Parallel Reinforcement Learning-Based Energy Efficiency Improvement for a Cyber-Physical System

Teng Liu, *Member, IEEE*, Bin Tian, *Member, IEEE*, Yunfeng Ai, *Member, IEEE*, and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*—As a complex and critical cyber-physical system (CPS), the hybrid electric powertrain is significant to mitigate air pollution and improve fuel economy. Energy management strategy (EMS) is playing a key role to improve the energy efficiency of this CPS. This paper presents a novel bidirectional long short-term memory (LSTM) network based parallel reinforcement learning (PRL) approach to construct EMS for a hybrid tracked vehicle (HTV). This method contains two levels. The high-level establishes a parallel system first, which includes a real powertrain system and an artificial system. Then, the synthesized data from this parallel system is trained by a bidirectional LSTM network. The lower-level determines the optimal EMS using the trained action state function in the model-free reinforcement learning (RL) framework. PRL is a fully data-driven and learning-enabled approach that does not depend on any prediction and predefined rules. Finally, real vehicle testing is implemented and relevant experiment data is collected and calibrated. Experimental results validate that the proposed EMS can achieve considerable energy efficiency improvement by comparing with the conventional RL approach and deep RL.**

***Index Terms*—Bidirectional long short-term memory (LSTM) network, cyber-physical system (CPS), energy management, parallel system, reinforcement learning (RL).**

I. Introduction

CYBER physical systems (CPSs) are defined as a system wherein the physical components are deeply intertwined with the software components to exhibit various and distinct behavioral patterns [1]. Recent increased demands of performance and complex usage pattern accelerate the development and research of CPSs [2]–[4]. Being a typical application of CPS in green transportation, hybrid electric vehicles (HEVs) show great potential to reduce energy consumption and air pollution [5], [6]. In such a system, hybrid electric powertrain and driving environments constitute the physical resources, communication and control data compose the cyber part of this system [7], [8]. Strong nonlinearities and uncertainties of the interactions between the cyber and physical resources increase difficulties in control, management, and optimization of HEVs [9], [10]. Especially, energy management of HEV is critical, and several challenges remain to be resolved, such as optimization, calculation time, and adaptability [11], [12].

Energy management strategies (EMSs) has been active research for decades because it can achieve remarkable energy efficiency. Existing EMSs are generally classified into three different categories, rule-based, optimization-based, and learning-based ones. Rule-based strategies depend on a set of predefined criterions without knowledge of real-world driving conditions [13], [14]. Binary control as a typical example is used to adjust power split between battery and engine as the state of charge (SOC) exceeds the threshold values. When the trip information is prior known, many approaches have been applied to search the optimal control strategies, such as dynamic programming (DP) [15], stochastic dynamic programming (SDP) [16], Pontryagin's minimum principle (PMP) [17], model predictive control (MPC) [18], and equivalent consumption minimization strategy (ECMS) [19]. However, these strategies are usually inappropriate for various driving environments [20]. Due to the ultrafast development of computing capability, learning-based methods emerge great potential in learning control strategies from the recorded historical driving data [21], [22]. This type of methods needs to be further developed.

As a complex CPS, hybrid electric powertrain still faces several issues to handle energy management problems. The first one is data lack [23]. The controller needs to collect new data and learn new model parameters to derive different strategies for new driving conditions. The second one is data inefficiency [24]. Large-dimension actions and states of complex CPS need to be calibrated and scheduled reasonably to guide the controller. The final one is universality. Adaptive and efficient control strategies need to be generated to

Manuscript received July 5, 2018; revised October 10, 2018; accepted December 12, 2018. The work was supported in part by the National Natural Science Foundation of China (61533019, 91720000), Beijing Municipal Science and Technology Commission (Z181100008918007), and the Intel Collaborative Research Institute for Intelligent and Automated Connected Vehicles (pICRI-IACVq). Recommended by Associate Editor Dongpu Cao. *(Corresponding author: Bin Tian.)*

Citation: T. Liu, B. Tian, Y. F. Ai, and F.-Y. Wang, "Parallel reinforcement learning-based energy efficiency improvement for a cyber-physical system," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 2, pp. 617–626, Mar. 2020.

T. Liu is with the Department of Automotive Engineering, Chongqing University, Chongqing 400044, and also with the Vehicle Intelligence Pioneers Inc., Qingdao 266109, China (e-mail: tengliu17@gmail.com).

B. Tian is with the Vehicle Intelligence Pioneers Inc., Qingdao 266109, and also with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: bin.tian@ia.ac.cn).

Y. F. Ai is with the Vehicle Intelligence Pioneers Inc., Qingdao 266109, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 101408, China (e-mail: aiyunfeng@ucas.ac.cn).

F.-Y Wang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: feiyue@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2020.1003072

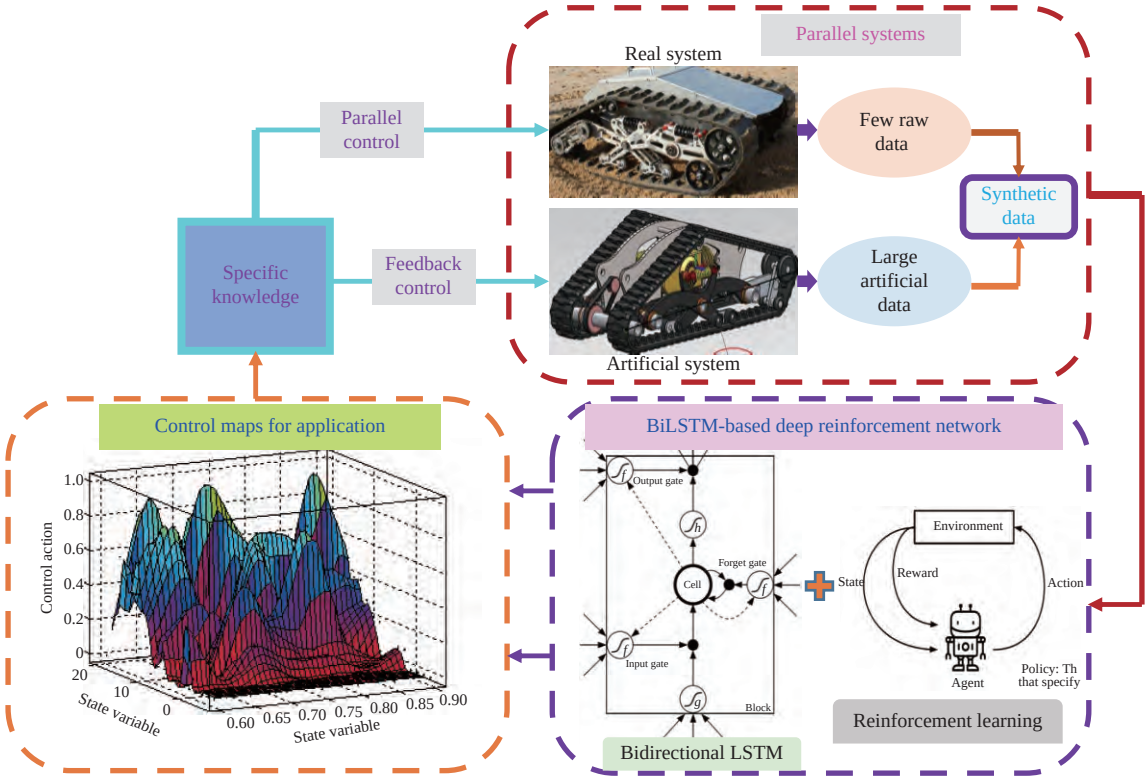

Fig. 1. Bidirectional LSTM network-based parallel reinforcement learning framework.

accommodate the dynamic real-world driving conditions.

To address these difficulties, we develop a novel bidirectional long short-term memory (LSTM) network based parallel reinforcement learning (PRL) framework to construct EMS for a hybrid tracked vehicle (HTV), see Fig. 1 as an illustration. This framework involves two levels. In the high-level structure, an artificial vehicle powertrain system is built analogy to the real vehicle to constitute the parallel powertrain system. The large synthesized data from this parallel system is utilized to relieve the data lack problem. A bidirectional LSTM network is proposed to represent dependence between multi-actions and state. This network can capture more details of the interactions between multi-action embeddings to solve the data inefficiency problem. In the lower-level skeleton, model-free reinforcement learning (RL) algorithm is finally used to compute the adaptive control strategy based on the trained data.

This literature involves three perspectives of contributions: 1) A parallel system of the HTV is constructed to generate large synthesized data based on the limited real historical data; 2) A bidirectional LSTM network is proposed to train the available data to model effectively the action state function; 3) Mode-free RL technique is applied to derive the adaptive EMS to accommodate different driving conditions. Experimental results illustrate that the proposed EMS can achieve considerable energy efficiency improvement by comparing with the conventional RL approach and deep RL.

The remainder of this paper is organized as follows. Section II describes the high-level architecture of a deep neural network for data estimation and the bidirectional LSTM network framework. Section III describes the modeling of the hybrid electric powertrain, wherein the optimal control problem is constructed, and the structure of the lower-level model-free RL algorithms are also introduced. In Section IV, the data collection in real vehicle tests and synthesized data processing are elaborated, and experiment results of three control strategies comparison are presented. Key takeaways are summarized in Section V.

II. Bidirectional LSTM Framework

Bidirectional LSTM network framework for action state function estimation is introduced in this section. First, multilayer deep neural network is constructed via considering powertrain state and actions as inputs. The states are the SOC in battery and generator speed, and actions are the engine torque, power demand, and motor speed. Based on this network, bidirectional LSTM theory is formulated to approximate the action value function. The detailed components are illustrated as follows.

A. Multilayer Neural Network

A deep neural network is a logical-mathematical model that seeks to simulate the behavior and function of a biological neuron [25]. Three layers named input layer, hidden layer, and output layer are included in this network, see Fig. 2(a) as an illustration. The input vector $z = [z_1, z_2, \ldots, z_N]$ are weighted by elements $\omega_1, \omega_2, \ldots, \omega_N$, then summed with a bias b and imposed by an activation function f to generate the neuron output as follow:

$$\begin{cases} x = \sum_{j=1}^{N} \omega_j z_j + b \\ y = f(x) \end{cases} \tag{1}$$

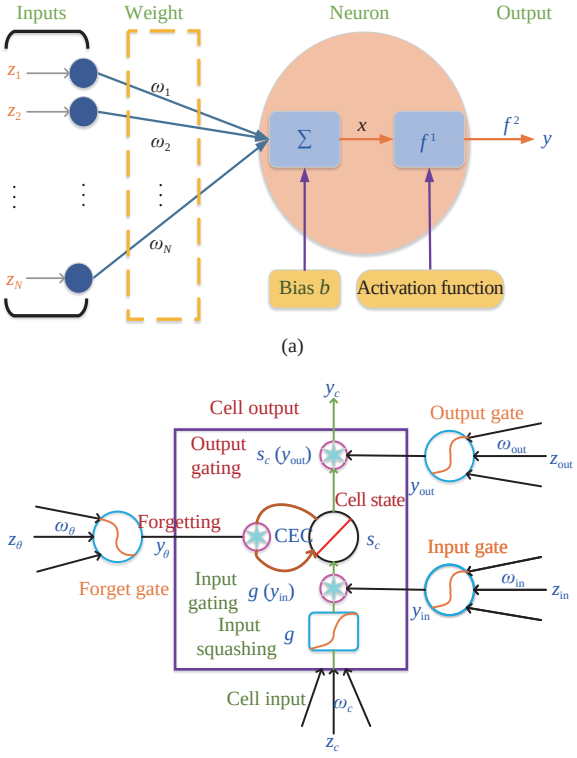

Fig. 2. Multilayer neural network for states and actions training. (a) Deep neural network construction; (b) LSTM memory block.

where x depicts the net input, y means the neuron output, N is total inputs index and z_j is the jth input.

The log-sigmoid activation function is adopted in this paper, and thus the output of the overall networks is depicted as

$$\begin{cases} f(x) = \dfrac{1}{1+e^{-x}} \\ y_{\text{all}} = f^2\left(\sum_{i=1}^{S} \omega_{1i}^2 f^1\left(\sum_{j=1}^{N} \omega^1{}_{ij} z_j + b_i^1\right) + b_{\text{all}}^2\right) \end{cases} \tag{2}$$

where f^2 and f^1 represent the activation function of the hidden layer and output layer, respectively. S is the number of neurons in the hidden layer, ω_{ij}^1 is the weight connecting the jth input and ith neuron in the hidden layer, ω_{1i}^2 represents the weight connecting the ith source of hidden layer to the output layer neuron. b_i^1 depicts the bias of the ith neuron in the hidden layer and b_{all}^2 is the bias of the neuron in the output layer.

B. Forward Pass of LSTM

A memory block is the key constituent part of an LSTM network. For each block, three adaptive and multiplicative gating units are shared by multiple cells, as shown in Fig. 2(b). Furthermore, a recurrently self-connected linear unit called constant error carousel (CEC) is the core of each block. The CEC can provide short-term memory storage for extended time periods by recirculating activation and error signals indefinitely. The three gating units are able to be trained to recognize, store and read information from the memory block. All the cells are combined into the block to share the same gates and reduce the number of adaptive parameters [26].

In this paper, the LSTM network is operated in bidirectional courses and the time steps are discreted as $t = 0, 1, 2, \ldots$. The two courses are named forward pass and backward pass, which mean the updating of the units' activation and the calculation of the error signals for weights. The notations in the following manuscript are defined as: j is the index of the memory block, v depicts the sequence of cells in the block j, and thus c_j^v means the vth cell in the jth memory block. y_c is the output of the memory cell, which is calculated by the cell state s_c, cell input z_c, input gate z_{in}, output gate z_{out}, and forget gate z_θ. ω_{lm} is the weight connecting the cell l and m. The components of one cell are described as follow.

1) Input: In the forward pass, the cell input is first computed as

$$z_{c_j^v}(t) = \sum_m \omega_{c_j^v m} y_m(t-1). \tag{3}$$

This variable is affected by the input squashing function g to generate the new cell state next.

The input gate activation y_{in} is derived by applying a logistic sigmoid squashing function f_{in} with range [0, 1] to the gate's net input z_{in}

$$\begin{cases} z_{\text{in}_j}(t) = \sum_m \omega_{\text{in}_j m} y_m(t-1) \\ y_{\text{in}_j}(t) = f_{\text{in}_j}(z_{\text{in}_j}(t)) \end{cases} \tag{4}$$

where $y_{\text{in}} \approx 1$ means the input gate is open and the relevant information can be stored in the block and $y_{\text{in}} \approx 0$ indicates the gate is close to shield the irrelevant one.

2) Cell State: The memory cell state s_c is initialized to zero when $t = 0$, and then it accumulates based on the input and discounted factor of the forget gate. First, the forget gate activation is defined as

$$\begin{cases} z_{\theta_j}(t) = \sum_m \omega_{\theta_j m} y_m(t-1) \\ y_{\theta_j}(t) = f_{\theta_j}(z_{\theta_j}(t)) \end{cases} \tag{5}$$

where f_θ represents a logistic sigmoid function and ranges from 0 to 1. Then, the new cell state is derived as follow:

$$s_{c_j^v}(t) = y_{\theta_j}(t) s_{c_j^v}(t-1) + y_{\text{in}_j}(t) g(z_{c_j^v}(t)), s_{c_j^v}(0) = 0. \tag{6}$$

What information to store in the memory block is decided by the input gate and when to erase the outdated information is determined by the forget gate. By doing this, the memory block can retain fresh data and the cell state cannot grow to infinity.

3) Output: The read access to the information is controlled by the output gate via multiplying the output from the CEC. The relevant activation is calculated by applying the squashing function ([0, 1]) into the net input

$$\begin{cases} z_{\text{out}_j}(t) = \sum_m \omega_{\text{out}_j m} y_m(t-1) \\ y_{\text{out}_j}(t) = f_{\text{out}_j}(z_{\text{out}_j}(t)). \end{cases} \tag{7}$$

Then, the cell output y_c is described by the cell state and output gate activation as follow:

$$y_{c_j^v}(t) = y_{\text{out}_j}(t) s_{c_j^v}(t). \tag{8}$$

Finally, the activation of the output units k is depicted as

$$y_k(t) = f_k(z_k(t)),\ z_k(t) = \sum_m \omega_{km} y_m(t) \tag{9}$$

where m range over all units, and f_k is the output squashing function.

C. Backward Pass of LSTM

LSTM's backward pass is a truncated version of real-time recurrent learning (RTRL) for weights to cell input, input gates, and forget gates. Also, it fuses the error of back-propagation (BP) in the output units and output gates efficiently.

1) Output Units and Gates: Based on the target t_k, the squared error objective function is depicted as

$$E(t) = \frac{1}{2}\sum_k e_k(t)^2,\ \ e_k(t) = t_k(t) - y_k(t) \tag{10}$$

where e_k is the externally injected error. Gradient descent algorithm is used to minimize the objective function. The weight ω_{lm} is decided by the variation $\Delta\omega_{lm}$, which is calculated via the negative gradient of E times the learning rate α. Hence, the standard BP weight changes of output units are

$$\Delta\omega_{km}(t) = \alpha\delta_k(t) y_m(t-1),\ \delta_k(t) = -\frac{\partial E(t)}{\partial z_k(t)}. \tag{11}$$

The standard BP is also utilized to compute the weight changes for connections to the output gate from source units m

$$\begin{cases} \Delta\omega_{\text{out}_j m}(t) = \alpha\delta_{\text{out}_j}(t) y_m(t) \\ \delta_{\text{out}_j}(t) \overset{\text{tr}}{=} f'_{\text{out}_j}(z_{\text{out}_j}(t)) \left(\sum_{v=1}^{S_j} s_{c_j^v}(t) \sum_k \omega_{k c_j^v} \delta_k(t) \right) \end{cases} \tag{12}$$

where $\overset{\text{tr}}{=}$ sign indicates error truncation, which indicates that the errors will not get propagated back further. Therefore, the LSTM learning algorithm becomes efficient.

2) Truncated RTRL Partials: The forward propagation is necessary in time for the partial derivatives in RTRL. These partials for weights at the cell (c^v_j), input gate (in), and forget gate (θ) are updated as follow:

$$\frac{\partial s_{c_j^v}(t)}{\partial \omega_{c_j^v m}} \overset{\text{tr}}{=} \frac{\partial s_{c_j^v}(t-1)}{\partial \omega_{c_j^v m}} y_{\varphi_j}(t) + g'(z_{c_j^v}(t)) y_{\text{in}_j}(t) y_m(t-1) \tag{13}$$

$$\frac{\partial s_{c_j^v}(t)}{\partial \omega_{\text{in}_j m}} \overset{\text{tr}}{=} \frac{\partial s_{c_j^v}(t-1)}{\partial \omega_{\text{in}_j m}} y_{\varphi_j}(t) + g(z_{c_j^v}(t)) f'_{\text{in}_j}(z_{\text{in}_j}(t)) y_m(t-1) \tag{14}$$

$$\frac{\partial s_{c_j^v}(t)}{\partial \omega_{\theta_j m}} \overset{\text{tr}}{=} \frac{\partial s_{c_j^v}(t-1)}{\partial \omega_{\theta_j m}} y_{\varphi_j}(t) + s_{c_j^v}(t-1) f'_{\theta_j}(z_{\theta_j}(t)) y_m(t-1) \tag{15}$$

where when $t = 0$, these partials equal to zero.

3) RTRL Weight Changes: In backward pass, the RTRL partials are employed to compute weight changes $\Delta\omega_{lm}$ for connections to the forget gate, cell and input gate as

$$\Delta\omega_{c_j^v m}(t) = \alpha e_{s_{c_j^v}}(t) \frac{\partial s_{c_j^v}(t)}{\partial \omega_{c_j^v m}} \tag{16}$$

$$\Delta\omega_{\text{in}_j m}(t) = \alpha \sum_{v=1}^{S_j} e_{s_{c_j^v}}(t) \frac{\partial s_{c_j^v}(t)}{\partial \omega_{\text{in}_j m}} \tag{17}$$

$$\Delta\omega_{\theta_j m}(t) = \alpha \sum_{v=1}^{S_j} e_{s_{c_j^v}}(t) \frac{\partial s_{c_j^v}(t)}{\partial \omega_{\theta_j m}}. \tag{18}$$

At each memory cell, the internal state error $e_{s_{c_j^v}}$ is determined as

$$e_{s_{c_j^v}}(t) \overset{\text{tr}}{=} y_{\text{out}_j}(t) \left(\sum_k \omega_{k c_j^v} \delta_k(t) \right). \tag{19}$$

D. Bidirectional LSTM Outline

In bidirectional recurrent nets, the forward and backward sequences of each training are regarded as two independent recurrent nets and are connected to the same output layer. Taking the time sequence from $t-1$ to t as an example, the outline that combines the bidirectional algorithm and LSTM is illustrated as follow.

1) Forward Pass: Feed all input data of the sequence into the LSTM and decide all the output units.

a) For the forward states (from time $t-1$ to t) and backward states (from time t to $t-1$), realize the forward pass process in Section II-B;

b) For the output layer, realize the forward pass process in Section II-B.

2) Backward Pass: Compute the relevant partial derivatives of error for the sequence used in the forward pass.

a) For the output neurons, achieve the backward pass process introduced in Section II-C;

b) For the forward states (from time t to $t-1$) and backward states (from time $t-1$ to t), achieve the backward pass process discussed in Section II-C.

3) Update Weight Changes: Finally, (16) to (19) are used to update RTRL weight changes.

III. Powertrain Model and Parallel Reinforcement Learning

In this section, the energy management of a hybrid tracked vehicle (HTV) is constructed as an optimization control problem. Modeling of the battery pack and engine-generator set (EGS) combine with the optimization objective are first introduced. To resolve the data lack problem of a complex CPS, a parallel system of the hybrid electric powertrain is then proposed to generate the artificial data. Real and artificial driving data constitute the synthesized data, which is trained to approximate the action state function. Finally, Q-learning algorithm is applied to compute the optimal control action according to the trained data from the bidirectional LSTM network.

The studied complex CPS is a self-built HTV and Fig. 3 depicts the sketch of the powertrain architecture. The main energy sources to propel the powertrain are the EGS and battery [10]. Table I lists the key characteristics of the HTV powertrain.

A. Powertrain Modeling

For EGS, the rated engine power is 52 kW at the speed

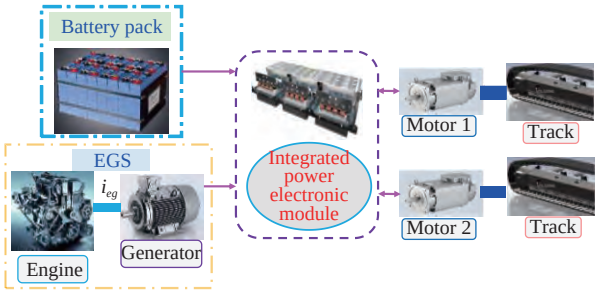

Fig. 3. Sketch of the self-built HTV architecture.

TABLE I
SPECIFICATION OF THE HTV

Symbol	Value	Unit
Vehicle mass $\boldsymbol{M}_v$	2500	kg
Inertia of generator $\boldsymbol{J}_g$	0.1	kg·m^2
Inertia of engine $\boldsymbol{J}_e$	0.2	kg·m^2
Gear ratio parameter $\boldsymbol{i}_{eg}$	1.2	/
Electromotive force parameter $\boldsymbol{K}_e$	0.8092	Vsrad^{-2}
Electromotive force parameter $\boldsymbol{K}_x$	0.0005295	NmA^{-2}
Minimum state of charge $\boldsymbol{SOC}_{\min}$	0.5	/
Maximum sate of charge $\boldsymbol{SOC}_{\max}$	0.9	/
Battery capacity $\boldsymbol{C}_{\text{bat}}$	37.5	Ah

6200 rpm. The rated generator output power is 40 kW within the speed range from 3000 rpm to 3500 rpm. The generator speed is the first state variable and is computed based on the torque equilibrium restraint

$$\begin{cases} \dfrac{dn_g}{dt} = \dfrac{\frac{T_e}{i_{eg}} - T_g}{0.1047\left(\frac{J_e}{i^2_{eg}} + J_g\right)} \\ n_e = \dfrac{n_g}{i_{eg}} \end{cases} \tag{20}$$

where n_g and n_e are the rotational speeds, T_g and T_e are the torques of generator and engine, respectively. T_e is one of the control variables in this work. J_e and J_g are the rotational moment of inertias for the engine and generator, severally. i_{eg} is the gear ratio connects the generator and engine, and 0.1047 is the transformation parameter which means 1 r/min = 0.1047 rad/s.

The output voltage and torque of the generator are derived as follow:

$$\begin{cases} T_g = K_e I_g - K_x I_g^2 \\ U_g = K_e n_g - K_x n_g I_g \end{cases} \tag{21}$$

where K_e is the electromotive force coefficient, U_g and I_g are the generator current and voltage, respectively. $K_x n_g$ is the electromotive force, and $K_x = 3PL^g/\pi$, in which L^g is the armature synchronous inductance, P is the poles number.

In the hybrid electric powertrain, SOC of battery is selected as another state variable. The output voltage and derivative of SOC in the battery are depicted via the equivalent first-order model

$$\begin{cases} \dfrac{dSOC}{dt} = -\dfrac{I_{\text{bat}}(t)}{C_{\text{bat}}} = \dfrac{(V_{\text{oc}} - \sqrt{V_{\text{oc}}^2 - 4r_{\text{in}}P_{\text{bat}}(t)})}{2C_{\text{bat}}r_{\text{in}}} \\ U_{\text{bat}} = \begin{cases} V_{\text{oc}} - I_{\text{bat}}r_{\text{ch}}(SOC), & I_{\text{bat}} \geq 0 \\ V_{\text{oc}} - I_{\text{bat}}r_{\text{dis}}(SOC), & I_{\text{bat}} < 0 \end{cases} \end{cases} \tag{22}$$

where I_{bat} and C_{bat} are the battery current and capacity, respectively. P_{bat} is the battery power, r_{in} is the battery internal resistance and V_{oc} is the open circuit voltage. U_{bat} is the output voltage of battery, $r_{\text{dis}}(SOC)$ and $r_{\text{ch}}(SOC)$ describe the internal resistance during discharging and charging, respectively.

The optimization control goal to be minimized is a trade-off between the charge sustaining constraint and fuel consumption over a finite horizon as

$$\begin{cases} J = \int_{t_0}^{t_f} [\dot{m}_f(t) + \lambda(\Delta SOC)^2]dt \\ \Delta SOC = \begin{cases} SOC(t) - SOC_{\text{ref}}, & SOC(t) < SOC_{\text{ref}} \\ 0, & SOC(t) \geq SOC_{\text{ref}} \end{cases} \end{cases} \tag{23}$$

where $[t_0, t_f]$ denotes the given time horizon, $\dot{m}_f$ is the fuel consumption rate, λ is a large positive weighting factor (λ = 10 000 in this paper) to restrict the terminal value of SOC, and SOC_{ref} is a pre-allocated constant to guarantee the charge sustaining constraint [27].

Furthermore, the instantaneous physical limits need to be observed to guarantee the reliability and safety of the powertrain:

$$\begin{cases} SOC_{\min} \leq SOC(t) \leq SOC_{\max} \\ n_{g,\min} \leq n_g(t) \leq n_{g,\max} \\ T_{e,\min} \leq T_e(t) \leq T_{e,\max} \\ n_{e,\min} \leq n_e(t) \leq n_{e,\max} \\ P_{dem,\min} \leq P_{dem}(t) \leq P_{dem,\max} \\ n_{m,\min} \leq n_m(t) \leq n_{m,\max} \end{cases} \tag{24}$$

where $n_{e,\min}$, $n_{e,\max}$, $T_{e,\min}$, and $T_{e,\max}$ are the permitted lower and upper bounds of the engine speed and torque, respectively. n_m is the motor speed, $n_{m,\min}$ and $n_{m,\max}$ are its boundary values. $P_{dem,\min}$, and $P_{dem,\max}$ are the threshold of power demand admissible sets, same as the $n_{g,\min}$ and $n_{g,\max}$.

Note that the core of this article focuses on discussing the PRL technique for a complex CPS, the traction motors are assumed as the power conversion devices with the identical efficiency and the battery aging is not considered in this study [9], [10].

B. Parallel Powertrain System

Fei-Yue Wang first initialized the parallel system theory in 2004 [28], [29], in which ACP method was proposed to deal the

complex CPS problem. ACP approach represents artificial societies (A) for modeling, computational experiments (C) for analysis, and parallel execution (P) for control. An artificial system is usually built by modeling, to explore the data and knowledge as the real system does. Through executing independently and complementally in these two systems, the learning model can be more efficient and less data-hungry. ACP approach has been employed in several fields to discuss the different problems in complex CPSs [30]–[32].

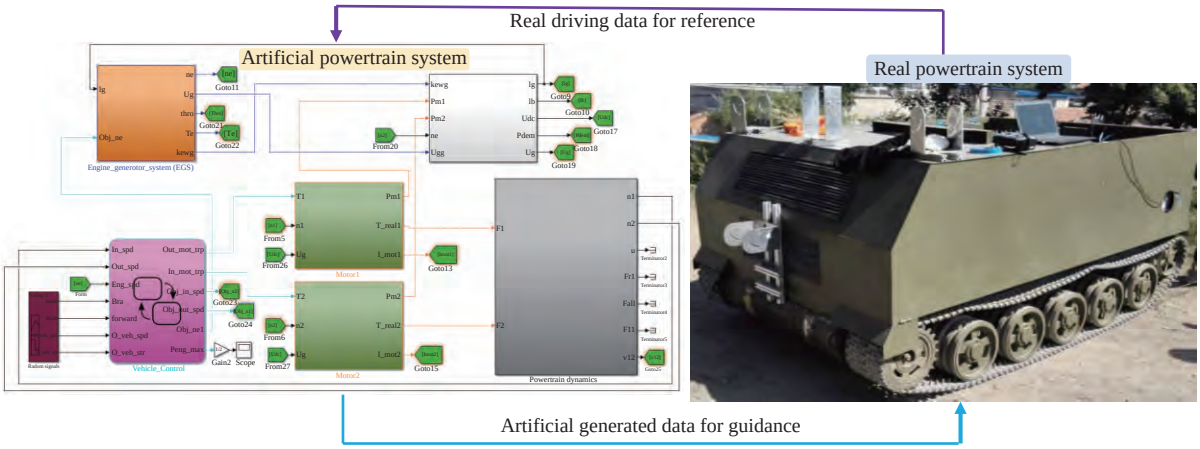

Fig. 4. Parallel powertrain system for the self-built HTV.

For a self-built HTV, there are not sufficient environments provided for it to operate to obtain enough actual data. Hence, we build an artificial powertrain system in MATLAB/Simulink to address the data lack problem in action state function training. This artificial system combines with the real powertrain system constitute the parallel system, see Fig. 4(a) as an illustration. To neglect the steering power (too small), the speeds of two tracks are equivalent to the average speed of them. By taking a few field test data as guidance and regulating the parameters of powertrain model and environments, large artificial data is acquired, including SOC, generator speed, engine torque, engine speed, power demand, battery current, battery voltage, and two motors speed. The synthesized data from the parallel system is collected and calibrated to derive the optimal EMS using the bidirectional LSTM network and reinforcement learning.

C. Reinforcement Learning

A learning agent interacts with a stochastic environment in reinforcement learning (RL) framework, and this interaction is modeled as a discrete discounted Markov decision process (MDP). The MDP is expressed as a quintuple ($\boldsymbol{S}$, $\boldsymbol{A}$, $\boldsymbol{\Pi}$, $\boldsymbol{R}$, γ), where $\boldsymbol{A}$ and $\boldsymbol{S}$ are the set of control actions and state variables, $\boldsymbol{\Pi}$ is the transition probability matrix (TPM), $\boldsymbol{R}$ is the reward function, and $\gamma \in (0, 1)$ is a discount factor. Especially, the state variables in this paper involves SOC and generator speed, control actions consist of the engine torque, power demand, and motor speed, reward function $r(s, a)$ represents the fuel consumption rate, and $p_{sa,\,s'}$ denotes the probability related with the transfer from state s to next state s' taking action a.

The value function is defined as the expected future reward

$$V(s) = E\left(\sum_{t=t_1}^{t_f} \gamma^{t_1-t_0} r(s)\right). \tag{25}$$

Then, the finite expected discounted and accumulated rewards is summarized as the optimal value function

$$V^*(s) = \min_{\pi} E\left(\sum_{t=t_0}^{t_f} \gamma^t r(s)\right) \tag{26}$$

where π is the control policy, which depicts the control action distribution with the time sequence. To deduce the optimal control action at each time instant, (26) is reformulated recursively as

$$V^*(s) = \min_a (r(s,a) + \gamma \sum_{s' \in S} p_{sa,s'} V^*(s')),\ \forall s \in S. \tag{27}$$

The optimal control policy is determined based on the optimal value function in (27)

$$\pi^*(s) = \arg\min_a (r(s,a) + \gamma \sum_{s' \in S} p_{sa,s'} V^*(s')). \tag{28}$$

Furthermore, the action value function and its corresponding optimal measure are described as follow:

$$\begin{cases} Q(s,a) = r(s,a) + \gamma \sum\limits_{s' \in S} p_{sa,s'} Q(s',a') \\ Q^*(s,a) = r(s,a) + \gamma \sum\limits_{s' \in S} p_{sa,s'} \min\limits_{a'} Q^*(s',a'). \end{cases} \tag{29}$$

Fig. 5 shows the bidirectional LSTM-based deep reinforcement network is utilized to estimate the action value function in RL. This structure includes two deep neural networks, one for state variables s_t^j embeddings and another for control sub-actions a_t^i embeddings. The bidirectional LSTM network is supposed to capture more information on how the individual embeddings combined into an integrated embedding due to its nonlinear structure.

The inner product is used to compute new $Q(s_t, a_t)$ through combining the states and sub-actions neuron output as

$$Q(s_t, a_t) = \sum_{j=1}^{K_1} \sum_{i=1}^{K_2} Q(s_t^j, a_t^i) \tag{30}$$

where K_1 and K_2 are the number of the states and sub-actions, respectively. $Q(s_t,\ a_t^i)$ denotes the expected accumulated future rewards relevant with the specific state variable s_t.

Finally, the action value function corresponding to an optimal control policy can be computed using the Q-learning algorithm as [33]

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \mu(r(s_t,a_t) + \gamma \min_{a'} Q(s_t',a_t') - Q(s_t,a_t)) \tag{31}$$

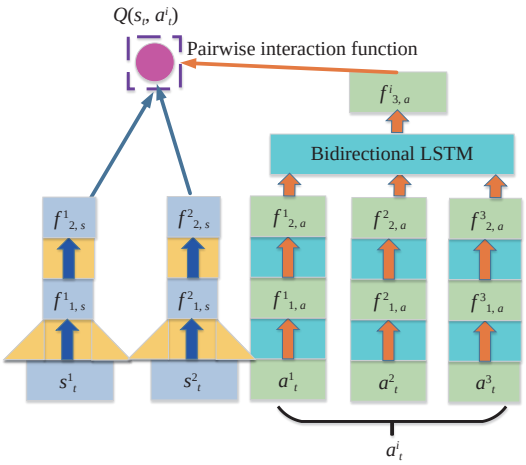

Fig. 5. Bidirectional LSTM-based deep neural network.

where $\mu \in [0, 1]$ is a decaying factor in Q-learning algorithm.

Algorithm 1 describes the pseudo-code of the Q-learning algorithm. The discount factor γ is settled as 0.96, the decaying factor μ is related with the time instant k and given as $1/\sqrt{k+2}$ to accelerate the convergence rate, the iterative times N_{it} is 10 000, and the sample time is 1 s.

Algorithm 1: Q-learning Algorithm

1. Extract $Q(s, a)$ from training and initialize iteration number N_{it}
2. Repeat time instant $k = 1, 2, 3, \ldots$
3. Based on $Q(s, \cdot)$, choose action a (ε-greedy policy)
4. Executing action a and observe r, s'
5. Define $a^* = \arg\min_a Q(s', a)$
6. $Q(s, a) \leftarrow Q(s, a) + \mu(r(s, a) + \gamma \min_{a'} Q(s', a') - Q(s, a))$
7. $s \leftarrow s'$
8. until s is terminal

The TPMs of power demand and vehicle modeling are inputs of RL technique for optimal EMS calculation. The RL algorithm is realized in Matlab via the Markov decision process (MDP) toolbox presented in [34] and a micro-processor with an Intel quad-core CPU of 2.70 GHz and RAM 3.8 GB. The proposed EMS is compared with the conventional RL approach and deep RL to demonstrate its optimality and availability in the next section.

IV. Experiment Results and Discussions

The proposed bidirectional LSTM enabled PRL-based energy management strategy (EMS) is assessed on the self-built HTV powertrain in this section. First, data collection and processing are introduced in detail. We operate the HTV in the real scenarios to collect real vehicle driving data. Based on this data, we generate the synthesized data from the parallel system to use for action value function estimation, including all the states and control variables. Then, the presented PRL-based energy management strategy is compared with the conventional RL and deep RL approaches to evaluate its availability and optimality. Simulation results indicate that the proposed energy management strategy is superior to the two benchmarking techniques in control performance.

A. Data Collection and Processing

The real vehicle experiment is implemented on the self-built HTV in the suburb to represent the cross-country scenarios, and the real and target driving cycles are depicted in Fig. 6. The vehicle data and powertrain states are collected with a sampling frequency of 100 Hz from the CAN bus. The collected driving data is applied to create large artificial data in the parallel powertrain system. Observing the physical constraints of powertrain, the inputs of the parallel system are engine throttle and rotational speed randomly, and outputs are the state variables and control actions. Fig. 7 illustrates a period of the generated driving data from the parallel powertrain system.

Furthermore, to eliminate the influence of different variable

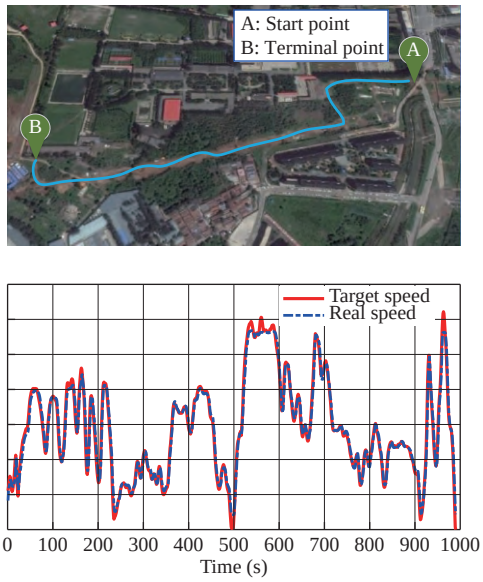

Fig. 6. Real vehicle testing scenario and the corresponding driving cycles.

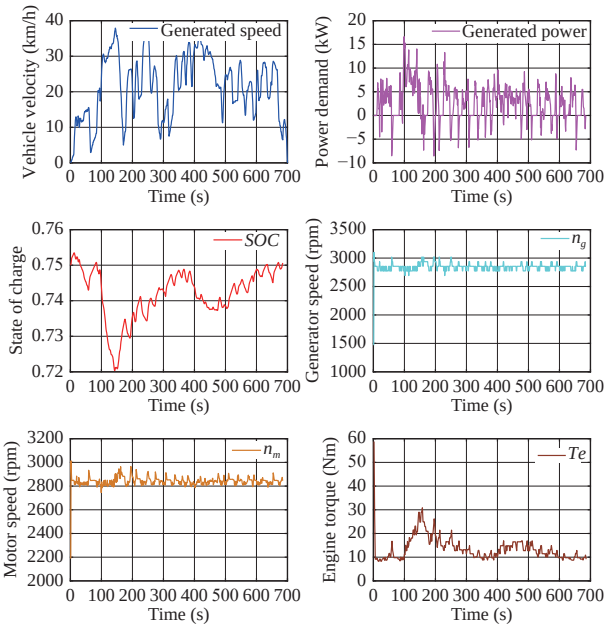

Fig. 7. One period of the generated driving data from the parallel system.

units on training, the input state variables and control actions of the network are scaled to the range from 0 to 1.

B. Comparison of Different EMSs

Based on the trained action-value function, the proposed bidirectional LSTM enabled PRL-based EMS is compared with the conventional RL and deep RL controls to certify its availability and optimality in this section. In the energy management problem, the simulation cycle is a real vehicle driving cycle, and the initial values of the state variable SOC and generator speed are 0.7 and 1200 rpm, respectively.

The SOC trajectories of a certain driving cycle and the corresponding generator speed are illustrated in Fig. 8. It can be discerned that the SOC trajectory imposed on the proposed model free EMS is close to that in deep RL control, and they are different from that in conventional RL control. This can be explained by the different power split between the EGS and battery, which is decided by the action value functions. It demonstrates that the training process in the deep neural network can improve the accuracy and optimality of control policy derived by the Q-learning algorithm. An analogous result in the generator speed trajectory is also given in Fig. 8.

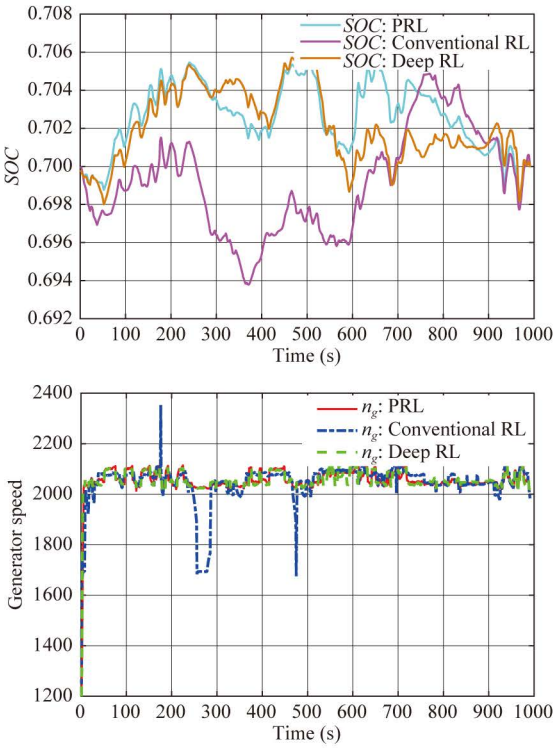

Fig. 8. State variables SOC and generator speed trajectories for the different control strategies.

Taking engine torque as an example, the above observation can be explained by the different distribution of engine torque with the state variables. Being a control variable, different values of engine torque decide multiple operative modes of the powertrain, as shown in Fig. 9.

The convergence processes of the action value function in the proposed EMS, conventional RL and deep RL are illustrated in Fig. 10. The mean discrepancy depicts the deviation of two action value functions per 100 iterations.

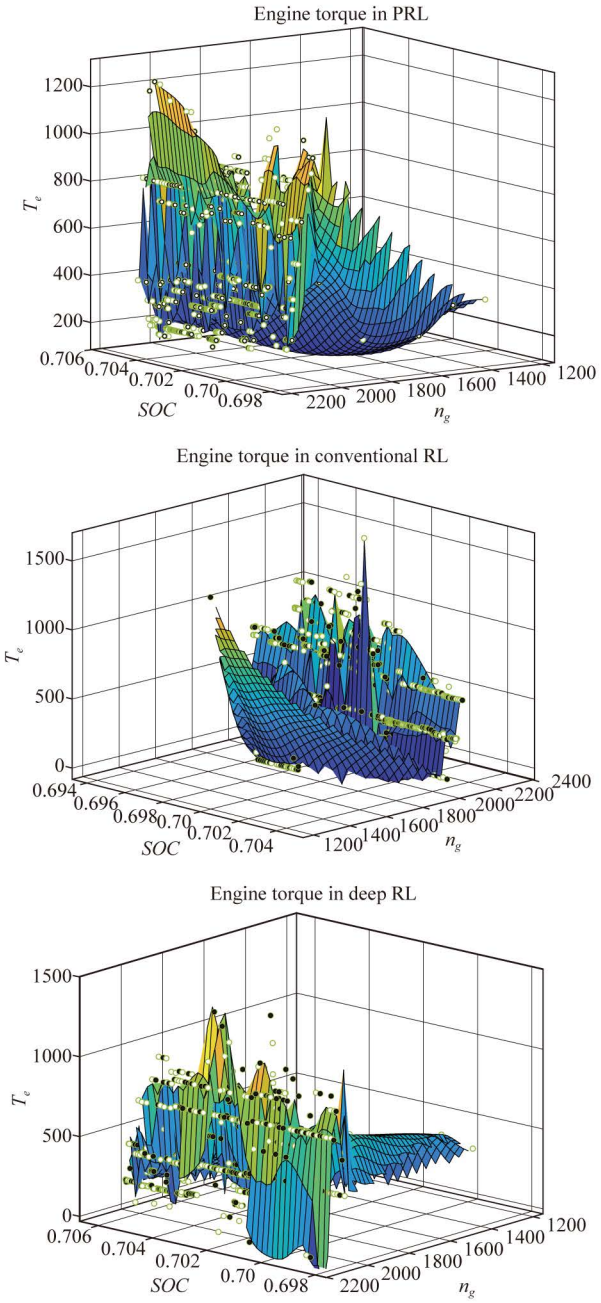

Fig. 9. Control action engine torque with the state variables.

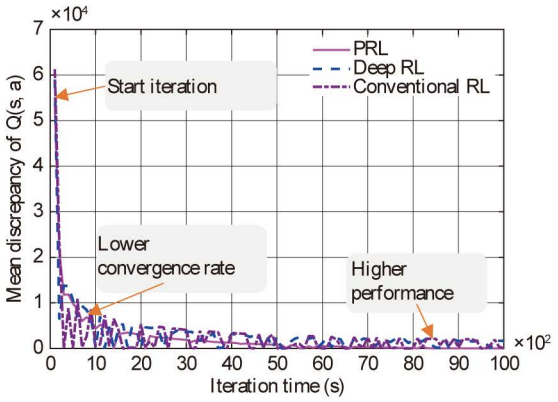

Fig. 10. Convergence rates of the action value function in three controls.

Note that, the increase of iterative number accompanies with a decreased mean discrepancy, which implies the convergence characteristic of the Q-learning algorithm.

Fig. 10 also describes that the proposed control is superior to the conventional RL and deep RL controls in control performance, and the convergence rate is a little slower than them. This can be illuminated by the additional training process of the action value function in the bidirectional LSTM network. With an accepted calculation speed, the proposed EMS adapts to the real-time driving conditions more suitably than the conventional RL and deep RL controls, which demonstrates its availability.

Table II describes the fuel consumption after SOC correction and computation time for the three control strategies. It is apparent that the fuel consumption under the PRL-enabled EMS is lower than those in conventional RL-based and deep RL controls, which demonstrates its optimality. Also, the consumed time of PRL is lower than that of deep RL and conventional RL, which implies that it is potential to be applied in real-time.

TABLE II
THE FUEL CONSUMPTION IN THREE CONTROL STRATEGIES

Algorithms	Consumed fuel (g)	Time cost (s)
PRL	416.8	46.8
Deep RL	441.3	61.5
Conventional RL	465.7	53.7

V. CONCLUSION

We propose a novel bidirectional LSTM network based PRL framework to construct EMS for an HTV in this paper. First, the up-level builds an artificial vehicle powertrain system analogy to the real vehicle to constitute the parallel powertrain system. Second, a bidirectional LSTM network is proposed to train the large synthesized data from this parallel system to represent dependence between multi-actions and states. Third, in the lower-level skeleton, model-free RL algorithm is finally used to compute the adaptive control strategy based on the trained data.

Tests prove the optimality and availability of the proposed energy management strategy. In addition, the advantages in control performance and energy efficiency imply that the proposed adaptive control can be applied in real situations.

The proposed combination of bidirectional LSTM network and RL is indeed a simplified specification of the so-called parallel learning [35] which aims to build a more general framework for data-driven intelligent control. Future work focuses on applying the parallel learning and PRL framework into different research fields of automated vehicles, such as driving style recognition [36], braking intensity estimation [37], [38], and lane changing intention prediction [39], [40]. The parallel system could generate abundant driving data and evaluate the performance of different controllers easily.

REFERENCES

[1] F.-Y. Wang, "The emergence of intelligent enterprises: from CPS to CPSS," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 4, pp. 85–88, 2010.

[2] F.-Y. Wang, "Control 5.0: from Newton to Merton in popper's cyber-social-physical spaces," *IEEE/CAA J. Autom. Sinica*, vol. 3, no. 3, pp. 233–234, 2016.

[3] X. L. Tang, X. S. Hu, W. Yang, and H. S. Yu, "Novel torsional vibration modeling and assessment of a power-split hybrid electric vehicle equipped with a dual mass flywheel," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 1900−2000, 2018.

[4] T. Liu, X. S. Hu, W. H. Hu, and Y. Zou, "A heuristic planning reinforcement learning-based energy management for power-split plug-in hybrid electric vehicles," *IEEE Trans. Industrial Informatics*, Mar. 2019. DOI: 10.1109/TII.2019.2903098.

[5] T. Liu, X. S. Hu, S. E. Li, and D. P. Cao, "Reinforcement learning optimized look-ahead energy management of a parallel hybrid electric vehicle," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 4, pp. 1497–1507, 2017.

[6] Y. Zou, T. Liu, D. X. Liu, and F. C. Sun, "Reinforcement learning-based real-time energy management for a hybrid tracked vehicle," *Applied Energy*, vol. 171, pp. 372–382, 2016.

[7] C. Lv, Y. H. Liu, X. S. Hu, H. Guo, D. P. Cao, and F.-Y. Wang, "Simultaneous observation of hybrid states for cyber-physical systems: a case study of electric vehicle powertrain," *IEEE Trans. Cybernetics*, vol. 48, no. 8, pp. 2357–2367, 2018.

[8] X. S. Hu, H. Wang, and X. L. Tang, "Cyber-physical control for energy-saving vehicle following with connectivity," *IEEE Trans. Indus. Electron.*, vol. 64, no. 11, pp. 8578–8587, 2017.

[9] Y. Zou, Z. H. Kong, T. Liu, and D. X. Liu, "A real-time Markov chain driver model for tracked vehicles and its validation: its adaptability via stochastic dynamic programming," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 3571–3582, 2017.

[10] T. Liu, Y. Zou, D. X. Liu, and F. C. Sun, "Reinforcement learning of adaptive energy management with transition probability for a hybrid electric tracked vehicle," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7837–7846, 2015.

[11] C. M. Martinez, X. S. Hu, D. P. Cao E. Velenis, B. Gao, and M. Wellers, "Energy management in plug-in hybrid electric vehicles: recent progress and a connected vehicles perspective," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 4534–4549, 2017.

[12] Y. C. Qin, F. Zhao, Z. F. Wang L. Gu, and M. M. Dong, "Comprehensive analysis for influence of controllable damper time delay on semi-active suspension control strategies," *J. Vibration and Acoustics-Trans. ASME*, vol. 139, no. 3, pp. 031006-1–031006-12, 2017.

[13] T. Liu, B. Wang, and C. L. Yang, "Online Markov chain-based energy management for a hybrid tracked vehicle with speedy Q-learning," *Energy*, vol. 160, pp. 544–555, 2018.

[14] H. S. Ramadan, M. Becherif, and F. Claude, "Energy management improvement of hybrid electric vehicles via combined GPS/rule-based methodology," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 586–597, 2017.

[15] K. Li, F. C. Chou, and J. Y. Yen, "Real-time, energy-efficient traction allocation strategy for the compound electric propulsion system," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 3, pp. 1371–1380, 2017.

[16] M. Muratori and G. Rizzoni, "Residential demand response: dynamic energy management and time-varying electricity pricing," *IEEE Trans. Power syst.*, vol. 31, no. 2, pp. 1108–1117, 2016.

[17] S. Delprat, T. Hofman, and S. Paganelli, "Hybrid vehicle energy management: singular optimal control," *IEEE Trans. Veh. Technol*, vol. 66, no. 6, pp. 9654–9666, 2017.

[18] L. L. Guo, B. Z. Gao, Q. F. Liu, J. H. Tang, and H. Chen, "On-line optimal control of the gearshift command for multispeed electric vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 4, pp. 1519–1530, 2017.

[19] J. H. Han, D. Kum, and Y. Park, "Synthesis of predictive equivalent consumption minimization strategy for hybrid electric vehicles based on closed-form solution of optimal equivalence factor," *IEEE Trans. Veh.*

Technol., 2017. DOI: 10.1109/tvt.2017.2660764.

[20] P. Nyberg, E. Frisk, and L. D. Nielsen, "Using real-world driving databases to generate driving cycles with equivalence properties," *IEEE Trans. Veh. Technol*, vol. 65, no. 6, pp. 4095–4105, Jun. 2016.

[21] T. Liu, X. L. Tang, H. Wang, H. Yu, and X. S. Hu, "Adaptive hierarchical energy management design for a plug-in hybrid electric vehicle," *IEEE Trans. Veh. Technol.*, Jul, 2019. DOI: 10.1109/TVT.2019.2926733.

[22] T. Liu, Y. Zou, D. X. Liu, and F. C. Sun, "Reinforcement learning-based energy management strategy for a hybrid electric tracked vehicle," *Energies*, vol. 8, no. 7, pp. 7243–7260, 2015.

[23] M. Deniša, A. Gams, A. Ude, and T. Petric, "Learning compliant movement primitives through demonstration and statistical generalization," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 5, pp. 2581–2594, 2017.

[24] V. Mnih, K. Kavukcuoglu, D. Silver, and A. Graves, "Playing atari with deep reinforcement learning, " arXiv preprint, arXiv: 1312.5602, 2013.

[25] M. Hagan, H. Demuth, M. Beale, and O. De Jess, *Neural Network Design,* Boston, MA, Martin Hagan, 2014.

[26] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *J. Machine Learning Research*, vol. 3, no. 1, pp. 115–143, 2002.

[27] L. Li, S. X. You, C. Yang, B. J. Yan, J. Song, and Z. Chen, "Driving-behavior-aware stochastic model predictive control for plug-in hybrid electric buses," *Appl Energy*, vol. 162, pp. 868–879, 2016.

[28] F.-Y. Wang, "Artificial societies, computational experiments, and parallel systems: a discussion on computational theory of complex social-economic systems," *Complex Syst. Complex. Sci.*, vol. 1, no. 4, pp. 25–35, Oct. 2004.

[29] F.-Y. Wang, "toward a paradigm shift in social computing: the ACP approach," *IEEE Intell. Syst.*, vol. 22, no. 5, pp. 65–67, Sept.–Oct. 2007.

[30] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: concepts, architectures, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.

[31] F.-Y. Wang and S. N. Tang, "Artificial societies for integrated and sustainable development of metropolitan systems," *IEEE Intell. Syst.*, vol. 19, no. 4, pp. 82–87, Jul.–Aug. 2004.

[32] F.-Y. Wang, H. G. Zhang, and D. R. Liu, "Adaptive dynamic programming: an introduction," *IEEE Comput. Intell. Magazine*, vol. 4, no. 2, pp. 39–47, Jun. 2009.

[33] T. Liu, H. L. Yu, H. Y. Guo, Y. C. Qin, and Y. Zou, "Online energy management for multimode plug-in hybrid electric vehicles," *IEEE Trans. Industrial Informatics*, vol. 15, no. 7, pp. 4352–4361, Jul. 2019.

[34] P. Shan, R. Li, S. H. Ning, and Q. Yang, "Markov decision process toolbox, " in *Proc. of IEEE Int. Workshop on Open-Source Software for Scientific Computation* (OSSC), Sep. 2009. DOI: 10.1109/ossc.2009.5416859.

[35] L. Li, Y. L. Lin, N. N. Zheng, and F.-Y. Wang, "Parallel learning: a perspective and a framework," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 3, pp. 389–395, 2017.

[36] C. Lv, X. S. Hu, A. Sangiovanni-Vincentelli, Y. T. Li, C. M. Martinez, and D. P. Cao, "Driving-style-based codesign optimization of an automated electric vehicle: a cyber-physical system approach," *IEEE Trans. Indus. Electron*, vol. 66, no. 4, pp. 2965–2975, 2018.

[37] C. Lv, Y. Xing, C. Lu, Y. H. Liu, H. Y. Guo, H. B. Gao, and D. P. Cao, "Hybrid-learning-based classification and quantitative inference of driver braking intensity of an electrified vehicle," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 5718–5729, 2018.

[38] C. Lv, Y. Xing, J. Z. Zhang, X. X. Na, Y. T. Li, T. Liu, D. P. Cao, and F.-Y. Wang, "Leven-berg-marquardt backpropagation training of multilayer neural networks for state estimation of a safety-critical cyber-physical system," *IEEE Trans. Industrial Informatics*, vol. 14, no. 8, pp. 3436–3446, 2017.

[39] Y. Xing, C. Lv, H. J. Wang, D. P. Cao, E. Velenis, and F.-Y. Wang, "Driver lane change intention inference for intelligent vehicles: framework, survey, and challenges," *IEEE Trans. Veh. Technol*, vol. 68, no. 5, pp. 4377–4390, 2019.

[40] T. Liu and X. S. Hu, "A Bi-level control for energy efficiencyimprovement of a hybrid tracked vehicle," *IEEE Trans. Industrial Informatics*, vol. 14, no. 4, pp. 1616–1625, 2018.

Teng Liu (M'18) received the B.S. degree in mathematics from Beijing Institute of Technology, in 2011. He received the Ph.D. degree in automotive engineering from Beijing Institute of Technology (BIT), in 2017. He worked as a Research Fellow at Vehicle Intelligence Pioneers Ltd. for one year. Now, he is a member of IEEE VTS, IEEE ITS, IEEE IES, IEEE TEC. Dr. Liu is now a Professor in the Department of Automotive Engineering, Chongqing University. Dr. Liu has more than 8 years' research and working experience in renewable vehicle and connected autonomous vehicle. His research interests include reinforcement learning (RL)-based energy management in hybrid electric vehicles, RL-based decision making for autonomous vehicles, and CPSS-based parallel driving. He has published over 30 SCI papers and 10 conference papers in these areas. He received the Merit Student of Beijing in 2011, the Teli Xu Scholarship (Highest Honor) of Beijing Institute of Technology in 2015, "Top 10" in 2018 IEEE VTS Motor Vehicle Challenge and sole outstanding winner in 2018 ABB Intelligent Technology Competition.

Bin Tian (M'18) received the B.S. degree from Shandong University, in 2009 and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2014. He is currently an Associate Professor of the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision, machine learning, and automated driving.

Yunfeng Ai (M'18) received the B.S. degree from Shandong University, in 2001 and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2006. He was a Visiting Scholar from December 2015 to October 2016 and Postdoctoral Researcher from November 2016 to April 2017 at Carnegie Mellon University. He is currently a Research Scientist of University of Chinese Academy of Sciences. His research interests include computer vision, machine learning, parallel robots, and automated driving.

Fei-Yue Wang (S'87–M'89–SM'94–F'03) received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, New York in 1990. He joined the University of Arizona in 1990 and became a Professor and Director of the Robotics and Automation Lab (RAL) and Program in Advanced Research for Complex Systems (PARCS). Dr. Wang's research focuses on methods and applications for parallel systems, social computing, and knowledge automation. Currently he is EiC of *IEEE Transactions on Computational Social Systems*, Founding EiC of *IEEE/CAA Journal of Automatica Sinica*, and *Chinese Journal of Command and Control*. Since 1997, he has served as General or Program Chair of more than 20 IEEE, INFORMS, ACM, and ASME conferences. In 2007, he received the National Prize in Natural Sciences of China and was awarded the Outstanding Scientist by ACM for his research contributions in intelligent control and social computing. He received IEEE ITS Outstanding Application and Research Awards in 2009, 2011 and 2015, and IEEE SMC Norbert Wiener Award in 2014.