Knowledge Implementation and Transfer With an Adaptive Learning Network for Real-Time Power Management of the Plug-in Hybrid Vehicle

Quan Zhou, *Member, IEEE*, Dezong Zhao, *Senior Member, IEEE*, Bin Shuai, Yanfei Li, Huw Williams, and Hongming Xu

***Abstract*— Essential decision-making tasks such as power management in future vehicles will benefit from the development of artificial intelligence technology for safe and energy-efficient operations. To develop the technique of using neural network and deep learning in energy management of the plug-in hybrid vehicle and evaluate its advantage, this article proposes a new adaptive learning network that incorporates a deep deterministic policy gradient (DDPG) network with an adaptive neuro-fuzzy inference system (ANFIS) network. First, the ANFIS network is built using a new global K-fold fuzzy learning (GKFL) method for real-time implementation of the offline dynamic programming result. Then, the DDPG network is developed to regulate the input of the ANFIS network with the real-world reinforcement signal. The ANFIS and DDPG networks are integrated to maximize the control utility (CU), which is a function of the vehicle's energy efficiency and the battery state-of-charge. Experimental studies are conducted to testify the performance and robustness of the DDPG-ANFIS network. It has shown that the studied vehicle with the DDPG-ANFIS network achieves 8% higher CU than using the MATLAB ANFIS toolbox on the studied vehicle. In five simulated real-world driving conditions, the DDPG-ANFIS network increased the maximum mean CU value by 138% over the ANFIS-only network and 5% over the DDPG-only network.**

***Index Terms*— Deep deterministic policy gradient (DDPG) network, fuzzy inference system, plug-in hybrid vehicle, power management, transfer learning.**

I. Introduction

RECENT advances in artificial intelligence (AI) and informatics have significantly promoted the development of connected and autonomous vehicles [1]. AI techniques will be equipped to future vehicles for enhancing safety and achieving the optimal energy efficiency. On the contrary, electrification brings another revolution to the automotive industry. By harnessing conventional thermal propulsion with electric drives, the hybridized vehicles can achieve high efficiency and low emissions simultaneously. Hybrid electric vehicles, as a mainstream ultralow emission solution, will account for more than 60% of the world automotive market by 2030 according to predictions from the International Energy Agency [2].

The power management systems (PMSs) regulate the energy flows between the power units (e.g., engine and battery) within the hybrid vehicle. The optimization of energy efficiency in the PMS is one of the most challenging decision-making tasks because of the uncertainties in real-world driving and constraints in operations [3], [4]. PMSs are expected to be optimized such that vehicles comply with the stricter regulations in fuel consumption and emissions. Taking Europe as an example, the new European driving cycle (NEDC) for road vehicles has been replaced by the worldwide-harmonized light-duty testing cycle (WLTC), where an increasing number of transient operation points is included to evaluate energy efficiency and emissions [5]. New legislations on examining real-world driving emissions (RDEs) have been enforced [6]–[8], which bring more uncertain transient operation conditions to be considered for vehicle development.

Offline optimization of the power management strategy under testing cycles is essential to help automakers comply with legislations. Offline optimization determines the optimal settings that achieve the maximum energy efficiency [9], where dynamic programming (DP) is considered as the benchmark method. However, DP requires large computational efforts and, therefore, is not feasible to be implemented in real-time control directly [10], [11].

Implementing the DP results in real-time control is critical to enable optimal power management [12]. This will be achieved by finding the optimal parameters in power management control models that achieve the minimum mean square error (MSE) with the DP results. Meta-heuristic algorithms, e.g., particle swarm optimization (PSO) [13]–[16] and genetic algorithms (GAs) [17], [18], have been developed to minimize the MSEs between model data and testing data. The learning performance heavily depends on the data used in training and validation. Khayyam and Bab-Hadiashar [18]

Manuscript received July 28, 2020; revised December 29, 2020 and April 5, 2021; accepted June 25, 2021. Date of publication July 14, 2021; date of current version December 1, 2021. This work was supported in part by the State Key Laboratory of Automotive Safety and Energy under Project KF2029, in part by Innovate U.K. under Grant 102253, and in part by U.K. Engineering and Physical Science Research Council (EPSRC) Innovation Fellowship under Grant EP/S001956/1. *(Corresponding author: Hongming Xu.)*

Quan Zhou and Hongming Xu are with the School of Engineering, University of Birmingham, Birmingham B15 2TT, U.K., and also with the State Key Laboratory of Automotive Safety and Energy, Tsinghua University, Beijing 10084, China (e-mail: h.m.xu@bham.ac.uk).

Dezong Zhao is with the James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, U.K.

Bin Shuai and Huw Williams are with the School of Engineering, University of Birmingham, Birmingham B15 2TT, U.K.

Yanfei Li is with the State Key Laboratory of Automotive Safety and Energy, Tsinghua University, Beijing 10084, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3093429.

Digital Object Identifier 10.1109/TNNLS.2021.3093429

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

modeled a fuzzy logic power management controller using five groups of datasets, while the size of each dataset is 30k. Tian *et al.* [19] used 1120 datasets to train a fuzzy power management controller, while the size of each dataset is more than 4k. Xing *et al.* [20] used 10k data to train recurrent neural networks for driver behavior prediction. These articles demonstrated model learning based on a huge amount of data, but such approaches are time consuming and may cause overfitting.

Building precise and robust control models with limited source data is challenging for knowledge implementation of the offline optimization results. Ideally, the optimal model built from knowledge implementation will use the data collected from the test cycle (e.g., WLTC and RTS-95) [8]. Cross-validation is a statistical method to estimate the robustness of machine-learning models [21]. It divides the source data into the training dataset and the validation dataset. The K-fold cross validation is widely used for learning with labeled data [21]. Lv *et al.* [22] implemented a fivefold cross validation to train a neural network for driver intention prediction. Zuo *et al.* [23] developed a fivefold method to train a fuzzy model in solving regression problems. Tivive used a tenfold method to train a convolutional neural network for pattern recognition [24]. However, using K-fold cross-validation methods for power management has not been reported.

Online optimization is necessary for hybrid powertrain control because only limited future-trip information is available in real-world driving. Model predictive control (MPC) has been applied in online optimization of energy management strategies [25]–[28]. It works on a rolling basis to generate the optimal control policy based on the vehicle model [29]. Because the vehicle model is normally fixed with offline calibrated results, MPC is less adaptive in noncalibrated conditions, e.g., real-world driving [30], [31].

Reinforcement learning (RL) is an emerging and promising technology for online optimal control [32]. It is a plant-model-free method based on Bellman's theory [33], which updates its knowledge based on reinforcement information to fulfill online optimization in unknown environments [10]. The effectiveness of RL has been demonstrated in various vehicle-control-related applications [34]. Remarkable improvements in vehicle energy efficiency have been achieved by RL methods, e.g., Q-learning [35], deep Q-learning [36], double Q-learning [37], and multiple-step Q-learning [38]. Most research on RL-based power management control focus on learning from scratch [39], [40]. However, this approach requires a long time to develop a proper control policy, so it is not practical in real-world applications [41].

The learning speed and performance of the RL can be theoretically improved if the offline optimization knowledge can be translated into online learning. This can be formulated as a transfer learning paradigm, which leverages the previously acquired knowledge (e.g., offline optimization) to improve the efficiency and accuracy of learning in another domain (e.g., real-world driving). The transferred knowledge include characteristics [42], feature representations [43], model parameters [44], and relational information [45]. Recently, adaptive neuro-fuzzy inference systems (ANFIS) have elaborated superiorities in transfer learning as they incorporate heuristic human knowledge with data for accurate modeling of uncertainties [46]–[48]. There has been a big volume of research on applying transfer learning to solve classification problems [49]. However, employing transfer learning to improve the real-time control performance is scarcely reported.

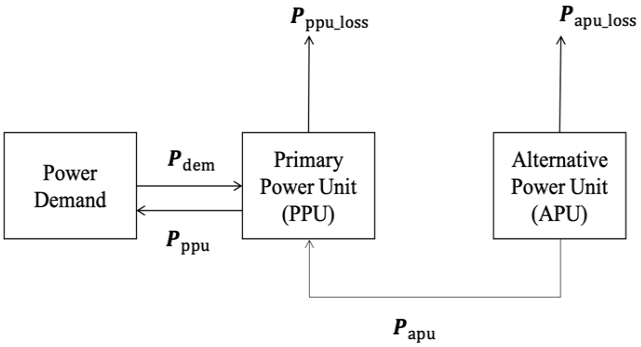


Fig. 1. Power flow of a hybrid powertrain system.

To enable knowledge implementation and transfer in power management of the plug-in hybrid electric vehicle (PHEV), this article proposes an adaptive learning network, which incorporates a deep deterministic policy gradient (DDPG) network [50] with an ANFIS network. Experimental evaluations are conducted to show how advanced artificial neural networks and learning systems help improve control performance in real-world driving. This work has two main contributions:

1) A new method named global K-fold fuzzy learning (GKFL) is proposed to build the ANFIS network that is used to implement the knowledge learned from offline DP to real-time control.
2) The DDPG network is combined with the ANFIS to optimize the control performance in real-world driving with the capability of online knowledge reinforcement.

The rest of the article is organized as follows. Section II describes the energy flow within a PHEV PMS, and Section III proposes the adaptive learning network for power management. Experimental evaluations are conducted in Section IV. Conclusions are summarized in Section V.

II. Energy Flow Optimization and Power Management

This section formulates the optimization problem in power management based on energy flow modeling. Afterward, a baseline PMS for the PHEV is introduced.

A. Energy Flow Optimization

In general, a PHEV consists of two power units (e.g., battery package and engine-generator) to match the power demand for vehicle operations. The power flows of power units are shown in Fig. 1, where P_{dem} is the power demand for vehicle operation; P_{ppu} is the power output from the battery pack; the battery is discharging when $P_{\text{ppu}} > 0$, and is charging when $P_{\text{ppu}} < 0$; P_{apu} is the power output from the engine generator. The battery package works as the primary power

unit of the PHEV. The engine generator is the alternative power unit for maintaining the battery's state-of-charge (SoC) to ensure longer driving distance. The PHEV has three working modes including the pure electric drive mode, the hybrid power mode, and the engine-generator power mode. When the battery SoC is relatively high, the powertrain is solely driven by the battery (e.g., the pure electric drive mode). When the battery SoC is relatively low, the engine-generator works at the rated power to guarantee that the battery is not over-discharged (e.g., the engine-generator power mode). Otherwise, both the battery package and engine generator supply power to the PHEV (e.g., the hybrid power mode).

1) Energy Flow Model: From the perspective of energy transmission, the power flow in the PHEV is expressed as

$$P_{\text{dem}}(t) = P_{\text{ppu}}(t) + P_{\text{apu}}(t). \tag{1}$$

Both power units have energy losses when generating electricity. The power losses of the battery and the engine generator can be modeled by

$$\left.\begin{aligned} P_{\text{ppu\_loss}}(t) &= R_{\text{loss}}(\text{SoC}) \cdot I_{\text{batt}}(u_{\text{batt}}(t))^2 \\ P_{\text{apu\_loss}}(t) &= \dot{m}_f\big(u_{\text{egu}}(t)\big) \cdot H_f - P_{\text{apu}}(t) \end{aligned}\right\} \tag{2}$$

where R_{loss} is the battery internal resistance; I_{batt} is the battery current; u_{batt} is the battery control signal; u_{egu} is the engine-generator control signal; $\dot{m}_f$ is the fuel mass flow rate; and H_f is the heat value of gasoline.

Achieving maximum vehicle energy efficiency is the primary objective for power management. The energy efficiency of the PHEV is defined as

$$\eta = \frac{\sum_{t=t_0}^{t_\tau} P_{\text{dem}}(t) \cdot \Delta t}{\sum_{t=t_0}^{t_\tau} P_{\text{dem}}(t) \cdot \Delta t + \sum_{t=t_0}^{t_\tau} P_{\text{loss}}(t) \cdot \Delta t} \tag{3}$$

where η is the vehicle energy efficiency; t_0 and t_t are the starting and terminal time of a driving cycle; Δt is the sampling time; and $P_{\text{loss}}(t) = P_{\text{ppu\_loss}}(t) + P_{\text{apu\_loss}}(t)$ is the total power loss.

Maintaining the battery SoC is a critical constraint to be met in power management. The battery SoC at time t_l is

$$\text{SoC}(t_l) = \text{SoC}(t_{l-1}) - \frac{I_{\text{batt}}(u_{\text{batt}}(t_l))}{Q_{\text{batt}}} \cdot \Delta t \tag{4}$$

where Q_{batt} and I_{batt} are the capacity and current of the battery, respectively.

2) Energy Flow Optimization Problem Formulation: To achieve the maximum vehicle energy efficiency while maintaining the battery SoC, a control utility (CU) function is defined by introducing $\mathbb{p}(\text{SoC}(t))$ to the denominator of (3) as

$$\mathcal{U} = \frac{\sum_{t=t_1}^{t_\tau} P_{\text{dem}}(t) \cdot \Delta t}{\sum_{t=t_1}^{t_\tau} P_{\text{dem}}(t) \cdot \Delta t + \sum_{t=t_1}^{t_\tau} \left(P_{\text{loss}}(t) \cdot \Delta t + \mathbb{p}(\text{SoC}(t))\right)} \tag{5}$$

where $\mathbb{p}(\text{SoC}(t)) = \beta \cdot e^{\alpha \cdot (SoC(t) - \text{SoC}^+ - \text{SoC}^-)}$ is the penalty function that is defined based on the degradation of the battery SoC [36]; and SoC^+ and SoC^- are the higher and lower boundaries of battery SoC for the hybrid power mode.

The optimization problem can be formulated by

$$\max_{\begin{bmatrix} \boldsymbol{u}_{\text{batt}} \\ \boldsymbol{u}_{\text{egu}} \end{bmatrix}} \mathcal{U}(\boldsymbol{u}_{\text{bat}}, \boldsymbol{u}_{\text{egu}}, \boldsymbol{P}_{\text{dem}})$$
$$\text{s.t.} \begin{cases} P_{\text{ppu\_loss}}(t) = R_{\text{loss}}(\text{SoC}) \cdot I_{\text{batt}}(u_{\text{batt}}(t))^2 \\ P_{\text{apu\_loss}}(t) = \dot{m}_f\big(u_{\text{egu}}(t)\big) \cdot H_f - P_{\text{apu}}(t) \\ \text{SoC}(t_l) = \text{SoC}(t_{l-1}) - \dfrac{I_{\text{batt}}(u_{\text{batt}}(t_l))}{Q_{\text{batt}}} \cdot \Delta t \\ \text{SoC}^- < \text{SoC}(t) < \text{SoC}^+ \\ t_1 \le t \le t_\tau \end{cases} \tag{6}$$

where $\boldsymbol{u}_{\text{batt}} = [u_{\text{batt}}(t_1), u_{\text{batt}}(t_2), \ldots, u_{\text{batt}}(t_\tau)]$ and $u_{\text{egu}} = [u_{\text{egu}}(t_1), u_{\text{egu}}(t_2), \ldots, u_{\text{egu}}(t_\tau)]$ are vectors of control signals in a driving cycle; and $\boldsymbol{P}_{\text{dem}} = [P_{\text{dem}}(t_1), P_{\text{dem}}(t_2), \ldots, P_{\text{dem}}(t_\tau)]$ is a vector of power demands in a driving cycle.

B. PMS of the PHEV

For the series PHEV, power from the battery and the engine generator yields

$$P_{\text{dem}}(t) = u_{\text{batt}}(t) \cdot P_{\text{ppu\_max}} + u_{\text{egu}}(t) \cdot P_{\text{apu\_max}}. \tag{7}$$

The battery command u_{batt} can be derived from (7) by

$$u_{\text{batt}}(t) = \frac{P_{\text{dem}}(t) - u_{\text{egu}}(t) \cdot \boldsymbol{P}_{\text{apu\_max}}}{u_{\text{batt}}(t) \cdot \boldsymbol{P}_{\text{ppu\_max}}} \tag{8}$$

where $P_{\text{ppu\_max}}$ and $P_{\text{apu\_max}}$ are the maximum powers that can be provided by the battery and engine generator, respectively.

1) Double-Input Single-Output Energy Management System: Typically, energy management of series plug-in hybrid powertrains uses power demand and battery SoC as system inputs to determine the control command of the engine-generator unit $u_{\text{egu}}(t)$ [25], [37], [38]

$$u_{\text{egu}}(t) = \mathcal{M}(P_{\text{dem}}(t), \text{SoC}(t), \mathcal{C}) \tag{9}$$

where $\mathcal{M}(\cdot)$ is a nonlinear mapping function that projects the inputs of $P_{\text{dem}}(t)$ and $\text{SoC}(t)$ to the relevant control command $u_{\text{egu}}(t)$; $\mathcal{C}$ is a vector of parameters for model $\mathcal{M}(\cdot)$. Then, the battery control command $u_{\text{batt}}(t)$ can be calculated using (8).

2) Takagi–Sugeno Fuzzy Inference Network for Energy Management: The energy management strategy is based on a Takagi–Sugeno model, as shown in Fig. 2. This strategy is easy to be implemented with data-driven learning [51]. The battery SoC and power demand from the PHEV are gathered as an input vector $\boldsymbol{x} = [\text{SoC}(t), P_{\text{dem}}(t)]^{\text{T}}$ in the input layer, and the control command $u_{\text{egu}}(t) = y$ is generated in the output layer. The output y is calculated in three hidden layers based on $\boldsymbol{x}$.

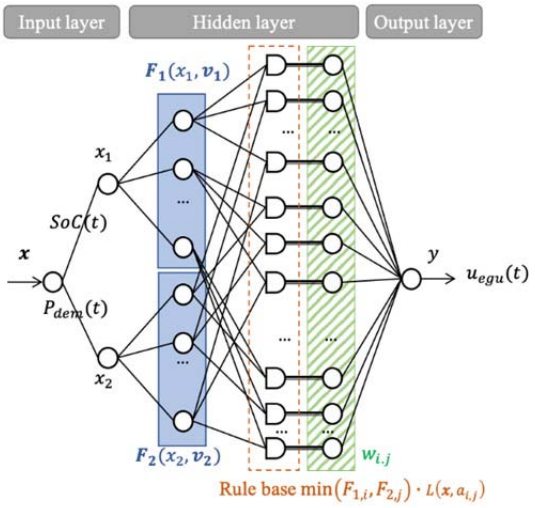


Fig. 2. Takagi–Sugeno fuzzy model for energy management.

The first hidden layer fuzzifies the inputs with triangular membership functions, $F_{1,i}$ and $F_{2,i}$, such that

$$\left.\begin{aligned} &F_{1,i}\left(x_1, \boldsymbol{v}_{1,i}\right) \\ &\quad = \max\left(\min\left(\frac{x_1 - \boldsymbol{v}_{1,i}(1)}{\boldsymbol{v}_{1,i}(2) - \boldsymbol{v}_{1,i}(1)} \quad \frac{\boldsymbol{v}_{1,i}(3) - x_1}{\boldsymbol{v}_{1,i}(3) - \boldsymbol{v}_{1,i}(2)}\right), 0\right) \\ &F_{2,j}\left(x_2, \boldsymbol{v}_{2,j}\right) \\ &\quad = \max\left(\min\left(\frac{x_2 - \boldsymbol{v}_{2,j}(1)}{\boldsymbol{v}_{2,j}(2) - \boldsymbol{v}_{2,j}(1)} \quad \frac{\boldsymbol{v}_{2,j}(3) - x_2}{\boldsymbol{v}_{2,j}(3) - \boldsymbol{v}_{2,j}(2)}\right), 0\right) \end{aligned}\right\} \tag{10}$$

where $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m$; x_1 and x_2 are the first and second element of $\boldsymbol{x}$; $F_{1,i}$ is the ith membership function for the first input; $F_{2,j}$ is the jth membership function for the second input; and $\boldsymbol{v}(k)$, $k = 1,2,3$, is the kth element of $\boldsymbol{v}$.

The second hidden layer connects the outputs of the input membership functions based on fuzzy rules. Each fuzzy rule applies the following linguistic logic:

$$\text{If } \boldsymbol{x}(1) \text{ is } F_{1,i}(\boldsymbol{x}(1), \boldsymbol{v}_{1,i}) \text{ and } \boldsymbol{x}(2) \text{ is } F_{2,j}(\boldsymbol{x}(2), \boldsymbol{v}_{2,j}) \text{ then } y \text{ is } L(\boldsymbol{x}, a_{i,j}) \tag{11}$$

where $L(\boldsymbol{x}, a_{i,j})$ is the output membership function that maps the $\boldsymbol{x}$ and $a_{i,j}$ to a constant [11]; and $a_{i,j}$ is a scaling factor.

The third hidden layer uses a vector of weighting values $\boldsymbol{W} = [w_{1.1}, w_{1.2}, \ldots, w_{1.n}, w_{2.1}, \ldots, w_{2.n}, \ldots, w_{m.1}, \ldots, w_{m.n}]$ to scale the outputs of fuzzy rules

$$y = \sum_{j=1}^{m} \sum_{i=1}^{n} \left\{\min\left(F_{1,i}, F_{2,j}\right) \cdot L\left(\boldsymbol{x}, a_{i,j}\right) \cdot w_{i,j}\right\} \tag{12}$$

where $w_{i.j} \in [0, 1]$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$.

For the generic controller, the model parameter vector $\mathbb{C}$ is

$$\mathbb{C} = [\boldsymbol{V}_1, \boldsymbol{V}_2, \boldsymbol{A}, \boldsymbol{W}] \tag{13}$$

where $\boldsymbol{V}_1 = [\boldsymbol{v}_{1,1}, \boldsymbol{v}_{1,2}, \ldots, \boldsymbol{v}_{1,n}]$ and $\boldsymbol{V}_2 = [\boldsymbol{v}_{2,1}, \boldsymbol{v}_{2,2}, \ldots, \boldsymbol{v}_{2,m}]$ are the parameter vectors of the inputs membership functions; $\boldsymbol{A}_1 = [a_{1,1}, \ldots, a_{1,n}, a_{2,1}, \ldots, a_{2,n}, \ldots, a_{m,n1}, \ldots, a_{m,n}]$ is a vector of parameter in the output membership functions. It should be noticed that the vector $\mathbb{C}$ cannot be determined using conventional gradient-based algorithms because the ANFIS model includes components that are not derivable. Therefore, $\mathbb{C}$ should be obtained by using the derivative-free algorithm.

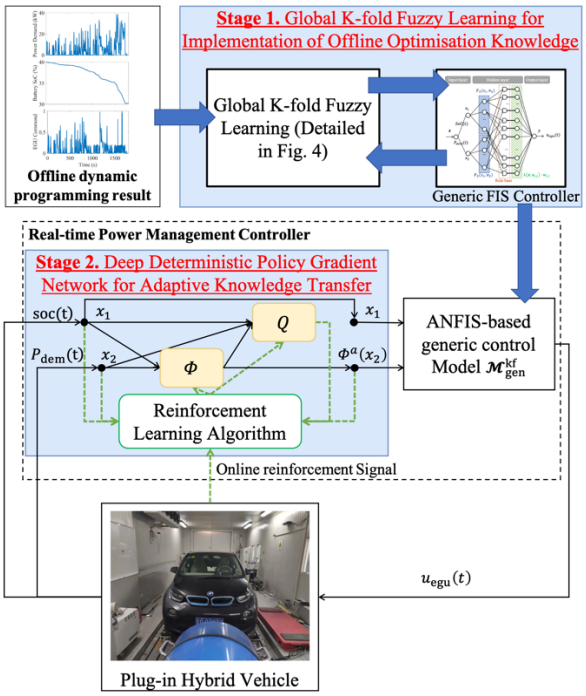


Fig. 3. Adaptive learning network in power management of the plug-in hybrid vehicle.

III. Adaptive Learning Network for Power Management

This section introduces the development procedure for the adaptive learning network, as shown in Fig. 3. This procedure incorporates a DDPG network with an ANFIS network to enable knowledge implementation and transfer for real-time energy management with two stages. In Stage 1, a new GKFL method is developed to implement the offline optimization results in the ANFIS-based generic control model. In Stage 2, the DDPG network is developed for transfer learning, which regulates the inputs of the ANFIS network with the reinforcement signals from real-world driving. Integration of the development in both stages ensures robust and efficient online learning that can be conducted in the onboard power management controller. Therefore, the vehicle's CU can be continuously improved in real-world driving.

A. GKFL for Implementation of Offline Optimization Knowledge in Real-Time Control

The GKFL method is proposed to implement the offline optimization knowledge in a control model $\mathcal{M}^{kf}$. It conducts k-fold cross validation for model learning with the offline DP

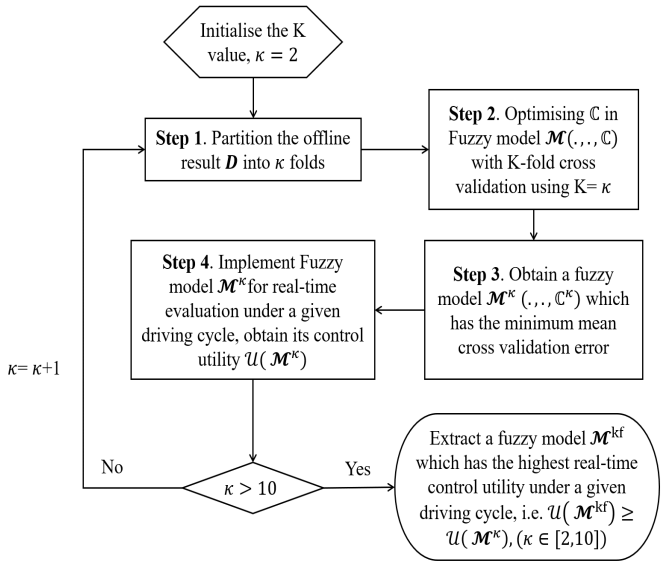


Fig. 4. Procedure of GKFL for implementing the offline optimization result into real-time control.

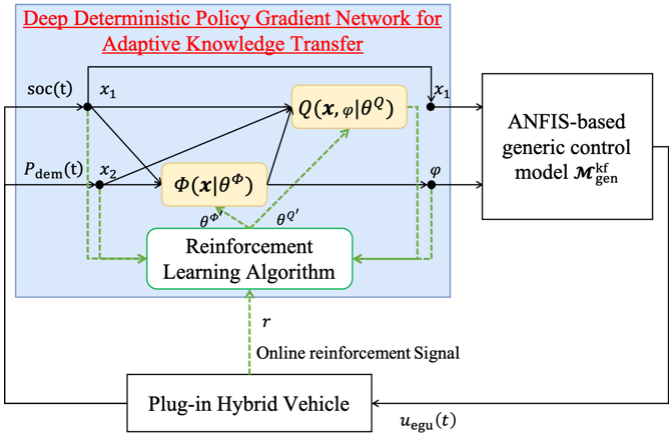


Fig. 5. DDPG network for adaptive knowledge transfer.

results to achieve the maximum CU value, $\mathcal{U}(\mathcal{M}^{\mathrm{kf}})$, in real-time.

Remark 1: The proposed GKFL method can simultaneously determine the optimal energy management model $\mathcal{M}^{\mathrm{kf}}$ and the optimal setting, κ^*, by conducting a global search with all possible κ values (e.g., $\kappa = 2, 3, 4, \ldots, 10$). Therefore, the study on GKFL is important to contribute know-how on selection of K values for cross-validation-based model learning because it is scarcely studied in this field. The working procedure of the proposed GKFL is illustrated in Fig. 4.

After an initialization of the K value by setting $\kappa = 2$, a rotational model learning process will be conducted by repeating Steps 1–4:

Step 1: The dataset of the offline DP results under a given driving cycle, $\boldsymbol{D} = [\boldsymbol{x}, y]^{\mathrm{T}}$, is divided into κ folds randomly, i.e., $\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots, \boldsymbol{D}_\kappa$, with similar data size.

Step 2: The parameter vector C in fuzzy model $\mathcal{M}$ is optimized using the derivative-free algorithm (e.g., Generic Algorithm) in κ rounds. For each round, $\boldsymbol{D}^r_{\mathrm{trn}} = [\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots, \boldsymbol{D}_{r-1}, \boldsymbol{D}_{r+1}, \ldots, \boldsymbol{D}_\kappa]$ $(r = 1, 2, 3, \ldots, \kappa)$ is used for training and $\boldsymbol{D}^r_{\mathrm{tst}} = \boldsymbol{D}_r$ is used for testing.

Step 3: A fuzzy model $\mathcal{M}^\kappa(.,.,\mathbb{C}^\kappa)$ is selected based on the results from Step 2, and it has the minimum cross-validation MSE (CV-MSE)

$$\mathrm{CVMSE}_{(\kappa)} = \frac{1}{\kappa} \cdot \sum_{r=1}^{\kappa} \frac{\sum_{t=1}^{\tau'} \left(\mathcal{M}^r(x^r_{\mathrm{tst}}(t)) - y^r_{\mathrm{tst}}(t)\right)^2}{\tau'} \quad (14)$$

where $\mathcal{M}^r(x^r_{\mathrm{tst}}(t))$ is the model output at time t using the model learned from training data $\boldsymbol{D}^r_{\mathrm{trn}}$ during round r; and $x^r_{\mathrm{tst}}(t)$ and $y^r_{\mathrm{tst}}(t)$ are, respectively, the model input and output at time t in testing dataset $\boldsymbol{D}^r_{\mathrm{tst}}$ during round r.

Step 4: The fuzzy model $\mathcal{M}^\kappa$ is implemented for real-time energy management control under the given driving cycle. The CU value, $\mathcal{U}(\mathcal{M}^\kappa)$, is collected as an indicator to select the optimal learning result.

Once the termination term is met (i.e., $\kappa > 10$), the rotational process stops. Thereafter, the optimal setting κ^* is extracted, and the optimal model $\mathcal{M}^{\mathrm{kf}}$ is generated to satisfy

$$\mathcal{U}(\mathcal{M}^{\mathrm{kf}}) = \mathcal{U}(\mathcal{M}^{\kappa^*}) \geq U(\mathcal{M}^\kappa), \quad \kappa \in [2, 10] \quad (15)$$

where $\mathcal{U}(\mathcal{M}^{\kappa^*})$ is the CU value that the vehicle achieved under a driving cycle with the optimal fuzzy model $\mathcal{M}^{\kappa^*}$ $(\kappa = \kappa^*)$.

B. DDPG Network for Adaptive Knowledge Transfer

To make the ANFIS network more adaptive to new driving conditions that has not been used for model learning, the input signals of the ANFIS need to be regulated. This will be achieved by developing a nonlinear input space mapping network that has the capability of online RL with real-time feedback.

Remark 2: Online adaptive knowledge transfer is enabled by the DDPG network, which has the capability of self-learning based on real-time reinforcement signals. As illustrated in Fig. 5, the DDPG network will allow the generic model $\mathcal{M}^{\mathrm{kf}}_{\mathrm{gen}}$ to adapt the new driving scenarios by regulating the power demand x_2 to a corresponding level φ.

The DDPG network uses a deep critic network Q to optimize the deep actor-network Φ online. The actor-network Φ maps a deterministic action execution policy, and a noise $N^{\mathcal{E}}$ is added to enable the exploration, such that

$$\varphi(t) = \Phi\left(\boldsymbol{x}(t)|\boldsymbol{\theta}^\Phi\right) + \mathcal{N}^{\mathcal{E}} \quad (16)$$

where $\varphi(t)$ is the regulated state variable at time t; $\boldsymbol{x}(t)$ is a vector of inputs at time t; $\boldsymbol{\theta}^\Phi$ is a vector of parameters in the neural network Φ; and $\mathcal{N}^{\mathcal{E}} \sim N(0, {\sigma_{\mathcal{E}}}^2)$ is the exploration noise generated by a random number generator in MATLAB, and it enables more global exploration at the early stage of the learning process and guarantee convergence of the control policy. The derivation value $\sigma_{\mathcal{E}}$ of the exploration noise decreases along with the online learning progress [52]

$$\sigma_{\mathcal{E}} = \sigma_0 - \Delta\sigma \cdot E \quad (17)$$

where σ_0 is the initial variation value; $\Delta\sigma$ is the variance decay rate; and $\mathcal{E}$ is the number of epis ParaFirstLine-Indodes that the

learning agent experienced in online learning. Once the power demand signal is regulated by (16), the control command is represented by

$$u_{\text{egu}}(t) = \sum_{j=1}^{m}\sum_{i=1}^{n}\Big\{\min\big(\boldsymbol{F}_{1,i}(\text{SoC}(t)), \boldsymbol{F}_{2,j}(\varphi(t))\big) \cdot L\big([\text{SoC}(t), \varphi(t)]^{\text{T}}, a_{i,j}\big) \cdot w_{i,j}\Big\}. \quad (18)$$

Subsequently, the vehicle's fuel mass flow rate and battery's open-circuit voltage (for battery SoC estimation) are measured and used as feedback to the controller. To guarantee the convergence of the algorithm, the reinforcement signal at each time step is defined based on (6)

$$r(t) = -\Big(P_{\text{loss}}(t) \cdot \Delta t + \beta \cdot e^{\alpha\cdot\left(\text{SoC}(t)-\text{SoC}^{+}-\text{SoC}^{-}\right)}\Big). \quad (19)$$

Multiple signals are restored in a replay buffer R, including the real-time state signals $\boldsymbol{x}(t) = [\text{SoC}(t), P_{\text{dem}}(t)]^{\text{T}}$, the action signal $\varphi(t)$, and the reinforcement signal $r(t)$. In each time interval, a minibatch of N transitions $(\boldsymbol{x}_i, \varphi_i, r_i, \boldsymbol{x}_{i+1})$ is created by random sampling from $\mathbb{R}$ to train both the actor and critic networks.

The parameter vector in the critic network, $\boldsymbol{\theta}^{Q}$, is optimized by minimizing the loss, L, such that

$$L = \frac{\sum_{i=1}^{N}\left(\hat{Q}_i - Q\left(\boldsymbol{x}_i, \varphi_i|\boldsymbol{\theta}^{Q}\right)\right)}{N} \quad (20)$$

where $Q(\boldsymbol{x}_i, \varphi_i|\boldsymbol{\theta}^{Q})$ is a deep critic network with parameter vector $\boldsymbol{\theta}^{Q}$, which calculates the merit function value using vehicle state, $\boldsymbol{x}_i$, and action, φ_i. Based on Bellman's theory [33], the estimated merit function value $\hat{Q}_i$ is calculated with the reinforcement signal, such that

$$\hat{Q}_i = r_i + \gamma \cdot Q'\Big(\boldsymbol{x}_{i+1}, \Phi'\Big(\boldsymbol{x}_{i+1}|\boldsymbol{\theta}^{\Phi'}\Big)|\boldsymbol{\theta}^{Q'}\Big) \quad (21)$$

where Q' with its parameter vector $\boldsymbol{\theta}^{Q'}$ is an estimated target critic network; Φ' with its parameter vector $\boldsymbol{\theta}^{\Phi'}$ is an estimated target actor network; and γ is the learning rate.

The parameter vector in actor network, $\boldsymbol{\theta}^{\Phi}$, is optimized based on the deterministic policy gradient theory [52], such that

$$\boldsymbol{\theta}^{\Phi} \leftarrow \boldsymbol{\theta}^{\Phi} + \alpha \cdot \nabla_{\boldsymbol{\theta}^{\Phi}} J\left(\Phi\left(\boldsymbol{x}|\boldsymbol{\theta}^{\Phi}\right)\right) \quad (22)$$

where $\nabla_{\boldsymbol{\theta}^{\Phi}} J(\Phi(\boldsymbol{x}|\boldsymbol{\theta}^{\Phi}))$ is the policy gradient, i.e., the gradient of vehicle performance J using the parameter vector $\boldsymbol{\theta}^{\Phi}$ in the actor network Φ; α is the learning rate. The policy gradient can be estimated using the minibatch data as

$$\nabla_{\boldsymbol{\theta}^{\Phi}} J\left(\Phi\left(\boldsymbol{x}|\boldsymbol{\theta}^{\Phi}\right)\right) = \frac{\sum_{i=1}^{N}\left[\nabla_{\boldsymbol{\theta}^{\Phi}}\Phi\left(\boldsymbol{x}_i|\boldsymbol{\theta}^{\Phi}\right)\cdot\nabla_{\Phi\left(\boldsymbol{x}_i|\boldsymbol{\theta}^{\Phi}\right)} Q\left(\boldsymbol{x}_i, \Phi\left(\boldsymbol{x}_i|\boldsymbol{\theta}^{\Phi}\right)|\boldsymbol{\theta}^{Q}\right)\right]}{N} \quad (23)$$

where $\nabla_{\boldsymbol{\theta}^{\Phi}}\Phi(\boldsymbol{x}_i|\boldsymbol{\theta}^{\Phi})$ is the gradient of the actor network Φ with regard to $\boldsymbol{\theta}^{\Phi}$ as variables; and $\nabla_{\Phi(\boldsymbol{x}(i)|\boldsymbol{\theta}^{\Phi})} Q(\boldsymbol{x}_i, \Phi(\boldsymbol{x}_i|\theta^{\Phi})|\theta^{Q})$ is the gradient of the critic network Q with regard to $\Phi(\boldsymbol{x}_i|\boldsymbol{\theta}^{\Phi})$ as variables. The pseudo-code for the DDPG algorithm for knowledge transfer is shown in Fig. 6.

DDPG Algorithm for Knowledge Transfer in PHEV Power Management

Randomly initialise critic network $\text{Q}\left(\boldsymbol{x}, \varphi|\boldsymbol{\theta}^{\text{Q}}\right)$ and actor network $\Phi\left(\boldsymbol{x}|\boldsymbol{\theta}^{\Phi}\right)$ with parameter vectors $\boldsymbol{\theta}^{\text{Q}}$ and $\boldsymbol{\theta}^{\Phi}$
Initialise target critic network Q' and target actor network Φ' with parameter vectors $\boldsymbol{\theta}^{Q'} \leftarrow \boldsymbol{\theta}^{\text{Q}}$and $\boldsymbol{\theta}^{\Phi'} \leftarrow \boldsymbol{\theta}^{\Phi}$
Initialise replay buffer $\mathbb{R}$
For $\mathcal{E}$=1, M **do**
  Initialise a random process $\mathcal{N}^{\mathcal{E}} \sim N(0, {\sigma_{\mathcal{E}}}^2)$ with (17)
  Receive initial observation state $\boldsymbol{x}(1)$
  For t=1, T **do**
    Select an action $\varphi(t) = \Phi\left(\boldsymbol{x}(t)|\boldsymbol{\theta}^{\Phi}\right) + \mathcal{N}^{\mathcal{E}}$
    Obtain the vehicle control signal using (18) and measure vehicle fuel mass flow rate and battery voltage
    Calculate reinforcement signal use (19)
    Observe new state $\boldsymbol{x}(t+1)$
    Store transitions $(\boldsymbol{x}(t), \varphi(t), r(t), \boldsymbol{x}(t+1))$ in $\mathbb{R}$
    Sample a random minibatch of N transitions $(\boldsymbol{x}_i, \varphi_i, r_i, \boldsymbol{x}_{i+1})$ from $\mathbb{R}$
    Estimate merit function value $\hat{Q}_i$ with (21)
    Update $\boldsymbol{\theta}^{\text{Q}}$ by minimising $L = \frac{\sum_{i=1}^{N}(\hat{Q}_i - Q(\boldsymbol{x}_i, \varphi_i|\boldsymbol{\theta}^{\text{Q}}))}{N}$
    Update $\boldsymbol{\theta}^{\Phi}$ by using the policy gradient in (23)
    Update $\boldsymbol{\theta}^{Q'} \leftarrow \tau \cdot \boldsymbol{\theta}^{\text{Q}} + (1-\tau) \cdot \boldsymbol{\theta}^{Q'}$and $\boldsymbol{\theta}^{\Phi'} \leftarrow \tau \cdot \boldsymbol{\theta}^{\Phi} + (1-\tau) \cdot \boldsymbol{\theta}^{\Phi'}$;
  End For
End For

Fig. 6. Pseudocode of adaptive knowledge transfer with DDPG.

TABLE I
KEY PARAMETERS FOR VEHICLE PLANT MODELING

Specification	Value	Unit
Vehicle Mass	1315	kg
Wheel rolling radius	0.35	m
Front Area	2.38	m^2
Drag coefficient	0.30	-
Rolling resistance	0.001	-

IV. EXPERIMENTAL EVALUATIONS

The experimental evaluations were conducted on a plug-in hybrid passenger car, which has a 36.6-kW engine generator with a 0.65L engine, a 125-kW electric motor, and a 360-V high-volt battery with the capacity of 22 kWh. The vehicle is modeled using the Simulink Powertrain Toolbox based on the dynamometer data. The inputs of the vehicle model are the desired vehicle speed and the power management control signals. The model outputs are the battery SoC, battery voltage/current, fuel mass flow rate, and power demand for vehicle operation. The key parameters are listed in Table I.

Both offline software-in-the-loop (SiL) and online hardware-in-the-loop (HiL) testing platforms were used in experimental evaluations. The offline evaluation was conducted in MATLAB 2020a on a PC with an i7 CPU and a 16-GB RAM. A Speedgoat real-time target machine (Intel Core i7 2.5 GHz with 4 GB RAM) is used for online HiL testing, as shown in Fig. 7. The control prototype and the real-time vehicle plant model are compiled on a host PC, downloaded onto the Speedgoat target machine through

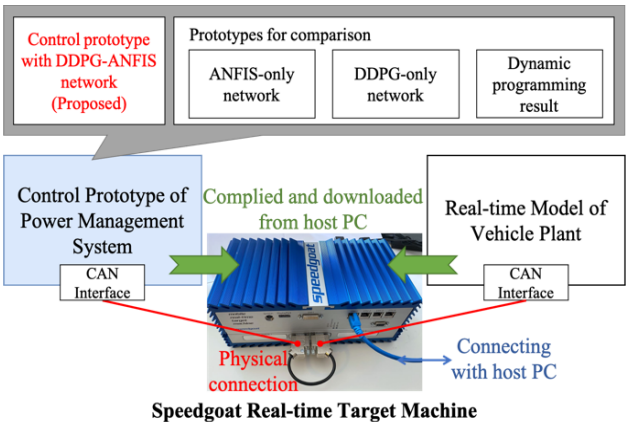


Fig. 7. Online HiL testing platform.

Ethernet, and physically connected using controller area networks (CANs). The proposed DDPG-ANFIS network was implement in the prototype controller with two steps: 1) the network was built with MATLAB/Simulink with its inputs/outputs connected to the CAN interface blocks that are provided in the Simulink real-time block set and 2) the Simulink model was compiled into executive code with the Simulink code generator for real-time control. The prototype controllers are developed in the same way with the ANFIS-only network, DDPG-only network, and a lookup table system (built with DP result).

TABLE II
KNOWLEDGE IMPLEMENTATION PERFORMANCE WITH GA

Learning Method	Learning performance		Veri. MSE	CU value
	Train. MSE	Min. CV-MSE		
Conv-GA	**0.0895**	**-**	**0.0941**	**0.2674**
GKFL-2-GA	-	0.1086	0.1157	0.2019
GKFL-3-GA	-	0.1144	0.1298	0.2571
GKFL-4-GA	-	0.1102	0.1101	0.2518
GKFL-5-GA	**-**	**0.0939**	**0.0932**	**0.2778**
GKFL-6-GA	-	0.0978	0.0911	0.2667
GKFL-7-GA	-	0.0975	0.1018	0.2486
GKFL-8-GA	-	0.1046	0.1017	0.2078
GKFL-9-GA	**-**	**0.0908**	**0.0924**	**0.2802**
GKFL-10-GA	-	0.0970	0.0993	0.2213

TABLE III
KNOWLEDGE IMPLEMENTATION PERFORMANCE WITH PSO

Learning Method	Learning Performance		Veri. MSE	CU value
	Train. MSE	Min. CV-MSE		
Conv-PSO	**0.0925**	**-**	**0.0955**	**0.2573**
GKFL-2-PSO	-	0.0928	0.0953	0.2462
GKFL-3-PSO	-	0.0965	0.0980	0.2462
GKFL-4-PSO	-	0.0969	0.0955	0.2383
GKFL-5-PSO	**-**	**0.0938**	**0.0912**	**0.2780**
GKFL-6-PSO	-	0.0927	0.0979	0.2680
GKFL-7-PSO	-	0.0898	0.0913	0.2777
GKFL-8-PSO	-	0.0909	0.0909	0.2508
GKFL-9-PSO	**-**	**0.0931**	**0.0904**	**0.2789**
GKFL-10-PSO	-	0.0954	0.0934	0.2356

A. Knowledge Implementation From Offline Optimization

Experimental evaluation of the knowledge implementation performance was conducted under the WLTC, which is currently used for vehicle certification. The benchmark power management result was obtained by DP, which is a dataset containing 1800 data pairs. A total of 70% of DP data was used for learning and 30% was for verification. The advantage of the proposed GKFL method was demonstrated by comparing the model learning performance with the conventional method (default in MATLAB ANFIS toolbox). The conventional method used the whole learning data, whereas GKFL further divided the learning data into κ folds ($\kappa = 2, 3,\ldots,10$) of training and validation data pairs. To ensure the effectiveness of learning results, GA and PSO algorithms were both used for model learning and the results are compared in Tables II and III.

Three performance metrics were used to testify the robustness of the proposed GKFL method including the learning performance, the verification MSE (Veri. MSE), and the real-time CU value. Training MSEs (Train. MSEs) and minimum CV-MSE (Min. CV-MSE) were used to evaluate the learning performance of the conventional method and the GKFL methods (with different κ values), respectively. The experimental evaluation suggests that the optimal power management control model under the WLTC can be obtained by the proposed GKFL method when $\kappa = 9$. It achieves the minimal Veri. MSE and highest CU values. This indicates that the result of fuzzy model learning using both GA and PSO is robust. The number of folds for cross-validation-based statistical learning should be carefully chosen to robustly achieve the optimal result. In this study, fuzzy learning based on the widely used fivefold cross validation can achieve acceptable learning performance, which achieves higher CU value compared to the conventional method. However, the performance using another widely used method, i.e., tenfold cross validation, is not as good as expected because it achieves a less CU value than the conventional method.

By implementing the control models in the HiL testing platform, the real-time performance of the fuzzy model obtained by GKFL-9-GA ($\kappa = 9$, using GA for learning) is compared in Fig. 8 with both benchmark strategy (obtained by DP) and the model obtained by conventional Conv-PSO (ANFIS toolbox with PSO). The PMS generates control commands for the engine generator unit (EGU) as shown in Fig. 8(b). It is to satisfy the power demand in Fig. 8(a) while maintaining the battery SoC at a certain level as shown in Fig. 8(c). It optimizes the vehicle energy efficiency by minimizing fuel consumption. The power management real-time control model obtained by GKFL-9-GA achieves 2.8% lower fuel consumption than using the benchmark strategy. However, this approach achieves 3.4% lower CU value because it has 5.9% less remaining battery SoC. The fuzzy model obtained by GKFL-9-GA achieves the highest CU value compared to other learning methods. The fuzzy model is chosen to implement the offline optimization knowledge for the rest of this article.

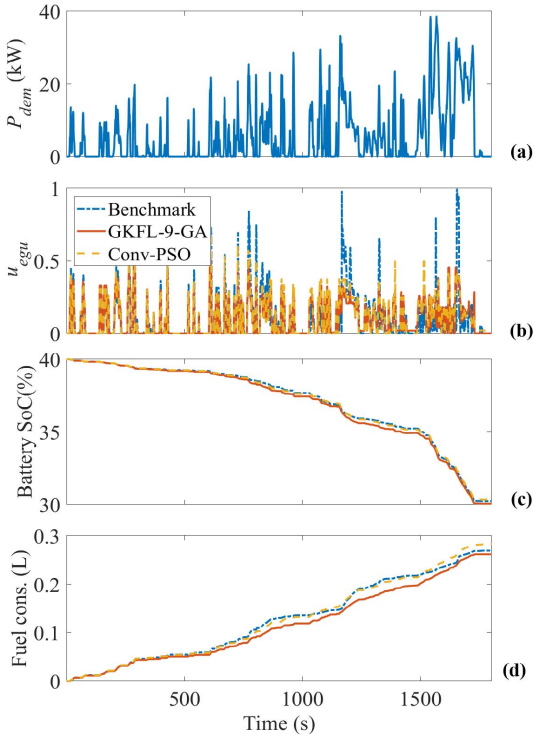


Fig. 8. Real-time performance under WLTC. (a) Power demand for vehicle operation. (b) EGU control command. (c) battery SoC. (d) Fuel consumptions.

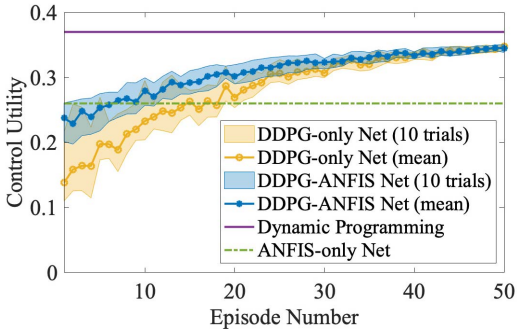


Fig. 9. Online learning progress for knowledge transfer.

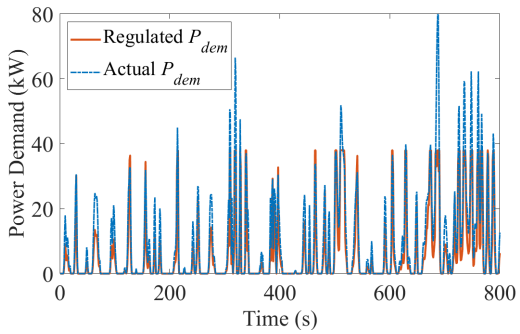


Fig. 10. Regulation of power demand in the RTS-95 cycle.

B. Knowledge Transfer Across Different Testing Cycles

The experimental evaluation on knowledge transfer performance is conducted under the RTS-95 cycle, which is recommended by the EU commission to emulate real-world driving conditions [8]. The practical significance of this study is to show how advanced learning networks help guarantee vehicle compliance with different legislations. DP was conducted to obtain the benchmark CU over the RTS-95 cycle.

The ANFIS network obtained by GKFL-9-GA was chosen as the generic energy management model, and a DDPG network was connected in front of the ANFIS network as described in Section III-B. Two baseline energy management systems developed based on a DDPG-only network and an ANFIS-only network, respectively, were used for comparison of the learning progresses shown in Fig. 9. Each learning episode (containing 800 steps with 1 s step length) runs repetitively under the RTS-95 cycle with an initial battery SoC of 40%. Because online learning contains random factors, the learning of using both DDPG–ANFIS and DDPG-only networks is repeated ten times independently. This is to examine their CU value at the end of each episode.

The blue-colored area covers all the CU values that were achieved in each episode of the ten trials when using the DDPG–ANFIS network for power management. The blue dot-curve shows the average CU values achieved by the DDPG–ANFIS network. The yellow-colored area covers the CU values achieved by the DDPG-only network and the yellow dot-curve comprises their mean values. The DDPG–ANFIS network has a greater opportunity to gain better CU values in the early stage with the transferred knowledge. Both DDPG–ANFIS and DDPG-only networks develop their policy to achieve better CU from real-world interactions. They can converge to the same CU level that is closed to the DP result. The converged CU is 38% higher than that obtained by the ANFIS-only network.

To monitor how the DDPG–ANFIS network enables adaptive transfer learning in the RTS-95 cycle, the original power demand and the regulated power demand (the output of the DDPG subnetwork) are illustrated in Fig. 10. The data were collected at the 50th episode, where the DDPG–ANFIS network had developed a proper control policy for power management. Because the ANFIS subnetwork transfers the knowledge learned from the WLTC cycle, it achieves the maximum performance in the scenarios where the power demands are between 0 and 40 kW. The proposed DDPG subnetwork has shown a capability to map the power demand into the domains where the ANFIS achieves its optimal performance.

C. Online Learning in Simulated Real-World Conditions

To investigate the feasibility of the proposed DDPG–ANFIS network in real-world learning, evaluations were conducted under simulated real-world driving conditions (SRDCs) with HiL testing. The baseline power management controllers were developed based on DDPG-only network and ANFIS-only network, respectively, for comparison. Five SRDC cycles (SRDC1-5) were generated using the AVL Random Driving Cycle Generator [53], which assume that the speed profiles are collected from different drivers and driving scenarios. The control utilities are measured in 200 learning episodes.

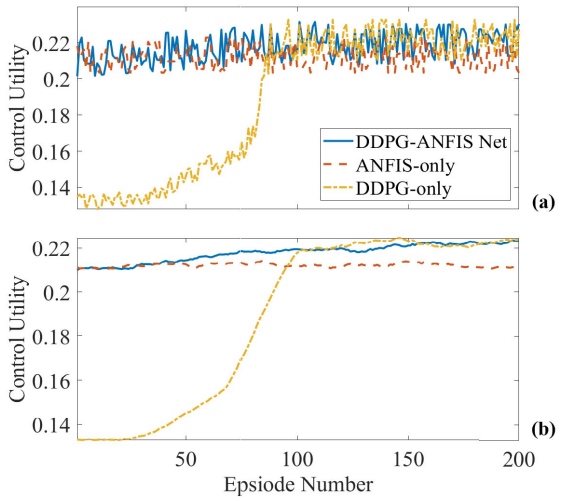


Fig. 11. Learning Performance in the SRDC-1. (a) Row data. (b) Moving average of every 30 episodes.

TABLE IV
ONLINE LEARNING PERFORMANCE IN SRDCS

Cycle name	Mean P_{dem}	Method	Mean CU value		
			All 200	First 50	Last 50
SRDC1	10.55 kW (σ=16.8)	ANFIS-only	0.212	0.212	0.212
		DDPG-only	0.189	0.136	0.222
		DDPG-ANFIS	**0.220**	**0.219**	**0.222**
SRDC2	8.60 kW (σ=12.7)	ANFIS-only	0.199	0.199	0.199
		DDPG-only	0.151	0.099	0.274
		DDPG-ANFIS	**0.261**	**0.259**	**0.273**
SRDC3	5.49 kW (σ=9.54)	ANFIS-only	0.173	0.173	0.173
		DDPG-only	0.352	0.288	0.345
		DDPG-ANFIS	**0.367**	**0.342**	**0.376**
SRDC4	5.03 kW (σ=8.33)	ANFIS-only	0.152	0.153	0.153
		DDPG-only	0.345	0.344	0.345
		DDPG-ANFIS	**0.363**	**0.362**	**0.363**
SRDC5	4.81 kW (σ=8.91)	ANFIS-only	0.162	0.162	0.162
		DDPG-only	0.245	0.091	0.359
		DDPG-ANFIS	**0.323**	**0.166**	**0.378**

Each episode contains 800 learning steps (with 1 s step length), which equals to the RTS-95 cycle. Battery SoC is randomly initialized between 35% and 40%.

The CU achieved by the DDPG–ANFIS network at each episode under SRDC-1 are compared with the ANFIS-only network and the DDPG-only network in Fig. 11(a). To illustrate the trends of CU improvements, moving averages of the control utilities in every 30 episodes are illustrated in Fig. 11(b). In SRDC-1, the moving average of the CU achieved by the ANFIS-only network remains at the same level because the ANFIS subnetwork only transfers the knowledge learned from WLTC but without the online learning capability. The moving average of the CU achieved by both DDPG–ANFIS and DDPG-only networks tends to be continuously improved. Both DDPG–ANFIS and DDPG-only networks achieve similar CU values when they acquired enough experience in real world, but DDPG–ANFIS network performs much better at the early stage.

To testify the robustness of the proposed DDPG–ANFIS network in real-time control, a comparison study with ANFIS-only and DDPG-only networks was conducted under five simulated real-world conditions. The statistical results of the online learning performance are summarized in Table IV. With mean power demands and standard derivations listed in Table IV, SRDC1 simulates a highway drive, which has a relatively higher power demand; SRDC2 simulates a suburban drive, which has combined highway and urban sections; and SRDC3-5 simulates urban driving with varying traffic congestion.

To measure the online performance at different stages, the mean CU values were calculated with 1) all 200 episodes' data for global performance; 2) the first 50 episodes' data for the initial stage; and 3) the last 50 episodes' data for the final stage. Generally, the DDPG–ANFIS network robustly outperforms both ANFIS-only and DDPG-only networks by achieving higher mean CU values under the five SRDC cycles. The highest mean CU value is achieved under SRDC4, which is 138% higher than the ANFIS-only network and 5.2% higher than the DDPG-only network. Because of the prior knowledge implemented in the ANFIS subnetwork and online learning capability enabled by DDPG subnetwork, the DDPG–ANFIS network achieves at least 4.9% higher mean CU values for the first 50 episodes than DDPG-only network. Also, the proposed method achieves more than 4.5% higher mean CU values for the last 50 episodes than the ANFIS-only network.

V. CONCLUSION

This article proposes an adaptive learning network to enable adaptive knowledge implementation and transfer in the power management of a plug-in hybrid vehicle. This network combines a DDPG network with an ANFIS network and the superiorities over both individual networks have been demonstrated. Experimental evaluations have been conducted on both SiL and HiL platforms. The conclusions drawn from this work are as follows.

1) The proposed GKFL for building ANFIS network is effective in the implementation of offline optimization knowledge. The highest CU has been achieved with ninefold fuzzy learning for the studied vehicle, which is 8% higher than using conventional fuzzy learning in the WLTC condition.
2) In the study under RTS-95 cycle representing new legislation conditions, the proposed DDPG–ANFIS network benefits from online reinforcement signals so that it achieved 38% higher CU compared to the ANFIS-only network.
3) In five SRDCs, the proposed DDPG–ANFIS network achieves higher control utilities, which is up to 138% higher than the ANFIS-only network and 5.2% higher than the DDPG-only network.

The planned future work will develop online multiple-objective optimization method to improve energy efficiency, safety, and drivability in autonomous driving. The parameter sensitivity in DDPG-based online learning will also be studied to guarantee the robustness.

REFERENCES

[1] M. A. Raposo, B. Ciuffo, M. Makridis, and C. Thiel, "The revolution of driving: From connected vehicles to coordinated automated road transport (C-ART)," EU Joint Res. Centre, Ispra, Italy, Tech. Rep. JRC106565, 2017, doi: 10.2760/225671.

[2] *Global EV Outlook 2019*, Int. Energy Agency, Paris, France, 2019.

[3] F. Zhang, X. Hu, R. Langari, and D. Cao, "Energy management strategies of connected HEVs and PHEVs: Recent progress and outlook," *Prog. Energy Combustion Sci.*, vol. 73, pp. 235–256, Jul. 2019.

[4] A. A. Malikopoulos, "Supervisory power management control algorithms for hybrid electric vehicles: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1869–1885, Oct. 2014.

[5] J. Pavlovic, A. Marotta, and B. Ciuffo, "CO_2 emissions and energy demands of vehicles tested under the NEDC and the new WLTP type approval test procedures," *Appl. Energy*, vol. 177, pp. 661–670, Sep. 2016.

[6] *Worldwide Emission Standards and Related Regulations*, Continental Automotive GmbH, Hanover, Germany, 2017.

[7] Q. Zhou *et al.*, "Modified particle swarm optimization with chaotic attraction strategy for modular design of hybrid powertrains," *IEEE Trans. Transport. Electrific.*, vol. 7, no. 2, pp. 616–625, Jun. 2021.

[8] E. G. Giakoumis, *Driving and Engine Cycles*. Cham, Switzerland: Springer, doi: 10.1007/978-3-319-49034-2.

[9] K. T. Chau and Y. S. Wong, "Overview of power management in hybrid electric vehicles," *Energy Convers. Manage.*, vol. 43, no. 15, pp. 1953–1968, Oct. 2002.

[10] Y. Hu, W. Wang, H. Liu, and L. Liu, "Reinforcement learning tracking control for robotic manipulator with kernel-based dynamic model," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3570–3578, Sep. 2019.

[11] J. Li, Q. Zhou, Y. He, H. Williams, and H. Xu, "Driver-identified supervisory control system of hybrid electric vehicles based on spectrum-guided fuzzy feature extraction," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 11, pp. 2691–2701, Nov. 2020.

[12] J. Peng, H. He, and R. Xiong, "Rule based energy management strategy for a series–parallel plug-in hybrid electric bus optimized by dynamic programming," *Appl. Energy*, vol. 185, pp. 1633–1643, Jan. 2017.

[13] A.-A. Mamun, Z. Liu, D. M. Rizzo, and S. Onori, "An integrated design and control optimization framework for hybrid military vehicle using lithium-ion battery and supercapacitor as energy storage devices," *IEEE Trans. Transport. Electrific.*, vol. 5, no. 1, pp. 239–251, Mar. 2019.

[14] Q. Zhou, W. Zhang, S. Cash, O. Olatunbosun, H. Xu, and G. Lu, "Intelligent sizing of a series hybrid electric power-train system based on chaos-enhanced accelerated particle swarm optimization," *Appl. Energy*, vol. 189, pp. 588–601, Mar. 2017.

[15] C. Yang, Y. Shi, L. Li, and X. Wang, "Efficient mode transition control for parallel hybrid electric vehicle with adaptive dual-loop control framework," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1519–1532, Feb. 2020.

[16] Q. Zhou *et al.*, "Global optimization of the hydraulic-electromagnetic energy-harvesting shock absorber for road vehicles with human-knowledge-integrated particle swarm optimization scheme," *IEEE/ASME Trans. Mechatronics*, early access, Feb. 1, 2021, doi: 10.1109/TMECH.2021.3055815.

[17] Y. He *et al.*, "Multiobjective co-optimization of cooperative adaptive cruise control and energy management strategy for PHEVs," *IEEE Trans. Transport. Electrific.*, vol. 6, no. 1, pp. 346–355, Mar. 2020.

[18] H. Khayyam and A. Bab-Hadiashar, "Adaptive intelligent energy management system of plug-in hybrid electric vehicle," *Energy*, vol. 69, pp. 319–335, May 2014.

[19] H. Tian, S. E. Li, X. Wang, Y. Huang, and G. Tian, "Data-driven hierarchical control for online energy management of plug-in hybrid electric city bus," *Energy*, vol. 142, pp. 55–67, Jan. 2018.

[20] Y. Xing, C. Lv, D. Cao, and C. Lu, "Energy oriented driving behavior analysis and personalized prediction of vehicle states with joint time series modeling," *Appl. Energy*, vol. 261, Mar. 2020, Art. no. 114471.

[21] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. Los Angeles, CA, USA: Springer, 2013.

[22] C. Lv *et al.*, "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, Jul. 2018.

[23] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, "Fuzzy regression transfer learning in Takagi–Sugeno fuzzy models," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1795–1807, Dec. 2017.

[24] F. H. C. Tivive and A. Bouzerdoum, "Efficient training algorithms for a class of shunting inhibitory convolutional neural networks," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 541–556, May 2005.

[25] Q. Zhou, Y. Zhang, Z. Li, J. Li, H. Xu, and O. Olatunbosun, "Cyber-physical energy-saving control for hybrid aircraft-towing tractor based on online swarm intelligent programming," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4149–4158, Sep. 2018.

[26] J. Hou and Z. Song, "A hierarchical energy management strategy for hybrid energy storage via vehicle-to-cloud connectivity," *Appl. Energy*, vol. 257, Jan. 2020, Art. no. 113900.

[27] A. M. Ali, A. Ghanbar, and D. Soffker, "Optimal control of multi-source electric vehicles in real time using advisory dynamic programming," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10394–10405, Nov. 2019.

[28] C. Yang, S. You, W. Wang, L. Li, and C. Xiang, "A stochastic predictive energy management strategy for plug-in hybrid electric vehicles based on fast rolling optimization," *IEEE Trans. Ind. Electron.*, vol. 67, no. 11, pp. 9659–9670, Nov. 2020.

[29] Y. Huang *et al.*, "A review of power management strategies and component sizing methods for hybrid vehicles," *Renew. Sustain. Energy Rev.*, vol. 96, pp. 132–144, Nov. 2018.

[30] Y. Huang, H. Wang, A. Khajepour, H. He, and J. Ji, "Model predictive control power management strategies for HEVs: A review," *J. Power Sources*, vol. 341, pp. 91–106, Feb. 2017.

[31] C. M. Martínez and D. Cao, "Integrated energy management for electrified vehicles," in *Ihorizon-Enabled Energy Management for Electrified Vehicles*. Oxford, U.K.: Elsevier, 2019, pp. 15–75.

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

[33] R. Bellman, "Dynamic programming and a new formalism in the calculus of variations," *Proc. Nat. Acad. Sci. USA*, vol. 40, no. 4, pp. 231–235, Apr. 1954.

[34] M.-B. Radac and R.-E. Precup, "Data-driven model-free slip control of anti-lock braking systems using reinforcement Q-learning," *Neurocomputing*, vol. 275, pp. 317–329, Jan. 2018.

[35] T. Liu, Y. Zou, D. Liu, and F. 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, Dec. 2015.

[36] C. Liu and Y. L. Murphey, "Optimal power management based on Q-learning and neuro-dynamic programming for plug-in hybrid electric vehicles," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 1942–1954, Jun. 2020.

[37] B. Shuai *et al.*, "Heuristic action execution for energy efficient charge-sustaining control of connected hybrid vehicles with model-free double Q-learning," *Appl. Energy*, vol. 267, Jun. 2020, Art. no. 114900.

[38] Q. Zhou *et al.*, "Multi-step reinforcement learning for model-free predictive energy management of an electrified off-highway vehicle," *Appl. Energy*, vol. 255, pp. 588–601, Dec. 2019.

[39] G. Du, Y. Zou, X. Zhang, Z. Kong, J. Wu, and D. He, "Intelligent energy management for hybrid electric tracked vehicles using online reinforcement learning," *Appl. Energy*, vol. 251, Oct. 2019, Art. no. 113388.

[40] H. Lee, C. Song, N. Kim, and S. W. Cha, "Comparative analysis of energy management strategies for HEV: Dynamic programming and reinforcement learning," *IEEE Access*, vol. 8, pp. 67112–67123, 2020.

[41] X. Hu, T. Liu, X. Qi, and M. Barth, "Reinforcement learning for hybrid and plug-in hybrid electric vehicle energy management: Recent advances and prospects," *IEEE Ind. Electron. Mag.*, vol. 13, no. 3, pp. 16–25, Sep. 2019.

[42] X.-F. Liu *et al.*, "Neural network-based information transfer for dynamic optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1557–1570, May 2020.

[43] H. Zuo, G. Zhang, V. Behbood, and J. Lu, "Feature spaces-based transfer learning," in *Proc. 16th World Congr. Int.-Fuzzy-Systems-Assoc. (IFSA)/9th Conf. Eur. Soc. Fuzzy-Logic Technol. (EUSFLAT)*, 2015, pp. 1000–1005.

[44] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.

[45] D. Wang, Y. Li, Y. Lin, and Y. Zhuang, "Relational knowledge transfer for zero-shot learning," in *Proc. 30th AAAI Conf. Artif. Intell. (AAAI)*, 2016, pp. 2145–2151.

[46] P. M. Ashok Kumar and V. Vaidehi, "A transfer learning framework for traffic video using neuro-fuzzy approach," *Sādhanā*, vol. 42, no. 9, pp. 1431–1442, Sep. 2017.

[47] J. Cervantes, W. Yu, S. Salazar, and I. Chairez, "Takagi–Sugeno dynamic neuro-fuzzy controller of uncertain nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1601–1615, Dec. 2017.
[48] Q. Zhou, C. Wang, Z. Sun, J. Li, H. Williams, and H. Xu, "Human-knowledge-augmented Gaussian process regression for state-of-health prediction of lithium-ion batteries with charging curves," *J. Electrochem. Energy Convers. Storage*, vol. 18, no. 3, pp. 1–10, Aug. 2021.
[49] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.
[50] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *Proc. 4th Int. Conf. Learn. Represent., (ICLR) Conf. Track*, 2016, pp. 1–14.
[51] J. Li, Q. Zhou, H. Williams, and H. Xu, "Back-to-back competitive learning mechanism for fuzzy logic based supervisory control system of hybrid electric vehicles," *IEEE Trans. Ind. Electron.*, vol. 67, no. 10, pp. 8900–8909, Oct. 2020.
[52] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, vol. 1, 2014, pp. 605–619.
[53] K. McAleer, "RDE—Development process & tools," in *Proc. 9th AVL Calibration Symp.*, 2015, p. 51.

Quan Zhou (Member, IEEE) received the B.Eng. and M.Eng. degrees in vehicle engineering from the Wuhan University of Technology, Wuhan, China, in 2012 and 2015, respectively, and the Ph.D. degree in mechanical engineering from the University of Birmingham (UoB), Birmingham, U.K., in 2019, that was distinguished by being the school's sole recipient of the UoB Ratcliffe Prize.

He is currently a Research Fellow and leads the Research Group of Connected and Autonomous Systems for Electrified Vehicles (CASE-V) Team, UoB. His research interests include evolutionary computation, fuzzy logic, reinforcement learning, and their application in vehicular systems.

Dezong Zhao (Senior Member, IEEE) received the B.Eng. and M.S. degrees in control engineering from Shandong University, Jinan, China, in 2003 and 2006, respectively, and the Ph.D. degree in control engineering from Tsinghua University, Beijing, China, in 2010.

Since 2020, he has been a Senior Lecturer with the James Watt School of Engineering, University of Glasgow, Glasgow, U.K. His research interests focus on autonomous vehicles and control engineering.

Dr. Zhao is an EPSRC Fellow and a Royal Society-Newton Advanced Fellow.

Bin Shuai received the B.Eng. degree in mechanical engineering from the Harbin University of Science and Technology, Harbin, China, in 2015. He is currently pursuing the Ph.D. degree in mechanical engineering with the Connected and Autonomous Systems for Electrified Vehicles (CASE-V) Team, University of Birmingham, Birmingham, U.K.

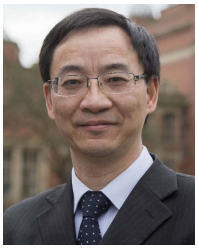
His research interests include intelligent vehicle control, reinforcement learning, system modeling, and energy management.

Yanfei Li received the B.Eng. degree in power machinery from Jilin University, Changchun, China, in 2006, and the Ph.D. degree in mechanical engineering from the University of Birmingham, Birmingham, U.K., in 2012.

He is currently a Research-Focused Assistant Professor with Tsinghua University, Beijing, China. His research interests include emissions and two-phase flows in the internal combustion engines (ICE), and dedicated ICE design for hybrid vehicles.

Huw Williams received the B.A. and M.A. degrees in mathematics from the University of Oxford, Oxford, U.K., in 1978 and 1983, respectively, and the Ph.D. degree in theoretical mechanics from the University of East Anglia, Norwich, U.K.

He joined Jaguar Land Rover (JLR), Coventry, U.K., in 1986, and then became one of only two people to hold the position of Senior Engineering Specialist in JLR product development. He is an Honorary Professor with the University of Birmingham (UoB), Birmingham, U.K., and the Director of DEEPPOWER Innovation Ltd., London, U.K. He has more than 20 years' of experience in the automotive industry.



Hongming Xu received the Ph.D. degree from the Imperial College London, London, U.K., in 1995.

He is a Professor of energy and automotive engineering with the University of Birmingham, Birmingham, U.K., and the Head of the Vehicle and Engine Technology Research Centre, Birmingham. He has six years of industrial experience with Jaguar Land Rover, Coventry, U.K. He has authored and coauthored more than 400 journal and conference publications on advanced vehicle powertrain systems.

Dr. Xu is a Fellow of SAE and IMechE.