An Unbiased Fuzzy Double Q-Learning based Energy Management for Fuel Cell Hybrid Electric Vehicles

Liang GUO
Laboratory LIS (UMR CNRS 7020)
Aix-Marseille University
Marseille, France
liang.guo@lis-lab.fr

Zhongliang LI
CNRS, institut FEMTO-ST
Université de Franche-Comté
Belfort, France
zhongliang.li@univ-fcomte.fr

Rachid OUTBIB
Laboratory LIS (UMR CNRS 7020)
Aix-Marseille University
Marseille, France
rachid.outbib@lis-lab.fr

***Abstract*—In the paper, a fuzzy double Q-learning (FDQL) based energy management strategy is proposed for fuel cell hybrid electric vehicles (FCHEV). Model-free characteristic of the proposed novel reinforcement learning enable the agent to enhance performance through environment interactions without relying on specific models. To enable the continuous space application for the studied energy system, fuzzy logic is involved to approximate the state-action value function of conventional Q-Learning. Moreover, the introduction of dual estimators solves its inherent overestimation problem. With python-based environment, low computation and fast convergence of the proposed FDQL are reflected in the training process. Also, adaptability to the changes in driving conditions and initial states are verified in the tests. Finally, the goal of reducing hydrogen fuel consumption and maintaining battery operation of FCHEV are both achieved.**

Keywords— reinforcement learning, energy management strategy, function approximation, fuzzy logic, double Q-learning.

I. Introduction

The energy crisis and environmental pollution have brought enormous pressure and challenges to the traditional internal combustion engine vehicles, and the development of renewable energy vehicles is gradually becoming the main direction of the world automobile industry. Fuel cell hybrid electric vehicles (FCHEV) are attracting increasing attention because of their highly efficient, green, pollution-free, fast charging (hydrogen), and competitive vehicle performance [1]. FCHEV usually consists of fuel cells and batteries to form an energy system. Energy management strategies (EMS) usually plays a very important role in FCHEV. It is used for power distribution among different energy sources to make the system work more efficiently and reliably.

However, the difficulty of modeling fuel cell systems, the time-varying model, and the uncertainty of operation conditions bring great challenges to the design of EMS. Scholars have proposed various EMS methods for different applications, and all those methods can be classified into three categories: rule-based, optimization control based, and learning based methods. Rule-based EMS is highly relayed on historical data and experience makes it difficult to obtain optimal solutions [2]. Among optimization based strategy, dynamic programming (DP) based EMS is a global optimal solution [3]. However, it can not be used for real-time control due to its backward calculation chrematistic. Stochastic dynamic programming (SDP) [4], Pontryagin's minimum principle (PMP) [5] and equivalent consumption minimization strategy (ECMS) [6] are applied to solve the real-time optimization problem. However, high-precision models and prior knowledge are required, and it is difficult to exploit their performance in complex models and time-varying systems. [7]

Reinforcement learning (RL), as a kind of machine learning, is attracting increasing attention in EMS development [8]. The process of the reinforcement learning agent making decisions is a Markov decision process (MDP). The most discriminant property of RL is that the control policy can be learned by interacting with the environment, and without relying on the system model and prior knowledge of driving conditions. Thanks to the property, RL-based methods have the potential to adapt to environment changes, such as the degradation of vehicle components and the changes in vehicle driving conditions.

Q-Learning is a typical model-free algorithm, which was proposed by Watkins [9]. It is a milestone in the development of reinforcement learning and is currently the most widely used reinforcement learning. Q-learning has the advantage of environment model independence, fewer parameters requirement, off-policy with high training efficiency, and convergence guarantee. However, the application of Q-Learning is limited by three drawbacks:

- Overestimation of value due to maximization policy.
- Impractical when the state-action space is very large.
- Only works for discrete action and state space.

To address the overestimation problem, Hasselt proposed double Q-learning on NIPS in 2010 [10]. It aims to replace the overestimation problem of Q-learning with underestimation by introducing double-estimator for the state-action value.

Fuzzy logic imitates the human brain's uncertainty concept judgment, and reasoning way of thinking, which is believed to enable function approximation with good generalization ability. Combined with fuzzy logic to approximate Q-function, fuzzy Q-learning (FQL) is proposed in [11], which is the first fuzzy reinforcement learning (FRL). In FRL, fuzzy inference system (FIS) is utilized to approximate the value function of state or action, which solves the discrete space problem. A FQL-based EMS has been developed for an off-hybrid electric vehicle.[12]. And a fuzzy rule value reinforcement learning (FRVRL) based EMS is proposed for fuel cell hybrid electric vehicles [13], which directly learns the value of each rule through reinforcement learning. However, the problem of overestimation of Q-Learning has not been solved in those fuzzy reinforcement learning methods.

In the paper, a novel fuzzy double Q-learning (FDQL) based EMS is proposed for FCHEV, which approximates the state-action value function based on fuzzy logic to reduce computation and solve continuous space problem. It is also based on double-learning to reduce the overestimation effect of Q-learning. Meanwhile, this model-free self-reinforcing learning can break the policy performance limit of experience-based design and overcome the difficulty modeling and time-varying problems of complex model. Therefore, a self-learning FDQL-based EMS will be studied in this paper, its principle will be elaborated and analyzed, and its performance on FCHEV will be tested by changing the initial state of the environment and different driving conditions.

II. System Modeling

The energy system of FCHEV consists of a fuel cells system, lithium batteries system and motor system. The fuel cell system and the battery system are connected to the DC bus through DC/DC converters to supply power to the load motor system or recover braking or deceleration energy from the motor system. The detailed scheme of the studied FCHEV energy system is shown in Fig. 1.

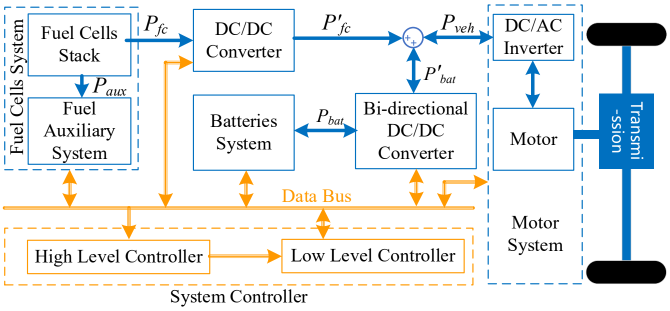


Fig. 1. Energy system for fuel cell hybrid electric vehicle.

A. Vehicle dynamics model

The dynamic model of the vehicle is shown as (1) with the velocity v and the road slope θ.

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

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:

$$P_{veh} = F_m \cdot v / \eta_m \quad (2)$$

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

$$P_{veh} = P'_{fc} + P'_{bat} \quad (3)$$

For the studied vehicle, the vehicle weight is 2500 kg, the windward area is 1.8 m^2, the air density is 1.25 kg/m^2, the air resistance coefficient is 0.3, the rolling friction coefficient is 0.01, and the total mechanical transmission efficiency is set as 90%, the gravity acceleration is 9.8 m/s^2

a) Fuel cell model

The mathematical model of the fuel cells stack is as follows:

$$\begin{aligned} V_{fc} &= n_{cell} \cdot (E_{nst} - V_{act} - V_{con} - V_{ohm}) \\ I_{fc} &= A_{fc} \cdot i_{fc} - I_{aux} \end{aligned} \quad (4)$$

where n_{cell} is the number of single fuel cells, E_{nst} is the theoretical voltage called the Nernst electromotive force, V_{act} is the activated polarization voltage, V_{con} is the concentration polarization voltage, and V_{ohm} is the ohmic voltage loss. I_{fc} is the output current of the fuel cell stack, I_{aux} is the auxiliary system current, i_{fc} is the current flowing through the unit area of the electrode plate. The specific model of each voltage part is expressed as:

$$\begin{cases} E_{nst} = E_0 + \dfrac{\Delta TS}{nF} - \dfrac{RT}{nF} \ln\left(\dfrac{P_{H_2O}}{P_{H_2}\sqrt{P_{O_2}}}\right) \\ V_{act} = \dfrac{RT}{\alpha F} \ln\left(\dfrac{i_{fc} + i_{loss}}{i_0}\right) \\ V_{con} = \dfrac{RT}{nF} \ln\left(\dfrac{I_{lim}}{I_{lim} - i_{fc}}\right) \\ V_{ohm} = i_{fc} R_{ohm} \end{cases} \quad (5)$$

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, $\Delta T = T - 273.15$, $n = 2$, $F = 96485$ is Faraday constant, $\alpha = 1$ is the transfer coefficient, P is the local pressure of the reactants and products at this atmospheric pressure. 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} is the fuel cell resistance. Then the hydrogen consumption model of the FC stack can be derived as follows:

$$\dot{m}_{H_2} = M_{H_2} \frac{I_{fc}}{nF} = \frac{M_{H_2} P_{fc}}{n V_{fc} F} \quad (6)$$

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 total power of fuel cells stack.

For the fuel cells side DC/DC converter, only its efficiency model is considered, since its response time scale is much smaller than that of the fuel cell systems:

$$\eta_{dc} = \frac{P'_{fc}}{P_{fc} - P_{aux}} = \frac{P'_{fc}}{P_{fc} - V_{fc} I_{aux}} \quad (7)$$

where P'_{fc} is the output power of the FC system to the DC bus. η_{dc} is the efficiency of DC/DC converter. P_{aux} is the auxiliary system power, and $I_{aux} = 2.0\ A$.

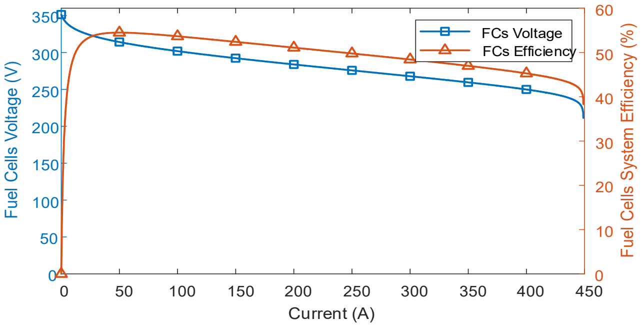


Fig. 2. The output voltage and efficiency of fuel cells

In our application, the cells number is $n_{cell} = 370$, the electrode area is set as $A_{fc} = 324\ cm^2$, the anode hydrogen pressure is 50 kPa, and cathode oxygen is obtained from the air by natural aspiration. Fig. 2 depicts the characteristics of fuel cells stack. The maximum power point is the current of 437A and the power of 104kW with the efficiency of 60%. The highest efficiency point is the current of 63.2A, and the efficiency is: 54.49% with the power of 15.7kW.

B. Battery model

The modeling of the batteries is as shown in (8), which is first order circuit model [14].

$$\begin{cases} I_{bat} = \dfrac{V_{oc} - \sqrt{V_{oc}^2 - 4R_{bat}P_{bat}}}{2R_{bat}} \\ SOC_{bat} = SOC_{bat}(0) - \int_0^t I_{bat} / Q_{bat} dt \end{cases} \tag{8}$$

where I_{bat} is the output current of the batteries. When $I_{bat} > 0$, the battery is discharged, and when $I_{bat} < 0$, the battery is charged. SOC_{bat} is the state of charge (SOC) of batteries. Q_{bat} is the battery capacity. Especially, the open-circuit voltage V_{oc} and the internal resistance R_{bat} are dependent on SOC_{bat}, and their characteristics are shown in Fig. 3. For the battery-side DC/DC converter, its efficiency η_{bdc} can be expressed as:

$$\eta_{bdc} = \begin{cases} P'_{bat} / P_{bat} & (P_{bat} > 0) \\ P_{bat} / P'_{bat} & (P_{bat} < 0) \end{cases} \tag{9}$$

where P'_{bat} is the output power of the bi-directional converter. The capacity of the studied battery is 6.6 Ah, and the standard voltage is 244.8V.

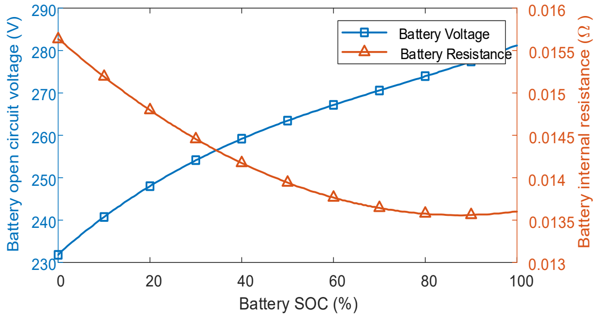


Fig. 3. The characteristics of the batteries

III. The Proposed Novel Fuzzy Q-Learning based EMS

A. EMS problem formulation

The objective of EMS is to optimize vehicle performance by adjusting the distribution of energy between different energy sources under constraints. In the paper, the objective is to minimize the fuel (hydrogen) consumption and maintain the battery SOC. The objective is formulated mathematically as the integral of instantaneous reward $r(t)$:

$$\begin{aligned} &\max J = \int_0^\infty r(t)dt \\ &r(t) = -\dot{m}_{H_2}(t) - k_{soc}(SOC_{bat}(t) - SOC_{ref})\Delta SOC_{bat}(t) \\ &\Delta SOC_{bat}(t) = SOC_{bat}(t) - SOC_{bat}(t-1) \end{aligned} \tag{10}$$

where SOC_{ref} is the reference of SOC corresponding to the battery characteristics, k_{SOC} is the weight factor of the SOC, and ΔSOC_{bat} is the difference of the current SOC compared to the previous moment. 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 maximazation of the objective function J. Considering the system characteristics and security, the following constraints $\mathcal{S}$ and $\mathcal{A}(s)$ should be satisfied:

$$\begin{aligned} &\mathcal{S}: \begin{cases} -50kW \le P_{veh} \le 50kW \\ 0 \le SOC_{bat} \le 100\% \end{cases} \\ &\mathcal{A}: 0 \le P_{veh} \le 100kW \end{aligned} \tag{11}$$

B. Fuzzy logic approximator:

Fuzzy logic imitates the fuzzy cognition and reasoning ability of the human brain. As shown in Fig. 4, a basic fuzzy inference system (FIS) consists of four parts: fuzzifier, defuzzifier, inference engine, and knowledge base.

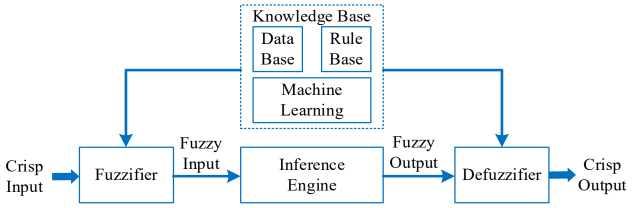
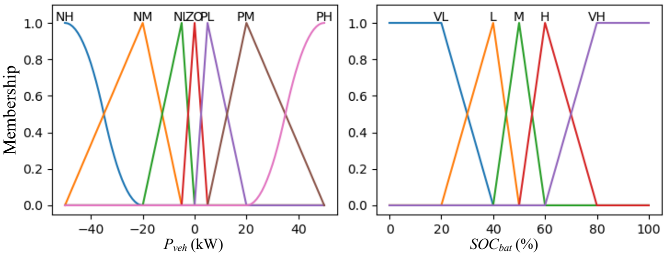


Fig. 4. Fuzzy Interrace System scheme

With fuzzifier, the FIS converts the crisp states s to fuzzy values by membership functions. Then the obtained membership degrees are combined with each other by logical "AND", thus the fuzzy state $\phi(s) = [\phi_1, \phi_2, \ldots, \phi_M]$ of each rule can be derived. In our EMS application, the input state is the state $s = [P_{veh}, SOC_{bat}]$, and the crisp output is the action $a = [P_{fc}]$.

The memebership functions of 2 states are designed as Fig. 5. For state $s_1 = P_{veh}$, 7 membership functions are designed. And the typical values of the s_1 memebership funciton is set as [-50,-20,-10,0,10,20,50] (kW) with fuzzy labels ["NH", "NM", "NL", "ZO", "PL", "PM", "PH"]. For state $s_2 = SOC_{bat}$, 5 membership functions are designed with typical values [20%,40%,50%,60%,80%] whose labels are ["VL", "L", "M", "H", "VH"]. Hence, the dimensional number of fuzzy state $\phi(s)$ is $M = 35$.



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

Fig. 5. Membership functions of 2 input states

Here, the output of FIS is defuzzied with a Takagi-Sugeno fuzzy model, and the average weighted method is adopted in TS fuzzy model as shown in (12).

$$y_k = \frac{\sum_{i=1}^{M} y_{k,i}\phi_i(s)}{\sum_{i=1}^{M} \phi_i(s)}, k = 1, 2, \ldots, N_{out} \tag{12}$$

where y_k is the k^{th} output, N_{out} is the number of the output variables, and $y_{k,i}$ means the i_{th} fuzzy ouput of the y_k. Traditionally, the value of $y_{k,i}$ will be determined by fuzzy rules. Here the fuzzy rules will not be specfic rules, but instead they will be established by reinforcement learning and then infer fuzzy outputs.

In the paper, when the output of FIS is $a = P_{fc}$, the typical value of the fuzzy output sets are U = [0, 1, 2, 5, 10, 20, 50, 100] (kW). Thus the number of fuzzy action is $N_U = 8$. Their fuzzy labels are [“ZO”, “SL”, “VL”, “L”, “M”, “H”, “VH”, “SH”]. Then the fuzzy actions a_i , $i = 1,2, \ldots, M$ will be chosen from the typical values U of fuzzy output sets. As a result, the ouput action a can be deffuzied with fuzzy actions a_i and fuzzy state $\phi_i(s)$ by average weighted method, which is the first-order TS fuzzy model.

C. Reinforcement learning principle

Reinforcement learning realizes continuous self-learning by the interaction of agent and environment. Meanwhile, the environment is required to have Markov properties, which means that the transition probability of the next state can be only determined by the current state $s(t)$ and the action a(t). Thus, a sequence $[s(0), a(0), r(0), s(1), a(1), r(1), \ldots]$ can be obtained during the learning process until the terminal state. This process is a Markov decision process.

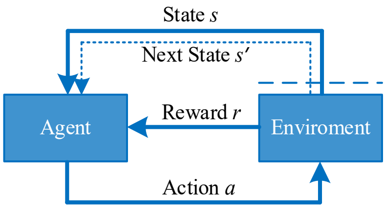


Fig. 6. Reinforcement Learning Principle

The goal of the RL agent is to maximize the cumulative rewards J from the initial state s_0 to the terminal state s_T at time $t = T$ by optimizing the policy.

$$J(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r(t)\right] \tag{13}$$

where π is the policy to obtain the action, and γ is a discount factor, $0 \le \gamma < 1$. $\gamma = 0$ means immediate return, γ tends to 1 means future return. And γ determines how much the future time will affect the return. The cumulative return is used as an evaluation function to evaluate the performance of the policy. The value function of the state s at time t is donated as

$$V(s) = \mathop{\mathbb{E}}_{s \sim \rho(s)}\left[\sum_{k=t}^{k=T} \gamma^{k-t} r(k)\right] \tag{14}$$

However the cumulative reward is difficult to calculate directly due to unknow future rewards. Q-learning uses the Q-function to solve the problem, which is a time-difference method, thus it can be updated step by step by bootstrapping. The Q-value represents the state-action value at the time t:

$$Q(s,a) = r(t) + \gamma \mathop{\mathbb{E}}_{s \sim s'}\left[V(s')\right] \tag{15}$$

where s' is the next state, and $Q(s,a)$ means the expectation of the discounted communitive rewards from state s and action a to the terminal state, and $V(s')$ is the value function of the next state s'. In Q-learning, maximization estimation method is used to obtain the estimated value function donated as $V^*(s')$ [9].

$$\begin{aligned} Q^*(s,a) &= \mathbb{E}_{s \sim \rho(s), a \sim \pi(s)}\left[r + \gamma \mathbb{E}_{s \to s'}\left[V^*(s')\right]\right] \\ &= \mathbb{E}_{s \sim \rho(s), a \sim \pi(s)}\left[r + \gamma \max \mathbb{E}_{a' \sim \pi(s')}\left[Q(s',a')\right]\right] \end{aligned} \tag{16}$$

where $Q^*(s,a)$ the optimal state-action value function, then the optimal policy π^* is specified as follows:

$$\pi^*(s) = \arg \max_{a \sim \pi(s')} (Q^*(s,a)) \tag{17}$$

The Q-Learning will be updated with the time-difference learning of (16). The update law is expressed as follows [15]:

$$\begin{aligned} Q(s,a) &= Q(s,a) + \alpha \cdot \Delta Q(s,a,s') \\ \Delta Q(s,a,s') &= r + \gamma \max_{a' \sim \pi(s')} Q(s',a') - Q(s,a) \end{aligned} \tag{18}$$

where $\alpha \in (0,1)$ is the learning rate of Q-learning, which represents the update speed of the Q-table $Q(s,a)$. According to general experience, α is set close to 0, and γ is close to 1. However, only for discrete space and overestimation of Q-Learning limit its application in energy management.

D. Fuzzy Double Q-learning

Overestimation of the value function of Q-learning will lead the agent to make non-optimal decisions in some states. "Overestimation" usually refers to finding the maximum value and then finding the expectation for a series of numbers, which is usually larger than finding the expectation first and then finding the maximum value. The mathematical expression is:

$$\mathbb{E}[\max(X_1, X_2, \ldots)] \ge \max(\mathbb{E}[X_1], \mathbb{E}[X_2], \ldots) \tag{19}$$

To solve the overestimation problem of Q-Learning, double-estimator is proposed to evaluate the state-action value function of next state s', which is double Q-learning [10]. Double Q-learning stores two Q-functions as the double-estimator: $Q_1(s,a)$ and $Q_2(s,a)$, and each Q-function will be updated with the other’s maximum action a_2^* and a_1^* of the next state s'. For each step, there will be only one Q-function can be updated with s' with probability 0.5. After updated by enough samples, the expectations of $Q_1(s', a_2^*)$ and $Q_2(s', a_1^*)$ will approach the optimal state-action value $Q^*(s', a^*)$.

$$\mathbb{E}[Q_1(s', a_2^*)] = Q^*(s', a^*) = \mathbb{E}[Q_2(s', a_1^*)] \tag{20}$$

where the actions a^*, a_1^* and a_2^* will be based on the greedy policy. Equation (20) is the application of double-estimator in Q-Learning. Theoretically, either of $Q_1(s,a)$ and $Q_2(s,a)$ can be used as the estimate of $Q(s,a)$, but it is usually the average value of two Q-functions. In addition, fuzzy logic is applied to approximate the value function to overcome the weakness of double Q-Learning.

In the proposed FDQL, double-estimator is used to suppress the overestimation of Q-learning. And the fuzzifier is applied to transfer continuous states into fuzzy states [16], while the action is obtained by the defuzzifier with the fuzzy actions. In fuzzy double Q-learning, 2 Q-functions need to be estimated with average weighted defuzzifier.

$$Q_1 = \frac{\boldsymbol{q}_1^{\mathrm{T}} \phi(s)}{\sum \phi(s)}, Q_2 = \frac{\boldsymbol{q}_2^{\mathrm{T}} \phi(s)}{\sum \phi(s)} \tag{21}$$

where Q_1 , Q_2 are the 2 evaluated Q-functions of double-estimator, and q_1, q_2 are 2 fuzzy q-arrays. Donated V_1, V_2 are the two value functions of the next state s':

$$\begin{aligned} V_1(s') &= Q_1(s', a_2^*) \\ V_2(s') &= Q_2(s', a_1^*) \end{aligned} \tag{22}$$

which are determined by each other's greedy actions a_2^* and a_1^*:

$$\left.\begin{aligned} a_{1,i}^* &= \arg \max_{a_{1,i}^* \in U} q_{1,i}(\phi_i(s'), a_{1,i}^*) \\ a_{2,i}^* &= \arg \max_{a_{1,i}^* \in U} q_{2,i}(\phi_i(s'), a_{2,i}^*) \end{aligned}\right., i = 1, \ldots, M \tag{23}$$

where $q_{1,i}$, $q_{2,i}$ are the i^{th} fuzzy q-arrays of q_1, q_2, and $a_{1,i}^*$ and $a_{2,i}^*$ are the i^{th} greedy fuzzy actions to maximum $q_{1,i}$, $q_{2,i}$. In the paper, the sizes of each $q_{1,i}$ and $q_{2,i}$ are $[1 \times N_U]$ Both of greedy fuzzy actions will sample from the predefined fuzzy output set U. Then the i^{th} fuzzy evaluated value functions $v_{1,i}$ and $v_{2,i}$ can be update with each other's greedy fuzzy actions $a_{2,i}^*$ and $a_{1,i}^*$.

$$\begin{aligned} v_{1,i}(\phi_i(s')) &= q_{1,i}(\phi_i(s'), a_{2,i}^*) \\ v_{2,i}(\phi_i(s')) &= q_{2,i}(\phi_i(s'), a_{1,i}^*) \end{aligned} \tag{24}$$

To obtain the crisp value functions V_1 and V_2, the defuzzifier of the fuzzy inference system should be employed with the next fuzzy state $\phi(s')$.

$$V_1(s') = \frac{\boldsymbol{v}_1^\top \phi(s')}{\sum \phi(s')}, V_2(s') = \frac{\boldsymbol{v}_2^\top \phi(s')}{\sum \phi(s')} \tag{25}$$

where V_1, V_2 are obtained by defuzzing the fuzzy state values $\boldsymbol{v}_1 = [v_{1,1}, v_{1,2}, \ldots, v_{1,M}]$ and $\boldsymbol{v}_2 = [v_{2,1}, v_{2,2}, \ldots, v_{2,M}]$ corresponding to each rule fired strength of the next state s' by applying the weighted average method. Then the i_{th} rule of FDQL with fuzzy language can be formed as follows:

***IF** the fuzzy state $\phi_i(s)$, **THEN** take $a_{2,i}^*$ with $q_{2,i}$ to update $v_{1,i}$*
take $a_{1,i}^*$ with $q_{1,i}$ to update $v_{2,i}$

And there will be M fuzzy rules. The fuzzy double Q-Learning will also be updated with the time-difference learning, and the target Q-function is set as: $Q_{1,target}(s,a) = r + \gamma V_1(s')$ and $Q_{2,target}(s,a) = r + \gamma V_2(s')$. Then $\Delta q_{1,\mathrm{i}}$ and $\Delta q_{2,\mathrm{i}}$ the increments of the fuzzy double q-arrays corresponding to the i^{th} rule are shown as:

$$\begin{aligned} \Delta q_{1,i} &= [r + \gamma V_1(s') - Q_1(s,a)] \frac{\phi_i(s)}{\sum_{i=1}^{M} \phi_i(s)} \\ \Delta q_{2,i} &= [r + \gamma V_2(s') - Q_2(s,a)] \frac{\phi_i(s)}{\sum_{i=1}^{M} \phi_i(s)} \end{aligned} \tag{26}$$

where $\Delta \boldsymbol{q}_1$ and $\Delta \boldsymbol{q}_2$ are based on the fuzzifier with fuzzy state $\phi(s)$. And they are used in (27) to update the fuzzy q-arrays: $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ which are corresponding to each rule update:

$$\begin{aligned} q_{1,i}(\phi_i(s), a_i) &:= q_{1,i}(\phi_i(s), a_i) + \alpha \Delta q_{1,i} \\ q_{2,i}(\phi_i(s), a_i) &:= q_{2,i}(\phi_i(s), a_i) + \alpha \Delta q_{2,i} \end{aligned} \tag{27}$$

where $\alpha \in (0,1)$ is the learning rate of the FDQL. a_i are the actual fuzzy actions corresponding to the i^{th} rule. Here, the average value of the two fuzzy q-functions is used as the estimate fuzzy state-action value $q = (q_1 + q_2)/2$. Then the action can be determined by maximum policy.

$$\left.\begin{aligned} q_i &= \frac{1}{2}\left(q_{1,i} + q_{2,i}\right) \\ a_i &= \arg\max_{a_i \in U} q_i(\phi_i(s), a_i) \end{aligned}\right. , i = 1, \ldots, M \tag{28}$$

To avoid difficulty in convergence due to the accumulation of errors in q_1 and q_2, it is necessary to synchronize q_1 and q_2 with q every certain period N_{syn} of the episodes, that is: $q_1 = q_2 = q$. At that time, add the following rules need to be added in the i^{th} fuzzy rule of the proposed fuzzy method:

***IF** the fuzzy state $\phi_i(s)$, **THEN** update $q_{1,i}$, $q_{2,i}$ with ΔQ_1, ΔQ_2*
and take a_i with $(q_{1,i} + q_{1,i})/2$

The fuzzy actions need to be transformed into the actual action with average weighted defuzzifier. Then the optimal action is:

$$a^\dagger(s) = \frac{\sum_{i=1}^{M} a_i \phi_i(s)}{\sum_{i=1}^{M} \phi_i(s)} \tag{29}$$

where $a^\dagger(s)$ is the action derived by maximum policy. To avoid the agent of the RL falling into a local optimum during the learning process, it is necessary to balance exploration and exploitation. Therefore, the behavior policy needs to be selected with a certain probability between the optimal action and the random action.

$$a = \begin{cases} a^\dagger(s) & , \varepsilon \le \mathcal{N}_t(0,1) \\ a_{\mathcal{N}} \in \mathcal{A}(s) & , \varepsilon > \mathcal{N}_t(0,1) \end{cases} \tag{30}$$

where a is the behavior policy with ε-greedy, and $a_{\mathcal{N}}$ is the random action according to the present state s. $\mathcal{N}_t$ is a random value that sampled from [0,1]. In the paper, the exploration rate ε decays exponentially from 1 to close to 0. With FDQL, the requirement of continuous state and action space problems are solved with fuzzy inference system, and the overestimation problem is suppressed by the double-estimator. Then the pseudocode of the proposed FDQL is as follows:

TABLE I. THE PROCEDURE OF THE PROPOSED FDQL

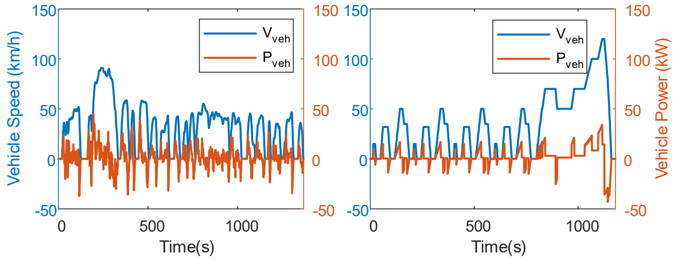
```
Fuzzy Doube Q-Learning (FDQL)
Randomly initialize q-arrays: $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, with the size of $[M \times N]$
M: the number of fuzzy rules; N: the total number of fuzzy outputs
for $episode = 1$ to $L$ do:
    Reset the environment with the initialized state $s_0$
    Obtain fuzzy state $\phi(s)$ with membership fucntions of each rules
    for $t = 1$ to $T$ do:
        Obtain fuzzy actions: $a_i$, with maximizing $(q_{1,i} + q_{2,i})/2$
            $q_i = (q_{1,i} + q_{2,i})/2,\ a_i = \arg\max_{u_i \in U} q_i(\phi_i(s), u_i), i = 1, \ldots, M$
        Select action $a$ by defuzzing fuzzy actions and $\varepsilon$-Greedy
            $a^\dagger(s) = \frac{\sum_{i=1}^{M} a_i \phi_i(s)}{\sum_{i=1}^{M} \phi_i(s)}, a = \begin{cases} a^\dagger(s) & , \varepsilon \le \mathcal{N}_t(0,1) \\ a_{\mathcal{N}} \in \mathcal{A}(s) & , \varepsilon > \mathcal{N}_t(0,1) \end{cases}$
        Observe the reward r and next state $s'$
        Obtain the next fuzzy state $\phi(s')$
        Evaluate fuzzy value functions:
            $a_{k,i}^* = \arg\max_{a_{k,i}^* \in U} q_{k,i}(\phi_i(s'), a_{1,i}^*)$    $k = 1,2, i = 1, \ldots, M$
            $v_{k,i}(\phi_i(s')) = q_{k,i}(\phi_i(s'), a_{3-k,i}^*)$
        Get value funtions and Q-functions with deffuzifier:
            $V_k(s') = \frac{\boldsymbol{v}_k^\top \phi(s')}{\Sigma \phi(s')}, Q_k = \frac{\boldsymbol{q}_k^\top \phi(s)}{\Sigma \phi(s)}, k = 1,2$
        with 0.5 probability:
            Get the increment of the 1st fuzzy q-array: $\Delta q_{1,i}$, with $V_1(s')$
                $\Delta q_{1,i} = [r + V_1(s') - Q_1(s,a)] \frac{\phi_i(s)}{\sum_{i=1}^{M} \phi_i(s)}, i = 1, \ldots, M$
            Update the fuzzy q-array $\boldsymbol{q}_1$:
                $q_{1,i}(\phi_i(s), a_i) := q_{1,i}(\phi_i(s), a_i) + \alpha \Delta q_{1,i}, i = 1, \ldots, M$
        else:
            Get the increment of the 2nd fuzzy q-array: $\Delta q_{2,i}$, with $V_2(s')$
                $\Delta q_{2,i} = [r + V_2(s') - Q_2(s,a)] \frac{\phi_i(s)}{\sum_{i=1}^{M} \phi_i(s)}, i = 1, \ldots, M$
            Update the fuzzy q-array $\boldsymbol{q}_2$:
                $q_{2,i}(\phi_i(s), a_i) := q_{2,i}(\phi_i(s), a_i) + \alpha \Delta q_{2,i}, i = 1, \ldots, M$
                Update state: $s \leftarrow s', \phi(s) \leftarrow \phi(s')$
    end for
    if $episode$ % $N_{syn}$ == 0:
        $q = (q_1 + q_2)/2,\ q_1 = q_2 = q$
    end if
end for
```

IV. SIMULATIONS AND RESULTS ANALYSIS

The proposed FDQL-based EMS for FCHEV will be trained and tested by a python-based platform with version 3.8.13. The processor is Intel(R) Core (TM) i5-9400H CPU @ 2.50GHz, and there is no need to use GPU due to its less computation. The environment model of FCHEV is referred to the report of Toyota's *Mirai* FCHEV given by the Argonne National Laboratory [17].

A. Test driving cycles

The proposed FDAL-based EMS is tested under 4 standard driving cycles *Urban Dynamometer Driving Schedule* (UDDS), *New European Driving Cycle* (NEDC). The velocity and power of the specific FCHEV under those 2 driving cycle are shown in Fig. 7.



(a) "UDDS" (b) "NEDC"

Fig. 7. Velocity and power of the FCHEV under different driving cycles

The detailed information of those driving cycles are shown in TABLE II. For the proposed EMS, the "UDDS" will be used as the external input of the training environment with the initial state $SOC_{bat} = 50\%$, and the test process will be based on those 2 driving cycles with the different initial states.

TABLE II. Different Driving Cycles Information

Driving Cycle	Duration (s)	Distance (km)	Average speed (km/h)	Minimum Power (kW)	Maximum Power (kW)
UDDS	*1369*	11.99	31.53	-37.19	44.89
NEDC	*1180*	10.93	33.35	-43.07	34.22

B. Training Process of the proposed FDQL-based EMS

In the training process, the learning rates for both two fuzzy Q-functions of the proposed FDQL are set as $\alpha = 0.002$, the decay factor for the cumulative reward is $\gamma = 0.999$, the reference for SOC trajectory is set as $SOC_{ref}(t) = 50\%$, and the weight factor of SOC in the objective function is $k_{SOC} = 200$. The training environment is a python-based energy system of FCHEV with the driving condition of "UDDS" with the initial SOC of batteries $SOC_{bat}(0) = 50\%$.

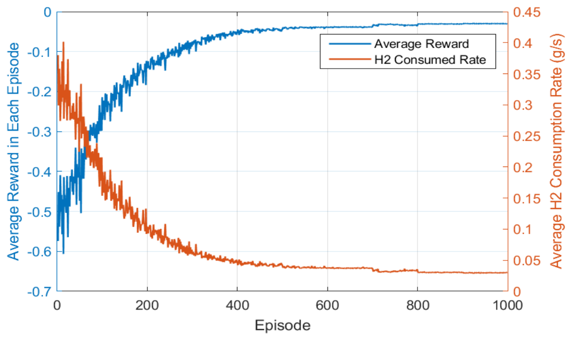


Fig. 8. The average reward and the average fuel consumption rate in each episode during the training process of the proposed FDQL-based EMS for FCHEV.

For the training process, the average reward and the average hydrogen consumption for each episode are shown in Fig. 8. The agent converges steadily until the 800th episode, and the training time is 20 minutes, which verifies the fast convergence and effectiveness of the proposed FDQL during the training process.

C. Test Process of the proposed FDQL-based EMS

The test process will be based on 4 different driving conditions mentioned before and 3 different initial state of SOC: 25%, 50% and 75%. For the well-trained agent of FDQL with the driving condition "UDDS", the other 3 driving cycles are total unknown. This will be a challenging ordeal for the adaptability of the proposed FDQL to unknown dynamics.

In Fig. 9, the detail power allocation strategy is illustrated under the driving condition of the cycle "UDDS", and the initial state of SOC is set as 50%, the same as the training environment. The cumulative reward during the test task is -42.56, and the average reward is -0.032. The cumulative hydrogen consumption is 42.56g, hence the hydrogen consumption per 100 km is 354.96g.

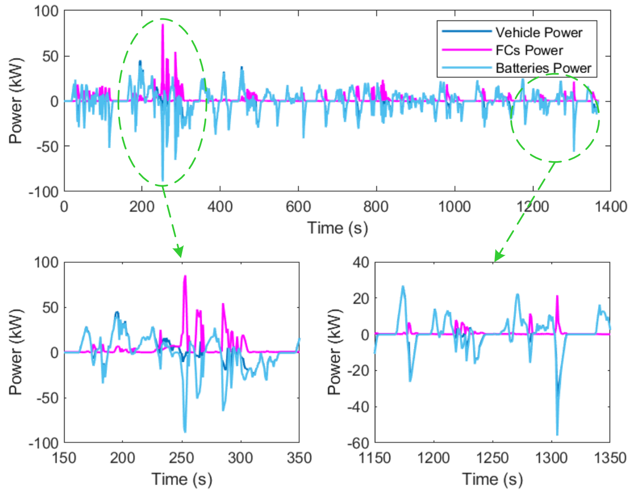
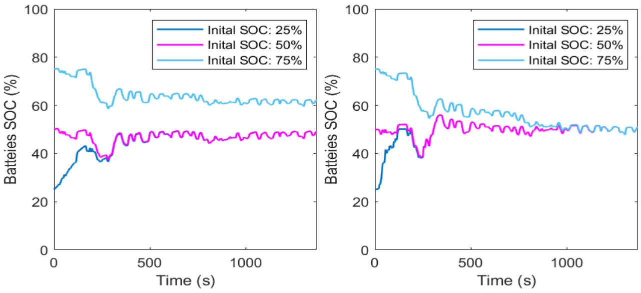


Fig. 9. Power allocation strategy of the proposed FDQL based EMS under the driving cycle "UDDS" with initial state of SOC 50%.



(a) FQL (b) FDQL

Fig. 10. SOC trajectory of the FQL and the FDQL based EMS under the driving cycle "UDDS" with different initial state of SOC.

To further verify that the well-trained agent is not sensitive to the initial state, and the agent can adjust the SOC trajectory to be near the preset reference value under different SOC initial values, several tasks are being tested with the initial state of SOC 25%, 50% and 75%. SOC trajectory of the FQL and the FDQL based EMS are shown in Fig. 10. As shown in Fig. 10 (a), the terminal SOC state of the FQL based EMS under initial SOC 25%, 50% and 75% are 49.06%, 49.07% and 62.13%. And in Fig. 10 (b), the same tests based on the proposed FDQL are carried out with the results of 50.93%, 50.93% and 50.94%. The test results show that the proposed FDQL have better the adaptability performance on initial state change.

The next test is about the adaptability of the proposed EMS to different driving conditions. The "UDDS" cycle is the training condition, while "NEDC" is test condition, which are completely unknown to the well-trained agent. In Fig. 11, it illustrates SOC trajectories of the batteries under 10 repeat cycles of those 2 standard driving conditions with the FQL-based EMS. The terminal states of SOC are 49.06% and 49.67 with the driving conditions of "UDDS" and "NEDC", separately. And in Fig. 12, it illustrates that the terminal SOC states of two diving conditions are 50.93% and 50.84% under the same test environment. The test results shows that the proposed FDQL based-EMS has faster convergency than that of the FQL on SOC trajectories, which means better adaptability performance to unknow driving conditions.

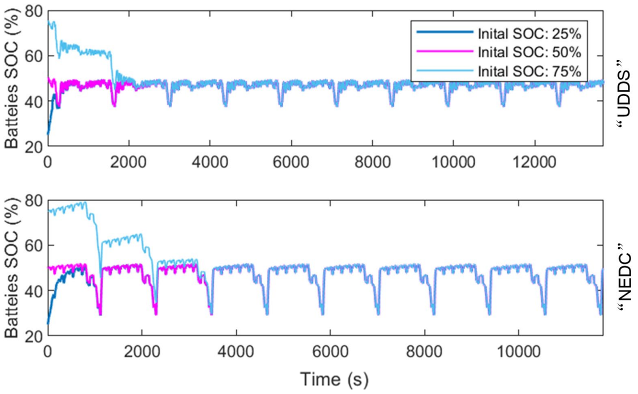


Fig. 11. The SOC trajectary of 10 repeat cycles of 2 different driving conditions under different initial state of SOC with the FQL-based EMS.

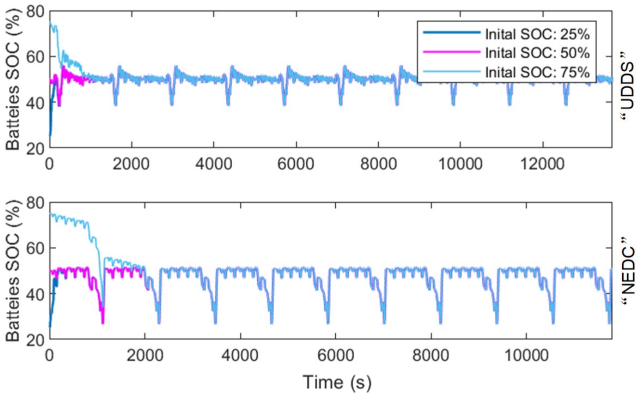


Fig. 12. The SOC trajectary of 10 repeat cycles under different initial state of SOC with the FDQL-based EMS.

TABLE III. TEST RESULTS OF THE FQL-BASED AND THE PROPOSED FDQL-BASED EMS

Driving Cycle	**Initial SOC**	**FQL-based EMS**			**FDQL-based EMS**		
		Average Reward	***H2 Rate (g/km)***	***Average SOC***	***Average Reward***	***H2 Rate (g/km)***	***Average SOC***
UDDS	*25%*	-0.049	60.63	44.78%	-0.052	64.44	48.75%
	50%	**-0.030**	**40.74**	**46.91%**	**-0.032**	**42.56**	**49.93%**
	75%	-0.028	30.73	64.11%	-0.021	22.05	56.95%
NEDC	*25%*	-0.055	58.72	43.54%	-0.058	62.08	45.80%
	50%	**-0.033**	**39.10**	**47.75%**	**-0.035**	**40.77**	**47.50%**
	75%	-0.030	27.97	73.35%	-0.025	23.14	67.54%

V. Conclusion

In the paper, a novel fuzzy reinforcement learning algorithm FDQL-based energy management strategy is proposed to suppress the overestimation of fuzzy Q-learning. The proposed FDQL combines the fuzzy logic and reinforcement learning to approximate the state-action value function of Q-learning, so that continuous action and state space operations can be enabled. Besides, the training time can be significantly reduced, which makes it possible to apply for real-time training and tests. Moreover, since the proposed FDQL is a model-free self-learning method, it does not need to model the energy system of FCHEV, and can adjust itself to adapt to the changed environment. Finally, the Python-based training and testing platform verify that the proposed FDQL-based EMS has good adaptability to the unknown driving conditions, and uncertain initial states. and it shows excellent performance in those difficulty tests to achieve the goals of saving hydrogen fuel consumption and maintaining the batteries operation for a long time.

Acknowledgment

This work has been supported by the ANR DEAL (contract ANR-20-CE05-0016-01). This work has also been partially funded by the CNRS Energy unit (Cellule Energie) through the project "PEPS GIALE".

References

[1] I. Sorlei *et al.*, "Fuel Cell Electric Vehicles — A Brief Review of Current Topologies and Energy Management Strategies," *Energies*, vol. 14, no. 1, pp. 252–280, 2021.

[2] T. Liu, W. Tan, X. Tang, J. Zhang, Y. Xing, and D. Cao, "Driving conditions-driven energy management strategies for hybrid electric vehicles : A review," *Renew. Sustain. Energy Rev.*, vol. 151, no. July, p. 111521, 2021, doi: 10.1016/j.rser.2021.111521.

[3] O. Sundström, L. Guzzella, and P. Soltic, "Optimal Hybridization in Two Parallel Hybrid Electric Vehicles using Dynamic Programming," *IFAC Proc. Vol.*, vol. 41, no. 2, pp. 4642–4647, 2008, doi: 10.3182/20080706-5-kr-1001.00781.

[4] B. Liu, L. Li, X. Wang, and S. Cheng, "Hybrid Electric Vehicle Downshifting Strategy Based on Stochastic Dynamic Programming during Regenerative Braking Process," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 4716–4727, 2018, doi: 10.1109/TVT.2018.2815518.

[5] B. H. Nguyen, R. German, J. P. F. Trovao, and A. Bouscayrol, "Real-time energy management of battery/supercapacitor electric vehicles based on an adaptation of pontryagin's minimum principle," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 203–212, 2019, doi: 10.1109/TVT.2018.2881057.

[6] W. Zhang, J. Li, L. Xu, and M. Ouyang, "Optimization for a fuel cell/battery/capacity tram with equivalent consumption minimization strategy," *Energy Convers. Manag.*, vol. 134, pp. 59–69, 2017.

[7] E. Charniak, *Introduction to Deep Learning*, vol. 91, no. 5. 2012.

[8] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[9] C. J. C. H. Watkins and P. Dayan, "Q-Learning," *Mach. Learn.*, vol. 292, pp. 279–292, 1992.

[10] H. Van Hasselt, "Double Q-learning," *Adv. Neural Inf. Process. Syst. 23 24th Annu. Conf. Neural Inf. Process. Syst. 2010, NIPS 2010*, pp. 1–9, 2010.

[11] P. Y. Glorennec and L. Jouffe, "Fuzzy Q-learning," *IEEE Int. Conf. Fuzzy Syst.*, vol. 2, no. 3, pp. 659–662, 1997.

[12] L. Bo, L. Han, C. Xiang, H. Liu, and T. Ma, "A Q-learning fuzzy inference system based online energy management strategy for off-road hybrid electric vehicles," *Energy*, vol. 252, p. 123976, 2022.

[13] L. Guo, Z. Li, and R. Outbib, "Fuzzy Rule Value Reinforcement Learning based Energy Management Strategy for Fuel Cell Hybrid Electric Vehicles," in *IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2022, pp. 1–7.

[14] R. Lian, J. Peng, Y. Wu, H. Tan, and H. Zhang, "Rule-interposing deep reinforcement learning based energy management strategy for power-split hybrid electric vehicle," *Energy*, vol. 197, p. 117297, 2020.

[15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction second edition*, vol. 9, no. 5. 2018.

[16] L. Guo, Z. Li, R. Outbib, and F. Gao, "Function Approximation Reinforcement Learning of Energy Management with the Fuzzy REINFORCE for Fuel Cell Hybrid Electric Vehicles," *Energy AI*, p. 100246, Feb. 2023.

[17] H. Lohse-Busch, K. Stutenberg, M. Duoba, and S. Iliev, "Technology assessment of a fuel cell vehicle: 2017 Toyota Mirai," 2018.