DOI: 10.3901/CJME.2011.**.***, available online at www.cjmenet.com; www.cjmenet.com.cn

Online Learning Control for Hybrid Electric Vehicle

LI Weimin[1, *], XU Guoqing[2, 3], and XU Yangsheng[3]

1 *Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China*
2 *Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China*
3 *Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, China*

Received May 12, 2010; revised October 9, 2010; accepted July, 2011; published electronically July, 2011

Abstract: Improvements in hybrid electric vehicle (HEV) fuel economy and emissions heavily depend on an efficient energy management strategy (EMS). However, the uncertainty of future driving conditions generally cannot be easily tackled in EMS design. Most existing EMSs act upon fixed parameters and cannot adapt to varying driving conditions. Therefore, they usually fail to fully explore the potential of these advanced vehicles. In this paper, a novel EMS design procedure based on neural dynamic programming (NDP) is proposed. The NDP is a generic online learning algorithm, which combines stochastic dynamic programming (SDP) and the temporal difference (TD) method. Instead of computing the utility function and optimal control actions through Bellman equations, the NDP algorithm uses two neural networks to approximate them. The weights of these neural networks are updated online by the TD method. It avoids the high computational cost that SDP suffers from and is suitable for real-time implementation. The main advantages of NDP EMS is that it does not rely on prior information related to future driving conditions, and can self-tune with a wide variance in operating conditions. The NDP EMS has been applied to "Qianghua-I", a prototype of a parallel HEV, using a revolving drum test bench for verification. Experiment results illustrate the potential of the proposed EMS in terms of fuel economy and in keeping state of charge (SOC) deviations at a low level. The proposed research ensures the optimality of NDP EMS, as well as real-time applicability.

Key words: hybrid electric vehicle, neural dynamic programming, energy management strategy

1 Introduction

Hybrid electric vehicles (HEVs) have been widely studied in recent years because of their potential to significantly improve fuel economy and reduce emissions. They have been regarded as a commercially viable alternative to either traditional vehicles or electric vehicles. HEV design combines an engine and a motor together with an energy storage that can act independently or cooperatively. Consequently, it can significantly reduce fuel consumption by operating the engine at an optimum efficiency range. Additionally, hybridization advantages, such as energy recovery during breaking, further improve fuel economy. However, due to the complexity of the hybrid structures and the uncertainty of future driving conditions, it is non-trivial to design an efficient real-time energy management strategy (EMS).

Many existing EMSs are rule-based, such as the Thermostatic strategy, the Power Follow strategy, and the Parallel Hybrid Electric Assist strategy[1]. These EMSs have been developed based on the results of extensive experimental trials and human expertise. Some other EMSs employ heuristic control techniques, with the resultant strategies formalized as fuzzy rules[2–3]. Though these rule-based strategies are effective and can be easily implemented, their optimality and flexibility are critically limited by working conditions. Therefore, an EMS that performs well under certain conditions may be not satisfactory under other conditions.

According to the literature, to optimize the operation of the HEV powertrain, some model-based global optimization methods have been employed in EMS design, such as dynamic programming (DP)[4–5], sequential quadratic programming (SQP)[6], genetic algorithms (GA)[7], 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. But the optimal power-split solutions obtained are only optimal with respect to that specific drive 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 implement these control laws directly. More critically, these methods suffer from the "curse of dimensionality" problem, which prevents their wide adoption in real-time applications. In summary, EMS designs built upon global optimization techniques can serve to evaluate the potential fuel economy of a given powertrain configuration, as well as the optimality of realizable control strategies.

How to gather the necessary information about future

* Corresponding author. E-mail: liweimin@sjtu.edu.cn
This project is supported by Innovation Technology Fund of the Hong Kong Special Administrative Region of China (Grant No. GHP/011/05)

driving conditions remains an open question. In the literature, statistical approaches are often employed to address this problem. For example, LIN, et al[8–9], presented an EMS design procedure based on stochastic dynamic programming (SDP). In their approach, a Markov process that represents future uncertainty under diverse driving conditions is applied to model the driver's power demand. The transition probability matrix of the power demand from the driver can be obtained from the driving cycles or the collected history data of driving operation. By doing so, the SDP EMS is optimized over a family of driving cycles that follow the same transition probability matrix in an average sense. The SDP problem can in principle be solved by the value iteration algorithm or the policy iteration algorithm[10]. The resultant EMS takes the form of a stationary full-state feedback control law that maps the current state to actions. However, the relevant techniques for the SDP problem are also time-consuming and highly computational. Moreover, they rely on the exact knowledge of the transition probability matrix, which may be difficult, if not impossible, to obtain. Therefore, the SDP algorithms cannot be readily applied in real-time applications.

A real-time EMS should adapt itself to varying driving conditions in order to get good performance. Hence, a learning control approach based on neural dynamic programming (NDP) is presented here. The NDP is a generic online learning algorithm which hybridizes the SDP and temporal difference (TD) reinforcement learning[11–12]. Instead of computing the utility function and optimal control actions through solving Bellman equations, the NDP algorithm uses two neural networks to approximate them. This approximation approach generally produces a suboptimal solution with a substantially reduced computational cost. Basically, the developed NDP EMS controls the system by learning its characteristics and updating the neural network weights simultaneously in real time. Consequently, the NDP EMS can tune itself adaptively to widely changing operating conditions. The RBF neural network is utilized here to implement the NDP algorithm due to its fast convergence. Furthermore, the NDP approach does not need an explicit environmental model, such as the transition probability matrix of the power demand in the SDP approach. This feature extensively enlarges its application scope.

The problem formulation and the HEV model used for evaluating the EMS are described in section 2. Section 3 presents a comprehensive description of the NDP mechanism. Section 4 provides the details of the NDP design procedure as well as experiment results. Finally, conclusions are given in section 5.

2　Problem Formulation

The baseline vehicle studied is a single axis parallel HEV, Qianghua-I, whose powertrain structure is shown in Fig. 1. The powertrain integrates an engine, an electric traction motor/generator, Ni-Hi batteries, an automatic clutch, and an automatic/manual transmission system. The motor is directly linked between the output of the auto clutch and the input to the transmission. 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.

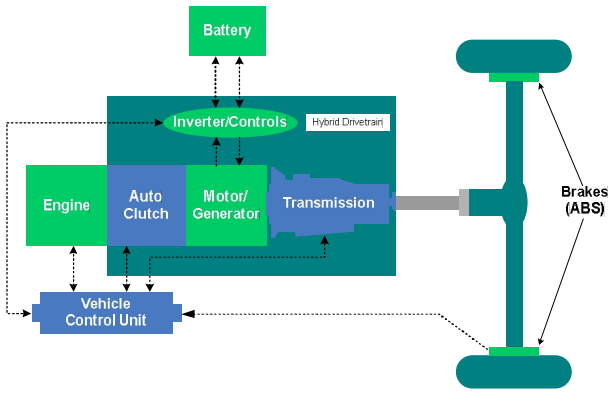


Fig. 1.　Schematic diagram of the parallel hybrid electric vehicle drivetrain

Table 1.　Summary of the HEV parameters

Spark Ignition (SI) engine	Displacement: 1.0 L Maximum power: 50 kW/5 700 r/min Maximum power: 89.5 N • m/5 600 r/min
Permanent magnet motor	Maximum Power:10 kW Maximum Torque:46.5 N • m
Advanced Ni-Hi 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: 1 000 kg

The system model has been developed according to the well-known quasi-static approach. Accordingly, the dynamics of the system are inverted and the torque required at the wheels T_w and the angular velocity of the front axle ω_w can be calculated as

$$\omega_w = \frac{v}{r_w(v)}, \tag{1}$$

$$T_w = r_w(v)\left(\frac{\rho}{2}A_f c_d v^2 + f_r(v)mg\cos\alpha + mg\sin\beta\right) + \frac{J_{tot}}{r_w(v)}\frac{dv}{dt}, \tag{2}$$

for a given vehicle speed v and a known road slope α. The vehicle parameters are the frontal area A_f, the drag coefficient c_d, the rolling resistance coefficient f_r, the vehicle mass m, the wheel radius r_w and the total vehicle inertia J_{tot}. The angular velocity ω_{in} and the torque request

T_{in} at the gearbox input are then given

$$\omega_{in} = \omega_w R(g), \tag{3}$$

$$T_{in} = \begin{cases} \dfrac{T_w + T_{loss}(\omega_w, g)}{R(g)\eta(g)}, & T_w + T_{loss} \geq 0, \\ \dfrac{T_w + T_{loss}(\omega_w, g)}{R(g)}\eta(g), & T_w + T_{loss} < 0. \end{cases} \tag{4}$$

Where T_{loss} denominates additional losses caused by friction, $R(g)$ the total transmission ratio, and $\eta(g)$ the total transmission efficiency from the gearbox input to the front wheels. The corresponding gear g can be calculated by using the specified shift schedules of the automatic gearbox as follows:

$$g(k+1) = \begin{cases} 5, & g(k)+q(k) > 5, \\ 1, & g(k)+q(k) < 1, \\ g(k)+q(k), & \text{otherwise}. \end{cases} \tag{5}$$

The control to transmission $q(k)$ is constrained to take the discrete values of 1, 0, and −1. These values represent the downshift, the keep unchanged, and the up-shift, respectively.

The torque request T_{in} has to be satisfied by the engine and the motor yielding:

$$T_{in} = T_e + T_m. \tag{6}$$

The fuel consumption rate and emissions are assumed to be static functions of the engine speed and the engine torque, and can be obtained by referencing the engine map.

The motor efficiency is modeled as a function of the motor torque and the speed, i.e., $\eta_m = f(T_m, \omega_m)$. The final motor torque output is limited by the battery capacity and motor power limit.

The battery is modeled as a voltage source with an open circuit voltage V_{oc} and an inner resistance R_{int}, both depending on the state of charge (SOC) of the battery, ξ. The evolution equation of the SOC is represented as

$$\xi(k+1) = \xi(k) - \frac{V_{oc} - \sqrt{V_{oc}^2 - 4(R_{int} + R_t)T_m \omega_m \eta_m^{-\mathrm{sgn}(T_m)}}}{2(R_{int} + R_t)Q_{max}}, \tag{7}$$

where Q_{max} is the maximum charge capacity of the battery; R_t is the terminal resistance.

Finally, the state vector of the HEV system includes four state variables, i.e., $\boldsymbol{X}(k) = (T_{dem}(k), \omega_w(k), g(k), \xi(k))^T$. The control vector is $\boldsymbol{U}(k) = (T_e(k), q(k))^T$. The motor output torque command $T_m(k)$ can then be obtained through Eq. (6). As can be seen from the above description, the energy management strategy includes two sub-strategies: 1) the gear shifting strategy, which selects the gear from a discrete set to optimize the operation of the engine, and 2) the torque-split strategy, which defines the best torque split between the engine and the motor.

Formally, the energy management of the HEV is to find the optimal control strategy, π^*, that maps the observed states $\boldsymbol{X}(k)$ to the control action $\boldsymbol{U}(k)$ so as to minimize vehicle fuel consumption and emissions along a transport mission. In the meantime, the vehicle drivability and charge-sustaining of the battery have to be satisfied. Mathematically, the energy management of HEV can be formulated as an infinite horizon dynamic optimization problem as follows:

$$J(\boldsymbol{X}) = \sum_{k=0}^{\infty} \gamma^k R(k), \tag{8}$$

where $R(k)$ is the immediate cost function incurred by $\boldsymbol{U}(k)$ at time k; $\gamma \in (0, 1)$ is a discount factor that assures the infinite sum of cost function convergence. A key benefit of the 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 consumption, emissions, and some other additional cost functions, as shown in Eq. (9):

$$R(k) = R_{fuel}(k) + a_1 R_{ems}(k) + a_2 R_{soc}(k) + a_3 R_{gs}(k). \tag{9}$$

The constraint on the charge-sustaining operation is incorporated in the cost function so that the SOC depletion can also be minimized:

$$R_{soc} = (\xi(k) - \xi_{ref})^2, \tag{10}$$

where ξ_{ref} is the desired SOC at the final time (which is usually equal to the initial SOC in simulation), and a_2 is a positive weight factor.

The gear-shifting schedule is crucial to the fuel economy of the HEV. If no constraint is imposed on gear-shifting frequency, the optimal gear trajectory will result in frequent shifting, and thereby influence comfort and drivability. This additional cost is represented as R_{gs}:

$$R_{gs} = |q(k)|. \tag{11}$$

Finally, the optimization problem is subject to the following system constraints:

$$\begin{cases} \omega_{e\_min} \leq \omega_e(k) \leq \omega_{e\_max}, \\ \xi_{min} \leq \xi(k) \leq \xi_{max}, \\ T_{e\_min}(\omega_e(k)) \leq T_e(k) \leq T_{e\_max}(\omega_e(k)), \\ T_{m\_min}(\omega_m(k), \xi(k)) \leq T_m(k) \leq T_{m\_max}(\omega_m(k), \xi(k)). \end{cases} \tag{12}$$

3 NDP Mechanism

For an infinite horizon optimal control problem, the

utility function J gives the expected accumulated future cost for each state $\boldsymbol{X}$. The optimal control action $\boldsymbol{U}^*$ is the control value that yields the minimum utility function J^*. According to the Bellman equation, there is a direct relationship between the optimal utility function of a state and its neighbors, as shown in Eq. (13):

$$J^*(\boldsymbol{X}(k)) = \min_u \left\{ \gamma J^*(\boldsymbol{X}(k+1)) + R(k) \right\}. \quad (13)$$

For the sake of simplification, $J(\boldsymbol{X}(k))$ are rewritten as $J(k)$ in the following.

The NDP approach uses two neural networks, the critic network and the action network to approximate each state's utility function $J(\boldsymbol{X})$ and the corresponding optimal control action $\boldsymbol{U}(\boldsymbol{X})$, respectively. The whole training process is operated along with the system operation. The states' utility function is updated online by the TD method. Then the experienced system state $\boldsymbol{X}(k)$ and their updated value $J(\boldsymbol{X})$ are used as a sample to adapt the weights of the critic network and the action network. So, the essence of NDP is to utilize the generalization ability of neural networks to approximate all states' $J(\boldsymbol{X})$ and $\boldsymbol{U}(\boldsymbol{X})$ in the whole state space. Although this approximation may cause little degradation in optimality, it can greatly reduce computational cost. The structure of the NDP configuration is shown in Fig. 2. The solid lines represent the signal flow, while the dashed lines are the paths for parameter tuning. TDL denotes a tapped delay line.

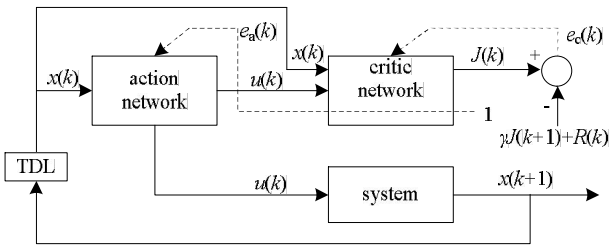


Fig. 2. Schematic diagram of the NDP

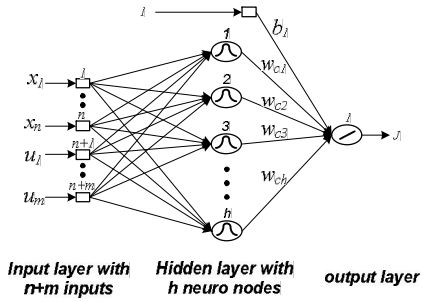
To reduce training time, a three-layer RBF neural network is adopted here for both critic and action networks. As shown in Fig. 3, the input values are each assigned to a node in the input layer and passed directly to the hidden layer without weights. The RBF units in the hidden layer take the Gaussian density function as the activation function, which are specified by a parameter vector $\boldsymbol{C}_i$ (called center), and a scalar β_i (called width). The overall input–output mapping equation of the RBF network is as follows:

$$y_j = b_j + \sum_{i=1}^{h} w_{ji} P_i = b_j + \sum_{i=1}^{h} w_{ji} \exp\left(-\frac{\left\| \boldsymbol{S} - \boldsymbol{C}_i \right\|^2}{\beta_i^2} \right), \quad (14)$$

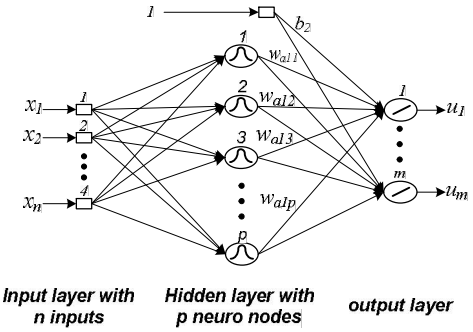
where $\boldsymbol{S}$ is the input vector, h is the number of RBF units in the hidden layer, b_j and w_{ji} are the bias terms and the weight between the hidden and output layers respectively, and P_i is the corresponding output of the ith RBF unit in the hidden layer; y_j is the jth output. Once the centers of the RBF units are established, the width of the centers in the hidden layer can be calculated by Eq. (15):

$$\beta_1 = \beta_2 = \cdots = \beta_h = \frac{d_{\max}}{\sqrt{2h}}, \quad (15)$$

where $d_{\max}$ represents the maximum Euclidean distance between centers. In the following, the critic and action networks are described in detail.



(a) Critic network



(b) Action network

Fig. 3. Three layer RBF neural networks

3.1 Critic network

The critic network has h hidden layer nodes. The input vector to critic network $\boldsymbol{S}(k)$ is composed of n-dimensional state vectors $\boldsymbol{X}(k)$ along with m-dimensional action network outputs $\boldsymbol{U}(k)$. The output of the critic network is an approximation of the utility function $J(k)$.

According to the TD method, the prediction error of the critic network can be defined as

$$e_c(k) = \gamma J(k) - \left[J(k-1) - R(k) \right]. \quad (16)$$

The critic network is trained to approximate the "utility function" $J(k)$ by minimizing the objective function:

$$E_c(k) = \frac{1}{2} e_c^2(k). \quad (17)$$

The weights of the critic network are then updated

according to the following gradient-descent algorithm:

$$\boldsymbol{W}_{\mathrm{c}}(k+1)=\boldsymbol{W}_{\mathrm{c}}(k)+\Delta\boldsymbol{W}_{\mathrm{c}}(k), \tag{18}$$

$$\Delta\boldsymbol{W}_{\mathrm{c}}(k)=-\eta_{\mathrm{c}}\frac{\partial E_{\mathrm{c}}(k)}{\partial\boldsymbol{W}_{\mathrm{c}}(k)}, \tag{19}$$

where η_{c} is the learning rate of the critic network, and $\boldsymbol{W}_{\mathrm{c}}$ is the weight vector of hidden to output layer in the critic network. By applying chain rules to Eq. (19), the adaptation of the critic network can be represented as

$$\frac{\partial E_{\mathrm{c}}(k)}{\partial w_{ci}(k)}=\frac{\partial E_{\mathrm{c}}(k)}{\partial e_{\mathrm{c}}(k)}\frac{\partial e_{\mathrm{c}}(k)}{\partial J(k)}\frac{\partial J(k)}{\partial w_{ci}(k)}=e_{\mathrm{c}}(k)P_{ci}(k), \tag{20}$$

where w_{ci} represents the ith value in $\boldsymbol{W}_{\mathrm{c}}$; $P_{ci}(k)$ is the corresponding output of the ith RBF unit. To reduce the computational complexity during the training process, the locations of the RBF centers are determined offline using the k-means clustering algorithm. Thereafter, the gradient-descent algorithm is calculated to update the output linear weights online.

3.2 Action network

The action network generates the desired control actions based on the measurements of the system states and operates as the actual controller of the system. The action network has p nodes in the hidden layer, as shown in Fig. 3(b). The input to action network is an n-dimensional system state vector $\boldsymbol{X}(k)$ and the output is the approximation of an m-dimensional action vector $\boldsymbol{U}(k)$. According to the Bellman equation, the objective of the action network adaptation is to find the optimal control actions $\boldsymbol{U}^{*}(k)$ to minimize the utility function $J(k)$ over time:

$$\boldsymbol{U}^{*}(k)=\arg\min_{U}[J(k)]. \tag{21}$$

This is achieved by training the action network with an error vector $\boldsymbol{e}_{\mathrm{a}}(k)$:

$$\boldsymbol{e}_{\mathrm{a}}(k)=\frac{\partial J(k)}{\partial\boldsymbol{U}(k)}. \tag{22}$$

The error vector can be obtained by back propagating constant "1" through the critic network[11]. Therefore, the action network should be trained simultaneously with the critic network training process. By applying chain rules, the mathematical closed form of the error signal is given in Eq. (23):

$$\frac{\partial J(k)}{\partial u_j(k)}=2\sum_{i=1}^{h}\left[\frac{c_{i,n+j}-u_j(k)}{\beta_i^2}w_{ci}(k)P_{ci}(k)\right], \tag{23}$$

where $u_j(k)$ is the jth value in action vector $\boldsymbol{U}(k)$; $c_{i,n+j}$ is the $(n+j)$th value in $\boldsymbol{C}_i$.

The training of the action network is to minimize the following error function over time:

$$E_{\mathrm{a}}(k)=\frac{1}{2}\boldsymbol{e}_{\mathrm{a}}^{\mathrm{T}}(k)\boldsymbol{e}_{\mathrm{a}}(k). \tag{24}$$

The weights of the action network can be updated similarly to the critic network training according to:

$$\boldsymbol{W}_{\mathrm{a}}(k+1)=\boldsymbol{W}_{\mathrm{a}}(k)+\Delta\boldsymbol{W}_{\mathrm{a}}(k), \tag{25}$$

$$\Delta\boldsymbol{W}_{\mathrm{a}}(k)=-\eta_{\mathrm{a}}\frac{\partial E_{\mathrm{a}}(k)}{\partial\boldsymbol{W}_{\mathrm{a}}(k)}, \tag{26}$$

$$\frac{\partial E_{\mathrm{a}}(k)}{\partial w_{aji}(k)}=P_{ai}(k)\sum_{s=1}^{m}\left[\frac{\partial J(k)}{\partial u_s(k)}\frac{\partial}{\partial u_j(k)}\left(\frac{\partial J(k)}{\partial u_s(k)}\right)\right], \tag{27}$$

Where η_{a} is a positive learning rate of the action network, which can be different from η_{c}; $P_{ai}(k)$ is the corresponding output of the ith hidden layer RBF unit in the action network, and $j=1, 2, \cdots, m$, $i=1, 2, \cdots, p$.

3.3 Overall training procedure

The online training procedure for the critic and action networks consists of two stages. The critic network is trained first, following the training of the action network. In the first stage, the critic network's weights are initialized with small positive random values, and in its training cycle, the incremental optimization is carried out according to Eqs. (16)–(19). The optimization terminates when either the error has been sufficiently reduced ($e_{\mathrm{c}}(k)<E_{\mathrm{cthr}}$) or the internal update cycles of the weights have been reached ($N_{\mathrm{c}}>N_{\mathrm{cmax}}$). In the second stage, the critic network's weights are fixed, and the training of the action network is carried out by using Eqs. (24)–(26) until convergence. The convergence of the action network training means that the training procedure has found weights that yield optimal control actions for the plant under consideration.

The action network's weights will remain fixed until the plant operating condition has changed. The training of the critic network will start again to adapt to the changes. In this way, the training alternates between the critic and action networks, while changes the plant operating points from time to time. The flowchart for the complete training process is given in Fig. 4.

Normalization is performed in both action and critic networks to confine the values of weights into some appropriate range by

$$\boldsymbol{W}_{\mathrm{c}}(k+1)=\frac{\boldsymbol{W}_{\mathrm{c}}(k)+\Delta\boldsymbol{W}_{\mathrm{c}}(k)}{\left\|\boldsymbol{W}_{\mathrm{c}}(k)+\Delta\boldsymbol{W}_{\mathrm{c}}(k)\right\|_1}, \tag{28}$$

$$\boldsymbol{W}_{\mathrm{a}}(k+1)=\frac{\boldsymbol{W}_{\mathrm{a}}(k)+\Delta\boldsymbol{W}_{\mathrm{a}}(k)}{\left\|\boldsymbol{W}_{\mathrm{a}}(k)+\Delta\boldsymbol{W}_{\mathrm{a}}(k)\right\|_1}. \tag{29}$$

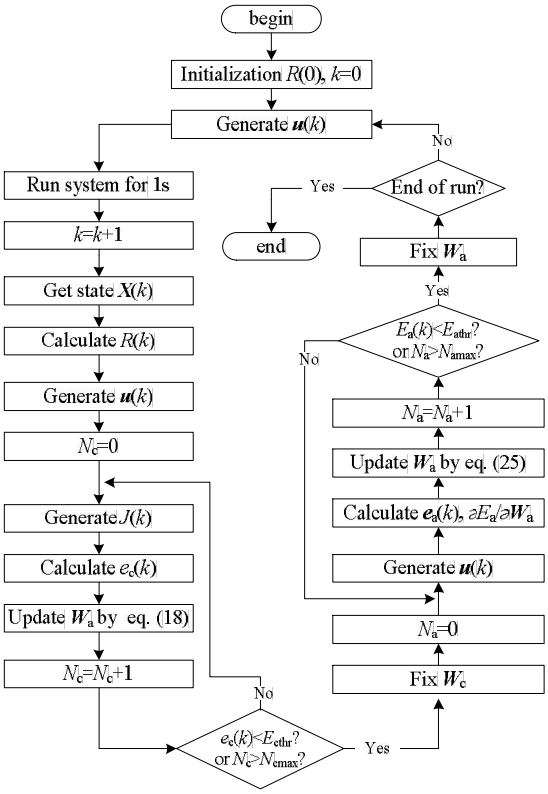


Fig. 4. Flow chart of the complete process

4 Implementation and Experiment Results

The NDP algorithm described in section 3 was then applied to an HEV energy management problem. It has been successfully implemented and tested using the Qianghua-I HEV. The details of the implementation and results will be given in the following subsections.

In this problem, the inputs to the critic network are the 6-dimensional vector $\boldsymbol{S}(k)=(T_{\text{dem}}(k),\ \omega_{\text{w}}(k),\ g(k),\ \xi(k),\ T_{\text{e}}(k),\ q(k))^{\text{T}}$. The output of the critic network is the utility function $J(k)$. The inputs to the action network are the 4-dimensional system states $\boldsymbol{X}(k)=(T_{\text{dem}}(k),\ \omega_{\text{w}}(k),\ g(k),\ \xi(k))^{\text{T}}$ and the outputs are the optimal control actions $\boldsymbol{U}(k)=(T_{\text{e}}(k),\ q(k))^{\text{T}}$.

4.1 NDP EMS test in simulation experiments

The NDP EMS had been predesigned using simulation before actual testing on vehicles to test stability and performance. It is also essential to predesign the EMS under a great variety of driving cycles to gain a level of insight which would normally take an impractical amount of time and effort using actual experiments. The simulation model for the Qianghua-I HEV is built in electric vehicle simulation software, ADVISOR.

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

The reason for the selection of $a_1=0$ is simply because that the emission maps are not provided for this engine. So the resultant EMS is a fuel-economy only strategy.

Table 2. Summary of the NDP algorithm parameters

Parameter	Value
Learning rate of the critic network η_{c}	0.8
Learning rate of the action network η_{a}	0.8
Internal cycle of the critic network N_{cmax}	30
Internal cycle of the action network N_{amax}	20
Internal training error threshold for the critic network E_{cthr}	0.01
Internal training error threshold for the action network E_{athr}	0.01
Number of hidden nodes in critic network h	20
Number of hidden nodes in action network p	25
Emission cost weight a_1	0
SOC deviation cost weight a_2	20
Gear shifting cost weight value a_3	0.4
Initial SOC value ξ_0	0.7
Reference SOC value ξ_{ref}	0.7
Discount factor γ	0.8

The developed NDP algorithm was written in Matlab version 6.5. By trial and error, 25 neurons for the critic network and 20 neurons for the action network in the hidden layer were optimally chosen for this case. Initially, the learning EMS had no prior knowledge about the plant as both the weights of the action network and the critic network were randomly initialized. Along with the training procedure over time, the developed EMS can learn from the history to set the weights more intelligently and more efficiently.

This integrated model is simulated under standard driving cycle EPA Urban Dynamometer Driving Schedule (UDDS). To illustrate the resultant NDP EMS more clearly, a convenient method is applied to represent it in an intuitive manner. A torque-split-ratio(TSR) $\tau=T_{\text{e}}/T_{\text{dem}}$ is defined to quantify the positive power flows in the powertrain[4]. Four positive power operation modes are defined, including motor-only ($\tau=0$), engine-only ($\tau=1$), power-assist ($0<\tau<1$), and charging mode ($\tau>1$). Fig. 5 shows the torque-split-ratio map of the resultant NDP strategy under standard driving cycle UDDS at a certain speed. The torque-split-ratio map gives us an intuitive impression of the EMS. From these maps one can easily partition different operation mode zones according to the values of τ.

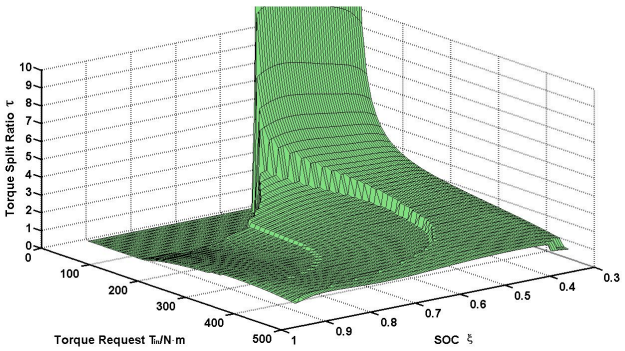


Fig. 5. TSR maps for NDP EMS at $\omega_{\text{w}}=86$ rad/s

It can be seen from Fig. 5 that when the SOC value is high ($\xi>0.7$), the NDP EMS uses the motor-only mode in

the low T_{dem} region ($T_{dem}<130$ N • m), and the power-assist mode in the high T_{dem} region ($T_{dem}\geq 130$ N • m). When the SOC is low ($\xi<0.5$) the engine will charge the battery ($\tau>1$). In general, the τ increases as the SOC decreases at the same T_{dem} in order to ensure charge sustainability. Note that negative torque requests can be handled with a relatively simple strategy: the motor recovers the maximum possible regeneration energy within the constraints imposed by the motor and the battery. The brakes will supply whatever is left.

4.2 NDP EMS implementation on VCU hardware

The proposed NDP EMS was then finally implemented on a vehicle control unit (VCU) hardware and tested on the Qianghua-I HEV by running it on a revolving drum test bench. The VCU and experiment details are shown in Fig. 6 and Fig. 7, respectively.

Fig. 6. VCU based on TMS320F2812 DSP chip

Fig. 7. Experiments on revolving drum test bench

Texas Instruments digital signal processor (DSP) TMS320F2812 was selected as the main chip of the VCU. It is a 32-bit fixed-point digital signal processor, and can operate at a maximum frequency of 150 MHz, providing adequate computational performance. The VCU uses a CAN communication interface. This allows for rapid communication with subsystem controllers to transfer orders and data. The fuel consumption and SOC values are provided by the engine control unit (ECU) and the battery management system (BMS), respectively. The VCU was also equipped with a storage cell FM25640, a 64 KB FRAM memory chip with an SPI interface, to save important parameters, such as network weights, when power is removed.

The system control cycle is 20 ms. Unlike other control tasks, the energy management of an HEV is an optimization problem, which does not require high real-time accuracy. Therefore, the update cycle for critic and action network weights was set at 1 s. The average of parameter changes in this period was taken to train the neural network. To calculate the neuron output, a look-up table approach was applied to determine exponential function. Due to the fact that the input vector to the neuron and the center are both normalized, most of the input values for the exponential function are located at $[-1, 0]$. Thus, the exponential function is decomposed according to Eq. (30). n_{int} is the nearest integer to n towards 0. A fine discretization is used for values between -1 and 0:

$$\exp(n)=\exp(n-n_{int})\bullet\exp(n_{int}). \tag{30}$$

All the calculations are done using fixed-point math to reduce the MIPS requirement for a real-time application. The clock cycle takes about 10ns when the TMS320F2812 is operated at its maximum clock speed of 150 MHz on our system. To facilitate convergence speed, the action network was pretrained offline based on the simulation data. After testing, according to the parameters given in Table 2, the whole training process took 39 558 743 cycles, that is, 400 ms, which can meet the real-time requirements of an HEV system. The whole training time includes the time taken to perform two while loops, as well as various other operations not directly related to the neural network.

4.3 Experiment results

To evaluate the performance and effectiveness of the NDP approach, the experiment results are compared with a heuristic rule-based EMS known as "Parallel Electric Assist Control Strategy"[1];

The experiment results for the UDDS driving cycles are listed in Table 3. It can be seen that the NDP strategy achieves very good results. Both the fuel consumption and component efficiency are improved significantly compared to the rule-based control strategy. Here, the fuel economy results have undergone SOC correction to compensate for the error caused by the SOC change before and after the experiment.

Table 3. Experiment results of the UDDS driving cycle

EMS	Rule-based	NDP
Fuel economy R_{fuel}/(miles • gal^{-1})	53.9	81.6
Engine efficiency η_e	24.6	34.6
Motoring efficiency η_m	87.2	91.6
Generating efficiency η_m	79.9	92.7

The experiment results shown in Fig. 8 indicate that the NDP EMS tends to keep the battery SOC within the range of 55%–70%. This leaves enough capacity to handle an extended period of battery discharge, and enough capacity

to absorb a long period of charging. Additionally, the battery SOC is maintained near a balance point to ensure the charge sustainability of the system.

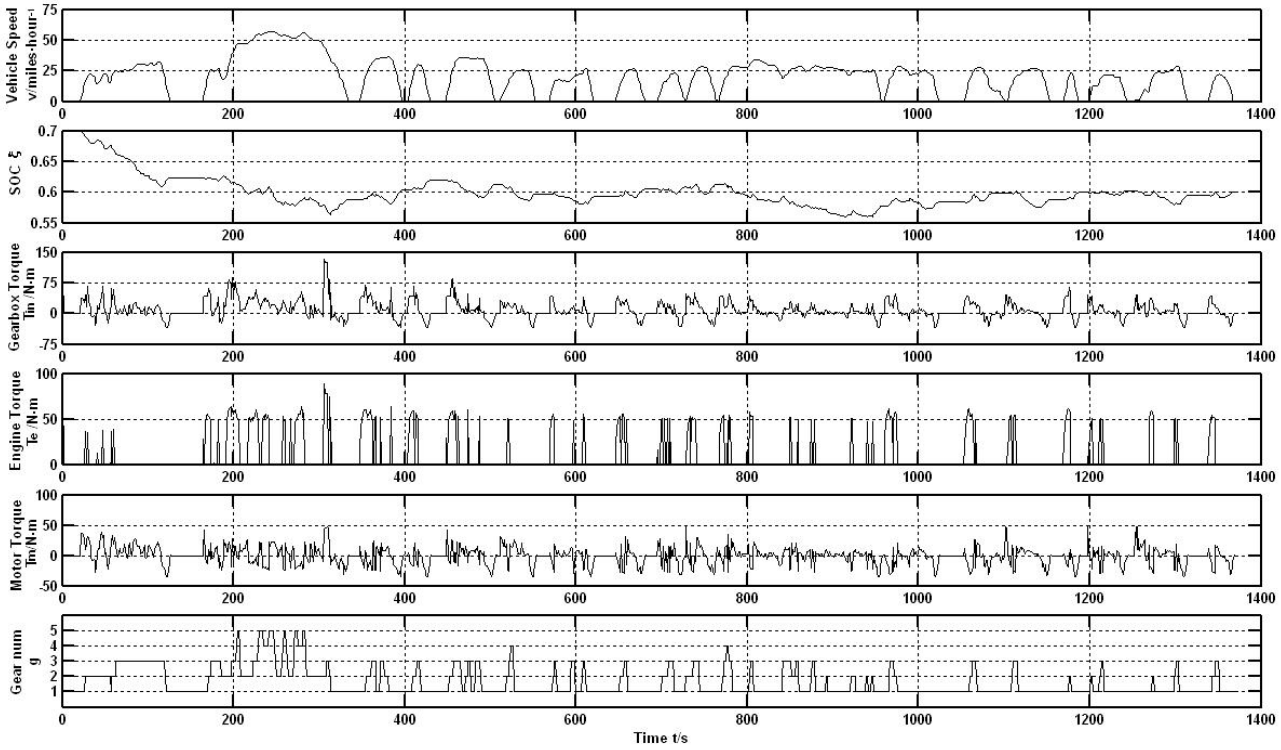


Fig. 8. NDP EMS experiment results under UDDS cycle

Fig. 9 and Fig. 10 report the distribution of engine and motor operating points under NDP EMS respectively. As shown by Fig. 9, the NDP EMS forces the engine operate at mid-range speed and load conditions (50–65 N • m) most of the time. The engine operating points under NDP EMS are mostly concentrated in the high efficiency region of 35.6%–38.6%, which will no doubt greatly improve the fuel economy. The motor is used under high torque request or regenerative braking. Its operating points are scattered in its whole working area as shown in Fig. 10. That means the motor participates more fully in propulsion of the vehicle.

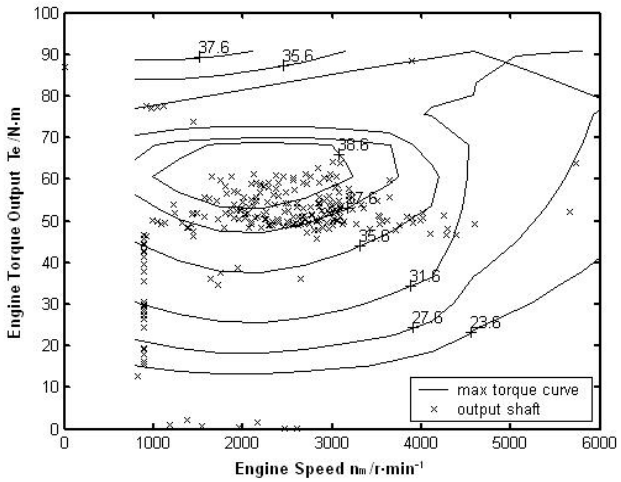


Fig. 9. Engine operation points distribution under NDP EMS

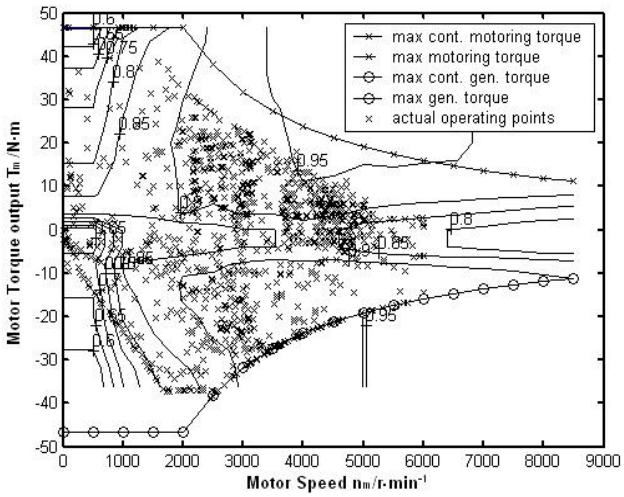


Fig. 10. Motor operation points distribution under NDP EMS

Fig. 11 shows the torque distribution trajectories. The figure clearly explains how the fuel economy is improved by using the NDP EMS. For illustration purposes, only the 160–300 s period of the torque distribution trajectory is shown. It can be seen that the engine provides the bulk of the torque demand, while the motor helps with the transient. The figure also shows a relatively smooth profile of the output torque of the engine compared with the driver torque demand and the motor output torque. The smoother engine torque output from the NDP EMS indicates that it helps improve fuel economy and alleviate emission problems.

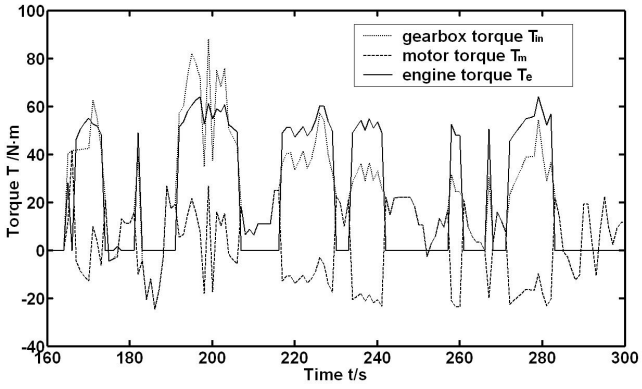


Fig. 11. Torque split trajectory by using NDP EMS

5 Conclusions

(1) The NDP technique provides an effective method to construct a suboptimal EMS with significantly reduced computational costs. A prototype HEV vehicle was used to evaluate the performance and effectiveness of this method. Experiment results indicate that this approach has good control performance and can significantly improve the average efficiency of the powertrain when compared with a traditional rule-based strategy.

(2) The resulted NDP control strategy takes the form of system state feedback. The requirement of a priori knowledge of driving conditions (necessary to implement the backward algorithm in the DP method) is unnecessary for the NDP EMS, allowing for its implementation in an actual vehicle.

(3) Emphasis also has been placed on the real-time capability analysis of the operation strategies. It verified that NDP EMS is very suitable to develop a fully digital controller and a complicated intelligent control algorithm for HEV control. Future work will concentrate on the determination of neural network size and detailed suboptimality analysis.

References

[1] JOHNSON V H, WIPKE K B, RAUSEN D J. HEV control strategy for real-time optimization of fuel economy and emissions[G]. SAE Paper no. 2000-01-1543.

[2] SCHOUTEN N J, SALMAN M A, KHEIR N A. Fuzzy logic control for parallel hybrid vehicles[J]. *IEEE Transactions on Control Systems Technology*, 2002, 10(3): 460–468.

[3] WON J S, LANGARI R. Intelligent energy management agent for a parallel hybrid vehicle—Part II: torque distribution, charge sustenance strategies, and performance results[J]. *IEEE Transactions on Vehicular Technology*, 2005, 54(3): 935–953.

[4] LIN C C, PENG H, GRIZZLE J W, et al. Power management strategy for a parallel hybrid electric truck[J]. *IEEE Transactions on Control Systems Technology*, 2003, 11(6): 839–849.

[5] DELPRAT S, LAUBER J, GUERRA T M, et al. Control of a parallel hybrid powertrain: optimal control[J]. *IEEE Transactions on Vehicular Technology*, 2004, 53(3): 872–881.

[6] ZHANG Rongjun, CHEN Yaobin. Control of hybrid dynamical systems for electric vehicles[C]//*Proceedings of American Control Conference*, Arlington, VA , USA, June 25─27, 2001: 2 884–2 889.

[7] PICCOLO A, IPPOLITO L, GALDI V, et al. Optimization of energy flow management in hybrid electric vehicles via genetic algorithms[C]//*IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, July 8─12, Como, Italy, 2001: 434–439.

[8] LIN C C, PENG H, GRIZZLE J W. A stochastic control strategy for hybrid electric vehicles[C]//*2004 American Control Conference*, Boston, USA, 30 June─2 July, 2004: 4 710–4 715.

[9] LIN C C, KIN M J, PENG H, et al. System-level model and stochastic optimal control for a PEM fuel cell hybrid vehicle[J]. *Journal of Dynamic Systems, Measurement, and Control*, 2006, 128(4): 878–890.

[10] BERTSEKAS D P, HOMER M L, LOGAN D A, et al. Missile defense and interceptor allocation by neuro-dynamic programming[J]. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 2000, 30(1): 42–51.

[11] PARK J W, HARLEY R G, VENAYAGAMOORTHY G K, Adaptive-critic-based optimal neurocontrol for synchronous generators in a power system using MLP/RBF neural networks[J]. *IEEE Transactions on Industry Applications*, 2003, 39(3): 1 529–1 540.

[12] CHAKRABORTY S, SIMOES M G. Neural dynamic programming based online controller with a novel trim approach[J]. *IEE Proc.-Control Theory Appl.*, 2005, 152(1): 95─104.

Biographical notes

LI Weimin was born in 1976. He received his BS and MS degrees in *Shandong University of Science and Technology, China*, in 1998 and 2003, respectively. Now he is a PhD candidate in *Department of Automation, Shanghai Jiao Tong University, China*. And his research interests focus on hybrid electric vehicle, motor control and neural network design.
E-mail: liweimin@sjtu.edu.cn

XU Guoqing received his PhD degree from *Zhejiang University, China*, in 1994. From 1997, he worked in *Tongji University, China* as associate professor, professor, and chairman of *Department of Electrical Engineering*. He is now a research professor in *The Chinese University of Hong Kong, China*, director of *Shenzhen Institute of Advanced Integration Technology*, director of *Center for Automotive Electronics*. His research interests have been in the areas of electric traction vehicle and automation, novel energy converter and control, and vehicle safety electronics.
E-mail: gq.xu@siat.ac.cn

XU Yangsheng (SM'95–F'03) received his PhD degree in robotics from the *University of Pennsylvania, Philadelphia, USA*, in 1989. He has been with *The Chinese University of Hong Kong, China*, since 1997 where he was chairman of *Department of Automation and Computer-Aided Engineering* from 1997 to 2004, and where currently he is chair professor. He was a faculty member with the *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA*, from 1989 to 1999 (on leave 1997 to 1999). His research has been in the areas of robotics, automation, and interface. Dr. XU is a fellow of *IEEE*, fellow of *HKIE*, academician of *Chinese Academy of Engineering*, academician of *Eurasian Academy of Sciences*, and corresponding member of *International Academy of Astronautics*.
E-mail: ysxu@mae.cuhk.edu.hk