SCIENCE CHINA
Technological Sciences

•Article•

March 2022 Vol.65 No.3: 529–540
https://doi.org/10.1007/s11431-021-1994-7

Ecological cruising control of connected electric vehicle: A deep reinforcement learning approach

WANG Qun[1], JU Fei[1], ZHUANG WeiChao[2] & WANG LiangMo[1*]

[1] *School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China;*
[2] *School of Mechanical Engineering, Southeast University, Nanjing 211189, China*

Received July 19, 2021; accepted December 30, 2021; published online January 28, 2022

Ecological cruising control methods of vehicles have been extensively studied to further cut down energy consumption by optimizing vehicles' speed profiles. However, most controllers cannot be put into practical application because of future terrain data requirements and excessive computational demand. In this paper, an eco-cruising strategy with real-time capability utilizing deep reinforcement learning is proposed for electric vehicles (EVs) propelled by in-wheel motors. The deep deterministic policy gradient algorithm is leveraged to continuously regulate the motor torque in response to road elevation changes. By comparing the proposed strategy to the energy economy benchmark optimized with dynamic programming (DP), and traditional constant speed (CS) strategy, its learning ability, optimality, and generalization performance are verified. The simulation results show that without a priori knowledge about the future trip, the proposed strategy provides 3.8% energy saving compared with the CS strategy. It also yields a smaller gap than the globally optimal solution of DP. By testing on other driving cycles, the trained strategy reveals good generalization performance and impressive computational efficiency (about 2 ms per simulation step), making it practical and implementable. Additionally, the model-free characteristic of the proposed strategy makes it applicable for EVs with different powertrain topologies.

eco-cruising, speed optimization, deep reinforcement learning, electric vehicle, optimal control

Citation: Wang Q, Ju F, Zhuang W C, et al. Ecological cruising control of connected electric vehicle: A deep reinforcement learning approach. Sci China Tech Sci, 2022, 65: 529–540, https://doi.org/10.1007/s11431-021-1994-7

1 Introduction

According to the United States Energy Information Administration, the transportation sector consumed approximately 26% of energy use in the United States in 2020 [1]. Among diverse energy sources, petroleum products contribute about 90% of the U.S. transportation sector's energy consumption. The exhaustion of non-renewable resources and growing environmental issues necessitate the reduction of fuel consumption of ground vehicles and greenhouse gas emissions. However, vehicle electrification offers an alternative way to cut down the dependency on fossil fuels. Consequently, electric vehicles (EVs) are regarded as a crucial tool in emissions reduction and energy conservation [2]. In addition to the development of new powertrain systems with higher efficiency, eco-driving, which is the optimization of vehicle longitudinal dynamics, has been regarded as a feasible method for reducing the impact of transportation on the environment over the past few years [3]. According to the NEXTCAR program [4], eco-driving can improve energy efficiency to about 11.4%, with relatively higher commercialization potential. The eco-driving controller optimizes the driving speed profile to enable the most energy-efficient operation of the vehicle. Recently, the penetrations of connection and automation techniques have provided vehicles with access to surrounding information and future driving conditions, thus, broadening the possibilities of speed planning for EVs [5]. Since daily driving largely comprises

*Corresponding author (email: liangmo@njust.edu.cn)

cruising, many related studies emphasized cruising scenarios.

Generally, eco-driving strategies are classified into rule-based, optimization-based, and learning-based strategies. Rule-based methods are the most widely used strategies due to their simplicity and real-time capability [6]. Moreover, input state variables can be mapped into corresponding output control variables using predefined thresholds and logic rules. However, limited optimality, massive calibration efforts, and poor flexibility impede its further applications [2].

To solve the cruising speed profile with optimal energy consumption characteristics, researchers usually construct the energy consumption of a vehicle as an optimal control problem (OCP). The optimal energy consumption problem is a nonlinear optimization problem with time-varying constraints. Optimization methods can be further subdivided into analytical and numerical-based ones. A typical analytical method called Pontryagin's minimum principle (PMP) is used to derive the optimal control law with respect to vehicle speed. Saerens and Van den Bulck [7] used PMP to obtain the driving principle with the least fuel consumption for a simplified vehicle model, which was treated as a point mass. The fuel-saving potential of a passenger car was exploited by employing PMP to analytically derive an optimal periodic control method [8]. Dynamic programming (DP) is another method used to numerically calculate the global optimal solutions [9]. Considering the waiting queue at signalized intersections, a collaborative eco-driving strategy was developed to cut down the overall fuel bill using time-based DP to obtain the energy-optimal velocity [10]. Zhuang et al. [11] developed a hierarchical framework to enable eco-cruising on slope-varying highways. They used DP to calculate the optimal vehicle speed considering energy efficiency and battery aging. However, DP does not scale well to real-time implementation since the computational cost grows exponentially with state complexity [12]. In light of this, real-time optimization approaches, such as model predictive control (MPC), have been proposed [13]. For a dual-mode hybrid electric vehicle (HEV), Xiang et al. [14] proposed an MPC-based control method with adaptive Markov-chain prediction. An improvement of about 16% in fuel economy can be obtained against a rule-based strategy. With the penetration of connected HEVs, Zhuang et al. [15] proposed an MPC-based cooperative control scheme to achieve safe and efficient platoon formation. Simulation results show that the proposed cooperative control strategy can achieve safe and efficient platoon formation. Nevertheless, the above methods have limitations, such as great computation expense, inferior optimality, and less adaptability to complex driving environments.

More recently, the emerging learning-based methods have offered a promising solution, e.g., reinforcement learning (RL) algorithms. In contrast to supervised learning that learns from labeled data, RL algorithms can obtain the control policy from raw observation input and scalar reward feedback directly [16,17]. RL has been widely used in several fields, such as lane-change decision-making [18], power-split for HEVs [19], and on-ramp merging [20]. The application of RL in eco-driving has recently attracted significant attention. Shi et al. [21] employed Q-learning to improve traffic performance and reduce exhaust emissions at signalized intersections. However, in this tabular method, the discretization requirements of state and action spaces may lead to the curse of dimensionality, making it challenging to handle situations with complex inputs and outputs [22]. Therefore, combined with deep learning, deep RL (DRL) algorithms can partially or completely eliminate the need for discretization. For connected and automated vehicles navigating through signalized intersections, Guo et al. [23] proposed a hybrid eco-driving control framework comprising of longitudinal (accelerate/brake) and lateral (lane-change) operational control. Two representative RL algorithms were used to control the longitudinal and lateral maneuvers simultaneously, saving energy by about 46% with traveling time slightly or not at all lengthened. In ref. [24], a partially observable Markov decision process was used to formulate the eco-driving problem solved using the proximal policy optimization algorithm. The controlled vehicle can autonomously pass signalized intersections and enhance the average fuel efficiency by 17.4% while maintaining comparable travel time.

However, most of the above studies presume that vehicles drive on flat routes, ignoring that road grade factors have a remarkable influence on energy consumption. Results show that overall fuel economy on flat routes is estimated by about 15%–20% better than that on hilly routes [25]. Lee et al. [26] employed a model-based Q-learning algorithm to realize eco-cruising for EVs considering road slope. However, Q-learning can only deal with discrete action space, inevitably leading to discretization error. Thus, in this paper, speed profile optimization for EVs considering road slope using deep deterministic policy gradient (DDPG) was investigated. Integrating the Actor-Critic (AC) architecture [27], the proposed DDPG algorithm can continuously output a deterministic control signal, which is more suitable for realistic vehicle control.

The main contributions of this paper are as follows. First, a DRL-based eco-cruising strategy with continuous action space is systematically designed to optimize the driving torque toward saving energy. Second, considering the huge impact of road slope on energy consumption, an energy-efficient vehicle speed profile is obtained without requiring prior driving or terrain information. Finally, the generalization performance and real-time capability of the proposed strategy are evaluated on testing driving cycles. To the best

of our knowledge, this is the first study to apply the DDPG algorithm to realize eco-cruising considering road slope, and its potential of practical application can also be guaranteed.

The remainder of this paper is structured as follows. Section 2 describes the dynamics modeling of powertrain components and the simulation environment. Section 3 elaborates the design of the DRL-based eco-cruising strategy. Section 4 presents the simulation results against two benchmark strategies, where the generalization evaluation is also discussed. Finally, Section 5 presents the conclusion and future work.

2 Modeling of electric vehicle and simulation environment

In this section, an energy-oriented simulation model of EVs is firstly constructed, including the vehicle longitudinal dynamics, modeling of the in-wheel motor (IWM), and battery dynamics. Then, three segments of the real-world urban route are selected as our driving environment.

Figure 1 shows the vehicle configuration adopted as a two-wheel independent drive (2WID) EV. In this configuration, the motor is integrated into the wheel to eliminate transmission losses and simplify mechanically complex components. To further simplify the model, the converter's efficiency is represented as a constant value 1. Table 1 presents detailed information about the vehicle and its components.

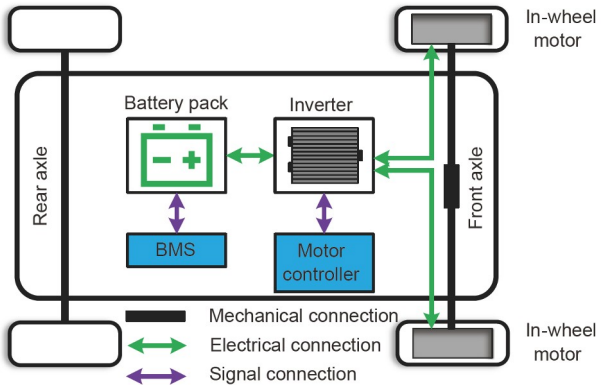

Figure 1 (Color online) Architecture of EV with IWMs.

Table 1 Specifications of the 2WID EV

Components	Parameters	Symbol	Value
Vehicle	Curb weight	m	2500 kg
	Tire radius	R_{tire}	0.3 m
	Frontal area	A_{f}	1.8 m^2
IWM	Maximum power	$P_{\max}$	76 kW
	Maximum torque	$T_{\max}$	1250 N m
	Maximum speed	$\omega_{\max}$	1600 r min^{-1}
Battery	Battery capacity	Q_{batt}	125 Ah
	Open-circuit voltage	V_{oc}	368 V

2.1 Vehicle longitudinal dynamics

As shown in eq. (1), the vehicle's motion is governed by the regular longitudinal dynamics model. The vehicle resistance consists of four parts: rolling resistance F_{f}, aerodynamic drag F_{w}, gradient resistance F_{i}, and inertia force F_{j}.

$$\begin{aligned} F_{\text{d}} &= F_{\text{f}} + F_{\text{w}} + F_{\text{i}} + F_{\text{j}}, \\ F_{\text{f}} &= m \cdot g \cdot f \cdot \cos\theta, \\ F_{\text{w}} &= \frac{1}{2} \cdot C_{\text{d}} \cdot A_{\text{f}} \cdot \rho \cdot v^2, \\ F_{\text{i}} &= m \cdot g \cdot \sin\theta, \\ F_{\text{j}} &= m \cdot a, \end{aligned} \tag{1}$$

where F_{d} denotes the request driving force, m denotes the curb weight, g denotes the gravitational acceleration, f denotes the rolling resistance coefficient, θ denotes the road grade, C_{d} denotes the aerodynamic coefficient, A_{f} denotes the frontal area, ρ denotes the air density, v denotes the instantaneous car speed, a denotes the car acceleration.

2.2 IWM model

A quasi-static technique is used to model the energy consumption of the power unit. The IWM adopted here is PD18 from Protean Electric [28], a permanent magnet synchronous motor. Figure 2 shows the efficiency map of PD18 based on the bench experiment data. The energy efficiency η_{e} of PD18 can be obtained from the interpolation of its rotational speed ω_{m} and torque T_{m} (positive during driving and negative during braking). The electric power P_{m} consumed by IWM can be calculated using the following equation:

$$P_{\text{m}} = T_{\text{m}} \cdot \omega_{\text{m}} \cdot \eta_{\text{e}}^{k}, \tag{2}$$

where the superscript k indicates working status. When the torque of IWM is positive (k=1), it works as a motor, converting electric power into mechanical power. Conversely, when the torque of IWM is negative (k=−1), it works as a generator, recovering braking energy.

2.3 Battery dynamics

The equivalent circuit model, consisting of open-circuit voltage and internal resistance, is used to represent battery dynamics (Figure 3). Due to its extensive application on energy management involving optimization methods [29–32] and learning methods [33–36], it is supposed to be sufficient for this research. Note that the thermal effect of the battery is not considered in this paper.

In Figure 3, P_{batt} denotes the power at the battery terminals,

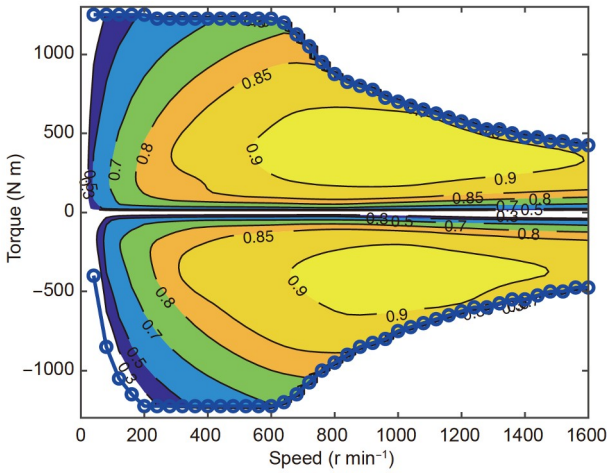

Figure 2 (Color online) Efficiency map of PD18.

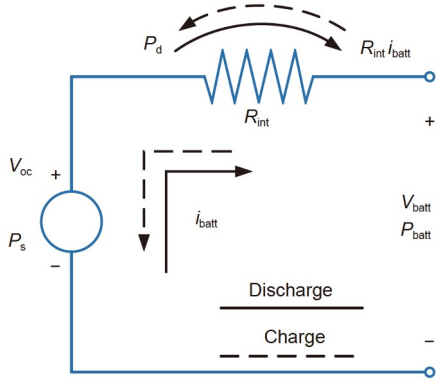

Figure 3 (Color online) Equivalent circuit battery cell model.

i_{batt} denotes the battery current (positive during discharge phase), V_{oc} and R_{int} denote open-circuit voltage and internal resistance of the battery, respectively.

Then, the battery power P_{batt} can be expressed as

$$P_{batt} = P_s - P_d = V_{oc} \cdot i_{batt} - {i_{batt}}^2 \cdot R_{int}, \tag{3}$$

where P_d is the dissipative power of the battery. Eq. (3) is solved with respect to i_{batt}. The solution can be expressed as

$$i_{batt} = \frac{V_{oc} - \sqrt{V_{oc}^2 - 4P_{batt}R_{int}}}{2R_{int}}. \tag{4}$$

The dynamics of the $LiFePO_4$ battery pack is governed by the following equation:

$$\frac{d}{dt}SOC = -\frac{i_{batt}}{Q_{batt}}, \tag{5}$$

where Q_{batt} is the battery capacity.

Applying eq. (4) to reformulate eq. (5), the differential equation of state of charge (SOC) with respect to P_{batt} can be achieved by

$$\dot{SOC} = -\frac{i_{batt}}{Q_{batt}} = -\frac{V_{oc} - \sqrt{V_{oc}^2 - 4P_{batt}R_{int}}}{2Q_{batt}R_{int}}. \tag{6}$$

2.4 Driving environment

Three pieces of road segments are selected as the driving environment. They are extracted from a real-world urban route in Nanjing, as shown in Figure 4. The length of each segment is 10 km. With access to a global positioning system (GPS), latitude, longitude, and altitude can be obtained. A five-point smoothing method is used to smooth the altitude profile for better visualization of the simulation results. Figure 5 shows the altitude profiles of the three segments and their corresponding slope variations. Driving cycle A is used to train the DRL-based eco-cruising strategy, whereas driving cycles B and C are used to test the generalization of the trained strategy.

3 DRL-based eco-cruising strategy

This section aims to develop a learning-based eco-cruising control strategy for EVs, minimizing the electricity consumption between two designated locations under speed constraints. Figure 6 shows a schematic overview of eco-cruising considering road slope. With access to GPS and high-definition maps, along with onboard sensors, the ego vehicle can obtain its real-time velocity, altitude, and slope of the road. Incorporating the multi-source information with the vehicle dynamics, the eco-cruising controller optimizes motor torque, maximizing the efficiency over the entire journey under certain speed limits.

3.1 OCP formulation

The eco-cruising problem of electric vehicles is framed as a nonlinear optimization formulation to minimize the overall energy consumption of the battery between two designated locations. The optimization object is constructed as follows:

$$\min J = \int_0^T \dot{SOC}(x(t), u(t))dt, \tag{7}$$

subject to

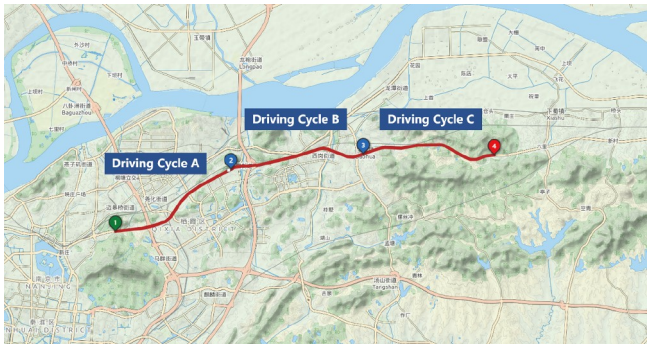

Figure 4 (Color online) Terrain profile of the selected real-world route.

$$
\begin{aligned}
&\dot{v} = F(x(t), u(t)),\\
&\dot{s} = v,\\
&v_{\min} \le v(t) \le v_{\max},\\
&\mathrm{SOC}_{\min} \le \mathrm{SOC} \le \mathrm{SOC}_{\max},\\
&I_{\mathrm{batt,min}} \le I_{\mathrm{batt}} \le I_{\mathrm{batt,max}},\\
&a_{\min} \le a_{\mathrm{e}} \le a_{\max},\\
&T_{\mathrm{mot,min}}(\omega_{\mathrm{mot}}(t)) \le u(t) \le T_{\mathrm{mot,max}}(\omega_{\mathrm{mot}}(t)),\\
&s(0) = 0,\\
&s(T) = s_{\mathrm{f}},\\
&v(0) = v_0,
\end{aligned}
\tag{8}
$$

where F denotes the vehicle longitudinal dynamics described in Section 2.1. $v_{\min}$ and $v_{\max}$ are the legal speed limits. $I_{\mathrm{batt,min}}$ and $I_{\mathrm{batt,max}}$ denote the maximum charging and discharging current, respectively. a_{e} denotes the vehicle acceleration. $a_{\min}$ and $a_{\max}$ are its lower and upper bound, respectively. $u(t)$ denotes the control variable referring to the motor torque within the range $[T_{\mathrm{mot,min}}, T_{\mathrm{mot,max}}]$. s_{f} and v_0 denote the entire distance and initial velocity.

3.2 The learning framework

In this paper, the DDPG algorithm is employed to learn the optimal eco-cruising strategy. Different from deep Q-network (DQN) algorithm, DDPG establishes two separate neural networks: Q-network (Critic) and policy-network (Actor), together making up the AC networks [27].

The Q-network, parameterized by θ^Q, is the same as DQN. State s and action a are fed into the Q-network, and it outputs estimated action-value $Q(s,a)$. The network architecture is pyramid-like, consisting of three fully connected layers (200-100-50) [37,38], which are activated by the rectified linear unit (ReLU) [39,40]. Following the three hidden layers is a linear output layer that outputs a scalar value $Q(s,a)$.

The policy-network, parameterized by θ^μ, generates a deterministic action a. The architecture of the policy-network is the same as the Q-network, which also consists of three fully connected layers (200-100-50), except the output layer processed by a tanh activation function, mapping the output value into [−1,1]. Then, a linear mapping function transforms the output value into the boundary value of motor torque.

The objective of Actor-Critic networks' training is to deterministically map the state into a specific action with maximal Q value by policy-network, deriving an optimal parameterized eco-cruising strategy π^*. TD-error is given by $y_k = r_k + \gamma Q'\left(s_{k+1}, \mu'\left(s_{k+1} \mid \theta^{\mu'}\right) \mid \theta^{Q'}\right)$. Here, γ is the discount factor. The loss function is minimized by gradient descent algorithm:

$$
L = \frac{1}{N}\sum_i \left(y_k - Q\left(s_k, a_k \mid \theta^Q\right)\right)^2, \tag{9}
$$

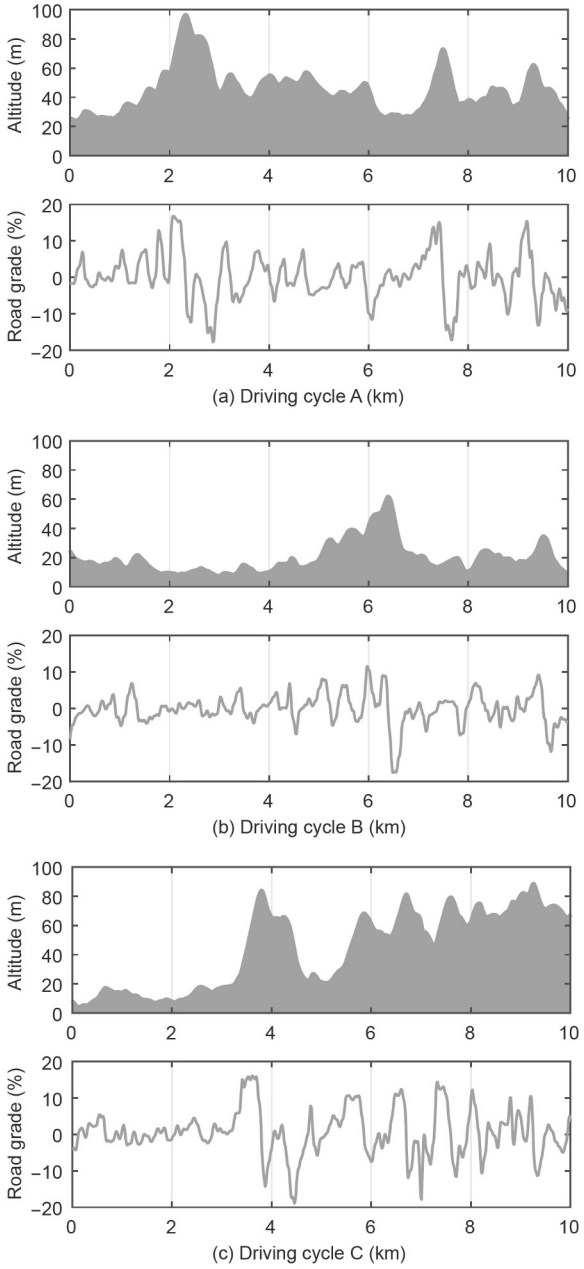

Figure 5 Profiles of road altitude and grade. (a) Driving cycle A; (b) driving cycle B; (c) driving cycle C.

$$
\begin{aligned}
&\nabla_{\theta^Q} L\left(s_k, a_k \mid \theta^Q\right) = 2\nabla_{\theta^Q} Q(s_k, a_k)\\
&\cdot\left[\left(r_k + \gamma Q'\left(s_{k+1}, \mu'\left(s_{k+1} \mid \theta^{\mu'}\right) \mid \theta^{Q'}\right)\right) - Q(s_k, a_k)\right].
\end{aligned}
\tag{10}
$$

Once the Q-network is updated, the policy-network will be guided by the estimated action-value function Q. The loss gradient is given as follows:

$$
\nabla_{\theta^Q} J \approx \frac{1}{N}\sum_i \nabla_a Q(s, a \mid \theta^Q) \cdot \nabla_{\theta^Q} \mu(s \mid \theta^\mu). \tag{11}
$$

Additionally, a replica of the Actor-Critic networks, namely target Q-network ($\theta^{Q'}$) and target policy-network ($\theta^{\mu'}$), is introduced [41,42]. The target Actor-Critic networks

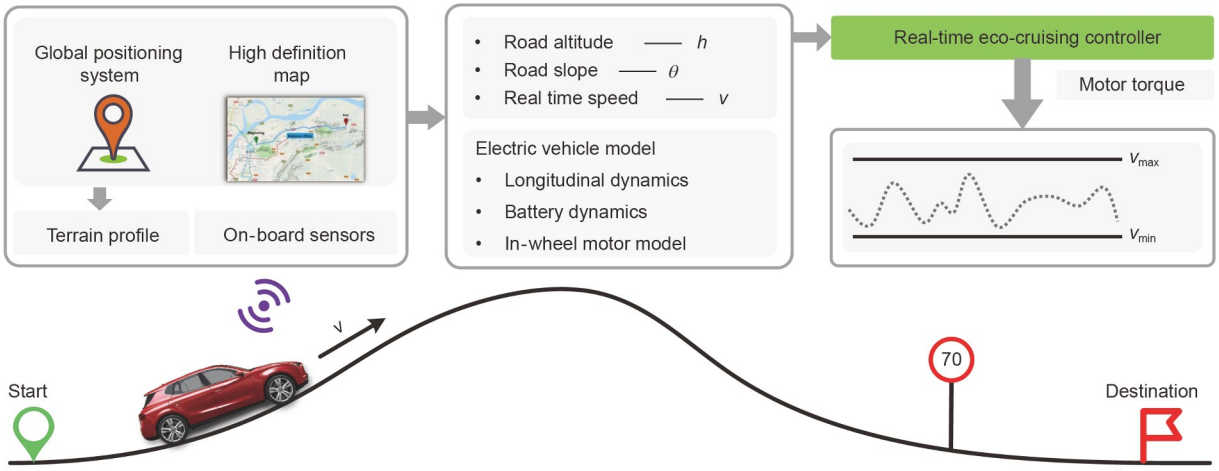

Figure 6 (Color online) Schematic overview of eco-cruising control considering a varying slope.

share the same architecture as the original Actor-Critic networks. Instead of directly copying the weight parameters, it updates network parameters by slowly tracking the learned networks [28]:

$$\theta_{k+1}^{Q'} = \tau\theta_k^{Q} + (1-\tau)\theta_k^{Q'},$$
$$\theta_{k+1}^{\mu'} = \tau\theta_k^{\mu} + (1-\tau)\theta_k^{\mu'}. \tag{12}$$

Because of the tracking rate $\tau \ll 1$, the target network updates at a slow rate, greatly improving the learning stability.

Another trick adopted here is prioritized experience replay (PER) to accelerate convergence [43]. According to the experimental results in ref. [44], DDPG with PER can improve training efficiency and stabilize the training process.

Algorithm 1 presents the entire procedure of the DDPG algorithm.

Algorithm 1 Pseudocode of the DDPG algorithm

1 Randomly initialize critic network and actor network with weights θ^{Q} and θ^{μ}.
2 **for** episode=1, M **do**
3 Initialize a random process $\mathcal{N}$ for action exploration;
4 Receive initial states: v_1, h_1, θ_1;
5 **for** k=1, K(number of discretization steps) **do**
6 Select action $a_k = \mu(s_k \mid \theta^{\mu}) + \mathcal{N}_k$ according to the current policy and exploration noise;
7 Execute action a_k, receive reward r_k and the new state s_{k+1};
8 Store transition (s_k, a_k, r_k, s_{k+1}) in replay buffer R;
9 Sample a minibatch of transitions (s_k, a_k, r_k, s_{k+1}) from R with priority experience replay;
10 Set $y_k = r_k + \gamma Q'(s_{k+1}, \mu'(s_{k+1} \mid \theta^{\mu'}) \mid \theta^{Q'})$;
11 Update critic by minimizing the loss:
$L = \frac{1}{N}\sum_i \left(y_k - Q\left(s_k, a_k \mid \theta^{Q}\right)\right)^2$;
12 Update the actor using the sampled policy gradient:
$\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_i \nabla_a Q(s, a \mid \theta^{Q}) \cdot \nabla_{\theta^{\mu}} \mu(s \mid \theta^{\mu})$;
13 Update the target networks:
$\theta_{k+1}^{Q'} = \tau\theta_k^{Q} + (1-\tau)\theta_k^{Q'}$,
$\theta_{k+1}^{\mu'} = \tau\theta_k^{\mu} + (1-\tau)\theta_k^{\mu'}$;
14 **end for**
15 **end for**

3.3 State, action and reward function

This section presents a detailed definition of the states, action, reward function, and settings of hyperparameters. We will also present the eco-cruising framework based on DDPG.

State: For the eco-cruising problem, in particular, we define a three-dimensional state vector consisting of the three most influential factors: vehicle speed, altitude, and road slope. The state-space can be defined as $S = \{v, h, \theta\}$. Vehicle speed must be within a predefined safety range $[v_{\min}, v_{\max}]$. An assumption is made that vehicle speed can be obtained through onboard sensors and road altitude, and slope can be accessed through GPS.

Action: Motor torque is selected as the action variable. The action space can be described as $A = \{T_{\text{mot}}\}$. Similarly, the motor torque is subject to the constraint $T_{\text{mot,min}}(\omega_{\text{mot}}) \le T_{\text{mot}} \le T_{\text{mot,max}}(\omega_{\text{mot}})$.

Reward: In the eco-cruising problem, the reward function can be formulated from electricity consumption, traveling time, speed constraint, and ride com comfort. In terms of ride comfort, vehicle acceleration a_e is used to reflect the driver's sensation. Intuitively, the multi-objective reward function is given as follows:

$$r = -\left(\alpha \cdot \Delta\text{SOC} + \beta \cdot \Delta t + \eta \cdot M_v + \delta \cdot a_e\right), \tag{13}$$

where α, β, η, and δ denote the weighting coefficient of electricity consumption, traveling time, speed constraint, and ride comfort, respectively; ΔSOC and Δt denote the battery

SOC variation and the traveling time within the interval of two adjacent sampling points, respectively, and their specific expressions will be given in Section 4.1; M_v denotes the penalty for the speed limits, expressed as follows:

$$M_v = \begin{cases} 0, & v_{\min} \le v \le v_{\max}, \\ (v - v_{\min})^2 + (v - v_{\max})^2, & \text{otherwise}. \end{cases} \tag{14}$$

For the policy-network, the learning rate is set as 0.0001; however, for the Q-network, it is set as 0.001. After trying different discount factors, we set the learning rate as 0.99. Similarly, different memory capacity values are evaluated, which is set as 10000 eventually. The batch size N is set as 64, and the soft replacement parameter τ is set as 0.01. After repeated tuning, the above weighting coefficients α, β, η, and δ are set as 1×10^5, 1, 1, and 1, respectively.

After elaborating all elements of the DDPG algorithm, the systematic flowchart of the DRL-based eco-cruising control framework is depicted in Figure 7. To enable the agent to better explore the environment and choose optimal actions, an exploration noise is added to the output of the policy network, which follows the normal distribution. Its variance decays as the training progresses from the initial value of 3.

4 Simulation results and discussion

In this section, firstly, deterministic DP-based strategy and constant speed (CS) strategy are used as benchmarks against the proposed DRL-based strategy. Secondly, the learning process and trajectories of vehicle speed, motor torque, and acceleration are illustrated. Thirdly, the energy-saving performance and calculation time are compared. Finally, the generalization performance of the proposed strategy is further validated on testing driving cycles.

4.1 Benchmark strategies

4.1.1 *Deterministic DP-based strategy*

Using Bellman's principle of optimality, DP converts the multistep optimal decision problem into a series of single-step ones. Thus, a complicated problem is broken down into several steps. Then, the OCP is solved from the last interval backward. Finally, an optimal control law is retrieved in reverse order.

However, when the DP method is used to design an eco-cruising strategy, an issue concerning DP implementation must be first solved [45]. For diverse vehicle speed trajectories, time durations for the same origin-destination (O-D) pair will differ, leading to different numbers of sampling steps if the OCP is formulated in the discrete-time format as eq. (7). Whereas, DP requires the same number of sampling steps to compare the costs among different velocities at each sample step and finally achieve the global minimum. Therefore, DP cannot be directly used to find the optimal eco-cruising strategy. The aforementioned OCP can be transformed in discrete-distance as follows:

$$\min J = \sum_{k=0}^{N-1} L(x(k), u(k)), \tag{15}$$

subject to

$$\begin{aligned} &\dot{v} = F(x(k), u(k)), \\ &\mathrm{d}s = v \cdot \mathrm{d}t, \\ &v_{\min} \le v(k) \le v_{\max}, \\ &\mathrm{SOC}_{\min} \le \mathrm{SOC} \le \mathrm{SOC}_{\max}, \\ &I_{\mathrm{batt,min}} \le I_{\mathrm{batt}} \le I_{\mathrm{batt,max}}, \\ &a_{\min} \le a_{\mathrm{e}} \le a_{\max}, \\ &T_{\mathrm{mot,min}}(\omega_{\mathrm{mot}}(k)) \le u(k) \le T_{\mathrm{mot,max}}(\omega_{\mathrm{mot}}(k)), \\ &s(0) = 0, \\ &v(0) = v_0, \end{aligned} \tag{16}$$

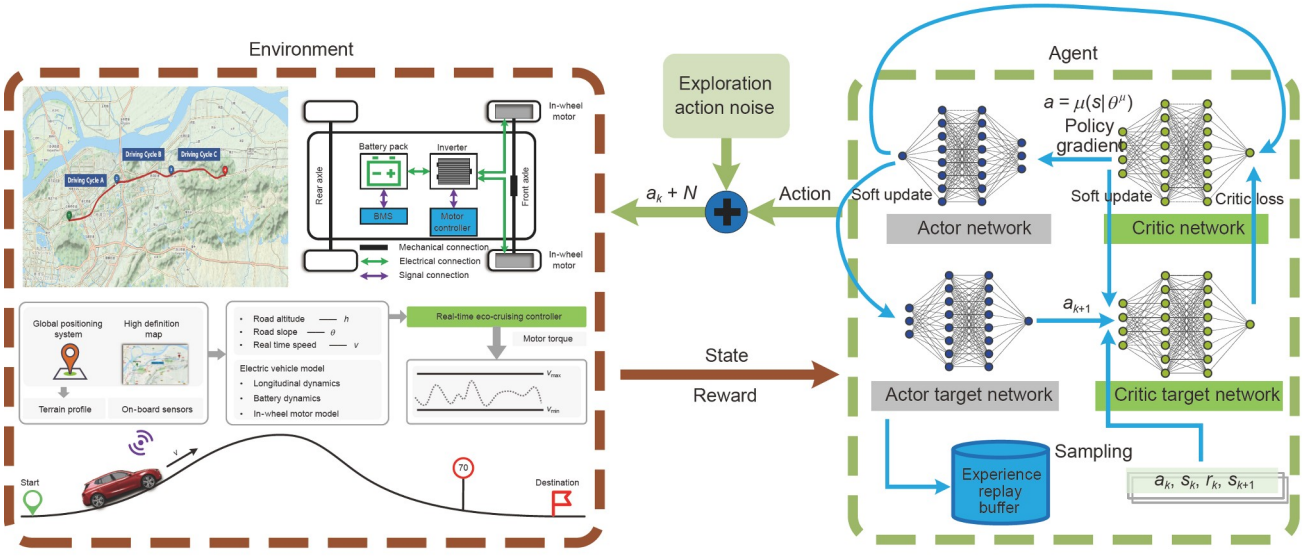

Figure 7 (Color online) DRL-based eco-cruising framework.

where the index k indicates the discretization step for N segments, which is evenly divided by unit distance Δs (set as 10 m); $L(x(k), u(k))$ denotes the instantaneous cost, expressed as follows:

$$L(x(k), u(k)) = \Delta \mathrm{SOC}(k) + \omega \cdot \Delta t(k), \tag{17}$$

$$\Delta \mathrm{SOC}(k) = \dot{\mathrm{SOC}}(k) \cdot \frac{2\Delta s}{v(k)+v(k+1)}, \tag{18}$$

$$\Delta t(k) = \frac{2\Delta s}{v(k)+v(k+1)}. \tag{19}$$

4.1.2 Constant speed strategy

The inefficiency of the DP algorithm makes it inefficient when used in real-time control systems. CS strategy, which is common in real life, is used as a baseline strategy. According to this strategy, the vehicle drives at constant speed equal to the average speed of the proposed DRL-based strategy.

4.2 Training result

All simulation experiments are implemented using Python 3.7 with the deep learning platform TensorFlow 1.15. The agents are trained for 500 episodes at an initial speed of 60 km/h. Since stochasticity has a considerable influence on the training progress, it is repeated seven times with different random seeds. Figure 8 shows the corresponding training progress. Within less than 50 episodes, the mean reward increases dramatically after decreasing during the startup period. After that, the mean reward increases slowly over time, fluctuating a bit and eventually stabilizing at a level around 0. Taking the third random seed (RL-3) as an example, the proposed strategy starts to converge from the 46th episode, demonstrating the effectiveness of PER in learning efficiency.

Figure 9 shows the altitude profile and corresponding velocity profile, along with the motor torque and acceleration profiles. Generally, the vehicle speed decreases when the vehicle travels uphill and increases when the vehicle travels downhill. By combining the speed constraints in the reward function, the speed profile can be controlled well within the predefined speed limit between 50 and 70 km/h. Figure 9(c) shows that the acceleration amplitudes are well maintained below 1 m/s^2 for most of the journey, with a maximum value of 1.09 m/s^2 and a minimum value of −1.13 m/s^2, leading to acceptable ride comfort.

4.3 Comparison analysis

The performance of the proposed DRL-based eco-cruising strategy needs to be evaluated to demonstrate its optimization capability. DP-based and CS-based strategies are implemented as benchmarking strategies. The average speed

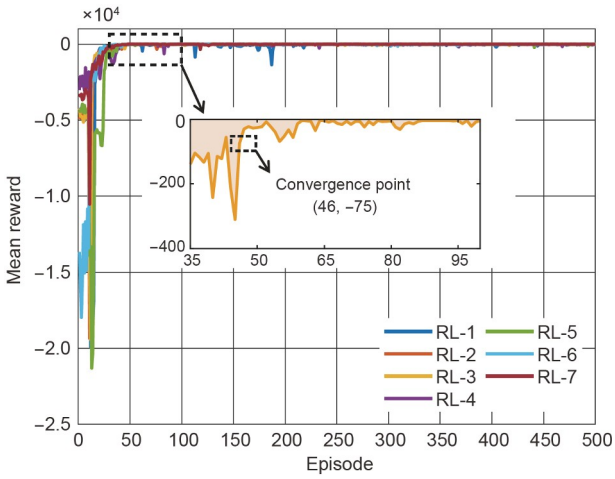

Figure 8 Training progress for seven agents with different random seeds.

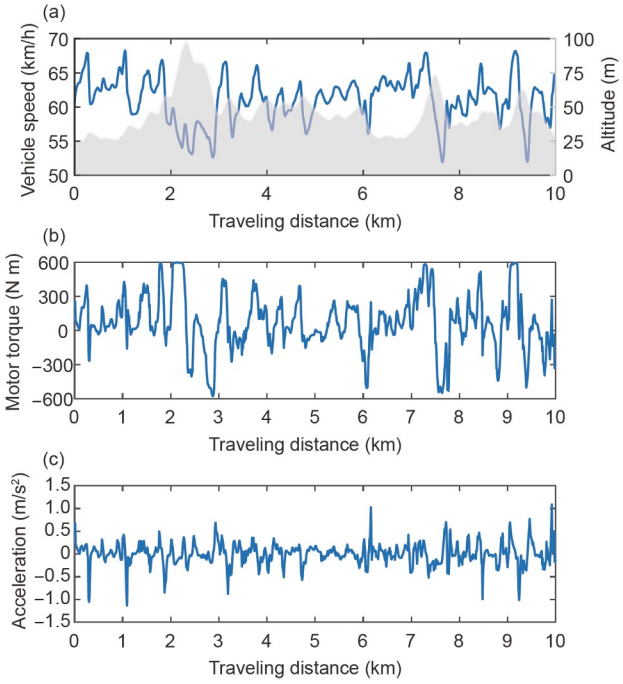

Figure 9 Training results. (a) Vehicle speed and altitude trajectories; (b) motor torque trajectory; (c) acceleration trajectory.

value of the above speed profile in Figure 9 is 60.2 km/h. For a fair comparison of energy consumption, we tune the weighting factor ω in the objective function of DP to keep the average speed of the journey approximately identical among three different strategies.

Table 3 and Figure 10 show the simulation results. Table 3 presents the SOC depletion, traveling time, and energy saving in percentage compared with the optimal solution via DP and rule-based CS. Traveling time is a significant factor since driving a certain distance faster tends to consume more electricity. In this simulation, CS, DP, and DRL have a similar traveling time, with a maximum gap of 1.1%, which

Table 3 Simulation results of CS, DP, and DRL

Algorithm	DP	DRL	CS
Initial SOC	0.8	0.8	0.8
Terminal SOC	0.7655	0.7650	0.7636
ΔSOC (%)	3.45	3.50	3.64
Traveling time (s)	598.2	591.6	591.6
Calculation time (s)	32.32	1.75	0.97
Energy saving (%)	5.2	3.8	–

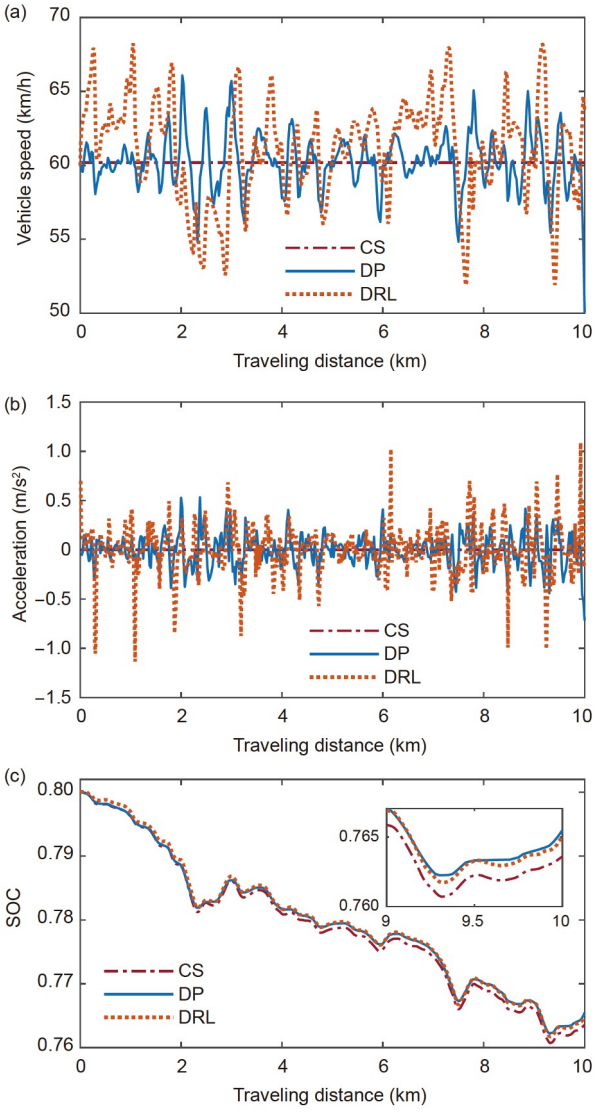

Figure 10 (Color online) Simulation results of three strategies. (a) Vehicle speed trajectories; (b) acceleration trajectories; (c) SOC trajectories.

can be ignored. In terms of SOC usage, the DP-based strategy consumes the least electricity among the three strategies, followed closely by the DRL-based strategy. Compared with CS, DP and DRL exhibit 5.2% and 3.8% energy-saving performance, respectively. Figure 10 shows the trajectories of vehicle speed, vehicle acceleration, and battery SOC. For speed profiles, the DRL and DP data show similar patterns. Despite the fact that the entire driving cycle information was used during the training, there is still a gap between results based on DP and DRL. The possible reason could be the definition of the problem. In DP, the OCP is defined in a finite horizon, while in DRL, it is defined in an infinite horizon. In terms of the acceleration amplitudes, both DP and DRL can control acceleration in an acceptable range $(-1.5,\ 1.5)$ m/s^2, whereas DP shows a relatively smaller amplitude of fluctuation. Aside from the near-optimal capability of DRL, the derived policy can also serve as an offline real-time controller. Similar to rule-based CS, the DRL-based eco-cruising strategy also exhibits quite remarkable computational speed (about 1.7 ms per simulation step) on a desktop computer with Intel Xeon CPU E3-1231.

4.4 Generalization evaluation

Some existing eco-cruising strategies, e.g., rule-based methods and fuzzy-logic methods, usually require elaborate design or much calibration effort and lack adaptability to other unknown conditions. However, in real applications, it may encounter uncertainty and problems for unseen driving conditions. For learning-based methods, its robustness across diversified contexts needs to be guaranteed. Thus, the trained strategy is further verified on testing cycles which are driving cycles B and C (see Figure 7(b) and (c)). The two testing cycles are completely unseen during the training process. Simulation results are presented in Table 4 and Figure 11.

Figure 11 depicts the results of the DRL strategy under the testing cycle. As shown in Figure 11(a) and (b), the initial vehicle speed for both DP and DRL is 60 km/h. The initial value of SOC is set to 0.8. According to the vehicle speed profiles of DP and DRL, we can conclude that the speed trajectory of DRL shows a similar trend with that of DP, indicating strong adaptability and self-learning ability of the DRL agent. However, in terms of the amplitude of acceleration, both DP and DRL seem to climb to a larger range $(-3,\ 3)$ m/s^2. However, the acceleration curve of DRL still shares a similar trend with DP's. Table 4 summarizes the energy-saving performance and traveling time. Similar to the previous comparison, the traveling time of DP is kept as close as possible to that of DRL. For driving cycle B, the SOC depletion gap between the DRL-based strategy and CS strategy is about 3.2%, whereas that between DP and CS is about 6.3%. For driving cycle C, DP and DRL improve energy-saving performance by 6.4% and 2.4%, respectively. The results demonstrate that there is a comparative energy conservation, despite that the percentage improvement is reduced compared with the results on the training cycle in Table 3. Consequently, we can conclude that the decision-making ability of RL and the generalization ability of deep

Table 4 Simulation results of optimization performance on testing driving cycles

Algorithm	Driving cycle B			Driving cycle C		
	DP	DRL	CS	DP	DRL	CS
Initial SOC	0.8	0.8	0.8	0.8	0.8	0.8
Terminal SOC	0.7704	0.7694	0.7687	0.7578	0.7560	0.7549
ΔSOC (%)	2.96	3.06	3.16	4.22	4.40	4.51
Traveling time (s)	664.3	680.4	680.4	613.8	615.8	615.8
Calculation time (s)	31.20	1.69	0.76	30.74	1.63	0.67
Energy saving (%)	6.3	3.2	–	6.4	2.4	–

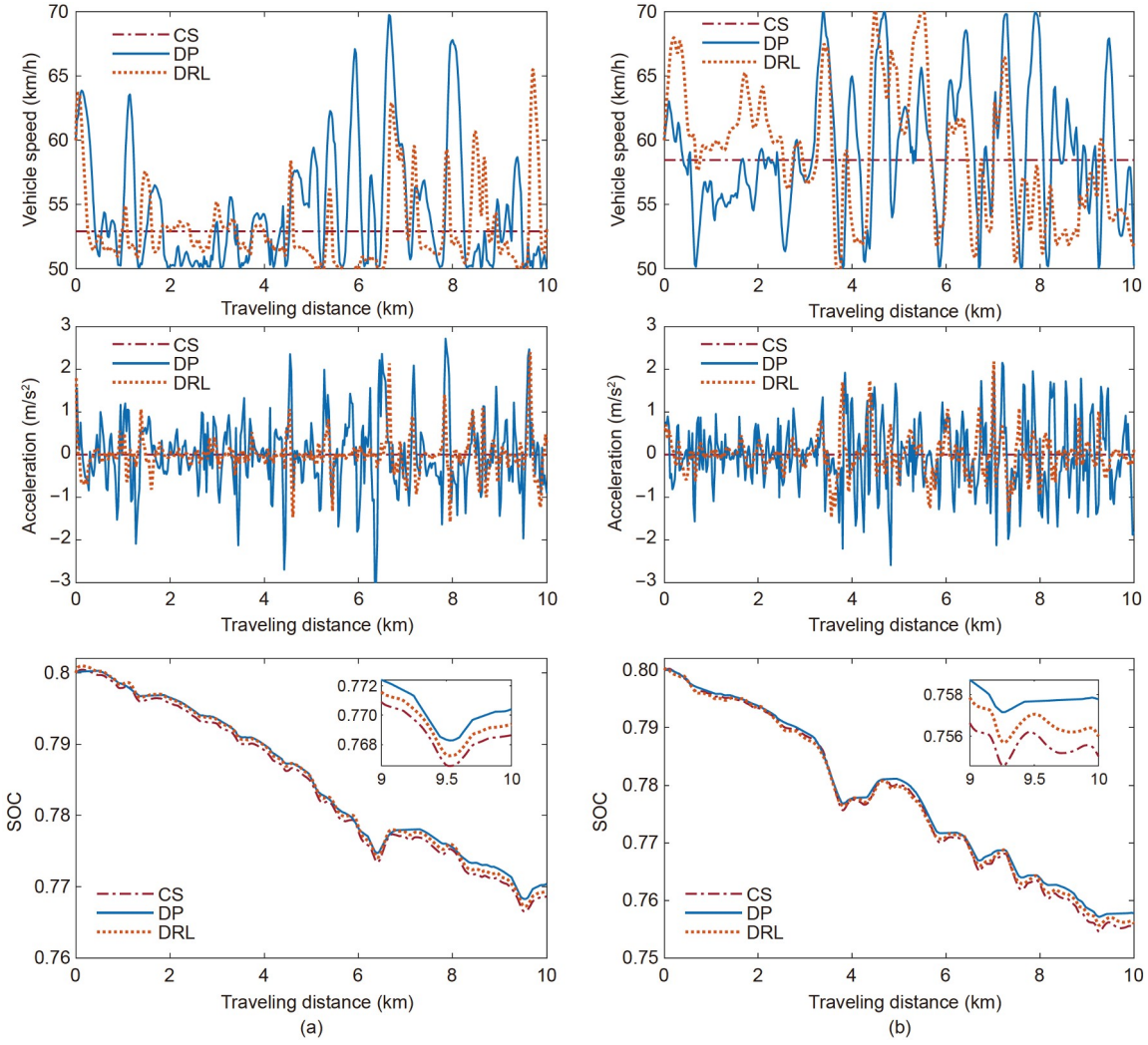

Figure 11 (Color online) Vehicle speed, acceleration, and SOC trajectories on testing cycles. (a) Driving cycle B; (b) driving cycle C.

neural networks together ensure its applicability toward other different driving cycles. Meanwhile, the trained strategy (represented by parameters of the policy-network) shows a comparative computation speed against the rule-based CS strategy. In the simulation environment, it takes only 2 ms on average per simulation step, making it feasible for real-time implementation.

5 Conclusions

In this paper, a DRL-based eco-cruising strategy for EVs considering road slope is developed. The DDPG algorithm incorporating a customized reward function is used to learn the eco-cruising strategy. The results of several case studies are as follows. (1) Compared with the CS strategy, the DRL-

based eco-cruising strategy yields great energy-saving capability (about 3.8%) and a small gap compared with the global optimal solution of DP. (2) For generalization ability, the trained policy is validated on two different testing driving cycles, showing good robustness. The average energy-saving rate of all driving cycles reaches 2.8%. (3) Although offline training is time-consuming, the trained strategy shows excellent computational efficiency during online implementation. Compared with DP, the average calculation efficiency is improved by about 94.6%, close to the rule-based controller. Therefore, the proposed DRL-based eco-cruising strategy exhibits the potential for real-time implementation.

Future work will be conducted from the following aspects. (1) In this paper, the speed limits along the given routes are set as fixed values. However, the road speed limit is related to travel distance in reality and should be integrated into the eco-driving strategy design. (2) Similarly, the introduction of the leading vehicle and upcoming traffic will make the strategy design more interesting and challenging.

This work was supported by the Graduate Student Innovation Project of Jiangsu Province, China (Grant No. KYCX20_0258).

1 U.S. Energy Information Administration, monthly energy review, https://www.eia.gov/energyexplained/use-of-energy/transportation
2 Liu T, Hu X, Hu W, et al. A heuristic planning reinforcement learning-based energy management for power-split plug-in hybrid electric vehicles. IEEE Trans Ind Inf, 2019, 15: 6436–6445
3 Barkenbus J N. Eco-driving: An overlooked climate change initiative. Energy Policy, 2010, 38: 762–769
4 Next-generation energy technologies for connected and automated on-road vehicles. https://arpa-e.energy.gov/technologies/programs
5 Xie L, Luo Y, Zhang D, et al. Intelligent energy-saving control strategy for electric vehicle based on preceding vehicle movement. Mech Syst Signal Processing, 2019, 130: 484–501
6 Chen B C, Wu Y Y, Tsai H C. Design and analysis of power management strategy for range extended electric vehicle using dynamic programming. Appl Energy, 2014, 113: 1764–1774
7 Saerens B, Van den Bulck E. Calculation of the minimum-fuel driving control based on Pontryagin's maximum principle. Transpation Res Part D-Transp Environ, 2013, 24: 89–97
8 Shen D, Karbowski D, Rousseau A. Fuel-optimal periodic control of passenger cars in cruise based on pontryagin's minimum principle. IFAC-PapersOnLine, 2018, 51: 813–820
9 Ye Z, Li K, Stapelbroek M, et al. Variable step-size discrete dynamic programming for vehicle speed trajectory optimization. IEEE Trans Intell Transp Syst, 2019, 20: 476–484
10 Dong H, Zhuang W, Yin G, et al. Energy-optimal velocity planning for connected electric vehicles at signalized intersection with queue prediction. In: Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM). Boston, 2020. 238–243
11 Zhuang W C, Qu L H, Xu S B, et al. Integrated energy-oriented cruising control of electric vehicle on highway with varying slopes considering battery aging. Sci China Tech Sci, 2020, 63: 155–165
12 Sciarretta A, Guzzella L. Control of hybrid electric vehicles. IEEE Control Syst Mag, 2007, 27: 60–70
13 Xie S, Hu X, Liu T, et al. Predictive vehicle-following power management for plug-in hybrid electric vehicles. Energy, 2019, 166: 701–714
14 Xiang C L, Ding F, Wang W D, et al. MPC-based energy management with adaptive Markov-chain prediction for a dual-mode hybrid electric vehicle. Sci China Tech Sci, 2017, 60: 737–748
15 Zhuang W, Xu L, Yin G. Robust cooperative control of multiple autonomous vehicles for platoon formation considering parameter uncertainties. Automot Innov, 2020, 3: 88–100
16 Sutton R S, Barto A G. Reinforcement Learning: An Introduction. 2nd ed. Cambridge: MIT Press, 2018
17 Li Y, He H, Khajepour A, et al. Energy management for a power-split hybrid electric bus via deep reinforcement learning with terrain information. Appl Energy, 2019, 255: 113762
18 Xu C, Zhao W Z, Chen Q Y, et al. An actor-critic based learning method for decision-making and planning of autonomous vehicles. Sci China Tech Sci, 2021, 64: 984–994
19 Zhou Q, Li J, Shuai B, et al. Multi-step reinforcement learning for model-free predictive energy management of an electrified off-highway vehicle. Appl Energy, 2019, 255: 113755
20 Wang P, Chan C Y. Formulation of deep reinforcement learning architecture toward autonomous driving for on-ramp merge. In: Proceedings of IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). Yokohama, 2017. 1–6
21 Shi J, Qiao F, Li Q, et al. Application and evaluation of the reinforcement learning approach to eco-driving at intersections under infrastructure-to-vehicle communications. Transpation Res Record, 2018, 2672: 89–98
22 Vázquez-Canteli J R, Nagy Z. Reinforcement learning for demand response: A review of algorithms and modeling techniques. Appl Energy, 2019, 235: 1072–1089
23 Guo Q, Angah O, Liu Z, et al. Hybrid deep reinforcement learning based eco-driving for low-level connected and automated vehicles along signalized corridors. Transpation Res Part C-Emerging Technologies, 2021, 124: 102980
24 Zhu Z, Gupta S, Gupta A, et al. A deep reinforcement learning framework for eco-driving in connected and automated hybrid electric vehicles. 2021. ArXiv: 2101.05372
25 Boriboonsomsin K, Barth M. Impacts of road grade on fuel consumption and carbon dioxide emissions evidenced by use of advanced navigation systems. Transpation Res Record, 2009, 2139: 21–30
26 Lee H, Kim N, Cha S W. Model-based reinforcement learning for eco-driving control of electric vehicles. IEEE Access, 2020, 8: 202886
27 Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning. 2015. ArXiv: 1509.02971
28 ProteanDrive. https://www.proteanelectric.com/technology/
29 Xie S, Hu X, Xin Z, et al. Time-efficient stochastic model predictive energy management for a plug-in hybrid electric bus with an adaptive reference state-of-charge advisory. IEEE Trans Veh Technol, 2018, 67: 5671–5682
30 Zhang F, Xi J, Langari R. Real-time energy management strategy based on velocity forecasts using V2V and V2I communications. IEEE Trans Intell Transp Syst, 2017, 18: 416–430
31 Sun C, Moura S J, Hu X, et al. Dynamic traffic feedback data enabled energy management in plug-in hybrid electric vehicles. IEEE Trans Contr Syst Technol, 2015, 23: 1075–1086
32 Guo J Q, He H W, Peng J K, et al. A novel MPC-based adaptive energy management strategy in plug-in hybrid electric vehicles. Energy, 2019, 175: 378–392
33 Murphey Y L, Park J, Chen Z, et al. Intelligent hybrid vehicle power control—part I: Machine learning of optimal vehicle power. IEEE Trans Veh Technol, 2012, 61: 3519–3530
34 Liu T, Zou Y, Liu D, et al. Reinforcement learning of adaptive energy management with transition probability for a hybrid electric tracked vehicle. IEEE Trans Ind Electron, 2015, 62: 7837–7846
35 Wu J, He H, Peng J, et al. Continuous reinforcement learning of energy management with deep Q network for a power split hybrid electric bus. Appl Energy, 2018, 222: 799–811
36 Liu T, Hu X, Li S E, et al. Reinforcement learning optimized look-ahead energy management of a parallel hybrid electric vehicle. IEEE/ASME Trans Mechatron, 2017, 22: 1497–1507

37 Lian R, Peng J, Wu Y, et al. Rule-interposing deep reinforcement learning based energy management strategy for power-split hybrid electric vehicle. Energy, 2020, 197: 117297
38 Larochelle H, Bengio Y, Louradour J, et al. Exploring strategies for training deep neural networks. J Mach Learn Res, 2019, 10: 1–40
39 He K, Zhang X, Ren S, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of IEEE International Conference on Computer Vision (ICCV). Santiago, 2015. 1026–1034
40 Zhang K, Sun M, Han T X, et al. Residual networks of residual networks: Multilevel residual networks. IEEE Trans Circ Syst Video Technol, 2018, 28: 1303–1314
41 Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518: 529–533
42 Dong H, Ding Z, Zhang S. Deep Reinforcement Learning: Fundamentals, Research and Applications. Singapore: Springer, 2020
43 Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay. 2016. ArXiv: 1511.05952
44 Hou Y, Liu L, Wei Q, et al. A novel DDPG method with prioritized experience replay. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC). Banff, 2017. 316–321
45 Chen Y, Li X, Wiet C, et al. Energy management and driving strategy for in-wheel motor electric ground vehicles with terrain profile preview. IEEE Trans Ind Inf, 2014, 10: 1938–1947