Energies **2015**, *8*, 11167-11186; doi:10.3390/en81011167

OPEN ACCESS

energies

ISSN 1996-1073

www.mdpi.com/journal/energies

Article

An Online Learning Control Strategy for Hybrid Electric Vehicle Based on Fuzzy Q-Learning

Yue Hu [1,2,3], Weimin Li [1,3,*], Hui Xu [3] and Guoqing Xu [1]

[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China; E-Mails: huy@jiat.ac.cn (Y.H.); gq.xu@siat.ac.cn (G.X.)

[2] Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen 518055, China

[3] Jining Institutes of Advanced Technology, Chinese Academy of Sciences, Jining 272000, China; E-Mail: xuh@jiat.ac.cn

* Author to whom correspondence should be addressed; E-Mail: wm.li@siat.ac.cn; Tel.: +86-159-1537-4149; Fax: +86-537-3255-897.

Academic Editor: K. T. Chau

Received: 30 June 2015 / Accepted: 12 August 2015 / Published: 9 October 2015

Abstract: In order to realize the online learning of a hybrid electric vehicle (HEV) control strategy, a fuzzy Q-learning (FQL) method is proposed in this paper. FQL control strategies consists of two parts: The optimal action-value function $Q^*(x,u)$ estimator network (QEN) and the fuzzy parameters tuning (FPT). A back propagation (BP) neural network is applied to estimate $Q^*(x,u)$ as QEN. For the fuzzy controller, we choose a Sugeno-type fuzzy inference system (FIS) and the parameters of the FIS are tuned online based on $Q^*(x,u)$. The action exploration modifier (AEM) is introduced to guarantee all actions are tried. The main advantage of a FQL control strategy is that it does not rely on prior information related to future driving conditions and can self-tune the parameters of the fuzzy controller online. The FQL control strategy has been applied to a HEV and simulation tests have been done. Simulation results indicate that the parameters of the fuzzy controller are tuned online and that a FQL control strategy achieves good performance in fuel economy.

Keywords: hybrid electric vehicle; fuzzy Q-learning (FQL) control strategy; $Q^*(x,u)$ estimator network (QEN); fuzzy parameters tuning (FPT)

1. Introduction

Hybrid electric vehicles (HEV), which combine the advantages of the fuel vehicle and pure electric vehicle, is the future of the road vehicle. Control strategy is one of the key technologies for hybrid electric vehicle and plays a decisive role on the performance of the vehicle. However, designing a highly-efficient and real time control strategy is a challenging task due to the complex structure of a HEV and an uncertain driving cycle.

Many existing control strategies are rule-based [1–4], such as the thermostatic strategy, the load-following strategy and electric assist strategy. These control strategies have been developed based on the results of extensive experimental trials and human expertise. Some other control strategies employ heuristic control techniques, with the resultant strategies formalized as fuzzy rules. Though these rule-based strategies are effective and can be easily implemented, their optimality and flexibility are critically limited by working conditions. Therefore, a control strategy that performs well under certain conditions may not provide satisfactory results under other conditions.

According to the literature [5–10], to optimize the operation of the HEV drivetrain, some model-based global optimization methods have been employed in control strategy design, such as dynamic programming (DP), sequential quadratic programming (SQP), genetic algorithms (GA), and so on. Usually, these algorithms can manage to determine the optimal power split between the engine and the motor for a particular driving cycle. However, the optimal power-split solutions obtained are only optimal with respect to a specific driving cycle and, in general, it is neither optimal nor charge-sustaining for other cycles. Unless future driving conditions can be predicted during real-time operation, there is no way to imply these control laws directly. Moreover, these methods suffer from the "curse of dimensionality" problem, which prevents their wide adoption in real-time applications. In conclusion, control strategy designs built upon global optimization techniques can serve to evaluate the potential fuel economy of a given drivetrain configuration, as well as the optimality of realizable control strategies.

Several studies, which developed neural networks to optimize the parameters of fuzzy controllers, show good fuel economy and system efficiency [11–13]. In these studies, fuzzy controllers can be easily and directly designed by optimizing parameters, such as the shape of membership functions. The strategy result shows about 2%–4% better fuel economy than the "fuzzy controller only" optimization result, but this strategy uses fixed parameters for optimization, which makes it an offline optimization strategy; thus, the parameters of fuzzy controller cannot vary from environment to environment.

To adapt different driving cycles, researchers have proposed a model predictive control (MPC) for HEV which is a closed-loop optimal control strategy [14–19]. To obtain the current control action, the optimal control problem in the finite domain is solved at each sampling instant. A dynamic model based on a predictive future, control action based on online rolling optimization, and feedback correction of the model error are the core features of the algorithm. This control strategy has the advantages of good control effect and strong robustness. During this process the limitations, uncertainty, nonlinearity, controlled variable, and manipulated variables, are dealt with effectively. However, when the prediction or control domain is very long, the MPC algorithm needs to solve an optimal control problem at each decision step and the algorithm is hardly executed in real-time for the great amount of calculation.

In order to make a control strategy adaptive to different driving cycles and convenient for practical application, we propose an approach to tune fuzzy controllers based on fuzzy Q-learning (FQL). The

FQL algorithm consists of two parts: a Q-function estimation network (QEN) and fuzzy parameters tuning (FPT). A back propagation (BP) neural network is adopted to estimate and generalize the optimal action-value function $Q(x,a)$, then $Q(x,a)$ and an evaluation signal are used to guide the fuzzy controller to tune parameters so that the fuzzy controller achieves better performance. Unlike traditional Q-learning algorithms, the optimal action is not obtained directly based on approximated values of $Q(x,a)$ and candidate discrete actions; rather, a fuzzy inference system (FIS) is applied to provide continuous control output. Compared with the Q-learning algorithm, FIS is introduced to enhance the generalizability of the state space and generate continuous action, to avoid the problem which is known as the "curse of dimensionality" in continuous systems, to tune the parameters and structure of FIS online so that FIS can be more adaptive to the external changes caused by the environment. The decrease of computational load makes FQL algorithm more convenient for practical applications.

2. Problem Formulation

The prototype vehicle is a single axis parallel HEV, and the drivetrain structure of the HEV is shown in Figure 1. The drivetrain is composed of an engine, an electric traction motor/generator, Ni-MH batteries, an automatic clutch, and an automatic/manual transmission system. The motor is directly linked between the auto clutch output and transmission input. This architecture provides the regenerative braking during deceleration and allows an efficient motor assist operation. To provide pure electrical propulsion, the engine can be disconnected from the drivetrain by the automatic clutch. Important parameters of this vehicle are given in Table 1.

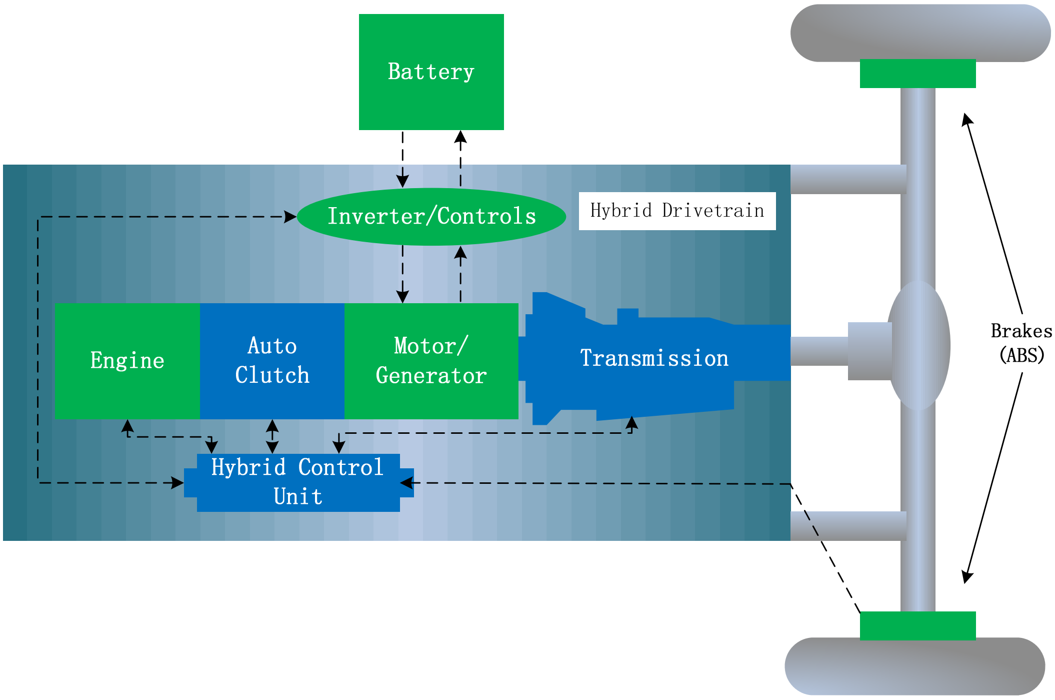


Figure 1. Schematic diagram of the parallel hybrid electric vehicle drivetrain.

Table 1. Summary of the hybrid electric vehicle (HEV) parameters.

Item	Parameter
Spark ignition (SI) engine	Displacement: 1.0 L Maximum power: 50 kW at 5700 r/min Maximum power: 89.5 N·m at 5600 r/min
Permanent magnet motor	Maximum power: 10 kW Maximum torque: 46.5 N·m
Advanced Ni-MH battery	Capacity: 6.5 Ah Nominal cell voltage: 1.2 V Total cells: 120
Automated manual transmission	5 speed GR: 2.2791/2.7606/3.5310/5.6175/11.1066
Vehicle	Curb weight: 1000 kg

The state vector of the HEV system includes three state variables, *i.e.*, $X(k) = (T_{\mathrm{dem}}(k), v(k), SOC(k))^T$, where $T_{\mathrm{dem}}(k)$ stands for required torque at time k, $v(k)$ is the vehicle speed, and $SOC(k)$ represents the remaining charge of the battery at time k. The control vector is $U(k) = T_{\mathrm{e}}(k)$, where $T_{\mathrm{e}}(k)$ represents the output torque from the engine. The motor output torque $T_{\mathrm{m}}(k)$ can be obtained by subtracting $T_{\mathrm{e}}(k)$ from $T_{\mathrm{dem}}(k)$. A torque split control strategy, which defines the best torque split between the engine and the motor, is adopted.

The control strategy goal of the HEV is to find the optimal control strategy that maps the observed states $X(k)$ to the control action $U(k)$ so as to minimize vehicle fuel consumption and emissions along a traveling route [20]. In the meantime, the vehicle drivability and battery health should be satisfied. Mathematically, the control strategy of the HEV can be formulated as an infinite-horizon dynamic optimization problem as follows:

$$J(x) = \sum_{k=0}^{\infty} \gamma^k R(k) \tag{1}$$

where $R(k)$ is the immediate cost function incurred by $U(k)$ at time k and $\gamma \in (0,1)$ is a discount factor that assures the convergence of the infinite sum of cost function. One of the key benefits of an infinite horizon problem is that the generated control strategy is time-invariant and, thus, can be easily implemented.

The cost function $R(k)$ consists of the sum of the weighted fuel economy, emissions, and SOC, as shown in Equation (2):

$$R(k) = R_{\mathrm{fuel}}(k) + a_1 R_{\mathrm{ems}}(k) + a_2 R_{\mathrm{SOC}}(k) \tag{2}$$

3. Fuzzy Q-Learning (FQL) Mechanism

The schematic diagram of a FQL control strategy is shown in Figure 2. FQL control strategies consists of two parts: *i.e.*, $Q^*(x,u)$ estimator network (QEN) and FIS parameters tuning (FPT). A BP neural network as the QEN is used to estimate $Q^*(x,u)$. For the fuzzy controller, we choose a Sugeno-type fuzzy controller.

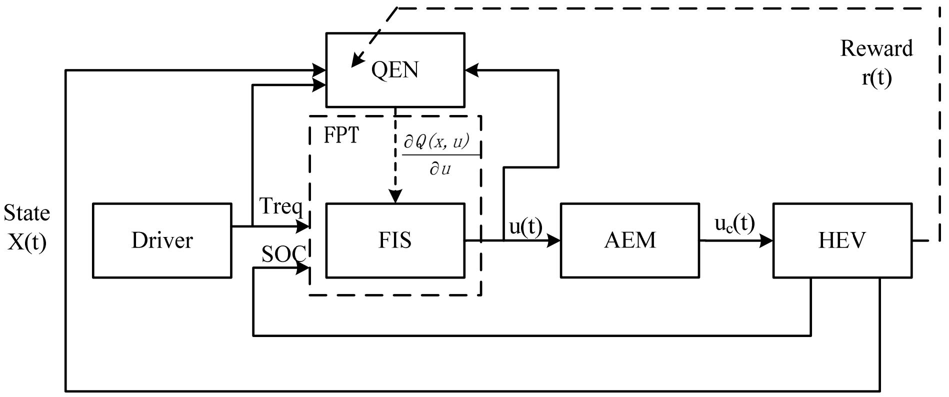


Figure 2. The schematic diagram of fuzzy Q-learning (FQL) control strategy. QEN: *Q**(*x*,*u*) estimator network; FIS: fuzzy inference system; AEM: action exploration modifier.

3.1. Back Propagation (BP) Neural Network for Estimating Q(x,u) (QEN)*

The application of reinforcement learning in control problems focuses on two main types of algorithms: actor-critic learning and Q-learning. The actor-critical learning system is a two-step process: *i.e.* to estimate the state value function $J(x)$ and to choose the optimal action for each state. For Q-learning, the system estimates an action value function $Q(x,u)$ for all state-action pairs and selects the optimal control algorithm based on $Q(x,u)$ [21].

The action value function $Q(x,u)$ is the expected discounted sum of rewards with the initial state x and initial action u which can be written as:

$$Q(x,u) = E\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid x_t = x, u_t = u\} \tag{3}$$

where u is the action that acts on the system, and $E(\ )$ is the expected value function. The optimal action-value function $Q^*(x,u)$ is represented as:

$$Q^*(x,u) = E\{r(x_{t+1}) + \gamma \max_{u'} Q^*(x_{t+1}, u') \mid x_t = x, u_t = u\} \tag{4}$$

The QEN plays the role of approximating or predicting the optimal action-value function $Q^*(x,u)$ associated with different input states and control output. A BP neural network is adopted to estimate $Q^*(x,u)$ due to its good approximation property. The architecture of the QEN is shown in Figure 3. The topology of the QEN is considered to be a three-layer structure having 4-10-1 nodes. The inputs of QEN are state variables of a HEV, and are vehicle speed $v(k)$, battery *SOC*, required torque $T_{\mathrm{dem}}(k)$, and control action $U(k)$. The output of QEN is $Q(x,u)$.

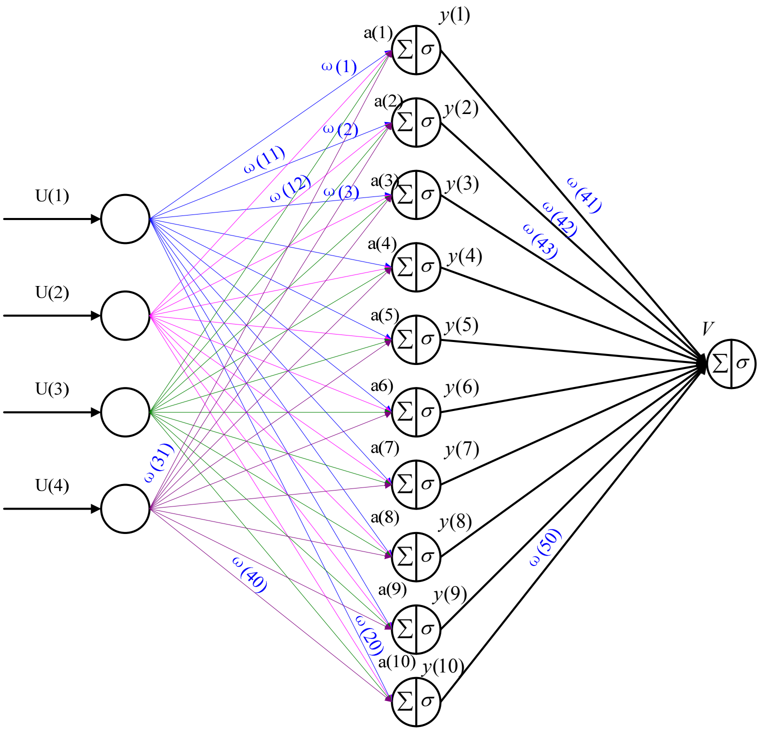


Figure 3. Architecture of the QEN.

In the QEN, $Q(x,u)$ is represented by:

$$Q(x,u) = f(V) \tag{5}$$

$$V = \sum_{i=1}^{10} \omega(40+i)y(i) \tag{6}$$

$$y(i) = f(a(i)) \tag{7}$$

$$a(i) = \sum_{j=1}^{4} U(j)\omega(j-1,i) \tag{8}$$

where,

V is the summed input of the output node;
ω$(40 + i)$ is the weight between hidden node and the output node;
$y(i)$ is the output of the hidden node;
$a(i)$ *is* the summed input of ith hidden node;
ω$(j - 1,i)$ is the weight between input node and hidden node;
$U(i)$ is the input of QEN; and
f is the activation function of the node.

Here, a sigmoid function is adopted as an activation function of the node, *i.e.*, $f(x) = 1/[1 + \exp(-x)]$.

The parameters of the QEN are tuned based on generalized policy iteration (GPI). We can approximate the optimal action-value function with the neural network by reducing the TD error δ_t continuously:

$$\delta_t = r_{t+1} + \gamma \max_{u'} Q(x_{t+1}, u') - Q(x_t, u_t) \tag{9}$$

The objective of the neural network is to minimize the following expression:

$$E = \frac{1}{2}\delta_t^2 \tag{10}$$

The weight-update rule for the neural-network-based gradient-descent method is given by:

$$\omega(t+1) = \omega(t) - \eta \frac{\partial E}{\partial \omega} \tag{11}$$

$$\frac{\partial E}{\partial \omega} = \sigma_t \frac{\partial \delta_t}{\partial \omega} = -\delta_t \frac{\partial Q(x_t, u_t)}{\partial \omega} \tag{12}$$

Combining the above two equations, we can obtain:

$$\omega(t+1) = \omega(t) + \eta \delta_t \frac{\partial Q(x_t, u_t)}{\partial \omega} \tag{13}$$

We can obtain $\frac{\partial Q(x_t, u_t)}{\partial \omega}$ based on the chain rules for $\omega(40+i)$ and $\omega(j-1, i)$:

$$\begin{aligned} \frac{\partial Q(x_t, u_t)}{\partial \omega(40+i)} &= \frac{\partial Q(x_t, u_t)}{\partial V} \frac{\partial V}{\partial \omega(40+i)} = f'(V) y(i) \\ &= y(i) Q(x_t, u_t)[1 - Q(x_t, u_t)] \end{aligned} \tag{14}$$

(for i = 1,…, 10)

$$\begin{aligned} \frac{\partial Q(x_t, u_t)}{\partial \omega(j-1, i)} &= \frac{\partial Q(x_t, u_t)}{\partial V} \frac{\partial V}{\partial y(i)} \frac{\partial y(i)}{\partial a(i)} \frac{\partial a(i)}{\partial \omega(j-1, i)} \\ &= f'(V) y(i) \omega(40+i) f'(a(i)) u(j) \\ &= \omega(40+i) u(j) Q(x_t, u_t)[1 - Q(x_t, u_t)] y(i)[1 - y(i)] \end{aligned} \tag{15}$$

(for i =1,…,10; j =1,…,4)

also obtaining $\frac{\partial Q(x_t, u_t)}{\partial u}$:

$$\begin{aligned} \frac{\partial Q(x_t, u_t)}{\partial u} &= \frac{\partial Q(x_t, u_t)}{\partial V} \sum_{i=1}^{10} \left(\frac{\partial V}{\partial y(i)} \frac{\partial y(i)}{\partial \omega(i)} \frac{\partial \omega(i)}{\partial u} \right) \\ &= f'(V) \sum_{i=1}^{10} (w_i^{(1)} f'(\omega(i)) w_{i,4}^{(2)}) \\ &= Q(x_t, u_t)[1 - Q(x_t, u_t)] \times \sum_{i=1}^{10} (w_i^{(1)} w_{i,4}^{(2)} y(i)[1 - y(i)]) \\ \frac{\partial Q(x_t, u_t)}{\partial u} &= \frac{\partial Q(x_t, u_t)}{\partial V} \sum_{i=1}^{10} \left(\frac{\partial V}{\partial y(i)} \frac{\partial y(i)}{\partial a(i)} \frac{\partial a(i)}{\partial u} \right) \\ &= f'(V) \sum_{i=1}^{10} [\omega(40+i) f'(a(5)) \omega(30+i)] \\ &= Q(x_t, u_t)[1 - Q(x_t, u_t)] \times \sum_{i=1}^{10} (\omega(40+i) \omega(30+i) y(i)[1 - y(i)]) \end{aligned} \tag{16}$$

where the control output of FIS is the fourth input of the neural network.

3.2. Fuzzy Interface System (FIS) Parameters Online Tuning-Based on Q(x,u) (FPT)*

This section focuses on how to tune the parameters of the fuzzy controller based on the approximated *Q*(*x,u*) obtained from the previous section. In order to optimize the output of the FIS, update the parameters of the FIS to maximize the action value function *Q*(*x,u*) with respect to the control output *u* for the current state. We can tune the parameters of FIS using gradient rules:

$$\xi(t+1)=\xi(t)+\beta\frac{\partial Q(x_t,u_t)}{\partial \xi} \tag{17}$$

$$\frac{\partial Q(x_t,u_t)}{\partial \xi}=\frac{\partial Q(x_t,u_t)}{\partial u}\frac{\partial u}{\partial \xi} \tag{18}$$

where, ξ is the parameter to be tuned in FIS such as K^l_j, c^l_i, and δ^l_i. We have obtained $[\partial Q(x_t,u_t)]/\partial u$ already through Equation (16), thus will only need to deduce $\partial u/\partial \xi$.

The Sugeno-type fuzzy inference system is chosen in our FQL control strategy. If the state vector is represented by $x=(x_1,x_2,...,x_n)^T\in R^n$ and the control output $u\in R$, the IF-THEN rules of the fuzzy controller may be expressed as:

$$\begin{aligned} &\mathrm{R}_l : IF\ x_1\ is\ F^l_1,...,and\ x_n\ is\ F^l_n \\ &\qquad THEN\, u=K^l_0+K^l_1x_1+K^l_2x_2+...+K^l_nx_n \end{aligned} \tag{19}$$

where F_i^l is the label of the fuzzy set in x_i, for l = 1, 2, …, M. K^l_0, K^l_1, K^l_2,...and K_n^l are the constant coefficients of the consequent part of the fuzzy rule. We use product inference for the fuzzy implication, singleton fuzzifier, and center-average defuzzifier, respectively. The final output value is:

$$u(\underline{x})=\frac{\sum_{l=1}^{M}((\prod_{i=1}^{n}\mu^{F^l_i}(x_i))\cdot(\sum_{j=0}^{n}K^l_jx_j))}{\sum_{l=1}^{M}(\prod_{i=1}^{n}\mu^{F^l_i}(x_i))} \tag{20}$$

where $\mu^{\mathrm{F}^l_i}$ is the membership degree of the fuzzy set F^l_i, $x_0=1$.

A Gaussian function is used as the membership function of the fuzzy system, *i.e.*:

$$\mu^{F^l_i}(x_i)=e^{-\frac{(x_i-C^l_i)^2}{(\sigma^l_i)^2}} \tag{21}$$

for i = 1, 2, …, n (n is the number of input variable) and l = 1, 2, …, M (M is number of fuzzy rule).

Now we know the parameters that need to be tuned, *i.e.*, c and σ, in our proposed Sugeno-type FIS.

If we let:

$$z^l=\prod_{i=1}^{n}\exp(-(\frac{x_i-c^l_i}{\sigma^l_i})^2) \tag{22}$$

Equation (22) represents the product of different input membership functions in one fuzzy rule:

$$\overline{y}^l=K^l_0+K^l_1x_1+K^l_2x_2+...+K^l_nx_n \tag{23}$$

Equation (23) represents the output of one fuzzy rule:

$$a = \sum_{l=1}^{M} (\overline{y}^{l} z^{l}), b = \sum_{l=1}^{M} z^{l}, u = \frac{a}{b} \tag{24}$$

where, *a*, *b*, and *u* represents the weighted summation, summation of weight of M rules, and total output respectively.

Thus we can calculate $\partial u / \partial \xi$ by the following equations:

$$\frac{\partial u}{\partial K_j^l} = \frac{z^l}{b} x_j \tag{25}$$

$$\frac{\partial u}{\partial c_i^l} = \frac{\partial u}{\partial z^l} \cdot \frac{\partial z^l}{\partial c_i^l} = \frac{\overline{y}^l - u}{b} z^l \frac{2(x_i - c_i^l)}{(\sigma_i^l)^2} \tag{26}$$

$$\frac{\partial u}{\partial \sigma_i^l} = \frac{\partial u}{\partial z^l} \cdot \frac{\partial z^l}{\partial \sigma_i^l} = \frac{\overline{y}^l - u}{b} z^l \frac{2(x_i - c_i^l)^2}{(\sigma_i^l)^2} \tag{27}$$

3.3. Exploration Policy and Action Modifier

Witkins has shown that $Q(x,a)$ converges to $Q^*(x,a)$ with a probability 1, if all actions continue to be tried from all states [22]. In order to guarantee all actions to be tried, we implemented an exploration policy for the control output u recommended by the FIS. The action exploration modifier (AEM) is introduced to generate the control command u_c. The u_c is the sum of u and an additive disturb action u_d, which has a normal distribution with the mean equal to zero and the standard deviation $\sigma_Q(t)$ recommended by the FIS. The AEM can solve the dilemma of "exploration" in reinforcement learning, and is added after the FIS and before the system input, *i.e.*, $u_c = u + u_d$, and u_d ~N(0,σ_Q(t)).

The σ_Q(t) calculated as follows:

$$\sigma_Q(t) = k / [1 + 2\exp(\max Q(x,a))] \tag{28}$$

where *k* is coefficient, which can expand or shrink the disturb action.

3.4. Overall Implementation Procedure

The detailed implementation procedure is presented as follows.

1) Initialize $Q(x_t,u_t)$, the parameters $\omega(1)$–$\omega(40)$, $\omega(41)$–$\omega(50)$ of the QEN, and the parameters ξ of the FIS.
2) Obtain the new control output u_t based on (20) and input of the FIS.
3) Before it is fed to the actual system, *u* is processed by the action modifier according to $u_c = u + u_d$.
4) The action modifier provides u_c, which acts as the control value of the system.
5) Based on our requirements for the system, we evaluate the performance of the controller as r and obtain the states of the system.
6) Obtain the approximated $Q(x_{t+1}, u_{t+1})$ from the QEN based on the current control action, and current states, and some previous states.

7) From r, $Q(x_t,u_t)$, and $Q(x_{t+1}, u_{t+1})$, we can calculate the TD error δ_t based on Equation (11). Here, we assume $Q(x_{t+1}, u_{t+1}) \approx \max_{u'} Q(x_{t+1}, u')$ because u_{t+1} is obtained from the FIS, which continuously maximizes $Q(x_t,u_t)$ with respect to the control output u.
8) Based on δ_t obtained from Step 7, we can update the parameters of the QEN according to Equations (14) and (15).
9) Tune the parameters of the FIS based on Equations (17)–(27).
10) Substitute $Q(x_t,u_t)$ with $Q(x_{t+1},u_{t+1})$.
11) If the parameters of the QEN and the FIS are not changed any more or after predefined iterations, the learning procedure is terminated; otherwise, return to Step 2 after a fixed sampling time Δ.

4. Simulation Results and Discussion

In order to know the effectiveness of the FQL algorithm, simulation experiments were done in ADVISOR. Using a simulation to test the algorithm in a variety of driving cycles can eliminate the huge cost and time needed for actual experimentation. The simulation model for the HEV mentioned in section 2 was built in ADVISOR and is shown in Figure 4.

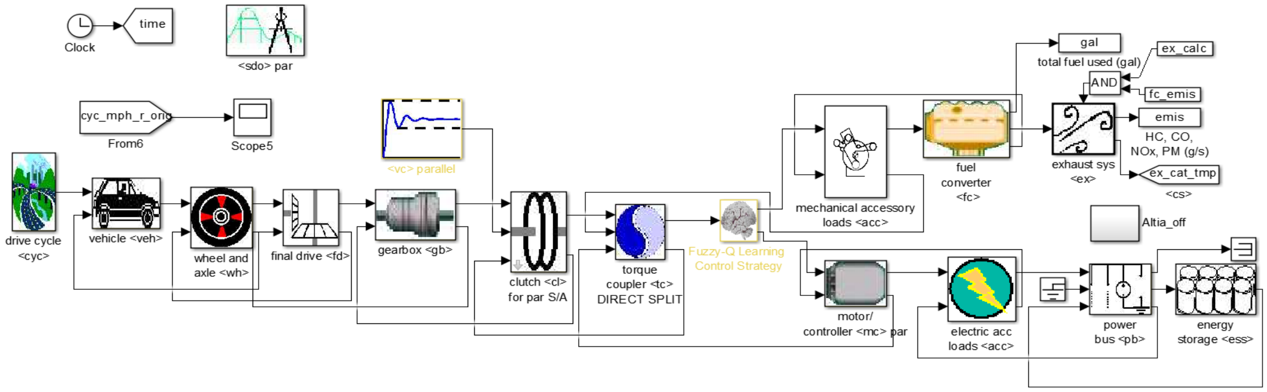


Figure 4. HEV model in ADVISOR.

For this particular HEV system, the parameters of the algorithm used in the simulations are summarized in Table 2, with proper notations defined in it.

Table 2. Summary of FQL algorithm parameters.

Parameter	Value
Number of input nodes in QEN	4
Number of hidden nodes in QEN	10
Learning rate of QEN η	0.34
Rate of Gradient descent β	0.32
Coefficient of AEM h	0.40
Discount factor γ	0.90
Emission cost weight a_1	0
SOC deviation cost weight a_2	1
Coefficient of $\sigma_Q(t)$ k	0.41

The next step is to define cost function *R*(*k*). The reason for the selection of $a_1 = 0$ is simply because the emission maps are not provided for the engine. Thus, the resultant control strategy is a fuel-economy only strategy. In order to consider the power economy influence on fuel economy, we let a_2 equals 1:

$$R(k) = R_{\text{fuel}}(k) + a_1 R_{\text{ems}}(k) + a_2 R_{\text{SOC}}(k) \tag{29}$$

$\mathrm{R}_{\text{fuel}}(k)$ and $\mathrm{R}_{\text{SOC}}(k)$ can be defined as follows:

$$\mathrm{R}_{\text{fuel}}(k) = \begin{cases} -1, & x > 0.88g/s, \\ -0.5, & 0.53g/s < x < 0.88g/s, \\ 0.5, & 0 < x < 0.53g/s. \end{cases} \tag{30}$$

$$\mathrm{R}_{\text{SOC}}(k) = \begin{cases} -0.3, & y < 0, \\ 0, & 0 < y < 0.001, \\ 0.3, & y > 0.001. \end{cases} \tag{31}$$

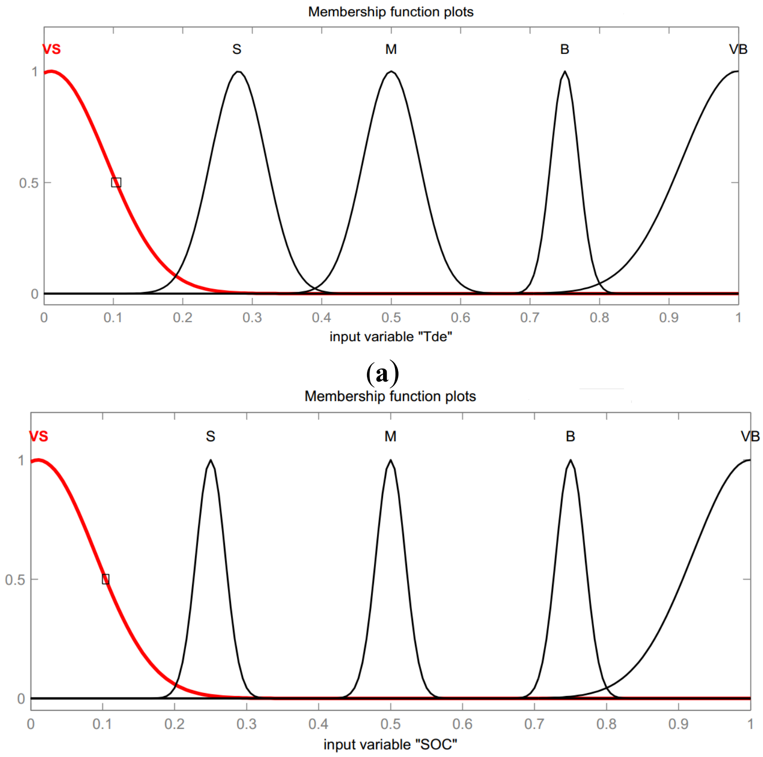
where *x* is the instantaneous fuel consumption value and *y* the *SOC* change rate.

The FQL algorithm was written in MATLAB. The fuzzy rules were predesigned according to engineering experience, and complete rules details are given in Table 3, where the parameter of T_e satisfies the relationship: VS < S < M < B < VB < SC < MC < BC, the parameters of T_{dem} and *SOC* satisfy the relationship: VS < S < M < B < VB.

Table 3. Summary of fuzzy rules.

***T*~e~**		***T*dem**				
		VS	**S**	**M**	**B**	**VB**
SOC	VS	BC	BC	MC	SC	VB
	S	BC	MC	MC	SC	VB
	M	SC	SC	M	M	B
	B	S	S	M	B	B
	VB	VS	VS	S	S	M

Initially, the membership functions of the fuzzy controller were randomly initialized. In order to illustrate the control strategy more clearly, a convenient method is applied to represent it in an intuitive manner. A torque-split-ratio (TSR) $\tau = T_e / T_{\text{dem}}$ is defined to quantify the positive power flows in the powertrain [23]. Four positive power operation modes are defined, including motor only ($\tau = 0$), engine only ($\tau = 1$), power-assist ($0 < \tau < 1$), and charging mode ($\tau > 1$). Figures 5 and 6 show the initial membership functions and TSR map for initial fuzzy controller.



(**a**)

(**b**)

Figure 5. (**a**) Initial membership functions of T_{dem}; and (**b**) initial membership functions of *SOC*.

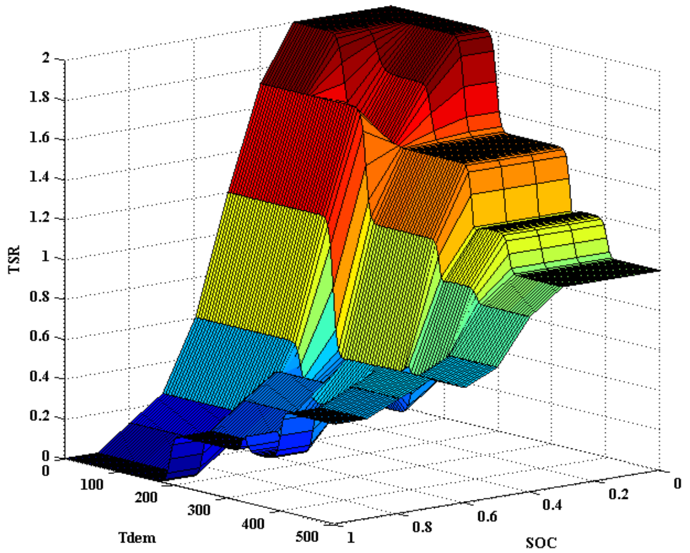
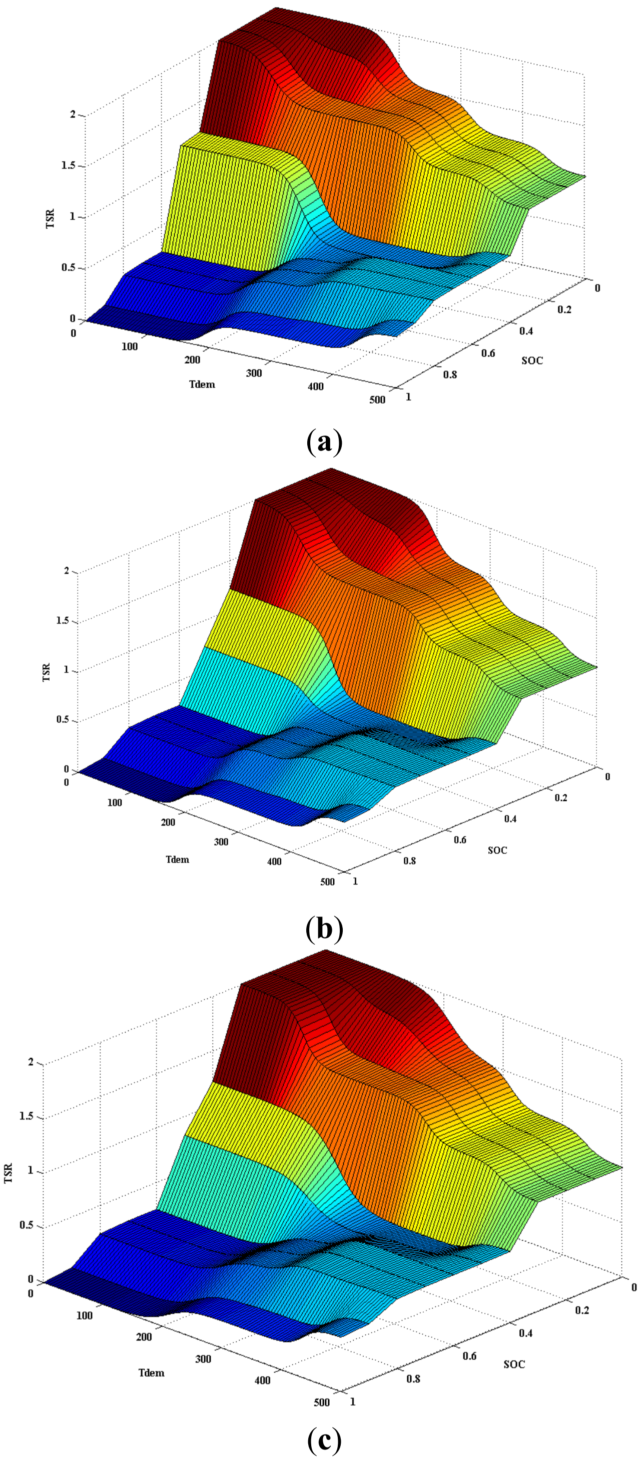


Figure 6. Torque-split-ratio (TSR) map for the initial fuzzy controller.

A. Simulation under Urban Dynamometer Driving Schedule (UDDS)

Simulation test was done under standard driving cycle UDDS. Figure 7 depicts the changing trend of the TSR map for fuzzy controller during the driving cycle. From 0 s to 1369 s, we can see that the surface of the TSR map is becoming smoother. The online learning of the fuzzy Q learning control strategy is the reason behind the TSR map smoothing. Figure 8 shows the final membership functions of the fuzzy controller.

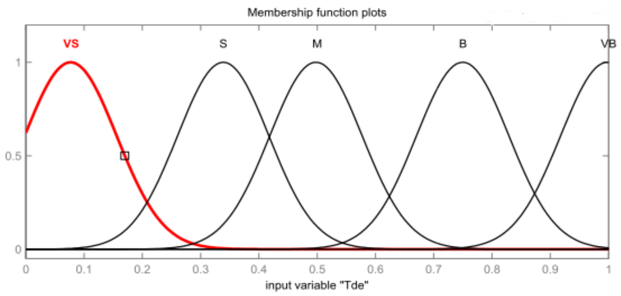


(**a**)

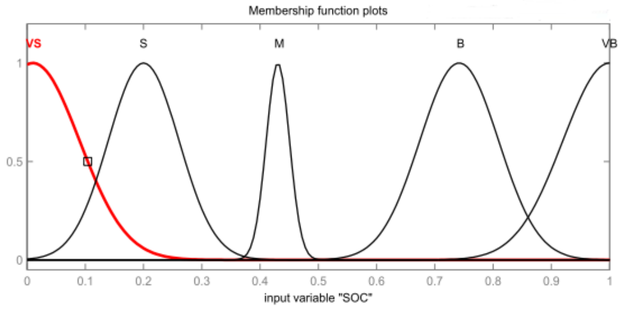
(**b**)

(**c**)

Figure 7. Changing trend of fuzzy controller TSR map under Urban Dynamometer Driving Schedule (UDDS): (**a**) TSR map for fuzzy controller in 500 s; (**b**) TSR map for fuzzy controller in 1000 s; and (**c**) TSR map for fuzzy controller in 1369 s.



(**a**)



(**b**)

Figure 8. (**a**) Final membership functions of T_{dem} under UDDS; and (**b**) initial membership functions of *SOC* under UDDS.

The simulation results for the UDDS driving cycles are shown in Figure 9. The FQL control strategy tends to maintain the battery *SOC* near 50%, finally. This leaves enough capacity to handle an extended period of battery discharge and enough capacity to absorb a long period of charging.

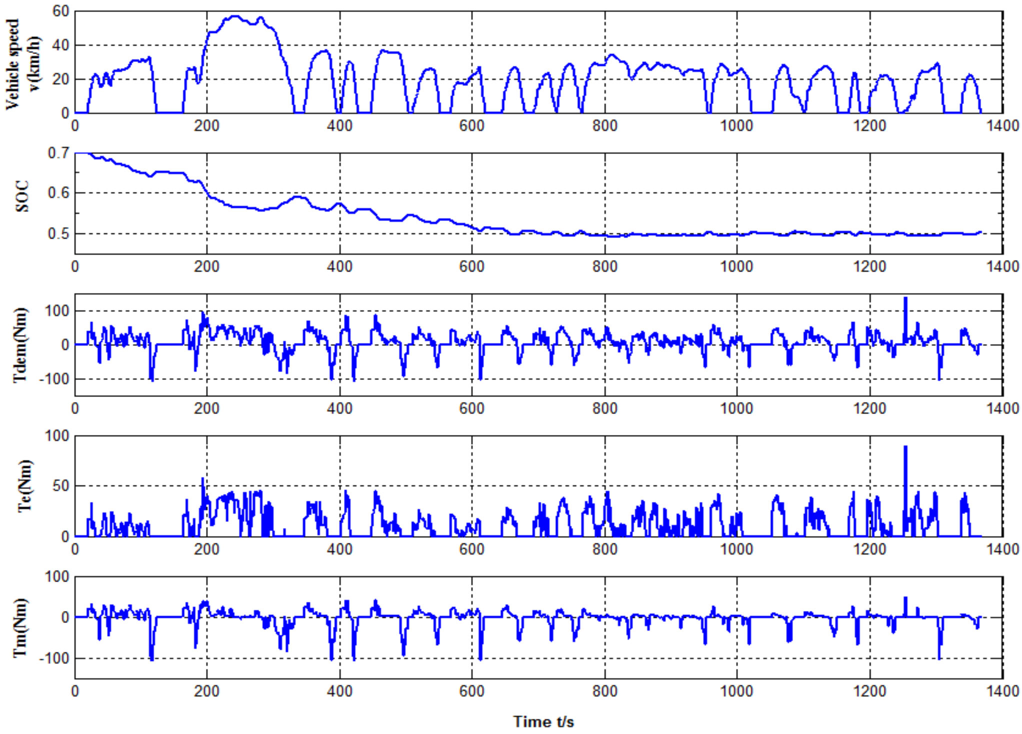


Figure 9. Simulation results under UDDS.

In order to evaluate the performance and effectiveness of the FQL control strategy, the experiment results are compared with a heuristic rule-based control strategy known as "Parallel Electric Assist Control Strategy" and the fuzzy logic control strategy. The comparison results are listed in Table 4. Power consumption is converted to fuel consumption; equivalent fuel consumption is obtained by adding the converted power consumption and fuel consumption. As shown by the results of Table 4, equivalent fuel consumption of fuzzy control is decreased by 3.10% compared with the rule-based control strategy. Meanwhile, the equivalent fuel consumption of the FQL is decreased by 2.67% compared with the fuzzy control strategy. The FQL control strategy achieves good performance.

Table 4. Compare results under UDDS.

Control strategy	Fuel consumption	Equivalent fuel consumption
Rule-based (L/100 km)	3.88	3.87
Fuzzy control (L/100 km)	3.67	3.75
FQL (L/100 km)	3.48	3.65

Figures 10 and 11 depict the distribution of engine and motor operating points under the rule-based control strategy and FQL control strategy. The FQL control strategy is a fuel strategy which limits the instantaneous fuel consumption. This strategy is not based on the efficiency of the engine, but it primarily limits the fuel use to a particular value. As shown in Figure 10, most of the engine operation points under the FQL control strategy are below the 0.3 g/s fuel use line, while a great amount of engine operation points under the rule-based control strategy are below the 0.55 g/s fuel use line. That means the instantaneous fuel consumption of the FQL control strategy is less than the rule-based control strategy most of the time. The motor operation point distribution of the FQL control strategy is shown in Figure 11. The efficiency of motor operating under the FQL control strategy is better than the rule-based control strategy.

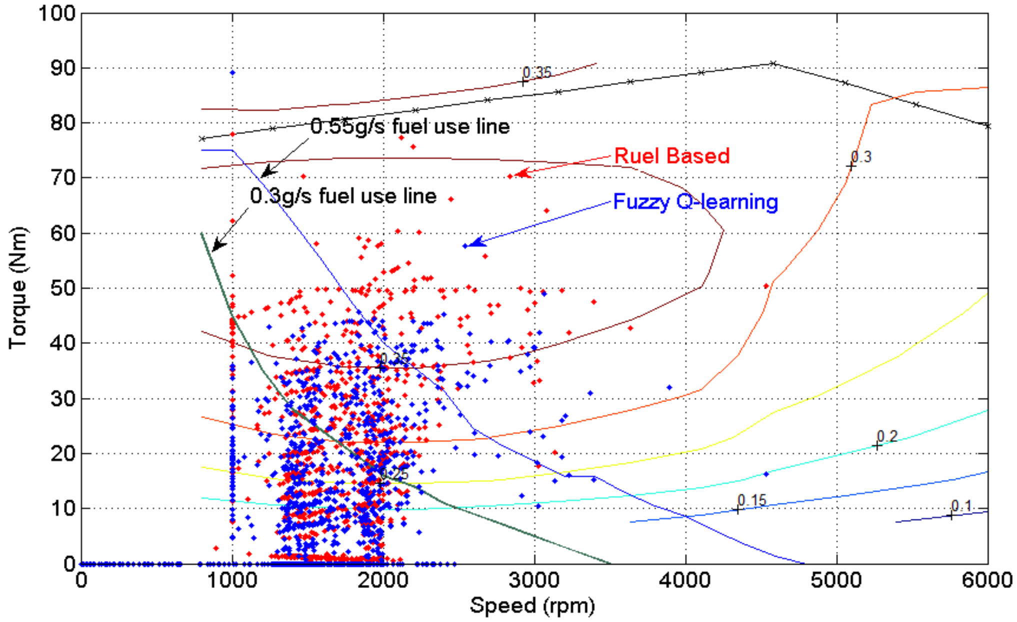


Figure 10. Engine operation point distribution under UDDS.

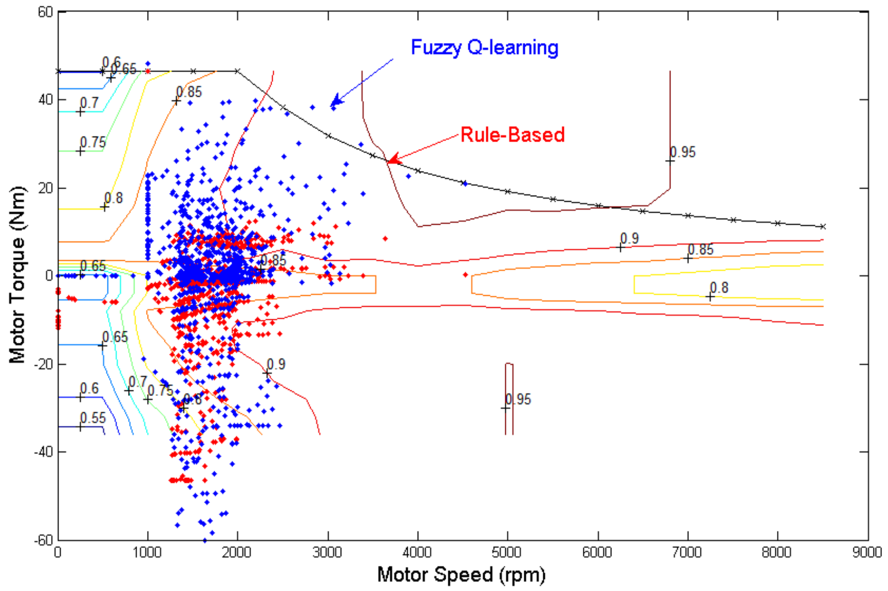


Figure 11. Motor operation point distribution under UDDS.

Torque split trajectory by using the FQL control strategy is shown in Figure 12. In order to illustrate the torque split trajectory, we choose 160–320 s period from the driving cycle. It is illustrated that the engine provides most of the torque demand, while the motor helps when more torque is needed. The figure also depicts a relatively smooth profile of the engine torque compared with the demand torque and the motor torque. The smoother engine torque from the fuzzy Q learning control strategy indicates that it helps improve the operating conditions of the engine.

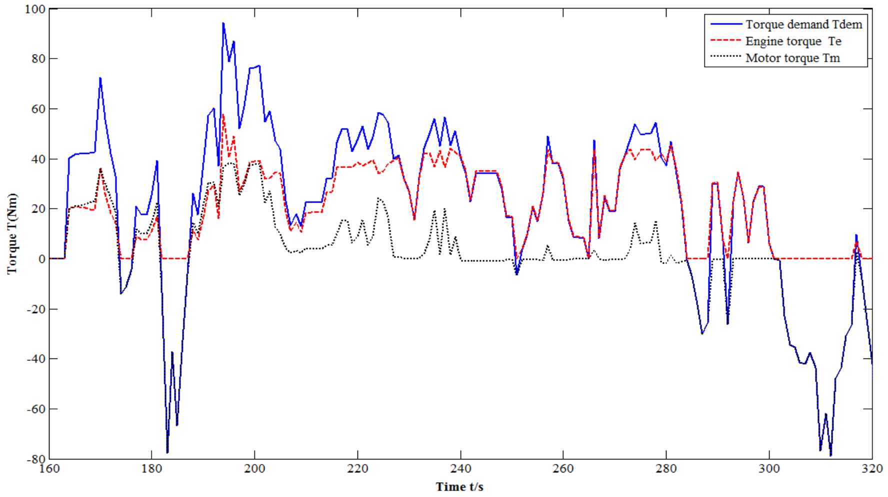
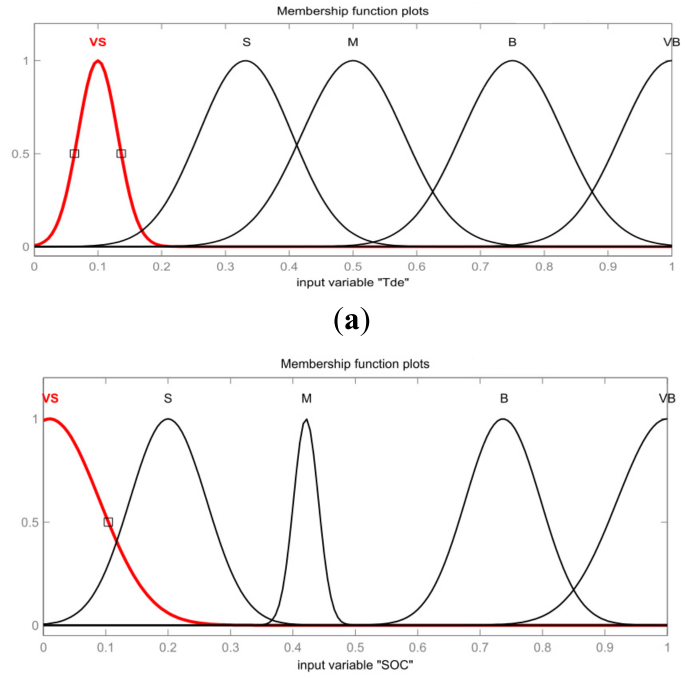


Figure 12. Torque split trajectory by using the FQL control strategy under UDDS.

B. Simulation under New European Driving Cycle (NEDC)

In order to check the effectiveness of the proposed method, it is tested on different driving conditions (NEDC driving cycles) starting from the same initial conditions (the same parameters for both neural network and fuzzy controller). Figure 13 show the final membership functions of the FQL control strategy under NEDC cycle are different from that under UDDS cycle; as a result we get two different controllers. That is, the control strategy really learns from the environment.



(**a**)

(**b**)

Figure 13. (**a**) Final membership functions of T_{dem} under New European Driving Cycle (NEDC); (**b**) Initial membership functions of *SOC* under NEDC.

Simulation results for the NEDC driving cycles are shown in Figure 14. We can see that the *SOC* also maintains near 50%, finally. The rules of the fuzzy controller have a significant effect on the final value of *SOC* to make sure that *SOC* is maintained near 50%, and the proposed method in this paper only changes the parameters of the membership function of the fuzzy controller during the cycle. Table 5 shows the comparison between fuel consumption and equivalent fuel consumption with different control strategies, and we can see the FQL control strategy also achieves good performance.

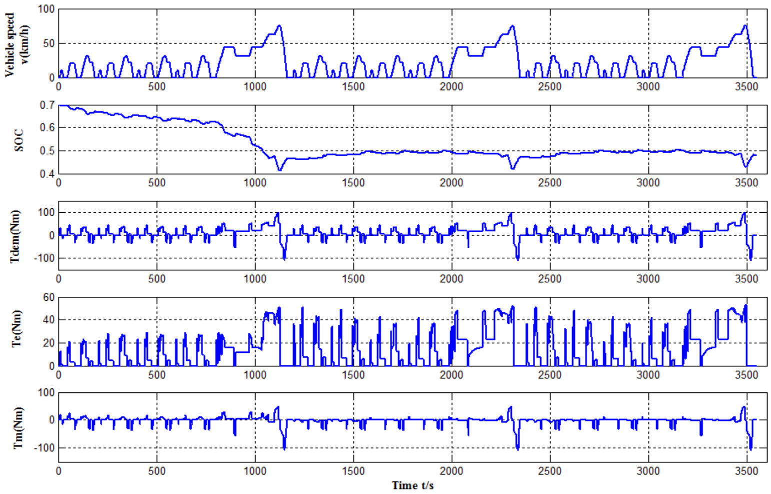


Figure 14. Simulation results under NEDC.

Table 5. Compare results under NEDC.

Control strategy	Fuel consumption	Equivalent fuel consumption
Rule-based (L/100 km)	3.90	3.92
Fuzzy control (L/100 km)	3.67	3.79
FQL (L/100 km)	3.43	3.66

5. Conclusions

An online learning control strategy based on FQL has been proposed to improve the fuel economy for a hybrid electric vehicle. The FQL control strategy contains two parts: QEN and FPT. We used a BP neural network as QEN to estimate $Q^*(x,u)$. For the fuzzy controller, we chose a Sugeno-type fuzzy controller and the parameters of the fuzzy controller were tuned online based on $Q^*(x,u)$. The action exploration modifier (AEM) is introduced to guarantee all actions are tried. Simulation results indicate that the parameters of the FIS are tuned online and the FQL control strategy achieves good performance in fuel economy.

Acknowledgments

This research is supported by National Natural Science Foundation of China (61273139), Natural Science Foundation of Shandong (ZR2014FP001).

Author Contributions

Yue Hu designed the FQL control strategy in matlab and wrote the main parts of the manuscript. Weimin Li proposed the method of FQL control strategy and checked the whole manuscript. Hui Xu did simulation experiments and wrote the part of Simulation results and discussion. Guoqin Xu checked the results and the whole manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Pisu, P.; Rizzoni, G. A comparative study of supervisory control strategies for hybrid electric vehicles. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 506–518.
2. Wirasingha, S.G.; Emadi, A. Classification and review of control strategies for plug-in hybrid electric vehicles. *IEEE Trans. Veh. Technol.* **2011**, *60*, 111–122.
3. Li, C.; Liu, G. Optimal fuzzy power control and management of fuel cell/battery hybrid vehicles. *J. Power Sources* **2009**, *192*, 525–533.
4. Odeim, F.; Roes, J.; Lars, W.; Angelika, H. Power management optimization of fuel cell/battery hybrid vehicles with experimental validation. *J. Power Sources* **2014**, *252*, 333–343.
5. Zhang, C.; Vahidi, A.; Pisu, P.; Li, X.; Tennant, K. Role of terrain preview in energy management of hybrid electric vehicles. *IEEE Trans. Veh. Technol.* **2010**, *59*, 1139–1147.

6. Lu, S.; Hillmansen, S.; Roberts, C. A power-management strategy for multiple-unit railroad vehicles. *IEEE Trans. Veh. Technol.* **2011**, *60*, 406–420.
7. Zheng, C.; Chris, C.M.; Xiong, R.; Xu, J.; You, C. Energy management of a power-split plug-in hybrid electric vehicle based on genetic algorithm and quadratic programming. *J. Power Sources* **2014**, *248*, 416–426.
8. Zhang, C.; Vahidi, A. Route preview in energy management of plug-in hybrid vehicles. *IEEE Trans. Control Systems Technol.* **2012**, *20*, 546–553.
9. Han, J.; Park, Y.; Kum, D. Optimal adaption of equivalent factor of equivalent consumption minimization strategy for fuel cell hybrid electric vehicles under active state inequality constraints. *J. Power Sources* **2014**, *267*, 491–502.
10. Ansarey, M.; Masoud, S.P.; Hussein, Z.; Mohammad, M. Optimal energy management in a dual-storage fuel-cell hybrid vehicle using multi-dimensional dynamic programming. *J. Power Sources* **2014**, *250*, 359–371.
11. Chen, H.; Wu, G.; Luo, L.; Tan, W. Simulation of hybrid electric vehicle control strategy based on compensation fuzzy neural network. In Proceedings of the 7th Intelligent Control and Automation World Congress, Chongqing, China, 25–27 June 2008; pp. 8697–8701.
12. Meng, X.; Langlois, N. Optimized fuzzy logic control strategy of hybrid vehicles using ADVISOR. In Proceedings of the IEEE Computer, Mechatronics, Control and Electronic Engineering (CMCE) International Conference, Changchun, China, 24–26 August 2010; pp. 444–447.
13. Chen, R.; Li, C.; Meng, X.; Yu, Y. The application of fuzzy-neural network on control strategy of hybrid vehicles. In Proceeding of the 27th Chinese Control Conference, Kunming, China, 16–18 July 2008; pp. 281–284.
14. Bichi, M.; Ripaccioli, G.; Di, C.S.; Bemardini, D.; Bemporad, A.; Kolmanovsky, I.V. Stochastic model predictive control with driver behavior learning for improved powertrain control. In Proceeding of the 49th IEEE Decision and Control Conference, Atlanta, GA, USA, 15–17 December 2010; pp. 6077–6082.
15. Borhan, H.; Vahidi, A.; Phillips, A.M.; Kuang, M.L.; Kolmanovsky, I.V.; Di, C.S. MPC-based energy management of a power-split hybrid electric vehicle. *IEEE Trans. Control Syst. Technol.* **2012**, *20*, 593–603.
16. Bubna, P.; Brunner, D.; Advani, S.G.; Prasad, A.K. Prediction-based optimal power management in a fuel cell/battery plug-in hybrid vehicle. *J. Power Sources* **2010**, *195*, 6699–6708.
17. Santucci, A.; Sorniotti, A.; Constantina, L. Power split strategies for hybrid energy storage systems for vehicular applications. *J. Power Sources* **2014**, *258*, 395–407.
18. Bordons, C.; Ridao, M.A.; Pérez, A.; Marcos, D. Model predictive control for power management in hybrid fuel cell vehicles. In Proceedings of the IEEE Vehicle Power and Propulsion Conference, Lille, France, 1–3 September 2010; pp. 1–6.
19. Yan, Y.; Xie, H. Model predictive control for series-parallel plug-in hybrid electrical vehicle using GPS system. In Proceedings of Electrical and Control Engineering International Conference, Yichang, China, 16–18 September 2011; pp. 2334–2337.
20. Li, W.; Xu, G.; Xu, Y. Online learning control for hybrid electric vehicle. *Chin. J. Mech. Eng.* **2012**, *25*, 98–106.

21. Dai, X.; Li, C.K.; Ahmad, B.R. An approach to tune fuzzy controllers based on reinforcement learning for autonomous vehicle control. *IEEE Trans. Intell. Transp. Syst.* **2005**, *6*, 285–293.
22. Watkins, C. Learning from delayed rewards. Ph.D. Thesis, University of Cambridge, Cambridge, UK, 1989.
23. Lin, C.C.; Peng, H.; Grizzle, J.M.; Kang, J.M. Power management strategy for a parallel hybrid electric truck. *IEEE Trans. Control Syst. Technol.* **2003**, *11*, 839–849.