Journal of Cleaner Production 248 (2020) 119302

Contents lists available at ScienceDirect

Journal of Cleaner Production

journal homepage: www.elsevier.com/locate/jclepro

Energy optimization of electric vehicle's acceleration process based on reinforcement learning

Hongwen He [a, *], Jianfei Cao [a], Xing Cui [b]

[a] National Engineering Laboratory for Electric Vehicles, School of Mechanical Engineering, Beijing Institute of Technology, Beijing, 100081, China
[b] China North Vehicle Research Institute, Beijing, 100072, China

ARTICLE INFO

Article history:
Received 6 May 2019
Received in revised form
9 October 2019
Accepted 12 November 2019
Available online 13 November 2019

Handling editor:

Keywords:
Unmanned driving
Electric vehicles
Pedal control stratgey
Energy optimization
Q-learning
Deep Q-learning

ABSTRACT

Under the situation of unmanned driving, the energy consumption in an electric vehicle's acceleration process can be reduced by controlling the driving behavior. So in this paper, a pedal control strategy which could optimize the energy consumption of electric vehicle's acceleration process is proposed. The strategy is generated by the training results of reinforcement learning framework and the specific method of building such framework is discussed in details. Based on the training results of Q-learning-based algorithm, the relationship between the proportion of energy consumption reduction and vehicle's acceleration time is analyzed, which illustrates the energy-saving potential of the algorithm. In order to improve the control effect of the strategy, an updated algorithm framework based on Deep Q-learning (DQN) is proposed and an improved pedal's control strategy is obtained. Compared with the strategy obtained by Q-learning-based algorithm, the improved strategy not only achieves the same energy-saving effect, but also guarantees the stability of control effect, which is more suitable for actual use.

1. Introduction

The progress of vehicle technology is always accompanied by the pursuit of improving energy efficiency and promoting social sustainable development. Compared with traditional vehicle that equipped with internal combustion engines, electric vehicles show great potentials to solve the problems of energy shortage and environmental pollution. Usually, the energy consumption level of electric vehicles can be mainly improved by adjusting the energy management strategy. Thus, much related effort has been invested in the areas of battery management, braking energy recovery and etc. In addition, the driver behavior is also an important factor in affecting electric vehicles' energy consumption level due to its direct influence on the energy efficiency of vehicle's power system. Traditionally, vehicle's movement is dependent on driver's behavior and thus can not be affected by the vehicle's control system. However, with the recent rapid development of artificial intelligence in autonomous driving, an electric vehicle can also control its movement using the control system, and make it possible to reduce the energy consumption by controlling driving behavior, which also provides a feasible development direction for improving energy efficiency of electric vehicles.

1.1. Literature review

Many studies have focused on the optimization of energy consumption in the driving process of new energy vehicles especially electric vehicles (Zhang et al., 2012; Hung and Wu, 2012; Williamson et al., 2007). These studies include not only the optimization of vehicle's power distribution, but also the optimization of some key components connected within the transmission system to reduce energy consumption (Wang et al., 2016; He et al., 2012; Feng and Zhang, 2017). These studies show that through the rational design of energy optimization strategies, the working points of main power components can be optimized, thus enhancing energy efficiency during vehicles' driving. For a hybrid electric vehicle, the optimization objects mainly include the battery, the engine, the motor and the transmission system; while for a pure electric vehicle or a plug-in hybrid electric vehicle who mainly works under electric mode, those objects mainly include the motor and the battery. The former mainly involves improving the working efficiency of driving motor (Hung and Wu, 2012); the latter mainly

* Corresponding author.
E-mail address: hwhebit@bit.edu.cn (H. He).

https://doi.org/10.1016/j.jclepro.2019.119302
0959-6526/

includes controlling the thermal load of the power battery under extreme conditions (Panchal et al., 2017, 2018).

With the development of intelligent driving technology, vehicles can decide their driving behavior independently without manual intervention. The existing research on intelligent driving decision-making control mainly focuses on obstacle avoidance and path planning (Linhui et al., 2009; Azouaoui and Chohra, 2002; Chu et al., 2012), whereas few researchers have devoted their efforts in the process of energy consumption. Some scholars have carried out research on intelligent assistant driving. By actively controlling the acceleration and deceleration during vehicle driving process, vehicle's active safety could be guaranteed (Zheng et al., 2004; Tawari et al., 2014). These studies did not take vehicle's energy consumption into account as well. To our best knowledge, energy consumption is one of the core factors affecting the power economic performance of electric vehicles and energy consumption optimization deserves further attention in intelligent driving, especially for those intelligent vehicles based on pure electric classic platform (Fritsch and Liu-Henke, 2017; Murphey, 2008).

Under the condition of manned driving, it is nearly impossible to improve electric vehicles' energy consumption level by controlling driving behavior because the vehicle is controlled by human drivers; while under the condition of unmanned driving, it is possible. Similar to traditional vehicles, the energy consumption of electric vehicles under steady-state crushing driving conditions is generally relatively low. As an attempt to fill the gap, the focus of the study is put on the transient-state driving conditions including both accelerating and braking processes, which has also been a principle that many researchers consider when optimizing energy consumption (Borhan et al., 2009, 2012). For braking process, many studies on braking energy recovery technology for manned vehicles can be directly applied in unmanned vehicles (Cikanek and Bailey, 2002; Gao et al., 2001); while for accelerating process, relevant research is rare because the demand for research on energy consumption of vehicles' acceleration process just emerged with the development of autonomous driving technology.

Based on the research of many scholars, many optimization algorithms have been applied to optimizing vehicle's energy consumption. The classical algorithm is dynamic programming, which can calculate the global optimal solution of optimization problems under discrete conditions (Wang et al., 2015a; Gausemeier et al., 2010; Ozatay et al., 2014). In (Wang et al., 2015a), dynamic programming is used to solve the optimal power distribution relationship of a plug-in hybrid electric vehicle. Reference (Gausemeier et al., 2010) deals with the development of a method for multi-objective optimization of vehicle's velocity profiles. In (Ozatay et al., 2014), a dynamic programming solution for optimizing vehicle's driving behavior in spatial domain, which increases fuel economy of a passenger vehicle. It is commonly thought that dynamic programming could help to fully explore the energy-saving potential of the target vehicle but its online application requires several specific design for the costly demand of calculation time and space. In some cases, the combination of off-line calculation and online table-lookups is also an enlightened solution.

However, the off-line solution of dynamic programming still faces the problem of dimension disaster when computational accuracy is required to be continuously improved. To overcome such difficulty in calculation demand, several rule-based logics are proposed for online optimal control of vehicle's energy consumption, but the effect is limited (Williamson et al., 2007; Padmarajan et al., 2016). Besides these traditional optimization methods, reinforcement learning has been a hot research topic at present with the development of artificial intelligence technology. It is an important part of machine learning and suitable for solving continuous multi-step decision problems (Lillicrap et al., 2015; Menda et al., 2018; Ma et al., 2019). In (Xu et al., 2018), a reinforcement learning algorithm is applied to obtain the closed-loop optimal/suboptimal solutions of the control quantity in model prediction control; in (Ipek et al., 2008), a reinforcement learing approach is proposed to finish the self-optimizing memory control mission.

Reinforcement learning has also been applied to optimizing vehicle's energy consumption and vehicle's intelligent control system (Wu et al., 2018; Kober et al., 2013). One of the basic reinforcement learning framework is Q-learning algorithm and it can achieve self-learning optimization in discretizing state space and store in tabular form (Watkins and Dayan, 1992; Hirashima et al., 1999). When the state-space is small, Q-learning is effective; however as the complexity of problem increases, the corresponding state-space will also become larger and the computation time will increase exponentially so that the algorithm would not work so well with its slow convergence rate (Lange and Riedmiller, 2010; Tsitsiklis, 1994). For those optimization problems with a continuous state space, deep reinforcement learning, also called DQN (Deep Q Learning), can be adopted. In deep reinforcement learning, the state-action value matrix in Q-learning is replaced by a neural network, so this algorithm could make decisions in continuous state space (Lillicrap et al., 2015; Ma et al., 2019; Arulkumaran et al., 2017). At present the convergence of Q-learning algorithm has been guaranteed, but the convergence of DQN algorithm has not been proved directly (Singh et al., 2000; Tsitsiklis and Van Roy, 1997). The main improvement methods for DQN algorithm are to improve the convergence effect and reduce the time cost by designing reasonable operation rules (Van Hasselt et al., 2015; Wang et al., 2015b).

1.2. Contributions of the work

(1) Under the situation of unmanned driving, an energy consumption optimization strategy for electric vehicle's acceleration process is proposed. The strategy is generated by the reinforcement learning framework and ensures that the motor's working points could be properly controlled so that the energy consumption of vehicle's acceleration process would get reduced.

(2) The specific method of building such reinforcement learning framework for the optimization problem is discussed in details, which includes the definition of state, action and reward, the method of how to choose the appropriate training parameters, the way to define the environmental model (vehicle model) and how to adjust the algorithm so that it could reasonably balance different optimization objectives (dynamic performance or economic performance).

(3) The relationship between the proportion of energy consumption reduction and vehicle's acceleration time is analyzed by the training results of Q-learning-based algorithm, which illustrates the energy-saving potential of this algorithm. In order to improve the control effect of the strategy, an updated algorithm framework based on DQN is proposed with its initial training conditions being determined by referring to the training results of Q-learning-based algorithm. The updated control strategy is more stable and ensures that the change curve of motor's working points does not have sharp jump, which is more suitable for actual control.

1.3. Organization of this paper

The remainder is organized as follows. Section 2 introduces how the reinforcement learning framework is established. Section 3 shows the necessary model parameters for training the algorithm. In Section 4, the training results and discussion is presented. The conclusions are finally given in Section 5.

2. Establishment of reinforcement learning framework

The optimization work in this paper is based on reinforcement learning framework. A general reinforcement learning framework includes five main concepts: environnement, agent, state, action and reward. The relationship is illustrated as follows. The agent explores the environment by choosing different actions, and then the environment gives the agent a reward to evaluate that if the action is good or bad. If the action is good, reward will be high, otherwise it will be low. By exploring the environment and getting the corresponding reward, the agent will gradually determine a decision sequence with a higher average cumulative reward. When the average cumulative reward becomes stable, the algorithm will be considered to be convergent. In such case, the experience learned by the agent will be regarded as the final optimization result.

In this study, the definitions of these five concepts are as follows.

2.1. Environement

The main function of the environment part is to evaluate the action made by the agent. For an acceleration process, time consumption and energy consumption are the main two points. Here, a vehicle model is defined to reflect both time and energy consumption under different control actions.

In the acceleration process, vehicle's velocity varies from 0 to a definite target value. Assuming that the current velocity changes from v_0 to v_1, energy consumption and time consumption by the vehicle can be calculated as formula (1) and (2) show, in which $acc(v)$ represents the acceleration of the vehicle.

$$T_{consum} = \int_{v_0}^{v_1} \frac{1}{acc(v)} dv \tag{1}$$

$$E_{consum} = \int_{v_0}^{v_1} \frac{T_m n}{\eta acc(v)} dv \tag{2}$$

Vehicle's acceleration process accon a gradientless road can be described by Equation (2).

$$acc = \frac{dv}{dt} = \frac{g}{\delta}(D - f) \tag{3}$$

The dynamic factor $D$is defined in the following way.

$$D = \frac{F_t - F_w - F_f}{G} \tag{4}$$

The driving force $F_t$can be defined as:

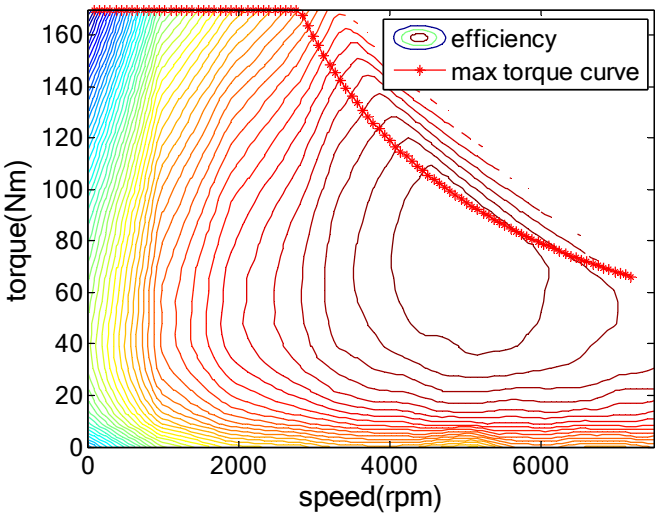


Fig. 1. Motor's efficiency map.

$$F_t = \frac{T_m i_0 i_g \eta_T}{r} \tag{5}$$

The gravity $G$is defined by the following formula.

$$G = mg \tag{6}$$

Air resistanceF_w is defined by the following formula. When the vehicle's velocity is low, the influence of it on vehicle's acceleration process can be neglected; when the vehicle's velocity is high, it should be calculated by equation (7).

$$F_w = \frac{C_d A v^2}{21.15} \tag{7}$$

Assuming that the rolling resistance coefficient f is constant, the rolling resistance is also constant in general. However, the rolling resistance can be considered to increase with the increasement of vehicle's velocity in the transient process from stationary to starting motion. Therefore, the mathematical expression of rolling resistance can be defined as follows. According to the relevant references, the velocity threshold can be defined as generally 1 km/h.

$$F_f = \begin{cases} vfG, & v \le v_{thrshold} \\ fG, & v > v_{thrshold} \end{cases} \tag{8}$$

The motor's output torque is a function of motor's speed and the pedal's travel as equation (9) shows. The specific relationship is determined by motor's external characteristic curve.

$$T_m = A_{\text{pedal}} \times T_{m\_\max}(n), n = \frac{v i_g i_0}{0.377 r} \tag{9}$$

The working efficiency of motor is calculated by the efficiency map, which is measured by experiment as Fig. 1 shows. (**Remark:** The data is provided by Nanjing Yuebo Power System Co., Ltd.)

Based on the above analysis, both time and energy consumption during the vehicle's acceleration process can be accurately calculated.

2.2. State and action

In the framework of reinforcement learning, the definition of action is closely related to the definition. In this study, we employ two methods to define the state and action.

(1) If the vehicle's velocity is chosen as the state variable, the pedal's travel is defined as the action variable. Specifically, the state is described as $s = (v)$ and the action is described as $a = (A_{pedal})$.
(2) If both vehicle's velocity and the pedal's travel is chosen as the state variables, the variation of pedal's travel is defined as the action variable. Specifically, the state is described as $s = (v, A_{pedal})$ and the action is described as $s = (v, \Delta A_{pedal})$.

Based on the former method, the number of actions increases with the discrete accuracy of pedal's travel getting improved; based on the latter method, obviously it does not, as long as the value of ΔA_{pedal} is limited in a certain range. However, the latter method only works in the situation that the action does not change too frequently, otherwise such definitions of state and action will limit the variation of action variables.

2.3. Reward

Reward is the key feedback from environment to agent (Kaelbling et al., 1995). The value of reward helps the agent to measure the merits of an action, so the strategy could get improved in the framework of reinforcement learning algorithm. In this study, it is the vehicle's energy consumption and acceleration time that are affected by the pedal's travel, so the reward is defined as a combination of the changes of time and energy consumption in an acceleration process, as is seen in Equation (10).

$$\begin{aligned} &R = (1-\lambda)R_1 + \lambda R_2, 0 \le \lambda \le 1 \\ &R_1 = -T_{consum} \\ &R_2 = -W_{consum} \end{aligned} \tag{10}$$

In equation (10), R_1 represents the reward corresponding to the time consumption and R_2 represents the reward corresponding to the energy consumption. The real reward value of R in the training process of reinforcement learning algorithm is defined as a linear combination of R_1 and R_2. Thus, λ, a coefficient, is defined to represent this combination relationship. λ is a real number with its value range being $\lambda \in [0, 1]$.

When $\lambda = 0$, the following equation is valid: $R = R_1 = -T_{consum}$. In such case, the agent trained by reinforcement learning algorithm takes time cost as the only optimization objective to control the working points of the motor.

When $\lambda = 1$, the following equation is valid: $R = R_2 = -W_{consum}$. In such case, the agent trained by reinforcement learning algorithm takes energy cost as the only optimization objective to control the working points of the motor.

When $0 < \lambda < 1$, the following equation is valid: $R = (1-\lambda)(-T_{consum}) + \lambda(-W_{consum})$. In such case, both energy cost and time cost are considered as the optimization objectives.

The value of λ directly affects the proportion of time cost and energy cost in the reward value. With the value being close to 0, the proportion of time cost in reward value gets larger; with the value being close to 1, the proportion of energy cost in reward value gets larger. When the proportion of energy cost in reward value, agent's exploring process and action selection process according to reward value may more likely to be affected by energy cost. As less energy cost which means better economic performance during the acceleration process corresponding to a higher reward value, the decision-making process of agent will be more inclined to select actions that could reduce energy consumption. Thus, with λ getting closer to 1, the agent pays more attention to the economic performance of the vehicle during acceleration process; otherwise, more attention will be paid to the dynamic performance.

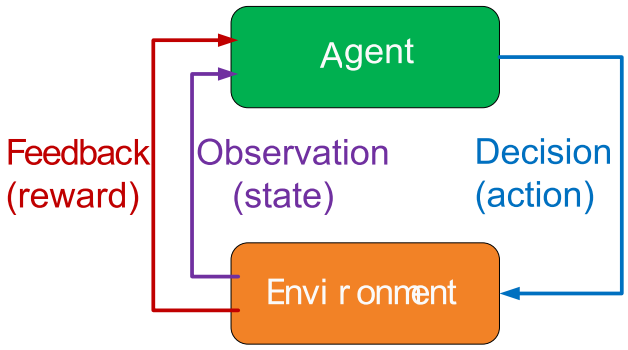


Fig. 2. Reinforcement learning framework.

2.4. Agent

Agent is the trained object to obtain the control strategy through multiple interactions with the environment to learn the control knowledge from cumulative experience. As a classical reinforcement learning algorithm (Q-learning), the core of the Q-learning agent is described as a matrix which can record state-action value and update its value directly through changing the corresponding element; for the agent in DQN, the core of the agent is represented by a neural network and the update of state-action value function is achieved through updating the parameters of the neural network.

Compared with DQN, the advantage of Q-learning is that the training results are easier to converge to the global optimzal solution. Although the convergence rate may not be fast and data efficient, the convergence of this method has been proved by relevant literatures (Hirashima et al., 1999; Watkins, 1989). In the case that Q-learning is chosen to train the agent, the algorithmic strategy is expressed as a $n_{state} \times n_{action}$ matrix. The update process of the algorithm is to continuously update the data in this matrix (Hirashima et al., 1999). When the algorithm converges, the matrix will finally record the average cumulative reward of each state and can determine the optimal action for each state, which can give the optimal pedal's control strategy.

As a typical discrete optimization method, Q-learning also inevitably has the problem of dimension disaster (Human-level control throu, 2015). As the discretization accuracy of state or action is improved, the iteration computation time of Q-learning will become very large, so it can hardly converge. Therefore, DQN is more suitable for such cases because of its generality and robustness with neural network as the core of the agent. When the number of state variables increases, the size of the neural network used to express the strategy will not change significantly, which means that DQN is more suitable for solving those optimization problems with large number of state variables.

2.5. Structure of the algorithm

The relationship between the above components could be

expressed simply as Fig. 2 shows.

Agent starts from an initial state, constantly explore the environment, and migrate from one state to another. In this process, the environment will give the agent some feedback information, which is defined as the reward. When the reward value is high, the next time the agent will continue to take the same action in a similar environment with a higher probability. When the reward value is low, the agent may reduce the probability of the choosing the action when facing the same condition.

From the perspective of algorithm execution, agent and environment could be regarded as two functions (The agent function is generally called as Q-function). The former outputs the decision quantity by inputting the current state; the latter outputs the state information after executing the decision and the feedback reward corresponding to the current action through the current state and decision-making information.

The specific mathematical form of the agent function could be either discrete or continuous, which also determines whether the reinforcement learing algorithm is discrete or continuous. In the following part, we present the details of the discrete reinforcement learning framework based on Q-learning (Algorithm 1) and the continuous reinforcement learning framework based on DQN (Algorithm 2).

2.5.1. Q-learning-based optimal algorithm

Q-learning-based algorithm is as shown in the following flowchart.

Algorithm1. Q-learning-based algorithm.

Updating process of Q-matrix

Initialize the state-action matrix Q arbitrarily

Repeat (for each episode):

Initialize s

Repeat (for each step of episode):

Choose action: $a=\begin{cases}\arg\max_a Q(s,a), probability:\varepsilon\\ random\ action\ a, probability:1-\varepsilon\end{cases}$

Take action a, observe reward R and next state s'

$Q(s,a)\leftarrow Q(s,a)+\alpha[R+\gamma\max_{a'}Q(s',a')-Q(s,a)], s\leftarrow s'$

Until $s=s_{terminal}$

Notice that MATLAB parallel computing toolbox is adopted to speed up the computing process to get a convergency result in less time. The toolbox provides up to 12 core parallel computing capabilities, which can speed up the learning process. This method could accelerate the convergence of the algorithm by using multi-agent to explore the environment, which is one of the current development directions of reinforcement learning (Tampuu et al., 2017). The whole parallel computer flow is shown in Fig. 3.

In Fig. 3, the computation process of each working pool is independent. Different workers represent a different calculation unit. The results of each workers are expressed as a different matrix,

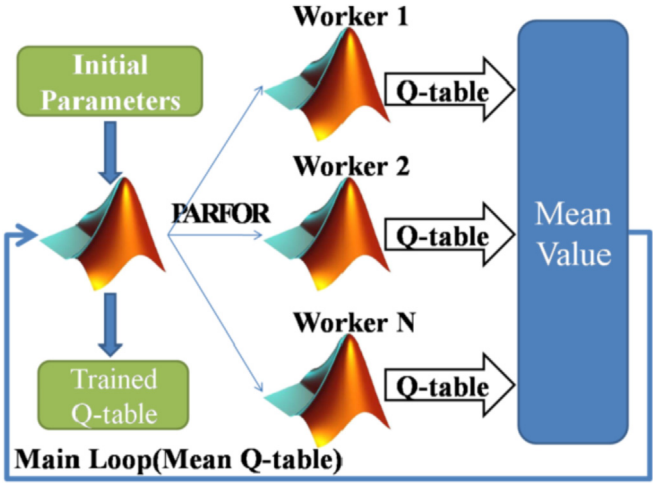


Fig. 3. Parallel computing flow diagram.

called sub-matrix. After several iteration computation, different results are expressed by different sub-matrix which has been obtained by different working pools. By setting the main Q-matrix as the average value of sub-matrix, the main Q-matrix can accumulate more knowledge in the same time than single core. After that, each sub-matrix will be reset within the value of main Q-matrix for the next iteration calculation. Based on such parallel computing framework, the computational speed will be significantly improved. If the core number is set to 12, the calculation time will be reduced by more than 90%, which would alleviate the pressure in computation time and improve the algorithm performance with Q-learning.

The convergence of Q-learning-based algorithm can be determined by equation (11). As the number of iterations increases, the $\overline{q}_{mean}$ becomes more and more stable. When $\overline{q}_{mean}$ does not change any more, the iterative process can be terminated.

$$\overline{q}_{mean}=\frac{1}{m\cdot n}\sum_{i=1}^{m}\times\sum_{j=1}^{n}(d\overline{Q}_k)_{ij},\begin{cases}q_{\max}=\max(Q_{ij}),1\le i\le m,1\le j\le n\\ q_{\min}=\min(Q_{ij}),1\le i\le m,1\le j\le n\\ \overline{Q}=2\dfrac{Q-q_{\min}}{q_{\max}-q_{\min}}-1\\ d\overline{Q}_k=\overline{Q}_k-\overline{Q}_{k+1},k=0,1,2...\end{cases}\tag{11}$$

2.5.2. DQN-based optimal algorithm

In order to improve the training effect of DQN algorithm, an improved version of DQN called Double DQN is chosen in this paper to construct the optimization algorithm. With two different Q-networks being used to complete the tasks of action selection and action evaluation respectively, double-DQN could minimize the impact of over-estimation problem. The whole framework of the algorithm is as the following flow-chart shows.

Algorithm 2. DQN-based algorithm.

Updating process of the Q network

Initialize replay memory D as capacity N

Initialize action-value function Q with random weights θ

Initialize target action-value function $\hat{Q}$ with weights $\hat{\theta} = \theta$

Initialize $n_{\text{train\_loop}} = 0$, $k = N_{\text{mini}}$

Repeat (for each episode):

Initialize s

Repeat (for each step of episode):

Choose action: $a = \begin{cases} \arg\max_a Q(s,a,\theta), \textit{probability} : \varepsilon \\ \textit{random action } a, \textit{probability} : 1-\varepsilon \end{cases}$

Take action a, observe reward R and next state s'

Define one exploratory experience as $d = (s,a,r,s')$

Store d in D

Sample random mini-batch $(d_1, d_2 ... d_k)$ from D,

$$\text{Set } y_k = \begin{cases} R_k & , s'_k = s_{\text{terminal}} \\ R_k + \gamma \hat{Q}(s'_k, \arg\max_a Q(s'_k, a; \theta_k); \hat{\theta}_k) & , s'_k \neq s_{\text{terminal}} \end{cases}$$

Perform a gradient descent step of α_{net} on $(y_k - Q(s_k, a_k, \theta_k))^2$ with respect to the network parameters θ

$$n_{\text{train\_loop}} = \begin{cases} n_{\text{train\_loop}} + 1, & n_{\text{train\_loop}} < C \\ 0, & \textit{otherwise} \end{cases}$$

If $n_{\text{train\_loop}} = C$, reset $\hat{Q} = Q$

Until $s = s_{\text{terminal}}$

The main difference between DQN-based algorithm and Q-learning-based algorithm is the expression form of Q function. Q function in DQN-based algorithm is expressed as a neural network instead of a two-dimensional matrix as that in Q learning-based algorithm. Therefore, the training and updating process of reinforcement learning algorithm is no longer a direct updating process of Q function, but an indirect updating by adjusting the parameters of the neural network.

Updating process of the network's parameters is realized by gradient descent method, which is an effective way for solving many optimization problems. In order to avoid the disadvantageous effect of random disturbance on parameters' updating, training process is performed with multiple sets of data rather than one single set, which is quite different from that in Q learning-based algorithm.

Such several sets of data together are called as a batch. Each batch of data is provided by the memory pool that stores each exploratory data. When the amount of data in memory pool is less than that of one batch, the algorithm only performs the process of agent exploring the environment, but not the process of parameter updating. When the amount of data in memory pool exceeds a certain limit, the data earlier entering into the memory pool will be discarded.

Fig. 4. Experimental vehicle.

3. Parameters' explanation

3.1. Parameters of the vehicle

Vehicle's parameters are used to construct the vehicle model in the algorithm framework. The specific data is given by the experimental vehicle provided by the author's laboratory (see Fig. 4). The vehicle parameters are listed in Table 1, and the driving motor and power battery's parameters are listed in Table 2.

3.2. Parameters of the Q-learning-based algorithm

The present parameters of reinforcement learning algorithm may have a serious effect on the training results. The parameters setting in Q-learning-based algorithm are given in Table 3.

Since the vehicle's velocity is chosen as one of the state variable, the number of state is proportional to the accuracy of vehicle's velocity as equation (12) describes.

$$n_{state} = \frac{v_{target} - v_{start}}{\Delta v} \tag{12}$$

In order to improve the calculation accuracy of vehicle model, the smaller the discrete value of velocity, the better; in order to reduce the training time cost of the algorithm, the larger the discrete value of velocity, the better. While discretizing the value of velocity with high resolution can improve the calculation accuracy of vehicle model, it can lead to the exponential increase in the training time. Under different target velocity, the relationship between time and energy consumption calculated by the vehicle model and discrete value of velocity Δv is shown in Fig. 5 (for ease of observation $\log\Delta v$ is used to represent Δv). As is shown, the calculation error of both time consumption and energy consumption increases with the increase of Δv. In the case of high resolution (Δv<1 km/h), the error is very small and almost unchanged; while in the opposite case (Δv>1 km/h), the error begins to increase sharply. Therefore, setting discrete value (Δv) as 1 km/h can attain a balance between the calculation accuracy and training time.

Table 1
Parameters of the vehicle model.

Parameters	Values	Parameters	Values
m (kg)	3500	i_g (/)	5.857
r(m)	0.447	i_0 (/)	2.604
f (/)	0.01	C_d (/)	0.65
Wheelbase(m)	2.65	A (m^2)	3.90

Table 2
Parameters of driving motor and power battery.

Parameters of motor	Values	Parameters of battery	Values
Max Power (kW)	50	Rated Voltage(V)	347.8
Speed (r/min)	2800/7200	Capacity (Ah)	50
Rated Voltage(V)	360	Sustained Current(A)	100
Max Torque (Nm)	170	Peak Current(A)	150

Table 3
Parameters of Q-learning-based algorithm.

Parameters	Implication	Value
α	learning rate	0.9
γ	discount rate	0.8
ε	greedy coefficient	0.9
$n_{ep\_max\_main}$	max episode in main loop	20
$n_{ep\_max\_pool}$	max episode in pool loop	$n_{state} \times n_{action}$
n_{state}	the number of state	adjustable
n_{action}	the number of action	adjustable
n_{pool}	the number of parallel pool	12

Table 4
Parameters of DQN-based algorithm.

Parameters	Implication	Value
n_{state}	the number of state	adjustable
n_{action}	the number of action	adjustable
α	learning rate of neural network	0.01
ε	greedy coefficient	0.6
N	size of memory pool	6000
N_{mini}	size of mini-batch	1024
C	update frequency of target network	per 50 steps

3.3. Parameters of the DQN-based algorithm

The parameters setting in DQN-based algorithm are given in Table 4.

In DQN algorithm, a neural network instead of Q-matrix is functioned as the core to explore the environment and learn the experience. As the neural network estimates the state-action value by function fitting, the increase of state variables will not affect the scale of the neural network, nor will it increase time cost of the algorithm significantly.

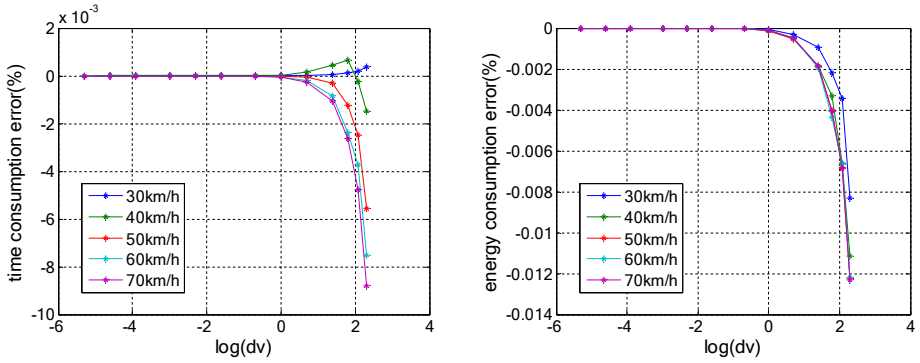
4. Training results and discussion

4.1. Results of Q-learning-based algorithm

The acceleration curve of the vehicle with pedal's travel equaling 100% is shown in Fig. 6. As is shown, the maximum velocity is about 80 km/h with acceleration time being 28s. Therefore, the acceleration process can be approximately divided into three different situations according to the target velocity, corresponding to different acceleration requirements: 30 km/h represents a low accelerated demand, 50 km/h represents a middle accelerated demand and 70 km/h represents a high accelerated demand.

By adjusting the coefficients λ, a set of different acceleration strategies will be obtained with different energy and time consumption as Figs. 7–9 show.

Within a certain range, energy consumption in an acceleration process decreases with the increase of time consumption, which



(a) time consumption error (b) energy consumption error

Fig. 5. The relation between vehicle model's time and energy consumption error and log (Δv).

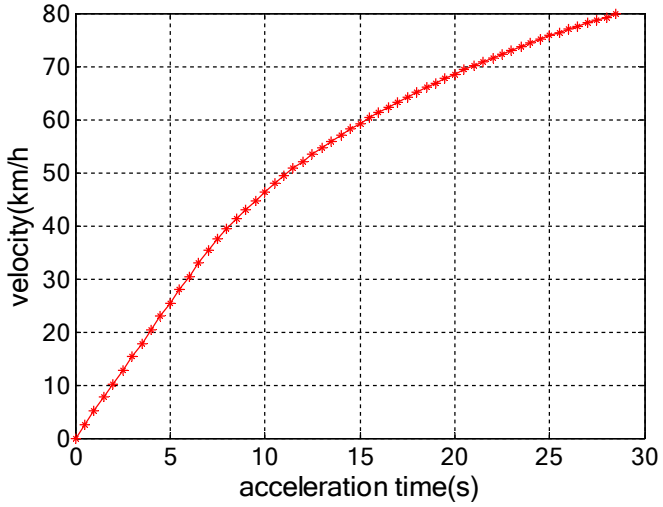


Fig. 6. Full-throttle-acceleration curve of the prototype vehicle.

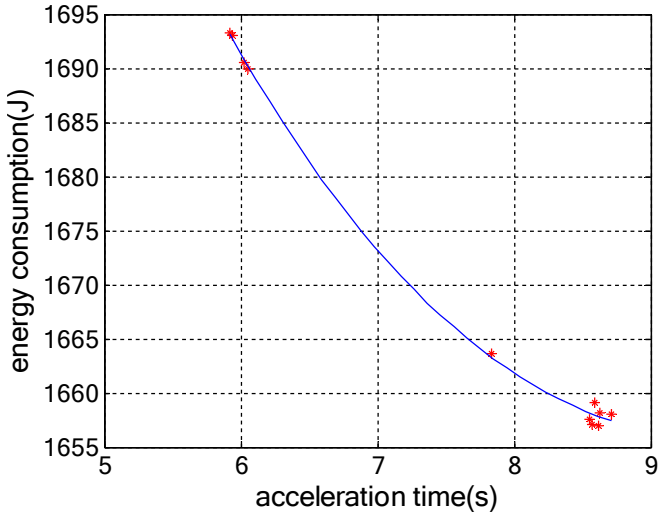


Fig. 7. The relation between energy and time consumption (target velocity being 30 km/h).

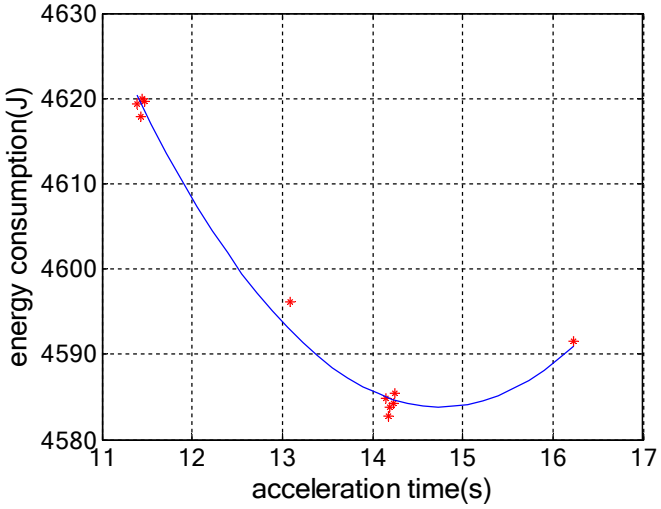


Fig. 8. The relation between energy and time consumption (target velocity being 50 km/h).

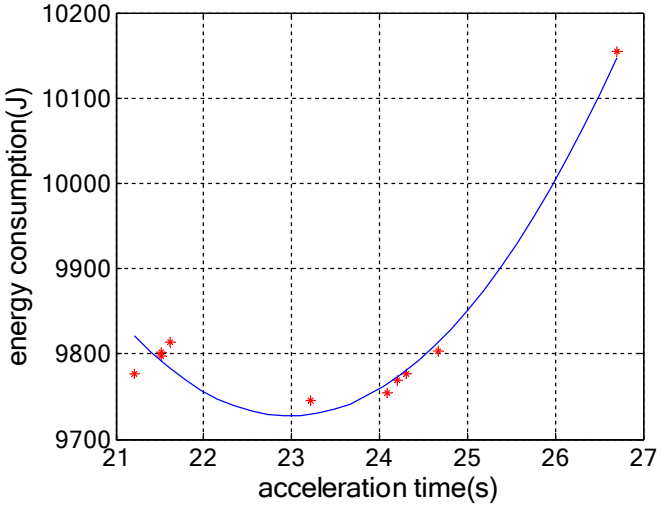
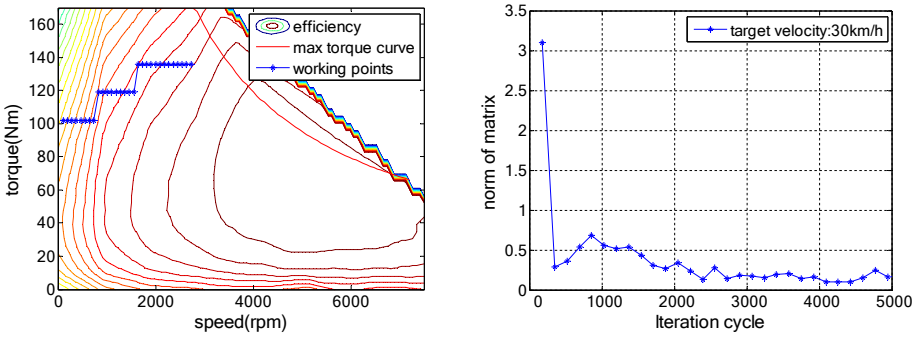


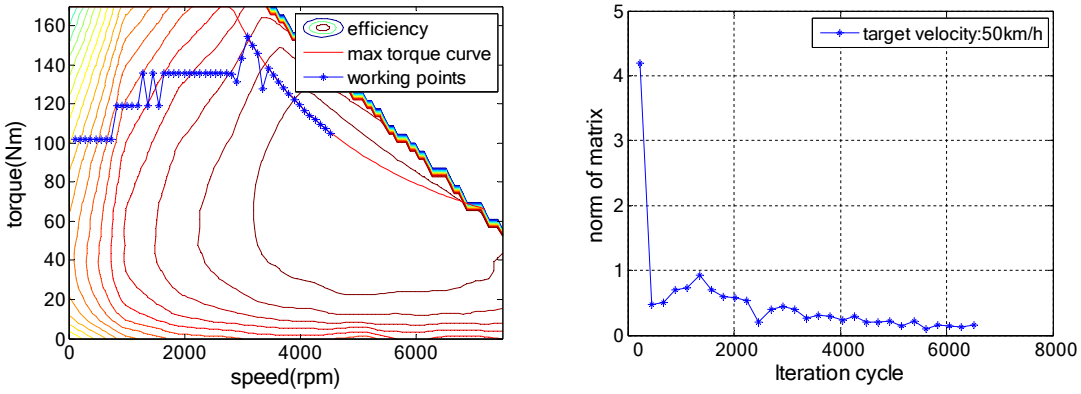
Fig. 9. The relation between energy and time consumption (target velocity being 70 km/h).

means the energy consumption during an acceleration process can be optimized. Different adjustment coefficients correspond to different control strategies; the strategy that reach the maximum energy reduction potential is considered to be the best strategy. The control effect can be shown in Figs. 10–12.



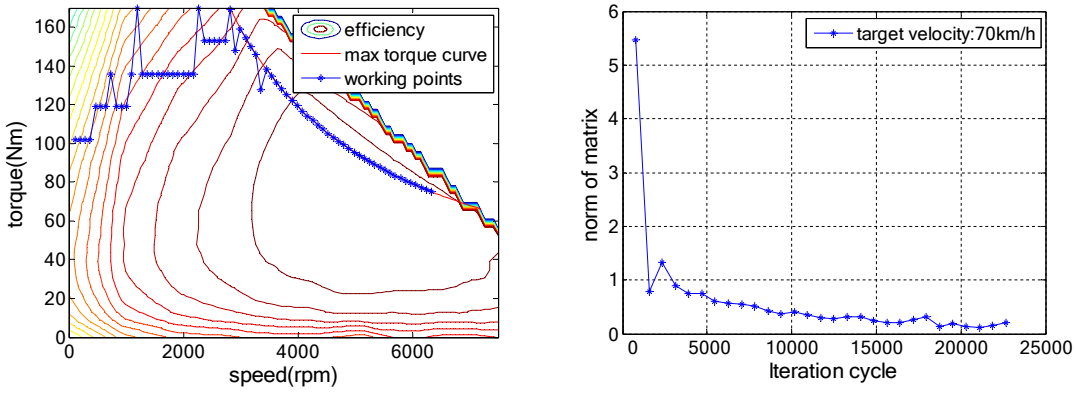
(a) motor's working points (b) convergence curve

Fig. 10. Optimal strategy with velocity being 30 km/h.



(a) motor's working points (b) convergence curve

Fig. 11. Optimal strategy with velocity being 50 km/h.



(a) motor's working points (b) convergence curve

Fig. 12. Optimal strategy with velocity being 70 km/h.

Table 5
Energy-saving potential of the control strategy under different target velocity.

Target velocity	30 km/h	50 km/h	70 km/h
energy reduction potential (%)	2.1	1	0.9
increment of acceleration time(s)	2.79	3.34	1.71
coefficientλ(/)	0.8	0.8	0.4
initial pedal's travel(/)	0.6	0.6	0.6

Based on equation (11), the iterative convergence curve is shown in Figs. 10(b), Fig. 11(b) and Fig. 12(b). These curves show that the Q-learning-based algorithm can converge effectively in three cases and generate stable control strategies. The control effect of the strategy is shown in Figs. 10(a), Fig. 11(a) and Fig. 12(a). As is shown, motor's working points are effectively controlled by the strategy. When the motor's speed is low, the control strategy adopts a lower driving torque; when the motor's speed is high, the control

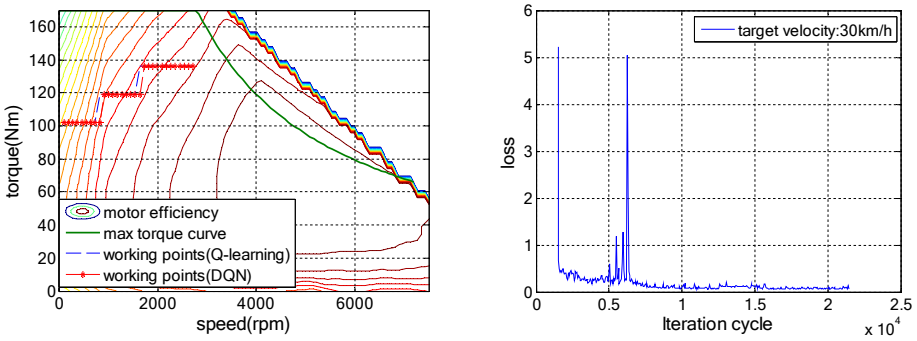
strategy adopts a higher driving torque.

Taking the time and energy consumption in the full pedal acceleration as a base case, the relationship between the increment of acceleration time and the percentage of energy reduction under different target velocity is illustrated in Table 5. With target velocity being 30 km/h, 50 km/h and 70 km/h, the energy reduction during an acceleration process can be 2.1%, 1% and 0.9%, which indicates that the energy-saving potential of the control strategy is greater following a lower target velocity.

Meanwhile, due to the influence of the adjustment coefficient, the dynamic performance reduction (increment of acceleration time) caused by energy saving can also be controlled in an acceptable range (less than 3.5s). Besides, the adjustment coefficient and the initial pedal's travel of the most energy-saving algorithms in the three cases are also recorded for further research in the following contents as Table 5 shows.

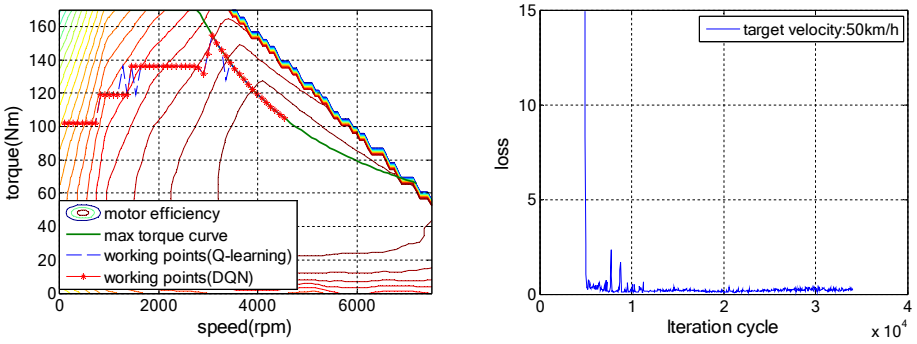
4.2. Results of DQN-based algorithm

Based on the test results of the state-discrete algorithm, the potential and control effect of the algorithm are given, but the control effect is still not so stable enough. In order to improve the stability of the pedal control strategy for electric vehicle's acceleration process and make the algorithm easy to realize online application, an updated pedal control strategy with continuous-state based on DQN framework is designed.



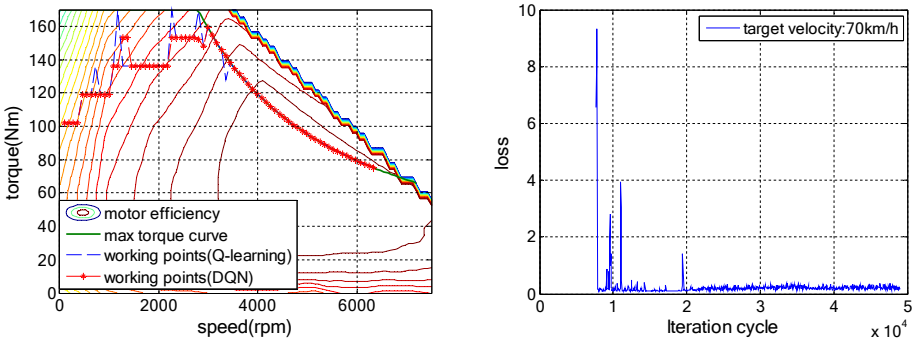
(a) motor's working points (b) convergence curve

Fig. 13. Performance of the strategy with target velocity being 30 km/h.



(a) motor's working points (b) convergence curve

Fig. 14. Performance of the strategy with target velocity being 50 km/h.



(a) motor's working points (b) convergence curve

Fig. 15. Performance of the strategy with target velocity being 70 km/h.

Table 6
Energy-saving potential of the control strategy under different target velocity.

target velocity		30 km/h	50 km/h	70 km/h
time(s)	DQN	8.7134	14.1461	23.2654
	Q-learning	8.6204	14.1767	23.2149
Energy(J)	DQN	1656.9	4581.9	9742.8
	Q-learning	1656.9	4582.8	9745.6

Table 7
Error comparison (Taking the algorithm obtained by Q-learning as the standard).

target velocity		30 km/h	50 km/h	70 km/h
error	time(s)	almost zero	−0.03059	0.05051
	energy(J)	almost zero	−0.8395	−2.748

Table 8
Descrpition of all the symbols.

Symbol	description	Unit
v	Velocity	km/h
acc	Acceleration	m/s^2
η	Efficiency	%
T_{consum}	Time consumption	s
E_{consum}	Energy consumption	J
g	Gravity acceleration	m/s^2
D	Dynamic factor	/
t	Time	s
f	Rolling resistance coefficient	/
F_t	Traction force	N
F_w	Air resistance force	N
F_f	Rolling resistance force	N
m	Mass of the vehicle	kg
G	Gravity of the vehicle	N
C_d	Air drag coefficient	/
A	Windward area	m^2
A_{pedal}	Pedal's travel	/
T_{m_max}	Motor's external characteristic torque	Nm
n	Motor's speed	r/min
r	Rolling radius of the vehicle	m

Compared with the discrete reinforcement learning framework, the following improvement measures have been taken.

(1) The action is defined as the increment of pedal's travel rather than pedal's travel, and the state is defined as the velocity and pedal's travel. In such case, motor's working points could change smoothly, which will also meet the requirement of actual control.
(2) In order to improve the stability of the algorithm, the dimension of state variables is extended with two more variables including cumulative time consumption $T_{cum} = \int_0^t T_{\text{consum}} dt$and cumulative energy consumption$E_{cum} = \int_0^t E_{\text{consum}} dt$. Although the dimension of state has increased, the time cost of the algorithm under DQN framework will not significantly increase.

In such case, action is finally defined as $a = (\Delta acc)$ and state is finally defined as $s = (v, acc, T_{cum}, E_{cum},)$.The performance of the strategy can be shown in Figs. 13–15. Figs. 13(a), Fig.14(a) and Fig.15(a) show the control effect of the trained strategy; Figs. 13(b), Fig.14(b) and Fig.15(b) show the convergence curve in the training process.

The convergence curves show that the proposed DQN-based algorithm converges in all the three cases (with target velocity being 30 km/h, 50 km/h and 70 km/h), which means the final strategy we got can provide a stable control under different situations. The trend of the convergence curve and the number of iterations is similar in these three cases, which suggests that the time cost of training the pedal control strategy based on this framework will not increase significantly with the change of target velocity.

Table 9
Descrpition of all the nomenclatures.

Nomenclature	Description
	Action in reinforcement learning
	State in reinforcement learning
R or r	Reward in reinforcement learning
	Weighting coefficient
	Dimension of state
	Dimension of action
	A value function, also called as Q function
	learning rate
	discount rate
	greedy coefficient
	max episode in main loop
	max episode in pool loop
	Dimension of state
	Dimension of action
	the number of parallel pool
	Learning rate of neural network
	Weights of the neural network
	Size of memory pool
	Size of mini-batch
	Renewal cycle

The change curve of motor's working points shows that the control effect trained by DQN-based algorithm is better than that by Q-learning-based algorithm. Compared with the latter (Q-learning based algorithm), motor's working points obtained by the former (DQN-based algorithm) is more stable with nearly no sharp jump in the change curve of motor's working points. Through the results, we can conclude that the pedal control strategy trained by DQN-based algorithm is more conducive to meet the requirement of actual control.

Energy and time consumption corresponding to the control strategy obtained by DQN-based algorithm and Q-learning algorithm are shown in Table 6. The error of time and energy consumption between the two strategies is very small as shown in Table 7. Thus, it can be concluded that compared to the strategy of Q-learning, the control strategy of DQN-based algorithm can obtain a similar performance in terms of energy saving potential and dynamic performance degradation.

Meanwhile, since parts of initial training conditions (including initial value of pedal's travel and the adjustment coefficient) of DQN-based algorithm are determined by referring to the training results of Q-learning-based algorithm, such training process is easier to converge than those training process with random initial conditions.

5. Conclusions

In this paper, an energy consumption optimization strategy for electric vehicle's acceleration process is proposed, aiming at reducing the energy consumption by controlling the change of acceleration pedal's travel reasonably during an acceleration process. The strategy is suitable to be applied in unmanned electric vehicles. The discrete reinforcement learning algorithm (Q-learning) and the state-continuous reinforcement learning algorithm (DQN) are combined together to train the strategy. The former is used for training and getting a basic strategy; the latter is used for improving and upgrading the strategy.

The results of case study show that energy consumption of

acceleration process can be effectively reduced by the strategy's control with the acceleration time being extending within an appropriate range. Especially under low target-velocity conditions, energy-saving potential is better. The strategy obtained by the two algorithms can achieve almost the same energy-saving potential. While the corresponding change process of motor's working points of DQN-based algorithm is more stable and could provide a more stable control strategy that is suitable for practical application. The Symbol and Nomenclature appled in this article are listed in Table 8 and Table 9.

Acknowledgments

This project is supported by National Key R&D Program of China. (Grand No. 2018YFB0105900) in part, and supported by the National Natural Science Foundation of China (grant number 51675042).

References

Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A., 2017. A brief survey of deep reinforcement learning. IEEE Signal Process. Mag. 34 (6), 26–38.
Azouaoui, O., Chohra, A., 2002. Soft computing based pattern classifiers for the obstacle avoidance behavior of intelligent autonomous vehicles. Appl. Intell. 16 (3), 249–272.
Borhan, H.A., Vahidi, A., Phillips, A., Kuang, M., Kolmanovsky, I., 2009. Predictive energy management of a power-split hybrid electric vehicle. In: 2009 American Control Conference. IEEE, pp. 3970–3976.
Borhan, H., Vahidi, A., Phillips, A.M., Kuang, M., Kolmanovsky, I., Di Cairano, S., 2012. MPC-based energy management of a power-split hybrid electric vehicle. In: IEEE Transactions on Control Systems Technology, vol 20. IEEE, pp. 593–603, 3.
Chu, K., Lee, M., Sunwoo, M., 2012. Local path planning for off-road autonomous driving with avoidance of static obstacles. IEEE Trans. Intell. Transp. Syst. 13 (4), 1599–1616.
Cikanek, S.R., Bailey, K.E., 2002. Regenerative braking system for a hybrid electric vehicle. In: Proceedings of the 2002 American Control Conference, vol 4. IEEE, pp. 3129–3134.
Feng, Y., Zhang, C., 2017. Core loss analysis of interior permanent magnet synchronous machines under SVPWM excitation with considering saturation. Energies 10 (11), 1716.
Fritsch, M., Liu-Henke, X., 2017. Optimization of energy consumption by using an intelligent assistance system for an electric vehicle. In: 2017 Twelfth International Conference on Ecological Vehicles and Renewable Energies. IEEE, pp. 1–9.
Gao, H., Gao, Y., Ehsani, M., 2001. A neural network based SRM drive control strategy for regenerative braking in EV and HEV. In: IEEE International Electric Machines and Drives Conference. IEEE, pp. 571–575.
Gausemeier, D.W.I.S., Karl-Peter, J.I., Ansgar, T.I.H., 2010. Multi-objective optimization of a vehicle velocity profile by means of dynamic programming. IFAC Proceedings Volumes 43 (7), 366–371.
He, H., Xiong, R., Guo, H., Li, S., 2012. Comparison study on the battery models used for the energy management of batteries in electric vehicles. Energy Convers. Manag. 64 (4), 113–121.
Hirashima, Y., Iiguni, Y., Inoue, A., Masuda, S., 1999. Q-learning algorithm using an adaptive-sized Q-table. In: Proceedings of the 38th IEEE Conference on Decision and Control, vol 2. IEEE, pp. 1599–1604.
Human-level control through deep reinforcement learning. Nature 518 (7540), 2015, 529–533.
Hung, Y.H., Wu, C.H., 2012. An integrated optimization approach for a hybrid energy system in electric vehicles. Appl. Energy 98 (1), 479–490.
Ipek, E., Mutlu, O., Martínez, J.F., Caruana, R., 2008. Self-optimizing memory controllers: a reinforcement learning approach. Computer Architecture News 36 (3), 39–50.
Kaelbling, L.P., Littman, M.L., Moore, A.W., 1995. An introduction to reinforcement learning. In: The Biology and Technology of Intelligent Autonomous Agents. Springer, Berlin, Heidelberg, pp. 90–127.
Kober, J., Bagnell, J.A., Peters, J., 2013. Reinforcement learning in robotics: a survey. Int. J. Robot. Res. 32 (11), 1238–1274.
Lange, S., Riedmiller, M., 2010. Deep auto-encoder neural networks in reinforcement learning. In: The 2010 International Joint Conference on Neural Networks. IEEE, pp. 1–8.
Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Wierstra, D., 2015. Continuous Control with Deep Reinforcement Learning arXiv preprint arXiv: 1509.02971.
Linhui, L., Mingheng, Z., Lie, G., Yibing, Z., 2009. Stereo vision based obstacle avoidance path-planning for cross-country intelligent vehicle. In: 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, vol 5. IEEE, pp. 463–467.
Ma, Y., Zhu, W., Benton, M.G., Romagnoli, J., 2019. Continuous control of a polymerization system with deep reinforcement learning. J. Process Control 75, 40–47.
Menda, K., Chen, Y., Grana, J., Bono, J., Tracey, B., Kochenderfer, M., Wolpert, D., 2018. Deep reinforcement learning for event-driven multi-agent decision processes. IEEE Trans. Intell. Transp. Syst. 20 (4), 1259–1268.
Murphey, Y.L., 2008. Intelligent vehicle power management: an overview. In: Computational Intelligence in Automotive Applications. Springer, Berlin, Heidelberg, pp. 169–190.
Ozatay, E., Onori, S., Wollaeger, J., Ozguner, U., Rizzoni, G., Filev, D., Michelini, J., Cariano, S., 2014. Cloud-based velocity profile optimization for everyday driving: a dynamic-programming-based solution. IEEE Trans. Intell. Transp. Syst. 15 (6), 2491–2505.
Padmarajan, B.V., McGordon, A., Jennings, P.A., 2016. Blended rule-based energy management for PHEV: system structure and strategy. IEEE Trans. Veh. Technol. 65 (10), 8757–8762.
Panchal, S., Mcgrory, J., Kong, J., Fraser, R., Fowler, M., Dincer, I., Agelin-Chaab, M., 2017. Cycling degradation testing and analysis of a LiFePO4 battery at actual conditions. Int. J. Energy Res. 41 (15), 2565–2575.
Panchal, S., Mathew, M., Dincer, I., Agelin-Chaab, M., Fraser, R., Fowler, M., 2018. Thermal and electrical performance assessments of lithium-ion battery modules for an electric vehicle under actual drive cycles. Electr. Power Syst. Res. 163, 18–27.
Singh, S., Jaakkola, T., Littman, M.L., 2000. Convergence results for single-step on-PolicyReinforcement-learning algorithms. Mach. Learn. 38 (3), 287–308.
Tampuu, A., Matiisen, T., Kodelja, D., Kuzovkin, I., Korjus, K., Aru, J., Vicente, R., 2017. Multiagent cooperation and competition with deep reinforcement learning. PLoS One 12 (4), e0172395.
Tawari, A., Sivaraman, S., Trivedi, M., Shannon, T., Tippelhofer, M., 2014. Looking-in and looking-out vision for Urban Intelligent Assistance: estimation of driver attentive state and dynamic surround for safe merging and braking. In: Intelligent Vehicles Symposium Proceedings. IEEE, pp. 115–120.
Tsitsiklis, J.N., 1994. Asynchronous stochastic approximation and Q-learning. Mach. Learn. 16 (3), 185–202.
Tsitsiklis, J., Van Roy, B., 1997. An analysis of temporal-difference learning with function approximation. IEEE Trans. Autom. Control 42 (5), 674–690.
Van Hasselt, H., Guez, A., Silver, D., 2015. Deep reinforcement learning with double Q-learning. Computer Science arxiv preprint arxiv:1509.06461. http://arxiv.org/abs/1509.06461.
Wang, X., He, H., Sun, F., Zhang, J., 2015. Application study on the dynamic programming algorithm for energy management of plug-in hybrid electric vehicles. Energies 8 (4), 3225–3244.
Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., De Freitas, N., 2015. Dueling Network Architectures for Deep Reinforcement Learning arXiv preprint arXiv:1511.06581.
Wang, C., He, H., Xiong, R., Zhang, Y., 2016. A novel efficiency modeling method for a DC-DC converter in the hybrid energy storage system for electric vehicles. Energy Procedia 88, 935–939.
Watkins, C., 1989. Learning from Delayed Rewards. King's College, London.
Watkins, C.J.C.H., Dayan, P., 1992. Q-learning. Mach. Learn. 8 (3–4), 279–292.
Williamson, S.S., Emadi, A., Rajashekara, K., 2007. Comprehensive efficiency modeling of electric traction motor drives for hybrid electric vehicle propulsion applications. IEEE Trans. Veh. Technol. 56 (4), 1561–1572.
Wu, J., He, H., Peng, J., LI, Y., Li, Z., 2018. Continuous reinforcement learning of energy management with deep Q network for a power split hybrid electric bus. Appl. Energy 222, 799–811.
Xu, X., Chen, H., Lian, C., Li, D., 2018. Learning-based predictive control for discrete-time nonlinear systems with stochastic disturbances. IEEE transactions on neural networks and learning systems 29 (12), 6202–6213.
Zhang, P., Qian, K., Zhou, C., Stewart, B.G., 2012. A methodology for optimization of power systems demand due to electric vehicle charging load. IEEE Trans. Power Syst. 27 (3), 1628–1636.
Zheng, N., Tang, S., Cheng, H., Li, Q., Lai, G., Wang, F., 2004. Toward intelligent driver-assistance and safety warning system. IEEE Intell. Syst. 19 (2), 8–11.