energies

MDPI

Article

Control of Hybrid Electric Vehicle Powertrain Using Offline-Online Hybrid Reinforcement Learning

Zhengyu Yao [1], Hwan-Sik Yoon [1,*] and Yang-Ki Hong [2]

[1] Department of Mechanical Engineering, The University of Alabama, Tuscaloosa, AL 35487, USA
[2] Department of Electrical and Computer Engineering, The University of Alabama, Tuscaloosa, AL 35487, USA
* Correspondence: hyoon@eng.ua.edu

Abstract: Hybrid electric vehicles can achieve better fuel economy than conventional vehicles by utilizing multiple power sources. While these power sources have been controlled by rule-based or optimization-based control algorithms, recent studies have shown that machine learning-based control algorithms such as online Deep Reinforcement Learning (DRL) can effectively control the power sources as well. However, the optimization and training processes for the online DRL-based powertrain control strategy can be very time and resource intensive. In this paper, a new offline–online hybrid DRL strategy is presented where offline vehicle data are exploited to build an initial model and an online learning algorithm explores a new control policy to further improve the fuel economy. In this manner, it is expected that the agent can learn an environment consisting of the vehicle dynamics in a given driving condition more quickly compared to the online algorithms, which learn the optimal control policy by interacting with the vehicle model from zero initial knowledge. By incorporating a priori offline knowledge, the simulation results show that the proposed approach not only accelerates the learning process and makes the learning process more stable, but also leads to a better fuel economy compared to online only learning algorithms.

Keywords: hybrid electric vehicle; reinforcement learning; powertrain control

Citation: Yao, Z.; Yoon, H.-S.; Hong, Y.-K. Control of Hybrid Electric Vehicle Powertrain Using Offline-Online Hybrid Reinforcement Learning. *Energies* **2023**, *16*, 652. https://doi.org/10.3390/en16020652

Academic Editors: Islam Safak Bayram and Stefania Santini

Received: 25 October 2022
Revised: 23 November 2022
Accepted: 24 December 2022
Published: 5 January 2023

1. Introduction

Due to environmental and economic considerations, the importance of energy-efficient vehicles continues to grow. One approach to improving vehicle energy efficiency is developing energy-efficient Hybrid Electrical Vehicles (HEVs). HEVs combine the benefits of a combustion engine and electric motor(s) to provide an effective method for the overall improvement of energy efficiency [1]. For HEVs to operate optimally for energy efficiency, the amount of power drawn from each power source, known as the power split, must be optimally controlled in real time. A systematic approach to this task is known as power flow control [2]. In general, approaches to power flow control can be roughly classified into three categories: rule-based methods, optimization-based methods and Deep Reinforcement Learning (DRL)-based methods.

Among the three general approaches, two types of power flow control are considered in this paper: optimization-based and DRL-based. Optimization-based controllers derive an optimal solution by minimizing a cost or objective function with a given set of constraints. They can be further divided into global and real-time controllers. Several real-time methods have been applied to solving the optimization problem, including real-time Dynamic Programming (DP) [3], the Adaptive Equivalent Consumption Minimization Strategy (A-ECMS) [4], and Model predictive control (MPC) [5]. These methods make instantaneous power handling decisions to minimize the cost function based on the equivalence assumptions for energy consumption [6]. Although these algorithms can determine the optimal power split for a given driving cycle, they require either a priori knowledge of the driving cycle or high computation power, which prohibits their wide adoption in real-time applications.

Recently, it has been shown that reinforcement learning (RL) can be applied to the HEV power flow control by taking optimal control actions to maximize a vehicle's fuel economy without a priori driving route information, making it desirable for real-time control [7]. In RL, an agent learns an optimal policy that returns the maximum accumulated rewards from a series of actions that the agent takes at each time step [8,9]. For HEV power flow controls, Pu et al. proposed a HEV power management framework based on DRL for optimizing the fuel economy [10]. The DRL technique comprises an offline deep neural network construction phase and an online Deep Q-learning phase. The research showed that it is possible to handle high-dimensional state and action spaces in a decision-making process. These methods, however, require the continuous state as well as the control actions for the vehicle powertrain to be discretized. Wu et al. [11] applied the Deep Deterministic Policy Gradient (DDPG) algorithm [12] to the energy management of a Plug-in Hybrid Electric Bus (PHEB). The simulation results showed that the proposed approach outperformed the One Q-learning approach over multiple driving cycles, with performance close to that of dynamic programming, which is globally optimal. Roman et al. implemented the DDPG on a mild HEV and achieved a nearly optimal fuel consumption result using a locally trained strategy [13]. These policy-based methods update the policy directly without relying on the value estimations. Yao et al. [14] applied a new approach for controlling a mild HEV using a relatively new reinforcement learning algorithm called Twin Delayed DDPG (TD3) [15] to maximize the fuel economy. TD3 is an extension of the DDPG algorithm with the capability to prevent the overestimation of the value function and further improve the performance. Despite all of their achievements and promise, these DRL-based methods all suffer from a significant drawback, which is their time and resource intensive training process.

The aforementioned online model-free DRL methods rely on a large data set sampled from the environment for improved performance, which often suffers from low sampling efficiency [16]. In many cases, human experience can provide reasonably good training samples or preferences to guide the learning agent in exploring the environment during the training process [17]. Lian et al. applied a rule-interposing DRL-based energy management strategy (EMS) to a Prius model [18]. With the inclusion of expert knowledge such as the Brake-Specific Fuel Consumption (BSFC) map within the DRL-based EMS, the engine can operate along its optimal BSFC curve for improved fuel economy. Liu et al. applied a DP-based optimal control policy to a DDPG framework as expert knowledge to guide the DDPG-based EMS [19]. The simulation results showed that its control performance is better than that of the conventional DDPG, and even closer to that of the DP-based EMS. However, a priori knowledge of the vehicle model is required during the training process, and DP requires a lot of memory to store the calculated result of every subproblem without any assurance as to whether the stored value will be utilized or not.

Another approach has been to combine DRL with Transfer Learning (TL) [20]. Guo et al. proposed an adaptive EMS for a hybrid electric tracked vehicle (HETV) by combining DDPG and TL [21]. In their approach, the vehicle velocity is divided into three-speed intervals to train the DDPG algorithm separately until the algorithm converges. Then, the TL method is employed to transfer the pretrained neural network to a new driving cycle. In this way, an optimal control strategy can be quickly obtained for a new driving cycle. Since one of the biggest limitations of transfer learning is the problem of negative transfer, however, TL works only if the initial and target problems are sufficiently similar to make the first round of training relevant.

In order to overcome these inherent difficulties in the existing DRL-based methods, a new offline and online hybrid DRL strategy has been developed and is presented in this paper. The strategy exploits offline vehicle data as well as exploring a new control policy online to further improve the fuel economy. To test this idea, offline vehicle dynamics data are generated from a HEV simulation model operated by an optimization-based control algorithm, and the obtained data is used to train the policy network. Then, the trained network is embedded in an online DDPG-based HEV powertrain control model. In this manner, the agent can learn the environment comprising the vehicle in a driving condition

faster than the online only algorithms that learn the optimal control policy starting from zero initial knowledge by interacting with the vehicle model.

The major contribution of this paper is the development and performance analysis of the new offline and online hybrid DRL strategy, named HDDPG, for HEV energy flow control. Also, the performance of the proposed HDDPG is compared with another state-of-the-art algorithm, DDPG, over various driving cycles including real-world driving cycles to highlight the potential capabilities of the HDDPG.

The remainder of this paper is organized as follows. The second section describes the vehicle simulation model used for the study. The third section introduces the different control algorithms used for power split in the study. Then, a new offline and online hybrid DRL strategy for the HEV powertrain control is explained in the fourth section. Lastly, the simulation results and the conclusion follow.

2. HEV System Model

2.1. Powertrain Architecture

The vehicle model used for this research is a modified version of the full HEV P4 reference model that comes standard as part of the Vehicle Dynamics Block set in MATLAB/Simulink. The general vehicle architecture is shown in Figure 1. This P4 HEV uses a parallel through the ground (PTTG) architecture [22], in which the internal combustion engine (ICE) and electric motor provide propulsion power to the front and rear axles, respectively. The two propulsion sources are not mechanically coupled. The power sources are instead coupled through the road. The electric motor is electrically connected to the battery, and power can flow bidirectionally between the two, with the direction being dependent on which operation mode is active. Although the PTTG architecture is not as popular as other options, some of the production HEVs, including the Volvo V60 PHEV, Peugeot 3008, and BMW 2 Series, have adopted this architecture. Due to the convenience of studying HEV control using a vehicle model in a simulation environment, this HEV model replaces an actual physical vehicle in both generating vehicle data for the offline training of a network and online reinforcement learning using the trained network model. Five different operation modes can be realized for this PTTG HEV architecture, as summarized in Table 1.

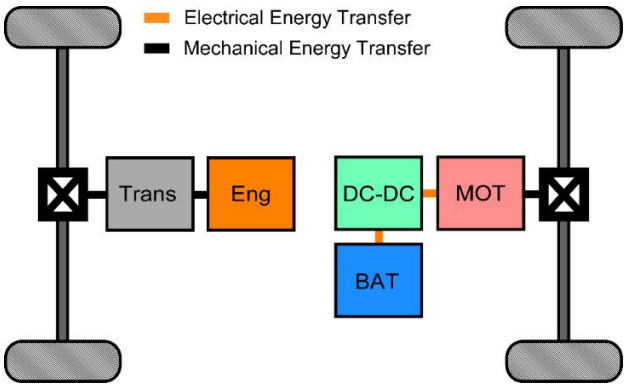

Figure 1. Full HEV powertrain architecture.

Table 1. Operation modes of full HEV.

Mode A	Tractive power is produced by IC engine only
Mode B	Tractive power is produced by electric motor only while IC engine idles
Mode C	Tractive power is produced by both IC engine and electric motor
Mode D	Regenerative power is produced by electric motor while vehicle is braking
Mode E	IC engine produces both tractive power and electric power through the generator (motor) to charge the battery

2.2. Longitudinal Vehicle Dynamics

The longitudinal vehicle dynamics can be represented by Equation (1). The wheel tractive force, F_w, enables the vehicle to accelerate or decelerate at a rate $\dot{v}$ depending on its sign. The vehicle's mass m is simplified as a point mass at the vehicle's center of gravity (CG). The magnitude of the gravitational force, F_g, acting through the vehicle CG is modified by the road grade, β. The resistive force, F_r, is a combination of rolling friction and aerodynamic drag and is resolved into a parallel and a perpendicular component to the road, both acting through the vehicle CG.

$$F_w - F_g - F_r = m\dot{v} \tag{1}$$

Since the focus of this research is to introduce a new offline and online hybrid DRL-based HEV powertrain control strategy, the model fidelity is not of primary importance as long as the same model is used for each of the different control algorithms for comparison. Additionally, the effects of the wind speed and road grade are not considered in this study, so both are assumed to be zero. Some of the important environmental parameters and vehicle specifications are presented in Table 2**.**

Table 2. Vehicle dynamics parameters.

Vehicle mass (m)	1623 kg
Vehicle frontal area	2.46 m^2
Aerodynamic drag coefficient	0.25
Rolling resistance coefficient	0.01
Tire radius (R)	0.327 m

2.3. Engine and Transmission Models

The HEV uses a Spark Ignition (SI) engine with a 6-speed transmission, both of which are modeled using lookup tables. The engine and transmission specifications are summarized in Table 3, and a Brake-Specific Fuel Consumption (BSFC) map representing the engine efficiency is shown in Figure 2.

2.4. Electric Motor and Battery Models

Similar to the engine, the P4 motor is modeled using a lookup table. The battery pack is modeled based on a lithium-ion battery model that is a part of the HEV P4 reference model in Simulink. Table 4 shows detailed specifications for the motor and battery pack models. Also, the motor efficiency map is shown in Figure 3, and the open circuit voltage curve for a single battery cell is shown in Figure 4.

Table 3. Full HEV powertrain specifications.

Engine displacement	1.5 L
Number of cylinders	4
Maximum power	92.9 kW
Maximum speed	5068.6 RPM
Maximum torque @ RPM	182.0 N·m @ 1964.3 RPM
Engine idle speed	750 RPM
Number of transmission gears	6
Transmission gear ratio	4.212/2.637/1.8/1.386/1/0.772

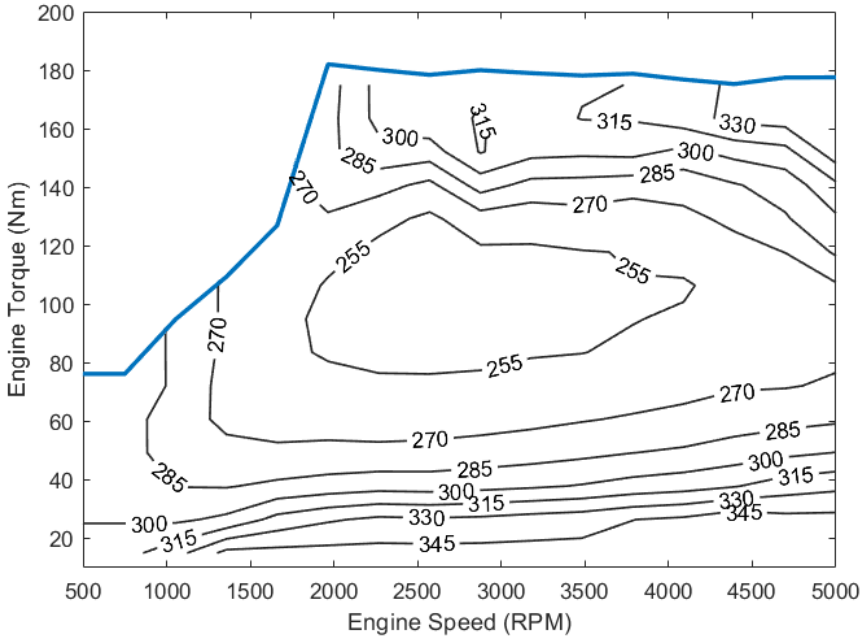

Figure 2. SI engine BSFC map (g/kWh).

Table 4. Electric motor and battery pack specifications.

Motor power	30 kW
Maximum torque @ RPM	200 N·m @ (0~1432.4 RPM)
Battery capacity	5.3 Ah
Battery cell voltage	2.8~4.2 V
Number of cells in series	72
Number of cells in parallel	1
Initial battery SOC	60%

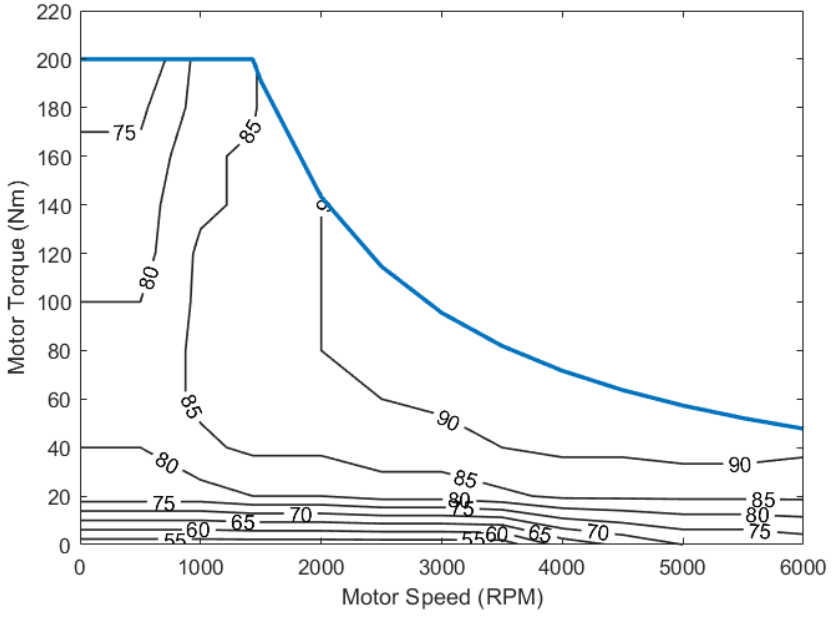

Figure 3. P4 motor efficiency map (%).

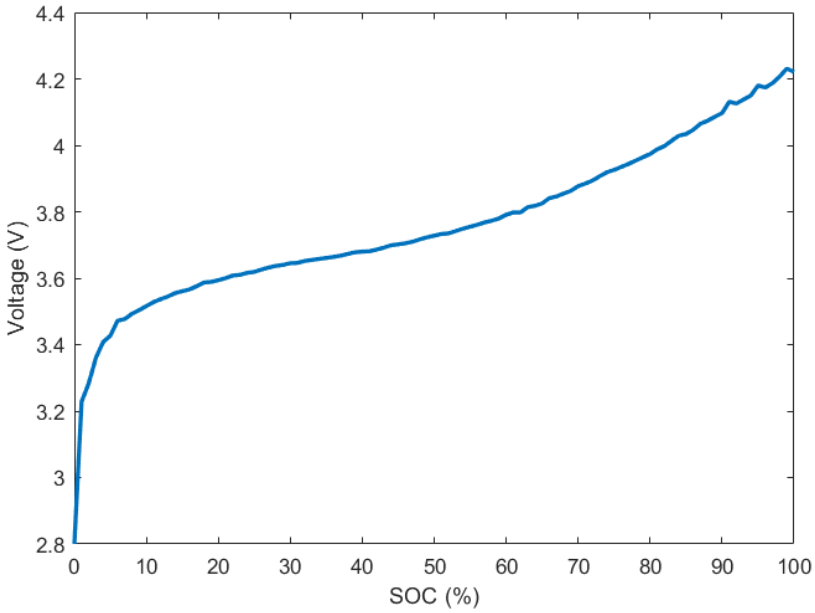

Figure 4. Open circuit voltage of a single battery cell.

The state of charge (SOC) of the battery pack is calculated using the relationship shown in Equation (2).

$$S(t) = S(t)|_{t=0} + \left(\frac{1}{C_b}\right) \int Idt \times 100 \tag{2}$$

where

$S(t)$ = Battery SOC at time t (%)
I = Charge(+)/discharge(−) current (A)
C_b = Battery capacity (Ah)

3. Power Flow Control Algorithms

This section provides a brief introduction to a DRL-based control algorithm and an optimization-based control algorithm, both used in this study. In the vehicle simulation model, a longitudinal driver block is used to control longitudinal speed tracking. The driver block generates normalized accelerator and brake pedal position commands represented by $APP \in [-1,\ 1]$, based on the reference and feedback vehicle speeds. Then, the wheel torque command, T_{cmd}, is calculated by multiplying the normalized accelerator pedal position by the maximum torque available at the current RPM with speed reduction ratios as shown in Equation (3).

$$T_{cmd} = APP \cdot \left(D \cdot T_{eng,max} + N \cdot T_{mot,max}\right), \tag{3}$$

where

$T_{eng,max}$ = Maximum engine torque (N·m) at a given RPM,
$T_{mot,max}$ = Maximum motor torque (N·m) at a given RPM,
D = Overall speed reduction ratio from the transmission to the wheel,
N = Speed reduction ratio between the motor and wheel.

The core problem of the HEV power flow control is determining the optimal torque split between the two power sources. In this research, both control approaches are designed to generate the electric motor torque output as a control action, A_t. Detailed vehicle operating conditions and torque splits are summarized for each vehicle operation mode in Table 5.

Table 5. Torque split and operation modes based on action value.

Mode	Conditions	Torque Split
A	$T_{cmd} > 0$ $A_t = 0$	$T_{eng} = T_{cmd}/D$ $T_{mot} = 0$
B	$T_{cmd} > 0$ $A_t = T_{cmd}/N$	$T_{eng} = 0$ $T_{mot} = A_t$
C	$T_{cmd} > 0$ $0 < At < T_{cmd}/N$	$T_{eng} = (T_{cmd} - N{\cdot}T_{mot})/D$ $T_{mot} = A_t$
D	$T_{cmd} < 0$ $A_t < 0$	$T_{eng} = 0$ $T_{mot} = A_t$
E	$T_{cmd} > 0$ $A_t < 0$	$T_{eng} = (T_{cmd} - N{\cdot}T_{mot})/D$ $T_{mot} = A_t$

In Table 5, T_{eng} and T_{mot} represent the torque commands (N·m) to the engine and electrical motor, respectively. During regenerative braking (Mode D), the vehicle's kinetic energy is converted to electrical energy by the motor in order to charge the battery following a simple rule-based algorithm. When $T_{cmd} > 0$, the vehicle can operate in Modes A, B, C and E, as determined by the powertrain control algorithms.

3.1. Optimization-Based Control

The Equivalent Consumption Minimization Strategy (ECMS) is a well-known optimization-based power flow control method for HEVs. As its name suggests, the goal of ECMS is to minimize the equivalent fuel consumption, which includes both the actual amount of consumed fuel and equivalent amount of consumed electrical energy. The instantaneous equivalent fuel consumption is calculated as shown in Equation (4).

$$\dot{m}_{eq}(t) = \dot{m}_f(t) + \frac{s(t)}{Q_{Lhv}} \cdot P_{batt}(t) \cdot p(SOC), \tag{4}$$

where

$\dot{m}_{eq}(t)$ = instantaneous equivalent fuel consumption rate
$\dot{m}_f(t)$ = fuel mass flow rate
Q_{Lhv} = fuel lower heating value
$P_{batt}(t)$ = instantaneous battery power
$s(t)$ = equivalence factor
$p(SOC)$ = penalty function on the usage of out-of-range SOC

In ECMS, $\dot{m}_{eq}(t)$ is calculated at each time step using several different candidate values for the control variable, $P_{batt}(t)$, and the value that produces the lowest value of $\dot{m}_{eq}(t)$ is selected as the control input. The equivalence factor is used to account for the theoretical process efficiency for the conversion of electrical power into fuel power to maintain consistency of units. Because the efficiency of the process is dependent on the direction of the power flow, $s(t)$ is a vector of two values, one for charge and one for discharge. Thus, the equivalence factor can be represented as $s(t) = \left[s_{chg}(t),\ s_{dis}(t)\right]$ [4].

The Adaptive Equivalent Consumption Minimization Strategy (A-ECMS) performs similarly to the ECMS, except that a proportional integral (PI) controller is used to update $s(t)$ based on the SOC feedback. Using the SOC feedback allows A-ECMS to not rely on a priori knowledge of the drive cycle. As a result, A-ECMS generally has slightly lower performance than ECMS on fully known drive cycles but performs better in real-world applications, where drive cycles are not typically known in advance.

3.2. DRL-Based Control

Reinforcement learning is one of the machine learning approaches, specialized in optimally solving a Markov Decision Process (MDP) [8,9]. An MDP consists of five components

such as $M = (S, A, R, P, \gamma)$, where S is a set of states, A is a set of actions, R is the reward function, P is the state transition probability and γ is the discount factor for future rewards. In reinforcement learning, the learning agent takes action A_t in state S_t at each time t according to a policy π. After an action has been taken, the environment returns the next state S_{t+1} with a reward R_{t+1} as feedback. This process is repeated by the agent to update its policy in order to maximize the expected return E_π, where $E_\pi = \Sigma_{k=0}^{\infty}(\gamma^k \cdot R_{t+k+1})$. The States, Action and Reward functions selected for this research are shown in Table 6.

Table 6. States, Action and Reward functions.

States	S_t = (Wheel Torque Command, Battery SOC, Battery Voltage, Transmission Gear Number, Motor Speed, Vehicle Speed)
Action	Motor Torque Command: $A_t \in [-50,\ 200]$
Reward	$R_t = w_i + w_j \cdot \lvert SOC_t - SOC_{init} \rvert - w_k \cdot FC_t$

The selected set of states in reinforcement learning should be representative of the current state of the vehicle and closely related to its efficiency characteristics. For this study, a state vector S_t containing the five state variables is used to approximate the Markov property, and the DRL agent is designed to generate the Motor Torque Command as an action.

The reward function shows how the algorithm's objective is explicitly defined. In the case of power split control on an HEV, the objective is set to minimize the fuel consumption while keeping the battery SOC within a specified range. In this research, the vehicle is intended to operate in a charge sustaining mode with the SOC range defined as ±10% from the initial SOC. In the Reward function shown in Table 1, w_i, w_j and w_k are weighting factors where w_i, w_k are positive values and w_j changes its sign depending on whether the deviation of the current SOC, SOC_t, from the initial SOC, SOC_{init}, is greater than the predefined range or not. FC_t is defined as

$$FC_t = F_{flow} + CP_{batt} \tag{5}$$

where

F_{flow} = fuel mass flow rate
P_{batt} = battery power
C = battery-to-fuel equivalence factor

In Equation (5), C is the equivalence factor that is used to convert the battery power to the equivalent fuel consumption rate. The Environmental Protection Agency (EPA) uses the standard conversion of 33.7 kilowatt-hours (121 megajoules) of electrical energy to one gallon of gasoline [23]. The properties of the gasoline are based on the California Reformulated Gasoline (CaRFG2) standards.

Different approaches exist for finding the optimal policy π^* in reinforcement learning. The Deep Deterministic Policy Gradient (DDPG) is a model-free off-policy policy gradient algorithm [12]. It is a combination of two other reinforcement learning algorithms, Deterministic Policy Gradient (DPG) and Deep Q-Network (DQN) [24,25]. DQN uses experience replay and a frozen target network to stabilize the learning of the Q-function. DDPG extends the normally discrete DQN to a continuous space for learning a deterministic policy. The learning algorithm is represented by an actor-critic framework. The policy function, known as the actor, takes the given state as the input and produces the best possible action. The value function, known as the critic, takes the action and environments as inputs and outputs the reward value. In DDPG, the parameters for both the actor and the critic receive soft updates at every time step using a smoothing factor τ, such that a target weighting parameter θ' is updated according to $\theta' = \tau \cdot \theta + (1 - \tau) \cdot \theta'$, where θ is a recent weighting

parameter. This constraint forces the target network weights to change slowly, as opposed to the performance of DQN, where the target network is frozen for some period.

4. Offline and Online Hybrid DRL-Based HEV Powertrain Control Strategy

In this research, a new offline and online hybrid DRL strategy based on DDPG has been developed to exploit offline vehicle data as well as exploring a new control policy to further improve the fuel economy. For the training of the offline–online Hybrid DDPG (HDDPG), offline vehicle dynamics data are generated from a PTTG HEV model whose powertrain is controlled by a built-in A-ECMS algorithm in the reference HEV model in MATLAB/Simulink. The generated vehicle operation data are used to train the actor target Neural Network (NN) in HDDPG. Then, the trained actor network is embedded in the HDDPG for online training of the powertrain control algorithm for the same PTTG HEV simulation model. The basic premises for using the same HEV model is that, in actual vehicle development, the vehicle dynamics data will be collected using a basic control algorithm and the actor target NN will be offline trained using the collected data. Then, the fuel economy of the same vehicle will be further improved via online training of the HDDPG model. Therefore, it is required to use the same vehicle model in this scenario. Since the DRL-based algorithm can learn how to control an unknown system, practically speaking, a small mismatch in the vehicle models for the offline and online training processes would not be an issue.

4.1. Offline Training

Based on heuristic considerations and performance evaluations, the NN is designed in this research such that the actor network has six fully connected hidden layers, with all hidden layers having 300 nodes each. The architecture of the actor network is shown in Figure 5. It should be noted that the reward values depend highly on the critic target network. If the critic target network is not well trained, it will limit the agent's capability to explore an optimal policy. For this reason, the offline data are used only for training the actor target network of the HDDPG agent.

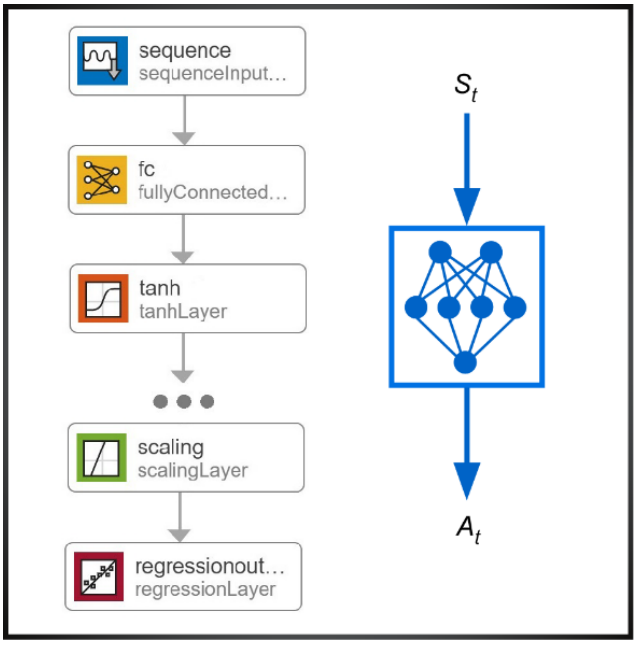

Figure 5. Architecture of the Actor network.

For the generation of offline data, multiple drive cycles with different speed patterns and lengths are selected to simulate the HEV model with the A-ECMS controller in MATLAB/Simulink. Based on the predefined states and actions, a total 10,513 samples were

collected and used to train the actor target network. In Figure 5, it is shown that the actor network uses the state vector as an input and produces an action as an output. The training progress for the actor network is shown with the network output error trend in Figure 6, and some of the key values during the training process are presented in Table 7.

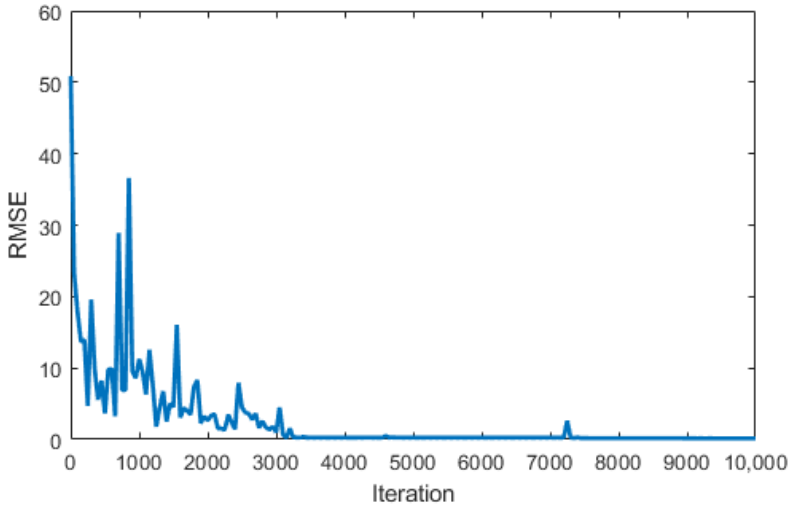

Figure 6. Actor network error trend during iterative offline training process.

Table 7. Key values during actor network training process.

Iteration	Time Elapsed (s)	Mini-Batch RMSE	Mini-Batch Loss
1	1	50.88	1294.4
1000	70	11.25	63.3
2500	167	4.50	10.1
5000	323	0.26	3.3×10^{-2}
10,000	642	0.16	1.3×10^{-2}

It can be seen from Figure 6 and Table 7 that the root-mean-square error (RMSE) between the network output and the target value converges after 3000 iterations. When the network error converges, the RMSE becomes as small as 0.16, which means that the trained network is able to replace the A-ECMS algorithm and can achieve a similar fuel economy. However, the DRL algorithm can further be optimized through online training in new driving conditions.

4.2. Online Training

The proposed HDDPG is based on DDPG, which is a combination of DPG and DQN. It uses an experience replay memory to store past transitions and uses target networks to stabilize learning. In this research, it is assumed that the offline A-ECMS controller can provide optimal training samples for the learning agent to effectively explore the environment at the beginning of the online training process. By inserting a well-trained target network in the DDPG framework, the exploration space will be narrowed, and thus the related computational efficiency and performance will be improved. The overall DDPG architecture utilized in this research for the HEV control is shown in Figure 7.

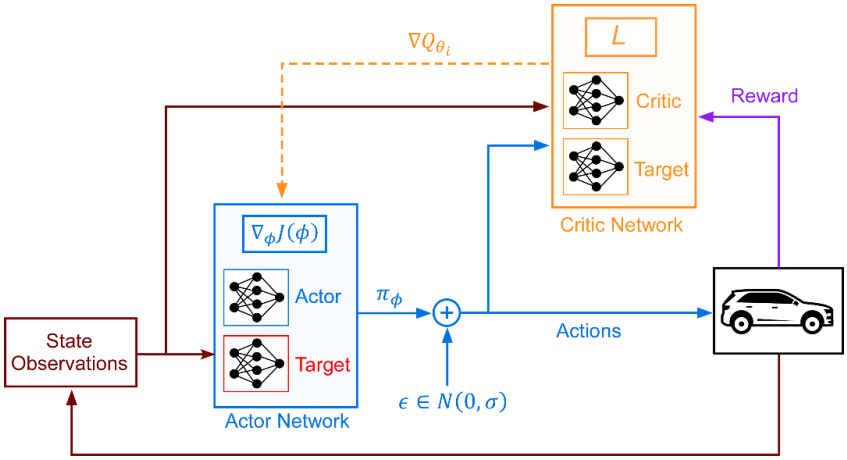

Figure 7. DDPG actor-critic architecture for the full HEV control.

DDPG uses multilayer NNs to learn in large state and action spaces. The actor network generates an action (A_t) based on the current state input (S_t) according to the current policy. A difference from the original DDPG algorithm is that the action is determined by the offline-trained target network with added exploration noise in this HDDPG algorithm. The critic network calculates the Q-Value based on the current state and the action chosen by the actor network. Finally, the untrained critic target network and pretrained actor target network are updated by using a sliding average. In this manner, the agent can effectively explore the environment at the beginning of the online training process. A pseudo-algorithm for the HDDPG is shown in Algorithm 1.

Algorithm 1. Computational procedure of the HDDPG algorithm with offline-trained target networks [12].

1. Initialize critic network, Q_θ and actor network, π_ϕ with θ, ϕ.
2. Initialize critic target network, $\theta'$and load initial actor target network ϕ', which is pretrained with offline data.
3. Initialize replay buffer, $\mathcal{B}$.
4. **for** episode = 1 **to** E **do**
5. Initialize local buffer $\mathcal{L}$.
6. Initialize S_t.
7. **for** t = 1 **to** T **do**
8. Select action $A_t = \pi(s) + \epsilon$ with exploration noise $\epsilon \sim N(0,\sigma)$ according to the current target policy.
9. Apply action A_t to the model. Observe reward R_t and next state $S\prime_t$.
10. Store the experience $(S_t, A_t, R_t, S\prime_t)$ in $\mathcal{B}$ and $\mathcal{L}$.
11. Sample a random mini batch of M experiences $(S_j, A_j, R_j, S\prime_j)$ from $\mathcal{B}$.
12. **if** $S\prime_j$ is terminal, **then** set $y_j = R_j$
13. **else** set $y_j = R_j + \gamma \cdot Q_{\theta'}\left(S\prime_j, \pi_{\phi'}\left(S\prime_j \middle| \phi'\right) \middle| \theta'\right)$
14. **end if**
15. Update critic θ by minimizing the loss L across all sampled experiences:

$$L = \frac{1}{M}\sum_{j=1}^{M}\left(y_j - Q_\theta(S_j, A_j|\theta)\right)^2$$

16. Update ϕ by the deterministic policy gradient:

$$\nabla_\phi J(\phi) = \frac{1}{M}\sum_{j=1}^{M} \nabla_A Q_\theta\left(S_j, A\right)\Big|_{A=\pi_\phi(s_j|\phi)} \cdot \nabla_\phi \pi_\phi\left(s_j \middle| \phi\right)$$

17. Update target network using smoothing factor, τ:

$$\theta' = \tau\cdot\theta + (1-\tau)\cdot\theta'$$
$$\phi' = \tau\cdot\phi + (1-\tau)\cdot\phi'$$

18. **end for**
19. **end for**

5. Simulation Results and Discussion

In this research, it was hypothesized that the offline–online hybrid DDPG has a higher convergence rate and shows better fuel economy performance than the original DDPG algorithm. When trained with various driving patterns and durations, the HDDPG agent actually showed a better convergence rate than the DDPG agent in most cases. Using the learning parameters shown in Table 8, the HDDPG and DDPG algorithms were trained, and the results were recorded. Example reward change plots over episodes during training are shown in Figures 8 and 9 for HDDPG and DDPG, respectively.

Table 8. Learning parameters used for DDPG and HDDPG.

Critic network learning rate	0.0001
Actor network learning rate	0.0001
Minimum batch size	32
Discount factor	0.95

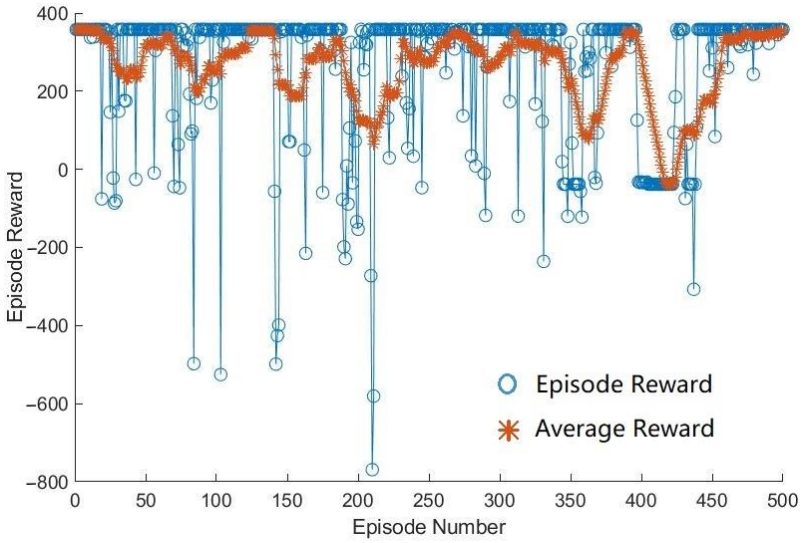

Figure 8. Example reward change of HDDPG over episodes during training.

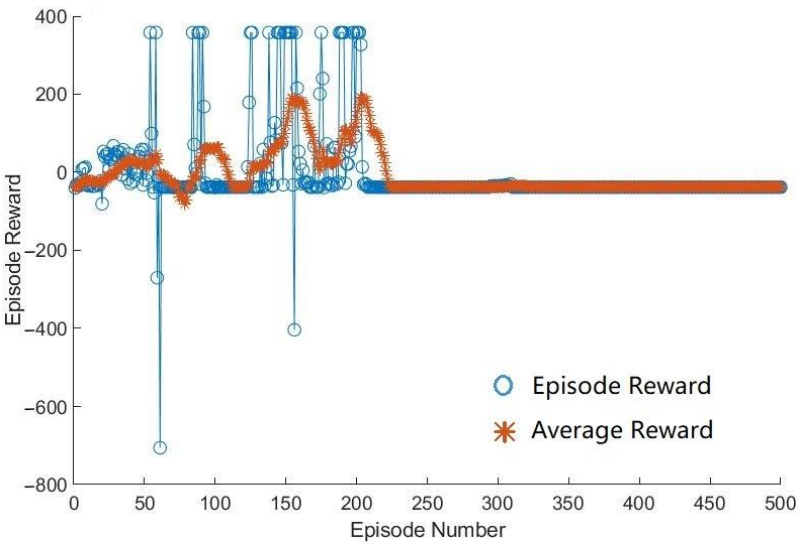

Figure 9. Example reward change of DDPG over episodes during training.

In Figures 8 and 9, it can be seen that the HDDPG agent quickly converges at the beginning to a greater reward value than the DDPG agent. After around 200 episodes, DDPG agent converges to a negative reward value and does not improve any more. When the learning rate of the critic network in the DDPG agent was changed to 0.001, it showed an improved performance in the episode reward as shown in Figure 10. From this, it can be learned that the training performance is highly dependent on the learning rate, and thus it is important to find and use a proper learning rate for each network.

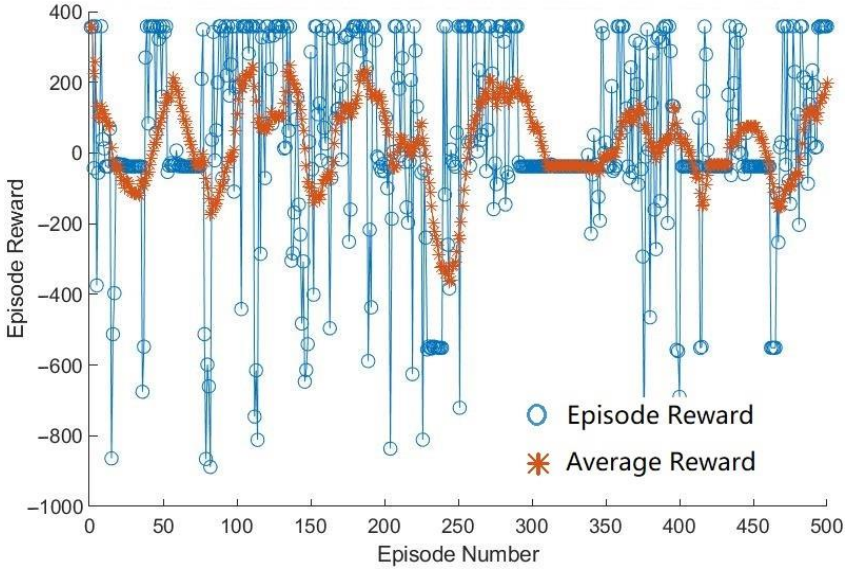

Figure 10. Example reward change of DDPG over episodes during training using a different learning rate.

Although the training results vary depending on the learning parameters used, HDDPG always shows better convergence than DDPG in the simulation. After being trained over different drive cycles, the agents (HDDPG and DDPG) with the best fuel performance were selected and compared with the A-ECMS control algorithm. For the evaluation of the HDDPG algorithm in different scenarios, three standard drive cycles and two real-world drive cycles were selected. The resulting fuel consumptions for the three control algorithms over the selected drive cycles are presented in Table 9. In the table, the percent improvements in the fuel consumption by the HDDPG agent over the other methods are calculated by Equation (6) and presented in the parentheses.

$$Imp\% = \frac{FC_i - FC_{HDDPG}}{FC_i} \times 100 \tag{6}$$

where FC_{HDDPG} is the fuel consumption by the HDDPG agent and FC_i represents the fuel consumption by the other two control algorithms.

In Table 9, UDDS, US06 and HWFET are three commonly used standard drive cycles, where UDDS represents an urban drive pattern, US06 shows the highest average speed and extreme accelerations and HWFET is a typical highway drive cycle featuring high speed and zero stop. Two additional real-world drive cycles, T-Town and T-Highway, are also used to represent distinct driving characteristics near the city of Tuscaloosa, Alabama, where T-Town shows a typical urban driving condition with low average speed and many stops at traffic signals and T-Highway shows the traffic condition on Highway US-82, which include a higher speed and more complicated traffic situations. It can be seen in the table that the HDDPG algorithm shows the best energy performance in all cases and has significant improvements over the DDPG algorithm. When compared with the A-ECMS algorithm, HDDPG also shows improvements in most cases except US06. The engine BSFC maps with the operation points are shown in Figures 11–13 for the three algorithms over the UDDS cycle with sampling at every 5 s.

Table 9. Comparison of fuel consumptions (gal) by three control algorithms over different drive cycles.

Drive Cycles	Fuel Consumptions by Different Algorithms		
	A-ECMS	**DDPG**	**HDDPG**
UDDS	0.2519 (3.02%)	0.2471 (1.13%)	0.2443
US06	0.2821 (−1.81%)	0.2896 (0.83%)	0.2872
HWFET	0.2862 (9.19%)	0.2670 (2.66%)	0.2599
T-Town	0.3730 (0.43%)	0.3716 (0.05%)	0.3714
T-Highway	0.4094 (8.38%)	0.3867 (3.00%)	0.3751

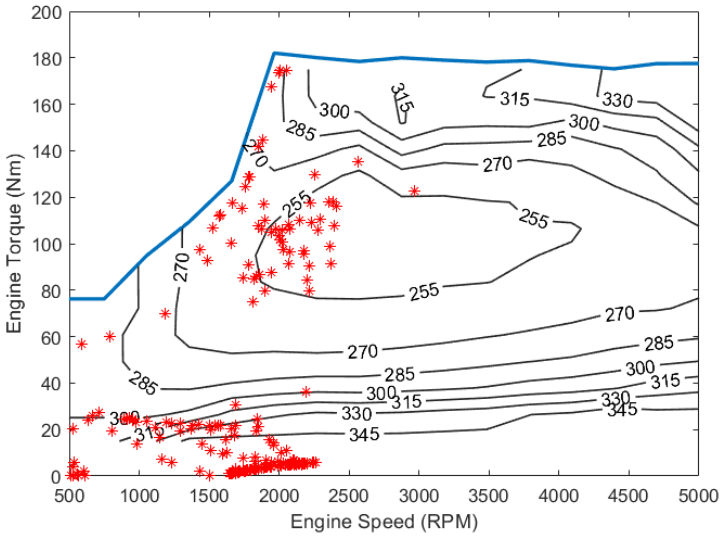

Figure 11. SI engine operation points by A-ECMS over UDDS.

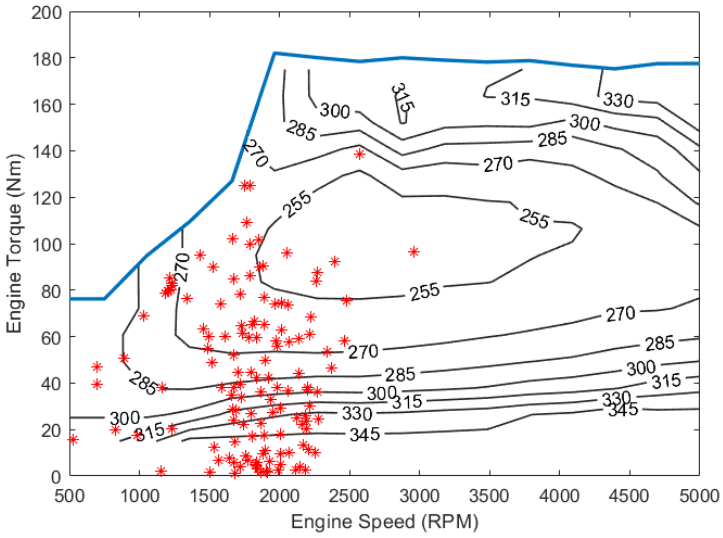

Figure 12. SI engine operation points by HDDPG over UDDS.

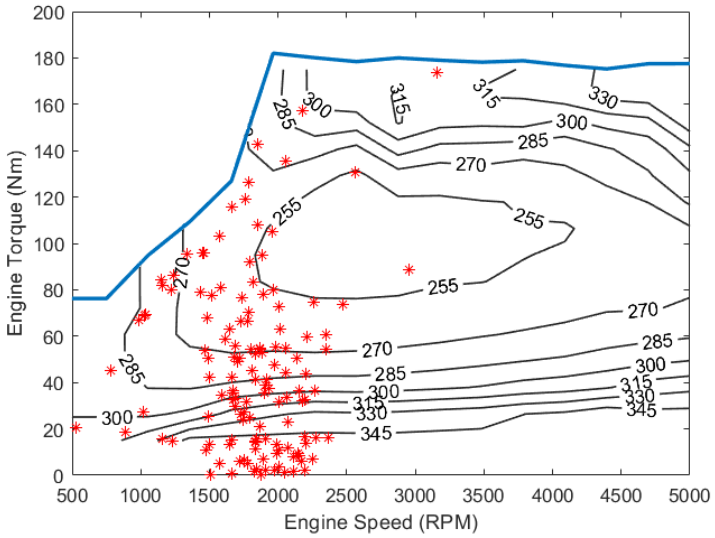

Figure 13. SI engine operation points by DDPG over UDDS.

The engine BSFC map shows the efficiency of the engine at different torque-speed combinations. It can be seen from Figure 11 that A-ECMS controls the engine's operation frequently in the high-efficiency region (255 g/kWh). A-ECMS also operates the engine in the low-efficiency region quite often (345 g/kWh). On the other hand, the two DRL-based algorithms show a more diverse distribution of the engine operation points. From Figures 12 and 13, it can be seen that the engine operates more around the region of 270 g/kWh compared to A-ECMS. This is due to the probabilistic nature of the learning-based control algorithms. HDDPG tries to explore a new control policy to further improve the fuel economy, which makes it behave differently from A-ECMS. Meanwhile, since the HDDPG utilizes offline-trained target networks, the engine works more efficiently when compared with the traditional DDPG algorithm. In Figure 13, it can be seen that, without the pretrained network, the DDPG shows a wider distribution of engine operation points compared to the HDDPG algorithm, especially at engine torques higher than 140 Nm. At the same time, only a very small number of operation points appear in the high-efficiency region, which reduces the overall fuel economy. Due to the difference in the power flow control, the three control algorithms utilize the battery power differently. Figure 14 shows how the battery SOC changes over the UDDS cycle with the three different control algorithms. The battery current changes produced by the three algorithms over the UDDS cycle are also shown in Figures 15–17.

It can be seen from Figure 14 that the HDDPG uses more battery energy than the other two algorithms without actively recharging the battery using the engine often. In Figure 15, it can be seen that A-ECMS uses the electric motor more frequently with larger amplitudes and applies more engine power to recharge the battery, which decreases the overall fuel economy. The HDDPG algorithm achieves a better fuel economy than the other two algorithms by using the electric motor more efficiently. As a result, the battery current for HDDPG shows a smoother profile with lower peaks in Figure 16. In all cases, the final SOC values are within 2% of each other.

In summary, the HDDPG algorithm performs better than the conventional DDPG algorithm and the optimization-based algorithms in two respects. First, the HDDPG algorithm exhibits a higher training efficiency than the conventional DDPG algorithm, as can be seen in the reward value changes. The HDDPG converges to a higher reward value much faster than DDPG. Second, the HDDPG algorithm can efficiently control the HEV powertrain over unknown drive cycles with comparable or better fuel economy in most cases compared to the conventional optimization-based control algorithms as well as the DDPG algorithm. This is enabled by using an experience replay memory to store past transitions and update the control policy based on changing drive patterns.

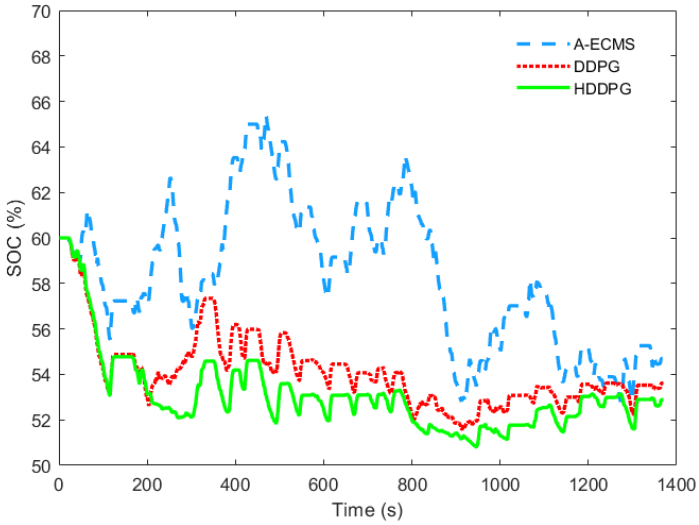

Figure 14. Variations of battery SOC over UDDS for three algorithms.

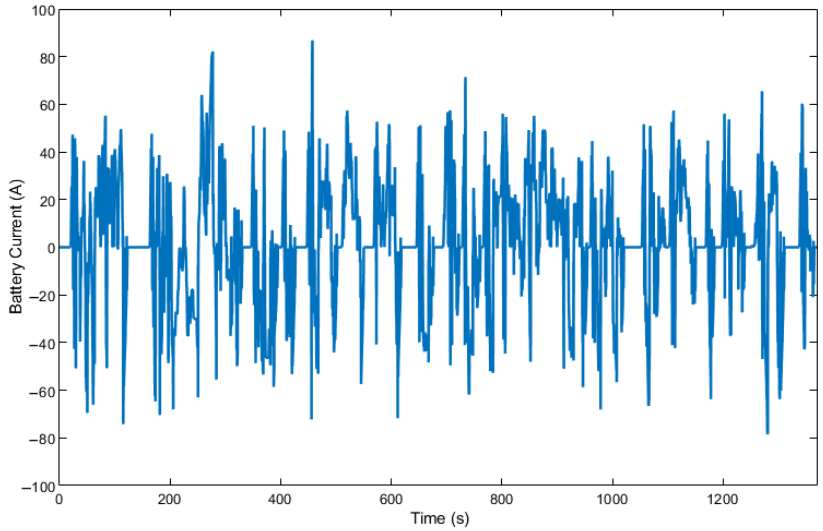

Figure 15. Battery current change by A-ECMS over UDDS.

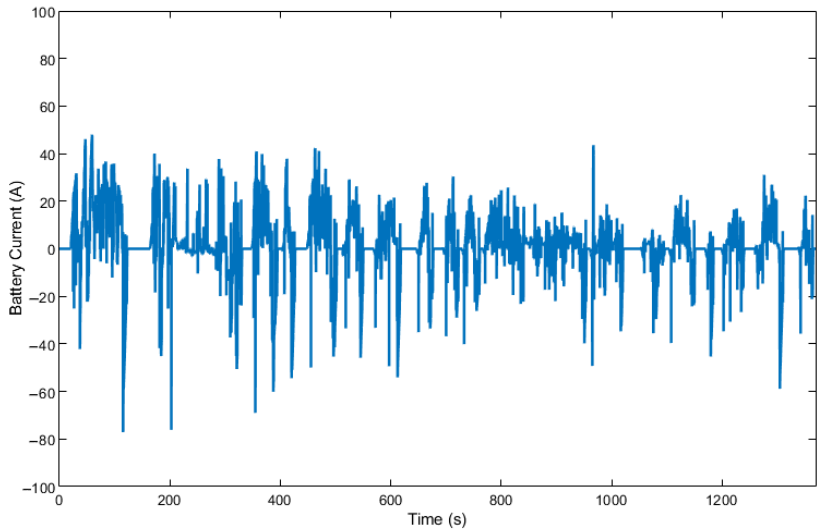

Figure 16. Battery current change by HDDPG over UDDS.

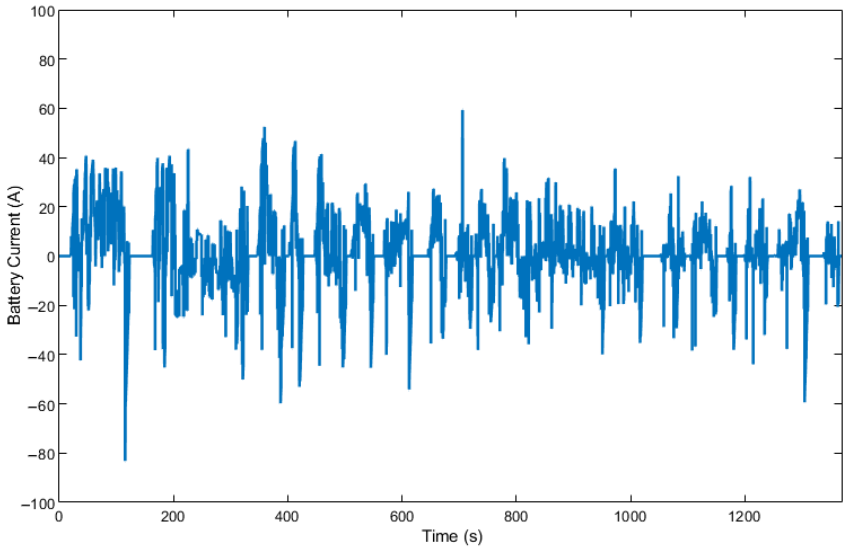

Figure 17. Battery current change by DDPG over UDDS.

6. Conclusions

This paper presents a new approach to the power flow control of a PTTG full HEV using a new offline and online hybrid deep reinforcement learning strategy for improving the training efficiency as well as maximizing the fuel economy. To train the proposed Hybrid Deep Deterministic Policy Gradient (HDDPG) algorithm, vehicle dynamics data are produced by a PTTG HEV simulation model controlled by an optimization-based method, A-ECMS, and the resulting data are used to pretrain the actor target network in the HDDPG algorithm offline. Then, the trained network is inserted into the HDDPG for online training of the HEV controller. In this manner, the agent can learn how to operate the vehicle powertrain in a given environment more quickly than online only algorithms, which learn the optimal control policy by interacting with the vehicle model from zero initial knowledge. The proposed HDDPG algorithm shows an average fuel economy improvement of 2.69% when compared to A-ECMS and DDPG algorithms. The HDDPG algorithm also shows a higher convergence rate in the learning process than the DDPG algorithm. Since the HDDPG algorithm uses an experience replay memory to store past transitions and updates the control policy based on changing driving conditions, it can effectively control the HEV powertrain over unknown drive cycles with comparable or better fuel economy in most cases.

Author Contributions: Conceptualization, Z.Y. and H.-S.Y.; methodology, Z.Y. and H.-S.Y.; software, Z.Y. and H.-S.Y.; validation, Z.Y. and H.-S.Y.; formal analysis, Z.Y. and H.-S.Y.; investigation, Z.Y. and H.-S.Y.; resources, H.-S.Y. and Y.-K.H.; writing—original draft preparation, Z.Y. and H.-S.Y.; writing—review and editing, Z.Y., H.-S.Y. and Y.-K.H.; visualization, Z.Y. and H.-S.Y.; supervision, H.-S.Y. and Y.-K.H.; project administration, H.-S.Y.; funding acquisition, Y.-K.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Science Foundation, grant EEC 1650564, and the E. A. "Larry" Drummond Endowment at the University of Alabama.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Enang, W.; Bannister, C. Modelling and control of hybrid electric vehicles (A comprehensive review). *Renew. Sustain. Energy Rev.* **2017**, *74*, 1210–1239. [CrossRef]
2. Becerra, G.; Alvarez-Icaza, L.; Pantoja-Vázquez, A. Power flow control strategies in parallel hybrid electric vehicles. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2016**, *230*, 1925–1941. [CrossRef]
3. Bellman, R. *Dynamic Programming*; Courier Corporation: New York, NY, USA, 2013; Volume 707.
4. Onori, S.; Serrao, L.; Rizzoni, G. *Hybrid Electric Vehicles: Energy Management Strategies*; Springer: London, UK, 2016.
5. Camacho, E.F.; Alba, C.B. *Model Predictive Control*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
6. Ali, A.M.; Söffker, D. Towards optimal power management of hybrid electric vehicles in real-time: A review on methods, challenges, and state-of-the-art solutions. *Energies* **2018**, *11*, 476. [CrossRef]
7. 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]
8. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
9. Puterman, M.L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
10. Zhao, P.; Wang, Y.; Chang, N.; Zhu, Q.; Lin, X. A deep reinforcement learning framework for optimizing fuel economy of hybrid electric vehicles. In Proceedings of the 2018 23rd Asia and South Pacific design automation conference (ASP-DAC), Jeju, Republic of Korea, 22–25 January 2018; pp. 196–202.
11. 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]
12. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
13. Liessner, R.; Schroer, C.; Dietermann, A.M.; Bäker, B. Deep reinforcement learning for advanced energy management of hybrid electric vehicles. In Proceedings of the ICAART (2), Funchal, Madeira, Portugal, 16–18 January 2018.
14. Yao, Z.; Yoon, H.-S. Hybrid Electric Vehicle Powertrain Control Based on Reinforcement Learning. *SAE Int. J. Electrified Veh.* **2021**, *11*, 165–176. [CrossRef]
15. Fujimoto, S.; Hoof, H.; Meger, D. Addressing function approximation error in actor-critic methods. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018.
16. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. Trust region policy optimization. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015.
17. Christiano, P.F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; Amodei, D. Deep reinforcement learning from human preferences. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
18. 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]
19. Li, Y.; He, H.; Peng, J.; Wang, H. Deep reinforcement learning-based energy management for a series hybrid electric vehicle enabled by history cumulative trip information. *IEEE Trans. Veh. Technol.* **2019**, *68*, 7416–7430. [CrossRef]
20. Olivas, E.S.; Guerrero, J.D.M.; Martinez-Sober, M.; Magdalena-Benedito, J.R.; Serrano, L. *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques: Algorithms, Methods, and Techniques*; IGI Global: Hershey, PA, USA, 2009.
21. Guo, X.; Liu, T.; Tang, B.; Tang, X.; Zhang, J.; Tan, W.; Jin, S. Transfer deep reinforcement learning-enabled energy management strategy for hybrid tracked vehicle. *IEEE Access* **2020**, *8*, 165837–165848. [CrossRef]
22. Zulkifli, S.; Mohd, S.; Saad, N.; Aziz, A. Split-parallel through-the-road hybrid electric vehicle: Operation, power flow and control modes. In Proceedings of the 2015 IEEE Transportation Electrification Conference and Expo (ITEC), Dearborn, MI, USA, 14–17 June 2015; pp. 1–7.
23. EPA. New Fuel Economy and Environment Labels for a New Generation of Vehicles. In *Regulatory Announcement EPA-420-F-11-017*; United States Environmental Protection Agency: Washington, DC, USA, 2011.
24. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic policy gradient algorithms. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014.
25. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]