applied sciences

MDPI

Article

Energy Management Strategy for a Hybrid Electric Vehicle Based on Deep Reinforcement Learning

Yue Hu [1,2,3,†], Weimin Li [1,3,4,*], Kun Xu [1,†], Taimoor Zahid [1,2], Feiyan Qin [1,2] and Chenming Li [5]

1 Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China; yue.hu@siat.ac.cn (Y.H.); kun.xu@siat.ac.cn (K.X.); taimoor.z@siat.ac.cn (T.Z.); fy.qin@siat.ac.cn (F.Q.)
2 Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen 518055, China
3 Jining Institutes of Advanced Technology, Chinese Academy of Sciences, Jining 272000, China
4 Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong 999077, China
5 Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong 999077, China; 1155101099@link.cuhk.edu.hk
* Correspondence: wm.li@siat.ac.cn; Tel.: +86-159-1537-4149
† These authors contributed equally to this work.

Received: 28 December 2017; Accepted: 24 January 2018; Published: 26 January 2018

Abstract: An energy management strategy (EMS) is important for hybrid electric vehicles (HEVs) since it plays a decisive role on the performance of the vehicle. However, the variation of future driving conditions deeply influences the effectiveness of the EMS. Most existing EMS methods simply follow predefined rules that are not adaptive to different driving conditions online. Therefore, it is useful that the EMS can learn from the environment or driving cycle. In this paper, a deep reinforcement learning (DRL)-based EMS is designed such that it can learn to select actions directly from the states without any prediction or predefined rules. Furthermore, a DRL-based online learning architecture is presented. It is significant for applying the DRL algorithm in HEV energy management under different driving conditions. Simulation experiments have been conducted using MATLAB and Advanced Vehicle Simulator (ADVISOR) co-simulation. Experimental results validate the effectiveness of the DRL-based EMS compared with the rule-based EMS in terms of fuel economy. The online learning architecture is also proved to be effective. The proposed method ensures the optimality, as well as real-time applicability, in HEVs.

Keywords: hybrid electric vehicle; energy management strategy; deep reinforcement learning; online learning

1. Introduction

An energy management strategy (EMS) is one of the key technologies for hybrid electric vehicles (HEVs) due to its decisive effect on the performance of the vehicle [1]. The EMS for HEVs has been a very active research field during the past decades. However, how to design a highly-efficient and adaptive EMS is still a challenging task due to the complex structure of HEVs and the uncertain driving cycle.

The existing EMS methods can be generally classified into the following three categories: (1) Rule-based EMS, such as the thermostatic strategy, the load following strategy, and electric assist strategy [2,3]. These methods rely heavily on the results of extensive experimental trials and human expertise without the a priori knowledge of the driving conditions [4]. Other related control strategies employ heuristic control techniques, with the resultant strategies formalized as fuzzy rules [5,6]. Though these rule-based strategies are effective and can be easily implemented, their optimality

and flexibility are critically limited by working conditions and, consequently, are not adaptive to different driving cycles. (2) Optimization-based EMS: some optimization methods employed in control strategy are either based on the known driving cycles or predicted future driving conditions, such as dynamic programming (DP) [7–9], sequential quadratic programming (SQP), genetic algorithms (GA) [10], the Pontryagin minimum principle (PMP) [11], and so on. Usually, these algorithms can manage to determine the optimal power split between the engine and the motor for a particular driving cycle. However, the obtained optimal power-split solutions are only optimal with respect to a specific driving cycle. In general, it is neither optimal nor charge-sustaining for other cycles. Unless future driving conditions can be predicted during real-time operation, there is no way to imply these control laws directly. Moreover, these methods suffer from the "curse of dimensionality" problem, which prevents their wide adoption in real-time applications. Model predictive control (MPC) [12] is another type of optimization-based method. The optimal control problem in the finite domain is solved at each sampling instant and control actions are obtained based on online rolling optimization. This method has the advantages of good control effect and strong robustness. (3) Learning-based EMS: some strategies can learn from the historical data or use the previous driving data for online learning or application [13,14]. Some researchers propose that traffic information and cloud computing in intelligent transportation systems (ITSs) can enhance HEV energy management since vehicles obtain real-time data via intelligent infrastructures or connected vehicles [15,16]. Regardless of the learning from historical data or predicted data, these EMS methods also need complex control models and professional knowledge from experts. Thus, these EMS methods are not end-to-end control methods. Reinforcement learning-based control methods have also been used for HEV energy management [17,18]. However, reinforcement learning must be able to learn from a scalar reward signal that is frequently sparse, noisy, and delayed. Additionally, the sequence of highly-correlated states is also a large problem of reinforcement learning, in addition to the data distribution changes, as the algorithm learns new behaviors in reinforcement learning.

The learning-based EMS is an emerging and promising method because of its potential ability of self-adaption according to different driving conditions, even if there are still some problems. In our previous research, online learning control strategies based on neural dynamic programming (NDP) [19], fuzzy Q-learning (FQL) [20], were proposed. These strategies do not rely on prior information related to future driving conditions and can self-tune the parameters of the algorithms. A back propagation (BP) neural network was used to estimate the Q-value which, in turn, tuned the parameter of fuzzy controller [20]. However, it also requires designing the fuzzy controller, as well as professional knowledge.

Deep reinforcement learning (DRL) has shown successful performance in playing Atari [21] and Go games [22] in recent years. The DRL method is a powerful algorithm to solve complex control problems and handle large state spaces by establishing a deep neural network to relate the value estimation and associated state-action pairs. As a result, the DRL algorithm has been quickly applied in robotics [23], building HVAC control [24], ramp metering [25], and other fields. In the automotive field, DRL has been used for lane keeping assist [26], autonomous braking system [27], and autonomous vehicles [28]. However, motion control of autonomous vehicles needs very high precision from our perspective. The mechanism of DRL has not been explained very deeply and may not meet this high requirement.

Nevertheless, DRL is a powerful technique that can be used in HEV EMS in this research as it concerns the fuel economy compared to the control precision. A DRL-based EMS has been designed for plug-in hybrid electric vehicles (PHEVs) [29]. This is the first time DRL has been applied to a PHEV EMS. However, there are several problems in this study: (1) The learning process is still offline, which means that the trained deep network can only work well in the same driving cycle, but would not be able to obtain good performance in other driving conditions. As a result, this method can be used in buses with fixed route however it is not acceptable for vehicles with route variation; (2) The immediate reward is important as it affects the performance of DRL. The optimization objective is

the vehicle fuel economy, but the reward is a function based on the power supply from the engine. The relationship between fuel economy and engine power is complex and the paper lacks the ability to justify this phenomena; (3) The structure of deep neural network can be well designed by fixing the Q targets network, which can make the algorithm more stable.

In this research, an energy management strategy based on deep reinforcement learning is proposed. Our work achieves good performance and high scalability by (1) building the system model of the HEV and formulating the HEV energy management problem; (2) developing a DRL-based control framework and an online learning architecture for a HEV EMS, which is adapted to different driving conditions; and (3) facilitating algorithm training and evaluation in the simulation environment. Figure 1 illustrates our DRL-based algorithm for HEV EMS. The DRL-based EMS can autonomously learn the optimal policy based on data inputs, without any prediction or predefined rules. For training and validation, we use the HEV model built in ADVISOR software (National Renewable Energy Laboratory, Golden, CO, USA). Simulation results reveal that the algorithm is able to improve the fuel economy while meeting other requirements, such as dynamic performance and vehicle drivability.

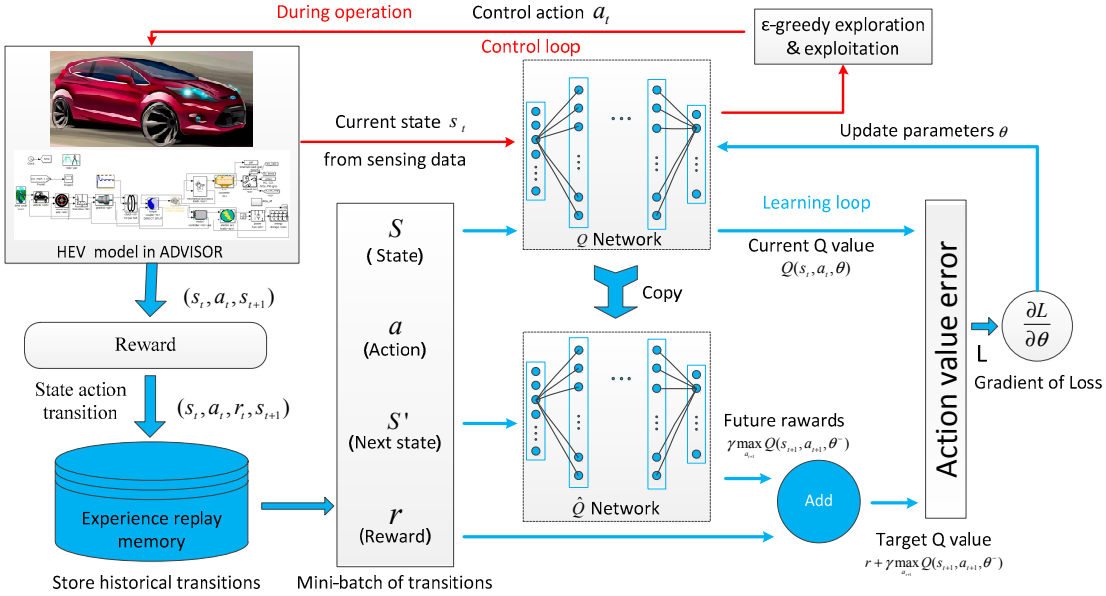

Figure 1. Deep reinforcement learning (DRL)-based framework for HEV EMS.

The proposed DRL-based EMS uses a fixed target Q network which can make the algorithm more stable. The immediate reward is a function directly related to fuel consumption. More importantly, a DRL-based online learning architecture is presented. It is a critical factor to apply the DRL algorithm in HEV energy management under different driving conditions.

The rest of this paper is organized as follows: Section 2 introduces the system model of HEV and describes the mathematics formulation of HEV EMS. Section 3 explains our deep reinforcement learning-based control strategy, including offline learning and online learning application. The experimental results are given in Section 4, followed by the conclusions in Section 5.

2. Problem Formulation

The prototype vehicle is a single-axis parallel HEV, the drivetrain structure of which is shown in Figure 2. The drivetrain integrates an engine, an electric traction motor/generator, Ni-Hi batteries, an automatic clutch, and an automatic/manual transmission system. The motor is directly linked between the auto clutch output and the transmission input. This architecture provides the regenerative braking during deceleration and allows an efficient motor assist operation. To provide pure electrical propulsion, the engine can be disconnected from the drivetrain by the automatic clutch. We have

adopted the vehicle model from our previous work [19,20] for this research. The key parameters of this vehicle are given in Table 1.

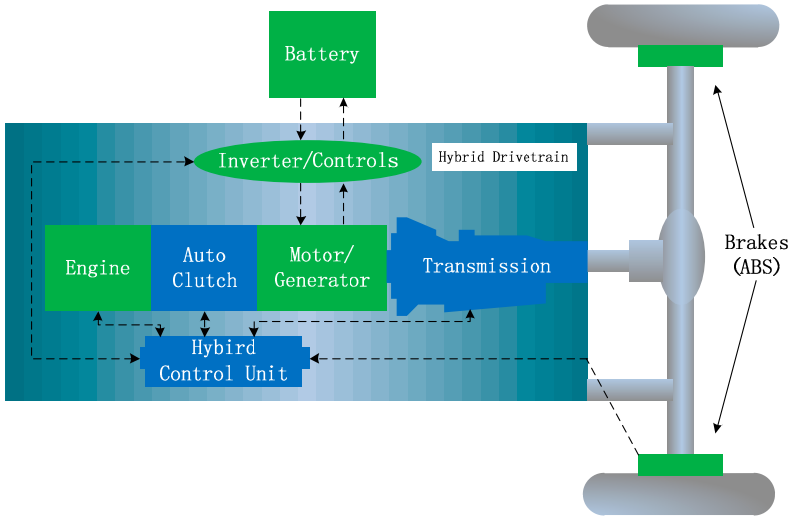

Figure 2. Drivetrain structure of the HEV.

Table 1. Summary of the HEV parameters.

Part or Vehicle	Parameters Value
Spark Ignition (SI) engine	Displacement: 1.0 L Maximum power: 50 kW/5700 r/min Maximum torque: 89.5 Nm/5600 r/min
Permanent magnet motor	Maximum power: 10 kW Maximum torque: 46.5 Nm
Advanced Ni-Hi battery	Capacity: 6.5 Ah Nominal cell voltage: 1.2 V Total cells: 120
Automated manual transmission	5-speed GR: 2.2791/2.7606/3.5310/5.6175/11.1066
Vehicle	Curb weight: 1000 kg

In the following, key concepts of the DRL-based EMS are formulated:

System state: In the DRL algorithm, control action is directly determined by the system states. In this study, the total required torque (T_{dem}) and the battery state-of–charge (SOC) are selected to form a two-dimensional state space, i.e., $s(t) = (T_{dem}(t), SOC(t))^T$, where $T_{dem}(t)$ represents the required torque at time t, and $SOC(t)$ represents the battery state of charge at time t.

Control action: The decision-making on the torque-split ratio between the internal combustion engine (ICE) and battery is the core problem of the HEV energy management strategy. We choose the output torque from the ICE as the control action in this study, denoted as $A(t) = T_e(t)$, where t is the time step index. $T_e(t)$ should be discretized in order to apply the DRL-based algorithm, i.e., the entire action space is $A = \{A^1, A^2, ..., A^n\}$, where n is the degree of discretization. In this research, we consider n as 24. The motor output torque $T_m(t)$ can be obtained by subtracting $T_e(t)$ from $T_{dem}(t)$.

Immediate Reward: Immediate reward is important in the DRL algorithm because it directly influences the parameters tuning of the deep neural network (DNN). The DRL agent is always trying to maximize the reward which it can obtain by taking the optimal action at each time step.

Therefore, the immediate reward should be defined according to the optimization objective. The control objective of the HEV EMS is to minimize vehicle fuel consumption and emissions along a driving mission. Meanwhile, the vehicle drivability and battery health should be satisfied. In this work, we focus more on fuel economy of the HEV; the emissions are not taken into consideration. Keeping this objective in mind, the reciprocal of the ICE fuel consumption at each time step is defined as the immediate reward. A penalty value is introduced to penalize the situation when the SOC exceeds the threshold. Immediate reward is defined by the following equations:

$$r_{ss'}^{a} = \begin{cases} \frac{1}{C_{ICE}} & C_{ICE} \neq 0 \cap 0.4 \leq SOC \leq 0.85 \\ \frac{1}{C_{ICE}+C} & C_{ICE} \neq 0 \cap SOC < 0.4 \; or \; SOC > 0.85 \\ \frac{2}{Min_{C_{ICE}}} & C_{ICE} = 0 \cap 0.4 \leq SOC \\ -\frac{1}{C} & C_{ICE} = 0 \cap SOC < 0.4 \end{cases} \tag{1}$$

where $r_{ss'}^{a}$ is the immediate reward generated when state changes from s to s' by taking action a; C_{ICE} is the instantaneous fuel consumption value of the ICE; C is the numerical penalty, as well as the maximum instantaneous power supply from the ICE; $Min_{C_{ICE}}$ is the minimum nonzero value of the ICE instantaneous fuel consumption value. The SOC variation range is from 40% to 85% in this study. This definition can guarantee the lower ICE fuel consumption while satisfying the SOC constrains.

Formally, the goal of the EMS of the HEV is to find the optimal control strategy, π^*, that maps the observed states s_t to the control action a_t. Mathematically, the control strategy of the HEV can be formulated as an infinite horizon dynamic optimization problem as follows:

$$R = \sum_{t=0}^{\infty} \gamma^t r(t) \tag{2}$$

where $r(t)$ is the immediate reward incurred by a_t at time t; and $\gamma \in (0,1)$ is a discount factor that assures the infinite sum of cost function convergence. We use $Q^*(s_t, a_t)$, i.e., the optimal value, to represent the maximum accumulative reward which we can obtain by taking action a_t in state s_t. $Q^*(s_t, a_t)$ is calculated by the Bellman Equation as follows:

$$Q^*(s_t, a_t) = E[r_{t+1} + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}) | s_t, a_t] \tag{3}$$

The Q-learning method is used to update the value estimation, as shown in Equation (4).

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \eta(r_{t+1} + \gamma \max_{a_{t+1}} Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)) \tag{4}$$

where $\eta \in (0,1]$ represents the learning rate. Such a value iteration algorithm converges to the optimal action value function, $Q_t \to Q^*$ as $t \to \infty$.

3. Deep Reinforcement Learning-Based EMS

Deep reinforcement learning-based EMS is developed which combines a deep neural network and conventional reinforcement learning. The EMS makes decisions only based on the current system state since the proposed EMS is an end-to-end control strategy. This deep reinforcement neural network can also be called a deep Q-network (DQN). In the rest of this section, value function approximation, DRL algorithm design, and the DRL-based algorithm online learning application are presented.

3.1. Value Function Approximation

The state-action value is represented by a large, but limited, number of states and actions table, i.e., the Q table, in conventional reinforcement learning. However, a deep neural network is taken in this work to approximate the Q-value by Equation (3). As depicted in Figure 3, the inputs of the

network are the system states, which are defined in Section 2. The rectified linear unit (ReLU) is used as the activation function for hidden layers, and the linear layer is used for obtaining the action value at the output layer. In order to balance the exploration and exploitation, the $\varepsilon - greedy$ policy is used for action selection, i.e., the policy chooses the maximum Q-value action with probability $1 - \varepsilon$ and selects a random action with probability ε.

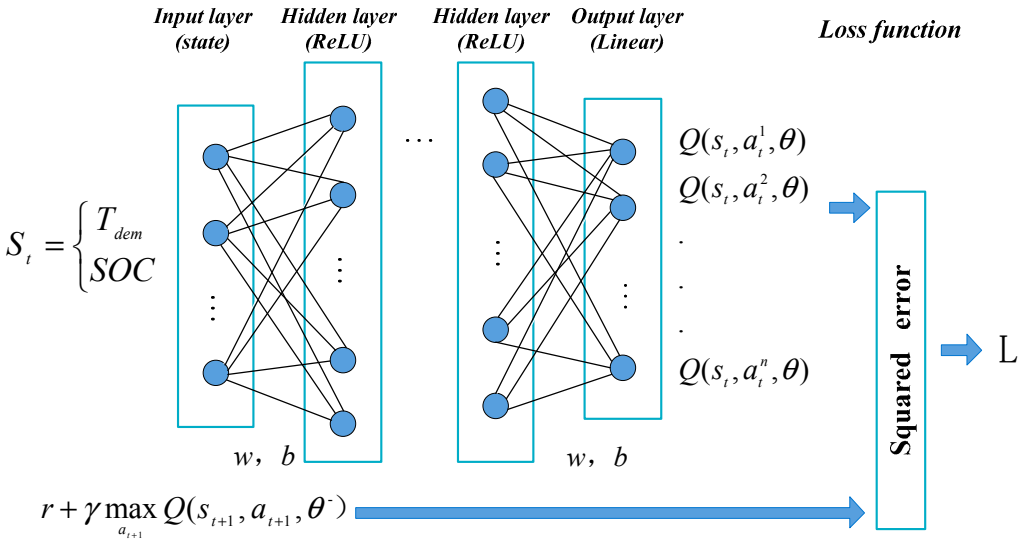

Figure 3. Structure of the neural network.

The Q-value estimation for all control actions can be calculated by performing a forward calculation in the neural network. The mean squared error between the target Q-value and the inferred output of neural network is defined as loss function in Equation (5):

$$L(\theta) = E[(r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}, \theta^-) - Q(s_t, a_t, \theta))^2] \quad (5)$$

where $Q(s_t a_t, \theta)$ is the output of the neural network with the parameters θ. $r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}, \theta^-)$ is the target Q-value, using parameters θ^- from some previous iteration. This fixed target Q network makes the algorithm more stable. Parameters in the neural network are updated by the gradient descent method.

The inputs of DQN are total required torque T_{dem} and battery SOC. The variation range of SOC is from 0 to 1 and does not need preprocessing. However, the total required torque T_{dem} can vary significantly. In order to facilitate the learning process, we scale the total required torque T_{dem} to the range $[-1, 1]$ before feeding to the neural network as shown in Equation (6). The minimum and maximum values for T_{dem} can be obtained from historical observation:

$$T'_{dem} = \frac{T_{dem} - \min(T_{dem})}{\max(T_{dem}) - \min(T_{dem})} \quad (6)$$

3.2. DRL Algorithm Design

Our DRL-based EMS control algorithm is presented in Algorithm 1. The outer loop controls the number of training episodes, while the inner loop performs the EMS control at each time step within one training episode.

Algorithm 1: Deep Q-Learning with Experience Replay

Initialize replay memory D to capacity N
Initialize action-value function Q with random weights θ
Initialize target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$
1: **For** episode = 1, M **do**
2: Reset environment: $s_0 = (SOC_{Initial}, T_0)$
3: **For** t = 1, T, **do**
4: With probability ε select a random action a_t
otherwise select $a_t = \max_{a_t} Q(s_t, a; \theta)$
5: Choose action a_t and observe the reward r_t
6: Set $s_{t+1} = (SOC_{t+1}, T_{t+1})$
7: Store (s_t, a_t, r_t, s_{t+1}) in memory D
8: Sample random mini-batch of (s_t, a_t, r_t, s_{t+1}) from D
9: **if** terminal s_{j+1}: Set $y_j = r_j$
else set $y_j = r_j + \gamma \max_{a_{j+1}} \hat{Q}(s_{j+1}, a_{j+1}; \theta-)$
10: Perform a gradient descent step on $(y_j - Q(s_j a_j; \theta))^2$
11: Every C steps reset $\hat{Q} = Q$
12: **end for**
13: **end for**

In order to avoid the strong correlations between the samples in a short time period of conventional RL, experience replay is adopted to store the experience (i.e., a batch of state, action, reward, and next state:(s_t, a_t, r_t, s_{t+1})) at each time step in a data experience pool. For each certain time, random samples of experience are drawn from the experiment pool and used to train the Q network.

We initialize memory D as an empty set. Then we initialize weights θ in the action-value function estimation Q neural network. In order to break the dependency loop between the target value and weights θ, a separate neural network $\hat{Q}$ with weights θ^{-} is created for calculating the target Q value.

We can set the maximum number of episode as M. During the learning process, in step 4, the algorithm selects the maximum Q value action with probability $1 - \varepsilon$ and selects a random action with probability ε based on the observation of the state. In step 5, action a_t is executed and reward r_t is obtained. In step 6, the system state becomes the next state. In step 7, the state action transition tuple is stored in memory. Then, a mini-batch of transition tuples is drawn randomly from the memory. Step 9 calculates the target Q value. The weights in neural network Q are updated by using the gradient descent method in step 10. The network $\hat{Q}$ is periodically updated by copying parameters from the network Q in step 11.

3.3. DRL-Based Algorithm Online Learning Application

In Section 3.2, the DRL-based algorithm is proposed, however, it is an offline learning algorithm which can only be applied in the simulation environment. More importantly, the training process can only be applied in limited driving cycles, therefore, the trained DQN only performs well under the learned driving conditions, which may not provide satisfactory results under other driving cycles. This is unacceptable in HEV real-time applications. As a result, online learning is necessary for DRL-based algorithms in HEV EMS applications.

The DRL-based online learning architecture is presented in Figure 4. Action execution and network training should be separated. There is a controller which contains a Q neural network and selects an action for the HEV while storing the state action transitions. When the HEV needs to learn a new driving cycle, the method of action selection will be the $\varepsilon - greedy$ method. Otherwise, the HEV can always select the maximum Q-value action. There is another on-board computer or remote computing center which is responsible for Q neural network training. The on-board computer or remote computing center obtains state action transitions from the action controller and trains the

neural network based on the DRL algorithm. The *Q* neural network is periodically updated by copying parameters from the on-board computer or remote computing center.

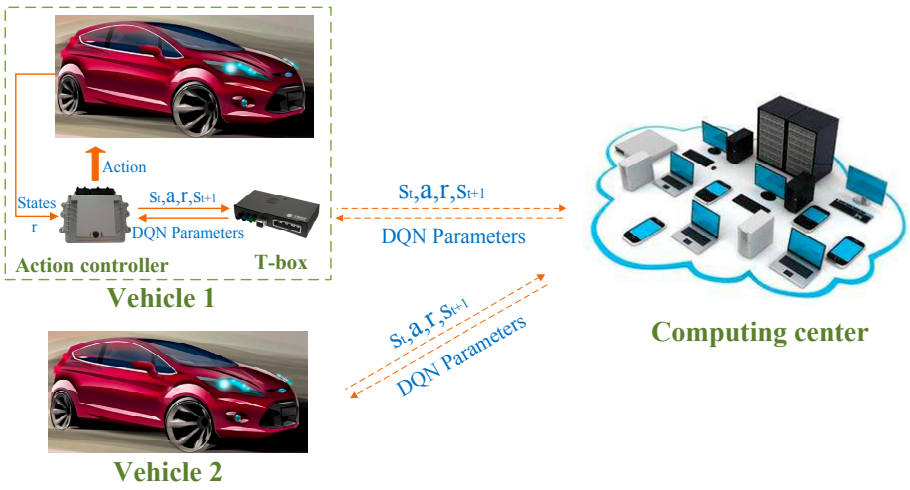

Figure 4. DRL-based online learning architecture.

The communication mode between the action controller and the on-board computer can be via CAN bus or Ethernet. If the *Q* neural network training is completed by a remote computing center, a vehicle terminal named Telematics-box (T-box) should be installed in the HEV in order to communicate with the remote computing center through the 3G communication network. A remote computing center can obtain state action transitions from other connected HEVs, as is shown in Figure 4. This is useful to train a large *Q* neural network which can deal with different driving conditions.

The main differences between online learning and offline learning are as follows: (1) online learning can adapt to varying driving conditions, while offline can only learn from the given driving cycles; (2) action execution and network training should be separated in online learning because of the limited on-board controller computing ability; and (3) online training efficiency should be higher than offline training since the vehicle must learn the optimal EMS with the shortest time. Thus, it is necessary to cluster the representative state action transitions and use the recent data in the experience pool.

Interestingly, offline learning and online learning can be combined to realize a good effect of EMS. For instance, we can train the DQN offline under the Urban Dynamometer Driving Schedule (UDDS), and then apply the online learning under the New European Driving Cycle (NEDC).

4. Experimental Results and Discussion

4.1. Offline Application

4.1.1. Experiment Setup

In order to evaluate the effectiveness of proposed DRL-based algorithm, simulation experiments are done in MATLAB and the ADVISOR co-simulation environment. The offline learning application is evaluated firstly and the UDDS driving cycle is used in the learning process. The simulation model for the HEV mentioned in Section 2 is built in ADVISOR. Meanwhile, the hyper parameters of the DRL-based algorithm used in the simulations are summarized in Table 2.

Table 2. Summary of the DRL-based algorithm hyper parameters.

Hyper Parameters	Value
mini-batch size	32
replay memory size	1000
discount factor γ	0.99
learning rate	0.00025
initial exploration	1
final exploration	0.2
replay start size	200

In this application, the input layer of the network has two neurons, i.e., T_{dem} and SOC. There are three hidden layers having 20, 50, and 100 neurons, respectively. The output layer has 24 neurons representing the discrete ICE torque. All these layers are fully connected. The network is trained with 50 episodes and each episode means a trip (1369 s).

We evaluate the performance of DRL-based EMS by comparing them with the rule-based EMS known as "Parallel Electric Assist Control Strategy" [20]. The initial SOC is 0.8.

4.1.2. Experimental Results

Firstly, we evaluate the learning performance of DRL-based algorithm. The track of average loss is recorded in Figure 5. It is clear that the average loss decreases quickly along the training process. Figure 6 depicts the track of the total reward of one episode along the training process. Even though the curve is oscillating, the overall trend of the track is rising. There are also some dramatic drops in the total reward during the training process. This is because of the adding of a large penalty when the algorithm selects actions that results in the violation of the SOC constraint.

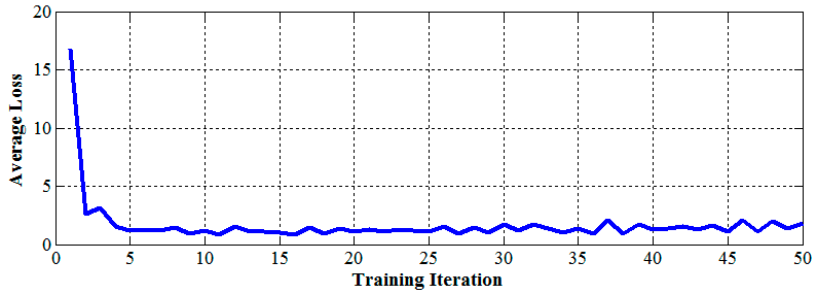

Figure 5. Track of loss.

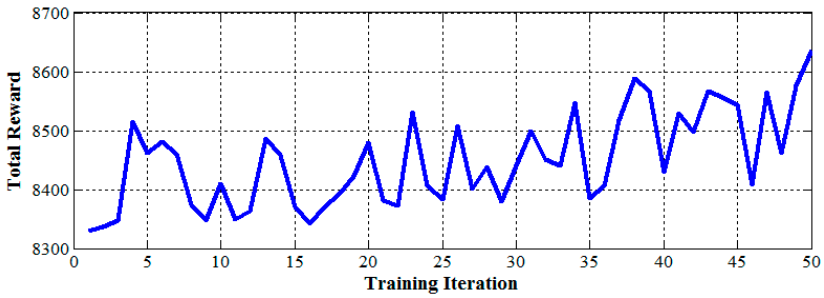

Figure 6. Track of the total reward.

Then, the simulation results of the trained DRL-based EMS for the UDDS driving cycle are shown in Figure 7. In order to evaluate the performance and effectiveness of the trained DRL-based EMS, comparison results are listed in Table 3. Power consumption is converted to fuel consumption; equivalent

fuel consumption is obtained by adding the converted power consumption and fuel consumption. As shown by the results of Table 3, fuel consumption is improved significantly compared to the rule-based control strategy, as fuel consumption is decreased by 10.09%. Meanwhile, the equivalent fuel consumption is also decreased by 8.05%. The DRL-based EMS achieves good performance. Notably, the rule-base EMS is designed by the experts while the DRL-based EMS only learns from the states and historical data.

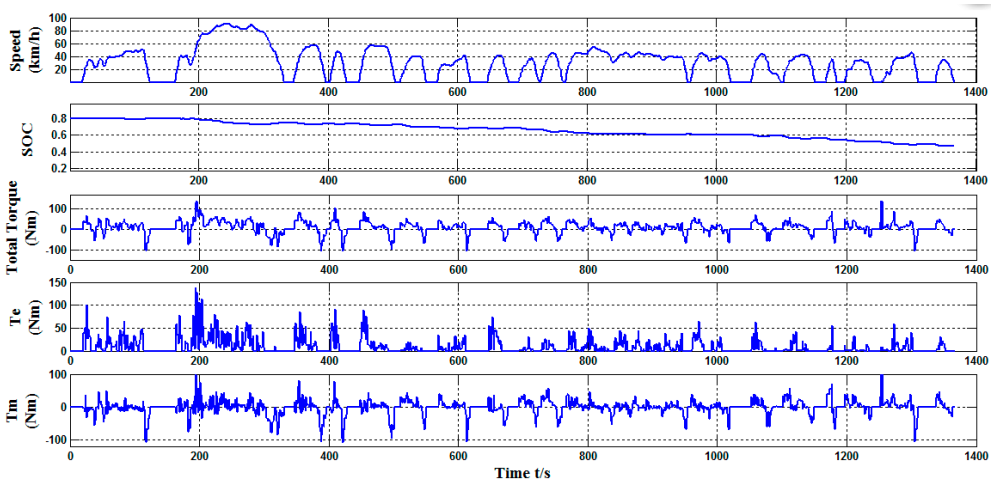

Figure 7. Simulation results of the trained EMS under UDDS.

Table 3. Comparison of the results under UDDS.

Control Strategy	Fuel Consumption (L/100 km)	Equivalent Fuel Consumption (L/100 km)
Rule-Based	3.857	3.861
DRL-based	3.468	3.550

4.2. Online Application

4.2.1. Experiment Setup

The DRL-based online learning architecture is presented in Section 3.2. In order to evaluate the online learning performance conveniently, we also use ADVISOR software to simulate the online learning working process. In the online learning application, the neural network setting is the same as the offline application. Two different kinds of simulations are performed. In the first scenario, the neural network parameters are random at the beginning and, in the second one, the neural network is pre-trained offline under the existing driving cycle before the online learning process. In the first case, the online learning simulation without any pre-training under the NEDC driving cycle is done. In the second case, we pre-train the neural network offline under the UDDS driving cycle firstly, and then apply the online learning under the NEDC driving cycle.

4.2.2. Experimental Results

In the first case, we trained the neural network 50 times under the NEDC driving cycle with the same initial condition. Unlike the offline learning, this process is online and simulates the vehicle running under the NEDC driving cycle.

The track of loss is depicted in Figure 8. The loss also decreases quickly along the training process in the online application. Figure 9 depicts the track of total reward and the fuel consumption of one driving cycle along the training process, and the overall trend of the total reward is the same as the offline application. This reveals the proposed DRL-based online learning architecture is effective.

As we can see from Figure 9, the trend of the total reward and the fuel consumption is nearly opposite. This reflects that the definition of the reward is suitable.

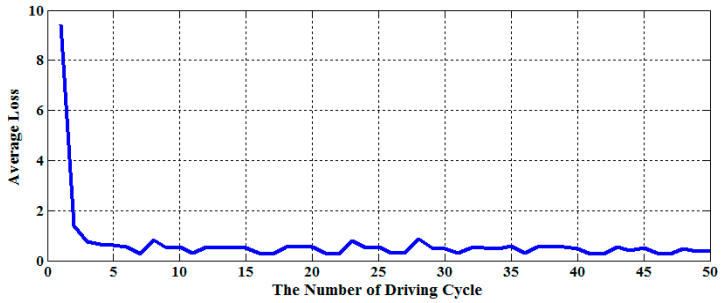

Figure 8. Track of loss in the online application.

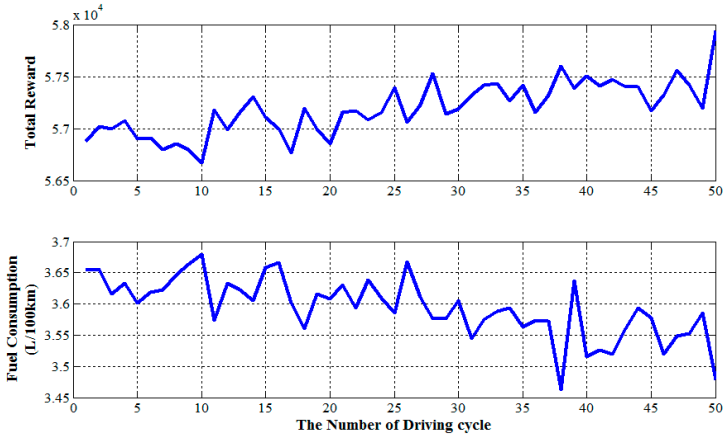

Figure 9. Track of the total reward and fuel consumption in the online application.

Simulation results of the online trained DRL-based EMS for the NEDC driving cycle are shown in Figure 10. The comparison of the results are listed in Table 4. Fuel consumption is also improved compared to the rule-based control strategy, as fuel consumption is decreased by 10.29%, while the equivalent fuel consumption is decreased by 2.57%.

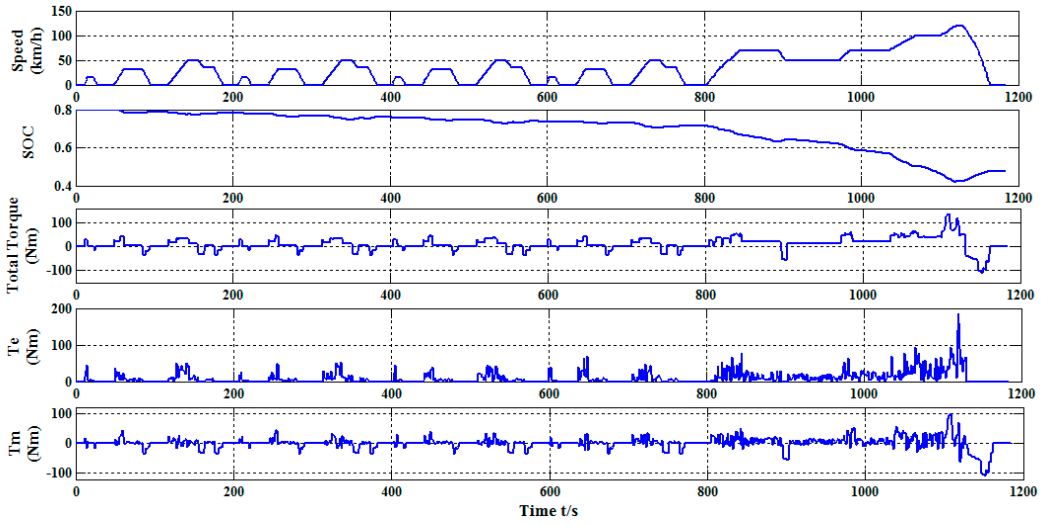

Figure 10. Simulation results of the trained EMS under NEDC.

Table 4. Comparison of the results under NEDC.

Control Strategy	Fuel Consumption (L/100 km)	Equivalent Fuel Consumption (L/100 km)
Rule-Based	3.877	3.892
DRL-based	3.478	3.792

In the second case, the neural network was pre-trained offline under the UDDS driving cycle, such that the DRL-based EMS can adapt to the UDDS driving cycle but have no a priori knowledge about the NEDC driving cycle. The comparison of the results between the offline trained DRL-based EMS under the UDDS driving cycle for the NEDC driving cycle and other control strategies are listed in Table 5. It is obvious that the offline-trained EMS under the UDDS driving cycle does not adapt well to the NEDC driving cycle.

Table 5. Comparison of the results under NEDC.

Control Strategy	Fuel Consumption (L/100 km)	Equivalent Fuel Consumption (L/100 km)
Rule-Based	3.877	3.892
DRL-based EMS trained under NEDC online	3.478	3.792
DRL-based EMS only pre-trained under UDDS offline	3.690	3.872

Based on the offline pre-trained EMS under the UDDS driving cycle, we can apply the online learning process under the NEDC driving cycle. We trained the pre-trained neural network 20 times under the NEDC driving cycle with the same initial conditions. After the training process, we tested the trained EMS under the NEDC driving cycle. The simulation results are shown in Figure 11. The comparison of the results are listed in Table 6. The results show that the pre-training process can contribute to effectively decrease the online training time. This is because the DRL-based EMS learns some of the same features between different driving conditions.

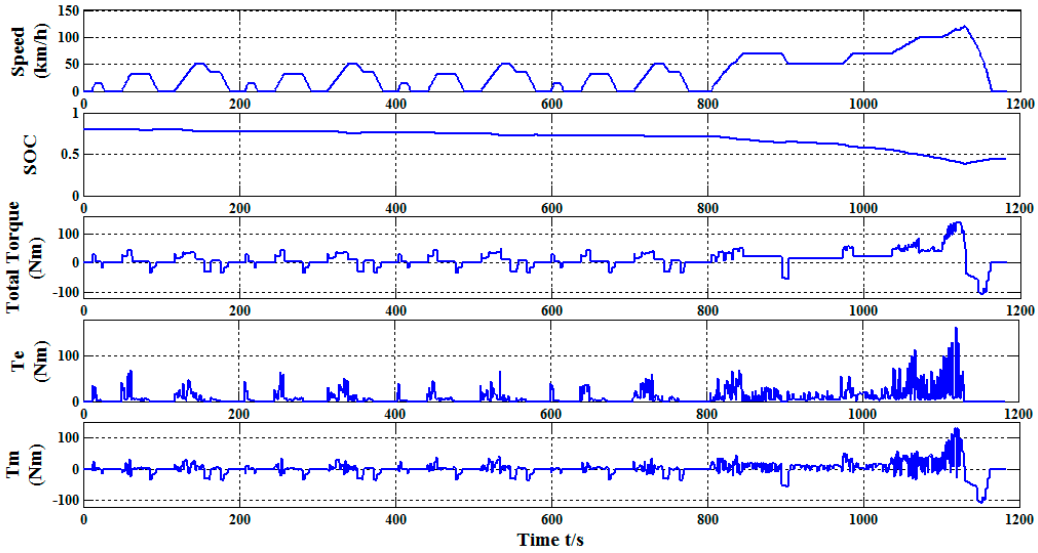

Figure 11. Simulation results under NEDC of the online trained EMS which was pre-trained under UDDS.

Table 6. Comparison of the results under NEDC.

Control Strategy	Fuel Consumption (L/100 km)	Equivalent Fuel Consumption (L/100 km)
DRL-based EMS trained under NEDC online	3.478	3.792
DRL-based EMS only pre-trained under UDDS offline	3.690	3.872
DRL-based EMS which pre-trained offline and trained under NEDC online	3.440	3.795

Figure 12 depicts the track of the total reward and the fuel consumption of one driving cycle along the training process, the curves are smoother than the curves without pre-training. This also reveals that pre-training offline can improve the online learning efficiency even though the driving condition is different.

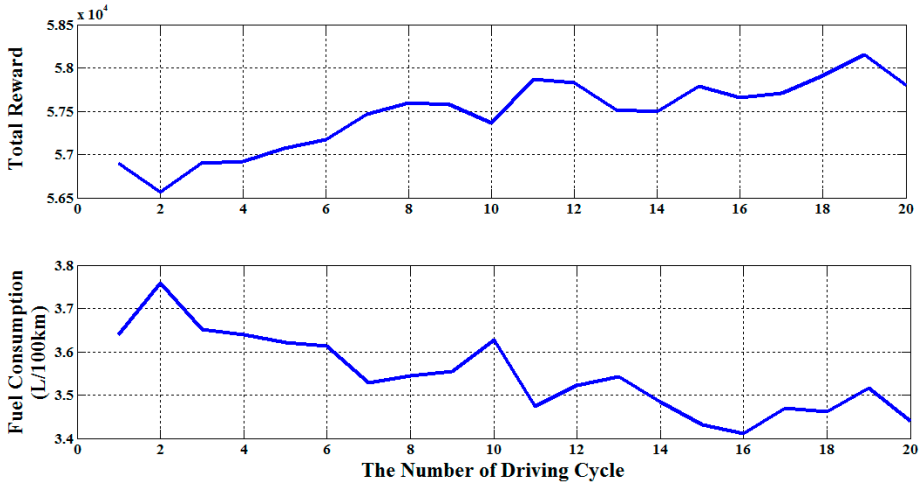

Figure 12. Track of the total reward and fuel consumption in the online application in which the EMS was pre-trained firstly.

5. Conclusions

This paper presents a deep reinforcement learning-based data-driven approach to obtain an energy management strategy of a HEV. The proposed method combines Q learning and a deep neural network to form a deep Q network which can obtain action directly from the states. Key concepts of the DRL-based EMS have been formulated. Value function approximation and DRL algorithm design have been described in detail in this paper. In order to adapt to varying driving cycles, a DRL-based online learning architecture has been presented. Simulation results demonstrate that the DRL-based EMS can obtain better performance than the rule-based EMS in fuel economy. Furthermore, the online learning approach can learn from different driving conditions. The future work will focus on how to improve the online learning efficiency and testing on a real vehicle. Another important issue is how to output continuous actions. In this paper, the output actions are discretized and this may leads to the violent oscillation of the ICE output torque. A deep deterministic policy gradient (DDPG) algorithm can output the continuous actions and may solve this problem. This will be a future work. However, DDPG is also based on DRL. The contribution of this paper will speed up the application of deep reinforcement learning methods in energy management of HEVs.

Acknowledgments: This research was supported by National Natural Science Foundation of China (61573337), National Natural Science Foundation of China (61603377), National Natural Science Foundation of China (61273139), and the Technical Research Foundation of Shenzhen (JSGG20141020103523742).

Author Contributions: Yue Hu and Kun Xu proposed the method and wrote main part of this paper; Weimin Li supported this article financially and checked the paper; Taimoor Zahid checked the grammar of this paper; and Feiyan Qin and Chenming Li analyzed the data.

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

References

1. Un-Noor, F.; Padmanaban, S.; Mihet-Popa, L.; Mollah, M.N.; Hossain, E. A Comprehensive Study of Key Electric Vehicle (EV) Components, Technologies, Challenges, Impacts, and Future Direction of Development. *Energies* **2017**, *10*, 1217. [CrossRef]

2. Wirasingha, S.G.; Emadi, A. Classification and review of control strategies for plug-in hybrid electric vehicles. *IEEE Trans. Veh. Technol.* **2011**, *60*, 111–122. [CrossRef]
3. Gokce, K.; Ozdemir, A. An instantaneous optimization strategy based on efficiency maps for internal combustion engine/battery hybrid vehicles. *Energy Convers. Manag.* **2014**, *81*, 255–269. [CrossRef]
4. Odeim, F.; Roes, J.; Wulbeck, L.; Heinzel, A. Power management optimization of fuel cell/battery hybrid vehicles with experimental validation. *J. Power Sources* **2014**, *252*, 333–343. [CrossRef]
5. Li, C.; Liu, G. Optimal fuzzy power control and management of fuel cell/battery hybrid vehicles. *J. Power Sources* **2009**, *192*, 525–533. [CrossRef]
6. Denis, N.; Dubois, M.R.; Desrochers, A. Fuzzy-based blended control for the energy management of a parallel plug-in hybrid electric vehicle. *IET Intell. Transp. Syst.* **2015**, *9*, 30–37. [CrossRef]
7. Ansarey, M.; Panahi, M.S.; Ziarati, H.; Mahjoob, M. Optimal energy management in a dual-storage fuel-cell hybrid vehicle using multi-dimensional dynamic programming. *J. Power Sources* **2014**, *250*, 359–371. [CrossRef]
8. Vagg, C.; Akehurst, S.; Brace, C.J.; Ash, L. Stochastic dynamic programming in the real-world control of hybrid electric vehicle. *IEEE Trans. Contr. Syst. Techol.* **2016**, *24*, 853–866. [CrossRef]
9. Qin, F.; Xu, G.; Hu, Y.; Xu, K.; Li, W. Stochastic optimal control of parallel hybrid electric vehicles. *Energies* **2017**, *10*, 214. [CrossRef]
10. Chen, Z.; Mi, C.C.; Xiong, R.; Xu, J.; You, C. Energy management of a power-split plug-in hybrid electric vehicle based on genetic algorithm and quadratic programming. *J. Power Sources* **2014**, *248*, 416–426. [CrossRef]
11. Xie, S.; Li, H.; Xin, Z.; Liu, T.; Wei, L. A pontryagin minimum principle-based adaptive equivalent consumption minimum strategy for a plug-in hybrid electric bus on a fixed route. *Energies* **2017**, *10*, 1379. [CrossRef]
12. Guo, L.; Guo, B.; Guo, Y.; Chen, H. Optimal energy management for HEVs in Eco-driving applications using bi-level MPC. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2153–2162. [CrossRef]
13. Tian, H.; Li, S.E.; Wang, X.; Huang, Y.; Tian, G. Data-driven hierarchical control for online energy management of plug-in hybrid electric city bus. *Energy* **2018**, *142*, 55–67. [CrossRef]
14. Daming, Z.; AI-Durra, A.; Fei, G.; Simoes, M.G. Online energy management strategy of fuel cell hybrid electric vehicles based on data fusion approach. *J. Power Sources* **2017**, *366*, 278–291.
15. Martinez, C.M.; Hu, X.; Cao, D.; Velenis, E.; Gao, B.; Wellers, M. Energy Management in Plug-in Hybrid Electric Vehicles: Recent Progress and a Connected Vehicles Perspective. *IEEE Trans. Veh. Technol.* **2017**, *66*, 4534–4549. [CrossRef]
16. Zheng, C.; Xu, G.; Xu, K.; Pan, Z.; Liang, Q. An energy management approach of hybrid vehicles using traffic preview information for energy saving. *Energy Convers. Manag.* **2015**, *105*, 462–470. [CrossRef]
17. Liu, T.; Zou, Y.; Liu, D.; Sun, F. Reinforcement learning-based energy management strategy for a hybrid electric tracked vehicle. *Energies* **2015**, *8*, 7243–7260. [CrossRef]
18. Qi, X.; Wu, G.; Boriboonsomsin, K.; Barth, M.J.; Gonder, J. Data-driven reinforcement learning-based real-time energy management system for plug-in hybrid electric vehicles. *Transp. Res. Rec.* **2016**, *2572*, 1–8. [CrossRef]
19. Li, W.; Xu, G.; Xu, Y. Online learning control for hybrid electric vehicle. *Chin. J. Mech. Eng.* **2012**, *25*, 98–106. [CrossRef]
20. Hu, Y.; Li, W.; Xu, H.; Xu, G. An online learning control strategy for hybrid electric vehicle based on fuzzy Q-learning. *Energies* **2015**, *8*, 11167–11186. [CrossRef]
21. Volodymyr, M.; Kavukcuoglu, K.; Silver, D.; Rusu, A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533.
22. Silver, D.; Aja, H.; Maddison, J.C.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, L.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef] [PubMed]
23. Springenberg, T.; Boedecker, J.; Riedmiller, M.; Obermayer, K. Autonomous learning of state representations for control. *Künstliche Intell.* **2015**, *29*, 1–10.
24. Wei, T.; Wang, Y.; Zhu, Q. Deep reinforcement learning for building HVAC control. In Proceedings of the 54th Annual Design Automation Conference, Austin, TX, USA, 18–22 June 2017.

25. Belletti, F.; Haziza, D.; Gomes, G.; Bayen, A.M. Expert level control of ramp metering based on multi-task deep reinforcement learning. *IEEE Trans. Intell. Transp. Syst.* **2017**, *99*, 1–10. [CrossRef]
26. Sallab, A.E.; Abdou, M.; Perot, E.; Yogamani, S. End-to-End deep reinforcement learning for lane keeping assist. *arXiv* **2016**.
27. Chae, H.; Kang, C.M.; Kim, B.D.; Kim, J.; Chung, C.C.; Choi, J.W. Autonomous Braking System via Deep Reinforcement Learning. *arXiv* **2017**.
28. Tianho, Z.; Kanh, G.; Levine, S.; Abbeel, P. Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 528–535.
29. Qi, X.; Luo, Y.; Wu, G.; Boriboonsomsin, K.; Brath, M.J. Deep reinforcement learning-based vehicle energy efficiency autonomous learning system. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Redondo, Beach, CA, USA, 11–14 June 2017.