Research on Adaptive Control Strategy of Plug-in Hybrid Electric Vehicle Based on Internet of Vehicles Information

Chao Ma, Jianhui Chen, Hang Yin, Lei Cao and Kun Yang

Abstract—**In order to better improve the fuel economy of plug-in hybrid electric vehicle (PHEV), an adaptive control strategy is proposed with the application of traffic information obtained from internet of vehicles technology. Firstly, the P2-configuration PHEV simulation model is developed based on MATLAB/Simulink. Secondly, a virtual scenario based on SUMO is built to simulate internet of vehicles technology to obtain traffic information. Through the experimental vehicle speed compared with average Baidu API to extract the traffic speed, verify the validity of the virtual scene. Based on the extracted average traffic flow speed, approximate global driving condition is generated by the exponential weighted moving average method. Then, the SOC reference trajectory is generated by the dynamic programming (DP) algorithm based on the acquired approximate global driving condition information. PI control is employed to follow the SOC reference trajectory, enabling adjustment of the equivalent factor adaptively. Finally, the SUMO-MATLAB co-simulation platform is built to validate the effectiveness. It demonstrates that the adaptive equivalent fuel consumption minimization strategy (A-ECMS) with information of internet of vehicles saves 3.6% of fuel consumption compared with ECMS strategy without information of Internet of Vehicles (IoV). To verify the possibility of applying the proposed strategy to a vehicle, a Linux board that can acquire real-time road condition information is developed, applying real-time traffic information to the strategy. The experiment outcomes demonstrate that, in comparison to the ECMS strategy without IoV information, the proposed approach improves fuel efficiency by 3.8%.**

Index Terms—**PHEV, Energy management strategy, Traffic information, A-ECMS, Linux board.**

I. Introduction

With the depletion of fossil fuels and the worsening environmental conditions, the new energy vehicle has become the general trend of the development of China's automobile industry [1]. The Plug-in hybrid electric vehicle has the advantages of long driving range and good fuel saving effect. It has gradually become the research hotspot of new energy vehicle [2]. As one of the core technologies of PHEVs, energy management strategy has been widely studied by domestic and foreign scholars [3].

Energy management strategies for PHEVs are mainly divided into rule-based and optimization-based. Rule-based strategies primarily consist of methods like logical threshold and fuzzy logic, among others. It mainly relies on the experience of engineers and cannot adapt to variable driving condition and complex environment [4,5]. Optimization-based energy management strategies involve dynamic programming (DP) algorithm [6], pontryagin's minimum principle (PMP) [7], model predictive control (MPC) [8], and equivalent fuel consumption minimization strategy (ECMS) [9], etc. ECMS strategy is an optimization strategy derived from PMP. The knowledge of global driving conditions is unnecessary for the ECMS strategy. The global optimization is replaced by local optimization, applying in real-time because of the small computation complexity. Wang W [10,11] et al. used the firefly algorithm and particle swarm optimization algorithm to optimize the equivalent factor of ECMS strategy respectively. Compared with the traditional ECMS strategy, the fuel economy is effectively improved. Li P [12,13] et al. respectively proposed a way to generate reference SOC trajectories of battery using a fuzzy inference system (FIS) and recurrent neural network algorithm (RNN). The equivalent factor in the ECMS strategy is adaptively adjusted according to the error between the actual SOC and the reference SOC. Fuel consumption is further reduced.

Recently, with the emergence of internet of vehicles (IoV) technology and intelligent transportation system (ITS), the energy management strategy has provided new direction for improving fuel economy and vehicle performance [14]. Tang [15] et al. proposed a multi-objective hierarchical optimization (MOHO) strategy. The upper layer by integrating road speed limit, slope and traffic light information based on model predictive control speed planning strategy is developed. The lower layer realizes the optimal power allocation based on the A-ECMS strategy. Lei [16] et al. designed an A-ECMS strategy considering traffic information. Simulation and experimental results show that the fuel economy of PHEV is improved. Sun [17] et al. designed an A-ECMS method. Predicted speed and traffic information are used to adjust the equivalent factors in ECMS strategy, which shows good economy.

In summary, the ECMS strategy with fixed equivalent factor cannot adapt to the change of driving condition and complex environment. With the emergence of IoV and ITS technology, the vehicle can obtain real-time traffic information. Therefore, PHEV of P2 configuration is selected as

Manuscript received May 15, 2024; revised October 28, 2024. This work was supported by the National Natural Science Foundation of China under Grant 51605265.

Chao Ma is a professor of Shandong University of Technology, Zibo 255000, China (corresponding author to provide email: mc@sdut.edu.cn).

Jianhui Chen is a postgraduate student of Shandong University of Technology, Zibo 255000, China (e-mail: chenjianhui0105@163.com).

Hang Yin is a postgraduate student of Shandong University of Technology, Zibo 255000, China (e-mail: yinhang15589164351@163.com).

Lei Cao is a postgraduate student of Shandong University of Technology, Zibo 255000, China (e-mail: cl15264609309@163.com).

Kun Yang is a professor of Shandong University of Technology, Zibo 255000, China (e-mail: yangkun@sdut.edu.cn).

the focus of this study. An A-ECMS strategy is proposed, which considers initial SOC, travel distance and information of IoV. The energy saving potential of PHEV is further explored.

II. PHEV POWERTRAIN STRUCTURE AND MODELLING

A. PHEV system architecture and key parameters

The powertrain of PHEV with P2 configuration is mainly composed of engine, motor, power battery and transmission system. By controlling the switch of the clutch, it can be divided into five operating modes: engine drive, pure electric, hybrid drive, driving charge and regenerative braking. The system's structure is depicted in Figure 1. And Table I display the parameters for the PHEV and key components.

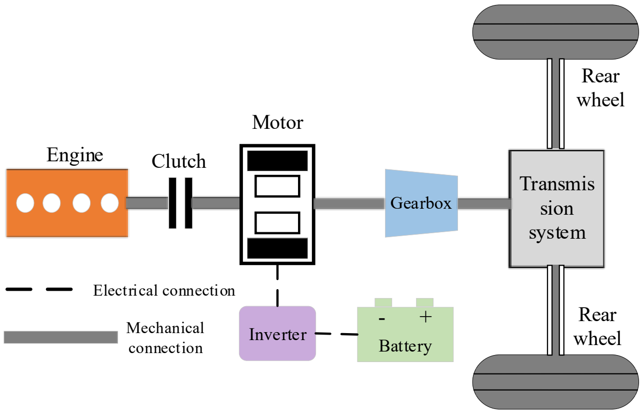

Fig. 1. Schematic diagram of PHEV powertrain structure

TABLE I
PARAMETERS OF THE PHEV AND KEY COMPONENTS

Parameter name	Value
Full-load mass m/kg	1575
Air drag coefficient C_D	0.31
Tire radius r/m	0.3
Windward area A/m^2	2
Rolling resistance coefficient f	0.013
Main reducer ratio i_0	3.8
Automatic transmission ratio i_g	3.6/2.24/1.4/1/0.82
Battery rated voltage /V	360
Battery capacity /Ah	33
Motor rated/peak power /kW	35/80
Motor rated/peak speed /rpm	3000/8000
Motor peak torque /Nm	260
Maximum engine power /kW	85
Maximum engine torque /Nm	165

B. Vehicle modelling

1) Vehicle dynamics model: The vehicle dynamics model can be constructed according to the longitudinal vehicle dynamics driving equation. When travelling on the road, longitudinal dynamics equation of the vehicle is written by:

$$\begin{aligned} F_t &= F_f + F_w + F_i + F_j \\ &= mgf\cos\alpha + \frac{C_D A v^2}{21.15} + mg\sin\alpha + \delta m\frac{du}{dt} \end{aligned} \tag{1}$$

where F_t is the driving force, F_f is the rolling resistance, F_w is the air resistance, F_i is the slope resistance, F_j is the acceleration resistance, α is the slope angle, δ is the rotating mass conversion factor, du/dt is the acceleration.

2) Engine model: In this paper, the engine model can be developed using the look-up table method. The data obtained from the bench test is interpolated and fitted. Taking the engine speed and torque as input, the instantaneous fuel consumption rate is interpolated. The fuel consumption rate can be represent using the following formulas:

$$\begin{cases} P_{ice} = \dfrac{T_{ice} n_{ice}}{9550} \\ b_{ice} = f(n_{ice}, T_{ice}) \\ Q_{ice} = \dfrac{P_{ice} b_{ice}}{3600} \end{cases} \tag{2}$$

where P_{ice} is the engine power, n_{ice} is the engine speed, T_{ice} is the engine torque, b_{ice} is the engine fuel consumption rate, Q_{ice} is the fuel consumption per unit time.

3) Drive motor model: Motor is modeled in a similar way to engine. The efficiency of the motor is also interpolated by the motor's torque and motor's speed. The motor has two operating modes of driving and generating according to the vehicle's requirement. The power calculation formulas in different modes are as follows:

$$\begin{cases} \eta_m = f(n_m, T_m) \\ P_m = \dfrac{T_m n_m}{9550\eta_m}, & T_m > 0 \\ P_m = \dfrac{T_m n_m \eta_m}{9550}, & T_m < 0 \end{cases} \tag{3}$$

where η_m is the motor efficiency, n_m is the motor speed, T_m is the motor torque, P_m is the motor power.

4) Power battery model: The power battery serves as the primary energy storage system for the PHEV. This study emphasizes PHEV fuel economy, disregarding the effects of complex chemical reactions and temperature variations. A simplified yet accurate equivalent circuit model is utilized to process modeling, as depicted in Figure 2.

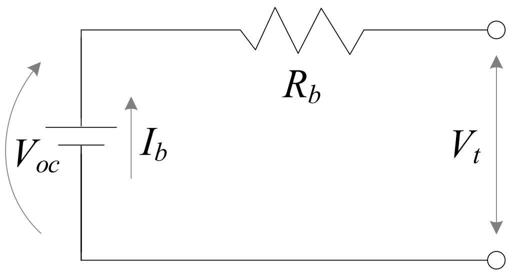

Fig.2. Battery equivalent circuit model

On the basis of Figure 2, the battery circuit current I_b is as follows:

$$I_b = \frac{V_{oc} - \sqrt{V_{oc}^2 - 4R_b P_b}}{2R_b} \tag{4}$$

where V_{oc} is the battery open-circuit voltage, I_b is the battery current, R_b is the battery internal resistance, P_b is the battery power.

Using the ampere-time integration method, the formula for calculating the battery SOC can be as follows:

$$SOC(t) = SOC_0 - \frac{\int_0^t (V_{oc} - \sqrt{V_{oc}^2 - 4R_b P_b})dt}{2R_b Q_b} \tag{5}$$

where SOC_0 is the initial SOC value of the battery, Q_b is the battery capacity.

III. Research on Global Traffic Information Generation Technology

With the emergence of IoV technology and ITS, it can record and upload the speed and location information of all vehicles on the road. The vehicle can obtain real-time road condition information through on-board navigation. Using traffic data and historical patterns, intelligent systems can predict future traffic conditions and pre-schedule the vehicle's energy use strategy. The PHEV operating mode is rationally arranged to achieve the best fuel economy.

A. Virtual simulation scenario construction based on SUMO

1) Road network model construction: To replicate the operation of vehicles under real-world road conditions, this paper selects Shandong University of Technology to Wenchang Lake Provincial Tourist Resort as the traffic simulation section. The experimental section covers various driving conditions such as urban road driving, high-speed road driving and urban road congestion.

SUMO provides a few ways to import and generate road network model, among which SUMO provides users with a free editable map OSM (OpenStreetMap). OSM is an open-source world mapping tool that allows people to freely edit and download. The road network of the selected area can be obtained by running the osmWebWizard.py script file using Python. Then the base road network file required from the simulation is generated according to the configured parameters.

Due to some inaccuracies and lags in the maps provided by OSM, it is necessary to repair and improve the road network model generated by OSM according to the actual road information. Netedit software is used to edit a missing number of lanes, road speed limit, road length, and partially inaccurate intersections. Eventually, the edited road network model is shown in Figure 3. The section marked by the blue solid line is the selected traffic simulation road. Compared the road network model with the enlarged satellite map, the road network model closely aligns with real-world road conditions, providing a solid foundation for traffic simulation.

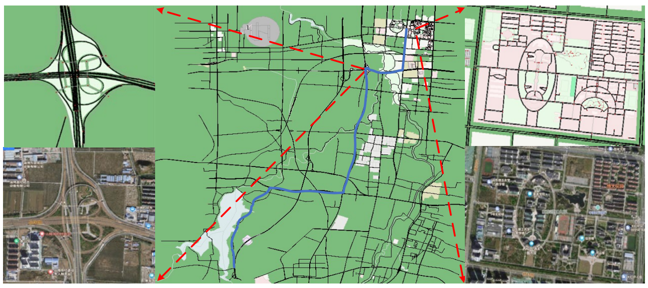

Fig.3. Road network model and actual satellite comparison

2) Traffic flow construction: At present, traffic data acquisition technology is mainly divided into two kinds: artificial investigation method and intelligent sensor device method. Artificial investigation method needs to invest a lot of human and material resources, and the cost is high. With the continuous improvement and promotion of urban intelligent transportation system, automated traffic data acquisition technology has become the mainstream application. The Baidu map API real-time road condition query service is calculated based on the analysis of big data collected by road condition information providers. The result is more accurate and can reflect the congestion situation on the road. Therefore, this paper uses the average traffic speed obtained from Baidu map API to configure the traffic flow.

The Baidu Map API provides an interface to query real-time traffic information, which can help developers check the congestion situation and trends of specific roads or areas. To reflect the situation of vehicle in traffic jams, speed data will be collected during the rush hour. The average traffic speed of the experimental section collected is shown in Figure 4. When there is less vehicle data on the road, real-time road condition information will not be returned. Therefore, when this happens, the average speed setting speed limit for road transportation. The experimental road section is congested in many places during the evening rush hour. Among them, serious congestion occurred at about 6-7 kilometers, and the average traffic speed was only 5.36 km/h.

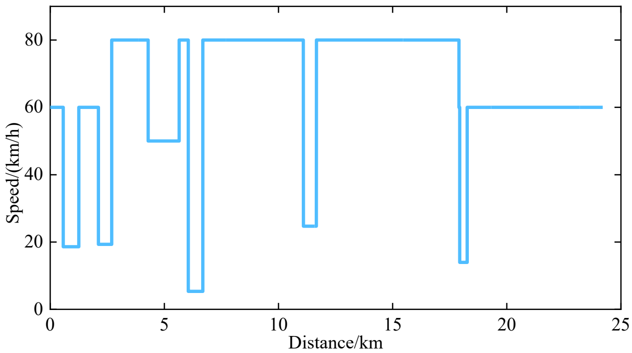

Fig.4. Average traffic speed obtained from Baidu Map API

The construction of traffic flow is based on the acquired average speed to simulate variations in road traffic. This process primarily involves utilizing the vehicle flow file in SUMO, which defines fundamental parameters such as vehicle type, acceleration and deceleration rates, maximum speed, and the number of vehicles. A speed factor can be introduced, allowing some vehicles to exceed the speed limit, thereby simulating speeding behavior. Additionally, a portion of the traffic flow can be designated as trucks or other vehicle types reflecting real-world traffic conditions. By adjusting the number of vehicles at each intersection and setting driving rules, the speed on each road section can be aligned with the average traffic speed obtained from the Baidu Map API. The resulting traffic flow at an intersection is illustrated in Figure 5.

B. Validation of the effectiveness of virtual simulation scenario

After completing the construction of the virtual scenario, it is essential to validate details such as vehicle flow, road distance within the virtual scenario. In this paper, SUMO-MATLAB co-simulation is used to obtain the data of the experimental vehicle. According to TraCI4MATLAB plug-in provided by SUMO, the co-simulation platform is built. The TraCI interface and related sample code are used to obtain information such as traffic lights, vehicles, and roads. After the traffic flow is loaded and established in the virtual simulation scenario, the experimental vehicle is added, and the speed data of the experimental vehicle is output through the TraCI interface. The collected data from the experimental vehicle are shown in Figure 6.

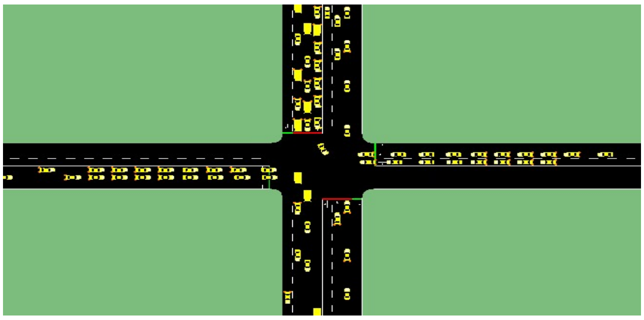

Fig.5. Traffic flow simulation diagram

Based on the data gathered from the experimental vehicle, its speed noticeably dropped at several locations, indicating traffic congestion. The vehicle's speed closely aligns with the average traffic speed obtained from the Baidu API, confirming the accuracy of the virtual scenario. This provides a basis for extracting the average global traffic flow speed in subsequent analysis.

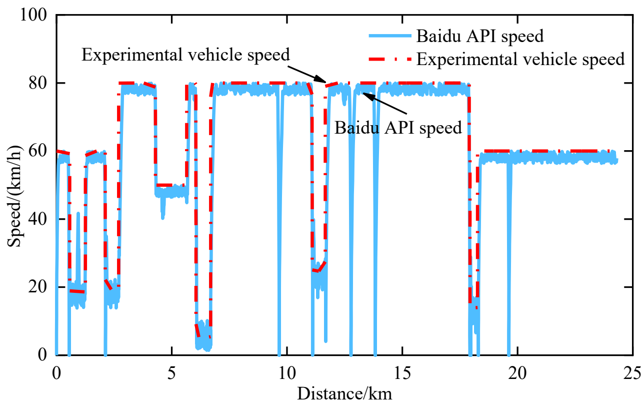

Fig.6. Speed comparison

C. Global traffic flow average speed generation

There are three types of detectors built into SUMO software. The E2 detector can set the detector length, enabling it to meet the requirement of collecting the average traffic flow speed at specified intervals. Therefore, the E2 detector is selected as a tool to obtain measure the average flow speed. E2 detectors are installed every 200 meters along the simulated road. Where the road length is less than 200 meters, it is placed according to the road length. Through Traci interface, each detector invokes the collected data through MATLAB software to obtain the traffic flow average speed. The current global traffic flow average speed information is obtained. Since the E2 detector can only be arranged in a single lane, it is necessary to arrange the detector in each lane of the section. In this way, it is necessary to do an arithmetic average of the extracted traffic flow average speed in each lane. In addition, to ensure the real-time data, the transmission frequency of the detector is set to 60s. The detector data is received every 60s in MATLAB software, and the average speed information of global traffic flow is updated at the same time. Figure 7 shows the collected one-hour average speed of traffic flow.

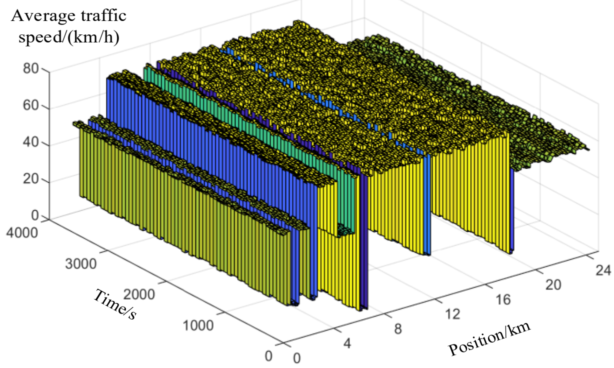

Fig.7. Average speed of traffic flow

The information the collected global traffic flow show that congestion occurs at several places in the spatial domain. At about 6km, the average traffic speed is only about 5km/h, and serious congestion occurs. It is basically consistent with the average traffic speed obtained from Baidu API, which verifies the validity of the acquisition method.

D. Approximate global driving condition generation based on smooth filter algorithm

Since the collected average traffic speed is a step signal, it does not match the driving condition of vehicle on the actual road. It is necessary to use the smooth filtering algorithm to smooth the data to generate the driving condition data of vehicle. The exponential weighted moving average method can quickly adapt to the data changes and has better smoothing results for the step data. Therefore, this paper selects the exponential weighted moving average method to smooth the extracted step data. The formula of exponential weighted moving average method is as follows:

$$F_t = \alpha Y_t + (1-\alpha) F_{t-1} \tag{6}$$

where F_t represents predicted value at time t, α represents a smooth coefficient, the value range is between 0 and 1. Y_t is the actual observed value at the moment t, $F_{t\text{-}1}$ is the predicted value at the moment t-1. The effect of the smoothing process is shown in Figure 8.

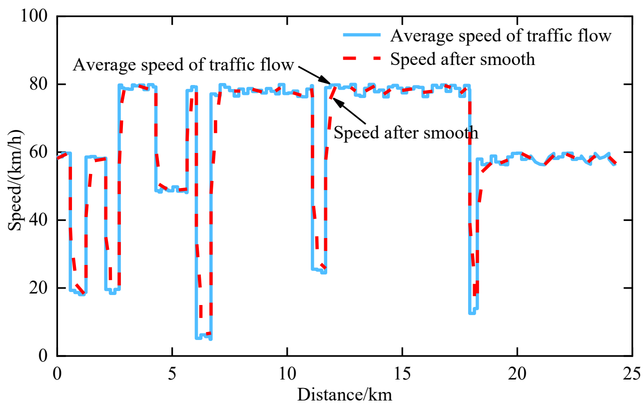

Fig.8. Smoothing the traffic flow average speed

After applying the exponential weighted moving average method, the step data has transformed into a smooth data curve, enhancing its clarity and interpretability. This smoothing process aids in accurately generating the vehicle's driving condition. Concurrently, the effectiveness of the exponential weighted moving average method is confirmed, demonstrating its capability to effectively process and refine the data. This validation ensures that the driving conditions derived from the smoothed data are reliable and representative of real-world scenarios.

IV. ADAPTIVE INFORMATION-DRIVEN CONTROL STRATEGY OF IOV

A. Modelling of equivalent fuel consumption minimization strategy

The ECMS aims to minimize the total equivalent fuel consumption of vehicle by rationally distributing the output torque of the engine and the motor. The ECMS strategy considers the instantaneous low fuel consumption operating interval of the engine and the efficiency of the motor. Therefore, it enables the power system to operate at its best instantaneous level, enhancing fuel economy. The mathematical model of the ECMS strategy can take the shape of:

$$\dot{m}_{eq}(t) = \dot{m}_{ice}(t) + s(t)\frac{\dot{P}_{bat}(t)}{H_{lhv}} \quad (7)$$

where $\dot{m}_{eq}(t)$ is the instantaneous total equivalent fuel consumption, $\dot{m}_{ice}(t)$ is the instantaneous fuel consumption of the engine, $s(t)$ is the equivalent factor, H_{lhv} is the low calorific value of fuel; $\dot{P}_{bat}(t)$ is the instantaneous power consumption of the power battery.

ECMS strategy originates the Pontryagin Minimum Principle. The global optimization challenge across the entire journey is reduced to a local optimal problem by using the minimum principle. In this paper, the SOC of the power battery is used as the state variable and the engine output torque is used as the control variable. The Hamilton function of the energy optimization problem of PHEV can be expressed as:

$$H(SOC(t), T_{ice}(t), \lambda(t), t) = \dot{m}_{ice}(T_{ice}(t), t) + \lambda(t)\dot{SOC}(t) \quad (8)$$

$$\dot{\lambda}(t) = -\frac{\partial H}{\partial SOC} = -\lambda(t)\frac{\partial \dot{SOC}(t)}{\partial SOC} \quad (9)$$

$$\dot{SOC}(t) = -\frac{I_b}{Q_b} = -\frac{P_{bat}(t)}{Q_b V_{oc}} \quad (10)$$

here $\lambda(t)$ denotes the co-state variable.

Based on the equivalence between the PMP and ECMS strategy, it can be seen from the comparison of Equation 7 and Equation 8:

$$s(t) = -\lambda(t)\frac{H_{lhv}}{Q_b V_{oc}} \quad (11)$$

The state variable and control variable should also meet the following constraints:

$$\begin{cases} SOC_{\min} \le SOC(t) \le SOC_{\max} \\ T_{ice\_\min}(n_{ice}(t)) \le T_{ice}(t) \le T_{ice\_\max}(n_{ice}(t)) \\ n_{ice\_\min} \le n_{ice}(t) \le n_{ice\_\max} \\ T_{m\_\min}(n_m(t)) \le T_m(t) \le T_{m\_\max}(n_m(t)) \\ n_{m\_\min} \le n_m(t) \le n_{m\_\max} \end{cases} \quad (12)$$

After above analysis, the ECMS strategy model is constructed using MATLAB/Simulink according to Equation 7. The equivalent factor affects the trajectory of SOC and determines the torque distribution ratio during the vehicle operation. Figure 9 shows SOC trajectories with different initial equivalent factors under UDDS condition. The SOC trajectory reveals that the larger the initial equivalent factor, the more the ECMS strategy tends to consume fuel. And frequent charging of the battery increases the SOC trajectory gradually. The smaller the initial equivalent factor, the more the ECMS strategy tends to consume power. The battery will discharge frequently during the driving process to gradually reduce the SOC. Hence, selecting an appropriate equivalent factor is crucial to make that the battery's SOC precisely reaches the lower SOC limit at trip's end and achieves good fuel economy.

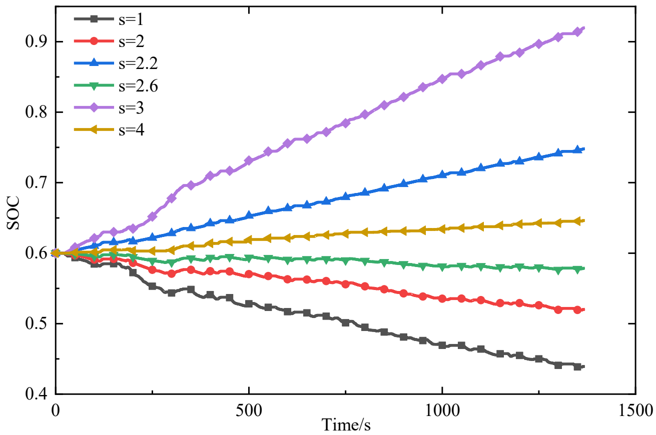

Fig.9. SOC trajectories with different initial equivalent factors

B. A-ECMS strategy based on information of IoV

The fixed equivalent factor cannot adapt to changing condition and complex environment. Fixed equivalent factor will cause vehicle to enter the charge sustain mode prematurely, leading to higher fuel consumption. Or the battery SOC fails to take the specified SOC lower limit by the trip's end, which fails to take full advantage of the large capacity battery of PHEV. Therefore, an adaptive ECMS strategy that can change with driving condition and an environment has been put forward to ensure the battery's SOC just reaches the target SOC level upon completion of the trip. Presently, the main focus of research is the impact of driving distance and battery SOC on A-ECMS strategy [18-19]. This paper further considers the impact of information of IoV on A-ECMS strategy. After the above analysis, to fully excavate the energy saving potential of PHEV, this paper proposes an adaptive equivalent fuel consumption minimization strategy that integrates the initial SOC, driving distance and information of IoV. The strategy diagram is shown in Figure 10.

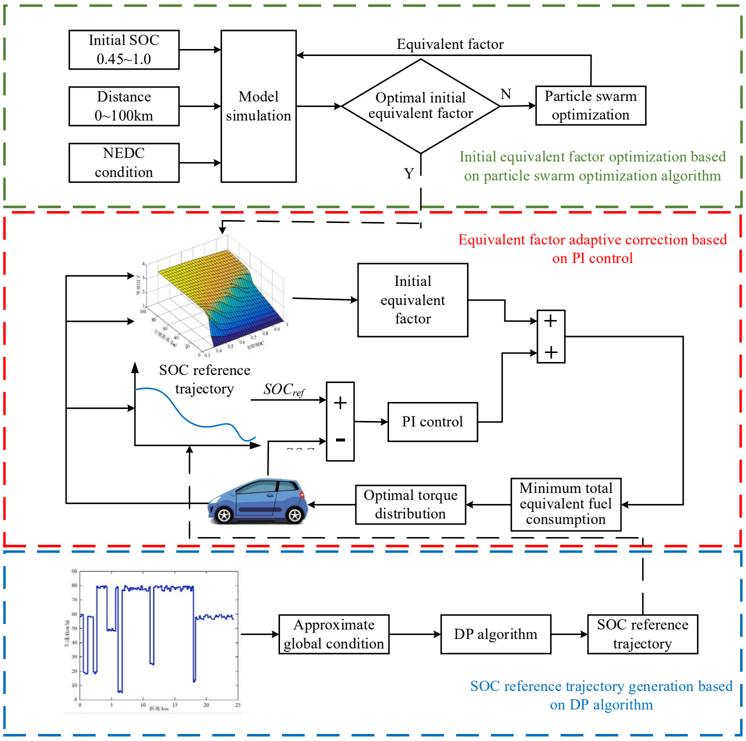

Fig.10. A-ECMS strategy based on information of IoV

1) Initial equivalent factor offline optimization utilizing particle swarm optimization techniques: PSO techniques is usually used to solve optimization problems, such as function optimization, parameter optimization [20]. Therefore, the ECMS strategy's equivalent factor for its vital control parameters is optimized through PSO techniques. Given that the research paper is to enhance the fuel economy of PHEV, the total equivalent fuel consumption is used as the fitness function, given by the equation below:

$$J_{fit} = \int_{t_0}^{t_f} \dot{m}_{ice}(t) + \dot{m}_{ele}(t)dt \tag{13}$$

where $\dot{m}_{ele}(t)$ is the electrical energy equivalent of fuel consumption. The offline optimization flow chart of equivalent factor is shown in Figure 11.

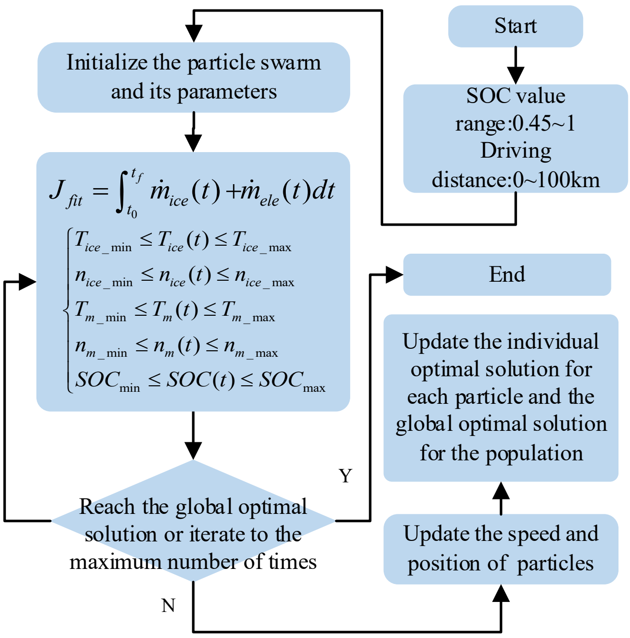

Fig.11. Flow chart of PSO techniques

Particle swarm algorithm is employed to optimize the initial equivalent factor for varying initial SOC and varying driving distances. The optimization objective is set to minimize the total equivalent fuel consumption, with the initial SOC ranging from 0.45~1. The range for driving distance is between 0~100km. Multiple simulations are carried out under NEDC condition. Finally, the initial SOC driving distance initial equivalent factor MAP is generated, as shown in Figure 12.

2) Generation of a reference SOC trajectory utilizing the DP algorithm: For the PHEV to exploit electrical energy efficiently so that it just reaches the lower SOC limit by the conclusion of the trip. The reference SOC curve needs to be added to limit the actual SOC. In this paper, it is generated using the acquired approximate global driving condition information combined with the DP algorithm.

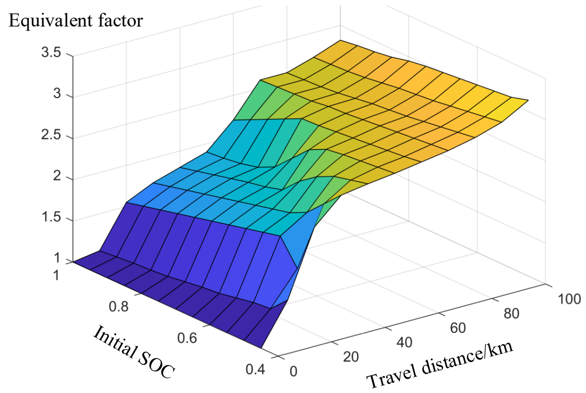

Fig.12. MAP of initial SOC -travel distance - initial equivalence factor

The battery SOC is selected as the state variable. The engine torque is used as the control variable. The total equivalent fuel consumption of Equation 13 is used as a cost function. The DP algorithm serves as the method for the specific solution procedure for PHEV control problem, as shown in Figure 13.

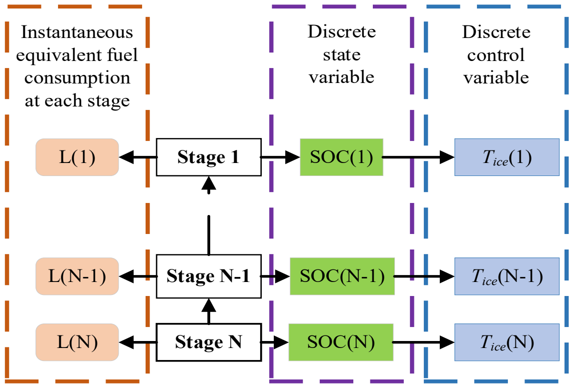

Fig.13. DP algorithm solution process

The SOC reference trajectory is generated using the acquired approximate global condition information combined with the DP algorithm. The resulting SOC reference curve is illustrated in Figure 14. The generated SOC reference trajectory is used to limit the actual SOC.

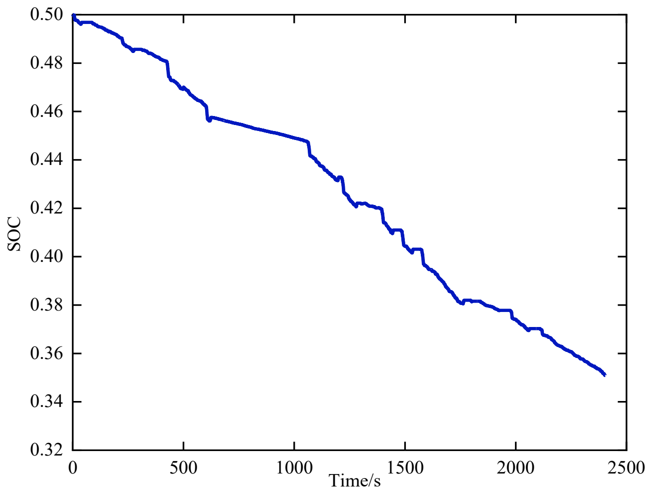

Fig.14. SOC reference trajectory

3) Equivalent factor adaptive correction based on PI control: To achieve a good following effect on the SOC reference trajectory, the equivalent factor is used as a control variable. According to the error values of the actual SOC and reference SOC, the equivalent factor is adaptively corrected by PI control. The modified equivalent factor change is:

$$s_{ame}(k) = K_p \cdot e_t + K_i \int_0^t e_t dt \tag{14}$$

$$e_k = SOC_r(k) - SOC(k) \tag{15}$$

where $s_{ame}(k)$ denotes the equivalent factor change of PI control correction at time k, e_k is the deviation of the actual SOC from the reference SOC at time k.

C. Simulation result analysis

In this section, different energy management strategies will be simulated and compared through the experimental vehicle speed and global traffic flow average speed information obtained by SUMO. The validity of the adaptive control strategy leveraging IoV data is confirmed. The simulation process extracts the global traffic flow average speed information of the expected travel before the experimental vehicle starts, which is used to generate the reference trajectory of SOC. In addition, the global traffic flow information is updated every 60s, and the corresponding SOC reference trajectory is also updated every 60s. The real-time and correctness of SOC reference trajectory are guaranteed. The initial SOC is set to be 0.5, and the lower limit value of SOC is 0.35. According to the total length of the selected experimental road of 24.2 km, the initial SOC is 0.5 and the optimal initial equivalent factor is 2.81. The collected experimental vehicle driving condition is selected. Simulation experiments of ECMS strategy considering only driving distance and initial SOC and A-ECMS strategy considering driving distance, initial SOC, and information from IoV were carried out respectively.

1) Analysis of SOC simulation results: One can observe from Figure 15 that the ECMS strategy without incorporating data from IoV has poor driving condition and environmental adaptability due to the adoption of fixed equivalent factor. The PHEV enters the charge sustain mode before the final stage of the trip, without fully exploiting electric power. According to the A-ECMS strategy, which is based on vehicular networking information and illustrated in Figure 16, the SOC target path produced by PI control is followed to achieve dynamic modification of the equivalent factor. The SOC curve closely follows the reference trajectory without significant fluctuations. At the final stage of the trip, the SOC reaches the lower limit, thereby making full use of the available electric power. Figure 17 and Figure 18 illustrate the variations in motor torque and engine torque under different energy management strategies.

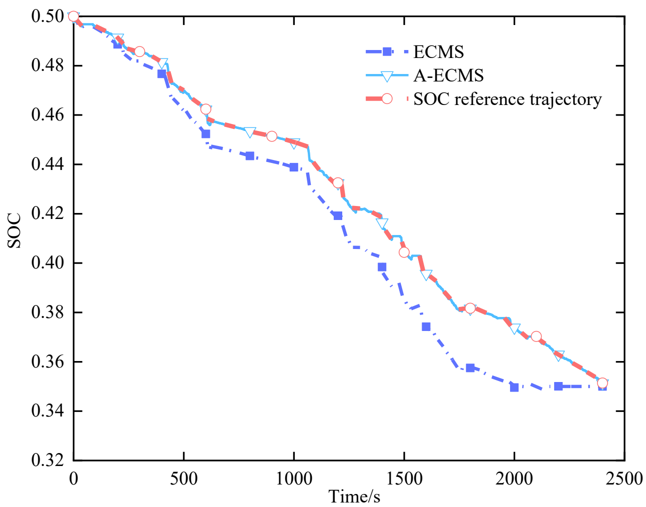

Fig.15. SOC change trajectories under different strategies

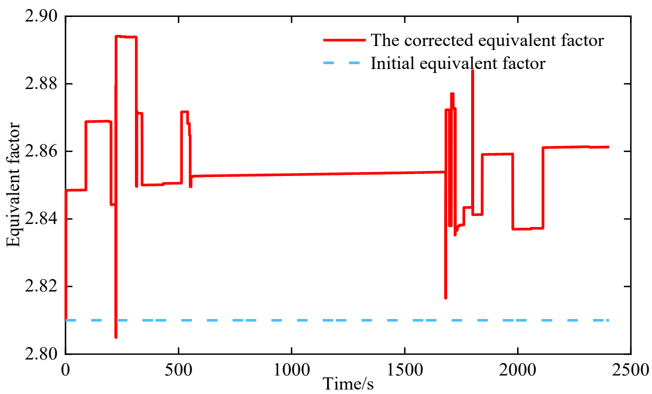

Fig.16. Equivalent factor comparison

2) Fuel economy simulation results analysis: The analysis of Figure 19 indicates that the ECMS strategy without

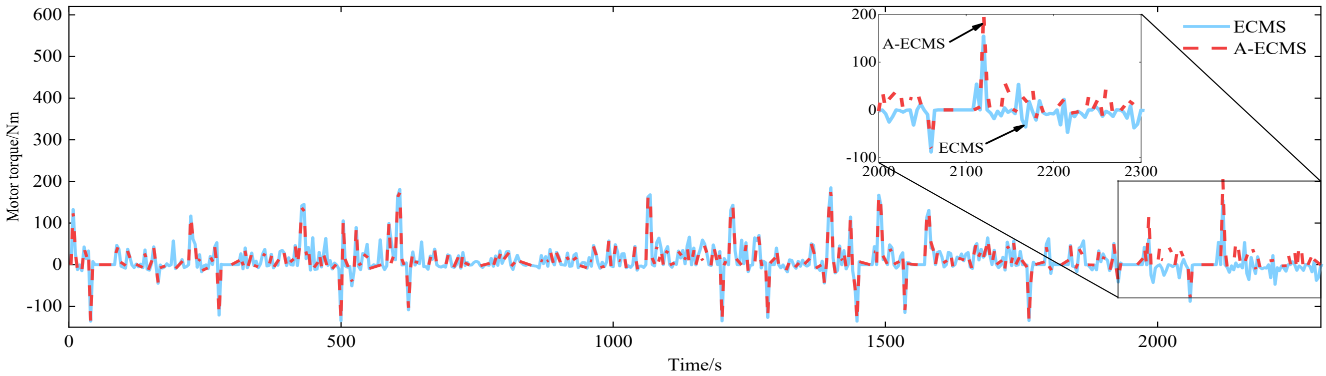

Fig.17. Simulation results of motor torque

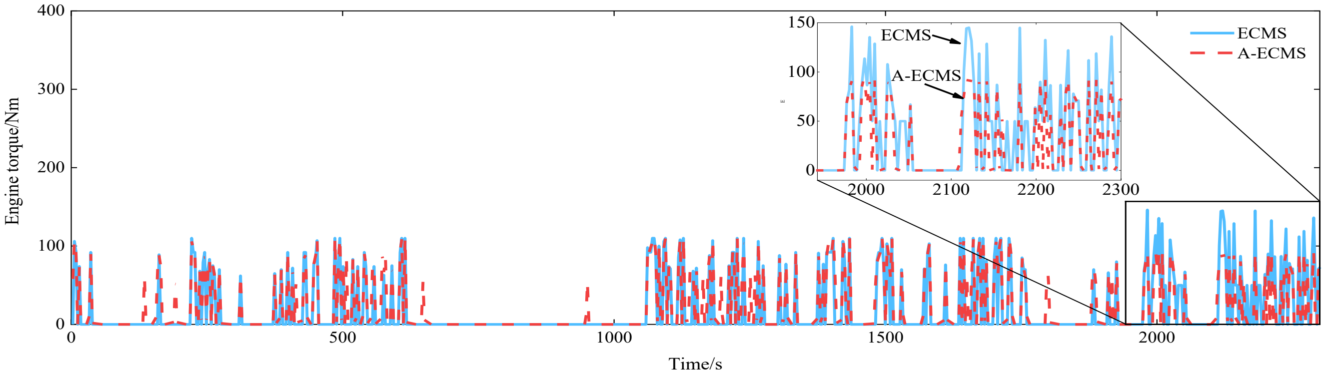

Fig.18. Simulation results of engine torque

information of IoV enters the charge sustain mode at the end of the trip due to the fixed equivalent factor. Fuel consumption has increased. The A-ECMS strategy based on information of IoV realizes the adaptive adjustment of equivalent factor by following the SOC reference trajectory. PHEV can just reach the specified SOC lower limit by the trip's end, which makes full use of power. And the engine is engaged at the right time, which works in the high efficiency range. Fuel consumption rises slowly. Therefore, the fuel economy performance is better.

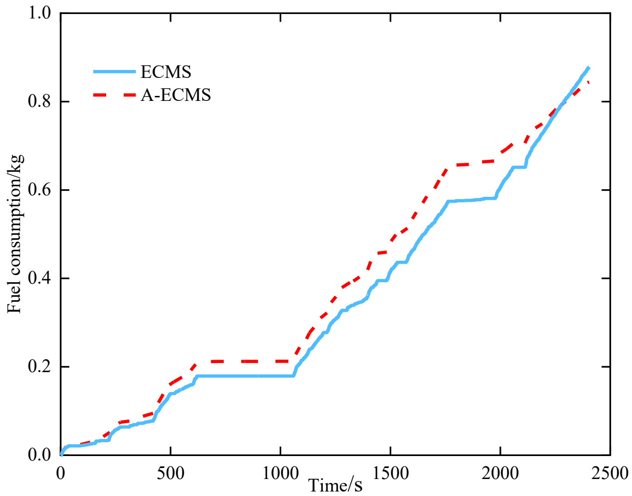

Fig.19. Fuel consumption under different strategies

From the comparison results of equivalent fuel consumption per 100 km in Table II, compared with the ECMS strategy without information of IoV, the proposed strategy improves fuel economy by 3.6%. This paper demonstrates the energy-saving effect of this strategy. The validity of the adaptive control strategy based on the IoV is verified.

TABLE II
FUEL ECONOMY COMPARISON

Energy management strategy	Equivalent fuel consumption per 100 km	Fuel economy comparison
ECMS strategy without information of IoV	4.16L	-
A-ECMS strategy with information of IoV	4.01L	3.6%

V. VERIFICATION OF REAL-TIME ROAD CONDITION INFORMATION ACQUISITION AND ENERGY MANAGEMENT STRATEGY BASED ON LINUX SYSTEM BOARD

Modern vehicles can connect to the internet through 4G/5G modules, Wi-Fi, Bluetooth, and other technologies, and can obtain road condition information through the on-board interconnection terminal. This chapter describes the network configuration of the 4G module. A program to obtain real-time traffic information is developed on the Linux system board using the configured 4G module. The data is transmitted to the host computer by means of serial communication. The validity of obtaining real-time traffic information and developing a program based on Linux system board is verified. The real-time road information obtained by A Linux system board is implemented for the adaptive control strategy utilizing information from IoV, and effectiveness of the strategy is validated.

A. 4G module configuration

Linux system board hardware resources are shown in Figure 20. The 4G module is shown in Figure 21. First, the board is connected to the network through the computer sharing network. Secondly, the board is started by SSH

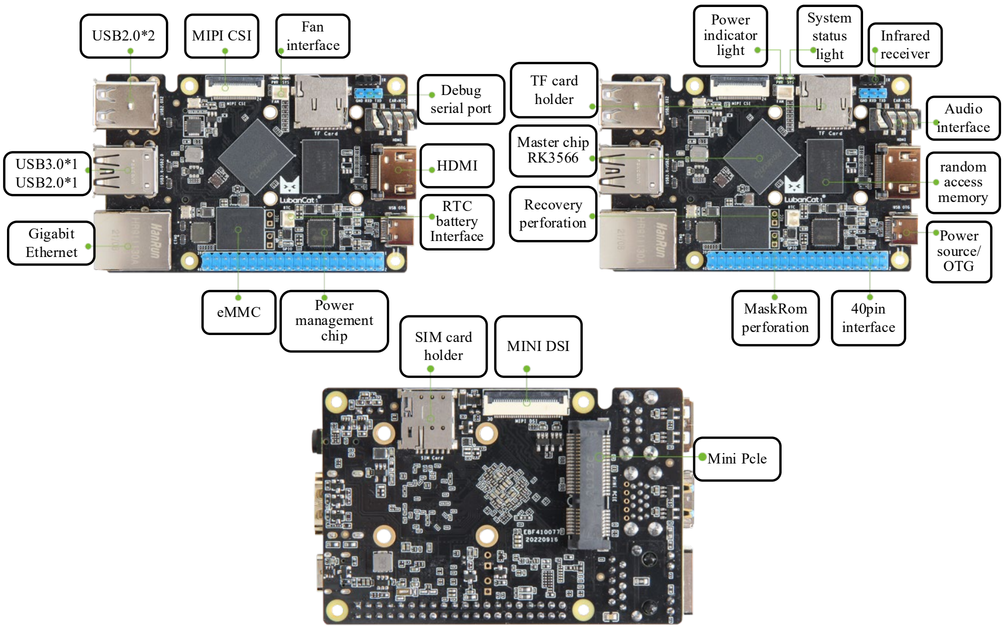

Fig.20. Linux system board hardware resources

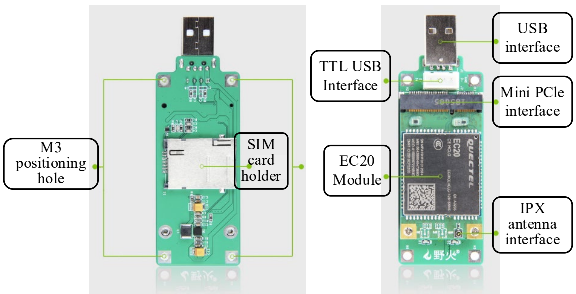

Fig.21. 4G module

protocol. After completing the above steps, the 4G module can be used to dial up the Internet through Linux commands. The board can also be set to the 4G module self-start mode and can directly connect to the board through SSH for subsequent operations in the next connection.

B. Development of real-time road condition information acquisition program

After the hardware resources of the board are configured, the built-in Python software of the board can be used to write the code. The general flow of the program is to first initialize the road parameters and generate txt text for the road name of the expected trip. It then reads the road information in the file and saves each road name to "Line_list". The roads in the text are assigned road speed limits in turn through "For" loop. Then use "Request" to call Baidu map API interface to return real-time road information. If the information is successfully obtained, the speed limit in the text is updated and the average speed is filled into the corresponding position. If the return is unsuccessful, exception handling is performed. The Baidu map API updates the road condition information every 60 seconds. Therefore, the main program is set to call Baidu map API every 60 seconds to ensure the real-time acquisition of road condition information. The pseudo-code of the program is shown in Table III.

C. Data transmission and result acquisition

After completing the program of obtaining real-time road condition information, transmission of acquired road information data is necessary. Validation of the development program is completed. Since URAT is asynchronous communication, no clock signal is required. The clock synchronization between the sender and receiver is not required. Full duplex data transmission can be achieved by using only two signal lines. Therefore, this paper uses USB to TTL serial line and URAT communication protocol to realize data transmission between board and terminal. The connection of the specific serial port is shown in Figure 22. The sending end of the serial port is linked to the receive side of the pin. The receive side of the serial port is connected to the sending end of the pin. The hardware connection be-tween the board and the terminal can be completed.

TABLE III
REAL-TIME TRAFFIC INFORMATION ACQUISITION PROGRAM PSEUDO-CODE

Pseudo-code of real-time road information acquisition programme

```
Initialize: Road information
Initialize: Speed V_list
While True do
  Read the file, get the name of the queried road, and save to
    Line_list
  For line in Line_list do:
    Road speed limit assignment
  End
  Read real-time speed
  For num in roadnum do:
    Res = Request(num road section)
    If Successful information acquisition do:
      Speed = Res['Average traffic speed']
      V_list.append(Speed)
    Else:
      Exception handling
    End
  End
  Sleep(60)
End
```

After the hardware connection of the serial port is completed, you also need to configure the software to realize data transmission. Firstly, the periphery library is installed on the board. Then the connection between the board and the host computer is realized by writing a program. Using this program can realize the function of bidirectional data transmission with the host computer. After the system board's data transfer program is configured, the corresponding configuration is also needed in MATLAB to realize the data transmission. Simulink provides serial communication modules. First, the parameters of the parameter configuration module such as baud rate are set to the same as the program. Then, after the data is obtained by the real-time road condition information acquisition program, the data receiving module receives the data. After completing the above hardware resource connection and software parameter configuration, the real-time traffic information data acquisition is realized. The final data acquisition results are shown in Figure 23.

Physical pin		Function	Serial number	GPIO	USB to URAT module
1	2	5V			
3	4	5V			
5	6	GND			
7	8	UART3_T	111	GPIO0_D	RXD
9	10	UART3_R	112	GPIO3_C0	TXD
11	12	PWM8	105	GPIO3_B1	
13	14	GND			GND
15	16	GPIO2_D	95		
17	18	GPIO3_A	96		
19	20	GND			
21	22	GPIO0_C2	18		
23	24	CS0	150	GPIO4_C6	
25	26	CS1	148	GPIO3_C4	
27	28	I2C5_SCL	107	GPIO3_B3	
29	30	GND			
31	32	PWM9	106	GPIO3_B2	
33	34	GND			
35	36	GPIO1_B0	40		
37	38	GPIO1_B1	41		
39	40	GPIO1_B2	42		

→ wire

Fig.22. Serial port connection diagram of the board

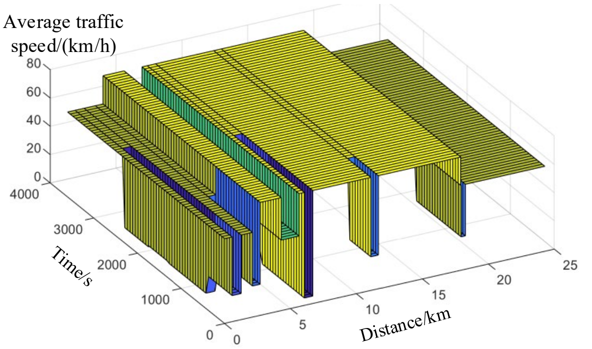

Fig.23. Real-time traffic information obtained based on Linux system board

The figure shows the data collected from Shandong University of Technology to Wenchang Lake Resort during the evening peak. From the real-time road condition information obtained in the figure, congestion occurs in many places in the spatial domain. Most of the congested sections in the time domain realize the smooth road after a period of congestion. The collected data is basically consistent with the congestion level in the map. The validity of the collected data and the correctness of the program for obtaining real-time traffic information are verified. The ECMS applying collected real-time road condition information effectively enhance fuel economy.

D. Validation of energy management strategy based on real-time road condition information

In this section, the obtained real-time traffic information is applied to an adaptive control strategy based on data from IoV. The simulation process is like Section 4.3. The difference is that the average global traffic speed obtained in this chapter is directly obtained based on the Linux system board. The process of obtaining information using Internet of Vehicles technology is simulated. Reconfigure traffic flow based on the average traffic speed obtained by the Linux system board. The speed of the experimental vehicle was obtained by using the Traci interface as a verification driving condition for the energy management strategy, as shown in Figure 24.

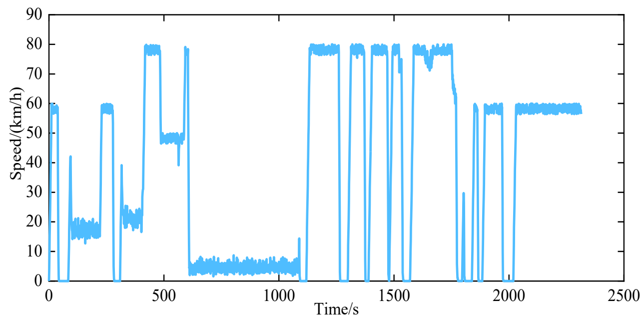

Fig.24. Experimental vehicle driving condition

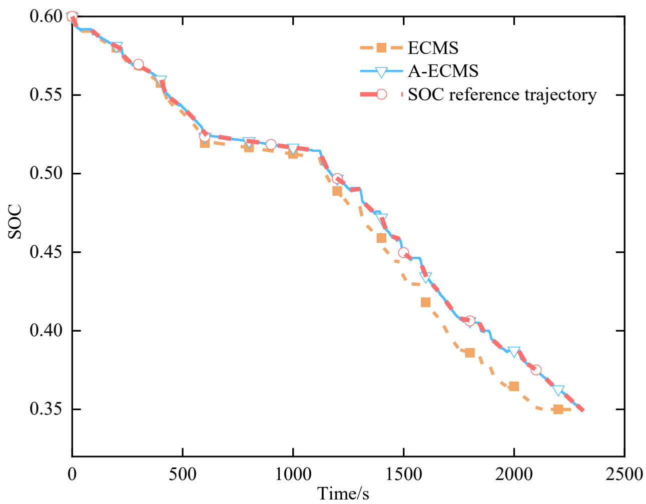

Fig.25. SOC change trajectories under different strategies

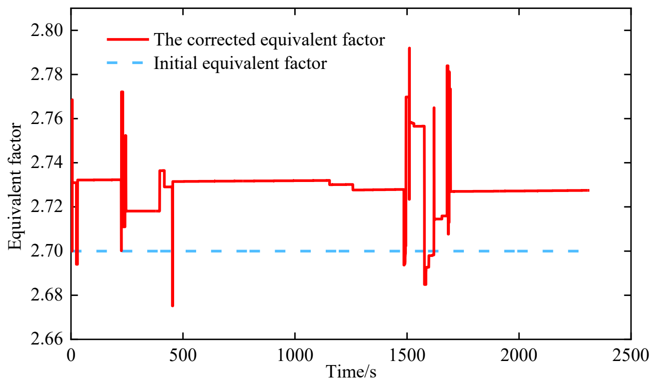

Fig.26. Equivalent factor comparison

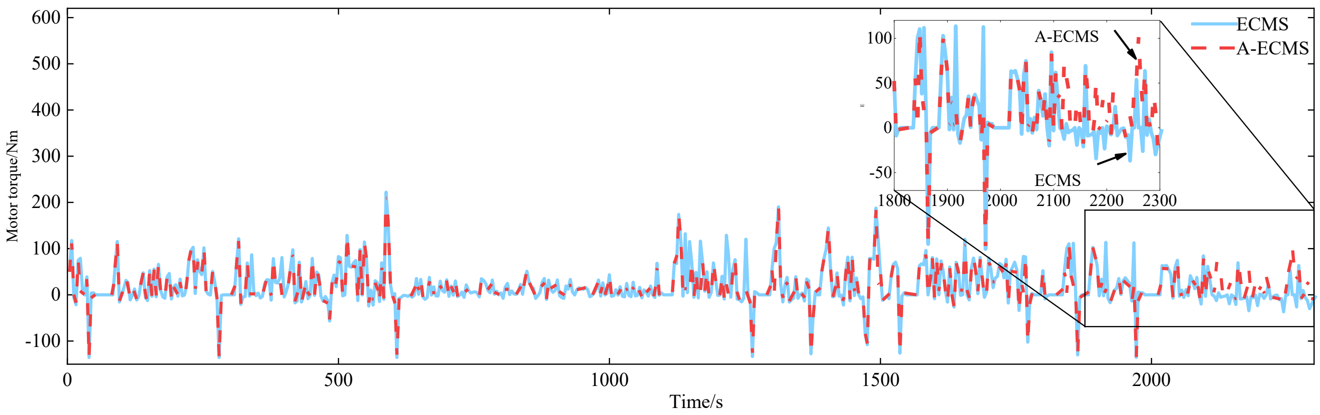

Fig.27. Experiment results of motor torque

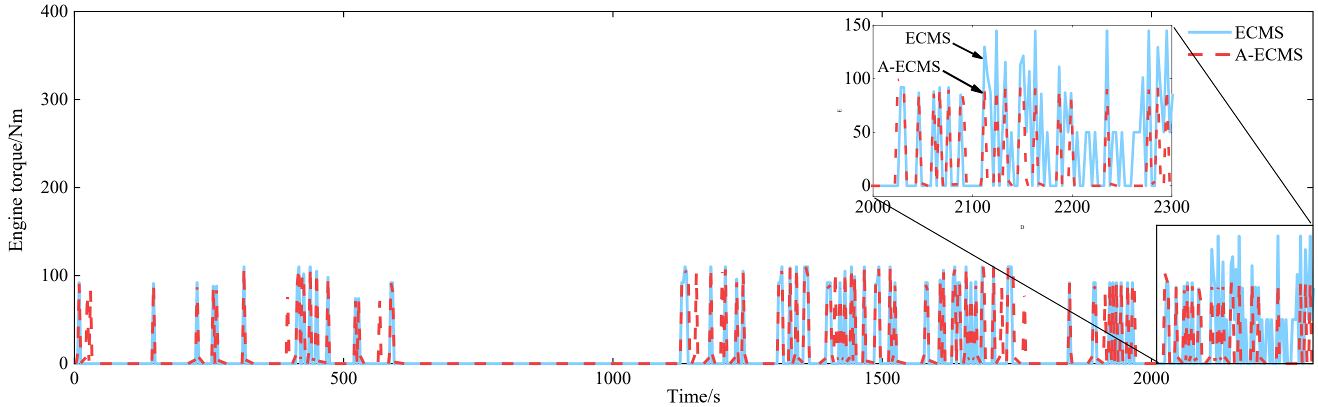

Fig.28. Experiment results of engine torque

The ECMS strategy without information of IoV is selected as the comparison strategy. The initial SOC is 0.6, with a lower SOC limit of 0.35. According to the MAP generated by the PSO algorithm, the initial equivalent factor is 2.7. The comparison results of energy management strategies are as follows:

1) Analysis of SOC simulation results: Combined with Figures 25 and 26, the ECMS strategy without the information from IoV cannot adapt to changing driving conditions due to the fixed initial equivalent factor. The SOC trajectory drops to the set lower limit before the end of the trip, failing to fully utilize the PHEV's large-capacity battery. In contrast, the adaptive ECMS strategy based on the information of IoV employs PI control to follow the SOC reference trajectory generated by the DP algorithm. This strategy achieves adaptive correction of the equivalent factor, resulting in a good following effect. At the same time, the proportion of engine and motor drive is reasonably planned throughout the trip, ensuring that the SOC decreases gradually. At the trip's end, the SOC just reaches set lower limit, thereby making full use of the electric energy.

2) Fuel economy simulation results analysis: From Figures 27, 28, and 29, the proportion of motor participation in driving in the ECMS strategy is higher than that of the engine at the beginning of the trip without the data of IoV. Fuel consumption is slightly lower than the A-ECMS strategy based on information of IoV. When the SOC drops to the set lower limit, the vehicle enters the charge sustain mode. The engine participates in the drive as the main power source. The engine works in the inefficient zone. The fuel consumption increases rapidly. The ECMS strategy with the fixed equivalent factor does not have good adaptability to driving condition. The A-ECMS strategy is based on information from IoV, the engine participates in the drive at the appropriate time during the whole trip. The fuel consumption rises slowly. At the trip's end, the fuel consumption is lower, which demonstrates good fuel economy.

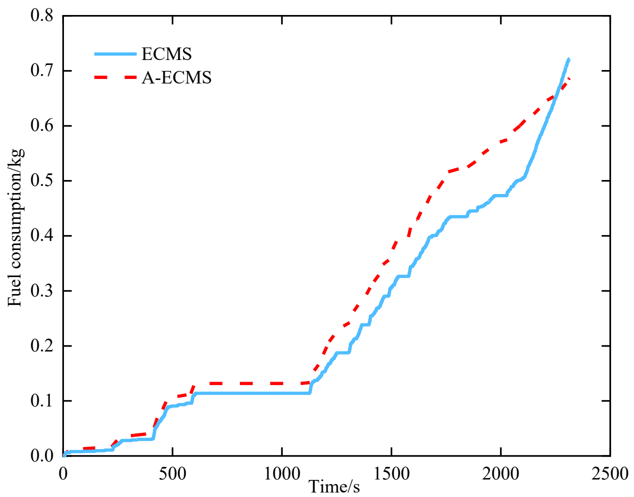

Fig.29. Fuel consumption under different strategies

TABLE IV
FUEL ECONOMY COMPARISON

Energy management strategy	Equivalent fuel consumption of 100 km	Fuel economy comparison
ECMS strategy without information of IoV	5.33L	-
A-ECMS strategy with information of IoV	5.13L	3.8%

As shown in Table IV, the A-ECMS strategy utilizing IoV information enhances fuel economy by 3.8% compared to the ECMS strategy with a fixed equivalent factor. This demonstrates the energy-saving potential of the proposed strategy, further validating the effectiveness of the IoV-based adaptive control approach.

VI. CONCLUSION

(1) For the P2 configuration PHEV, the vehicle power system architecture is analyzed. The vehicle model is developed using MATLAB/Simulink.

(2) Based on the actual road information and traffic information, the virtual scenario is built. The speed of the experimental vehicle verifies the validity of the virtual scenario. The global traffic flow average speed information is extracted by SUMO built-in detector. The extracted step data is smoothed by smoothing filter algorithm. The approximate global driving condition is generated.

(3) Using the obtained data, an A-ECMS strategy considering information of IoV is proposed. The SUMO-MATLAB co-simulation platform is built to verify the effectiveness of the proposed strategy. The simulation results show that the A-ECMS strategy with the information from IoV saves 3.6% of fuel consumption compared with the ECMS strategy without information from IoV. The energy saving effect of the proposed strategy is demonstrated.

(4) In order to verify the possibility that the strategy can be applied to real vehicle, a Python program that can obtain real-time road condition information is developed using Linux board with 4G module. The data transmission is done through serial communication. The validity of obtaining real-time traffic information based on Linux board is verified. The real-time traffic data obtained is incorporated into the ECMS. In comparison to the ECMS strategy lacking IoV information, the proposed approach enhances fuel economy by 3.8%. This confirms the effectiveness of the proposed strategy.

REFERENCES

[1] F. Zhang, L. Wang, S. Coskun, H. Pang and Y. Cui, "Energy management strategies for hybrid electric vehicles: Review, classification, comparison, and outlook," *Energies*, vol. 13, no. 13, p. 3352, 2020.

[2] A. Ahmadian, B. Mohammadi-Ivatloo, and A. Elkamel, "A review on plug-in electric vehicles: Introduction, current status, and load modeling techniques," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 3, pp. 412-425, 2020.

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

[4] H. A. Trinh, H. V. Truong, K. K. Ahn, "Development of fuzzy-adaptive control based energy management strategy for PEM fuel cell hybrid tramway system," *Applied Sciences*, vol. 12, no. 8, p. 3880, 2022.

[5] S. Najjaran, Z. Rahmani, M. Hassanzadeh, "Fuzzy predictive control strategy for plug-in hybrid electric vehicles over multiple driving cycles," *International Journal of Dynamics and Control*, vol. 10, no.3, pp. 930-941, 2022.

[6] S. Tao, W. Chen, R. Gan, L. Li, G. Zhang, Y. Han and Q Li, "Energy management strategy based on dynamic programming with durability extension for fuel cell hybrid tramway," *Railway Engineering Science*, vol. 29, pp. 299-313, 2021.

[7] D. Shi, S. Liu, Y. Cai, S.Wang, H. Li and L. Chen, "Pontryagin's minimum principle based fuzzy adaptive energy management for hybrid electric vehicle using real-time traffic information," *Applied Energy*, vol. 286, p. 116467, 2021.

[8] C. Ma, Y. Shang, S. Jin, K. Yang, and Z. Li, "Research on real-time energy management strategy of dual motor coupled PHEV based on model predictive control," *IAENG International Journal of Applied Mathematics*, vol. 53, no. 1, pp. 76–85, 2023.

[9] X. Liu, D. Qin, S. Wang, "Minimum energy management strategy of equivalent fuel consumption of hybrid electric vehicle based on improved global optimization equivalent factor," *Energies*, vol. 12, no. 11, p. 2076, 2019.

[10] W. Wang, S. Tian, Q. Zheng and Y. Luo, "Optimization of equivalent fuel consumption minimization strategy based on firefly algorithm," *Journal of Jiangsu University (Natural Science Edition)*, vol. 43, no. 2, pp. 147-153, 2022.

[11] Y. Zeng, J. Sheng, M. Li, "Adaptive real-time energy management strategy for plug-in hybrid electric vehicle based on simplified-ECMS and a novel driving pattern recognition method," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[12] P. Li, X. Jiao, Y. Li, "Adaptive real-time energy management control strategy based on fuzzy inference system for plug-in hybrid electric vehicles," *Control Engineering Practice*, vol. 107, p. 104703, 2021.

[13] L. Han, X. Jiao, Z. Zhang, "Recurrent neural network-based adaptive energy management control strategy of plug-in hybrid electric vehicles considering battery aging," *Energies*, vol. 13, no. 1, p. 202, 2020.

[14] C. Yang, M. Zha, W. Wang and K. Liu, "Efficient energy management strategy for hybrid electric vehicles/plug‐in hybrid electric vehicles: review and recent advances under intelligent transportation system," *IET Intelligent Transport Systems*, vol. 14, no. 7, pp. 702-711, 2020.

[15] X. Tang, Z. Duan, X. Hu, H. Pu, D. Cao and X. Lin, "Improving ride comfort and fuel economy of connected hybrid electric vehicles based on traffic signals and real road information," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3101-3112, 2021.

[16] Z. Lei, D. Qin, L. Hou, J. Peng, Y. Liu and Z. Chen, "An adaptive equivalent consumption minimization strategy for plug-in hybrid electric vehicles based on traffic information," *Energy*, vol. 190, p.116409, 2020.

[17] X. Sun, Y. Cao, X. Tian and M. Xue, "An adaptive ECMS based on traffic information for plug-in hybrid electric buses," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 9, pp. 9248-9259, 2022.

[18] T. Zeng, C. Zhang, Y. Zhang, C. Deng, D. Hao, Z. Zhu, H. Ran and D. Cao, "Optimization-oriented adaptive equivalent consumption minimization strategy based on short-term demand power prediction for fuel cell hybrid vehicle," *Energy*, vol. 227, p. 120305, 2021.

[19] J. Li, Y. Liu, D. Qin, G. Li and Z. Chen, "Research on equivalent factor boundary of equivalent consumption minimization strategy for PHEVs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6011-6024, 2020.

[20] M. A. M. De Oca, T. Stutzle, M. Birattari and M. Dorigo, "Frankenstein's PSO: A Composite Particle Swarm Optimization Algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1120-1132, 2009.