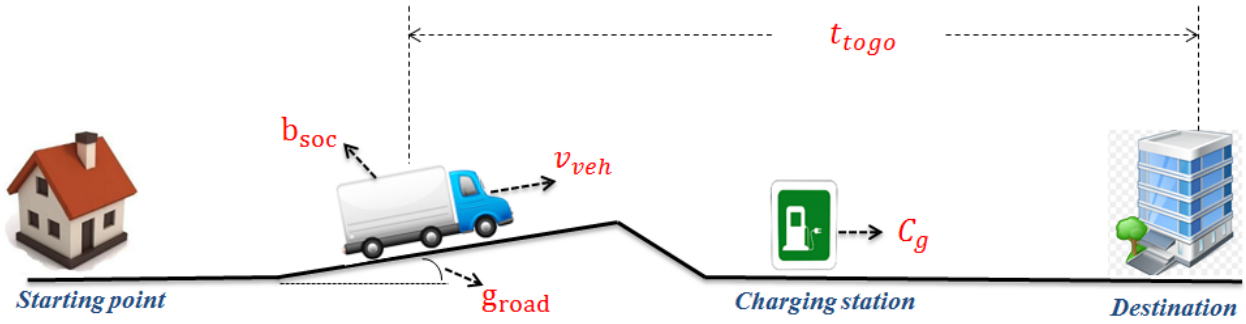

Figure 3. Illustration of environment states along a trip.

3.2 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. Immediate reward is calculated by the following equations:

$$r_{ss'}^{a}=\begin{cases}\frac{1}{P_{ICE}} & if\ P_{ICE}\neq 0\ and\ 0.2\leq SOC\leq 0.8\\ \frac{1}{P_{ICE}+P} & if\ P_{ICE}\neq 0\ and(SOC\leq 0.2\ or\ SOC\geq 0.8)\\ \frac{2}{Min_{P_{ICE}}} & if\ P_{ICE}=0\ and\ 0.2\leq SOC\leq 0.8\\ \frac{1}{2*P} & if\ P_{ICE}=0\ and(SOC\leq 0.2\ or\ SOC\geq 0.8)\end{cases} \quad (10)$$

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 study; 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 evolutionary algorithm (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 level of driving proficiency in the real world situation.

3.3 *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 kept updating 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 γ=0.5; Learning rate α=0.5; Exploration probability ε ϵ(0,1); Vehicle power demand profile P_d (*N* time steps)

Outputs: $Q(s, a)$ array; Control decisions P_d (*T* time steps)

1: Initialize $Q(s, a)$ arbitrarily
2: for each time step t=1:*T*
3: Observe current s_t $(v_{veh},\ g_{road}, t_{togo}, b_{soc}, C_g)$
4: Choose action a_t for the current state s_t:
5: temp=random(0,1);
6: if temp <= ε
7: $a_t = \arg\max_{a \epsilon A}\{ Q(s_t, a)\}$
8: else
9: a_t= randomly choose an action;
10: end
11: Take action a_t, observe next state s_{t+1} (P_{t+1}, SOC_{t+1})
12: if SOC_{t+1}<0.2
13: Switch into Charging-Sustaining mode;
14: Give big penalty to r_t according to Eq. (10)
15: else
16: Calculate reward r_t according to Eq. (10)
17: end
18: Update $Q(s_t, a_t)$ with following value:
19: $Q(s_t, a_t) + \alpha\{r_t + \gamma * \max_{a_{t+1}}\{ Q(s_{t+1}, a_{t+1})\} - Q(s_t, a_t)\}$
20: end

4. CASE STUDY

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

4.1 Data Description

To obtain a series of real trip data (second-by-second velocity trajectories), we apply the trajectory synthesis technique proposed in our previous work (21) to the inductive loops detector (ILD) data archived in the California Freeway Performance Measurement System (PeMS) (26). 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 9th, 2012 and January 17th, 2012. Twelve trips (including eastbound and westbound) are generated in total. The road grade information is 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 (21).

4.2 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 (21) are used for training. Although it has been proven that *Q*-learning is guaranteed to converge mathematically (20), an experimental analysis of convergence is conducted in this study. In the experiment, the trip data for the first six 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 the following Figure 4. 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.

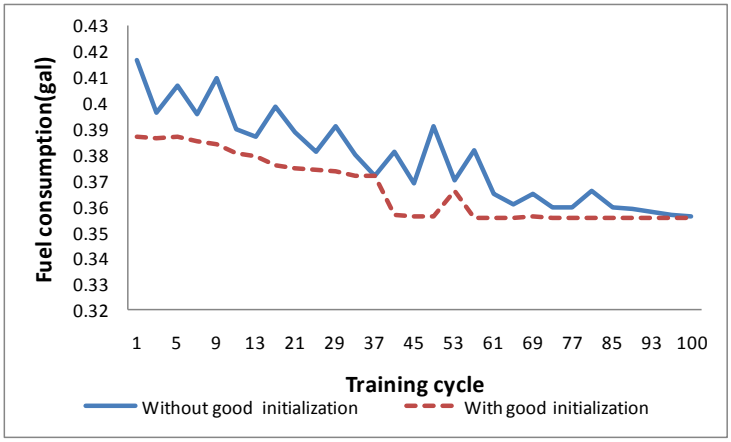

Fig.4. Convergence Analysis (ε =0.7; γ=0.5; α=0.5)

As previous described, the selected state space is 5-dimensional and the action space has 1 dimension. Therefore the *Q*(*s*, *a*) table is 6-dimensional. Figure 5 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., blue color means lower values) and 3 slices on the (ICE power supply, power demand) 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. (10). It is also noted that when SOC level is high, it is less likely 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.

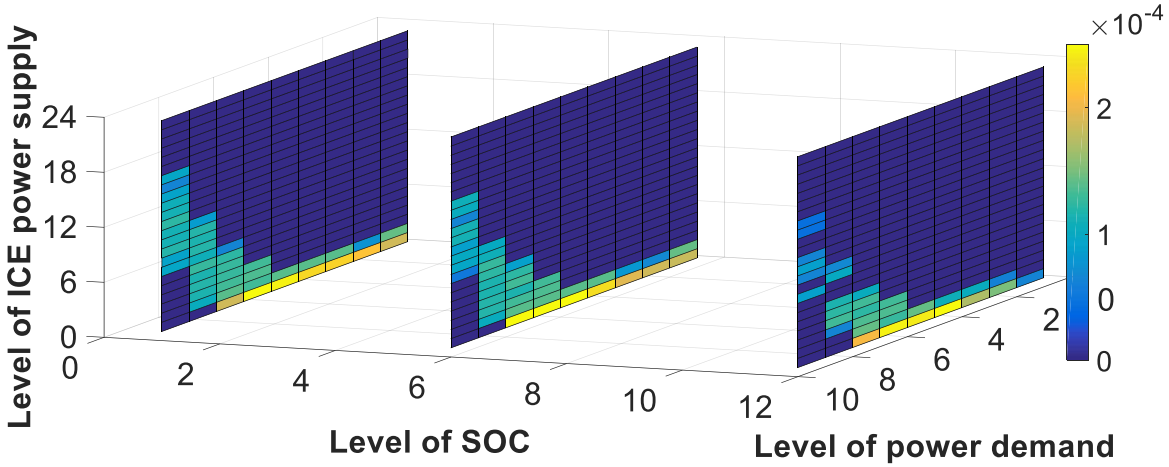

Fig.5. 4-D slice diagram of the learned *Q* table

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-ε 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 Figure 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 study 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 gallon, 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 Figure 8. This also implies that an adaptive strategy for determining exploration rate along the trip could be a useful. Figure 9(a) 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 gallon fuel consumption, which is 11.7% less than binary control shown in Figure 6.

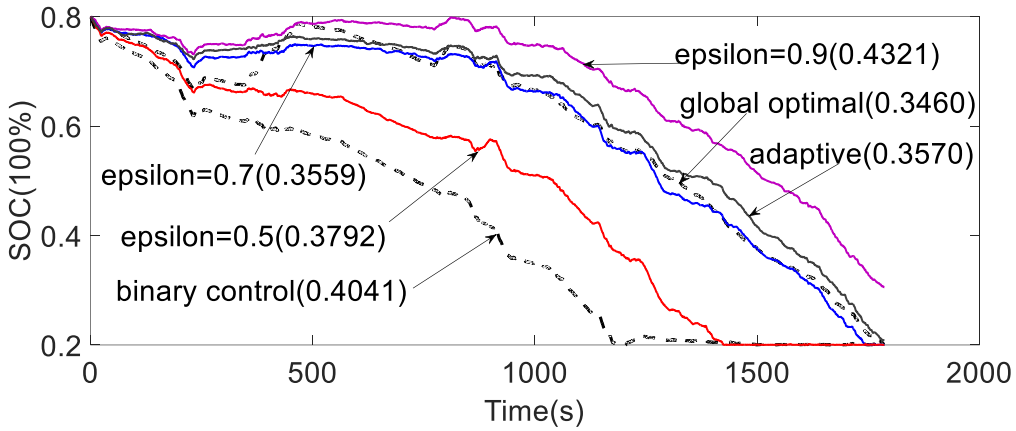

Fig.6. Fuel consumption in gallon (bracketed values) and SOC curves by different exploration probabilities

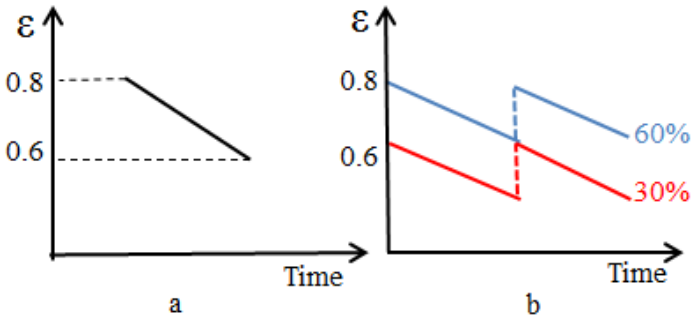

Fig.7. (a) Linear adaptive control of ε; (b) Linear adaptive control of ε with charging opportunity

4.3 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 further evaluate the impacts due to charging availability, we include this information 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 is constructed by combining a round trip between the origin and destination (21). 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% and 60%. In this case, a corresponding adaptive strategy of ε is also used as shown in Figure 7(b). 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.

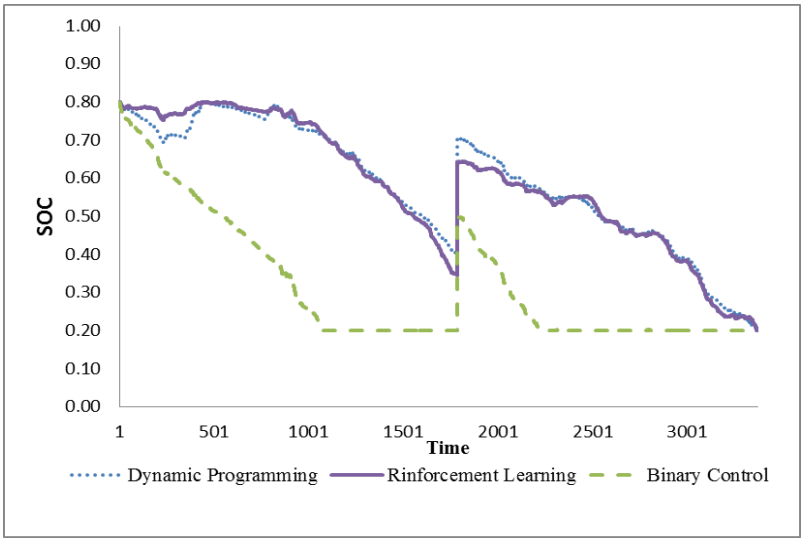

Fig.8. Optimal results when available charging gain is 0.3 (C_g=0.3)

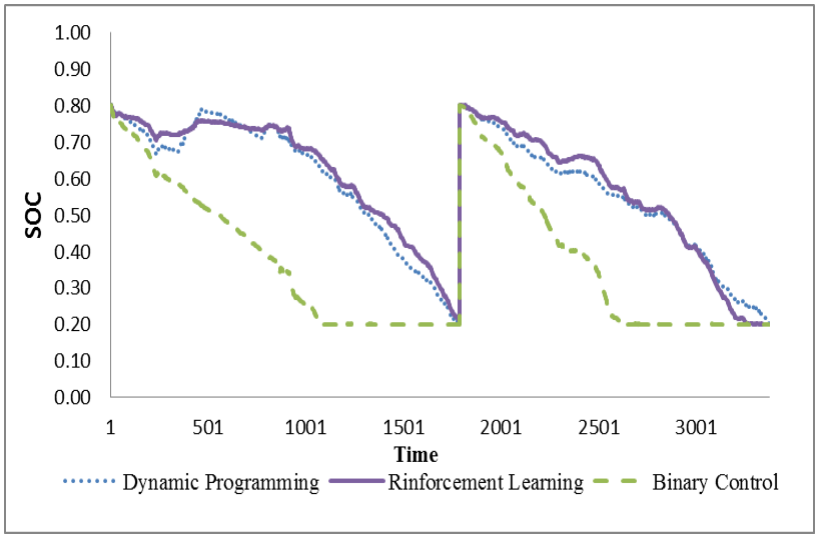

Fig.9. Optimal results when available charging gain is 0.6 (C_g=0.6)

The obtained optimal results are shown in Figure 8 and Figure 9. As we can see in both figures, the resultant SOC curves are much closer to the global optimal solutions obtained by DP than binary control. To obtain a statistical significance of the performance, the proposed model is tested with 30 different trips by randomly combining two trips and assume a charging station in between with a random C_g (randomly choose from 30% and 60%). By taking binary control as baseline, the reduced fuel consumption is given in the following Figure 10. 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. And also 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.

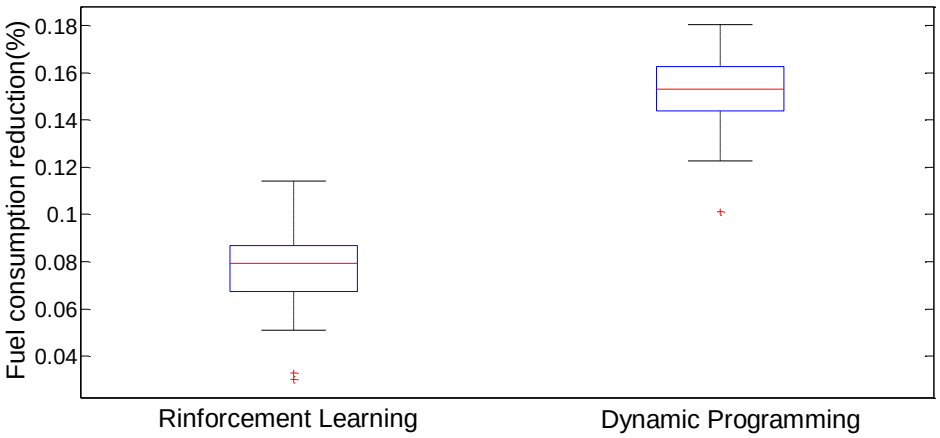

Fig.10. Fuel consumption reduction compared to binary control

5 CONCLUSIONS AND FUTURE WORK

This paper proposes a data-driven reinforcement learning based real-time energy management system for PHEVs, which 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% and 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.

ACKNOWLEDGMENTS

This work was partially supported by the U. S. Department of Transportation.

REFERENCE

[1] Bureau of Transportation Statistics (BTS). Available at: http://www.bts.gov/publications/national_transportation_statistic.

[2] U.S. Environmental Protection Agency (EPA). *DRAFT Inventory of U.S. Greenhouse Gas Emissions and Sinks: 1990 – 2013*. Final report, Feb, 2015.

[3] G. Wu, K. Boriboonsomsin, M. Barth. *Development and Evaluation of an Intelligent Energy-Management Strategy for Plug-in Hybrid Electric Vehicles*. IEEE Transactions on Intelligent Transportation Systems, Vol.15, No.3, June 2014, pp. 1091 – 1100

[4] L. Tribiloli, M Barbielri, R. Capata, E.Sciubba,E.Jannelli and 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.

[5] Denis, N.; Dubois, M.R.; Desrochers, A., *Fuzzy-based blended control for the energy management of a parallel plug-in hybrid electric vehicle*, Intelligent Transport Systems, IET , vol.9, no.1, pp.30,37, 2 2015

[6] Wang X., He, H. Sun, F., Sun, X., Tang,H., *Comparative Study on Different Energy Management Strategies for Plug-In Hybrid Electric Vehicles*, Energies 2013, 6, 5656-5675

[7] Wu J., *Fuzzy energy management strategy for plug-in hev based on driving cycle modeling*, Control Conference (CCC), 2014 33rd Chinese , vol., no., pp.4472,4476, 28-30 July 2014

[8] Tribioli, L.; Onori, S., *Analysis of energy management strategies in plug-in hybrid electric vehicles: Application to the GM Chevrolet Volt*, American Control Conference (ACC), 2013 , vol., no., pp.5966,5971, 17-19 June 2013

[9] Hai Yu; Ming Kuang; McGee, R., *Trip-Oriented Energy Management Control Strategy for Plug-In Hybrid Electric Vehicles*, Control Systems Technology, IEEE Transactions on , vol.22, no.4, pp.1323,1336, July 2014

[10] Qiuming Gong; Yaoyu Li; Zhong-Ren Peng, *Trip based optimal power management of plug-in hybrid electric vehicles using gas-kinetic traffic flow model*, American Control Conference, 2008 , vol., no., pp.3225,3230, 11-13 June 2008

[11] Feng, T.; Yang, L.; Gu, Q.; Hu, Y.; Yan, T.; Yan, B., *A supervisory control strategy for PHEVs based on energy demand prediction and route preview*, Vehicular Technology, IEEE Transactions on , vol.PP, no.99, pp.1,1

[12] Larsson, V.; Johannesson Mårdh, L.; Egardt, B.; Karlsson, S., *Commuter Route Optimized Energy Management of Hybrid Electric Vehicles*, Intelligent Transportation Systems, IEEE Transactions on , vol.15, no.3, pp.1145,1154, June 2014

[13] Xuewei Qi; Guoyuan Wu; Boriboonsomsin, K.; Barth, M.J., *An on-line energy management strategy for plug-in hybrid electric vehicles using an Estimation Distribution Algorithm*, Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on , vol., no., pp.2480,2485, 8-11 Oct. 2014

[14] M. P. O'Keefe and T. Markel, *Dynamic programming applied to investigate energy management strategies for a plug-in HEV*, National Renewable Energy Laboratory, Golden, CO, Report No. NREL/CP-540-40376, 2006.

[15] Zheng Chen, Chris Chunting Mi, Rui Xiong, Jun Xu, Chenwen You, *Energy management of a power-split plug-in hybrid electric vehicle based on genetic algorithm and quadratic programming*, Journal of Power Sources, Volume 248, 15 February 2014, Pages 416-426.

[16] Xiao Lin; Banvait, H.; Anwar, S.; Yaobin Chen, *Optimal energy management for a plug-in hybrid electric vehicle: Real-time controller*, American Control Conference (ACC), 2010 , vol., no., pp.5037,5042, June 30 2010-July 2 2010

[17] Qiuming Gong; Yaoyu Li; Zhong-Ren Peng, *Trip based optimal power management of plug-in hybrid electric vehicles using gas-kinetic traffic flow model*, American Control Conference, 2008 , vol., no., pp.3225,3230, 11-13 June 2008

[18] Cong Hou, Liangfei Xu, Hewu Wang, Minggao Ouyang, Huei Peng, *Energy management of plug-in hybrid electric vehicles with unknown trip length*, Journal of the Franklin Institute, Volume 352, Issue 2, February 2015, Pages 500-518,

[19] Xuewei Qi; Guoyuan Wu; Boriboonsomsin, K.; Barth, M.J., *Evolutionary algorithm based on-line PHEV energy management system with self-adaptive SOC control*, in Intelligent Vehicles Symposium (IV), 2015 IEEE , vol., no., pp.425-430, June 28 2015-July 1 2015. doi: 10.1109/IVS.2015.7225722

[20] J. Liu and H. Peng, *Modeling and Control of A Power-split Hybrid Vehicle*, IEEE Transactions on Control Systems Technology, vol. 16, no. 6, pp. 1242 – 1251, 2008.

[21] G.Wu, K. Boriboonsomsin, M.J.Barth. *Development and Evaluation of an Intelligent Energy-Management Strategy for Plug-in Hybrid Electric Vehicles*. IEEE Transactions on Intelligent Transportation Systems, Vol.15, No.3, 2014, June

[22] Bellman, R. E.. *Dynamic Programming*. Princeton University Press, Princeton, NJ. Republished 2003: Dover, ISBN 0-486-42809-5.

[23] Powell, W. B.. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. 2nd Edition, Wiley, September 2011, ISBN 978-0-470-60445-8

[24] Chen Cai, Chi Kwong Wong, Benjamin G. Heydecker, Adaptive traffic signal control using approximate dynamic programming, Transportation Research Part C: Emerging Technologies, Volume 17, Issue 5, October 2009, Pages 456-474

[25] R. S. Sutton and A. G. Barto. Reinforcement Learning: An Introduction. MIT Press, Cambridge MA, 1998.

[26] California Performance Measurement System (PeMS): http://pems.dot.ca.gov/. Accessed on July 7, 2015