Transferable representation modelling for real-time energy management of the plug-in hybrid vehicle based on k-fold fuzzy learning and Gaussian process regression

Zhou, Quan; Li, Yanfei; Zhao, Dezong; Li, Ji; Williams, Huw; Xu, Hongming; Yan, Fuwu

DOI:
10.1016/j.apenergy.2021.117853

Document Version
Peer reviewed version

Citation for published version (Harvard):
Zhou, Q, Li, Y, Zhao, D, Li, J, Williams, H, Xu, H & Yan, F 2022, 'Transferable representation modelling for real-time energy management of the plug-in hybrid vehicle based on k-fold fuzzy learning and Gaussian process regression', *Applied Energy*, vol. 305, 117853. https://doi.org/10.1016/j.apenergy.2021.117853

Transferable Representation Modelling for Real-time Energy Management of the Plug-in Hybrid Vehicle based on K-fold Fuzzy Learning and Gaussian Process Regression

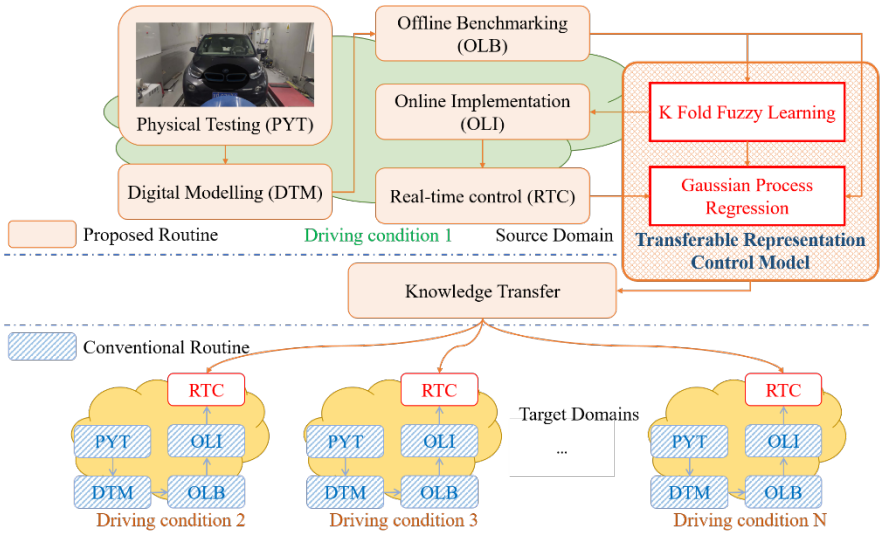
Quan Zhou [a,b], Yanfei Li [b], Dezong Zhao [c], Ji Li [a], Huw Williams [a], Hongming Xu [a,b,d*1], Fuwu Yan [d]

a. Department of Mechanical Engineering, University of Birmingham, B15 2TT, UK
b. State Key Laboratory of Automotive Energy and Safety, Tsinghua University, 100083, China
c. James Watt School of Engineering, University of Glasgow, G12 8QQ, UK
d. School of Automotive Engineering, Wuhan University of Technology, 430070, China

HIGHLIGHTS

- A transfer learning routine is proposed to reduce workload in control development.
- Fuzzy learning and Gaussian process regression are incorporated in a control model.
- Fuzzy learning achieves 27% higher distinction rate than a neural network.
- The 'deeper' architecture outpefroms the 'broader' one in transfer learning.
- Real-time control functionality is verified by hardware-in-the-loop testing.

GRAPICHAL ABSTRACT



ARTICLE INFO

Keywords:

Energy management
Hybrid vehicle
Adaptive neuro fuzzy inference
Gaussian process regression
Transfer learning

ABSTRACT

Electric vehicles, including plug-in hybrids, are important for achieving net-zero emission and will dominate road transportation in the future. Energy management, which optimizes the onboard energy usage, is a critical functionality of electric vehicles. It is usually developed following the model-based routine, which is conventionally costly and time-consuming and is hard to meet the increasing market competition in the digital era. To reduce the development workload for the energy management controller, this paper studies an innovative transfer learning routine. A new transferable representation control model is proposed by incorporating two promising artificial intelligence technologies, adaptive neural fuzzy inference system and Gaussian process regression, where the former applies k-fold cross valudation to build a neural fuzzy system for real-time implementation of offline optimization result, and the later connects the neural fuzzy system with a 'deeper' architecture to transfer the offline optimization knowledge learnt at source domain to new target domains. By introducing a concept of control utility that evaluates vehicle energy efficiency with a penalty on usage of battery energy, experimental evaluations based on the hardware-in-the-loop testing platform are conducted. Competitive real-time control ultility values (as much as 90% of offline benchmarking results) can be achieved by the proposed control method. They are over 27% higher than that achieved by the neural-network-based model.

[1]* Corresponding Author: Prof H. M. Xu (h.m.xu@bham.ac.uk), University of Birmingham, B15 2TT, UK
(The short version of the paper was presented at virtual CUE2020, Oct 10-17, 2020. This paper is a substantial extension of the short version of the conference paper.)

1. Introduction

The hybrid electric vehicle (HEV), as a mainstream ultra-low emission solution, will account for more than 60% of the world passenger market share by 2030 according to predictions of the International Energy Agency (IEA) [1]. The energy management system (EMS) is one of the most important systems within the HEV, and it controls the energy flows between power units (e.g., engine and battery) within the hybrid system [2]. The optimization of energy efficiency in the EMS is among the most challenging decision-making tasks because of uncertainties and constraints in real-world driving [3,4].

The control strategy of the EMS should be optimized to allow vehicles to comply with regulations in fuel consumption and emissions. New European Driving Cycle (NEDC) for road vehicles has been replaced by the Worldwide-harmonized Light-duty Testing Cycle (WLTC), in which an increasing number of transient operation points are included to evaluate energy efficiency and emissions [5]. New vehicle legislation encompassing real-world driving emissions (RDE) have been enacted [6,7] and they will bring more workload for the R&D of advanced EMSs. Therefore, advanced EMSs that have the self-learning capability are in urgent demand to help automakers comply with legislation and improve customer satisfaction.

Thanks to growing developments in artificial intelligence (AI) and the Internet-of-Things (IoT), learning-based energy management techniques have been shown to be significantly superior to the conventional rule-based and model-based methods [8]. Reinforcement learning (RL), which has the capability of online learning in real-world driving [9], is recognized as a promising technology for EMS. It is plant-model-free [10] and updates its knowledge base (Q table or deep Q network, DQN) using reinforcement information obtained in real-world interactions [11]. The effectiveness of RL has been demonstrated in various vehicle control applications [12]. Remarkable improvements in vehicle energy efficiency have been achieved by RL methods, e.g. Q-learning [13], deep Q-learning [14], double Q-learning [15], double deep Q-learning [16], and multiple-step Q-learning [9]. Most research on RL-based power management control focuses on learning from scratch [17,18], which requires a long time to develop a proper control policy and therefore is not practically acceptable [19].

Transfer learning (TL), which builds a robust representation model (RM) in a source domain and trains a smaller scale compensation model (CM) for target domains, is an emerging and promising technology that can significantly reduce the learning cost and time. Because TL is new to automotive engineering, only a few studies in TL-based energy management have been reported. Lian et al. transferred the control policy (of a DQN-based EMS) among four different HEV topologies by introducing an internal RM within the DQN, which can accelerate the RL process and save at least 40% time to obtain a proper control model [20]. Guo et al. proposed a speed classifier to enable TL of a bi-level energy management system, where the first level selects one of the RMs with the speed classifier, and the second level optimizes the associated CM with RL [21]. By evaluating the vehicle's fuel economy on a hardware-in-the-loop (HiL) testing platform under an author-defined driving cycle, the RL+TL method can save more than 9% fuel compared to the RL-only method [21].

RMs are important for knowledge transfer, and an ideal RM of HEV energy management strategy should achieve competitive real-time performance under a given driving cycle (usually defined by vehicle regulations) compared to the benchmark result obtained in offline dynamic programming (DP) [22]. However, the conventional DP-based control development is time consuming and expensive because new experiments are usually needed for benchmarking under different regulations [11,23]. This motivates the present work to develop a new transferable RM that can be adaptive to regulations in different countries worldwide without tedious DP-based benchmarking. To achieve this, advanced AI techniques are needed to address the challenges in two aspects: 1) an explainable and robust baseline RM is required to precisely implement DP result for real-time control; and 2) an effective knowledge transfer method is in demand to enable the RM to be adaptive to different regulations worldwide.

To address the first challenge, AI models are usually developed by artificial neural networks (ANN) or fuzzy inference systems (FIS), where the former is a backbox model with strong data-driven learning capability [24] while the later is explainable by linguistic logics [25]. Meta-heuristic algorithms, e.g. particle swarm optimization (PSO) [26,27] and genetic algorithms (GA) [28,29], are commonly used for AI modelling and can be used to optimize the real-time control model to achieve the minimum root mean squared error (RMSE) based on the DP result. Adaptive neural fuzzy inference system (ANFIS) is a new explainable AI technique [30]. With the advantages of NN and FIS, ANFIS is promising for real-time control because it implements data-driven learning in building interpretable rules/logics and is capable to assist decision making in vehicle controller. Therefore, this paper builds the baseline RM based on an ANFIS to implement the DP result for real-time control.

On the other hand, the performance of AI models heavily depends on quantity and quality of data. Khayyam et al. modelled a fuzzy logic power management controller using 5 groups of 30k data sets [29]. Xing et al. used 10k data pairs to train recurrent neural networks for driver behavior prediction [31]. To make breakthrough upon the above studies that are based on big volume of data, cross-validation provides an efficient tool for modelling with limited data [32]. K-fold cross-validation is widely used for learning with labelled data [32]. Lv et al. applied a five-fold method to train a neural network for driver intention prediction [33]. However, using K-fold cross-validation to build real-time representation control models has not yet been reported.

Transferable knowledge and topologies of the learning system are the keys to address the second challenge. The transferable knowledge includes characteristics [34], extracted features [35], model parameters [36], and relational information [37]. The transferable knowledge can be modelled by both parametric and non-parametric methods. ANN [38] and fuzzy systems [39] are normally developed for the parametric modeling. With significant superior representation capability than the parametric methods, the emerging Gaussian process regression (GPR) has been recognized as a promising non-parametric method [40]. It treats the input-to-output mapping as a random function with a probability density defined based on a Gaussian process hypothesis. Several published works have shown the advantage of using GPR-based transfer learning for battery state estimation [41,42], driver behavior prediction [43,44], and robot

control [45]. However, the research that applies GPR for transfer learning in controller development is lacking.

Regarding the topologies of transfer learning systems, 'broader' and 'deeper' architectures are usually adopted to transfer the knowledge from source domain to new target domains. Deng et al. proposed a 'broader' architecture for robot control, where the control outputs are selected from an empirical model (developed in source domain) or an adaptive model (learnt in target domain) [45]. Zou et al. enables fuzzy regression transfer learning in a 'deeper' architecture, where the knowledge transfer model is connected in the front of a baseline fuzzy model [46]. According to the 'no free lunch' theory [47], there is no confirmed optimal system architecture for the newly proposed transferable RM of energy management control.

To address the two challenges while contributing innovative ideas to more energy-effieicent electric vehicles, this paper studies a new transferable representation modelling scheme for the development of energy management controller. A new RM is proposed by incorporating ANFIS and GPR, where ANFIS is used to implement the DP result in source domain and GPR transfers the knowledge to allow optimal control in target domains. The work has two original contributions to design of dedicated learning systems: 1) a global k-fold fuzzy learning (GKFL) scheme is proposed, which applies k-fold cross validation to obtain a robust ANFIS model for knowledge implementation, and the optimal setting of k-fold learning is investigated; 2) GPR is deployed to enable knowledge transfer across different driving cycles, and a hybrid network with a 'deeper' architecture is selected as a unified topology for the transferable RM.

The rest of the paper is organized as follows: Section 2 formulates the studied problem by introducing the vehicle system and driving cycles. Section 3 proposes the transferable representation control modelling scheme for the PHEV energy management. Section 4 presents the results from experimental evaluations, and Section 5 summarizes the conclusions.

2. Problem Formulation

This section formulates the transfer learning problem based on mathematic modelling of the hybrid electric powertrain, which is managed by a fuzzy inference system. Standard driving cycles, proposed by the legislations in different countries, are also introduced to define the source domain and target domains. The source domain is defined as the driving cycle used for offline benchmarking, and the target domains are defined as the new driving conditions that have not employed the control strategy for offline optimization.

2.1. The hybrid electric powertrain

The studied vehicle is a plug-in HEV with a series topology, and has two power units to meet the power demand of a 125kW motor for vehicle operation. The power units include a 36.6kW generator powered by a 0.65L engine and a 360V/22kWh high-voltage battery package. The key specifications of the studied vehicle are summarized in Table I.

Table I Key specifications of the studied vehicle

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	-

The power flows across the 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 $P_{\mathrm{ppu}} > 0$, and is charging when $P_{\mathrm{ppu}} < 0$; 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 used as the alternative power unit for maintaining the battery's state-of-charge (SoC) to allow longer driving distance.

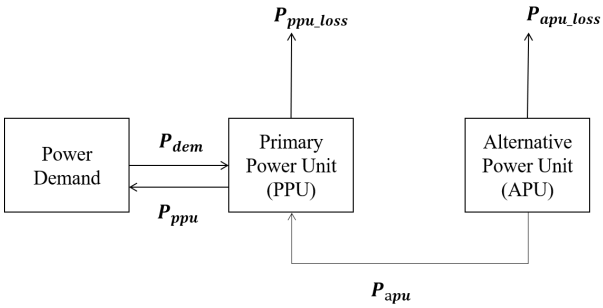


Fig. 1 Power Flow of a Hybrid Powertrain System

From the perspective of energy transmission, the power flow in the PHEV is expressed as

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

The power losses of the battery and the engine-generator can be modelled by

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

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 the fuel.

The primary objective of power management is to maximize the vehicle's energy efficiency:

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

where t_0 and t_t are the start and terminate of a driving cycle; Δt is the sampling time; and $P_{\mathrm{loss}}(t) = P_{\mathrm{ppu\_loss}}(t) + P_{\mathrm{apu\_loss}}(t)$ is the total power loss.

Maintaining the battery SoC is a critical constraint in power management. The battery SoC at time t_l is calculated by

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

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

To achieve the maximum vehicle energy efficiency while maintaining the battery SoC, a control utility (CU) function is

defined by introducing a penalty for degrading the battery SoC, $\beta \cdot e^{\alpha \cdot (SoC(t)-SoC^{+}-SoC^{-})}$[48], to the denominator of Eq. (3) as

$$\mathcal{U} = \frac{\sum_{t=t_1}^{t_\tau} P_{\text{dem}}(t)\cdot \Delta t}{\sum_{t=t_1}^{t_\tau} P_{\text{dem}}(t)\cdot \Delta t + \sum_{t=t_1}^{t_\tau} (P_{\text{loss}}(t)\cdot \Delta t + \beta \cdot e^{\alpha \cdot (SoC(t)-SoC^{+}-SoC^{-})})}, \quad (5)$$

where SoC^{+} and SoC^{-} are the higher and lower boundary of battery SoC for the hybrid power mode, respectively.

2.2. Energy Management with Fuzzy Inference

The energy management strategy determines the power ratio of the engine-generator $u_{\text{egu}}(t)$ by [8,15,49]:

$$u_{\text{egu}}(t) = \mathcal{M}(P_{\text{dem}}(t), SoC(t), \mathbb{C}) \quad (6)$$

where $\mathcal{M}(\cdot)$ can be a FIS that projects the inputs of $P_{\text{dem}}(t)$ and $SoC(t)$ to the relevant control command $u_{\text{egu}}(t)$; and $\mathbb{C}$ is a vector of parameters.

The FIS, $\mathcal{M}(\cdot)$, is developed based on a Takagi-Sugeno (TS) model because the TS model is easy to implement in data-driven learning [50]. It includes one input layer, three hidden layers and one output layer, as shown in Fig. 2.

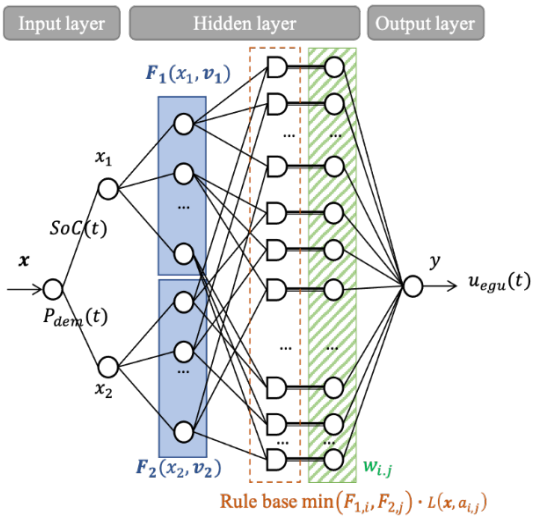


Fig. 2 Takagi-Sugeno fuzzy model for energy management

The input layer collects the battery SoC and power demand from the PHEV with an input vector $\boldsymbol{x} = [SoC(t), P_{\text{dem}}(t)]^{\text{T}}$. The output layer implements the control command $u_{\text{egu}}(t) = y$ based on the computing results from hidden layers. The hidden layers calculate y using $\boldsymbol{x}$.

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

$$\left.\begin{aligned} F_{1,i}(\boldsymbol{x_1}, \boldsymbol{v_{1,i}}) &= \mathbf{max}\left(\mathbf{min}\left(\frac{x_1 - v_{1,i}(1)}{v_{1,i}(2) - v_{1,i}(1)} \quad \frac{v_{1,i}(3) - x_1}{v_{1,i}(3) - v_{1,i}(2)}\right), \mathbf{0}\right) \\ F_{2,j}(\boldsymbol{x_2}, \boldsymbol{v_{2,j}}) &= \mathbf{max}\left(\mathbf{min}\left(\frac{x_2 - v_{2,j}(1)}{v_{2,j}(2) - v_{2,j}(1)} \quad \frac{v_{2,j}(3) - x_2}{v_{2,j}(3) - v_{2,j}(2)}\right), \mathbf{0}\right) \end{aligned}\right\} \quad (7)$$

where x_1 and x_2 are the elements in the input vector $\boldsymbol{x}$; $F_{1,i}$ is the i-th membership function for the first input; n is the total number of membership functions for the first input; $F_{2,j}$ is the j-th membership function for the second input; m is the total number of membership functions for the second input; and $\boldsymbol{v}(k)$, k=1,2,3, is the k-th element of the parameter vector $\boldsymbol{v}$.

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

$$\begin{aligned} &\text{If } \boldsymbol{x}(1) \text{ is } F_{1,i}(\boldsymbol{x}(1), \boldsymbol{v_{1,i}}) \text{ and } \boldsymbol{x}(2) \text{ is } F_{2,j}(\boldsymbol{x}(2), \boldsymbol{v_{2,j}}), \\ &\qquad \text{then } \boldsymbol{y} \text{ is } \boldsymbol{L}(\boldsymbol{x}, a_{i,j}), \\ &\qquad i = 1,2,\dots,n;\ j = 1,2,\dots,m \end{aligned} \quad (8)$$

where $L(\boldsymbol{x}, a_{i,j})$ is the output membership function in a constant type for this study as in [23]; and $a_{i,j}$ is a scale in an output membership function.

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

$$\boldsymbol{y} = \sum_{j=1}^{m}\sum_{i=1}^{n}\left\{\min\left(F_{1,i}(\cdot), F_{2,j}(\cdot)\right) \cdot \boldsymbol{L}(\boldsymbol{x}, a_{i,j}) \cdot w_{i,j}\right\} \quad (9)$$

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

Optimization of the energy management controller is to find the optimal setting of the parameter vector

$$\mathbb{C} = [\boldsymbol{V_1}, \boldsymbol{V_2}, \boldsymbol{A}, \boldsymbol{W}] \quad (10)$$

where $\boldsymbol{V}_1 = [v_{1,1}, v_{1,2}, \dots v_{1,n}]$ and $\boldsymbol{V}_2 = [v_{2,1}, v_{2,2}, \dots v_{2,m}]$ are the parameter vectors of the inputs membership functions; $\boldsymbol{A}_1 = [a_{1,1}, \dots, a_{1,n}, a_{2,1}, \dots, a_{2,n}, \dots a_{m,n1}, \dots, a_{m,n}]$ is the parameter vector of the output membership functions.

2.3. Source Domains and Target Domains

In this paper, five selected driving cycles built on the standard cycles will be used to testify the control models based on five-fold cross validation. In each fold, one of these cycles will be defined as the source domain for model learning and the others will be defined as target domains for validations. The profiles of the standard cycles are summarized in Table II.

Table II Profiles of the studied driving cycles

Cycle name	Abbreviation	Country/ Region
Federal Test Procedure (hot soak period included)	FTP75	US
Japanese Cycle	JC08	Japan
Aggressive Standardized Random Test	RTS95	Proposed for RDE
Urban Dynamometer Driving Schedule	UDDS	US
Worldwide-harmonized Light-vehicles Test Cycles	WLTC	UN, EU

* US: United States; RDE: Real-world Driving Emission (Evaluation); UN: United Nation; EU: European Union

To ensure each domain has equal sample size, the studied five driving cycles are generated by running the standard cycles repeatedly for 2474s (with a sampling time of 1s) which is equal to the cycle length of the longest cycle (FTP75). To make it easy to follow, 'extended' is added to the original names of the standard cycles to name the generated cycles (e.g., extended WLTC, Ex-WLTC for short) for the rest of this paper. The source domain is defined as one of the driving cycles that is selected for

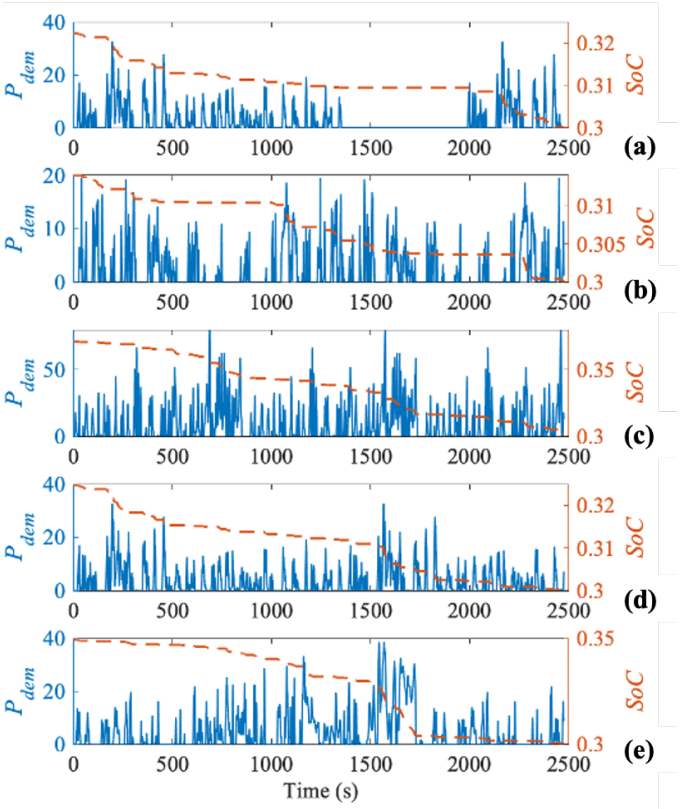


Fig. 3 Power demands and benchmark SoC trajectories of (a) Ex-FTP75, (b) Ex-JC08, (c) Ex-RTS95, (d) Ex-UDDS, and (e) Ex-WLTC

optimization of the energy management strategy with dynamic programming (DP). The target domains are the driving cycles where the HEV is controlled by the strategy optimized in the source domain while applies transfer learning for control strategy adaptation.

The power demands (shown in blue solid lines) of the PHEV under the five given driving cycles and the respective battery SoC trajectories (shown in red dashed lines) obtained by DP are compared in Fig. 3. Battery SoC was used as the state variable in the DP algorithm, and its value at the end of each cycle was limited to 0.30. For demonstration, this paper introduced a finite element state space, $\{0.3, 0.301, 0.302, \ldots 0.399, 0.40\}$, for battery SoC. The size of the state space is adjustable for different use cases, and it affects the optimality and computational efforts. Following the power demands at each time step, DP determines the optimal control signal to achieve the maximum CU value for each driving cycle.

3. Methodology

A transfer learning routine (TLR) as shown in Fig. 4 is proposed to enable rapid development of the representation control model for real-time energy management of PHEV. Following the conventional model-based controller development routine, the TLR includes five main steps.

- Step 1. Vehicle data, including energy flow and components' states (e.g., battery SoC), is collected from the chassis dynameter testing system at Tsinghua University under the selected driving cycle (source domain).

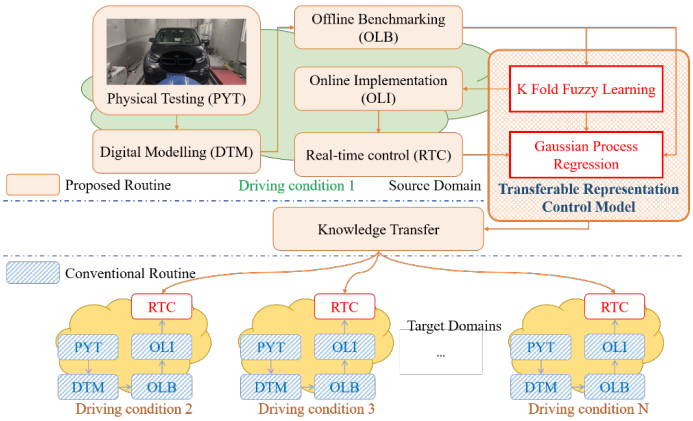


Fig. 4 The transfer learning routine for development of energy management controller

- Step 2. Digital vehicle model is developed using the vehicle data.
- Step 3. Dynamic programming (DP) is then carried out for offline benchmarking (OLB) of the ideal optimal signals under the source domain.
- Step 4. With the control parameters calibrated based on the OLB results, a fuzzy logic controller is developed for online implementation (OLI) of the energy management strategy.
- Step 5. Real-time control (RTC) functionality is verified in hardware-in-the-loop (HiL) testing.

The transferable representation control model (TRCM) is the core of the TLR to reduce the need for intensive physical testing and computationally-costly DP-based benchmarking. It is built on ANFIS and GPR, which are both promising AI technologies. ANFIS is used to implement the DP result under the source domain while GPR transfers the knowledge to enable optimal control in target domains. A new global k-fold fuzzy learning (GKFL) scheme is proposed to enable robust source domain learning, and a 'deeper' architecture is developed to incorporate GPR with ANFIS for the TRCM. Details of GKFL-based source domain learning and TRCM-based target domain transfer learning are introduced as follows.

3.1. Global k-fold fuzzy learning

A global k-fold fuzzy learning scheme, which implements k-fold cross-validation to optimize the vector $\mathbb{C}$ of the fuzzy inference system, is proposed for source domain learning. It allows accurate knowledge implementation for real-time control by obtaining a FIS model $\boldsymbol{\mathcal{M}}^{\mathbf{kf}}$ that allows the vehicle system to achieve high CU value, $\mathcal{U}(\boldsymbol{\mathcal{M}}^{\mathrm{kf}})$, in real-time. This model will be better than the one using the conventional model $\boldsymbol{\mathcal{M}}^{\mathrm{cov}}$ (with default setting in the MATLAB ANFIS Toolbox). Since the lacking of research into finding the best κ value for fuzzy learning in energy management control, this paper will investigate fuzzy learning performance with all possible κ values (i.e., $\kappa = 2,3,4, \ldots, 10$).

A global search method is proposed to determine both the optimal setting κ^* for K-fold fuzzy learning and the optimal online energy management model $\boldsymbol{\mathcal{M}}^{\mathbf{kf}}$. The overall working procedure of the proposed global K-fold fuzzy learning is presented in **Fig. 5**. After the initialization of the K value by setting $\kappa = 2$, a rotational learning process will repeat Steps 1-4:

Step 1. The offline optimization results $\boldsymbol{D}[\boldsymbol{x}, \boldsymbol{y}]^{\mathrm{T}}$ are randomly divided into κ folds which have the similar size, i.e., $\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots, \boldsymbol{D}_\kappa$.

Step 2. The parameter vector $\mathbb{C}$ in fuzzy model $\boldsymbol{\mathcal{M}}(.,.,\mathbb{C})$ is optimized in κ independent rounds, where $\boldsymbol{D}_{\mathrm{trn}}^r = [\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots, \boldsymbol{D}_{r-1}, \boldsymbol{D}_{r+1}, \ldots \boldsymbol{D}_\kappa]$ (r=1,2, …, κ) is for training and $\boldsymbol{D}_{\mathrm{tst}}^r = \boldsymbol{D}_r$ is for testing.

Step 3. $\boldsymbol{\mathcal{M}}^\kappa(.,.,\mathbb{C}^\kappa)$ is selected based on the results from Step 2, which has the minimum cross-validation mean square error (CVMSE),

$$\mathrm{CVMSE}_{(\kappa)} = \frac{1}{\kappa} \cdot \sum_{r=1}^{\kappa} \frac{\sum_{t=1}^{\tau'} (\boldsymbol{\mathcal{M}}^r(x_{\mathrm{tst}}^r(t)) - y_{\mathrm{tst}}^r(t))^2}{\tau'} \tag{11}$$

where $\boldsymbol{\mathcal{M}}^r(x_{\mathrm{tst}}^r(t))$ is the model output using the model learnt from training data $\boldsymbol{D}_{\mathrm{trn}}^r$ during round r; $x_{\mathrm{tst}}^r(t)$ is the model input in testing data $\boldsymbol{D}_{\mathrm{tst}}^r$ during round r; and $y_{\mathrm{tst}}^r(t)$ is the model output in testing data $\boldsymbol{D}_{\mathrm{tst}}^r$ in the round r.

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

Once the termination term is met ($\kappa > 10$), the process stops. Then the optimal setting κ^* is extracted, together with the optimal model $\boldsymbol{\mathcal{M}}^{\mathrm{kf}}$ that satisfies

$$\mathcal{U}(\boldsymbol{\mathcal{M}}^{\mathbf{kf}}) = \mathcal{U}(\boldsymbol{\mathcal{M}}^{\boldsymbol{\kappa}^*}) \geq \mathcal{U}(\boldsymbol{\mathcal{M}}^{\boldsymbol{\kappa}}), \qquad \kappa \in [2,10] \tag{12}$$

where $\mathcal{U}(\boldsymbol{\mathcal{M}}^{\kappa^*})$ is the CU value that the vehicle achieved using the model $\boldsymbol{\mathcal{M}}^{\kappa^*}$.

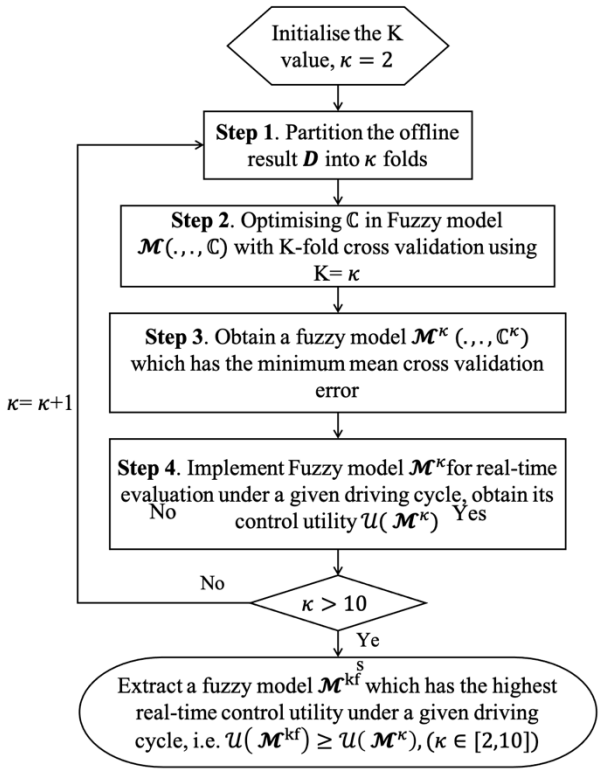


Fig. 5 Procedure of GKFL for implementing the offline optimisation result into real-time control.

3.2. Knowledge transfer with Gaussian process regression

The ANFIS model has been optimized under the source domain. To make it more adaptive to target domains, a Gaussian process model, $u' = \mathcal{G}(\boldsymbol{x})$, is developed to regulate the control output from the ANFIS model using feature state vector, $\boldsymbol{x}$. Based on the Gaussian process hypothesis, $\mathcal{G}(\boldsymbol{x})$ is treated as a random function with a Gaussian prior, $\mathcal{gp}(\cdot)$, as

$$p(\mathcal{G}(\boldsymbol{x})) = \mathcal{gp}(\mu(\boldsymbol{x}), K(\boldsymbol{x}, \boldsymbol{x})) \tag{13}$$

where $\mu(\boldsymbol{x})$ is the mean function of $\boldsymbol{x}$, and $K(\boldsymbol{x}, \boldsymbol{x})$ is the covariance function of $\boldsymbol{x}$. The GPR is trained by postulating a parametric form for the mean and the covariance function. The parameters of the mean and covariance functions are obtained as a hyper-parameter vector, $\boldsymbol{\theta}$, which is estimated by maximizing the log-likelihood of a training dataset $\{\boldsymbol{x}_{\mathrm{trn}} \quad \boldsymbol{u}'_{\mathrm{tr}}\}$.

During the learning process, $\boldsymbol{u}_p{}'$ at a set of inputs $\boldsymbol{x}_p$ is predicted by computing a conditional probability distribution $p(\boldsymbol{u}'_{\mathrm{tr}} | \boldsymbol{u}', \boldsymbol{x}, \boldsymbol{x}_{\mathrm{trn}}, \theta)$; p has a Gaussian distribution, $\mathcal{N}(\tilde{\mu}, \tilde{K})$, with the mean function, $\tilde{\mu}$, and covariance, $\tilde{K}$, given by

$$\left.\begin{aligned} \tilde{\mu}(\boldsymbol{x}_{\mathrm{trn}}) &= \mu(\boldsymbol{x}_{\mathrm{trn}}) + K(\boldsymbol{x}_{\mathrm{trn}}, \boldsymbol{F}^*) K^{-1}(\boldsymbol{x}_{\mathrm{trn}}, \boldsymbol{F}^*)[\mu(\boldsymbol{x}_p) - \boldsymbol{u}_p{}'] \\ \tilde{K}(\boldsymbol{x}_{\mathrm{trn}}, \boldsymbol{x}_{\mathrm{trn}}) &= K(\boldsymbol{x}_p, \boldsymbol{x}_p) - K(\boldsymbol{x}_{\mathrm{trn}}, \boldsymbol{x}_p) K^{-1}(\boldsymbol{x}_p, \boldsymbol{x}_p) K(\boldsymbol{x}_p, \boldsymbol{x}_{\mathrm{trn}}) \end{aligned}\right\} \tag{14}$$

As a data-driven learning, GPR is expected that the data points with similar state values naturally have close output values. To reduce the negative impact of the similarity on model learning performance, kernel functions are widely adapted. This paper uses a standard squared exponential kernel function that is built in MATLAB GPR toolbox [40]:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j | \Theta) = {\sigma_f}^2 \exp[-\frac{1}{2} \cdot \frac{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T (\boldsymbol{x}_i - \boldsymbol{x}_j)}{{\sigma_l}^2}] \tag{15}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ ($i \neq j$) are two randomly selected vectors from the training set $\boldsymbol{x}_{\mathrm{trn}}$; $\Theta = [\log \sigma_l, \log \sigma_f]$ is a parameter vector of the kernel function; σ_l is the characteristic length scale; and σ_f is the signal standard derivation.

A 'deeper' architecture, as shown in **Fig. 6**a) is developed in this paper to incorporates the GPR model with the ANFIS model for energy management. On the other hand, a 'broader' architecture, as shown in **Fig. 6**b), is also studied for comparison.

For the 'deeper' architecture, the inputs of the GPR model are

$$\boldsymbol{x}(\boldsymbol{t}) = [u_{\mathrm{anfis}}(t) \quad P_{\mathrm{dem}}(t) \quad \mu(\boldsymbol{P}_{\mathrm{dem}}) \quad \sigma(\boldsymbol{P}_{\mathrm{dem}})]^{\mathbf{T}} \tag{16}$$

where $u_{\mathrm{anfis}}(t)$ and $P_{\mathrm{dem}}(t)$ are the output of the ANFIS model and vehicle power demand at time t, respectively; $\mu(\cdot)$ and $\sigma(\cdot)$ are mean function and variation function, respectively; and $\boldsymbol{P}_{\mathrm{dem}} = [P_{\mathrm{dem}}(t_1), P_{\mathrm{dem}}(t_1), \ldots, P_{\mathrm{dem}}(t_t)]$ is a vector of power demands over a driving cycle. The output signal calculated with the GPR model, $u_{\mathrm{egu}}(t) = u'(t) = \mathcal{G}_D(\boldsymbol{x}(\boldsymbol{t}))$, is the final output of the TRCM with 'deeper' architecture.

For the 'broader' architecture, the inputs of the GPR model, $\mathcal{G}_B$, is

$$\boldsymbol{x}(\boldsymbol{t}) = [P_{\mathrm{dem}}(t) \quad \mu(\boldsymbol{P}_{\mathrm{dem}}) \quad \sigma(\boldsymbol{P}_{\mathrm{dem}})]^{\mathbf{T}} \tag{17}$$

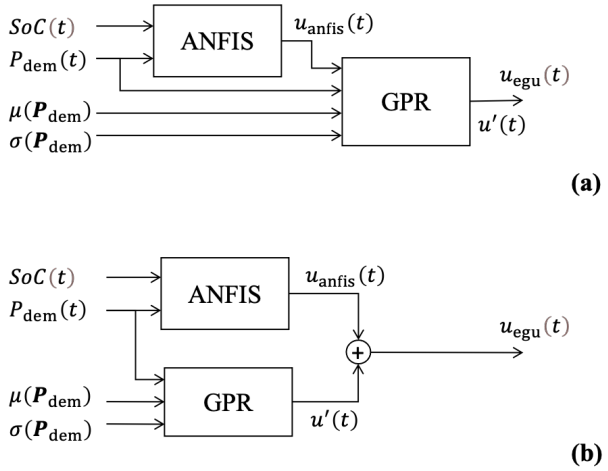


Fig. 6 Configuration of the transferable representation control model with (a) 'deeper' architecture and (b) 'broader' architecture.

The outputs of the TRCM with 'broader' architecture is represented as

$$u_{\text{egu}}(t) = u_{\text{anfis}}(t) + u^{'}(t) \quad (18)$$

where $u_{\text{anfis}}(t)$ is the output of the ANFIS model, and $u^{'}(t) = \mathcal{G}_B(\boldsymbol{x}(t))$ is the output of the GPR model.

4. Experimental evaluations

Both offline software-in-the-loop (SiL) and online HiL testing platforms were used in experimental evaluations. The SiL test was conducted in MATLAB 2020a on a PC with an i7 CPU and a 16GB RAM. The power control prototypes were developed in Simulink to allow closed-loop control of the PHEV model. A Speedgoat real-time target machine is used for HiL testing, as shown in **Fig. 7**. The control prototype and the real-time vehicle model are compiled in a host PC, downloaded onto the Speedgoat target machine through Ethernet, and physically connected via a CAN bus.

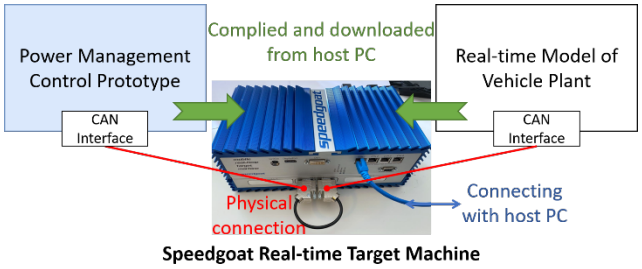


Fig. 7 Online hardware-in-the-loop testing platform

4.1. Fuzzy learning performance in source domain

Experimental evaluation on learning performance in the source domain was conducted under the Ex-WLTC condition, which is based on the WLTC cycle being used for vehicle certification in the European Union countries. The benchmark power management strategy under Ex-WLTC was obtained by dynamic programming. 70% of data was used for learning, and 30% was for verification. GA and PSO algorithms were employed for the GKFL. The conventional method (ANFIS toolbox) was selected as the baseline. The results obtained by GA and PSO are compared with the baseline in **Fig. 8**a) and Fig. 8b), respectively.

The models were developed using the training data, and their learning performances were evaluated based on the mean square errors with training data (Train. MSE), where the MSE with the whole training data set is measured for the conventional method, and the minimum cross-validation mean square is used for evaluation of the GKFL method with different κ values. The models were examined using the verification data to obtain the verification mean square error (Veri. MSE). The CU was evaluated by deploying the models in real-time control and is shown in yellow lines with markers. The baseline CU value is shown in red dash line as a reference.

The highest CU value under Ex-WLTC cycle is achieved by incorporating GKFL with GA when $\kappa = 9$. However, GKFL is more robust with PSO which achieves higher average CU value (0.2577) than GA (0.2481). The κ value is critical for GKFL and needed to be chosen very carefully. GKFL achieves higher CU value than the baseline when $\kappa = 5,6,9$ where $\kappa = 5$ is widely used in five-fold cross validation. Another widely used cross-validation method, i.e., ten-fold cross validation, is beneath expected in GKFL because it achieves lower CU value than the baseline with both GA and PSO. The ANFIS model, GKFL-9W, which is obtained with nine-fold fuzzy learning using PSO algorithm under the Ex-WLTC condition, is used as the base model in the TRCM for the rest of the paper.

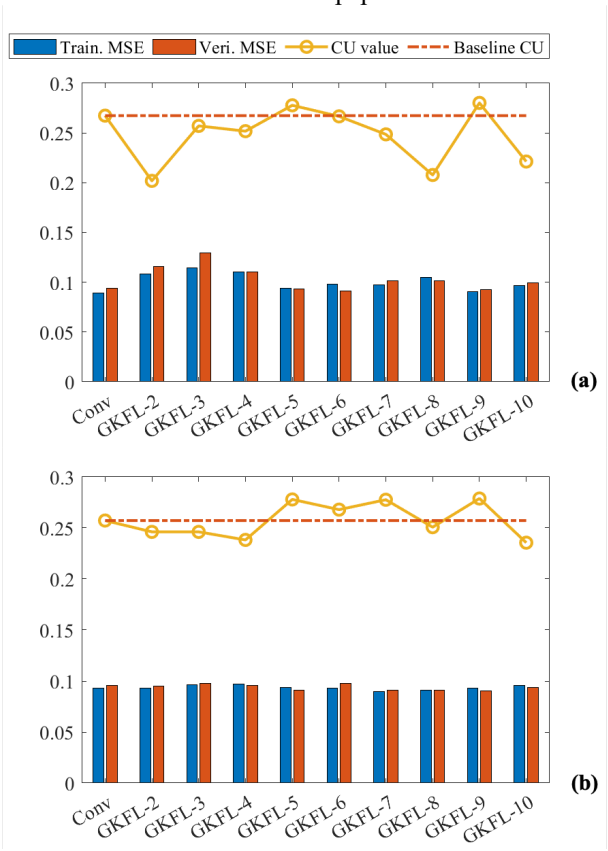


Fig. 8 Local learning performance under the Ex-WLTC cycle using (a) GA and (b) PSO algorithms.

4.2. Robustness of the GKFL model

To testify the robustness of the developed GKFL model, the model developed under Ex-WLTC conditions was implemented for real-time control under the other four driving cycles. The real-time controller built on the Artificial Neural Network (ANN), a widely-used AI model, is developed as a baseline method for comparison. The ANN has three hidden layers (10, 25, and 25 neurons for each layer) and has equal training efforts with the GKFL model. Using the same dataset obtained under Ex-WLTC condition. The CU values obtained by the GKFL-9W-based controller and the ANN-based controller are compared with the benchmark results (obtained with DP) under the five selected driving cycles in **Fig. 9**. The ANN-based controller can achieve higher CU value in Ex-WLTC than the GKFL-9W-based controller because ANN is more capable in data-based nonlinear system modelling. GKFL-9W-based controller is shown to be more adaptive in other driving cycles than the ANN-based controller.

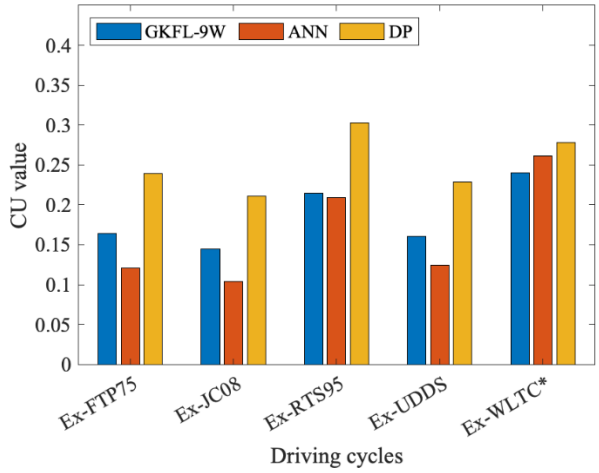


Fig. 9 CU values obtained under target domains and source domain (Ex-WLTC)

To quantify the performance of the real-time models, a distinction rate is introduced by

$$\partial = \frac{u_{\mathrm{rt}}}{u_{\mathrm{dp}}} \tag{19}$$

where u_{rt} is the CU value obtained with the real-time model (GKFL-9W or ANN) under the given driving cycle (e.g., Ex-FTP75), and u_{dp} is the CU value obtained with dynamic programming under the same driving cycle. The distinction rates obtained in target and source domains are compared in Table III. The distinction rates under the source domain (Ex-WLTC condition) obtained by the GKFL-9W-based controller and the ANN-based controller are 0.9107 and 0.9406, respectively. It indicates that both GKFL-9W-based controller and ANN-based controller can achieve competitive performance that is similar to the benchmarking result. The average distinction rate achieved by the GKFL-9W-based controller under the target domains is 0.7121, which is 27.53% higher than the distinction rate obtained by ANN-based controller (0.5583). It indicates that GKFL-9W-based controller is more robust than then ANN-based control model. Nevertheless, in target domains, the performance of the control model achieved is notas good as in the source domain.

Table III Distinction rates in target and source domains

	Distinction rate				
	Ex-FTP75	Ex-JC08	Ex-RTS95	Ex-UDDS	**Ex-WLTC***
GKFL-9W	0.6999	0.6861	0.7071	0.7554	**0.9107**
ANN	0.5040	0.4941	0.6915	0.5439	**0.9406**

4.3. Knowledge transfer across target domains

The knowledge transfer performance of the TRCM that incorporates the GPR model with the ANFIS model in a 'deeper' architecture (GS-AN-D for short) was evaluated by monitoring its CU value under the target domains. Another TRCM incorporating the GPR model with the ANFIS model in a 'broader' architecture (GS-AN-B for short) was developed for comparison. Both GPR model and ANFIS model (built with GKFL-9 method) were developed using the DP results under the

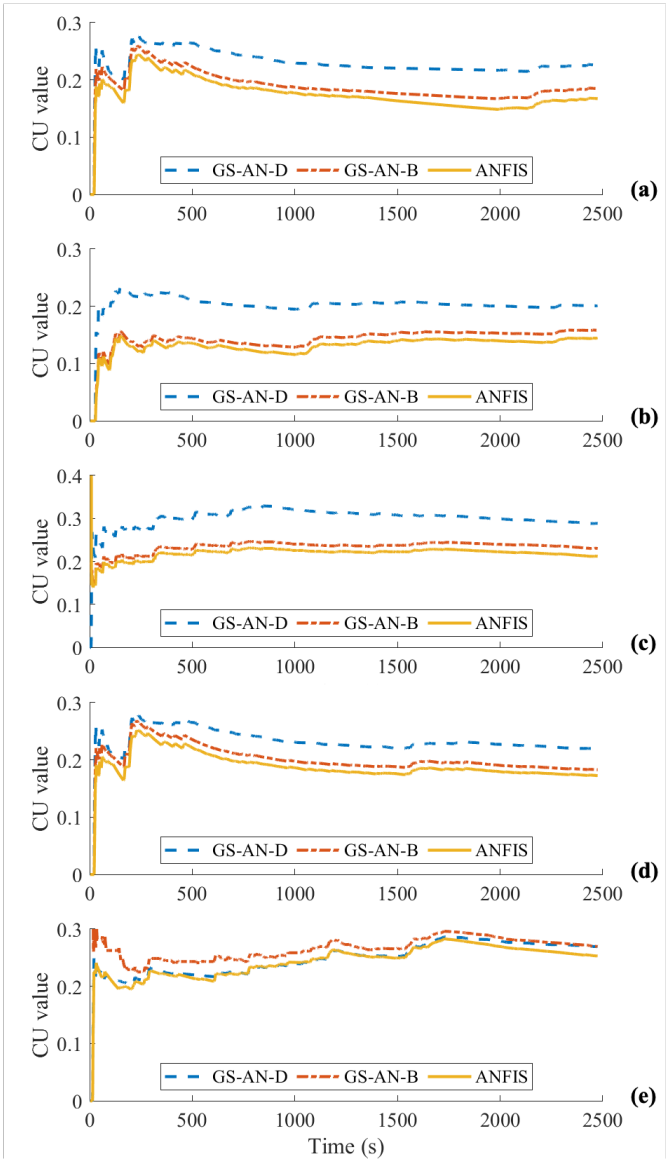


Fig. 10 CU value achieved under (a) Ex-FTP75, (b) Ex-JC08, (c) Ex-RTS95, (d) Ex-UDDS, and (e) Ex-WLTC.

Ex-WLTC condition. The real-time CU values achieved by GS-AN-D, GS-AN-B, and ANFIS are compared in Fig. 10. In target domains, as shown in **Fig. 10**a)-d), both GS-AN-D and GS-AN-B can achieve higher CU values than the ANFIS controller because both of them can transfer the knowledge (learnt from the source domain) to the target domains. The GS-AN-D achieves higher CU values than the GS-AN-B because the 'deeper' architecture includes the outputs of the ANFIS model as the input of the GPR model, and this can provide enhanced information for knowledge transfer.

To quantify the improvements of CU in target domains, the improving rate is introduced as

$$\Delta = \frac{\mathcal{U}_{\text{trcm}} - \mathcal{U}_{\text{anfis}}}{\mathcal{U}_{\text{anfis}}} \times 100\% \quad (20)$$

where $\mathcal{U}_{\text{trcm}}$ is the CU value obtained with the TRCM (GS-AN-D or GS-AN-B); and $\mathcal{U}_{\text{anfis}}$ is the CU value obtained with the ANFIS (GKFL-9W). The improving rates in target and source domains are compared in Table IV. In the target domains, an average improving rate of 34% can be achieved by the GS-AN-D method, and the GS-AN-B method can lead to an average improving rate of 8.1%. The maximum improving rate (38.6%) is achieved under Ex-JC08 condition with the GS-AN-D method. This paper also evaluated the GS-AN-D and GS-AN-B models in the source domain, and it indicates that both TRCMs can also contribute to CU value improvements in the source domain (by up to 6.3%).

Table IV Improving rates in target and source domains

	Improving rate					
	Ex-FTP75	Ex-JC08	Ex-RTS95	Ex-UDDS	**Ex-WLTC***	Mean
GS-AN-D	34.8%	38.6%	35.3%	27.3%	**6.3%**	34.0%
GS-AN-B	9.6%	9.4%	8.1%	5.2%	**4.3%**	8.1%

Performance of the PHEV equipped with the TRCM (GS-AN-D) is verified in **Fig. 11**. The illustrated variables include the real-time CU values, battery SoC, EGU's control command, and battery cell's voltage under Ex-FTP75, Ex-JC08, Ex-RTS95, Ex-UDDS, and Ex-WLTC conditions. The results are compared with benchmark results obtained by DP in the respective driving conditions. In general, the proposed method can achieve competitive performance with the benchmark results in real-time. It can maintain the battery SoC with similar trajectories to the DP results and ensure the voltages of battery cell satisfy its physical constraints (between 3.3V to 3.9V).

5. Conclusions

This paper studied a transfer learning routine to enable rapid development of real-time controller for energy management of PHEV. A new transferable representation control model is developed by incorporating two promising aritificial intelligence

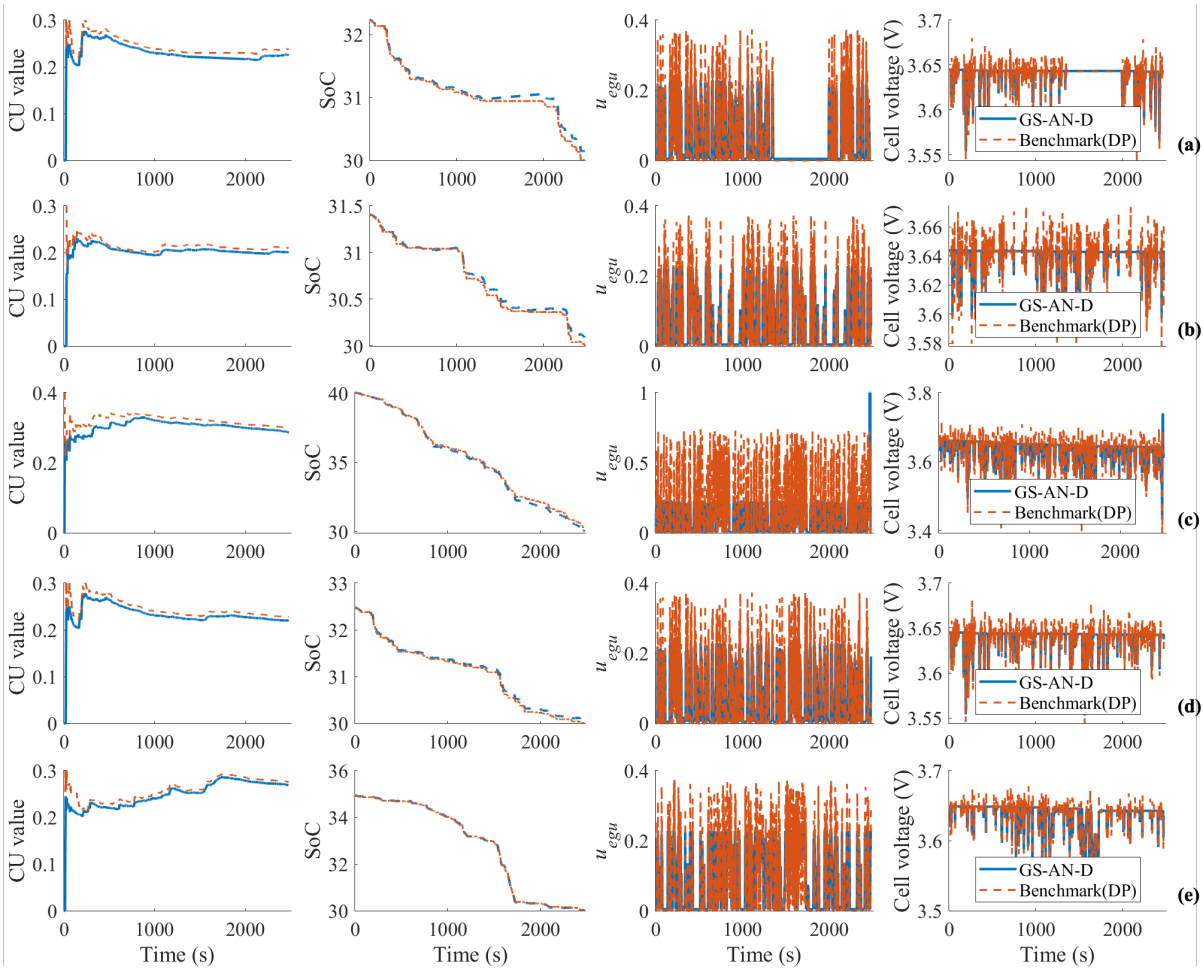


Fig. 11 Vehicle performance, including real-time CU value, battery SoC, EGU control command, and battery cell voltage under (a) Ex-FTP75, (b) Ex-JC08, (c) Ex-RTS95, (d) Ex-UDDS, and (e) Ex-WLTC.

technologies, adaptive neuro fuzzy inference and Gaussian process regression, where the former is used to build a generic control model and the later is implemented for transfer learning. By defining source and target domains based on five worldwide driving cycles, experimental evaluations based on hardware-in-the-loop testing were conducted. The conclusions drawn from this work are as follows:

1) By introducing k-fold cross validation in the learning process for development of the generic control model, the proposed global k-fold fuzzy learning is superior to the default MATLAB toolbox. The optimal control performance can be achieved when $\kappa = 9$.
2) The generic control model built on global k-fold fuzzy learning is robust in source domains and target domains. It achieves 27% higher distinction rate compared to the model built on artrificial neural network.
3) By incorporating the transfer learning model with the generic model in a 'deeper' architecture, the proposed control model is capable of knowledge transfer from source domain to target domains. It achieves average improvement of 34% in control ultility compared to control model without transfer learning.

In the future, the proposed control model will be incorporated within learning-based control systems, e.g., with reinforcement learning agents, to provide reliable control execution based on empirical knowledge and to guarantee the robustness of online learning.

Nomenclature

AI	Artifical Intelligence
ANN	Aritificial Neural Network
CM	Compensation Model
DQN	Deep Q Network
EMS	Energy Management System
FIS	Fuzzy Inference System
GA	Genetic Algorithm
GPR	Gaussian Process Regression
HEV	Hybrid Electric Vehicle
HiL	Hardware-in-the-Loop
IEA	International Energy Agency
IoT	Internet of Things
NEDC	New European Driving Cycle
PSO	Particle Swarm Optimization
RDE	Real-world Driving Emissions
RL	Reinforcement Learning
RM	Represeantion Model
RMSE	Root Mean Squared Error
TL	Transfer Learning
WLTC	Worldwide-harmonized Light-duty Testing Cycle

Acknowledgement

The authors are grateful to the State Key Laboratory of Automotive Safety and Energy (KF2029), the EPSRC Fellowship scheme (EP/S001956/1), and the Natural Science Foundation of China (51775393).

Reference

[1] IEA. Global EV Outlook 2019. 2019.

[2] Hu X, Han J, Tang X, Lin X. Powertrain Design and Control in Electrified Vehicles: A Critical Review. IEEE Trans Transp Electrif 2021. https://doi.org/10.1109/TTE.2021.3056432.

[3] Zhang F, Hu X, Langari R, Cao D. Energy management strategies of connected HEVs and PHEVs: Recent progress and outlook. Prog Energy Combust Sci 2019;73:235–56. https://doi.org/10.1016/j.pecs.2019.04.002.

[4] Malikopoulos AA. Supervisory Power Management Control Algorithms for Hybrid Electric Vehicles: A Survey. IEEE Trans Intell Transp Syst 2014;15:1869–85. https://doi.org/10.1109/tits.2014.2309674.

[5] Pavlovic J, Marotta A, Ciuffo B. CO2 emissions and energy demands of vehicles tested under the NEDC and the new WLTP type approval test procedures. Appl Energy 2016;177:661–70. https://doi.org/10.1016/j.apenergy.2016.05.110.

[6] Continental Automotive GmbH. Worldwide Emission Standards and Related Regulations. 2017.

[7] Giakoumis EG. Driving and engine cycles. Driv Engine Cycles 2016:1–408. https://doi.org/10.1007/978-3-319-49034-2.

[8] Zhou Q, Zhang Y, Li Z, Li J, Xu H, Olatunbosun O. Cyber-Physical Energy-Saving Control for Hybrid Aircraft-Towing Tractor based on Online Swarm Intelligent Programming. IEEE Trans Ind Informatics 2018;14:4149–58. https://doi.org/10.1109/TII.2017.2781230.

[9] Zhou Q, Li J, Shuai B, Williams H, He Y, Li Z, et al. Multi-step Reinforcement Learning for Model-Free Predictive Energy Management of an Electrified Off-highway Vehicle. Appl Energy 2019;255:588–601.

[10] Bellman R. Dynamic Programming and a New Formalism in the Calculus of Variations. Proc Natl Acad Sci 1954;40:231–5. https://doi.org/10.1073/pnas.40.4.231.

[11] Hu Y, Wang W, Liu H, Liu L. Reinforcement Learning Tracking Control for Robotic Manipulator With Kernel-Based Dynamic Model. IEEE Trans Neural Networks Learn Syst 2019:1–9. https://doi.org/10.1109/tnnls.2019.2945019.

[12] Radac MB, Precup RE. Data-driven model-free slip control of anti-lock braking systems using reinforcement Q-learning. Neurocomputing 2018;275:317–29. https://doi.org/10.1016/j.neucom.2017.08.036.

[13] Liu T, Zou Y, Liu D, Sun F. Reinforcement Learning of Adaptive Energy Management With Transition Probability for a Hybrid Electric Tracked Vehicle. IEEE Trans Ind Electron 2015;62:7837–46.

[14] Liu C, Murphey YL. Optimal Power Management Based on Q-Learning and Neuro-Dynamic Programming for Plug-in Hybrid Electric Vehicles. IEEE Trans Neural Networks Learn Syst 2019:1–13. https://doi.org/10.1109/tnnls.2019.2927531.

[15] Shuai B, Zhou Q, Li J, He Y, Li Z, Williams H, et al. Heuristic action execution for energy efficient charge-sustaining control of connected hybrid vehicles with model-free double Q-learning. Appl Energy 2020;267.

[16] Han X, He H, Wu J, Peng J, Li Y. Energy management based on reinforcement learning with double deep Q- learning for a hybrid electric tracked vehicle. Appl Energy 2019;254:113708. https://doi.org/10.1016/j.apenergy.2019.113708.

[17] Li W, Cui H, Nemeth T, Jansen J, Ünlübayir C, Wei Z, et al.

Cloud-based health-conscious energy management of hybrid battery systems in electric vehicles with deep reinforcement learning. Appl Energy 2021;293. https://doi.org/10.1016/j.apenergy.2021.116977.
[18] Qiu D, Ye Y, Papadaskalopoulos D, Strbac G. Scalable coordinated management of peer-to-peer energy trading: A multi-cluster deep reinforcement learning approach. Appl Energy 2021;292. https://doi.org/10.1016/j.apenergy.2021.116940.
[19] Hu X, Liu T, Qi X, Barth M. Reinforcement Learning for Hybrid and Plug-In Hybrid Electric Vehicle Energy Management: Recent Advances and Prospects. IEEE Ind Electron Mag 2019;13:16–25. https://doi.org/10.1109/mie.2019.2913015.
[20] Lian R, Tan H, Peng J, Li Q, Wu Y. Cross-Type Transfer for Deep Reinforcement Learning Based Hybrid Electric Vehicle Energy Management. IEEE Trans Veh Technol 2020;69:8367–80. https://doi.org/10.1109/TVT.2020.2999263.
[21] Guo X, Liu T, Tang B, Tang X, Zhang J, Tan W, et al. Transfer Deep Reinforcement Learning-enabled Energy Management Strategy for Hybrid Tracked Vehicle. IEEE Access 2020;8:165837–48. https://doi.org/10.1109/access.2020.3022944.
[22] Chau KT, Wong YS. Overview of power management in hybrid electric vehicles. Energy Convers Manag 2002;43:1953–68. https://doi.org/10.1016/S0196-8904(01)00148-0.
[23] Li J, Zhou Q, He Y, Williams H, Xu H. Driver-identified Supervisory Control System of Hybrid Electric Vehicles based on Spectrum-guided Fuzzy Feature Extraction. IEEE Trans Fuzzy Syst 2020;28:2691–701. https://doi.org/10.1109/tfuzz.2020.2972843.
[24] Profillidis VA, Botzoris GN, Profillidis VA, Botzoris GN. Artificial Intelligence—Neural Network Methods. Model. Transp. Demand, Elsevier; 2019, p. 353–82. https://doi.org/10.1016/B978-0-12-811513-8.00008-X.
[25] Ross TJ, Ross TJ. FUZZY LOGIC WITH ENGINEERING. 2010.
[26] Mamun A Al, Liu Z, Rizzo DM, Onori S. An integrated design and control optimization framework for hybrid military vehicle using lithium-ion battery and supercapacitor as energy storage devices. IEEE Trans Transp Electrif 2019;5:239–51. https://doi.org/10.1109/TTE.2018.2869038.
[27] Zhou Q, Zhang W, Cash S, Olatunbosun O, Xu H, Lu G. Intelligent sizing of a series hybrid electric power-train system based on Chaos-enhanced accelerated particle swarm optimization. Appl Energy 2017;189:588–601. https://doi.org/10.1016/j.apenergy.2016.12.074.
[28] He Y, Zhou Q, Makridis M, Mattas K, Li J, Williams H, et al. Multiobjective Co-Optimization of Cooperative Adaptive Cruise Control and Energy Management Strategy for PHEVs. IEEE Trans Transp Electrif 2020;6:346–55. https://doi.org/10.1109/TTE.2020.2974588.
[29] Khayyam H, Bab-Hadiashar A. Adaptive intelligent energy management system of plug-in hybrid electric vehicle. Energy 2014;69:319–35. https://doi.org/10.1016/j.energy.2014.03.020.
[30] Wu J Da, Hsu CC, Chen HC. An expert system of price forecasting for used cars using adaptive neuro-fuzzy inference. Expert Syst Appl 2009;36:7809–17. https://doi.org/10.1016/j.eswa.2008.11.019.
[31] Xing Y, Lv C, Cao D, Lu C. Energy oriented driving behavior analysis and personalized prediction of vehicle states with joint time series modeling. Appl Energy 2020;261:114471. https://doi.org/10.1016/j.apenergy.2019.114471.
[32] James G, Witten D, Hastie T, Tibshirani R. An Introduction to Statistical Learning. Springer; 2013. https://doi.org/10.1007/978-1-4614-7138-7_8.
[33] Lv C, Xing Y, Lu C, Liu Y, Guo H, Gao H, et al. Hybrid-Learning-Based Classification and Quantitative Inference of Driver Braking Intensity of an Electrified Vehicle. IEEE Trans Veh Technol 2018;67:5718–29. https://doi.org/10.1109/TVT.2018.2808359.
[34] Liu X-F, Zhan Z-H, Gu T-L, Kwong S, Lu Z, Duh HB-L, et al. Neural Network-Based Information Transfer for Dynamic Optimization. IEEE Trans Neural Networks Learn Syst 2019;31:1–14. https://doi.org/10.1109/tnnls.2019.2920887.
[35] Zuo H, Zhang G, Behbood V, Lu J. Feature Spaces-based Transfer Learning 2015:1000–5. https://doi.org/10.2991/ifsa-eusflat-15.2015.141.
[36] Duan L, Tsang IW, Xu D. Domain transfer multiple kernel learning. IEEE Trans Pattern Anal Mach Intell 2012;34:465–79. https://doi.org/10.1109/TPAMI.2011.114.
[37] Wang D, Li Y, Lin Y, Zhuang Y. Relational knowledge transfer for zero-shot learning. 30th AAAI Conf Artif Intell AAAI 2016 2016:2145–51.
[38] Zhao ZQ, Zheng P, Xu ST, Wu X. Object Detection with Deep Learning: A Review. IEEE Trans Neural Networks Learn Syst 2019;30:3212–32. https://doi.org/10.1109/TNNLS.2018.2876865.
[39] Ashok Kumar PM, Vaidehi V. A transfer learning framework for traffic video using neuro-fuzzy approach. Sadhana - Acad Proc Eng Sci 2017;42:1431–42. https://doi.org/10.1007/s12046-017-0705-x.
[40] Rasmussen CE, Williams CKI. Gaussian Processes for Machine Learning. MIT Press; 2006.
[41] Li Y, Sheng H, Cheng Y, Stroe DI, Teodorescu R. State-of-health estimation of lithium-ion batteries based on semi-supervised transfer component analysis. Appl Energy 2020;277:115504. https://doi.org/10.1016/j.apenergy.2020.115504.
[42] Chehade AA, Hussein AA. A Collaborative Gaussian Process Regression Model for Transfer Learning of Capacity Trends between Li-Ion Battery Cells. IEEE Trans Veh Technol 2020;69:9542–52. https://doi.org/10.1109/TVT.2020.3000970.
[43] Li Z, Gong J, Lu C, Xi J. Importance Weighted Gaussian Process Regression for Transferable Driver Behaviour Learning in the Lane Change Scenario. IEEE Trans Veh Technol 2020;69:12497–509. https://doi.org/10.1109/TVT.2020.3021752.
[44] Lu C, Hu F, Cao D, Gong J, Xing Y, Li Z. Transfer Learning for Driver Model Adaptation in Lane-Changing Scenarios Using Manifold Alignment. IEEE Trans Intell Transp Syst 2020;21:3281–93. https://doi.org/10.1109/TITS.2019.2925510.
[45] Deng Z, Guan H, Huang R, Liang H, Zhang L, Zhang J. Combining Model-Based Q-Learning with Structural Knowledge Transfer for Robot Skill Learning. IEEE Trans Cogn Dev Syst 2019;11:26–35. https://doi.org/10.1109/TCDS.2017.2718938.
[46] Zuo H, Zhang G, Pedrycz W, Behbood V, Lu J. Fuzzy Regression Transfer Learning in Takagi-Sugeno Fuzzy Models. IEEE Trans Fuzzy Syst 2017;25:1795–807. https://doi.org/10.1109/TFUZZ.2016.2633376.
[47] Yang X-S. Nature-inspired optimization algorithms. Elsevier; 2014.
[48] Liu C, Murphey YL. Optimal Power Management Based on Q−Learning and Neuro−Dynamic Programming for Plug−in Hybrid Electric Vehicles. n.d. https://doi.org/10.1109/tnnls.2019.2927531.
[49] Zhou Q, Li J, Shuai B, Williams H, He Y, Li Z, et al. Multi-step reinforcement learning for model-free predictive energy management of an electrified off-highway vehicle. Appl Energy 2019;255.

https://doi.org/10.1016/j.apenergy.2019.113755.
[50] Li J, Zhou Q, Williams H, Xu H. Back-to-back Competitive Learning Mechanism for Fuzzy Logic based Supervisory Control System of Hybrid Electric Vehicles. IEEE Trans Ind Electron 2019:1–1. https://doi.org/10.1109/tie.2019.2946571.