Predictive Energy Management for Fuel Cell Hybrid Electric Vehicles

Yang Zhou*[a,b], Alexandre Ravey[b], Marie-Cécile Péra[b]

a. School of Automation, Northwestern Polytechnical University, Xi'an, China

b. FEMTO-ST (UMR CNRS 6174), *FCLAB* (USR CNRS 2007), *Univ. Bourgogne Franche-Comté, UTBM*

Rue Thierry Mieg, 90000, BELFORT, FRANCE.

Email : yang.zhou@utbm.fr ; alexandre.ravey@utbm.fr ; marie-cecile.pera@univ-fcomte.fr.

* *Corresponding Author*.

Abstract

Fuel cells are gradually becoming the competitive alternative to conventional internal combustion engines due to their high system efficiency and zero-local-emission property. Nevertheless, the high manufacturing cost and the limited lifetime of fuel cell systems still remain the major barrier towards the massive promotion of fuel cell electric vehicles. To reduce the vehicle's operating cost, reliable energy management strategies should be devised to coordinate the outputs of multiple energy sources in hybrid powertrain.

This chapter intends to present the development of predictive energy management strategy for fuel cell hybrid electric vehicles, especially focusing on the possibility of combining the driving predictive information with the real-time optimization framework. To this end, two driving prediction techniques are proposed, namely a vehicle speed forecasting approach and a driving pattern recognition method. Thereafter, model predictive control is adopted for real-time decision-making with the assistance of the predicted information. Validation results indicate that the proposed control strategy outperforms the benchmark control strategies in terms of fuel economy and fuel cell durability, thereby verifying the control performance improvement imposed by driving prediction integration.

Key words: Fuel Cell Hybrid Electric Vehicle; Energy Management Strategy; Driving Prediction Techniques; Model Predictive Control; Real-time Optimization

1. Introduction

With the global warming and the depletion of fossil fuels, fuel cell technologies have gained increasingly growing attentions in transportation electrification field [1]. Fuel cell hybrid electric vehicles (FCHEVs) are associated with many appealing properties, like high-efficiency, zero-emission and rapid-refueling, which make them the competitive substitutions to internal combustion engine-based hybrid electric vehicles (HEVs) [2]. Nevertheless, vehicle's high ownership cost and the limited lifespan of fuel cell systems greatly hinder the massive commercialization of FCHEVs. To reduce the FCHEV's operating cost for better economic performance, one of the practical solutions at the current stage is to develop and implement the reliable energy management strategies (EMSs) [3].

EMSs are vehicular system-level control strategies dedicated to coordinating the outputs of multiple energy sources within the hybrid powertrain, so as to meet the requested propulsion power demand. Existing EMSs for FCHEVs can be roughly categorized into rule-based and optimization-based strategies. Rule-based strategies constitute of a series of predefined deterministic (or fuzzy) rules. These rules for power allocation are largely designed based on human intuition, engineering experience or expertise knowledge [4]. Generally, the advantages of rule-based strategies lie in their simplicity, real-time suitability and robustness against discrepancies in driving cycles. However, their performance optimality (i.e. fuel economy) cannot be guaranteed by the calibrated parameters and preset rules, which is the major deficiency of rule-based strategies. In contrary, global optimization-based strategies derive the optimal power-distributing effect based on the complete driving cycle knowledge *a priori* [5], which can dramatically improve the performance optimality versus rule-based strategies. Nevertheless, such optimal control effect cannot be directly applied to real-time scenarios due to the requirement on the entire driving cycle information beforehand.

Alternatively, to optimally distribute power demand in real-time, a diversity of control algorithms has been successfully applied to vehicular energy management problems, wherein model predictive control (MPC) is one of the most representative approaches [6]. Specifically, MPC anticipates the upcoming vehicle's behavior over the finite time horizon and takes control actions accordingly. With the advancement in onboard electronic control units and real-time optimization techniques, MPC-based strategies can be adopted for real-time power-allocation. Nevertheless, vehicle's future behaviors could be heavily affected by a variety of unpredictable traffic factors, like the stochastic distribution of traffic lights and the unexpected movement of pedestrians [3]. Such driving disturbance would differ the actual vehicle behaviors from the estimated ones, thus degrading the precision of predictive control framework. Therefore, how to properly characterize the future driving uncertainties has become a crucial aspect in terms of MPC performance enhancement [3].

Nowadays, the maturation of modern telematics systems and the advances in driving prediction techniques (DPTs) make it possible to acquire the previewed information about the vehicle's future

driving states, such as the traffic flow speed and the road slope profile. Benefiting from the previewed information, there would be more chances for predictive energy management strategies (PEMSs) to further enhance the vehicles' performance against traditional non-predictive EMSs. In this case, this chapter will especially focus on the development of PEMS for FCHEV, so as to underline the potential performance improvement imposed by driving prediction integration.

The rest of this chapter is organized as follows. Section 2 presents the development and validation of a layer recurrent neural network (LRNN) predictor. Moreover, to enhance the EMS's adaptability under multiple driving patterns, a Markov Chain based driving pattern recognizer is devised and verified in Section 3. With the driving prediction techniques and model predictive control, Section 4 presents a multi-mode predictive energy management strategy (PEMS) for FCHEV, whose functionality and real-time suitability are validated via Software-in-the-Loop (SIL) testing against the benchmark strategies. The major findings and future research directions are briefed in Section 5.

2. Data-driven speed-forecasting approach

Under the framework of model predictive control (MPC), the control actions are computed based on the estimation of future disturbances. In vehicular energy management field, vehicle's upcoming propulsion power requests are normally deemed as the system disturbances. Therefore, the prediction quality of vehicular future power demand would have a profound impact on the performance of MPC-based energy management strategies. Moreover, considering the dependency between the propulsion power request and vehicle speed, it is thus necessary to study how to precisely estimate the distribution of vehicle's upcoming speed profiles.

According to our previous study [3], data-driven approaches such as artificial neural network, Markov Chain, support vector machine are widely applied to forecast the vehicle's speed. As one of the representative methods, neural networks can learn predictive knowledge from historical dataset via training processes and reproduce the similar behaviors in future speed prediction tasks. However, conventional back propagation neural networks (BPNN) are suffering from two drawbacks, namely the slow convergence rate and the risk of being trapped into the local optima in training stage, which would eventually deteriorate the speed-forecasting performance [3]. To this end, we propose a novel speed-forecasting approach in this section, based on layer recurrent neural network (LRNN), with the purpose of enhancing the forecast precision with the assistance of an improved type of network structure.

2.1 Layer recurrent neural network speed predictor

LRNN is one type of recurrent neural network (RNN), which is a connectionist model including multiple self-connected hidden layers. The biggest advantage of the recurrent connection is that a "memory" of previous inputs remains in the network's internal state [3]. The structure of the proposed LRNN speed predictor is depicted in Fig. 1.

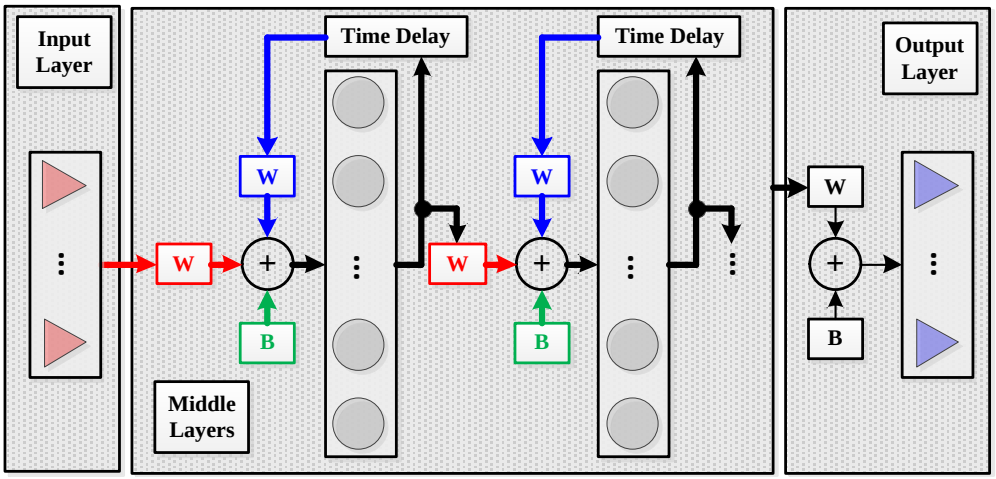

Fig. 1. Schematic diagram of the LRNN predictor.

As can be seen, the LRNN comprises an input layer, multiple middle layers and an output layer. The output of each middle layer is feedback to itself with a time delay. Such recurrent network structure helps the LRNN to store historical temporal information, thus better capturing the dynamics in a time-series. Mathematically, the functionality of LRNN is to receive the historical speed samples and to project the future ones, which can be expressed as follows:

$$\left[v_{k+1}^{*}, v_{k+2}^{*}, \ldots, v_{k+H_p}^{*}\right] = f_{\mathrm{LRNN}}\left(v_{k-H_q+1}, \ldots, v_{k-1}, v_k\right) \tag{1}$$

Where H_q and H_p are the length of input and output speed samples, respectively. As reported in [8] and [9], it is suggested to set the length of forecast horizon (H_p) to 1-10 seconds (with sampling time period being 1s) in vehicular energy management problems. Therefore, in this study, the upper limit for velocity prediction horizon is set to 10s, while the length of input speed sequence is set the same as the prediction horizon ($H_p = H_q$).

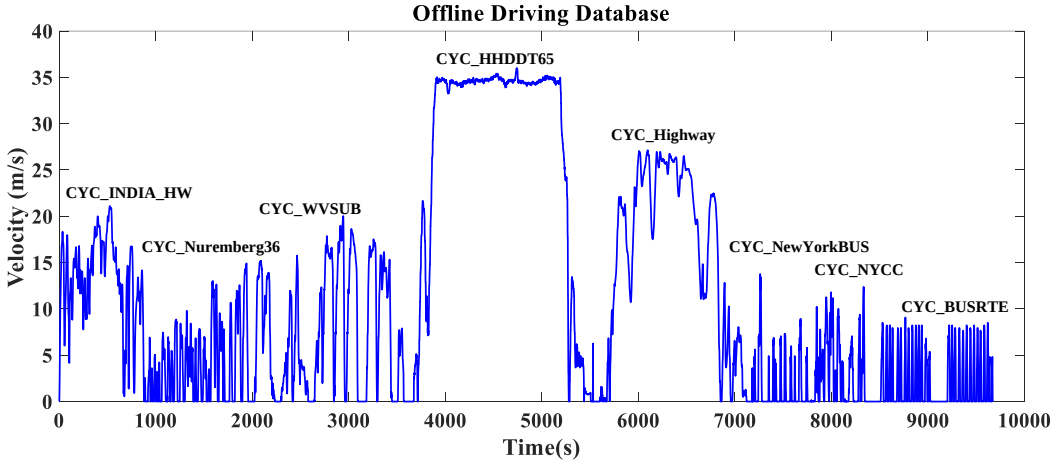

Fig. 2. Multiple standard driving cycles extracted from ADVISOR simulator.

The training of LRNN should be accomplished based on a comprehensive database. To guarantee the

satisfied forecast precision, this driving database should contain abundant driving cycles covering a wide range of driving scenarios. To this end, eight standard driving cycles with different driving patterns (urban/suburban/highway) are aggregated to establish the offline driving database, as shown in Fig. 2. Please note these standard driving cycles are taken from the vehicular simulator ADVISOR [10]. Thereafter, another standard driving cycle, Urban Dynamometer Driving Schedule (UDDS), is picked from ADVISOR to verify the performance of LRNN predictor. The root-mean-square-error (RMSE) is used as the evaluation metric for forecast precision [9].

2.1.1 Impact of percentage of training sample and middle layer configuration

Before online implementation, a sensitivity analysis is conducted to explore the impacts on forecast precision of LRNN caused by different percentage of network training samples and different node combinations in LRNN middle layer, so as to further improve the quality of speed prediction.

Firstly, to study the impact on prediction accuracy by different ratios of network training sample, the LRNN is trained with seven different percentage of driving data, and then the performance is tested under the UDDS driving cycle. In NN training phase, once the percentage x% of training sample is settled, the (1-x%) of driving data is used for validation accordingly, since we do not consider any testing sample (0%). The middle layer configuration of LRNN is {3,4,6}. The reason for using such configuration will be illustrated afterwards. Table 1 details the prediction results. As can be seen, with the increment of training ratio from 35% to 85%, the forecast accuracy of LRNN is improved when H_p = 3, 5 and 10s. This is mainly because, with a higher ratio of training sample, LRNN can learn predictive knowledge from a wider range of driving scenarios, thereby increasing its forecast precision in face of the newly-encountered driving conditions. Nevertheless, too much training sample (e.g. 95%) would degrade the prediction accuracy to some extent, since an over high ratio of training sample would compromise the generalization capacity of LRNN, thus reducing the forecast quality. As a result, the ratio of training sample is set to 85% since it can improve the prediction accuracy without over degrading the network generalization capacity.

Table 1. Average RMSE (km/h) under different training data percentage.

	35%	**45%**	**55%**	**65%**	**75%**	**85%**	**95%**
H_p = 3s	1.75	1.78	1.82	1.74	1.72	1.67	1.70
H_p = 5s	3.00	2.98	3.06	2.96	2.91	2.85	2.97
H_p =10s	6.31	6.31	6.43	6.29	6.20	6.09	6.28

Moreover, we keep using 85% of driving data (8227 out of 9479 speed samples) as the training sample for LRNN. Thereafter, by maintaining the three-middle-layer structure unchanged, the total number of middle nodes as a constant (e.g. in our case, 13), and altering the node numbers in the first two middle layers, LRNN predictor is tested under UDDS driving cycle, with the average prediction error (RMSE) listed in Table 2. As can be observed, when H_p = 3, 5 and 10s, the highest prediction accuracy is achieved

under the middle layer configuration III, namely {3,4,6}.

Table 2. Average RMSE (km/h) under different node combinations of LRNN middle layer

H_p	Config. I {1,6,6}	Config. II {2,5,6}	Config. III {3,4,6}	Config. IV {4,3,6}	Config. V {5,2,6}	Config. VI {6,1,6}
3s	2.66	1.81	1.67	2.08	3.04	2.89
5s	3.46	2.92	2.85	3.12	3.76	3.59
10s	6.72	6.21	6.09	6.22	6.75	6.70

Therefore, based on the results of sensitivity analysis, for LRNN predictor, 85% of data in offline driving database is used for network training while the remaining 15% is for performance validation, and the hidden layer node configuration is set to {3,4,6} for online implementation.

2.1.2 Benchmark speed predictor description

Two commonly-used speed predictors are introduced as the benchmark, namely a multi-step Markov Chain (MSMC) predictor and a back-propagation neural network (BPNN) predictor. MSMC predicts the vehicle's acceleration via the multi-step transition probability matrices (TPM), with the number of Markov state being 50 and the order of Markov Chain being 1. More details regarding the establishment of MSMC are available in [9]. In addition, the BPNN predictor has a three-layer structure, constituting of an input layer, a hidden layer and an output layer. It is a multi-input-multi-output mapping function, where the node number of BPNN hidden layer equals to the sum of nodes in LRNN middle layers, 13 in our case. More details regarding the elaboration of BPNN predictor are available in [9]. Furthermore, the estimation of TPM and the training of BPNN are accomplished using the driving data in Fig. 2.

2.1.3 Prediction results and discussions

Fig. 3 depicts the performance discrepancy of three predictors under UDDS testing cycle, where the blue and red curves respectively denote the real speed and the forecasted speed over each prediction horizon, where H_p = 10s and the sampling time interval $\Delta T = 1s$.

As shown in Fig. 3(a)-(c), due to the stochastic nature of Markov Chain, the predicted speed of MSMC tend to diverge heavily from the actual speed traces, thus causing the largest prediction error among three approaches. Besides, since the order of Markov Chain is set to one, the MSMC approach forecasts the upcoming speeds only relying on the current driving state, thus degrading its credibility in characterizing the complex and blended driving behaviors. Moreover, since more past speed samples are used for speed forecasting, the prediction quality of BPNN approach can thus be enhanced in contrast to MSMC method. In addition, thanks to the additional "memory" effect imposed by the recurrent network structure, the forecasted speed of LRNN distributes closer to the actual speed trajectories versus BPNN predictor, implying the improved forecast accuracy.

Furthermore, as underlined in the dashed regions in Fig. 3(d)-(f), the LRNN predictor shows an overall higher rate of re-convergence after the inflection points of the speed trajectory versus benchmark

predictors, indicating it can more promptly adapt to recent driving changes.

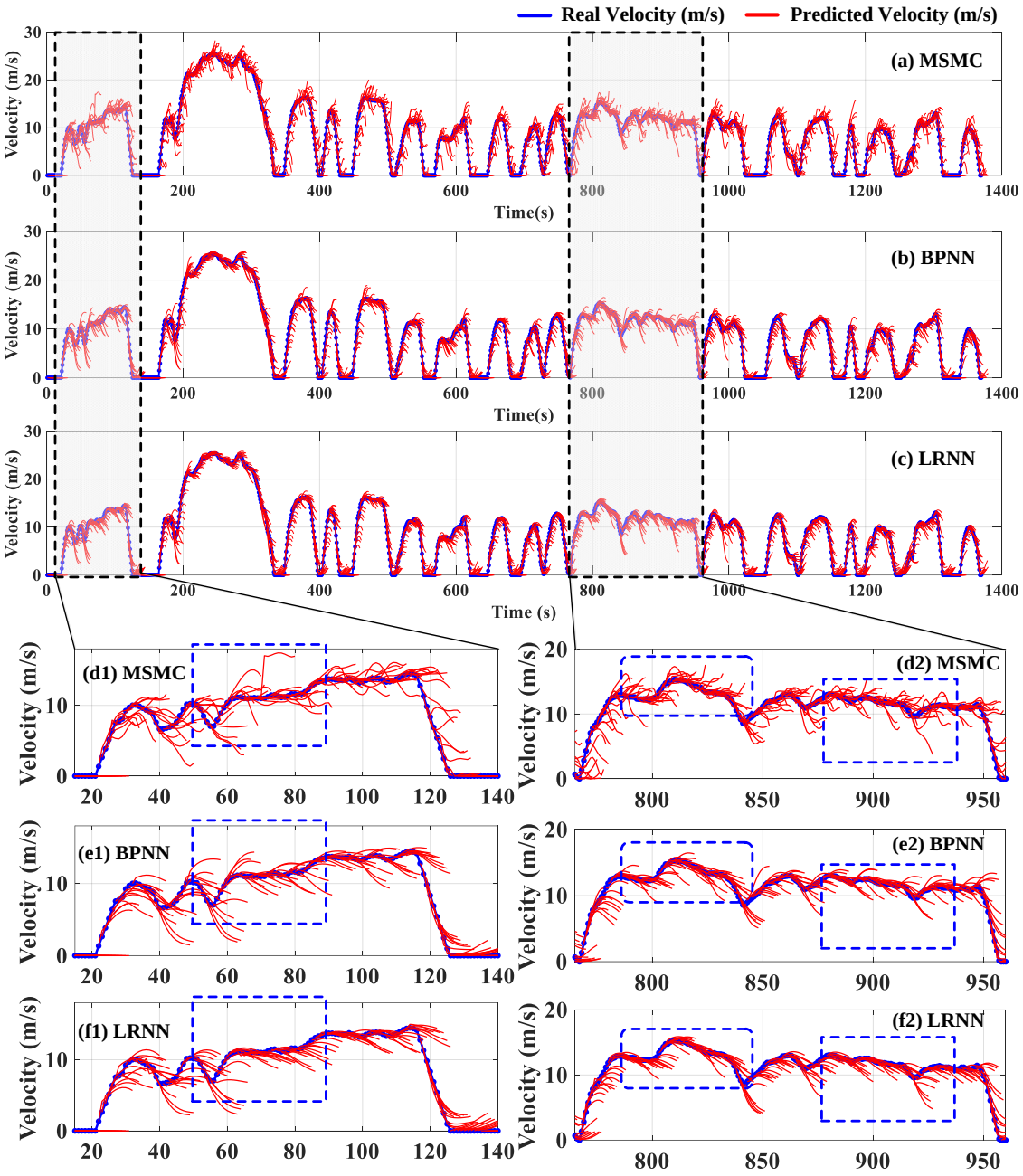

Fig. 3. Comparative speed forecasting results on UDDS driving cycle (H_p = 10): (a)-(c) global view of MSMC, BPNN and LRNN predictors; (d)-(f) Local view of MSMC, BPNN and LRNN predictors.

Table 3 lists the average RMSE ($\overline{\text{RMSE}}$) of three predictors with different H_p on UDDS testing cycle, where the percentage is the $\overline{\text{RMSE}}$ decrement brought by LRNN predictor. Specifically, in contrast to benchmark approaches, the proposed LRNN approach can respectively shrink the forecast error in average by at least 16.23% (MSMC) and 6.16% (BPNN), implying the improved forecast precision. Hence, the effectiveness of enhancing the prediction quality via using an improved type of network structure is verified.

Table 3. Average RMSE and prediction accuracy improvement on UDDS driving cycle

Hp		MSMC	BPNN	LRNN
3s	$\overline{\text{RMSE}}$ (km/h)	2.34	1.97	1.67
	Error reduction	**28.63%**	**15.23%**	N/A
5s	$\overline{\text{RMSE}}$ (km/h)	3.98	3.18	2.85
	Error reduction	**28.39%**	**10.38%**	N/A
10s	$\overline{\text{RMSE}}$ (km/h)	7.27	6.49	6.09
	Error reduction	**16.23%**	**6.16%**	N/A

3. Markov Chain based driving pattern recognizer

Driving pattern is an overall characterization of the combination of vehicle states and road environment [3], with urban (flowing or congested), suburban and highway being the representative patterns. Usually, the speed and the power demand profile under different classes of driving pattern exhibits different features. This would greatly challenge the adaptability and optimality of energy management strategies (EMS) under changeable driving scenarios. Yet, most of existing EMSs for FCHEVs still focuses on optimizing the power-splitting effects on specific driving cycles (e.g. [5]), which did not completely account for the impacts on control performance by different driving patterns.

In such context, a driving-pattern-conscious EMS with strengthened adaptability in changeable driving scenarios should be further investigated, which, in parallel, brings a challenging mission: driving pattern recognition (DPR). To address this issue, this section develops a DPR approach based on Markov Chain and moving window technique [11], which can effectively identify the pattern of the online driving fragment. The design process will be detailed in the following parts.

3.1 Working principle of the Markov Chain based DRP approach

In this section, the velocity-acceleration (v-a) transition behavior is deemed as the characteristic of each driving pattern, and it is quantified by the transition probability matrix (TPM) of Markov Chain. The workflow of the devised DPR method is illustrated in Fig. 4, including four working stages: (a) offline benchmark TPM building stage, (b) online TPM estimation stage, (c) resemblance quantification stage and (d) DPR precision enhancing stage, where stage (a) is accomplished offline, whereas others are fulfilled in real-time. The principle of each working stage will be detailed in the rest of this section.

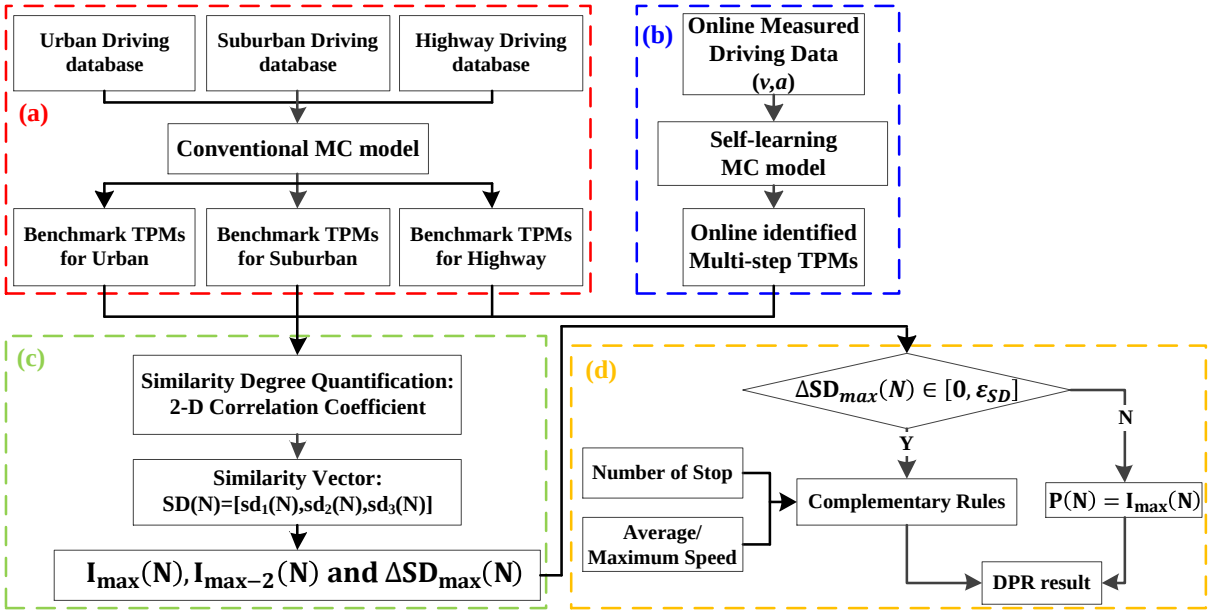

Fig. 4. Flowchart of the Markov driving pattern recognition (DPR) method.

3.2 Design of Markov-based driving pattern recognizer

As shown in Fig.4, conventional Markov model is leveraged to estimate offline-benchmark TPMs for three representative driving patterns, that is, urban, suburban and highway. In contrast, the TPMs related to recent driving changes are generated via the self-learning Markov model. Finally, the pattern recognition results can be attained through measuring the resemblance between offline-benchmark TPMs and online-recognized TPMs.

For DPR purpose, the Markov state is defined as the discrete (v-a) pair, respectively velocity and acceleration at the time k, denoted by $x(k) = \big(v(k), a(k)\big)$. Therefore, the $(i,j)^{th}$ element in the l-step TPM can be derived by [12]:

$$[T_l]_{ij} = \Pr\{x(k+l) = x_j | x(k) = x_i\} \approx \mathbf{Num}_{ij}^{l} / \mathbf{Num}_{oi}^{l} \tag{2}$$

$$\mathbf{Num}_{oi}^{l} = \sum_{j=1}^{s} \mathbf{Num}_{ij}^{l}, i, j \in \{1,2,\dots,s\}, l \in \{1,\dots,N_T\} \tag{3}$$

where $\mathbf{Num}_{ij}^{l}$ denotes the number of state transition event $x_i \to x_j$ in l steps ahead, $\mathbf{Num}_{oi}^{l}$ the state transition number originating from x_i , and N_T is the time-scale range of conventional Markov model.

To identify the TPM group via the real-time measurements, the number of state transition **Num** should be replaced by the frequency of state transition **Fre**. Hence, the original estimation model of transition probability (2) is reformulated as [12]:

$$[T_l(L)]_{ij} \approx \frac{\mathbf{Num}_{ij}^{l}(L)/L}{\mathbf{Num}_{oi}^{l}(L)/L} = \frac{\mathbf{Fre}_{ij}^{l}(L)}{\mathbf{Fre}_{oi}^{l}(L)} \tag{4a}$$

$$\mathbf{Fre}_{ij}^{l}(L) = \mathbf{Num}_{ij}^{l}(L)/L = \frac{1}{L}\sum_{t=1}^{L}\mathbf{flag}_{ij}^{l}(t) \tag{4b}$$

$$\mathbf{Fre}_{oi}^{l}(L) = \mathbf{Num}_{oi}^{l}(L)/L = \frac{1}{L}\sum_{t=1}^{L}\mathbf{flag}_{oi}^{l}(t) \tag{4c}$$

$$\mathbf{flag}_{oi}^{l}(t) = \sum_{j=1}^{s}\mathbf{flag}_{ij}^{l}(t) \tag{4d}$$

Where L is the length of observation. Besides, $\mathbf{flag}$ implies the occurrence of corresponding transition events, $i, j \in \{1, \dots, s\}$ and $l \in \{1, \dots, H_p\}$. For instance, $\mathbf{flag}_{ij}^{l}(t) = 1$ only when the state transition incident $a_i \to a_j$ occurs at time step t ($t \in [1, L]$), while $\mathbf{flag}_{oi}^{l}(t) = 1$ only when the state transition incident originates from state a_i at time step t. If the corresponding transition events do not occur, they both take zero values. Furthermore, the frequency of transition ($\mathbf{Fre}_{ij}^{l}$ and $\mathbf{Fre}_{oi}^{l}$) can be further expanded in the following way [12]:

$$\begin{aligned}\mathbf{Fre}_{ij}^{l}(L) &= \frac{1}{L}\sum_{t=1}^{L}\mathbf{flag}_{ij}^{l}(t) = \frac{1}{L}\cdot\left[(L-1)\mathbf{Fre}_{ij}^{l}(L-1) + \mathbf{flag}_{ij}^{l}(L)\right] \\ &= \mathbf{Fre}_{ij}^{l}(L-1) + \frac{1}{L}\cdot\left[\mathbf{flag}_{ij}^{l}(L) - \mathbf{Fre}_{ij}^{l}(L-1)\right] \\ &\approx \mathbf{Fre}_{ij}^{l}(L-1) + \boldsymbol{\varphi}\cdot\left[\mathbf{flag}_{ij}^{l}(L) - \mathbf{Fre}_{ij}^{l}(L-1)\right]\end{aligned} \tag{5a}$$

$$\begin{aligned}\mathbf{Fre}_{oi}^{l}(L) &= \frac{1}{L}\sum_{t=1}^{L}\mathbf{flag}_{oi}^{l}(t) = \frac{1}{L}\cdot\left[(L-1)\mathbf{Fre}_{oi}^{l}(L-1) + \mathbf{flag}_{oi}^{l}(L)\right] \\ &= \mathbf{Fre}_{oi}^{l}(L-1) + \frac{1}{L}\cdot\left[\mathbf{flag}_{oi}^{l}(L) - \mathbf{Fre}_{oi}^{l}(L-1)\right] \\ &\approx \mathbf{Fre}_{oi}^{l}(L-1) + \boldsymbol{\varphi}\cdot\left[\mathbf{flag}_{oi}^{l}(L) - \mathbf{Fre}_{oi}^{l}(L-1)\right]\end{aligned} \tag{5b}$$

To help TPMs adapt to recent driving changes, the varying decay coefficient $1/L$ is replaced by a constant forgetting coefficient $\boldsymbol{\varphi}$ ($0 < \boldsymbol{\varphi} < 1$) in (5a) and (5b), with the purpose of stepwise removing the impacts on transition probabilities by old measurements. A larger φ denotes a higher updating rate of TPM. In specific, all the measurements $\left[\mathbf{flag}_{ij}^{l}(1), \dots, \mathbf{flag}_{ij}^{l}(L)\right]$ and $\left[\mathbf{flag}_{oi}^{l}(1), \dots, \mathbf{flag}_{oi}^{l}(L)\right]$ are allocated with a group of exponentially declining weights $[\boldsymbol{\varphi}(1-\boldsymbol{\varphi})^{L-1}, \dots, \boldsymbol{\varphi}(1-\boldsymbol{\varphi}), \boldsymbol{\varphi}]$, wherein the sum of all weight elements is one. Thus, the probability $[T_l(L)]_{ij}$ can be updated online by [12]:

$$[T_l(L)]_{ij} \approx \frac{\mathbf{Fre}_{ij}^{l}(L-1) + \boldsymbol{\varphi}\cdot\left[\mathbf{flag}_{ij}^{l}(L) - \mathbf{Fre}_{ij}^{l}(L-1)\right]}{\mathbf{Fre}_{oi}^{l}(L-1) + \boldsymbol{\varphi}\cdot\left[\mathbf{flag}_{oi}^{l}(L) - \mathbf{Fre}_{oi}^{l}(L-1)\right]}, i, j \in \{1, \dots, s\}, l \in \{1, \dots, H_p\}. \tag{6}$$

By (6), the TPM can be gradually renewed based on the incrementally attained driving information.

3.2.1 Offline-benchmark transition probability matrix building stage

Fig. 5 gives the flowchart of offline-benchmark TPM building stage. In specific, the workflow of this stage is detailed as follows:

- **Step 1**: multiple standard driving cycles are selected from ADVISOR [10], including the Cruise3, HWFET, ARTEMIS_HW, HHDDT65, ARTEMIS_UB, US06_HW, BUSRTE, Manhattan, AQMDRTC2, NurembergR36, ARTEMIS_SUB, WVUINTER, WVUSUB, UNIF01 and IM240.

The driving cycles with identical patterns are aggregated to generate the related sub-database.

- **Step 2**: within each sub-database, collected speed profiles are discretized into related (v-a) pairs. Then, these time-index labelled driving samples are imported into the velocity-acceleration plane. The same Markov state indices are given to those samples which are distributed in the identical rectangle zone.
- **Step 3**: According to the measurements within the velocity-acceleration plane, the TPM groups of each driving pattern can be generated by (2). These TPM groups are deemed as the offline references for real-time resemblance quantification. Furthermore, as seen from the attained 3-D bar diagrams, every driving scenario has its own velocity-acceleration transition feature. Hence, the generated multi-step TPM groups can be adopted to represent corresponding driving scenarios.

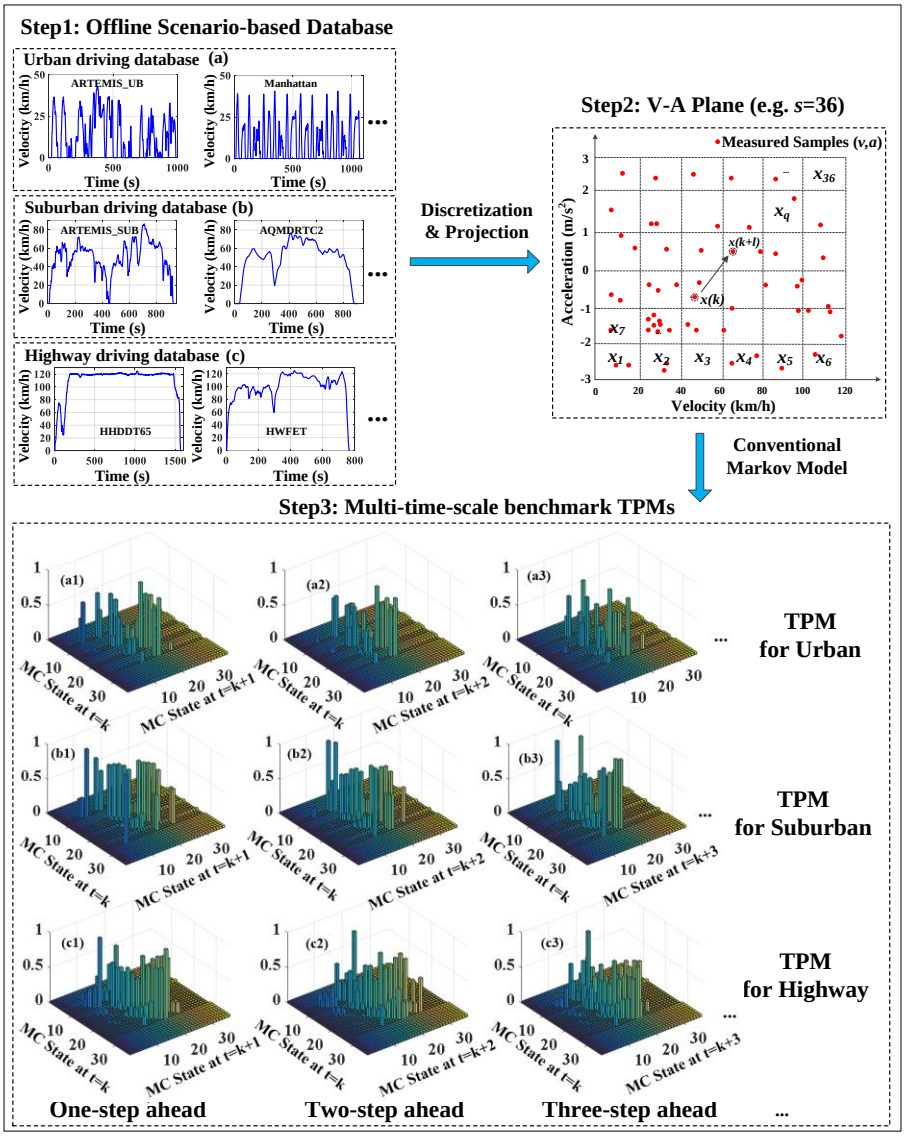

Fig. 5. Flowchart of offline scenario-based benchmark TPMs estimation phase (e.g. s = 36 and N_T = 3): Step 1. Establishment of the offline scenario-based driving database; Step 2. Discretion & projection speed samples into the V-A plane; Step 3. Estimation of offline benchmark TPM groups in different driving patterns.

3.2.2 Real-time transition probability matrix recognition stage

Fig. 6 presents the workflow of real-time TPM recognition stage. As shown in Fig.6, the online-learning Markov model (6) is applied to each driving fragment within the moving horizon. L_u and L_s represent the length of updating and sampling window, respectively.

Based on the (v-a) samples, the Markov transition probabilities are renewed at each sampling time instant, so as to evolve the real-time TPMs from the initial status to the terminal status. Thereby, at the end of each sampling phase, the similarity between the online-recognized TPMs and the offline-benchmark TPMs can be quantified. The quantification results keep unchanged within the whole updating phase (L_u seconds). Thereafter, to promptly remove the negative impacts imposed by old measurements, all the elements in the online-recognized TPMs are set to $1/s$ at every initialization moment (denoted by red line in Fig. 6). Hence, the quantification results are renewed every L_u seconds. More details regarding the TPM resemblance quantification will be introduced in subsection 3.2.3.

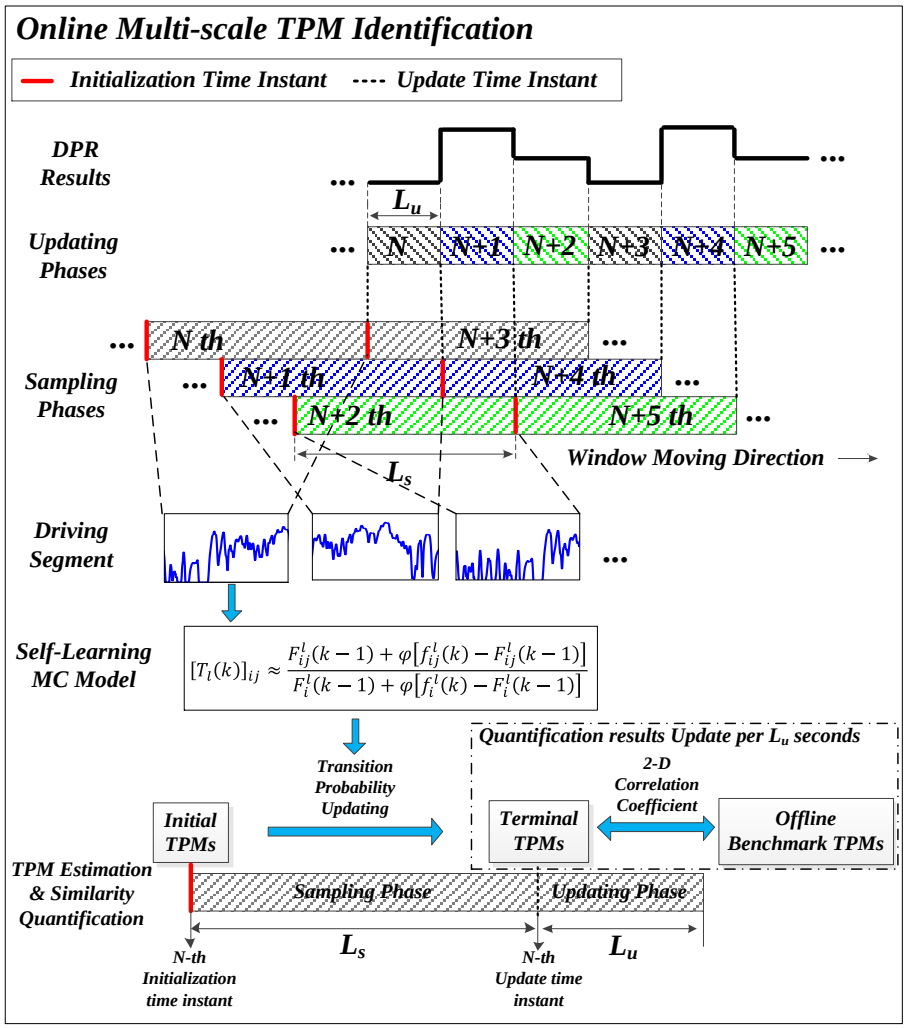

Fig. 6. Flowchart of online multi-scale TPM identification phase.

To guarantee the online-recognized TPMs fully reflecting the driving feature of current segment, the

Markov effective memory depth D_φ is set identical to the length of sampling window L_s. Obviously, a larger L_s covers a wider range of historical driving conditions. Yet, an overlarge L_s might include redundant information and also lead to heavier computational burden. As stated in [13], typically, the HEVs' driving period is approximately three minutes. In this case, as a rational tradeoff, L_s is set approximated to this threshold. In addition, the setting of updating window length L_u should guarantee the renewing rate of pattern recognition without frequent pattern switching. In light of these issues, L_u and L_s are set to 50s and 150s, respectively. It should be mentioned that the settings on L_u and L_s are attained via numerous cross-validations.

3.2.3 Quantification of similarity degree

The 2-D correlation factor $r \in [0,1]$ is adopted in this section to quantify the resemblance between the online-recognized TPMs and the offline-benchmark TPMs. Given two matrices $A, B \in R^{m\times n}$, $r(A,B)$ indicates the resemblance between them, as calculated by:

$$r(A,B)=\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}([A]_{i,j}-\bar{A})([B]_{i,j}-\bar{B})}{\sqrt{(\sum_{i=1}^{m}\sum_{j=1}^{n}([A]_{i,j}-\bar{A})^2)(\sum_{i=1}^{m}\sum_{j=1}^{n}([B]_{i,j}-\bar{B})^2)}} \tag{7}$$

where $[\cdot]_{i,j}$ represents the $(i,j)^{th}$ element of a matrix. $\bar{A}$ and $\bar{B}$ represent the average of elements in A and B, respectively. A larger $r(A,B)$ means a higher degree of similarity between the examined matrices. In addition, let N represent the index of updating window. Hence, at $t=k, N=fix(k/L_u)$, wherein $L_u = 50s$ and the function of fix is to output the integer part of k/L_u.

At the N^{th} updating time-step, the online-recognized TPMs, denoted by $T_l(N)$, are compared to the offline-benchmark TPMs, denoted by T_l^i, $l=1,2,\dots,N_T$, wherein i is the driving pattern index (1: urban, 2: suburban, 3: highway). Hence, the quantification results are expressed by a vector of similarity $SD(N)=[sd_1(N), sd_2(N), sd_3(N)]$, wherein $sd_i(N)\in[0,1], i=1,2,3$ measures the average similarity of the online-recognized TPMs versus each type of offline-benchmark TPMs, as denoted by:

$$sd_i(N)=\frac{1}{N_T}\sum_{l=1}^{N_T} r\left(T_l(N), T_l^i\right), i=1,2,3. \tag{8}$$

Besides, we define the discrepancy of the largest and the second largest element in $SD(N)$ by $\Delta SD_{max}(N)\in[0,1]$, the indices of the largest and the second largest element in $SD(N)$ by $I_{max}(N), I_{max-2}(N)\in\{1,2,3\}$, respectively, and the similarity threshold by $\varepsilon_{SD}\in(0,1)$. On this basis, two possible cases tend to occur within the N^{th} sampling horizon:

- **Case I ($\Delta SD_{max}(N) > \varepsilon_{SD}$):** such difference in resemblance is deemed adequate to split different driving patterns. Hence, the pattern identification results can be confidently derived based on the largest element in $SD(N)$, namely $P(N)=I_{max}(N)$. This case tends to occur if the (v-a) samples originate from single driving pattern, like the k^{th} and the r^{th} phases shown in Fig. 7(a).

- **Case II ($\Delta SD_{max}(N) \leq \varepsilon_{SD}$):** it is not persuasive to differentiate driving patterns based on such insignificant similarity differences. This case tends to occur within either the pattern-shifting phases (e.g. see the q^{th} phase of Fig. 7(a)) or the confusion phases (e.g. see the s^{th} phase of Fig. 7(a)).

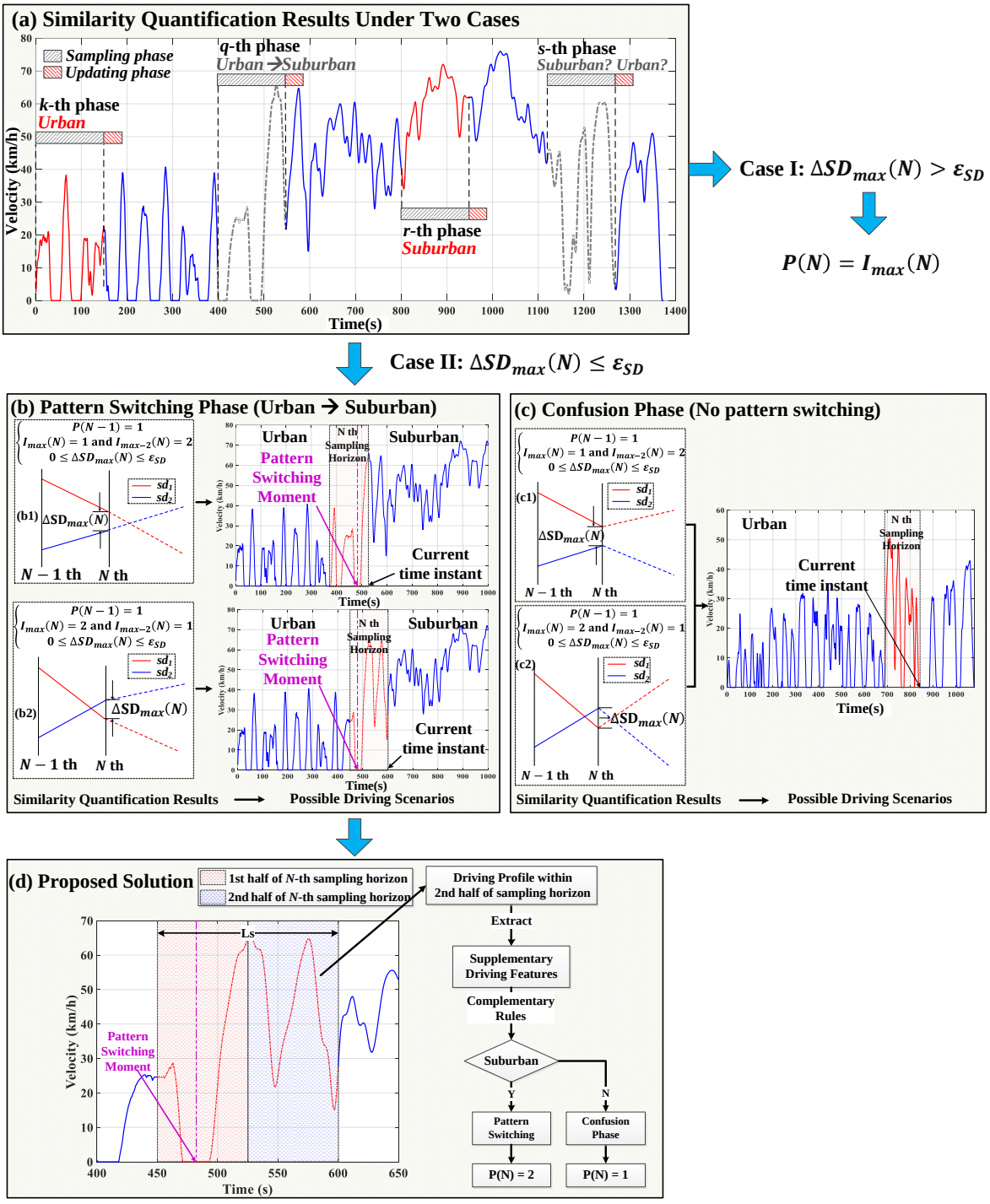

Fig. 7. Representation of (a) similarity quantification results in different driving scenarios; (b) real driving pattern-switching phases (e.g. urban to suburban); (c) confusion phases (e.g. urban vs. suburban); (d) proposed solution to separate pattern switching phases from confusion phases (e.g. urban vs suburban).

The similarity threshold ε_{SD} should be set as a balance between the overall pattern recognition accuracy

and the sensitivity towards the (v-a) transitions. In specific, the driving pattern updating rate might be restricted by a larger ε_{SD}, thus degrading the real-time suitability under rapid-changing driving conditions. In contrary, if an over-small ε_{SD} were used, the reliability of DPR would be reduced by the frequent pattern switching results. As a result, ε_{SD} is set to 0.05 after a large number of trials and errors.

The TPM resemblance quantification results in **Case II** make it impossible to discriminate two conflict patterns, and, thus, additional rules are required to enhance the precision of pattern recognition. In fact, distinct pattern identification decisions should be taken in terms of two different driving conditions. In specific, though $\Delta SD_{max}(N) \leq \varepsilon_{SD}$ under pattern-shifting phases (e.g. see Fig. 7(b1), where more driving samples are derived from "urban" pattern, and Fig. 7(b2), where more driving samples are derived from "suburban" pattern), it is thus rational to recognize the upcoming pattern as "suburban", since the pattern-switching moment (see the purple dashed curve) is located in the current sampling horizon. Nevertheless, to prevent mis-recognitions in confusion phases (Fig. 7(c1) and (c2)), it is suggested to recognize the current pattern as "urban" since the real driving pattern does not alter.

3.2.4 Complementary rules development

To split pattern-switching phases and confusion phases, the workflow of the proposed solution (see Fig. 7(d)) is detailed as below. Provided a) $P(N-1) = 1$, b) $I_{max}(N), I_{max-2}(N) \in \{1,2\}$ and c) $\Delta SD_{max}(N) \leq \varepsilon_{SD}$, the N^{th} pattern recognition result $P(N)$ can be picked from two candidates: $I_{max}(N)$ and $I_{max-2}(N)$. Hence, we split the N^{th} sampling horizon into two identical fragments. If the driving fragment in the second-half of sampling window has sufficient supplementary driving features related to "suburban" scenario, then $P(N) = 2$. Else, $P(N) = 1$. Likewise, if "highway" and "urban" or "highway" and "suburban" are the conflict pattern pairs, the aforementioned strategy can also be adopted to finalize the pattern identification results.

To fulfill the above-mentioned solution, it is supposed to extract the supplementary driving features over the second-half of sampling window, if $\Delta SD_{max}(N) \leq \varepsilon_{SD}$. With the extracted features, the corresponding complementary rules become effective to determine if the target driving segment can be categorized into the upcoming driving pattern. Hereafter, a brief illustration of the selection of supplementary driving features in urban, suburban scenarios and the design of complementary rules are given to detail the criterion to separate conflict driving patterns.

The number of stop incident (NoS) and the average speed (v_{mean}) are chosen as the supplementary features if "suburban" and "urban" become the conflict driving patterns. To explore the statistical distributions of the selected features, numerous driving data with fixed sampling length ($0.5L_s = 75\ s$) are picked up from the offline driving database (as shown in Fig. 5). As a result, with the extracted driving samples, Fig. 8 depicts the statistical distribution of the selected features. Table 4 summarizes several key figures of distribution. Note $Pr(\cdot)$ is the probability of the related incident. Given these

statistics, the complementary rule for separating urban and suburban scenarios is depicted in Fig. 9(a). Likewise, the complementary rules for other situations can be finalized, as shown in Fig. 9(b) and (c), where the detail design procedure is omitted to avoid repetitive illustrations.

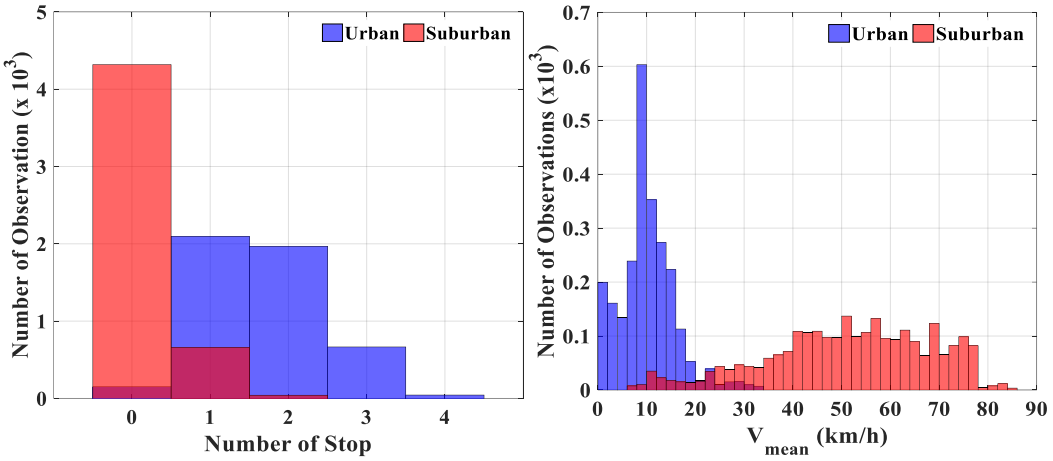

Fig. 8. Histogram on NoS and v_{mean} of driving samples (per 75s) under urban and suburban patterns.

Table 4. Statistical distributions (per 75s) for the supplementary driving features

	$\Pr(NoS = 0)$	$\Pr(NoS = 1)$	$\Pr(NoS > 1)$	$\Pr(v_{mean} > 20km/h)$
Urban	3.07%	42.55%	54.38%	4.57%
Suburban	86.01%	13.15%	0.84%	95.10%

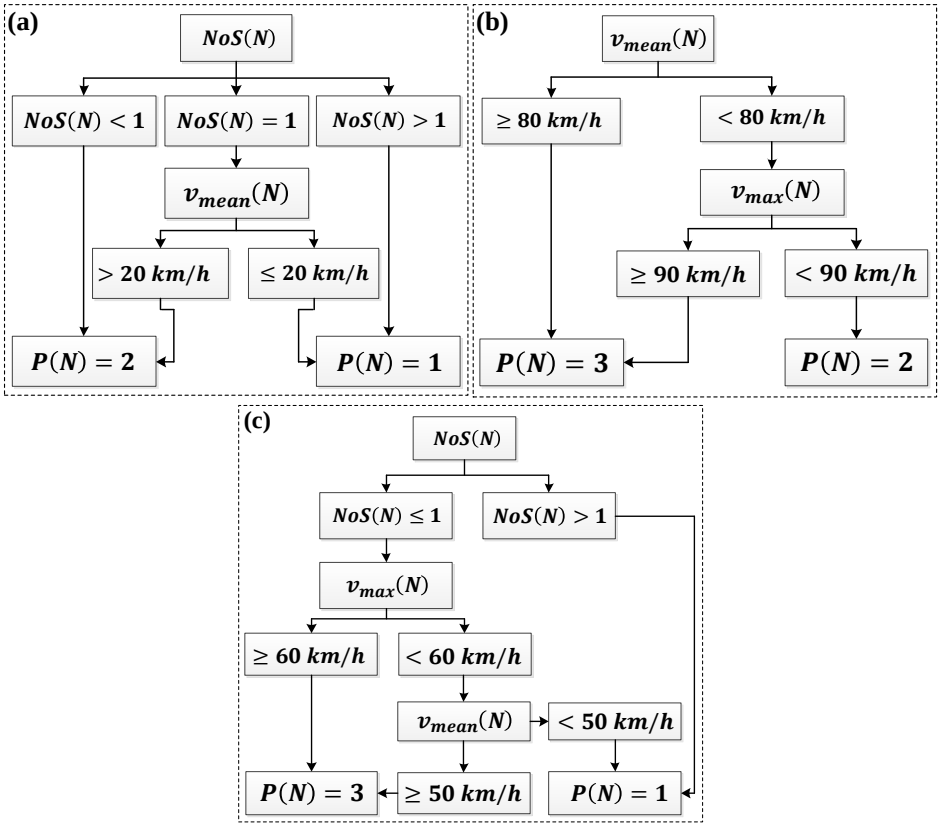

Fig. 9. Complementary rules for (a) urban/suburban, (b) highway/suburban and (c) urban/highway.

3.3 Driving pattern recognition performance validation

The devised Markov pattern recognizer is verified under combined testing cycles, wherein the Markov time scale N_T and the Markov state number s are set to 5 and 16, respectively.

3.3.1 Pattern identification results and discussions

To test the pattern-recognition performance, eight driving cycles are aggregated to generate the combined testing cycle, as depicted in Fig. 10(a).

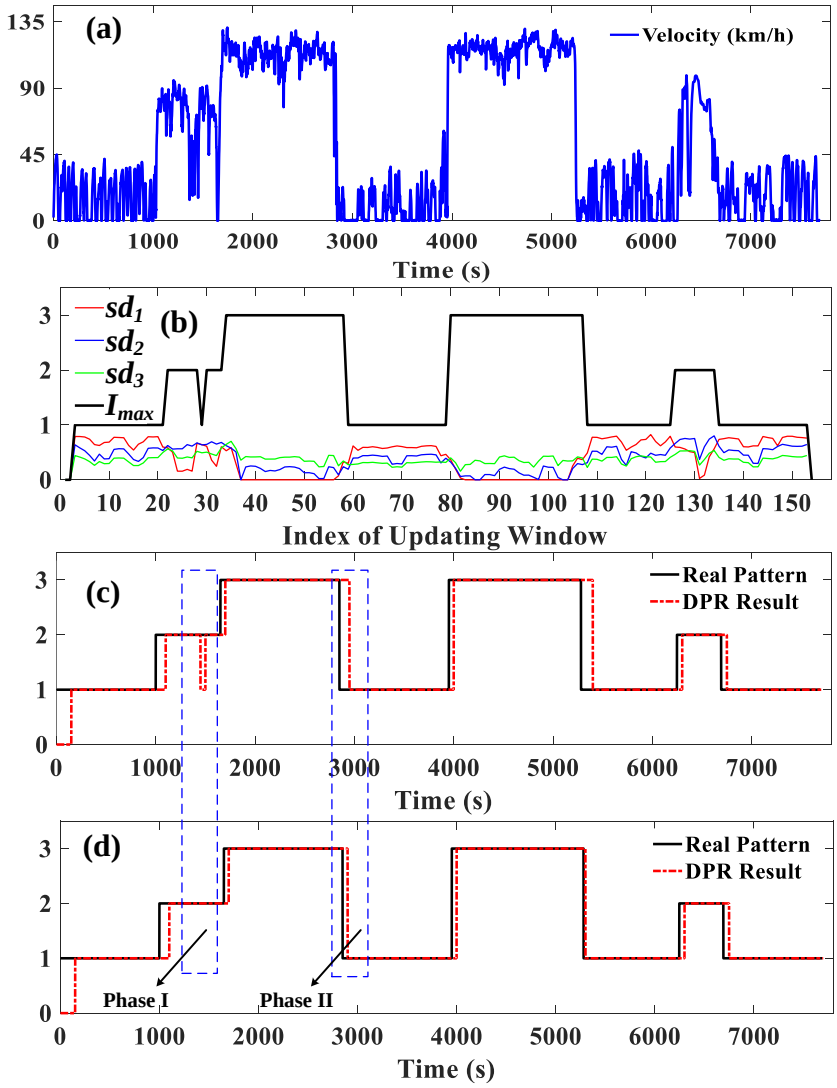

Fig. 10(a)-(d). DPR results on test cycle III. Fig. (a): speed profile of testing cycle, Fig. (b): similarity quantification results, Fig. (c) and (d): DPR results without and with complementary, respectively.

In addition, as seen in Fig. 10(b), the curves in red, blue, and green formats respectively represent the degrees of resemblance ($\mathrm{sd}_1, \mathrm{sd}_2, \mathrm{sd}_3$) versus three predefined modes, and the black curve represents the index trajectory of the largest similarity element ($\mathrm{I}_{\max}$). Moreover, the DPR results are plotted in Fig. 10(c) and (d). In general, the proposed method can properly recognize driving pattern according to $\mathrm{I}_{\max}$ if a stable driving condition is encountered. Yet, as displayed in Fig. 10(c), DPR errors tend to appear when $\Delta\mathrm{SD}_{\max} \leq \varepsilon_{\mathrm{SD}}$. Contrarily, after using the complementary rules, the DPR accuracy

improves greatly (e.g. Phase I and II in Fig. 10(d)). Note the pattern identification result is set to "unrecognized (0)" in the first 150s due to the lack of historical data for pattern recognition in this period. Overall, as depicted in Fig. 10(a)-(d), the proposed method can separate real-time driving patterns with high credibility in face of complex driving scenarios.

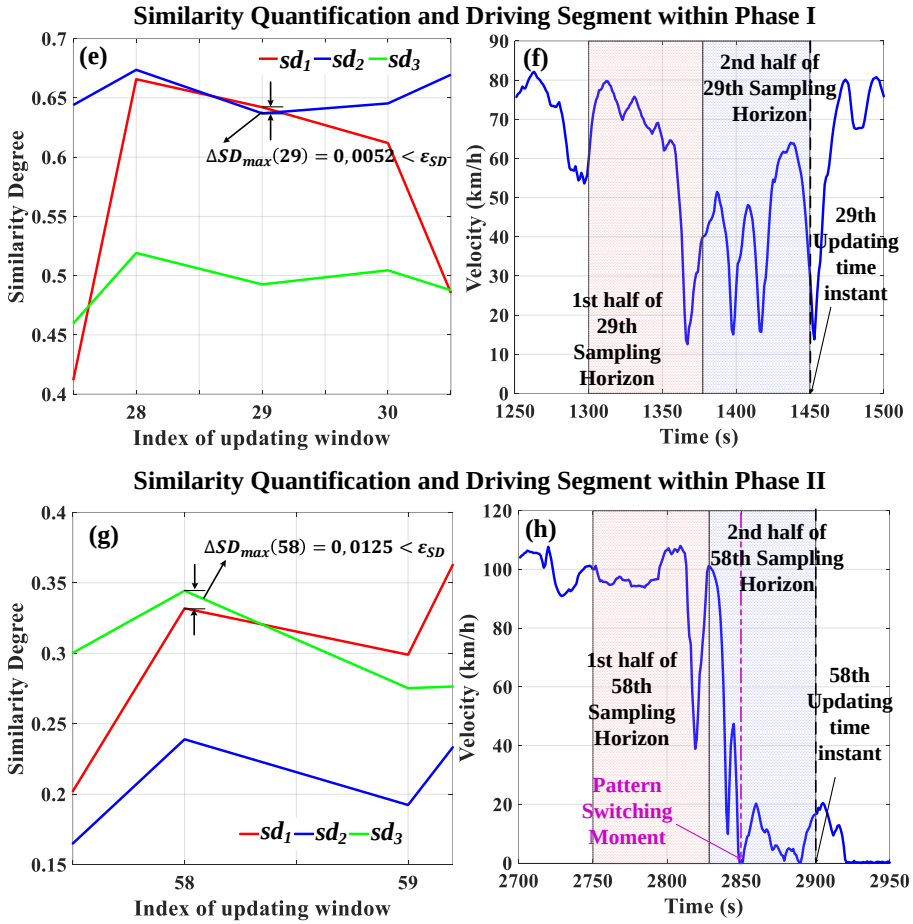

Fig. 10(e)-(h). DPR results on test cycle III: similarity quantification results in phase I and II.

In specific, with the adoption of the complementary rules, the pattern recognition performance has been improved from two aspects: 1) the hazards of mis-recognition are declined; 2) the latency before correctly identifying the upcoming pattern is reduced. For instance, as seen in Fig. 10(e), $\mathrm{sd}_1(29)$ is larger than $\mathrm{sd}_2(29)$, but their difference (0.0052) is less than the threshold $\varepsilon_{\mathrm{SD}}$ (0.05). Thus, without the help of complementary rules, current pattern is recognized as "urban" since $\mathrm{I}_{\max}(29) = 1$. This would lead to DPR errors, as depicted in phase I of Fig. 10(c). Contrarily, as depicted in Fig. 10(f), no vehicle stop is detected in the second-half of the 29th sampling window. Based on the complementary rules given in Fig. 9(a), current pattern is recognized as "suburban", so as to avoid the mis-recognition of driving pattern, as depicted in phase I of Fig. 10(d). Likewise, as seen in Fig. 10(g), since $\mathrm{I}_{\max}(58) = 3$, the 58th pattern would be recognized as "highway" if without the help of complementary rules, causing the delay of pattern identification, as depicted in phase II of Fig. 10(c). Contrarily, vehicle stops for three times ($\mathrm{NoS} > 1$) in the second-half of 58th sampling window (Fig. 10(h)). Based on the

complementary rules shown in Fig. 9(c), the 58th pattern identification result is set as "urban", thus reducing the latency of pattern identification, as depicted in phase II of Fig. 10(d).

Table 5. DPR Accuracy Comparison with/without Complementary Rules ($s = 16$ and $N_T = 5$)

	Test cycle I	**Test cycle II**	**Test cycle III**
Without complementary rules	92.32%	93.55%	92.89%
With complementary rules	94.97%	98.16%	95.55%
Accuracy Improvement	**+2.65%**	**+4.61%**	**+2.66%**

Beside the testing cycle in Fig. 10(a), two another testing cycles are adopted to validate the presented DPR method, with the average precision of identification given in Table 5. In specific, over 92.32% recognition precision can be attained by the proposed method, even without the help of complementary rules. On this basis, an additional DPR precision improvement (ranging from 2.65% to 4.61%) can be obtained with the use of the complementary rules. This proofs that the complementary rules are capable of compensating for DPR performance losses in case $\Delta SD_{max} \leq \varepsilon_{SD}$. To sum up, the proposed DPR method can reasonably differentiate online driving patterns, with the precision of 94.97% to 98.16%.

3.3.2 Impacts on pattern identification accuracy imposed by s and N_T

The settings of s and N_T would influence the precision of the presented pattern recognizer. Hence, a sensitivity analysis is conducted to explore the impacts on recognition precision by different settings on s and N_T. Related evaluation results on three driving cycles are listed in table 6.

Table 6. DPR Accuracy Comparison with Different Parameter Configurations

Parameter Settings		**Test cycle I**	**Test cycle II**	**Test cycle III**
s = 16	$\mathbf{N_T = 1}$	91.64%	88.19%	86.98%
	$\mathbf{N_T = 2}$	92.31%	92.90%	92.89%
	$\mathbf{N_T = 3}$	92.32%	94.87%	94.87%
	$\mathbf{N_T = 4}$	92.98%	97.49%	95.52%
	$\mathbf{N_T = 5}$	94.97%	98.16%	95.55%
s = 36	$\mathbf{N_T = 1}$	91.66%	86.87%	85.88%
	$\mathbf{N_T = 2}$	91.66%	89.50%	89.81%
	$\mathbf{N_T = 3}$	90.99%	93.44%	92.23%
	$\mathbf{N_T = 4}$	90.99%	93.44%	92.23%
	$\mathbf{N_T = 5}$	90.99%	94.09%	93.54%

As shown in table 6, the highest DPR precision is attained when $s = 16$ and $N_T = 5$. On one hand, if the size of Markov state-space continues to grow, more observations are requested to ensure the completeness of the online-identified TPMs. On the other hand, the limited number of driving data in the sampling window makes the expanded TPMs hard to fully capture the recent driving changes, thus degrading the precision of DPR. Moreover, a larger N_T can contribute to the higher DPR accuracy in most cases, since this can permit more online-recognized TPMs for resemblance quantification. Through such average filtering effect, the sensitivity towards the improper quantification results could be reduced, thus enhancing the precision of DPR. Yet, when N_T exceeds 5 steps, the accuracy increment effect can be neglected.

3.3.3 Performance comparison with existing DPR approaches

In pattern identification tasks, the recognition accuracy and the computation burden are two concerning issues for real applications. In this subsection, the proposed Markov-based DPR approach is compared to existing DPR approaches on these two issues, with the comparison results listed in Table 7, where SVM refers to support vector machine, MLPNN means multilayer perceptron neural network and LVQNN stands for learning vector quantization neural network. Note the average DPR accuracy of the proposed method on three testing cycles ($s = 16$ and $N_T = 5$) is adopted for comparison. Besides, the proposed DPR method uses five feature parameters, namely the velocity, the acceleration, the number of stops, the average and maximal speed.

Overall, the pattern identification accuracy of the proposed method is comparable to those in existing studies [13]-[16]. Although the DPR method in [16] results in slightly higher accuracy compared to this work, it adopts 19 feature parameters for pattern identification, which is nearly four times the amount of feature parameters used in this study. Using too many feature parameters would increase the complexity of NN structure, thus enlarging offline training time and increasing the risk of overfitting. To sum up, compared to existing DPR approaches, the proposed method can achieve a well balance between the identification accuracy and the online computation burden.

Table 7. DPR Performance Comparison Results

DPR methods	Number of Feature Parameters	Average DPR Accuracy
Proposed	5	96.22%
SVM-based [13]	4	95.20%
MLPNN-based [14]	6	95.82%
Clustering +SVM [15]	6	95.00%
LVQNN-based [16]	19	98.00%

In conclusion, the major advances of the proposed method against previous DPR methods are summarized as follows:

- The velocity-acceleration (v-a) transition behaviors, for the first time, are used as the driving feature parameters for DPR problems compared to stationary feature parameters used by traditional DPR approaches. This measure permits a more accurate description of each type of driving pattern;
- The proposed complementary rules can effectively compensate for DPR accuracy losses during the pattern-shifting phases, thus improving the reliability of pattern identification versus traditional DPR approaches.

Validation results demonstrate the proposed method can identify the real-time driving pattern with an average precision of 96.22%, where the periodically updated DPR results can facilitate the realization of multi-mode EMS framework under changeable driving scenarios.

4. Energy management strategy via model predictive control

Section 2 and 3 focus on the development of driving prediction techniques. On this basis, this section provides an example of embedding the predictive knowledge into the real-time optimization framework of model predictive control (MPC), leading to the birth of an integrated predictive energy management strategy (PEMS) for FCHEVs. The rest of this section is organized as follows. Subsection 4.1 presents the modelling of vehicle's hybrid powertrain. Subsection 4.2 illustrates the design and verification of a multi-mode PEMS, which can allocate power demand in face of changeable driving patterns.

4.1 Powertrain architecture and system modeling

This subsection presents the modeling of vehicle's hybrid powertrain system.

4.1.1 Vehicle model and powertrain architecture

As depicted in Fig. 11(a), this chapter focuses on a midsize vehicle model, which is taken from the database of the vehicular simulator ADVISOR [10]. Fig. 11(b) schemes the topology of the studied hybrid propulsion system, where the fuel cell system (FCS), attached to the DC bus via a unidirectional DC/DC converter, and the battery, directly linked to the DC bus, work cooperatively to response the power request from the electric machine. The sizes of PEMFC and battery are determined by the component-sizing method presented in our previous work [17]. The key specifications of the studied vehicle model are listed in Table 8.

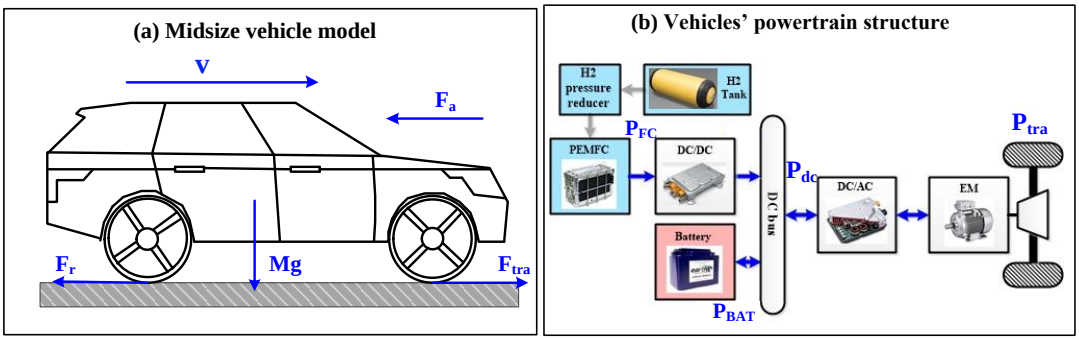

Fig. 11. Studied vehicle model: (a) midsize vehicle outline and (b) topology of the hybrid powertrain.

Moreover, the propulsion power (P_{tra}) needed by vehicle in motion can be calculated as a function of its weight (M) and speed (v), as denoted by (9) [18]. Accordingly, the output power of FCS (P_{FC}) and battery (P_{BAT}) together satisfy the DC bus power demand (P_d), as denoted by (10).

$$P_{tra} = v \cdot F_{tra} = v \cdot \left[\underbrace{c_r M g \cos(\theta)}_{F_r} + \underbrace{0.5 \rho_{air} S_f c_d v^2}_{F_a} + M\dot{v} \right] \tag{9}$$

$$P_d = \frac{P_{tra}}{\eta_{drive} \cdot \eta_{DC/AC} \cdot \eta_{EM}} = P_{BAT} + P_{FC} \cdot \eta_{DC/DC} \tag{10}$$

where c_r is the rolling resistance coefficient, ρ_{air} the air density (1.21 kg/m^3), S_f the front surface area, c_d the aerodynamic drag coefficient, g the gravitational acceleration, η_{drive} the driveline efficiency, $\eta_{DC/DC}$, $\eta_{DC/AC}$ the power converters' efficiency and η_{EM} the EM efficiency. Since a horizontal vehicle

model is used in this study, the road slope θ takes zero.

Table 8. Powertrain specifications of the vehicle models

Category	Item	Specifications
Vehicle Structural Parameters	Vehicle mass	1360 kg
	Vehicle front surface	1.746 m^2
	Tire radius	0.32 m
	Aerodynamic coefficient	0.3
	Rolling coefficient	0.0135
	Driveline efficiency	0.91
	Gravitational acceleration	9.81 m/s^2
PEMFC System	Rated power	30 kW
	Maximum efficiency	50.3 %
Lithium-ion Battery Pack	Nominal energy capacity	6.4 kWh
Electrical Machine	Maximum power	150 kW
	Maximum torque	220 N·m
	Maximum rotation speed	11000 rpm
Others	DC/DC converter Efficiency	0.90
	DC/AC converter Efficiency	0.95

4.1.2 Fuel cell model

Proton exchange membrane fuel cell (PEMFC) is used in the studied hybrid powertrain. As the core of PEMFC system, fuel cell stack converts hydrogen energy into useful electricity power (P_{stack}) via a series of electrochemical reactions. A fraction of electrical power generated by the stack is used in the auxiliaries around the stack (P_{AUX}) (e.g. air compressor, etc.) to ensure the normal operation of the entire system. In this case, the actual output (net) power from PEMFC system (P_{FC}) equals to the difference between P_{stack} and P_{AUX} . During the operation of PEMFC, the hydrogen mass consumption (M_{H_2}) can be calculated by [19]:

$$\mathrm{M_{H_2}} = \int_0^{\mathrm{t}} \frac{\mathrm{P_{FC}}(\tau)}{\eta_{\mathrm{FCS}} \cdot \mathrm{LHV_{H_2}}} \mathrm{d}\tau \tag{11}$$

Where $\mathrm{LHV_{H_2}}$ is the lower heating value of hydrogen. Moreover, let P_{H_2} denote the theoretical power supplied by hydrogen, the efficiency of fuel cell system can be expressed as:

$$\eta_{FCS} = \frac{P_{FC}}{P_{H_2}} = \frac{P_{stack} - P_{AUX}}{P_{H_2}} \tag{12}$$

More details about the modelling of auxiliaries' power consumption can be found in [18]. As a result, the efficiency curve of the studied fuel cell system (FCS) is given Fig. 12. To enhance the operating efficiency of FCS, the FCS net power with the highest system efficiency (η_{max}) is defined as the most efficient operating point, marked as P_η^{max}. Besides, the operating range $P_{FC} \in \left[P_\eta^{LOW}, P_\eta^{HIGH}\right]$, where the PEMFC system efficiency (η_{FCS}) is higher than 47%, is defined as the FCS's high efficiency area.

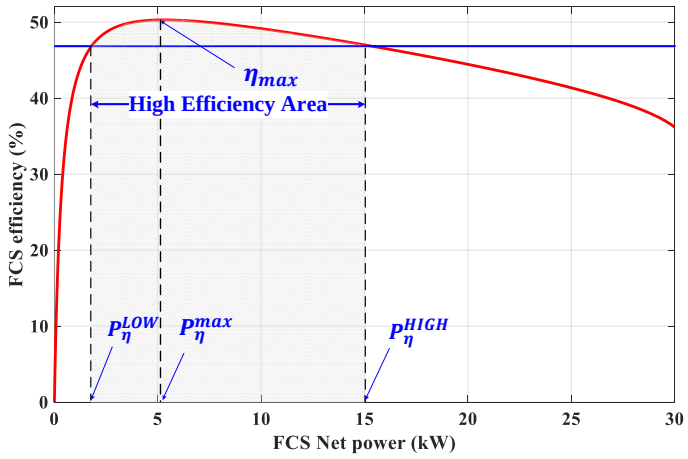

Fig. 12. Efficiency curves of a 30-kW PEMFC system efficiency.

4.1.3 Battery model

A simple but effective enough internal resistance model (R-int) is adopted to represent the behavior of a battery, with the equivalent circuit depicted in Fig. 13(a).

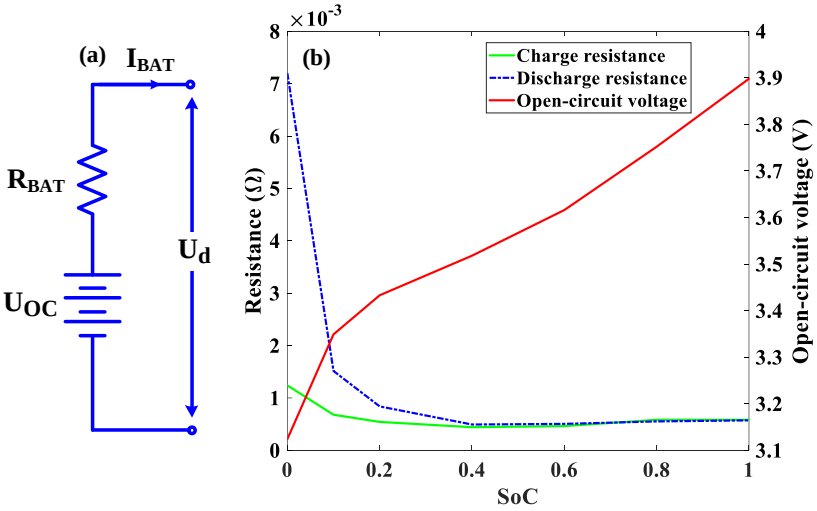

Fig. 13. Modelling of battery: (a) equivalent circuit of the R-int model and (b) relationship of the internal resistance and OCV of a single cell with respect to its SoC.

The state-of-charge (SoC) is a percentage indicator of the remaining battery capacity (in Ah) in contrast to its nominal one, as computed by:

$$SoC(t) = SoC_0 - \int_0^t \frac{\eta_{BAT} \cdot I_{BAT}(\tau)}{Q_{BAT}} d\tau \quad (13)$$

Where Q_{BAT} the nominal battery capacity, SoC_0 the initial SoC, I_{BAT} the battery current, η_{BAT} the battery efficiency (1 for discharge and 0.95 for charge). According to Kirchhoff's voltage law, the DC bus voltage (U_d) can be calculated as:

$$U_d = U_{OC} - I_{BAT} \cdot R_{BAT} \quad (14)$$

Where R_{BAT} is the battery internal resistance and U_{OC} the battery open-circuit voltage (OCV). Combine (14) with the expression of battery output power $P_{BAT} = U_d \cdot I_{BAT}$, battery current I_{BAT} can be given as:

$$I_{BAT} = \frac{U_{OC}(SoC) - \sqrt{U_{OC}(SoC)^2 - 4 \cdot R_{BAT}(SoC) \cdot P_{BAT}}}{2 \cdot R_{BAT}(SoC)} \quad (15)$$

According to [20], U_{OC} and R_{BAT} can be respectively casted into a function of SoC. Fig. 13(b) depicts how the OCV and internal resistance change with SoC. Please note the displayed battery characteristics are extracted from the experimental-validated lithium-ion battery model from ADVISOR [10].

4.1.4 Electric machine model

Electric machine (EM) is the supplier of vehicular propulsion power. Considering the vehicular maximum power/torque demands by the mission profiles, a 150-kW model is selected from the database of ADVISOR [10], with its rotation speed and torque operating ranges listed in Table 8. Moreover, the EM efficiency map, as shown in Fig.14, derived from ADVISOR is adopted to compute η_{EM} when the torque and speed requests from wheel side are specified.

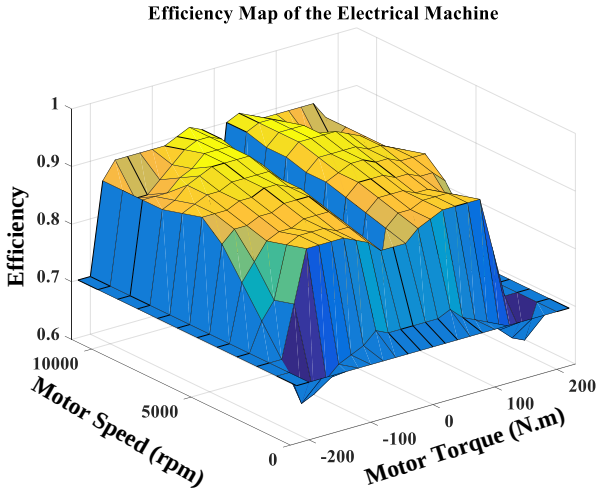

Fig. 14. Efficiency map of the studied electric machine (rated power @150 kW).

4.2 Multi-mode predictive energy management strategy

In face of changeable driving conditions in practice, energy management strategies (EMS) should be able to allocate power demands in multiple driving patterns. To this end, a multi-mode predictive EMS for FCHEV is devised, which can adapt to rapid-changing driving scenarios. Specifically, based on the Markov driving pattern recognizer (DPR) and the layer recurrent neural network (LRNN) predictor proposed in previous sections, model predictive control (MPC) is leveraged to derive the optimal power-allocating decisions at each sampling period.

Fig. 15 depicts the control framework of the proposed multi-mode EMS, which comprises a Markov DPR and a multi-mode MPC. In specific, the Markov recognizer, in the upper level, can periodically

refresh the pattern recognition results. Each identified pattern corresponds to one group of pre-optimized control parameters of MPC. Based on the speed-forecast results by LRNN and the adopted control parameters, MPC, in the lower level, takes the optimal control actions through solving a constrained optimization problem over each receding (prediction) horizon H_p. The sampling period ΔT is set as 1s. Following parts would detail the development of MPC-based EMS.

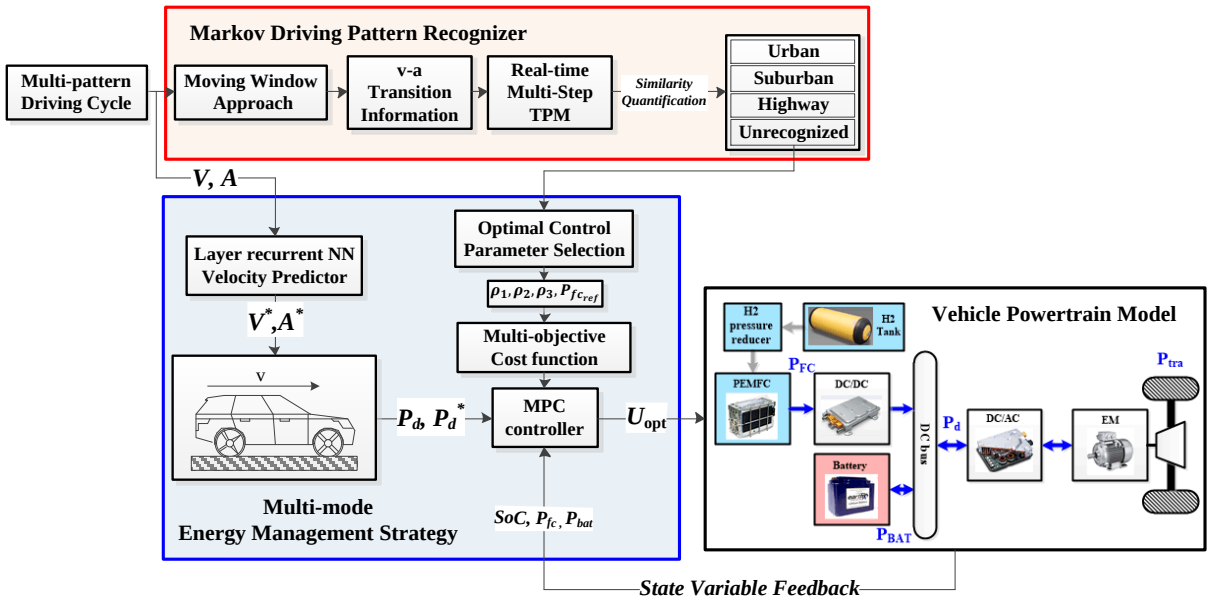

Fig. 15. Control framework of the multi-mode energy management strategy.

4.2.1 Model predictive control: a brief introduction

Model predictive control is one of the most widely-used advanced control methods in multiple industrial sectors [21], which comprises the following three elements, as shown in Fig. 16.

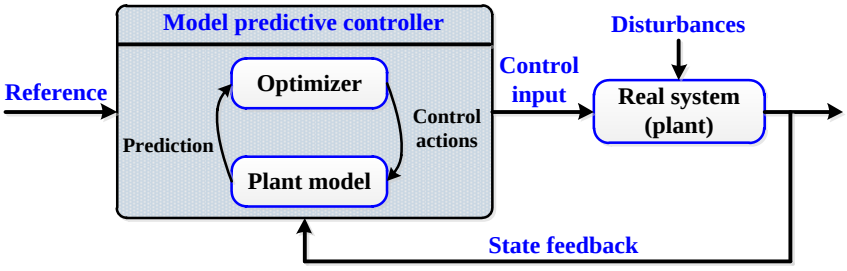

Fig.16. Illustration of model predictive control framework.

(a) Predictive Model: the term "model" in MPC refers to the control-oriented model (plant model), which is capable of representing the future dynamic behaviors of the real system (plant) according to the input information. The plant model is typically given in the form of state-space representation or transfer function, and the precision of system modelling can greatly affect the performance of MPC.

(b) Rolling Optimization: MPC takes control actions via optimizing the performance index (quantified

by the cost function) over a finite time horizon. Specifically, with the plant state sampled at time instant t = k, MPC optimizes the performance index over the time horizon [k, k + H_p-1], where H_p > 1 is the length of prediction horizon. At the next time instant, the time horizon shifts forward to [k + 1, k + H_p] with the optimization performed again. In this way, the optimization is repeatedly conducted online.

(c) Feedback correction: After obtaining the optimal control sequence, containing H_p elements, at time instant *k*, MPC only implements the first one to the real system while discards the others. This measure can prevent the control performance losses imposed by model distortion or disturbances in environment.

To sum up, as displayed in Fig. 17, the MPC working flow includes three steps: (i) Future system state trajectory estimation, (ii) MPC performance index optimization over finite time horizon, and (iii) Application of the first optimal control element to the real system. Once the plant states are updated, step (i) to (iii) is sequentially carried out. Afterwards, the prediction horizon moves forward, the system states are resampled and the calculation (step (i) to (iii)) is repeated starting from the new states.

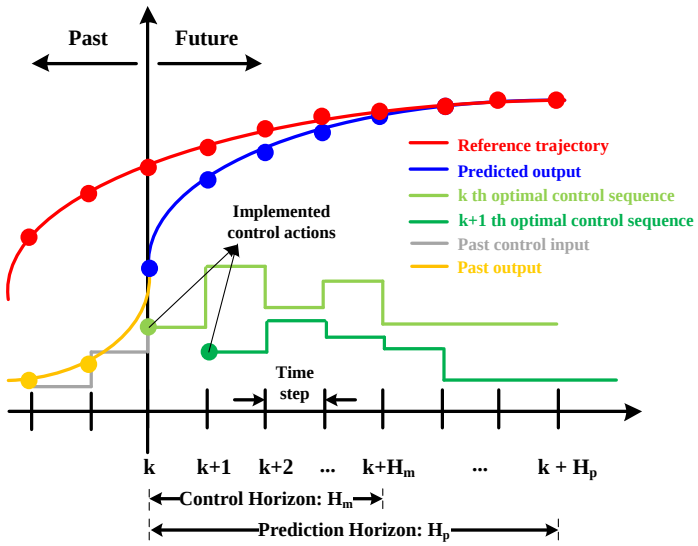

Fig.17. Representation of MPC working principle [6].

4.2.2 Multi-mode model predictive controller

This subsection presents the design of the multi-mode predictive controller.

4.2.2.1 Control oriented model

Let $\mathbf{x}, \mathbf{u}, \mathbf{y}, \mathbf{w}$ and $\mathbf{r}$ respectively denote the state, control (input), output, disturbance and reference vector, a linear discrete-time system model is adopted, with the system variable definitions and the state-space representation given as below:

$$x(k+1) = A(k)x(k) + B_u(k)u(k) + B_w w(k) \quad \text{(a)}$$
$$y(k) = Cx(k) \quad \text{(b)} \qquad (16)$$

$$\text{with}\begin{cases} x(k) &= [SoC(k)\ P_{FC}(k-1)]^T \\ u(k) &= \Delta P_{FC}(k) = \frac{P_{FC}(k)-P_{FC}(k-1)}{\Delta T} \\ y(k) &= [SoC(k)\ P_{FC}(k-1)]^T \\ w(k) &= P_d(k) \end{cases} \tag{17}$$

Besides, the reference vector $r(k) = \left[SoC_{ref}\ P_{fc_{ref}}(k)\right]^T$ includes the reference values of battery SoC and fuel cell power. Additionally, a first-order differential approximation of SoC dynamics [5] and the discrete form of DC-bus power balance can be denoted by (18) and (19), respectively.

$$SoC(k+1) = SoC(k) - \frac{\Delta T \cdot \eta_{BAT}}{U_d(k) \cdot Q_{BAT}} \cdot P_{BAT}(k) \tag{18}$$

$$P_d(k) = P_{FC}(k) \cdot \eta_{DC/DC} + P_{BAT}(k) \tag{19}$$

Combine (16)-(19), the system matrices $A(k), B_u(k), B_w(k), C$ can be specified as:

$$\begin{aligned} A(k) &= \begin{bmatrix} 1 & \frac{\Delta T \cdot \eta_{DC/DC} \cdot \eta_{BAT}}{U_d(k) \cdot Q_{BAT}} \\ 0 & 1 \end{bmatrix} \qquad B_u(k) = \left[\frac{\Delta T \cdot \eta_{DC/DC} \cdot \eta_{BAT}}{U_d(k) \cdot Q_{BAT}} \quad 1\right]^T \\ B_w(k) &= \left[-\frac{\Delta T \cdot \eta_{DC/DC} \cdot \eta_{BAT}}{U_d(k) \cdot Q_{BAT}} \quad 0\right]^T \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{20}$$

4.2.2.2 Cost function and constraints

The devised EMS intends to achieve three objectives: a) the saving of H2 consumption, b) the extension of fuel cell lifespan and c) the regulation of battery SoC. Note the second objective is transformed into limiting the power transients of FCS, since the steadier the fuel cell power is, the friendlier the operating conditions are, which will mitigate the degradation of FCS and thus contribute to a longer service time. Hence, at time step $t = k$, the objective function $J(k)$ is expressed as below:

$$J(k) = \sum_{i=1}^{H_p} [\boldsymbol{\rho}_1(k) \cdot \mathbf{C}_1(k+i) + \boldsymbol{\rho}_2(k) \cdot \mathbf{C}_2(k+i-1) + \boldsymbol{\rho}_3(k) \cdot \mathbf{C}_3(k+i)]$$

$$\text{with}\begin{cases} \mathbf{C}_1(k+i) &= \left(\frac{P_{FC}(k+i-1)-P_{ref}(k)}{P_{FC}^{max}}\right)^2 \\ \mathbf{C}_2(k+i-1) &= \left(\frac{\Delta P_{FC}(k+i-1)}{\Delta P_{FC}^{max}}\right)^2 \\ \mathbf{C}_3(k+i) &= \left(\frac{SoC(k+i)-SoC_{ref}}{SoC_{max}-SoC_{min}}\right)^2 \end{cases} \tag{21}$$

where $\mathbf{C}_1$ to $\mathbf{C}_3$ are the cost terms related to three EMS objectives, and $P_{FC}^{max} = 30\ \text{kW}, \Delta P_{FC}^{max} = 1\ \text{kW/s}, SoC_{max} = 0.8$ and $SoC_{min} = 0.6$. The major functions of $\mathbf{C}_1$ to $\mathbf{C}_3$ are specified as below:

- $\mathbf{C}_1$ is adopted to urge fuel cell working towards the reference point P_{ref};
- $\mathbf{C}_2$ is leveraged to limit the harsh power transients to mitigate the FCS performance degradation imposed by overlarge load dynamics [22];
- $\mathbf{C}_3$ is deployed to guarantee the SoC regulation performance. Since a non-plug-in vehicle configuration is adopted, the reference SoC value (SoC_{ref}) is identical to the initial SoC value

(SoC_0), so as to avoid battery over-discharge or over-charge, namely $SoC_{ref} = SoC_0 = 0.7$.

In addition, $\boldsymbol{\rho_1, \rho_2, \rho_3}$ are three penalty coefficients, reflecting the weights on cost terms $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$. The determination of $\boldsymbol{\rho_1, \rho_2, \rho_3}$ and $\mathbf{P_{ref}}$ under different driving patterns will be detailed thereafter. Moreover, the control horizon length of MPC is set identical to the prediction horizon length, where H_p is set at five steps. Within each receding horizon, following constraints should be enforced:

$$\begin{cases} \underline{SoC} \le SoC(k+i) \le \overline{SoC} & (a) \\ \underline{P_{FC}} \le P_{FC}(k+i-1) \le \overline{P_{FC}} & (b) \\ \underline{\Delta P_{FC}} \le \Delta P_{FC}(k+i-1) \le \overline{\Delta P_{FC}} & (c) \\ \underline{P_{BAT}} \le P_{BAT}(k+i) \le \overline{P_{BAT}} & (d) \\ w(k+i) = P_d^*(k+i), i \ge 1 & (e) \end{cases} \tag{22}$$

where constraint (22a) guarantees the battery operation safety, where $\underline{SoC} = 0.55, \overline{SoC} = 0.85$. If SoC < 0.6 or SoC > 0.8, the EMS emergency working mode will be activated to enforce SoC return to the desired operating range [0.6, 0.8] as soon as possible. In addition, the operating boundaries for P_{FC}, ΔP_{FC} and P_{BAT} are indicated by constraints (22b)-(22d), where $\underline{P_{FC}} = 0\text{W}$, $\overline{P_{FC}} = 30$ kW, $\overline{\Delta P_{FC}} = -\underline{\Delta P_{FC}} = 1$ kW, $\underline{P_{BAT}} = -50$ kW and $\overline{P_{BAT}} = 100$ kW. Constraint (22e) assigns the k^{th} disturbance as the forecasted power demand $\left[P_d^*(k+1), \dots, P_d^*(k+H_p)\right]$, which is derived based on the speed-forecast results $V_k^* = \left[v_{k+1}^*, \dots, v_{k+H_p}^*\right]$ by LRNN and the vehicles' dynamics (9) and (10).

It should be noted that since a quadratic performance index J(k) is adopted as the MPC cost function, the k^{th} control actions $U^*(k) = \left[u_1^*(k), \dots, u_{H_p}^*(k)\right]$ can be derived by minimizing (21) with respect to linear constraints (22). Such a problem can be converted into a quadratic programming (QP) problem, and thus can be solved using the open-source solver *qpOASES* [23]. Thereafter, only the first element of $U^*(k)$ is implemented to vehicle model, while the others are abandoned.

4.2.2.3 Multiple working modes and parameter design of energy management strategy

To adapt to changeable driving scenarios, multiple EMS working modes are defined and the switching of working mode is accomplished via using different sets of MPC control parameters. In specific, we consider the following EMS working modes:

Normal working stage. Based on the power demand under urban/suburban/highway scenarios, three groups of control parameters for MPC are tuned in offline. Then, with the periodically renewed DPR results, one group of well-tuned parameters is picked for real-time control to deal with related driving conditions. The offline parameters tuning process will be introduced afterwards.

SoC emergency stage. If SoC < 0.6 or SoC > 0.8, $\boldsymbol{\rho_3}$ is increased to ten times of the normal value to enforce SoC return to [0.6,0.8]. If SoC emergency incident occurs, the setting of control parameters is

turned to the "SoC emergency" mode and keeps unchanged until the next pattern updating moment.

Start-up stage. Without available recent driving information, the pattern identification result is set to "unrecognized" over the first sampling window ($t \in [1,150]$). In the start-up stage, the control parameters for MPC are tuned to make battery as the primary energy provider, whereas the fuel cell only starts to work in case SoC < 0.6.

- **Flowchart of parameter tuning for model predictive control**

The performance of MPC relies highly on its control parameter settings, namely $(\boldsymbol{\rho_1}, \boldsymbol{\rho_2}, \boldsymbol{\rho_3})$ and $\mathbf{P_{ref}}$. To obtain the suitable parameter setting of each driving pattern, the flowchart of parameter tuning is shown in Fig. 18, including four steps: (i) dynamic programming (DP) is executed over each type of mission profile to extract the global-optimal results. (ii) Related $\mathbf{P_{ref}}$ is attained based on the statistical distributions of the DP-extracted fuel cell working points. (iii) Given the fuel cell reference power and weighting coefficient candidates, several performance metrics of MPC-based EMS (e.g. hydrogen consumption, SoC final value etc.) over the identical driving cycles are compared versus DP-optimized results. (iv) According to the performance discrepancies, $\boldsymbol{\rho_1}, \boldsymbol{\rho_2}, \boldsymbol{\rho_3}$ are tuned via trials and errors.

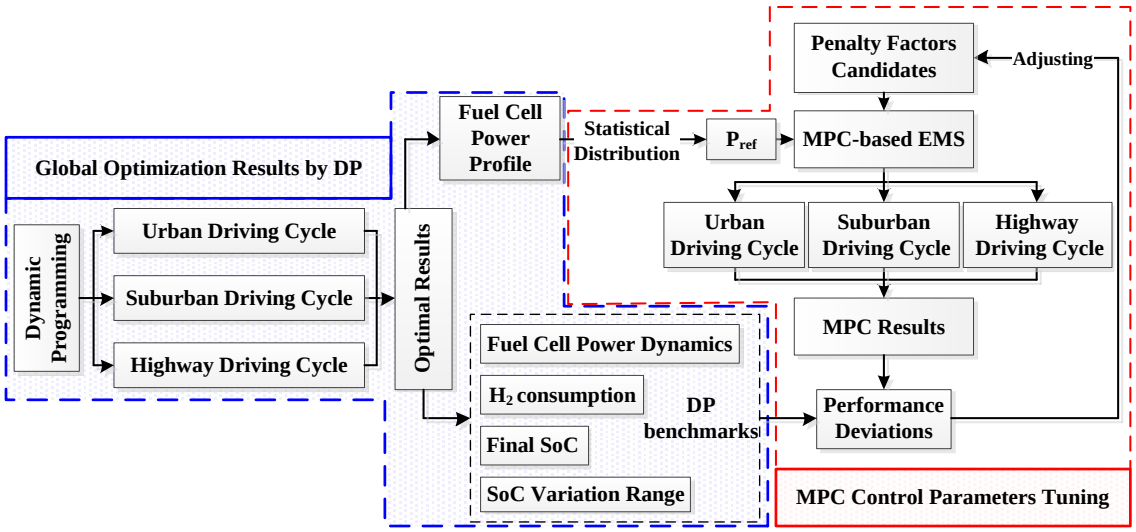

Fig. 18. Flowchart of MPC control parameter tuning process.

- **Selection of fuel cell reference working points**

With the entire driving cycle information beforehand, DP can obtain the global optimal power-allocating effect, which offers a benchmark for the selection of fuel cell reference working points. Considering the various power demand features, the objective functions of DP vary under different driving scenarios. Specifically, the optimization objective in urban regions is to limit the power transients of fuel cell (the summation of fuel cell power variation along the trip) against the fast-dynamic power requests to prolong the lifespan of FCS. In comparison, under suburban and highway regions, the objective is to improve the average fuel cell working efficiency to save H2 consumption. Additionally, during the

optimization, the constraints for DP and MPC based strategies are the same. Fig. 19(a)-(c) show the DP-optimized results over three driving patterns, with the related distributions of fuel cell power depicted in Fig. 19(d). In urban pattern, the optimal fuel cell working points are located in the range of [1.5, 2.3] kW. The optimal fuel cell working range, in suburban, is [6.0, 7.0] kW, whereas the optimal fuel cell power range is [13.5, 15.5] kW in highway pattern. Thus, the reference fuel cell power (P_{ref}) is set to the corresponding average values, specifically, 1.78 kW for urban, 6.80 kW for suburban and 14.40 kW for highway.

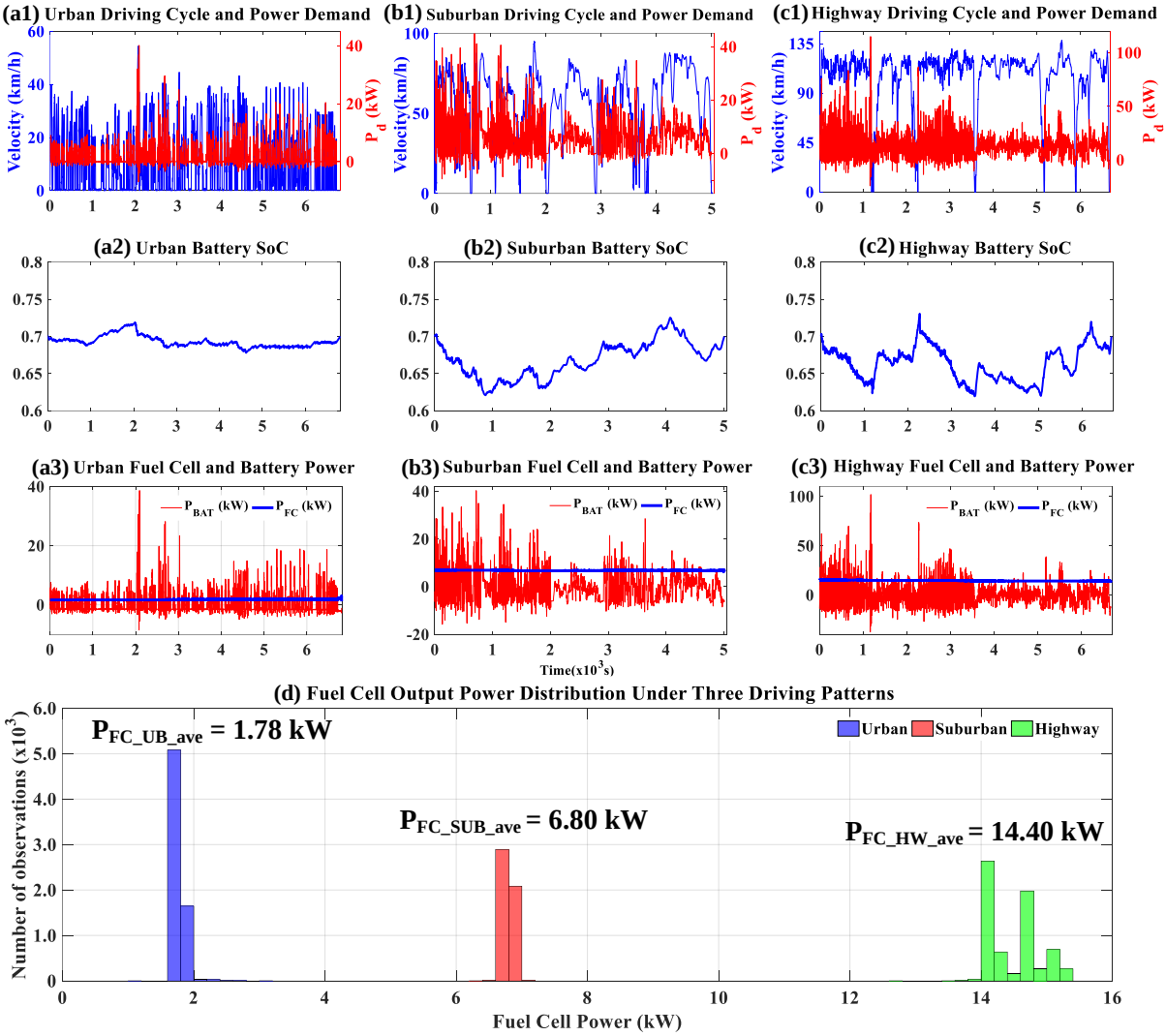

Fig. 19. DP-based optimization results under three driving patterns: (a1)-(a3) DP results under urban scenarios; (b1)-(b3) DP results under suburban scenarios; (c1)-(c3) DP results under highway scenarios; (d) distribution of fuel cell working points under three driving patterns.

- **Tuning results of penalty coefficients**

According to the P_{ref} obtained previously, Fig. 20 depicts the tuning results of MPC penalty coefficients. Please note the non-tuned MPC uses the initial penalty coefficient setting (e.g. $\boldsymbol{\rho}_1 = \boldsymbol{\rho}_2 = 1$, $\boldsymbol{\rho}_3 =$

1000), which intends to keep battery working in charge-sustaining mode. As shown in Fig. 20(a), with the tuned penalty coefficients (red curve), the power transients of fuel cell in urban pattern is greatly declined versus MPC with non-tuned coefficients (green curve). Likewise, as given in Fig. 20(b) and (c), with tuned coefficients, the fuel cell power transients are limited in a relatively narrow range.

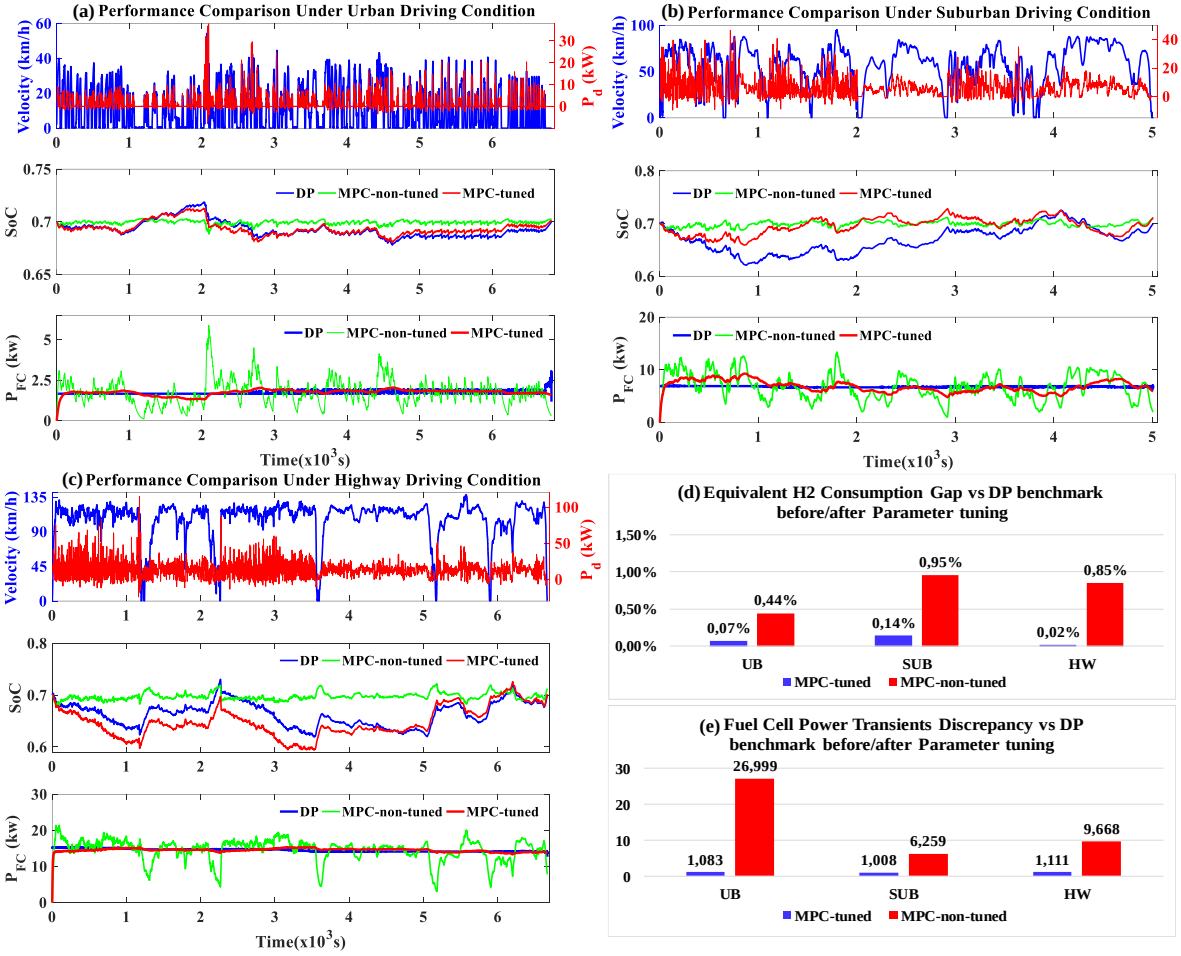

Fig. 20. EMS performance comparison before/after MPC penalty factor tuning: (a) urban scenarios; (b) suburban scenarios; (c) highway scenarios; (d) fuel economy discrepancy vs. DP benchmark; (e) fuel cell power dynamics discrepancy vs. DP benchmark.

Table 9. MPC Performance Gaps versus DP benchmark before/after parameter tuning

		Urban (UB)			Suburban (SUB)			Highway (HW)		
		DP	**MPC-T**	**MPC-N**	**DP**	**MPC-T**	**MPC-N**	**DP**	**MPC-T**	**MPC-N**
SoC_N		0.7000	0.7021	0.7032	0.7000	0.7123	0.7102	0.7000	0.7021	0.7131
m_{equ,H_2}(g)		135.10	135.20	135.71	418.30	418.90	422.29	1302.50	1302.71	1313.62
$\overline{\lvert\Delta P_{fc}\rvert}$ (W/s)		1.41	1.42	35.29	13.32	13.43	83.39	11.01	12.23	106.42
Opt. MPC parameters	**(ρ_1,ρ_2,ρ_3)**	**(1,2,100)**			**(1,1,60)**			**(1,0.2,54)**		
	P_{ref}	**1.78 kW**			**6.80 kW**			**14.40 kW**		

Compared to DP benchmark, the performance gaps by MPC-based strategies are given in Table 9. The acronym “MPC-N” and “MPC-T” represent MPC with non-tuned and tuned penalty coefficients, respectively, and m_{equ,H_2} is the equivalent hydrogen mass consumption, which takes the final SoC

deviation against the initial value (0.7 in our case) into consideration, as calculated by:

$$m_{equ,H_2} = \int_0^t \frac{P_{FC}(\tau)}{\eta_{FCS} \cdot LHV_{H_2}} d\tau + \frac{\Delta SoC_N \cdot E_{BAT} \cdot 3600}{\bar{\eta}_{FC} \cdot LHV_{H_2}} \quad (23)$$

where E_{BAT} is the nominal energy capacity of battery pack (6.4 kWh), $\Delta SoC_N = SoC_0 - SoC_N$, and $\bar{\eta}_{FC}$ the average fuel cell working efficiency. As depicted in Fig. 20(d), with the tuned penalty coefficients, the largest performance deviation on m_{equ,H_2} versus DP benchmark is merely 0.14%. In addition, as depicted in Fig. 20(e), the average fuel cell power transients caused by MPC-N strategy is 6.259 to 26.999 times of DP benchmark. In contrary, with the tuned parameters, this ratio has reduced sharply to 1.008 to 1.111 times of DP basis. Hence, it is confirmed that the penalty coefficients of MPC are well tuned, with the optimized parameters listed in Table 9.

4.2.3 Energy management strategy performance validation

In this subsection, Software-in-the-Loop (SIL) simulation is used as a tool to verify the performance of the proposed multi-mode energy management strategy.

4.2.3.1 Description of the software-in-the-loop testing platform

SIL testing is one of the important simulation-based tools for initial prototyping before the integration of any actual hardware; it is used to further validate the performance of control strategies [24], [25]. The major task of SIL testing is to justify the behavior of the generated code (functional), and gives further proofs of the generated code running on the embedded target (embeddability). In this study, a SIL testing platform is set up (see Fig. 21), which allows the proposed strategy to be tested in the dSPACE hardware (MicroAutoBox II 1401/1511 [26]), thereby further verifying its functionality and real-time suitability.

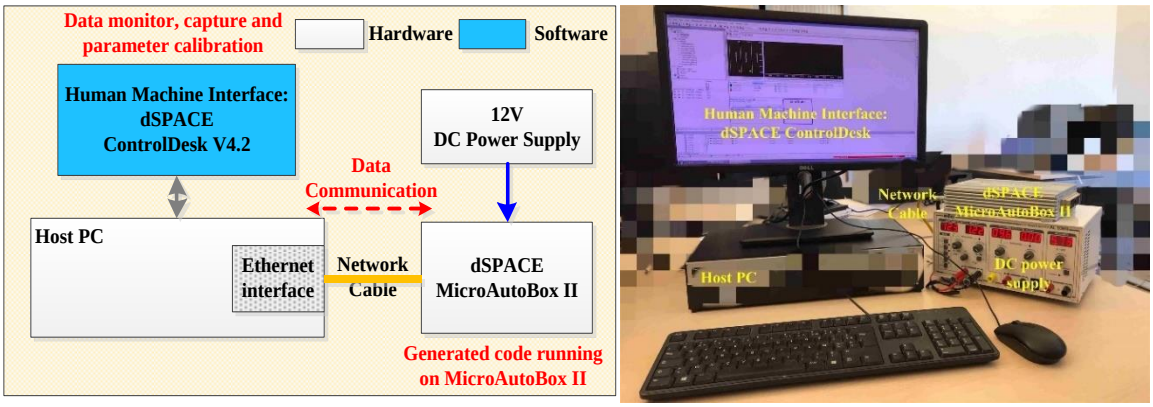

Fig. 21. (a) Block diagram and (b) real picture of the SIL testing platform.

The SIL platform comprises hardware and software subsystems, wherein the hardware subsystem includes a DC power supply, a host PC and a dSPACE MicroAutoBox II real-time system. The software subsystem contains the vehicular powertrain model and the control algorithms (PEMS) developed in the MATLAB/Simulink environment, which are compiled into C code and downloaded into the

MicroAutoBox II. Besides, the dSPACE ControlDesk software is installed in the host PC as the human machine interface (HMI) to calibrate the model parameters and to capture the experimental data during the online simulation. The host PC and the MicroAutoBox II is connected via a network cable through the Ethernet interface, and the data communication between them is enabled by the dSPACE real-time interface (RTI) module.

In SIL testing, the proposed control strategy has been successfully executed in real-time under three sampling period settings, namely 1.0s, 0.5s and 0.2s, meaning the required computational hardware expense is far from reaching the upper limits of the target CPU, thus proving that the resulting computation burden is acceptable for online applications. To avoid the repetitive illustrations, only the testing results at the sampling period of 1.0s are presented in the following parts.

4.2.3.2 Description of the benchmark energy management strategies

Upper benchmark: as expressed by (24), the global optimal result is derived by dynamic programming (DP), which intends to minimize the H2 consumption over the given driving cycles. The global-optima searching can only be finished in offline due to the demand of full cycle information beforehand.

$$\min_{\Delta P_{FC}\in\mu_{FC}} \sum_{k=0}^{N-1}\left[\frac{P_{FC}(k)}{\eta_{FCS}(P_{FC})\cdot LHV_{H_2}}\right]\cdot\Delta T$$

Subject to

$$\begin{cases} 0.6 \le SoC(k) \le 0.8 & (a)\\ 0 \le P_{FC}(k) \le 30\ kW & (b)\\ -1\ kW/s \le \Delta P_{FC}(k) \le 1\ kW/s & (c)\\ -50\ kW \le P_{BAT}(k) \le 100\ kW & (d)\\ SoC_0 = 0.7, P_{FC_0} = 0\ W & (e)\\ SoC_N = 0.7 & (f) \end{cases} \tag{24}$$

Where ΔP_{FC} is deemed as the manipulated variable. μ_{FC} is the feasible region of ΔP_{FC} in discretized domain, where the grid resolution is 1 W/s. Constraints (24a)-(24d) respectively indicate the operating boundaries for SoC, P_{FC}, ΔP_{FC} and P_{BAT}. In addition, (24e) defines the initial status of SoC and FC power, and (24f) poses a terminal constraint on battery SoC.

Lower benchmark is a single-mode MPC-based strategy, wherein the reference fuel cell power is set to the most efficient system working point ($P_{ref} = P_{\eta}^{max}$, as shown in Fig. 13). Furthermore, to deal with the unpredictable driving scenarios, the penalty coefficients are set to the initial configuration (the same as the non-tuned MPC controller in subsection 4.2.2.3, $\boldsymbol{\rho}_1 = \boldsymbol{\rho}_2 = 1,\ \boldsymbol{\rho}_3 = 1000$), with the purpose of keeping battery working in charge sustaining mode to the utmost extent, so as to ensure the operation safety of the hybrid propulsion system.

4.2.3.3 Evaluation results on multi-pattern testing cycles

Five combined testing cycles are adopted for performance validation, with the evaluation results over testing cycle I detailed in Fig. 22.

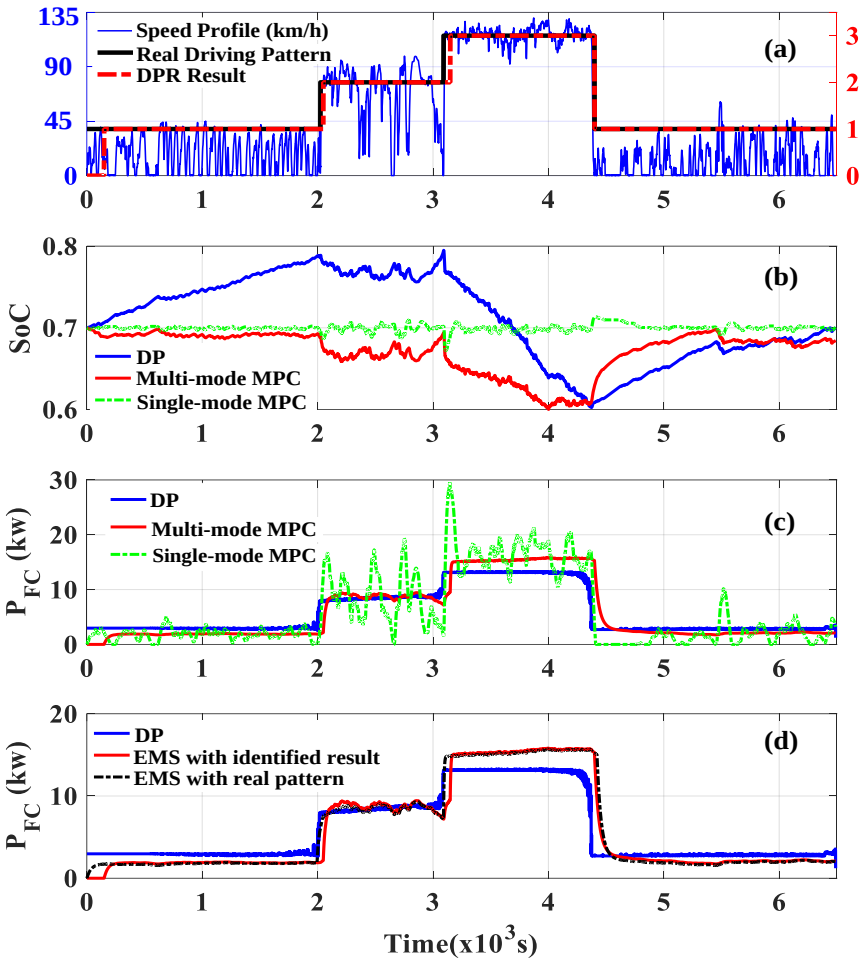

Fig. 22. Evaluation results over testing cycle I: (a) speed and driving pattern information; (b) battery SoC trajectories; (c) fuel cell power comparison; (d) influences of pattern recognition errors on fuel cell power.

As depicted in Fig. 22(a), the testing cycle I comprises urban, suburban and highway scenarios, with the real driving pattern marked with black solid format and the pattern recognition result displayed in red dashed format. On this testing cycle, 97.05% DPR precision can be attained, with the errors largely imposed by the latency in pattern-switching stages. Fig. 22(b) shows the SoC trajectories of three strategies, where, in urban scenarios, DP charges battery to response the peaking power requests in the upcoming suburban and highway scenarios. The multi-mode MPC strictly restricts battery SoC varying around 0.7 in urban scenarios, whereas the battery power is used in a more flexible way in other scenarios. Additionally, the single-mode MPC maintains SoC firmly 0.7 over the entire trip. As shown in Fig. 22(c), DP uses fuel cell power at different levels in multiple driving patterns with few transient loadings. Likewise, the multi-mode MPC controls fuel cell operating at different reference points in a stable way. In contrary, the single-mode MPC leads to more fuel cell transient loadings and start-stop cycling. Fig. 22(d) shows how DPR errors affect the output power of fuel cell. The outcome of a multi-

mode MPC assisted by real driving-pattern information (100% precision) is plotted in black dashed format. In terms of multi-mode MPC with pattern recognition results (red curve), the fuel cell power switching latency can be seen at each pattern-shifting moment. Their performance deviation on fuel cell power is neglectable if external driving conditions are stable.

Table 10 lists the numerical results on five testing cycles. The acronym "MPC-S" represents the single-mode MPC, "MPC-R" and "MPC-M" respectively stand for the multi-mode MPC assisted by real driving pattern and pattern-recognition results. As the upper benchmark, DP leads to the least H2 consumption m_{equ,H_2} and the smallest power transients of fuel cell $|\overline{\Delta P_{FC}}|$. Compared with the MPC-S strategy on five testing cycles, the MPC-M strategy can respectively reduce m_{equ,H_2} by 2.07% to 3.26% and $|\overline{\Delta P_{FC}}|$ by 87.75% to 88.98%. This implies the improved fuel efficiency and the decreased risk of fuel cell degradation by harsh transient loadings. In addition, it can be seen that the DPR errors can enlarge m_{equ,H_2} by 0.06% to 1.30%, if comparing the outcomes of MPC-R and MPC-M strategies.

Table 10. Numerical control performance on five combined testing cycles.

Type	Road information	Metrics	DP	MPC-R	MPC-M	MPC-S
Combined Cycle I (CYC_I)	**Type: "UB + SUB +HW +UB"**	**SoC_N**	0.7000	0.6998	0.6844	0.7010
		m_{H_2} (g)	**474.30**	479.21	480.50	502.10
	DPR accuracy = 97.05%	**m_{equ,H_2} (g)**		**479.50**	**486.02**	**501.72**
		$\overline{\lvert\Delta P_{fc}\rvert}$(w/s)	**9.07**	**9.87**	**9.99**	**89.71**
Combined Cycle II (CYC_II)	**Type: "UB + SUB +HW +SUB"**	**SoC_N**	0.7000	0.7149	0.7133	0.7030
		m_{H_2} (g)	**552.10**	566.10	566.51	576.1
	DPR accuracy = 96.26%	**m_{equ,H_2} (g)**		**560.84**	**561.85**	**575.03**
		$\overline{\lvert\Delta P_{fc}\rvert}$(w/s)	**8.89**	**9.58**	**9.63**	**87.40**
Combined Cycle III (CYC_III)	**Type: "UB + SUB +HW +SUB+UB"**	**SoC_N**	0.7000	0.7067	0.7086	0.7012
		m_{H_2} (g)	**488.90**	503.7	504.60	512.70
	DPR accuracy = 96.24%	**m_{equ,H_2} (g)**		**501.34**	**501.63**	**512.25**
		$\overline{\lvert\Delta P_{fc}\rvert}$(w/s)	**9.85**	**10.03**	**10.59**	**86.48**
Combined Cycle IV (CYC_IV)	**Type: "UB + SUB +HW +UB"**	**SoC_N**	0.7000	0.7055	0.7066	0.7012
		m_{H_2} (g)	**527.02**	541.10	542.05	553.62
	DPR accuracy = 94.95%	**m_{equ,H_2} (g)**		**539.14**	**539.66**	**553.18**
		$\overline{\lvert\Delta P_{fc}\rvert}$(w/s)	**8.27**	**8.83**	**8.95**	**79.27**
Combined Cycle V (CYC_V)	**Type: "UB + SUB +HW +UB"**	**SoC_N**	0.7000	0.6956	0.6966	0.7011
		m_{H_2} (g)	**450.40**	458.50	459.50	476.60
	DPR accuracy = 96.61%	**m_{equ,H_2} (g)**		**460.08**	**460.73**	**476.25**
		$\overline{\lvert\Delta P_{fc}\rvert}$(w/s)	**9.89**	**10.41**	**10.53**	**93.09**

As a conclusion, in contrast to single-mode strategy, the presented multi-mode strategy can lead to i) at least 87.00% reduction of power transient of fuel cell and ii) over 2.07% reduction of H2 consumption. Hence, under changeable driving scenarios, the FCS's operating and maintenance costs could be highly mitigated via the proposed strategy. This should be deemed as the major benefit concerning the actual implementation of the devised multi-mode energy management strategy.

5. Conclusion

This chapter develops a predictive energy management strategy (PEMS) for fuel cell hybrid electric

vehicle (FCHEV). Compared to traditional control strategies, the proposed one especially concentrates on the combination of driving predictive information and real-time optimization framework, so as to further improve vehicle's economic and durability performance. To this end, two driving prediction techniques are designed firstly:

i) a layer recurrent neural network speed predictor is proposed to estimate vehicle's upcoming speed profiles over each receding horizon;

ii) a Markov driving pattern recognizer is devised to differentiate real-time driving patterns, which establishes a solid basis for the realization of multi-mode energy management framework.

On this basis, combining the driving prediction techniques with the model predictive control (MPC) framework, a multi-mode PEMS is developed for a midsize sedan powered by fuel cell and battery, aiming at splitting power demand under changeable driving patterns. In order to verify the effectiveness of the proposed strategy, a Software-in-the-Loop (SIL) platform is established based on the dSPACE MicroAutoBox II real-time system. Validation results show that the proposed control strategy can be properly embedded into and correctly executed on the target hardware with the predefined objectives achieved, thus verifying the EMS's functionality and real-time suitability. This also justifies the possibility of the proposed strategy being integrated into the onboard electronic control units for real implementations.

It should be mentioned that the application scenarios of the proposed predictive energy management framework can also adapt to the change of vehicle models or mission profiles. For example, with the help of an online-learning enhanced Markov speed predictor [27] and an adaptive battery SoC reference generator, a PEMS has been designed for a midsize plug-in FCHEV, which can optimally deplete battery energy under multiple driving patterns for better fuel economy [28]. Moreover, combine the fuzzy C-means clustering enhanced Markov predictor and the battery energy planning approach, an integrated PEMS is devised for a light-duty plug-in FCHEV dedicated to postal delivery [29], where the GPS-collected real-world driving data in urban routes has been used to verify the effectiveness of the proposed strategy. To sum up, the devised predictive energy management framework has good versatility, making it capable of adapting to multiple application scenarios.

Despite the progresses regarding the energy management strategies for fuel cell electric vehicles in this chapter, further intensive studies should be conducted to improve the energy distribution performance. Specifically, future works would concentrate on the following perspectives:

- This chapter only focuses on retarding the fuel cell degradation imposed by harsh power transients, whereas other factors that may compromise the durability of fuel cell systems are not considered, such as working at extremely high/low loadings, frequent start-stop cycling, etc. In future works, it is expected to systematically consider these degrading factors by quantifying them within the cost

function when making power-allocating decisions [30].

- Powertrain component sizing plays an important role in vehicle's drivability and economic performance. In future works, a co-optimization framework for fuel cell hybrid electric vehicles considering the component degradations will be developed, which can simultaneously optimize the sizing parameters and the vehicle's total ownership cost given the desired driving profiles.
- Due to the abundant historical driving database of the postal-delivery vehicles, the past driving experience is useful in guiding future energy distributions. Therefore, it is expected in future works to develop a data-driven approach (e.g. deep neural networks) to plan the future usage of onboard electricity energy for further improving the fuel economy performance when charge-depleting mode is involved.

Reference

[1]. T. Wang, Q. Li, X. Wang, Y. Qiu, M. Liu, X. Meng, J. Li, W. Chen, An optimized energy management strategy for fuel cell hybrid power system based on maximum efficiency range identification, *J. Power Sources*, Vol. 445, 2020, 227333.

[2]. Z. Hua, Z. Zheng, M.-C. Péra, F. Gao, Remaining useful life prediction of PEMFC systems based on the multi-input echo state network, *Appl. Energy*, Vol. 265, 2020, 114791.

[3]. Y. Zhou, A. Ravey, M.-C. Péra, A survey on driving prediction techniques for predictive energy management of plug-in hybrid electric vehicles, *J. Power Sources*, Vol. 412, 2019, Pages 480-495.

[4]. D. Tran, M. Vafaeipour, M.E. Baghdadi, R. Barrero, J. V. Mierlo, O. Hegazy, Thorough state-of-the-art analysis of electric and hybrid vehicle powertrains: Topologies and integrated energy management strategies, *Renew. Sust. Energ. Rev.*, Vol. 119, 2020, 109596,

[5]. A. Ravey, B. Blunier, A. Miraoui, "Control Strategies for Fuel-Cell-Based Hybrid Electric Vehicles: From Offline to Online and Experimental Results," in *IEEE T. Veh. Technol.*, vol. 61, no. 6, pp. 2452-2457, 2012.

[6]. Y. Huang, H. Wang, A. Khajepour, H. He, J. Ji, Model predictive control power management strategies for HEVs: A review, *J. Power Sources*, Vol. 341, 2017, Pages 91-106.

[7]. A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, J. Schmidhuber, "A Novel Connectionist System for Unconstrained Handwriting Recognition," in *IEEE T. Pattern. Anal.*, vol. 31, no. 5, pp. 855-868, May 2009.

[8]. Liu, K., Asher, Z., Gong, X., Huang, M., Kolmanovsky, I. "Vehicle Velocity Prediction and Energy Management Strategy Part 1: Deterministic and Stochastic Vehicle Velocity Prediction Using Machine Learning," *SAE Technical Paper*, 2019-01-1051, 2019.

[9]. C. Sun, X. Hu, S. J. Moura, F. Sun, "Velocity Predictors for Predictive Energy Management in Hybrid Electric Vehicles," in *IEEE T. Contr. Syst. T.*, vol. 23, no. 3, pp. 1197-1204, May 2015.

[10]. ADVISOR Advanced Vehicle Simulator. http://adv-vehicle-sim.sourceforge.net/.

[11]. Y. Zhou, A. Ravey, M.-C. Péra, Multi-mode predictive energy management for fuel cell hybrid electric vehicles using Markov driving pattern recognizer, *Appl. Energy*, Vol. 258, 2020, 114057.

[12]. D. P. Filev, I. Kolmanovsky, "Generalized Markov Models for Real-Time Modeling of Continuous Systems," in *IEEE T. Fuzzy Syst.*, vol. 22, no. 4, pp. 983-998, Aug. 2014.

[13]. X. Huang, Y. Tan, X. He. An intelligent Multifeature Statistical Approach for the Discrimination of Driving Conditions of a hybrid electric vehicle. in *IEEE T. Intell. Transp.*, Vol. 12, No. 2, Jun. 2011.

[14]. R. Zhang, J. Tao, H. Zhou. Fuzzy Optimal Energy Management for Fuel Cell and Supercapacitor Systems Using Neural Network Based Driving Pattern Recognition. in *IEEE T. Fuzzy Syst.*, Vol. 27, No. 1, Jan. 2019.

[15]. Z. Chen, L. Li, B. Yan, C. Yang, C.M. Martinez, D. Cao. Multimode Energy Management for Plug-In Hybrid Electric Buses Based on Driving Cycles Prediction. in *IEEE T. Intell. Transp.*, Vol. 17, No.10, Oct. 2016.

[16]. Q. Zhang, W. Deng, G. Li. Stochastic Control of Predictive Power Management for Battery/Supercapacitor Hybrid Energy Storage Systems of Electric Vehicles. in *IEEE T. Ind. Inform.*, Vol. 14, No.7, Jul. 2018.

[17]. A. Ravey, N. Watrin, B. Blunier, D. Bouquain, A. Miraoui, "Energy-Source-Sizing Methodology for Hybrid Fuel Cell Vehicles Based on Statistical Description of Driving Cycles," in *IEEE T. Veh. Technol.*, vol. 60, no. 9, Nov. 2011.

[18]. Guzzella L, Sciarretta A. Vehicle Propulsion systems: introduction to modeling and optimization. *Berlin: Springer-Verlag*; 2005.

[19]. M.C. Péra, D. Hissel, H. Gualous, C. Turpin. Electrochemical Components, *John Wiley & Sons, Inc*, 2013.

[20]. V.H. Johnson, Battery performance models in ADVISOR, *J. Power Sources*, Vol. 110, Issue 2, 2002, Pages 321-329.

[21]. Y. Zhou. Predictive energy management for fuel cell hybrid electric vehicle. Other. Université Bourgogne Franche-Comté, 2020. English. ⟨NNT : 2020UBFCA020⟩. ⟨tel-03080574⟩.
[22]. C.H. Zheng, G.Q. Xu, Y.I. Park, W.S. Lim, S.W. Cha, Prolonging fuel cell stack lifetime based on Pontryagin's Minimum Principle in fuel cell hybrid vehicles and its economic influence evaluation, *J. Power Sources*, Vol. 248, 2014.
[23]. H. J. Ferreau et al., qpOASES User's Manual, Version 3.2 [Online], Apr. 2017. *Available at*: https://github.com/coin-or/qpOASES/blob/master/doc/manual.pdf.
[24]. C. Liu, H. Bai, S. Zhuo, X. Zhang, R. Ma, F. Gao, "Real-Time Simulation of Power Electronic Systems Based on Predictive Behavior," in *IEEE T. Ind. Electron*., vol. 67, no. 9, pp. 8044-8053, Sept. 2020.
[25]. Bai H, Liu C, Ma R, Paire D, Gao F. Device-level modelling and FPGA-based real-time simulation of the power electronic system in fuel cell electric vehicle. *IET Power Electron.* 2019;12(13):3479–87.
[26]. MicroAutoBox II: Compact and robust prototyping system for in-vehicle applications [Online]. *Available at:* https://www.dspace.com/en/pub/home/products/hw/micautob/microautobox2.cfm.
[27]. Y. Zhou, A. Ravey, M.-C. Péra, "A Velocity Prediction Method based on Self-Learning Multi-Step Markov Chain," *45th Annual Conference of the IEEE Industrial Electronics Society*, Lisbon, Portugal, 2019, pp. 2598-2603.
[28]. Y. Zhou, A. Ravey, M.-C. Péra, Multi-objective energy management for fuel cell electric vehicles using online-learning enhanced Markov speed predictor, *Energy Convers. Manag.*, Vol. 213, 2020, 112821.
[29]. Y. Zhou, H. Li, A. Ravey, M.-C. Péra, An integrated predictive energy management for light-duty range-extended plug-in fuel cell electric vehicle, *J. Power Sources*, Vol. 451, 2020, 227780.
[30]. Y. Zhou, A. Ravey, M.-C. Péra, Real-time cost-minimization power-allocating strategy via model predictive control for fuel cell hybrid electric vehicles, *Energy Convers. Manag.*, Vol. 229, 2021, 113721.