World Electric Vehicle Journal

Article

Soft Actor-Critic Algorithm-Based Energy Management Strategy for Plug-In Hybrid Electric Vehicle

Tao Li, Wei Cui and Naxin Cui *

School of Control Science and Engineering, Shandong University, Jinan 250061, China
* Correspondence: cuinx@sdu.edu.cn

Abstract: Plug-in hybrid electric vehicles (PHEVs) are equipped with more than one power source, providing additional degrees of freedom to meet the driver's power demand. Therefore, the reasonable allocation of the power demand of each power source by the energy management strategy (EMS) to keep each power source operating in the efficiency zone is essential for improving fuel economy. This paper proposes a novel model-free EMS based on the soft actor-critic (SAC) algorithm with automatic entropy tuning to balance the optimization of energy efficiency with the adaptability of driving cycles. The maximum entropy framework is introduced into deep reinforcement learning-based energy management to improve the performance of exploring the internal combustion engine (ICE) as well as the electric motor (EM) efficiency interval. Specifically, the automatic entropy adjustment framework improves the adaptability to driving cycles. In addition, the simulation is verified by the data collected from the real vehicle. The results show that the introduction of automatic entropy adjustment can effectively improve vehicle equivalent fuel economy. Compared with traditional EMS, the proposed EMS can save energy by 4.37%. Moreover, it is able to adapt to different driving cycles and can keep the state of charge to the reference value.

Keywords: hybrid electric vehicle; energy management strategy; deep reinforcement learning; SAC algorithm; automating entropy adjustment

Citation: Li, T.; Cui, W.; Cu, N. Soft Actor-Critic Algorithm-Based Energy Management Strategy for Plug-In Hybrid Electric Vehicle. *World Electr. Veh. J.* **2022**, *13*, 193. https://doi.org/10.3390/wevj13100193

Academic Editor: Vladimir Katic

Received: 6 September 2022
Accepted: 12 October 2022
Published: 18 October 2022

1. Introduction

Nowadays, the problems of environmental pollution and energy depletion are becoming more and more serious [1,2]. Unfortunately, traditional vehicles as the primary form of transportation not only emit large amounts of exhaust fumes, but also consume large amounts of petroleum resources [3]. Electric vehicles (EVs) are considered to be able to solve the problems of fuel consumption and emissions. However, there are several issues that prevent their actual diffusion, among which the most relevant is the limited driving range [4]. Meanwhile, the industry is developing PHEVs as an intermediate solution [5]. Compared with traditional vehicles, PHEVs are typically equipped with two power sources: the ICE and the EM that can be used as both a generator and EM [6]. Therefore, the EMS attempts to navigate energy between several energy sources considering one or more objectives while meeting the power needs of drivers [7,8].

1.1. Literature Review

Recently, various PHEV EMSs have been proposed. They can be broadly divided into three categories: rule-based EMSs [9], optimization-based EMSs [10] and machine learning-based EMSs [11].

The rule-based EMSs determine the operating state of the vehicle powertrain through pre-established rules. Due to its simplicity, rule-based EMSs are easily applied in practice [12,13]. However, rule-based EMSs are formulated empirically so that they cannot efficiently cope with all kinds of driving cycles, therefore cannot always ensure efficient control [14,15]. To further enhance fuel economy, the expert knowledge and optimization

algorithms are applied to the rule-based EMSs, such as the genetic algorithm (GA) and particle swarm optimization (PSO) algorithm [16]. Zhou et al. present a multi-objective optimization method that the membership functions of the integrated strategy are tuned by elitist nondominant GA [17]. Natella et al. implement an optimization problem for the off-line section of the velocity thresholds and the corresponding power splitting between the actuators [18].

Optimization-based EMSs are mainly divided into global optimization-based EMSs and real-time optimization-based EMSs [19]. The dynamic programming (DP) algorithm and pontryagin minimum principle (PMP) strategy are the widely used global optimization algorithms in PHEV EMSs [20]. The DP algorithm requires perfect knowledge over the entire optimization horizon to obtain or approximate the global optimum and requires much computing time, so it is not suitable for real-time control of PHEVs [21]. The PMP algorithm-based strategy is also a widely studied global optimization strategy, which solves the optimization problem by minimizing the Hamiltonian function [22]. However, entire driving information is still needed to achieve improvements in fuel economy [23]. The equivalent consumption minimization strategy (ECMS) and model predictive control (MPC) strategy are representative of vehicle EMSs based on real-time optimization technologies [24]. ECMS is the most popular real-time EMS, which gets the optimum fuel consumption through the rational allocation of the power of the EM and ICE according to the torque or power requirements of the vehicle in the current moment, and balances fuel consumption and the battery state of charge (SOC) by the equivalent factor [25]. The equivalent factor plays a very important role in the ECMS, and it directly affects the effectiveness of the optimization strategy. Li et al. [26] studied the equivalent factor boundary of the ECMS for PHEVs. However, it is still a challenge to precisely determine the equivalence factor. This limits the adaptive adjustment capabilities of the ECMS. Feng et al. [27] proposed an adaptive ECMS with energy demand prediction to improve the adaptive capability. The MPC strategy optimizes the fuel economy of PHEV on a moving finite horizon [28]. However, it is not possible to update the result of MPC frequently when computing on look-ahead horizons because the computation time increases with the horizon length. Therefore, Uebel et al. [29] proposed a two-level MPC approach to overcome the computational burden. Mariani et al. [30] propose the design of an MPC strategy for maximizing regenerative braking in a real vehicle that has been hybridized by means of a kit.

The above approaches lack self-learning capabilities, which are dependent on the accuracy of dynamic models, and can fail if such models run under various environments. Therefore, machine learning-based EMSs have received extensive attention in recent years, with reinforcement learning (RL) and deep reinforcement learning (DRL) being the most widely studied. RL's main idea is to train a fully autonomous agent by interacting directly with its potential environment [31]. It is different from supervised and unsupervised machine learning which need static data during the training process. In [32], model-free predictive EMS with multi-step learning capabilities was proposed, in which the Q learning-based EMSs, including the sum-to-terminal strategy, average-to-neighbor strategy and recurrent-to-terminal strategy were investigated. Compared to the conventional strategy, a real-time fast Q-learning-based reinforcement learning EMS was investigated to improve the fuel economy and reduce the computational efficiency [33]. Chen et al. [34] combine the Markov decision process (MDP) and Q-learning algorithms to design power flow EMS, which achieves fuel economy improvement under different driving cycles. The Q-learning algorithm needs to build a Q table whose size is determined by the dimension of the states and the action. When applied to complex HEV, Q-learning needs discrete continuous states and actions so that the curse of dimensionality, as well as discrete error, limits its application [35]. In order to solve the problem with high-dimensional and continuous state/action, the deep neural network is used in the RL approaches [36]. Wu et al. [37] proposed a continuous EMS based on deep Q learning (DQL), which approximates the action value function through a deep neural network. By using double deep Q-learning

(DDQL), Han et al. [38] addressed the problem that traditional DQL tended to fall into the trap of over-optimistic estimation of Q value during training. DDQL-based intelligent decision algorithm was proposed in [39], which achieves energy savings similar to offline optimization. Both DQL and DDQL sample data from empirical replay, then calculate the target Q value. All the samples in the replay buffer have the same probability of being sampled. This leads to a reduction in the learning efficiency of the RL agent. In order to improve the learning efficiency, Runna et al. [40] applied prioritized replay to the DQL-based EMS. DQL is a value function-based approach, but it has trouble with solving large action spaces, especially continuous spaces [41,42]. Yue et al. [43] introduce temporal-difference (TD) learning based on Q learning and achieve the management of HEV supercapacitors and power batteries through a model-free online strategy. A DRL EMS based on the TD algorithm was proposed and combined with road information to achieve self-learning power flow distribution of the hybrid vehicle power system in [44]. Lian et al. [45] used the deep deterministic policy gradient (DDPG) algorithm to solve the problem of multi-objective energy management with a large control variable space. However, the above deep reinforcement learning algorithms are often highly sensitive to hyperparameters which directly affect the convergence performance, and even the RL agent is difficult to adapt to different conditions [46]. Recent work has shown that the SAC deep reinforcement learning algorithm with maximum entropy learning can solve the above problems. The standard RL improves their performance only by maximizing the cumulative reward. Because the maximum entropy learning is to achieve entropy maximization, the SAC improves its performance by maximizing the weighted sum of expected cumulative reward and entropy. The maximum entropy learning frame is introduced to improve the ability of the action exploration and robustness. Furthermore, in order to promote the algorithm performance, auto-entropy tuning (AET) is proposed [47].

1.2. Contribution

Considering the good convergence and insensitivity to hyperparameters of the SAC algorithm with auto-entropy tuning (SAC−AET), the SAC−AET-based EMS is proposed to improve the control effect of the traditional RL algorithm-based EMS. The ability of the maximum entropy learning framework used in EMS to explore the vehicle efficiency space is investigated. An automatic entropy adjustment framework is introduced to enhance the EMS's adaptability to driving cycles. In addition, the driving cycles were collected from real vehicles. The fuel economy of the proposed SAC-based EMS is compared with that of the DDPG strategy, traditional SAC strategy and ECMS. In addition, the adaptability of the proposed strategy to different driving conditions is verified by combining driving cycles.

The remainder of the paper is organized as follows. The PHEV powertrain model is described in Section 2. The SAC algorithm-based EMS is described in detail in Section 3. In Section 4, simulations are designed to evaluate the performance of the proposed EMS, and simulation results are evaluated. The final section is the conclusion of the paper.

2. PHEV Powertrain Model

In this paper, the studied vehicle is a plug-in hybrid electric bus with a parallel configuration which is shown in Figure 1. The vehicle powertrain consists mainly of ICE, EM, battery pack and dual-clutch. The two clutches separate the EM and ICE so that the vehicle powertrain can operate in different modes, such as only the ICE operation, only the EM operation, and ICE and EM operation together. The relevant parameters of the vehicle are given in Table 1.

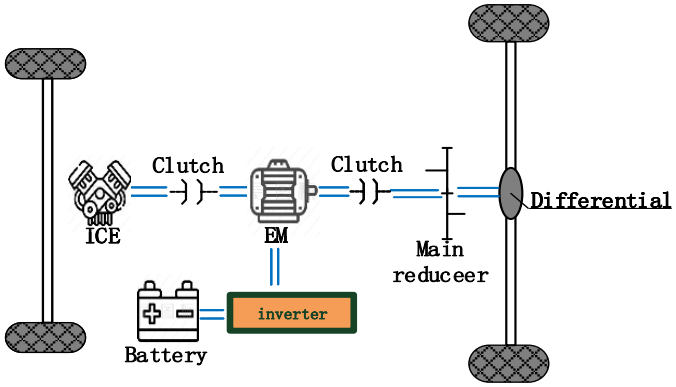


Figure 1. The PHEV powertrain framework.

Table 1. Vehicle parameters.

Symbol	Parameters	Value
Vehicle	Curb weight	10,500 kg
	Rolling resistance coefficient	0.015
	Air resistance coefficient	0.65
	Frontal area	6.75 m^2
EM	Maximum power	135 kW
	Maximum torque	1000 Nm
	Maximum speed	3500 rpm
ICE	Maximum power	159 kW
	Maximum torque	904 Nm
	Maximum speed	2300 rpm
Battery	Voltage	525 V
	capacity	96 Ah

The ICE and EM are modeled by efficiency maps collected from an experimental platform. The efficiency maps of the ICE and the electric EM depict the relationship between speed, torque and efficiency, as shown in Figures 2 and 3. The red curve of the ICE efficiency map in Figure 2 corresponds to the optimal torque at maximum efficiency. The Figure 3 shows that the EM maintains a relatively high efficiency whatever the operating conditions.

The balance equation of the vehicle longitudinal force is given as follows:

$$F_t = F_j + F_i + F_w + F_f \tag{1}$$

$$\begin{cases} F_j = \delta m a_{cc} \\ F_i = mg\sin(\theta) \\ F_w = \frac{1}{2}\rho C_d A_f v^2 \\ F_f = mgf\cos(\theta) \end{cases} \tag{2}$$

where F_t is traction force, F_j is the acceleration force, F_i is the road grade force, F_w is aerodynamic resistance force, F_f is rolling resistance force, m is the gross weight, g is the gravity constant, θ is the road slope, C_d is the aerodynamic coefficient, A_f is the vehicle frontal area, f is the rolling resistance coefficient, v is the vehicle velocity. δ is the rotational mass coefficient. a_{cc} is the vehicle acceleration, and ρ is the air density.

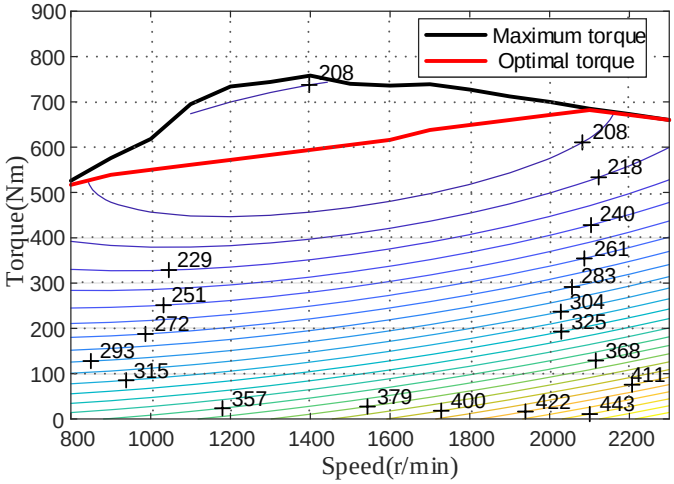


Figure 2. ICE efficiency map.

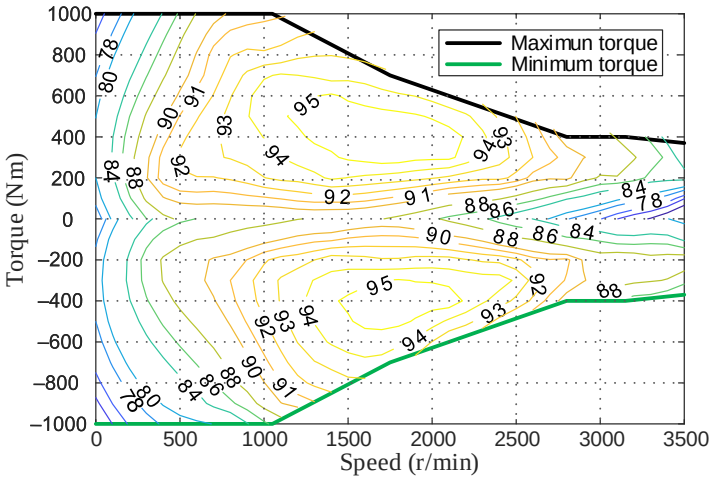


Figure 3. EM efficiency map.

The vehicle torque demand is given in Equation (3).

$$T_{req} = F_t \times r_w \div (i_g \times i_o \times \eta) \tag{3}$$

where r_w is the wheel radius, i_g is transmission ratio, i_o is the final drive ratio, η is the drive system efficiency. The powertrain structure of the studied PHEV is configured with a six-speed automated manual transmission (AMT) gearbox, and the AMT gearbox parameters are shown in Table 2. According to the vehicle velocity, the gear information G and the transmission ratio i_g are obtained by checking the table.

In this paper, the ICE torque demand T_e is determined by the SAC strategy. The EM torque demand T_m is calculated by the Equation (4).

$$T_m = T_{req} - T_e \tag{4}$$

where the T_{req} is the demand torque.

The battery is a significant part of the vehicle. It not only powers the vehicle but also stores the recovered energy when the vehicle is decelerating. There are many important parameters of the battery, including battery current, open-circuit voltage, internal resistance, SOC, and so on [48]. In this research, SOC is more important since it describes the remaining battery energy. The Li-Ion battery is modeled by an equivalent internal resistance model that can calculate the SOC as follows:

$$\begin{cases} P_b(t) = P_m \eta_m^n \\ I_b(t) = \frac{V_{oc}(t) - \sqrt{V_{oc}^2(t) - 4R_0 P_b(t)}}{2R_0} \\ SOC(t) = \frac{Q_{in} - \int_0^t I(t)dt}{Q_{max}} \end{cases} \quad (5)$$

$$n = \begin{cases} -1, if P_m > 0 \\ 1, if P_m \leq 0 \end{cases} \quad (6)$$

where P_b is the power of the battery, P_m is the EM power, η_m^k is the charge and discharge efficiency, I_b is the current, V_{oc} is the open-circuit voltage, R_0 is the internal resistance of the battery, Q_{in} is the initial capacity of battery, Q_{max} is the maximum battery capacity, n is a variable that varies according to the EM power.

Table 2. The AMT gearbox parameters.

Upshifting Velocity (km/h)	0–10	10–20	20–32	32–50	50–66	66–95
Downshifting Velocity (km/h)	0–7	7–15	15–28	28–45	45–58	-
Gear position	1	2	3	4	5	6
Gear ratio	6.39	3.97	2.40	1.48	1	0.73

3. Energy Management Strategy Based on SAC Algorithm

3.1. SAC Algorithm

MDP is a sequential decision problem with a fully observable stochastic environment. The goal of RL is to find the optimal strategy under MDP to maximize the final cumulative reward r. Each state under the MDP is related not only to the current state s but also to the current action a. Since RL relies on reward and punishment given by the environment to learn, the corresponding RL also includes the reward and punishment value r. Therefore, the RL process can be composed of a quadruplet M = (s, a, s′, r). The basic process of the agent–environment interaction for deep reinforcement learning is shown in Figure 4. The RL agent continuously interacts with the environment until it converges. In general, the process of the agent–environment interaction can be summarized as follows: at each moment of the discrete-time series, $t \in \{0, 1, 2, \cdots\}$, the RL agent samples action a_k from the policy network. After executing the action a_k, the agent gets a new state s_{k+1} from the environment and obtains a reward r based on the action performed.

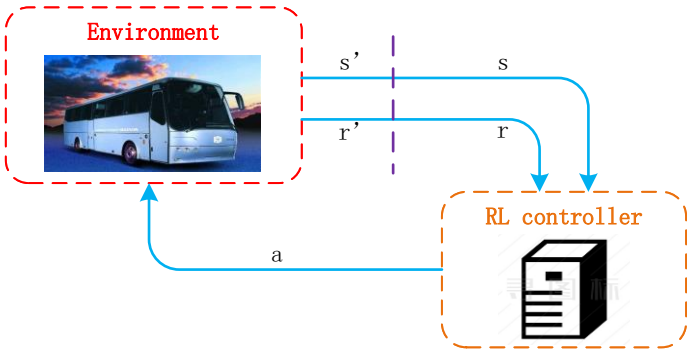


Figure 4. Basic process of agent–environment interaction for deep reinforcement learning.

This deep reinforcement learning algorithm consists of three important parts: an experience replay for storing previous experiences to reduce sampling complexity, maximum entropy for stabilization and exploration, and an actor-critic structure consisting of a policy network and four Q networks built by Multilayer Perceptron (MPL). Through the interplay

of the above three components, the SAC deep reinforcement learning algorithm can be divided into two processes: soft policy iteration and automating entropy adjustment.

3.1.1. Soft Policy Iteration

The aim of standard reinforcement learning is to learn a policy $\pi^*(a_t|s_t)$ to make the desired reward larger in the future. The SAC algorithm is different from standard reinforcement learning which only learns the optimal policy to maximize the reward. In order to explore all possible optimal paths, entropy term $\mathcal{H}$ is introduced:

$$\pi^* = argmax_\pi \sum_t \mathbb{E}_{\rho_\pi}[r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t))] \tag{7}$$

where α is the equilibrium parameter that is used to regulate the tradeoff between reward and entropy. The strategy can be made more randomized by increasing α, that is, the probability of each action is as uniform as possible, instead of concentrating on one action. $\mathcal{H}$ is the entropy value of strategy π, ρ_π is the probability of state-action tuple under the current polity, r is the reward, s is the state, a is the action.

To ensure that the sum of reward (and entropy) is finite over the entire time−step sequence, $k \in \{0, 1, 2, 3, \cdots\}$, discount factor γ, $0 \leq \gamma \leq 1$, is introduced. The policy with a discount factor can be defined as:

$$\pi^* = argmax_\pi \sum \mathbb{E}_{\rho_\pi}\left[\sum_{l=k}^{\infty} \gamma^{l-k} \mathbb{E}[r(s_k, a_k) + \alpha \mathcal{H}(\pi(\cdot|s_k))]\right] \tag{8}$$

l represents the time series from the current moment to the last moment.

Soft policy iteration is a general method to learn the optimal maximum entropy policy [49]. The actor−critic framework consists of two kinds of networks: the policy network and the Q network. The policy iteration process alternates between soft policy evaluation (Q network) and soft policy improvement (policy network) to maximize the reward during the iteration. The process of soft policy evaluation which computes the soft Q value $Q(s_k, a_k)$ can be achieved by repeatedly using the modified Bellman backup operator $\mathcal{T}^\pi$ for the fixed policy as follows:

$$\mathcal{T}^\pi Q(s_k, a_k) \triangleq r(s_k, a_k) + \gamma \mathbb{E}_\rho[V(s_{k+1})] \tag{9}$$

where

$$V(s_k) = \mathbb{E}_\pi[Q(s_k, a_k) - \alpha \log \pi(a_k|s_k)] \tag{10}$$

is a soft value function.

In order to deal with complex and multimodal behaviors, the SAC algorithm introduces the general energy-based policy [50]. In the policy improvement process, the new policy is not tractable in practice, so the Kullback–Leibler divergence is introduced, which can limit policy to a certain set of policies Π. The policy $\pi \in \Pi$ is updated as follows:

$$\pi_k^* = \arg\min_{\pi' \in \Pi} D_{KL}\left(\pi'(\cdot|s_k) \middle\| \frac{\exp(\frac{1}{\alpha} Q^{\pi_{old}}(s_k, \cdot))}{Z^{\pi_{old}}(s_k)}\right) \tag{11}$$

$Z^{\pi_{old}}(s_k)$ does not contribute to π_k^*, so it can be ignored to optimize the π_k^*, and it has been proven that $Q^{\pi_k^*}(s_k, a_k) \geq Q^{\pi_{old}}(s_k, a_k)$. More details can be seen in [50].

3.1.2. Automatic Entropy Adjustment

The parameter α in Equation (8) is a significant parameter that directly influences the optimization objective, which should be changed for different driving cycles by experienced engineers. Therefore, this section formulates a different objective from maximum entropy reinforcement learning to achieve the automatic entropy adjustment, where the entropy is

considered as a constraint. We want to maximize the desired reward under the constraint of a minimum expected entropy as follows:

$$\begin{aligned} &\max_{\pi_{0:T}} \mathbb{E}_{\rho_\pi}\left[\sum_{k=0}^{T} r(s_k, a_k)\right], \\ &s.t. \mathbb{E}_{\rho_\pi}[-\log \pi(\cdot|s_k)] \geq \mathcal{H} \end{aligned} \tag{12}$$

where $\mathcal{H}$ is an expected minimum entropy threshold.

The optimal policy π_k^* is given directly as follows, and its derivation is described in detail in [51].

$$\begin{aligned} \pi_k^* &= \arg\max_{\pi_k} \mathbb{E}[Q_k^*(s_k, a_k) - \alpha_k \log \pi(a_k|s_k)] \\ &= \arg\min_{\pi_k} D_{KL}\left(\pi_k \middle\| \frac{1}{Z(s_k)} \exp\left(\frac{1}{\alpha_k} Q_k^*(s_k, a_k)\right)\right) \end{aligned} \tag{13}$$

the Equation (13) is exactly the soft policy improvement step, which has the additional equilibrium parameter α_k. The dual variable α_k^* is a function of the optimal strategy at k. The dual variable α_k^* can be solved as

$$\alpha_k^* = \arg\min_{\alpha_k} \mathbb{E}_{\tilde{\gtrsim}}[-\alpha_k \log \pi_k^*(a_k|s_k; \alpha_k) - \alpha_k \mathcal{H}] \tag{14}$$

3.2. Practical Algorithm

In the last section, the problem has already been solved in the tabular case. In order to extend the above method to the continuous domain, the function approximator is used to represent the Q-function Q_π and policy π_ϕ. A number of effective value function approximation methods have been proposed [52]. In this paper, a neural network will be used to approximate the value function, which scores the effect produced by the current torque distribution. We use two artificial neural networks to approximate the Q-function with parameter θ_i, $i \in \{1, 2\}$, which aim at reducing the overestimation of the Q value. The Q value is chosen from the smaller of the double Q-function values. The target network with the parameters $(\bar{\theta}_1, \bar{\theta}_2)$ is introduced to prevent overfitting during the training of the two Q networks. In each iteration, the Q networks are trained by the gradient descent method and the parameters $(\bar{\theta}_1, \bar{\theta}_2)$ of the target networks are updated by an exponentially moving average of the value network weights as

$$J_Q(\theta_i) = \mathbb{E}_{\mathcal{D}}\left[\frac{1}{2}(Q_{\theta_i}(s_k, a_k) - (r(s_k, a_k) + \gamma V_{\bar{\theta}_1, \bar{\theta}_2}(s_{k+1})))^2\right] \tag{15}$$

$$\bar{\theta}_i(k+1) \leftarrow m\theta_i(k+1) + (1-m)\bar{\theta}_i(k), \;\; i \in \{1, 2\} \tag{16}$$

where $0 < m < 1$, is a smoothing factor. $\mathcal{D}$ is the replay buffer, from which minibatch are obtained. A policy network is a Gaussian model with the mean and covariance given by the neural network. The Gaussian policy is trained by minimizing the loss $J_\pi(\phi)$:

$$J_\pi(\phi) = \mathbb{E}_{\mathcal{D}}\left[\alpha \log \pi_\phi(a_k|s_k) - \min_{i \in \{1,2\}} Q_{\theta_i}(s_k, a_k)\right] \tag{17}$$

The learning of α can be obtained by approximating dual gradient descent to minimize the dual objective $J(\alpha)$:

$$J(\alpha) = \mathbb{E}_{\mathcal{D}}\left[-\alpha \log \pi_\phi(a_k|s_k) - \alpha \mathcal{H}\right] \tag{18}$$

The Equations (15)–(18) form the core of the SAC−AET. In the following work, the networks will be optimized by Equations (15)–(18). The process of the SAC algorithm is shown in Algorithm 1.

Algorithm 1 Soft actor-critic DRL with automating entropy adjustment algorithm.

initialization:
Q networks with weights $\theta_i, i = \{1, 2\}$,
policy network with weights ϕ,
target network with weights $\overline{\theta}_{1,2}$,
equilibrium parameter α,
replay buffer $\mathcal{D}$;
FOR EACH EPISODE DO
get initial state
for each environment step do
choose action $a_k \sim \pi(a_k|s_k)$
take action a_k, observe s_{k+1} and reward r
update replay buffer $\mathcal{D} \leftarrow \mathcal{D} \cup \{s_k, a_k, r, s_{k+1}\}$
end
for each gradient step do
sample a minibatch from $\mathcal{D}$
update the Q network parameters :
$$Q \leftarrow Q - \lambda \widehat{\nabla}_Q J(Q)$$
update the policy network parameters:
$$\pi \leftarrow \pi - \lambda \widehat{\nabla}_\pi J(\pi)$$
update the equilibrium parameter:
$$\alpha \leftarrow \alpha - \lambda \widehat{\nabla}_\alpha J(\alpha)$$
update the target network parameters
end
END

3.3. Design of SAC Algorithm-Based EMS

The three important components (state, action, reward) are defined as follows:

3.3.1. State

The gear is an important driving information factor for PHEV. Therefore, the state variables are given as follows:

$$s(t) = \{v(t), a_{cc}(t), SOC(t), T_{req}(t), G(t)\} \tag{19}$$

where v is the vehicle velocity, a_{cc} is the acceleration, G,$G \in [1, 2, 3, 4, 5, 6]$, is the current gear of the vehicle, s is the current state, the future state is denoted by $s^{'}$.

3.3.2. Action

The action $a(t)$, $0 \leq a \leq 1$, is used as the control signal for the ICE at moment t. When the action $a(t)$ is obtained by the SAC algorithm, the demand torque of the ICE and EM can be calculated:

$$\begin{cases} T_e(t) = T_{e\_max}(t)a(t) \\ T_m(t) = T_{req}(t) - T_e(t) \end{cases} \tag{20}$$

where T_{e_max} denotes the maximum value of the ICE torque. In order to ensure the safety and stability of the vehicle, the following constraints should be ensured, which are mainly the physical limitations for the vehicle controller and the maximum operating parameter limitations of the ICE and EM:

$$\begin{cases} SOC_{min} < SOC(t) < SOC_{max} \\ T_{e\_min} < T_e(t) < T_{e\_max} \\ T_{m\_min} < T_m(t) < T_{m\_max} \\ n_{e\_min} < n_e(t) < n_{e\_max} \\ n_{m\_min} < n_m(t) < n_{m\_max} \\ T_{req\_min} < T_{req}(t) < T_{req\_max} \\ I_{batt\_min} < I_{batt}(t) < I_{batt\_max} \end{cases} \tag{21}$$

where n_e and n_m represent the ICE speed and the EM speed, respectively. max and min denote the maximum and minimum values of the corresponding variable.

3.3.3. Reward

The objectives of the EMS are to achieve fuel economy and keep the battery *SOC* within a certain range. The reward is defined as a function of vehicle fuel consumption and battery *SOC*. The main purpose of the RL agent is to achieve the maximum reward. However, we want to decrease the instantaneous fuel consumption and keep the SOC to SOC_{ref}. The introduction of r_{init} can resolve the conflict. The reward is shown as:

$$r(t) = \begin{cases} r_{init} - cost(t), & SOC(t) \geq SOC_{ref} \\ r_{init} - \left(\beta\left(SOC_{ref} - SOC(t)\right)^2 + cost(t)\right), & else \end{cases} \tag{22}$$

$$cost(t) = b_e(t)\frac{T_e(t)n_e(t)}{9550} \tag{23}$$

where the $r_{init} = 1$ is a constant. β is a proportional factor used to balance the *SOC* and fuel consumption efficiency. In the study of this paper, β is taken as 15,000. SOC_{ref} is reference *SOC* value. *cost* is the instantaneous fuel consumption, and *be* is the effective specific fuel consumption.

The proposed EMS for PHEVs based on the SAC algorithm is shown in Figure 5. At each time step, the agent interacts with the environment to obtain samples $(s, a, r, s^{'})$ that are stored in the replay buffer whose capacity is set to hold one million data items. During the learning process, a minibatch with 256 data items is randomly selected from the replay buffer which is used to solve the problem of data correlation and non-stationary distribution. In addition, the algorithm is able to learn from past experiences to increase data utilization and learning efficiency. The state vector s combined with the action a is used as input to Q networks. The Q networks output Q values, which are used to evaluate the value of taking input action under the input state vector. The next state vector $s^{'}$ is used as input to the policy network to calculate the next action. The inputs of the target Q networks are $s^{'}$ combined with the next action. Then, the Q value at the next moment can be obtained, which evaluates the value of taking the next action under the next state. Then, Q networks and policy networks are trained by gradient descent algorithm according to Equations (15) and (17), respectively. The α is learned by dual gradient descent according to Equation (18). Policy network, Q networks and target Q networks have only one hidden layer with 300 neurons, respectively. More details about the SAC algorithm have been shown in Table 3.

Table 3. Parameters of SAC−AET.

Parameters	Value
discount factor	0.99
target smoothing coefficient	0.005
learning rate	0.0003
batch size	256
hidden size	300
replay size	1,000,000
entropy target	−3

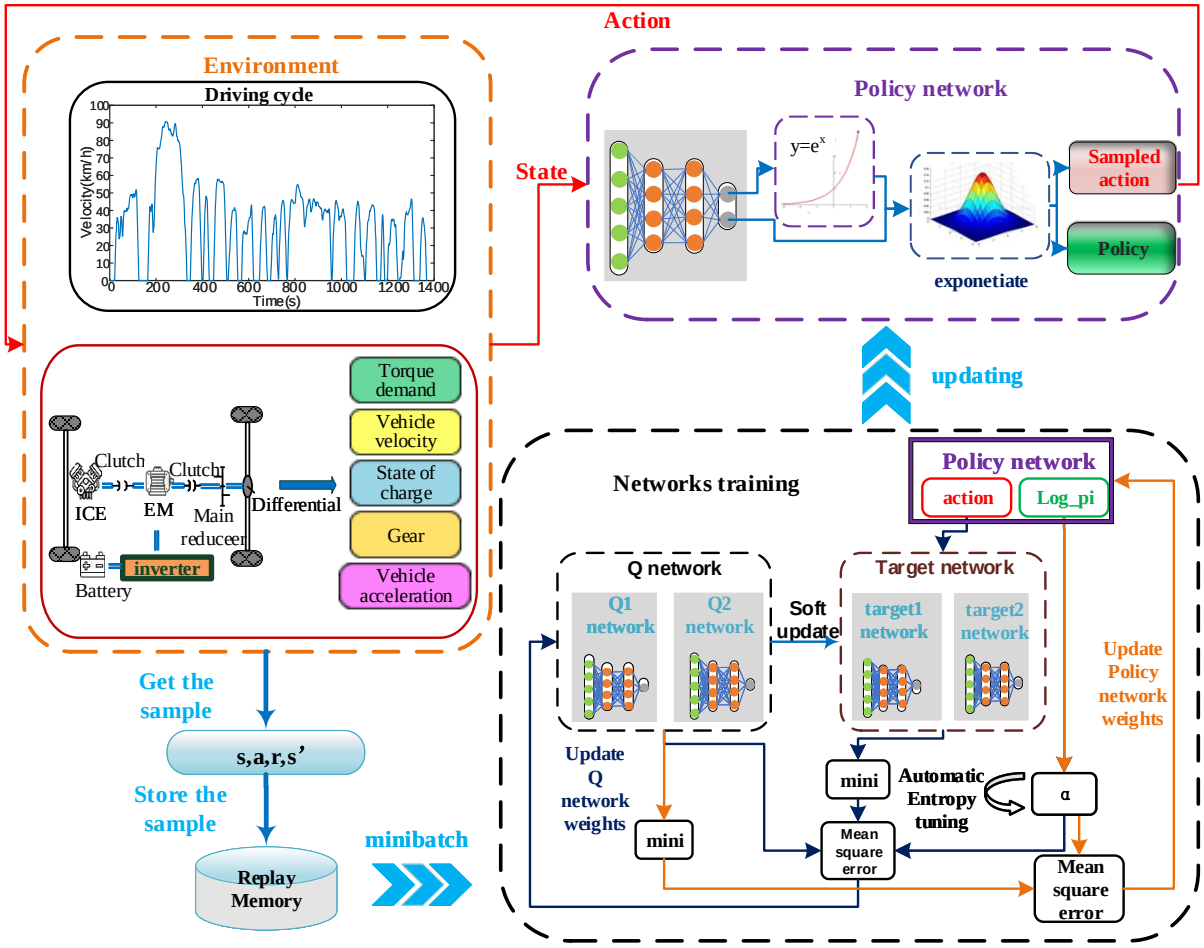


Figure 5. The proposed EMS framework.

4. Simulation and Discussion

The proposed SAC-based EMS is validated in this section. The convergence and performance of the proposed strategy are verified by a standard driving cycle. In addition, the performance of SAC−AET-based EMS is compared with that of the SAC fixed equilibrium parameter, DDPG strategy and ECMS. The adaptability of the proposed strategy is verified by the real driving cycle. The real driving cycle is obtained from the bus remote intelligent monitoring platform based on a mass of data, which is repeatedly collected from vehicle CAN as shown in Figure 6.

Figure 6. Real-world driving cycle collection.

4.1. The Performance of SAC Algorithm-Based EMS for UDDS

The standard driving cycle Urban Dynamometer Driving schedule (UDDS) in Figure 7 is used for the learning process of the proposed method. We use two UDDS driving cycles to simulate a long-distance trip of PHEV. In this section, the performance of the SAC−AET for UDDS is explored. The reference SOCs are set to 35%, 40%, 45%, 50%, respectively, to validate the control effects for battery SOC. Figure 8 represents the trajectories of the reference SOC from 35% to 50% under UDDS driving cycle. We can see that the battery SOC curves decrease from initial value 0.7 to reference SOC, which shows that the proposed strategy can achieve the goal to keep the SOC to the various reference SOC. In order to analyze the changes of each part of the EMS in detail, we randomly select one of the four reference SOCs to explain, and here we will show the results when the reference SOC is 45%.

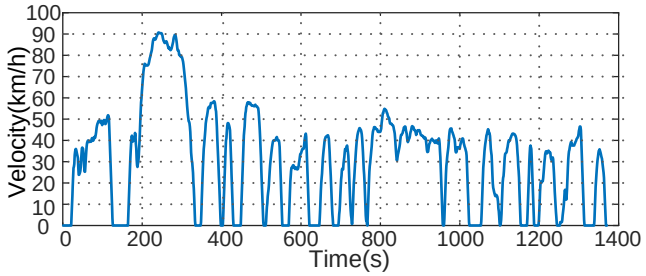


Figure 7. UDDS driving cycle.

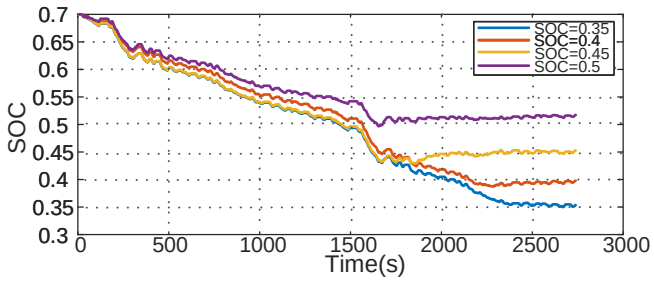


Figure 8. The SOC trajectories of different reference SOC.

Figure 9 shows the rewards of the iterations, where the rewards have a large fluctuation at the beginning of the learning process. In the beginning, the RL agent cannot figure out a better decision, and the exploration strategy is used to randomly explore the action for the current state to obtain the cumulative reward information. As the number of iterations increases, the exploration strategy is gradually weakened, and the selected actions can bring higher rewards according to the current policy. Therefore, the reward stabilizes after about 23 iterations. It can be found that the reward becomes larger than the initial condition and eventually remains within a certain range.

Figure 10 shows the adjustment process of α. The blue line is the loss of equilibrium parameter at each step throughout the training process, which shows that the α loss gradually fluctuates near zero. The α is adjusted and finally stabilized to a certain value, as the red line shows.

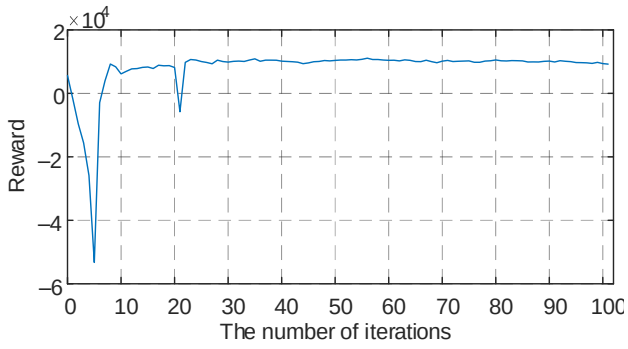


Figure 9. Cumulative rewards of each iteration.

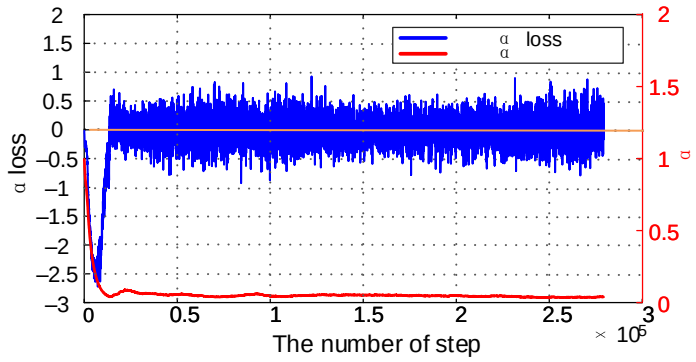


Figure 10. The process of α adjustment.

The EM power and SOC trajectory curve of the vehicle are shown in Figure 11. The EM not only acts as a power source but also charges the battery when the EM power is negative. In the SOC maintenance stage, which is the horizontal part of the yellow line, the probability of the negative power is larger to keep the SOC to reference SOC (0.45).

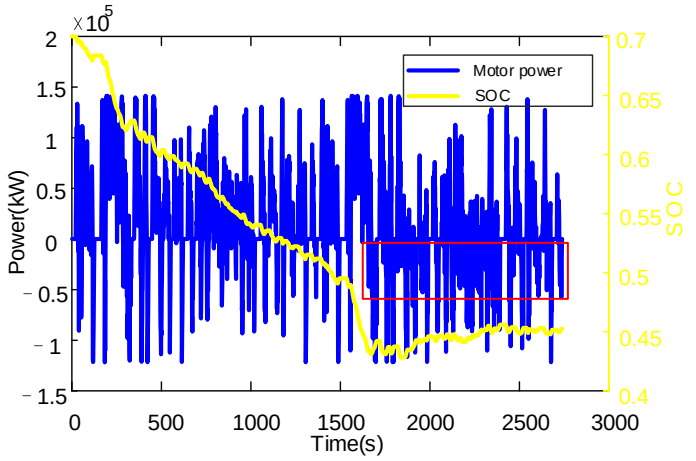


Figure 11. SOC trajectory at reference SOC of 0.45 and EM power.

The strategy is trained for a total of 100 iterations and the change in SOC is observed every 10 iterations, as in Figure 12. It can be seen from the figure that the value of the SOC is the largest in the first iteration. This is because the parameters of the strategy are randomly initialized at the initial time. During the 10th iteration, we can see that the terminal of SOC will drop to 0.45. As the training time continues to increase, the terminal of SOC gets closer to the reference SOC (0.45). This further reflects the stability of the strategy and the ability to learn. The equivalent fuel consumption per 100 km for each iteration is shown in Figure 13. With the increase in training, the fuel consumption per 100 km tends to stabilize.

As we can see from the graph, there are two places with the lowest fuel consumption, but in this case, the SOC of the battery is not controlled by the reference value. We simulated five times under this driving cycle and the average value of equivalent fuel consumption for 100 km is 24.42 L.

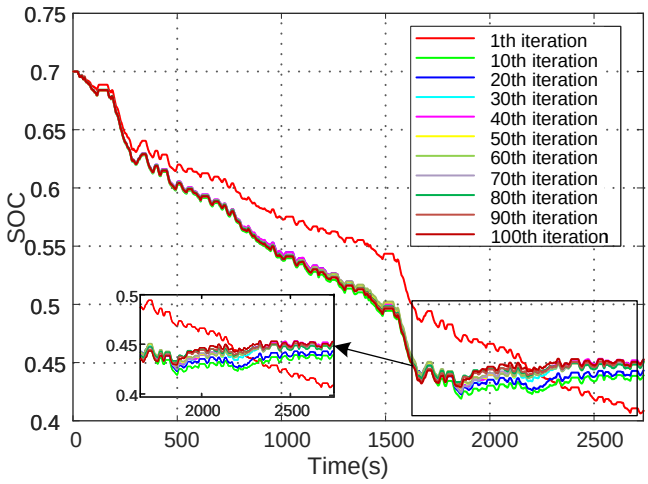


Figure 12. SOC trajectories with different iterations.

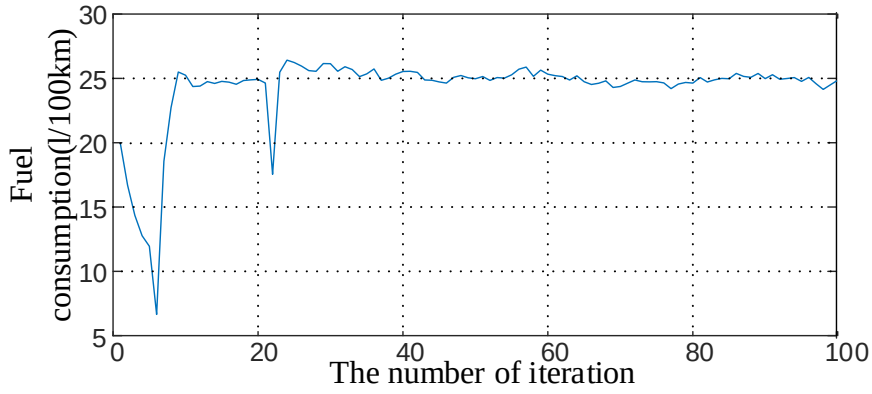
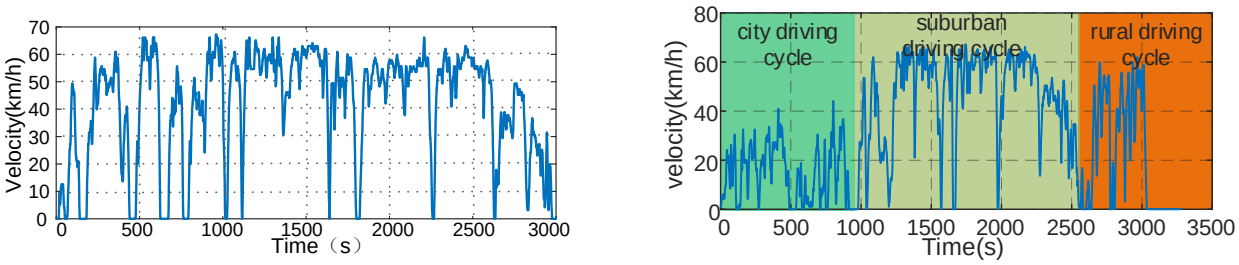


Figure 13. The equivalent fuel consumption variation with different iterations.

4.2. Comparison of Different Strategies

In this experiment, the optimality of SAC−AET-based EMS is compared with that of the SAC algorithm-based strategy with fixed equilibrium parameter, DDPG-based strategy and ECMS. The DDPG strategy is a deep reinforcement learning algorithm-based EMS that is highly sensitive to hyperparameters. In this section, the four strategies are simulated using real driving cycle c1, shown in Figure 14a.



(**a**) (**b**)

Figure 14. Testing driving cycle. (**a**) Real-world driving cycle c1. (**b**) Real-world driving cycle c2.

The four EMSs are simulated using the same initial state, objective function, control variable and constraints. Table 4 shows the fuel consumption of the strategies. In order to

reflect the stability of the proposed strategy, the results of the last five simulations were averaged to obtain the ICE fuel consumption and the equivalent fuel consumption. As we can see from the table, the SAC-AET has a fuel consumption of 9.4283 L/100 km, which is more economical compared to the other three strategies. It is also the smallest in terms of equivalent fuel consumption. The control effects of the DDPG strategy are close to that of the SAC strategy, but the efficiency distribution of ICE working points of the strategies according to Figure 15 shows that the proposed SAC algorithm-based EMS is able to explore a wider action space, which then facilitates the ICE to work in a more favorable operating zone for the strategy. It can be seen from the figure that most ICE operating points are located near a 200 (g/kWh) fuel consumption rate. In addition to the ECMS, the ICE operates in the 100–190 (g/kWh) range some of the time. Therefore, the SAC strategy saves more fuel. The comparison with the traditional SAC algorithm-based strategy is used to demonstrate that the introduction of automatic entropy adjustment not only improves the fuel economy, but also increases the adaptability and self-regulation capability of the strategy.

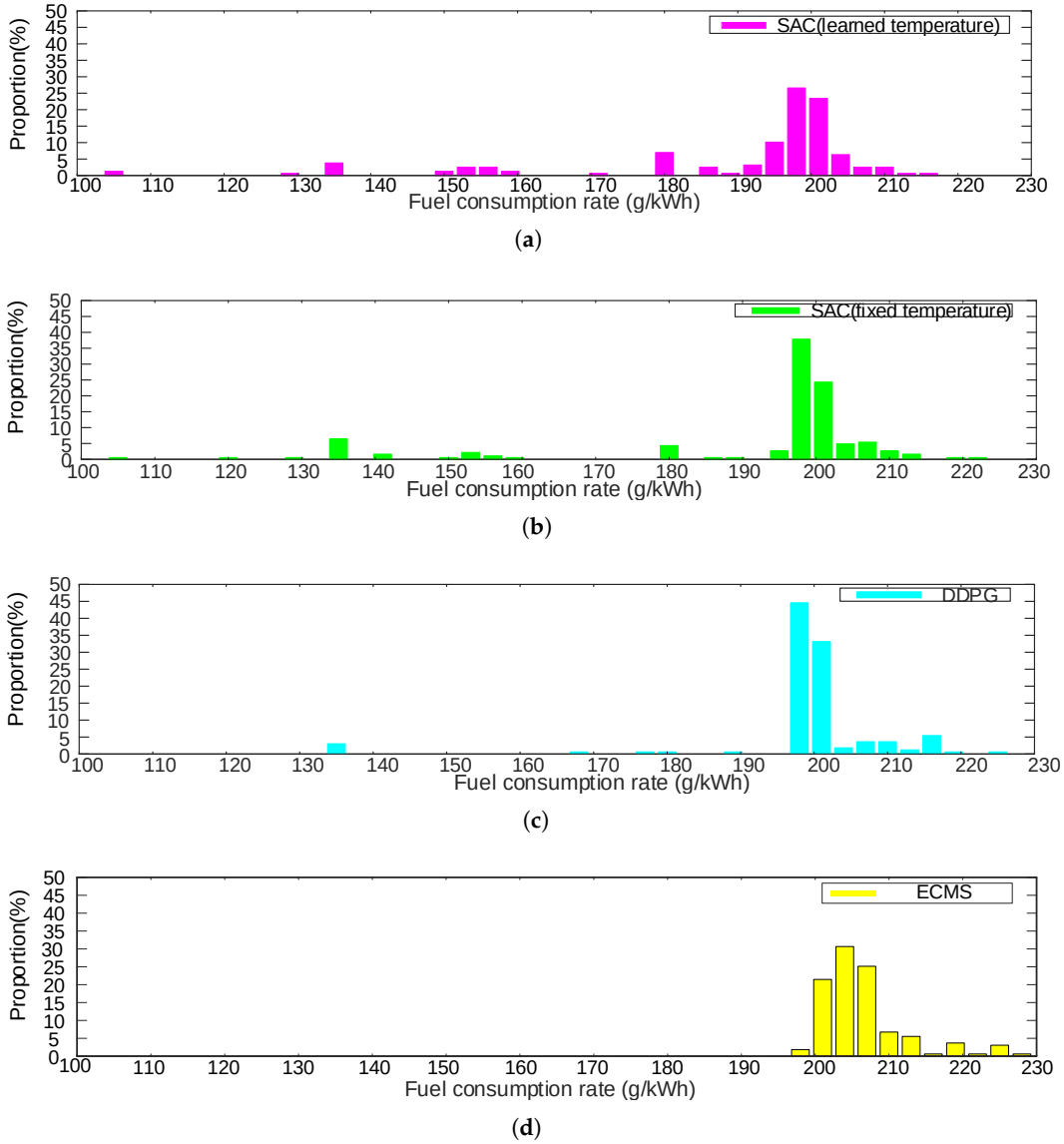


Figure 15. The efficiency distribution of ICE working points of the four strategies. (**a**) SAC-AET strategy. (**b**) SAC (fixed equilibrium parameter) strategy. (**c**) DDPG strategy. (**d**) ECMS strategy.

The SOC trajectories of the strategies are shown in Figure 16. For maximizing the use of electrical energy, we set the reference SOC to 0.3. In the figure, we can see that the SOC trajectories for the SAC-AET strategy and SAC with fixed equilibrium parameter strategy are close before 2400 s. After 2400 s, the SOC of the ECMS is the highest compared with others, and its final SOC is 0.3064. The final SOCs of the SAC-AET and DDPG strategies are 0.2908 and 0.2865, respectively.

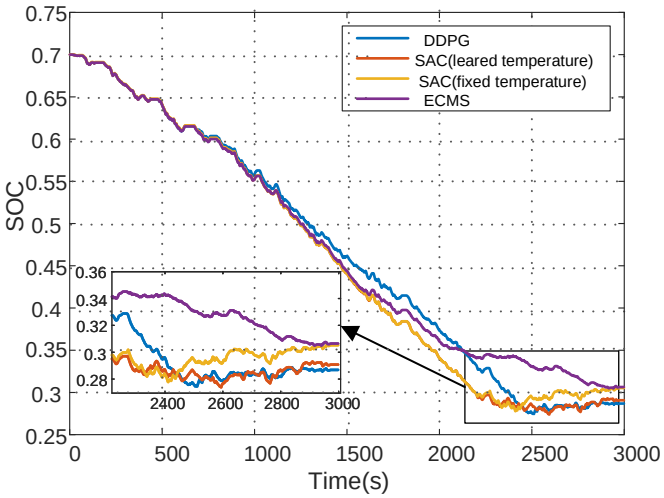


Figure 16. The SOC trajectories of four strategies.

Table 4. The comparison of fuel consumption in three strategies.

Algorithm	ICE Fuel Consumption (l/100 km)	Equivalent Fuel Consumption (l/100 km)	Saving Rate (%)	Final SOC
SAC (learned parameter)	9.4283	23.5767	4.37	0.29
DDPG	9.5372	23.9056	3.04	0.28
SAC (fixed parameter)	10.6914	24.5879	0.26	0.31
ECMS	10.9499	24.6541	-	0.31

4.3. The Adaptability of SAC Algorithm-Based EMS

Many of the existing methods require elaborate designs to adapt to different driving cycles. In order to verify the adaptability of SAC strategies to stochastic driving cycles, we use real-world driving cycle c2 including the city driving cycle, suburban driving cycle and rural driving cycle, shown in Figure 14b. The maximum velocity of the driving cycle is 67.25 km/h, and the average velocity is 29.36 km/h. The actual and desired vehicle-speed trajectories of the PHEV are shown in Figure 17. It can be seen from the figure that the reference speed of the real-world driving cycle is followed very well. The SOC trajectory of the real-world driving cycle c2 is shown in Figure 18. The demand power is relatively low in phase T1 and the SOC drops slowly. Phase T2 has a higher demand power and the SOC drops sharply, eventually to the reference SOC (0.3). Phase T3 enters the ICE charging process, keeping the SOC fluctuating around the reference SOC. Similarly, we simulated five times under this driving cycle. The average values of the ICE and equivalent fuel consumption for 100 km are 9.951 l and 18.056 l, respectively.

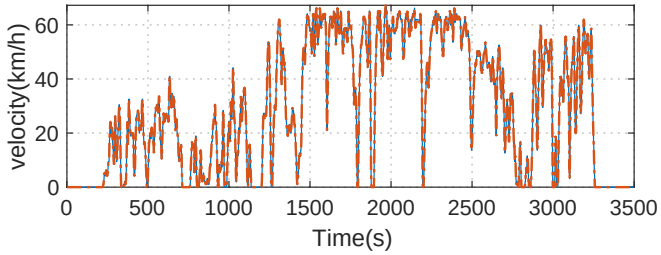


Figure 17. The actual and desired vehicle-speed trajectories.

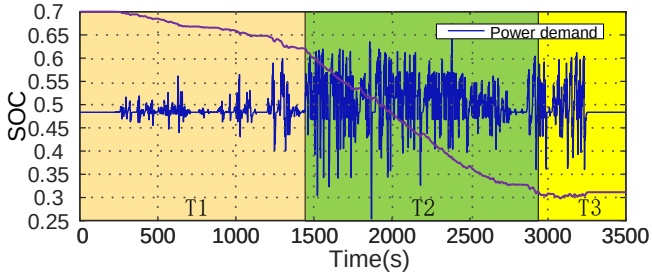


Figure 18. SOC trajectory of real-world driving cycle c2.

5. Conclusions

In this paper, an improved SAC algorithm-based EMS is proposed to improve the fuel economy of EMS. The two Q networks are used to solve the problem of overestimation of Q values. The equilibrium parameter, which is the most important parameter of SAC algorithm-based EMS, can be self-adjusted by learning. The simulation results show that the SOC is able to be maintained at the various reference value under a real-world driving cycle. Compared with the SAC strategy with fixed equilibrium parameters, the introduction of a self-learning equilibrium parameter can improve fuel economy. The fuel economy of the proposed SAC algorithm-based strategy gets 4.37% performance than ECMS. The control effects of the DDPG strategy are close to that of the proposed SAC strategy, but the proposed SAC algorithm-based EMS is able to explore a wider action space, which then facilitated the ICE to work in a more favorable operating zone for the strategy. The proposed SAC algorithm-based strategy is more adaptable to different driving cycles.

Author Contributions: Conceptualization, T.L. and W.C.; methodology, T.L.; software, T.L.; validation, T.L., N.C. and W.C.; formal analysis, T.L.; investigation, T.L.; resources, N.C.; data curation, T.L. and W.C.; writing—original draft preparation, T.L., N.C. and W.C.; writing—review and editing, T.L. and N.C.; visualization, T.L.; supervision, N.C.; project administration, N.C.; funding acquisition, N.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China grant number U1864205; Key Technology Research and Development Program of Shandong Province grant number 2019JZZY020814. The APC was funded by U1864205.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Deng, Z.; Weng, D.; Chen, J.; Liu, R.; Wang, Z.; Bao, J.; Zheng, Y.; Wu, Y. Airvis: Visual analytics of air pollution propagation, *IEEE Trans. Vis. Comput. Graph.* **2020**, *26*, 800–810.
2. Djelailia, O.; Necaibia, S.; Kelaiaia, M.S.; Merad, F.; Labar, H.; Chouial, H. Optimal fuel consumption planning and energy management strategy for a hybrid energy system with pumped storage. In Proceedings of the 2019 1st International Conference on Sustainable Renewable Energy Systems and Applications (ICSRESA), Tébessa, Algeria, 4–5 December 2019; pp. 1–6.
3. Ceraolo, M.; di Donato, A.; Franceschi, G. A general approach to energy optimization of hybrid electric vehicles. *IEEE Trans. Veh. Technol.* **2008**, *57*, 1433–1441. [CrossRef]

4. Mahyiddin, S.H.; Mohamed, M.R.; Mustaffa, Z.; Khor, A.C.; Sulaiman, M.H.; Ahmad, H.; Rahman, S.A. *Fuzzy Logic Energy Management System of Series Hybrid Electric Vehicle*; IET Conference Publications: Kuala Lumpur, Malaysia, 2016; pp. 1–6.
5. Zhang, F.; Hu, X.; Langari, R.; Cao, D. Energy management strategies of connected hevs and phevs: Recent progress and outlook. *Prog. Energy Combust. Sci.* **2019**, *73*, 235–256. [CrossRef]
6. Liu, K.; Jiao, X.; Yang, C.; Wang, W.; Xiang, C.; Wang, W. Event-triggered intelligent energy management strategy for plug-in hybrid electric buses based on vehicle cloud optimisation. *IET Intell. Transp. Syst.* **2020**, *14*, 1153–1162. [CrossRef]
7. Sierra, A.; Herrera, V.; González-Garrido, A.; Milo, A.; Gaztañaga, H.; Camblong, H. Experimental comparison of energy management strategies for a hybrid electric bus in a test-bench. In Proceedings of the 2018 Thirteenth International Conference on Ecological Vehicles and Renewable Energies (EVER), Monte-Carlo, Monaco, 5–7 May 2018; pp. 1–9.
8. Ma, G.; Ghasemi, M.; Song, X. Integrated powertrain energy management and vehicle coordination for multiple connected hybrid electric vehicles. *IEEE Trans. Veh. Technol.* **2018**, *67*, 2893–2899. [CrossRef]
9. Chen, H.; Xiong, R.; Lin, C.; Shen, W. Model predictive control based real-time energy management for hybrid energy storage system. *CSEE J. Power Energy Syst.* **2021**, *7*, 862–874.
10. Zhang, Y.; Ma, R.; Zhao, D.; Huang, F.; Liu, W. A Novel Energy Management Strategy Based on Dual Reward Function Q-learning for Fuel Cell Hybrid Electric Vehicle. *IEEE Trans. Ind. Electron.* **2022**, *69*, 1537–1547. [CrossRef]
11. Du, G.; Zou, Y.; Zhang, X.; Kong, Z.; Wu, J.; He, D. Intelligent energy management for hybrid electric tracked vehicles using online reinforcement learning. *Appl. Energy* **2019**, *251*, 113388. [CrossRef]
12. Lian, R.; Tan, H.; Peng, J.; Li, Q.; Wu, Y. Cross-type transfer for deep reinforcement learning based hybrid electric vehicle energy management. *IEEE Trans. Veh. Technol.* **2020**, *69*, 8367–8380. [CrossRef]
13. Wang, P.; Li, Y.; Shekhar, S.; Northrop, W.F. Actor-critic based deep reinforcement learning framework for energy management of extended range electric delivery vehicles. In Proceedings of the 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Hong Kong, China, 8–12 July 2019; pp. 1379–1384.
14. Qiu, L.; Qian, L.; Zomorodi, H.; Pisu, P. Global optimal energy management control strategies for connected four-wheel-drive hybrid electric vehicles. *IET Intell. Transp. Syst.* **2017**, *11*, 264–272. [CrossRef]
15. Ostadian, R.; Ramoul, J.; Biswas, A.; Emadi, A. Intelligent energy management systems for electrified vehicles: Current status, challenges, and emerging trends. *IEEE Open J. Veh. Technol.* **2020**, *1*, 279–295. [CrossRef]
16. Li, P.; Jiao, X.; Li, Y. Adaptive real-time energy management control strategy based on fuzzy inference system for plug-in hybrid electric vehicles. *Control Eng. Pract.* **2021**, *107*, 104703. [CrossRef]
17. Zhou, S.; Chen, Z.; Huang, D.; Lin, T. Adaptive real-time energy management control strategy based on fuzzy inference system for plug-in hybrid electric vehicles. *IEEE Trans. Power Electron.* **2021**, *36*, 5926–5940. [CrossRef]
18. Natella, D.; Mostacciuolo, E.; Baccari, S.; Vasca, F. A velocity-thresholds power splitting optimization for hybrid electric vehicles. In Proceedings of the 2019 18th European Control Conference (ECC), Naples, Italy, 25–28 June 2019; pp. 4148–4153.
19. Hu, B.; Li, J. An adaptive hierarchical energy management strategy for hybrid electric vehicles combining heuristic domain knowledge and data-driven deep reinforcement learning. *IEEE Trans. Transp. Electrif.* **2022**, *8*, 3275–3288. [CrossRef]
20. Zeng, X.; Wang, J. A parallel hybrid electric vehicle energy management strategy using stochastic model predictive control with road grade preview. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 2416–2423. [CrossRef]
21. Zhang, F.; Hu, X.; Langari, R.; Wang, L.; Cui, Y.; Pang, H. Adaptive energy management in automated hybrid electric vehicles with flexible torque request. *Energy* **2021**, *214*, 118873. [CrossRef]
22. Liu, T.; Zou, Y.; Liu, D.; Sun, F. Reinforcement learning of adaptive energy management with transition probability for a hybrid electric tracked vehicle. *IEEE Trans. Ind. Electron.* **2015**, *62*, 7837–7846. [CrossRef]
23. Miro-Padovani, T.; Colin, G.; Ketfi-Chérif, A.; Chamaillard, Y. Implementation of an energy management strategy for hybrid electric vehicles including drivability constraints. *IEEE Trans. Veh. Technol.* **2016**, *65*, 5918–5929. [CrossRef]
24. Park, S.; Ahn, C. Power management controller for a hybrid electric vehicle with predicted future acceleration. *IEEE Trans. Veh. Technol.* **2019**, *68*, 10477–10488. [CrossRef]
25. Zhang, Y.; Chu, L.; Fu, Z.; Xu, N.; Guo, C.; Zhao, D.; Ou, Y.; Xu, L. Energy management strategy for plug-in hybrid electric vehicle integrated with vehicle-environment cooperation control. *Energy* **2020**, *197*, 117192. [CrossRef]
26. Li, J.; Liu, Y.; Qin, D.; Li, G.; Chen, Z. Research on equivalent factor boundary of equivalent consumption minimization strategy for phevs. *IEEE Trans. Veh. Technol.* **2020**, *69*, 6011–6024. [CrossRef]
27. Feng, T.; Yang, L.; Gu, Q.; Hu, Y.; Yan, T.; Yan, B. A supervisory control strategy for plug-in hybrid electric vehicles based on energy demand prediction and route preview. *IEEE Trans. Veh. Technol.* **2015**, *64*, 1691–1700.
28. Borhan, H.; Vahidi, M. Phillips, A. L.; Kuang, M.; Kolmanovsky, I.V.; Di Cairano, S. Mpc-based energy management of a power-split hybrid electric vehicle. *IEEE Trans. Control Syst. Technol.* **2012**, *20*, 593–603. [CrossRef]
29. Uebel, S.; Murgovski, N.; Bäker, B.; Sjöberg, J. A two-level mpc for energy management including velocity control of hybrid electric vehicles. *IEEE Trans. Veh. Technol.* **2019**, *68*, 5494–5505. [CrossRef]
30. Mariani, V.; Rizzo, G.; Tiano, F.; Glielmo, L. A model predictive control scheme for regenerative braking in vehicles with hybridized architectures via aftermarket kits. *Control Eng. Pract.* **2022**, *123*, 105142. [CrossRef]
31. Tipaldi, M.; Iervolino, R.; Massenio, P. R. Reinforcement learning in spacecraft control applications: Advances, prospects, and challenges. *Annu. Rev. Control* **2022**, *in press*. [CrossRef]

32. Zhou, Q.; Li, J.; Shuai, B.; Williams, H.; He, Y.; Li, Z.; Xu, H.; Yan, F. Multi-step reinforcement learning for model-free predictive energy management of an electrified off-highway vehicle. *Appl. Energy* **2019**, *255*, 113755. [CrossRef]
33. Xu, B.; Malmir, F.; Filipi, Z. *Real-Time Reinforcement Learning Optimized Energy Management for a 48v Mild Hybrid Electric Vehicle*; SAE Technical Papers 2019-01-1208; SAE: Warrendale, PA, USA, 2019.
34. Chen, Z.; Hu, H.; Wu, Y.; Xiao, R.; Shen, J.; Liu, Y. Energy management for a power-split plug-in hybrid electric vehicle based on reinforcement learning. *Appl. Sci.* **2018**, *8*, 2494. [CrossRef]
35. He, D.; Zou, Y.; Wu, J.; Zhang, X.; Zhang, Z.; Wang, R. Deep q-learning based energy management strategy for a series hybrid electric tracked vehicle and its adaptability validation. In Proceedings of the 2019 IEEE Transportation Electrification Conference and Expo (ITEC), Detroit, MI, USA, 19–21 June 2019; pp. 1–6.
36. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
37. Wu, J.; He, H.; Peng, J.; Li, Y.; Li, Z. Continuous reinforcement learning of energy management with deep q network for a power split hybrid electric bus. *Appl. Energy* **2018**, *222*, 799–811. [CrossRef]
38. Han, X.; He, H.; Wu, J.; Peng, J.; Li, Y. Energy management based on reinforcement learning with double deep q-learning for a hybrid electric tracked vehicle. *Appl. Energy* **2019**, *254*, 113708.
39. Chen, Z.; Gu, H.; Shen, S.; Shen, J. Energy management strategy for power-split plug-in hybrid electric vehicle based on mpc and double q-learning. *Energy* **2022**, *245*, 123182. [CrossRef]
40. Zou, R.; Fan, L.; Dong, Y.; Zheng, S.; Hu, C. Dql energy management: An online-updated algorithm and its application in fix-line hybrid electric vehicle. *Energy* **2021**, *225*, 120174. [CrossRef]
41. Wu, Y.; Tan, H.; Peng, J.; Zhang, H.; He, H. Deep reinforcement learning of energy management with continuous control strategy and traffic information for a series-parallel plug-in hybrid electric bus. *Appl. Energy* **2019**, *247*, 454–466. [CrossRef]
42. Li, Y.; He, H.; Khajepour, A.; Wang, H.; Peng, J. Energy management for a power-split hybrid electric bus via deep reinforcement learning with terrain information. *Appl. Energy* **2019**, *255*, 113762. [CrossRef]
43. Yue, S.; Wang, Y.; Xie, Q.; Zhu, D.; Pedram, M.; Chang, N. Model-free learning-based online management of hybrid electrical energy storage systems in electric vehicles. In Proceedings of the IECON 2014-40th Annual Conference of the IEEE Industrial Electronics Society, Dallas, TX, USA, 29 October–1 November 2014; pp. 3142–3148.
44. Qiu, C.; Hu, Y.; Chen, Y.; Zeng, B. Deep deterministic policy gradient (ddpg)-based energy harvesting wireless communications. *IEEE Internet Things J.* **2019**, *6*, 8577–8588. [CrossRef]
45. Lian, R.; Peng, J.; Wu, Y.; Tan, H.; Zhang, H. Rule-interposing deep reinforcement learning based energy management strategy for power-split hybrid electric vehicle. *Energy* **2020**, *197*, 117297. [CrossRef]
46. Henderson, P.; Islam, R.; Bachman, P.; Pineau, J.; Precup, D.; Meger, D. Deep reinforcement learning that matters. In Proceedings of the AAAI conference on artificial intelligence, New Orleans, LA, USA, 2–7 February 2018.
47. Yang, J.; Zhang, J.; Xi, M.; Lei, Y.; Sun, Y. A deep reinforcement learning algorithm suitable for autonomous vehicles: Double bootstrapped soft-actor-critic-discrete. *IEEE Trans. Cogn. Dev. Syst.* **2021**, 1. [CrossRef]
48. Nguyen, B.-H.; German, R.; Trovão, J.P.F.; Bouscayrol, A. Real-time energy management of battery/supercapacitor electric vehicles based on an adaptation of pontryagin's minimum principle. *IEEE Trans. Veh. Technol.* **2019**, *68*, 203–212. [CrossRef]
49. Haarnoja, T.; Zhou, A.; Ha, S.; Tan, J.; Tucker, G.; Levine, S. Learning to walk via deep reinforcement learning. *arXiv* **2018**, arXiv:1812.11103.
50. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International Conference on Machine Learning 2018, Macau, China, 26–28 February 2018.
51. Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. Soft actor-critic algorithms and applications. *arXiv* **2018**, arXiv:1812.05905.
52. Fu, F.; Kang, Y.; Zhang, Z.R.; Zhang, Z.; Yu, F.R.; Wu, T. Soft actor–critic drl for live transcoding and streaming in vehicular fog-computing-enabled iov. *IEEE Internet Things J.* **2021**, *8*, 1308–1321. [CrossRef]