Computationally Efficient Energy Management for a Parallel Hybrid Electric Vehicle Using Adaptive Dynamic Programming

Tong Liu , *Member, IEEE,* Kaige Tan , *Member, IEEE,* Wenyao Zhu , and Lei Feng , *Member, IEEE*

***Abstract*—Hybrid electric vehicles (HEVs) rely on energy management strategies (EMSs) to achieve optimal fuel economy. However, both model- and learning-based EMSs have their respective limitations which negatively affect their performances in online applications. This paper presents a computationally efficient adaptive dynamic programming (ADP) approach that can not only rapidly calculate optimal control actions but also iteratively update the approximated value function (AVF) according to the actual fuel and electricity consumption with limited computation resources. Exploiting the AVF, the engine on/off switch and torque split problems are solved by one-step lookahead approximation and Pontryagin's minimum principle (PMP), respectively. To raise the training speed and reduce the memory space, the tabular value function (VF) is approximated by carefully selected piecewise polynomials via the parametric approximation. The advantages of the proposed EMS are threefold and verified by processor-in-the-loop (PIL) Monte Carlo simulations. First, the fuel efficiency of the proposed EMS is higher than that of an adaptive PMP and close to the theoretical optimum. Second, the new method can adapt to the changed driving conditions after a small number of learning iterations and thus has higher fuel efficiency than a non-adaptive dynamic programming (DP) controller. Third, the computation efficiencies of the proposed AVF and a tabular VF are compared. The concise data structure of the AVF enables faster convergence and saves at least 70% of onboard memory space without obviously increasing the average CPU utilization.**

***Index Terms*—Hybrid electric vehicle, Energy management strategy, Adaptive dynamic programming, Approximated value function.**

I. Introduction

THE urgent demand to reduce energy consumption and exhaust emission dramatically expedites vehicular electrification in contemporary society [1], [2]. Among various new-energy vehicles, the hybrid electric vehicle (HEV), characterized by an extra onboard electric energy storage (EES), e.g., a battery pack or a supercapacitor (SC), along with the traditional fuel tank on its powertrain, has shown fascinating advantages over its counterparts. Thanks to the electric motor (EM), the internal combustion engine (ICE) can either operate within its high-efficiency range or be switched off to save fuel. Hence, the HEV can achieve better fuel economy than the conventional fuel-powered vehicle and has less concern about range anxiety than the pure electric vehicle [3]. However, the dual onboard energy sources add an extra degree of freedom to the powertrain and thus require an appropriate energy management strategy (EMS) to flexibly assign torque demands to the fuel and electric paths for minimal fuel consumption without violating any system requirement [4], [5].

The published EMSs over past decades can be broadly classified into three groups, namely rule-based, optimization-based, and learning-based strategies [6], [7]. Rule-based EMSs, including thermostat (on/off) [8], power follower [9], state machine [10] and fuzzy logic strategy [11], have merits in component variability and system robustness. However, since predefined rules are extracted from heuristic inference and/or human expertise rather than rigorous optimization, these EMSs can hardly ensure close-to-optimal performances or even that all system constraints can be well satisfied.

Based on predefined optimization objectives and system constraints, optimization-based strategies search for optimal or suboptimal solutions by different approaches. Depending on the reliance on future driving information, they can be further divided into global and real-time optimization EMSs. The former subgroup, containing deterministic dynamic programming (DDP) [12], [13], genetic algorithm (GA) [14], simulated annealing (SA) [15], and particle swarm optimization (PSO) [16], usually cannot be directly applied to online applications due to the dependency on complete driving information as well as the enormous computation intensity. Benefiting from the rapid solving process, the latter subgroup, including Pontryagin's minimum principle (PMP) [17], equivalent consumption minimization strategy (ECMS) [18], and model predictive control (MPC) [19], can be utilized in online applications and has obtained favorable results. The performances of optimization-based EMSs, however, are not robust if vehicle models significantly deviate from real powertrain features or the predicted information fails to reflect actual driving scenarios.

To improve the robustness, a slew of learning-based EMSs have been studied recently, such as supervised learning [20], unsupervised learning [21], reinforcement learning (RL) [22], deep reinforcement learning (DRL) [23], and so forth. At each time step, they select a set of control actions and then update control strategies according to the real-time feedback and the accumulated historical information. Therefore, after adequate training in simulation and actual driving environment, they

This work was supported by China Scholarship Council (CSC), KTH Excellence in Production Research (XPRES), and Trustworthy Edge Computing Systems and Applications (TECoSA). *(Corresponding author: Lei Feng)*

Tong Liu, Kaige Tan and Lei Feng are with the Department of Engineering Design, KTH Royal Institute of Technology, Brinellvägen 83, SE-10044, Stockholm, Sweden e-mail: {tongliu, kaiget, lfeng}@kth.se.

Wenyao Zhu is with the Department of Electrical Engineering, KTH Royal Institute of Technology, Electrum 229, SE-16440, Kista, Sweden (e-mail:wenyao@kth.se).

can achieve a competitive performance very close to the optimum. Nevertheless, current learning-based EMSs still have several bottlenecks. For instance, the basic Q-learning method [24] represents Q functions as high-dimensional tables, which incur truncation errors to the control performance and require a large amount of onboard memory space. Approximating the Q table by a deep neural network (DNN), the deep Q-network (DQN)-based EMS [25] can effectively overcome the "curse of dimensionality" but can only output actions with discrete values. The deep deterministic policy gradient (DDPG) method [26] enables control actions in continuous spaces, but relies on four independent neural networks (NNs) and one replay buffer of large size to store historical experience. Such a complicated architecture brings in massive storage occupation and burdensome computation intensity in online applications. Moreover, almost all learning-based EMSs suffer low convergences rates, and their performances are highly dependent on the training database.

Among all aforementioned EMSs, DDP is regarded as the most effective method to realize the global optimum and has been extensively investigated. Since DDP is an offline method, a lot of online dynamic programming (DP) methods have been investigated in recent years [27]. Instead of requiring a precise driving cycle in advance, stochastic DP (SDP) [28] adopts a statistical model to predict future driving information and generates a stationary optimal control policy. The time-invariant and state feedback properties enable SDP to run rapidly online. However, its multi-dimensional control map is usually memory intensive and limits its prevalence. Adaptive DP (ADP)[1] [30] is another alternative, in which the explicit value function (VF) and state transitions are approximated by NNs. ADP can attain a comparable performance as DDP and significantly save the onboard memory, whereas the extra computation overhead for updating NNs is nontrivial in online applications.

In addition to the torque split between the fuel and electric paths, the ICE on/off switch is crucial for fuel economy, especially for parallel HEVs, whose wheel speeds are directly coupled with ICE spinning speeds. Due to the binary property, optimizing the ICE on/off switch together with the torque split simultaneously is time-consuming in online control. Consequently, the majority of research works ignore this binary variable or use heuristic rules to calculate it with ease [31]. The PMP method is utilized to regulate the real-time ICE status but cannot avoid rapid switches [32]. The DDP method is able to solve the optimal ICE switch problem with a receding horizon, but the computation load is too large [33].

In summary, the complex nonlinearity and non-convexity of the HEV powertrain impose enormous challenges on the development of advanced online EMSs. To the best of our knowledge, the majority of current research concentrates on whether the numeric results attained by a newly developed EMS are better than those by the benchmark methods, while little attention has been paid to how much computation resource an EMS will occupy and whether it can be implemented as a real-time controller on vehicular onboard processors.

To address these challenges, this paper presents a computationally efficient adaptive dynamic programming (ADP)-based EMS for a parallel HEV to improve its fuel economy. The proposed online EMS contains three interactive modules, namely powertrain mode selection, torque split control, and adaptive learning algorithm. With the aid of VF, the total equivalent fuel consumption in the remaining driving can be forecasted, and the optimal ICE on/off switch determined. If the ICE is switched on, PMP is employed to calculate the torque allocations on ICE and EM. For a close-to-optimal solution with efficient execution in real-time, the Hamiltonian is formulated as a constraint quadratic programming problem, and the costate of PMP is derived from the VF. To avoid the "curse of dimensionality", the tabular VF of explicit values is replaced by the approximated value function (AVF) of piecewise polynomials. The AVF parameters are initialized by the optimal VF obtained from offline DDP and then iteratively updated during online usage to overcome the deviation between the model and reality.

Processor-in-the-loop (PIL) simulations based on a low-cost microprocessor have been performed on two different driving routes to verify the following three primary advantages of the proposed ADP method. First, the ADP method achieves a close-to-optimal fuel efficiency, more than 97% of that by offline DDP and at least 5% better than that of an adaptive PMP (APMP). There is no frequent ICE on/off switch during driving, and the final state of charge (SOC) of the SC is close to its initial value. Second, this method can quickly improve itself by adapting to real driving conditions through online learning. The adaptation addresses both model errors and the deviation between prior knowledge and real driving conditions. Thus, it achieves higher fuel efficiency than a non-adaptive DP method. Third, this method adopts a much more compact data structure to represent the AVF than the DDP method directly using tabular VF. Hence, it enjoys a higher learning speed and saves at least 70% of onboard flash memory.

The main contributions that distinguish this paper from previous studies are summarized below.

1. Owing to the nonlinear and discrete dynamics of HEV powertrains, existing ADP-based EMSs approximate the VF by complex NNs to ensure the numeric accuracy, whereas NNs require long learning time and large memory spaces. Our method approximates the VF by piecewise third-order polynomials to reduce the complexity. The VF segmentation is determined by the optimal profile of powertrain mode obtained from offline DDP. To ensure optimality and accelerate the convergence speed, the parameters of piecewise polynomials are initialized by the optimal VF from offline DDP.
2. Online decisions on ICE switch and ICE torque are often calculated by mixed integer optimization or DDP for a limited time horizon in the literature and are thus time-consuming. Our approach constrains the ICE to operate at the peak efficiency point when evaluating its on/off switch command. Then, the ICE on/off switch control becomes a binary problem and can be rapidly solved by one-step lookahead with the AVF.

[1]Adaptive DP is sometimes also called approximate DP since they both employ adaptive critic designs [29].

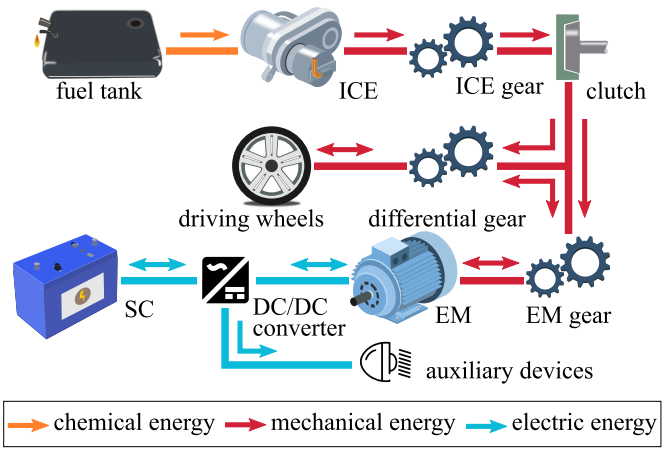


Fig. 1. Parallel HEV Powertrain Architecture

3. Unlike previous PMP methods that calculate the costate by either heuristic rules or partial derivative equations, the optimal costate in this paper is estimated through the AVF for a close-to-optimal solution. In addition, by simplifying the ICE and EM models, the Hamiltonian is reformulated as a convex constraint quadratic programming problem that can be rapidly solved.
4. Only numeric results are discussed in the majority of previous EMS studies, and a small number of researchers merely exhibited the EMS execution time in simulations. This paper systematically investigates the computation efficiency of the proposed EMS through PIL simulations, which measure both numerical results and the time and space complexities of the proposed EMS, including maximum/average CPU utilization and RAM/flash memory consumption.

The rest of this paper is organized as follows: Section II establishes a control-oriented dynamical model of a parallel HEV and its powertrain; Section III converts the HEV energy management problem into a constrained-optimal control problem (OCP); Section IV elaborates the framework of ADP-based EMS; Section V illustrates and discusses the PIL simulation results; and lastly, Section VI draws the main conclusion and raises the future work.

II. Dynamical Model of HEV and Powertrain

The HEV under investigation is a lightweight prototype and has a parallel powertrain depicted in Fig. 1. It consists of two independent propelling components: a petrol-driven ICE and a brushless direct current (BLDC) motor powered by an SC. During driving, the powertrain has two working modes, namely the electric mode when the ICE is off and the clutch is disengaged, and the hybrid mode when the ICE is on and the clutch is engaged. Essential parameters of this HEV are listed in TABLE I. Since the optimization objective is the accumulated energy consumption on a driving route, the quasi-static modeling method is employed to analyze the dynamical characteristics of each powertrain component. The fast dynamics, such as clutch engage/disengage and ICE on/off switch, are neglected.

TABLE I. Essential Parameters of the HEV

Parameter	Sign	Value	Unit
HEV gross mass	M	216	kg
Gravitational acceleration	g	9.81	$kg{\cdot}m{\cdot}s^{-2}$
Rotational mass conversion ratio	δ	1.04	/
Driving wheel radius	r	0.26	m
Windward area	A_f	1.05	m^2
Air drag coefficient	c_d	0.15	$kg{\cdot}m^{-3}$
Rolling resistance coefficient	c_r	0.011	/
ICE gear ratio	R_{ce}	1.23	/
EM gear ratio	R_{em}	1.06	/
Differential gear ratio	R_p	10	/
Lumped efficiency in drive shaft	η_d	0.9	/
Lumped efficiency to recharge SC	η_{rc}	0.25	/
Average SC efficiency	η_{sc}	0.98	/
SC terminal voltage	V_{sc}	40-50	V
SC Nominal capacitance	C	107	F
SC Nominal charge capacity	Q_{sc}	5350	C
ICE maximum torque	$T_{ce}^{\max}$	3.1	Nm
ICE maximum power	$P_{ce}^{\max}$	1.5	kW
EM maximum torque	$T_{em}^{\max}$	11	Nm
EM maximum power	$P_{em}^{\max}$	3.55	kW

A. HEV Longitudinal Model

Suppose that the total driving time t_f for a driving route is uniformly divided into N steps with interval $t_s = t_f/N$. At the k^{th} step, $k \in \{0, 1, ..., N-1\}$, the HEV longitudinal dynamics can be described by,

$$a_k = \frac{1}{\delta M}\left[\frac{T_{t,k}}{r} - \frac{1}{2}A_f c_d v_k^2 - Mg\left(c_r \cos\alpha_k + \sin\alpha_k\right)\right], \quad (1)$$

$$v_{k+1} = v_k + a_k t_s, \quad (2)$$

where a, v, T_t, and α denote the HEV acceleration, speed, net tractive torque on the driving wheel, and road slope angle, respectively. They are assumed fixed within one step t_s.

In the hybrid mode, T_t is supplied by both the ICE torque T_{ce} and the EM torque T_{em}, expressed by,

$$T_{t,k} = R_p\left(T_{ce,k}R_{ce}\eta_d + T_{em,k}R_{em}\eta_d^{sign(T_{em,k})}\right). \quad (3)$$

In the electric mode, (3) still holds with $T_{ce,k}$ equal to 0.

B. ICE Model

The transient fuel consumption by ICE during one step t_s contains two parts, one is the actual fuel consumption m_{ce} for generating driving torque T_{ce}, and another is the equivalent one m_{sw} for powertrain mode switch by switching ICE on/off and dis/engaging clutch. The first part, m_{ce}, is derived as,

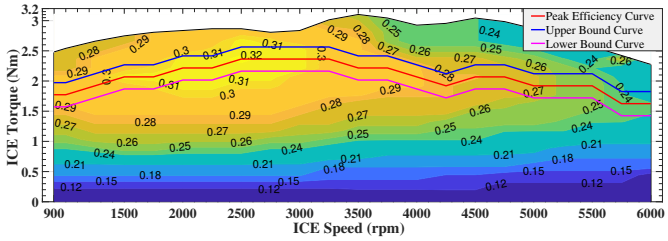
$$m_{ce,k} = \dot{m}_{ce,k} t_s, \quad (4)$$

$$\dot{m}_{ce,k} = \frac{P_{ce,k}}{Q_f} = \frac{T_{ce,k}\omega_{ce,k}}{\eta_{ce}(T_{ce,k}, \omega_{ce,k}) \cdot Q_f}, \quad (5)$$

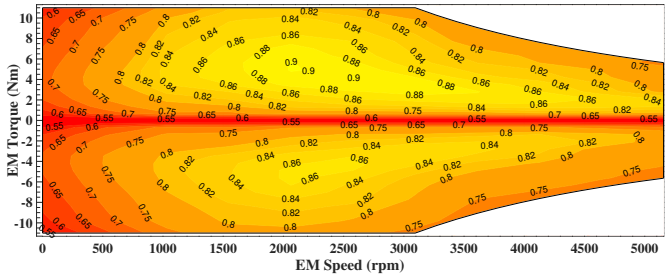
$$\omega_{ce,k} = v_k \frac{R_p R_{ce}}{r}, \quad (6)$$

where $\dot{m}_{ce}$ is the transient fuel consumption rate in grams per second (g/s), P_{ce} is the power consumption by ICE, Q_f is the lower heating value of gasoline, ω_{ce} is the spinning speed of ICE crankshaft after the clutch is engaged, and η_{ce} denotes the ICE net efficiency which is modeled as a 2D map with T_{ce} and ω_{ce} as inputs, shown in Fig. 2(a).

The general approach to obtain η_{ce} is to perform interpolation with T_{ce} and ω_{ce} in the 2D map of meshgrid format.

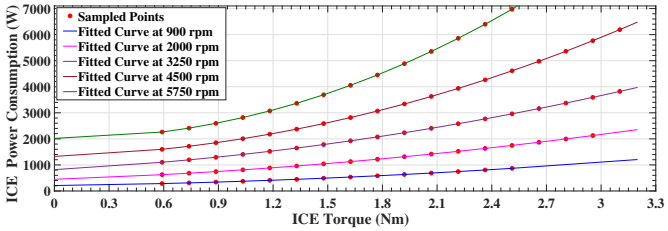


(a) ICE

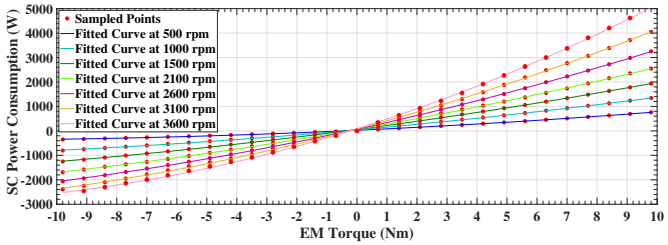


(b) EM

Fig. 2. Actuator Efficiency Maps



(a) ICE



(b) SC

Fig. 3. Power Consumption Approximation

However, this method will result in excessive computation and memory overheads in real-time optimization. Thus, it is unsuitable for developing computationally efficient EMSs. A common solution is to approximate P_{ce} of a given ω_{ce} as a second-order function of T_{ce} [34],

$$P_{ce,k} = p_2(\omega_{ce,k})T_{ce,k}^2 + p_1(\omega_{ce,k})T_{ce,k} + p_0(\omega_{ce,k}), \quad (7)$$

where p_2, p_1, p_0 are fitting coefficients of a specific ω_{ce}. The approximation results of several different ω_{ce} are plotted in Fig. 3(a), and the normalized root mean square error (NRMSE) between approximated efficiency values and real ones in the 2D map is 2.68%.

For better fuel economy, the ICE is expected to operate in the high-efficiency region. In this paper, a narrow band of width ΔT, around the peak efficiency line that is depicted by the red curve in Fig. 2(a), is selected as the admissible ICE operating range. Hence, the upper and lower bounds for T_{ce} at a given ω_{ce} are defined as,

$$T_{ce,k}^{\min} = T_{ce}^{\circ}(\omega_{ce,k}) - \Delta T/2, \quad (8)$$
$$T_{ce,k}^{\max} = T_{ce}^{\circ}(\omega_{ce,k}) + \Delta T/2, \quad (9)$$

where the superscripts $\min$ and $\max$ refer to the upper and lower bounds, T_{ce}° denotes the ICE torque with peak efficiency.

The value of m_{sw} can be either $m^{\star}$ if the powertrain mode is switched at the current step or 0 if it does not occur. Since the actual energy consumption for one mode switch varies a lot under different operation conditions [35], for simplification, $m^{\star}$ is set as the average equivalent fuel consumption of a large number of mode switches under different conditions. The fast dynamics of ICE switch and clutch dis/engagement are neglected in this quasi-static model since they have negligible impact on the analysis of energy consumption. Assume that one switch can be fully carried out within one step t_s. If the current ICE on/off status is represented by a binary variable $s_{ce} \in \{0, 1\}$ ("1" means on concerning the hybrid mode and "0" means off concerning the electric mode) and the ICE on/off command by another binary variable $u_{ce} \in \{0, 1\}$, then m_{sw} can be calculated by the followings,

$$m_{sw,k} = \begin{cases} 0; & s_{ce,k} = u_{ce,k} \\ m^{\star}; & s_{ce,k} \neq u_{ce,k} \end{cases}, \quad (10)$$
$$s_{ce,k+1} = u_{ce,k}, \quad (11)$$
$$s_{ce,k} = 0 \;\Rightarrow\; T_{ce,k} = 0, \quad (12)$$
$$s_{ce,k} = 1 \;\Rightarrow\; T_{ce,k} \in \left[T_{ce,k}^{\min},\, T_{ce,k}^{\max}\right]. \quad (13)$$

C. EM and SC Models

The EM can work in either the actuator mode when T_{em} is positive or the generator mode when T_{em} is negative. Its transient electric power consumption P_{em} is calculated by,

$$P_{em,k} = \frac{T_{em,k}\omega_{em,k}}{\eta_{em}\left(T_{em,k}, \omega_{em,k}\right)^{sign(T_{em,k})}}, \quad (14)$$
$$\omega_{em,k} = v_k \frac{R_p R_{em}}{r}, \quad (15)$$

where ω_{em} is the spinning speed of the EM rotor, and η_{em} is the EM net efficiency dependent on T_{em} and ω_{em}, shown by Fig. 2(b).

An SC is selected as the onboard EES mainly due to its longer life cycles and higher specific power than a battery pack [36]. The net power across the SC P_{sc} is the combination of P_{em} and P_{aux}, i.e, the power to support other onboard auxiliary devices. For simplification, P_{aux} and the SC efficiency η_{sc} are both treated as constants of their average values. Consequently, the SC dynamics can be expressed by (16)-(19),

$$P_{sc,k} = \frac{P_{em,k} + P_{aux}}{\eta_{sc}^{sign(P_{em,k}+P_{aux})}}, \quad (16)$$
$$\dot{V}_{sc,k} = -\frac{P_{sc,k}}{C \cdot V_{sc,k}}, \quad (17)$$
$$V_{sc,k+1} = V_{sc,k} + \dot{V}_{sc,k} t_s, \quad (18)$$
$$SOC_k = \frac{C}{Q_{sc}} V_{sc,k}. \quad (19)$$

Thanks to the linear relationship between V_{sc} and SOC, V_{sc} is employed to indicate the SOC level hereafter. Similar to the ICE model, to improve the computation efficiency for calculating P_{sc}, P_{sc} is also approximated as a second-order function of T_{em},

$$P_{sc,k} = q_2(\omega_{em,k})T_{em,k}^2 + q_1(\omega_{em,k})T_{em,k} + q_0(\omega_{em,k}), \quad (20)$$

where q_2, q_1, and q_0 are fitting coefficients associated to ω_{em}. The approximation results are plotted in Fig. 3(b), with the NRMSE of 4.88%.

III. Optimal Control Problem Statement

The EMS objective for this parallel HEV is to optimally regulate the powertrain mode and allocate torque demands to the ICE and the EM so that the total fuel consumption over a specified driving route can be minimized. Additionally, the final SC voltage is expected to be no less than its initial value; otherwise, the net electricity consumption over the whole driving route will be converted into an equivalent fuel consumption for recharging the SC afterward.

For evaluating EMS performances, standard driving cycles that define time sequences of HEV speed, acceleration, and road grade are typically used, e.g., Japan 10-15, New European Driving Cycle (NEDC), Artemis Urban, and so forth [37]. However, all speed profiles in these cycles cannot properly match road characteristics of the driving routes selected in this paper. The characteristics of a real driving route, including explicit information on geometry, altitude, and driving distance, significantly impact the HEV fuel economy.

The essential task of an optimal EMS is to find the optimal speed trajectory for a specific HEV on a given route with knowledge of the length-altitude profile to minimize the overall fuel consumption under the constraints of safe speed, maximum driving time, and actuator limits. This problem has been solved by offline DDP with a distance-based state update model, in which the HEV speed and accumulated driving time are set as state variables, and the ICE and EM torques are two independent control variables [38]. The optimized solutions contains a distance-based speed trajectory, which is then converted into a time-based one for online usage.

If the HEV can strictly follow the given speed profile, the net tractive torque at each step can be calculated by (1), and thereby the EM and ICE torques at that step must satisfy (3), (12), and (13). Consequently, this energy minimization problem is formulated as an optimal control problem (OCP) and expressed below.

$$J(\mathbf{x}_0) = \sum_{k=0}^{N-1} \left[m_{ce}(\mathbf{x}_k, \mathbf{u}_k) + m_{sw}(\mathbf{x}_k, \mathbf{u}_k) \right] + m_{rc}(\mathbf{x}_N), \quad (21)$$

subject to (1)-(3), (10)-(13), (16)-(18) and the following,

$$m_{rc}(\mathbf{x}_N) = \frac{C \cdot \left(V_{sc,0}^2 - V_{sc,N}^2\right)}{2\eta_{rc} Q_f}, \quad (21a)$$

$$\mathbf{x}_k = \left[V_{sc,k},\ s_{ce,k}\right]^T, \quad (21b)$$

$$\mathbf{u}_k = \left[T_{ce,k},\ u_{ce,k}\right]^T, \quad (21c)$$

$$\mathbf{x}_0 = \left[V_{sc,0},\ 0\right]^T, \quad (21d)$$

$$V_{sc}^{\min} \le V_{sc,k} \le V_{sc}^{\max}, \quad (21e)$$

$$V_{sc,N}^{\min} \le V_{sc,N} \le V_{sc}^{\max}, \quad (21f)$$

$$s_{ce,N} = 0, \quad (21g)$$

$$T_{em}^{\min}(v_k) \le T_{em}(k) \le T_{em}^{\max}(v_k), \quad (21h)$$

where m_{rc} is the equivalent fuel consumption to recharge SC if the final value of SC voltage $V_{sc,N}$ is less than its initial value $V_{sc,0}$; $V_{sc,N}^{\min}$ is much higher than $V_{sc}^{\min}$ and used as the lower bound for $V_{sc,N}$ to ensure the SC charge sustain. Note that s_{ce} must be 0 at both the start and the end to prevent the ICE from low operating efficiency at low-speed driving, and $T_{em}^{\min}$ and $T_{em}^{\max}$ are variables determined by v due to the rigid connection between the driving wheels and the EM rotor.

IV. ADP-Based EMS Design

The formulated OCP (21) is a mixed-integer nonlinear program (MINLP) problem because it contains both continuous and discrete variables in the state and control vectors. This type of OCPs are generally difficult to be solved by existing optimization solvers because of the huge exploration spaces caused by control decisions at many time steps. A general solution to these OCPs is DDP, which is applicable for complex nonlinear and non-convex OCPs [39]. However, due to the "curse of dimensionality", DDP can hardly be directly implemented online. Moreover, its solution is non-causal because it relies on an accurate powertrain model and complete prior knowledge.

The key reason for the huge complexity of DDP is the explicit representation of VFs as high-dimensional data arrays. To reduce the space complexity without losing too much accuracy, explicit VFs are usually approximated by DNNs, which are trained through simulations and/or experimental data. Close-to-optimal control decisions are derived from the trained DNN models of AVFs. The training and control processes using DNNs are often realized by ADP and DRL. Having less complexity and better adaptivity than DDP, these DNN-based EMSs are widely applied for online HEV energy management in recent years [22].

Nevertheless, DNN-based EMSs require lengthy training time to obtain accurate DNN parameters for estimating optimal state values before they can produce near-optimal control decisions. When the OCP is complex and the DNN is large, the training process becomes very tedious. In light of this, a computationally efficient ADP-based EMS combining the strengths of DDP and DRL is designed in this paper. The reasons why this EMS is named ADP are twofold. First and foremost, the AVF serves as the foundation to calculate optimal control actions for the ICE on/off switch and torque split. Second, the AVF is initialized by optimized solutions of offline DDP for a fast convergence and a robust performance.

Illustrated by Fig. 4, the complete ADP-based EMS framework consists of offline and online parts. The offline part, shown by the yellow block, performs DDP and parametric approximation in sequence. The tabular VF is first generated by DDP through solving the OCP (21). Before being sent to the online part, it is approximated by piecewise polynomials with the method elaborated in Subsection IV-A. The online

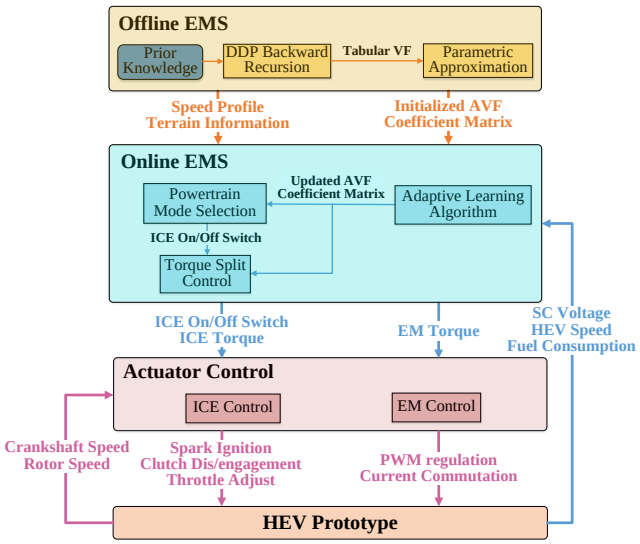


Fig. 4. ADP-Based EMS Framework

part, shown by the blue block, exploits the ADP approach to calculate optimal control actions and then refines the AVF based on real-time state feedback and fuel consumption. Since the complex OCP (21) is computationally intractable in real-time, it is decoupled into two sub-problems and solved by two control modules sequentially. According to (11)-(13), T_{ce} depends on s_{ce}. Thus, the powertrain mode selection module determines the optimal powertrain mode by one-step lookahead with the aid of AVF and generates the correspondingly optimal ICE on/off command u^*_{ce}, using the strategy in (29). If the powertrain works in the hybrid mode, the torque split control module employs the value-based PMP algorithm to calculate optimal torque demands on ICE and EM, T^*_{ce} and T^*_{em}. For a close-to-optimal solution with rapid calculation, the Hamiltonian is simplified to a constrained-quadratic programming problem in (33) and the PMP costate is derived from the AVF by (32). During online control, AVF parameters are iteratively updated by the adaptive learning algorithm elaborated in Subsection IV-D according to the real fuel and electricity consumption. Moreover, to reduce the response time of control actions, the learning algorithm is executed after all optimal control actions are determined and sent to the actuator control, shown by the magenta block, for subsequent operations.

A. Parametric Approximation of VFs

To solve the OCP (21) by DDP, all continuous variables, including the state variable V_{sc}, the control action T_{ce}, and the free variable driving time t, have to be discretized. To mitigate the performance degradation by truncation errors, relatively high resolutions are preferred in the offline calculation. In this case, the resolutions of V_{sc}, T_{ce} and t are $0.1\,V$, $0.05\,Nm$ and $0.5\,s$, respectively. The formulated problem is solved by a generic DP MATLAB function [40] and a tabular VF of three inputs $Y(V_{sc}, s_{ce}, t)$ is derived. Since s_{ce} is a binary variable, this 3D look-up table is separated into two 2D ones with different s_{ce} values, expressed as $Y_{on}(V_{sc}, t)$ and $Y_{off}(V_{sc}, t)$.

Because of dense grids, the tabular $Y(\cdot)$ contains tens of thousands of elements that will consume intractable memory space on onboard processors. To decrease the memory demand, the parametric approximation is adopted to convert tabular VFs into parametric functions. Among various basis function sets, such as polynomials, wavelets, radial basis functions, NNs, and so forth, DNN is the primary candidate to approximate the (state-action) VF in learning-based EMS, because it is sufficiently expressive to represent complicated problems with multiple inputs/outputs and/or non-convex properties [41]. However, the drawbacks of employing DNNs are evident as well. To guarantee the approximation accuracy, a DNN usually possesses a sophisticated architecture and contains at least hundreds of activation functions distributed over several hidden layers. As a consequence, it will consume considerable onboard computation resources.

To save the computation resource without compromising the numeric precision, a concise parametric approximation method should be designed to approximate tabular VFs. Since the powertrain mode imposes a significant impact on VF evolutions, one intuitive approach is to separate the entire $Y_{on}(\cdot)$ and $Y_{off}(\cdot)$ into several time-dependent sections according to the optimal trajectory of powertrain mode $s^\circ_{ce}(t)$ from DDP. Hence, the entire driving period $t \in [0, t_f]$ can be divided into a number of time intervals with the constant powertrain mode. Suppose the number of intervals of hybrid mode in one route is N_h, and thereby that of electric mode is $N_e = N_h + 1$ because the HEV must use the electric mode at the start and the end of one route. Therefore, $Y_{on}(\cdot)$ and $Y_{off}(\cdot)$ are both separated into $N_{md} = N_h + N_e$ sections. Furthermore, denote the boundary of each two adjacent sections by $t^\circ_1, t^\circ_2, \cdots, t^\circ_{N_{md}-1}$ in sequence. For convenience, 0 and t_f are used to label the start of the first section t°_0 and the end of the last section $t^\circ_{N_{md}}$, respectively. For any section with index $n \in [1, N_{md}]$ and $t \in [t^\circ_{n-1}, t^\circ_n)$, $Y_{on,n}(\cdot)$ and $Y_{off,n}(\cdot)$ are approximated as third order polynomials of V_{sc} and t and expressed as,

$$\begin{aligned}\tilde{Y}_{on,n} =& w^n_1 V^3_{sc} + w^n_2 V^2_{sc}\cdot t + w^n_3 V_{sc}\cdot t^2 + w^n_4 t^3 + \\ & w^n_5 V^2_{sc} + w^n_6 V_{sc}\cdot t + w^n_7 t^2 + w^n_8 V_{sc} + w^n_9 t + w^n_{10},\end{aligned} \tag{22}$$

$$\begin{aligned}\tilde{Y}_{off,n} =& w^n_{11} V^3_{sc} + w^n_{12} V^2_{sc}\cdot t + w^n_{13} V_{sc}\cdot t^2 + w^n_{14} t^3 + \\ & w^n_{15} V^2_{sc} + w^n_{16} V_{sc}\cdot t + w^n_{17} t^2 + w^n_{18} V_{sc} + w^n_{19} t + w^n_{20},\end{aligned} \tag{23}$$

where $\boldsymbol{w}^n = [w^n_1, w^n_2, \cdots, w^n_{20}]$ is the coefficient vector for the n^{th} section and obtained by surface fitting. All $\boldsymbol{w}^n$ compose a coefficient matrix $\boldsymbol{W} = \left[\boldsymbol{w}^1; \boldsymbol{w}^2; \cdots; \boldsymbol{w}^{N_{md}}\right]$ of dimension $N_{md} \times 20$.

Hence, at the k^{th} step, the value of a state $Y(\mathbf{x}_k, t_k)$ can be approximated by a section of the piecewise cubic function $\tilde{Y}(\mathbf{x}, t|\boldsymbol{W})$ with only 10 coefficients,

$$\begin{aligned}Y(\mathbf{x}_k, t_k) &\approx \tilde{Y}(\mathbf{x}_k, t_k|\boldsymbol{W}) \\ &= \begin{cases}\tilde{Y}_{on,n}\left(V_{sc,k}, t_k|\boldsymbol{w}^n\right); & s_{ce,k} = 1 \\ \tilde{Y}_{off,n}\left(V_{sc,k}, t_k|\boldsymbol{w}^n\right); & s_{ce,k} = 0\end{cases},\end{aligned} \tag{24}$$

where

$$t_k = k \cdot t_s \in \left[t^\circ_{n-1}, t^\circ_n\right). \tag{25}$$

The NRMSE for this method is less than 2% on several different driving routes, and relevant details concerning spe-

TABLE II. Average NRMSE by Piecewise Polynomials of Different Orders on Several Testing Routes

Polynomial Order	1^{st}	2^{nd}	3^{rd}	4^{th}
Coefficient Number	6	12	20	30
Average NRMSE	3.54%	2.32%	1.33%	1.18%

cific routes adopted in this paper are presented in Section V. Polynomials of different orders have been tested. The results in TABLE II indicate that polynomials of lower orders have larger NRMSE, while those of higher orders cannot obviously decrease the NRMSE but greatly increase computation loads.

B. Powertrain Mode Selection

With the aid of $Y(\cdot)$, the global OCP (21) can be converted into a local one to solve real-time optimal control actions $\mathbf{u}_k^*$ based on $\mathbf{x}_k$. Following the Bellman equation [42], the total fuel consumption to be minimized in the remaining route is described as a one-step lookahead approximation,

$$\tilde{Y}(\mathbf{x}_k,t_k)=m_{ce}(\mathbf{x}_k,\mathbf{u}_k)+m_{sw}(\mathbf{x}_k,\mathbf{u}_k)+\tilde{Y}(\mathbf{x}_{k+1},t_{k+1}), \quad (26)$$

$$\mathbf{u}_k^* = \underset{\mathbf{u}_k}{\operatorname{argmin}}\, \tilde{Y}(\mathbf{x}_k,t_k), \quad (27)$$

subject to the same constraints in OCP (21), where $m_{ce}(\cdot)$ and $m_{sw}(\cdot)$ together constitute the instant cost, and $\tilde{Y}(\mathbf{x}_{k+1},t_{k+1})$ represents the approximated cost-to-go.

Although the OCP (27) is simplified to a large extent, it is still an MINLP problem, and therefore solving it in real-time is still computationally consuming for low-cost onboard processors. Considering constraints (8), (9) and (13) that T_{ce} can only vary within a small range when s_{ce} equals to 1, we simplify the OCP (27) by assuming,

$$T_{ce,k}^* = s_{ce,k}\cdot T_{ce}^{\circ}(\omega_{ce,k}). \quad (28)$$

Thus, T_{ce} is no longer an independent control variable but determined by s_{ce}, and the only control variable left is u_{ce}, i.e., $\mathbf{u}=u_{ce}$. Then, the OCP (27) becomes a binary optimization problem and u_{ce}^* can be rapidly determined by (29),

$$u_{ce,k}^*=\begin{cases} 1; & \tilde{Y}(\mathbf{x}_k,t_k)\Big|_{u_{ce,k}=1} < \tilde{Y}(\mathbf{x}_k,t_k)\Big|_{u_{ce,k}=0} \\ 0; & \tilde{Y}(\mathbf{x}_k,t_k)\Big|_{u_{ce,k}=1} > \tilde{Y}(\mathbf{x}_k,t_k)\Big|_{u_{ce,k}=0} \\ s_{ce,k}; & \tilde{Y}(\mathbf{x}_k,t_k)\Big|_{u_{ce,k}=1} = \tilde{Y}(\mathbf{x}_k,t_k)\Big|_{u_{ce,k}=0} \end{cases}. \quad (29)$$

Note that the assumption (29) is only valid for solving u_{ce}^*. The explicit values of T_{ce}^* and T_{em}^* will be calculated in the following torque split control based on a known s_{ce}^*.

C. Torque Split Control

The torque split control is responsible for splitting T_t into T_{ce} and T_{em} for optimal fuel economy. Its solution is directly dependent on the transient powertrain mode, indicated by the value of s_{ce}. If the powertrain works in the electric mode, i.e., $s_{ce}=0$, then T_{ce} must be 0, and T_t is solely satisfied by T_{em}; otherwise, if it works in the hybrid mode, i.e., $s_{ce}=1$, T_{ce} is nonzero, and there are theoretically infinite admissible combinations of T_{ce} and T_{em} that can satisfy T_t. To efficiently pick out the optimal pair of T_{ce}^* and T_{em}^*, this torque split problem is solved by PMP. For this purpose, the Hamiltonian $\mathcal{H}$ is defined as,

$$\mathcal{H}_k = \dot{m}_{ce,k}+\lambda_k\dot{V}_{sc,k} = \frac{P_{ce,k}}{Q_f}-\lambda_k\frac{P_{sc,k}}{C\cdot V_{sc,k}}, \quad (30)$$

$$T_{ce,k}^* = \underset{T_{ce,k}}{\operatorname{argmin}}\,\mathcal{H}_k, \quad (31)$$

where λ is the costate of V_{sc} and is a scalar.

The optimal value of T_{ce} highly relies on the trajectory of optimal costate λ^* [43], [38]. General methods to obtain this trajectory must require full knowledge of future driving and usually perform tedious searches and complex computations, which are impractical for real-time applications with low-cost microprocessors. Some APMP-based EMSs [44], [45] use the deviation between the real and reference SOCs to calculate suboptimal λ but cannot guarantee robust performances once the driving conditions changed. Given the essential equivalence between PMP and DP, λ^* in PMP is equivalent to the derivative of the optimal VF in DDP with respect to the state variable. Consequently, λ^* can be rapidly estimated by (32),

$$\lambda_k^* \approx \frac{\partial\tilde{Y}_{on,n}(V_{sc,k},t_k|\boldsymbol{w}^n)}{\partial V_{sc,k}}. \quad (32)$$

Since the OCP (31) is a nonlinear programming problem, T_{ce}^* has to be generally solved by a nonlinear programming solver. Our solution is to simplify $\mathcal{H}$ for higher computation efficiency. Owing to the nonlinear item $T_{em}\eta_d^{sign(T_{em})}$ in (3), T_{em} cannot be easily substituted by a function of T_{ce}. To tackle this issue, a reasonable assumption is introduced that T_{em} is positive when $T_{ce}^{\circ}(\omega_{ce})R_{ce}\eta_d$ is less than T_t; otherwise, T_{em} is negative. As a result, $\mathcal{H}$ is transformed into a constrained-quadratic programming problem and illustrated by,

$$\mathcal{H}_k=\begin{cases} a_1T_{ce,k}^2+b_1T_{ce,k}+c_1;\ T_{ce}^{\circ}(\omega_{ce,k})<\dfrac{T_{t,k}}{R_{ce}R_p\eta_d} \\ a_2T_{ce,k}^2+b_2T_{ce,k}+c_2;\ T_{ce}^{\circ}(\omega_{ce,k})\geq\dfrac{T_{t,k}}{R_{ce}R_p\eta_d} \end{cases}, \quad (33)$$

where

$$a_1 = \frac{p_2}{Q_f}-\lambda_k^*\frac{q_2R_{ce}^2}{CR_{em}^2V_{sc,k}}, \quad (33a)$$

$$b_1 = \frac{p_1}{Q_f}+\lambda_k^*\frac{R_{ce}}{CV_{sc,k}}\left(\frac{2q_2T_{t,k}}{R_{em}^2R_p\eta_d}+\frac{q_1}{R_{em}}\right), \quad (33b)$$

$$a_2 = \frac{p_2}{Q_f}-\lambda_k^*\frac{q_2R_{ce}^2\eta_d^4}{CR_{em}^2V_{sc,k}}, \quad (33c)$$

$$b_2 = \frac{p_1}{Q_f}+\lambda_k^*\frac{R_{ce}}{CV_{sc,k}}\left(\frac{2q_2T_{t,k}\eta_d^3}{R_{em}^2R_p}+\frac{q_1\eta_d^2}{R_{em}}\right). \quad (33d)$$

The coefficients p_2, p_1, q_2 and q_1 are determined by ω_{ce} and ω_{em}. a_1, b_1, a_2 and b_2 are intermediate variables, and c_1 and c_2 are two constants independent of T_{ce}.

Since $\mathcal{H}$ is converted into a convex quadratic programming problem expressed by (33), T_{ce}^* can be solved efficiently, and T_{em}^* solved by (3) thereafter.

D. Adaptive Learning Algorithm

The accuracy of $\tilde{Y}_{on}(\cdot)$ and $\tilde{Y}_{off}(\cdot)$ from DDP solutions highly relies on the accuracy of the HEV powertrain model and driving information. However, inevitable model errors and unmeasured disturbances during driving degrade the effectiveness of $\tilde{Y}_{on}(\cdot)$ and $\tilde{Y}_{off}(\cdot)$. They will seriously worsen the HEV fuel economy and even destruct vehicular drivability.

To overcome this issue, an adaptive learning algorithm is designed and introduced into the online EMS framework to iteratively update the coefficient matrix $\boldsymbol{W}$ according to the feedback information from online implementation. In this way, the iteratively improved AVFs will approach the optima suitable to the actual vehicle dynamics and road conditions, and then support the control modules to generate real-time close-to-optimal control actions.

More specially, temporal difference (TD) learning [46] is applied to update $\boldsymbol{W}$ in this paper. At time step t_{k+1}, the HEV transits from $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ based on $\mathbf{u}_k^*$, and outputs running cost $m_{ce,k}$ and $m_{sw,k}$. According to (26), the estimated TD target $\tilde{Y}^{\circ}(\cdot)$ at a state $\mathbf{x}_k$ is expressed as,

$$\tilde{Y}^{\circ}\big(\mathbf{x}_k, t_k|\boldsymbol{W}\big) = m_{ce}\big(\mathbf{x}_k, \mathbf{u}_k^*\big) + m_{sw}\big(\mathbf{x}_k, \mathbf{u}_k^*\big) + \tilde{Y}\big(\mathbf{x}_{k+1}, t_{k+1}|\boldsymbol{W}\big), \tag{34}$$

where $\mathbf{u}^*$ can be obtained from the control modules elaborated in Subsections IV-B and IV-C.

Since $\tilde{Y}^{\circ}(\cdot)$ gives an unbiased estimation for $\tilde{Y}(\cdot)$ through bootstrapping, the TD error between $\tilde{Y}^{\circ}(\cdot)$ and $\tilde{Y}(\cdot)$ can be expressed as,

$$e\big(\mathbf{x}_k, t_k|\boldsymbol{W}\big) = \tilde{Y}^{\circ}\big(\mathbf{x}_k, t_k|\boldsymbol{W}\big) - \tilde{Y}\big(\mathbf{x}_k, t_k|\boldsymbol{W}\big). \tag{35}$$

Afterward, we define the loss function based on an individual sample as,

$$l_k = \frac{1}{2}e^2\big(\mathbf{x}_k, t_k|\boldsymbol{W}\big). \tag{36}$$

To efficiently eliminate the TD error and ensure the training robustness with limited onboard computation resources, batch gradient descent (BGD) [47] is employed to update $\boldsymbol{W}$ with a batch $\mathcal{K}$ of maximum size $\bar{N}$ samples $s_k = \{l_k, \mathbf{x}_k\}$ approximated by the same coefficient vector $\boldsymbol{w}^n$. As described in Subsection IV-A, according to the optimal trajectory $s_{ce}^{\circ}(t)$ by DDP, the entire VFs $Y_{on}(\cdot)$ and $Y_{off}(\cdot)$ are divided into N_{md} time-dependent sections with each one approximated by an individual coefficient vector $\boldsymbol{w}^n, n \in [1, N_{md}]$. During each sampling step t_k, if the batch is not full, i.e., $|\mathcal{K}| < \bar{N}$, and the new sample s_k is approximated based on the same $\boldsymbol{w}^n$ to those stored in $\mathcal{K}$, the learning algorithm will not perform any update on $\boldsymbol{w}^n$ but only append s_k into $\mathcal{K}$; otherwise, $\boldsymbol{w}^n$ will be updated immediately based on existing samples stored in $\mathcal{K}$, and then $\mathcal{K}$ will be reset to empty before a new sample s_k is appended into it. The coefficient update follows the rule of gradient descent. Since the parametric approximation is realized by piecewise polynomials, the gradient of the loss function $\boldsymbol{\Delta}_k \in \mathbb{R}^{20}$ of one sample s_k is calculated by,

$$\boldsymbol{\Delta}_k = \frac{\partial l_k}{\partial \boldsymbol{w}^n}. \tag{37}$$

Algorithm 1: Adaptive learning in one episode

Input: α, $\bar{N}$, $\mathbf{x}_0$, and $\boldsymbol{W}$
1 **Initialize** $m_{ce,0}=0$, $m_{sw,0}=0$, $\mathcal{K}=\emptyset$, $k=1$, $n=1$, and $n'=1$;
2 **while** $t_k < t_f$ **do**
3 Receive $\mathbf{x}_k$, $m_{ce,k-1}$, and $m_{sw,k-1}$;
4 Find $n' \in [1, N_{md}]$ where $t^{\circ}_{n'-1} \le t_k < t^{\circ}_{n'}$;
5 **if** ($|\mathcal{K}|=\bar{N}$ **or** $n \neq n'$) **then**
6 Update the n^{th} row of $\boldsymbol{W}$, i.e., $\boldsymbol{w}^n$, by (37) and (38);
7 $\mathcal{K} = \emptyset$;
8 $n \leftarrow n'$;
9 **end**
10 Calculate l_{k-1} by (34)-(36);
11 Append $s_{k-1}=\{l_{k-1}, \mathbf{x}_{k-1}\}$ into the buffer $\mathcal{K}$;
12 $k \leftarrow k+1$;
13 **end**
Output: $\boldsymbol{W}$

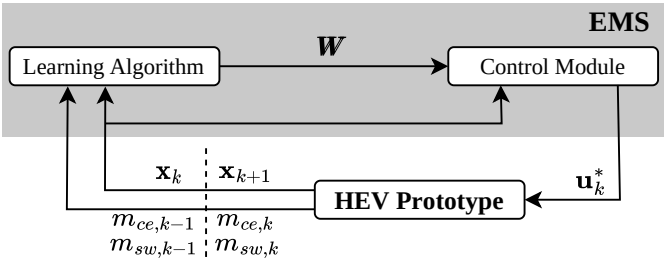


Fig. 5. Information Interaction between EMS and HEV

Note that at each step t_k, $\tilde{Y}(\cdot)$ is expressed by either $\tilde{Y}_{on,n}(\cdot)$ or $\tilde{Y}_{off,n}(\cdot)$ depending on the value of $s_{ce,k}$. Thus, only half of the entries in $\boldsymbol{\Delta}_k$ are non-zero. Consequently, the corresponding coefficient vector $\boldsymbol{w}^n$ can be updated by accounting for all the samples in $\mathcal{K}$ and expressed as,

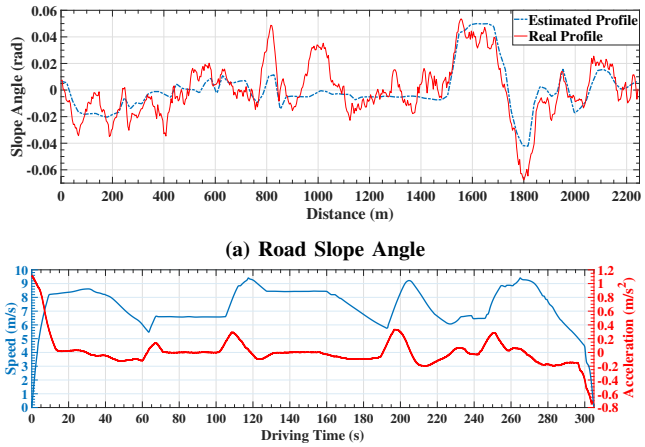
$$\boldsymbol{w}^n \leftarrow \boldsymbol{w}^n - \beta \sum_{k \in \mathcal{K}} \boldsymbol{\Delta}_k, \tag{38}$$

where β denotes the learning rate in the update process.

The running process of this adaptive learning algorithm in one episode is summarized by Algorithm 1 with the information interaction with the control module and vehicular system depicted in Fig. 5. In Algorithm 1, two variables, n and n', are used to identify the variation of time sections referring to $\boldsymbol{w}^n$. When the driving time t_k reaches a new time section, n' is updated in Line 4, which triggers the condition to update $\boldsymbol{w}^n$ and clear $\mathcal{K}$. After that, the updated index n' is assigned to n in Line 8, enabling $\mathcal{K}$ to accumulate new samples from the new section referring to $\boldsymbol{w}^{n'}$.

V. PIL Simulation Results

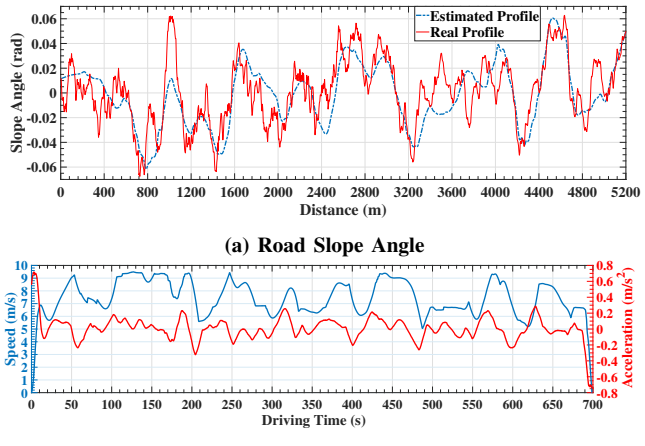
To manifest the superiority of the proposed ADP using piecewise polynomial AVF in terms of fuel economy and computation efficiency, three types of comparative studies have been performed through PIL simulations based on a portable microprocessor with limited computation resources. First, the



(a) Road Slope Angle

(b) HEV Speed and Acceleration

Fig. 6. Driving Information on Route SEM16



(a) Road Slope Angle

(b) HEV Speed and Acceleration

Fig. 7. Driving Information on Route STHLM

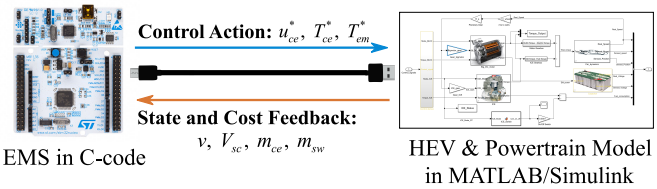


Fig. 8. PIL Simulation Platform

proposed method is compared with the optimal DDP and an APMP controller to verify the optimality of the equivalent fuel efficiency. Second, this method is compared with a non-adaptive DP to highlight the significance of the adaptivity in the actual driving environment, especially when the driving conditions become aggravated. Third, this method using polynomial AVF is compared with a similar ADP method using explicit tabular VF to demonstrate the advantages of learning efficiency and memory utilization.

A. Driving Routes

Two different driving routes are selected to test the real-time performance of the ADP-based EMS, and the geographic information and corresponding speed profiles are elaborated by Figs. 6 and 7, respectively. As briefly explained at the beginning of Section III, the optimal speed profile is computed by the distance-based DDP method elaborated in [38]. The first route, named SEM16, is a short route of roughly $2240\,m$ and relatively gentle because its slope angle is mostly within $\pm 0.02\,rad$. Correspondingly, its speed profile has small variations around the average speed and its acceleration profile has small variations around zero during the majority of the route except the starting and stopping phases, as illustrated by Fig. 6(b). By contrast, the second route is a section of public road in Stockholm and thereby named STHLM. This route is $5200\,m$ long and contains many steep uphills and downhills. Hence, both its speed and acceleration profiles vary much more dramatically and frequently, as illustrated in Fig. 7(b).

It is worth noting that there are two slope angle profiles exhibited for each route. The real slope angle profiles for the two driving routes are depicted by the red plots in Figs. 6(a) and 7(a). The blue plots represent the low-fidelity estimations of slope angles. To verify the effectiveness and optimality of the proposed EMS, real profiles are utilized by the offline DDP to find the theoretical optimum as a reference. By contrast, low-fidelity estimations are utilized to design online EMSs, including generating optimal speed profiles and initializing VFs, while real profiles are adopted in the simulation environment to test the adaptive learning ability of the ADP-based EMS. Furthermore, the fuel efficiency attained by the proposed EMS is compared with the optimum by DDP based on real profiles.

B. PIL Simulation Test Bench

To verify the online control performance and computation efficiency of the proposed EMS, a test bench is established to perform PIL simulations in which the designed EMS is converted into C code and executed by a real processor, as shown in Fig. 8. To demonstrate the advantages of the proposed EMS in terms of computation efficiency, a portable microprocessor *STM32L476RGT6*[2] (Arm Cortex-M4 MCU $80\,MHz$ with up to $1\,Mbyte$ flash memory and $128\,Kbyte$ of SRAM) is selected to run the EMS in real-time. The complete system model, including the HEV dynamics, actuator control and digital sensors, is built up in *MATLAB/Simulink*[3]. At each step, the microprocessor receives real-time state and cost feedback, including v, V_{sc}, m_{ce} and m_{sw}, from the system model, and then sends out optimized control actions, containing u_{ce}^*, T_{ce}^* and T_{em}^*. The real-time information interaction between the EMS and the system model is realized by serial communication.

For a balance between the computation load and the control performance, the sampling periods for the powertrain mode selection and the adaptive learning algorithm are $1\,s$, and that for the torque split control is $0.1\,s$. Besides, $V_{sc,0}$ is set as $48\,V$, a value close to but lower than the upper bound. Hence, the SC can be either discharged or recharged at the start of the route, even if there is a sharp uphill or downhill.

To investigate the robustness and adaptivity of the proposed EMS in complex working environments, the training process

[2]https://www.st.com/en/microcontrollers-microprocessors/stm32l476rg.html
[3]https://se.mathworks.com/products/simulink.html

is performed based on Monte Carlo simulations, in which various types of actuator disturbances, sensor noises as well as system uncertainties are imposed on the HEV model. Detailed information about Monte Carlo simulation is presented in [38]. Correspondingly, all results presented and summarized in the following subsections are from PIL Monte Carlo simulations. Moreover, our investigation finds that in addition to the road slope angle, the rolling resistance and air drag coefficients play important roles in HEV fuel consumption. On account of this, to further investigate the adaptivity of the proposed EMS under sudden variations of driving conditions, relevant EMSs will be tested under more aggravated driving scenarios, where both coefficients increase by 10%, but designed EMSs do not know this fact.

C. Benchmark EMSs

To manifest the superiority of the proposed ADP method on both control performance and computation efficiency, several benchmark EMSs are developed and tested for comparison.

The first one is the optimal DDP, which can only run offline on a backward quasi-static HEV model with complete prior knowledge, illustrated by the red plots in Figs. 6(a) and 7(a), and give rise to an optimal solution for a specific scenario. Hence, we only use its final result to evaluate the optimality of other EMSs.

The second one is an APMP, where the costate is calculated by a PID controller to regulate the error between the reference SOC and the real-time SOC [44], and the ICE on/off commands are determined by a thermostat controller [9]. The relevant formulas are given below.

$$u_{ce,k}=\begin{cases}1; & V_{sc,k}\le V_1 \; or \; T^{\max}_{em,k}<\dfrac{T_{t,k}}{R_{em}\eta_d}\\ 0; & V_{sc,k}\ge V_2 \; or \; T^{\min}_{em,k}>\dfrac{(T_{t,k}-T^{\min}_{ce,k})\eta_d}{R_{em}}\\ s_{ce,k}; & otherwise\end{cases}, \tag{39}$$

$$\begin{aligned}\lambda_k =&\lambda_0 + k_p\Delta V_{sc,k} + k_i\sum_{i=0}^{k}\Delta V_{sc,i}\\ &+ k_d\big(\Delta V_{sc,k}-\Delta V_{sc,k-1}\big),\end{aligned} \tag{40}$$

$$\Delta V_{sc,k} = V^{\circ}_{sc,k} - V_{sc,k}, \tag{41}$$

where V_1 and V_2 ($V^{\min}_{sc} < V_1 < V_2 < V^{\max}_{sc}$) are prescribed lower and upper thresholds that divide the admissible range of V_{sc} into 3 sections; λ_0 reflects an initial guess to λ^*; k_p, k_i and k_d are proportional, integral and derivative gains of the PID control, respectively; ΔV_{sc} is the deviation between the current SC voltage and the optimal one V°_{sc} from offline DDP.

Illustrated by (39), the ICE should be switched on when V_{sc} is lower than V_1 or the EM cannot solely satisfy the torque demand on the powertrain; on the contrary, the ICE should be off when V_{sc} is higher than V_2 or the EM cannot recuperate the surplus torque on the powertrain if the ICE keeps on working; otherwise, the ICE prefers to maintain its current on/off status. All tunable parameters, including V_1, V_2, λ, k_p, k_i and k_d are carefully selected after a long-time calibration for a possibly satisfactory performance.

TABLE III. Results Comparison on Driving Route SEM16

Control Strategy	ADP	APMP	NADP	ADP-NPA
Final Voltage Variation (V)	0.08	-0.78	-0.35	-1.67
Total Fuel Consumption (mL)	11.65	12.22	11.93	11.62
Equivalent Fuel Efficiency (km/L)	192.7	183.6	188.1	193.0
Flash Memory Occupation (Kbyte)	90.63	88.74	85.52	293.45
RAM Occupation (Kbyte)	39.68	25.28	33.16	33.27
Max. CPU Utilization (%)	23.96	4.31	7.60	5.72
Avg. CPU Utilization (%)	2.82	2.09	2.68	2.07

The third one adopts exactly the same control method as the proposed EMS but has no adaptive learning mechanism. It is hence named NADP-based EMS. It is used to assess the effectiveness of the adaptive learning algorithm as well as its computation resource consumption in online applications.

The last one is an ADP-based EMS without performing the parametric approximation, and thereby is named ADP-NPA. The principle of this EMS is very similar to the proposed one, and the essential difference is that real-time state values are acquired by interpolation on tabular VFs. For this reason, its learning algorithm totally differs from that of the proposed ADP-based EMS. Testing results of this EMS aim to evaluate the significance of the adopted parametric approximation method for ADP-based EMS. Denote the value of state $\mathbf{x_k}$ by $Y(\mathbf{x}_k, t_k)$. The TD target $Y^{\circ}(\cdot)$ is expressed as,

$$\begin{aligned}Y^{\circ}(\mathbf{x}_k, t_k) = m_{ce}\big(\mathbf{x}_k, \mathbf{u}^*_k\big) + m_{sw}\big(\mathbf{x}_k, \mathbf{u}^*_k\big)\\ + Y\big(\mathbf{x}_{k+1}(\mathbf{x}_k, \mathbf{u}^*_k), t_{k+1}\big).\end{aligned} \tag{42}$$

Then, the corresponding state value can be updated by the value iteration algorithm,

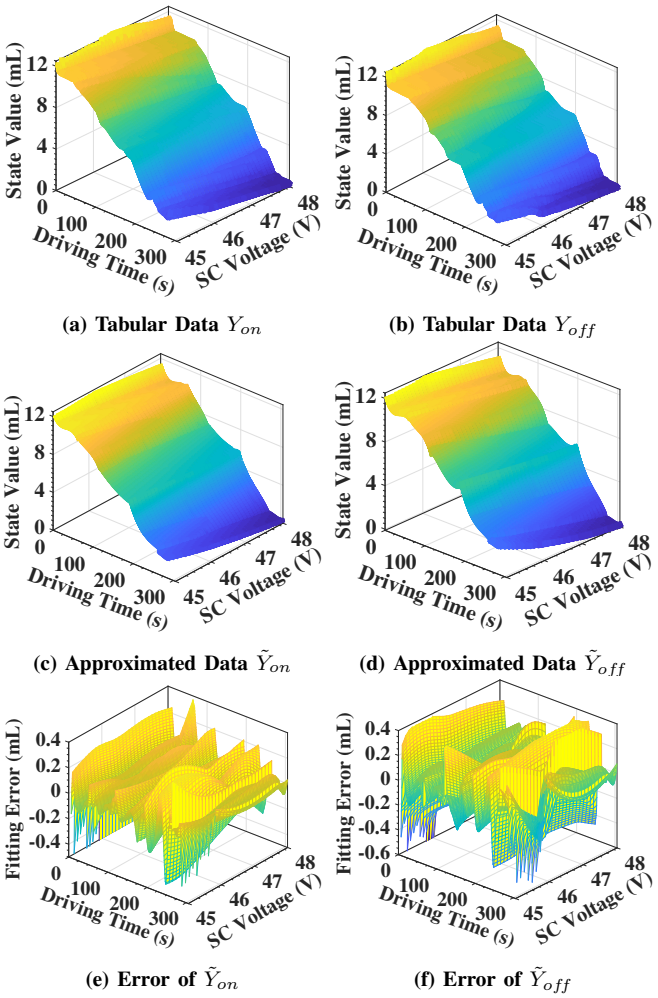
$$Y(\mathbf{x}_k, t_k) \leftarrow (1-\beta)\cdot Y(\mathbf{x}_k, t_k) + \beta\cdot Y^{\circ}(\mathbf{x}_k, t_k). \tag{43}$$

Note that the state variable V_{sc} is continuous, while $Y(\cdot)$ is a discrete representation of the VF. As a result, any state value not at the meshgrid of lookup tables is obtained by linear interpolation, and only the nearest meshgrid point to the sample will be updated by (43).

D. Results on Driving Route SEM16

All testing results on SEM16 by the proposed ADP-based EMS and comparison methods are illustrated in Figs. 9 - 11, and summarized in TABLE III. Figs. 9(a) and 9(b) show tabular VFs initialized by offline DDP, which are utilized by ADP-NPA. After the parametric approximation, the resulting AVFs, shown in Figs. 9(c) and 9(d), are sent to ADP and NADP for online usage. Fitting errors on all sampling points are shown in Figs. 9(e) and 9(f), with the NRMSE of only 1.59%. Due to the large size, all data in tabular VFs have to be converted into single-precision floating-point type before being loaded into the microprocessor, while other variables and parameters maintain the double-precision type.

Recall that VFs for all online DP strategies are computed based on low-fidelity estimations of road slope profiles, illustrated by the blue plots in Figs. 6(a) and 7(a), and only ADP and ADP-NPA have online learning capability among all tested EMSs. The learning capability can improve VFs during online control. Fig. 10(a) shows that both ADP and ADP-NPA

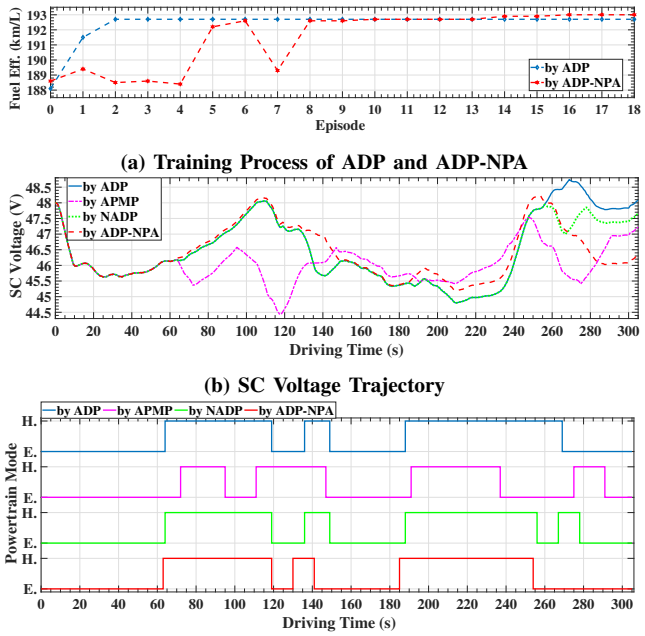


(a) Tabular Data Y_{on} (b) Tabular Data Y_{off}

(c) Approximated Data $\tilde{Y}_{on}$ (d) Approximated Data $\tilde{Y}_{off}$

(e) Error of $\tilde{Y}_{on}$ (f) Error of $\tilde{Y}_{off}$

Fig. 9. VFs on Driving Route SEM16

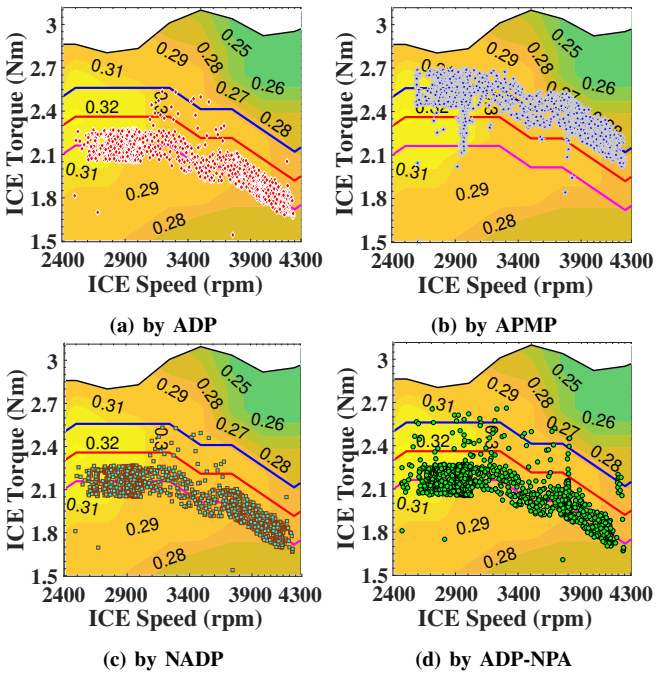


(a) Training Process of ADP and ADP-NPA

(b) SC Voltage Trajectory

(c) Powertrain Mode Trajectory (H. refers to Hybrid, E. to Electric)

Fig. 10. PIL Simulation Results on Driving Route SEM16



(a) by ADP (b) by APMP

(c) by NADP (d) by ADP-NPA

Fig. 11. ICE Operation Points on Driving Route SEM16

can improve fuel efficiencies after a few training episodes, but ADP converges only after 3 episodes while ADP-NPA requires at least 15 episodes. The main reason is that ADP has much fewer training parameters than the extraordinary large table of ADP-NPA. TABLE III compares PIL simulation results of four EMSs. The results for ADP and ADP-NPA are the ones after the training completes. The best fuel efficiency by ADP on this route reaches $192.7\,km/L$, only a little bit lower than ADP-NPA of $193.0\,km/L$ and very close to the optimal result of $195.4\,km/L$ by offline DDP. Thanks to its adaptivity, the proposed ADP is 2.5% higher than NADP and 5% higher than APMP in equivalent fuel efficiency, respectively. One essential reason for the slightly worse fuel efficiency over ADP-NPA is that, exhibited by Figs. 10(c) and 11, the ICE driven by ADP works for $148\,s$ in total, $13\,s$ longer than that by ADP-NPA, even though ICE operation points by two EMSs are concentrated in the same region and the total powertrain mode switching numbers are identical. From the aspect of charge sustainability, depicted by Fig. 10(b), the proposed ADP outperforms the other three because its net electricity consumption during driving is negative, so the terminal penalty for compensation is avoided.

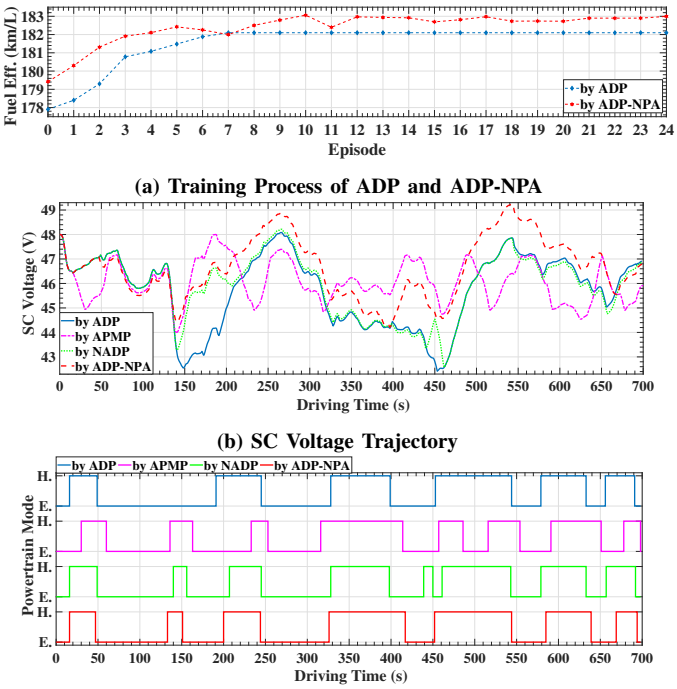
On the real-time computation efficiency, APMP consumes the least onboard computation resources in terms of RAM occupation and CPU utilization since it uses simple heuristic methods to determine the ICE on/off switch and calculate the costate for PMP, and does not have the learning algorithm. Nevertheless, its flash memory occupation is slightly larger than that of ADP and NADP because it requires the optimal SC voltage trajectory by offline DDP as a reference. In contrast, owing to the introduction of adaptive learning algorithm, ADP consumes more onboard computation resources than NADP, but the growths in flash memory occupation, RAM

occupation, and average CPU utilization are very limited, only $6\,Kbyte$, $7.7\,Kbyte$, and 0.18%, respectively, indicating the high efficiency of learning algorithm. Moreover, ADP has a great advantage over ADP-NPA in terms of flash memory occupation without evidently increasing the computation intensity. Thanks to the parametric approximation, ADP does not have to save huge lookup tables and thereby saves nearly 70% of onboard flash memory space. Since the learning algorithm for updating a set of parameters is more complex than that for updating several individual points, the RAM occupation and average CPU utilization by ADP are higher than those by ADP-NPA. Additionally, due to the batch usage, the learning algorithm in ADP performs parameter updates only when the batch is full or the powertrain mode is switched instead of at each sampling period. Consequently, its maximum CPU utilization is evidently higher than that of ADP-NPA. A batch of larger size can improve the robustness of the learning algorithm and slightly increase the RAM occupation, but can drastically raise the instant computation overhead, reflected as the surge of maximum CPU utilization. Thus, the batch size should be carefully selected based on the computing capability of the selected onboard processor.

E. Results on Driving Route STHLM

Testing results on STHLM, presented by Figs. 12 and 13, and TABLE IV, further reveal the strengths of the proposed ADP. First of all, approximated results on VFs on this route are very similar to those on route SEM16, with a smaller NRMSE of only 0.75%. Illustrated by Fig. 12(a), after training of 7 episodes, the fuel efficiency by ADP promptly converges to a steady state of $182.1\,km/L$, reaching more than 97% of DDP of $186.5\,km/L$, roughly 2.5% higher than that by NADP and 5.9% higher than that by APMP. In contrast, ADP-NPA achieves a slightly higher result of $182.9\,km/L$ after an obviously longer training process of more than 20 episodes. As exhibited by Fig. 12(c), ADP utilizes the ICE for $337\,s$ totally, almost identical to NADP, $13\,s$ longer than that by APMP and $17\,s$ shorter than that by ADP-NPA. However, the number of powertrain mode switches by ADP is the least, one couple less than that by ADP-NPA and two couples less than those by APMP and NADP, implying more robust operation on ICE and a longer ICE lifespan. Besides, Fig. 13 illustrates sampled ICE operation points by each EMS. Compared to DP-based EMSs, APMP distributes more points outside the peak efficiency region (over 30%). Although none of these EMSs can ensure the final SC voltage equal to its initial value, the SC voltage driven by ADP can well recover back to $47\,V$ at last, close to its initial value and better than its counterparts.

The consumption of onboard computation resources by each EMS on this route present a similar tendency to that on SEM16. The proposed ADP enjoys a larger advantage because the longer driving time increases the size of lookup tables used by ADP-NPA. As a result, the flash memory occupation of ADP-NPA on this longer route STHLM increases to $611.30\,Kbyte$, more than twice of that on the shorter route SEM16, whereas that of ADP increases by only $50\,Kbyte$ mainly resulting from the longer profiles of speed reference

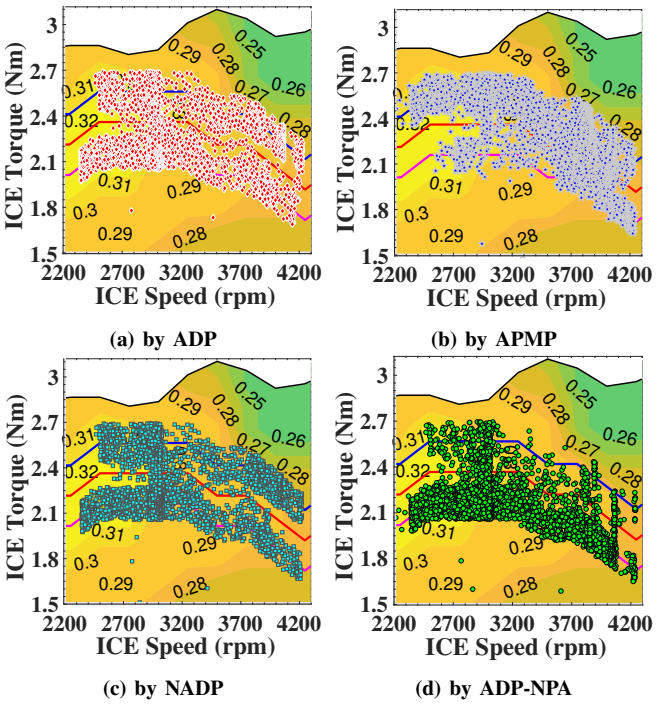


(a) Training Process of ADP and ADP-NPA

(b) SC Voltage Trajectory

(c) Powertrain Mode Trajectory

Fig. 12. PIL Simulation Results on Driving Route STHLM



(a) by ADP (b) by APMP

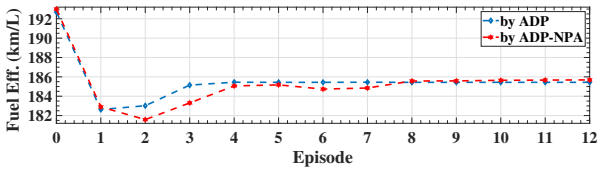
(c) by NADP (d) by ADP-NPA

Fig. 13. ICE Operation Points on Driving Route STHLM

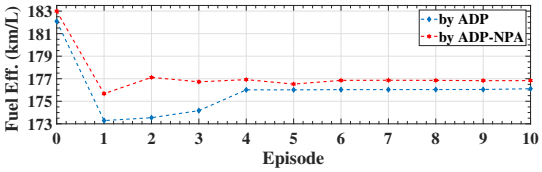
and terrain information. Admittedly, the size of coefficient matrix for approximating state values in this longer route will rise as the number of necessary powertrain switches ascends. Nonetheless, this increment has a negligible impact on the overall memory occupation since one more mode switch will only introduce 20 extra coefficients. As described in Subsection V-D, due to the complex computing process of

TABLE IV. Results Comparison on Driving Route STHLM

Control Strategy	ADP	APMP	NADP	ADP-NPA
Final Voltage Variation (V)	-1.09	-2.10	-1.13	-1.22
Total Fuel Consumption (mL)	28.49	30.17	29.17	28.36
Equivalent Fuel Efficiency (km/L)	182.1	172.0	177.9	182.9
Flash Memory Occupation (Kbyte)	145.60	156.86	139.66	611.30
RAM Occupation (Kbyte)	40.93	25.30	33.24	33.29
Max. CPU Utilization (%)	27.08	4.44	7.95	5.57
Avg. CPU Utilization (%)	2.95	2.16	2.77	2.06



(a) Driving Route SEM16



(b) Driving Route STHLM

Fig. 14. Training Process of Adaptivity Test

the learning algorithm, the RAM occupation and average CPU utilization by ADP on this route are still higher than those by APMP, ADP-NPA, and NADP. It is noteworthy that, the real-time computation overheads by each EMS on these two routes are similar, manifesting the consistency of all these online EMSs when applied to different driving tasks.

F. Adaptivity Test

To verify the adaptivity of the proposed ADP-based EMS against sudden variations during driving, an extra adaptivity test is performed in which larger rolling and aerodynamic resistances are imposed on the HEV. As aforementioned, only ADP and ADP-NPA have the adaptive learning mechanism. Hence, Fig. 14 illustrates the training processes of ADP and ADP-NPA on both driving routes, and TABLE V compares their after-training fuel efficiencies with those from APMP and NADP without learning ability.

Fig. 14 shows that fuel efficiencies of ADP and ADP-NPA suffer steep slumps immediately after driving conditions suddenly worsen. Nonetheless, ADP can readily adapt to this variation after 4 episodes on both routes owing to its small number of AVF parameters. By contrast, ADP-NPA requires longer training time to fully update its tabular VFs. Note that the fuel efficiency cannot return to the high value before the environmental change, because the increases in tire rolling friction and aerodynamic drag inevitably deteriorate the fuel efficiency. Numeric results in TABLE V summarize the significance of adaptive learning in practice when the system model and prior knowledge cannot accurately reflect actual driving scenarios. Due to the lack of a learning mechanism, the performances of APMP and NADP seriously degrade by around 10% on both routes. By comparison, the decreases in fuel efficiency of ADP and ADP-NPA are very limited after sufficient training, within 4% on SEM16 and 3.5% on STHLM, respectively.

TABLE V. Adaptivity Test Results on Both Driving Routes

Control Strategy	ADP	APMP	NADP	ADP-NPA
SEMI6	185.4 (km/L)	165.8	171.8	185.7
STHLM	176.1	159.8	164.2	176.8

VI. Conclusion and Future Work

To minimize the fuel consumption of a parallel HEV, this paper proposes a computationally efficient ADP-based EMS that combines the strengths of existing model-based and learning-based EMSs. On the one hand, by making use of AVFs initialized by offline DDP and approximated as piecewise cubic polynomials, the OCP containing ICE on/off switch and torque split can be rapidly solved to fulfill the real-time requirement; on the other hand, an adaptive learning algorithm is designed to iteratively update AVFs according to the actual energy consumption in real-time applications. PIL simulation results on two different driving routes figure out that the proposed ADP-based EMS can be efficiently executed by a portable microprocessor and generate close-to-optimal fuel efficiency, at least 5% higher than that by an APMP. Compared with two benchmark EMSs without the learning mechanism and parametric approximation, the proposed EMS fully exhibits the effectiveness of adaptive learning algorithm and its superiority in terms of both memory occupation and training speed, especially in the long-time driving route.

All EMSs studied in this paper are based on an HEV with a fixed powertrain configuration. However, it is well known that the compelling fuel economy relies on not only the appropriate EMS but also the proper powertrain configuration. In view of this, the future research orientation will favor excavating the potentiality of efficient cooperative optimization on both powertrain component sizing and real-time energy management. In this context, further improved fuel economy can be anticipated by virtue of advanced EMSs in the premise of the most suitable component sizes according to the variations of driving conditions and requirements in reality.

References

[1] A. Ibrahim and F. Jiang. The electric vehicle energy management: An overview of the energy system and related modeling and simulation. *Renew. Sustain. Energy Rev.*, 144(111049):1–28, 2021.

[2] S. Bai and C. Liu. Overview of energy harvesting and emission reduction technologies in hybrid electric vehicles. *Renew. Sustain. Energy Rev.*, 147(111188):1–17, 2021.

[3] Y. Huang, H. Wang, A. Khajepour, B. Li, J. Ji, K. Zhao, and C. Hu. A review of power management strategies and component sizing methods for hybrid vehicles. *Renew. Sustain. Energy Rev.*, 96:132–144, 2018.

[4] C. Yang, S. You, W. Wang, L. Li, and C. Xiang. A Stochastic Predictive Energy Management Strategy for Plug-in Hybrid Electric Vehicles Based on Fast Rolling Optimization. *IEEE Trans. Ind. Electron.*, 67(11):9659–9670, 2020.

[5] S. Sarvaiya, S. Ganesh, and B. Xu. Comparative analysis of hybrid vehicle energy management strategies with optimization of fuel economy and battery life. *Energy*, 228(120604):1–18, 2021.

[6] D. D. Tran, M. Vafaeipour, M. El Baghdadi, R. Barrero, J. Van Mierlo, and O. Hegazy. Thorough state-of-the-art analysis of electric and hybrid vehicle powertrains: Topologies and integrated energy management strategies. *Renew. Sustain. Energy Rev.*, 119(109596):1–29, 2020.

[7] X. Hu, J. Han, X. Tang, and X. Lin. Powertrain Design and Control in Electrified Vehicles: A Critical Review. *IEEE Trans. Transp. Electrif.*, 7(3):1990–2009, 2021.

[8] M. Kim, D. Jung, and K. Min. Hybrid thermostat strategy for enhancing fuel economy of series hybrid intracity bus. *IEEE Trans. Veh. Technol.*, 63(8):3569–3579, 2014.

[9] W. Shabbir and S. A. Evangelou. Threshold-changing control strategy for series hybrid electric vehicles. *Appl. Energy*, 235:761–775, 2019.

[10] A. Macias Fernandez, M. Kandidayeni, L. Boulon, and H. Chaoui. An Adaptive State Machine Based Energy Management Strategy for a Multi-Stack Fuel Cell Hybrid Electric Vehicle. *IEEE Trans. Veh. Technol.*, 69(1):220–234, 2020.

[11] D. Phan, A. Bab-Hadiashar, M. Fayyazi, R. Hoseinnezhad, R. N. Jazar, and H. Khayyam. Interval Type 2 Fuzzy Logic Control for Energy Management of Hybrid Electric Autonomous Vehicles. *IEEE Trans. Intell. Veh.*, 6(2):210–220, 2021.

[12] V. Larsson, L. Johannesson, and B. Egardt. Analytic solutions to the dynamic programming subproblem in hybrid vehicle energy management. *IEEE Trans. Intell. Veh.*, 64(4):1458–1467, 2015.

[13] L. Li, C. Yang, Y. Zhang, L. Zhang, and J. Song. Correctional DP-Based Energy Management Strategy of Plug-In Hybrid Electric Bus for City-Bus Route. *IEEE Trans. Intell. Veh.*, 64(7):2792–2803, 2015.

[14] X. Lü, Y. Wu, J. Lian, Y. Zhang, C. Chen, P. Wang, and L. Meng. Energy management of hybrid electric vehicles: A review of energy optimization of fuel cell hybrid power system based on genetic algorithm. *Energy Convers. Manag.*, 205(112474):1–26, 2020.

[15] B. Wang, J. Xu, B. Cao, and B. Ning. Adaptive mode switch strategy based on simulated annealing optimization of a multi-mode hybrid energy storage system for electric vehicles. *Appl. Energy*, 194:596–608, 2017.

[16] S. Y. Chen, C. H. Wu, Y. H. Hung, and C. T. Chung. Optimal strategies of energy management integrated with transmission control for a hybrid electric vehicle using dynamic particle swarm optimization. *Energy*, 160:154–170, 2018.

[17] J. M. Lujan, C. Guardiola, B. Pla, and A. Reig. Analytical Optimal Solution to the Energy Management Problem in Series Hybrid Electric Vehicles. *IEEE Trans. Intell. Veh.*, 67(8):6803–6813, 2018.

[18] M. Razi, N. Murgovski, T. McKelvey, and T. Wik. Design and Comparative Analyses of Optimal Feedback Controllers for Hybrid Electric Vehicles. *IEEE Trans. Veh. Technol.*, 70(4):2979–2993, 2021.

[19] Y. Huang, H. Wang, A. Khajepour, H. He, and J. Ji. Model predictive control power management strategies for HEVs: A review. *J. Power Sources*, 341:91–106, 2017.

[20] Z. Chen, C. C. Mi, J. Xu, X. Gong, and C. You. Energy management for a power-split plug-in hybrid electric vehicle based on dynamic programming and neural networks. *IEEE Trans. Intell. Veh.*, 63(4):1567–1580, 2014.

[21] J. Shi, B. Xu, Y. Shen, and J. Wu. Energy management strategy for battery/supercapacitor hybrid electric city bus based on driving pattern recognition. *Energy*, 243(122752):1–13, 2022.

[22] X. Hu, T. Liu, X. Qi, and M. Barth. Reinforcement Learning for Hybrid and Plug-In Hybrid Electric Vehicle Energy Management: Recent Advances and Prospects. *IEEE Ind. Electron. Mag.*, 13(3):16–25, 2019.

[23] R. Lian, H. Tan, J. Peng, Q. Li, and Y. Wu. Cross-Type Transfer for Deep Reinforcement Learning Based Hybrid Electric Vehicle Energy Management. *IEEE Trans. Intell. Veh.*, 69(8):8367–8380, 2020.

[24] T. Liu, X. Hu, S. E. Li, and D. Cao. Reinforcement Learning Optimized Look-Ahead Energy Management of a Parallel Hybrid Electric Vehicle. *IEEE/ASME Trans. Mechatron.*, 22(4):1497–1507, 2017.

[25] X. Han, H. He, J. Wu, J. Peng, and Y. Li. Energy management based on reinforcement learning with double deep Q-learning for a hybrid electric tracked vehicle. *Appl. Energy*, 254(113708):1–10, 2019.

[26] Y. Li, H. He, A. Khajepour, H. Wang, and J. Peng. Energy management for a power-split hybrid electric bus via deep reinforcement learning with terrain information. *Appl. Energy*, 255(113762):1–13, 2019.

[27] A. M. Ali, A. Ghanbar, and D. Soffker. Optimal Control of Multi-Source Electric Vehicles in Real Time Using Advisory Dynamic Programming. *IEEE Trans. Veh. Technol.*, 68(11):10394–10405, 2019.

[28] C. Vagg, S. Akehurst, C. J. Brace, and L. Ash. Stochastic Dynamic Programming in the Real-World Control of Hybrid Electric Vehicles. *IEEE Trans. Control Syst. Technol.*, 24(3):853–866, 2016.

[29] F. Wang, H. Zhang, and D. Liu. Adaptive dynamic programming: An introduction. *IEEE Comput. Intell. Mag.*, 4(2):39–47, 2009.

[30] Z. Shen, C. Luo, X. Dong, W. Lu, Y. Lv, G. Xiong, and F. Y. Wang. Two-Level Energy Control Strategy Based on ADP and A-ECMS for Series Hybrid Electric Vehicles. *IEEE Trans. Intell. Transp. Syst.*, 23(8):13178–13189, 2022.

[31] L. Tang, G. Rizzoni, and S. Onori. Energy management strategy for HEVs including battery life optimization. *IEEE Trans. Transp. Electrif.*, 1(3):211–222, 2015.

[32] N. Guo, J. Shen, R. Xiao, W. Yan, and Z. Chen. Energy management for plug-in hybrid electric vehicles considering optimal engine ON/OFF control and fast state-of-charge trajectory planning. *Energy*, 163:457–474, 2018.

[33] S. Uebel, N. Murgovski, B. Bäker, and J. Sjöberg. A Two-Level MPC for Energy Management Including Velocity Control of Hybrid Electric Vehicles. *IEEE Trans. Veh. Technol.*, 68(6):5494–5505, 2019.

[34] T. Liu, W. Zhu, K. Tan, M. Liu, and L. Feng. A Low-Complexity and High-Performance Energy Management Strategy of a Hybrid Electric Vehicle by Model Approximation. In *Proc. IEEE 18th Int. Conf. Autom. Sci. Eng. (CASE)*, pages 455–462, Mexico City, Mexico, 2022. IEEE.

[35] A. Sciarretta, M. Back, and L. Guzzella. Optimal control of parallel hybrid electric vehicles. *IEEE Trans. Control Syst. Technol.*, 12(3):352–363, 2004.

[36] R. Xiong, J. Cao, and Q. Yu. Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle. *Appl. Energy*, 211:538–548, 2018.

[37] S. Onori, L. Serrao, and G. Rizzoni. *Hybrid Electric Vehicles: Energy Management Strategies*. Springer London, Heidelberg New York Dordrecht, 2016.

[38] T. Liu, L. Feng, and W. Zhu. Fuel Minimization of a Hybrid Electric Racing Car by Quasi-Pontryagin's Minimum Principle. *IEEE Trans. Veh. Technol.*, 70(6):5551–5564, 2021.

[39] M. Razi, N. Murgovski, T. McKelvey, and T. Wik. Predictive Energy Management of Hybrid Electric Vehicles via Multi-Layer Control. *IEEE Trans. Veh. Technol.*, 70(7):6485–6499, 2021.

[40] O. Sundström and L. Guzzella. A generic dynamic programming Matlab function. In *18th IEEE Int. Conf. Control Appl.*, pages 1625–1630, Saint Petersburg, Russia, 2009.

[41] D. P. Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, Belmont, Massachusetts, 1st edition, 2019.

[42] R. E. Bellman and E. S. Lee. History and development of dynamic programming. *IEEE Control Syst. Mag.*, 4(4):24–28, 1984.

[43] T. Liu, W. Tan, X. Tang, J. Zhang, Y. Xing, and D. Cao. Driving conditions-driven energy management strategies for hybrid electric vehicles: A review. *Renew. Sustain. Energy Rev.*, 151(111521):1–16, 2021.

[44] S. Onori and L. Tribioli. Adaptive Pontryagin's Minimum Principle supervisory controller design for the plug-in hybrid GM Chevrolet Volt. *Appl. Energy*, 147:224–234, 2015.

[45] A. Nguyen, J. Lauber, and M. Dambrine. Optimal control based algorithms for energy management of automotive power systems with battery/supercapacitor storage devices. *Energy Convers. Manag.*, 87:410–420, 2014.

[46] F. L. Lewis and D. Vrabie. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits Syst. Mag.*, 9(3):32 – 50, 2009.

[47] P. Baldi. Gradient Descent Learning Algorithm Overview: A General Dynamical Systems Perspective. *IEEE Trans. Neural Netw.*, 6(1):182 – 195, 1995.

Tong Liu (Member, IEEE) received the B.Sc. degree in electronics information engineering, and the M.Sc. degree in traffic information engineering and control from Chang'an University, China, in 2010 and 2013, respectively. He has ever worked as a traffic engineer at China Communications Construction Company Ltd. since 2013. He is currently pursuing the Ph.D. degree at the Department of Engineering Design, KTH Royal Institute of Technology, Stockholm, Sweden. His research interests include optimal control of dynamic processes, energy management of new energy vehicles, and structure optimization of hybrid powertrains.

Kaige Tan (Member, IEEE) received the B.Sc. degree in Mechatronics from Harbin Institute of Technology, Harbin, China, in 2018, and the M.Sc. degree in Mechatronics from KTH Royal Institute of Technology, Stockholm, Sweden, in 2020. He is currently working towards the Ph.D. degree with the Department of Engineering Design, KTH Royal Institute of Technology, Stockholm, Sweden. His research interests include optimization, reinforcement learning, and optimal filtering.

Wenyao Zhu received the B.S. degree in Electrical and Computer Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2018, and the M.S. degree in Embedded Systems from KTH Royal Institute of Technology, Kista, Sweden, in 2020. He is currently pursuing the Ph.D. degree at School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Kista, Sweden. His current research interests include edge computing and real-time systems.

Lei Feng (Member, IEEE) received the B.S. and M.S. degrees from Xi'an Jiaotong University, Xi'an, China, in 1998 and 2001, respectively, and the Ph.D. degree from the Systems Control Group, the University of Toronto, Toronto, ON, Canada, in 2007. In 2012, he joined the Mechatronics and Embedded Control System Division, the KTH Royal Institute of Technology, Stockholm, Sweden, where he is currently an Associate Professor. His main research interests include energy management control of mechatronic systems, autonomous driving, verification and control synthesis of cyber-physical systems, and supervisory control of discrete event systems.