

Multi-agent Deep Reinforcement Learning for Charge-sustaining Control of Multi-mode Hybrid Vehicles

Min Hua¹, Quan Zhou^{1,*}, Cetengfei Zhang¹, Fanggang Zhang¹, Hongming Xu^{1,*}, Wei Liu²

¹ Department of Mechanical Engineering, University of Birmingham, Birmingham, B15 2TT, UK

² School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, US

Highlights

- MIMO control optimization is researched to save energy for a multi-mode HEV.
- DDPG is employed to address continuous control problems by self-learning under a predefined driving cycle obtained by a novel random method.
- Collaborative cyber-physical simultaneous learning is proposed to enhance energy efficiency up to approximately 4% by MARL for charge-sustaining control.
- Learning performance with the different relationships of MARL has been studied.

Abstract

Transportation electrification requires an increasing number of electric components (e.g., electric motors and electric energy storage systems) on vehicles, and control of the electric powertrains usually involves multiple inputs and multiple outputs (MIMO). This paper focused on the online optimization of energy management strategy for a multi-mode hybrid electric vehicle based on multi-agent reinforcement learning (MARL) algorithms that aim to address MIMO control optimization while most existing methods only deal with single output control. A new collaborative cyber-physical learning with multi-agents is proposed based on the analysis of the evolution of energy efficiency of the multi-mode hybrid electric vehicle (HEV) optimized by a deep deterministic policy gradient (DDPG)-based MARL algorithm. Then a learning driving cycle is set by a novel random method to speed up the training process. Eventually, network design, learning rate, and policy noise are incorporated in the sensibility analysis and the DDPG-based algorithm parameters are determined, and the learning performance with the different relationships of multi-agents is studied and demonstrates that the not completely independent relationship with Ratio 0.2 is the best. The comparison study with the single-agent and multi-agent suggests that the multi-agent can achieve approximately 4% improvement of total energy over the single-agent scheme. Therefore, the multi-objective control by MARL can achieve good optimization effects and application efficiency.

Keywords: Energy efficiency optimization; Charge-sustaining control; Multi-mode hybrid electric vehicle; Multi-agent reinforcement learning;

* Corresponding Authors: Q. Zhou (q.zhou@bham.ac.uk) and H. M. Xu (h.m.xu@bham.ac.uk)

1 Introduction

Facing the pressures of the environmental deterioration and energy shortage nowadays, the electrification of transportation is a considerably necessary and promising solution to alleviate vehicle emission concerns, such as SO_2 , CO , and NO_x harmful gases and greenhouse gas CO_2 emissions, and the threat of energy sources [1]. However, the electrified powertrain in the automotive industry, with diversified power sources, plays a tremendous role in reducing these emissions released into the atmosphere[2]. Battery electric vehicles (BEVs), hybrid electric vehicles (HEVs), and fuel cell vehicles (FCVs) are mainstream classifications, where HEVs with an internal combustion engine(ICE) and motor are an ideal transition from traditional ICE-based vehicles into new energy vehicles, when BEVs possess the limitations of short driving range, long charging time, and high battery costs and for FCVs, there are technical problems in the storage and transportation of hydrogen despite the characteristics of non-polluting, high efficiency, and diverse fuel sources. The technologies of energy management strategies (EMSs) with various powertrain architectures will be explored and enhanced to address the energy efficiency and emission issues.

Most literature divides the EMSs of HEVs into rule-based, optimization-based, and learning-based methods [3]. Among them, the rule-based strategies consist of deterministic and fuzzy logic methods and then obtain control rules founded on the parameters of the vehicle and expert knowledge, which is easy and fast to apply to real-world environments but tough and inaccurate to approach optimal fuel economy[4]. Considering the characteristics of the engine, motor, and other components, the hybrid power system switching mode is designed and the control torque is distributed between the engine and the electric motor according to the driver's acceleration and braking demands, the vehicle state, battery status, and other information[5]. Optimization-based strategies, such as Dynamic Programming(DP), Pontryagin's minimum principle(PMP), Equivalent consumption minimization strategy(ECMS), and Model predictive control(MPC) are proposed[6]. DP, as the benchmark, traverses all feasible solutions in the state space and obtains the optimal fuel economy based on the Bellman theory, however, it is difficult to apply in real-time scenarios due to the heavy computation cost, and the driving cycle is predefined and fixed[7]. PMP and ECMS, which are used as a reference for developing other strategies, solve the energy optimization by minimizing the Hamilton equation, the computation time is reduced, compared with DP, but the algorithm also requires the known working conditions[8]. There are also some studies using real-time optimization algorithms, like MPC, and online rolling optimization for control variables has been conducted by predicting the future dynamic model, whose energy-efficient and real-time performance depend on the length of the control and the prediction domain. If the driving conditions vary greatly with time, the MPC controller cannot obtain desired real-time control performance. Therefore, the main problem of optimization-based strategies possesses poor adaptability to the working conditions, and it is tough to obtain good results for multi-objective and multi-mode optimization problems due to a large amount of calculation and the accuracy of the control model.

In recent years, artificial intelligence (AI) algorithms applied in various fields, such as self-driving, robotics, and electric vehicles, have been attracting abundant attention, due to playing a momentous role in addressing multi-objective and multi-constraints issues. Deep learning (DL) and reinforcement learning (RL) have developed as potential solutions to deal with the limitations of the rule-based and optimization-based methods, especially from the aspects of self-adjusting and adaptivity capability.

Q-learning is the prevalent method of RL in the field of EMSs to optimize the discrete system, which is attributed to the computation cost and algorithmic complexity. A new blended real-time EMS based on Q-learning with Charging-Depletion mode(QCD) and a conventional Charge-Sustaining(CS) mode is proposed to deal with the tradeoff between real-time performance and optimality[9]. A Q-learning-based method combined with neuro-dynamic programming(NDP) with future trip information is presented to solve the power management application under two-stage deployment[10]. Four different RL algorithms including Q-learning, Sarsa, Actor-Critic, QV-learning, and ACLA with four ensembled methods, majority voting, rank voting, Boltzmann multiplication, and Boltzmann addition, have been implemented, then the value functions of these different RL algorithms are combined to represent and learn a single value function, the results of the experiment show the Boltzmann multiplication and majority voting ensemble outperforms the single one[11]. Q-learning algorithm incorporated into the stochastic model predictive control(SMPC) controller is presented by constructing a multi-step Markov velocity prediction model to achieve superior fuel economy[12]. A Q-function updating method combining direct learning and indirect learning is proposed, then a virtual world model is introduced to approximate the real-time environment, the results demonstrate that DYNA-Q outperforms Q-learning and rule-based strategy in terms of adaptivity, real-time performance, and optimality[13]. The warm-start method is utilized to reduce the learning iterations of Q-learning in hybrid electric vehicle applications with more detailed explanations of the warm-start process, and the heuristic strategy and the ECMS are employed to initialize the Q-table[14]. The adaptability demonstration with three main factors, driving cycle, vehicle load condition, and road grade is analyzed then the Q-learning was first investigated based on supervisory control for HEVs[15]. Although the Q-learning algorithm can obtain excellent results, require lower computing costs, and have good convergence when there are low state dimensions, with the increase of state and action variables and the complication of the energy management system (EMS), it becomes tough to compute all Q values corresponding to the discrete state-action from the perspectives of computing efficiency and the algorithm optimization. Thus, deep reinforcement learning (DRL)-based EMSs are effectively capable of dealing with high-dimensional continuous EMS issues by using the neural network to approximate the value function.

The DRL algorithm, a combination of DL and RL, can be more flexible and generally applicable to solving the dynamic environment. Then, with more diverse architectures of the HEV and more components of the HEV, various approaches have been presented to design the EMS and to enhance the energy-saving, flexible and real-time performance considering a more complicated control system[16]. Considering three aspects including the energy consumption, the real-time

condition, and the application in different scenarios, the RL-based online EMS is presented based on the power transition probability matrices updated by the new driving cycle and Kullback-Leibler (KL) divergence rate[17]. Based on DRL algorithms, a deep Q-network(DQN)-based energy and emission management strategy(E&EMS) with two distributed DRL, including asynchronous advantage actor-critic (A3C) and distributed proximal policy optimization (DPPO), is first presented to achieve near-optimal fuel economy and excellent computational efficiency[18]. Adversarial examples obtained from the fast gradient sign method(FGSM) can have a prominent influence on degrading the well-trained DRL-based EMS, and the form of cyber-attack should be carefully considered to improve the robustness of the EMS[19]. An offline RL training framework with the maximum possible utility of offline data is presented to solve the issues of sample and deployment inefficiency, safety constraints, and simulation-to-real gaps nowadays for connected hybrid electric vehicles[20]. Ensemble learning velocity prediction(ELVP) with Markov chain(MC), backpropagation(BP), and radial basis function(RBF) neural network(NN)-based methods are proposed to optimize the energy management strategy(EMS) considering the driving pattern[21]. A novel double deep reinforcement learning(DDRL), combining a deep Q-network(DQN) to learn the gear-shifting strategy and deep deterministic policy gradient(DDPG) to control the engine throttle, is presented to achieve the multi-objective synchronization control for parallel hybrid electric vehicles(HEVs) with start-stop strategy[22]. An ensemble reinforcement learning with a thermostatic strategy and ECMS, used as two single agents in the presented ensemble agents, is proposed to enhance the fuel economy[23]. A hierarchical power splitting structure is used to reduce the state-action space by an adaptive fuzzy filter based on a reinforcement learning-based algorithm using the ECMS to tackle high-dimensional state-action space for fuel cell hybrid electric vehicles[24]. A transfer learning between DRL-based EMSs, with cross-type knowledge transfer among four different types of HEVs, is presented to improve the efficiency of HEVs' development automatically[25]. The RL-related framework, objectives, and architectures for EMS are reviewed, in addition, a detailed comparative analysis for achieving different charging coordination objectives while satisfying multiple constraints is conducted[26]. Thus, DRL-based control strategies combined with other advanced algorithms, employed in the EMS, can provide more accurate control rules, compared to the tabular Q-learning method and rule-based scheme; and the reduced-order model dependency of many other algorithms, such as ECMS and MPC, can be addressed due to its model-free feature. In the meanwhile, as the EMSs of the HEVs become more complex due to the combination of multiple power sources to achieve the mode switch and keep the charge-sustaining (CS) and the control variables and objectives will increase, DRL-based methods of interdisciplinary integration have emerged. However, few scholars have focused their attention on the multi-agents of RL to solve multiple continuous control and state variables.

This paper investigates a new methodology of a model-free charge sustaining control of a multi-mode HEV based on collaborative cyber-physical learning with multiple agents, and the present work

includes the following new areas: 1) multiple inputs and multiple-output (MIMO) control optimization of a multi-mode HEV is proposed for the CS control; 2) the DDPG, one of deep reinforcement learning methods, is investigated to address continuous control problem and self-learn under different driving conditions; and 3) collaborative cyber-physical learning with multiple agents based on DDPG is studied to achieve simultaneous optimization control of multi-objective and multi-variables.

The rest of this paper is organized as follows: Section 2 demonstrates a multi-mode HEV powertrain modeling based on DRL-based EMS with DDPG. Then the structure of collaborative cyber-physical learning with multiple agents is proposed in Section 3, followed by the description of the experimental results and analysis for the simultaneous optimization control validation and evaluation in Section 4 from the different aspects. Section 5 summarizes the conclusions.

2 DRL-based EMS with multi-mode HEV Powertrain modeling

As shown in Fig. 1, a multi-mode HEV is mainly made up of two power sources(a battery and an engine generator) that work together to meet the vehicle's power needs. The engine generator (MG1) is a power source that keeps the battery's SoC constant for longer driving distances. The electric motor(MG2) is the other primary power source to drive and brake the vehicle. A multi-mode HEV has different working modes through different mechanical connections like clutch, including series hybrid mode, parallel hybrid mode, series-parallel hybrid mode, and regenerative brake, as shown in Fig. 1 with varying lines of color to achieve the charge-sustaining, which makes the EMS more complicated.

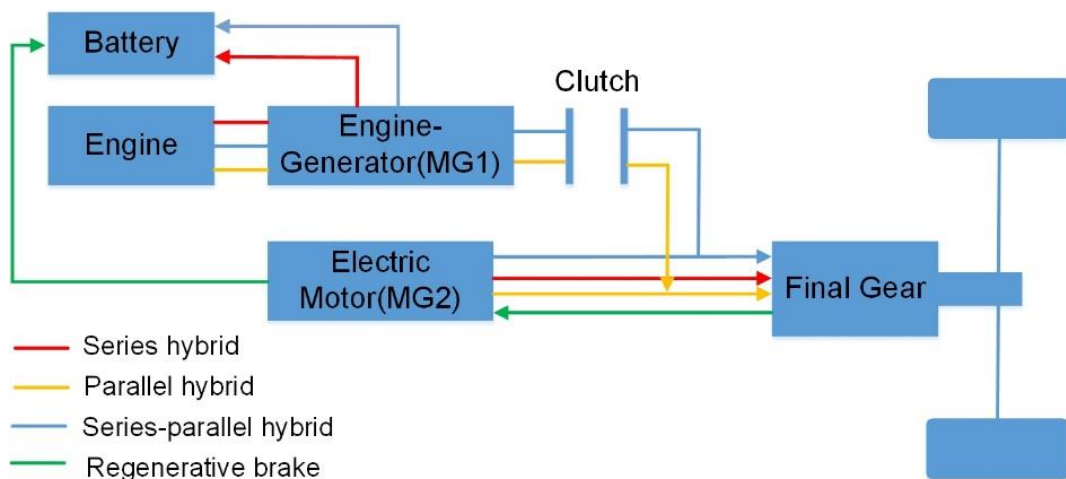


Fig. 1. Configuration of the multi-mode HEV powertrain energy-flow

Founded on the DRL-based method employed in the EMS, the environment, agent, state, action, and reward are the five basic ideas of a broad DRL framework, as depicted in Fig. 2. And the forward and backward techniques are two approaches to model the multi-mode HEV powertrain configuration for optimizing and evaluating EMSs. The backward one is selected in this paper with no need for a driver model since the desired speed(predefined driving cycle) is a direct input, and

the outputs are power distribution and energy consumption. The DRL agent investigates the complex and uncertain environment by performing various behaviors (continuous control actions), and the environment (the multi-mode HEV powertrain model) subsequently provides an immediate reward (energy consumption) to determine the action value and updates the vehicle state (e.g., SoC, power demand). The greater the action value is, the higher the reward will be. Eventually, the DRL agent will determine a decision sequence with a higher average cumulative reward by exploring the environment and receiving the relevant reward. The algorithm will be called convergent when the average cumulative reward becomes steady.

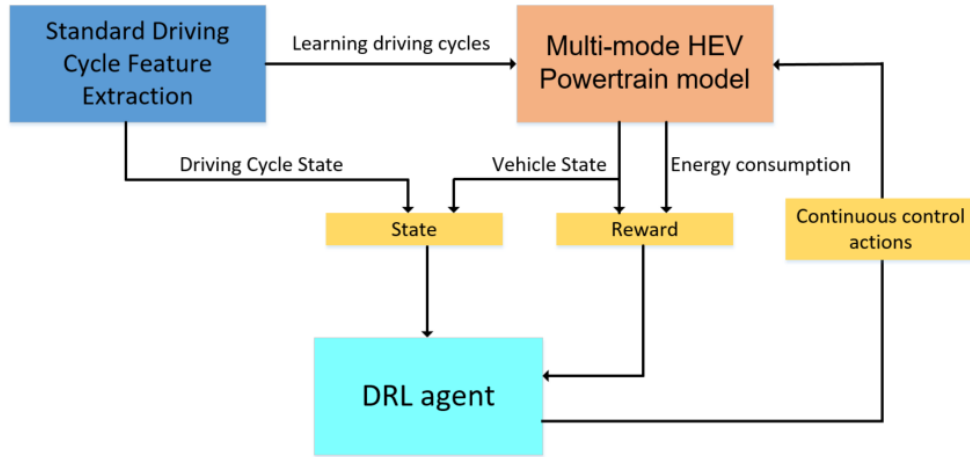


Fig. 2. A schematic overview of learning-based control strategy

2.1 Environment

The backward model of the multi-mode HEV powertrain is built firstly for the DRL-based environment, the EMS of which primarily deals with power allocation among multiple power sources, regardless of its various hybrid modes, it should satisfy the power balance equations with the longitudinal vehicle model, and then the vehicle power demand for a given driving cycle can be calculated.

a) Motion equations

The longitudinal force model mainly consists of rolling resistance force F_r , aerodynamic drag force F_a , The gradient resistance force F_g And inertial force F_i as described in Equation(1)[27].

$$\begin{aligned}
 F &= F_r + F_a + F_g + F_i \\
 F_r &= m \cdot g \cdot f \\
 F_a &= \frac{1}{2} \rho \cdot A_f \cdot C_d \cdot v^2 \\
 F_g &= m \cdot g \cdot \alpha \\
 F_i &= m \cdot a
 \end{aligned} \tag{1}$$

Where m is the vehicle mass, a is the vehicle acceleration, g is the gravity acceleration, f is the rolling resistance coefficient, and ρ is the air density, A_f is the front area of the vehicle, C_d is the air resistance coefficient, v is the longitudinal velocity, α is the road slope; not taken into account in this paper, these parameter values are described in Table.1.

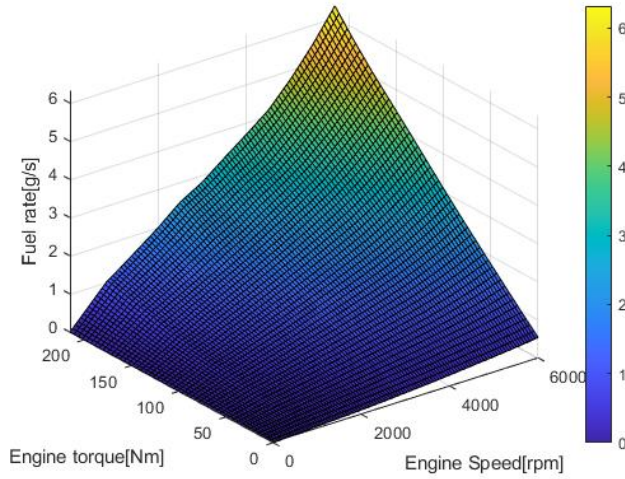
Table. 1 Main parameters of the multi-mode HEV specification

Parameters	Values
Vehicle mass m	1718.375 kg
Drag coefficient C_d	0.32
Vehicle front area A_f	2.455 m ²
Gravity coefficient g	9.8 m/s ²
Rolling resistance coefficient f	0.011
Air density ρ	1.2258 kg/m ³

b) Engine model

The fuel consumption rate \dot{m}_f (g/s) is calculated through a look-up table as shown in Fig. 3, which is a function of the engine speed w_{eng} and the effective engine torque T_{eng} .

$$\dot{m}_f(t) = f(w_{eng}(t), T_{eng}(t)) \quad (2)$$

**Fig. 3. Engine efficiency map****c) Motor model**

A quasi-static map defining the electric efficiency characteristic is used to model the electric motors (MG1 and MG2), as depicted in Fig. 4. The MG1 and MG2 electric power $P_{mot1,2}(t)$ that are utilized (in traction mode) or provided to the battery (in regenerative mode) are calculated by the efficiency maps $\eta_{mot1,2}(t)$ in equation (3) and equation (4), as a function of the electric rotational

speed $w_{mot1,2}(t)$ and the torque $T_{mot1,2}(t)$ respectively, the electric power is expressed as equation (5) and equation (6).

$$\eta_{mot1}(t) = f(w_{mot1}(t), T_{mot1}(t)) \quad (3)$$

$$\eta_{mot2}(t) = f(w_{mot2}(t), T_{mot2}(t)) \quad (4)$$

$$P_{mot1}(t) = \frac{w_{mot1}(t) \cdot T_{mot1}(t)}{9550} \cdot \eta_{mot1}(t) \quad (5)$$

$$P_{mot2}(t) = \begin{cases} \frac{w_{mot2}(t) \cdot T_{mot2}(t)}{9550} \cdot \eta_{mot2}(t); & T_{m2}(t) > 0 \\ \frac{w_{mot2}(t) \cdot T_{mot2}(t)}{9550} \cdot \frac{1}{\eta_{mot2}(t)}; & T_{m2}(t) \leq 0 \end{cases} \quad (6)$$

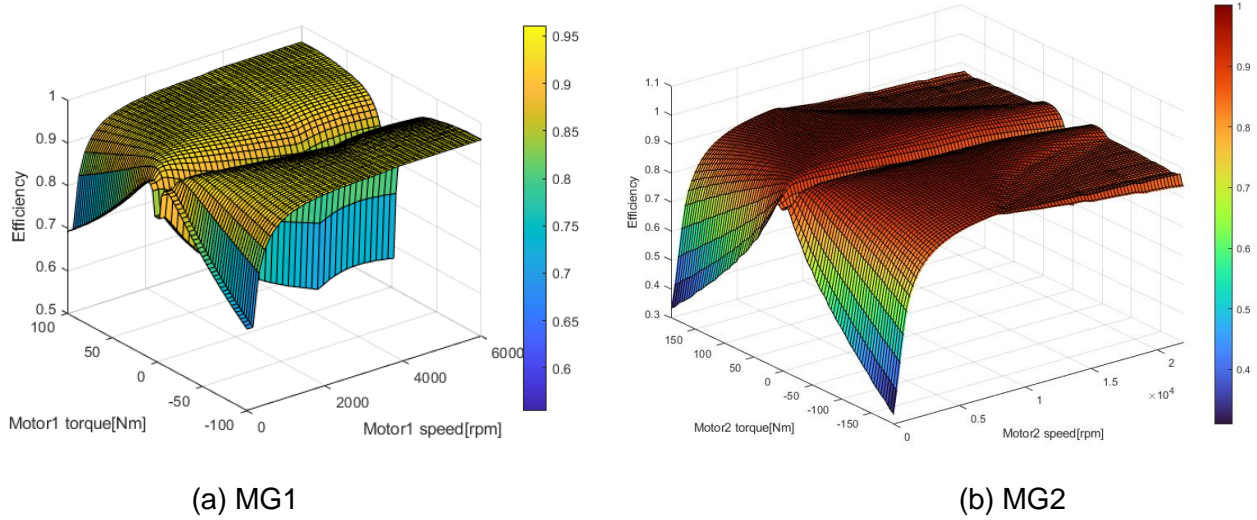


Fig. 4. Motor efficiency map

d) Battery model

The battery model is established by an equivalent circuit model (ECM) in equation (7), and this research mainly focuses on CS control of the EMS to maintain the SoC of the battery. Furthermore, the temperature and battery aging is not considered in this research, and the open-circuit voltage (OCV) $U(U = 350V)$ and the battery internal resistance $R(R = 0.15\Omega)$ are both constant.

$$P_{batt}(t) = P_{mot2}(t) - P_{mot1}(t)$$

$$P_{batt}(t) = U \cdot I_{batt}(t) - R \cdot I_{batt}^2(t)$$

$$I_{batt}(t) = \frac{U - \sqrt{U^2 - 4 \cdot R \cdot P_{batt}(t)}}{2 \cdot R} \quad (7)$$

$$SoC(t) = SoC(0) - \frac{\int_0^t I_{batt}(t) dt}{Q_{batt}}$$

where, $P_{batt}(t)$ is the output power in the charge-discharge process, $SoC(0)$ is the initial SoC value (for the health of the battery, $SoC(0)$ should be 28%), $I_{batt}(t)$ is the current of the battery, and Q_{batt} is the nominal battery capacity ($Q_{batt} = 54.3 \text{ Ah}$).

e) Power flow model

The power of multi-mode HEV from the battery and the engine, so the total power loss includes the engine and battery; the power loss of the engine $Loss_{eng}(t)$ and power loss of battery $Loss_{batt}(t)$ can be calculated by the fuel flow rate $\dot{m}_f(t)$, engine torque T_{eng} , engine rotational speed w_{eng} and the battery current I_{batt} respectively. The power flow model is defined as:

$$\begin{aligned}
 P_{dem}(t) &= P_{eng}(t) + P_{batt}(t) \\
 P_{eng}(t) &= \frac{w_{eng}(t) \cdot T_{eng}(t)}{9550} \\
 P_{loss}(t) &= Loss_{eng}(t) + Loss_{batt}(t) \\
 \begin{cases} Loss_{eng}(t) = \dot{m}_f(t) \cdot H_f - \frac{w_{eng}(t) \cdot T_{eng}(t)}{9550} \\ Loss_{batt}(t) = R \cdot I_{batt}(t)^2 \end{cases} & \quad (8)
 \end{aligned}$$

where, H_f is the heat value of fuel (for diesel, $H_f = 43.5 \times 10^6$ J/kg), R is the equivalent internal resistance in the battery model.

2.2 State and action

The states directly dictate control action based on the DRL algorithm. The battery SoC and the power demand are selected as the state variables in this study to create a two-dimensional continuous state space[28]:

$$s(t) = [T_{dem}(t), SoC(t)] \quad (9)$$

where, $s(t)$ is the current state at the t^{th} time step; $T_{dem}(t) \in \mathbf{T}_{dem}$ is the power demand value at the t^{th} time step.

The key problem with the multi-mode HEV EMSs is how to determine the torque-split ratio among the internal combustion engine (ICE), the MG1, and MG2. In this work, the MG1 output torque ratio $u_{mot1}(t)$ is selected as the control action firstly, which is denoted as :

$$a(t) = u_{mot1}(t) \quad (10)$$

$a(t)$ is the continuous action value at the time t^{th} time step will be used to control the MG1 torque ratio, where, $u_{mot1}(t) \in \mathbf{U}$, $\mathbf{U} = [0 \ 1]$. After obtaining the control signal $a(t)$, the MG2 torque output will be at maximum value, which is denoted as T_{mot2_max} , and the requirement for the engine torque for the output shaft can be obtained using

$$\begin{aligned}
 T_{eng} &= u_{mot1}(t) \cdot T_{mot1_max} + T_{GB} \\
 T_{dem} &= T_{mot2_max} + T_{eng} \\
 u_{eng}(t) &= \frac{u_{mot1}(t) \cdot T_{mot1_max} + T_{GB}}{T_{eng_max}}
 \end{aligned} \quad (11)$$

where, $u_{eng}(t)$ is the engine torque ratio, T_{mot1_max} is the maximum torque of the MG1; T_{eng_max} is the maximum torque that can be supplied by the engine; and T_{dem} is the torque demand for driving and braking the vehicle; T_{GB} is the torque of the output shaft provided by the engine when MG2 output torque cannot meet the requirement of the total torque output.

2.3 Reward

The instantaneous fuel consumption and the cost of the battery CS are considered as the multi-objective reward function based on the DDPG-based EMS. The control goal is to reduce power loss while keeping an appropriate battery SoC range during the driving cycle. The reward function is defined as follows:

$$r(t) = \begin{cases} -\alpha P_{loss}(t) & SoC(t) \geq SoC(0) \\ -\alpha P_{loss}(t) - \beta |SoC_{ref} - SoC(t)| & SoC(t) < SoC(0) \end{cases} \quad (12)$$

where, $SoC(0)$ is the reference battery SoC value that is chosen to maintain the battery SoC (keep the approximate equivalence of the initial SoC and the final SoC). α and β are scale factors used to balance the consideration of battery SoC level and power efficiency.

2.4 Agent

For environments with both a continuous action and observation space, Deep Deterministic Policy Gradient (DDPG) is the most straightforward compatible agent, followed by Twin-Delayed Deep Deterministic Policy Gradient (TD3), Proximal Policy Optimization (PPO), and Soft Actor-Critic (SAC). For such environments, TD3 is an improved, more complex version of DDPG; and PPO has more stable updates but requires more training. SAC is an enhanced, more complex version of DDPG that generates stochastic policies. Therefore, in this multi-mode HEV powertrain system, the DDPG method has been employed to design the agent of DRL-based EMS in the automotive field.

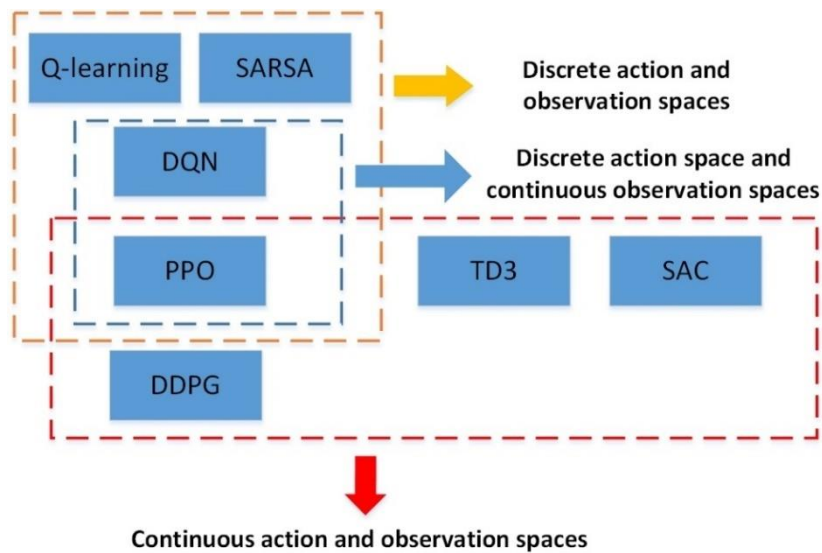


Fig. 5. Types of agents and application cases

2.5 DDPG-based optimal algorithm

The optimization problem can be formulated by

$$\begin{aligned}
 & \text{Minimize } P_{\text{loss}}(\mathbf{u}_{\text{eng}}, \mathbf{u}_{\text{mot1}}, \mathbf{T}_{\text{dem}}) \\
 & \text{s.t. } \begin{cases} \text{Loss}_{\text{eng}}(t) = m_f(t) \cdot H_f - \frac{w_{\text{eng}}(t) \cdot T_{\text{eng}}(t)}{9550} \\ \text{Loss}_{\text{batt}}(t) = R \cdot I_{\text{batt}}(t)^2 \\ \text{SoC}(t) = \text{SoC}(0) - \frac{\int_0^t I_{\text{batt}}(t) dt}{Q_{\text{batt}}} \\ \text{SoC}^- \leq \text{SoC}(t) \leq \text{SoC}^+ \end{cases} \quad (13) \\
 & \begin{cases} P_{\text{batt}_{\min}} \leq P_{\text{batt}} \leq P_{\text{batt}_{\max}} \\ 0 \leq T_{\text{eng}}(w_{\text{eng}}) \leq T_{\text{eng}_{\max}}(w_{\text{eng}}) \\ T_{\text{mot1}_{\min}}(w_{\text{mot1}}) \leq T_{\text{mot1}}(w_{\text{mot1}}) \leq T_{\text{mot1}_{\max}}(w_{\text{mot1}}) \\ T_{\text{mot2}_{\min}}(w_{\text{mot2}}) \leq T_{\text{mot2}}(w_{\text{mot2}}) \leq T_{\text{mot2}_{\max}}(w_{\text{mot2}}) \end{cases}
 \end{aligned}$$

The DDPG-based algorithm is shown in the following flow diagram in Fig.6. The outer loop is in charge of the number of training episodes, while the inner loop is in charge of the EMS control within the whole predefined driving cycle for a single training episode[29].

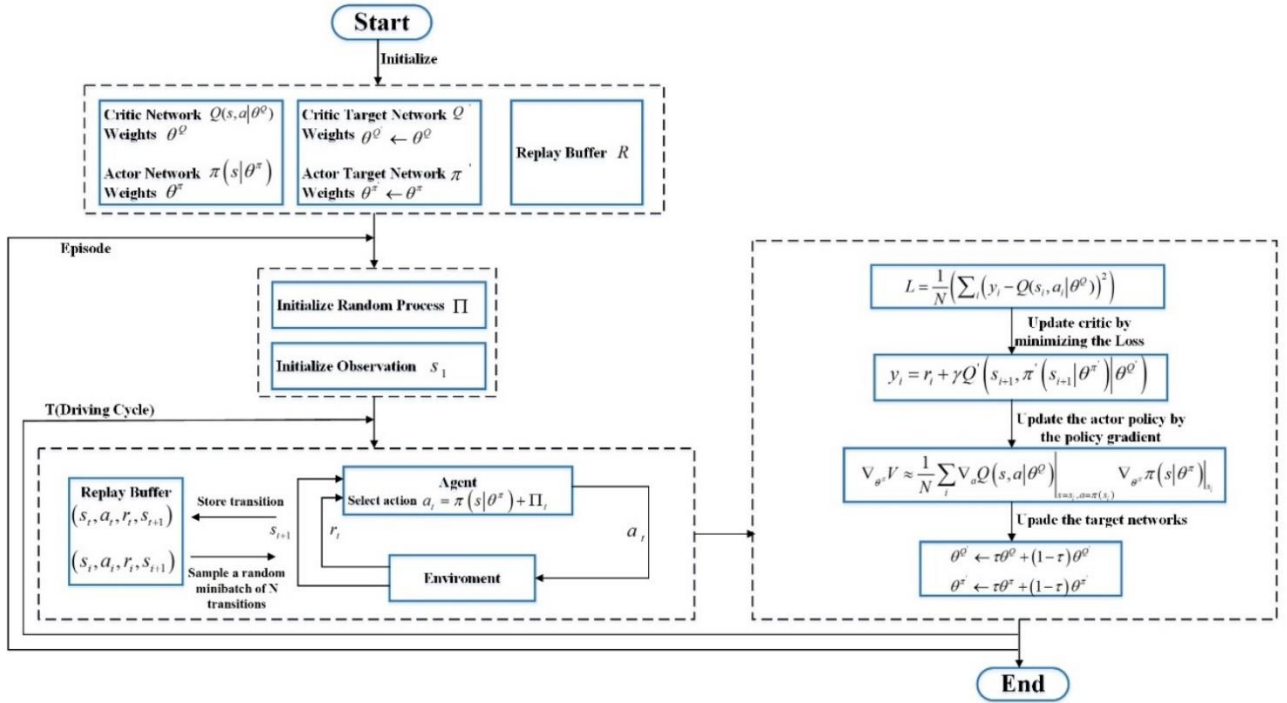


Fig. 6. DDPG-based control algorithm flow diagram

In RL, the agent receives an observation s_t , performs an action a_t , and is rewarded with a scalar reward r_t at each time step t . A policy μ , which maps states s_t (assuming the environment is fully observed, that is, $s_t = x_t$) to a probability distribution across actions a_t , defines an agent's behavior. The sum of discounted future reward R_t with a discounting factor γ of $[0, 1]$ is the return from a state:

$$R_t = \sum_{i=t}^T \gamma^{(i-t)} r_i(s_i, a_i) \quad (14)$$

The return is stochastic and is decided by the actions chosen and the policy, and the goal in RL is to obtain an approach that maximizes the expected return. Then the action-value function $Q^\mu(s_t, a_t)$, representing the expected return after taking an action a_t when in state s_t and then is described as:

$$Q^\mu(s_t, a_t) = E_{r_{i \geq t}, s_{i \geq t} \sim E, r_{i \geq t} \sim \pi} [R_t | s_t, a_t] \quad (15)$$

Then according to the Bellman equation, the recursive relationship in RL is described as:

$$Q^\mu(s_t, a_t) = E_{r_t, s_{t+1} \sim E} [r(s_t, a_t) + \gamma E_{a_{t+1} \sim \pi} [Q^\mu(s_{t+1}, a_{t+1})]] \quad (16)$$

The expectation is only decided by the environment when the policy μ is deterministic as a function $\pi : S \leftarrow A$, and a commonly used off-policy method Q^π is to choose the action maximizing $Q(s_t, a_t)$, which can be expressed as:

$$Q^\pi(s_t, a_t) = E_{r_t, s_{t+1} \sim E} [r(s_t, a_t) + \gamma Q^\pi(s_{t+1}, \pi(s_{t+1}))] \quad (17)$$

where

$$\pi(s) = \arg \max_a Q(s, a) \quad (18)$$

Therefore, based on the temporal difference(TD), the function approximators are parameterized by θ^Q are optimized by minimizing the loss L , which is expressed as:

$$L(\theta^Q) = E_{s_t \sim \pi, a_t \sim \pi, r_t \sim E} [(Q(s_t, a_t | \theta^Q) - y_t)^2] \quad (19)$$

where

$$y_t = r(s_t, a_t) + \gamma Q(s_{t+1}, \pi(s_{t+1}) | \theta^Q) \quad (20)$$

To obtain the significant and non-linear function approximators for learning action-value functions, the neural networks as function approximators are adopted; further, to scale Q-learning, two improvements are introduced, including the experience replay buffer and the target network for calculating y_t . Experience replay buffer R is used to store the transitions(e.g., a batch of state, action, reward, and next state, (s_t, a_t, r_t, s_{t+1})) at each time step in a data experience pool to avoid the strong correlations between the samples in a short period of conventional RL. And the replay buffer can be pretty significant, allowing the system to learn over many unrelated transitions since DDPG is an off-policy algorithm. N random samples of experience are selected from the experiment pool and utilized to train the networks at different times, in which batch normalization is used to scale the samples, so they are in a minibatch to have unit mean and variance.

Obtaining the greedy policy in continuous action spaces involves optimization at each time step, which is too slow to be practical with large, unconstrained function approximators and nontrivial action spaces. Instead, the Deterministic Policy Gradient(DPG)-based actor-critic method is employed, in which a parameterized actor function $\pi(s | \theta^\pi)$ determines the current policy by deterministically mapping states to a specific action; the critic function $Q(s, a | \theta^Q)$ is updated using the Bellman equation, and the actor is updated by applying the chain rule to the expected return:

$$\nabla_{\theta^\pi} V \approx \frac{1}{N} \sum_i \nabla_a Q(s, a | \theta^Q) \Big|_{s=s_i, a=\pi(s_i)} \nabla_{\theta^\pi} \pi(s | \theta^\pi) \Big|_{s=s_i} \quad (21)$$

Since the network $Q(s, a | \theta^Q)$ is updated and used to gain the target value y_t , the Q update is prone to be unstable in many environments; the target networks are employed, which means a copy of the actor and critic networks $Q'(s, a | \theta^{Q'})$ and $\pi'(s | \theta^{\pi'})$ respectively, are adopted to calculate the target value. Then making them slowly keep track of the learned networks, $\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$ with $\tau \ll 1$, is used to update the weights of these target networks; that is, the target networks change slowly and significantly improve the stability of learning.

Exploration is one of the most challenging problems of the learning process in continuous action domains. Off-policies methods, like DDPG, provide the advantage of separating the exploration problem from the learning process. An exploration policy π' by combining the actor policy with a noise process Π , which is described as:

$$\pi'(s_t) = \pi'(s_t | \theta_t^\pi) + \Pi \quad (22)$$

An Ornstein-Uhlenbeck (OU) process is chosen as a noise process, in which the decay rate β (how "strongly" the system reacts to perturbations) and the variation σ of the noise should be set and tuned, as shown in Fig.7. Since the OU process is time-series related and can be used to generate temporally correlated exploration in the action selection process of the previous step and the next step of RL to improve the exploration efficiency of control systems.

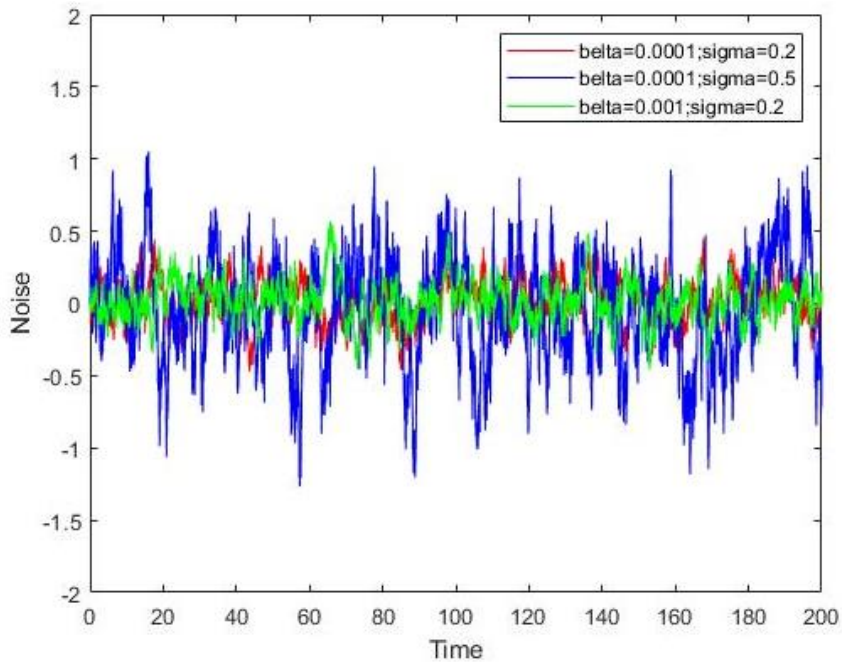


Fig. 7. Ornstein-Uhlenbeck (OU) noise process under different parameters

3 Collaborative cyber-physical learning with multi-agents

3.1 Multiple inputs and multiple-output (MIMO) synchronization optimization design

When the optimization problem described in Eq. (13) has been solved by the DDPG algorithm in a multi-mode HEV powertrain system, the action gained by the DDPG algorithm is taken to control the MG1 torque output to achieve the energy-saving within the range of SOC. The efficiency of MG2 has not been considered to achieve synchronous optimization. Therefore, multiple inputs(the demand torque and the battery SOC) and multiple-output (the control variables of MG1 and MG2) synchronization optimization problems are described to obtain the optimal energy consumption and the charge-sustaining.

$$\begin{aligned}
 & \text{Minimize } P_{loss}(\mathbf{u}_{eng}, \mathbf{u}_{mot1}, \mathbf{u}_{mot2}, \mathbf{T}_{dem}) \\
 & s. t. \begin{cases} Loss_{eng}(t) = m_f(t) \cdot H_f - \frac{w_{eng}(t) \cdot T_{eng}(t)}{9550} \\ Loss_{batt}(t) = R \cdot I_{batt}(t)^2 \\ SoC(t) = SoC(0) - \frac{\int_0^t I_{batt}(t) dt}{Q_{batt}} \\ SoC^- \leq SoC(t) \leq SoC^+ \end{cases} \\
 & \begin{cases} P_{batt_{min}} \leq P_{batt} \leq P_{batt_{max}} \\ 0 \leq T_{eng}(w_{eng}) \leq T_{eng_{max}}(w_{eng}) \\ T_{mot1_{min}}(w_{mot1}) \leq T_{mot1}(w_{mot1}) \leq T_{mot1_{max}}(w_{mot1}) \\ T_{mot2_{min}}(w_{mot2}) \leq T_{mot2}(w_{mot2}) \leq T_{mot2_{max}}(w_{mot2}) \end{cases}
 \end{aligned} \tag{23}$$

3.2 The collaborative cyber-physical learning process with multi-agents

Based on the above problem, collaborative cyber-physical learning with two agents is proposed to control the torque output of MG1 and MG2 simultaneously and reduce the fuel economy. And the DDPG-based control algorithm with multi-agents is shown in Fig.8.

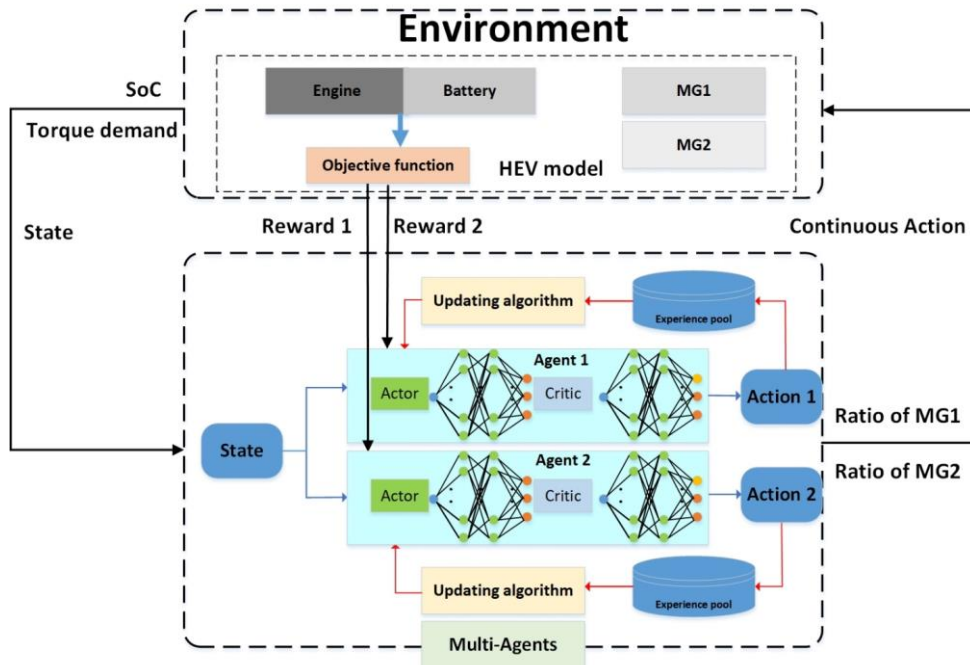


Fig. 8. DDPG-based control algorithm with multi-agents

In Section 2.5, the control signal of MG1 is calculated and optimized by the actor-critic method. The output torque of MG2 is directly obtained from the mechanical characteristic curve for the speed and the torque. In the multi-mode HEV powertrain system, as shown in Fig. 4, there is enough power for MG2 to drive or brake the vehicle under most typical scenarios. Instead, the output torque of MG2 needs to be optimized under different cases, and the DDPG algorithm is also adopted to achieve the control signal of MG2. Two DDPG agents are designed, as described in Fig.8 to reduce energy consumption, and the learning process is working as follows:

(1) The HEV's current state $s(t) = [T_{dem}(t), SoC(t)]$, including the total torque demand and the battery SoC, are observed and then transmitted to the multi-agents module with the blue line in Fig.8, of which two agents are the same and share.

(2) Similar to the DDPG scheme applied to the control of the MG1, the two continuous actions $a(t) = [u_{mot1}(t), u_{mot2}(t)]$ are executed and calculated by critic and actor networks updated respectively based on the DDPG-based optimal algorithm, as shown in the red line. Then transmitted as two signals to the MG1 and MG2 respectively in the multi-mode HEV model with the black line, where the MG1 and MG2 output torque ratio $u_{mot1}(t)$ and $u_{mot2}(t)$ are designed, followed by

$$a(t) = [u_{mot1}(t), u_{mot2}(t)] \quad (24)$$

where, $u_{mot2}(t) \in \mathbf{U}$, $\mathbf{U} = [0 \ 1]$. Further, the hyperparameters of the actor and critic networks for two agents, including the weights, the learning rate and the size of networks, and the size of their experience pool, are changed simultaneously and shared, but in one agent, the hyperparameters of the critic and actor are different.

(3) Within the multi-mode HEV powertrain model, the torque allocation relationship for the output shaft, according to Eq.11, can be obtained to achieve the multi-mode architecture as presented in Fig.1 using

$$\begin{aligned} T_{dem} &= u_{eng}(t) \cdot T_{eng_max} + u_{mot2}(t) \cdot T_{mot2_max} \\ u_{eng}(t) &= \frac{u_{mot1}(t) \cdot T_{mot1_max} + T_{GB}}{T_{eng_max}} \end{aligned} \quad (25)$$

(4) For the environment, the torque allocation becomes complex due to the increase of the control variables, therefore, the reward assessment of the multi-agents will differ from that of the single agent. Then since the relationship between the two agents is fully cooperative in this paper, their reward calculations are both prone to achieving the global objective r_{global} , which is denoted by

$$r_{global} = -P_{loss}(t) \quad (26)$$

The reward assessments for two agents evaluate whether the current action is appropriate for the current vehicle states under the predefined driving cycles, which means that the different objectives, including energy-saving and the charge-sustaining, are considered. Therefore, two different reward functions for two agents are provided to achieve the multi-objective optimization, and Reward 1 for MG2 with Agent2 is exhibited to reduce the engine energy consumption of the internal combustion

engine(ICE), instead, Reward 2 for MG1 with Agent1 is calculated by considering the battery SOC, which is denoted by

$$\begin{cases} \text{Reward 1: } r_{local,2} = -\alpha Loss_{eng}(t) \\ \text{Reward 2: } r_{local,1} = -\beta Loss_{batt}(t) \end{cases} \quad (27)$$

Local penalties received by Agent1 and Agent2 based on their separate objectives are $r_{local,1}$ and $r_{local,2}$. Therefore, the rewards r_1 and r_2 received by Agent1 and Agent2, respectively, are expressed as

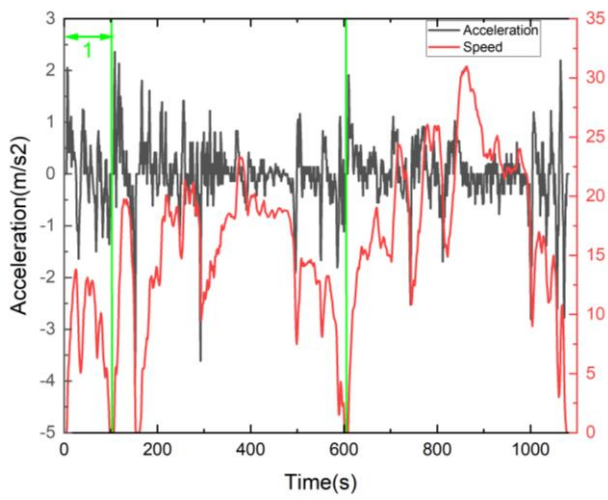
$$\begin{cases} r_1 = r_{global} \mp r_{local,1} \\ r_2 = r_{global} \mp r_{local,2} \end{cases} \quad (28)$$

By adjusting the relationship or ratio between r_{global} and $r_{local,1}$ and $r_{local,2}$, The rule of different rewards can be made, which means the contributions of two agents to the goal objective will differ.

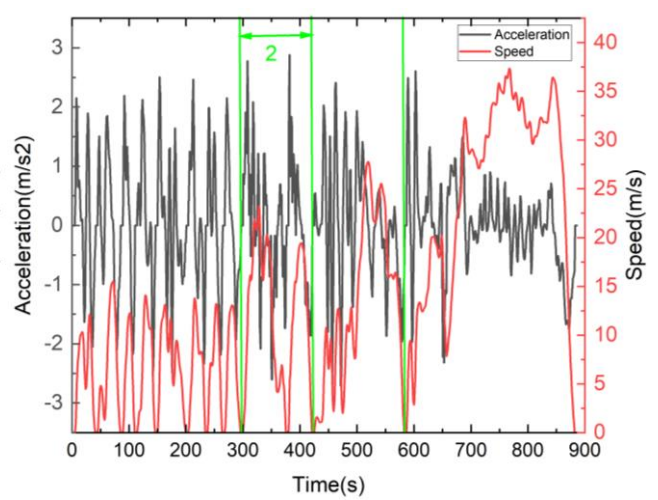
(5) The continuous actions at each time step are optimized by the updated control policy to adapt to different real-world driving conditions. A new round of the collaborative learning process will start again.

3.3 Driving Cycles Setup

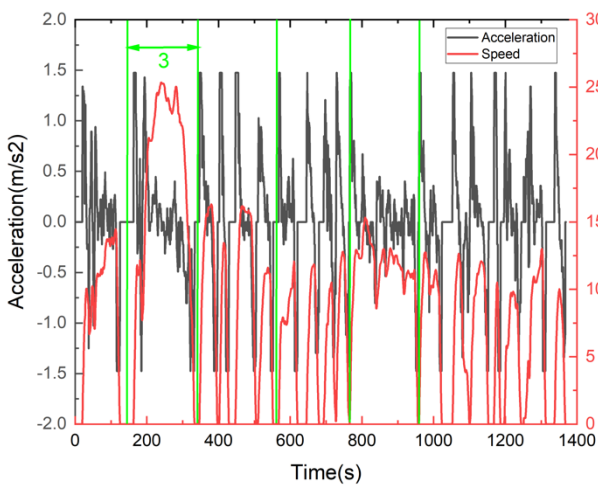
To speed up the learning process and enhance the generalization of the presented algorithms, the predefined learning driving cycle for training agents is essential to improve the learning performance of DRL-based algorithms. This paper's random learning driving cycle is designed with features, mainly including the maximum acceleration and speed from four standard driving cycles. First, the Artemis Rural, RTS95, UDDS, and WLTP Driving Cycle are adopted, as shown in Fig.9. Then, four standard driving cycles are separately divided into a series of segmented driving cycles according to the low-speed, medium-speed, and high-speed phase, as shown in Fig.9 with the green line. Finally, from four divided standard driving cycles, considering one-speed Phase 2 with the maximum acceleration, the other three speed phases with the above three different types of segmented driving cycles (Phase 1 represents the low-speed, Phase 3 represents the medium-speed, Phase 4 represents the high-speed), the learning driving cycles will be combined randomly with four phases from 1-4, as shown in Fig.10, which are the same under the single agent and multi-agents. In Fig.10, Phase 2 is with the maximum acceleration, and Phase 1, 3, and 4 are with the three different types.



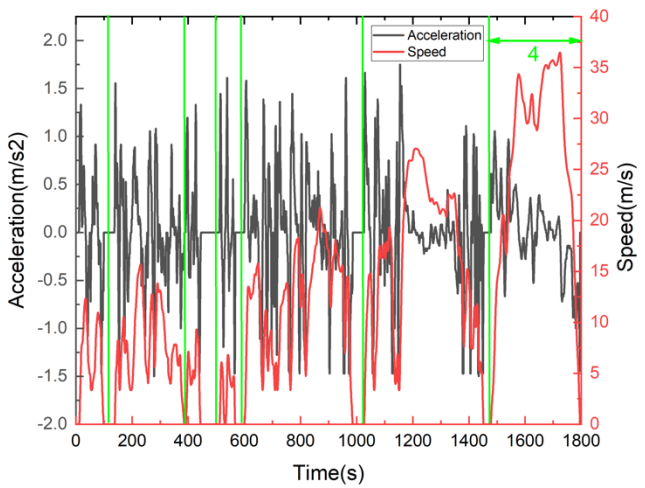
(a) Artemis Rural Driving Cycle



(b) RTS95 Driving Cycle



(c) UDDS Driving Cycle



(d) WLTP Driving Cycle

Fig. 9. Four standard driving cycles

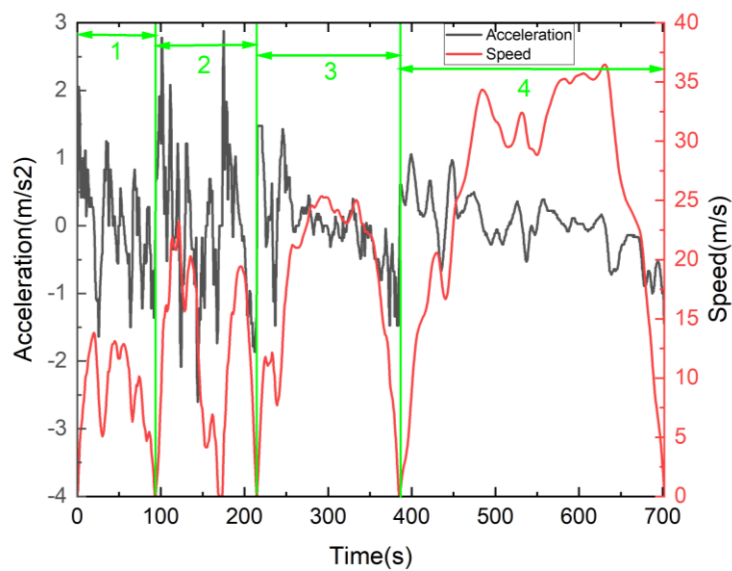


Fig. 10. Learning Driving Cycles

4 Results and Discussion

This section presents the results and comparisons of the DDPG-based algorithm with the single-agent and multi-agents for the multi-mode EMS. And then, the HEV model setup is first conducted in Matlab/Simulink, then the DDPG-based RL with the single-agent and the multi-agent are proposed, respectively, and the driving cycle is designed to train the agent of RL.

4.1 The sensibility of learning analysis with the single-agent method

DDPG-based EMS of multi-mode HEV with a single-agent method has been investigated, and different parameters of the presented algorithm should be compared and determined first. Network designs of the actor and critic in DDPG are an essential step to enhancing the learning performance of the DRL-based method; therefore, different network depths are conducted to obtain the optimal combination with the network architecture of the critic and actor. Then different learning rate is studied to improve the learning performance and learning efficiency; finally, the policy noise with the OU process is provided to enhance the exploration efficiency of the whole learning procedure in RL. In this Section, SOC initial value is set to 0.28, and further parameter analysis is conducted solely with the single method under the predefined driving cycle. Further, the optimal parameters calculated by comparing different cases are employed in the multi-agent system directly under learning driving cycles and other scenarios.

a) Networks layers

The network architecture should be researched and chosen under the single-agent case. The depth of the networks is changed and is mainly considered one of the essential influence factors in this paper. First, the parameters of a single DDPG agent are provided in Table. 2.

Table. 2 Hyperparameters of DDPG

Parameters	Values
Batch size	64
Discount factor	0.99
Optimizer	Adam
Learning rate of actor	1×10^{-4}
Learning rate of critic	1×10^{-4}
Experience Buffer Length	1×10^5
The decay rate of the policy noise	0.2

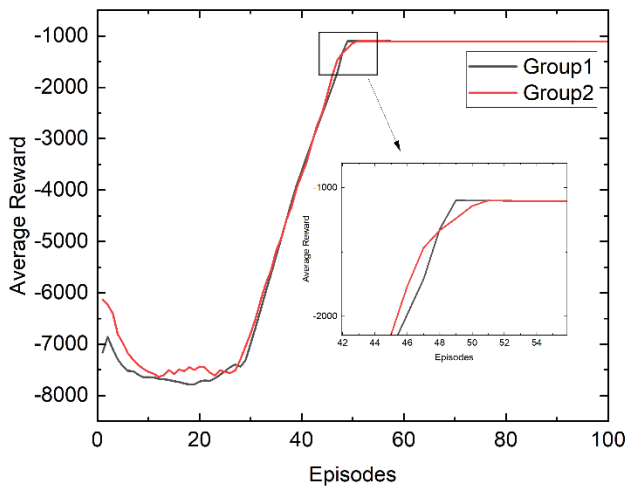
The variation of the policy noise	1×10^{-4}
-----------------------------------	--------------------

Then the two network architectures of the actor and the critic are described in Table.3. And Group 1 is set as the baseline to validate that network layers can affect the learning performance. The reduction of the critic network depth has been executed to be Group 2. However, the optimal combination cannot be gained.

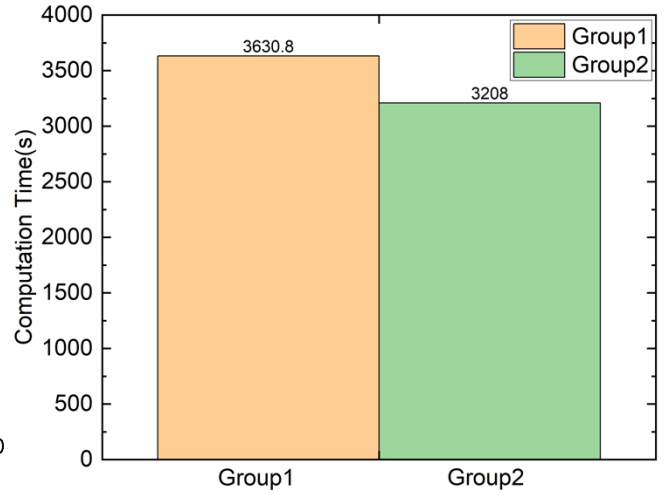
Table. 3 Networks designs with different depths

Group	Network	Layers	Size	Regularization Factor
Group1	criticNetwork	InputLayer	2	0.0001
		FullyConnectedLayer	50	
		ReLULayer	50	
		FullyConnectedLayer	25	
		ReLULayer	1	
		FullyConnectedLayer	25	
		ReLULayer	1	
		FullyConnectedLayer	1	
	actorNetwork	InputLayer	2	0.0001
		FullyConnectedLayer	64	
		ReLULayer	64	
		FullyConnectedLayer	32	
		FeatureInputLayer	1	
		FullyConnectedLayer	32	
		AdditionalLayer	2	
		TanhLayer	1	
		FullyConnectedLayer	1	
Group2	criticNetwork	InputLayer	2	0.0001
		FullyConnectedLayer	50	
		ReLULayer	50	
		FullyConnectedLayer	1	
	actorNetwork	InputLayer	2	0.0001
		FullyConnectedLayer	64	
		ReLULayer	64	
		FullyConnectedLayer	32	
		FeatureInputLayer	1	
		FullyConnectedLayer	32	
		AdditionalLayer	2	
		TanhLayer	1	
		FullyConnectedLayer	1	

When two groups of parameters are adopted respectively under the learning driving cycle, the average reward, the computer time, SOC of the battery, and the fuel consumption(L/100km) are given in Fig.11 and Table. 5.



(a) The average rewards



(b) Computation time

Fig. 11. The results of different networks layers

Table. 4 Comparison of SOC and Fuel consumption

Group	Initial SOC	End SOC	Fuel Consumption(L/100km)
Group1	0.28	0.318	4.826
Group2	0.28	0.311	4.620

In DL theory, the network depths represent the complexity of the trained objects; for the hybrid electrics, the environment is less complicated than the image identification field or other automated fields. By comparing the learning results of different network architectures, given in Fig. 11 and Table.4, Group 2 is smoother than Group1 in average rewards and better than that in charge-sustaining and fuel saving. The main reason is that Group 1 can be slightly overfitting due to more depths. On the other hand, compared with the computer time, the cost of Group 2 is less 11.6 % than that of Group1. From the aspect of the learning performance, the end of SOC is close to the same initial SOC; however, the fuel-saving is achieved at a 4.3% improvement by Group2. Thus, the better network architecture Group2 of the actor and the critic will be employed under the single-agent and multi-agent cases directly.

b) Learning rate

Different learning rates of the networks have a crucial influence on the learning performance, especially since there are two networks for the critic and the actor in the DDPG method, which can be the same or the different learning rates. Therefore, five groups of parameters are designed to validate the impact of the learning rates, as shown in Table.4.

Table. 5 Learning rates of the critic and the actor

Networks	Group	Actor	Critic
Learning rate	Group1	1×10^{-4}	1×10^{-4}
	Group2	1×10^{-3}	1×10^{-3}
	Group3	1×10^{-4}	1×10^{-3}
	Group4	1×10^{-3}	1×10^{-4}
	Group5	1×10^{-5}	1×10^{-5}

The network architecture of Group 2 is adopted to validate the influence of the learning rate. Then by comparing five groups of learning rates of the actor and the critic networks under the learning driving cycle, the average reward, the computer time, SOC of the battery, and the fuel consumption (L/100km) are obtained in Fig.12 and Table. 6.

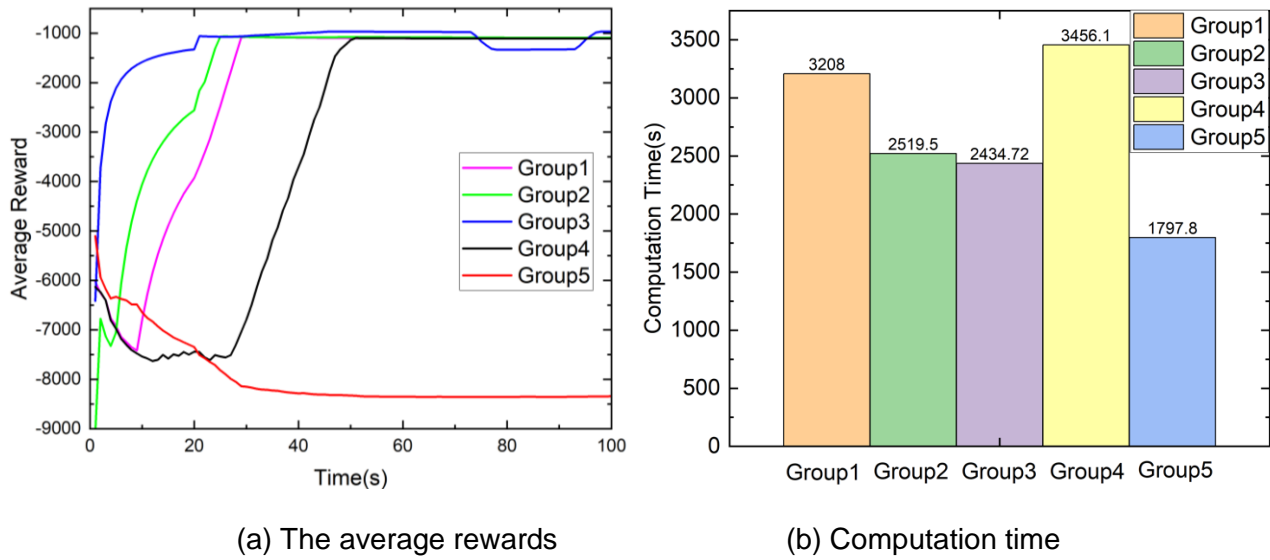


Fig. 12. The results of different learning rates

Table. 6 Comparison of SOC and Fuel consumption

Group	Initial SOC	End SOC	Fuel Consumption(L/100km)
Group1	0.28	0.311	4.620
Group2	0.28	0.294	4.547
Group3	0.28	0.303	4.589

Group4	0.28	0.316	4.726
Group5	0.28	---	---

From Fig.12, it is apparent different learning rates are crucial for learning and produce a distinguished impact on learning by updating networks. The learning rate is not as low as possible, as shown in Group 5. Although the computation time is least among all the Groups, the learning procedure worsens, which needs to be adjusted according to the network architecture. The reason for the least computation cost is that the unknown and known experiences are not learned by the agent during the exploration and exploitation. Then Group 4 has the longest computation burden, and it costs the longest time to achieve the convergence as well. Compared with Group 1, 2, and 3, the computation cost of Group 1 is similar to Group 4, which illustrates that the learning rate of the critic network has less impact on the learning performance than that of the actor-network. Group 3 is the least computation time and is the fastest to reach the coverage point. Still, it is more unstable than Group 2. To conclude, Group 2 will be the best option for the actor and the critic network.

c) Policy noise

The exploration in RL is one of the most significant problems, and it is worthy of optimization to choose an action with what probability at each step. Ant OU process can provide the temporal relationship between the previous step and the next step during the exploration to enhance efficiency when the agent takes action. Further, the network architecture of Group 2 and the learning rates of Group 2 are used to conduct the comparison of the policy noise combination.

Table. 7 Policy noise combination

OU process	Belta	Sigma
OU1	1×10^{-4}	0.2
OU2	1×10^{-4}	0.5
OU3	1×10^{-3}	0.2

Then three groups of policy noise combinations are used respectively under the learning driving cycle; the average reward, SOC of the battery, and the fuel consumption(L/100km) are given in Fig.13 and Table. 8, and the computer time is similar.

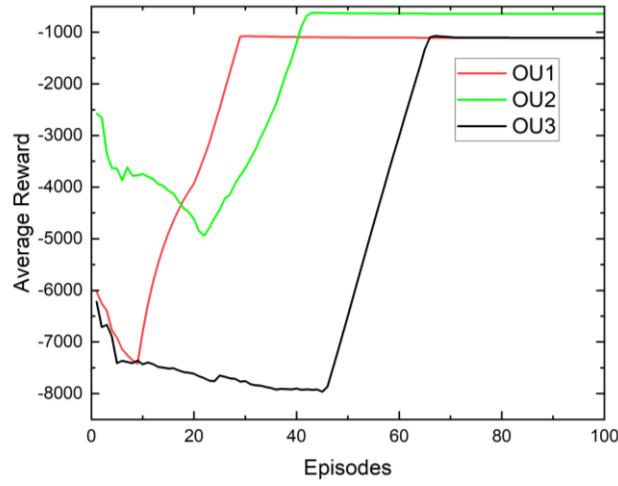


Fig. 13. The average rewards of different policy noise

Table. 8 Comparison of SOC and Fuel consumption

Group	Initial SOC	End SOC	Fuel Consumption(L/100km)
OU1	0.28	0.294	4.547
OU2	0.28	0.313	4.689
OU3	0.28	0.309	4.625

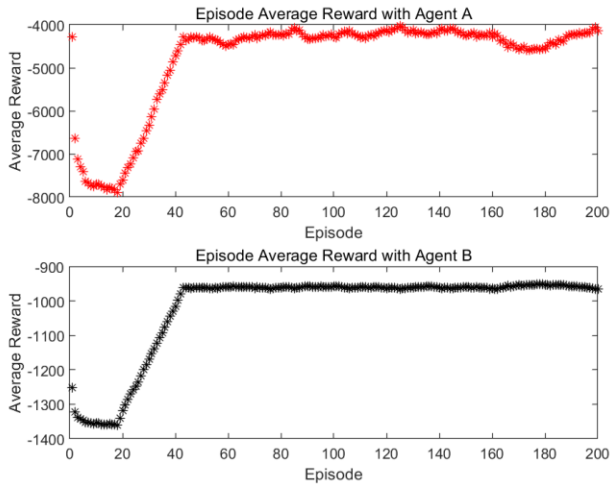
By comparing the results, the OU1 is the best since it can achieve the fastest convergence, and the learning results with the charge-sustaining and fuel-saving are a little superior to the other two groups. In the meanwhile, the results show that the Beta parameter of the OU process can impact the speed of the learning coverage by comparing with Group 1 and 3, and the Sigma parameter can have a primary influence on the stable point in contrast with Group 1 and 2.

4.2 Level of learning performance with the different relationships of multi-agents

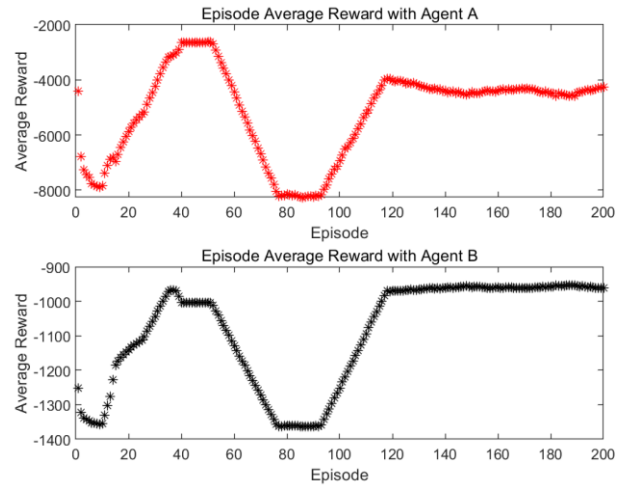
After determining the main hyperparameters of the single-agent RL, the multi-agent method employed two same DDPG agents. In the paper, the two DDPG agents in the collaborative cyber-physical learning system share the same network parameters, including the network layers, the learning rate, and other hyperparameters. Then in multi-agent RL(MARL), the majority of MARL settings use global rewards rather than explicitly disentangling reward systems for various agents, which presents a challenge when training[30]. Aiming to translate this global reward into a local reward for each agent, the contributions of each agent concerning global reward are analyzed. Therefore, the respective rewards of the two agents are described as:

$$\begin{cases} r_1 = Ratio * r_{global} + r_{local,1} \\ r_2 = Ratio * r_{global} + r_{local,2} \end{cases} \quad (29)$$

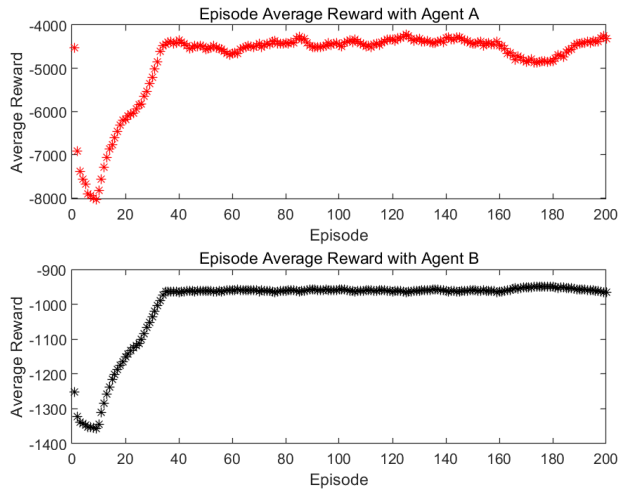
The Ratio value will change from 0 to 0.9 at 0.1 intervals; then the different relationships of multi-agent study for the level of learning performance have been investigated as shown in Fig.14, which displays the average rewards with Agent A and Agent B under 200 episodes from 0 to 0.9.



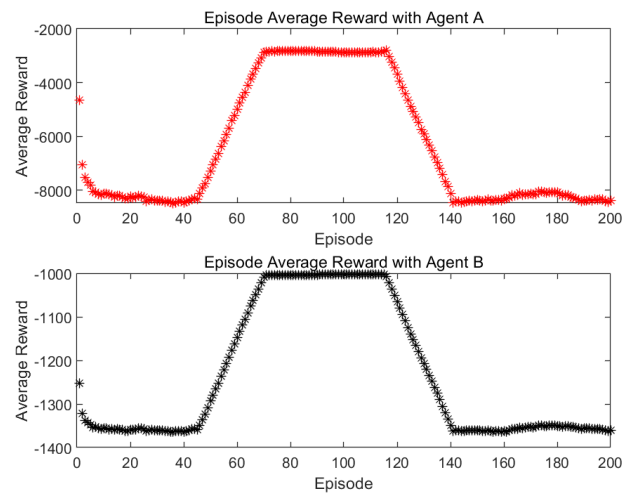
(1) Ratio = 0



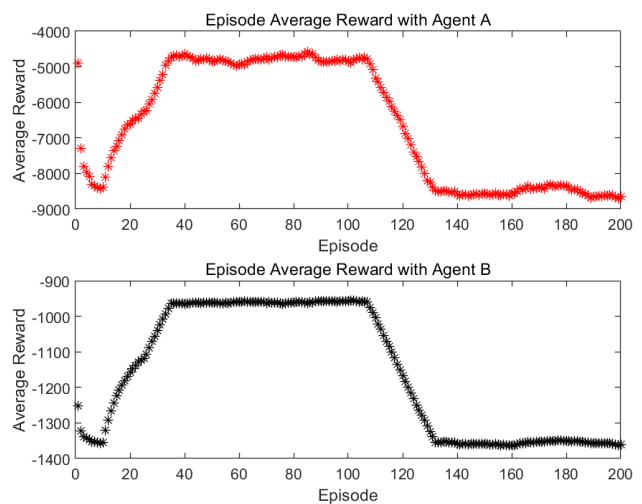
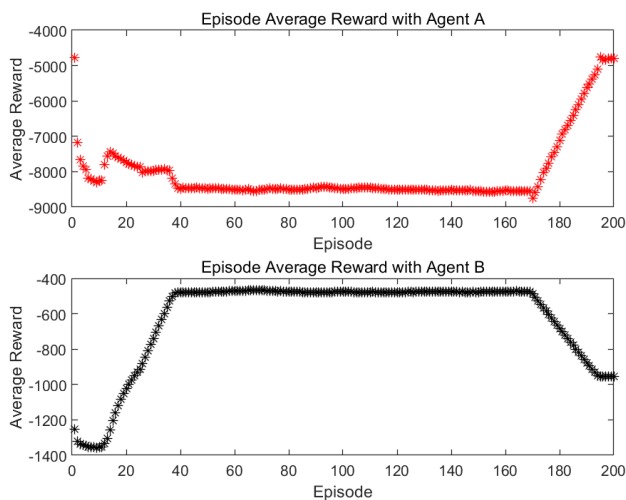
(2) Ratio = 0.1



(3) Ratio = 0.2



(4) Ratio = 0.3



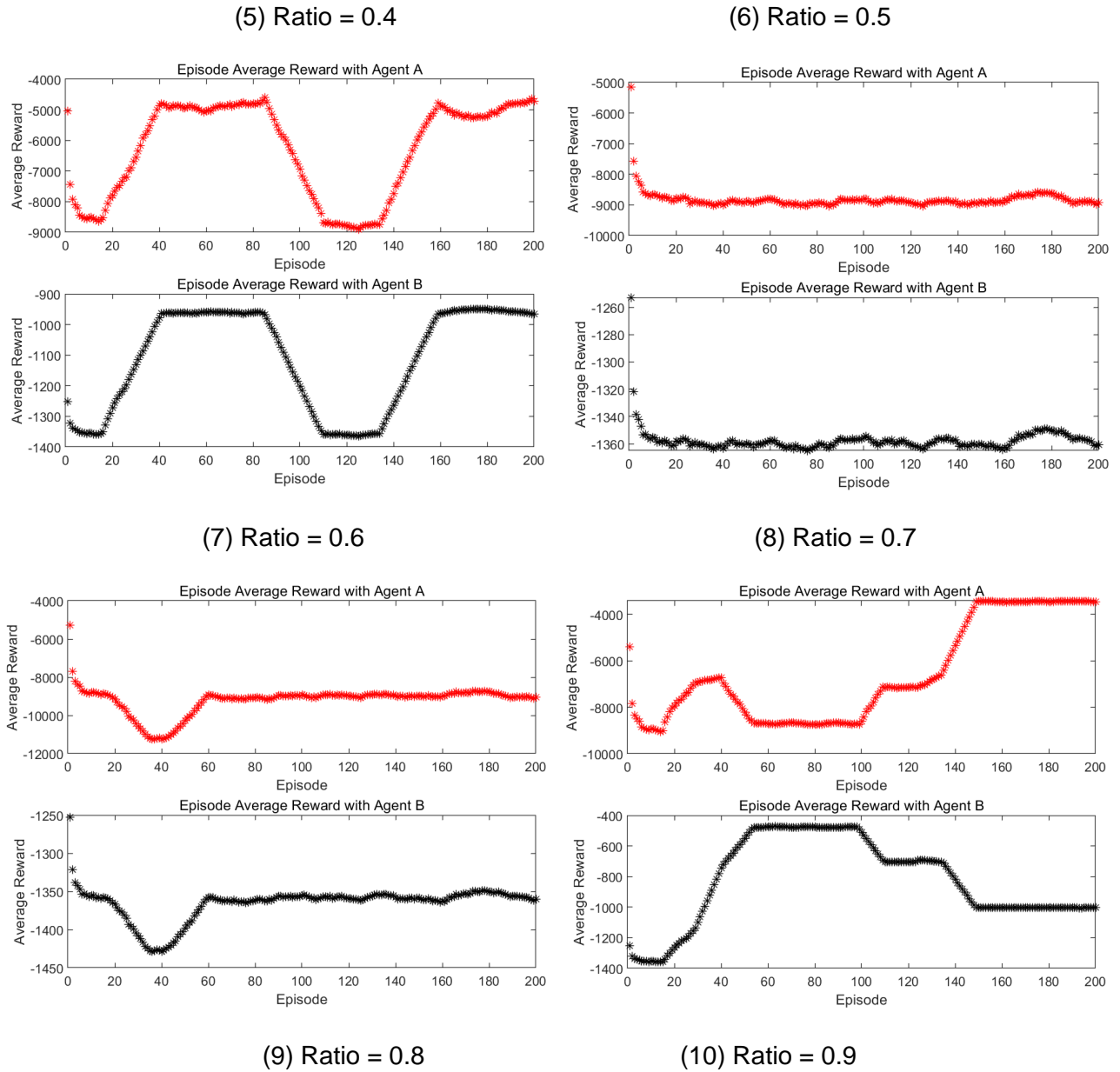


Fig. 14. Learning performance of different weights (SOC initial value is 0.28)

From Fig.14, we can find that Agent A and Agent B have a similar tendency for the average reward in most cases, except for Ratio 0.4 and 0.9, where two agents cannot reach a state of convergence and are trained in the exact opposite direction. The results are undesirable and unacceptable under these relationships, since the cooperative relationship, in which Agent A and Agent B can achieve the global goal in a union, is expected in this paper. Similarly, when the Ratio is 0.1, 0.3, 0.5, 0.6, 0.7, and 0.8, although two agents are trained to follow the same trend, they are all at unstable states and cannot converge at the steady point, therefore, these relationships between the global reward and the local reward are unreasonable. Then, when Ratio 0 and 0.2, two agents perform well and achieve the fast convergence at about 40 episodes and have the same tendency to reach the common goal, which reflects that two agents tend to be independent to ensure their stability. By calculating average values and standard deviation under Ratio 0(a completely independent relationship) and 0.2(not a completely independent relationship) cases, as shown in Table.9, when

the Ratio is 0.2, two agents are more stable to realize the global objective jointly since the standard deviation is smaller; in the meanwhile, the two agents have the same standard deviation under the same Ratio case.

Table. 9 Average values and standard deviation under Ratio 0 and 0.2 case

Ratio	Agent	Average value (200 episodes)	Standard deviation(after 40 episodes)
0	Agent A	-4782.1	48.064
	Agent B	-1018.8	48.064
0.2	Agent A	-4803.3	46.621
	Agent B	-999.1	46.621

To conclude, the relationship between the rewards of different agents, which represent the contributions to the overall goal for each agent, is also essential and complicated in MARL. In this paper, the not completely independent relationship with Ratio 0.2 is effective and set to realize the energy-efficient control for multi-mode HEV.

4.3 Comparison analysis for the learning performance with single-agent and multi-agents

After the main parameters of the DDPG-based method have been investigated and the relationship ratio between the multi-agents has been determined. The comparison is made to verify that the multi-agent system is more efficient and effective with parallel calculations than the single-agent. First, by designing the learning driving cycle to speed up the training process, a comparison test with two methods is conducted to validate the improvement of the learning performances (including the fuel consumption and the charge-sustaining of the battery). Then, to validate the adaptivity of the presented DDPG-based algorithms with the single-agent and multi-agent, four different standard driving cycles are adopted to test the learned agent trained through the predefined learning cycles. In the end, the learning performance of the multi-mode HEV is compared and described in Table.10 with two methods, where the SOC and fuel consumption(L/100km) are obtained and the error of SOC and saving are calculated by

$$SOC_{error} = \frac{|SOC_{End} - SOC_{Initial}|}{SOC_{Initial}} * 100\% \quad (30)$$

$$Saving = \frac{|Fuel_{Multi-agent} - Fuel_{Single-agent}|}{Fuel_{Single-agent}} * 100\% \quad (31)$$

Table. 10 Learning performance of the single-agent and multi-agent case

Driving Cycle`	Initial SoC	Method	End SoC	SOC Error(%)	Fuel/100km (L/100km)	Saving	Average Improvement
----------------	-------------	--------	---------	--------------	----------------------	--------	---------------------

Learning driving cycle	0.25	Single-agent	0.272	8.80	4.534	-	2.408
	0.25	Multi-agent	0.241	3.60	4.419	2.538	
	0.28	Single-agent	0.305	8.93	4.547	-	
	0.28	Multi-agent	0.271	3.21	4.450	2.130	
	0.30	Single-agent	0.328	9.33	4.534	-	
	0.30	Multi-agent	0.286	4.67	4.418	2.554	
Artemis Rural driving cycle	0.25	Single-agent	0.274	9.60	4.195	-	2.525
	0.25	Multi-agent	0.241	3.60	4.107	2.084	
	0.28	Single-agent	0.304	8.57	4.196	-	
	0.28	Multi-agent	0.269	3.93	4.081	2.746	
	0.30	Single-agent	0.324	8.00	4.196	-	
	0.30	Multi-agent	0.293	2.33	4.081	2.746	
RTS95 driving cycle	0.25	Single-agent	0.265	6.00	5.024	-	1.006
	0.25	Multi-agent	0.253	1.20	4.992	0.631	
	0.28	Single-agent	0.306	9.29	5.024	-	
	0.28	Multi-agent	0.271	3.21	4.961	1.263	
	0.30	Single-agent	0.326	8.67	5.024	-	
	0.30	Multi-agent	0.281	6.33	4.968	1.124	
UDDS driving cycle	0.25	Single-agent	0.262	4.80	4.430	-	2.309
	0.25	Multi-agent	0.261	4.40	4.351	1.770	
	0.28	Single-agent	0.292	4.29	4.430	-	
	0.28	Multi-agent	0.29	3.57	4.306	2.805	
	0.30	Single-agent	0.312	4.00	4.431	-	
	0.30	Multi-agent	0.31	3.33	4.327	2.353	
	0.25	Single-agent	0.261	4.40	4.324	-	3.911
	0.25	Multi-agent	0.246	1.60	4.155	3.897	

WLTP driving cycle	0.28	Single-agent	0.293	4.64	4.324	-
	0.28	Multi-agent	0.292	4.29	4.148	4.070
	0.30	Single-agent	0.314	4.67	4.325	-
	0.30	Multi-agent	0.304	1.33	4.162	3.767

- represents no comparison

The results in Table 10 indicate that the multi-agent method leads to a significant improvement in energy saving. Compared with the single-agent, the multi-agent algorithm can achieve the two control variables synchronously, which means that the output of MG2 is not always at maximum torque output. From the aspect of adaptivity, under four standard driving cycles, the error of SOC is controlled within 10% under the single-agent, and within 5% under the multi-agent. Specifically, during the Artemis Rural and RTS95 driving cycle, the error of SOC becomes larger since the two driving cycles change radically. Instead, the other two cycles, UDDS and WLTP, are in the range of 5% with the single-agent, and within about 3% with the multi-agent. Therefore, it demonstrates that the adaptivity performance has been enhanced and the effectiveness of the learning driving cycle has been validated.

From the perspective of fuel consumption, under distinct initial SOC values, the results are slightly different, and the average improvements are calculated during the learning and standard driving cycles. We can find that the multi-agent method can realize an average improvement of 2.408%, 2.525%, 1.006%, 2.309%, and 3.911% respectively, in which there is no obvious enhancement in the RTS95 driving cycle due to the drastic characteristics of the velocity profile. And there is a maximum improvement with the smallest error of SOC in the WLTP driving cycle, which demonstrates that the learned knowledge by the learning cycle is employed the best in this driving cycle.

5 Conclusions

This paper studied a novel charge-sustaining control strategy for a multi-mode HEV based on DRL-learning algorithms. The DDPG algorithm has been developed to solve the continuous control problem for EMS, in which the multi-mode HEV model is built, and the single DDPG agent is adopted to regulate the MG1 first. Then MIMO synchronization optimization problem is presented. The multi-agent architecture based on the proposed DDPG is used to manage MG1 and MG2, respectively, with parallel computation. To improve the learning performance and speed up the learning process, a random method is proposed to define the learning driving cycle according to four main features from four typical driving cycles. Ultimately, the learning performance with the different relationships of multi-agents and the comparison with the single-agent method and multi-agent method is conducted. The conclusions drawn from the investigation are as follows:

- (1) For the single-agent control method, the network design, the learning rate, and the policy noise are crucial to the DDPG-based RL scheme. In network layers, the cost of Group 2 is less 11.6 % than that of Group1 in computer time and can achieve a 4.3% improvement in fuel saving to Group1. In a comparison of the learning rate, the learning rate of Actor and Critic are both 1×10^{-3} , which is best and can achieve the least computation time and the fastest convergence. Then for policy noise, the parameter Belta and Sigma are 1×10^{-4} and 0.2 separately, which are the best combination and can achieve the fastest convergence and the most fuel saving.
- (2) For the multi-agent control method, by analyzing different relationships of multi-agents, the local rewards of Agent A and Agent B are solely relevant to the power loss of the engine and the battery respectively, however, the global reward is to realize the total loss. Then the investigation revealed that the independent relationship is most suitable for the EMS of multi-mode HEV.
- (3) For the comparison with the single-agent and multi-agent methods, under differing initial battery SOC and typical driving cycles, the performance of the multi-agent method is robust, saving approximately 4% of total energy over the single-agent scheme.
- (4) Different learning driving cycles have a momentous impact on learning performance, which needs to be optimized in the future.

References

- [1] Ganesh AH, Xu B. A review of reinforcement learning based energy management systems for electrified powertrains: Progress, challenge, and potential solution. *Renew Sustain Energy Rev* 2022;154:111833. <https://doi.org/10.1016/j.rser.2021.111833>.
- [2] Hua M, Chen G, Zhang B, Huang Y. A hierarchical energy efficiency optimization control strategy for distributed drive electric vehicles. *Proc Inst Mech Eng Part D J Automob Eng* 2019;233:605–21. <https://doi.org/10.1177/0954407017751788>.
- [3] Zhang F, Hu X, Langari R, Cao D. Energy management strategies of connected HEVs and PHEVs: Recent progress and outlook. *Prog Energy Combust Sci* 2019;73:235–56. <https://doi.org/10.1016/j.pecs.2019.04.002>.
- [4] Wirasingha SG, Emadi A. Classification and review of control strategies for plug-in hybrid electric vehicles. *IEEE Trans Veh Technol* 2011;60:111–22. <https://doi.org/10.1109/TVT.2010.2090178>.
- [5] Odeim F, Roes J, Wülbeck L, Heinzel A. Power management optimization of fuel cell/battery hybrid vehicles with experimental validation. *J Power Sources* 2014;252:333–43. <https://doi.org/10.1016/j.jpowsour.2013.12.012>.
- [6] Zhang F, Hu X, Langari R, Cao D. Energy management strategies of connected HEVs and PHEVs: Recent progress and outlook. *Prog Energy Combust Sci* 2019;73:235–56. <https://doi.org/10.1016/j.pecs.2019.04.002>.
- [7] Lee H, Kang C, Park Y II, Cha SW. A study on power management strategy of HEV using

dynamic programming. *World Electr Veh J* 2016;8:274–80. <https://doi.org/10.3390/wevj8010274>.

- [8] Chowdhury NR, Ofir R, Zargari N, Baimel D, Belikov J, Levron Y. Optimal Control of Lossy Energy Storage Systems with Nonlinear Efficiency Based on Dynamic Programming and Pontryagin's Minimum Principle. *IEEE Trans Energy Convers* 2021;36:524–33. <https://doi.org/10.1109/TEC.2020.3004191>.
- [9] Qi X, Wu G, Boriboonsomsin K, Barth MJ. A Novel Blended Real-Time Energy Management Strategy for Plug-in Hybrid Electric Vehicle Commute Trips. *IEEE Conf Intell Transp Syst Proceedings, ITSC 2015;2015-Octob:1002–7*. <https://doi.org/10.1109/ITSC.2015.167>.
- [10] Liu C, Murphey YL. Optimal Power Management Based on Q-Learning and Neuro-Dynamic Programming for Plug-in Hybrid Electric Vehicles. *IEEE Trans Neural Networks Learn Syst* 2020;31:1942–54. <https://doi.org/10.1109/TNNLS.2019.2927531>.
- [11] Hasselt MAW and H van. Ensemble algorithms in reinforcement learning. *Mach Tool Blue B* 1974;69:90–6.
- [12] Chen Z, Hu H, Wu Y, Zhang Y, Li G, Liu Y. Stochastic model predictive control for energy management of power-split plug-in hybrid electric vehicles based on reinforcement learning. *Energy* 2020;211:118931. <https://doi.org/10.1016/j.energy.2020.118931>.
- [13] Zhang W, Wang J, Liu Y, Gao G, Liang S, Ma H. Reinforcement learning-based intelligent energy management architecture for hybrid construction machinery. *Appl Energy* 2020;275:115401. <https://doi.org/10.1016/j.apenergy.2020.115401>.
- [14] Xu B, Hou J, Shi J, Li H, Rathod D, Wang Z, et al. Learning Time Reduction Using Warm-Start Methods for a Reinforcement Learning-Based Supervisory Control in Hybrid Electric Vehicle Applications. *IEEE Trans Transp Electrif* 2021;7:626–35. <https://doi.org/10.1109/TTE.2020.3019009>.
- [15] Xu B, Tang X, Hu X, Lin X, Li H, Rathod D, et al. Q-Learning-Based Supervisory Control Adaptability Investigation for Hybrid Electric Vehicles. *IEEE Trans Intell Transp Syst* 2021;1–10. <https://doi.org/10.1109/TITS.2021.3062179>.
- [16] Xu G, He X, Chen M, Miao H, Pang H, Wu J, et al. Hierarchical speed control for autonomous electric vehicle through deep reinforcement learning and robust control. *IET Control Theory Appl* 2022;16:112–24. <https://doi.org/10.1049/cth2.12211>.
- [17] Xiong R, Cao J, Yu Q. Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle. *Appl Energy* 2018;211:538–48. <https://doi.org/10.1016/j.apenergy.2017.11.072>.
- [18] Tang X, Chen J, Liu T, Qin Y, Cao D. Distributed Deep Reinforcement learning-based energy and emission management strategy for hybrid electric vehicles. *IEEE Trans Veh Technol*

2021;70:9922–34. <https://doi.org/10.1504/IJVP.2022.119433>.

- [19] Wang P, Li Y, Shekhar S, Northrop W. Adversarial Attacks on Reinforcement Learning based Energy Management Systems of Extended Range Electric Delivery Vehicles n.d.
- [20] Hu B, Li J. A Deployment-efficient Energy Management Strategy for Connected Hybrid Electric Vehicle based on Offline Reinforcement Learning. *IEEE Trans Ind Electron* 2021. <https://doi.org/10.1109/TIE.2021.3116581>.
- [21] Lin X, Wu J, Wei Y. An ensemble learning velocity prediction-based energy management strategy for a plug-in hybrid electric vehicle considering driving pattern adaptive reference SOC. *Energy* 2021;234. <https://doi.org/10.1016/j.energy.2021.121308>.
- [22] Tang X, Chen J, Pu H, Liu T, Khajepour A. Double Deep Reinforcement Learning-Based Energy Management for a Parallel Hybrid Electric Vehicle with Engine Start-Stop Strategy. *IEEE Trans Transp Electrif* 2021;7782. <https://doi.org/10.1109/TTE.2021.3101470>.
- [23] Xu B, Hu X, Tang X, Lin X, Li H, Rathod D, et al. Ensemble Reinforcement Learning-Based Supervisory Control of Hybrid Electric Vehicle for Fuel Economy Improvement. *IEEE Trans Transp Electrif* 2020;6:717–27. <https://doi.org/10.1109/TTE.2020.2991079>.
- [24] Sun H, Fu Z, Tao F, Zhu L, Si P. Data-driven reinforcement-learning-based hierarchical energy management strategy for fuel cell/battery/ultracapacitor hybrid electric vehicles. *J Power Sources* 2020;455:227964. <https://doi.org/10.1016/j.jpowsour.2020.227964>.
- [25] Lian R, Tan H, Peng J, Li Q, Wu Y. Cross-Type Transfer for Deep Reinforcement Learning Based Hybrid Electric Vehicle Energy Management. *IEEE Trans Veh Technol* 2020;69:8367–80. <https://doi.org/10.1109/TVT.2020.2999263>.
- [26] Abdullah HM, Gastli A, Ben-Brahim L. Reinforcement Learning Based EV Charging Management Systems-A Review. *IEEE Access* 2021;9:41506–31. <https://doi.org/10.1109/ACCESS.2021.3064354>.
- [27] Liu W, Xiong L, Xia X, Lu Y, Gao L, Song S. Vision-aided intelligent vehicle sideslip angle estimation based on a dynamic model. *IET Intell Transp Syst* 2020;14:1183–9. <https://doi.org/10.1049/iet-its.2019.0826>.
- [28] Zhou Q, Li J, Shuai B, Williams H, He Y, Li Z, et al. Multi-step reinforcement learning for model-free predictive energy management of an electrified off-highway vehicle. *Appl Energy* 2019;255. <https://doi.org/10.1016/j.apenergy.2019.113755>.
- [29] Antonio G, Maria-dolores C, Member S. Multi-Agent Deep Reinforcement Learning to Manage Connected Autonomous Vehicles at Tomorrow ' s Intersections. *IEEE Trans Veh Technol* 2022;PP:1. <https://doi.org/10.1109/TVT.2022.3169907>.
- [30] Schmidt LM, Brosig J, Plinge A, Eskofier BM, Mutschler C. An Introduction to Multi-Agent

Reinforcement Learning and Review of its Application to Autonomous Mobility 2022.