IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 11, 2020, accepted December 25, 2020, date of publication January 4, 2021, date of current version January 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3048966

Deep Deterministic Policy Gradient Based Energy Management Strategy for Hybrid Electric Tracked Vehicle With Online Updating Mechanism

ZHIKAI MA[1], QIAN HUO[1], TAO ZHANG[2], JIANJUN HAO[1], AND WEI WANG[1]
[1]College of Mechanical and Electrical Engineering, Hebei Agricultural University, Baoding 071001, China
[2]China North Vehicle Research Institute, Beijing 100072, China

Corresponding authors: Tao Zhang (ztao1208@126.com) and Jianjun Hao (hjj@hebau.edu.cn)

ABSTRACT In this paper, an online energy management strategy (EMS) for hybrid electric tracked vehicle (HETV) is developed based on deep deterministic policy gradient (DDPG) with time-varying weighting factor to further improve economic performance of HETV and reduce computational burden. The DDPG is applied to model the EMS problem for the target HETV. Especially, a time-varying weighting factor is introduced here to update old network parameters with experience learned from most recent cycle segment. Afterwards, simulation is conducted to verify the effectiveness and adaptability of the proposed method. Results show that DDPG-based EMS with online updating mechanism can achieve nearly 90% fuel economy performance as that of dynamic programming while computational time is greatly reduced. Finally, hardware-in-loop experiment is carried out to evaluate the real-world performance of the proposed method.

INDEX TERMS Energy management, hybrid electric tracked vehicle, online updating mechanism, deep deterministic policy gradient.

I. INTRODUCTION

Hybrid electric vehicles (HEVs), which integrate the advantages of internal combustion engine (ICE) vehicles and pure electric vehicles are regarded as one of the most important categories of new energy vehicles [1]. By coordinating the working states of ICE and battery, Hybrid electric vehicle (HEV) can make engine work in high efficiency area most of the time and thus achieving perfect fuel economy performance.

Energy management strategy (EMS) is the key factor for realizing the energy-saving potential of HEVs. Currently, the strategy can be divided into two categories: rule-based strategies and optimization-based strategies [2]. The rule-based strategies often have an 'if-else' structure, which is simple and practical. It has been widely used in engineering practice. However, its performance is heavily dependent on engineer's experience. The optimization-based strategies have better theoretical optimization performance. However, due to the computational complexity of the strategy or the need for prior knowledge of the journey, the online application the algorithm still confronts great challenges.

The associate editor coordinating the review of this manuscript and approving it for publication was Haiyong Zheng.

In order to combine the advantages of the above two strategies, many researchers attempt to extract the rules from optimization-based algorithms or other advanced algorithms to improve the control effect of rule-based strategy algorithms [3], [4]. For example, Peng *et al.* [4] used dynamic programming (DP) to locate the optimal action of the engine in plug in hybrid electric vehicles (PHEVs) and proposed a rule-based EMS based on the results calculated by DP. Yan *et al.* [5] extracted the rules by wavelet transform technique and proposed a rule-based EMS for hybrid electric buses. To further improve fuel-saving performance, it is necessary to analyze the advantages and disadvantages of different optimization-based EMS strategies.

Optimization-based methods realize the power distribution by minimizing the value of the objective function while satisfying system constraints. Existing relevant methods include offline methods like DP [6]–[8], the Pontryagin's Minimum Principle (PMP) [9]–[13], convex programming [14]–[16], particle swarm optimization (PSO) [17]–[19], and online methods like equivalent consumption minimization strategy (ECMS) [20]–[22], model predictive control (MPC) method [13], [23]–[25].

DP is the most commonly used approach to develop EMS due to its excellent performance in solving global optimal control policy. However, the typical application of DP is

offline because of the requirement of completely knowing the driving cycle in advance and the problem of ''curse of dimensionality'' [26]. The PMP based EMS is another widely-used approach [10], [12]. This method is used to formulate the analytical necessary condition equation. However, when dealing with nonlinearities and complex constraints, this method is often incapable of obtaining optimal solutions in an efficient manner. Additionally, the convex optimization has been applied and compared with DP by Xiao *et al.* [14]. Philipp *et al.* [16] derived the global engine on/off conditions analytically and proposed an EMS for a series hybrid electric bus using convex optimization. The PSO, a population-based optimization method, also attracts attention in the development of EMS because of its high efficiency. It is first introduced to optimize a fuzzy controller of the EMS for a HEV [27], then many PSO and its variation based EMSs are developed. Although these above mentioned EMSs have great performance in fuel-saving and do make a contribution to the development of energy management, most of them are offline optimization and can hardly be applied in real-time.

Lots of attempts have been carried out to develop online methods. ECMS is one of such methods. ECMS, stemming from PMP [28] and first introduced in [29], transfers the original global optimization into an instantaneous one to optimize the instantaneous power demand distribution. However the optimal equivalent factor can only be determined when the power demand of the whole cycle is fully known in advance. Further, the adaptive ECMS based on driving cycle recognition is proposed in [20]–[22], which can be applied in real-time. But its fuel economy is inferior to that of offline optimization. Considering the inevitable randomness existing in driving behavior, MPC-based methods still have limitations when predicted speed deviates largely from the real value.

With the popularization of artificial intelligence, reinforcement learning (RL), an intelligent control method, has attracted increasing attention in the fields of EMS for HEVs [9]. The researchers tried to construct a rule-based strategy based on RL, which eliminated the need for prior knowledge of the journey and realized the online application with better optimization performance. RL has achieved remarkable results in the field of control, including energy management for HEVs. Liu *et al.* proposed a real-time RL-based EMS, in which the Kullback-Leibler (KL) divergence rate was adopted as an on-line updating trigger condition for the RL-based policy [30]. Yuan *et al.* demonstrated the adaptability, optimality, and the learning ability of RL-based energy management by numerical simulation on several different driving schedules [31], [32]. Although the above RL-based energy management can achieve excellent performance, they all need discretization of the state and action variables when these variables are continuous rather than discrete, which means there is a trade-off between optimization performance and computational cost. In addition, the learning process is offline and cannot ensure good adaptability.

To overcome above shortcomings, recently Deep Deterministic Policy Gradient (DDPG) based EMS methods become increasingly popular. In DDPG, two networks, namely actor network and critic network, are used to realize value function processing with continuous state space. For example, Ref [33] solved the multi-objective energy management optimization problem with large control variable space by combining battery characteristics and prior knowledge of engine high-efficiency working area using DDPG. In Ref [34], the global optimal control policy obtained by DP is used as expert knowledge to train DDPG model, and driving data are collected to replace DP based control. Ref [35] systematically integrated terrain information into the energy management problem of power split hybrid electric bus. Through the improvement of DDPG algorithm, the optimal energy management strategy can be searched in the discrete continuous mixed action space. Despite above contributions, the parameters of DDPG in most existing researches are basically fixed when the training process is over. However, in real-world driving scenario, the driver's driving style will change dynamically with the traffic situation and driver's state. Thus above methods lack adaptiveness to ever-changing driving style.

To overcome above shortcomings, this paper proposed a Deep Deterministic Policy Gradient (DDPG) based EMS with updating mechanism for HETV. To increase the adaptability of the method, an online updating framework is proposed to update network parameters when desired driving power has changed greatly. A time-varying weighting factor or adjust factor is introduced here to combine the experience learned from history cycle and current cycle. The online updating framework is based on the assumption that the driver's driving behavior has similar characteristics in adjacent time intervals. Thus, the derived control policy based on recent driving data has desirable performance in the short future.

Compared with RL and DP, DDPG-based EMS with the proposed framework is observed to achieve a better fuel economy and lower computational burden. The adaptability of the updating framework is validated by numerical simulation on a combined driving cycle. The contribution of this paper can be summarized as follows:

1) An energy management strategy based on deep reinforcement learning is proposed. The obtained control strategy is presented in the form of an artificial neural network, which can be implemented in real-time. 2) Both of the state variables and control variables are continuous. Eliminating of discretization makes the obtained policy more accurate and reliable. 3) An online updating framework of the energy management strategy is proposed, increasing the adaptability of the EMS. 4) Simulation and hardware-in-loop tests are conducted to validate the fuel economy and real-time performance. The most outstanding novelty of our work is introduction of the time-varying weighting factor, which can greatly increase the optimality of the method in dynamic driving conditions.

The remainder of this paper is organized as follows. In Section II, the mathematical model of the series HETV is established and verified, and the energy management problem is formulated. The DDPG-based energy management strategy and an online updating framework are developed in Section III. Section IV shows the computer simulation and the hardware-in-loop experiment result, followed by the key conclusions in Section V.

II. ENERGY MANAGEMENT PROBLEM FORMULATION

A. VEHICLE CONFIGURATION

The structure of the series hybrid system is shown in Fig.1, which mainly includes engine-generator set (EGS), power battery pack, motor drive system, hybrid control unit (HCU) and power distribution unit (PDU). PDU is responsible for regulating the power output between engine and battery.

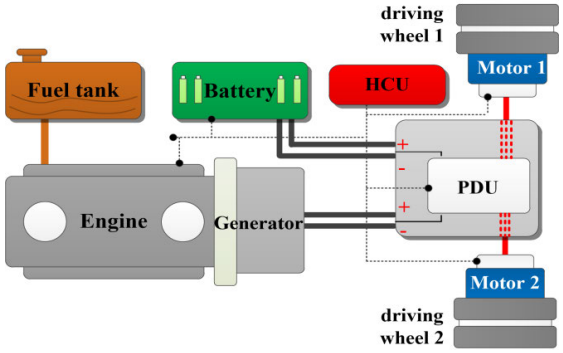


FIGURE 1. The configuration of the HETV powertrain.

The main component specifications of the HETV are shown in Table 1.

B. VEHICLE MODELING AND VALIDATION

1) VEHICLE POWER DEMAND MODEL

The primary purpose of EMS is to optimize the power distribution between different power sources. The power demand of a tracked vehicle contains two parts, namely the heading power and steering power, and can be calculated as follows:

$$\begin{cases} P_d = \left(mgf_r + \dfrac{C_D A}{21.15}v^2 + \delta m\dfrac{dv}{dt}\right)\cdot v \\ \qquad\quad + M\cdot\dfrac{v_{out}-v_{in}}{B} \\ M = \dfrac{1}{4}\mu_t mgL \\ \mu_t = \dfrac{\mu_{max}}{0.925+0.15R/B} \\ R = \dfrac{B}{2}\cdot\dfrac{v_{out}+v_{in}}{v_{out}-v_{in}} \\ P_m = P_d\cdot\eta^{-sgn(P_r)} \end{cases} \quad (1)$$

where v denotes the vehicle speed, M refers to the yaw moment of the vehicle, v_{out} and v_{in} denote the speed of the outside and inside track respectively, and R denotes the turning radius of the vehicle; P_m is the required electric power; η is the total efficiency coefficient of the track, the motor, and motor controller, which is obtained by field test of the HETV. Other vehicle parameters of the power demand model are listed in Table 2

TABLE 1. Main components specifications of the HETV.

Parameter Name		Value
EGS	Engine type	Diesel
	Engine displacement	1.06 liter, turbine
	Generator type	PMSM
	Power	30kW
Battery	Battery type	NCM
	Battery capacity	39Ah
	Normal voltage	345.6V
Driving motor	Motor type	PMSM
	Continuous power	25kW
	Continuous torque	60Nm

TABLE 2. Parameters of the power demand model.

Parameter	Physical meaning	Value
m	Curb weight of the HETV	1200kg
g	Gravitational acceleration	9.8kg/m^2
C_D	Coefficient of air resistance	0.7
A	Windward area	1.12
f_r	Rolling resistance coefficient	0.05
δ	Coefficient of mass increasing	1.2
L	Contact length of the track on the ground	1.6m
η	Efficiency coefficient	0.81
μ_{max}	Maximum coefficient of lateral resistance	0.8

Fig.2 shows the verification result of the power demand model against experimental data which is collected by the vehicle sensor. The relative error of the power demand model is 3.7%, which is calculated by the ratio of root mean square error to the mean value of the experimental data.

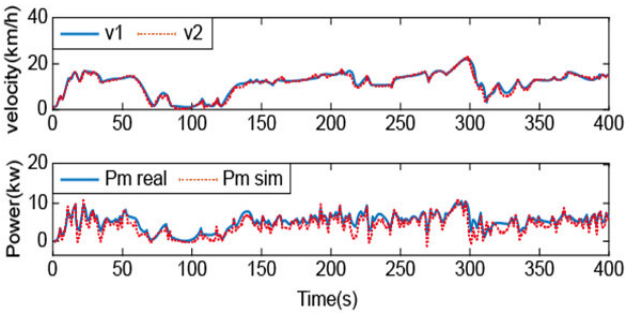


FIGURE 2. Framework of deep learning regression machine model.

2) EGS MODEL

In the EGS, a diesel engine and a permanent magnet synchronous AC generator are combined to generate electricity. An uncontrolled rectifier is used to transform the AC voltage to DC voltage. The model of the diesel engine and the generator are built based on the data achieved from the bench experiment. The engine fuel consumption map is expressed as the relationship in terms of engine speed and torque by a non-linear 3D map. Similarly, the modeling process of

the generator is the same as that of the diesel engine. Fig.3 presents the efficiency map of both the engine and generator.

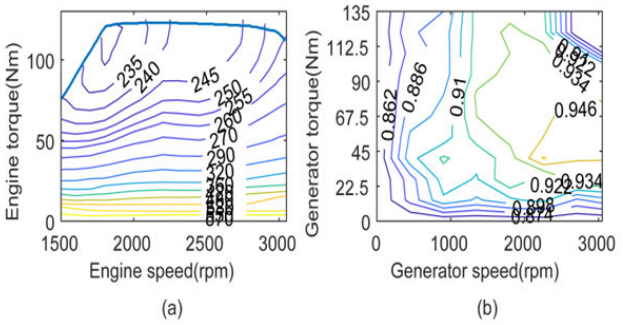


FIGURE 3. Efficiency map of engine and generator. (a) Efficiency map of the engine. (b) Efficiency map of the generator.

The output voltage and the electromagnetic torque of the generator are calculated as:

$$\begin{cases} U_g = K_e\omega_g - K_x\omega_g I_g \\ T_g = K_e I_g - K_x I_g^2 \end{cases} \tag{2}$$

where U_g and I_g are the output voltage and current of the generator, ω_g is the generator speed, T_g is the torque of the generator; K_e and K_x, the electromotive force coefficient and the electrical resistance coefficient, are 1.632 Vsrad^{-2} and 0.001NmA^{-2} respectively.

3) BATTERY MODEL

The battery in this research is a nominal 345.6V lithium-ion battery with a capacity of 15kWh. Since the HETV has a thermal management system, the impact of temperature change on the battery is not considered in this study. A straightforward and practical internal resistance battery model is used to characterize the dynamics of the battery, which can be expressed as follow:

$$\begin{cases} U_b = V_{oc} - I_b\left(R_{int}\right) \\ \dfrac{dSOC}{dt} = -\dfrac{V_{oc} - \sqrt{V_{oc} - 4R_{int}\cdot P_b}}{2R_{int}C} \end{cases}$$
$$R_{int} = \begin{cases} R_{int\_ch}, & I_b > 0 \\ R_{int\_dis}, & I_b < 0 \end{cases} \tag{3}$$

In the above formula, U_b is the output voltage of the battery, which is determined by the current and SOC of the battery, as shown in Fig.4; V_{oc} is the open-circuit voltage of the battery, which can be expressed as a relationship by a table in terms of the I_b and SOC; I_b is the battery current, SOC is the state of charge of the battery, C is the electric capacity of the battery, and P_b is the power of the battery; R_{int} is the electric resistance of the battery, which is different in the process of the charging and discharging; the charge resistance R_{ch} and discharge resistance R_{dis} can be obtained through the experiment, as shown in Fig.4.

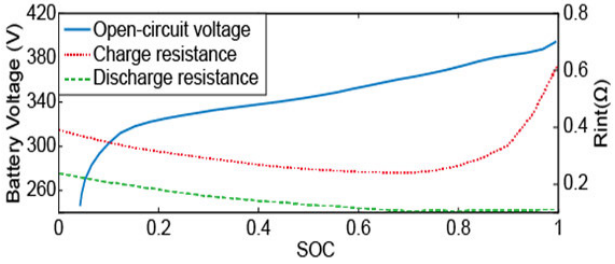


FIGURE 4. The open-circuit voltage and internal resistance of the battery.

4) POWER DISTRIBUTION MODEL

Due to the series architecture of the HETV, the power distribution of the powertrain must satisfy the following equations:

$$\begin{cases} P_m = P_g + P_b \\ P_g = U_{dc} I_g \\ P_b = U_{dc} I_b \\ U_{dc} = V_{oc} - I_b R_{int} \\ U_{dc} = K_e\omega_g - K_x\omega_g I_g \end{cases} \tag{4}$$

where U_{dc} is the bus voltage. The powertrain models are verified against experimental data as shown in Fig.5. The relative error of the EGS model is 3.5%. As for the battery model, we take experimental current as the input of the battery model and get the output of the battery model. The verification results indicate that the battery model accurately reflects the battery characteristics. The relative errors of the battery voltage and SOC are 4.3% and 3.9% respectively.

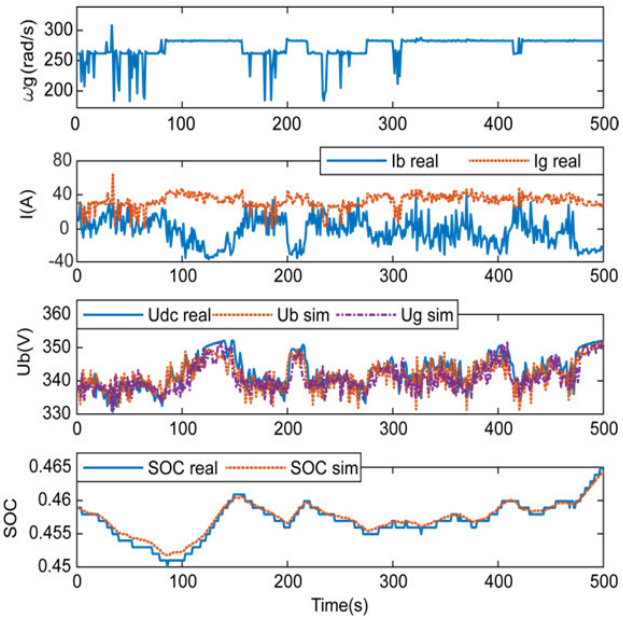


FIGURE 5. Validation of the powertrain models.

C. ENERGY MANAGEMENT PROBLEM FORMULATION

The main purpose of EMS is to minimize the fuel consumption while satisfying required constraints, which means it

needs to find an optimal control policy π thus best economy performance can be achieved for a driving cycle with start time t_0 and end time t_f. The objective function J is:

$$\min_{\pi}\left\{J=\int_{t_0}^{t_f} f_{rate}\left(n_{eng},T_{eng}\right)dt+k_f \cdot\left|\left(SOC-SOC(t_0)\right)\right|\right\} \tag{5}$$

subject to

$$\begin{cases} \dot{x}=f\left(x,u,P_r\right) \\ \left|\dot{n}_e\left(t\right)\right|<\Delta n_0 \\ T_e=thr\cdot T_{e\_max}\left(n_e\right) \\ 0<thr<1 \\ U_{dc\_min}<U_{dc}<U_{dc\_max} \\ n_{e\_min}<n_e<n_{e\_max} \\ I_{bat\_ch\_max}<I_{bat}<I_{bat\_dis\_max} \\ 0<I_g<_{g\_max} \\ SOC_{min}<SOC<SOC_{max} \\ k_f=\begin{cases}1000, t=t_f \\ 0, t_0<t<t_f\end{cases} \end{cases} \tag{6}$$

where f_{rate} is the instantaneous fuel consumption rate, which is a function of engine speed and engine torque. $x=[n_e, SOC, P_r, v]$ is chosen as the state variable and throttle opening of the engine is the control variable represented by u. The function f represents the system dynamics in Eq. (1)-(4). The system constraints in Eq.(6) needs to be satisfied. $\left|\dot{n}_e\left(t\right)\right|<\Delta n_0$ represents a physical limit on the change in speed per unit time; $T_e=thr\cdot T_{e\_max}\left(n_e\right)$ represents the engine torque limit; thr represents the opening of the throttle valve; $(U_{dc\_min},\ U_{dc\_max})$ represents the range of bus voltages; $(n_{e\_min},\ n_{e\_max})$ represents the range of engine speeds, $(I_{bat\_ch\_max},\ I_{bat\_dis\_max})$ represents the current range of the battery charging and discharging; $0<I_g<_{g\_max}$ means that the current of the generator is positive and less than the maximum. Since the HETV in this study cannot be charged externally, $SOC\left(t_f\right)$ is constrained to be equal to its initial value. A large penalty factor k_f is used for punishment if this constraint is not met. The penalty term $k_f\cdot\left|SOC-SOC(t_0)\right|$ is incorporated to ensure SOC-sustainability. The penalty is equal to zero only when the final SOC is exactly the same as the initial SOC, otherwise, additive penalty will be added to the objective function. The proposed method tries to minimize the objective function, thus the SOC-sustainability is considered and minimized in the optimization process.

III. DEEP DETERMINISTIC POLICY GRADIENT FOR ENERGY MANAGEMENT

A. DDPG STRATEGY STRUCTURE

According to the theory of RL, at each time step, the agent chooses an action a_t based on the control policy π, which maps current state s_t to action a_t. When agent applies the action, it will receive an instantaneous reward r_t sent by the environment and transits to a new state s_{t+1}. The target of RL is to find an optimal control policy thus accumulative reward weighted by discounting factors can be maximized. Because in this paper, the objective is to minimize the total fuel consumption, a negative sign is added to the instantaneous reward r_t.

$$\begin{cases} R_t=\sum_{k=0}^{\infty}\gamma^k r_{t+k+1} \\ r_t=-f_{rate}\left(n_{eng}(t),T_{eng}(t)\right)-k_f \cdot\left|SOC(t)-SOC(t_0)\right| \end{cases} \tag{7}$$

where $\gamma\in[0,\ 1)$ is the discounting factor, which discounts the future reward to current time stamp.

DDPG adopts Actor-Critic framework combining value-based and policy-based RL, and uses deep Q-network (DQN) to deal with continuous control problems. Firstly, a policy network with observation state as network input and control action as network output is defined. Then the control strategy can be parameterized as follows:

$$\pi_\theta(s,a)=P[a|s,\theta] \tag{8}$$

Because DDPG learns the deterministic strategy, the control action in the continuous space is a deterministic value determined by the observation state s. The problem shifts to find the appropriate network parameters to maximize the accumulative reward. For continuous problems, the Q-value function is parameterized directly by a Q-network with network parameters ϕ:

$$Q_\phi(s,a)=R_{t+1} \tag{9}$$

The objective function of DDPG can be further expressed as follow:

$$\begin{cases} J\left(\phi\right)=\min_\phi E_\pi\left[\frac{1}{2}\left(y_t-Q_\phi\left(s_t,a_t\right)\right)^2\right] \\ J\left(\theta\right)=\max_\theta E_\pi\left[Q_\phi\left(s_t,u\left(s_t\right)\right)\right] \end{cases} \tag{10}$$

With

$$y_t=r_t+\gamma Q_{\phi'}\left(s_{t+1},a_{t+1}\right) \tag{11}$$

In the above equations, $J\left(\theta\right)$ is set to maximize the expectation of $Q\left(s_t,u\left(s_t\right)\right)$ by updating the network parameters θ, so as to get the maximum cumulative reward. $J\left(\phi\right)$ is set to minimize the expectation of the loss function, which is defined as the square error between Q and y. y_t is the value of target Q with parameter of ϕ'. r_t is the instant reward at time t by taking action a_t, $Q_\phi\left(s_t,a_t\right)$ is the state-action value function at state s_t and a_t. Solve the gradient of the above equations, the updating of the corresponding parameters can be expressed as follow:

$$\begin{cases} \nabla_\phi J=E_\pi\left[\left(r_t+\gamma Q_{\phi'}\left(s_{t+1},u'\left(s_{t+1}|\theta'\right)\right)-Q_\phi\left(s_t,a_t\right)\right)\nabla_\phi Q_\phi\left(s_t,a_t\right)\right] \\ \nabla_\theta J=E_\pi\left[\nabla_a Q_\phi\left(s,a\right)|_{s=s_i,a=u(s_i)}\nabla_\theta u\left(s|\theta\right)|s_i\right] \end{cases} \tag{12}$$

where $\nabla_a Q\left(s,a|\phi\right)$ is the gradient of the action-value function $Q\left(s,a\right)$ in terms of a, which denotes the update direction in order to maximize Q. $\nabla_\theta u\left(s|\theta\right)|s_i$ is the gradient of the

control policy u in terms of its parameters θ, which denotes how to update θ to make the control policy more likely to do the action. By combining the two parts above, $\nabla_\theta J$ can update the policy network parameters to maximize the value of Q, and $\nabla_\phi J$ can update the critic network parameters to minimize the value of the loss. After obtained the gradients, the Adam Optimizer in Tensor Flow is used to update the online network parameters. To make the training process more stable, a copy of the online policy and Q networks are used for softly updating the target networks with weight τ as follows:

$$\begin{cases} \phi' \leftarrow \tau\phi' + (1-\tau)\,\phi \\ \theta' \leftarrow \tau\theta' + (1-\tau)\,\theta \end{cases} \tag{13}$$

In Section II, we specified the optimization target, state equation, state variables and control variables. Eq.(1)∼Eq.(4) are state equations. Eq.(5) is optimization target. State vector is $x = [n_e, SOC, P_r, v]$. Control variable is the throttle opening of the engine, which is also the "action" in DDPG algorithm. The negative value of instantaneous fuel consumption is the reward. The DDPG-based EMS for HETV in this paper is shown in Fig.6. The environment includes the vehicle model and the driving cycle for training. At each training step, the agent selects an action according to the current policy network and its current state. However, to make the agent has a more holistic understanding of the environment, at some time steps, the agent will randomly choose an action to enlarge its exploration scope. The agent will store a tuple (s_i, a_i, r_i, s_{i+1}) in its memory reply buffer at each time step and use mini-batch in the buffer to train the two networks at fixed time interval. The training process of the whole driving cycle is repeated until convergence when the optimal strategy network is obtained.

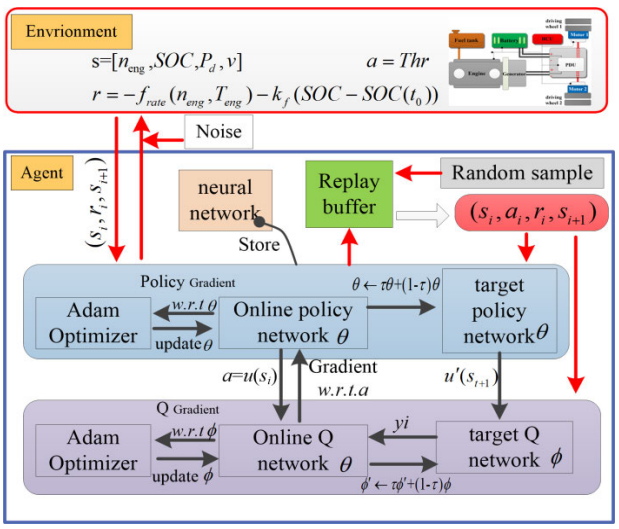


FIGURE 6. The basic structure of DDPG-based EMS.

B. ONONLINE UPDATING FRAMEWORK

Because driving conditions faced by the off-road tracked vehicle are very complex and random, it is crucial for HETV's EMS to have strong adaptability to different driving conditions. Traditionally, an optimal policy can only be the best when it is applied on the driving cycle for training. In order to improve the adaptability of DDPG-based EMS, an online updating EMS framework is proposed based on the assumption that the recent driving cycle is similar to the next future cycle.

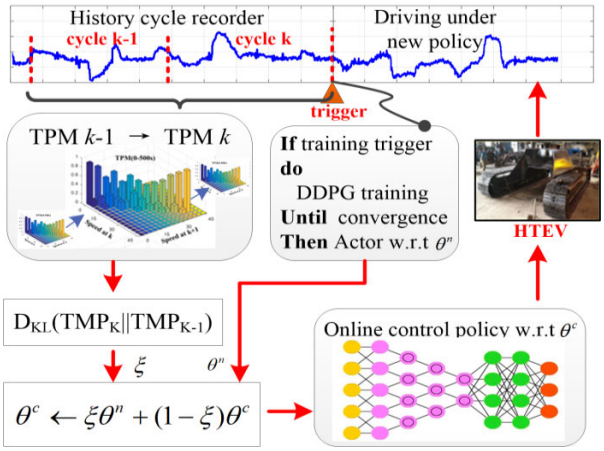


FIGURE 7. The diagram of the online updating framework.

Fig.7 shows the diagram of the online updating framework. Firstly, the network is trained using existing driving cycles to obtain the initial network parameters. Then, data are collected in real-time when HETV operates. When dataset reaches a predetermined length, a new training procedure is triggered. Thirdly, when the new training process converges, the control strategy is softly updated to obtain stable control performance. Update can be expressed as:

$$\theta^c \leftarrow \zeta\theta^n + (1-\zeta)\,\theta^c \tag{14}$$

where θ^c is the parameters of the online control policy network, θ^n is the parameters of the policy network. ζ is the time-varying weighting factor or adjust factor, which is used to update θ^c by adding information of history driving cycle introduced by θ^n. To determine the value of ζ, the characteristic of power demand needs to be modeled. In this paper, the transition probability matrix (TPM) is used to quantify driving cycle's power transition property, which can be calculated as follows:

$$\begin{cases} p_{ij}(n) = P(x_{n+1} = j | x_n = i) = N_{ij}/N_i \\ N_i = \sum_{j=1}^{m} N_{ij} \end{cases} \tag{15}$$

where $p_{ij}(n)$ represents the transition probability as the power demand transfers from i at time n to j at next time $n+1$, N_{ij} is the total transition number when power demand transfers from i to j, N_i denotes total transition number starts from i.

Kullback-Leiber (KL) divergence rate is adopted to quantify the difference between two adjacent TPMs of the nearest past two driving cycle segments. KL divergence rate is calculated as follows:

$$D_{KL}(P_k||P_{k-1}) = \sum_i P_k(i) \log \frac{P_k(i)}{P_{k-1}(i)} \tag{16}$$

where $P_k(i)$ is the TPM of kth driving cycle segment. To satisfy the existence of logarithm, $P_k(i)$ and $P_{k-1}(i)$ must be positive values. Thus a matrix, whose elements are all a same tiny constant, is added to the TPM with no change on the probability distribution:

$$P' = P + \sigma I \tag{17}$$

where σ is a tiny constant, I is a matrix with all elements are 1.

Because the vehicle's velocity at adjacent points are very close in most of time, TPM has highest value in diagonal direction. For those elements which are highly deviated from diagonal, values of most of them are zero, so the TPMs are sparse matrixes. If a matrix has too many zero velues, its KL divergence rate with another matrix can be bigger than 1. Therefore, KL divergence rate cannot be used as the weighting factor directly because (1-ζ) in Eq.(14) needs to be positive. In order to make the weighting factor fall in the range (0, 1), a modified sigmoid function is used to construct the relationship between KL divergence rate and the weighting factor:

$$\zeta = \frac{2}{1 + e^{-\xi D_{KL}}} - 1 \tag{18}$$

where D_{KL} is the KL divergence rate, ξ is a coefficient which controls the mapping relationship.

The procedure of the online updating DDPG-based EMS is shown in Table 3. All of the layers of actor and critic networks are fully connected layers. The DDPG algorithm settings and network parameters are listed in Table 4.

IV. RESULTS AND DISCUSSION

A. SIMULATION RESULT

To verify the effectiveness of the proposed DDPG-based EMS, a real-world driving cycle, as shown in Fig.8, is collected and used to train the DDPG network. Fig.9 shows the training loss and corresponding cumulative reward as the training process proceeds. It can be seen that after about 30 thousand steps, the loss becomes stable, which means the network parameters have converged. In addition, the cumulative reward achieves nearly its maximum value after corresponding episodes. It can be concluded from above observations that the proposed algorithm has ideal convergence property.

In order to validate the optimality of the proposed method, DP-based global energy management, together with Q-learning and DDPG, is adopted as a benchmark to make a comparative study. The Classic calculation flowchart of Q-learning is shown in Table 5. For detailed calculation process of DP, please refer to Ref [36]. The control problem definition and state equation of DP and Q-learning are the same

TABLE 3. The pseudocode of the online updating DDPG-based EMS.

Algorithm: The online updating DDPG-based EMS

1: Initialize parameters: online policy network θ^c, Q-function network ϕ^c, empty replay buffer B.
2: In the driving cycle:
3: For each time step, run the vehicle with online policy $\pi_{\theta^c}(s,a)$, and record the history cycle with timer C_l
4: **If** C_l mod L_c =0:
5: Trigger new DDPG training
6: Calculate TPM_k, $D_{KL}(TPM_k \| TPM_{k-1})$, and ζ
7: Initialize parameter for new training: $\theta^n \leftarrow \theta^c$, $\phi^n \leftarrow \phi^c$, $\theta' \leftarrow \theta^c$, $\phi' \leftarrow \phi^c$
8: **For each episode:**
9: Reset environment state to s_0
10: **For** t=1 to L_c **:**
11: Observe state s_t and select action $a_t = clip\left(u_{\theta^n}(s_t) + \varepsilon, a_{Low}, a_{High}\right)$, where ε~N
12: Execute a_t in the environment, observe next state s_{t+1}, reward r_t
13: Store (s_t, a_t, r_t, s_{t+1}) in the replay buffer B
14: Randomly sample a batch of transitions, $B = \{(s_i, a_i, r_i, s_{i+1})\}|_{i=1...N}$ from B
15: Compute Q gradient w.r.t ϕ

$$\begin{cases} y_i = r + \gamma Q_{\phi'}\left(s_{i+1}, u'(s_{i+1}|\theta')|\phi'\right) \\ L = \frac{1}{N}\sum_i \left(y_i - Q(s_i, a_i|\phi^n)\right)^2 \end{cases}$$

Back-propagation to get gradient: $\nabla_\phi L$

16: Update Q net: ϕ^n
17: Compute policy gradient w.r.t θ^n:

$$\nabla_{\theta^n} J(u) = \frac{1}{N}\sum_i \left[\nabla_a Q(s_i, u(s_i)|\phi^n)|_{s=s_i, a=u(s_i)} \nabla_{\theta^n} u(s_i|\theta^n)|s_i\right]$$

18: update action net: θ^n
19: update target net θ', ϕ' using Eq.(13)
20: **end for**
21: **until** convergence
22: update the online control policy network: $\theta^c \leftarrow \zeta\theta^n + (1-\zeta)\theta^c$
23: **until** the end of the driving cycle

TABLE 4. The parameter settings of the algorithm.

Parameter name	Value
Replay buffer size B	1000
Step size in each episode *max_steps*	Length of the driving cycle
Minibatch size N	100
Discount factor τ	0.95
Learning rate of Action La	0.01
Learning rate of Critic Lc	0.005
Action network layers number	3
Action network unit number	200
Critic network layers number	4
Critic network unit number	150

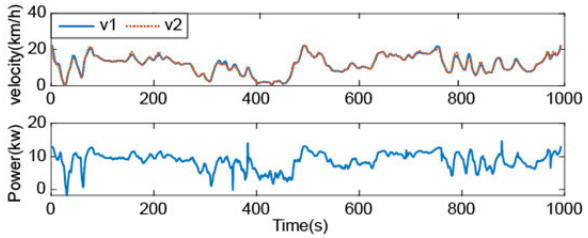


FIGURE 8. Driving schedule and power demand of the HETV.

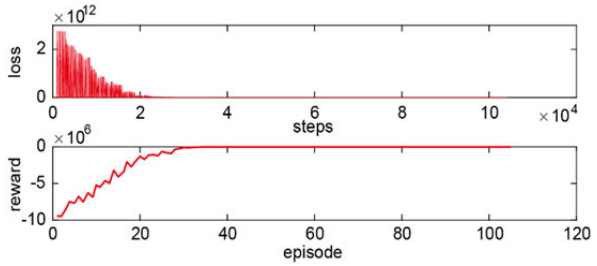


FIGURE 9. The loss and cumulative reward of the training process.

TABLE 5. The pseudocode of t DP-based EMS.

Algorithm: Q-learing (off-policy TD control) for estimating

Algorithm parameters: step size $\alpha \in (0,1]$, small $\varepsilon > 0$

Initialize $Q(s, a)$, for all $s \in S^+$, $\alpha \in A(s)$, arbitrarily except that Q(terminal, .)=0

Loop for each episode :

Initialize S

Loop for each step of episode:

Choose A from S using policy derived from Q(e.g., $\varepsilon-$greedy)

Take action A, observe R, S'

$Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,A)]$

$S \leftarrow S'$

Until S is terminal

as that of DDPG. Namely, the state equations of all above three algorithms are Eq.(1)∼Eq.(6), the control variable is the throttle opening, the state vector is $x = [n_e, SOC, P_r, v]$, which is a combination of engine speed, battery state-of-charge, required power and vehicle speed. The discretization of DP and Q-learning are the same in this paper. The discretization granulation of engine speed, battery state-of-charge, required power and vehicle speed are 20rpm, 5%, 10kW and 5km/h respectively.

Fig.10 shows the trajectories of the battery SOC, and it can be seen that all of the three methods have excellent performance to keep SOC sustainable at the final time. Overall, compared with SOC trajectories of Q-learning, the trend of DDPG is more similar to that of DP. During 0-100s, the curve of DDPG mostly coincides with the curve of DP where their EGSs stop at most of time. Similarly, during the 150-250s, the EGS of both DDPG and DP stop, leading to the same downward trend in SOC curves of the two methods. During 250s-521s, the EGS of DP is continuously at a stop state at most of time, which results in a fall of SOC to 0.731.

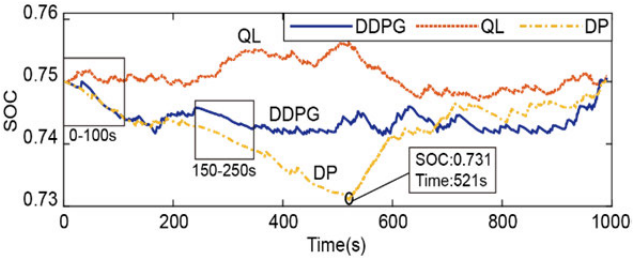


FIGURE 10. The trajectories of the battery SOC of the three methods.

While the EGS of DDPG starts to work to maintain SOC around 0.743. This is the main reason for the difference in fuel consumption between the two methods. After 700s, the EGS on/off state of both DDPG and DP are almost the same to finally drive the SOC back to the original value.

Fig.11 shows the engine working points for the three methods. It can be found that the engine points distributed in range where fuel consumption rate is 245∼270g/kWh for DP method are relatively fewer. More of them are scattered in range where fuel consumption rate is 235∼240g/kWh. Meanwhile Q-learning method has fewest engine working points distributed in the range where fuel consumption rate is below 240g/kWh. In addition, the result of DDPG is better than that of Q-learning while inferior to DP.

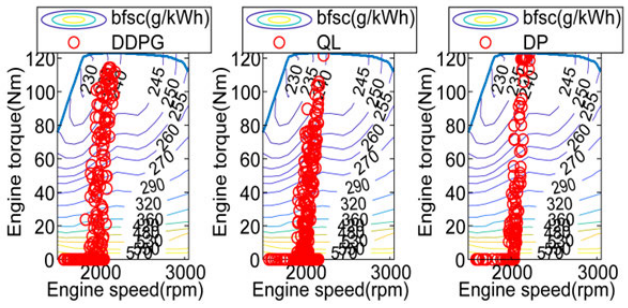


FIGURE 11. Engine working points of the three strategies.

To clearly evaluate the fuel economy of the proposed method, the result of DP is used as the benchmark. In Table 6, the economy performance of DP method is set as 100%, the calculation formula of fuel economy is defined as follows:

$$Fuel\ economy = (1 - \frac{FC_{QL} - FC_{DP}}{FC_{DP}}) \times 100\% \quad (19)$$

where FC is short for fuel consumption. As can be seen from the table, the DDPG-based EMS can achieve 91.3% fuel economy of DP benchmark, which is 6.4% better than Q-learning, benefiting from larger optimization space in DDPG algorithm with both continuous state and control variable. In addition, Table 7 shows the computation time of the three methods. Notably, Q-learning and DDPG-based strategies take less computation time than DP on the same driving cycle. The calculation time consumption of DDPG is 2.7% of the DP-based method and about 25% of the total

TABLE 6. Oil consumption of the three methods.

Algorithm	Fuel consumption (g)	Final SOC	Fuel economy (%)
DP	220.5	0.7501	100
Q-learning	259.7	0.7509	84.9
DDPG	241.3	0.7506	91.3

TABLE 7. Computation consumption of the three methods.

Algorithm	Computation time (s)	Proportion of the driving cycle (%)
DP	9005	908.7
Q-learning	7211	727.6
DDPG	251	25.3

length of the driving cycle. It can be seen that the calculation time is far shorter than the length of the driving cycle, that is to say, the online updating framework is feasible in terms of time consumption.

Generally, the computational burden of DP is usually more huge than the learning based method because of the following two reasons. Firstly, DP tries to find the optimal control strategy over the whole driving cycle, thus it needs to traverse all the possible state combinations for all time stamps. Secondly, DP needs discretization for all variables and the computational burden increases exponentially with the state and control variable numbers. In this paper, there are five variables needs to be discretized. therefore, the computation time of DP is relatively longer. However, for the learning-based methods, they adopt regression model, such as neural network to realize maneuver over continuous space and avoid discretization. The regression model parameters are usually fewer than all possible combinations of state and control variables, thus its computational burden is much lighter. In Ref [37], the computation time for DP is 46.11s while for Actor-Critic method is 3.62s.

B. ADAPTABILITY VALIDATION

In this scenario, a combined driving cycle, which consists of four different sub-cycles, is used to verify the adaptability of the proposed EMS. Fig.12 shows the combined cycle and the velocity characteristics of four sub-cycles. It can be seen that the four sub-cycles vary differently from each other. Thus unless the EMS has self-adaptive capability, the economic performance over the combined cycle cannot be desirable.

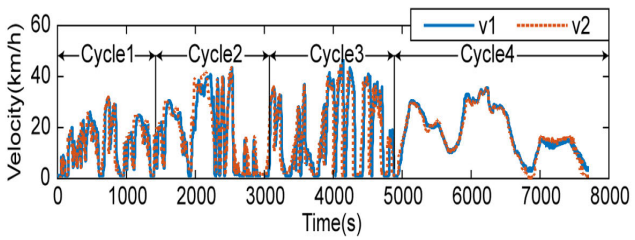


FIGURE 12. The combined driving cycle of HETV.

According to the proposed EMS, the combined cycle is divided into several segments with fixed length.

Fig.13 shows a part of the TPMs of the power demand for each cycle segment. In this paper, TPM is used to characterize the driving condition statistically. The difference of TPM in adjacent sampling intervals represents the change of driving condition. According to the quantitative change, the control strategy can be updated as follows: when the driving condition changes drastically, the control strategy believes more in the control strategy parameters obtained from most recent data. However, when the change of driving condition is not obvious, the change of control strategy is also small. The updated method is used to make the control strategy have better adaptability to driving conditions, so as to improve its ability to cope with the actual time-varying conditions.

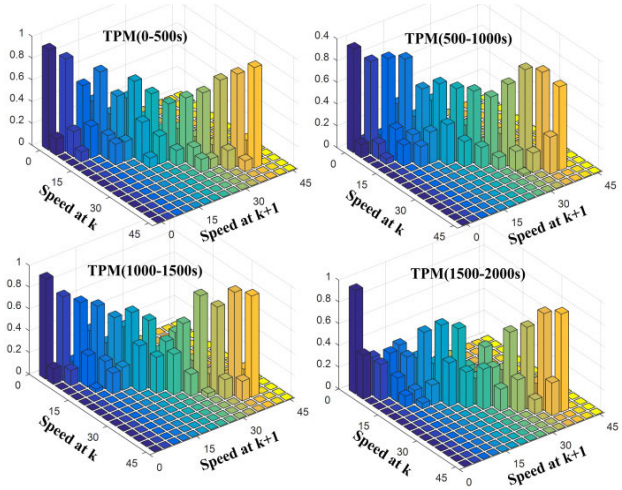


FIGURE 13. TPMs of several periods fixed-length driving cycle.

Fig.14 shows the KL divergence rate of TPMs of adjacent cycle segments, together with the corresponding adjust factor ζ. It can be seen that at 14^{th} update step, the KL divergence rate exceeds 1 but thanks to the sigmoid function of Eq.(18), the adjust factor is transformed below 1.

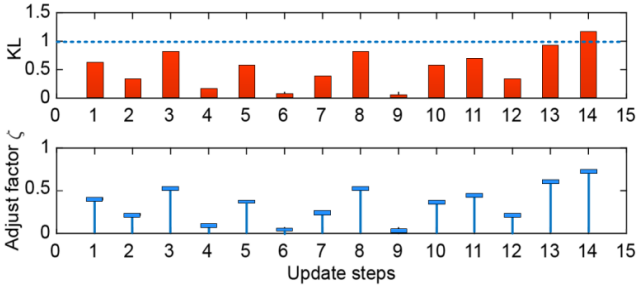


FIGURE 14. The KL divergence rate and adjust factor ζ.

Fig.15 shows the SOC trajectories of the three different methods, where DP represents dynamic programming, DDPG denotes the static DDPG-based method without policy updating and DDPGU is the updating DDPG-based method.

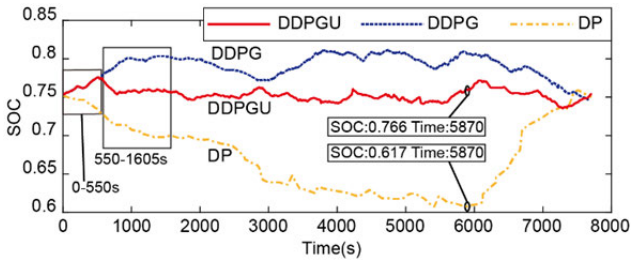


FIGURE 15. SOC trajectories of the three methods.

For DP method, the whole driving cycle is assumed to be known in advance. It can be seen that the global optimal solution obtained from DP results in the SOC mainly decreasing from start to 5870s and then fast increasing after 5870s. Finally, the SOC increases to its initial value at the end because of the SOC balance requirement. For the DDPG method, at most of time, its SOC is much higher than that of DP method. This is because its policy parameters is trained by the driving cycle shown in Fig.9, the policy is nearly optimal for the training cycle in Fig.9 but not for the test cycle in Fig.14. Thus its result SOC severely deviates from the optimal solution of DP. For the DDPGU method, its SOC fluctuates around its initial value because the policy is updated every 50s and the SOC balance constraint is introduced in every updating procedure. In addition, because the update is started from 500s, the SOC trajectories of DDPG and DDPGU are the same before 500s.

Fig.16 shows the engine's working points under three different control policies. It can be seen that all the three policies try to make the engine work around the area where engine speed is between 1800rpm and 2200rpm except for the situation when the engine torque is close to zero. In addition, DP method enables the engine have most points in high-efficiency area, where the BFSC of the engine is less than 230g/kWh. Compared with DDPG, DDPGU-based control policy has more points distributed in high-efficiency area. Additionally, fuel consumption after SOC-correction for the three policies is listed in Table 8. For the combined driving cycle, the fuel economy of the DDPGU can achieve 89.8% as that of the DP benchmark and increases 5.0% compared with DDPG without updating.

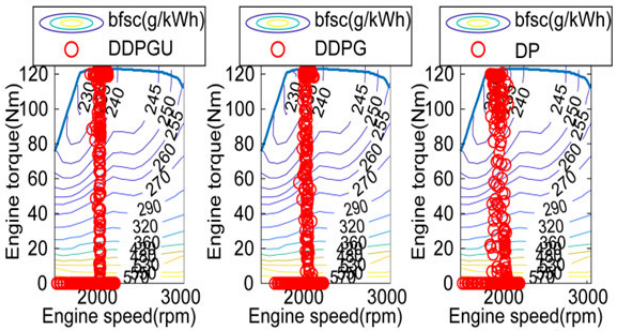


FIGURE 16. Engine working points of the three methods.

TABLE 8. Fuel economy comparison of the three policies.

algorithm	Fuel consumption (g)	Final SOC	Fuel economy(%)
DP	1520.7	0.751	100
DDPG	1789.1	0.7345	84.8
DDPGU	1691.3	0.7501	89.8

C. HARDWARE IN-LOOP EXPERIMENT

Because the driver's behavior is stable in most cases, the driving characteristic in the short future is similar to that at current in statistics. Thus, the derived control policy based on recent driving data has desirable performance in the short future despite driver's different actions.

To validate the performance of the proposed DDPG-based EMS with updating, the hardware-in-loop (HIL) experiment is conducted. As shown in Fig.17(b), the HIL test bench consists of an industrial personal computer (IPC), a powertrain plant of the HETV (dSPACE AutoBox), a driving simulator, an upper computer, and a CAN-Ethernet convertor (CANET). The driver manipulates the HETV vehicle model in the dSPACE target box through the driving simulator. The EMS controller outputs the target throttle opening according to the current vehicle state feedback from the vehicle model in the dSPACE AutoBox. The DDPG trainer collects the driving schedule data and the EMS training process will be triggered upon the data length reaches 500s. The newly trained control policy network parameters will be sent to the EMS controller for EMS updating through the Ethernet communication. The CANET is added to implement the communication between CAN and Ethernet.

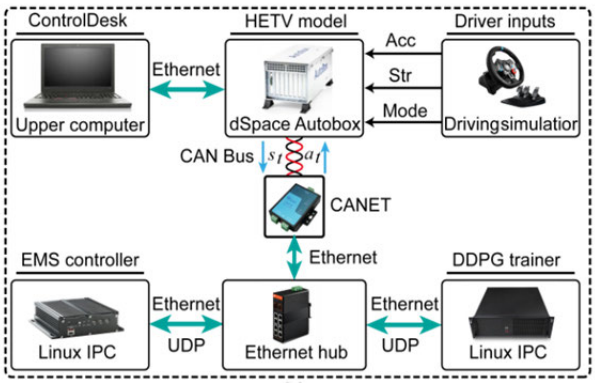


FIGURE 17. Architecture of the hardware-in-loop test bench.

The tests are carried out with the abovementioned test bench. The first one, named as HIL1, takes the driver's random operation as input to verify the real-time availability of the DDPGU. The input signals and the collected driving schedule during the HIL1 are plotted in Fig.18. The second HIL follows the driving schedule of HIL1 to reproduce the HIL process with DDPG. HIL2 is designed to work as a benchmark to illustrate the advantage of DDPGU. In addition, the third HIL test also follows the driving schedule shown in Fig.18 with DDPGU. However, the initial value and the

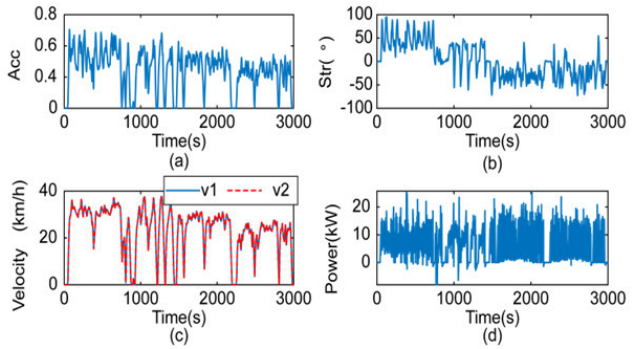


FIGURE 18. The input signals and driving schedule of the HIL. (a) The accelerator pedal signal. (b) The steering signal of the steering wheel. (c) The collected driving schedules of the two tracks. (d) The power demand on DC side of the two motors.

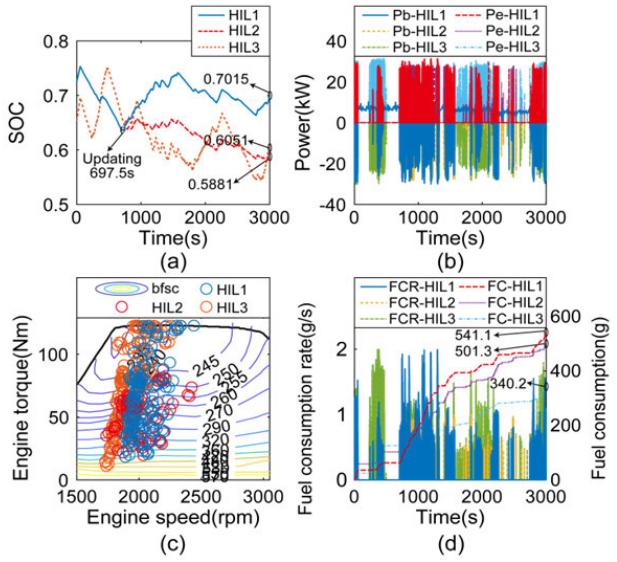


FIGURE 19. Results of three different HIL. (a) SOC trajectories. (b) Power distributions. (c) Engine working points. (d) Fuel consumption rate and fuel consumption.

final value in HIL3 are set as 0.65 to verify the impact of different SOC on fuel consumption. All three HIL tests can be implemented in real-time and the test results are shown in Fig.19. Fig.19(a) shows the SOC trajectories of the three HIL tests. The first updating happens at 697.5s, then, the SOC trajectories of HIL1 and HIL2 become significantly different. HIL1 can maintain the SOC around 0.75, while the SOC in HIL2 decreases to 0.5147 at the end. This is because the power demand of the HIL test is overall larger than the training driving cycle shown in Fig.8, which is used to train DDPG network parameters. This result in the throttle opening output by DDPG method is relatively lower for the HIL test situation. However thanks to the online updating mechanism, DDPGU can make the final SOC close to its initial value. Fig.19(c) shows that the engine working points of HIL1 are distributed in more efficient areas than HIL2. In addition, the engine working points of HIL3 are closer to the most efficient areas. It is because that lower SOC can make the engine operate at a lower speed. Therefore, on the premise of ensuring that the vehicle dynamic performance is not affected, appropriately reducing SOC is conducive to improving fuel economy. The power distribution of the three HIL tests is plotted in Fig.19(b). The sum cost of fuel and electric consumption is defined to measure the fuel economy of HETV. The price of diesel price is 6.67(CNY/L) and the electric price is 0.97 (CNY/kWh). The costs of the three HIL tests are 4.95 CNY, 5.27 CNY,4.59 CNY, respectively. The result shows that DDPGU increases fuel economy by 6.0% than DDPG.

V. CONCLUSION

In this paper, a continuous reinforcement learning algorithm, named DDPG, is applied to develop EMS for a HETV for better performance in terms of fuel economy and computation speed. Simulation results show that the proposed DDPG-based method is capable of achieving a sub-optimal fuel economy. Compared with discrete RL-based energy management strategies, the proposed method can increase fuel economy by 6.4%. It achieves 91.3% of the fuel-saving performance based on DP. Meanwhile, the calculation time of the proposed EMS is significantly shortened, accounting for about 25% of the length of the driving cycle. In addition, an online updating framework is proposed to improve the adaptability of DDPG-based EMS. Through scrolling self-learning from history driving data, an EMS, which dynamically adapts to the current driving cycle, is obtained. For the combined validation driving cycle, simulation results show that the proposed updating method increases fuel economy by 6% compared with the pre-trained original DDPG without updating and achieve 90% of the fuel economy performance based on DP. Moreover, in contrast to DP, no prior knowledge of the driving cycle is necessary for the proposed method, which is closer to the practical driving situation. Finally, a hardware-in-loop experiment is conducted and the result proves that the proposed algorithm can be applied in real-time.

REFERENCES

[1] C. Fabio, C. Andrea, and L. Giuseppe, "Hybrid electric vehicles: Some theoretical considerations on consumption behaviour," *Sustainability*, vol. 1, no. 4, p. 1302, 2018.
[2] Y. Zhou, A. Ravey, and M.-C. Péra, "A survey on driving prediction techniques for predictive energy management of plug-in hybrid electric vehicles," *J. Power Sources*, vol. 412, pp. 480–495, Feb. 2019.
[3] J. Wang, J. Wang, Q. Wang, and X. Zeng, "Control rules extraction and parameters optimization of energy management for bus series-parallel AMT hybrid powertrain," *J. Franklin Inst.*, vol. 355, no. 5, pp. 2283–2312, Mar. 2018.
[4] J. Peng, H. He, and R. Xiong, "Rule based energy management strategy for a series–parallel plug-in hybrid electric bus optimized by dynamic programming," *Appl. Energy*, vol. 185, pp. 1633–1643, Jan. 2017.
[5] M. Yan, M. Li, H. He, J. Peng, and C. Sun, "Rule-based energy management for dual-source electric buses extracted by wavelet transform," *J. Cleaner Prod.*, vol. 189, pp. 116–127, Jul. 2018.
[6] J. Liu, Y. Chen, W. Li, F. Shang, and J. Zhan, "Hybrid-trip-model-based energy management of a PHEV with computation-optimized dynamic programming," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 338–353, Jan. 2018.

[7] 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. Veh. Technol.*, vol. 64, no. 7, pp. 2792–2803, Jul. 2015.

[8] M. Zhao, J. Shi, and C. Lin, "Optimization of integrated energy management for a dual-motor coaxial coupling propulsion electric city bus," *Appl. Energy*, vol. 243, pp. 21–34, Jun. 2019.

[9] B.-H. Nguyen, R. German, J. P. F. Trovao, and A. Bouscayrol, "Real-time energy management of battery/supercapacitor electric vehicles based on an adaptation of pontryagin's minimum principle," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 203–212, Jan. 2019.

[10] C. Zheng, W. Li, and Q. Liang, "An energy management strategy of hybrid energy storage systems for electric vehicle applications," *IEEE Trans. Sustain. Energy*, vol. 9, no. 4, pp. 1880–1888, Oct. 2018.

[11] Z. Yuan, L. Teng, S. Fengchun, and H. Peng, "Comparative study of dynamic programming and Pontryagin's minimum principle on energy management for a parallel hybrid electric vehicle," *Energies*, vol. 6, no. 4, pp. 2305–2318, Apr. 2013.

[12] 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. Veh. Technol.*, vol. 67, no. 8, pp. 6803–6813, Aug. 2018.

[13] S. Xie, X. Hu, Z. Xin, and J. Brighton, "Pontryagin's minimum principle based model predictive control of energy management for a plug-in hybrid electric bus," *Appl. Energy*, vol. 236, pp. 893–905, Feb. 2019.

[14] R. Xiao, B. Liu, J. Shen, N. Guo, W. Yan, and Z. Chen, "Comparisons of energy management methods for a parallel plug-in hybrid electric vehicle between the convex optimization and dynamic programming," *Appl. Sci.-Basel*, vol. 8, no. 2, pp. 1–12, 2018.

[15] X. Hu, Y. Li, C. Lv, and Y. Liu, "Optimal energy management and sizing of a dual motor-driven electric powertrain," *IEEE Trans. Power Electron.*, vol. 34, no. 8, pp. 7489–7501, Aug. 2019.

[16] P. Elbert, T. Nuesch, A. Ritter, N. Murgovski, and L. Guzzella, "Engine On/Off control for the energy management of a serial hybrid electric bus via convex optimization," *IEEE Trans. Veh. Technol.*, vol. 63, no. 8, pp. 3549–3559, Oct. 2014.

[17] C. Yang, S. Du, L. Li, S. You, Y. Yang, and Y. Zhao, "Adaptive real-time optimal energy management strategy based on equivalent factors optimization for plug-in hybrid electric vehicle," *Appl. Energy*, vol. 203, pp. 883–896, Oct. 2017.

[18] 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*, vol. 160, pp. 154–170, Oct. 2018.

[19] T. Mesbahi, N. Rizoug, P. Bartholomeás, R. Sadoun, F. Khenfri, and P. L. Moigne, "Optimal energy management for a li-ion battery/supercapacitor hybrid energy storage system based on a particle swarm optimization incorporating Nelder–Mead simplex approach," *IEEE Trans. Intell. Veh.*, vol. 2, no. 2, pp. 99–110, Jun. 2017.

[20] A. Rezaei, J. B. Burl, B. Zhou, and M. Rezaei, "A new real-time optimal energy management strategy for parallel hybrid electric vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 2, pp. 830–837, Mar. 2019.

[21] Y. Zeng, J. Sheng, and M. Li, "Adaptive real-time energy management strategy for plug-in hybrid electric vehicle based on simplified-ECMS and a novel driving pattern recognition method," *Math. Problems Eng.*, vol. 2018, Oct. 2018, Art. no. 5816861.

[22] Z. Du, K. L. Cheong, and P. Y. Li, "Energy management strategy for a power-split hydraulic hybrid vehicle based on Lagrange multiplier and its modifications," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 233, no. 5, pp. 511–523, May 2019.

[23] X. Xing, L. Xie, and H. Meng, "Cooperative energy management optimization based on distributed MPC in grid-connected microgrids community," *Int. J. Electr. Power Energy Syst.*, vol. 107, pp. 186–199, May 2019.

[24] X. Zeng and J. Wang, "A parallel hybrid electric vehicle energy management strategy using stochastic model predictive control with road grade preview," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 6, pp. 2416–2423, Nov. 2015.

[25] C. Xiang, F. Ding, W. Wang, W. He, and Y. Qi, "MPC-based energy management with adaptive Markov-chain prediction for a dual-mode hybrid electric vehicle," *Sci. China Technological Sci.*, vol. 60, no. 5, pp. 737–748, May 2017.

[26] R. Bellman, "The theory of dynamic programming," *Bull. Amer. Math. Soc.*, vol. 60, no. 6, pp. 503–516, 1954.

[27] Z. Chenghui, S. Qingsheng, and C. Naxin, "Particle swarm optimization for energy management fuzzy controller design in dual-source electric vehicle," in *Proc. IEEE Power Electron. Spec. Conf.*, Jun. 2007, pp. 1405–1410.

[28] N. Kim, S. Cha, and H. Peng, "Optimal control of hybrid electric vehicles based on Pontryagin's minimum principle," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 5, pp. 1279–1287, Sep. 2011.

[29] G. Paganelli, "General supervisory control policy for the energy optimization of charge-sustaining hybrid electric vehicles," *JSAE Rev.*, vol. 22, no. 4, pp. 511–518, Oct. 2001.

[30] Y. Zou, T. Liu, D. Liu, and F. Sun, "Reinforcement learning-based real-time energy management for a hybrid tracked vehicle," *Appl. Energy*, vol. 171, pp. 372–382, Jun. 2016.

[31] T. Liu, Y. Zou, D. Liu, and F. Sun, "Reinforcement learning–based energy management strategy for a hybrid electric tracked vehicle," *Energies*, vol. 8, no. 7, pp. 7243–7260, Jul. 2015.

[32] T. Liu, Y. Zou, D. Liu, and F. Sun, "Reinforcement learning of adaptive energy management with transition probability for a hybrid electric tracked vehicle," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7837–7846, Dec. 2015.

[33] R. Lian, J. Peng, Y. Wu, H. Tan, and H. Zhang, "Rule-interposing deep reinforcement learning based energy management strategy for power-split hybrid electric vehicle," *Energy*, vol. 197, Apr. 2020, Art. no. 117297.

[34] T. Liu, X. Tang, X. Hu, W. Tan, and J. Zhang, "Human-like energy management based on deep reinforcement learning and historical driving experiences," 2020, *arXiv:2007.10126*. [Online]. Available: https://arxiv.org/abs/2007.10126

[35] 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*, vol. 255, Dec. 2019, Art. no. 113762.

[36] G. Du, Y. Zou, X. Zhang, Z. Kong, J. Wu, and D. He, "Intelligent energy management for hybrid electric tracked vehicles using online reinforcement learning," *Appl. Energy*, vol. 251, Oct. 2019, Art. no. 113388.

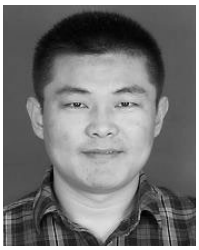
[37] H. Tan, H. Zhang, J. Peng, Z. Jiang, and Y. Wu, "Energy management of hybrid electric bus based on deep reinforcement learning in continuous state and action space," *Energy Convers. Manage.*, vol. 195, pp. 548–560, Sep. 2019.

ZHIKAI MA received the M.S. degree from the Hebei University of Technology, China, in 2013. He is currently a Lecturer with Hebei Agricultural University. His current research interests include vehicle dynamics controller, modeling and control for electrified vehicle, energy management strategy, and tractor navigation systems.

QIAN HUO received the M.S. degree from the Beijing Institute of Technology, in 2013. She is currently a Lecturer with Hebei Agricultural University. Her current research interests include automobile vibration and noise control.

TAO ZHANG received the M.S. degree in mechanical engineering from the Beijing University of Technology, China, in 2015, and the Ph.D. degree in mechanical engineering from the Beijing Institute of Technology, in 2020. He is currently a Research Assistant with the China North Vehicle Research Institute. His current research interests include hardware design of vehicle controller, vehicle dynamics control, advanced driver assistance systems, and reinforcement learning.

JIANJUN HAO received the M.S. degree from Hebei Agricultural University, China, in 2001, and the Ph.D. degree from the University of Science and Technology Beijing, China, in 2010. He is currently a Professor with Hebei Agricultural University, and the Dean of the Department of Agricultural Mechanization. His current research interests include surface engineering and intelligent control machinery (agricultural machinery) equipment and automatic control.



WEI WANG received the M.S. degree from the University of Jinan, China, in 2012. He is currently a Lecturer with Hebei Agricultural University. His current research interests include intelligent agricultural machinery and automatic processing equipment for seafood.