

Dual-loop online intelligent programming for driver-oriented predict energy management of plug-in hybrid electric vehicles

Ji Li, Quan Zhou, Yinglong He, Bin Shuai, Ziyang Li, Huw Williams, Hongming Xu*

Department of Mechanical Engineering, School of Engineering, University of Birmingham, B15 2TT, UK

HIGHLIGHTS

- Dual-loop online intelligent programming is proposed for HEV energy management.
- Deep fuzzy predictor is created for speed forecast via fuzzy granulation technology.
- The optimized predictive control strategy is validated by a driver-in-the-loop test.
- Up to 9% total energy can be saved by the proposed algorithm over real-world driving.

ARTICLE INFO

Keywords:

Driver-in-the-loop test
Fuzzy mathematics
Online predictive control
Particle swarm optimization
Plug-in hybrid electric vehicle

ABSTRACT

This paper investigates an online predictive control strategy for series-parallel plug-in hybrid electric vehicles (PHEVs), resulting in a novel online optimization methodology named the dual-loop online intelligent programming (DOIP) that is proposed for velocity prediction and energy-flow control. By reconsidering the change of driving behaviours at each look-ahead step, this methodology guarantees the effectiveness of optimal control sequence in the energy-saving efficiency of online predictive energy management. The design procedure starts with the simulation of a series-parallel PHEV using a systematic control-oriented model and the definition of a cost function. Inspired by fuzzy granulation technology, a deep fuzzy predictor is created to achieve driver-oriented velocity prediction, and a finite-state Markov chain is exploited to learn transition probabilities between vehicle speed and acceleration. To determine the optimal control behaviours and power distribution between two energy sources, chaos-enhanced accelerated swarm optimization is developed for the DOIP algorithm. The prediction capability of the deep fuzzy predictor is evaluated by comparing with two existing predictors over the WLTP-based driving cycle. The proposed control strategy is contrasted with short-sighted and dynamic programming based counterparts, and validated by a driver-in-the-loop test. The results demonstrate that the deep fuzzy predictor can effectively recognize driving behaviour and reduce at least 19% errors compared to involved Markov chain based predictors. Online predictive control strategy using the DOIP algorithm is able to significantly reduce 9.37% fuel consumption from the baseline and shorten computational time.

1. Introduction

Hybrid technology is a solution to environmental pollution that makes it possible to improve fuel economy and reduce exhaust emissions of vehicles [1,2]. Optimal energy management strategies are critical for hybrid electric vehicles (HEVs) to achieve best performance and greatest energy efficiency through power-split control [3]. Driver

behaviours are another primary element that affects energy consumption, emissions, and driving safety of a road vehicle. A study [4] of driver feedback, driver attitudes, and the adoption of eco-driving behaviours finds that drivers with high levels of technical proficiency can achieve a reduction of about 4.4% in fuel consumption. Thus, vehicle control strategies that seek highly optimized performance need to optimize the system composed of both the vehicle and the driver.

Abbreviations: PHEV, plug-in hybrid electric vehicle; DOIP, dual-loop online intelligent programming; WLTP, worldwide harmonized light vehicles test procedure; HEV, hybrid electric vehicle; MPC, model predictive control; MC, Markov chain; DP, dynamic programming; GA, genetic algorithm; CAPSO, chaos-enhanced accelerated particle swarm optimization; DiL, driver-in-the-loop; ISG, integrated starter-generator; ICE, internal combustion engine; DFP, deep fuzzy predictor; AR, auto-regression; NNP, nearest neighbour predictor; FEP, fuzzy encoding predictor

* Corresponding author.

E-mail address: h.m.xu@bham.ac.uk (H. Xu).

Nomenclature	
M	gross mass
A_f	windward area
R_{wh}	tire rolling radius
C_d	air drag coefficient
i_0	reducer ratio
i_g	transmission ratio
F_t	traction force
F_f	accelerative resistance force
F_w	aerodynamic resistance force
F_i	grade resistance force
F_j	rolling resistance force
θ	road slope
u_a	vehicle speed in km/h
P_d	demand power
T_d	demand torque
n_d	demand rotation speed
n_{mot}	motor rotation speed
T_{mot}	motor torque
T_{ICE}	ICE torque
n_{ICE}	ICE rotation speed
P_{ISG}	ISG power
P_{bra}	braking power
E_{fuel}	fuel consumption
k_{end}	driving cycle ending time
J	optimization target
w	weight coefficient
S_{vel}	pedal actions
k	sampling time
v	vehicle velocity
a	vehicle acceleration
p	transition probability
H	transition counts
λ	probability vector
Π	transition probability matrix
Φ	fuzzy subsets
μ	membership function
O	probability vector
z	possibility vector
\bar{c}	membership function centroid
\hat{V}	membership function volume
ϑ_r	AR coefficients
K	order of AR model
ε_k	i.i.d noise
τ	sample period
γ	vector of speed interval samples
ω	membership criterion
C	cluster centre
x	position of each particle
i	index of iterations
j	index of the particle
g	best position in iterations
β	attraction parameter
α	convergence parameter
ϵ	random number
$g_{\max_iter,*}$	best position at the end iteration
R	record matrix
R_{mirror}	mirror matrix

In 2015, a real-world driving emissions testing procedure [5] was proposed to further restrict vehicle emissions under on-road conditions. Classical control strategies have difficulty in meeting the requirements of this standard, because the driver's information is not easy to exploit in real time. Model predictive control (MPC) appears to be suitable for this purpose, and it can repeatedly optimize a control sequence over a receding horizon by exploiting a model to predict the future system behaviour [6]. Because of the capability of achieving high performance in multivariable systems subject to constraints, MPC has attracted considerable interest in the automotive industry (see reviews, [7–9] and references therein). However, the performance of MPC is also determined by the precision of future velocity or power forecasts and its prediction model is hard formulated due to strong randomness and uncertainty [10]. The authors have proposed a real-time nonlinear model-based predictive energy management method, which makes it possible to carry on online nonlinear control optimization with advanced predictive models [11]. This research will develop advanced driver behaviour prediction methods to further improve the vehicle efficiency with online nonlinear model-based predictive control.

To optimize the overall system composed of vehicle and driver, prediction models of the MPC must capture the driver behaviour. In some applications on vehicle control systems, Markov chain (MC) models have been utilized to predict the vehicle speed or torque demand signal as follows. Ref. [12] employs the K-means to classify driving behaviours with rigid boundaries but the uncertainty of driving behaviour is not considered. Ref. [13] integrates MC models and dynamic programming (DP) to implement stochastic model predictive control for plug-in hybrid electric buses. In fact, some dramatic driving states may be homogenized into a very low probability distribution or even ignored altogether in the training process of a MC model. Ref. [14] proposes a velocity predictor with fuzzy encoding to improve its prediction accuracy. But such learning-based prediction methods rely on previous driving data, wherein once it is finished the training, the

structure and weight of the prediction model cannot be changed. To recognize the driving pattern, a fuzzy logic pattern recognition [15,16] has been widely used for the adaptive energy management of HEVs, in which different driving cycle patterns are represented by a fuzzy membership function and thus characterizes the driving pattern. In Ref. [17], a radial basis function neural network is introduced for time series forecasting and it is trained based on pedal position and historical velocity. As another industrial application for driving pattern classifiers in Ref. [18] driven by neural networks are employed to identify driving patterns in real time. Other types of artificial intelligence-based predictive model contain Bayesian algorithms [19], fuzzy cognitive maps [20], and auto-regressive models [21,22]: these algorithms can be used for modelling, learning, and predicting. If the driving behaviours exhibit dramatic changes, however, the prediction will become poor so that online updates need to be considered.

In the field of HEV energy consumption optimization, there has been much research into improving the efficiency and adaptability of HEV energy management. In Refs. [23,24], a reinforcement learning algorithm is employed to optimally control HEVs using MC-based predictors. Compared with Ref. [23], the main improvement of Ref. [24] is to utilize an online updating policy for updating the transition probability of the MC models. Ref. [25] proposes a co-design optimization of the plant and controller parameters, where Gaussian mixture models are used to classify driving styles. However, for computational reasons, it can fail to work if the dimensionality of the clustering problem is too high. Particle swarm algorithms [26] with the fast convergence speed are widely used for solving optimal control sequence in the HEV energy management, which is adopted by Refs. [16,27]. Actually, the classical particle swarm optimization algorithm may easy result in a local optimum in high-dimensional space and has a low convergence rate in its iterative scheme. In Ref. [28], a genetic algorithm (GA) is used to optimize the power threshold at which engine is turned on, which makes the engine work more efficiently and thus reduces the fuel-

consumption. The DP as a representative of global optimization algorithms usually depends on a model to provide a provably optimal control strategy by searching all state and control grids exhaustively [15]. However, DP or GA is not applicable to real-time problems since the exact future driving information is seldom known in the real world [29].

To help improve the energy-saving efficiency of online predictive energy management, this paper proposes a dual-loop online intelligent programming (DOIP) to guarantee the effectiveness of optimal control sequence for HEV systems. By reconsidering the change of driving behaviours at each look-ahead step, the new methodology with higher precision of predicted velocity trajectories involves two online iteration loops to simultaneously update the predictive model and optimize the control sequence. Firstly, a PHEV under research is analysed and its optimal problem is formulated. Subsequently, inspired by the authors' past research [30], interval fuzzy and deep fuzzy prediction methods are developed and compared, in terms of the performance of velocity prediction; meanwhile, a finite-state MC is exploited to learn transition probabilities from the vehicle speed to acceleration. The chaos-enhanced accelerated particle swarm optimization (CASPO) algorithm is harnessed to realize the predictive optimal control for increasing fuel economy and maintaining battery charge level. Finally, the performance of different predictors is compared over the WLTP-based driving cycle and the DOIP-driven predict control strategy is contrasted with the benchmarking DP to validate its effectiveness. In addition, the DOIP-driven control strategy is verified through a driver-in-the-loop (DiL) experiment over the real world driving. Three perspectives are contributed to the related literature.

- (1) Design methodology of a novel online predictor is presented, and its performance is compared with the existing MC-based predictors.
- (2) A comparison between the DOIP-based optimal control and DP-based one is presented.
- (3) A DiL experiment is carried out to demonstrate the performance of the proposed energy management strategy and its online computational efficiency.

In Section 2, the configuration of the series-parallel PHEV is presented, and its optimal control problem is formulated. The mechanism of the proposed DOIP is described in Section 3, where two fuzzy-based prediction approaches and CAPSO algorithm are applied. Section 4

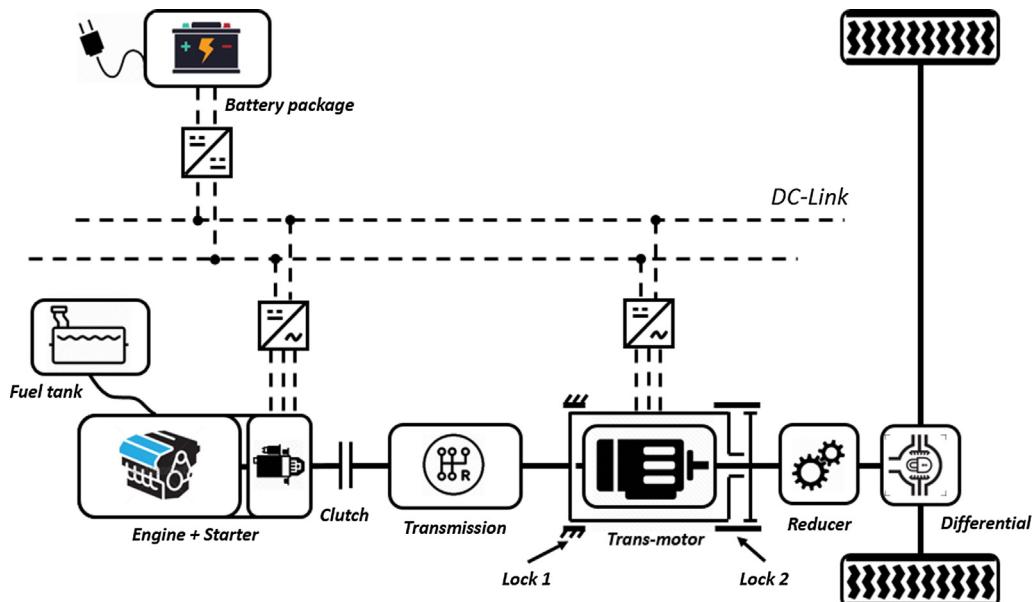


Fig. 1. The architecture of series-parallel PHEV powertrain.

Table 1
Main parameters of the PHEV model.

Symbol	Parameters	Values
M	Gross mass	1500 kg
A_f	Windward area	2m^2
R_{wh}	Tire rolling radius	0.3m
C_d	Air drag coefficient	0.3
i_0	Reducer ratio	3.75
i_g	Transmission ratio	3.55/1.96/1.30/0.89/0.71

defines the testing cycle, including the human driver who created it as well as the DiL experimental platform. Section 5 sets out a comparative study of different predictors and control policies for the PHEV energy management. Conclusions are summarized in Section 6.

2. Vehicle configuration and problem formulation

2.1. PHEV configuration

The series-parallel PHEV powertrain architecture includes one gasoline engine, one integrated starter-generator (ISG), one trans-motor and two energy sources of fuel and electricity as shown in Fig. 1. In this case, the post-transmission powers of an internal combustion engine (ICE) and a trans-motor are combined by coupling their speeds, while the speeds of the two power plants are decoupled to be chosen freely [31]. The peak power of the trans-motor is $P_{mot^*} = 75 \text{ kW}$ with 270 Nm peak torque. The peak power of the gasoline engine is $P_{ICE^*} = 63 \text{ kW}$ with 140 Nm peak torque. The peak power of the ISG is $P_{ISG^*} = 32 \text{ kW}$. The data for all the components is provided by ADVISOR software for vehicle driveline system analysis and optimization. The main parameters of the PHEV model are shown in Table 1.

In the series-parallel PHEV dynamics model, the traction force F_t needs to overcome various resistances for keeping the force balance in the vehicle. The traction force F_t and resistances can be sequentially expressed as:

$$F_t = F_f + F_w + F_i + F_r$$

$$\begin{cases} F_f = Mgf \cos\theta \\ F_w = \frac{C_d A_f}{21.15} u_a^2 \\ F_i = Mgsin\theta \\ F_j = \delta Ma \end{cases} \quad (1)$$

where F_f , F_w , F_i , F_j are accelerative, aerodynamic, grade and rolling resistance forces respectively; f is the rolling resistance coefficient; θ is the road slope; u_a is the vehicle speed in km/h; δ is the vehicle rotating quality conversion coefficient; a is the vehicle acceleration; u is the vehicle velocity. Derived by Eq. (1), the demand power P_d of the vehicle can be calculated as given by

$$P_d = P_f + P_w + P_i + P_j = \left(Mgf \cos\theta + \frac{C_d A_f}{21.15} u_a^2 + Mgsin\theta + \delta Ma \right) u \quad (2)$$

In which the demand torque T_d after a reducer is calculated as given by:

$$T_d = \left(Mgf \cos\theta + \frac{C_d A_f}{21.15} u_a^2 + Mgsin\theta + \delta Ma \right) \frac{R_{wh}}{i_0} \cdot \eta_{i_0} \quad (3)$$

where R_{wh} is tire rolling radius; i_0 is a reducer ratio; and $\eta_{i_0} = 0.95$ is the efficiency of the worm-gear speed reducer.

2.2. Multi-objective problem formulation

2.2.1. Search area and constraints

The rotation speed of the motor (the relative speed of the rotor to the stator) and the power of the ISG are two optimization variables involved in this research, their boundary conditions need to be constrained as given by:

$$\begin{cases} 0 < n_{mot}(k) < n_{mot}^* \\ -P_{ISG}^* < P_{ISG}(k) < 0 \end{cases} \quad (4)$$

Due to the limitation of peak powers and the layout of the PHEV powertrain, it is necessary to constrain for ICE, ISG and the traction motor during the optimization, which are formulated as:

$$\begin{cases} T_{mot} = T_d(k), & 0 < T_{mot}(k) < T_{mot}^* \\ T_{ICE} = T_d(k), & 0 < T_{ICE}(k) < T_{ICE}^* \\ n_{ICE} = \left(P_d(k) - \frac{T_{mot}(k) \cdot n_{mot}(k)}{9550} - P_{ISG}(k) \right) \cdot \frac{9550}{T_{ICE}(k)}, & 0 < n_{ICE}(k) < n_{ICE}^* \\ P_{bra} = \frac{T_d(k) \cdot n_d(k) - T_{mot}(k) \cdot n_{mot}(k)}{9550}, & P_{bra}(k) < 0 \end{cases} \quad (5)$$

where T_{mot} is the torque of the motor; T_{ICE} is the torque of the engine; n_{ICE} is the rotation speed of the engine; P_{bra} is the braking torque; n_d is the demand speed. To ensure the BP is performing in proper condition and protect the BP from over discharge, the battery's state of charge should obey [32]:

$$0.2 \leq SoC(k) \leq 0.8 \quad (6)$$

The total power generated by the powertrain needs to meet:

$$P_d = \frac{T_{mot}(k) \cdot n_{mot}(k) + T_{ICE}(k) \cdot n_{ICE}(k)}{9550} + P_{bra}(k) \quad (7)$$

2.2.2. Cost function

Two main targets are mainly concerned in this paper, one is the final fuel consumption from the fuel tank and the BP, and another is the BP's SoC. These optimization targets are defined as:

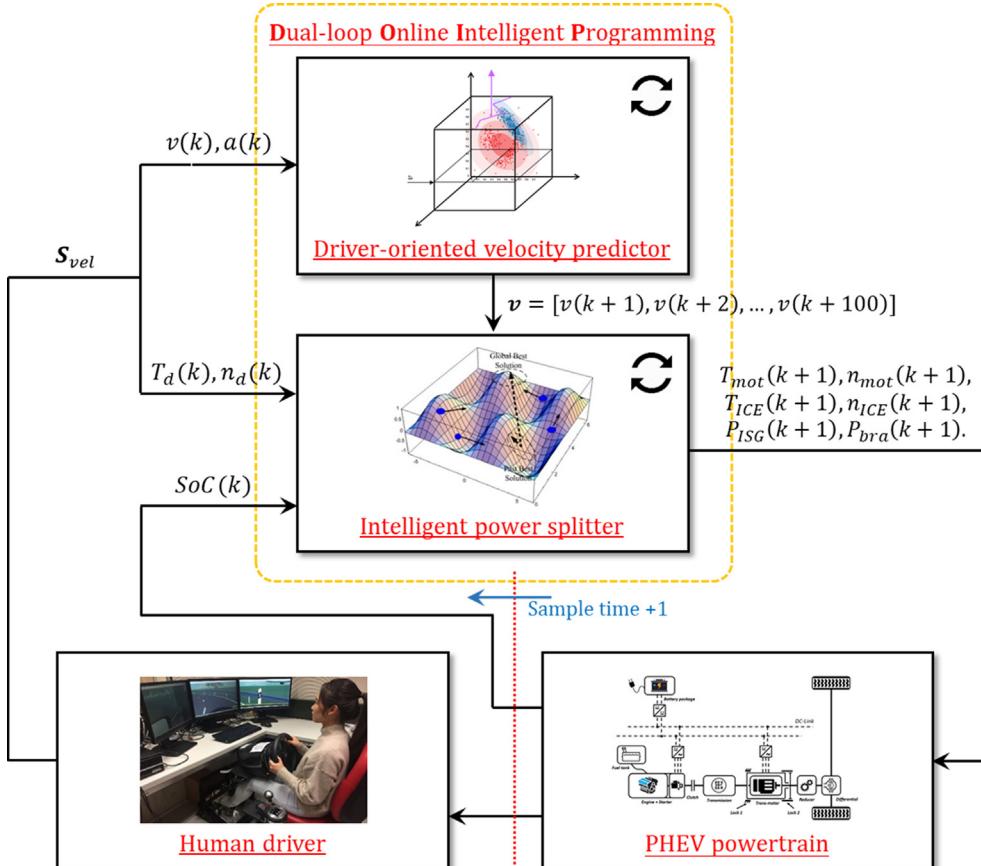


Fig. 2. Mechanism of dual-loop online intelligent programming.

$$\begin{cases} J_1 = \sum_{k=1}^{k_{end}} E_{fuel}(k) \\ J_2 = SoC(k_{end}) \end{cases} \quad (8)$$

where, E_{fuel} denotes the instantaneous fuel consumption at k time-step; k_{end} is the ending time of the driving cycle.

To convert the multi-objective optimization problem into a single objective optimization for CAPSO algorithm, in the present work, the multi-objective optimization is formulated by using the weighted sum method [33]. Therefore, the optimal energy-flow control problem is described as:

$$\min J = w \cdot J_1 \cdot \frac{1}{J_1^*} + (1 - w) \cdot J_2 \cdot \frac{1}{J_2^*}$$

$$\text{s. t. } \begin{cases} 0 < n_{mot}(k) < n_{mot}^* \\ -P_{ISG}^* < P_{ISG}(k) < 0 \\ T_{mot} = T_d(k), \quad 0 < T_{mot}(k) < T_{mot}^* \\ T_{ICE} = T_d(k), \quad 0 < T_{ICE}(k) < T_{ICE}^* \\ n_{ICE} = \left(P_d(k) - \frac{T_{mot}(k) \cdot n_{mot}(k)}{9550} - P_{ISG}(k) \right) \cdot \frac{9550}{T_{ICE}(k)}, \\ 0 < n_{ICE}(k) < n_{ICE}^* \\ P_{bra} = \frac{T_d(k) \cdot n_{d}(k) - T_{mot}(k) \cdot n_{mot}(k)}{9550}, \quad P_{bra}(k) < 0 \\ 0.2 \leq SoC(k) \leq 0.8 \\ P_d = \frac{T_{mot}(k) \cdot n_{mot}(k) + T_{ICE}(k) \cdot n_{ICE}(k)}{9550} + P_{bra}(k) \end{cases} \quad (9)$$

where w is a weight coefficient; J_1^* and J_2^* are scaling coefficients of optimization targets J_1, J_2 . Here, the optimization target J_2 is formulated as the penalty function in the cost function.

3. Dual-loop online intelligent programming

Dual-loop online intelligent programming (DOIP) involves one driver-oriented velocity predictor and one intelligent power splitter, which takes over the real-time optimal control system of the vehicle. Real pedal actions S_{vel} by the human driver and the vehicle state data are sent to the DOIP algorithm for online optimization, then a real optimal control signal will be sent back to the powertrain for the energy distribution. 0.1 s is chosen according to [29] as the sampling time k , which is approved to be able to track the system dynamics while reserving enough time slot for algorithm computing. The mechanism of the DOIP for the PHEV system is shown in Fig. 2.

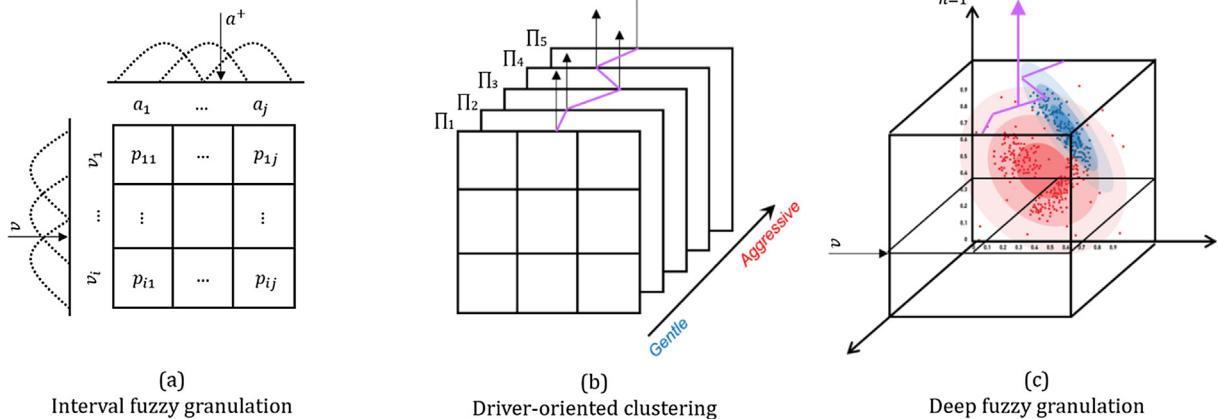


Fig. 3. Fuzzy granulation evolution for MC models.

3.1. Interval fuzzy predictor

In this paper, the vehicle velocity and acceleration are described as a finite-state MC [34], its state space is denoted as $V = v_i | i = 1, \dots, M \subset X \subset R$, and $W = a_j | j = 1, \dots, N \subset Y \subset R$. The transition probabilities may be estimated from the frequencies of observed transitions as given by

$$\begin{cases} p_{ij} = P(a^+ = a_j | v=v_i) = \frac{H_{ij}}{H_i} \\ H_i = \sum_{j=1}^N H_{ij} \end{cases} \quad (10)$$

where v is the present velocity; a^+ is the next step acceleration; p_{ij} is the transition probability from v_i to a_j ; H_{ij} indicates the transition counts from v_i to a_j ; H_i is the total transition counts initiated from v_i ; the transition probability matrix Π is filled with elements p_{ij} . Motivated by (10), the probability vector of the next state is defined as

$$(\lambda^+(a))^T = (\lambda(v))^T \Pi = \prod_j^T \quad (11)$$

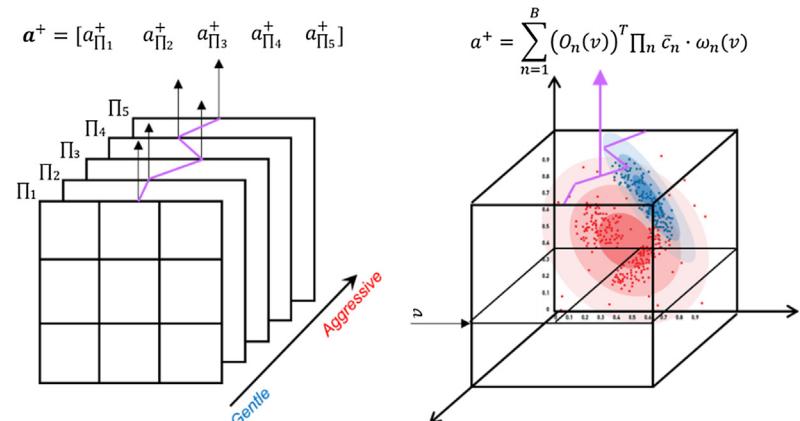
where $\lambda^+(v) = [0 \dots 1 \dots 0]$ is an N -dimensional probability vector with the j th element, to indicate a discrete state a_j in disjoint intervals $I_j, j = 1, \dots, N$; \prod_j^T denotes the j th row of the transition probability matrix Π . In the fuzzy encoding technique, X and Y are divided into finite sets separately with fuzzy subsets $\Phi_i, i = 1, \dots, M$ and $\Phi_j, j = 1, \dots, N$. The fuzzy subset Φ_i and Φ_j are pairs of $(X, \mu_i(\cdot))$ and $(Y, \mu_j(\cdot))$, where $\mu_i(\cdot), \mu_j(\cdot)$ are Lebesgue measurable membership functions that satisfy the following property:

$$\begin{cases} \mu_i: X \rightarrow [0, 1] \quad \text{s. t. } \forall v \in X, \exists i, \quad 1 \leq i \leq M, \quad \mu_i(v) > 0 \\ \mu_j: Y \rightarrow [0, 1] \quad \text{s. t. } \forall a \in Y, \exists j, \quad 1 \leq j \leq N, \quad \mu_j(a) > 0 \end{cases} \quad (12)$$

where $\mu_i(v)$ reflects the degree of membership of $v \in X$ in μ_i ; $\mu_j(a)$ reflects the degree of membership of $a \in Y$ in μ_j . Based on the theory of approximate reasoning [35], the transformation with normalization allocates an M -dimensional probability vector for each $v \in X$ as follows:

$$(O(v))^T = \left[\frac{\mu_1(v)}{\sum_{i=1}^M \mu_i(v)}, \frac{\mu_2(v)}{\sum_{i=1}^M \mu_i(v)}, \dots, \frac{\mu_M(v)}{\sum_{i=1}^M \mu_i(v)} \right] \quad (13)$$

This transformation is used to do normative fuzzification and map velocity in the space X to vector in M -dimensional probability vector space \bar{X} , and the sum of the elements in the probability vector $\sim O(v)$ equals to 1. The probability distribution of the next state in \bar{Y} is computed based on Eq. (11), then aggregated with membership function $\mu(a)$ to decode vectors in \bar{Y} back to the space Y as given by:



$$z^+(a) = (O^+(v))^T \mu(a) = (O(v))^T \prod \mu(a) \quad (14)$$

where the element p_{ij} in the transition probability matrix Π is interpreted as a transition probability between Φ_i and Φ_j . The membership function $\mu(a)$ is used to encode the probability vector of the next state in the space Y .

In this paper, it is assumed that membership functions have the same volume from which it follows that $\sum_{j=1}^N p_{ij} = 1$ and $\sum_{i=1}^M O_i(v) = 1$, and the next one-step-ahead velocity is calculated and simplified to

$$\left\{ \begin{array}{l} a^+ = \frac{\sum_{i=1}^M O_i(v) \sum_{j=1}^N p_{ij} \bar{V}_j \bar{c}_i}{\sum_{i=1}^M O_i(v) \sum_{j=1}^N p_{ij} V_j} = (O(v))^T \prod \bar{c} \\ v^+ = v + a^+ \end{array} \right. \quad (15)$$

where \bar{c}_i and \bar{V}_j are the centroid and volume of the membership function $\mu_j(v)$.

3.2. Deep fuzzy predictor

As driver-oriented predict models, the deep fuzzy predictor (DFP) is created with multi-dimensional fuzzification to improve the precision of future velocity predictions by reconsidering driving behaviours for each look-ahead step. Unlike a single MC model with fuzzy encoding, the DFP involves five driver-oriented MC models, which are classified by fuzzy C-mean clustering algorithm. A membership criterion vector solved by clustering is utilized as weighted sum coefficients to aggregate the predicted accelerations of the five driver-oriented MC models. The fuzzy granulation evolution for the MC models is drawn in Fig. 3 and the production process of the DFP is presented as follows.

The auto-regression (AR) model is a proven tool for generalizing the signal's average time regressive pattern and predicting by following the dynamic. The AR model used in this study follows the structure described in [21]:

$$v(k) = \sum_{r=1}^K \vartheta_r v(k-r) + \varepsilon_k \quad (16)$$

where ϑ_r are the AR coefficients; K is the order of AR model; ε_k refers to the i.i.d noise; $v(k)$ is the vehicle speed at time step k , with sample

period $\tau = 0.1$ s in this study.

Due to real-world driving involving frequent transitions in the driver behaviour, the AR model is used in the moving horizon way to extract driver-oriented velocity information, in which parameters of past measurement horizon length and the AR model order R need to be investigated. AR models of horizons ranging from 10 to 500 s and with orders from 1st to 4th have been tested on 9000 s WLTP-based driving cycles. Second order AR models with 200 s horizons show consistent advantage judged by the Corrected Akaike Information Criterion [36]. Its result is described as data vector γ , of speed interval samples, which contains four sets of information as given by

$$\gamma_r = [\vartheta_{r1} \ \vartheta_{r2} \ a_{r_avg} \ a_{r_maxR}] \quad (17)$$

where the AR coefficient set ϑ provides the trend of sampling speed change; average acceleration rate a_{r_avg} marks the average state and the maximum acceleration range a_{r_maxR} marks the range of acceleration changes.

Considering the computation efficiency and prediction accuracy, 5-layer Markov-chain model is reliable for training purpose as has been proved by Ref. [21]. In this paper, the AR model coefficient sets are classified into 5 clusters representing different acceleration states to label the estimated AR models to some specific driver states. The five clusters are fuzzified to reflect the acceleration range relationship among different driving behaviours, the driving behaviours are noted as 1. Very Gentle, 2. Gentle, 3. Normal, 4. Aggressive, 5. Very Aggressive. As there is no prior information on the predicted vehicle's performance or the driver's preference, the fuzzy C-mean method with unsupervised learning process is recommended for dividing information with less strict internal borders and unpredictable external borders. It is more sensitive to the isolated point i.e. dramatic driving state. The method uses a membership criterion $\omega_{k,n}$ and the Euclidean distance from data $x_k (k \in \{1, \dots, A\})$ to identify multiple cluster centres $C_n (n \in \{1, \dots, B\})$. The cluster centers C_n are iterated till the total distance is minimized [37]:

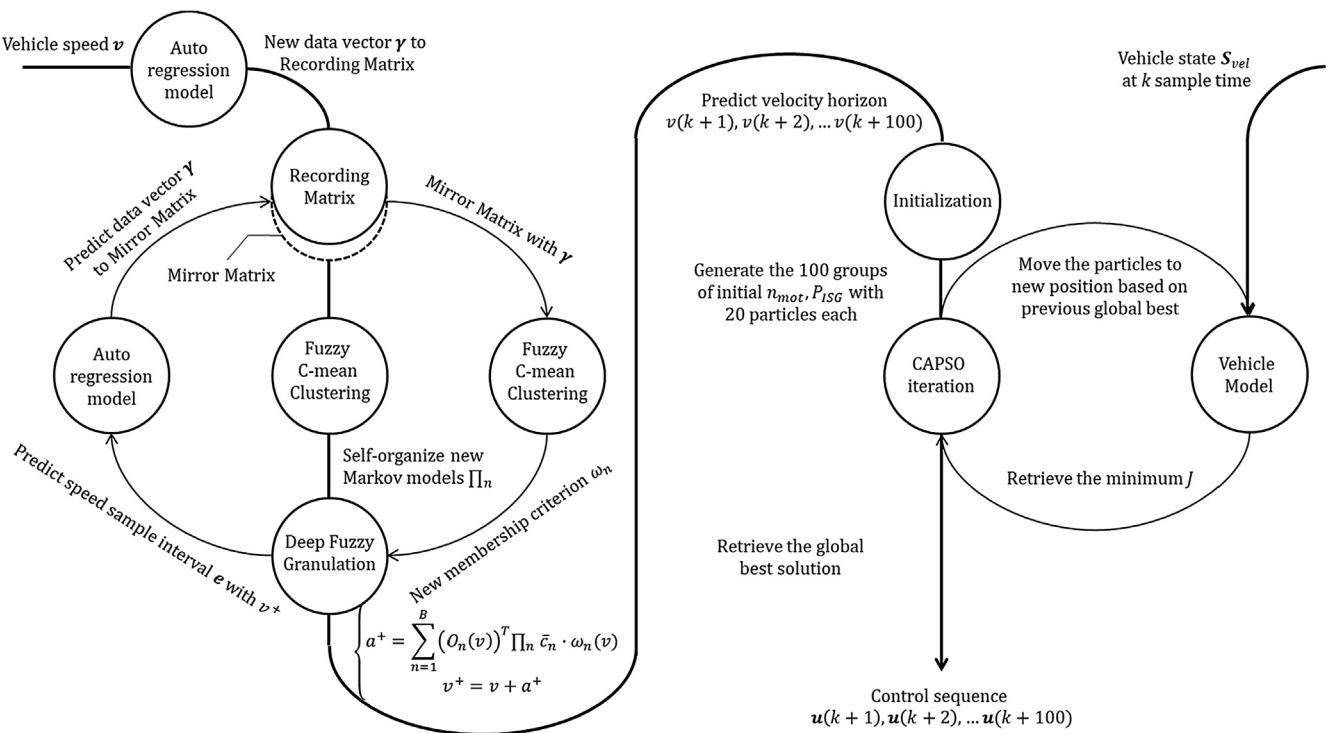


Fig. 4. Workflow of dual-loop online intelligent programming.

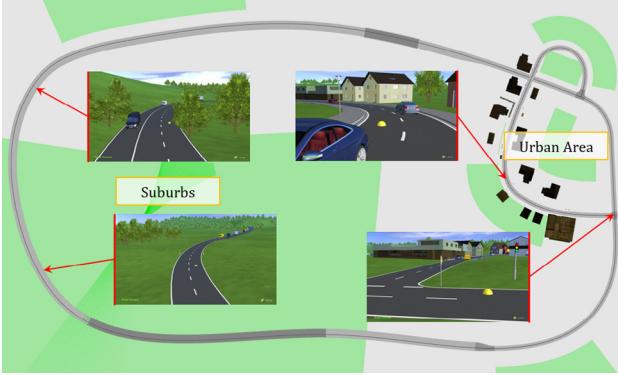


Fig. 5. Driving scenario with traffic provided by IPG CarMaker.

Table 2
Specification of real-world driving cycle.

Human driver	Traffic type	Driving time (s)	Driving distance (km)
A	Urban	1880	11.6
B	Urban	1590	17.9
C	Urban	940	8.1
D	Highway	1350	22.5
E	Highway	2240	40.0



Fig. 6. Driving simulator used in this research.

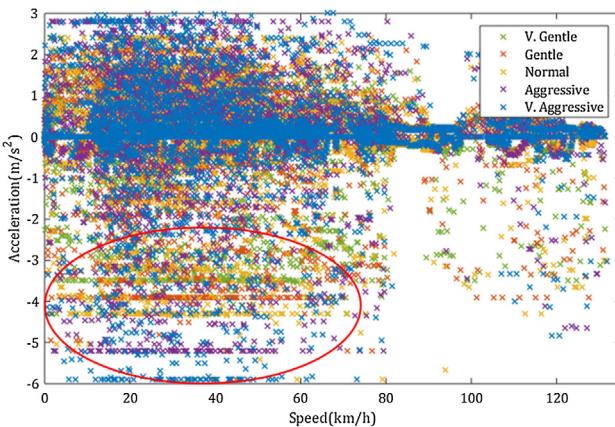


Fig. 7. Clustering results of driving behaviour by the DFP.

$$J = \sum_{k=1}^A \sum_{n=1}^B \omega_{k,n}^m \|x_k - C_n\|^2 \quad (18)$$

$$\begin{cases} \omega_{k,n} = \frac{1}{\sum_{k=1}^B \left(\frac{\|x_k - C_n\|}{\|x_k - C_n\|} \right)^{m-1}} \\ C_n = \frac{\sum_{k=1}^B \omega_{k,n}^m \gamma_k}{\sum_{k=1}^B \omega_{k,n}^m} \end{cases} \quad (19)$$

where, parameter m ($m > 1$) controls the fuzziness of cluster overlapping, which is set at $m = 2$ in this study. The classification results provide data's membership distributions with respect to all of the clusters. According to driving behaviour classification, the transition probability matrix Π in Eq. (11) is expanded into five transition probability matrixes as given by

$$|\Pi_1 \ \Pi_2 \ \Pi_3 \ \Pi_4 \ \Pi_5| \quad (20)$$

where the transition probability matrix with customized division can more accurately reflect the probability distribution of acceleration under different driving behaviours. Therefore, the next one-step-ahead accelerations by driver-oriented clustering (Fig. 3b) can be translated to

$$a_n^+ = (O_n(v))^T \prod_n \bar{c}_n \quad (21)$$

Here, the membership criterion is utilized as weighted sum coefficient to aggregate predict acceleration of five driver-oriented MC models. Based on Eq. (21), the next one-step-ahead velocity is calculated as given by

$$\begin{cases} a^+ = \sum_{n=1}^B (O_n(v))^T \prod_n \bar{c}_n \cdot \omega_n(v) \\ v^+ = v + a^+ \end{cases} \quad (22)$$

where, the membership criterion vector $\omega_n(v)$ corresponds to the data vector γ_r of the speed interval sample.

3.3. Intelligent power splitter and DOIP workflow

In this section, an intelligent power splitter is designed based on CAPSO algorithm, which has three main procedures, namely, initialization, main iteration, and optimal position retrieving. The details and principle of the CAPSO algorithm working procedure are discussed in the author's previous work in [11]. To solve the optimization problem in Eq. (9) online, the algorithm is customized and modified in the following aspects. At the initialization procedure, the position of each particle is defined as:

$$x^{(i,j)} = [n_{mot}^{(i,j)} \ P_{ISG}^{(i,j)}] \quad (23)$$

Here, the superscript i is an index of iterations, for a swarm intelligent algorithm that has $M = 15$ iterations, $i = [1, 2, 3 \dots M]$. The superscript j is the index of the particle, for swarm that has $N = 20$ particles, $j = [1, 2, 3 \dots N]$. $n_{mot}^{(i,j)}$, $P_{ISG}^{(i,j)}$ are the rotation speed of motor and the demand power of an ISG in the j th agent and i th iteration. For the CAPSO, the particles position updates with the following equation:

$$x^{(i+1,j)} = (1 - \beta) \cdot x^{(i,j)} + \beta \cdot g^{(i,*)} + \alpha^{(i)} \cdot \epsilon^{(i,j)} \quad (24)$$

In Eq. (24), $g^{(i,*)}$ is the best position in the i th iteration, β is the attraction parameters of CAPSO, α is the convergence parameters of CAPSO and ϵ is the $[0, 1]$ random number. Here, α and β could be updated respectively in each iteration as:

$$\begin{cases} \alpha^{(i)} = \alpha^{(0)} \cdot \gamma^i \\ \beta^{(i+1)} = a \cdot \beta^{(i)} \cdot (1 - \beta^{(i)}) \end{cases} \quad (25)$$

where the setting range of $\alpha^{(0)}$ and γ are $\alpha^{(0)} = 0.9$, $\gamma = 0.95$ in this paper; the attraction parameter is mapped by the logistic map [38], in which the initial value $\beta^{(1)} = 0.6$ and $a = 4$ are used. When $\beta \rightarrow 0$ in any step, the algorithm may lead to slow changes. After the convergence has been achieved, the algorithm ends the main iteration and outputs the best position at the end iteration $g_{\max_iter,*}$ as the global optimal solution.

The proposed DOIP has two online iteration loops for updating

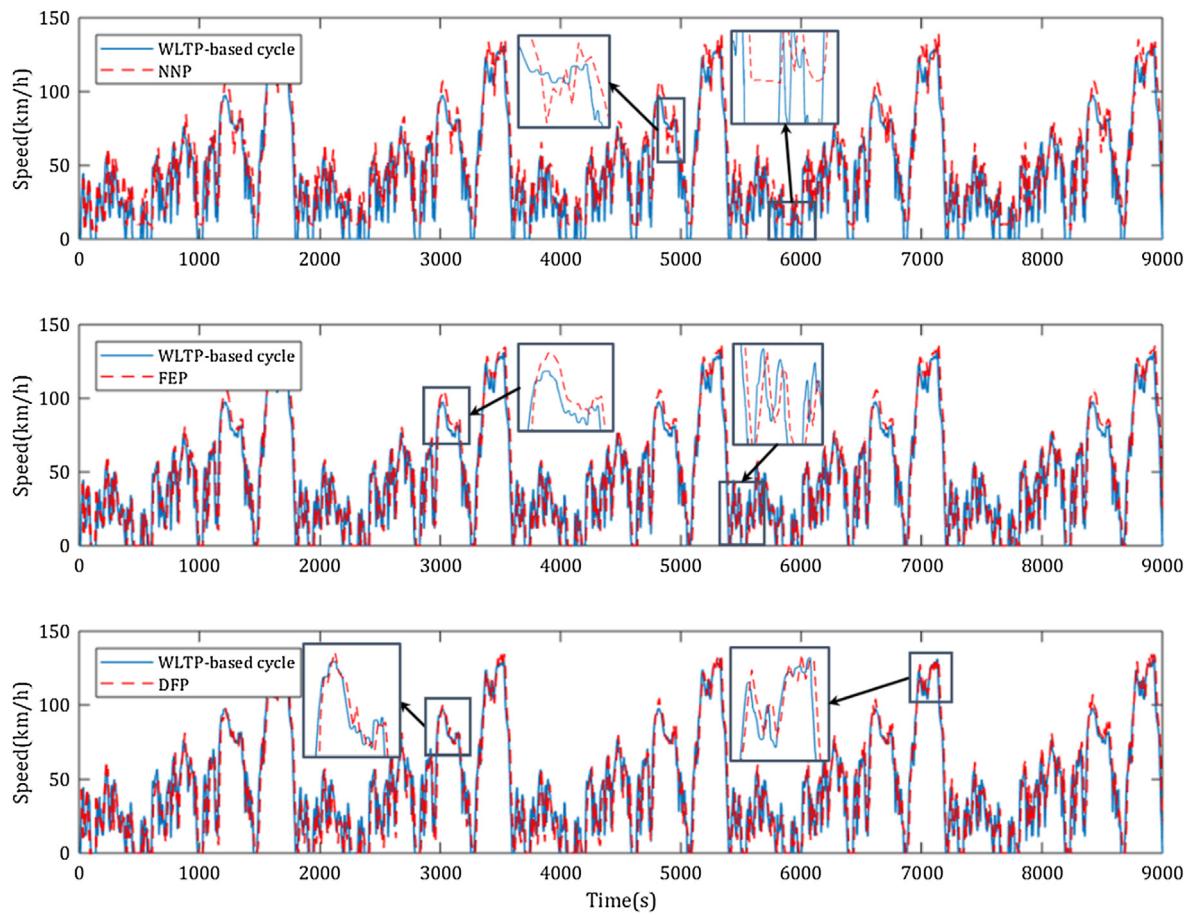


Fig. 8. Velocity prediction result of three MC-based predictors.

Table 3
Velocity prediction comparison of three MC-based predictors.

Predictor	Driving behaviour recognition	Maximum error	ITAE (10^5)	Reduce (%)
NNP	NA	14.81%	3.3412	–
FEP	NA	4.81%	2.7074	18.97%
DFP	Yes	2.22%	2.4237	27.46%

predict model and optimizing control sequence, which can obtain a real-time optimal control signal for energy flow distribution. Inspired by fuzzy granulation technology, the DOIP algorithm improves the precision of future velocity by reconsidering driving behaviours for each look-ahead step to guarantee the effectiveness of the optimal control sequence solved by CAPSO algorithm. The DOIP workflow (at k sampling time) is shown in Fig. 4.

In the prediction process of the DOIP algorithm, a recording matrix and its mirror matrix are involved and used as short-term length moving windows for driving data collection. Although the size of the training data is limited by length of moving windows, the DOIP algorithm will continuously motivate the training data to keep fresh all the time. It can be described as given by:

$$\mathbf{R} = \mathbf{R}_{\text{mirror}} = [\gamma_k, \gamma_{k-1}, \dots, \gamma_{k-5000}]^T \quad (26)$$

where γ_k indicates data vector of the moving memory horizon at sampling time k ; the update frequency of recording matrix \mathbf{R} is 5 s; the update frequency of mirror matrix $\mathbf{R}_{\text{mirror}}$ is consistent with the communication frequency of the real-time PHEV system (0.1 s).

Between two updates of the record matrix \mathbf{R} , the mirror matrix $\mathbf{R}_{\text{mirror}}$ will participate in the iterative calculation of multi-step

prediction and its procedure is as follows. Based on Eq. (22), the new data vector γ_{k+1} related to the next one-step-ahead velocity $v(k+1)$ is calculated and used to squeeze out the last data vector γ_{k-5000} in the mirror matrix. Therefore, the mirror matrix $\mathbf{R}_{\text{mirror}}$ is replaced as given by

$$\mathbf{R}'_{\text{mirror}} = [\gamma_{k+1}, \gamma_k, \dots, \gamma_{k-4999}]^T \quad (27)$$

Based on the new mirror matrix $\mathbf{R}'_{\text{mirror}}$, the membership criterion of the next one-step-ahead $u_n(\gamma_{k+1})$ is calculated as the weighted sum coefficient to aggregate the next two-step-ahead velocity from predict results of five driver-oriented Markov models. The next-two-step velocity $v(k+2)$ can be calculated as given by

$$\begin{cases} a(k+2) = \sum_{n=1}^B (O_n(v(k+1)))^T \prod_n \bar{c}_n \cdot \omega_n(v(k+1)) \\ v(k+2) = v(k+1) + a(k+2) \end{cases} \quad (28)$$

After iterative calculation, the future horizon velocity $v(k+1), v(k+2), \dots, v(k+100)$ is obtained, then used in the second iteration loop for optimizing the control sequence via the CAPSO algorithm. Finally, the first fifty elements of the control sequence $\mathbf{u}(k+1), \mathbf{u}(k+2), \dots, \mathbf{u}(k+100)$ will be sent back to the powertrain of the PHEV system for real-time energy management. Once the next updating trigger comes, the recording matrix \mathbf{R} and the mirror matrix $\mathbf{R}_{\text{mirror}}$ will both be updated following which the driver-oriented MC models \prod_n will be re-learned respectively.

4. Testing and validation set-up

4.1. Real-world driving cycles

In this paper, experimental studies are conducted on the cockpit

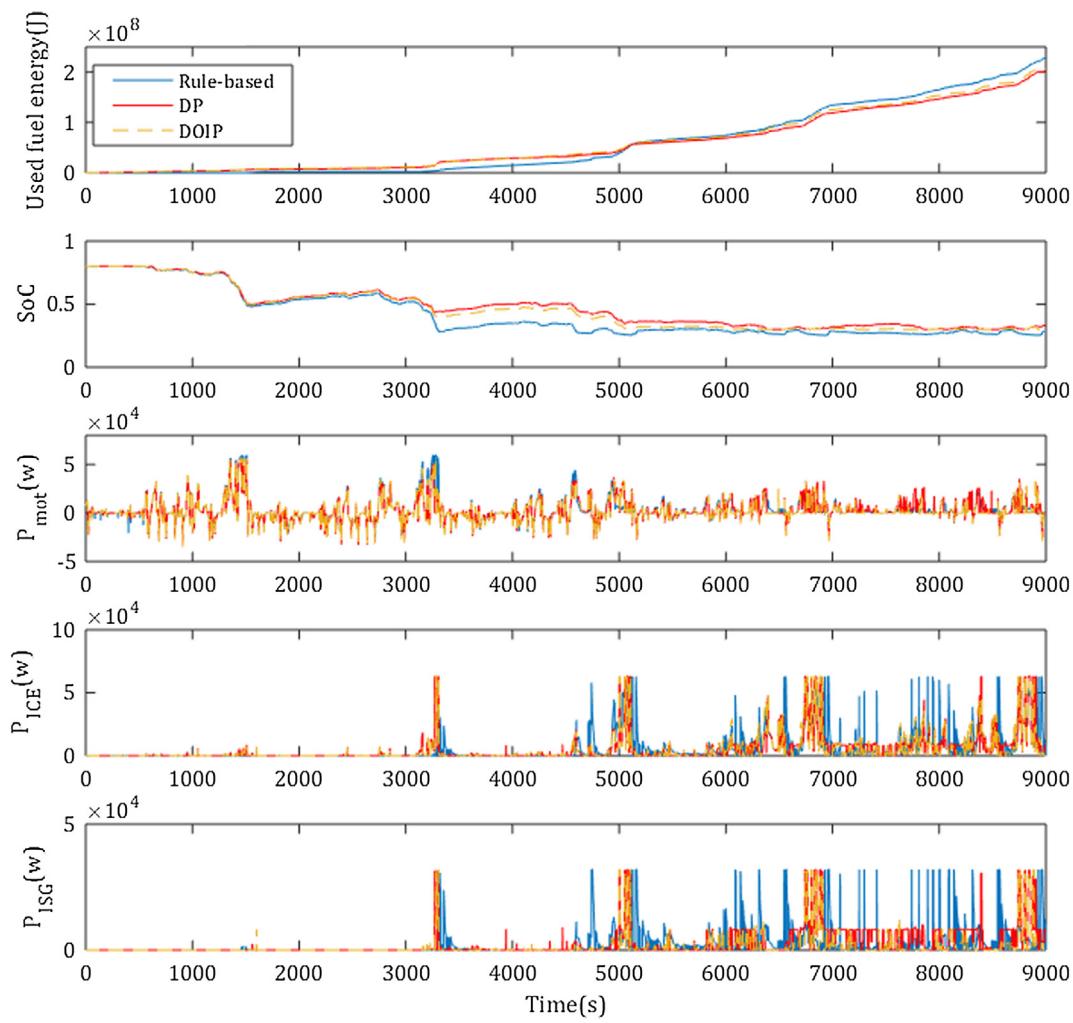


Fig. 9. Vehicle system performance comparison when initial battery SoC = 0.8.

package with the same PHEV model, in which five human drivers are invited as experimental objects to participate in 8000 s real-world driving. The road map model used with traffic is a mixture of highway and local roads with multiple stop signs, traffic lights, and speed limit changes provided by IPG CarMaker as Fig. 5. The human driver is required to follow the speed limits, stop signs, traffic lights, and other traffic regulations. It should be noted that the driver's pedal behaviour may be dependent on the vehicle, the pedal to its torque map, and even the physical pedal resistance feedback. This dependence is not studied in this research. The proposed model in this paper is for generating/learning the drivers' behaviour for a given vehicle. The specification of the real-world driving cycle is shown in Table 2.

4.2. Driving simulation platform

A static system experience platform driving simulator is involved in this research as shown in Fig. 6, which is the ideal tool for subjectively testing vehicle functions through direct experience. It makes the most of the advantages offered by the combination of a detailed and realistic human machine interface simulation and a real-world driving experience, coupled with the CarMaker open integration and test platform [39]. There is one cockpit package supported by a Thrustmaster T500RS and one host PC with I5-6500 3.2 GHz processor and 8 GB RAM. Their communication relies on a 3.0 USB cable, in which sampling frequency of the PHEV system and pedal data acquisition are both 10 Hz. All online programming is encoded as Matlab m-code and it operates on the PC host to obtain an optimal control signal for the PHEV

system in real time, in which m-code can be recognized directly under the CarMaker simulation environment.

5. Results and discussion

5.1. Velocity prediction comparison

Existing MC-based predictors including the nearest neighbour predictor (NNP) [40] and the fuzzy encoding predictor (FEP) i.e. the interval fuzzy predictor [14] is considered in this paper and compared with the proposed DFP. As the unique ability, the DFP can efficiently differentiate driving behaviours at each driving state using fuzzy C-mean clustering algorithm. In Fig. 7, it can be seen that five-level driving behaviours are clearly discriminated through deep fuzzy granulation, in which the driver with more aggressive actions have more wide operation range. Especially at deceleration range [-6, -3], the DFP differentiates the operation border of each driving behaviour.

Fig. 8 presents 5-s velocity prediction result of three predictors on WLTP-based cycles. In fact, it is very hard for the NNP to make a good prediction in the low-speed area because the transition probability of this area is very small by using one discrete MC model. Through fuzzy granulation, the FEP fixes the problem from the low-speed area but to treat different driving habits in a unified way makes its prediction performance in the medium-high-speed area still unsatisfied. Compared with both MC-based predictors of the NNP and the FEP, it is apparent that the DFP can achieve more excellent accuracy because the training database of the predictive model is continuously updating during real-

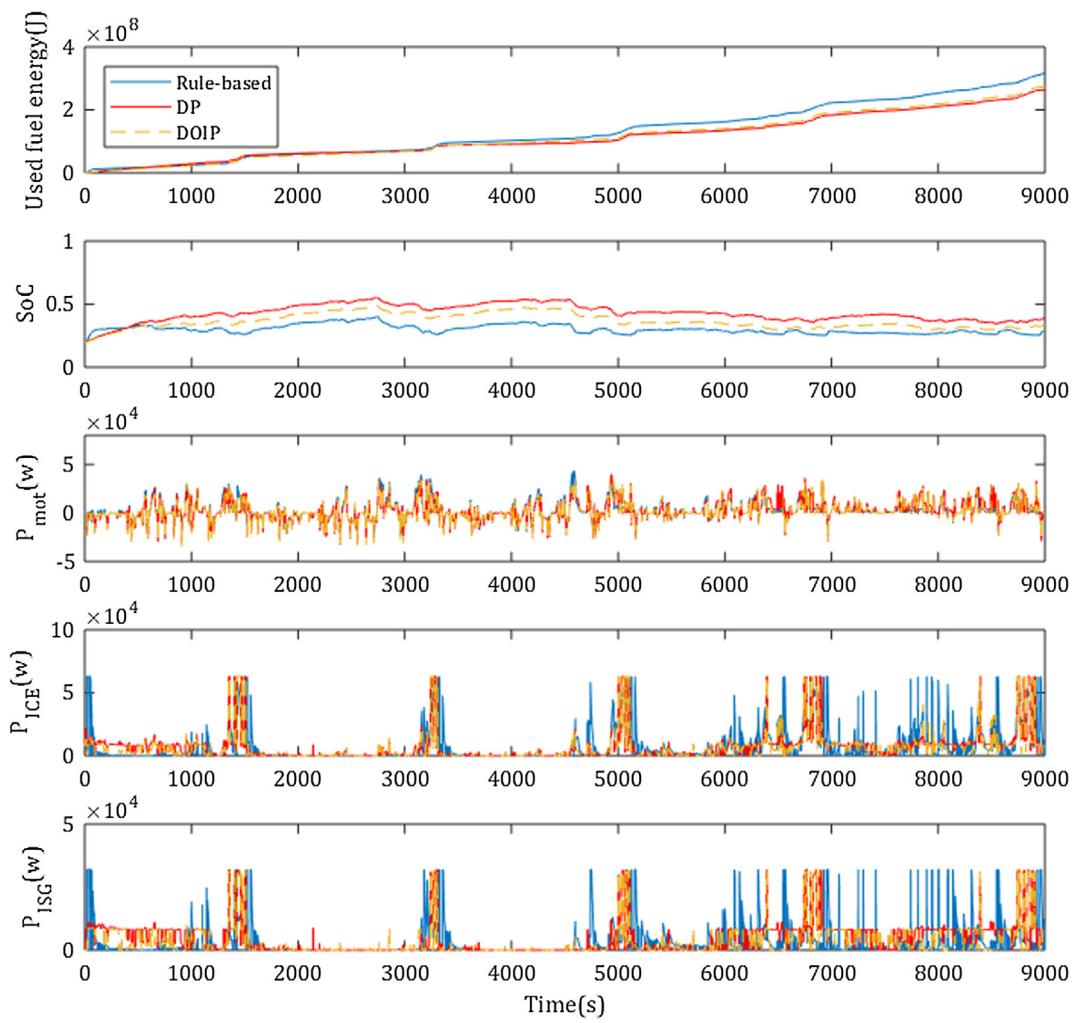


Fig. 10. Vehicle system performance comparison when initial battery SoC = 0.2.

Table 4
Vehicle system performance over WLTP-based driving.

Optimization strategy	Initial SoC	Final SoC	Used fuel energy (10^8 J)	Savings (%)
Rule-based	0.8	0.2893	2.2730	–
DP	0.8	0.3339	2.0146	11.37%
DOIP	0.8	0.3093	2.0644	9.18%
Rule-based	0.2	0.2893	3.0951	–
DP	0.2	0.3891	2.6383	14.76%
DOIP	0.2	0.3348	2.7300	11.80%

world driving. The prediction model can realize more personalized prediction in time-series through driver-oriented continuous adjusting among 5-layer Markov-chain models. The ITAE in DFP (2.4237×10^5) is less than that in NNP (3.3412×10^5) and FEP (2.7074×10^5), in which the maximum error in DFP decreases 2.59% compared to that in FEP. More comparison detail is shown in Table 3.

5.2. Performance over cycle-based driving

Here, the DOIP-based online control strategy using the DFP (10-second look ahead) is further compared with DP-based one and a rule-based control strategy over the WLTP-based driving cycle. Fig. 9 illustrates the vehicle system performance comparison when initial SoC = 0.8. It can be discerned that the power split trajectory in the DOIP-based online control strategy is close to that of DP-based control

strategy and clearly differs from the rule-based control. Especially during the hybrid mode, the rule-based control strategy more often let the engine work at maximum power. Relatively, the DP and DOIP based control strategies let engine keep working in a high-efficiency region for fuel saving.

An analogous result also can be observed in Fig. 10 when initial SoC = 0.2. Obviously, the fuel consumption under the DOIP-based online control strategy is closest to that of the DP-based control, 9.18% and 11.8% energy-saving from the rule-based control strategy at SoC = 0.8 and SoC = 0.2. Compared to the rule-based control strategy, the DP and DOIP based them have a higher SoC level during the entire journey as backup energy for the potential high power requirements. Compared with when the initial battery SoC = 0.2, energy-saving performance of productive control strategies is more significant when the initial battery SoC = 0.8. The vehicle performance with different control strategies is summarized in Table 4.

5.3. Performance over real-world driving

All the experiments here were conducted on the driving simulation platform. Fig. 11 shows the DiL experiment result operated by five human drivers in simulation driving scenarios, where each driver's independently driven section is separated by a black dotted line. After 600-second initialization of the recording matrix, the DFP starts to produce 10 s look-ahead horizon and its prediction models are real-time updating per five seconds. The predicted velocity feature relies on last-one-step driving behaviour not related to the driver's change. This

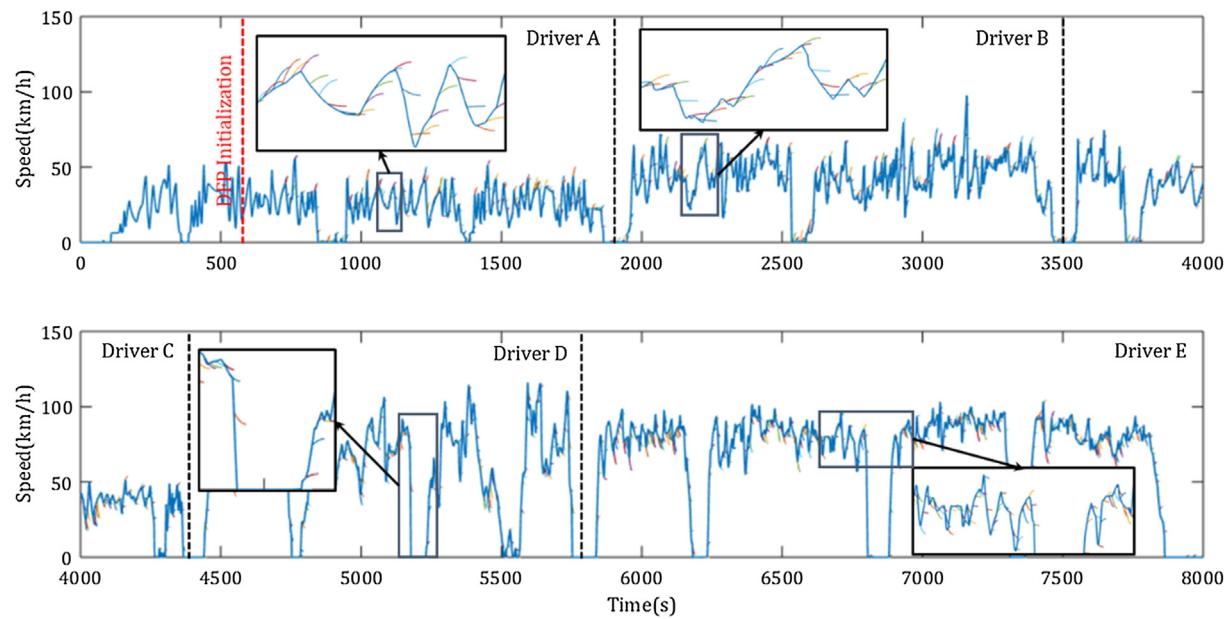


Fig. 11. Online prediction results over real-world driving.

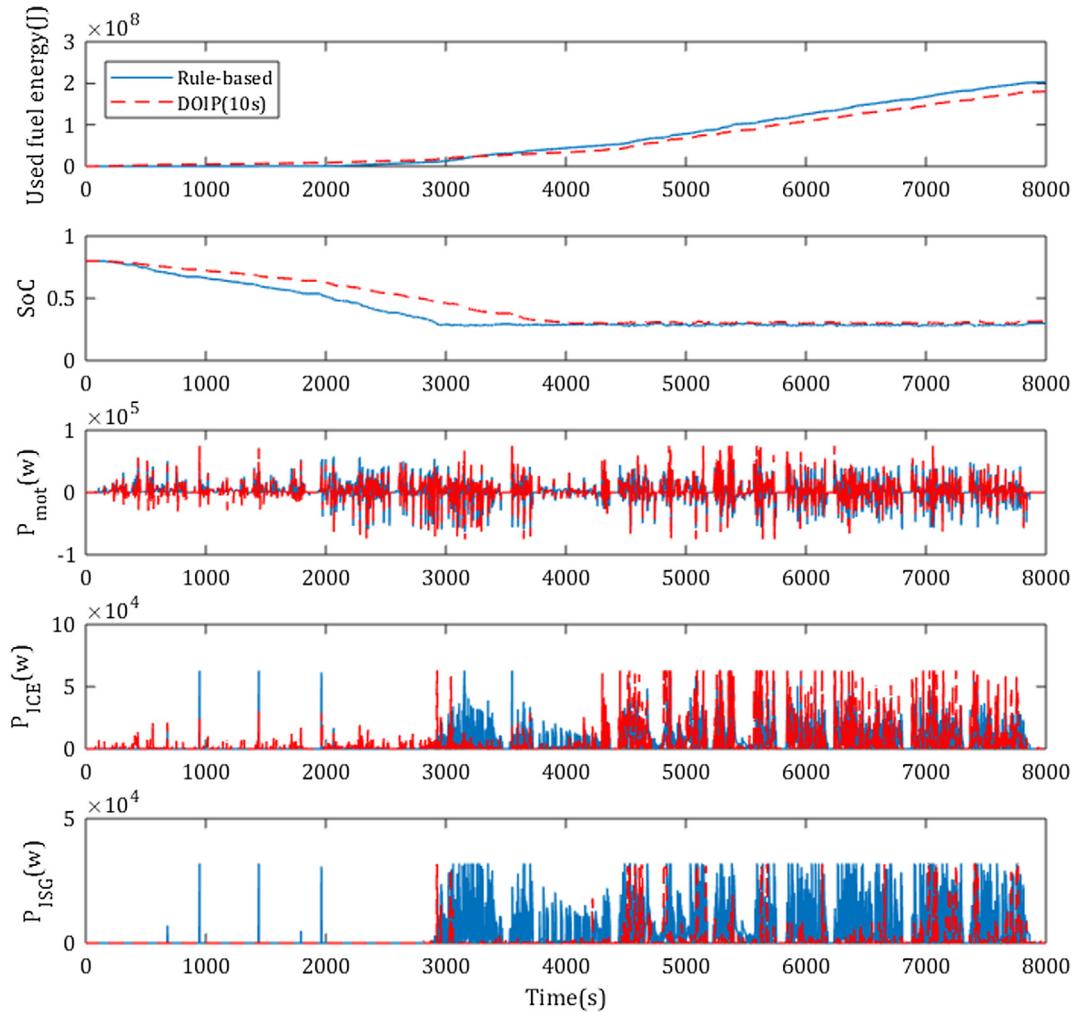


Fig. 12. Vehicle performance over real-world driving when initial battery SoC = 0.8.

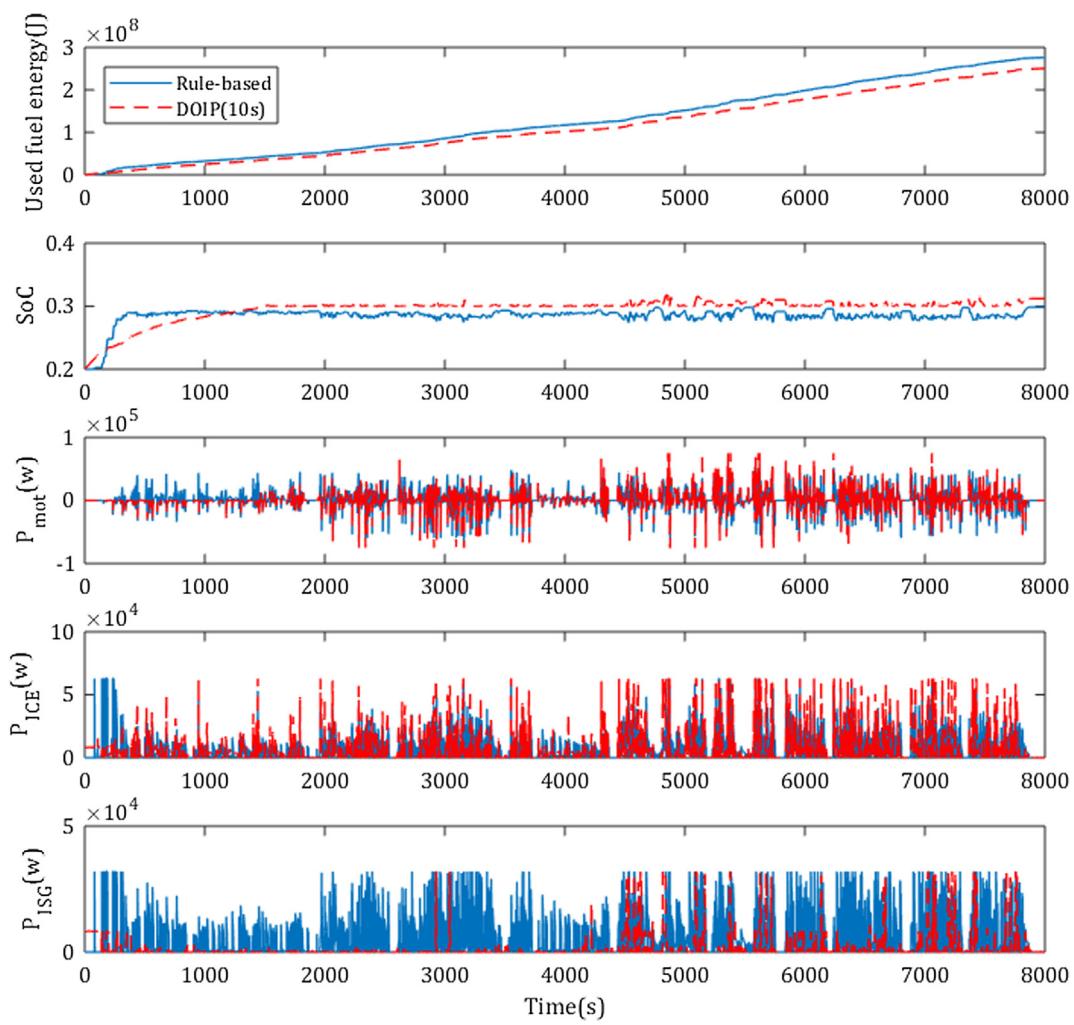


Fig. 13. Vehicle performance over real-world driving when initial battery SoC = 0.2.

Table 5
Online performance comparison over real world driving.

Optimization strategy	Initial SoC	Final SoC	Used fuel energy (10^8 J)	Saving (%)
Rule-based	0.8	0.2985	2.0217	–
DOIP	0.8	0.3144	1.8668	7.66%
Rule-based	0.2	0.2985	2.7638	–
DOIP	0.2	0.3123	2.5049	9.37%

Table 6
Computational time in different look-ahead horizon length.

Optimization strategy	Look-ahead horizon	Time (per second)	Relative increase (%)
DOIP	5 s	0.54 s	–
DOIP	10 s	0.67 s	24.07%
DOIP	20 s	0.79 s	46.30%

means that if the driving behaviour of a single driver has changed dramatically, its predicted velocity will be adjusted adaptively according to a new pedal action (shown in magnified views). It is emphasized that the recording matrix of the DFP will be completely replaced within 600 s so that to relearn a new driver behaviour takes up to 10 min whenever the driver changes.

Fig. 12 indicates when BP's SoC is high, the DOIP-based control

strategy allows the engine to compensate for the total power demand for low power. Fig. 13 indicates when BP's SoC is low, this control strategy will give the engine a priority in compensating the total power demand compared to the generator to compensate the BP. Compared with the rule-based strategy, Up to 9.37% total energy can be saved while maintain the higher SoC value by the DOIP-based control strategy. During the real-world driving, the DOIP algorithm breaks the bondage of conventional rule-based strategy and freely explore the more efficient way for the PHEV's energy-split. Due to no prior information on the predicted vehicle's performance or the driver's preference, global optimal algorithms i.e. the DP-based control strategy is no longer suitable for online optimization of the HEV energy management. More specification comparison over real-world driving is reported in Table 5.

The computational time of the DOIP algorithm in the DiL experiment is investigated and contrasted in Table 6. The solving speed of the DOIP algorithm is affected by look-ahead horizon length and computational efficiency of the processor, where the latter will not be discussed in this paper. As an increase of look-ahead horizon length, iterative calculations in this algorithm increase and its computational time appears a linear upward trend. When the look-ahead horizon is 20 s, the computational time increases to 0.79 s per second but computing resources still have a surplus depending on its concise solving frame. As the rapid development of computer science, it is feasible to operate the DOIP algorithm on the actual on-board controller of HEVs for real-time energy saving.

6. Conclusions

This paper develops an online predictive control strategy using the proposed DOIP algorithm for series-parallel PHEVs. Its prediction performance is demonstrated and compared with existing MC-based predictors. Cycle-based driving (WLTP) and real world driving are conducted on the driving simulator experimental platform for vehicle performance validation. The conclusions drawn from the investigation are as follows:

1. The proposed DFP has the ability to differentiate driving behaviours at each driving state in real time. Its prediction result shows excellent accuracy with the lowest maximum error (2.22%), compared with the NNP (14.81%) and the FEP (4.81%).
2. From the cycle-based driving results, energy management efficiency of the DOIP-based control strategy is close to DP-based control strategy. It is clearly superior to the rule-based one over the WLTP-based driving cycle with up to 11.80% reduction in fuel consumption.
3. During real world driving, up to 9.37% total energy can be saved compared with the rule-based control strategy. The DP-based control strategy cannot work in this online environment.
4. Whether cycle-based driving or real world driving scenarios, the energy-saving performance of the DOIP-based control strategy is more outstanding when the initial SoC is low (SoC = 0.2).
5. The computational time of the DOIP algorithm is investigated. It is feasible to operate the DOIP algorithm for 20 s look-ahead horizon and computing resources still have a surplus.

References

- [1] Trovão JPF, Santos VDN, Antunes CH, Pereirinha PG, Jorge HM. A real-time energy management architecture for multisource electric vehicles. *IEEE Trans Ind Electron* 2015;62:3223–33. <https://doi.org/10.1109/TIE.2014.2376883>.
- [2] Hu X, Jiang J, Member S. Advanced power-source integration in hybrid electric vehicles: multicriteria optimization approach. *IEEE Trans Ind Electron* 2015;62:7847–58. <https://doi.org/10.1109/TIE.2015.2463770>.
- [3] 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>.
- [4] Stillwater T, Kurani KS, Mokhtarian PL. The combined effects of driver attitudes and in-vehicle feedback on fuel economy. *Transp Res Part D* 2017;52:277–88. <https://doi.org/10.1016/j.trd.2017.02.013>.
- [5] Stewart A, Hope-Morley A, Mock P, Tietge U. Quantifying the impact of real-world driving on total CO₂ emissions from UK cars and vans. *Element Energy* 2015.
- [6] Huang Y, Wang H, Khajepour A, He H, Ji J. Model predictive control power management strategies for HEVs: a review. *J Power Sources* 2017;341:91–106. <https://doi.org/10.1016/j.jpowsour.2016.11.106>.
- [7] Sabri MFM, Danapalasingam KA, Rahmat MF. A review on hybrid electric vehicles architecture and energy management strategies. *Renew Sustain Energy Rev* 2016;53:1433–42. <https://doi.org/10.1016/j.rser.2015.09.036>.
- [8] Martinez CM, Hu X, Cao D, Velenis E, Gao B, Wellers M. Energy management in plug-in hybrid electric vehicles: recent progress and a connected vehicles perspective. *IEEE Trans Veh Technol* 2017;66:4534–49.
- [9] Wang F, Zheng N, Cao D, Martinez CM, Li L, Liu T. Parallel driving in CPSS: a unified approach for transport automation and vehicle intelligence. *IEEE/CAA J Autom Sin* 2017;4:577–87.
- [10] Sun C, Hu X, Moura SJ, Sun F. Velocity predictors for predictive energy management in hybrid electric vehicles. *IEEE Trans Control Syst Technol* 2015;23:1197–204. <https://doi.org/10.1109/TCST.2014.2359176>.
- [11] 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. *IEEE Trans Ind Inf* 2018;14:4149–58. <https://doi.org/10.1109/TII.2017.2781230>.
- [12] Li L, You S, Yang C, Yan B, Song J, Chen Z. Driving-behavior-aware stochastic model predictive control for plug-in hybrid electric buses. *Appl Energy* 2016;162:868–79. <https://doi.org/10.1016/j.apenergy.2015.10.152>.
- [13] Xie S, He H, Peng J. An energy management strategy based on stochastic model predictive control for plug-in hybrid electric buses. *Appl Energy* 2017;196:279–88. <https://doi.org/10.1016/j.apenergy.2016.12.112>.
- [14] Liu T, Hu X, Li SE, Cao D. Reinforcement learning optimized look-ahead energy management of a parallel hybrid electric vehicle. *IEEE/ASME Trans Mechatron* 2017;22:1497–507.
- [15] Zhang S, Xiong R. Adaptive energy management of a plug-in hybrid electric vehicle based on driving pattern recognition and dynamic programming. *Appl Energy* 2015;155:68–78. <https://doi.org/10.1016/j.apenergy.2015.06.003>.
- [16] Chen Z, Xiong R, Cao J. Particle swarm optimization-based optimal power management of plug-in hybrid electric vehicles considering uncertain driving conditions. *Energy* 2016;96:197–208. <https://doi.org/10.1016/j.energy.2015.12.071>.
- [17] Xiang C, Ding F, Wang W, He Y. Energy management of a dual-mode power-split hybrid electric vehicle based on velocity prediction and nonlinear model predictive control. *Appl Energy* 2017;189:640–53. <https://doi.org/10.1016/j.apenergy.2016.12.056>.
- [18] Zhang R, Tao J, Zhou H. Fuzzy optimal energy management for fuel cell and supercapacitor systems using neural network based driving pattern recognition. *IEEE Trans Fuzzy Syst* 2018;PP:1. <https://doi.org/10.1109/TFUZZ.2018.2856086>.
- [19] Unger J, Kozek M, Jakubek S. Nonlinear model predictive energy management controller with load and cycle prediction for non-road HEV. *Control Eng Pract* 2015;36:120–32. <https://doi.org/10.1016/j.conengprac.2014.12.001>.
- [20] Stach W, Kurgan LA, Pedrycz W. Numerical and linguistic prediction of time series with the use of fuzzy cognitive maps. *IEEE Trans Fuzzy Syst* 2008;16:61–72.
- [21] Jing J, Filev D, Kurt A, Özatay E, Michelini J, Özgüner Ü. Vehicle speed prediction using a cooperative method of fuzzy markov model and auto-regressive model. *Intelligent vehicles symposium (IV)*. IEEE; 2017. p. 881–6.
- [22] Vatanparvar K, Faezi S, Burago I, Levorato M, Abdullah M, Faruque A. Extended range electric vehicle with driving behavior estimation in energy management. *IEEE Trans Smart Grid* 2018;14:1–10. <https://doi.org/10.1109/TSG.2018.2815689>.
- [23] 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.
- [24] Liu T, Wang B, Yang C. Online Markov chain-based energy management for a hybrid tracked vehicle with speedy Q-learning. *Energy* 2018;160:544–55. <https://doi.org/10.1016/j.energy.2018.07.022>.
- [25] Lv C, Hu X, Sangiovanni-A, Li Y, Martinez CM, Cao D. Driving-style-based co-design optimization of an automated electric vehicle: a cyber-physical system approach. *IEEE Trans Ind Electron* 2018;1. <https://doi.org/10.1109/TIE.2018.2850031>.
- [26] Kennedy J. Particle swarm optimization. Encyclopedia of machine learning. Springer; n.d., p. 760–6.
- [27] Zhang Q, Deng W, Li G. Stochastic control of predictive power management for battery/supercapacitor hybrid energy storage systems of electric vehicles. *IEEE Trans Ind Inf* 2018;14:3023–30. <https://doi.org/10.1109/TII.2017.2766095>.
- [28] Chen Z, Mi CC, Xiong R, Xu J, You C. Energy management of a power-split plug-in hybrid electric vehicle based on genetic algorithm and quadratic programming. *J Power Sources* 2014;248:416–26. <https://doi.org/10.1016/j.jpowsour.2013.09.085>.
- [29] Zeng X, Wang J, Member S. A stochastic driver pedal behavior model incorporating road information. *IEEE Trans Hum-Mach Syst* 2017;47:614–24.
- [30] Li J, Li Z, Zhou Q, Zhang Y, Xu H. Improved scheme of membership function optimisation for fuzzy air-fuel ratio control of GDI engines. *IET Intel Transport Syst* 2018;13:209–17. <https://doi.org/10.1049/iet-its.2018.5013>.
- [31] Ehsani M, Gao Y, Longo S, Ebrahimi K. Modern electric, hybrid electric, and fuel cell vehicles. CRC Press; 2018.
- [32] Chen Z, Xiong R, Wang C, Cao J. An on-line predictive energy management strategy for plug-in hybrid electric vehicles to counter the uncertain prediction of the driving cycle. *Appl Energy* 2017;185:1663–72. <https://doi.org/10.1016/j.apenergy.2016.01.071>.
- [33] Marler RT, Arora JS. The weighted sum method for multi-objective optimization: new insights. *Struct Multidiscip Optim* 2010;41:853–62. <https://doi.org/10.1007/s00158-009-0460-7>.
- [34] Filev DP, Kolmanovsky I. Generalized Markov models for real-time modeling of continuous systems. *IEEE Trans Fuzzy Syst* 2014;22:983–98.
- [35] Johannesson L, Åsbögård M, Egardt B. Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming. *IEEE Trans Intell Transp Syst* 2007;8:71–83.
- [36] Akaike H. Information theory and an extension of the maximum likelihood principle. Selected papers of hirotugu akaike. Springer; 1998. p. 199–213.
- [37] Bezdek JC. Objective function clustering. Pattern recognition with fuzzy objective function algorithms. Springer; 1981. p. 43–93.
- [38] 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>.
- [39] IPG Carmaker. System experience platform. Karlsruhe (Germany): IPG Automotive; 2014<https://ipg-automotive.com/products-services/test-systems/driving-simulators/#system-experience-platform>.
- [40] Lin C, Peng H, Grizzle JW. A stochastic control strategy for hybrid electric vehicles. American control conference. IEEE; 2004. p. 4710–5.