Optimization of Energy Management Algorithm for Hybrid Power Systems Based on Deep Reinforcement Learning

Lan BAN

School of Mechanical Engineering, University of Science and Technology Beijing, Beijing, 100083, China
Tianjin College University of Science and Technology Beijing, Tianjin, 301830, China
banlan-521@163.com

Abstract: As new energy technologies mature and become more widely available, hybrid vehicles are increasingly being adopted by consumers. However, the existence of at least two power modes and energy management methods has had a significant impact on their related operating costs. This paper proposes an energy management algorithm for this type of vehicles based on an improved deep Q-Learning neural network. This way the energy consumption for hybrid vehicles could be reduced, the efficiency of their energy utilisation could be improved, and their related cost advantages could be enhanced in comparison with traditional fuel-powered vehicles. By using a mixture of multiple representative operating conditions for testing purposes, the conducted experiment confirmed that the fuel consumption per 100 kilometers for the algorithm based on double deep Q-networks under random operating conditions 1 and 2 was 4.05L/100km and 3.64L/100km, respectively. Moreover, the average fuel consumption per 100 seconds for this algorithm was 43ml/100s, which was significantly lower than that of the other three employed automotive powertrain energy management algorithms. The obtained experimental results proved that the energy management algorithm for the hybrid electric vehicle powertrain presented in this paper featured excellent energy control and management capabilities. To sum up, this study could have certain reference significance for improving the cost-effectiveness of hybrid electric vehicles in China.

Keywords: Neural network, Reinforcement learning, Hybrid power, Energy management, Fuel consumption.

1. Introduction

The Hybrid Electric Vehicle (HEV), as a type of vehicle with an excellent energy-saving and environmental performance, has been increasingly accepted by the public (Lu & Li, 2020). This type of car combines the advantages of an internal combustion engine and an electric motor, aiming to provide better fuel efficiency and lower exhaust emission levels (Zhao et al., 2021). Hybrid Electric Vehicle Energy Management (HEVEM) has become a key technical challenge as it directly affects the fuel efficiency and performance of these vehicles. Optimizing the powertrain of HEVs to improve their energy conversion efficiency and reduce their energy consumption is a standard powertrain issue in the automotive industry. Designing excellent energy management methods can effectively reduce the operating costs of HEVs, thereby enhancing their market competitiveness and value.

Traditional HEVEM technology, Model Predictive Control (MPC), and Dynamic Programming (DP) require the use of heuristic algorithms in areas such as fuzzy control, parameter optimization, and motion mode construction (Qi et al., 2022). DP is a commonly used dynamic optimization technique, and MPC is specifically designed to handle planning problems with severe information loss in the dynamic optimization. These methods have certain advantages, such as fast processing speed, easy implementation, and being easy to understand. However, their main drawbacks are also very obvious. Their generalization ability is weak and they can only handle certain specific working conditions, such as urban road conditions, highway conditions, etc. In addition, their optimization objectives are usually fixed and cannot meet the needs triggered by multi-objective and dynamic changes. In recent years, Deep Reinforcement Learning (DRL) has shown a strong performance in multiple fields, a thing which has the potential to change the current state of HEVEM. As a way of imitating human learning, DRL can optimize decision-making through self-learning methods during continuous interaction with the environment. Although DRL has been recognized to some extent in HEV, there is still limited research on optimization of DRL-based HEVEM. Based on this background, it was necessary to conduct this study. The contribution of this study lies in the design of an energy management model for HEV based on improved double deep Q-networks. This model uses deep learning and reinforcement learning techniques to carry out the energy system management of HEV under nonlinear and high latitude data conditions, effectively reducing fuel consumption during operation, and providing a feasible new solution for the HEVEM.

The remainder of this paper is as follows. Section 2 introduces the research results of domestic and foreign scholars on HEV vehicle energy

https://doi.org/10.24846/v33i2y202402

management, as well as the application value of deep reinforcement learning algorithms in improving vehicle energy management. Section 3 mainly elaborates on the design process of the energy management algorithm for the HEV hybrid power system based on an improved deep Q-learning network (DQN). Section 4 refers to the experiments conducted for verifying the performance of the proposed algorithm and applying it to actual HEV vehicle operating conditions. Finally, Section 5 includes the conclusion of this paper, related to the obtained experimental results and the shortcomings of this research.

2. Related Works

Vehicle energy management has become increasingly important due to the increasing demand for environmental protection and energy efficiency. Given the increasing advancement of decentralized power systems and electric vehicles, Tan & Chen (2020) sought to improve the energy management performance of multiple microgrid systems under the uncertainty of electric vehicle charging. A multi-objective optimization model was established to minimize transmission losses, operating costs, and carbon emissions in a multi-microgrid system. The simulation results confirmed the superiority of the improved algorithm in terms of global search performance and fast convergence performance (Tan & Chen, 2020). Ghaderi et al. (2020) studied the impact of cell and fuel cell degradation on HEVEM. To this end, an online energy management strategy that simultaneously adapted to both battery and Fuel Cell (FC) models was proposed. These test scenarios using standard driving cycles confirmed that the electromagnetic system could successfully solve the model uncertainty caused by power performance drift in the above situations (Ghaderi et al., 2020). Demircali & Koroglu (2020) stated that in a multi-power hybrid structure, an energy management system was needed to improve system efficiency and provide optimal power sharing between the battery and power supply. The performance of Jaya optimization method was compared with that of DP, one of the global optimization methods, and that of particle swarm optimization, another heuristic real-time application method. These simulation results confirmed that the Jaya optimization method had a loss of nearly 3.1%, achieving the best effect in terms of total energy loss (Demircali & Koroglu, 2020). Cui et al. (2022) proposed a multi-objective hierarchical strategy with low computational complexity by combining resistance network-triggered motion planning and convex torque optimization based on a variable direction multiplier. Finally, the superiority of this method was verified through simulation and hardware in loop experiments (Cui et al., 2022).

The development of efficient HEVEM strategies has become a key task due to the variability of the topology structure of electrified power systems and the uncertainty of driving scenarios. By utilizing DRLs, a heuristic rule-based local controller was embedded in the loop to eliminate unreasonable torque allocation while considering the characteristics of the powertrain components. In addition, a hybrid experience playback method based on a hybrid experience buffer composed of offline optimal computing experience and online learning experience was proposed to address the impact of environmental interference. These experiments confirmed that under different operating conditions, the improved DRL obtained the best fuel optimality, fastest convergence speed, and highest robustness compared with typical value-based and policy-based optimization methods Wang et al. (2020) combined computer vision with DRL to improve the fuel economy of HEV, which could autonomously learn the optimal control strategy from visual input. These experiments confirmed that visual information-based DRL systems reduced fuel consumption by 4.3% to 8.8% in comparison with systems without visual information, achieving a global optimal DP fuel economy of 96.5% .

In summary, although previous studies included extensive research on the energy-saving issues related to mechanical and electronic products such as HEV, most of them did not employ cutting-edge neural network algorithms to construct energy-saving methods. Meanwhile, the effectiveness of energy-saving design methods has not been verified using multiple complex operating conditions, so this study was conducted with the aim to compensate for these shortcomings.

3. Research Methods

Before embarking on a specific research, it is necessary to first clarify the object of the research. There are three types of HEVs: parallel, series, and hybrid HEVs. The hybrid HEV can optimize the motor and engine under different operating

conditions and environments. This is equivalent to simultaneously possessing the advantages of the other two HEVs, which can improve the power performance of vehicles while ensuring excellent fuel economy (Alfaverh, Denai & Sun, 2023). However, the disadvantage of hybrid HEV is that the system structure is too complex, making control operations more difficult, which results in significant energy loss in the entire vehicle. Therefore, it is of practical significance to investigate and adopt more suitable control methods for a hybrid HEVEM.

3.1 The Design of HEV Hybrid Energy Management Strategy Based on DQN

As it was mentioned above, the object of this study is the power split HEV in the hybrid HEV. The power split HEV includes an engine and several drive motors, which can transform multiple operating modes according to different driving needs, thus achieving a superior performance (Wei et al., 2021). By contrast, series hybrid vehicles may have reduced energy efficiency due to multiple energy conversions (Mellit, Pavan & Lughi, 2021). In comparison with parallel hybrid vehicles, planetary gear design compensates for the construction shortcomings of power coupling systems in urban driving scenarios, which creates conditions for the transmission system to operate in high-efficiency areas (Paterova & Prauzek, 2021). However, the relatively high complexity of the double-array planetary structure also increases the difficulty of control.

The research on the energy management control strategy for a vehicle cannot be separated from the power system structure of the vehicle itself. Figure 1 shows the power transmission system of a power split HEV (Zhang et al., 2020). The power system features a double-row planetary structure, with Motor 1 (MG1) connected to a sun gear on the planetary row, while the planet carrier is connected to the engine. The planet carrier of planetary row 2 is also fixed, and the sun gear is connected to Motor 2 (MG2). The corresponding gear rings of the two planetary rows are fixed and also connected to the output end. The double-array planetary can decouple the engine from the driving load, which can also adjust the speed and torque of MG1 (Huang et al., 2022). MG2 can provide power in pure electric mode and recover energy by providing braking torque when the car slows down.

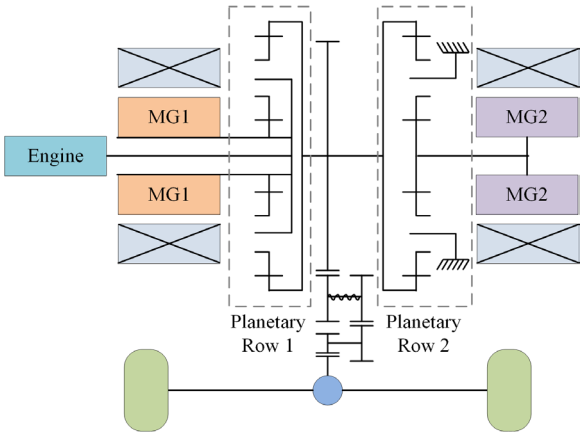

Figure 1. Power Transmission System Structure for a Power Split HEV

Considering the physical size and performance of such vehicles, the basic parameters corresponding to the transmission structure of the HEV in Figure 1 are set as follows. In terms of the entire vehicle, the total mass, wheel radius, windward area, rolling resistance coefficient, air resistance coefficient, and air density are 1400kg, 0.288 m, 2.54 m^2, 0.015, 0.28, and 1.2 kg/m^3, respectively. In terms of engine, the maximum power and maximum torque are 73kW@5200r/min and 145N·m@4000r/min. The MG1 and MG2 are rated at 42kW, 10,000r/min and 60kW, 12,000r/min, respectively, in terms of power and top speed. The above parameters are prerequisites for building a vehicle driving model and designing energy management.

The engine dynamic effects of the power split HEV are short-lived and can be ignored. Therefore, this study used data from engine bench experiments to construct a quasi-steady state model of the engine, motor, battery, planetary gear, and vehicle driving dynamics in the automotive hybrid power system.

The energy distribution of the motor and engine during driving is essentially a sequential decision-making problem. Therefore, it is more appropriate to use Q-learning in reinforcement learning. However, this algorithm lacks the ability to parameterize data fitting and has poor computational performance for high latitude inputs (there is a high number of features or dimensions in the input data). Therefore, this study presents the design of a DQN for a power split HEVEM. Traditional DP and MPC methods have a poor energy planning capability under complex road conditions. Processing complex nonlinear data is the advantage of deep learning algorithms, which is also the main reason for choosing to use the DQN algorithm to construct energy management models.

When dealing with the energy management problem for a power split HEV, Q-learning needs to discretize the state and control in advance. The high accuracy of discretization indicates good computational performance, but the corresponding computational workload also increases significantly (Han & Yang, 2021). However, the state of the vehicle is a continuous variable and may not fall on the pre-discretized grid points of the Q table in the actual driving process. In this case, using interpolation to estimate the Q value of state-action pairs may result in errors. On the contrary, DQN directly inputs the vehicle state into the deep network processing method, which can avoid interpolation errors in Q-learning and ensure the continuity of state changes. Based on this, the reward function, state, and action of the DQN network are first designed.

In DQN, the vehicle speed V, acceleration A, engine speed W_e, and battery SOC are considered as system state variables. The value of the control action is the torque T_{MG1} of MG1. The setting of instant rewards is consistent with Q-learning. Equation (1) expresses the state and the value of the control action:

$$S = \left\{ s = \left[V, A, W_e, SOC \right]^T \right\} \tag{1}$$

In equation (1), s and S represent the state variable and its set, respectively.

In equation (2), a represents the execution of the action:

$$A = \left\{ a = T_{MG1} \right\} \tag{2}$$

Equation (3) represents the reward $R(s,a)$ (Zhang et al. ,2022):

$$R(s,a) = -\int_0^T \left\{ fuel + \lambda \left(SOC(t) - SOC_{ref} \right)^2 \right\} dt \tag{3}$$

In equation (3), T is the endpoint of the time period for measuring rewards, *fuel* represents the fuel consumption, λ is the penalty coefficient and SOC_{ref} is the reference battery status. A five-layer fully connected neural network is used to construct a DQN to adapt to state action pairs' Q value. The input layer has four neurons corresponding to the vehicle's speed, acceleration, engine speed, and battery. The number of neurons for the three hidden layers is 200, 100, and 100, respectively, using ReLU activation function. Figure 2 shows the designed network structure.

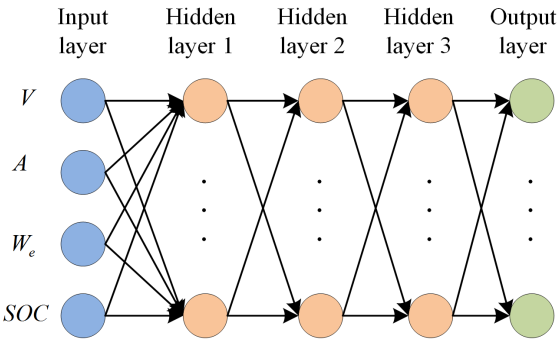

Figure 2. Deep Neural Network Structure in DQN (Kose & Oktay, 2023)

The output layer in Figure 2 discretizes T_{MG1} into 32 parts using the linear activation function in equation (4). Corresponding to the output layer neurons, each output represents the Q value of the corresponding control action in the current state:

$$A = \left\{ A_1, A_2, ..., A_{32} \right\} \tag{4}$$

The variation range for the battery SOC is small, but the variation range for the vehicle speed and engine speed is wide. The vehicle state is normalized to ensure the convergence speed of the model. In DQN, the ε-greedy strategy is used to execute control actions, where ε represents the exploration rate. In strategy exploration, actions are randomly selected with probability ε (Yuan et al., 2021). Then, for strategy utilization, the action corresponding to deep network's maximum Q value is selected with probability 1-ε. So far, the operation mode for the HEV energy management policy based on DQN can be obtained, which is illustrated in Figure 3.

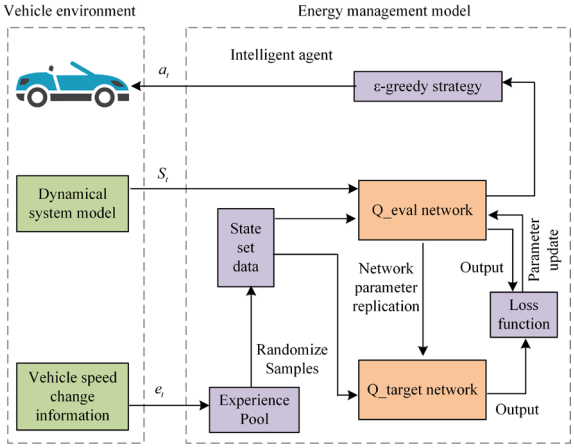

Figure 3. HEV Energy Management Operation Mode Based on DQN Algorithm

The management process includes a control cycle and a learning cycle. In the control cycle, the controller employs the ε-greedy strategy based on the current state information s_t for the vehicle. There is a probability ε to select a random action for exploration, while there is a probability 1-ε to select the action with the maximum Q value after executing Q_eval network (Peng et al., 2021). After completing the action, the vehicle state will change and the relevant state action sequence will be stored in the experience pool. When the experience pool data storage space is full, the learning task begins to execute. A dual network structure including Q_eval and Q_target is set up to achieve strategy learning and optimization in the learning cycle. Q_eval is used to calculate the Q value of the current state action pair and generate the optimal control action, while its network parameter θ undergoes gradient updates at each step. Q_target is used to calculate the target Q value, and its network parameters do not require gradient updates. Instead, they are copied from Q_eval parameter θ to Q_target parameter θ^- at every fixed step. This delayed update system enhances the stability of DQN. The loss function $L(\theta)$ of DQN is defined as the square of the difference between the target Q and the predicted Q in equation (5):

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

Here, $r+\gamma\cdot\max_{a_{t+1}}Q\left(s_{t+1},a_{t+1},\theta^-\right)$ represents the target Q. r and γ are the parameters of the median function in the Q table, namely the output of Q_target. θ^- is a parameter copied from Q_eval at regular intervals. $Q(s_t,a_t,\theta)$ and θ represent the output of Q_eval and the parameter updated by real-time gradient, respectively.

3.2 The Design of DQN Algorithm Calculation Process

Although the designed DQN algorithm has a higher fitting ability and high latitude data processing ability in comparison with the Q-learning algorithm, there is still an overestimation problem with Q-learning. Overestimation refers to the fact that the true value in learning is lower than the updated estimated value. The problem with DQN arises from the maximization calculation for the median function, as follows. The median function $Q(s_t,a_t,)$ in the Q table is updated according to equation (6) below. The advantage of these equations is that they incorporate Double Deep Q-networks (DoubleDQN) by contrast with traditional reinforcement learning algorithms, thereby effectively reducing overestimation of state values (Xiaofei et al., 2022):

$$Q\left(s_t,a_t\right)\leftarrow Q\left(s_t,a_t\right)+\alpha\left[R+\gamma\max_{a'}Q\left(s',a'\right)-Q\left(s_t,a_t\right)\right] \quad (6)$$

In equation (6), s' and a' are the corresponding state variables when Q reaches its maximum values s_t and a_t at time t. R is the parameter γ of the median function in Q table. Equation (7) represents the update of the corresponding DQN median function θ_t.

$$\theta_{t+1}=\theta_t+\alpha\left[r+\gamma\max_{a'}Q\left(s',a';\theta^-\right)-Q\left(s,a;\theta\right)\right]\nabla Q(s,a;\theta) \quad (7)$$

From equations (7) and (8), both Q-learning and DQN value function updates include the max operation. The max operation resulted in the actual $Q(s',a')$ value being lower than the estimated value. It is assumed that all states of $Q(s',a')$ were overestimated with the same magnitude. The greedy strategy only focuses on the maximum Q value and corresponding actions. Even if all value functions are uniformly overestimated, this strategy will not affect the generation of the optimal strategy, because the goal of reinforcement learning is to find the optimal strategy, rather than accurately calculating the value function corresponding to each state. However, the overestimation of the target Q is not uniform for the solution of practical problems, which may result in the learning strategy being suboptimal rather than optimal.

Yang et al. (2022) have proposed the DoubleDQN method to solve this problem, which can effectively reduce state value overestimation and outperform DQN for many problems. Here, this method is also used for action selection and evaluation to calculate the target Y_t^Q in equation (8):

$$Y_t^Q=r+\gamma\max_{a'}Q\left(s',a';\theta^-\right) \quad (8)$$

The choice of action is to solve Y_t^Q, and it is necessary to select an action a^* to maximize $Q(s',a')$ at state s'. The evaluation of actions consists in calculating the corresponding state action value function of a^*, which will form the target Q.

In DoubleDQN, different neural network parameters are used for action selection and evaluation, and its target $Y_t^{DoubleQ}$ is calculated according to equation (9):

$$Y_t^{DoubleQ} = r + \gamma Q\left(s', \arg\max_{a'} Q(s', a', \theta); \theta^-\right) \quad (9)$$

The action value function in DoubleDQN is calculated based on parameter θ in equation (10):

$$\arg\max_{a'} Q(s', a', \theta) \quad (10)$$

After selecting the maximum action a^*, the action is evaluated using equation (11), and the network parameter used this time is θ:

$$Y_t^{DoubleQ} = r + \gamma Q\left(s', a^*; \theta^-\right) \quad (11)$$

DQN reduces the time series correlation for different kinds of data through experience playback. However, it is difficult for intelligent agents to learn efficiently by using uniform sampling methods to sample data from the experience pool. This is because the empirical data generated by the interaction between the environment and the agent is not equally important. Some states related to it may have a higher learning value than others. Therefore, this study adopts the prioritised experience replay method to extract data from the experience pool and give higher sampling weights to states with higher learning efficiency. Prioritised experience replay mainly includes two steps: defining priority and sampling. Priority is first defined, and the absolute value $|\delta|$ of the temporal difference deviation for the sample is selected to evaluate the priority of the sample. Therefore, the temporal difference deviation of sample m can be calculated as the absolute value $|\delta(m)|$ of the difference between the target Q and the estimated Q according to equation (12):

$$|\delta(m)| = \left| r + \gamma Q\left(s', \arg\max_{a} Q(s', a, \theta); \theta^-\right) - Q(s, a; \theta) \right| \quad (12)$$

If δ is larger, it indicates that the estimated Q of the state is greater than the target Q, which has a higher learning value. Therefore, prioritizing the playback of samples with larger δ can enable reinforcement learning algorithms to converge quickly. However, if samples with larger δ are consistently selected according to the greedy pattern, sample diversity will be lost, which may lead to over-fitting. Therefore, a very small constant ξ greater than 0 is added in this study to the temporal difference deviation of each sample as the sampling priority p_m of the sample in equation (13):

$$p_m = \delta + \xi \quad (13)$$

Then a sampling section is designed. Figure 4 shows the experience pool storage structure in DQN. A binary tree is used to store experience pool sample data with priority. The leaf nodes in the binary tree store the data and priority of the samples, while non-leaf nodes store the sum of the priority values for their child nodes.

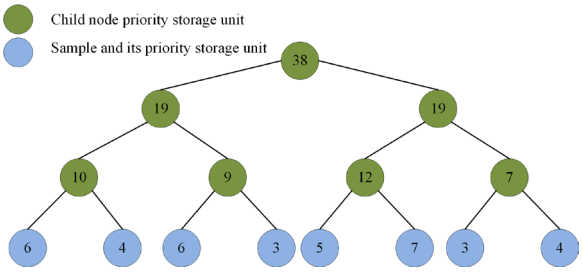

Figure 4. DQN Experience Pool Storage Structure (Wu et al., 2020)

The sampling probability $P(m)$ of sample m is described in equation (14):

$$P(m) = p_m^\alpha \Big/ \sum_k p_m^\alpha \quad (14)$$

In equation (14), the parameter α determines the priority, and its value range is (0,1). When α drops to 0, it means uniform sampling is adopted. On the contrary, the adopted degree of priority sampling is greater.

The prerequisite for updating the value function using batch gradient descent is that the sampling distribution should be consistent with the value function distribution. But prioritizing experience replay may break this consistency. To correct the deviation, it is now necessary to calculate an importance sampling coefficient w_k in equation (15) before the gradient:

$$w_k = \left[\frac{1}{N} \frac{1}{P(m)} \right]^\beta \quad (15)$$

In equation (15), N represents the size of the experience pool, while β represents the weight coefficient. The sampling coefficient w_k is often standardized to ensure the stability of the algorithm during use. At this point, the DoubleDQN with prioritized experience playback is constructed and applied to the power split HEVEM, replacing the traditional DQN as it is illustrated in Figure 5.

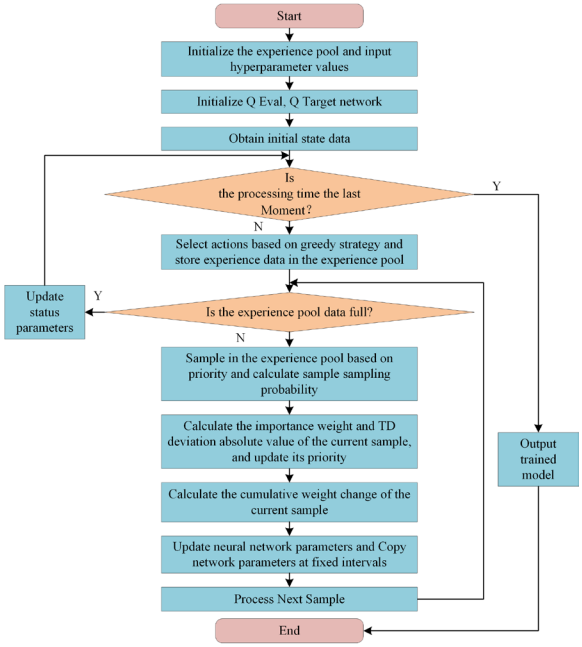

Figure 5. Double DQN Algorithm Flow with Prioritised Experience Playback

4. Performance Testing for Energy Management Algorithms

A simulation experiment was conducted using the condition recognition method for the purpose of testing the energy management capability of this design method. This experiment included various common hybrid vehicle energy control methods, which were analyzed according to the loss function, battery SOC, model memory consumption, and vehicle fuel consumption indicators.

4.1 Experimental Design

Some typical operating conditions were selected to construct combined operating conditions, which were used as data for testing various controllable models and methods. The selected typical operating conditions should comprehensively reflect the actual driving conditions and road environment for the vehicle. They should at least cover suburban and high-speed conditions, idle mode, constant speed, rapid deceleration conditions, and other operating conditions. Table 1 shows the selected typical operating conditions.

The above operating conditions were concatenated into a complete testing condition according to the number sequence. Figure 6 shows the acceleration and velocity variation curves for this full operating condition. Figures 6(a) and 6(b) show the acceleration curve and the corresponding velocity curve of the process, respectively. The total operating time after splicing was 10985 seconds. Further on, Q-learning, DP, and Faster-RCNN were selected to construct a comparative model, and compare the mode parameters and select the optimal solution through multiple experiments.

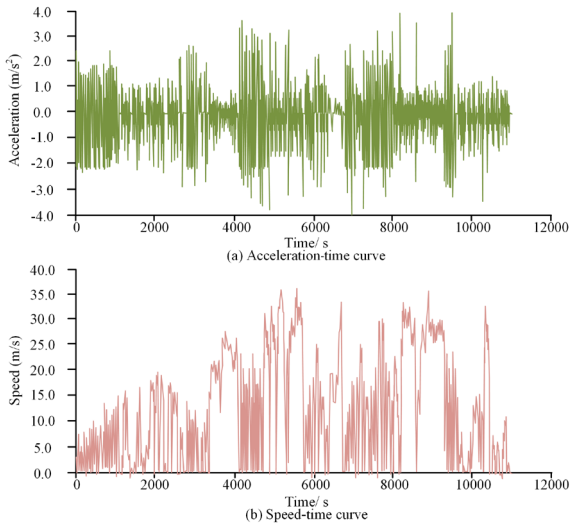

Figure 6. Complete Speed and Acceleration Variation Curves Spliced from Typical Operating Conditions

Table 1. Typical operating conditions Information Table

Number	Condition name	Time length (s)	Data Size (KB)	Number of types of road conditions covered
#1	WVUSUB	1665	322	3
#2	NYCC	599	58	1
#3	NurembergR36	1082	126	2
#4	WVUCITY	1408	249	4
#5	LA92	1436	301	1
#6	SC03	601	69	1
#7	ARB02	1640	285	2
#8	HWFET	766	41	5
#9	REP05	1401	364	2
#10	EUDC	401	72	5

4.2 Experimental Results Analysis

First, the changes in the loss function for each model were analysed during the training phase as it is illustrated in Figure 7. The horizontal axis and vertical axis represent the number of iterations and loss function, respectively, and different curves represent different models. When the number of iterations exceeded 300, each model completed training. At this time, the loss functions values for DoubleDQN, Q-learning, DP, and Faster-RCNN were 1.82, 4.25, 5.36, and 6.29, respectively. The DoubleDQN illustrated in Figure 7 obtained the smallest value for the loss function.

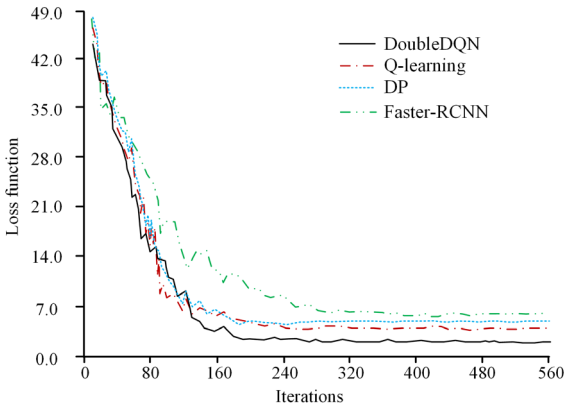

Figure 7. Changes in Loss Functions of Each Model during the Training Phase

According to practical experience in the industry, the core operating conditions that should be included in the data for testing automotive energy management strategies included congestion conditions, suburban conditions, and high-speed conditions. Due to their size relationship in terms of speed, they were assigned the values of "1", "2", and "3", respectively. Due to the high computational complexity of testing the complete operating conditions, at least one segment with numbers "1", "2", and "3" was randomly selected from the complete operating conditions to form random testing conditions 1 and 2. Figure 8 shows the vehicle speed under random condition 1 and the corresponding SOC calculation results. Figure 8(a) shows the speed and condition number variation curve for random condition 1. Figure 8(b) shows the SOC curve for random condition 1. As it can be seen in Figure 8, the variation of operating conditions within random condition 1 was relatively complex, including three basic operating conditions, and the switching between basic working conditions was frequent. Except for the energy management model based on Faster-RCNN, the trend of SOC curve change for the other models was generally consistent, while the former had a higher overall value. Among the other three management modes, DoubleDQN had the lowest average SOC value, which meant it worked the most with a motor. The analysis results for random condition 2 were basically consistent with Figure 8.

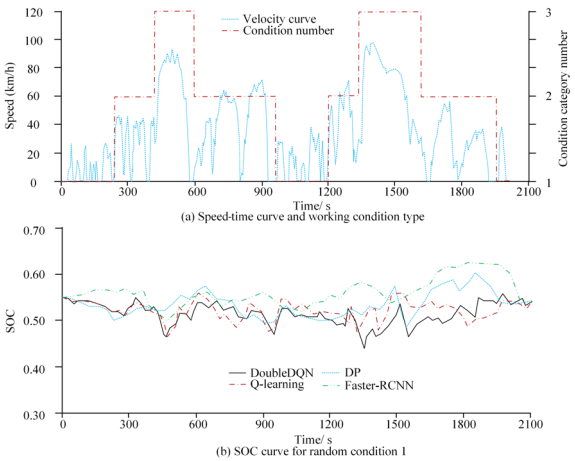

Figure 8. Random Condition 1 Vehicle Speed Mode and SOC Calculation Results

The employed models were also compared from the perspective of fuel consumption indicators. Figure 9 shows the overall fuel consumption under various operating conditions for the four energy management models. The four energy management models of various hybrid systems are displayed on the horizontal axis, while the vertical axis represents the fuel consumption of each model under comparative operating conditions, in L/100km. The two filling styles represent the two random operating conditions. Each experimental protocol was repeated 10 times to improve the reliability of statistical results, and these results were computed in the form of mean and standard deviation. The management model based on Faster-RCNN had the highest fuel consumption per 100 kilometers for both random operating conditions than all other management models. The management model based on DoubleDQN had the lowest fuel consumption per 100 kilometers in random operating conditions 1 and 2 among all the employed models, while the fuel consumption of the Q-learning management model was slightly higher than that of DoubleDQN. Specifically, the fuel consumption per 100 kilometers for DoubleDQN, Q-learning, DP, and Faster-RCNN

under random conditions 1 and 2 was 4.05, 4.13, 4.19, and 4.46, and 3.64, 3.81, 3.94, and 4.25 L/100km, respectively.

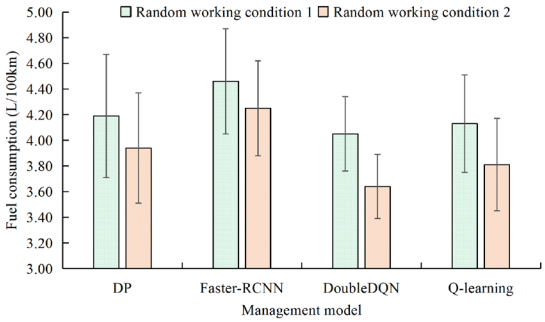

Figure 9. Comparison of Overall Fuel Consumption for the employed Energy Management Models

Then a specific analysis was conducted on the fuel consumption changes for each management model under complete operating conditions taking random working condition 1 as an example. The evaluation index was changed to the fuel consumption within a hundred seconds. Figure 10 shows the obtained statistical results. The horizontal axis represents time, in seconds, and the vertical axis represents the fuel consumption within 100 seconds for each of the two schemes, in ml/100s. Figure 10(a) illustrates the fuel consumption for the Q-learning model and DoubleDQN model over time for intervals of 100 seconds, and Figure 10(b) illustrates the fuel consumption for the DP model and Faster RCNN model over time for intervals of 100 seconds.

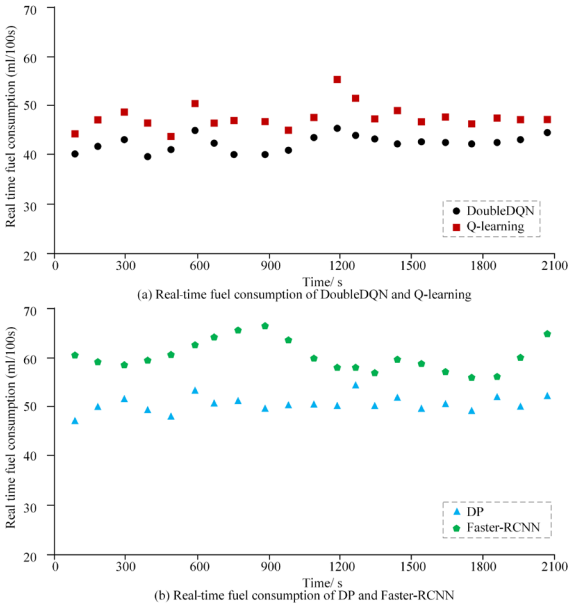

Figure 10. Real-time Fuel Consumption Changes for the employed Energy Management Models under Random Condition 1

There was a negative correlation between the fuel consumption within 100 seconds and the SOC value for each model, because the power of HEV came from fuel-powered engines and electric motors, and the battery energy consumption was low. Under the same conditions, it meant that the engine energy consumption was high. The average fuel consumption per 100 seconds for DoubleDQN as it is illustrated in Figure 10 was 43ml/100s, which was significantly lower than that of the other energy management models.

Finally, the consumption of hardware system memory for each model was compared to determine their deployment difficulty, as it is illustrated in Figure 11. Due to the low computational complexity in this case, all models were directly tested under the complete splicing condition. The memory consumption for DoubleDQN was relatively high, second only to Faster-RCNN, because the DoubleDQN model contains a deep neural network structure internally, which increases the overall complexity and total number of parameters of the model. Due to the simple calculation logic, the DP-based model had fewer parameters and involved fewer calculation processes, resulting in minimal memory consumption. Specifically, the average memory consumption for DoubleDQN, Q-learning, DP, and Faster-RCNN under the complete splicing condition was 238MB, 176MB, 89MB, and 271MB, respectively.

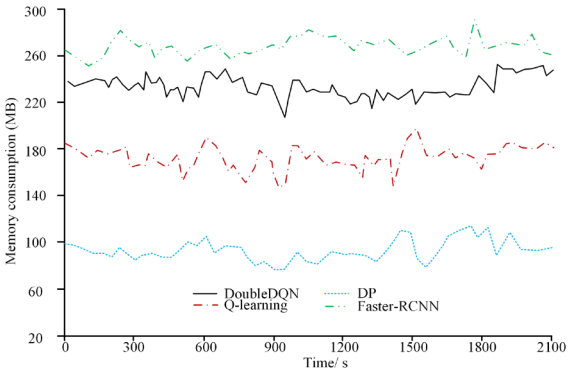

Figure 11. Comparison of Memory Consumption during the Running Process for the Employed Models

Comparing the time complexity of each model from a mathematical perspective, the calculation of the number of operating conditions for the four models in Table 2 was carried out by randomly concatenating the various conditions listed in

Table 2. Comparison Results for Average Calculation Time and Time Complexity for the Employed Models (Unit: second)

The number of operating conditions to be calculated	Improve DoubleDQN	DP	Q-learning	Faster-RCNN
1	2.4	0.7	1.3	3.2
10	15.2	1.3	2.5	32.5
100	46.8	2.5	6.4	75.9
1000	285.1	9.2	28.2	463.1
10000	763.5	26.7	105.1	1254.8

Table 1. Table 2 includes the mixed comparison results. As it can be seen in Table 2, the improved DoubleDQN model proposed in this paper took 2.4s, 15.2s, 46.8s, 285.1s, and 763.5s to calculate the number of operating conditions for the cases involving 1, 10, 100, 1000, and 10000 working conditions, respectively, which was longer than for all other compared models except for the Faster-RCNN. The time complexity of this model was relatively high.

The computational complexity of the HEVEM model based on the improved DQN presented in this paper was higher than that of similar traditional models. A detailed analysis is given below. Firstly, the model proposed in this paper used a fully connected neural network with a five-layer structure, which contained a large number of neuron structures and coefficients, thereby improving the overall computational complexity of this model. Secondly, prioritising experience replay calculation also increased the computational complexity of the employed algorithm, as this method required first adding high-priority experiences to the experience replay time. The prerequisite for this step was to scroll through all experience elements and find the experience with the highest priority. Finally, the model had a dual network structure, which included two types of networks: Q_eval and Q_target. Both networks started optimizing parameters during the training process, which also increased the overall complexity of this model (originating from experimental results and original images).

5. Conclusion

This paper proposed a HEVEM based on an improved DQN to enhance the energy management efficiency of HEV hybrid systems. When the number of iterations exceeded 300, each model completed the training. At this point, the values of the loss functions of DoubleDQN, Q-learning, DP, and Faster-RCNN were 1.82, 4.25, 5.36, and 6.29, respectively. The proposed DoubleDQN obtained the smallest value for the loss function. The SOC variation patterns for DoubleDQN, Q-learning, and DP algorithms were consistent, but the average value of SOC obtained by DoubleDQN was the smallest, namely 0.49. The energy management model based on Faster-RCNN had a higher fuel consumption per 100 kilometers in the two random operating conditions than all other management models. The management model based on DoubleDQN presented in this study had the lowest fuel consumption per 100 kilometers in random conditions 1 and 2 among all the employed models. The fuel consumption per 100 kilometers for DoubleDQN, Q-learning, DP, and Faster-RCNN under random condition 1 and 2 was 4.05, 4.13, 4.19, and 4.46, and 3.64, 3.81, 3.94, and 4.25L/100km, respectively. The average fuel consumption per 100 seconds for DoubleDQN was 43ml/100s, which was significantly lower than that of the other management models. Therefore, the energy management algorithm for the HEV hybrid system proposed in this paper had a stronger energy management capability. However, due to research limitations, subjective evaluation research on this algorithm has not been carried out, which is also an area that future research should focus on.

REFERENCES

Alfaverh, F., Denai, M. & Sun, Y. (2023) Optimal vehicle-to-grid control for supplementary frequency regulation using deep reinforcement learning. *Electric Power Systems Research*. 214(B), 108949. doi: 10.1016/j.epsr.2022.108949.

Cui, W., Cui, N., Li, T., Cui, Z., Du, Y. & Zhang, C. (2022) An efficient multi-objective hierarchical energy management strategy for plug-in hybrid electric vehicle in connected scenario. *Energy*. 257, 124690. doi: 10.1016/j.energy.2022.124690.

Demircali, A. & Koroglu, S. (2020) Jaya algorithm-based energy management system for battery- and ultracapacitor-powered ultralight electric vehicle. *International Journal of Energy Research*. 44(6), 4977-4985. doi: 10.1002/er.5248.

Ghaderi, R., Kandidayeni, M., Soleymani, M., Boulon, L. & Chaoui, H. (2020) Online energy management of a hybrid fuel cell vehicle considering the performance variation of the power sources. *IET Electrical Systems in Transportation*. 10(4), 360-368. doi: 10.1049/iet-est.2020.0035.

Han, B. A. & Yang, J. J. (2021) A Deep Reinforcement Learning Based Solution for Flexible Job Shop Scheduling Problem. *International Journal of Simulation Modelling.* 20(2), 375-386. doi: 10.2507/IJSIMM20-2-CO7.

Huang, R., He, H., Zhao, X., Wang, Y. & Li, M. (2022) Battery health-aware and naturalistic data-driven energy management for hybrid electric bus based on TD3 deep reinforcement learning algorithm. *Applied Energy*. 321, 119353. doi: 10.1016/j.apenergy.2022.119353.

Kose, O. & Oktay, T. (2023) Simultaneous design of morphing hexarotor and autopilot system by using deep neural network and SPSA. *Aircraft Engineering and Aerospace Technology.* 95(6), 939-949. doi: 10.1108/AEAT-07-2022-0178.

Lu, M. & Li, F. (2020) Survey on lie group machine learning. *Big Data Mining and Analytics*. 3(4), 235-258. doi: 10.26599/BDMA.2020.9020011.

Mellit, A., Pavan, A. M. & Lughi, V. (2021) Deep learning neural networks for short-term photovoltaic power forecasting. *Renewable Energy*. 172(C), 276-288. doi: 10.1016/j.renene.2021.02.166.

Paterova, T. & Prauzek, M. (2021) Estimating Harvestable Solar Energy from Atmospheric Pressure Using Deep Learning. *Elektronika ir Elektrotechnika*. 27(5), 18-25. doi: 10.5755/j02.eie.28874.

Peng, L., Wang, Y., Zhang, F., Zhang, J. & Li, Z. (2021) Evaluation of emergency driving behaviour and vehicle collision risk in connected vehicle environment: A deep learning approach. *IET Intelligent Transport Systems*. 15(4), 584-594. doi: 10.1049/itr2.12053.

Qi, C., Song, C., Xiao, F. & Song, S. (2022) Generalization ability of hybrid electric vehicle energy management strategy based on reinforcement learning method. *Energy*. 250, 123826.1-123826.9. doi: 10.1016/j.energy.2022.123826.

Tan, B. & Chen, H. (2020) Multi-objective energy management of multiple microgrids under random electric vehicle charging. *Energy*. 208(2), 118360.1-118360.18. doi: 10.1016/j.energy.2020.118360.

Wang, Y., Tan, H., Wu, Y. & Peng, J. (2020) Hybrid Electric Vehicle Energy Management With Computer Vision and Deep Reinforcement Learning. *IEEE Transactions on Industrial Informatics*. 17(6), 3857-3868. doi: 10.1109/TII.2020.3015748.

Wei, S., Tien, P. W., Wu, Y. & Calautit, J. K. (2021) The impact of deep learning based equipment usage detection on building energy demand estimation. *Building Services Engineering Research & Technology*. 42(5), 545-557. doi: 10.1177/01436244211034737.

Wu, C., Cui, Y., Ji, C., Kuo, T. W., & Xue, C, J. (2020) Pruning deep reinforcement learning for dual user experience and storage lifetime improvement on mobile devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.* 39(11), 3993-4005. doi: 10.1109/TCAD.2020.3012804.

Xiaofei, Y., Yilun, S., Wei, L., Hui, Y., Weibo, Z. & Zhengrong, X (2022) Global path planning algorithm based on double DQN for multi-tasks amphibious unmanned surface vehicle. *Ocean Engineering*. 266(1), 112809.1-112809.14. doi: 10.1016/j.oceaneng.2022.112809.

Yuan, S., Zhang, Y., Qie, W., Ma, T. & Li, S. (2021) Deep Reinforcement Learning for Resource Allocation with Network Slicing in Cognitive Radio Network. *Computer Science and Information Systems*. 18(3), 979-999. doi: 10.2298/CSIS200710055Y.

Zhang, H., Shi, D., Cai, Y., Zhou, W. & Yang, H. (2020) Research on Transmission Efficiency Oriented Predictive Control of Power Split Hybrid Electric Vehicle. *Mathematical Problems in Engineering*. 2020(Pt.7), 7024740.1-7024740.14. doi: 10.1155/2020/7024740.

Zhang, Y., Ma, R., Zhao, D., Huangfu, Y. & Liu, W. (2022) A Novel Energy Management Strategy Based on Dual Reward Function Q-learning for Fuel Cell Hybrid Electric Vehicle. *IEEE Transactions on Industrial Electronics*. 69(2), 1537-1547. doi: 10.1109/TIE.2021.3062273.

Zhao, T., Li, G., Pan, H. & Yuan, H. (2021) Dynamic characteristics analysis for vehicle parts based on parallel optimization algorithm with CUDA. *Engineering Computations*. 38(9), 3622-3642. doi: 10.1108/EC-04-2020-0232.