Stochastic Model Predictive Control for Energy Management of Power-Split Plug-in Hybrid Electric Vehicles Based on Reinforcement Learning

Zheng Chen[1,3**], Hengjie Hu[1], Yitao Wu[1], Yuanjian Zhang[2], Guang Li[3], and Yonggang Liu[4*]
[1]Faculty of Transportation Engineering, Kunming University of Science and Technology, Kunming, 650500, China
[2]Sir William Wright Technology Center, Queen's University Belfast, Belfast, BT9 5BS, United Kingdom
[3]School of Engineering and Materials Science, Queen Mary University of London, London, E1 4NS, United Kingdom
[4]State Key Laboratory of Mechanical Transmissions & School of Automotive Engineering, Chongqing University, Chongqing, 400044, China
Email: chen@kust.edu.cn, huhengjie1995@163.com, yitaowumail@gmail.com, y.zhang@qub.ac.uk, g.li@qmul.ac.uk, andylyg@umich.edu
Correspondence: Yonggang Liu (andylyg@umich.edu), and Zheng Chen (chen@kust.edu.cn).

Abstract: In this paper, a stochastic model predictive control (MPC) method based on reinforcement learning is proposed for energy management of plug-in hybrid electric vehicles (PHEVs). Firstly, the power transfer of each component in a power-split PHEV is described in detail. Then an effective and convergent reinforcement learning controller is trained by the Q-learning algorithm according to the driving power distribution under multiple driving cycles. By constructing a multi-step Markov velocity prediction model, the reinforcement learning controller is embedded into the stochastic MPC controller to determine the optimal battery power in predicted time domain. Numerical simulation results verify that the proposed method achieves superior fuel economy that is close to that by stochastic dynamic programming method. In addition, the effective state of charge tracking in terms of different reference trajectories highlight that the proposed method is effective for online application requiring a fast calculation speed.

Keywords: energy management strategy, reinforcement learning, Markov chain, velocity prediction, stochastic model prediction control.

NOMENCLATURE

Abbreviations		*Symbols*	
PHEV	plug-in hybrid electric vehicle	F_{total}	total fuel consumption
HEV	hybrid electric vehicle	F_{rate}	fuel rate
AER	all-electric range	ω_{eng}	speed of petrol engine
EMS	energy management strategy	ω_{mot}	speed of motor
CD/CS	charge depletion/charge sustaining	T_{eng}	torque of petrol engine
SOC	state of charge	T_{mot}	torque of motor
DP	dynamic programming	P_{req}	vehicle request power
PMP	Pontryagin's minimum principle	P_{final}	final drive power
GA	genetic algorithm	P_{mot}	motor power
CV	convex optimization	P_{eng}	engine power
DDP	deterministic dynamic programming	P_{bat}	battery power

SDP	stochastic dynamic programming	P_{acc}	accessory power
HIL	hardware-in-the-loop	η_{fianl}	drive efficiency factor of final drive
ECMS	equivalent consumption minimization strategy	η_{gear}	drive efficiency factor of gear
MPC	model predictive control	η_c	transmission efficiency of motor
RL	reinforcement learning	OCV	open-circuit voltage
RBFNN	radial basis function neural network	C_{bat}	battery capacity
WT	wavelet transform	i_{bat}	battery current
SMPC	stochastic model predictive control	R_{int}	battery internal resistor
BPNN	back propagation neural network	SOC_{init}	battery initial SOC
SQP	sequential quadratic programming	β	gear ratio of ring gear and sun gear
AI	artificial intelligence	p	transition probability
ML	machine learning	$X_{i,j}^{k}$	number transiting from i to j at speed v_k
OOL	optimal operation line	X_{i}^{k}	total number for transfers of i at speed v_k
MD	Markov decision	$P_{s\to s'}^{a}$	probability of state transferring from s to s' by executing action a
QL	Q-learning	γ	discount factor
MC	Monte Carlo	η	learning efficiency
RMSE	root mean square error	F_i	total number of possible next step transitions of
QL-SMPC	QL-based SMPC	F_{ij}	number of transiting from a_i to a_j
SQL-SMPC	single-step Markov-based SMPC	$Erri(t)$	mean error between the predicted speed and the real speed
MQL-SMPC	multi-step Markov-based SMPC	$SOC_{line}(k)$	linear SOC reference value
		SOC_{low}	SOC terminal value at the end of driving

I. INTRODUCTION

Nowadays, reckless mining of fossil energy and increasing power demand incur massive greenhouse gas emission and environmental pollution, and enormous efforts have been devoted to investigating energy saving and emission reduction technologies [1]. For road transportation industry, electrification provides an effective solution to mitigate negative impact imposed by traditional fossil driving solutions [2]. Plug-in hybrid electric vehicles (PHEVs) have been becoming attractive promising alternative solution due to the improved fuel economy and reduction of emission, compared to traditional fossil fuel vehicles. Different from conventional hybrid electric vehicles (HEVs), PHEVs can be directly charged from the power grid, thereby attaining an extra all-electric range (AER) and further mitigating fuel consumption of engine. In PHEVs, two and more energy sources are deployed to supply driving power individually or jointly, and energy delivery routes can be transmitted and optimized to reduce fuel consumption [3]. Presently, the gasoline engines are employed as the main driving source in PHEVs, the electric energy storage devices such as batteries and supercapacitor are

employed as the auxiliary power sources [4]. In addition, the fuel cell electric vehicles attracting wide attention have been turning into a research hotspot due to their zero emission characteristics, and compatibility coupling with electric energy storage devices to drive vehicles [5-6]. Nonetheless, the difficulties underlying transition and output power ratio among different energy routes complicate design of energy management strategies (EMSs), which need wide research attention from academia and industry.

The control target of EMSs includes reduction of fuel consumption and emission and extension of service life of energy storage systems (usually lithium-ion batteries) [7]. In the literature, EMSs of PHEVs can be divided into two main types: rule-based methods and optimization-based methods [8]. Rule-based methods, characterized by simple structure, ease implementation as well as reliable and stable controlling performance, are generally composed of the predefined logical relationship and fuzzy rules. As a typical solution, charge depletion (CD)/charge sustaining (CS) method has been commonly implemented in real applications. In the CD stage, PHEVs hold to release the stored battery energy until the battery state of charge (SOC) drops to a certain threshold, then the driving mode switches to the CS stage, and the vehicle attempts to maintain the SOC at a given threshold, of which the working state is similar with that appears in HEVs [9]. The CD/CS method shows satisfactory performance of fuel saving and emission reduction in a short-term driving scenario (such as the all-electric driving); however, as the target driving distance becomes longer, the fuel consumption will obviously increase due to the simple sustaining optimization in the CD stage. In short, rule-based methods mainly rely on engineering experiences, and the optimal control cannot be ensured all the time. For this reason, research on EMSs progressively turns to optimization-based methods.

Optimization-based methods can be divided into two main categories: global optimization based and instantaneous optimization based. To achieve global optimization, it is necessary to know the detailed information of driving schedule in advance. Then, the optimal control theory is imposed to optimally allocate the energy among multiple power sources to attain global optimization. Dynamic programming (DP) [10], Pontryagin's minimum principle (PMP) [11], genetic algorithm (GA) [12] and convex optimization (CV) [13] are typical representative candidates. Unfortunately, all of them are difficult to implement online due to the time-varying driving conditions; nevertheless, these offline solutions can be employed as evaluation criteria for other methods or be served as optimal knowledge for rule extraction and development of online algorithms. DP is a mature algorithm that can be roughly divided into two categories: deterministic DP (DDP) and stochastic DP (SDP), which are distinguished according to whether the global disturbance (usually the required driving

power) is known in advance. In [14], DP is exploited to extract the rules of engine on/off command, gear shifting and torque distribution. By combining with the K-means clustering algorithm, a blended method considering driving conditions is proposed to achieve better fuel economy and faster calculation speed. In [15], DP is employed to enable the energy allocation of a series-parallel PHEV, and a recalibration method based on optimized rules is proposed and verified by the hardware-in-the-loop (HIL) experiment to manifest the improvement of fuel economy. In [16], DP is employed for solving the energy distribution of multi hybrid energy storage vehicles including fuel cells, lithium-ion batteries and supercapacitors. The DP based strategy with a multiple-grained speed prediction method is proposed, and the HIL experiment is conducted to verify the effectiveness of the proposed method. Different from DDP, SDP shows strong adaptability to different driving conditions with limited driving knowledge in advance, where the demanded power of historical driving cycles or standard driving schedules are regarded as a stochastic model with Markov property. In a random Markov process, a probabilistic model of driving condition model is established, and then the energy control problem can be solved by the SDP [17] .The resulting output is usually a determined table for state control, which can be applied online [18]. In [19], the SDP is applied in a parallel HEV, and its controlling performance is close to that of DDP. Even though, compared with instantaneous optimization algorithms, SDP still requires too much calculation intensity, hindering its online application potential.

Instantaneous optimization algorithms, as the name implies, can instantly optimize energy distribution of powertrain in PHEVs and commonly guarantees a local optimum. In general, the energy management performance by instantaneous optimization algorithms is inferior to that by global optimized method. Representative instantaneous optimization algorithms, such as equivalent consumption minimization strategy (ECMS) [20], model predictive control (MPC) [21], reinforcement learning (RL) (usually it can be applied online) [22], have been widely employed to search the optimal result instantaneously or in a short horizon with different control targets. Among them, ECMS, grounded on the PMP theory, usually transfers the battery power to equivalent fuel consumption and thus converts the multi-dimensional optimization problem into an instant single optimization issue. As an effective real-time algorithm, MPC forms close-loop system according to the inner three processes of prediction model, rolling optimization and feedback correction and conducts optimization control at each moment to ensure the optimality and real-time performance in control time domain [23]. Ref. [24] builds up a novel hierarchy MPC based energy management framework, in which the radial basis function neural network (RBFNN) and wavelet transform (WT) are employed cooperatively to achieve the

speed prediction, and the MPC is applied to derive preferable fuel economy online. Ref. [25] establishes a Markov chain model for driving power demand, and then a stochastic MPC (SMPC) is proposed for instantaneous and predictive energy management of HEVs. In [26], a Markov speed prediction model is deployed, and DP is applied in the rolling optimization of SMPC. The HIL experiment validation indicates that the SMPC method can lead to better energy savings. In [27], a multi-objective strategy is proposed based on a global fast SOC planning, and the back propagation neural network (BPNN) is employed to predict the vehicle speed. Then, the direct configuration method and sequential quadratic programming (SQP) are employed as the optimization scheme in the controlling horizon. The simulation results validate the effectiveness of proposed method in mitigating operating costs.

More recently, with the rapid explosion of artificial intelligence (AI) technology, machine learning (ML) has been incrementally explored and exploited in energy management field. RL, a fundamental ML algorithm, has been operated in various applications such as robot control [28], transport [29] and other intelligent systems. In particular, RL exhibits better online and optimal performance in the energy management of PHEVs. In [30], an adaptive method based on RL is proposed, and its optimality of RL is justified. In [31], a RL-based method is proposed to reasonably distribute the power flow between the battery and super-capacitor, by which not only the energy consumption can be effectively reduced, but also the maximum discharge current of battery can be limited. Ref. [32] reveals that the RL-based method can yield better fuel performance while ensuring safe operation of battery, compared with the ECMS. In [33], a deep Q-learning method is proposed, and the results manifest that the convergence and training speed of the proposed method is faster than that of the conventional Q-learning algorithm, and the fuel economy improvement can reach 89% of that by DP.

Based on the above literature review, it can be found that most of the MPC algorithms attempt to follow the designated curve (such as SOC trajectories), which is solved by the global optimization algorithm, in a short-term rolling horizon [34], and occasionally, the computational efficiency can be low. In addition, to the authors' knowledge, RL is rarely applied in MPC for real-time energy management controller design [35]. Motived by this, the study designs a SMPC controller based on RL to improve the fuel economy of PHEV. To achieve this, the power distribution of a typical power-split PHEV is analyzed in detail first. Subsequently, to construct the stochastic model in the SMPC controller, different driving scenarios are selected, and the Markov chain model is built to pave the way for the application of one typical RL algorithm, i.e., the Q-learning method. Then, two diverse Markov speed predictors, namely single-step and multi-step Markov speed predictor, are involved in the

controller to conduct speed prediction. By comparing the predictive error, the multi-step Markov speed prediction method with better accuracy is selected as the stochastic prediction model. Finally, numerical simulation validations are conducted to evaluate the fuel economy improvement, and the computation speed of the proposed method and its applicability in different SOC reference trajectories are analyzed. The main contributions of this study are attributed to the following two aspects: 1) The RL is employed to resolve online rolling control optimization required by SMPC. 2) An RL based controller incorporating speed prediction is established, and it provides an effective support for online application of ML based energy management strategies of PHEVs.

The rest of this paper is structured as follows. Section II introduces the dynamic system structure of studied power-split PHEV and elaborates the mathematical analysis. In Section III, the principle of Q-learning is demonstrated, and a convergent Q-learning controller is constructed. Section IV introduces and establishes a framework of SMPC based on the RL. In Section V, the effectiveness, adaptability and scalability of the proposed method is verified. Section VI draws the main conclusions and findings of this study.

II. MODELING OF PHEV

A power-split PHEV is taken as a case study in this paper, and its powertrain framework is shown in Fig. 1. The vehicle powertrain consists of a petrol engine, a lithium-ion battery pack, two electric motors, a planetary gear set, a final drive and two electric convertors. Among them, two motors can be employed as driving motors to propel the vehicle or as used as generators to charge the battery pack. The corresponding parameters are listed in Table I. In this study, the main purpose is to minimize the fuel consumption of the PHEV in a certain driving range, and the objective function can be expressed as:

$$J = \min F_{total} = \min \int_0^T F_{rate} dt \tag{1}$$

$$F_{rate} = f(\omega_{eng}, T_{eng}) \tag{2}$$

where F_{total} represents the total fuel consumption of whole driving process, F_{rate} is the fuel rate, T is the entire driving schedule, ω_{eng} and T_{eng} denote the speed and torque of petrol engine, respectively. For the sake of minimizing the engine fuel consumption, the relevant factors affecting the instantaneous fuel consumption of the engine needs to be analyzed in detail.

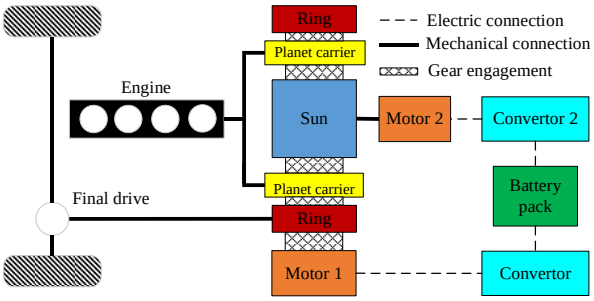


Fig. 1. The power-split PHEV powertrain framework.

Table I. Vehicle parameters.

Units	Parameters	Value
Vehicle	Mass	1801 kg
Lithium-ion battery	Rated capacity	39 Ah
	Rated voltage	220 V
Motor 1	Peak power	50 kW
	Rated power	25 kW
Motor 2	Peak power	30 kW
	Rated power	15 kW
Engine	Peak power	57 kW
Planet gear set	Sun gear	30
	Ring gear	78

As shown in Fig. 1, the required power requirement at wheel is supplied by the engine and two motors, of which the latter power is obtained from the battery pack. The detailed relationships can be expressed as:

$$\begin{cases} P_{req} = P_{final} \cdot \eta_{final} \\ P_{final} = (P_{mot1} + P_{mot2} + P_{eng}) \cdot \eta_{gear} \\ P_{bat} = (P_{mot1}/\eta_{c1} + P_{mot2}/\eta_{c2}) + P_{acc} \\ \quad = \omega_{mot1} T_{mot1}/\eta_{c1} + \omega_{mot2} T_{mot2}/\eta_{c2} + P_{acc} \end{cases} \tag{3}$$

where P_{req}, P_{final}, P_{mot1}, P_{mot2}, P_{eng} and P_{bat} represent the power of vehicle chassis, final drive, motor 1, motor 2, engine and battery, respectively. P_{acc} denotes the accessory power, which is supposed to be a constant value: 220 W in this study. η_{fianl} and η_{gear} denote the drive efficiency factor of final drive and gear. η_{c1} and η_{c2} express the transmission efficiency of motors 1 and 2, respectively. ω_{mot1}, ω_{mot2}, T_{mot1} and T_{mot2} are the speed and torque of two motors. In addition, a simplified equivalent circuit model, including an internal resistor and an ideal voltage source, is elicited to characterize the battery's electric performance. On this basis, P_{bat}, the battery current i_{bat}, and the battery SOC $SOC(t)$ can be formulated as:

$$\begin{cases} P_{bat} = OCV \cdot i_{bat} - {i_{bat}}^2 R_{\text{int}} \\ i_{bat} = OCV - \sqrt{{OCV}^2 - 4R_{\text{int}} P_{bat}}\,/2R_{\text{int}} \\ SOC(t) = SOC_{init} - 1/C_{bat} \int_0^t i_{bat} dt \end{cases} \tag{4}$$

where OCV is the open-circuit voltage, C_{bat} is the parallel connected capacitance, R_{int} is the internal resistance, and SOC_{init} denotes the initial SOC. In this simplified model, OCV and R_{int} are attained by the interpolation with SOC, as shown in Fig. 2. As can be seen, OCV varies from 165 V to 220 V with the increase of SOC, and R_{int} ranges from 0.09 ohm to 0.14 ohm.

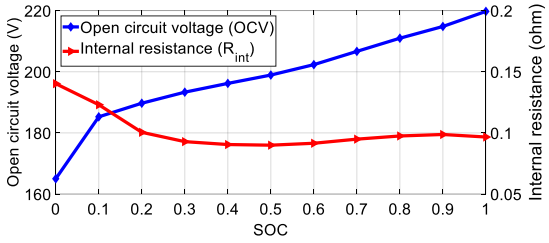


Fig. 2. OCV and R_{int} variation with SOC.

The engine and two motors are dynamically coupled via a planetary set. According to its working principle, the relationship between the engine and the two motors can be described as:

$$\begin{cases} \omega_{eng} = (\beta/1+\beta)\omega_{mot2} + (1/1+\beta)\omega_{mot1} \\ T_{eng} = (1+\beta)T_{mot1} = (1+1/\beta)T_{mot2} \end{cases} \tag{5}$$

where β denotes the gear ratio of ring gear and sun gear. According to (2) to (5), we can find that the control degrees of freedom of energy management problem is two, causing difficulty of directly solving it. To mitigate the complexity, the engine optimal operation line (OOL), as shown in Fig. 3, is introduced to simplify the control design [36]. The OOL of engine denotes the optimal engine speed with respect to a determined power output, and under the current group of power and speed, the fuel efficiency is highest among all the combinations. The corresponding mathematical relation can be formulated, as:

$$\omega_{eng} = g^*(P_{eng}) \tag{6}$$

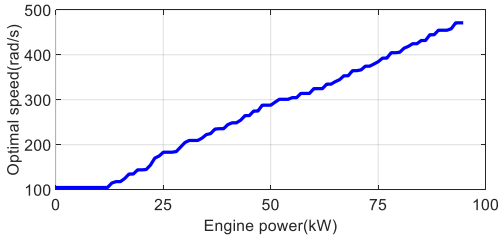


Fig. 3. Optimal operating point at different engine power.

According to the above analysis, we may conclude that only if the battery power is given, the engine power and the corresponding instantaneous fuel rate can be determined. From this point of view, the battery power is the only controlling input determining the fuel rate. Due to the power limitations and performance requirement

of each part, the optimization problem is subjected to the following constraints:

$$\begin{cases} P_{bat\_\min} \le P_{bat} \le P_{bat\_\max} \\ P_{mot1\_\min} \le P_{mot1} \le P_{mot1\_\max} \\ P_{mot2\_\min} \le P_{mot2} \le P_{mot2\_\max} \\ P_{eng\_\min} \le P_{eng} \le P_{eng\_\max} \\ P_{req\_\min} \le P_{req} \le P_{req\_\max} \\ SOC_{\min} \le SOC \le SOC_{\max} \end{cases} \tag{7}$$

where the subscripts of min and max denote the minimum and maximum values of corresponding variables, respectively.

Based on the above energy relationship, a novel RL-based SMPC method is proposed to determine the optimal battery power at each moment and consequently ensure the reduction of fuel consumption in the whole driving range, which is to be presented in the next Section.

III. Reinforcement Learning Application

To apply the RL to SMPC, it is imperative to theoretically analyze the RL principle, and then an effective RL controller can be established to optimize the SMPC process.

A. Markov Chain Model

As well known, RL is built on the Markov decision (MD) theory, and the Markov chain supplies the fundamental framework k for the MD [30]. Actually, the required propelling power in practice can be treated as a random process with Markov properties, and the required power state at the next moment of the vehicle is only related to the current power state, independent of the historical state [32]. To obtain the state transition matrix, different driving cycles such as SC03, REP05, LA92, US06, WLTC and JC08 are merged and employed as the cycle database, trying to involve different driving conditions. The selected cycles stand for highway, suburb, urban and crowded urban roads, including low-, medium- and high-speed driving conditions and therefore exhibit the random characteristics of most traffic environment. The speed profile and the required power profile are shown in Fig. 4. It can be found that the vehicle speed ranges from 0 m/s to 35 m/s, and the required power ranges from -100 kW to 100 kW. The maximum likelihood estimation method is exploited to estimate the transfer probability of the required power at different speed, as:

$$\begin{cases} p_{i,j}^{k} = \dfrac{X_{i,j}^{k}}{X_{i}^{k}} \\ \sum_{j=1}^{n} X_{i,j}^{k} = X_{i}^{k} \end{cases} \tag{8}$$

where $p_{i,j}^{k}$ is the transition probability of required power from P_{req}^{i} to P_{req}^{j} at speed v_k. $X_{i,j}^{k}$ denotes the number transiting from P_{req}^{i} to P_{req}^{j} at speed v_k, and X_{i}^{k} represents the total number for variation of P_{req}^{i} at speed v_k.

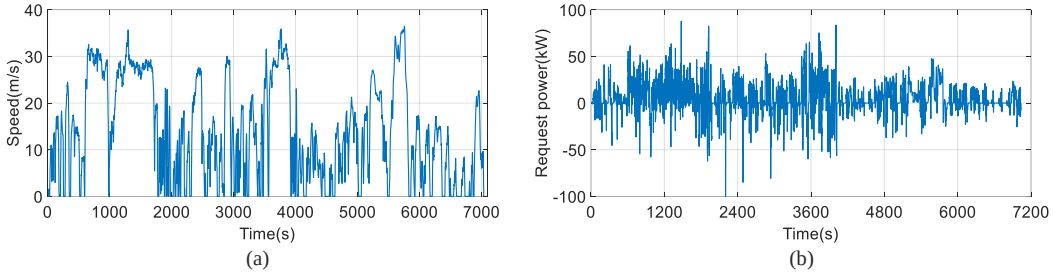


Fig. 4. The speed profile and the required power profile of training cycles.

The required power transition probability at different speed is shown in Fig. 5. As can be seen, the transition of the required power occurs between two adjacent states, and obviously the probability of large variation of the required power is small.

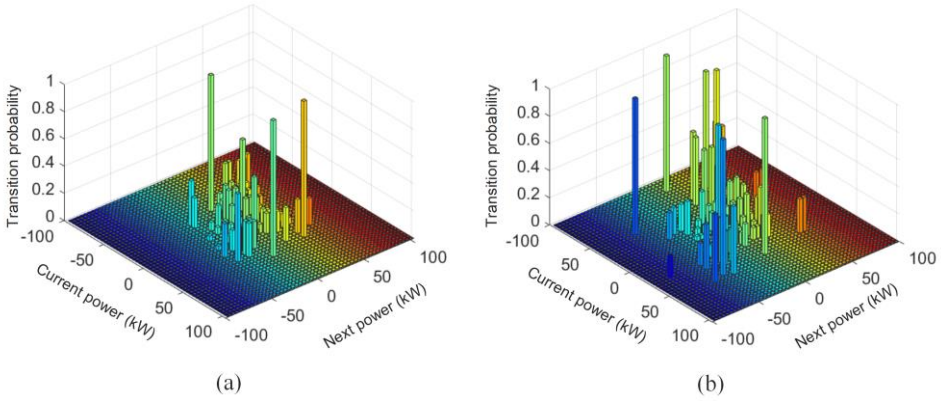


Fig. 5. Examples of required power transition probability at different speed. (a) v=25km/h; (b) v=50km/h.

B. RL Algorithm

As a profoundly popular branch of machine learning, RL is widely employed to solve optimal solution problem in the control field. The main RL mechanism is that the agent perceives the surrounding environment (i.e. the controller object) and performs reasonable actions to interact with the controller object, so as to maximize the benefits of agent. As a popular candidate of RL methods with an intelligible structure and a feature of easy implementation, Q-learning (QL) algorithm has been widely applied. The main content of QL is to build a Q table that can be directly iterated and optimized via the state-action pairs, and performs direct action selection according to the iteratively updated Q value to obtain the desired benefit result. The QL algorithm is employed as the main method for training the QL controller to attain the consequent optimization of SMPC.

In this study, the QL table can be described as a five-element group $\{S, A, P^a_{s\to s'}, \gamma, R\}$. Among them, the state variables S of the model include the required power P_{req}, vehicle speed v and battery SOC; the control action A is battery power; $P^a_{s\to s'}$ represents the probability of state transfer from s to s' by executing control action a; γ denotes the discount factor in the learning process; and R is the reward function, which is defined as the negative number of fuel consumption at each moment. The corresponding relationship can be expressed as:

$$\begin{cases} s = \left\{ P_{req}(t), v(t), SOC(t) \right\} \in S \\ a = \left\{ P_{bat}(t) \right\} \in A \\ r(t) = \left\{ -F_{rate}(t) \right\} \in R \end{cases} \tag{9}$$

The QL-based strategy is a mapping function from state to action $\pi : S \to A$; that said, as long as a state s is given, the current action can be determined according to the strategy $a = \pi(s)$. For each state, the value function is defined as the sum of the mathematical expectations of the discounted reward function, as:

$$V^*(s) = \max_{a\in\pi} E\left(\sum_{t=0}^{T} \gamma^t r_t \right) \tag{10}$$

Based on the Bellman principle, equation (10) can be reformulated as:

$$V^*(s) = \max_{a\in\pi} r(s) + \gamma \sum_{s'\in S} p^a_{s\to s'} V^*(s') \tag{11}$$

For a given state s, the Q function $Q(s,a)$ is defined as the expectation of the total number of discount rewards for the agent when performing the action a and the follow-up policy in this state. The relationship between value functions V and Q can be described as:

$$\begin{cases} Q(s,a) = r(s,a) + \gamma \sum_{s'\in S} p^a_{s\to s'} V^*(s') \\ V^*(s) = \max_{a\in A} Q(s,a) \end{cases} \tag{12}$$

Then, the optimal Q function in the QL algorithm can be expressed as:

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s'\in S} p^a_{s\to s'} \max_{a'} Q^*(s',a') \tag{13}$$

The update rule of Q function is:

$$Q(s,a) \leftarrow Q(s,a) + \eta\left(r + \gamma \max_{a} Q(s',a) - Q(s,a) \right) \tag{14}$$

where η is the learning efficiency. The higher value η, the faster convergence speed of the QL method will

be. Nevertheless, an extreme high learning efficiency will lead to overfitting [37]. Fig. 6 shows the mean error of Q value under different SOC at the same speed with a discount factor of 0.9 and a maximum number of iterations of 10000. The mean error can be expressed as:

$$Mean_{error} = \sum_{N}^{N+100} \left(\left| r + \gamma \max_{a} Q(s',a) - \eta Q(s,a) \right| \right) \Big/ 100 \tag{15}$$

According to (15), the average increment of the Q value after every 100 iterations is calculated, as depicted in Fig. 6. With the increase of iteration operation, the mean error of Q value gradually decreases towards 0, indicating the convergence of the QL method. It can be also found that when the controller starts the learning process, the exploration process is employed to expand the leaning samples and enrich the information of the reward function. After a certain period of learning, the learned sample information is applied to optimize the control action, and the optimal control strategy is attained after the final convergence.

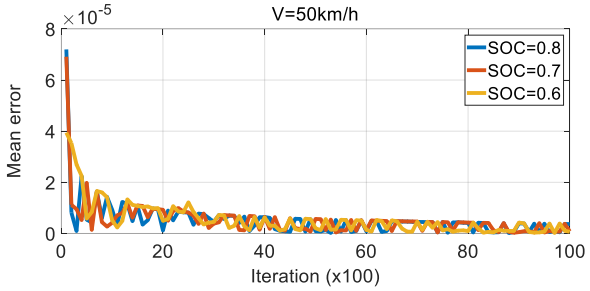


Fig. 6. The mean error of Q value.

After a convergent QL controller is yielded, a SMPC control framework is introduced in the next section, and the trained QL controller is incorporated into the SMPC optimization process.

IV. Application of SMPC in Energy Management Strategy

MPC is an online optimization control method based on the concept of receding horizon. In cooperation with the local optimization, the rolling optimization mechanism is exploited in prediction time domain. It advances preferable robustness, strong stability and near-optimal control performance in dealing with linear or nonlinear problems. As a special MPC framework, the exclusive difference between SMPC and traditional MPC is the stochasticity of the prediction model. Specifically, the traditional MPC is based on a fixed prediction model; whereas for the SMPC, a random prediction target is taken into account, and the stochastic process prediction is applied to update the prediction model. The SMPC framework proposed in this study is shown in Fig. 7. First, the Markov Monte Carlo method is employed to construct a random speed prediction model. Then, based on the previously introduced QL controller, the SOC curve obtained by the QL offline controller is taken

as the reference trajectory for the SMPC, and the input information of the stochastic prediction model is optimized by the QL controller. Finally, the control of the first second in the prediction horizon is imposed to the PHEV model after feedback correction.

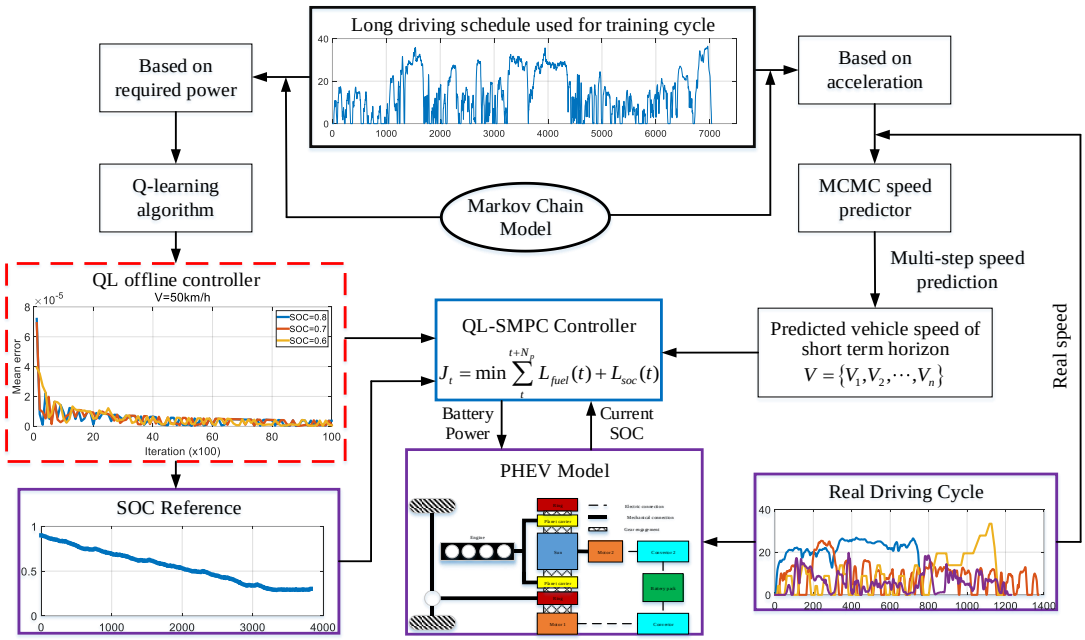


Fig. 7. The proposed SMPC framework.

A. *Markov Based Speed Prediction*

The vehicle acceleration varies arbitrarily due to external environment and driver's operations during actual driving. It can be understood that the acceleration change is only related to the current acceleration, independent of the historical acceleration information. In other words, it shows the characteristic of Markov properties. Similarly, the acceleration of the vehicle is also regarded as a stochastic process with Markov properties, and in this study, the short-term speed prediction process is performed by the Monte Carlo (MC) algorithm [38]. To integrate the QL controller, six standard driving cycles, as shown in Fig. 4, are employed as the training set, and the corresponding acceleration profiles are shown in Fig. 8. It can be found that the acceleration, ranging from -4 to 4 m/s^2, includes rapid accelerations, rapid decelerations and normal driving, thus effectively involving different conditions of acceleration. Note that in the process of acceleration discretization, excessive sparse interval of deviation leads to distorted acceleration profiles; and instead an extremely narrow one can raise much more computational labor. To ensure the accuracy of speed prediction without a heavy computational burden, a tradeoff of 80 discrete points is determined. Additionally, the maximum likelihood estimation is employed to reckon the state transition probability. The single-step and the multi-step state transition probability are calculated by:

$$\begin{cases} F_i = \sum_{j=1}^{N} F_{ij} \\ p_{ij} = \dfrac{F_{ij}}{F_i} \end{cases} \tag{16}$$

where F_i represents the total number of possible next step transitions of a_i, F_{ij} denotes the number of transiting from a_i to a_j, and p_{ij} is the probability of the acceleration shifting from a_i to a_j.

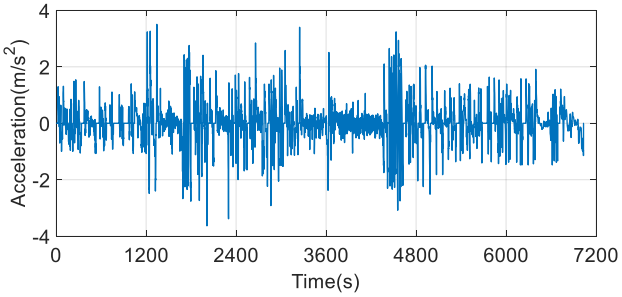
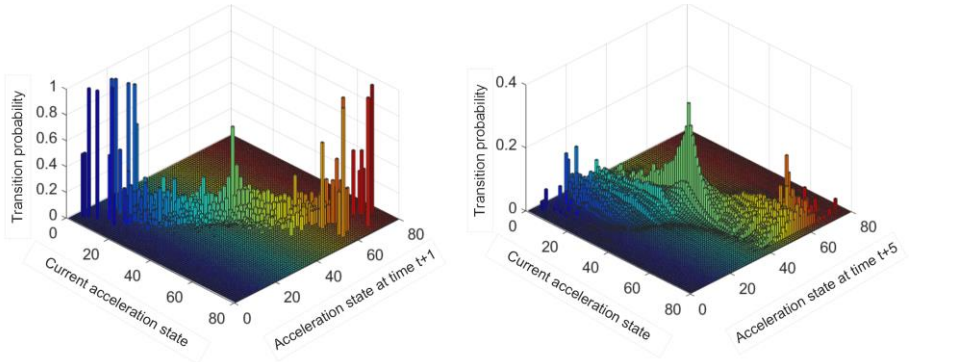
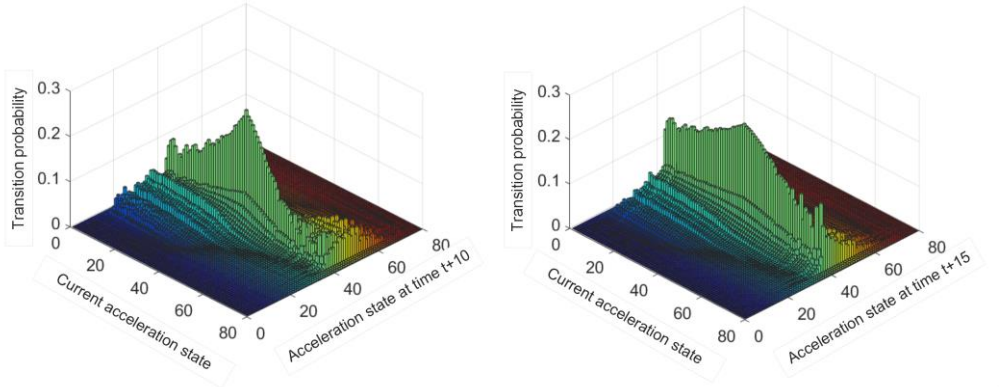


Fig. 8. The acceleration profile of training cycle.

The acceleration transition probability in different steps are exhibited in Fig. 9. The probability of one-step state transition matrix is basically diagonally distributed, indicating few deviations between adjacent state interval. In the multi-step state transition matrix, thanks to the longer prediction time, the diagonal characteristics are less obvious, and the state transition trends to be scattered. The reason is that as the number of steps increases, the probability of wider state variation increases, thus enabling more dispersed probability distribution.



(a)



(b)

Fig. 9. The acceleration transition probability with different steps. (a) One-step and five-step acceleration transition probability; (b) Ten-step and fifteen-step acceleration transition probability.

In this study, the root-mean-square error (RMSE) is adopted as the evaluation index to verify the precision of speed prediction, as:

$$\begin{cases} RMSE = \sum_{t=1}^{T} Erri(t) \Big/ T \\ Erri(t) = \sqrt{\sum_{i=1}^{t_p} (v_{t,i}^{*} - v_{t,i})^2 \Big/ t_p} \end{cases} \tag{17}$$

where T represents the total duration of the predicted driving schedule, t_p is the predictive horizon, $v_{t,i}^{*}$ is the ith predicted speed, $v_{t,i}$ denotes the ith real speed, and $Erri(t)$ is the mean error between the predicted speed and the real speed in the tth time domain.

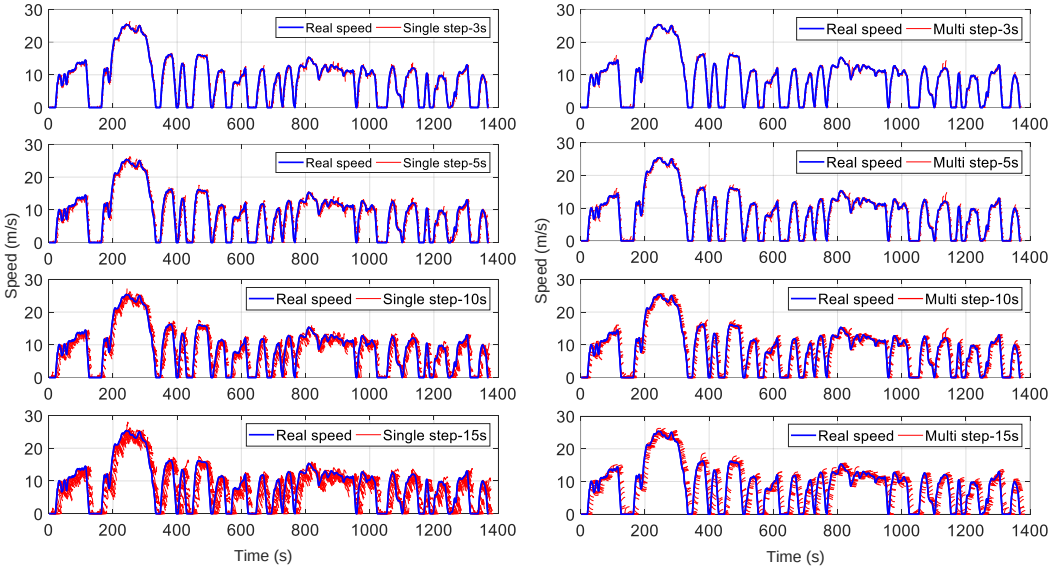


Fig. 10. The results of Markov speed prediction. (a) Single-step method; (b) Multi-step method.

Fig. 10 shows the results of single-step and multi-step Markov speed prediction for under different prediction horizons. We can find that with the extension of prediction horizon, the deviation between predicted speed and actual speed becomes more noticeable. For example, when the prediction horizon is 15 s, the forecast error apparently appears. From this point of view, it can be confirmed that the terminal deviation by a single-step Markov method is better than that of multi-step Markov method. To further validate the accuracy of multi-step prediction method. Table II shows the RMSE of four driving cycles, including HWFET, NEDC, WVUSUB and UDDS, in different prediction horizons under two Markov speed prediction methods. As detailed in Table II, the prediction results of multi-step method are better than those of single-step method under different test cycles, especially in high-speed driving cycles: HWFET and WUVSUB. As such, the multi-step Markov method

is eventually taken as the stochastic prediction algorithm. Besides, by trading off the computational ability and the prediction accuracy, the prediction horizon is selected to be 10 s in this study.

Table II. Markov speed prediction results.

Prediction accuracy	HWFET			NEDC		
	Single-step	Multi-step	Improved accuracy	Single-step	Multi-step	Improved accuracy
3s	0.5489	0.4822	12.15%	0.7258	0.6627	7.42%
5s	0.7899	0.6807	13.82%	1.0635	0.9833	7.54%
10s	1.3200	1.0945	17.08%	1.8978	1.7636	7.07%
15s	1.8773	1.4520	22.65%	2.6923	2.5473	5.39%
Prediction accuracy	WVUSUB			UDDS		
	Single-step	Multi-step	Improved accuracy	Single-step	Multi-step	Improved accuracy
3s	0.7222	0.6291	12.89%	0.9890	1.9662	2.3%
5s	1.0694	0.9086	15.04%	1.4705	1.4219	3.3%
10s	1.8554	1.5372	17.15%	2.5508	2.4724	3.1%
15s	2.5390	2.1180	16.58%	3.4924	3.3957	2.8%

B. Rolling Optimization Process Based on RL

In this study, the state equation of SMPC can be expressed as:

$$x(t+1) = f\left(x(t), u(t), w(t)\right) \tag{18}$$

where $x(t) = SOC(t)$ is the state variable, $u(t) = P_{bat}(t)$ represents the control variable, and $w(t)$ is considered as the system stochastic disturbance, i.e., the predictive speed. It should be pointed out that considering the random disturbance caused by the prediction speed error, the prediction horizon of the SMPC controller is with the same length as that of the control horizon. The rolling optimization criterion J_t in each predictive horizon can be expressed as:

$$J_t = \min \sum_{t}^{t+N_p} L_{fuel}(t) + L_{soc}(t) \tag{19}$$

where J_t is the optimization criterion in prediction time domain, $L_{fuel}(t) = F_{rate}(t)$ denotes the instantaneous fuel consumption function at each step, and $L_{soc}(t)$ is the cost for the SOC penalty. Considering the SOC deviation from the reference trajectory at step t, it can be formulated as:

$$L_{soc}(t) = \begin{cases} 0 & SOC(t) > SOC_{ref}(t) \\ \alpha\left(SOC(t) - SOC_{ref}(t)\right)^2 & SOC(t) < SOC_{ref}(t) \end{cases} \tag{20}$$

where α represents the negative weighting factor. Note that the purpose of specifying the battery SOC here is to ensure that the real SOC trajectory follows the referred curve. The realization process of the QL-based SMPC (QL-SMPC) is designed as follows:

(1) Calculate the predicted speed sequence $v_{t+1}, v_{t+2}, \ldots, v_{t+N_p}$ based on the multi-step Markov speed prediction model.

(2) Calculate the required power sequence $P_{req,t+1}, P_{req,t+2}, \ldots, P_{req,t+N_p}$ in the prediction horizon based on the predicted speed sequence $v_{t+1}, v_{t+2}, \ldots, v_{t+N_p}$.

(3) Employ the QL controller to incorporate the prediction speed sequence, the required power and the SOC reference trajectory for achieving the online receding-horizon optimization of QL-SMPC, of which the detailed optimization process is shown in Table III. The matrix in the QL controller is denoted as $Q_{original}(s,a)$, and the matrix participating in rolling optimization at each step is denoted as Q_{new}. Note that in Table III, the learning process of each step in the proposed rolling optimization is based on the QL controller, and all actions in the predictive horizon are employed to maximize the reward function in the current step, as formulated in (21).

(4) Set the first control element in the predictive horizon to the PHEV model after feedback correction.

Table III. The rolling optimization implementation process of QL-SMPC.

The rolling optimization process:
1. Extract the matrix in the QL-SMPC: $Q_{original}(s,a)$, S, A and R
2. The predictive speed, required power and SOC reference are numbered to obtain the serial numbers $s_t, s_{t+1}, \ldots s_{t+N_p}$
3. Put forward all action sequence corresponding to the serial number from $Q_{original}(s,a)$, and create a new learning matrix Q_{new}.
4. For $k = t : t + N_p$
Initialization state $s = k$
Perform a selected action a_k based on the ε-greedy algorithm.
Modify the instantaneous reward by equation (20): $r_{soc} = L_{soc}$.
Update the Q_{new}:
$$Q_{new}(s(t),a(t)) \leftarrow Q_{new}(s(k),a(k)) + \eta\left[\sum_{j=i}^{N_p} r_{soc}(k) + \max Q_{new}(s(k),:) - Q_{new}(s(k),a(k))\right] \quad (21)$$
$s = k + 1$
End

In the next step, the simulation is conducted, and the validation analysis is performed to verify the feasibility of the proposed algorithm.

V. Simulation Results and Analysis

In this paper, all the simulations are conducted based on Autonomie, which is a vehicle simulation software developed by Argonne National Laboratory [39]. Four standard driving schedules, i.e., HWFET, UDDS, NEDC and a real driving schedule in Kunming, China (denoted by KM), as shown in Fig. 11, are adopted to represent the most typical driving conditions. A sequential combination of these driving cycles, that is, 5 consecutive HWFET, 6 consecutive UDDS, 8 consecutive NEDC and 12 consecutive KM (simplified as 5 HWFE, 6 UDDS, 8 NEDC and 12 KM) are constructed as long cycles to evaluate the control effect of the proposed algorithm.

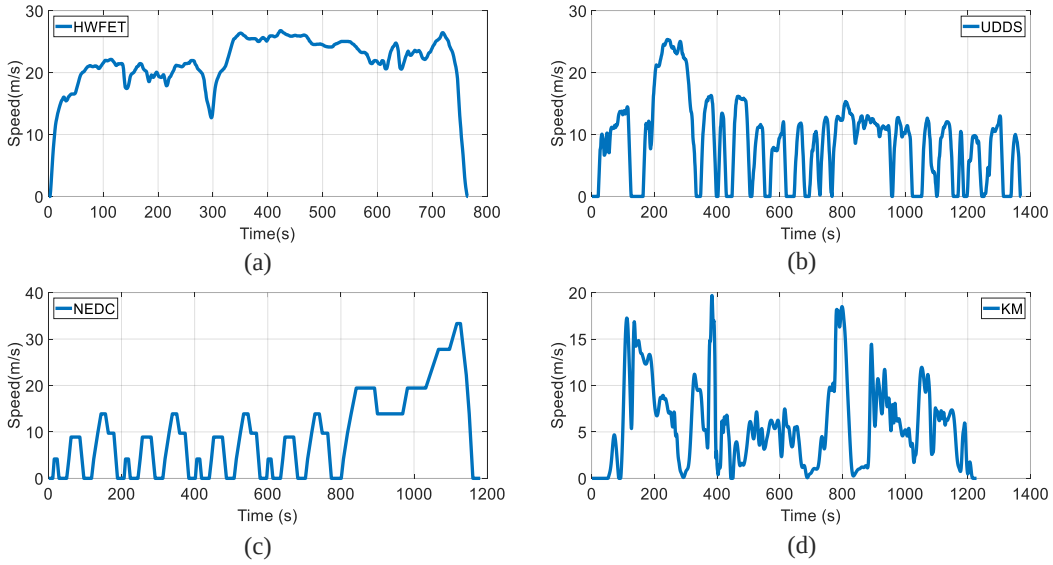


Fig. 11. Four testing cycles. (a) HWFET; (b) UDDS; (c) NEDC; (d) KM.

The simulation results of this paper are divided into four parts. Firstly, the SDP method is regarded as a evaluation benchmark. Compared with the SDP method and the QL method, the effectiveness and adaptability of the proposed method can be properly evaluated. Then, based on the two different speed prediction methods, the impact of speed prediction accuracy is addressed in terms of the controlling performance of the proposed method. Next, considering the influence of different SOC reference trajectories in practical application, two different SOC reference trajectories are employed to expand the application of the proposed method. Finally, the computational efficiency of the proposed method is analyzed to assess practical application potential.

The SDP algorithm is considered as an energy control method with great adaptability to various driving conditions. Based on the Markov Chain model established in Section III, the SDP strategy based on the modified policy iteration method is taken as a benchmark example, of which the target function can be formulated as:

$$J = \min \int_0^T F_{rate}(s_t, a_t) dt \tag{22}$$

In the SDP strategy, the setting of state and control variables remains almost the same with those of the QL controller, and the only difference lies in that the cost function of SDP method is in the instantaneous time domain.

A. *Analysis of Fuel Economy*

Table IV shows the comparison of the fuel consumption with SOC correction under different strategies [40]. It can be observed that the proposed method is close to the global SDP strategy, and the fuel consumption of the proposed method is 0.7%, 3.1%, 0% and 3.5% higher than those of the SDP method under four testing cycles. In addition, the same fuel consumption can be achieved under 8 NEDC, compared with the SDP result.

Furthermore, in comparison with the QL algorithm, the SOC results of the proposed method are similar to those of the QL method, and the fuel economy is better than the QL method most of time, except under 12 KM cycles. Fig. 12 sketches the SOC profiles of 5 HWFET and 6 UDDS under different strategies, and it can be found that the proposed method can track the reference trajectory effectively.

Table IV. Comparison of fuel consumption and engine performance.

Driving cycles	Method	Fuel consumption (kg)	Ending SOC	Fuel saving (%)
5 HWFET	SDP Method	1.8597	0.3024	-
	QL Method	1.8492	0.3024	-0.6
	Proposed method	1.8785	0.3052	+0.7
6 UDDS	SDP Method	1.1478	0.2960	-
	QL Method	1.1750	0.2958	+2.4
	Proposed method	1.1896	0.2992	+3.1
8 NEDC	SDP Method	1.8317	0.3133	-
	QL Method	1.8200	0.3131	-0.6
	Proposed method	1.8310	0.3133	+0
12 KM	SDP Method	1.2897	0.2887	-
	QL Method	1.3386	0.2887	+3.7
	Proposed method	1.3388	0.2907	+3.5

Note: +: Increment; -: Reduction.

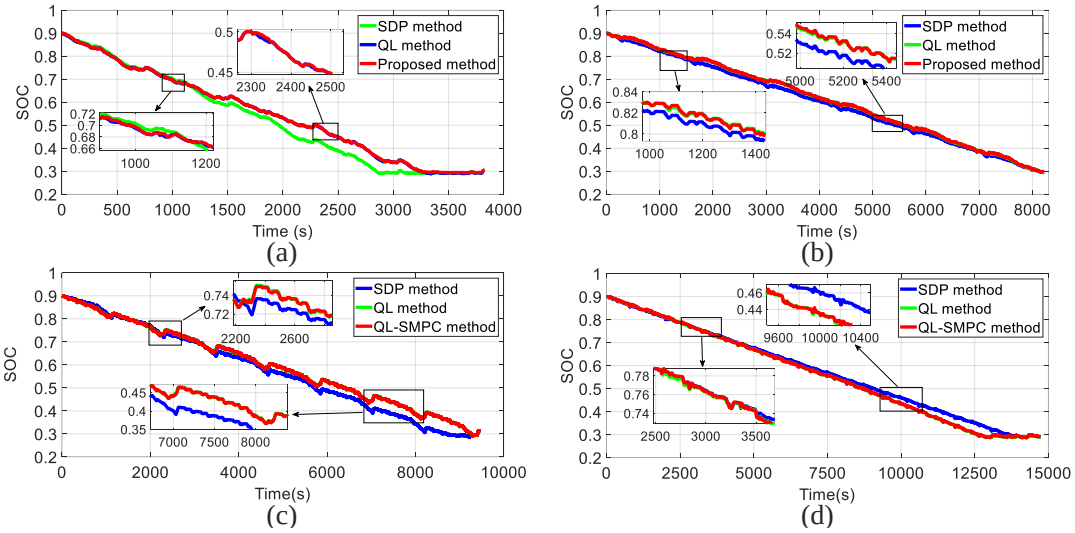


Fig. 12. The SOC profiles under four testing cycles. (a) 5 HWFET; (b) 6 UDDS; (c) 8 NEDC; (d) 12 KM.

To further verify the efficacy of the proposed method, Fig. 13 shows the engine efficiency when driving on the four test cycles. We can observe that the engine efficiency of the proposed method is close to that by the SDP method and that by the QL method. All the three methods can enable the engine to work in more efficient region, and try to avoid the engagement in the low torque area, thus improving the vehicle's overall fuel economy. Table V shows the engine working rate regarding the three methods. The engine on/off frequency of the proposed method is similar to that of the QL method and lower than that of the SDP method. This is caused by the negative value of the reward function for instantaneous fuel consumption in the QL controller. In this case, when the engine is off, only the battery pack supplies the propelling power, and the fuel consumption is zero for the QL control at this point, thus maximizing the reward. Due to the learning principle of RL, the

learning tends to attain more awards. Thus, the possibility of larger reward function value will increase, and the engine tends to remain off. Similarly, the engine on/off frequency based on the propose algorithm is less than that by the SDP method. On the other hand, this can, to some extent, explain why the fuel consumption of two methods becomes similar.

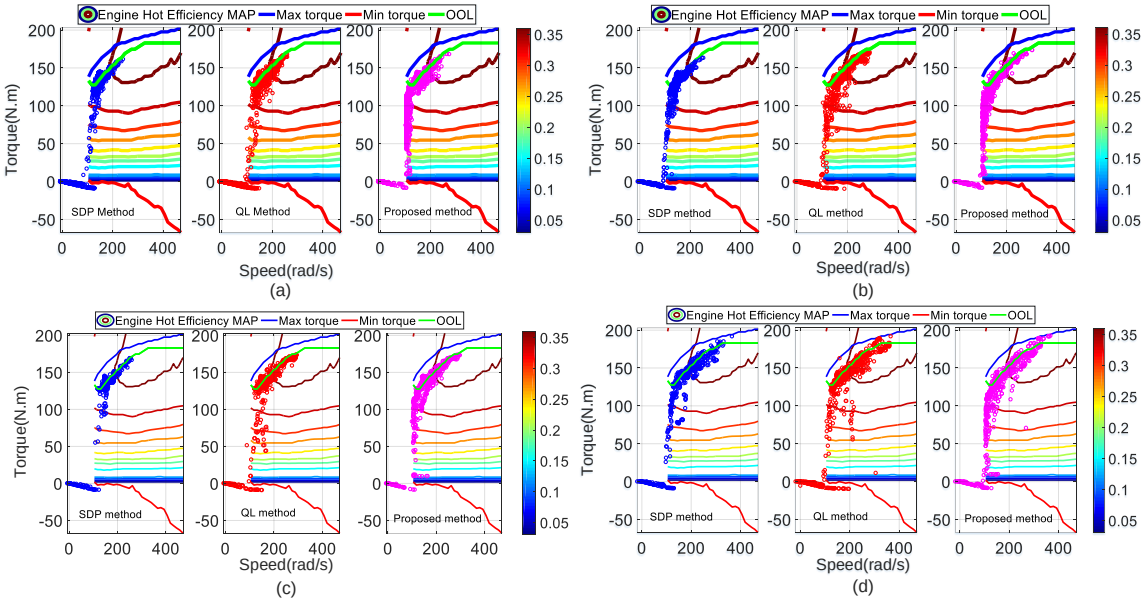


Fig. 13. The engine efficiency under four test cycles. (a) 5 HWFET; (b) 6 UDDS; (c) 8 NEDC; (d) 12 KM.

Table V. Comparison of engine working rate.

Driving cycles	Method	Working rate (%)	Driving cycles	Method	Working rate (%)
5 HWFET	SDP Method	40.84	8 NEDC	SDP Method	14.5
	QL Method	34.82		QL Method	11.31
	Proposed method	41.75		Proposed method	11.37
6 UDDS	SDP Method	12.46	12 KM	SDP Method	5.8
	QL Method	10.23		QL Method	5.1
	Proposed method	11.56		Proposed method	5.48

In summary, by comparing with the SDP method and QL method, the proposed method is verified effective and robust in terms of fuel economy, engine efficiency and engine working rate, under different driving cycles.

B. Fuel Consumption Comparison of Speed Predictive Accuracy

For the sake of comparing the fuel economy influenced by the speed prediction accuracy, one single-step Markov speed prediction method and multi-step Markov speed prediction method are integrated into the SMPC controllers as different prediction models discussed in Section IV. In this part of comparison, all the other settings remain the same. The fuel consumption by the two predictive models is shown in Fig. 14. As can be found, the fuel consumption of single-step Markov-based SMPC (SQL-SMPC) controller is more than that of multi-step Markov-based SMPC (MQL-SMPC) controller in these four test cycles. The MQL-SMPC can effectively increase the fuel economy by 0.6%, 0.4%，0.4% and 0.6%, respectively. In short, it can be found that the higher the accuracy of speed prediction in advance is, the better the fuel economy will be.

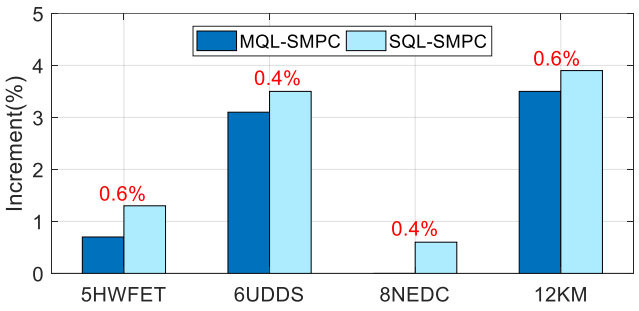


Fig. 14. Influence of prediction accuracy on fuel consumption.

C. Tracking Effect Analysis of Different SOC Reference Trajectories.

The proposed method of this paper is based on the QL controller. To extend the application range, two different SOC references are employed to further validate the learning effect of the proposed method. Here, two SOC curves generated by the SDP and an ideal linear time averaging method are taken as the references, of which the latter can be formulated as:

$$SOC_{line}(k) = SOC_{init} - \frac{k}{T}(SOC_{init} - SOC_{low}) \tag{23}$$

where $SOC_{line}(k)$ denotes the linear SOC reference value at step k; SOC_{low} is the SOC terminal value at the end of driving, and is set to 0.3 in this study. Fig. 15 shows the tracking effect of two SOC references under 5 HWFET and 6 UDDS cycles. From Fig. 15 (a), we can find that although the decline trend of two SOC references is quite different, the proposed method can perform the SOC tracking with high accuracy. As observed from Fig. 15 (b), the decline trend of the SDP reference and the ideal linear time reference remains nearly consistent, and it can also be easily found that from the two partially enlarged figures, the SOC curve of proposed method is always close to the SOC reference, thus manifesting the effectiveness of proposed controller.

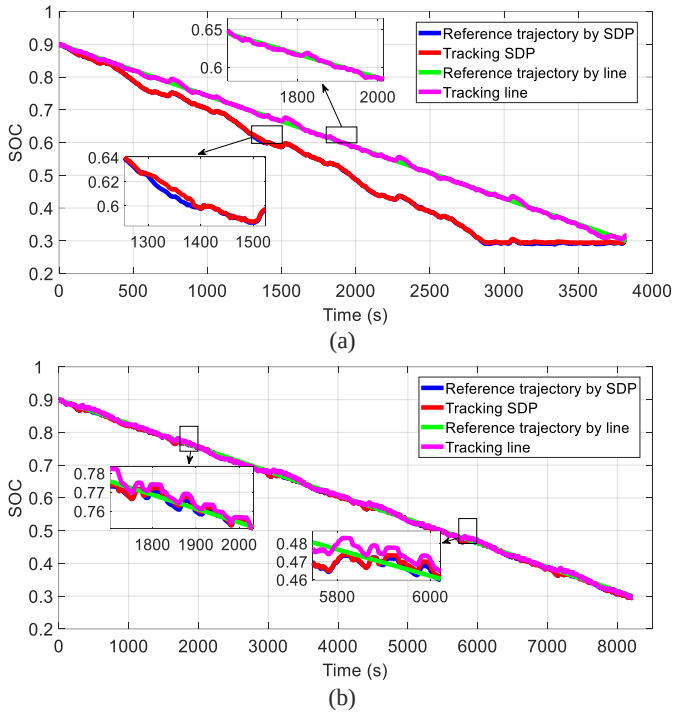


Fig. 15. The following effect of different SOC references. (a) Five HWFET. (b) Six UDDS.

Table VI compares the fuel consumption based on different SOC references. As can be found, the proposed method can achieve preferable fuel economy under both SOC references. In comparison with the SDP method, the fuel consumption of different SOC references increases by 1.7% and 2% under 5 HWFET cycles and by 3% and 3.3% under 6 UDDS cycles. Moreover, the fuel consumption based on the SDP reference is lower than that based on the linear method. The reason can be explained as follows. The SDP enables the global sub-optimality; and on this account, the SOC reference generated by the SDP method also highlights the characteristics of the global sub-optimality to certain extent. In contrast, the ideal linear SOC reference is simply an average curve and does not feature any global optimal or sub-optimal characteristics.

Table VI. Comparison of fuel consumption under different SOC references.

Driving cycles	SOC reference	Ending SOC (%)	Fuel consumption (kg)	Fuel saving (%)
5 HWFET	SDP (no reference)	30.24	1.8597	-
	SDP reference	30.51	1.8962	+1.7
	Line reference	31.63	1.9231	+2
6 UDDS	SDP (no reference)	29.60	1.1478	-
	SDP reference	29.88	1.1826	+3
	Line reference	29.74	1.188	+3.3

Note: +: Increment; -: Decrement.

D. *Analysis of Computational Efficiency*

In this study, the computation is performed in Matlab/Simulink through a computer with 8GB RAM and a core i5 processor @ 2.6GHz. It should be mentioned that the calculation time does not include the time of the QL controller, but only accounts for the computational load of the speed prediction and the QL-SMPC

controller. The computation duration is listed in Table VII. The total computation time for the speed prediction and QL-SMPC at each step ranges from 35.9 ms to 57.15 ms, which is verified qualified in real-time application with 1 s as a step. Thus, it can be concluded that the proposed strategy shows certain potential of online implementation.

Table VII. Comparison of the calculation time under different cycles.

	Single step calculation time (ms)			
	5 HWFET	6 UDDS	8 NEDC	12 KM
Speed prediction	0.3	0.35	0.36	0.36
SMPC	35.6	56.8	39.9	38.4

VI. CONCLUSION

In this study, a stochastic model predictive control energy management strategy based on Q-learning is proposed to improve the fuel economy of plug-in hybrid electric vehicles. Firstly, different driving conditions are considered, and the propelling power is modeled by Markov Chain. The Q-learning algorithm is applied to train a convergent optimal reinforcement learning controller. Then, based on the stochastic characteristics of vehicle acceleration, the Markov stochastic speed prediction model is established. After verifying the speed predicting accuracy, the multi-step Markov speed prediction is applied to the stochastic model predictive control, and the Q-learning controller is employed in the rolling optimization process. Finally, a series of simulations are performed to evaluate the performance of the proposed controller. The simulation results manifest that the proposed algorithm can achieve similar fuel economy as that of the offline stochastic dynamic programming method. Furthermore, one single step calculation time of the proposed controller is less than 57.15 ms, indicating its real-time application potential.

Our future work will focus on further improving the fidelity of the speed prediction model and the reinforcement learning controller performance. In addition, hardware-in-the-loop experiments and real vehicle validation will be conducted to further improve the performance of the proposed method.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China (No. 2018YFB0104000), in part by the National Natural Science Foundation of China (No. 61763021 and No. 51775063), and in part by the EU-funded Marie Skłodowska-Curie Individual Fellowships Project under Grant 845102-HOEMEV-H2020-MSCA-IF-2018. Moreover and most importantly, the authors would also like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] S. Zhang, W. Dou, Y. Zhang, W. Hao, Z. Chen, and Y. Liu, "A Vehicle-Environment Cooperative Control Based Velocity Profile Prediction Method and Case Study in Energy Management of Plug-in Hybrid Electric Vehicles," *IEEE Access,* vol. 7, pp. 75965-75975, 2019 2019, doi: 10.1109/access.2019.2921949.

[2] Y. Shang, N. Cui, B. Duan, and C. Zhang, "Analysis and Optimization of Star-Structured Switched-Capacitor Equalizers for Series-Connected Battery Strings," *IEEE Transactions on Power Electronics,* vol. 33, no. 11, pp. 9631-9646, Nov 2018, doi: 10.1109/tpel.2017.2787909.

[3] M. Sabri, K. Danapalasingam, and M. J. R. Rahmat, "A review on hybrid electric vehicles architecture and energy management strategies," *Renewable Sustainable Energy Reviews,* vol. 53, pp. 1433-1442, 2016.

[4] Y. Wang, L. Wang, M. Li, and Z. Chen, "A review of key issues for control and management in battery and ultra-capacitor hybrid energy storage systems," *eTransportation,* p. 100064, 2020.

[5] Y. Wang, Z. Sun, X. Li, X. Yang, and Z. Chen, "A comparative study of power allocation strategies used in fuel cell and ultracapacitor hybrid systems," *Energy,* vol. 189, Dec 15 2019, Art no. 116142, doi: 10.1016/j.energy.2019.116142.

[6] Y. Wang, Z. Sun, and Z. Chen, "Energy management strategy for battery/supercapacitor/fuel cell hybrid source vehicles based on finite state machine," *Applied Energy,* vol. 254, Nov 15 2019, Art no. 113707, doi: 10.1016/j.apenergy.2019.113707.

[7] X. Shu, G. Li, J. Shen, Z. Lei, Z. Chen, and Y. Liu, "An adaptive multi-state estimation algorithm for lithium-ion batteries incorporating temperature compensation," *Energy,* vol. 207, p. 118262, 2020/09/15/ 2020.

[8] J. Du, J. Chen, Z. Song, M. Gao, and M. Ouyang, "Design method of a power management strategy for variable battery capacities range-extended electric vehicles to improve energy efficiency and cost-effectiveness," *Energy,* vol. 121, pp. 32-42, 2017.

[9] S. Overington and S. Rajakaruna, "High-efficiency control of internal combustion engines in blended charge depletion/charge sustenance strategies for plug-in hybrid electric vehicles," *IEEE Transactions on Vehicular Technology,* vol. 64, no. 1, pp. 48-61, 2014.

[10] J. Hou and Z. Song, "A hierarchical energy management strategy for hybrid energy storage via vehicle-to-cloud connectivity," *Applied Energy,* vol. 257, p. 113900, 2020.

[11] X. Li, Y. Wang, D. Yang, and Z. Chen, "Adaptive energy management strategy for fuel cell/battery hybrid vehicles using Pontryagin's Minimal Principle," *Journal of Power Sources,* vol. 440, 2019.

[12] M. Wieczorek and M. Lewandowski, "A mathematical representation of an energy management strategy for hybrid energy storage system in electric vehicle and real time optimization using a genetic algorithm," *Applied energy,* vol. 192, pp. 222-233, 2017.

[13] X. Wu, X. Hu, X. Yin, L. Li, Z. Zeng, and V. J. J. o. P. S. Pickert, "Convex programming energy management and components sizing of a plug-in fuel cell urban logistics vehicle," *Journal of Power Sources,* vol. 423, pp. 358-366, 2019.

[14] Z. Lei, D. Qin, P. Zhao, J. Li, Y. Liu, and Z. Chen, "A real-time blended energy management strategy of plug-in hybrid electric vehicles considering driving conditions," *Journal of Cleaner Production,* vol. 252, 2020.

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

[16] Y. Wang, X. Li, L. Wang, and Z. Sun, "Multiple-grained velocity prediction and energy management strategy for hybrid propulsion systems," Journal of Energy Storage, vol. 26, Dec 2019, Art no. 100950, doi: 10.1016/j.est.2019.100950.

[17] Y. Du, Y. Zhao, Q. Wang, Y. Zhang, and H. Xia, "Trip-oriented stochastic optimal energy management strategy for plug-in hybrid electric bus," *Energy,* vol. 115, pp. 1259-1271, 2016.

[18] F. Qin, G. Xu, Y. Hu, K. Xu, and W. Li, "Stochastic optimal control of parallel hybrid electric vehicles," *Energies,* vol. 10, no. 2, p. 214, 2017.

[19] C.-C. Lin, H. Peng, and J. Grizzle, "A stochastic control strategy for hybrid electric vehicles," in *Proceedings of the 2004 American control conference*, 2004, vol. 5: IEEE, pp. 4710-4715.

[20] B. Geng, J. K. Mills, and D. Sun, "Energy management control of microturbine-powered plug-in hybrid electric vehicles using the telemetry equivalent consumption minimization strategy," *IEEE transactions on Vehicular Technology,* vol. 60, no. 9, pp. 4238-4248, 2011.

[21] J. Hou, J. Sun, and H. Hofmann, "Adaptive model predictive control with propulsion load estimation and prediction for all-electric ship energy management," *Energy,* vol. 150, pp. 877-889, 2018.

[22] Y. Wu, H. Tan, J. Peng, H. Zhang, and H. He, "Deep reinforcement learning of energy management with continuous control strategy and traffic information for a series-parallel plug-in hybrid electric bus," *Applied energy,* vol. 247, pp. 454-466, 2019.

[23] G. Li, J. Zhang, and H. He, "Battery SOC constraint comparison for predictive energy management of plug-in hybrid electric bus," *Applied Energy,* vol. 194, pp. 578-587, 2017.

[24] Z. Chen, N. Guo, J. Shen, R. Xiao, and P. J. I. A. Dong, "A hierarchical energy management strategy for power-split plug-in hybrid electric vehicles considering velocity prediction," *IEEE Access,* vol. 6, pp. 33261-33274, 2018.

[25] G. Ripaccioli, D. Bernardini, S. Di Cairano, A. Bemporad, and I. Kolmanovsky, "A stochastic model predictive control approach for series hybrid electric vehicle power management," in *Proceedings of the 2010 American*

Control Conference, 2010: IEEE, pp. 5844-5849.
[26] S. Xie, H. He, and J. Peng, "An energy management strategy based on stochastic model predictive control for plug-in hybrid electric buses," *Applied energy,* vol. 196, pp. 279-288, 2017.
[27] Y. Liu, J. Li, Z. Chen, D. Qin, and Y. Zhang, "Research on a multi-objective hierarchical prediction energy management strategy for range extended fuel cell vehicles," *Journal of Power Sources,* vol. 429, pp. 55-66, 2019.
[28] T. Kobayashi, "Student-t policy in reinforcement learning to acquire global optimum of robot control," *Applied Intelligence,* vol. 49, no. 12, pp. 4335-4347, 2019.
[29] R. C. Hsu, C.-T. Liu, and D.-Y. Chan, "A reinforcement-learning-based assisted power management with QoR provisioning for human–electric hybrid bicycle," *IEEE Transactions on Industrial Electronics,* vol. 59, no. 8, pp. 3350-3359, 2011.
[30] T. Liu, Y. Zou, D. Liu, and F. Sun, "Reinforcement learning of adaptive energy management with transition probability for a hybrid electric tracked vehicle," *IEEE Transactions on Industrial Electronics,* vol. 62, no. 12, pp. 7837-7846, 2015.
[31] R. Xiong, J. Cao, and Q. Yu, "Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle," *Applied Energy,* vol. 211, pp. 538-548, 2018.
[32] Z. Chen, H. Hu, Y. Wu, R. Xiao, J. Shen, and Y. Liu, "Energy management for a power-split plug-in hybrid electric vehicle based on reinforcement learning," *Applied Sciences,* vol. 8, no. 12, p. 2494, 2018.
[33] J. Wu, H. He, J. Peng, Y. Li, and Z. Li, "Continuous reinforcement learning of energy management with deep Q network for a power split hybrid electric bus," *Applied Energy,* vol. 222, pp. 799-811, 2018.
[34] Y. Wang, X. Wang, Y. Sun, and S. You, "Model predictive control strategy for energy optimization of series-parallel hybrid electric vehicle," *Journal of Cleaner Production,* vol. 199, no. PT.1-1130, pp. 348-358, 2018.
[35] Q. Zhou *et al.*, "Multi-step reinforcement learning for model-free predictive energy management of an electrified off-highway vehicle," *Applied Energy,* vol. 255, p. 113755, 2019.
[36] Z. Chen, B. Xia, C. You, and C. C. Mi, "A novel energy management method for series plug-in hybrid electric vehicles," *Applied Energy,* vol. 145, pp. 172-179, 2015.
[37] T. Liu, Y. Zou, D. Liu, and F. Sun, "Reinforcement learning–based energy management strategy for a hybrid electric tracked vehicle," *Energies,* vol. 8, no. 7, pp. 7243-7260, 2015.
[38] D. Gamerman and H. F. Lopes, *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. CRC Press, 2006.
[39] A. Aziz, M. S. Shafqat, M. A. Qureshi, and I. Ahmad, "Performance analysis of power split hybrid electric vehicles using autonomie," in *2011 IEEE Student Conference on Research and Development*, 2011: IEEE, pp. 144-147.
[40] C. Hou, M. Ouyang, L. Xu, and H. J. A. E. Wang, "Approximate Pontryagin's minimum principle applied to the energy management of plug-in hybrid electric vehicles," *Applied Energy,* vol. 115, pp. 174-189, 2014.