Energy Conversion and Management 301 (2024) 117973

Contents lists available at ScienceDirect

Energy Conversion and Management

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

Deep stochastic reinforcement learning-based energy management strategy for fuel cell hybrid electric vehicles

Basel Jouda [a], Ahmad Jobran Al-Mahasneh [b,*], Mohammed Abu Mallouh [c]

[a] Electrical Engineering Department, College of Engineering, University of Prince Mugrin, Madinah, Saudi Arabia
[b] Mechatronics Engineering Department, Faculty of Engineering and technology, Philadelphia University, Amman, Jordan
[c] Mechatronics Engineering Department, Faculty of Engineering, The Hashemite University, Zarqa, Jordan

ARTICLE INFO

Keywords:
Fuel Cell Hybrid Vehicles
Energy Management Strategy
Fuel Economy Improvement
Deep Reinforcement Learning
Stochastic Reinforcement Learning

ABSTRACT

Fuel cell hybrid electric vehicles offer a promising solution for sustainable and environment friendly transportation, but they necessitate efficient energy management strategies (EMSs) to optimize their fuel economy. However, designing an optimal leaning-based EMS becomes challenging in the presence of limited training data. This paper presents a deep stochastic reinforcement learning based approach to address this issue of epistemic uncertainty in a midsize fuel cell hybrid electric vehicle. The approach introduces a deep REINFORCE framework with a deep neural network baseline and entropy regularization to develop a stochastic policy for EMS. The performance of the proposed approach is benchmarked against three EMSs: i) a state-of- art deep deterministic reinforcement learning technique called Double Deep Q-Network (DDQN), Power Follower Controller (PFC) and Fuzzy Logic Controller (FLC). Using New York City cycle as a validation drive cycle, the deep REINFORCE approach improves fuel economy by 7.68%, 13.53%, and 10% compared to DDQN, PFC, and FLC, respectively. The deep REINFORCE approach improves fuel economy by 5.31 %,9.78 %, and 9.93 % compared to DDQN, PFC, and FLC, respectively under another validation cycle, Amman cycle. Moreover, the training results show that the proposed algorithm reduces training time by 38% compared to the DDQN approach. The proposed deep REINFORCE-based EMS shows superiority not only in terms of fuel economy, but also in terms of dealing with epistemic uncertainty.

1. Introduction

The transportation sector is one of the main contributors to the air pollution and global warming [1]. Therefore, there has been a growing demand for electrified vehicles that utilize electric motor in their powertrain and thus, emit fewer pollutants. The transition from conventional vehicles to green electrified version is anticipated to happen in an increasing rate in near future. Not only customer state of mind and government policies towards greener transportation are important, but also innovation and technology development in the area of Electrified Vehicles (EVs). There are many powertrain configurations of electrified vehicles, however, fuel cell hybrid powertrain is one of the most promising solutions [2]. In general, Fuel Cells (FC) generate electricity through an electrochemical reaction that combines hydrogen with ambient air. FC's function is similar to a battery but consumes hydrogen and oxygen instead of producing electricity from stored chemical energy. Therefore, FC produces electricity as long as fuel is supplied, while battery requires frequent recharging. Compared to Internal Combustion Engines (ICEs), FCs are more efficient and emit only water vapor and heat. When FCs are used in the powertrain of an EV; usually, they are combined with other energy storage elements (e.g., batteries, ultracapacitor, …etc.) that have better dynamic response and have capability of restoring braking energy. This hybridization of FCs with energy storage elements in the powertrain is very helpful to deal with sudden changes in load or power demand, since FCs are known for their slow dynamic response, compared to batteries. [3]. Therefore, Fuel Cell Hybrid Electric Vehicles (FCHEVs) use multiple energy sources, such as FCs and batteries. This hybridization increases the complexity of operation and control of each component in the powertrain. To manage this complexity, FCHEVs need an efficient Energy Management Strategy (EMS). An EMS is a supervisory control algorithm that determines the optimal operating point of each utilized energy source to simultaneously meet the driver's demand and performance requirements. Improper distribution of power may result in lowering the system efficiency, shortening the vehicle lifespan, and endangering driving safety.

* Corresponding author.
E-mail address: almahasneh87@gmail.com (A. Jobran Al-Mahasneh).

https://doi.org/10.1016/j.enconman.2023.117973
Received 11 October 2023; Received in revised form 26 November 2023; Accepted 6 December 2023
Available online 28 December 2023

Nomenclature

A	reinforcement learning action
A_f	vehicle's front area
$a^{`}$	next action
a_t	current action
a_{t+1}	next action
a_{veh}	acceleration of the vehicle
$b(\bullet)$	baseline function
C	number of steps needed to update the target network
C_D	air drag coefficient
c	the number of continuous actions generated by the actor
c_r	rolling resistance coefficient
$f(\bullet)$	function that relates state and control variables
f_t	tractive force
G_t	return function
g	gravitational acceleration
$\mathscr{H}$	entropy loss term
I_b	battery current
J	performance index
k_{FC}	fuel consumption penalty factor
k_{soc}	state of charge maintaining penalty factor
$L(\bullet)$	cost function
M_{veh}	Mass of the vehicle
$\dot{m}_{FC}$	hydrogen fuel consumption
N	number of samples
P_b	battery output power
P_{demand}	demand power
P_{FC}	fuel cell power
P_{motor}	motor power demand
$P_\theta(\tau)$	probability distribution
Q	utility value
Q_b	battery remaining charge
Q_{max}	battery maximum charge capacity
RGR	reduction gear ratio
R_{int}	internal resistance
r	reward
r_{wheel}	wheel radius
S	chosen variables as states
s	state
$s^{`}$	next state
s_t	current state
s_{t+1}	next state
SoC	state of charge
$\dot{SoC}$	rate of change of state of charge
SoC_{ref}	reference state of charge
SoC_{max}	maximum limit of state of charge
SoC_{min}	minimum limit of state of charge
T	terminal episode
t	time
t_0	trip start time
t_f	trip end time
u(t)	control variables
$u(t)_{min}$	minimum constraints of control variables
$u(t)_{max}$	maximum constraints of control variables
V_b	battery's terminal voltage
V_{OC}	open circuit voltage
V_{veh}	vehicle speed
w_{motor}	rotation speed of the motor
$x(t)_{min}$	minimum constraints of the state variables
$x(t)_{max}$	maximum constraints of the state variables
x(t)	state variables
y^{DDQN}	target network Q-values
α	road gradient
α_c	actor learning rate
α_{eval}	evaluation network learning rate.
α_v	learning rate of value function network
γ	discount factor
δ_t	advantage function
ε	probability of choosing to select a random action
θ	parameters of the REINFORCE action policy
θ_{eval}	evaluation network parameters
θ^*	the optimal REINFORCE actor parameters
λ	entropy loss weight
μ_{coul}	coulombic efficiency
$\mu_{DC/DC}$	efficiency of the unidirectional dc/dc
μ_{fd}	reduction gear efficiency
μ_{motor}	efficiency of the motor
$\pi(a\|s,\theta)$	reinforce policy
π^*	optimal policy
ρ_a	density of the air
σ	standard deviation
τ	the trajectories τ that are sampled from the probability distribution
$\varphi(\bullet)$	soft constraints

Therefore, EMS design and analysis is crucial for FCHEVs operation. Researchers have developed different EMSs for FCHEVs, which can be classified into three groups: rule-based, optimization-based, and learning-based EMSs.

The power distribution in the rule-based EMS requires the use of rule and lookup tables. The commonly used rule-based strategies are deterministic rules-based EMS and Fuzzy Logic Controllers (FLCs). The authors in [4] compared the classical proportional-integral controller with different rule-based methods namely FLC, frequency decoupling, and state machine. The lowest fuel consumption was achieved using FLC with good response time. Luciani et. al [5] also compared different rule-based EMSs for an FCHEV. The FLC achieved the lowest fuel consumption and highest FC efficiency. However, the rule-based EMSs need frequent calibration and it is difficult to achieve multi-objective energy management using it. In [6], an adaptive, causal, and scalable rule-based EMS was proposed. The rules were initiated using two-dimensional dynamic programming, and power adjustments were made based on the estimated average power. Different optimization techniques have been used to tune the parameters of rule-based EMS in [7]. However, the resulting strategies may suffer from a lack of adaptability when driving conditions (i.e., driving cycles) change.

The optimization-based EMS is another class of EMSs. This type of EMS is further classified into two categories: global optimization and instantaneous optimization. One of the most common global optimization EMS approaches is dynamic programming. Dynamic programming is used to control FCHEV by considering the relationship between driving conditions and different control modes [8]. In [9], Pontryagin's minimum principle, another global optimization technique, was found to achieve near-optimal solutions for the energy distribution problem. However, global optimization methods are typically used for fixed and known driving conditions [10]. On the other hand, instantaneous optimization is concerned with optimizing vehicle's operation at a particular moment based on the current vehicle state and environmental conditions. Ferrara et al. [11]. proposed an instantaneous optimization-based EMS named Equivalent Consumption Minimization Strategy (ECMS) for managing energy in heavy-duty hybrid vehicles and compared it against various optimization techniques for real-life driving cycles. Similarly, the authors in [12] developed a hierarchical EMS for FCHEV based on terrain information using predictive ECMS, FC longevity-conscious strategy, and battery State of Charge (*SoC*) trajectory planning. This EMS enhanced fuel consumption and the FC lifespan considering uphill conditions. Even though these instantaneous optimization algorithms can achieve excellent results, they also pose challenges as they depend heavily on the vehicle's model accuracy and can be computationally complex [13]. Therefore, they are still active research topic.

The third EMS class is Learning-based EMSs. These EMSs utilize Machine Learning (ML) algorithms to reach a near-optimal instantaneous energy distribution. Learning-based EMSs provide excellent flexibility to unanticipated driving conditions, quick reaction, and little computing cost. In general, they are divided into Neural Network (NN)-based EMS and Reinforcement Learning (RL) based-EMSs. NN-based EMSs are mostly used to optimize operating conditions in electrified powertrains. NN can be trained to predict or classify speed and other driving conditions [14], and can also learn the optimal policy by mimicking other optimal algorithms [15]. Recently, RL-based EMSs are receiving significant attention owing to the advantages, such as interactive learning, that RL offers compared to other methods. RL is an ML approach where agents learn to make decisions by trial and error to satisfy an objective function. RL agents can learn different types of policies. These policies can be either deterministic or stochastic, depending on how the RL actions are chosen. Deterministic RL agents make decisions by following deterministic policies. Such agents are widely used in literature for example, the authors in [16] developed Q-learning algorithm as the EMS in a FCHEV. Similarly, the authors in [17] introduced a Q-learning based- EMS, and the authors initialized the Q-table with different preset rules obtained from rule-based EMSs. In [18], Li et al. proposed a method to manage the energy in a FCHEV that considers both the energy demand and the fuel consumption. The authors used a Deep Neural Network (DNN) to approximate the Q-function instead of a Q-table, which reduces the computational cost. The authors of [19] used a deterministic RL algorithm called Deep Deterministic Policy Gradient (DDPG) to adjust the parameters of the artificial potential field that controls the energy flow for optimal efficiency and performance. However, their DDPG algorithm may overestimate the Q-value and cause policy degradation [20]. To overcome this problem, Twin Delay Deep Deterministic Policy Gradient (TD3) was developed. Zhou et al. [21] developed a TD3-based EMS for a parallel hybrid vehicle that counts for the road slope and the number of passengers. In [22], a similar TD3-based EMS was applied to manage the energy sources in a hybrid railway vehicle.

On the other hand, stochastic RL agents are agents that incorporate probabilistic decision-making processes to take an action. Stochastic RL agents have an edge over deterministic RL in dealing with uncertainties in the environment [23]. Such uncertainties may arise from changes in driving and road conditions, and energy sources in FCHEVs [24]. Moreover, stochastic RL-algorithm can learn effective policies even in the presence of limited training data or uncertainty. For example, a stochastic RL-based demand management of EV charging stations was proposed in [25]. The proposed algorithm was able to handle uncertainties in the environment, where it outperformed both DDPG and TD3.

1.1. *Motivation and contribution*

Despite the numerous advantages associated with stochastic RL agents, their utilization as EMSs for FCHEVs remains limited. This article seeks to investigate the implementation of a deep stochastic RL-based EMS on a midsize FCHEV, particularly when dealing with limited training data, obtained from drive cycles. The chosen variant of deep stochastic RL agents is the Monte Carlo policy gradient algorithm known as REINFORCE [26]. A DNN is utilized in this article to approximate the policy probability distribution for the stochastic RL-based EMS. To mitigate the variance effects from the Monte Carlo approximation, this article uses a DNN to estimate the state-value function that serves as a baseline. Moreover, to enhance the agent's exploration an entropy regularization term that increases the uncertainty when the agent takes actions. In this study, the proposed algorithm will be referred to as deep REINFORCE. To the best of the authors' knowledge, the application of deep REINFORCE-based EMS in hybrid electric vehicles has not been explored previously. The effectiveness of the proposed deep REINFORCE-based EMS in terms of fuel economy will be assessed through a comparison with three common EMSs used for FCHEVs: a state-of-the art deep deterministic RL-based EMS known as the Double Deep Q-Network (DDQN), ii) a Power Follower Controller (PFC) and iii) an FLC-based EMS.

1.2. *Organization*

After this introduction, the paper is structured as follows: Section 2 provides the modeling of the FCHEV utilized in this study. Section 3 formulates the energy management problem and introduces both the deep REINFORCE and the DDQN-based EMSs. In Section 4, training and validation results are presented and analyzed. Finally, conclusions are presented in section 0.

2. FCHEV modeling

Fig. 1 shows the powertrain configuration of a front wheel driven FCHEV. As depicted in Fig. 1, the FC and battery pack are connected in parallel using a unidirectional DC/DC converter. An AC electric motor, connected with the wheels through a reduction gear mechanism, receives power from these energy sources through a DC/AC inverter. The generated torque is transferred to the wheels though the transmission. This configuration is referred to as a direct parallel connection for the battery, as it connects directly to the power bus. This connection enables four operating modes depending on the power demand and the *SoC* of the battery pack:

1. Propelling using FC alone: the FC powers the motor alone and the excess power will charge the battery.
2. Propelling using the battery alone: the electronic interface will isolate the FC from the bus and the battery will power the traction motor alone.
3. Hybrid propelling: both the FC and the battery pack power source will supply power to the motor.
4. Regenerative braking: when the motor decelerates, the motor acts as a generator, and the induced regenerative energy will charge the battery under its constraints (e.g., maximum charging current, temperature … etc.)

Based on the vehicle's status and energy demand, the EMS will determine the appropriate mode to be selected from the previously described modes, and the amount of power each energy source contributes toward the satisfying the driver's demand power. FCHEV dynamic modelling.

The force applied to the wheels to propel the vehicle is known as the

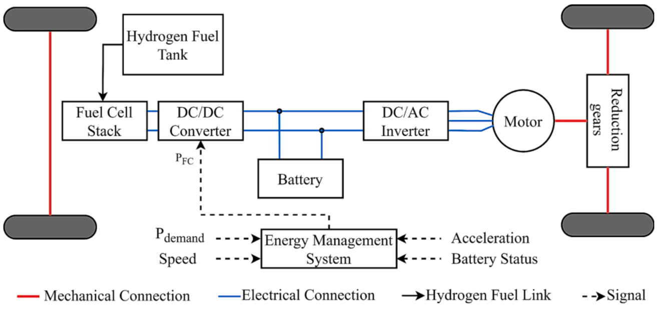

Fig. 1. The powertrain configuration of the FCHEV.

tractive force, f_t, and it is calculated as the summation of forces acting on a vehicle as it accelerates uphill, , as shown in Fig. 2. The demand power, P_{demand}, is the tractive force multiplied by the vehicle speed. f_t and P_{demand} are given in Eq. (1).

$$\begin{cases} f_t = M_{veh} g c_r cos\alpha + 0.5 C_D \rho_a A_f {V_{veh}}^2 + M_{veh} g sin\alpha + M_{veh} a_{veh} \\ P_{demand} = f_t V_{veh} \end{cases} \tag{1}$$

where M_{veh} is the vehicle's mass, g is the gravitational acceleration, c_r rolling resistance coefficient, α is the road gradient, C_D is the air drag coefficient, ρ_a is the density of the air, A_f is the vehicle's front area, V_{veh} speed of the vehicle, and a_{veh} is the acceleration of the vehicle.

2.1. Fuel cell system modelling

FCs generate electricity through an electrochemical reaction between hydrogen and oxygen, and produce water vapor a byproduct. FCs have some advantages over classical ICEs, such as having higher efficiency, producing zero pollutants, low noise, and low vibration [27]. Due to these advantages, utilizing FCs in transportation are very promising solution to reduce air pollution and to reduce dependency on fossil fuels. There are several types of FCs, and they are often categorized based on the electrolyte they use. Proton Exchange Membrane (PEM) FCs are the most common choice in the automotive industry and are expected to see increased usage in cars, buses, trucks, heavy-duty vehicles, trains, and airplanes [28]. To describe the relation between the fuel consumption and the output power from a FC, a quasi-static model is utilized. In this article, the FC-ANL50H2 model, developed by ADVISOR team [29], is used. Fig. 3 shows the efficiency and fuel consumption according to the output power for the utilized 55 kW PEM FC stack system.

2.2. Battery modelling

A battery pack is needed in the powertrain to store regenerative braking energy and to provide power to the motor when needed. Among the different types of batteries available for transportation applications, lithium-ion batteries are the most used one. Lithium-Ion batteries exhibit high energy density and long life-cycle. Additionally, they are less likely to experience memory effects and have a low self-discharge rate, which explains their popularity in electric vehicles [30]. Therefore, a 3.45 kWh lithium-ion battery pack is employed for the FCHEV in this research. Lithium-ion batteries can be modeled using internal resistance model as shown in Fig. 4. This model consists of an open circuit voltage source denoted by V_{OC}, and the internal resistance of the battery denoted by R_{int}. The battery output power, P_b, can be calculated as follows:

$$P_b = V_b I_b = V_{OC} I_b \text{-} R_{int}\ {I_b}^2 \tag{2}$$

The battery's terminal voltage, denoted as V_b, takes into account the open circuit voltage and the voltage drop across R_{int}. Additionally, I_b , given in Eq (3), represents the battery's current.

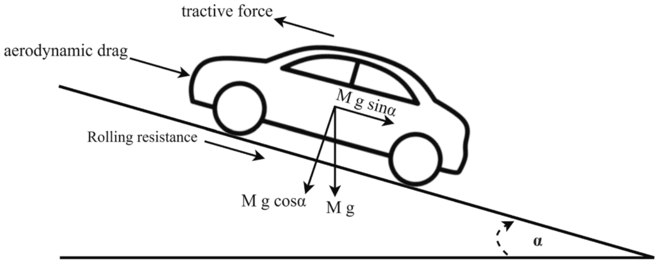

Fig. 2. The forces acting on a vehicle accelerating uphill.

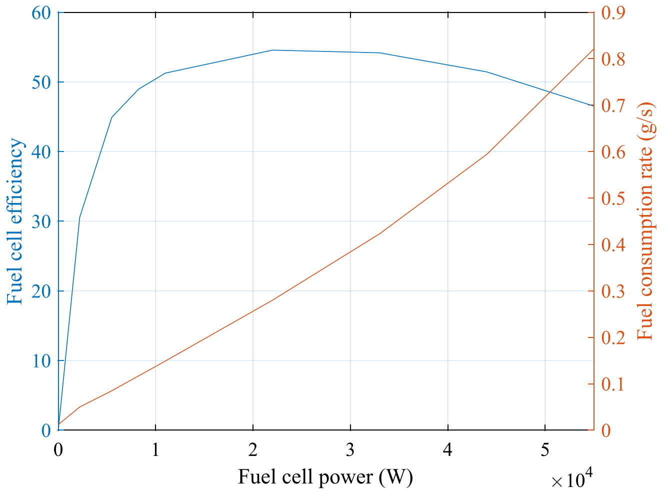

Fig. 3. The efficiency and fuel consumption of the chosen 55-kW PEM fuel cell.

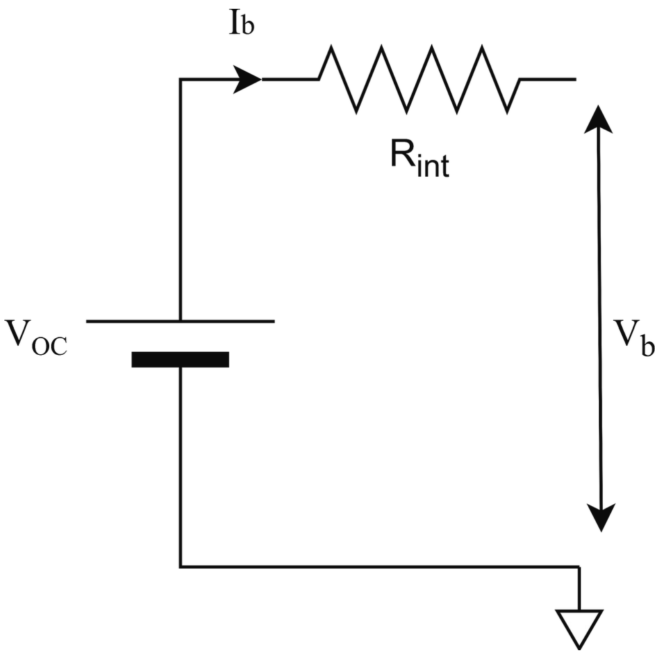

Fig. 4. Equivalent circuit diagram of the lithium-ion battery.

$$I_b = \left(V_{OC} \text{-} \sqrt{{V_{OC}}^2 \text{- } 4R_{int}\ P_b} \right) \Big/ (2R_{int}) \tag{3}$$

The amount of stored energy in a battery can be measured by determining the remaining charge available for use. This is expressed as the battery's SoC, which is defined as the ratio of the remaining charge Q_b to the maximum charge capacity Q_{max} of the battery, as shown in Eq. (4)

$$SoC = Q_b / Q_{max} \tag{4}$$

According to [31], the equation used to calculate the rate of change of *SoC*, $S\dot{o}C$, during battery charging or discharging is as follows:

$$S\dot{o}C = \begin{cases} -I_b / Q_{max} & I_b > 0 \\ -I_b \mu_{coul} / Q_{max} & I_b < 0 \end{cases} \tag{5}$$

where μ_{coul} is Coulombic efficiency and it accounts for charge losses. Fig. 5 shows how V_{OC} and R_{int} change with the battery's SoC.

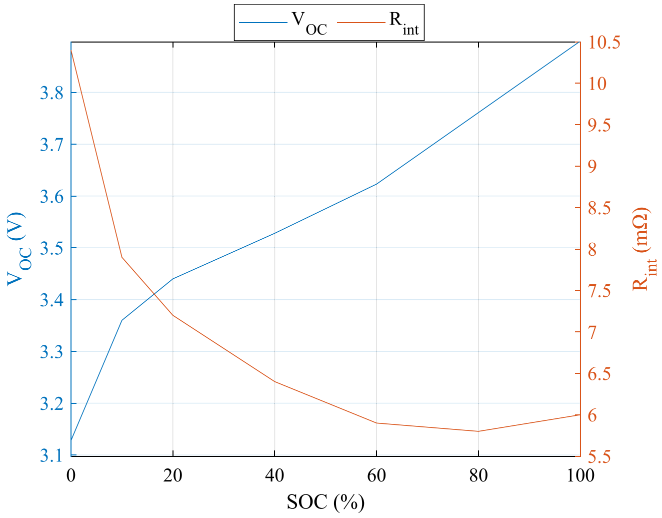

Fig. 5. Equivalent circuit diagram of the lithium-ion battery.

2.3. Electric motor modelling

An electric propulsion system forms the heart of an electrified powertrain, comprising essential components such as an electric traction motor, power converters, and electronic controllers. The primary role of the electric motor within this system is to transform electrical energy into mechanical energy, propelling the vehicle forward. Additionally, the motor can function as a generator, recovering energy during braking or deceleration. The motor power demand P_{motor}, can be calculated according to the efficiency of the motor, μ_{motor}, as follows:

$$\begin{cases} P_{motor} = \begin{cases} r_{wheel}f_t w_{motor}/(\mu_{rd}\ \mu_{motor}RGR) & \text{motor} \\ r_{wheel}f_t w_{motor}\mu_{rd}\mu_{motor}/RGR & \text{generator} \end{cases} \\ w_{motor} = V_{veh}RGR/r_{wheel} \end{cases} \tag{6}$$

where r_{wheel} is the wheel radius, w_{motor} is the rotation speed of the motor, μ_{fd} reduction gear efficiency, and *RGR* is the Reduction Gear Ratio. Since the efficiency is associated with speed of rotation and motor torque, the efficiency can be determined using an efficiency map. Fig. 6 shows the efficiency map of the utilized 107 kW AC induction motor. The motor power demand in the FCHEV is provided by the two energy sources, the FC stack and the battery pack as follows:

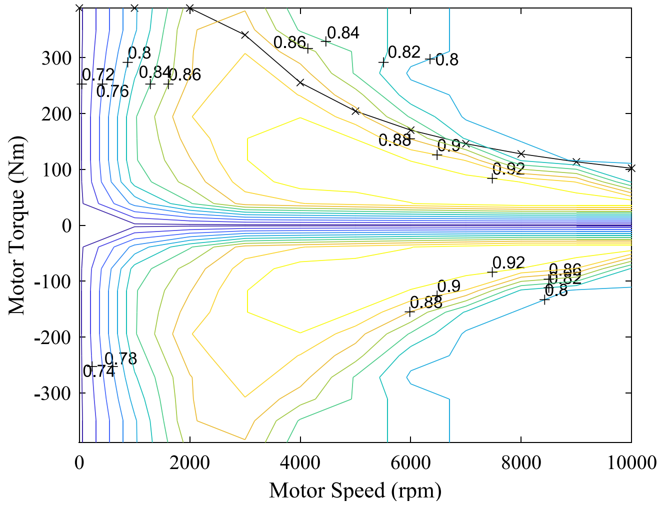

Fig. 6. The efficiency map of the traction motor for the FCHEV.

$$P_{motor} = P_{FC}\mu_{DC/DC} + P_b \tag{7}$$

where P_{FC} is the FC stack power, and $\mu_{DC/DC}$ is the efficiency of the unidirectional DC/DC converter.

3. Rl-based EMS for FCHEV

The task of the EMS is to determine the optimal energy split between the FC and the battery pack. The EMS tries to minimize fuel consumption without violating the system constraints (e.g., output power limits, battery *SoC* limits…etc.). In this section, the energy management problem is first formulated,

then the proposed deep REINFORCE-based EMS is described, and finally DDQN-based EMS used for comparison is explained.

3.1. Rl-based energy management problem formulation

In general, energy control problems are typically formulated by

considering several critical factors, including the state variable, control variable, and the objective function. The system dynamic equation can be expressed as follows:

$$\dot{x}(t) = f\ (x(t),\ u(t),\ t\ ) \tag{8}$$

where *x*(*t*) and *u*(*t*) are the state and control variables at time *t*, respectively. *f*(•) is the function that relates *x*(*t*) and *u*(*t*). For FCHEVs, the vehicle status (e.g., speed, acceleration, power demand … etc.) should be efficiently represented by the state variables; the control variable should be the power that the FC needs to provide. In order to solve the FCHEV energy management problem, one must identify the optimal power split, *u*(t), for minimizing a performance index, *J*, over a certain trip from the start, t_0, to the end of the trip, t_f. The performance index is given by Eq. (9).

$$J = \varphi(x(t_f)) + \int_{t_0}^{t_f} L(x(t), u(t), t)dt \tag{9}$$

The function *φ*(•) represents soft constraints. It is incorporated as a penalty term in the cost function to prevent the optimal solution from violating these constraints. *L*(•) is the cost function that represents the objective of the control problem The objective function, in this research, is the objective of improving the fuel economy and maintaining the battery *SoC*. The optimal solution for the energy management problem should not violate the constraints on the state and control variables shown below:

$$\begin{aligned} x(t)_{min} \le x(t) \le x(t)_{max} \\ u(t)_{min} \le u(t) \le u(t)_{max} \end{aligned} \tag{10}$$

where $x(t)_{min}$ and $x(t)_{max}$ are the minimum and maximum constraints of the state variables, respectively. Additionally, $u(t)_{min}$ and $u(t)_{max}$ are the minimum and maximum constraints of control variables, respectively.

RL is a learning method that employs the interactions between an agent and its environment to iteratively find a set of optimal control actions that maximize a long-term reward function [32]. Fig. 7 shows how the RL-based EMS interacts with the vehicle's environment. As depicted from Fig. 7, the controller gives an action (energy distribution) and observes the states of the vehicle (speed, acceleration, power demand, *SoC*). According to the performance index (i.e., reward), the controller learns the optimal control action. With this iterative learning process, the controller can learn how to efficiently distribute the energy between the energy sources in the vehicle to achieve the lowest fuel consumption without violating any constraints.

The adopted RL-based EMSs in this research can handle continuous observations, thus they can process more state variables that represent the vehicle status. The vehicle speed, acceleration, power demand, and

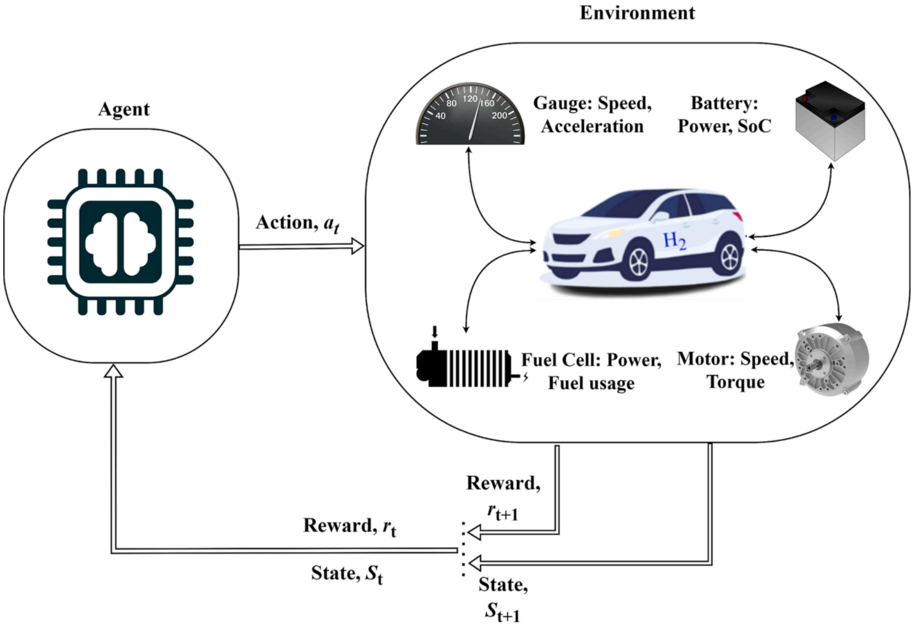

Fig. 7. The RL controller (agent) interacts with the FCHEV (environment).

battery *SoC* were chosen to be the states for the RL-based EMSs as expressed in Eq. (11).

$$S = \{V_{veh}, a_{veh}, P_{demand}, SoC\} \tag{11}$$

The agent's action is the power demand from the FC, P_{FC}, which ranges from 0 to 55 kW. The deep REINFORCE algorithm can produce continuous actions, while the DDQN can only outputs discrete actions. Therefore, the DDQN's output is discretized with a step size of 0.275 kW, as shown below:

$$A = P_{FC} = [0\ 0.275\ 0.55\ \cdots\ 54.45\ 54.725\ 55]\ \text{kW} \tag{12}$$

The reward function influences how to optimize the network and influences the policy that the agent learns, i.e., the reward function is the performance index for the RL-based EMS. The reward function, *r*, is designed to minimize fuel consumption and sustain the battery *SoC* between the minimum ($SoC_{\min} = 0.6$) and maximum ($SoC_{\max} = 0.8$) limit as depicted in Eq. (13). The RL-based EMS should not violate the system constraints shown in Eq. (14).

$$r = -\begin{cases} k_{FC}\dot{m}_{FC} + k_{soc}(SoC_{ref} - SoC)^2 & SoC > 0.8 or SoC < 0.6 \\ k_{FC}\dot{m}_{FC} & otherwise \end{cases} \tag{13}$$

$$\begin{array}{c} 0.6 \le SoC \le 0.8 \\ 0kW \le P_{FC} \le 50kW \end{array} \tag{14}$$

where SoC_{ref} is the reference *SoC* for charge sustaining task and it is set to 0.7, k_{FC} is the fuel consumption penalty factor, $\dot{m}_{FC}$ is the hydrogen fuel consumption, and k_{soc} is a penalty factor for maintaining the *SoC* in the specified range.

3.2. Deep stochastic RL-based EMS design

Stochastic RL agents are agents that follow stochastic policies to select actions. These agents introduce randomness into their learning process to develop a policy that can explore different actions and be able to handle uncertainties. Introducing randomness to the agent can help develop more adaptable agents by subjecting them to different scenarios and mitigate the negative effects of any changes that may occur in the environment.

Deep learning is a branch of machine learning that uses neural networks with multiple layers to process and learn from raw data. The depth of the network, which refers to the number of layers, enables deep learning algorithms to automatically extract higher-level features from raw data [33]. This is different from traditional machine learning-based applications, where a lot of work is required to manually extract useful features. Deep learning algorithms can learn hierarchical representations from the data itself without human intervention. Deep stochastic RL agents use deep learning methods to handle uncertainty or randomness. These agents have several advantages, such as learning meaningful representations from the data without manual feature engineering, thanks to the complex pattern and feature extraction abilities of DNNs. Moreover, deep stochastic RL agents can adapt well to new or partially observed states because of DDNs approximation capabilities. Therefore, the vehicle can adapt and generalize by utilizing deep stochastic RL agents to manage energy in FCHEVs, while taking into consideration the uncertainties that may arise in the environment. For this work, the chosen variant of stochastic RL algorithms is the Monte Carlo agent or commonly known as REINFORCE. Deep stochastic RL is employed to develop an energy management strategy during the training stage. This strategy is achieved through a cost function known as the reward, carefully designed to account for considerations such as fuel economy and the SoC of the battery. In contrast to static rule-based methods traditionally used in energy management, deep stochastic RL involves dynamic learning process.

REINFORCE is a stochastic, policy-based, on-policy, and model-free RL algorithm. The policy, $\pi(a|s, \theta)$, can be modeled as a stochastic function that assigns a probability of choosing an action "*a*" for any given state "*s*" [26]. In this work, a DNN with a set of parameters θ is used to represent this stochastic function. The actor network can output the mean and standard deviation of a Gaussian distribution over the actions. Then the action can be sampled from this distribution using a normal distribution as shown in Fig. 8. This figure shows how the actor interacts with the FCHEV environment. The actor takes an action a_t, based on the current state s_t, and then observes the next state s_{t+1} and the reward r_{t+1}. A sequence of experience, $[s_0, a_0, r_1, s_1, \ldots, a_{T-1}, r_{T-1}, s_T]$, is generated during this interaction. This process continues until the actor reaches the terminal state s_T.

Learning a policy involves the process of maximizing an objective function, denoted as J. In continuous RL algorithms, this objective

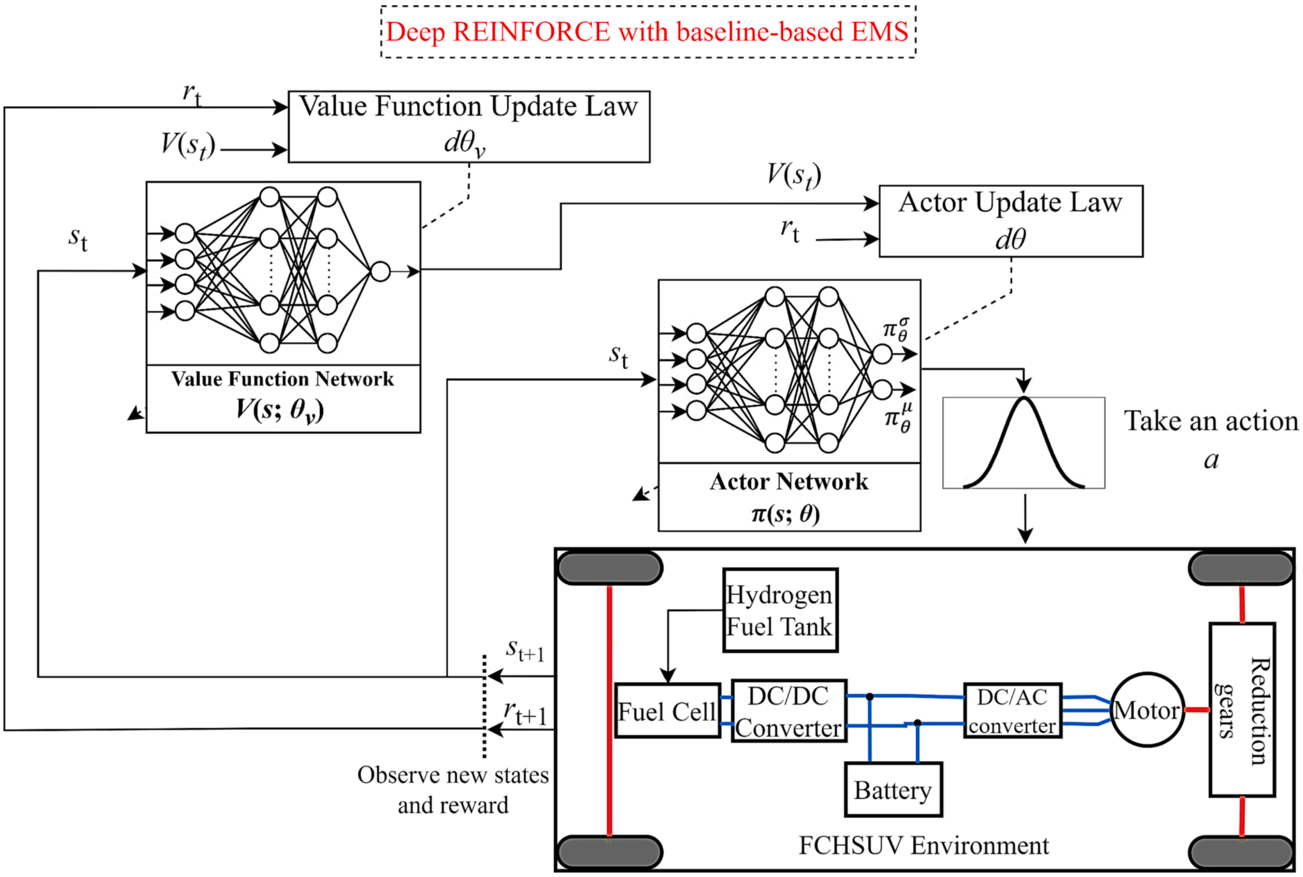

Fig. 8. Block diagram shows the interaction of Deep REINFORCE based-EMS with the FCHEV.

function is designed to maximize the cumulative discounted reward known as the return G_t as shown in Eq (15).

$$J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\sum_{k=t}^{T}\gamma^{k-t}r(s_k,a_k)\right] = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[G_t] = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[G(\tau)] \tag{15}$$

where γ is the discount factor, τ is trajectories sampled from following a policy π_θ, and θ is the parameters of a policy π_θ. Eq. (15) represents the expected discounted reward obtained by following the policy π_θ over a certain time horizon t. The expectation is taken over the trajectories τ that are sampled from the probability distribution, $P_\theta(\tau)$, of the policy approximation, which depends on θ. This distribution can be computed by multiplying the conditional probabilities of each state and action in the trajectory as given in Eq. (16).

$$P_\theta(\tau) = P_\theta(s_1,a_1,\cdots,s_T,a_T) = P(s_1)\prod_{t=1}^{T}\pi_\theta(a_t|s_t)P(s_{t+1}|s_t,a_t) \tag{16}$$

where $P(s_1)$ is the initial state distribution, and $P(s_{t+1}|s_t,\ a_t)$ is the transition probability from state s_t to state s_{t+1} after taking action a_t.

According to Eq (17), to find the optimal policy, π^*, we need to maximize the objective function. In REINFORCE, the gradient ascent is used to maximize J, i.e., the policy parameters are updated in the direction of the gradient of J.

$$\pi^* = \theta^* = \underset{\theta}{\text{argmax}}J(\pi_\theta) = \underset{\theta}{\text{argmax}}\mathbb{E}_{\tau\sim\pi_\theta(\tau)}[G(\tau)] \tag{17}$$

The objective function can be reformulated to simplify the implementation of gradient ascent as given in Eq (18).

$$J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[G(\tau)] = \int\pi_\theta(\tau)G(\tau)d\tau \tag{18}$$

The gradient ascent of J with respect to policy parameters θ:

$$\nabla_\theta J(\pi_\theta) = \int\nabla_\theta\ \pi_\theta(\tau)G(\tau)\ d\tau \tag{19}$$

The integral can be eliminated by using the log-likelihood trick, which states that:

$$\nabla_\theta\pi_\theta(\tau) = \pi_\theta(\nabla_\theta\pi_\theta(\tau)/\pi_\theta(\tau)) = \pi_\theta(\tau)\nabla_\theta\ \ln\ \pi_\theta(\tau) \tag{20}$$

Substituting Eq (19) in (20), the gradient ascent becomes:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[\nabla_\theta\ \ln\ \pi_\theta(\tau)G(\tau)] \tag{21}$$

The term ∇_θ ln $\pi_\theta(\tau)$ can be further simplified as follows:

$$\nabla_\theta ln\pi_\theta(\tau) = \sum_t\nabla_\theta ln\pi_\theta(a_t|s_t) \tag{22}$$

The simplified objective function is given in Eq (23).

$$\begin{aligned}\nabla_\theta J(\pi_\theta) &= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\sum_t\nabla_\theta ln\pi_\theta(a_t|s_t)G(\tau)\right] = \nabla_\theta J(\pi_\theta)\\ &= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\sum_t G_t\nabla_\theta ln\pi_\theta(a_t|s_t)\right]\end{aligned} \tag{23}$$

Sometimes the expectation of $J(\pi_\theta)$ may be difficult or impossible to evaluate analytically. To estimate this expectation, Monte Carlo approximation is used. The fundamental concept of such algorithm is generating a number of N samples from the distribution $P_\theta(\tau)$. Subsequently, the sample average of the gradients is utilized as an estimate for the gradient of $J(\theta)$.

$$\nabla_\theta J(\pi_\theta) \approx 1/N\sum_{i=1}^{N}\left[\sum_t G_t^{(i)}\nabla_\theta ln\pi_\theta\left(a_t^{(i)}|s_t^{(i)}\right)\right] \tag{24}$$

Using Monte Carlo approximation to evaluate the expectation introduces high variance to the learning process, i.e., the high variance causes the policy to be unstable and slow to converge [34]. On possible solution for reducing variance is introducing an appropriate baseline, $\mathrm{b}(\mathrm{s}_t)$, to the gradient of J [35] as follows:

$$\nabla_\theta \mathrm{J}(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\sum_t \nabla_\theta \ln \pi_\theta(\mathrm{a}_t|\mathrm{s}_t)(\mathrm{G}_t - \mathrm{b}(\mathrm{s}_t))\right] \tag{25}$$

$$\nabla_\theta \mathrm{J}(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\sum_t \nabla_\theta \ln \pi_\theta(\mathrm{a}_t|\mathrm{s}_t)(\delta_t)\right] \tag{26}$$

where δ_t is referred to as the advantage function. To encourage the exploration in policy gradient algorithms, a regularization term called entropy loss, $\mathscr{H}$, is added to increase the uncertainty in the policy distribution [36]. The agent's uncertainty about the best action to choose is reflected by the entropy value. The higher the entropy, the more the agent explores different actions. Therefore, increasing the entropy loss term helps the agent to explore more. More gradients are added to reduce the entropy loss function. Hence the gradient of J with the entropy loss becomes:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\sum_t \nabla_\theta \ln\pi_\theta(a_t|s_t)(\delta_t) + \lambda\nabla_\theta \mathscr{H}(\pi_\theta(\bullet|s_t))\right] \tag{27}$$

where λ is the entropy loss weight. The use of the dot notation indicates that the entropy is a function of the policy distribution over all possible action. The entropy can be computed as:

$$\mathscr{H}(\pi_\theta(\bullet|s_t)) = (1/2)\sum_{k=1}^{c} \ln(2\pi e\sigma_k^2) \tag{28}$$

where c is the number of continuous actions generated by the actor, and σ_k is the standard deviation of action k in state s under the current policy. Finally, using Eq. (27), the update law for the actor parameters can be formulated as follows:

$$\theta \leftarrow \theta + \alpha_c \nabla_\theta J(\pi_\theta) \tag{29}$$

where α_c is the actor learning rate. The baseline can be any variable or function that does not depend on the action, i.e., $\nabla_\theta b(s_t) = 0$. A possible choice of baseline function is state-value approximator

$\mathrm{V}(\mathrm{s}_t)$. A DNN with parameters θ_v is employed to approximate the state-value. θ_v can be updated according to Eq (30).

$$\theta_v \leftarrow \theta_v + \alpha_v \delta_t \nabla_{\theta_v} V(s_t) \tag{30}$$

where α_v is the learning rate of value function network. The pseudocode of deep REINFORCE with baseline and entropy regularization-based EMS is shown in Table 1.

Table 1
The pseudocode of the proposed deep REINFORCE-based EMS.

Deep REINFORCE with baseline and entropy regularization -based EMS algorithm

```
Initialize the actor $\pi(a|s,\theta)$
Initialize the value function $V(s_t)$
For each episode:
Initialize state $s$
Generate episode experience by following actor policy $\pi(s)$, $s_0$, $a_0$, $r_1$,$s_1$, …, $s_T$,$a_\mathrm{T}$,$r_T$
For $t = 1, 2, \ldots, T$:
Calculate the return $G_t = \sum_{k=t}^{T}\gamma^{k-t}r(s_k, a_k)$
Compute the advantage function $\delta_t = G_t - V(s_t)$
Accumulate the gradients for the value function network
$\nabla_{\theta_v} V(s_t)$
Update the value-function network
$\theta_v \leftarrow \theta_v + \alpha_v\delta_t\nabla_{\theta_v} V(s_t)$
Accumulate the gradients for the actor network
$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\sum_t \nabla_\theta \ln\pi_\theta(a_t|s_t)(\delta_t) + \lambda\nabla_\theta\mathscr{H}(\pi_\theta(\bullet|s_t))\right]$
Update the actor network
$\theta \leftarrow \theta + \alpha_c\nabla_\theta J(\pi_\theta)$
End for
End for
```

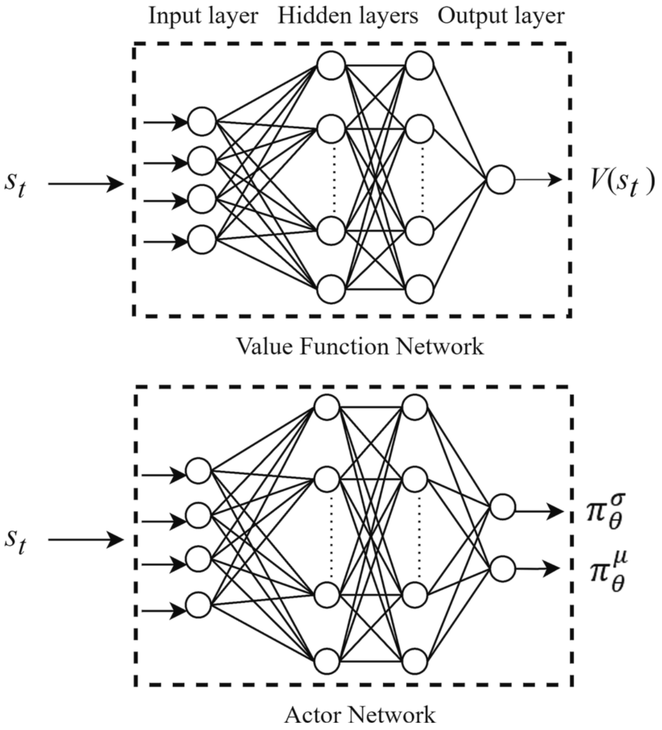

Fig. 9. Deep REINFORCE neural network architecture.

The proposed deep REINFORCE-based EMS utilizes two DNNs: the actor and the value function approximator, as illustrated in Fig. 9. The actor network comprises an input layer, hidden layers, and an output layer. The hidden layers are composed of three fully connected layers, each with 50, 100, and 150 neurons. Additionally, nonlinear tansigmoid activation layers are incorporated between the fully connected layers to enhance the neural network's flexibility. The actor network takes the vehicle's states as input and computes the mean and standard deviation of the probability distribution for each possible continuous action. To approximate the state-value function, another DNN with the same layer structure as the actor network is employed. The value function network calculates the expected return from that state following a specific policy given the vehicle's state. Because the driving conditions and driver preferences are constantly changing, and the vehicle's power allocation decisions must adapt to those changes, the proposed deep REINFORCE-based EMS learns a stochastic policy. This means that the algorithm can explore different potential actions and choose the one that is most likely to lead to a good outcome, even in unseen conditions. This adaptability is essential for navigating the complex and unpredictable driving environment.

3.3. DDQN-based EMS

To showcase the effectiveness of utilizing a deep stochastic RL-based EMS, it becomes important to compare its performance with a deterministic RL-based EMS. For this purpose, a deep deterministic RL-based EMS named DDQN is employed as a point of comparison. This subsection proceeds to describe the framework of DDQN-based EMS, outlining its key components and operations for a comprehensive understanding of its functioning.

Q-learning is a deterministic, value-based, off-policy RL algorithm that employs a *Q*-table to store the rewards associated with different actions. Thus, the mapping of action-state pairs is done using a table

[32]. However, using a table for this mapping becomes impractical in complex environments with a large state space. This is referred to as the problem of curse of dimensionality, where the size and complexity of the table grow exponentially as the state space expands. Deep *Q*-Network (DQN) addresses this limitation by utilizing DNN to represent the *Q*-table. Using DNNs, the RL algorithm can recognize the changes in the state variables, which improves its decision-making and allows it to generalize and adapt to new data [37]. To improve the convergence ability, two DNNs will be used. for the DQN, called evaluation and target networks. This configuration is known as DDQN. The target network computes the target *Q*-values during the training process, while the evaluation network chooses the action based on the state i.e., it interacts with the environment. Fig. 10 shows how the DDQN-based EMS interacts with the FCHEV. The agent interacts with the vehicle and collects observations, actions, and rewards; then store them in a replay buffer memory.

During the training process, mini-batches are randomly sampled from the replay buffer. The evaluation network employs these sampled mini-batches to estimate the *Q*-values for state-action pairs, $Q(s, a; \theta_{\mathrm{eval}})$. The estimated *Q*-values represent the expected cumulative rewards for taking a particular action "*a*" in a given state "*s*" parameterized by the network's parameters θ_{eval}. While the target network utilizes these sampled mini-batches to generate target *Q*-values, y^{DDQN}, as given in Eq. (31). These target *Q*-values serve as a reference for training and updating the evaluation network. Finally, the evaluation network is trained using backpropagation that minimizes the cost function *L* as given in Eq. (32). *L* is defined as the mean squared error between the estimated *Q*-value of the target *Q*-value. Every *C* step, the target network copies the parameters of the evaluation network.

$$y_i^{DDQN} = r_i + \gamma Q(s_i^{'}, \underset{a^{'}}{argmax} Q(s_i^{'}, a^{'}; \theta_{eval}); \theta^{-}) \tag{31}$$

$$L(\theta_{eval}) = (1/2M)\sum_{i=1}^{M} (y_i^{DDQN} - Q(s_i, a_i; \theta_{eval\,i}))^2 \tag{32}$$

where θ^{-} is the parameters of the target network, *M* is the minibatch size, and *i* is the *i*-th step. Finally, the evaluation network is updated as follows:

$$\theta_{eval} \leftarrow \theta_{eval} + \alpha_{eval} \nabla_{\theta_{eval}} L(\theta_{eval}) \tag{33}$$

where α_{eval} is the evaluation network learning rate. In order to discover optimal actions, an agent must explore the environment. If the agent explores too frequently, it may struggle to establish a stable policy. Therefore, the agent also needs to exploit its existing knowledge by choosing actions that yield maximum rewards. On the other hand, complete exploitation in every episode can lead the agent to get stuck in a local optimum. It is crucial to deploy an algorithm to balance the exploration and exploitation. The epsilon (ε)-greedy is used to select the action that balances between exploration and exploitation using probability a distribution as given in Eq. (34).

$$a(t) = \begin{cases} \underset{a(t)}{\max} Q(s(t+1),\ a(t)) & \text{with probability } (1\text{-}\varepsilon) \\ \text{random } a(t) & \text{with probability } (\varepsilon) \end{cases} \tag{34}$$

where ε is the probability of choosing to explore. The pseudocode of the DDQN-based EMS can be found in Table 2. The DDQN employs two DNNs. Each DNN consists of three fully connected layers with 50, 100, and 150 neurons each. Nonlinear tan-sigmoid activation layers were also added between the fully connected layers.

4. Simulation and performance assessment

The study involves comparing the developed RL-based EMSs with two widely used rule-based controllers, namely the PFC, deterministic-rule based strategy, and the FLC. This comparison aims to

Table 2
The pseudocode of DDQN-based EMS.

```
DDQN-base EMS algorithm

Initialize $Q(s,a; \theta_{eval})$
Initialize $Q(s,a; \theta^{-})$
Initialize replay buffer D
For each episode:
Initialize state ($s$)
For each step:
Select random action $a(t)$ with probability $\varepsilon$, otherwise select action $a(t) = \underset{a(t)}{max} Q(s(t+1), a(t))$
Store transition T ($s$, $a$, $r$, $s'$) in D
Sample random minibatch from D
If s' is terminal state set the value function $y_i^{DDQN}$ to r, otherwise set it to $y_i^{DDQN} = r_i + \gamma Q(s_i^{'}, \underset{a^{'}}{argmax} Q(s_i^{'}, a^{'}; \theta_{eval}); \theta^{-})$
Update evaluation network parameters by minimizing:
$L(\theta_i) = (1/2M)\sum_{i=1}^{M} (y_i^{DDQN} - Q(s, a; \theta_{eval}))^2$
$\theta_{eval} \leftarrow \theta_{eval} + \alpha_{eval} \nabla_{\theta_{eval}} L(\theta_{eval})$
Every C step reset $\theta^{-} = \theta_{eval}$
Update $s = s'$
End for
End for
```

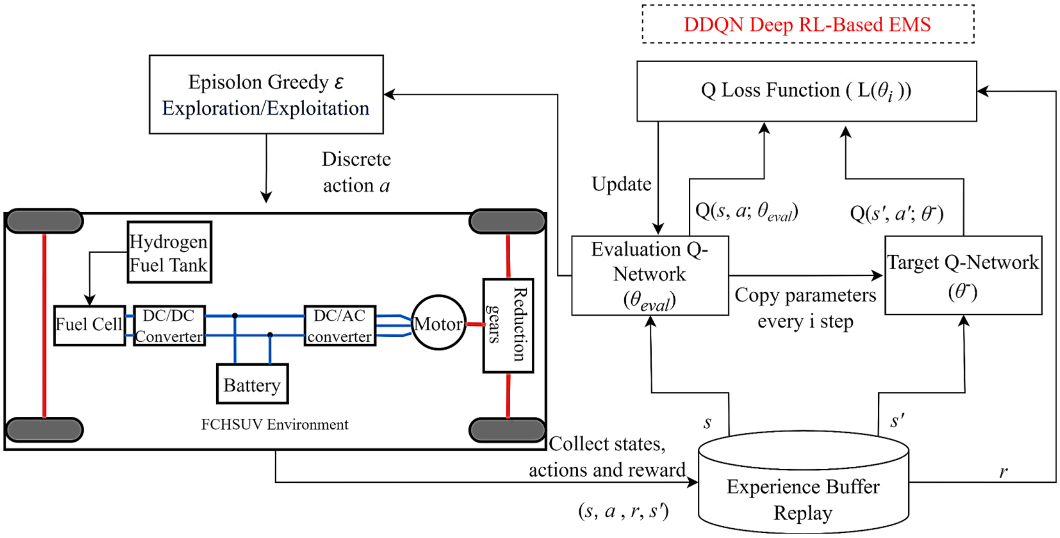

Fig. 10. DDQN based-EMS interacts with the FCHEV.

demonstrate the effectiveness of utilizing deep RL-based EMSs. The midsize FCHEV model is created in the MATLAB/Simulink environment using ADVISOR [29]. The specification of the adopted FCHEV was primarily taken from [38], are outlined in Table 3. The adopted models of the FC stack, battery pack, motor, and vehicle body all were previously utilized and validated [29,18] and [39].

4.1. Driving cycles

Four driving cycles were chosen to train and validate the developed deep RL-based EMSs. Fig. 11a and Fig. 11b illustrate the training drive cycles, namely the Urban Dynamometer Driving Cycle (UDDS) and the High Way Fuel Economy Test (HWFET). They were selected to encompass both urban and highway driving characteristics. In contrast, Fig. 11c and Fig. 11d present the validation cycles—the New York City Cycle (NYCC) and an experimental drive cycle for Amman city developed by Mallouh et al. [40], respectively. UDDS, HWFET, and NYCC, as standard environmental protection agency cycles. Including Amman cycle ensures generalizability in the deep REINFORCE-based EMS.

4.2. Fuel economy assessment method

The developed EMSs in this article are designed to improve fuel economy and charge sustaining task. Therefore, the difference in the final and the initial *SoC*, denoted as Δ*SoC*, should be as close to zero as possible. However, if Δ*SoC* values for the all EMSs are far apart, it becomes challenging to impartially assess the fuel economy achieved by each strategy. To address this issue and eliminate any discrepancies in Δ*SoC* among the EMSs, Δ*SoC* should be considered in the calculation of fuel economy. Various.

methods have been suggested to address this issue, such as a simple method proposed in [41]. This method varies the initial *SoC* for each run, then computes the difference in fuel economy caused by changing the initial *SoC*, denoted as Δ Fuel economy. As a result, pairs of Δ*SoC* and Δ Fuel economy are obtained, which are then used to fit a linear line, as depicted in Fig. 12. By leveraging the linear relationship between Δ*SoC* and Δ Fuel economy, it becomes possible to calculate the fuel economy while taking into consideration the Δ*SoC*. In Fig. 12, the Δ*SoC* versus Δ Fuel economy curve is illustrated. It is worth mentioning that the inclusion of ΔSoC in the calculation depends on the drive cycle. Fig. 12a shows Δ Fuel economy with UDDS and Fig. 12b with NYCC for the PFC-based EMS. The same procedure was applied to assess the fuel economy resulted by the other EMSs with the chosen drive cycle.

4.3. Deep RL-based EMSs training results

The deep REINFORCE algorithm uses two DNNs, one for the.

actor and one for the value function approximation. The DDQN also utilizes two DNNs, the evaluation network and the target network. All the RL-based EMSs have the same DNN structure to evaluate their training results. Table 4 and Table 5 present the hyperparameters for

Table 3
Specification of the developed FCHEV from ADVISOR.

Component	Specifications	Value
Vehicle	Total mass:	1843 kg
	Electric Accessory:	700 W
	vehicle frontal area:	2.66 m^2
	Wheel radius	0.34 *m*
	aerodynamic drag coefficient:	0.44
	Rolling resistance coefficient	0.009
	reduction gear ratio	10.55
	Reduction gear efficiency	0.94
PEM fuel cell	Maximum power:	55 kW
	Maximum efficiency:	0.55
Motor	Maximum power	107 kW
	Maximum efficiency	0.94
Li-ion battery pack	Capacity	3.45 kWh
	Coulombic efficiency	0.99
DC/DC converter	Efficiency	0.95

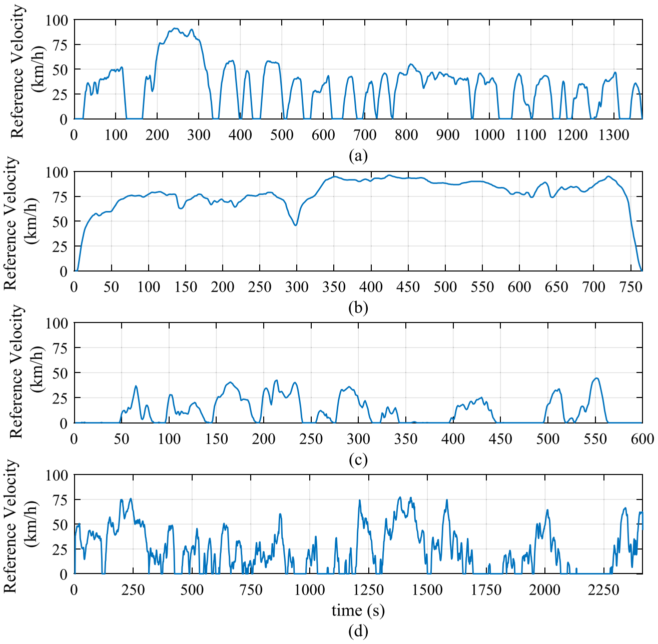

Fig. 11. The chosen drive cycles (a) UDDS, (b) HWFET, (c) NYCC, and (d) Amman.

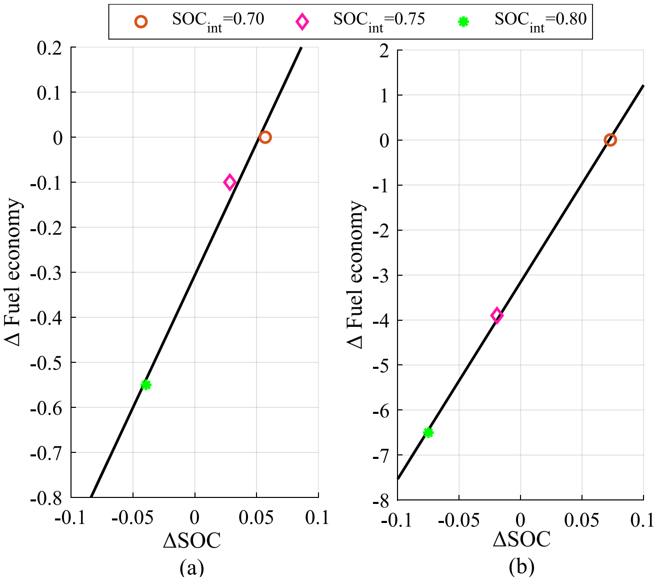

Fig. 12. ΔSOC versus Δ Fuel economy curve for PFC under (a) UDDS and (b) NYCC, the dotted points are Δ Fuel economy when the initial SOC changes.

Table 4
The chosen hyperparameter for the deep REINFORCE-based EMS.

Hyperparameter	Value
Discount factor,γ	0.99
Value-function learning rate,α_v	0.001
Actor learning rate,α_c	0.0001
Entropy loss weight,λ	0.01
Penalty factor,k_{FC}	2.5
Penalty factor,k_{soc}	10

deep REINFORCE and DDQN, respectively. These hyperparameters were selected based on common RL practices and trial and error. The training was conducted for three hundred episodes on core I5-10210U running at clock frequency of 1.6 GHz. The cumulative average reward curves for the developed RL algorithms under the training cycles are shown in Fig. 13. The figure shows that the DDQN algorithm learns quickly, as indicated by the sharp rise in average reward during the early episodes. Moreover, the DDQN curve has relatively small fluctuations, which become even smaller around the 220th episode. This may be due to the use of a target network that stabilizes the Q-value updates in DDQN. The deep REINFORCE algorithm also learns fast, with a significant increase in average reward in the initial episodes. However, the deep REINFORCE curve has higher fluctuations than the DDQN curve. The main reason for the fluctuations in the REINFORCE algorithm is its reliance on the Monte Carlo estimation of the policy gradient. Moreover, the oscillations in training stage for REINFORCE is partially attributed to the fact that the algorithm is an on-policy RL method, i.e., it updates its policy by directly sampling trajectories during each episode. In contrast to the DDQN, an off-policy method, which updates its policy using experience replay, storing and sampling past experiences, and updating less frequently than REINFORCE. The fluctuations in the deep REINFORCE curve gradually decrease over time, as the policy is improved. Table 6 shows that the REINFORCE algorithm reduced the training time by 38 % compared to the DDQN algorithm. However, the DDQN network converged faster than the REINFORCE algorithm.

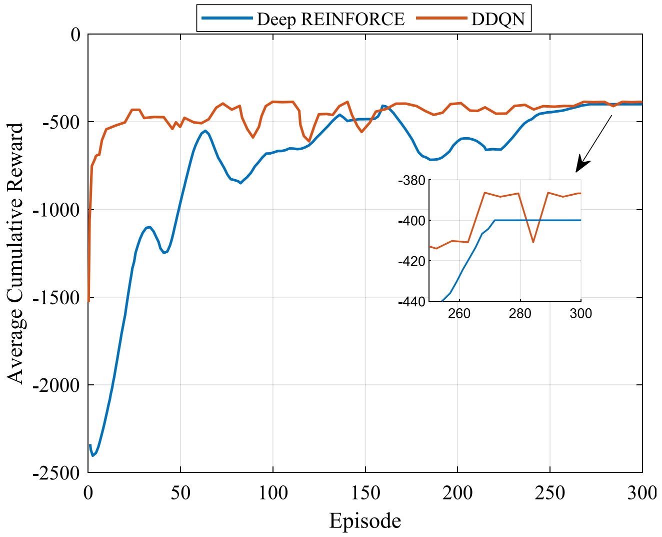

Fig. 13. Average cumulative reward curves for the developed deep RL-based EMSs.

Table 6
Training performance comparison of the developed deep RL-based EMSs.

Deep RL-based EMS	Final average Reward	Convergence Episode	Training time (hours)
Deep REINFORCE	−400	263	5.08
DDQN	−386	226	8.2

4.4. Performance assessment under training cycles

The power allocation profile presented in Fig. 14 shows the dynamic power management within the FCHEV using the proposed deep REINFORCE algorithm during the UDDS cycle. The deep REINFORCE-based EMS showcases its adaptability in optimizing power for various speed profiles. As the vehicle.

accelerates, the intelligent EMS efficiently engages the FC to enhance propulsion, ensuring a consistent power supply to the electric motor. Moreover, the figure illustrates the performance of the proposed algorithm emphasizing the system's flexibility and efficiency in managing power in the vehicle.

In Fig. 15, the battery *SoC* trajectories of the deep RL-based EMSs are compared with the benchmark EMSs, PFC, and FLC. The figure vividly demonstrates how each EMS manages to keep the SoC within the predefined limits. As depicted by Fig. 15a, the PFC and FLC tend to continuously charge the battery without effectively utilizing its stored energy. In contrast, the deep RL-based EMSs showcase a more intelligent approach in their ability to utilize the battery while maintaining its charge. Moreover, Fig. 15a and Fig. 15b show that the SoC trajectory falls below the lower limit when employing the deep REINFORCE algorithm. However, the proposed algorithm proves its effectiveness by successfully restoring the battery charge and maintaining it above the lower limit, as depicted in the figure.

The method described in section 4.2 is used to fairly assess the fuel economy while considering the differences between the *SoC* resulted by each EMS. The fuel economy, measured by gasoline equivalent (Le/100 km), results are listed in Table 7 in which the last column shows the improvement provided by the proposed algorithm. The deep RL-based EMSs where able to outperform the rule-based benchmark controllers in both urban and highway driving conditions. The Deep REINFORCE algorithm was able to improve the fuel economy by 5.32 % and 5.92 % compared to the PFC and FLC under UDDS cycle, respectively. While it achieved 1.25 % and 1.75 % improvement compared to the PFC and FLC under HWFET, respectively. The proposed algorithm was outperformed by DDQN, which achieved 2.30 % and 3.14 % higher performance under UDDS and HWFET, respectively. This can be.

Table 5
The chosen hyperparameters for the DDQN-based EMS.

Hyperparameter	Value
Experience buffer replay size	100,000
Mini-batch size, M	512
Critic learning rate,α_{eval}	0.001
Discount factor,γ	0.99
Epsilon greedy coefficient, ε	0.01
Penalty factor,k_{FC}	2.5
Penalty factor,k_{soc}	10

attributed to the replay buffer mechanism of DDQN, which enables it to learn from the past experiences. This could lead to better exploitation of the training data and a more stable training process, resulting in better performance under training data.

To ensure the reliability of the developed deep RL-based EMSs, the powertrain efficiency under training cycles is reported in Table 7. The deep REINFORCE and DDQN-based EMSs are able to achieve higher powertrain efficiency. This is attributed to the adaptability of the deep RL-based EMSs over conventional PFC and FLC-based EMSs. However, since DDQN can exploit training data, the DDQN was able to achieve the highest powertrain efficiency.

4.5. Adaptability analysis under epistemic uncertainty

The chosen training cycles have limited speed and acceleration profiles, which means they do not represent the full range of driving patterns. As a result, an RL-based EMS trained exclusively on these cycles may not have enough exposure to handle the complexities and variations present in other drive cycles. This is known as epistemic uncertainty [42], which is the uncertainty that stems from limited data or incomplete knowledge. This case represents the actual learning process in real life, where training data may be limited. To test the adaptability of the developed deep RL-based EMSs under epistemic uncertainty, they are tested on two different drive cycles namely NYCC and Amman. Fig. 16 illustrates the *SoC* trajectories of the battery during the validation cycles. The deep RL-based EMSs depicted in the figure effectively

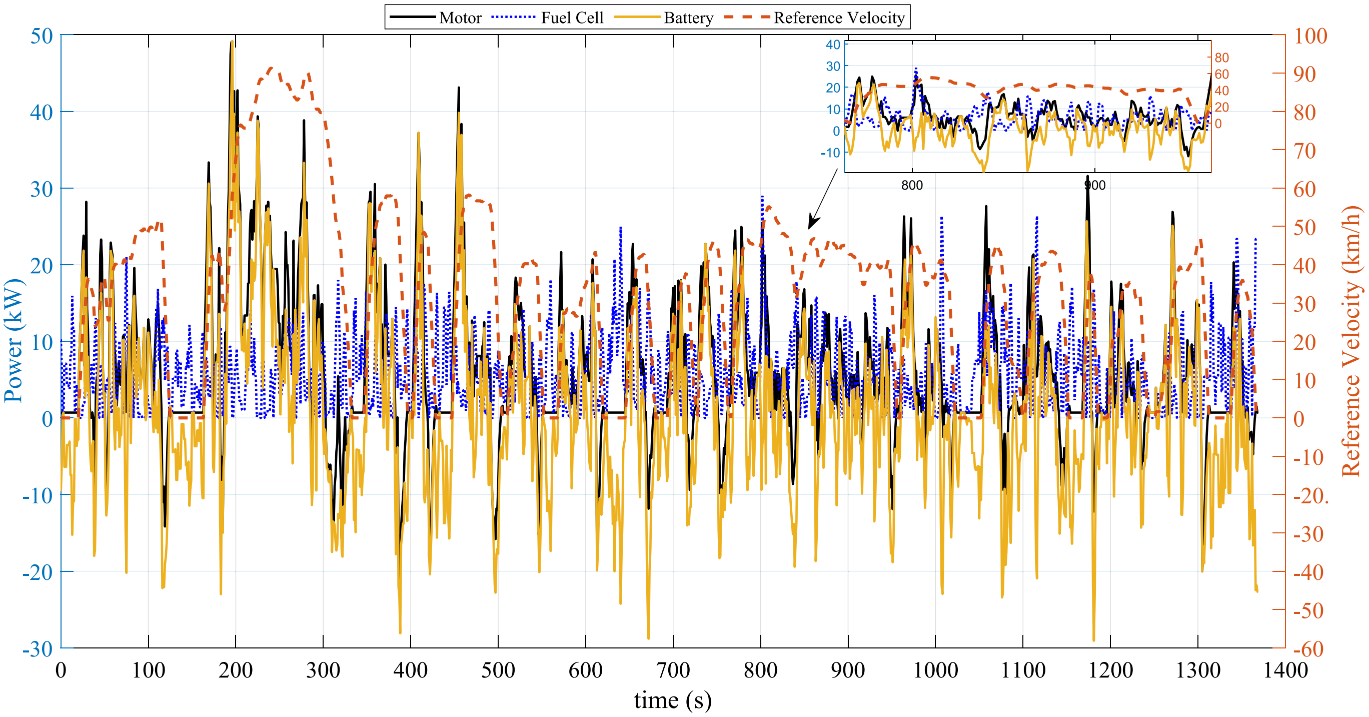

Fig. 14. Power allocation for the proposed deep REINFORCE-based EMS under UDDS cycle.

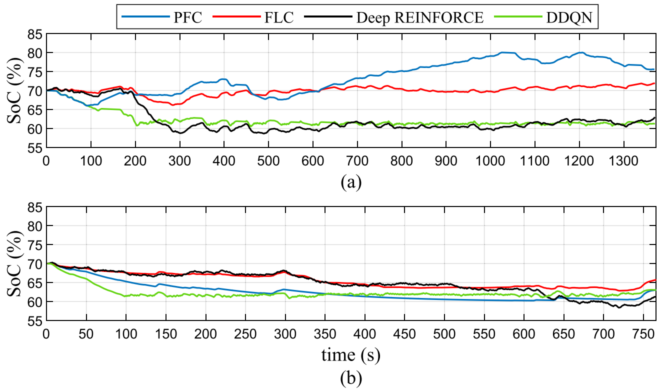

Fig. 15. Battery *SoC* trajectories of different strategies under training cycles (a) UDDS, (b) HWFET.

maintain the battery's charge while intelligently utilizing its energy. However, the proposed algorithm has managed to maintain a relatively higher *SoC* values at the end of the validation cycles. The results of the controllers in terms of fuel economy under the validation cycles are shown in Table 8. The proposed algorithm outperformed PFC and FLC by 13.53 % and 10.75 %, respectively, under NYCC. It also achieved improvements of 9.78 % and 9.93 % under Amman cycle when compared to PFC and FLC, respectively. As shown in Table 7, DDQN achieved the best fuel economy. However, when DDQN was applied to the validation cycles, its performance deteriorated compared to the deep REINFORCE algorithm. The proposed algorithm was able to improve the

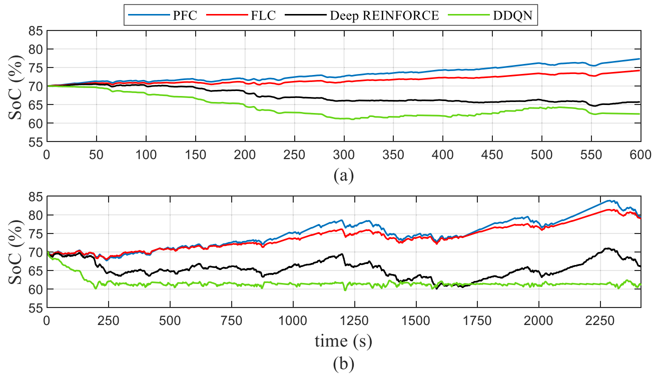

Fig. 16. Battery SoC trajectories of different strategies under training cycles (a) NYCC, (b) Amman.

Table 7
Fuel economy results under training cycles.

Drive cycle	EMS	Δ*SoC* (%)	Powertrain Efficiency (%)	Fuel economy (gasoline equivalent Le/100 km)	Improvement (%)*
UDDS	PFC	5.50	39.72 %	4.7	5.32 %
	FLC	2.00	38.76 %	4.73	5.92 %
	Deep REINFORCE	−7.00	40.38 %	4.45	–
	DDQN	−8.80	**41.34 %**	**4.35**	−2.30 %
HWFET	PFC	−7.00	40.46 %	3.99	1.25 %
	FLC	−4.24	39.51 %	4.01	1.75 %
	Deep REINFORCE	−8.80	41.49 %	3.94	–
	DDQN	−7.00	**41.87 %**	**3.82**	−3.14 %

* Improvement provided by deep REINFORCE.

Table 8
Fuel economy results under validation cycles.

Drive cycle	EMS	Δ*SoC* (%)	Powertrain Efficiency (%)	Fuel economy (gasoline equivalent Le/100 km)	Improvement (%)
NYCC	PFC	7.33	30.65 %	9.31	13.53 %
	FLC	4.18	31.32 %	9.02	10.75 %
	Deep REINFORCE	−4.00	**33.42 %**	**8.05**	–
	DDQN	−7.50	31.41 %	8.72	7.68 %
Amman	PFC	9.32	39.22 %	5.93	9.78 %
	FLC	9.00	38.54 %	5.94	9.93 %
	Deep REINFORCE	−4.00	**43.49 %**	**5.35**	–
	DDQN	−8.70	40.64 %	5.65	5.31 %

*Improvement provided by deep REINFORCE.

fuel economy by 7.68 % and 5.31 % under NYCC and Amman cycles, respectively, compared to DDQN. The power allocation presented in Fig. 17, shows the dynamic performance using the proposed deep REINFORCE algorithm under Amman cycle. The deep REINFORCE-based EMS showcases its adaptability in optimizing power for real life cycles ensuring its adaptability when faced with unseen cycles. As the vehicle accelerates, the developed EMS engages the FC to ensure a.

consistent power supply to the electric motor. To ensure the reliability of the developed deep RL-based EMSs under validation cycles, the powertrain efficiency under validation cycles is reported in Table 8. Similar to the training cycles, both deep REINFORCE and DDQN achieved higher powertrain efficiency than PFC and FLC-based EMSs, showcasing their ability to adapt to different drive cycles. Notably, during validation cycles, the deep REINFORCE-based EMS demonstrated superior powertrain efficiency compared to DDQN, indicating its capacity to maintain efficiency even with unfamiliar cycles. The average FC efficiency with deep REINFORCE-based EMS is 48.00 %, 49.00 %, 41.00 %, and 49.50 % under UDDS, HWFET, NYCC, and Amman respectively. The algorithm consistently performs well, in terms of average FC efficiency, during UDDS, HWFET, and Amman, yielding high average powertrain efficiency. However, in NYCC, the lower FC efficiency contributes to a reduced powertrain efficiency, as detailed in Table 8.

This demonstrates how the proposed algorithm can effectively learn to optimize the vehicle's performance even with limited training data. The deep REINFORCE algorithm uses a stochastic policy to make decisions. This stochasticity helps the agent explore different actions and encourages it to discover more effective strategies. By exploring different possibilities, the algorithm can find better policies and adapt to different scenarios, which leads to improved performance when exposed to unseen data.

5. Conclusion

In this paper, a deep stochastic reinforcement learning-based EMS for FCHEV under epistemic certainty is proposed. First, the model of the FCHEV along with its components are developed. Then, the framework of improving fuel economy and maintaining the battery charge using deep REINFORCE-based EMS is formulated. The variance in deep REINFORCE algorithm is reduced by utilizing a DNN to approximate the value function. Moreover, to improve the exploration in the proposed stochastic policy, an entropy regularization term is added. The epistemic uncertainty is introduced by training the developed algorithm on training cycles with limited speed and acceleration profiles.

To showcase the effectiveness of utilizing deep REINFORCE-based EMS for FCHEVs, the developed algorithm performance is compared with DDQN, a deterministic algorithm, and two rule-based EMS benchmarks: PFC and FLC. The training results show that the deep

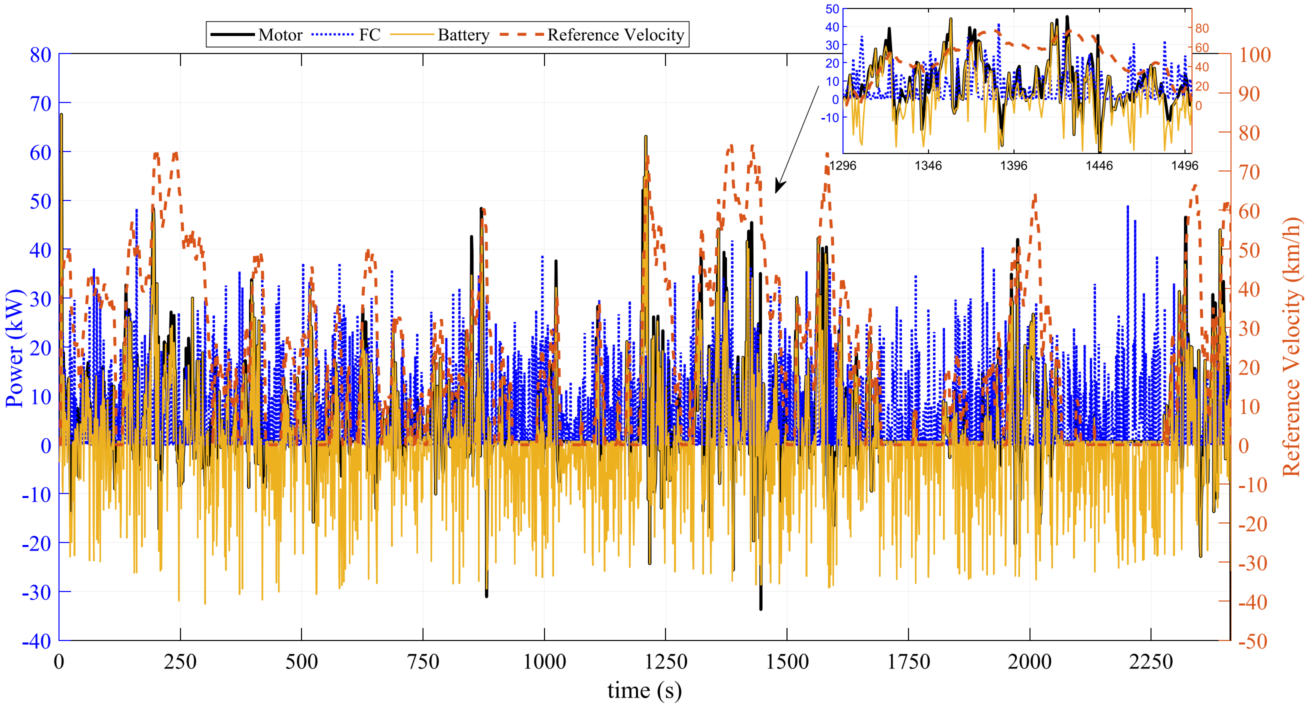

Fig. 17. Power allocation for the proposed deep REINFORCE-based EMS under Amman cycle.

REINFORCE was able to reduce training time by 38 % compared to the DDQN algorithm. While DDQN was able to exploit training data, its performance was worse compared to deep REINFORCE under never seen before data, i.e., the validation cycles. The proposed algorithm improved fuel economy by 7.68 % and 5.31 % under NYCC and Amman cycles compared to DDQN, respectively. Furthermore, the deep REINFORCE consistently outperformed the PFC and FLC-based EMSs across training and validation cycles. At the core of the proposed EMS algorithm lies its utilization of a stochastic policy, which facilitates diverse action exploration. This attribute empowers the agent to learn more effective strategies, leading to improved performance under unseen data.

6. Limitations and future work

When implementing deep stochastic RL methods some limitations might be present including the computational demand since such models require more computational resources than the classical EMS methods. Also, the initial training stage should be carried out before implementing such systems in the hardware. To ensure safety and reliability when deploying deep stochastic reinforcement learning-based energy management strategy in actual fuel cell hybrid electric vehicles, limitations on the vehicle speed and acceleration are to be utilized to avoid sudden changes in vehicle speed and exceeding speed limits. For this purpose, a saturation on the vehicle speed between desired low and maximum speeds should be utilized in the EMS deployment. Although the developed EMS in this paper can effectively improve fuel economy under epistemic uncertainty, the powertrain durability was not investigated. Future research will explore prolonging the life of FCHEV using deep stochastic RL-based EMSs.

CRediT authorship contribution statement

Basel Jouda: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Visualization, Writing – original draft, Writing – review & editing. **Ahmad Jobran Al-Mahasneh:** Conceptualization, Methodology, Software, Resources, Writing – review & editing, Supervision, Project administration. **Mohammed Abu Mallouh:** Conceptualization, Funding acquisition, Methodology, Resources, Supervision, 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.

Data availability

Data will be made available on request.

Acknowledgments

This research was supported by the Deanship of Scientific Research and Graduate Studies at Philadelphia University, and the Deanship of Scientific Research at Hashemite University (project No. 13/338/2019).

References

[1] U.S. Environmental Protection Agency, "The Sources and Solutions: Fossil Fuels," [Online]. Available: https://www.epa.gov/nutrientpollution/sources-and-solutions-fossil-fuels. [Accessed: January 1,2023].

[2] Song K, Chen H, Wen P, Zhang T, Zhang B, Zhang T. A comprehensive evaluation framework to evaluate energy management strategies of fuel cell electric vehicles. Electrochim Acta 2018;292:960–73. https://doi.org/10.1016/j.electacta.2018.09.166.

[3] Fathabadi H. Combining a proton exchange membrane fuel cell (PEMFC) stack with a Li-ion battery to supply the power needs of a hybrid electric vehicle. Renew Energy 2019;130. https://doi.org/10.1016/j.renene.2018.06.104.

[4] Soumeur MA, Gasbaoui B, Abdelkhalek O, Ghouili J, Toumi T, Chakar A. Comparative study of energy management strategies for hybrid proton exchange membrane fuel cell four wheel drive electric vehicle. J Power Sources 2020;462: 228167. https://doi.org/10.1016/j.jpowsour.2020.228167.

[5] Luciani S, Tonoli A. Control strategy assessment for improving PEM fuel cell system efficiency in fuel cell hybrid vehicles. Energies 2022;15:2004. https://doi.org/10.3390/en15062004.

[6] Peng H, Li J, Thul A, Deng K, Ünlübayir C, Löwenstein L, et al. A scalable, causal, adaptive rule-based energy management for fuel cell hybrid railway vehicles learned from results of dynamic programming. eTransportation 2020;4:100057. https://doi.org/10.1016/j.etran.2020.100057.

[7] Ye K, Li P, Li H. Optimization of hybrid energy storage system control strategy for pure electric vehicle based on typical driving cycle. Math Probl Eng 2020;2020: 1–12. https://doi.org/10.1155/2020/1365195.

[8] Zhou W, Yang L, Cai Y, Ying T. Dynamic programming for new energy vehicles based on their work modes Part II: Fuel cell electric vehicles. J Power Sources 2018;407:92–104. https://doi.org/10.1016/j.jpowsour.2018.10.048.

[9] Xu L, Ouyang M, Li J, Yang F, Lu L, Hua J. Application of Pontryagin's Minimal Principle to the energy management strategy of plugin fuel cell electric vehicles. Int J Hydrog Energy 2013;38:10104–15. https://doi.org/10.1016/j.ijhydene.2013.05.125.

[10] Pisu P, Rizzoni G. A comparative study of supervisory control strategies for hybrid electric vehicles. IEEE Trans Control Syst Technol 2007;15:506–18. https://doi.org/10.1109/TCST.2007.894649.

[11] Ferrara A, Jakubek S, Hametner C. Energy management of heavy-duty fuel cell vehicles in real-world driving scenarios: Robust design of strategies to maximize the hydrogen economy and system lifetime. Energy Convers Manag 2021;232: 113795. https://doi.org/10.1016/j.enconman.2020.113795.

[12] Lin X, Xu X, Lin H. Predictive-ECMS based degradation protective control strategy for a fuel cell hybrid electric vehicle considering uphill condition. eTransportation 2022;12:100168. https://doi.org/10.1016/j.etran.2022.100168.

[13] Zhu Y, Li X, Liu Q, Li S, Xu Y. Review article: A comprehensive review of energy management strategies for hybrid electric vehicles. Mech Sci 2022;13:147–88. https://doi.org/10.5194/ms-13-147-2022.

[14] Liu Y, Li J, Chen Z, Qin D, Zhang Y. Research on a multi-objective hierarchical prediction energy management strategy for range extended fuel cell vehicles. J Power Sources 2019;429:55–66. https://doi.org/10.1016/j.jpowsour.2019.04.118.

[15] Yavasoglu HA, Tetik YE, Ozcan HG. Neural network-based energy management of multi-source (battery/UC/FC) powered electric vehicle. Int J Energy Res 2020;44: 12416–29. https://doi.org/10.1002/er.5429.

[16] 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, IEEE transportation electrification conference and expo (ITEC). IEEE 2019;2019:1–6. https://doi.org/10.1109/ITEC.2019.8790451.

[17] Li W, Ye J, Cui Y, Kim N, Cha SW, Zheng C. A speedy reinforcement learning-based energy management strategy for fuel cell hybrid vehicles considering fuel cell system lifetime. International Journal of Precision Engineering and Manufacturing-Green Technology 2022;9:859–72. https://doi.org/10.1007/s40684-021-00379-8.

[18] Tang X, Zhou H, Wang F, Wang W, Lin X. Longevity-conscious energy management strategy of fuel cell hybrid electric Vehicle Based on deep reinforcement learning. Energy 2022;238:121593. https://doi.org/10.1016/j.energy.2021.121593.

[19] Zhou J, Liu J, Xue Y, Liao Y. Total travel costs minimization strategy of a dual-stack fuel cell logistics truck enhanced with artificial potential field and deep reinforcement learning. Energy 2022;239:121866. https://doi.org/10.1016/j.energy.2021.121866.

[20] Tiong T, Saad I, Teo KTK, L H. Deep Reinforcement Learning with Robust Deep Deterministic Policy Gradient. 2020 2nd International Conference on Electrical, Control and Instrumentation Engineering (ICECIE), https://doi.org/10.1109/ICECIE50279.2020.9309539.

[21] Zhou J, Xue S, Xue Y, Liao Y, Liu J, Zhao W. A novel energy management strategy of hybrid electric vehicle via an improved TD3 deep reinforcement learning. Energy 2021;224:120118. https://doi.org/10.1016/j.energy.2021.120118.

[22] Deng K, Liu Y, Hai D, Peng H, Löwenstein L, Pischinger S, et al. Deep reinforcement learning based energy management strategy of fuel cell hybrid railway vehicles considering fuel cell aging. Energy Convers Manag 2022;251:115030. https://doi.org/10.1016/j.enconman.2021.115030.

[23] Kuang NL, Leung CHC, Sung VWK. Stochastic Reinforcement Learning. 2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE). https://doi.org/10.1109/AIKE.2018.00055.

[24] Rathor S, Saxena D, Khadkikar V. Electric vehicle trip chain information-based hierarchical stochastic energy management with multiple uncertainties. IEEE Trans Intell Transp Syst 2022;23:18492–501. https://doi.org/10.1109/TITS.2022.3161953.

[25] Hussain A, Bui VH, Musilek P. "Local demand management of charging stations using vehicle-to-vehicle service: A welfare maximization-based soft actor-critic model". eTransportation 2023;18:100280. https://doi.org/10.1016/j.etran.2023.100280.

[26] Williams RJ. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach Learn 1992;8:229–56. https://doi.org/10.1007/BF00992696.

[27] Sharaf OZ, Orhan MF. An overview of fuel cell technology: Fundamentals and applications. Renew Sustain Energy Rev 2014;32:810–53. https://doi.org/10.1016/j.rser.2014.01.012.

[28] M. Moein Jahromi, H. Heidary, Automotive applications of PEM technology, in: PEM Fuel Cells, Elsevier, 2022: pp. 347–405. https://doi.org/10.1016/B978-0-12-823708-3.00009-2.

[29] Wipke KB, Cuddy MR, Burch SD. ADVISOR 2.1: a user-friendly advanced powertrain simulation using a combined backward/forward approach. IEEE Trans Veh Technol 1999;48:1751–61. https://doi.org/10.1109/25.806767.

[30] Grey CP, Hall DS. Prospects for lithium-ion batteries and beyond—a 2030 vision. Nat Commun 2020;11:6279. https://doi.org/10.1038/s41467-020-19991-4.

[31] Onori S, Serrao L, Rizzoni G. Hybrid electric vehicles. London: Springer; 2016. https://doi.org/10.1007/978-1-4471-6781-5.

[32] Sutton RS, Barto AG. Reinforcement Learning: An Introduction. Second edition. MIT Press; 2015.

[33] IBM, "What is deep learming?" [Online]. Available: https://www.ibm.com/topics/deep-learning, [Accessed: January 1, 2023].

[34] MNIH Volodymyr, et al. Asynchronous methods for deep reinforcement learning. *International conference on machine learning.* PMLR 2016.

[35] Greensmith E, Bartlett PL, Baxter J. Variance reduction techniques for gradient estimates in reinforcement learning. J Mach Learn Res 2004;5:1471–530.

[36] Ahmed Z, Le Roux N, Norouzi M, Schuurmans D. Understanding the impact of entropy on policy optimization, *International conference on machine learning.* PMLR 2019.

[37] Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, et al. Playing Atari with Deep Reinforcement Learning 2013.

[38] Anbarasu A, Dinh TQ, Sengupta S. Novel enhancement of energy management in fuel cell hybrid electric vehicle by an advanced dynamic model predictive control. Energy Convers Manage 2022;267:115883. https://doi.org/10.1016/j.enconman.2022.115883.

[39] National Renewable Energy Laboratory, "ADVISOR Documentation", [online]. Available: https://adv-vehicle-sim.sourceforge.net/advisor_doc.html, [Accessed: November 2,2023].

[40] Abu Mallouh M, Surgenor BW, Abdelhafez E, Salah M, Hamdan M. Development of a driving cycle for amman city with performance evaluation for ICE vehicle. Proceedings of the ASME 2014 12th Biennial Conference on Engineering Systems Design and Analysis, Jul 2014. https://doi.org/10.1115/ESDA2014-20600.

[41] Zheng CH, Park YI, Lim WS, Cha SW. Fuel consumption of fuel cell hybrid vehicles considering battery SOC differences. Int J Automot Technol 2012;13:979–85. https://doi.org/10.1007/s12239-012-0100-x.

[42] Lockwood O, Si M. A Review of Uncertainty for Deep Reinforcement Learning. Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment 2022.