Function Approximation Reinforcement Learning of Energy Management with the Fuzzy REINFORCE for Fuel Cell Hybrid Electric Vehicles

Liang GUO[a]*, Zhongliang LI[a,b]*, Rachid OUTBIB[a], Fei GAO[b]

[a]Aix-Marseille University, LIS UMR CNRS 7020, Marseille, France
[b]Université de Franche-Comté, UTBM, CNRS, institut FEMTO-ST, F-90000 Belfort, France[c]UTBM, CNRS, institut FEMTO-ST, F-90000 Belfort, France

ARTICLE INFO

Keywords:
Energy management strategy
Fuel cell hybrid electric vehicle
Reinforcement learning
Fuzzy inference system
Fuzzy policy gradient
Hardware-in-loop

ABSTRACT

In the paper, a novel self-learning energy management strategy (EMS) is proposed for fuel cell hybrid electric vehicles (FCHEV) to achieve the hydrogen saving and maintain the battery operation. In the EMS, it is proposed to approximate the EMS policy function with fuzzy inference system (FIS) and learn the policy parameters through policy gradient reinforcement learning (PGRL). Thus, a so-called Fuzzy REINFORCE algorithm is first proposed and studied for EMS problem in the paper. Fuzzy REINFORCE is a model-free method that the EMS agent can learn itself through interactions with environment, which makes it independent of model accuracy, prior knowledge, and expert experience. Meanwhile, to stabilize the training process, a fuzzy baseline function is adopted to approximate the value function based on FIS without affecting the policy gradient direction. Moreover, the drawbacks of traditional reinforcement learning such as high computation burden, long convergence time, can also be overcome. The effectiveness of the proposed methods were verified by Hardware-in-Loop experiments. The adaptability of the proposed method to the changes of driving conditions and system states is also verified.

1. Introduction

In recent years, under the dual pressure of energy crisis and low-carbon requirements, the traditional fossil fuel automobile industry is facing a severe situation. Hydrogen energy, due to its environment friendly nature, large reserves and diverse production routes, has attracted more and more attention from various countries. Accordingly, fuel cell hybrid electric vehicles (FCHEV) are attracting increasing attention because of their environment friendly nature and competitive vehicle performance [1]. In the control framework of FCHEV, energy management strategy (EMS) is a key element to make the whole powertrain system work more efficiently by adjusting the power distribution among different energy sources.

EMS methods can be generally classified into three categories: Rule based, optimal control based, and learning based methods. Rule-based EMS establishes rules based on the characteristics of the concerned system. Among different rule based EMS methods, fuzzy logic rule based EMS is one of the most attractive due to its practical effectiveness [2]. However, the design of the EMS rules highly rely on on expert experience in most cases [3].

Energy management problems can be regarded as a constrained horizontal sequence optimization problem [4]. Among various resolution methods, dynamic programming (DP) based EMS can provide a numerical global optimal solution [5]. However, the vehicle operating condition and model have to be known in prior to implement DP based EMS which is unrealistic in practice. Stochastic dynamic programming (SDP) [6], Pontryagin's minimum principle (PMP) [7] and equivalent consumption minimization strategy (ECMS) [8] have also been proposed to solve the optimization problem. Using these methods, quasi-optimal solution can be obtained via real-time implementation. However, accurate system models and the information of future driving condition are needed to achieve high performance [9][10].

Reinforcement learning (RL), as a kind of machine learning, is attracting increasing attention in EMS development [11]. When applying RL, the solution of a constrained sequential decision optimization problem can be learned progressively by interacting with the environment without relying on the system model and prior knowledge of driving condition. Thanks to the property, RL-based methods have potential to adapt to environment changes, such as the model degradation of energy sourses and the variations of vehicle driving conditions or drivers' behivor. Q-Learning, as a basic value-based RL method, has been proposed in various hybrid vehicles to solve energy management problems [12][13]. The main drawback of Q-Learning based methods is that it only works for discrete action space and

-
    - * Corresponding authors.
    - E-mail address: liang.guo@lis-lab.fr (L.Guo), zhongliang.li@univ-fcomte.fr (Z. Li)

- Peer review under responsibility of xxxxx. xxxx-xxxx/$ – see front matter © 2023 xxxxxxxx. Hosting by Elsevier B.V. All rights reserved.
- http://dx.doi.org/10.0000/x.xxxx.2022.xx.xxx

state space while EMS needs to be implemented in continuous action and state spaces.

For continuous space problems in reinforcement learning, its solution usually involves a function approximation to approximate value function or/and policy function. The used approximative function can be linear or nonlinear. The linear function approximation is realized through a linear combination of the extracted state features. For instance, polynomial, Fourier basis, corse coding and tile coding are commonly used functions to extract state features [14] [15].

Nonlinear function approximators such as neural networks and Gaussian functions generally have better generalization capabilities and adapt to more scenarios. A Gaussian based non-linear function approximation for RL is proposed in [16], which solves the information loss problem of linear function approximation. Deep neural networks have been extensively applied as a function approximation method for reinfoecement learning. A large number of deep reinforcement learning (DRL) algorithms have been developed and researched rapidly [17]. For EMS problems, DRL such as Deep Q-Networks (DQN) [18] and Deep Deterministic Policy Gradient (DDPG) [19] have been proposed and achieved interesting results [20]. However, the nonlinear function approximation based on deep neural networks has drawbacks, such as high computational complexity, hyperparameter tuning difficulty, hurdling its practical applications [21].

Fuzzy inference system (FIS) is a system that defines input, output and state variables on fuzzy sets. It captures the ambiguity of human brain thinking and can imitate human comprehensive inference to deal with the problems that are difficult to solve by conventional mathematical methods[22]. Similar to neural networks, FIS also has good generalization ability and function approximation ability. The first application of the FIS function approximator in reinforcement learning is fuzzy Q-learning (FQL), in which the state-action value function is approximated using FIS engine[23]. In energy demain, FQL are applied to solve the optial problem of energy management [24][25], to reduce energy loss, improve efficiency and economy. In our previous work, a FQL-based EMS for a fuel cell hybrid electric vehicle is also proposed in [26] to prolong the lifetime of fuel cells. Even low computation requirement has been justified in this work, the inherent drawbacks of Q-Learning, such as value overestimation and large training variance,are not well addressed.

To overcome the drawbacks of FQL, FIS can also be used as the policy function approximator to form a fuzzy policy gradient (FPG), and faster and smoother convergency can often be achieved in policy gradient learning paradigm [27]. Monte-Carlo Policy Gradient, referred as REINFORCE, is the a policy gradient RL method that can be used to solve the continuous state discrete action space. Meanwhile, in the learning process, the agent can explore the action space and avoid getting stuck in a local optimum by making small perturbations near the target action [28].

In the paper, a fuzzy REINFORCE-based EMS method is proposed for FCHEV, which apply FIS to approxiamtare policy function in Monte-Carlo Policy Gradient. Moreover, to suppress of the training variance of REINFORE, a fuzzy basline function [29] is also used in the paper. Thanks to the model-free characteristic of the proposed methods, the EMS controller does not require an accurate system model. The optimal control can be achieved by interacting with the environment . In addition, the learned fuzzy REINFORCE based EMS also demonstrates satisfactory robustness to the unknown external input and the different system states. The main contributions of the paper are thus as follows:

1) Fuzzy REINFORCE, as a model-free learning based EMS, is initially proposed to tackle energy management problems for FCHEVs.
2) Fuzzy baseline function is proposed to suppress the training variance of Fuzzy REINFORE;
3) The adaptability of the propose methods to the variation of driving conditions and system states are justified.
4) The proposed Fuzzy REINFORE has been implemented successfully in a single-chip microcomputer and in real-time, which justifies its low computational requirements.

The paper is organized as follows: Section 2 introduces the modeling of FCHEV energy system, including fuel cells, batteries and vehicle kinetic models. Section 3 describes the principle of the proposed Fuzzy REINFORCE based energy management strategy. Section 4 shows the analysis of the simulation and Hardware-in-Loop experiment results. The paper is finally concluded in Section 5.

2. Modeling of FCHEV Energy System

The studied energy system of the FCHEV is shown in Fig. 1. Its power source consists of a fuel cell (FC) system and a battery system. Due to the slow dynamic response of the FC, the battery is mainly used to absorb the power surplus and provide instantaneous high-power output. They all share the DC bus through the DC/DC converter, which is used to provide the power required by the load or to absorb the load energy.

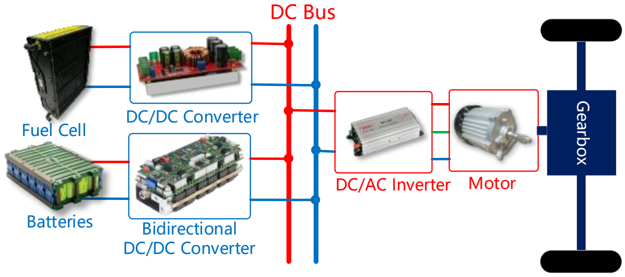


Fig. 1 - Energy system for fuel cell hybrid electric vehicle.

The specific model of each part will be analyzed in the sequel.

2.1. Fuel cells model

Proton exchange membrane fuel cell (PEMFC) is a low-temperature fuel cell, which is an electrochemical device that uses hydrogen as fuel and polymer proton exchange membrane as the conductive medium. The product is only water. The single-cell voltage V_{cell} of the PEMFC stack can be expressed as [30]:

$$\begin{aligned} V_{cell} = E_0 + \frac{\Delta TS}{nF} - \frac{RT}{nF}\ln\left(\frac{P_{H_2O}}{P_{H_2}\sqrt{P_{O_2}}}\right) \\ -\frac{RT}{\alpha F}\ln\left(\frac{i_{fc}+i_{loss}}{i_0}\right)-\frac{RT}{nF}\ln\left(\frac{I_{lim}}{I_{lim}-i_{fc}}\right)-i_{fc}R_{ohm} \end{aligned} \tag{1}$$

where $E_0 = 1.23\,V$ is the open-circuit voltage of fuel cell reaction at standard atmospheric pressure, $R = 8.3145$ is the gas constant, $T = 333.15\,K$ is the fuel cell temperature, $\Delta T = T - 273.15 = 60\text{K}$, $n = 2$, $F = 96485$ is Faraday constant, $\alpha = 1$ is the transfer coefficient, P is the local pressure of the reactants and products. i_{fc} is the current density. $i_{loss} = 2mA/cm^2$ is the current loss, $i_0 = 0.003mA/cm^2$ is the exchange current density. $I_{lim} = 1.6A/cm^2$ is the limiting current density. $R_{ohm} = 0.11\Omega$ is the fuel cell resistance.

For the FC stack, the model is as follows:

$$\begin{aligned} V_{fc} &= n_{cell} \cdot V_{cell} \\ I_{fc} &= A_{fc} \cdot i_{fc} \end{aligned} \tag{2}$$

where n_{cell} is the number of single FCs, and A_{fc} is the active area of the FC electrode plate. Then the hydrogen consumption model of the FC stack can be derived as follows [31]:

$$\dot{m}_{H_2} = M_{H_2}\frac{I_{fc}}{nF} = \frac{M_{H_2}P_{fc}}{nV_{fc}F} \tag{3}$$

where $\dot{m}_{H_2}$ is the rate at which hydrogen is consumed, and M_{H_2} is the molar mass of hydrogen. P_{fc} is the output power of FCs. The converter model will only be concerned about its power characteristics. The efficiency model of the DC/DC converter for FCs is that:

$$P_{fc} = P_{fc}' / \eta_{dc} + P_{aux} \tag{4}$$

where P_{fc}' is the output power of the FC system. It is considered that P_{fc}' is equal to the power command from the control strategy. η_{dc} is the efficiency of DC/DC converter for fuel cells. P_{aux} is the auxiliary system, and it includes the air compressor motor power, water-heat management system power and other auxiliary control systems power. Then the auxiliary system power model is as follows:

$$\begin{aligned} P_{aux} &= P_{air} + I_{aux} \cdot V_{fc} \\ P_{air} &= \frac{2\pi}{60} T_{air} N_{air} \end{aligned} \tag{5}$$

where T_{air} and N_{air} are the torque and speed of the air compressor, respectively. The power of the air compressor is related to the operating power of the fuel cell. When the power of the fuel cell is small, the air compressor does not work. When the power of the fuel cell is large, the required power of the air compressor increases. For the other parts in the auxiliary system are consider as a constant current load I_{aux}.

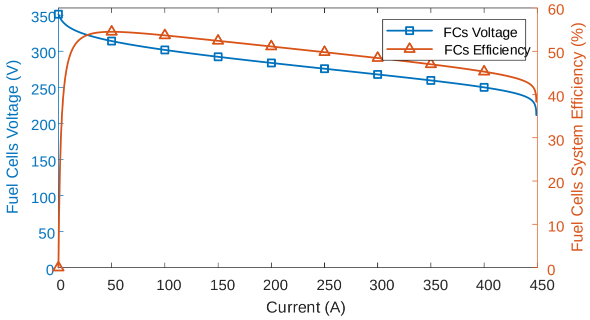


Fig. 2 - The output voltage and efficiency of fuel cells

The model parameters of fuel cell vehicles refer to the Argonne National Lab (ANL) 's test report for Toyota Mirai I [32]. In the model, $n_{cell} = 370$, the effective area of the electrode is $A_{fc} = 282\ cm^2$, the pressure of anode hydrogen is 50 kPa over atmosphere pressure, and cathode oxygen is obtained from the air by air compressor . As shown in Fig. 2 when the current is 439.8 A, the FC power reaches the max power of 117.8 kW and the efficiency is 48.57%; When the current is 63.2 A, the FC efficiency reaches the max efficiency of 55.21%, and the power is 19.9 kW. The parameters of the fuel cell model are fitted to match Mirai I, more details of Mirai I vehicle data can be found in [32].

2.2. Battery model

The battery cell is modeled using a simple one-order circuit model [33]. The model of the battery pack is realized in series and parallel according to the single battery cell model. The open circuit voltage V_{oc} and equivalent resistance R_{bat} of the battery pack can be expressed as:

$$\begin{aligned} V_{oc} &= N_S \cdot V_{oc,cell} \\ R_{bat} &= \frac{N_s}{N_P} \cdot R_{cell} \end{aligned} \tag{6}$$

where $V_{oc,cell}$ is the open circuit voltage of the single cell, R_{cell} is the cell equivalent resistance, N_S is the number of series cells, and N_P is the parallel cells of the battery pack. For the single cell battery, $V_{oc,cell}$ and R_{cell} are dependent on the state of charge (SOC) of the battery according a map functions shown in Fig. 3 (a). The output current I_{bat} of the battery pack and the evolution of the battery state of charge SOC_{bat} are characterized by the following equations.

$$\begin{aligned} I_{bat} &= \frac{V_{oc} - \sqrt{V_{oc}^2 - 4R_{bat}P_{bat}}}{2R_{bat}} \\ SOC_{bat} &= SOC_{bat}(0) - \int_0^t I_{bat} / Q_{bat} dt \end{aligned} \tag{7}$$

When $I_{bat} > 0$, the battery is discharged, and when $I_{bat} < 0$, the battery is charged. Q_{bat} is the battery capacity. Considering the power loss of the battery-side DC/DC converter, the battery output power can be expressed as:

$$P_{bat}' = \begin{cases} P_{bat} \cdot \eta_{discharge} & (P_{bat} > 0) \\ P_{bat} / \eta_{charge} & (P_{bat} < 0) \end{cases} \tag{8}$$

where P_{bat}' is the output power of the bi-directional converter, and its charge efficiency is considered as constant value of $\eta_{charge} = 95\%$ and discharge effiency is similar with $\eta_{discharge} = 94.44\%$. In our application, the capacity of the studied battery is set as 6.6 Ah, and the standard voltage is 244.8V, which is also referred to the parameters of Toyota Mirai I reported by ANL [32].

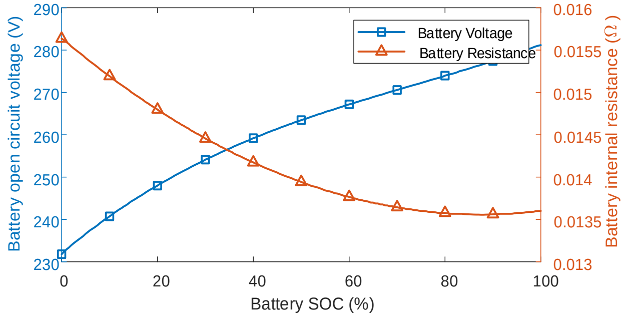


Fig. 3 - The characteristics of the batteries

2.3. Vehicle dynamics model

Supposing a vehicle is moving forward at velocity v on a road with gradient θ, its dynamic function is:

$$\begin{aligned} F_m &= F_{air} + F_f + F_s + F_a \\ &= \frac{1}{2}C_D A\rho v^2 + Gf\cos\theta + G\sin\theta + m\frac{dv}{dt} \end{aligned} \tag{9}$$

where F_m represents the driving force provided by the motor, F_{air} is air resistance, F_f is rolling resistance, F_s denotes slope resistance and F_a represents acceleration resistance. ρ and C_D represent air density and air resistance coefficient respectively. A represents the windward surface volume of the vehicle body, and v represents the vehicle velocity. m represents the vehicle mass. $G = mg$ represents the gravity of the vehicle, and f represents the sliding resistance coefficient.

The required power for the vehicle is:

$$P_{veh} = F_m \cdot v / \eta_m \tag{10}$$

where, P_{veh} represents the required power of the motor, η_m represents the transmission efficiency of the electric machine. According to the power balance, the required power of the motor is provided by the fuel cell and battery:

$$P_{veh} = P'_{fc} + P'_{bat} \tag{11}$$

For the studied vehicle, the paremetes are show in Table 1, and the total mechanical transmission efficiency is set as a contant value 90%. The parameters of the vehicle is designed to match Toyota Mirai I FCHEV reported by ANL [32].

Table 1 – Vehicle parameters

Vehicle	Parameters
Mass	$2500kg$
Windward area	$1.8\ m^2$
Air density	$1.25kg/m^3$
Air resistance coefficient	$0.3Cd$
Rolling friction coefficient	0.01
Gravity acceleration	9.8m/s^2

3. Fuzzy Policy Gradient EMS

3.1. EMS problem formulation

The objective of EMS is to optimize vehicle performance by dispatching instantaneously the demand power among different energy sources. In this work, the objective is to minimize fuel (hydrogen) consumption while maintaining the battery SOC. The objective function is formulated mathematically as the integral of instantaneous reward:

$$\begin{aligned} &\max J = \int_0^T r(t)dt \\ &r(t) = -\dot{m}_{H_2}(t) - k(SOC_{bat}(t) - SOC_{ref})^2 \end{aligned} \tag{12}$$

where $r(t)$ is the instant reward, and it contains two parts. The first part is about the hydrogen consumption rate $\dot{m}_{H_2}(t)$, and the second part is to maintain the SOC of batteries in a safety range to make its long operation, and SOC_{ref} is the reference of SOC corresponding to the battery characteristics. The EMS is dedicated to determining $P_{fc}(t), t \in [0, T]$ using the observables $[P_{veh}(t), SOC_{bat}(t)]$ to achieve the maximum of the objective function J.

3.2. Fuzzy Interrace System for the Energy Management Problem of FCHEV

Fuzzy logic imitates the human brain's uncertain concept judgment and reasoning thinking. A basic FIS consists of four parts: fuzzifier, defuzzifier, inference engine, and knowledge base.

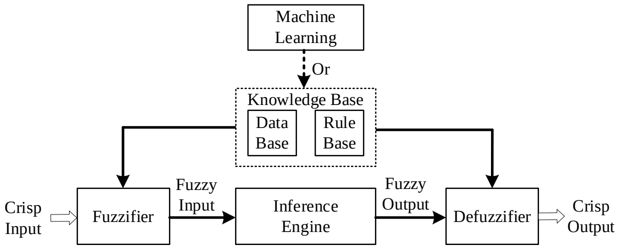


Fig. 4 - Fuzzy Interrace System scheme

The control based on FIS is an effective and widely used method to deal with energy management problems. In our application, the EMS deals with a multi-input single-output FIS control system. As shown in Fig. 4, the crisp input is the system state $\boldsymbol{s} = [P_{veh}, SOC_{bat}]$, and the crisp output is the action of the control system $\boldsymbol{a} = [P_{fc}]$.

Table 2 - Vehicle Required Power Fuzzy Table

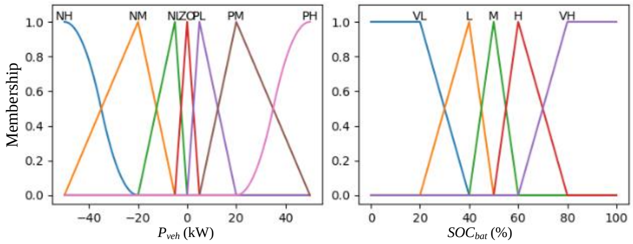
$\boldsymbol{P_{veh}}$ Fuzzy sets	NH	NM	NL	ZO	PL	PM	PH
Typical value (kW)	-50	-20	-10	0	10	20	50

Table 3 - Batteries System Power Fuzzy Table

$\boldsymbol{SOC_{bat}}$ Fuzzy sets	VL	L	M	H	VH
Typical value (%)	20	40	50	60	80

With fuzzifier, the fuzzy state $\boldsymbol{x} = [x_1, x_2, \ldots, x_m]^T$of the energy system can be derived by predefined membership functions of state variables in $\boldsymbol{s}$. The meaning of the fuzzy sets [“NH”, “NM”, “NL”, “ZO”, “PL”, “PM”, “PH”] for P_{veh} are “Negative High”, “Negative Middle”, “Negative Low”, “Zero”, “Positive Low”, “Positive Middle”, and “Positive High”. As shown in Fig.5(a), the membership functions of "NH" and "PH" adopt Z-shape and S-shape, while other membership functions adopt triangular shape, so that the power points that near the boundary values of membership functions -50kW and 50kW will have higher weights. This will help the EMS to be more sensitive to the input state boundary value when making strategic decisions, to avoid actions that are out of bounds and limited. The triangle functions are used due to their simplicity.

For another state SOC_{bat}, the meaning of [“VL”, “L”, “M”, “H”, “VH”] are “Very Low”, “Low”, “Middle”, “High”, and “Very High”. Their membership functions are mainly triangular, but changes are made to the membership functions of "VL" and "VH". As shown in Fig.5(b), the membership functions are mainly triangular, but the membership functions of "VL" and "VH" are changed a little. The weights of these two membership functions are 1 in the state where the SOC is less than 20% or greater than 80%. This is because we set the SOC range of [20%, 80%] to be safe, and for the unsafe range, giving the highest weight 1 for battery safety. Similarly, the triangle functions are used due to their simplicity, and non-equal membership function values were used. The reason why 50% is the center is because the set battery SOC reference value is 50%, and it is located at the midpoint of the SOC physical range [0%, 100%], so higher fuzzy accuracy is required.



(a) Membership of P_{veh} (b) Membership of SOC_{bat}

Fig. 5 - Membership functions of state variables

The membership functions of input states are shown in Fig. 5. Then the two crisp input states can be transformed into fuzzy states $\boldsymbol{x}$ with fuzzy logic operation "AND" of two sets of membership functions. The number of states in $\boldsymbol{x}$ is identical to the number of rules. The membership functions of P_{veh} and SOC_{bat} are 7 and 5 respectively. Hence, the dimensional number of fuzzy states $\boldsymbol{x}$ is $m = 35$.

Traditionally, fuzzy rules can be constructed using experienced data and/or engineering experience. The logic rules are usually formed like:

IF P_{veh} is " Positive High" (PH), **AND** SOC_{bat} is "Very Low"(VL), **THEN** P_{fc} is "Super High" (SH).

The inference engine deduces then the fuzzy output based on each rule. The control action is calculated by a defuzzifier combining all fuzzy outputs. For instance, the calculation can be realized using the weighted average defuzzification method:

$$\boldsymbol{a} = \frac{\sum_{i=1}^{n} u_i \mu_i}{\sum_{i=1}^{n} \mu_i} \tag{13}$$

where μ_i is the i^{th} weight for the output fuzzy action sets, u_i is the i^{th} compoment of the typical value vector $\boldsymbol{U}$, which is related to each fuzzy set of P_{fc}, and $n = 8$ is the number of fuzzy action. Their values in this study are shown in Table 4. Here, a non-equidistant method is used for the division of fuzzy sets, and the state with lower power is divided more finely.

Table 4 - Fuel Cells System Power Fuzzy Table

P_{fc} Fuzzy Sets	SL	SL	VL	L	M	H	VL	SH
Typical value (kW)	0	1	2	5	10	20	50	100

In traditional FIS, the performance of the fuzzy logic controller is limited by the designed rules due to the shortage of data and experience. In this work, policy gradient reinforcement learning is employed to explore the rules automatically. Combining fuzzy logic technology and reinforcement learning will break the empirical performance limits and be able to adapt to model changes such as fuel cell degradation.

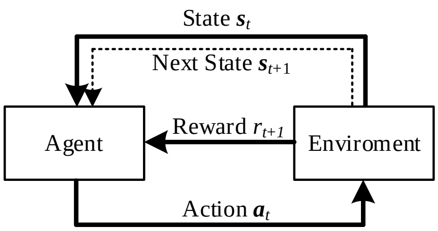


Fig. 6 - Reinforcement Learning Principle

3.3. REINFORCE: Monte-Carlo Policy Gradient

Reinforcement Learning (RL) is a type of machine learning that which the agent takes actions through interacting with the environment for maximum cumulative rewards [34] (see Fig. 6). As a result, a sequence $[\boldsymbol{s}_0, \boldsymbol{a}_0, r_1, \boldsymbol{s}_1, \boldsymbol{a}_1, r_2, \boldsymbol{s}_2 \ldots]$ can be obtained during the learning process. $\boldsymbol{s}_t, \boldsymbol{a}_t$ denote respectively the state, action and reward r_{t+1} denotes the instantaneous reword between instant t and $t+1$.

Policy Gradient (PG) Reinforcement Learning is one of the main class methods of reinforcement learning. In policy gradient, the policy can be parameterized as a stochastic function $\pi_\theta(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{\theta})$ which represents the probability of taking action $\boldsymbol{a}$ in a given state $\boldsymbol{s}$. $\boldsymbol{\theta}$ is the parameter of the policy function. The goal is to adjust $\boldsymbol{\theta}$ to maximize the expected accumulated reward or return denoted by $G(\tau)$, expressed by the formula:

$$J(\pi_\theta) = \underset{\tau\sim\pi_\theta}{\mathrm{E}}\left[G(\tau)\right] \tag{14}$$

where $\tau = [\![0, T-1]\!]$ means a complete time sequence from the initial state to the terminal state in each episode, and $\tau\sim\pi_\theta$ means that we have different time sequence τ due to the policy π_θ is a stochastic policy. In theory, it is necessary to generate different trajectories t through the interaction between the agent and the environment, and then calculate the expected value of the objective function once by calculating the average value of the cumulative rewards of multiple trajectories [35], which will cost a lot of computation.

In the paper, it is propoed to chose Monte-Carlo Policy Gradient (MCPG), also called REINFORCE as the PG method for two reasons: 1) It does not require additional estimation of the value function which facilitates the configuration and training of the algorithm; 2) Monte-Carlo is an unbiased estimation method for $\underset{\tau\sim\pi_\theta}{\mathbb{E}}[G(\tau)]$ [35]. Thus, function $G(\tau)$ can be connected with the discrete form of EMS objective function J by introducing discounted function as:

$$G_0 = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{T-1} r_T = \sum_{i=1}^{T} \gamma^{i-1} r_i \tag{15}$$

where γ is a discount factor that is highly close to 1, and G_0 is the discounted accumulative reward from the initial state of time $t = 0$ under the trajectory τ. Usually, for maximization problems, we can use the gradient ascent algorithm to find the maximum value.

$$\boldsymbol{\theta}^* = \boldsymbol{\theta} + \alpha_\theta \cdot \nabla_\theta J(\pi_\theta) \tag{16}$$

To optimize the parameters step by step, we need to get $\nabla_\theta J(\pi_\theta)$, the gradient of the final reward function $J(\pi_\theta)$ with respect to θ, which is the policy gradient.

$$\begin{aligned} \nabla_\theta J(\pi_\theta) &= \nabla_\theta \underset{\tau\sim\pi_\theta}{\mathrm{E}}\left[G(\tau)\right] = \nabla_\theta \int_\tau P(\tau|\boldsymbol{\theta}) G(\tau) \\ &= \int_\tau P(\tau|\boldsymbol{\theta}) G(\tau) \frac{\nabla_\theta P(\tau|\boldsymbol{\theta})}{P(\tau|\boldsymbol{\theta})} \\ &= \underset{\tau\sim\pi_\theta}{\mathrm{E}}\left[G(\tau) \nabla_\theta \ln P(\tau|\boldsymbol{\theta})\right] \end{aligned} \tag{17}$$

where $P(\tau|\theta)$ is the product of the probabilities at each time step t in a trajectory τ, and the compact expression $\ln P(\tau|\theta)$ is used for the fractional vector $\frac{\nabla_\theta P(\tau|\theta)}{P(\tau|\theta)}$. With further derivation according to the maximum likelihood method, we have the basic theory of policy gradient [14]:

$$\nabla_\theta J(\pi_\theta) = \underset{\tau\sim\pi_\theta}{\mathrm{E}}\left[G(\tau) \sum_{t=0}^{T+1} \nabla_\theta \ln \pi_\theta(\boldsymbol{a}_t | \boldsymbol{s}_t)\right] \tag{18}$$

In this case, $G(\tau)$ represents the discount rewards for the entire trajectory of each episode. For each step here is:

$$G_t = \sum_{i=t+1}^{T} \gamma^{i-1} r_i = r_{t+1} + \gamma G_{t+1} \tag{19}$$

Therefore, to update parameters for each step, (18) can be rewritten as follows:

$$\nabla_\theta J(\pi_\theta) \propto \underset{s,\tau\sim\pi_\theta}{\mathrm{E}}\left[\gamma^t G_t \nabla_\theta \ln \pi_\theta(\boldsymbol{a}_t | \boldsymbol{s}_t)\right] \tag{20}$$

where ∝ means a proportional relationship between two sides. Since the policy gradient only needs to be guaranteed to be in the same direction as the gradient, the scaling factor will be included in α. In this way, policy parameters can be updated step by step as follows according to (16):

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha_\theta \gamma^t G_t \nabla_{\boldsymbol{\theta}} \ln \pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t \mid \boldsymbol{s}_t) \tag{21}$$

where $\boldsymbol{\theta}$ is the update result of the last episode parameter at step 0. The update method is the back-to-front method of the gradient and reward estimation from the current state to the terminal state in each trajectory $t = T-1, T-2, \ldots, 1, 0$.

Traditional REINFORCE is designed for a discrete and stochastic action application. In our case, the algorithm should be constructed with continuous $\boldsymbol{a}$ and $\boldsymbol{s}$.

3.4. *Fuzzy REINFORCE: a Policy Gradient Method with Function Approximation*

In the paper, a novel fuzzy policy gradient method named Fuzzy REINFORCE is proposed. In this method, fuzzy logic is used to construct the policy function. The proposed fuzzy policy function mainly consists of a fuzzifier, linear process, soft-max function and defuzzifier as shown in Fig. 7.

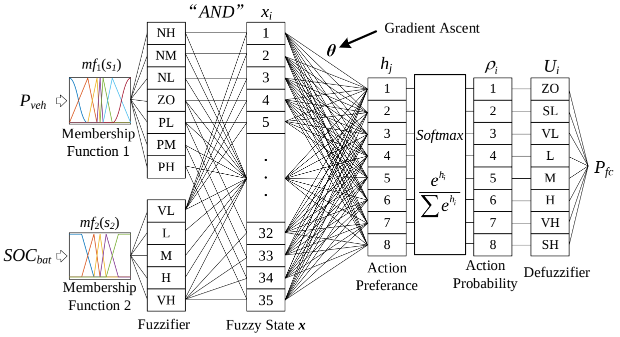


Fig. 7 - The proposed Fuzzy REINFORCE scheme

In the fuzzifier, the state variable $\boldsymbol{s}$ is projected to $\boldsymbol{x}$ as mentioned in Section 3.2. Then, from fuzzy states to action preferences, a linear process is integrated as:

$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^{\bullet}\ \boldsymbol{x} \tag{22}$$

where $\boldsymbol{h} = [h_1, h_2, \ldots h_n]^T$ are action preferences. The dimension of $\boldsymbol{\theta} \in \boldsymbol{R}^{m \times n}$ depends on the fuzzy state $\boldsymbol{x}$ and the action preference $\boldsymbol{h}$ as shown in Fig. 7. The actions with higher preferences in each state also have higher probabilities of being chosen.

To choose an action, the stochastic policy for each typical action vector $\boldsymbol{U} = \{u_1, u_2, \ldots, u_n\}$ is chosen as an exponential soft-max function.

$$\pi_{\boldsymbol{\theta}}(u_i \mid \boldsymbol{x}_t, \boldsymbol{\theta}) = \frac{e^{h_j}}{\sum_{j=1}^{n} e^{h_j}} \tag{23}$$

where $\pi_{\boldsymbol{\theta}}(u_i|\boldsymbol{x}_t, \boldsymbol{\theta})$ is the target policy for u_i, and h_j represents j^{th} anction preference. The stochastic policy we use is an exponential soft-max function. This is because soft-max has a compact form in the derivation involving logarithms, which facilitates the calculation of the inverse gradient in our algorithm. The gradient term in (20) can be deduced as [14]:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}_j} \ln \pi_{\boldsymbol{\theta}}(u_i|\boldsymbol{x}_t, \boldsymbol{\theta}) &= \frac{\nabla_{\boldsymbol{\theta}_j} \pi_{\boldsymbol{\theta}}(u_i \mid \boldsymbol{x}_t, \boldsymbol{\theta})}{\pi_{\boldsymbol{\theta}}(u_i \mid \boldsymbol{x}_t, \boldsymbol{\theta})} = \frac{\boldsymbol{x}_t}{\pi_{\boldsymbol{\theta}}(u_i \mid \boldsymbol{x}_t, \boldsymbol{\theta})} \frac{\partial \pi_{\boldsymbol{\theta}}(u_i \mid \boldsymbol{x}_t, \boldsymbol{\theta})}{\partial h_j} \\
&= \begin{cases} \dfrac{\boldsymbol{x}_t}{\pi_{\boldsymbol{\theta}}(u_i \mid \boldsymbol{x}_t, \boldsymbol{\theta})} \pi_{\boldsymbol{\theta}}(u_i \mid \boldsymbol{x}_t, \boldsymbol{\theta})(1 - \pi_{\boldsymbol{\theta}}(u_i \mid \boldsymbol{x}_t, \boldsymbol{\theta})) & , i = j \\ \dfrac{\boldsymbol{x}_t}{\pi_{\boldsymbol{\theta}}(u_i \mid \boldsymbol{x}_t, \boldsymbol{\theta})} (-\pi_{\boldsymbol{\theta}}(u_i \mid \boldsymbol{x}_t, \boldsymbol{\theta}) \pi_{\boldsymbol{\theta}}(u_j \mid \boldsymbol{x}_t, \boldsymbol{\theta})) & , i \neq j \end{cases} \\
&= \begin{cases} \boldsymbol{x}_t (1 - \pi_{\boldsymbol{\theta}}(u_i \mid \boldsymbol{x}_t, \boldsymbol{\theta})) & , i = j \\ -\boldsymbol{x}_t \pi_{\boldsymbol{\theta}}(u_j \mid \boldsymbol{x}_t, \boldsymbol{\theta}) & , i \neq j \end{cases}
\end{aligned} \tag{24}$$

where $\theta_j \in \boldsymbol{R}^{\boldsymbol{m}}$ is the parameter vector of $\boldsymbol{\theta}$, which is connected to the i^{th} action preference h_j as shown in Fig. 7. In the fuzzy policy gradient RL, the probability for each fuzzy action set can be obtained with policy $\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{x}_t, \boldsymbol{\theta})$ in (23). A common way is to select the action with the greatest probability as the output. However, this approach requires the action space to be discrete. In our application, the action space is continuous. To tackle the issue, the probability corresponding to an action set can also be seen as the weight of the action. Denoting weight vector corresponding to each typical value u_i of the fuzzy action set as $\boldsymbol{\rho} = [\rho_1, \rho_2, \ldots \rho_n]$, the elements can be calculated as:

$$\rho_i = \pi_{\boldsymbol{\theta}}(u_i \mid \boldsymbol{x}_t, \boldsymbol{\theta}), i = 1, 2, \ldots, n \tag{25}$$

For stochastic application with discrete action space, the proposed fuzzy intensification has been done with $\boldsymbol{\rho}$. For the deterministic case of continuous action, the proposed method is to apply a defuzzifier for the probability of fuzzy action sets. Here, we use each action's weight $\boldsymbol{\rho}$ as the membership $\boldsymbol{\mu}$ of the action fuzzy sets in (13). Thus, the weighted average method shown in (13) can be deployed and the action $\boldsymbol{a}_t$ can be derived as:

$$\begin{aligned} \boldsymbol{a}_t &= \sum\nolimits_{i=1}^{n} \rho_i u_i \\ P_{fc} &= \boldsymbol{a}_t + \mathrm{N}_t \end{aligned} \tag{26}$$

where $\mathcal{N}_t$ is a random exploration noise. The exploration rate is set from 10% to 0.01% of $\boldsymbol{a}_t$ in our application during the process. Based on the above analysis, the pseudo-code of the algorithm Fuzzy REINFORCE is summarized in Table 5.

Table 5 - The pseudo-code of the Fuzzy REINFORCE

```
Fuzzy REINFORCE: Fuzzy Monte-Carlo Policy Gradient
Initialize policy parameter $\boldsymbol{\theta} \in \boldsymbol{R}^{\boldsymbol{m \times n}}$ with a random seed
Repeat for each episode:
  Empty the sequence memory $\boldsymbol{M}$
  Reset the environment with $\boldsymbol{s}_0$
  Get fuzzy state $\boldsymbol{x}_t$ from state $\boldsymbol{s}_t$ with Fig.5
  Repeat for each step $t = 0, 1, \ldots, T-2, T-1$:
    Get the preferance $\boldsymbol{h}$ by with $\boldsymbol{\theta}^T \boldsymbol{x}_t$
    Get the fuzzy action weight $\boldsymbol{\rho}$ with softmax($\boldsymbol{h}$)
    Obtain and take action $\boldsymbol{a}_t$ with defuzzifier of $\boldsymbol{\rho}$ and random $\mathcal{N}_t$
    Observe the reward $r_{t+1}$ and next state $\boldsymbol{s}_{t+1}$
    Get fuzzy state $\boldsymbol{x}_{t+1}$ from state $\boldsymbol{s}_{t+1}$ with fig.5
    Add $[\boldsymbol{x}_{t+1}, \boldsymbol{a}_t, r_{t+1}]$ to the sequential memory $\boldsymbol{M}$.
    Update fuzzy state $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_{t+1}$
  Until $\boldsymbol{s}_{t+1}$ is terminal
  Repeat for $t = T-1, T-2, \ldots, 1, 0$:
                    $G_t \leftarrow r_{t+1} + \gamma G_{t+1}$
                $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_\theta \gamma^t G_t \nabla \ln \pi(\boldsymbol{a}_t|\boldsymbol{x}_t, \boldsymbol{\theta})$
Until $G_0$ is convergent.
```

3.5. Fuzzy REINFORCE with the baseline of fuzzy value function

A typical problem with the traditional REINFORCE is that it suffers from high variance during the gradient updates [29]. To reduce variance and stabilize learning, a baseline $b(\boldsymbol{s_t})$ is introduced into the function (20) [14] as:

$$\nabla_{\theta_t} J(\pi_\theta) \propto \underset{s,\tau\sim\pi_\theta}{\mathrm{E}}\left[(G_t - b(\boldsymbol{s}_t))\gamma^t \nabla_{\theta_t} \ln \pi_\theta(\boldsymbol{a}_t \mid \boldsymbol{s}_t)\right] \tag{27}$$

The baseline $b(\boldsymbol{s_t})$ can be designed as a constant value, or a function related to the state $\boldsymbol{s_t}$. The gradient direction of the objective function J is not affected since $\nabla_{\theta} b(\boldsymbol{s_t}) = 0$. Therefore, the parameters $\boldsymbol{\theta}$ can be rewritten as:

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha_\theta [G_t - b(\boldsymbol{s}_t)]\gamma^t \nabla_\theta \ln \pi_\theta(\boldsymbol{a}_t \mid \boldsymbol{s}_t) \tag{28}$$

A value function $V(s_t)$ is proposed to estimate the accumulated reward and used as the baseline function $b(\boldsymbol{s_t})$.

Therefore, the update method of the parameters $\boldsymbol{\theta}$ can be written as:

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha_\theta \gamma^t \nabla_\theta \ln \pi_\theta(\boldsymbol{a}_t \mid \boldsymbol{s}_t)[G_t - V(\boldsymbol{s}_t)] \tag{29}$$

In this work, FIS is also used to be the function approximator of value function for the baseline. Specifically, the fuzzy value function can be constructed using the weighted average method:

$$V(\boldsymbol{s}_t) = \frac{\sum_{i=1}^{M} x_i(\boldsymbol{s}_t) v_i}{\sum_{i=1}^{M} x_i(\boldsymbol{s}_t)} \tag{30}$$

where $v_i \in \boldsymbol{v}$ is the i^{th} fuzzy state value corresponding to the i^{th} fuzzy state $x_i(\boldsymbol{s_t})$. v_i can be updated according to the weight of the fired strength of each fuzzy state:

$$v_i = v_i + \alpha_v (G_t - V(s_t)) \frac{x_i(\boldsymbol{s}_t)}{\sum_{i=1}^{M} x_i(\boldsymbol{s}_t)} \tag{31}$$

where v_i will also be updated as a Monte-Carlo method from time $t = T - 1$ to $t = 0$ in each episode. The pseudo-code of the Fuzzy REINFORCE with fuzzy baseline value function is shown in Table 6.

Table 6 - The pseudo-code of the Fuzzy REINFORCE with baseline

```
Fuzzy REINFORCE with Fuzzy Baseline Function
Initialize policy parameter $\boldsymbol{\theta} \in \boldsymbol{R}^{m\times n}, \boldsymbol{v} \in \boldsymbol{R}^{m}$ with random seeds
Repeat for each episode:
    Empty the sequence memory $\boldsymbol{M}$
    Reset the environment with $\boldsymbol{s_0}$
    Get fuzzy state $\boldsymbol{x_t}$ from state $\boldsymbol{s_t}$ with Fig.5
    Repeat for each step $t = 0,1,\dots,T-2,T-1$:
        Get the preferance $\boldsymbol{h}$ by with $\boldsymbol{\theta}^T\boldsymbol{x_t}$
        Get the fuzzy action weight $\boldsymbol{\rho}$ with softmax$(\boldsymbol{h})$
        Obtain and take action $\boldsymbol{a_t}$ with defuzzifier of $\boldsymbol{\rho}$ and random $\mathcal{N}_t$
        Observe the reward $r_{t+1}$ and next state $\boldsymbol{s_{t+1}}$
        Get fuzzy state $\boldsymbol{x_{t+1}}$ from state $\boldsymbol{s_{t+1}}$ with fig.5
        Add $[\boldsymbol{x}_{t+1}, \boldsymbol{a}_t, r_{t+1}]$ to the sequential memory $\boldsymbol{M}$.
        Update fuzzy state $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_{t+1}$
    Until $\boldsymbol{s}_{t+1}$ is terminal
    Repeat for $t = T-1,\dots,1,0$:
            $G_t \leftarrow r_{t+1} + \gamma G_{t+1}, V(\boldsymbol{s}_t) = \frac{\sum_{i=1}^{M} x_i(\boldsymbol{s}_t)v_i}{\sum_{i=1}^{M} x_i(\boldsymbol{s}_t)}$
            $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_\theta \gamma^t \nabla \ln \pi(\boldsymbol{a}_t|\boldsymbol{x}_t,\boldsymbol{\theta})\left(G_t - V(\boldsymbol{s}_t)\right)$
            $\boldsymbol{v} \leftarrow \boldsymbol{v} + \alpha_v\left(G_t - V(s_t)\right)\frac{\boldsymbol{x}(s_t)}{\sum \boldsymbol{x}(\boldsymbol{s}_t)}$
Until $G_0$ is convergent.
```

4. Simulations and Results Analysis

A Python-based training and testing platform have been established for the proposed Fuzzy REINFORCE based EMS. Moreover, a Hardware-in-Loop (HIL) experimental platform has also been built with dSPACE MicroLabBox, and a microcontroller ESP32. The performance of each calculation unit is as shown in Table 7. Note that, the model of FCHEV is based on Toyota's *Mirai* whose database can be found in [32]. In this section, the results of the proposed EMS are analyzed and discussed.

Table 7 - Vehicle Required Power Fuzzy Table

Platforms	Processors
Host PC	Intel Core i5 9400H @ 2.5GHz
dSPACE MicroLabBox	Dual-core real-time processor @ 2GHz
ESP32	Xtensa dual-core 32-bit LX6 microprocessor @ 240 MHz

4.1. Test driving cycles

The proposed EMS is tested using 2 standard driving cycles *Urban Dynamometer Driving Schedule* (UDDS) and *New European Driving Cycle* (NEDC). The velocity and power of the specific FCHEV under those 2 driving cycle are shown in Fig. 8.

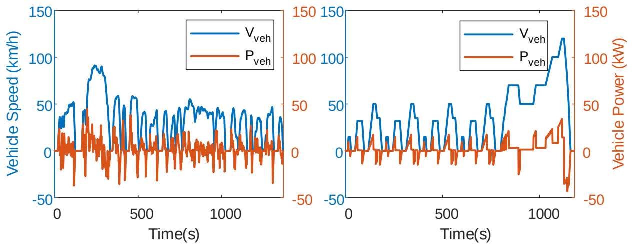


Fig. 8 - Velocity and power of the FCHEV under different driving cycles: (a) "UDDS"; (b) "NEDC"

4.2. Test Results Analysis

The decay rate in (15) is set as $\gamma = 0.999$. The system states and control action are constrained as $P_{veh}(t) \in [-50kW, 50kW]$, $SOC_{bat}(t) \in [0\%, 100\%]$, and $P_{fc}(t) \in [0, 50\ kW]$. The learning curves of the proposed Fuzzy REINFORCE are shown in Fig. 9.

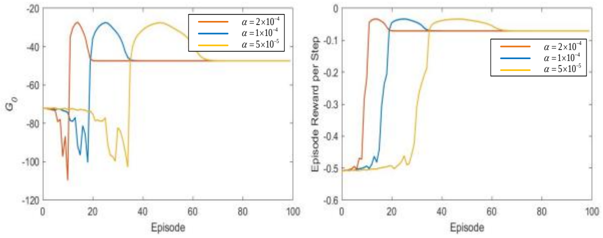


Fig. 9 - Training Process of the proposed Fuzzy REINFORCE based EMS: (a) G_0; (b) average reward

As shown in Fig. 9, G_0 is the main criterion for evaluating the reinforcement learning training process, which represents the discounted cumulative reward from the initial state to the terminal state. The second figure is about the average reward for each episode. The tests were conducted with different learning rates α, and the results show that $\alpha_\theta = 0.0002$ has a faster convergence rate and can achieve the same training results. In addition, the training process of the proposed fuzzy REINFORCE with fuzzy baseline function is shown in Fig. 10. The learning rate parameters are chosen as: $\alpha_\theta = 0.0002$, $\alpha_v = 0.002$. With baseline, G_0 value and average reward at stablised region are both higher than those without baseline setting. With fuzzy baseline function, a better policy optimization path can be found and local optimum can be avoided to a great extent.

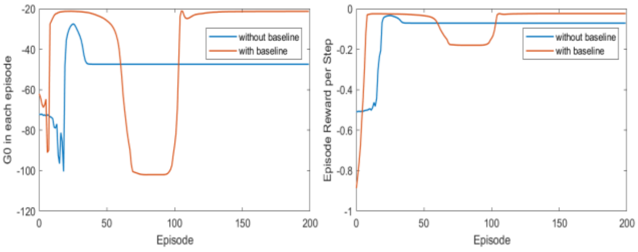


Fig. 10 - Training Process of the proposed Fuzzy REINFORCE with baseline: (a) G_0; (b) average reward

Here we compare the training time and convergency episode of 4 different RL algorithms as shown in Table 8. For the Q-learning and Fuzzy Q-learning from our previous works in [13] and [26], the training environments of them are same with FCHEV model and driving conditions in this paper. Due to the difference in algorithm principles, they have different levels of demand for computation.

Table 8 The training time and convergency of 4 RL algorithms

RL Agent	Training Time	Convergency Episode	Computation Size
Q-Learning [13]	4 hours	95000	$[10201 \times 101]$
Fuzzy Q-Learning [26]	15 minutes	450	$[35 \times 8]$
Fuzzy REINFORCE	4 minutes	20	$[35 \times 8]$
Fuzzy RIENFORCE with Baseline	10 minutes	120	$[35 \times 8]$

For the computation of those RL moethods, Q-Learning needs to store the learned experience to a Q-table whosesize in our application is $[10201 \times 101]$. It means that all states and actions are discretized to an accuracy of 1% of the feasible ranges. In Q-Learning, only one item in the table is updated in one learning step. For the Fuzzy Q-Learning, a $[35 \times 8]$ fuzzy Q-table is stored and 8 items in the table are updated in one learning step. For the 2 proposed Fuzzy REINFORCE methods, $[35 \times 8]$ parameter matrix $\boldsymbol{\theta}$ is stored and all elements are updated in each learning step. Compared to traditional reinforcement learning and fuzzy Q-learnig, the proposed 2 Fuzzy REINFORCE methods significantly reduces the convergence time of training. Fast training and less computational resource ensure its possibility as a real-time online learning algorithm. Moreover, Fuzzy REINFORCE with baseline, achieves better performance while maintaining satisfactory learning speed and light data storage space.

After the training process, different tests with 3 different initial SOC are carried out, which aims to validate the adaptability of the proposed methods to initial state changes. The vehicle is tested under the driving cycles of "UDDS" with different initial state $SOC_{bat} = 25\%, 50\%, 75\%$. Fig. 11 shows the cumulative loss during the test process, and a higher loss means a lower reward for the agent. When there is no baseline, the loss increases significantly. Especially when the initial value is 75%, the loss is much higher than that with baseline. This situation in reinforcement learning generally means that the region is less explored. In Fig. 12, it shows the SOC trajectories of the proposed Fuzzy REINFORCEE based EMS. To ensure the continuous cycle operation of the battery, the SOC is always required to be within a certain range according to the actual situation of the vehicle. In the paper, the SOC trajectory is set to be close to 50%. It is shown that the SOC trajectory with a baseline can move closer to the reference SOC value more quickly.

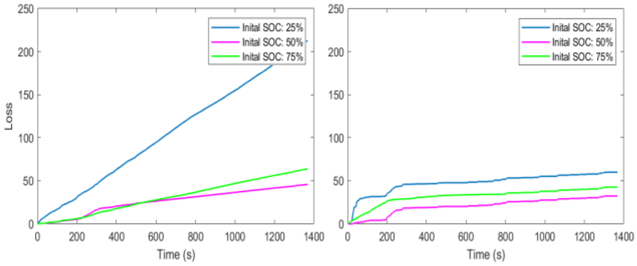


Fig. 11 - Loss of the proposed Fuzzy REINFORCEE based EMS under the diving condition "UDDS": (a) without baseline; (b) with baseline

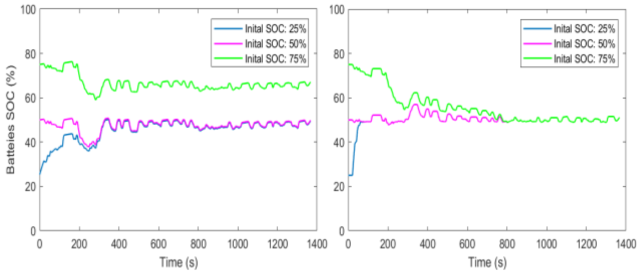


Fig. 12 - The SOC trajectories of the proposed Fuzzy REINFORCEE based EMS under the diving condition "UDDS": (a) without baseline; (b) with baseline

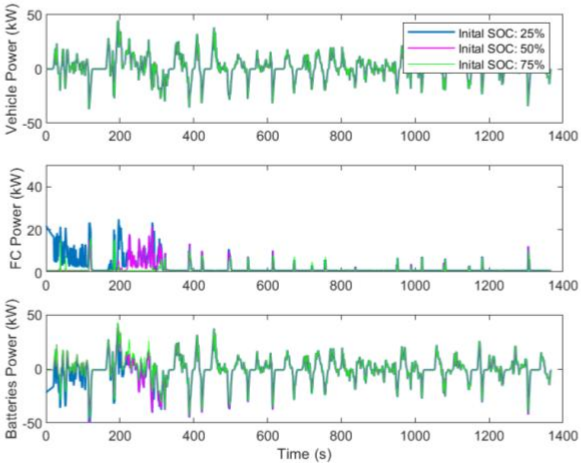


Fig. 13 - The power allocation of the proposed Fuzzy REINFORCE based EMS under the driving cycle "UDDS" (simulation test)

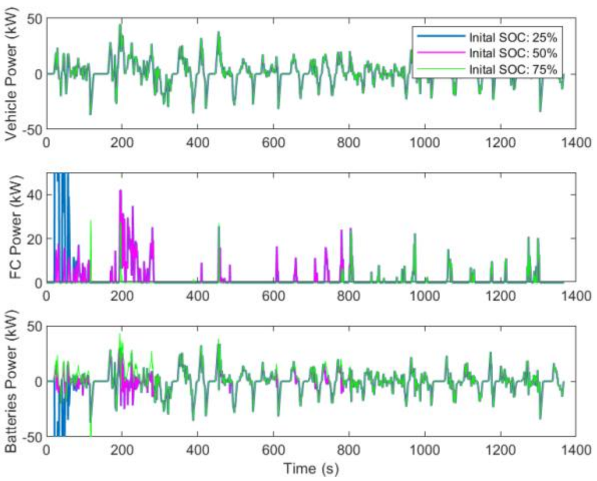


Fig. 14 - The power allocation of the proposed Fuzzy FEINFORCE with baseline based EMS under the driving cycle "UDDS" (simulation test)

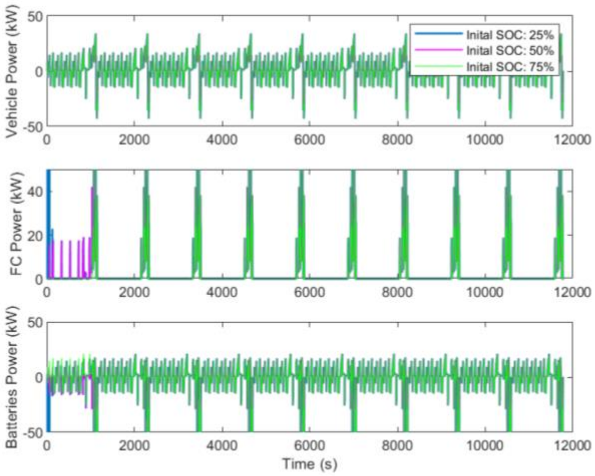


Fig. 16 - The power allocation of the proposed Fuzzy REINFORCE with baseline based EMS under 10 driving cycles of "NEDC" (simulation test)

In addition, Fig. 13 and Fig. 14 show the power allocation of the proposed 2 methods under the driving condition of "UDDS" with 3 different initial SOC. Fig. 15 and Fig. 16 illustrate the power allocation of the proposed EMSs under 10 driving cycles of "NEDC", which is unknown to the EMS agents trained using "UDDS". In contrast, behaviors of the one with baseline function are more conservative, which is more beneficial in actual vehicle operation. The test results show that even in different initial states and unknown external input conditions, the proposed method still has good adaptability to those environment changes, to realizes power allocation strategies between different energy sources, and thus improves the overall work efficiency.

The battery SOC trajectories of 10 driving cycles of "UDDS" and "NEDC" for the FCHEV are shown in Fig. 17 and Fig. 18. The test results show that the SOC trajectory is well maintained at around 50% under different driving conditions. For the proposed Fuzzy REINFORCE without baseline function, the final SOC of batteries is 49.68% and 53.72% under the driving cycle "UDDS" and "NEDC". After 10 driving cycles, with the same initial state SOC set as 50%, the total hydrogen consumption of "UDDS" is 416.5g, and the hydrogen consumption of "NEDC" is 397.4g, which is at the level near the optimal references provided using DP ("UDDS:"362.4g, "NEDC": 344.7g). For the proposed Fuzzy REINFORCE with baseline function, the final SOC of batteries is 50.90% and 67.12% under the driving cycle "UDDS" and "NEDC". After 10 driving cycles, with the same initial state SOC set as 50%, the total hydrogen consumption of "UDDS" is 307.53, and the hydrogen consumption of "NEDC" is 316.5g, which are better than the EMS without baseline. The detailed results are summarized in Table 9 and Table 10. The hydrogen consumption performance of the two proposed methods in the two test driving cycles is shown in Fig. 19.

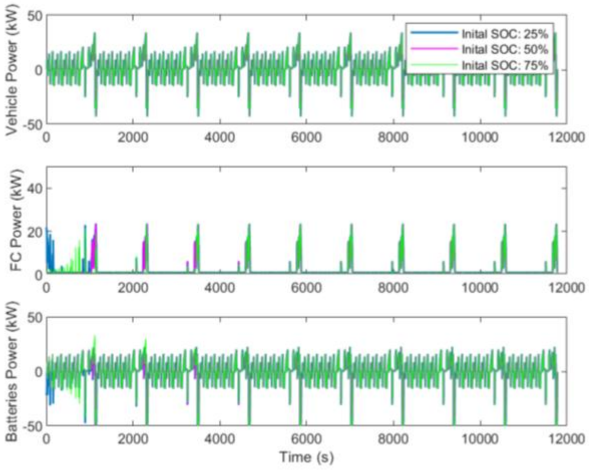


Fig. 15 - The power allocation of the proposed Fuzzy REINFORCE based EMS under 10 driving cycles of "NEDC" (simulation test)

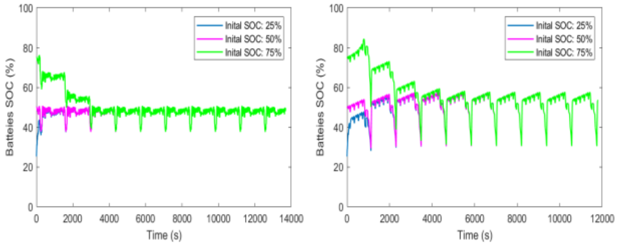


Fig. 17 - The SOC trajectories of the proposed Fuzzy REINFORCE based EMS under 10 driving cycles: (a) UDDS; (b) NEDC

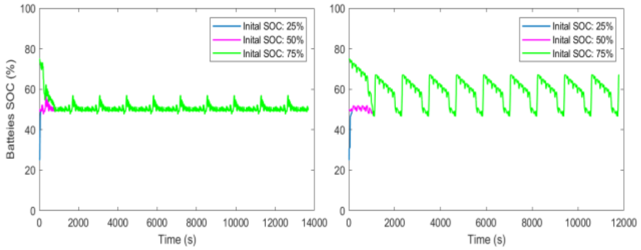


Fig. 18 - The SOC trajectories of the proposed Fuzzy REINFORCE with baseline based EMS under 10 driving cycles: (a) UDDS; (b) NEDC

Table 9 - Test Results after 10 Driving Cycles Time (without Baseline)

Driving Cycle	Initial SOC	Average reward	H2 Consumed (g)	Fuel Rate (g/100km)	Final SOC
	25%	-0.1585	436.4	364.0	49.68%
UDDS	**50%**	**-0.0336**	**416.5**	**347.4**	**49.68%**
	75%	-0.1912	396.2	327.1	49.68%
	25%	-0.2169	417.4	381.9	53.71%
NEDC	**50%**	**-0.0422**	**397.4**	**363.6**	**53.71%**
	75%	-0.1353	377.9	345.8	53.71%

Table 10 Test Results after 10 Driving Cycles Time (with Baseline)

Driving Cycle	Initial SOC	Average reward	H2 Consumed (g)	Fuel Rate (g/100km)	Final SOC
	25%	-0.0252	330.4	275.56	50.90%
UDDS	**50%**	**-0.0232**	**307.5**	**256.46**	**50.90%**
	75%	-0.024	286.5	238.95	50.90%
	25%	-0.0557	339.7	310.80	67.12%
NEDC	**50%**	**-0.0532**	**316.5**	**289.57**	**67.12%**
	75%	-0.0587	295.4	270.27	67.12%

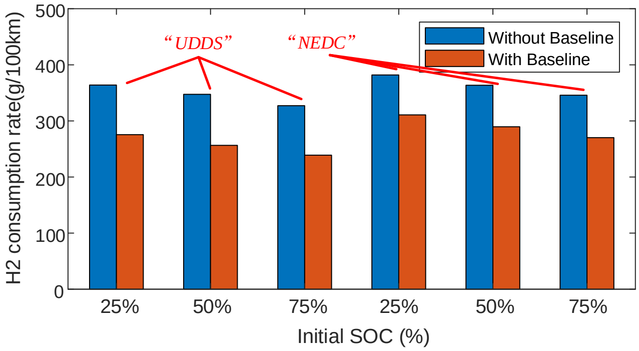


Fig. 19 – H2 consumption per 100km of the FCHEV with the proposed 2 Fuzzy REINFORCE methos after 10 driving cycles

4.3. Experiment results:

The experiment platform dedicated to Hardware-in-Loop (HIL) tests is formed as shown in Fig. 20. During the HIL experiment test, The controlled object is virtual, and its model is executed in the real time in dSPAECE MicroLabBox. As for the controller, ESP32microcontroller is used to deploy the EMS program. FCHEV simulator interacts with the microcontroller through the analog inputs and outputs offered by RTI and HIL libraries after they are downloaded to dSPACE. The information of states P_{veh} , SOC_{bat} and $\dot{m}_{H_2}$ are transferred to ESP32. Then ESP32 feedbacks with the power command for the fuel cells system output power according to the system states. The command is sent through its digital-to-analog conversion (DAC) block to FCHEV simulator. The algorithm deployment of Fuzzy REINFORCE is entirely realized on ESP32. All test data are monitored and saved in the designed user interface with ControlDesk and Matlab/Simulink. The data exchange of dSPACE is through Ethernet, while the data exchange of ESP32 is achieved through Wi-Fi or serial port.

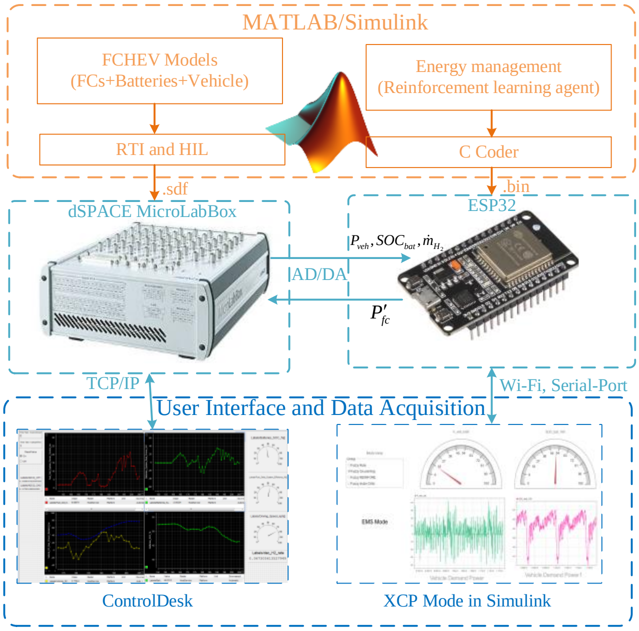


Fig. 20 – Hardware-in-Loop Experiment platform

As shown in Fig. 21 and Fig. 22, the effective of the power allocation of the proposed Fuzzy REINFORCE with baseline is validated under 1 UDDS driving cycle and 10 NEDC driving cycles. Compared to simulation results, the strategy in the HIL experiment tends to output larger fuel cell system power reference, while the overall strategy trends of the two are basically the same.

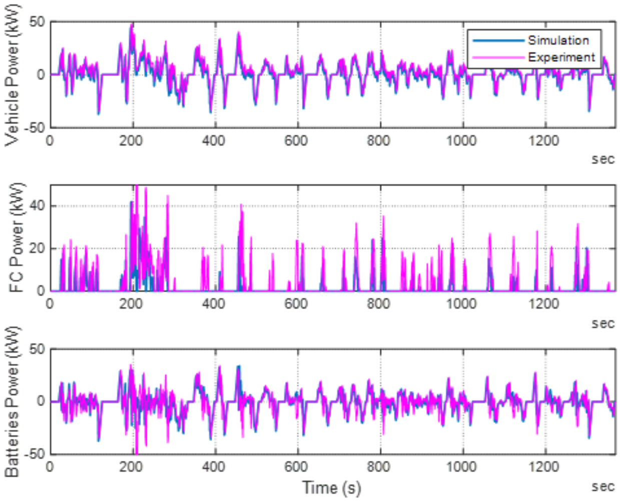


Fig. 21 - The power allocation of the proposed Fuzzy FEINFORCE with baseline based EMS under the driving cycle "UDDS" (experimental test)

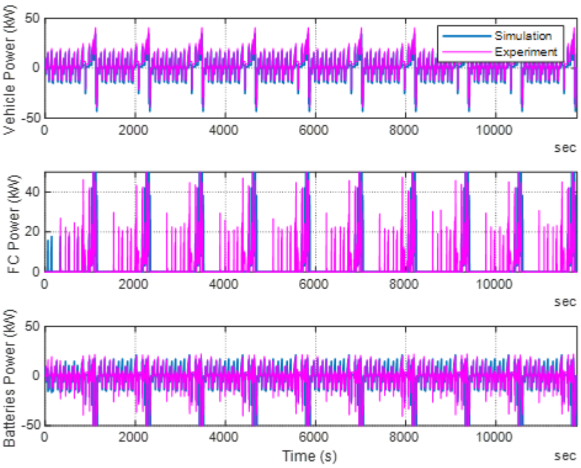


Fig. 22 - The power allocation of the proposed Fuzzy REINFORCE with baseline based EMS under 10 driving cycles of "NEDC" (experimental test)

Compared to simulation results, the strategy in the HIL experiment tends to output larger fuel cell system power reference, while the overall strategy trends of the two are basically the same. Fig. 23 and Fig. 24 show the SOC trajectories with proposed Fuzzy FEINFORCE with baseline under 1 UDDS driving cycle and 10 NEDC driving cycles. The trajectory of the SOC in HIL is more constrained compared with the simulation results. In Fig. 23, the average value of SOC under "UDDS" is 49.75%, which is close to the preset 50% SOC reference value. The standard deviation of the SOC is reduced from 1.62% to 1.20% compared to simulations. In Fig. 24, the average SOC is 51.25%, while the standard deviation of the SOC is reduced from 6.47% to 3.64%.

The difference between the simulation and experimental results mainly comes from the deviation of numerical calculation and the sampling error from AD/DA. As there is a trend towards using fuel cells rather than batteries, increased hydrogen consumption and a more stable SOC trajectory are observed.

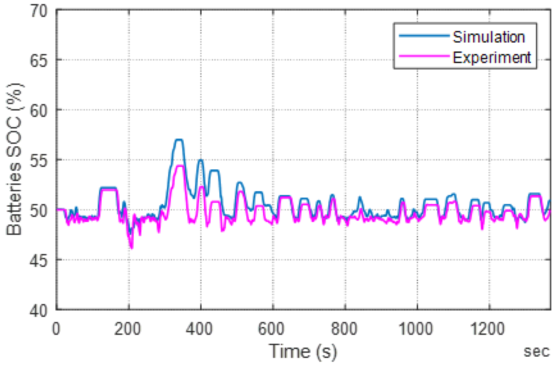


Fig. 23 - The SOC trajectories of the proposed Fuzzy REINFORCE based EMS under 1 driving cycles of UDDS (experimental test)

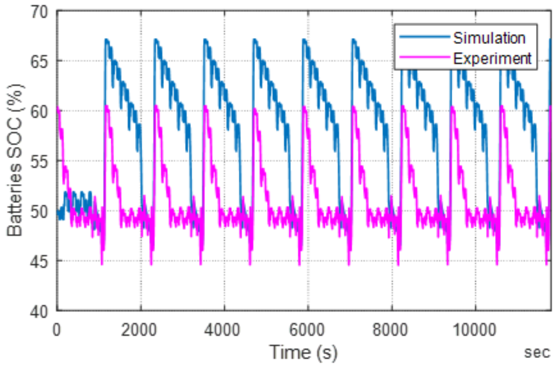


Fig. 24 - The SOC trajectories of the proposed Fuzzy REINFORCE based EMS under 10 driving cycles of NEDC (experimental test)

5. Conclusions

In the paper, Fuzzy FEINFORCE based energy management strategies are studied for fuel cell hybrid electric vehicles. The proposed Fuzzy REINFORCE methods utilize fuzzy inference system to approximate the policy function and the policy parameters are updated with Monte-Carlo method. Moreover, a fuzzy baseline function is proposed to achieve more stable convergency. Since the proposed methods are model-free reinforcement learning, the well-trained EMS agent can obtain near-optimal results without accurately modelling, without highly relying on experience or prior knowledge.

The simulation and HIL experiments test results show that the proposed Fuzzy REINFORCE has fast and smooth convergence and can self-adapt to environment changes such as the initial state change and unknown driving condition. The originally proposed fuzzy baseline function makes the training convergemore stable and faster, while better performance, in terms of hydrogen consumption reduction and SOC preservation, is also achieved. Faster convergence and less computation also make the proposed methods suitable for online self-learning. The implementation of the proposed methods in a microcontroller has justified their online applications.

Acknowledgement

This work has been supported by the ANR DEAL (contract ANR-20-CE05-0016-01). This work has also been partially funded by Region Sud Provence-Alpes-Côte d'Azur via project AMULTI (2021_02918).

REFERENCES

[1] Sorlei I, Bizon N, Thounthong P, et al. Fuel Cell Electric Vehicles — A Brief Review of Current Topologies and Energy Management Strategies. Energies. 2021;14(1):252-280.

[2] Phan D, Bab-Hadiashar A, Fayyazi M, Hoseinnezhad R, Jazar RN, Khayyam H. Interval Type 2 Fuzzy Logic Control for Energy Management of Hybrid Electric Autonomous Vehicles. *IEEE Trans Intell Veh*. 2021;6(2):210-220. doi:10.1109/TIV.2020.3011954.

[3] Liu T, Tan W, Tang X, Zhang J, Xing Y, Cao D. Driving conditions-driven energy management strategies for hybrid electric vehicles : A

review. *Renew Sustain Energy Rev*. 2021;151(July):111521. doi:10.1016/j.rser.2021.11152.

[4] Zhou Y, Ravey A, Péra MC. Multi-mode predictive energy management for fuel cell hybrid electric vehicles using Markov driving pattern recognizer. *Appl Energy*. 2020;258:114057. doi:10.1016/j.apenergy.2019.114057.

[5] Sundström O, Guzzella L, Soltic P. Optimal Hybridization in Two Parallel Hybrid Electric Vehicles using Dynamic Programming. *IFAC Proc Vol*. 2008;41(2):4642-4647. doi:10.3182/20080706-5-kr-1001.00781.

[6] Liu B, Li L, Wang X, Cheng S. Hybrid Electric Vehicle Downshifting Strategy Based on Stochastic Dynamic Programming during Regenerative Braking Process. *IEEE Trans Veh Technol*. 2018;67(6):4716-4727. doi:10.1109/TVT.2018.2815518.

[7] Nguyen BH, German R, Trovao JPF, Bouscayrol A. Real-time energy management of battery/supercapacitor electric vehicles based on an adaptation of pontryagin's minimum principle. *IEEE Trans Veh Technol*. 2019;68(1):203-212. doi:10.1109/TVT.2018.2881057.

[8] Zhang W, Li J, Xu L, Ouyang M. Optimization for a fuel cell/battery/capacity tram with equivalent consumption minimization strategy. *Energy Convers Manag*. 2017;134:59-69. doi:10.1016/j.enconman.2016.11.007.

[9] Kermani S, Delprat S, Guerra TM, Trigui R, Jeanneret B. Predictive energy management for hybrid vehicle. *Control Eng Pract*. 2012;20(4):408-420. doi:10.1016/j.conengprac.2011.12.001.

[10] Zhang S, Xiong R, Sun F. Model predictive control for power management in a plug-in hybrid electric vehicle with a hybrid energy storage system. *Appl Energy*. 2017;185:1654-1662. doi:10.1016/j.apenergy.2015.12.035.

[11] Silver D, Huang A, Maddison CJ, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*. 2016;529(7587):484-489. doi:10.1038/nature16961.

[12] Watkins CJCH, Dayan P. Q-Learning. *Mach Learn*. 1992;292:279-292.

[13] Guo L, Li Z, Outbib R. Reinforcement Learning based Energy Management for Fuel Cell Hybrid Electric Vehicles. *IECON 2021– 47th Annu Conf IEEE Ind Electron Soc*. Published online 2021:1-6.

[14] Sutton RS, Barto AG. *Reinforcement Learning: An Introduction Second Edition*. Vol 9.; 2018.

[15] LA P, Bhatnagar S. Reinforcement Learning With Function Approximation for Traffic Signal Control. *IEEE Trans Intell Transp Syst*. 2011;12(2):412-421. doi:10.1109/TITS.2010.2091408.

[16] Haider A, Hawe G, Wang H, Scotney B. Gaussian Based Non-linear Function Approximation for Reinforcement Learning. *SN Comput Sci*. 2021;2(3):1-12. doi:10.1007/s42979-021-00642-4.

[17] Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning. Published online December 19, 2013. http://arxiv.org/abs/1312.5602.

[18] Wu P, Partridge J, Anderlini E, Liu Y, Bucknall R. Near-optimal energy management for plug-in hybrid fuel cell and battery propulsion using deep reinforcement learning. *Int J Hydrogen Energy*. 2021;(xxxx). doi:10.1016/j.ijhydene.2021.09.196.

[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. doi:10.1016/j.energy.2021.121866.

[20] Du G, Zou Y, Zhang X, Liu T, Wu J, He D. Deep reinforcement learning based energy management for a hybrid electric vehicle. *Energy*. 2020;201:117591. doi:10.1016/j.energy.2020.117591.

[21] Liessner R, Schmitt J, Dietermann A, Bäker B. Hyperparameter optimization for deep reinforcement learning in vehicle energy management. *ICAART 2019 - Proc 11th Int Conf Agents Artif Intell*. 2019;2(Icaart):134-144. doi:10.5220/0007364701340144.

[22] Mitiku T, Manshahia MS. Neuro Fuzzy Inference Approach : A Survey. 2018;4(April):505-519.

[23] Glorennec PY, Jouffe L. Fuzzy Q-learning. *IEEE Int Conf Fuzzy Syst*. 1997;2(3):659-662. doi:10.1109/fuzzy.1997.622790.

[24] Kofinas P, Dounis AI, Vouros GA. Fuzzy Q-Learning for multi-agent decentralized energy management in microgrids. *Appl Energy*. 2018;219(December 2017):53-67. doi:10.1016/j.apenergy.2018.03.017.

[25] Bo L, Han L, Xiang C, Liu H, Ma T. A Q-learning fuzzy inference system based online energy management strategy for off-road hybrid electric vehicles. *Energy*. 2022;252:123976. doi:10.1016/j.energy.2022.123976.

[26] Guo L, Li Z, Outbib R. A Lifetime Extended Energy Management Strategy for Fuel Cell Hybrid Electric Vehicles via Self-Learning Fuzzy Reinforcement Learning. *2022 10th Int Conf Syst Control*. Published online 2022:161-167.

[27] Wang XN, Xu X, He HG. Policy gradient fuzzy reinforcement learning. *Proc 2004 Int Conf Mach Learn Cybern*. 2004;2(August):992-995. doi:10.1109/icmlc.2004.1382332.

[28] Liu EZ, Raghunathan A, Liang P, Finn C. Decoupling Exploration and Exploitation for Meta-Reinforcement Learning without Sacrifices. *Int Conf Mach Learn PMLR*. Published online 2020. http://arxiv.org/abs/2008.02790.

[29] Surita G, Lemos A, Gomide F. *Fuzzy Baselines to Stabilize Policy Gradient Reinforcement Learning*. Vol 258. Springer International Publishing; 2022. doi:10.1007/978-3-030-82099-2_39.

[30] Barbir F. *PEM Fuel Cells: Theory and Practice Second Edition*. Academic Press; 2012. doi:10.1016/B978-0-12-078142-3.X5000-9.

[31] Zhang R, Tao J, Zhou H. Fuzzy optimal energy management for fuel cell and supercapacitor systems using neural network based driving pattern recognition. *IEEE Trans Fuzzy Syst*. 2019;27(1):45-57. doi:10.1109/TFUZZ.2018.2856086.

[32] Lohse-Busch H, Stutenberg K, Duoba M, Iliev S. *Technology Assessment of a Fuel Cell Vehicle: 2017 Toyota Mirai*.; 2018.

[33] Onori S, Serrao L, Rizzoni G. *Hybrid Electric Vehicles: Energy Management Strategies*.; 2016. doi:10.1007/978-1-4471-6781-5.

[34] Nian R, Liu J, Huang B. A review On reinforcement learning: Introduction and applications in industrial process control. *Comput Chem Eng*. 2020;139:106886. doi:10.1016/j.compchemeng.2020.106886.

[35] Willia RJ. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Mach Learn*. 1992;8(3):229-256. doi:10.1023/A:1022672621406.