Applied Energy 267 (2020) 114900

Contents lists available at ScienceDirect

Applied Energy

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

Heuristic action execution for energy efficient charge-sustaining control of connected hybrid vehicles with model-free double Q-learning

Bin Shuai[a], Quan Zhou[a,b,*], Ji Li[a], Yinglong He[a], Ziyang Li[a], Huw Williams[a], Hongming Xu[a,*], Shijin Shuai[b]

[a] Department of Mechanical Engineering, University of Birmingham, Birmingham B15 2TT, UK
[b] State Key Laboratory of Automotive Safety and Energy, Tsinghua University, Beijing 100084, China

HIGHLIGHTS

- Model-free double Q-learning is researched to save energy in a hybrid vehicle.
- Two heuristic action execution policies are proposed for improving energy efficiency.
- The random execution policy is the most effective and stable for double Q-learning.
- The methods can save energy at least 4% in selected real-world driving conditions.

ARTICLE INFO

Keywords:
Energy efficiency optimisation
Charge-sustaining control
Hybrid vehicle
Reinforcement learning
Double Q-learning

ABSTRACT

This paper investigates a model-free supervisory control methodology with double Q-learning for the hybrid vehicle in charge-sustaining scenarios. It aims to improve the vehicle's energy efficiency continuously while maintaining the battery's state-of-charge in real-world driving. Two new heuristic action execution policies, the max-value-based policy and the random policy, are proposed for the double Q-learning method to reduce overestimation of the merit-function values for each action in power-split control of the vehicle. Experimental studies based on software-in-the-loop (offline learning) and hardware-in-the-loop (online learning) platforms are carried out to explore the potential of energy-saving in four driving cycles defined with real-world vehicle operations. The results from 35 rounds of offline undisturbed learning show that the heuristic action execution policies can improve the learning performance of conventional double Q-learning by achieving at least 1.09% higher energy efficiency. The proposed methods achieve similar results obtained by dynamic programming, but they have the capability of real-time online application. Double Q-learnings are shown more robust to turbulence during the disturbed learning: they realise at least three times improvement in energy efficiency compared to the standard Q-learning. Random execution policy achieves 1.18% higher energy efficiency than the max-value-based policy for the same driving condition. Significant tests show that deciding factor in the random execution policy has little impact on learning performance. By implementing the control strategies for online learning, the proposed model-free control method can save energy by more than 4.55% in the predefined real-world driving conditions compared to the method using standard Q-learning.

1. Introduction

Heavy-duty vehicles contributed 27% of road transport carbon dioxide (CO2) emissions and 5% of EU greenhouse gas (GHG) emissions in 2016 [1]. The EU Commission has proposed that the average CO2 emission from new heavy-duty vehicles in 2025 should be 15% lower than in 2019, and a further 30% reduction should be achieved in 2030 [2]. These targets are expected to be achieved via electrification of heavy-duty vehicles, i.e. powertrain electrification [3,4], braking system electrification [5,6], and suspension electrification [7,8]. The feasible alternative to conventional oil-based road vehicles should be represented by electric and hybrid vehicles [9,10]. This will minimize the vehicle's energy consumption and emissions with the help of advanced energy management strategies in real-world driving [11,12]. The optimisation problems in energy management should be subjected to physical constraints, such as battery state-of-charge [13], power

[*] Corresponding authors at: Department of Mechanical Engineering, University of Birmingham, Birmingham B15 2TT, UK (Q. Zhou).
E-mail addresses: q.zhou@bham.ac.uk (Q. Zhou), h.m.xu@bham.ac.uk (H. Xu).

https://doi.org/10.1016/j.apenergy.2020.114900
Received 21 November 2019; Received in revised form 20 March 2020; Accepted 25 March 2020

Nomenclature

P	power (W)
s	state
a	action
r	reward
t	time
SoC	state of charge
R	resistance (Ω)
I	current (A)
$Loss$	power loss (W)
$\mathscr{D}$	deciding variable
Q^A	the first Q table
Q^B	the second Q table
Q	expected system performance
Π	action execution policy
$\boldsymbol{U}$	set of actions
θ	updating variable
π_{exe}	action execution policy
OEM	original equipment manufacturer

Subscripts

dem	power demand
eng	engine-generator set
$batt$	battery package
ini	initial
ref	reference

demand [14], and gear shifting [15].

Rule-based and model-based predictive control methods are traditionally used for energy management control of hybrid vehicles. The rule-based energy management strategy has been successfully implemented in real vehicle products [16], where control rules are predefined and optimised offline based on standard driving cycles [17]. Dynamic programming is considered as the global optimisation method; however, it requires large computational effort and is impossible for online applications [18]. Particle swarm optimisation (PSO) [19,20], nondominated sorting genetic algorithms (NSGA-III) [21], and convex optimisation [22] have been developed to achieve acceptable optimisation results in a much faster manner. Offline optimisation cannot guarantee the optimum vehicle performance in real-world driving since the standard driving cycles cannot fully include all scenarios in real-world driving.

Online optimisation is necessary for the real-time hybrid powertrain control with limited information on the future trip. Model-based predictive control (MPC) is a widely used method for online optimisation of hybrid vehicles [23]. MPC operates a rolling optimisation process in the vehicle controller, which is based on the prediction of the vehicle's future power demands over an optimisation horizon with mathematic model [24]. However, the performance of MPC is heavily dependent on the prediction of the driving conditions and vehicle states [25]. Recently, the unveiled legislation evaluates the vehicle emissions in real-world driving [26], therefore, the development of learning-based adaptive control is necessary where both rule-based and model-based energy management methods have their limitations.

Reinforcement learning is an emerging and promising technology for online optimal control [27]. It has been implemented in varies of vehicle control applications, e.g. active safety control [28], car following control [29]. The scientific novelty of Q-learning is the implementation of the knowledge base to allow online optimisation in an unknown environment based on Bellman's theory [30]. Remarkable improvement in vehicle energy efficiency has been achieved by reinforcement learning, compared to the conventional rule-based and model-based methods. The reinforcement learning methods, including Q-learning and deep Q-learning, can improve the vehicle's energy efficiency by at least 5% compared to the conventional MPC-based optimal control method [31].

Q-learning is commonly used for energy management of series or plug-in hybrid vehicles because it normally requires no more than three state variables and Q-table is capable of mapping the merit function values with the state variable inputs. Liu et al. implement Q-learning algorithms for energy management of series hybrid vehicles [32]. Zhou et al. proposed multi-step reinforcement learning for energy management of a hybrid vehicle [33]. Cao et al. optimize the energy use of a plug-in hybrid vehicle based on Q-learning [34]. Reddy et al. develop the energy management method for fuel cell/battery hybrid electric vehicle with Q-learning [35].

If the number of state variables is too large, it is not computationally efficient to use Q-table anymore. A deep neural network is therefore needed for deep Q-learning. Roman et al. demonstrate a model-based hyperparameter optimization of the hybrid vehicle using deep Q-learning [36]. Li et al. implement deep reinforcement learning for the energy management of a series hybrid bus considering its trip history information [37]. Q-learning and neuro-dynamic-programming are proposed to enable optimal convergence of vehicle energy efficiency without the model of vehicle plant [38]. Pengyue et al. use a neuro network to build an actor-critic for energy management of a hybrid electric vehicle [39].

Furthermore, the performance of reinforcement learning, in terms of learning stability and speed, can be improved by preventing the overestimation of the merit functions [40]. Breakthrough methods, which introduce an additional knowledge base (can be either a Q-table [41] or deep Q-network [42]), are proposed to minimize the positive turbulence caused by overestimation. The first attempt of using conventional double Q-learning for energy management of a hybrid car achieves 7.1% fuel saving compared to standard Q-learning [43]. Apart from the practical study using the double Q-learning algorithms with default settings, there are still some theoretical issues, e.g. how action execution policies affect the learning performance, need to be solved for specified engineering applications including energy management of the hybrid vehicle. To the best of the authors' knowledge, the research into action execution policies for energy management of hybrid vehicle with double Q-learning has not been found in any present publications.

To explore practically robust reinforcement learning methods for energy management of the plug-in hybrid vehicle, this paper carries out theoretical and experimental studies on the action execution policies for the energy management system using double Q-learning. Double Q-learning is chosen because it enables a straightforward learning process and is capable of energy management of the plug-in hybrid vehicle that only needs two-state variables. The new energy management method focuses on charge-sustaining control because more safety issues should be considered in this scenario. Robust model-free charge-sustaining control strategy should prevent the over-discharge of the battery since the battery is working in a low State-of-Charge (SoC) domain.

The present work includes the following new features: (1) two heuristic action execution policies are proposed and modelled for the charge-sustaining control of the hybrid vehicle with double Q-learning; (2) The concept of disturbed and undisturbed learnings are used for the first time to study the potential of energy efficiency improvement with the double Q-learning methods, and (3) statistic methods, including significant test and robustness test, are used for evaluations of the proposed method in the real-world applications based on the results of software-in-the-loop and hardware-in-the-loop testing.

The rest of this paper is organized as follows: Section 2 introduces the connected vehicle system. The framework and function modules of the model-free charge-sustaining control are presented in Section 3.

Two heuristic action execution policies for the model-free charge-sustaining control are proposed in Section 4, followed by the description of the experimental system for validation and evaluation of the control functionalities in Section 5. Section 6 demonstrates and analyses how the proposed methods outperform the conventional reinforcement learnings in energy saving from both experimental and theoretical aspects. The real-time performance of the proposed model-free charge-sustaining control system is also presented for energy flow analysis and robustness test in Section 6. Section 7 summarizes the conclusions.

2. The connected vehicle system

This paper is partially supported by an Innovate UK project. An aircraft-towing tractor manufacturer is involved as an industrial partner. They support the vehicle parameters and real-world vehicle operation data of a connected aircraft-towing tractor so that the proposed model-free charge-sustaining energy management control is demonstrated with the connected vehicle system as shown in Fig. 1. The proposed technology is not limited to the application of the aircraft-towing tractor. It can be implemented in any hybrid vehicles. A vehicle-to-everything (V2X) network connecting the tractor, the Roadside Unit (RSU), and the Control Tower (CT), is used for data exchange within the system. The tractor-to-aircraft communication and the CT-to-RSU communication are both enabled by the Ethernet, and the tractor communicates with the RSU via Wi-Fi. Vehicle information (e.g. state, action, and reward of the vehicle system) will be transferred into the RSU then uploaded to the control tower for online reinforcement learning. Afterwards, the optimised control policy will be implemented in the local energy management control by downloading the key parameters of the control policy model (e.g. Q-value) regularly, which allows online optimisation of the energy management control strategy in real-world driving.

The aircraft-towing tractor has a plug-in series hybrid powertrain as shown in Fig. 2, which uses electricity from a battery package (consisted of 8200 NCR-18650 series lithium-ion cells) as the primary power to drive a 245 kW traction motor. An 86.2 kW engine-generator is equipped as an alternative power unit to provide extra power for vehicle operation and battery charging. The energy management system (EMS) determines the amount of power contributed by the engine-generator and the battery package to satisfy the power demand and maintain the State-of-Charge (SoC) of the battery pack. This paper will focus on charge-sustaining control of the energy management system.

3. The model-free charge-sustaining control with double Q-learning

The model-free charge-sustaining control system is developed with a layered energy management system [33] shown in Fig. 3, which includes two layers: the control layer (installed in the onboard tractor controller) and the learning layer (deployed in the server computer). The two layers communicate through the V2X network. The proposed control strategy aims to continuously optimize the vehicle's energy efficiency in real-world operation while maintaining the battery SoC close to 30% for longer battery life and enhanced battery safety. There are four function modules to allow the model-free charge-sustaining control. Three of the function modules, including the states perception module, the action execution module, and the reward assessment module, are located in the control layer. The double Q-learning module is located in the learning layer. The inputs and outputs of each layer and algorithms used to enable the functionalities are described as follows:

3.1. States perception module

The states perception module is responsible for determining the current state of the vehicle system based on the sensor signal. The battery SoC and the driver's power demand are selected as the state variables for the best performance of the learning system with the minimum computational effort [33]. The precepted state variables will be sent to the double Q-learning module for parallel learning and the action execution module for control of the downstream controllers. The variables are measured at each sampling time and discretized into the finite state vector,

$$s(t) = [P_{dem}(t),\ SoC(t)] \tag{1}$$

where, $s(t)$ is the current state at the t^{th} time step; $P_{dem}(t) \in \{0\text{kW} \le P_{dem} \le 253\text{kW}\}$ is the driver's power demand value at thet^{th} time step; $SoC(t) \in \{20\% \le SoC \le 80\%\}$ is the battery SoC value at the t^{th} time step.

3.2. Action execution module

The action execution module connects with the state perception module, the double Q-learning module and the downstream controllers, including the engine-generator controller and battery management controller. The primary purpose of the action execution module follows the action execution policy to pick an action from the two Q-tables and implement the charge-sustaining control signal to downstream controllers (e.g. engine-generator controller). The action taken in each sampling time will be measured and forward to the double Q-learning module. The execution policy π_{exe} is used to determine the action $a(t) \in \boldsymbol{A}$ based on the current vehicle state $s(t)$, which is defined below:

$$a(t) \leftarrow \pi_{exe}(\mathbb{Q}(s(t), \boldsymbol{A})) \tag{2}$$

where, $\mathbb{Q}$ is the knowledge base that stores the merit-function values corresponding to different states and action variables; $\mathbb{Q}$ can be either Q^A or Q^B, depending on the action execution policy. The policy π_{exe} is the action execution policy, which will be continuously optimised via reinforcement learning; $a(t)$ is the action value at the time t^{th} time step, and it will be used to control the power rate of the engine-generator:

$$u_{apu}(t) = a(t) \tag{3}$$

where, $u_{apu}(t) \in \{0\% \le u_{apu} \le 100\%\}$ is the control signal to the engine-generator. The requirement for the battery pack can be obtained using:

$$u_{batt}(t) = \frac{P_{req} - u_{apu}(t) \bullet P_{apu\_max}}{P_{batt\_max}} \tag{4}$$

where, P_{batt} is the power supplied by the battery package; $P_{apu\_max} = 86.2$ kW is the maximum power that can be supplied by the engine generator; $P_{batt\_max} = 365$ kW is the maximum power that can be supplied by the battery pack; P_{req} is the required power for driving the traction motor and powering the on-boarded auxiliary devices.

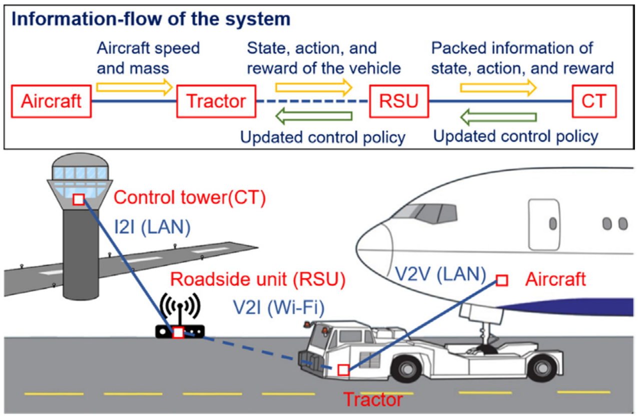

Fig. 1. The connected vehicle system [33].

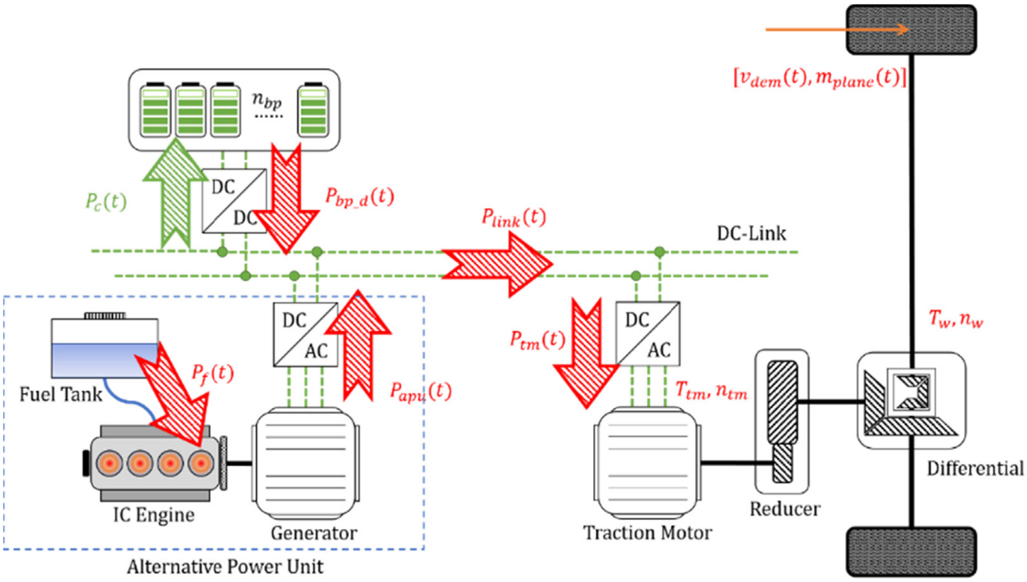

Fig. 2. Configuration of the plug-in series hybrid powertrain for the aircraft-towing tractor [44].

3.3. Reward assessment module

The reward assessment module measures the powertrain performance, including energy consumption and remaining battery SoC. It uses a merit-function to evaluate vehicle performance after taking each action. The merit function calculates at each sampling time will be sent to the double Q-learning module. This helps the training of the optimal control policy which minimizes the vehicle power loss P_{loss} while maintaining battery SoC simultaneously. The merit-function is defined as [33]:

$$r(t) = \begin{cases} r_{ini} - P_{loss}(t) SoC(t) \geq 30\% \\ r_{ini} - P_{loss}(t) - \alpha \; |SoC_{ref} - SoC(t)| \; SoC(t) < 30\% \end{cases} \tag{5}$$

where, SoC_{ref} is the reference battery SoC value that is chosen to maintain the battery SoC within an acceptable range (for the best performance and health of the battery SoC_{ref} should be 28%). α is a scale factor to balance the consideration of battery SoC level and power efficiency; $P_{loss}(t) = Loss_{eng}(t) + Loss_{batt}(t)$ is the total power loss of engine and battery; the power loss of engine $Loss_{eng}(t)$ and power loss of battery $Loss_{batt}(t)$ can be calculated by:

$$\begin{cases} Loss_{eng}(t) = \dot{m_f}(t) \bullet H_f - \frac{T_{eng}(t) \bullet n_{eng}(t)}{9550} \\ Loss_{batt}(t) = R_{loss}(SoC) \bullet I_{batt}(t)^2 \end{cases} \tag{6}$$

where, $\dot{m_f}$ is the real-time measurement of fuel rate in kg/s; T_{eng} and n_{eng} are the engine torque and speed; I_{batt} is the current of the battery pack; H_f is the heat value of fuel (for diesel, $H_f = 44 \times 10^6$J/kg), R_{loss} is equivalent internal resistant of the battery, and R_{loss} is a function of battery SoC.

3.4. Double Q-learning module

The double Q-learning module receives the state, action, and reward variables from the other three function modules. It implements double Q-learning algorithm to optimize the action execution policy by updating the merit-function values in the knowledge bases. Two knowledge bases (Q^A and Q^B) are used to predict the merit function value with the observation of the state variables. Theoretically, this

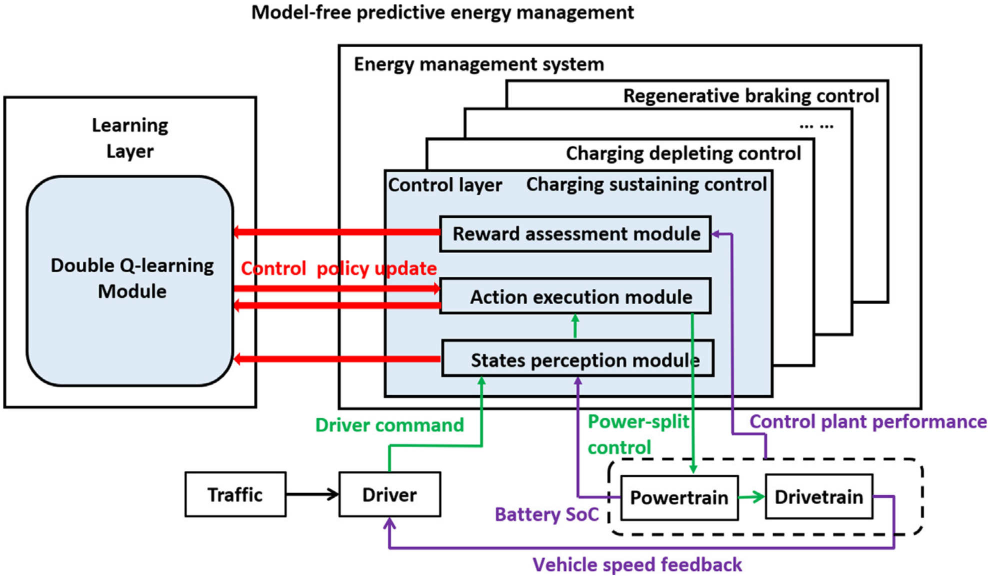

Fig. 3. Layered control framework for model-free charge-sustaining control.

mechanism can reduce the overestimation between the actual value and the approximation [45].

The learning process of double Q-learning is mathematically modelled as the updating of the two Q-tables using

$$\begin{cases} Q^A(s(t), a(t)) \leftarrow Q^A(s(t), a(t)) \\ \qquad + \beta(r(t) + Q^B(s(t+1), a^{'}) \\ \qquad - Q^A(s(t), a(t))) \\ Q^B(s(t), a(t)) \leftarrow Q^B(s(t), a(t)) \\ \qquad + \beta(r(t) + Q^A(s(t+1), b^{'}) \\ \qquad - Q^B(s(t), a(t))) \end{cases} \tag{7}$$

where, $Q^x(s(t), a(t))$ is the element in the knowledge bases Q^x ($x = A$ or B); the element value is indexed by the state variables $s(t)$ and action variable $a(t)$; $a^{'}$ and $b^{'}$ are the action for the next step predicted with the knowledge bases Q^A and Q^B with the maximum merit-function values. β is the learning rate, the default setting $\beta = 0.5$ is chosen for this paper [46]. Conventional double Q-learning applies a rolling alternate progressing policy [43], i.e. Q^A and Q^B will change the role as the leading knowledge base in each time cycle, when Q^A is used as the leading knowledge base for action execution, the feedback merit-function value will be used to update the Q^B. Similar policy is applied if Q^B is used for as the leading knowledge base for action execution.

4. The heuristic action execution policies

Action execution is the most important procedure for model-free charge sustaining control as described above. It determines the control signals to the downstream controllers so that the powertrain feeds backthe system with reward. Both action and reward variables significantly affect the learning process in the double Q-learning module. Deterministic exploration (based on the maximum merit function value) and the random exploration are the two heuristic elements working collaboratively to allow continuous evolution in reinforcement learning [46]. They are conventionally used for the learning module for standard Q-learning and double Q-learning [33,43]. Two new action execution policies are proposed with the consideration of these two heuristic elements as follows:

4.1. Max-value-based execution policy

The max-value-based execution policy is proposed based on the deterministic exploration for system evolution. The action is picked up with the maximum merit-function value in both Q-tables:

$$\pi_{exe}\text{: } a(t) \leftarrow \underset{a(t)\in \boldsymbol{U}}{arg\,max}\,[Q^A(s(t), :)\ \ Q^B(s(t), :)] \tag{8}$$

where, $Q^A(s(t), :)$ and $Q^B(s(t), :)$ are two arrays of the merit function values; they are indexed by the state variables $s(t)$ from the two knowledge bases Q^A and Q^B respectively.

After the action $a(t)$ is executed, the following rules will be used for updating of the knowledge bases with the reward feedback $r(t)$ from the vehicle:

$$\begin{cases} Q^A(s(t), a(t)) \leftarrow Q^A(s(t), a(t)) \\ \quad + \beta(r(t) + Q^B(s(t+1), a^{'}) - Q^A(s(t), a(t))) if \theta \geq 0.5 \\ \quad Q^B(s(t), a(t)) \leftarrow Q^B(s(t), a(t)) \\ \qquad + \beta(r(t) + Q^A(s(\mathrm{t}+1), b^{'}) \\ \qquad - Q^B(s(t), a(t))) if \theta < 0.5 \end{cases} \tag{9}$$

where, θ is a random number between 0 and 1· Because the action execution policy is deterministic based on the maximum value in $Q^A(s(t), :)$ and $Q^B(s(t), :)$. The random number θ is therefore needed to include a stochastic process to allow the generating of new features for the system evolution.

4.2. Random execution policy

The random execution policy is proposed based on random exploration. A deciding variable $\mathscr{D}$ is introduced to compare with a random comparing variable $\mathscr{C}$. The action is executed using the comparison result as:

$$\pi_{exe}\text{: } a(t) \leftarrow \begin{cases} \underset{a(t)\in \boldsymbol{U}}{arg\,max}\ Q^A(s(t), :) & if \quad C \geq D \\ \underset{a(t)\in \boldsymbol{U}}{arg\,max}\ Q^B(s(t), :) & if \quad C < D \end{cases} \tag{10}$$

where $\mathscr{C}$ is the random comparing variable that is between 0 and 1; $\mathscr{D}$ is the deciding variable which determines the Q-table that will be used for selecting actions. If $\mathscr{C} \geq \mathscr{D}$, the action execution module will select an action $a(t)$ from Q^A. Otherwise, an action $a(t)$ will be collected from Q^B.

After the action $a(t)$ is executed, the following rules will be used for updating the Q-tables with the reward feedback $r(t)$ from the vehicle:

$$\begin{cases} \quad Q^A(s(t), a(t)) \leftarrow Q^A(s(t), a(t)) \\ \qquad + \beta(r(t) + Q^B(s(t+1), a^{'}) - Q^A(s(t), a(t))), if \\ \qquad : a(t) \leftarrow \pi_{exe}(Q^A) \\ Q^B(s(t), a(t)) \leftarrow Q^B(s(t), a(t)) \\ \qquad + \beta(r(t) + Q^A(s(t+1), b^{'}) - Q^B(s(t), a(t))), if \\ \qquad : a(t) \leftarrow \pi_{exe}(Q^B) \end{cases} \tag{11}$$

where, $a(t) \leftarrow \pi_{exe}(Q^x)$ means the action $a(t)$ is executed based on the knowledge bases $Q^x (x = A\ or\ B)$. Since the random policy involves uncertainties in action execution as in Eq. (10), the updated policy for both knowledge bases should be deterministic, that is if action$a(t)$is randomly executed from a knowledge base (e.g. Q^A), the vehicle system feedback will update the merit-function value in another knowledge base (e.g. Q^B).

5. Testing and validation set up

Testing and validation of the proposed control strategies follow a simplified model-based development (MBD) procedure in the automotive industry. Both software-in-the-loop and hardware-in-the-loop platforms will be developed for the evaluations in different MBD stages. Initially, the performance of two action execution policy methods is first investigated by tracking the evolution of the vehicle's energy efficiency with a driving cycle in the software-in-the-loop platform. The software-in-the-loop platform is built with MATLAB/Simulink on a workstation (configured with i7-8700 CPU and 32 GB RAM). The improvements in vehicle energy efficiency in a repeated driving cycle for machine learning (described in Section 5.1) are compared with the standard Q-learning, double Q-learning, and dynamic programming. Both disturbed and undisturbed learning scenarios are considered for the software-in-the-loop evaluation. Next, the feasibility for real-time implementation was investigated by monitoring the performance of each learning algorithm in the hardware-in-the-loop platform (described in Section 5.2). The proposed model-free charge-sustaining control methods are validated in two rounds of real-time hardware-in-the-loop tests (described in Section 5.1) to observe the robustness of the proposed control methods. The results are compared with the standard Q-learning method was carried out for energy flow analysis and robustness test.

5.1. Driving cycles for machine learning and validation

Four driving cycles defined by the tractor manufacturer [23,44] are used in the present work. All the driving cycles are defined based on real vehicle operation data collected from London Heathrow Airport. The power demand profiles of each driving cycle are shown in Fig. 4. The vehicle model following the driving cycle 1 (3000 s) for machine

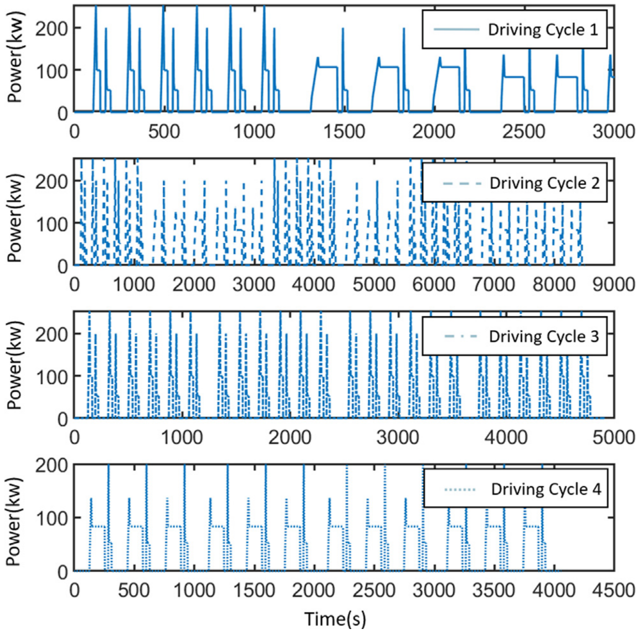

Fig. 4. The power demand of the four driving cycles.

learning was first run repetitively for 35 rounds. The other three driving cycles were used to simulate the power demand diversity in real-world operation, enabling the evaluation of the model-free predictive energy management strategy.

5.2. Hardware-in-the-loop testing facilities

The hardware-in-the-loop platform is built with the ETAS's vehicle control development facility [37] as shown in Fig. 5. The control strategies were developed using MATLAB/Simulink initially, and then be compiled into C-code using ETAS' INTECRIO/INCA software in Host PC-1. The complied control strategies are deployed into an ETAS ES910 prototype controller through a USB-to-CAN interface. The ES910 is configured with a 1.5 GHz CPU, a 4 GB RAM, and a 1Gbps Ethernet interface. DESK-LABCAR implements the real-time model of the hybrid aircraft towing tractor on host PC-2 via Ethernet. The real-time vehicle model was developed in Simulink and was verified using the testing data from a prototype vehicle [38]. LABCAR and the ES910 were connected via a CAN bus to emulate the communication between the controller and the vehicle plant in real-time. The performance of the vehicle was monitored and recorded by ETAS' Experimental Environment.

6. Results and discussion

6.1. Improvement in energy efficiency with double Q-learning

Model-free charge-sustaining control performance with double Q-learning is initially investigated by monitoring the vehicle's energy efficiency at different learning stages on the software-in-the-loop platform. This will demonstrate how the double Q-learning with the

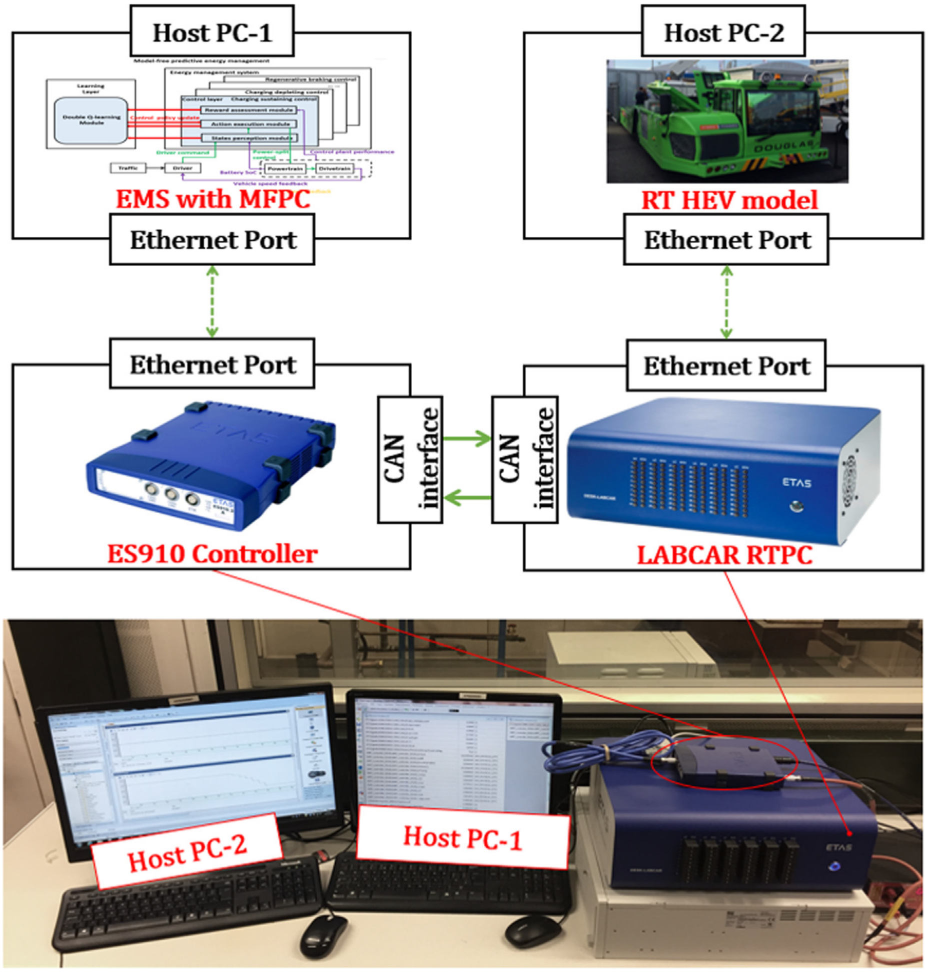

Fig. 5. Facilities for hardware-in-the-loop test.

proposed action execution policies outperform the conventional double Q-learning method. The results obtained with conventional double Q-learning methods and the double Q-learnings with the heuristic action execution policies are compared in Fig. 6. The black round line shows the improvement in vehicle energy efficiency with conventional double Q-learning. The blue cross line monitors the vehicle performance using the random action execution policy, and the red plus-sign line shows the vehicle performance using the max-value-based action execution policy. The learning processes can be classified into two stages based on the improving rate of vehicle energy efficiency. In stage one, the energy efficiency improvement of the vehicle fluctuate dramatically. This is due to the range of attempts of the differing actions for reinforcement learning. Stage two slows down the improvement of the vehicle's energy efficiency after a 'knee point'. This is due to the progressive reduction in the probability of new action exploration over time [31], which leads to a theoretical logarithmic improvement in the vehicle's energy efficiency.

All the double Q-learning algorithms can achieve an improvement in the vehicle's energy efficiency, starting with an average initial vehicle energy efficiency of 31.29% at around one by learning from scratch (all the knowledge bases are zero sets). 2.56%, 3.23%, and 4.80% improvements in vehicle energy efficiency are realized by conventional double Q-learning, double Q-learning with the max-value-based execution method, and double Q-learning with the random execution method, respectively. The two proposed double Q-learning control methods can obtain better vehicle performance than the standard Q-learning at the end of the learning process. The double Q-learning with the proposed action execution policies outperform the conventional double Q-learning method by additionally improving by at least 1.09% the vehicle energy efficiency. The model-free charge-sustaining control with the random execution policy can further increase the vehicle energy efficiency by at least 3% compared to the max-valued-based method.

6.2. Efficiency improvements in disturbed and undisturbed learning scenarios

In real-world driving, the learned experience stored in the knowledge base may provide negative effects on the learning progress. This motivates the study of the model-free charge-sustaining control in the undisturbed and disturbed learning scenarios. Three groups have been carried out in offline learning, including double Q-learning with max-value-based action execution policy (Group A), double Q-learning with random action execution (Group B), and double Q-learning with random action execution (Group C), to observe the adaptivity under turbulence (defects from the algorithm). Dynamic programming (DP) is used to obtain the global optimisation results offline to justify the maximum efficiency that can be achieved ideally. Based on the authors' previous development [18], the DP algorithm uses the battery SoC as a state variable, the cost for the transient from one state to another is calculated by Eq. (5). In each group, the reinforcement learning algorithm operates two individual optimisations for model-free charge-sustaining control at the same driving condition (repeating driving cycle 1 for 35 rounds) with initial battery SoC of 30%. The first optimisation is learning from scratch (undisturbed learning), and the second optimisation is learning from a pre-defined knowledge based on a poor control policy (disturbed learning). Disturbed learning includes some turbulences into the learning process by simulating bad attempts. Fig. 7(a), (b) and (c) show the vehicle's energy efficiency (calculated at the end of each round) based on standard Q-learning, double Q-learning method with the max-value-based action execution policy, and double Q-learning method with the random action execution policy, respectively. Each subplot compares the vehicle's energy efficiency obtained in undisturbed learning (red cross line) to that obtained in disturbed learning (blue round line).

The global optimal energy efficiency calculated by the DP for the driving cycle is 32.81% (shown in pink solid line) with the assumption that the power demand of the driving cycle is known in advance. This can only provide a reference to show the effectiveness of the online learning-based control methods (achieve more than 31.53%). But it is impossible to use DP in real-time practice because it requires 100% actuate future power demand profiles in real-world driving applications. The model-free energy management method is not limited by the driving conditions and less computation (after the training process). It can be practically applied online for achieving optimal energy efficiency through continuously interacting with the characteristic of unknown driving cycles.

Generally, undisturbed learning achieves better vehicle energy efficiency than disturbed learning, because a poor control policy is applied in the pre-defined knowledge base. For the standard Q-learning (Group C), undisturbed learning experiences some oscillations during the learning process and finally achieves an energy efficiency of 31.82%. The disturbed Q-learning has a gradual improvement in energy efficiency and finally converges to the energy efficiency of 31.53%.

Both double Q-learning methods (Group A and B) can achieve higher vehicle energy efficiency than standard Q-learning with either undisturbed learning or disturbed learning. More specifically, for the double Q-learning method with the max-value-based execution policy (Group A), the vehicle energy efficiency can achieve 32.21% in disturbed learning and a very similar energy efficiency level of 32.30% in undisturbed learning. The random action execution policy (Group B) has more potential to underestimate or reduce the value of each action in both Q-tables. It obtained the highest vehicle energy efficiency with 32.59% and 32.75% in disturbed learning and undisturbed learning, respectively.

Considering the improvements (from the initial energy efficiency to end energy efficiency) in disturbed learning, the standard Q-learning gives an improvement of 1.05%, much less than that in undisturbed learning (3.09% improvement). This is because the pre-defined knowledge base in disturbed learning leads to the overestimation of the actions, which limits the probability of attempting other actions for global searching. The double Q-learning with the max-value-based execution policy achieves an improvement of 3.22%, while 4.80% energy efficiency improvement is achieved by the random execution method. Both of the double Q-learning methods can achieve at least three times higher energy efficiency improvement than the conventional Q-learning during the disturbed learning process.

6.3. Stability of learning in vehicle's energy efficiency improvement

Double Q-learning involves many stochastic processes that may affect learning stability. This is more significant for the method using random execution policy, in which, an additional random number $\mathscr{C}$ is

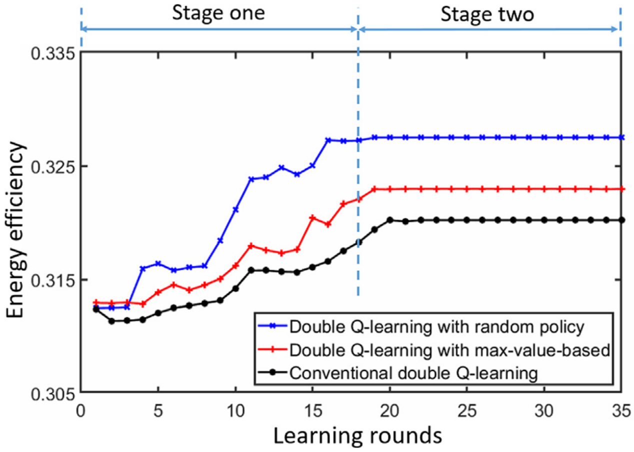

Fig. 6. Learning performance of three learning-based control strategies.

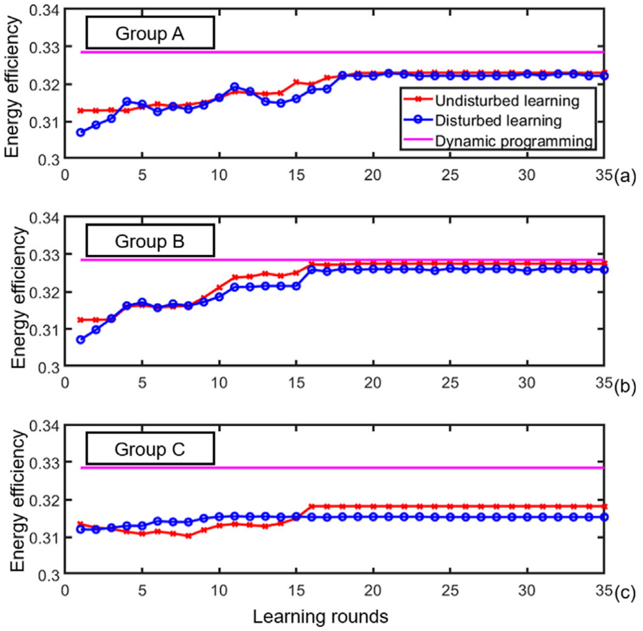

Fig. 7. Vehicle's energy efficiency obtained in undisturbed learning and disturbed learning.

initially compared to the deciding factor $\mathscr{D}$, if $\mathscr{C}$ is larger than $\mathscr{D}$ then an action from Q-table A is executed. The default value of the deciding factor is 0.5, which assumes an equal probability for attempting an action from one of the two tables. The influence of the deciding factor on the learning level was investigated by changing the value from 0.1 to 0.5. The tests were carried out based on repeating 35 rounds of driving cycle 1 with an initial battery SoC of 30%. Each test with a different deciding variable value was repeated 20 times. Table 1 compares the level of learning at stage two for different deciding factor values, in which, the vehicle energy efficiency at the end of the learning process and the variance of results in stage two are used to evaluate the stability of learning.

The data from Table 1 was analysed in the statistical software package, Minitab, to ascertain any correlation between the outcomes and the level of the deciding variable.

The results in Table 2 show that (at say 5% significance) the null hypothesis (that there is no correlation) cannot be rejected for either the energy efficiency or the variance, so it is concluded that there is no significant correlation with the level of the deciding variable.

The stability of learning for control policy optimisation based on standard Q-learning, double Q-learning method with the max-value-based execution policy, and double Q-learning method with the random execution policy is compared in Table 3. The variance of the results obtained by all the methods is very small, which shows that the learning process in stage two is very stable. Among these methods, the double Q-learning with the random execution policy has the lowest variance of results. This includes both the best and the worst average energy efficiency obtained by the random execution policy with different deciding factor values in Table.3. In terms of the average vehicle energy efficiency in stage two, all double Q-learning methods outperform Q-learning method by achieving higher energy efficiency. The double Q-learning method with the random execution policy can achieve at least 32.56% energy efficiency, which is marginally better than the double Q-learning method with the max-value-based policy (32.35%).

6.4. Energy flow in real-world operation

The energy flow in real-world operation with battery initial SoC values of 30% was investigated on the hardware-in-the-loop platform under real-world cycle-1 for two stages. Firstly, we compared all the double Q-learning methods to select the most effective double Q-learning method that achieves the minimum total energy loss at the end of the driving cycle. In the second stage, we compare the selected double Q-learning method with the standard Q-learning and the results obtained offline with dynamic programming.

The results obtained by double Q-learning (DQL) methods, including the DQL with conventional policy (DQL-CON), DQL with max-value-based action execution policy (DQL-MEP), and DQL with random action execution policy (DQL-REP), are compared in Fig. 8. The real-time data, including the accumulated energy loss, battery state-of-charge, equivalent energy loss in battery and engine generation, is collected in each subfigure. The vehicle with DQL-CON method (magenta dash line) achieves almost the same value of total energy loss like the one with DQL-MEP (blue solid line) method. DQL-REP method (red solid line) outperforms the other two methods by losing less energy in real-world driving. According to the real-time battery state-of-charge, DQL-REP allows smoother operation of battery charge and discharge, and maintain battery state-of-charge close to the predefined level, i.e. 28%. This results in significant energy saving from the battery for the vehicle with DQL-REP.

The vehicle performance with DQL-REP method is then compared with the vehicles using standard Q-learning (SQL) method, and the offline results obtained by DP. The vehicle performances are presented in Fig. 9. The DQL-REP (red solid line) outperforms the SQL (blue dash-dot line) method by achieving less total energy loss at the end of the driving cycle. The energy consumption curve obtained with the DQL-REP is closer to the one obtained with the DP (cyan dash line) method, especially for the energy loss in the battery. The battery SoC obtained with the DP remains at a relatively high level compared to the SQL and DQL-REP method, because DP plans each action from a global optimisation perspective with the consideration of the whole driving information. However, the DP results are practically inaccessible because it is impossible to obtain the whole trip information in advance in real-world operations.

The energy flows of the vehicle with different energy management methods are further compared in Fig. 10. The accumulated total energy losses achieved by SQL, DQL-CON, DQL-MEP, DQL-REP, and DP are 246.84 MJ, 243.33 MJ, 240.61 MJ, 231.45 MJ, and 209.19 MJ, respectively. DP achieves the minimum energy losses with the offline optimisation. DQL-REP is shown to be the best online learning method which has the lowest total energy loss compared to SQL and other DQL methods. DQL-REP saves 15.39 MJ energy compared to the SQL method. This is contributed by 4.49 MJ energy saving in battery loss (58.20% lower than SQL method) and 10.90 MJ energy saved in engine generator (4.56% lower than SQL method). In charge-sustaining control scenarios, the energy saving in the battery has more contribution to the total energy saving.

Table 1
Stability of learning at stage two with different deciding factor values.

Deciding variable value	Energy efficiency	Variance of results
0.10	32.65%	8.94e-8
0.15	32.63%	3.46e-8
0.20	32.73%	8.73e-8
0.25	32.56%	4.20e-7
0.30	32.57%	7.68e-8
0.35	32.75%	2.13e-8
0.40	32.74%	1.12e-8
0.45	32.65%	6.50e-8
0.50	32.74%	3.20e-8

Table 2
Correlation of the results and performance of the models with the deciding variable.

x	y	Correlation coefficient	P-value
Deciding variable	Energy efficiency	0.386	0.304
	Variance of results	−0.250	0.517

Table 3
Mean efficiency and variance of 20 samples.

Algorithm	Execution method	Average efficiency	Variance of results
Q Learning	–	31.47%	4.78e-7
Double Q-Learning	Max-value-based	32.35%	1.03e-6
	Random (best: $\mathscr{D}$ = 0.35)	32.75%	2.13e-8
	Random (worst: $\mathscr{D}$ = 0.25)	32.56%	4.20e-7

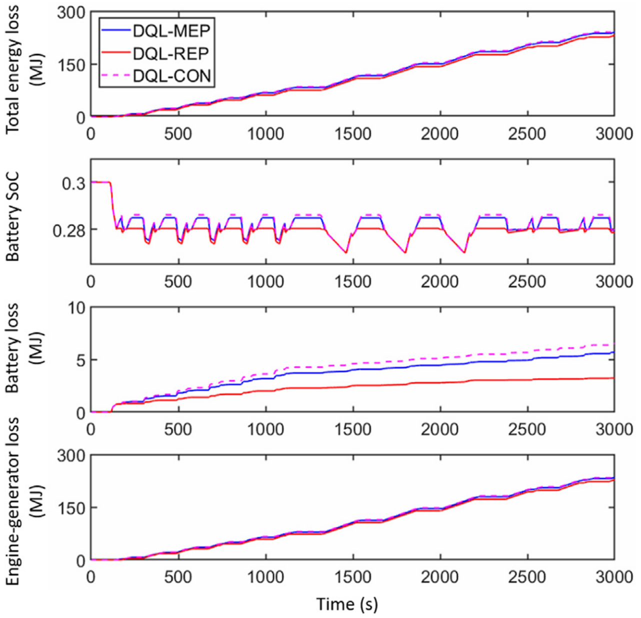

Fig. 8. HiL testing results of three double Q-learning methods.

6.5. Robustness of the performance in real-world operations

Robustness test is carried out based on the hardware-in-the-loop platform to validate the adaptability of the proposed charge-sustaining control method in real-world driving. Three driving cycles, comprising 9 scenarios with different initial battery SoC values (30%, 28%, and 25%), are used to emulate the unknown driving conditions in real-world driving. None of the driving conditions is used for offline learning of the control policy, online learnings capability will be evaluated in these unknown conditions. Both the battery SoC level at the end of each cycle (end SoC) and total energy usage are listed in Table 4.

The total energy usage is the sum of the energy loss and the effective energy used for vehicle operation (e.g. driving and aircraft-towing). The proposed double Q-learning methods with both max-value-based execution policy (DQL-MEP) and random execution policy (DQL-REP) are compared with standard Q-learning (SQL) and double Q-learning with conventional policy (DQL-CON). The standard Q-learning method is chosen as the baseline method, and the energy-saving rate (savings) is calculated by:

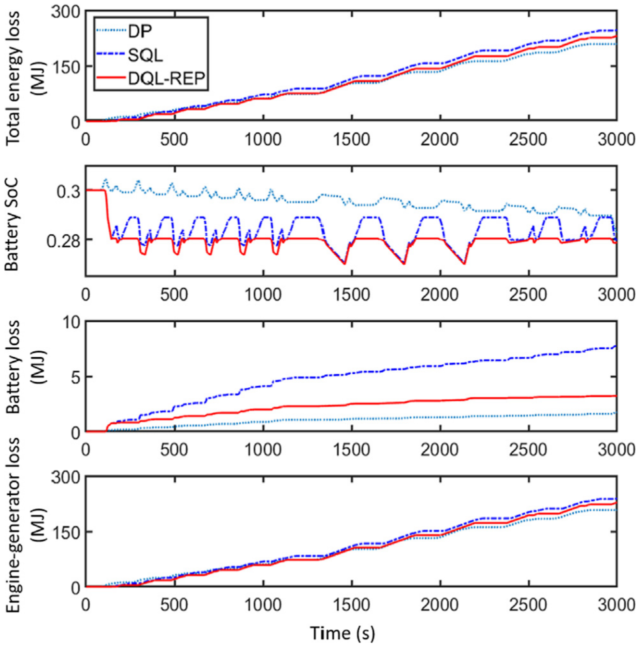

Fig. 9. HiL testing results of DQL-REP and SQL, and the offline results of DP.

$$\Delta = \frac{E_Q - E_{DQ}}{E_Q} \tag{12}$$

where, Δ is the energy-saving rate; E_Q is the total energy used for vehicle operation with the standard Q-learning method; E_{DQ} is the total energy used for vehicle operation with the two double Q-learning methods. According to the real-time variation of battery SoC, the saving is achieved by charging and smoothly discharging the battery following the power demand.

The double Q-learning with the random policy (DQL-REP) can save more than 4.55% of additional energy compared to the standard Q-learning (SQL). The highest energy saving of 7.78% is obtained on real-world cycle-3 with initial battery SoC set at 28%. The average energy saving is 5.78% for the nine pairs of experiments. The highest saving rates of the other two double Q learning methods are 2.07% (DQL-CON) and 3.46% (DQL-MEP). This should be noticed that the Q learning algorithm used here can save at least 7.8% energy compared to the model-based method [33].

7. Conclusions

The work in this paper studied a new energy-efficient charge-sustaining control strategy for a hybrid off-highway vehicle using double Q-learning. Two heuristic action execution policies were proposed for the improvement of the energy efficiency of hybrid vehicles by reducing overestimation of the merit-function values. Software-in-the-loop and hardware-in-the-loop tests were used to evaluate the optimisation performance in the charge-sustaining control scenarios, using a standard Q-learning algorithm as the baseline, for both offline and online investigations. Four driving cycles were defined in the study based on real-world vehicle operation data, in which driving cycle 1 was repeated 35 times in offline training (the data started to converge after 25 times).

The conclusions drawn from the investigation are as follows:

(1) The heuristic action execution policies can improve the learning performance of conventional double Q-learning. They achieve at least 1.09% higher energy efficiency after 35 rounds of offline undisturbed learning. The improving rate in energy efficiency, compared to the first round, is three times higher than the Q-learning

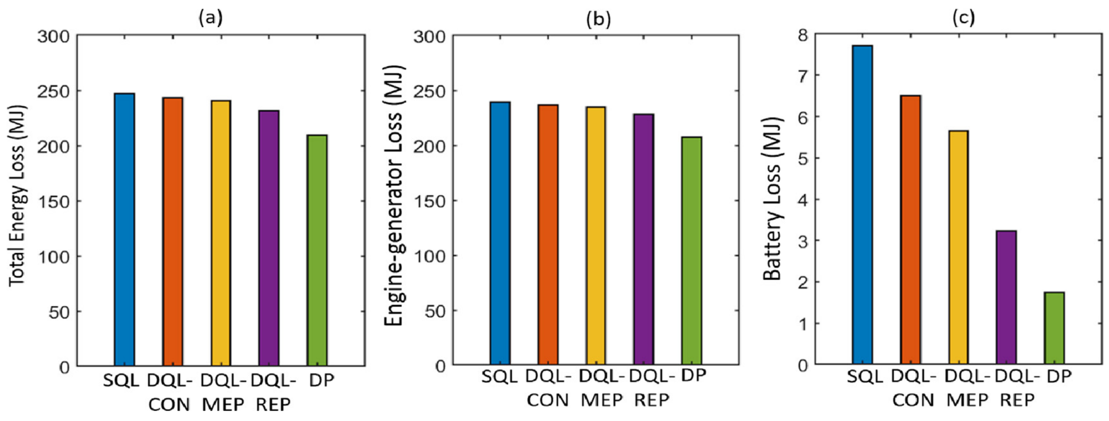

Fig. 10. Energy flows of the Vehicle with Different Control Methods.

Table 4
Performance of double Q-learning methods and standard Q-learning method.

Driving Cycle‘	Initial SoC	Method	End SoC	Energy Usage (MJ)	Savings
Driving Cycle 2	25%	SQL	28.80%	1082.51	–
	25%	DQL-CON	28.61%	1070.61	1.06%
	25%	DQL-MEP	28.49%	1062.32	1.85%
	25%	DQL-REP	28.04%	1032.46	4.55%
	28%	SQL	28.90%	1066.84	–
	28%	DQL-CON	28.61%	1054.13	1.21%
	28%	DQL-MEP	28.49%	1045.42	2.03%
	28%	DQL-REP	28.04%	1015.61	4.85%
	30%	SQL	28.80%	1056.13	–
	30%	DQL-CON	28.61%	1044.31	1.11%
	30%	DQL-MEP	28.49%	1035.92	1.93%
	30%	DQL-REP	28.04%	1006.44	4.76%
Driving Cycle 3	25%	SQL	28.90%	556.22	–
	25%	DQL-CON	28.61%	545.95	1.88%
	25%	DQL-MEP	28.49%	538.79	3.17%
	25%	DQL-REP	28.04%	515.97	7.24%
	28%	SQL	28.90%	539.75	–
	28%	DQL-CON	28.61%	528.93	2.07%
	28%	DQL-MEP	28.49%	521.57	3.46%
	28%	DQL-REP	28.04%	498.67	7.73%
	30%	SQL	28.90%	529.64	–
	30%	DQL-CON	28.61%	519.25	2.06%
	30%	DQL-MEP	28.48%	512.06	3.45%
	30%	DQL-REP	28.04%	489.54	7.78%
Driving Cycle 4	25%	SQL	28.90%	494.56	–
	25%	DQL-CON	28.61%	489.98	1.02%
	25%	DQL-MEP	28.49%	486.94	1.68%
	25%	DQL-REP	28.04%	471.86	4.82%
	28%	SQL	28.90%	477.74	–
	28%	DQL-CON	28.61%	472.88	1.13%
	28%	DQL-MEP	28.49%	469.67	1.86%
	28%	DQL-REP	28.04%	454.58	5.13%
	30%	SQL	28.90%	467.33	–
	30%	DQL-CON	28.61%	462.83	1.08%
	30%	DQL-MEP	28.49%	459.81	1.81%
	30%	DQL-REP	28.04%	445.22	5.12%

method.

(2) Double Q-learning with random execution policy is robust to the turbulence and achieve the energy efficiency of 32.59% in disturbed learning. The effectiveness of the result is confirmed by comparing to the efficiency of 32.81% achieved by the benchmark dynamic programming.

(3) The double Q-learning method with the random execution policy (DQL-MEP) has the lowest variance and highest average energy efficiency after 35 learning rounds in the defined driving cycle 1 for offline learning. The investigation has revealed that the deciding factor in the random execution policy has little impact on the performance of the vehicle.

(4) For online real-time control in driving cycles 1–4 with differing initial battery state-of-charge, the performance of double Q-learning with the random execution policy is robust, saving at least 4% of total energy usage over the baseline method.

CRediT authorship contribution statement

Bin Shuai: Methodology, Formal analysis, Software, Writing - original draft. **Quan Zhou:** Funding acqusition, Supervision, Conceptualisation, Writing - review & editing. **Ji Li:** Software, Validation. **Yinglong He:** Software, Visualization. **Ziyang Li:** . **Huw Williams:** Writing - review & editing. **Hongming Xu:** Funding acqusition, Supervision, Writing - review & editing. **Shijin Shuai:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The work is partially funded by the Innovate UK (Grant 102253) for the development of world-leading off-highway vehicular technologies. It is also sponsored by the State Key Laboratory of Automotive Safety and Energy (Tsinghua University) under Project No. KF2029 for the development of digital-twin enabled online reinforcement learning. The authors gratefully acknowledge the support from Textron Ground Support Equipment UK and Hyper-Drive Innovation.

References

[1] Gregor E. EU legislation in progress CO 2 emission standards for heavy-duty vehicles 2019.
[2] European Commission. Proposal for post-2020 CO2 targets for cars and vans | Climate Action 2017.
[3] Cash S, Zhou Q, Olatunbosun O, Xu H, Davis S, Shaw R. Development of a series hybrid electric aircraft pushback vehicle: a case study. Engineering 2019;11:33–47. https://doi.org/10.4236/eng.2019.111004.
[4] Cash S, Zhou Q, Xu H, Olatunbosun O, Davis S, Shaw R. A new traction motor sizing strategy for a HEV/EV based on an overcurrent-tolerant prediction model. IET Intell Transp Syst 2018. https://doi.org/10.1049/iet-its.2018.5016.
[5] Zhou Q, Guo X, Tan G, Shen X, Ye Y, Wang Z. Parameter analysis on torque stabilization for the eddy current brake: a developed model, simulation, and sensitive analysis. Math Probl Eng 2015;2015:1–10. https://doi.org/10.1155/2015/436721.
[6] Zhou Q, Tan G, Guo X, Fang Z, Gong B. Relationship between braking force and pedal force of a pedal controlled parallelized energy-recuperation retarder system. SAE Tech Pap 2014. https://doi.org/10.4271/2014-01-1783.
[7] Guo S, Chen Z, Guo X, Zhou Q, Zhang J. Vehicle Interconnected suspension system based on hydraulic electromagnetic energy harvest: design, modeling and simulation tests. SAE Tech Pap 2014-01-2299 2014. 10.4271/2014-01-2299.
[8] Zhou Q, Guo X, Xu L, Wang G, Zhang J. Simulation based evaluation of the electro-

hydraulic energy-harvesting suspension (EHEHS) for off-highway vehicles. SAE Tech Pap 2015;2015-April. 10.4271/2015-01-1494.

[9] Ancillotti E, Bruno R, Palumbo S, Capasso C, Veneri O. Experimental set-up of DC PEV charging station supported by open and interoperable communication technologies. 2016 int symp power electron electr drives, autom motion, SPEEDAM 2016 2016:677–82. 10.1109/SPEEDAM.2016.7526036.

[10] Capasso C, Rubino G, Rubino L, Veneri O. Power architectures for the integration of photovoltaic generation systems in DC-microgrids. Energy Procedia 2019;159:34–41. https://doi.org/10.1016/j.egypro.2018.12.014.

[11] Schmalfuß F, Mühl K, Krems JF. Direct experience with battery electric vehicles (BEVs) matters when evaluating vehicle attributes , attitude and purchase intention 2017;46:47–69. 10.1016/j.trf.2017.01.004.

[12] Wu J, He H, Peng J, Li Y, Li Z. Continuous reinforcement learning of energy management with deep Q network for a power split hybrid electric bus. Appl Energy 2018;222:799–811. https://doi.org/10.1016/j.apenergy.2018.03.104.

[13] Guo H, Wang X, Li L. State-of-charge-constraint-based energy management strategy of plug-in hybrid electric vehicle with bus route. 111972 Energy Convers Manag 2019;199. https://doi.org/10.1016/j.enconman.2019.111972.

[14] Lee H, Jeong J, Park Y il, Cha SW. Energy management strategy of hybrid electric vehicle using battery state of charge trajectory information. Int J Precis Eng Manuf - Green Technol 2017;4:79–86. https://doi.org/10.1007/s40684-017-0011-4.

[15] Wang F, Zhang J, Xu X, Cai Y, Zhou Z, Sun X. A comprehensive dynamic efficiency-enhanced energy management strategy for plug-in hybrid electric vehicles. Appl Energy 2019;247:657–69. https://doi.org/10.1016/j.apenergy.2019.04.016.

[16] Chau KT, Wong YS. Overview of power management in hybrid electric vehicles. Energy Convers Manag 2002;43:1953–68. https://doi.org/10.1016/S0196-8904(01)00148-0.

[17] Peng J, He H, Xiong R. Rule based energy management strategy for a series–parallel plug-in hybrid electric bus optimized by dynamic programming. Appl Energy 2016;185:1633–43. https://doi.org/10.1016/j.apenergy.2015.12.031.

[18] Li J, Zhou Q, He Y, Williams H, Xu H. Driver-identified Supervisory Control System of Hybrid Electric Vehicles based on Spectrum- guided Fuzzy Feature Extraction 2020;6706. 10.1109/TFUZZ.2020.2972843.

[19] Al Mamun A, Liu Z, Rizzo DM, Onori S. An integrated design and control optimization framework for hybrid military vehicle using lithium-ion battery and supercapacitor as energy storage devices. IEEE Trans Transp Electrif 2019;5:239–51. https://doi.org/10.1109/TTE.2018.2869038.

[20] Li J, Zhou Q, Williams H, Xu H. Back-to-back competitive learning mechanism for fuzzy logic based supervisory control system of hybrid electric vehicles. IEEE Trans Ind Electron 2019:1. https://doi.org/10.1109/tie.2019.2946571.

[21] He Y, Zhou Q, Makridis M, Mattas K, Li J, Williams H, et al. Multi-objective co-optimization of cooperative adaptive cruise control and energy management strategy for PHEVs 2020;XX:1–10. 10.1109/TTE.2020.2974588.

[22] Pourabdollah M, Egardt B, Murgovski N, Grauers A. Convex optimization methods for powertrain sizing of electrified vehicles by using different levels of modeling details. IEEE Trans Veh Technol 2018;67:1881–93. https://doi.org/10.1109/TVT.2017.2767201.

[23] Zhou Q, Zhang Y, Li Z, Li J, Xu H, Olatunbosun O, et al. Cyber-physical energy-saving control for hybrid aircraft-towing tractor based on online swarm intelligent programming. IEEE Trans Ind Informatics 2018;14:4149–58. https://doi.org/10.1109/TII.2017.2781230.

[24] Soriano F, Moreno-Eguilaz M, Álvarez-Flórez J. Drive cycle identification and energy demand estimation for refuse-collecting vehicles. IEEE Trans Veh Technol 2015;64:4965–73. https://doi.org/10.1109/TVT.2014.2382591.

[25] Sun C, Hu X, Moura SJ, Sun F. Velocity predictors for predictive energy management in hybrid electric vehicles. IEEE Trans Control Syst Technol 2015;23:1197–204. https://doi.org/10.1109/TCST.2014.2359176.

[26] Continental Automotive GmbH. Worldwide emission standards and related regulations; 2017.

[27] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, et al. Mastering the game of Go with deep neural networks and tree search. Nature 2016;529:484–9. https://doi.org/10.1038/nature16961.

[28] Radac MB, Precup RE. Data-driven model-free slip control of anti-lock braking systems using reinforcement Q-learning. Neurocomputing 2018;275:317–29. https://doi.org/10.1016/j.neucom.2017.08.036.

[29] Q-learning C. Model-free optimal tracking control. IEEE Trans Neural Networks Learn Syst 2016;27:2134–44. https://doi.org/10.1109/TNNLS.2016.2585520.

[30] Bellman R. Dynamic programming and a new formalism in the calculus of variations. Proc Natl Acad Sci 1954;40:231–5. https://doi.org/10.1073/pnas.40.4.231.

[31] Liu T, Hu X, Li SE, Cao D. Reinforcement learning optimized look-ahead energy management of a parallel hybrid electric vehicle. IEEE/ASME Trans Mechatronics 2017;22:1497–507. https://doi.org/10.1109/TMECH.2017.2707338.

[32] Liu T, Hu X, Hu W, Zou Y. A heuristic planning reinforcement learning-based energy management for power-split plug-in hybrid electric vehicles. IEEE Trans Ind Informatics 2019:1. https://doi.org/10.1109/TII.2019.2903098.

[33] 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;255:588–601.

[34] Cao J, Xiong R. Reinforcement learning-based real-time energy management for plug-in hybrid electric vehicle with hybrid energy storage system. Energy Procedia 2017;142:1896–901. https://doi.org/10.1016/j.egypro.2017.12.386.

[35] Reddy NP, Pasdeloup D, Zadeh MK, Skjetne R. An intelligent power and energy management system for fuel cell/battery hybrid electric vehicle using reinforcement learning. ITEC 2019–2019 IEEE Transp Electrif Conf Expo 2019. https://doi.org/10.1109/ITEC.2019.8790451.

[36] Roman Liessner, Jakob Schmitt AD and BB. Hyper-parameter optimization for deep learning 2016.

[37] Li Y, He H, Peng J, Wang H. Deep reinforcement learning-based energy management for a series hybrid electric vehicle enabled by history cumulative trip information. IEEE Trans Veh Technol 2019;68:7416–30. https://doi.org/10.1109/tvt.2019.2926472.

[38] 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 2019:1–13. https://doi.org/10.1109/tnnls.2019.2927531.

[39] Wang P, Li Y, Shekhar S, Northrop WF. Actor-critic based deep reinforcement learning framework for energy management of extended range electric delivery vehicles. IEEE/ASME Int Conf Adv Intell Mechatronics, AIM 2019:1379–84. https://doi.org/10.1109/AIM.2019.8868667.

[40] Hasselt H Van, Group AC, Wiskunde C. Double Q-learning. Nips 2010:1–9.

[41] Schilperoort J, Mak I, Wiering MA. Learning to play pac-xon with Q-learning and two double Q-learning variants. IEEE Symp Ser Comput Intell 2018;2018:1151–8.

[42] Zhang Y, Sun P, Yin Y, Lin L, Wang X. Human-like autonomous vehicle speed control by deep reinforcement learning with double Q-learning. 2018-June IEEE Intell Veh Symp Proc2018:1251–6. https://doi.org/10.1109/IVS.2018.8500630.

[43] Han X, He H, Wu J, Peng J, Li Y. Energy management based on reinforcement learning with double deep Q- learning for a hybrid electric tracked vehicle. 113708 Appl Energy 2019;254. https://doi.org/10.1016/j.apenergy.2019.113708.

[44] Zhou Q, Zhang W, Cash S, Olatunbosun O, Xu H, Lu G, et al. Intelligent sizing of a series hybrid electric power-train system based on Chaos-enhanced accelerated particle swarm optimization. Appl Energy 2017;189:588–601. https://doi.org/10.1016/j.apenergy.2016.12.074.

[45] Zhang Q, Lin M, Member S, Yang LT, Member S, Chen Z, et al. A double deep Q-learning model for energy-efficient edge scheduling. IEEE Trans Serv Comput 2019;12:739–49. https://doi.org/10.1109/TSC.2018.2867482.

[46] Wang YH, Li THS, Lin CJ. Backward Q-learning: The combination of Sarsa algorithm and Q-learning. Eng Appl Artif Intell 2013;26:2184–93. https://doi.org/10.1016/j.engappai.2013.06.016.