2019 IEEE Intelligent Transportation Systems Conference (ITSC)
Auckland, NZ, October 27-30, 2019

A Deep Reinforcement Learning-Based Approach to Intelligent Powertrain Control for Automated Vehicles *

I-Ming Chen, *Member, IEEE*, Cong Zhao and Ching-Yao Chan, *Member, IEEE*

***Abstract*—The development of a powertrain controller for automated driving systems (ADS) using a learning-based approach is presented in this paper. The goal is to create an intelligent agent to learn from driving experience and integrate the operation of each powertrain unit to benefit vehicle driving performance and efficiency. The feature of reinforcement learning (RL) that the agent interacts with environment to improve its policy mimics the learning process of a human driver. Therefore, we have adopted the RL methodology to train the agent for the intelligent powertrain control problem in this study. In this paper, a vehicle powertrain control strategist named Intelligent Powertrain Agent (IPA) based on deep Q-learning network (DQN) is proposed. In the application for an ADS, the IPA receives trajectory maneuver demands from a decision-making module of an ADS, observes the vehicle states and driving environment as the inputs, and outputs the control commands to the powertrain system. As a case study, the IPA is applied to a parallel hybrid vehicle to demonstrate its capability. Through the training process, the IPA is able to learn how to operate powertrain units in an integrated way to deal with varied driving conditions, and vehicle's speed chasing ability and power management are improved along with its driving experience.**

I. Introduction

The development of autonomous vehicles (AVs) and automated driving systems (ADS) is a multi-disciplinary integration as depicted in an exemplar system architecture in Fig. 1. The system requires efforts across many technological fields including computer vision, sensor fusion, information connection, planning and control. While the topics of perception and planning for AVs have drawn a great deal of attention from researchers, the lower-level control problem related to the operational control of steering, throttle and brake earns relatively less notice. This is due to the fact that the conventional control approaches used by the vehicle industry have long been proven to be solid and efficient for low-level controllers, and is regarded to be sufficient for the applications of ADS. However, for an ADS, there will be a higher level of integration and coordination, which is an essential difference from conventional vehicles. Therefore, the powertrain performance of an ADS has the potential to be further improved in a systematic manner in comparison with the component level optimization conventionally.

*Research supported by Ministry of Science and Technology, Taiwan with project number 107-2917-I-564-039.

I-M. Chen and C.-Y. Chan are with California PATH, University of California, Berkeley, CA 94804, USA. (e-mail: imchen@berkeley.edu; cychan@berkeley.edu). C. Zhao is with Key Laboratory of Road and Traffic Engineering of the Ministry of Education, Tongji University, Shanghai 201804, China. (e-mail: zhc@tongji.edu.cn).

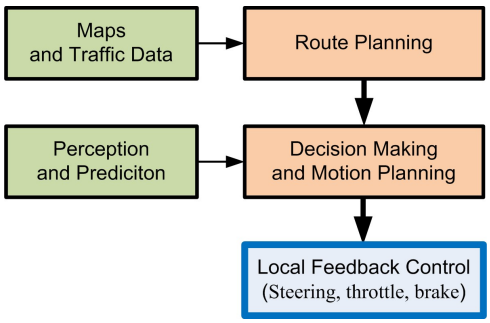

Figure. 1 Architecture for ADS technology

The approaches for vehicle powertrain control can be classified into three categories as shown in Fig. 2. Rule-based methods use principles developed with heuristics, experiment, and analysis to formulate a control strategy with look-up tables of previously saved parameters for real-time control. Being simple, fast, and stable, the rule-based approach is the most commonly used approach in practical applications. However, the rule-based methods are usually inflexible by their nature, and it cannot adapt to changes in driving situation. Therefore, the system performance is far from being optimal, and calibration is required to ensure system performance in different conditions. The optimization-based methods use an established system mathematical model with an optimized or sub-optimized algorithm to calculate the control strategy. Some examples of the optimization-based methods are equivalent consumption minimization strategies (ECMS) [1], dynamic programming (DP), stochastic dynamic programming (SDP) [2] and model predictive control (MPC). While these methods offer a mechanism for optimization, such methods require highly accurate system models, high computational demands and prediction for future driving condition, thus rendering the implementation of this type of approach impractical.

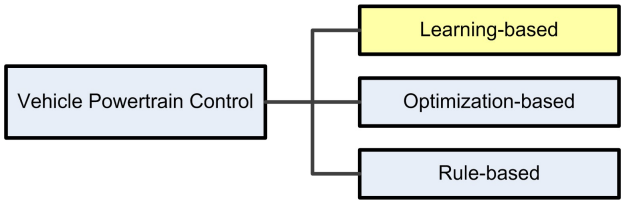

Figure. 2 Control category of vehicle powertrain control

The third category is the learning-based method, which is an emerging approach in recent years. Learning-based methods structure and train a neural network (NN) to establish the relationship between the input and output data, and are able to adapt and optimize powertrain operation for different driving conditions. With the data-driven mechanism, the difficulty of deriving accurate system models can be

dismissed, and real measurement data are utilized to practically improve system control strategy. When the training process is completed, the stored NN can be applied to real-time control. The learning-based method possesses the advantages of both rule-based and optimization-based approaches. The NN system can be used for driving environment prediction as suggested in [3], [4], or performing power management illustrated in [5], [6]. However, the aforementioned previous studies did not consider the integration of NN with lower level controllers, namely, the throttle, brake, and gear shift operation. Since an integrated powertrain controller is beneficial to ADS, we propose a learning-based control approach to establish an adaptive and intelligent agent.

In this paper, we use the name of Intelligent Powertrain Agent (IPA) to refer to the learning-based integrated control agent for ADS. The development of this agent is based on the concept of deep Q-learning network (DQN). Fig. 3 shows the architecture of the IPA. The IPA receives motion planning commands from a high-level decision maker in the ADS. The IPA uses the motion command, as well as the vehicle state and the surrounding environment as its inputs. The IPA then outputs control commands to powertrain units including internal combustion engine (ICE), motor/generator (MG), transmission and brake. The IPA utilizes a NN to figure out the relationship between the inputs and outputs through its training process. The task of IPA is to learn a powertrain operation strategy to maximize a reward that is defined to reflect the vehicle performance and efficiency under varied driving conditions. The IPA can be applied to various types of powertrain configurations including conventional ICE vehicle, electric vehicle (EV), and hybrid electric vehicle (HEV). In this paper, a parallel HEV composed of one ICE, one MG, and one 5-speed transmission is used as an example to demonstrate IPA's functionality.

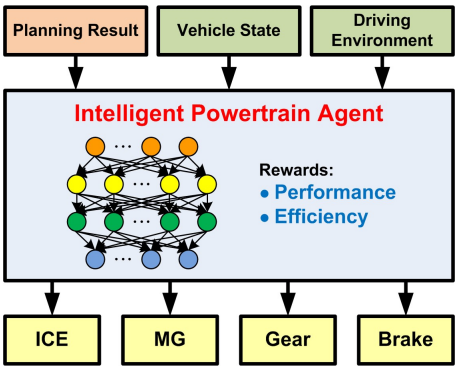

Figure. 3 Architecture for intelligent powertrain agent

This paper is organized as follows: Section II introduces the mathematical models of the powertrain system in the IPA. Section III presents the DQN architecture, states, action, reward and hyperparameters. A rule-based PID controller is also introduced at the end of this section to establish a benchmark for comparison. Section IV describes the learning results and powertrain operation of the IPA. Finally, the findings and the future work of this study are summarized in the conclusion section.

II. Parallel Hybrid Electric Vehicle

In this study, a model of a parallel HEV is used as a case study to explain the formulation and development of the IPA. It should be noticed that in real applications, the IPA does not require a mathematical model in its algorithm, the IPA interacts directly with the real environment and obtains feedback from onboard sensors of a vehicle. However, in order to conduct preliminary training before driving the vehicle on a real way, the powertrain model of a parallel hybrid system is built in a simulation scenario.

The parallel hybrid system investigated in this paper consists of one ICE, one MG, and one 5-speed transmission, its configuration is depicted in Fig. 4, where I_{ice} and I_{mg} are the inertias of the ICE and MG, N_t and N_f represent ratio of the 5-speed transmission and final drive, M_v is the vehicle mass, T_{ice} and T_{mg} are the torques applied to the ICE and MG.

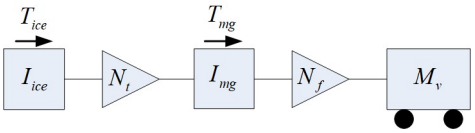

Figure. 4 Configuration of parallel hybrid system

The speed equation of the powertrain is shown in (1), where ω_w, ω_{mg}, and ω_{ice} denote the angular velocities of wheel, MG and ICE. The dynamic equation of the system is arranged in (2), where M_e is the vehicle equivalent mass, r_w is the wheel radius, T_b is the torque applied to the brake, g is the gravitational acceleration (9.81 m/s^2), C_r is the rolling resistance coefficient, ρ is the density of air, A is the vehicle frontal area, and C_d is the aerodynamic drag coefficient. TABLE I lists the parameters and their values used for the simulation. Please note that the MG possesses two operation modes, a positive MG torque corresponds to motor mode and a negative MG torque refers to generator mode.

$$\omega_w = \frac{\omega_{mg}}{N_f} = \frac{\omega_{ice}}{N_t N_f} \tag{1}$$

$$\begin{aligned} &\dot{\omega}_w \left(M_e r_w^2 + I_{mg} N_f^2 + I_{ice} N_t^2 N_f^2 \right) \\ &= T_{ice} N_t N_f + T_{mg} N_f - T_b \\ &- \left[M_v g C_r - 0.5 \rho A C_d \left(\omega_w r_w \right)^2 \right] r_w \end{aligned} \tag{2}$$

TABLE I. Parameters of Parallel Hybrid Vehicle Model

Parameters	Value
Vehicle mass	1400 kg
Vehicle equivalent mass	1600 kg
Transmission ratio	[3.46, 1.75, 1.10, 0.86, 0.71]
Final drive ratio	3.21
Tire radius	0.28 m
Maximum ICE power	101.7Nm @ 4000rpm
Maximum MG torque	305.0Nm @ 940rpm (Motor)
Minimum MG torque	-305.0Nm @ 940rpm (Generator)
Maximum brake torque	2000 Nm
Battery capacity	1.3 kWh
Rolling resistance coefficient	0.009
Air density	1.2 kg/m^3
Vehicle frontal area	2.3 m^2
Aerodynamic drag coefficient.	0.3

III. METHODOLOGY

In order to develop an integrated and intelligent powertrain controller for ADS to perform the driving task used to be handled by a human driver, reinforcement learning (RL) is selected for building the IPA in this study. RL involves two major elements: an agent and an environment, which interacts with each other and the interaction is described with a set of states $\boldsymbol{S}$, and a set actions $\boldsymbol{A}$. By executing an action a_t at time t, the agent transitions from state s_t to state s_{t+1}, and a reward r_t is obtained during the process. The goal of the agent is to maximize its total reward.

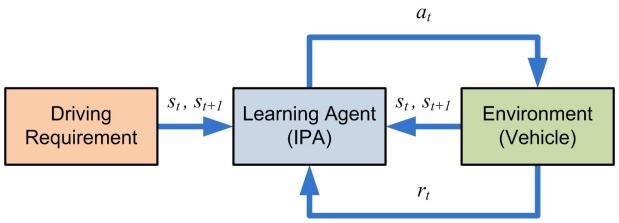

Figure. 5 IPA based on reinforcement learning

The flowchart of IPA based on reinforcement learning architecture is illustrated in Fig. 5. The IPA receives and observes states from two sources, one is from the driving requirement which is the longitudinal vehicle target speed generated from a high-level planner of an ADS, the other part is from the driving environment, i.e., vehicle powertrain conditions. After getting the target speed and observing the current vehicle condition, the IPA chooses an action or operation to the powertrain such as pushing throttle or gear shifting according to its driving policy. After the action is executed, the vehicle changes to its new state and a reward referring to vehicle performance and fuel efficiency is calculated during the process. Then the IPA adjusts its driving policy according to this reward to improve its driving behavior. This process repeats and the IPA can keep improving its driving skill along with its driving experience.

A. Deep Q-learning

The vehicle powertrain control problem can be regarded as a Markov decision process (MDP) when states and actions are observed and commanded in discrete time steps, and the state transition satisfies the Markov property, that is, the next vehicle state s' depends only on the current state s and the decision maker's action a, and is independent of all previous states and actions. Deep Q-learning network is selected as the RL algorithm to solve the MDP in this study. Since the IPA needs to deal with the vehicle dynamics problem including infinite states, which is unmanageable with traditional Q-learning table, a neural network used in the DQN provides a feasible approach to connect the infinite vehicle states and the actions of powertrain operation to solve this problem. The NN structure of IPA is shown in Fig. 6. The Q value of each state and action is recorded and the weights θ of the NN are updated through the learning process. The goal of the DQN is to find an optimal action-value function $Q^*(s, a)$ that achieves the maximum return by following the policy. DQN uses techniques such as experience replay and delayed target network to solve the Bellman equation problem. More detailed introduction of DQN can be found in the reference [7].

After the $Q^*(s, a)$ is found, optimal action under certain state can be selected as $a = \text{argmax}\, Q^*(s, a)$, and an ε-greedy policy can be applied to ensure adequate exploration of the state space. Thereafter, the IPA learns how to operate the powertrain properly under different driving conditions.

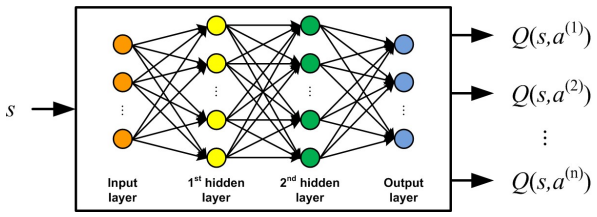

Figure. 6 Structure of the Q-network in IPA

B. States, Actions and Reward

To apply DQN to the control of the parallel hybrid system, states, actions and reward of the IPA are defined in this section. Ten system states including information of ICE, MG, transmission, brake, vehicle and the assigned driving speed requirement are arranged in TABLE II. Please note that the DQN doesn't receive the entire target speed profile but only the current and next target speed at its current time step; this is to ensure that the IPA is feasible in real-time applications. The states are then normalized and input to the DQN.

TABLE II. STATES OF IPA

	States	
1	Current ICE speed	$\omega_{ice}^{(t)}$
2	Current ICE torque	$T_{ice}^{(t)}$
3	Current MG speed	$\omega_{mg}^{(t)}$
4	Current MG torque	$T_{mg}^{(t)}$
5	Current engaged gear number	$N_g^{(t)}$
6	Current brake torque	$T_b^{(t)}$
7	Current battery state of charge (SOC)	$SOC^{(t)}$
8	Current vehicle speed	$V_{vehicle}^{(t)}$
9	Current target speed	$V_{target}^{(t)}$
10	Next target speed	$V_{target}^{(t+1)}$

The action space of the IPA is defined by 11 discrete actions referring to different powertrain operations listed in TABLE III. The mathematic operation of each action is also arranged in the table, where N_g is the gear number of the 5-speed transmission. The actions include torque modification of the ICE, MG and brake, and gear shift of the transmission. Along with the main action being chosen, some auxiliary operations may also be executed to prevent conflict between system units. For example, when action 2, increase ICE torque, is chosen, the brake torque at the next time step will simultaneously be set to zero to avoid unnecessary waste of energy. Increments of the control variables are tuned to ensure operation smoothness in the simulation, where ΔT_{ice} equals to 5Nm , ΔT_{mg} is 10Nm, and ΔT_b is set to be 10Nm. The action 1 which no action is applied is designed to avoid unnecessary or redundant action taken by the IPA.

TABLE III. ACTIONS OF IPA

	Actions	Operation
1	No action	$T_{ice}^{(t+1)} = T_{ice}^{(t)}$, $T_{mg}^{(t+1)} = T_{mg}^{(t)}$, $N_g^{(t+1)} = N_g^{(t)}$, $T_b^{(t+1)} = T_b^{(t)}$
2	Increase ICE torque	$T_{ice}^{(t+1)} = T_{ice}^{(t)} + \Delta T_{ice}$, $T_b^{(t+1)} = 0$
3	Decrease ICE torque	$T_{ice}^{(t+1)} = T_{ice}^{(t)} - \Delta T_{ice}$, $T_b^{(t+1)} = 0$
4	Increase Motor torque	$T_{mg}^{(t+1)} = T_{mg}^{(t)} + \Delta T_{mg}$,if $T_{mg}^{(t)} \geq 0$, $T_b^{(t+1)} = 0$
5	Decrease Motor torque	$T_{mg}^{(t+1)} = T_{mg}^{(t)} - \Delta T_{mg}$,if $T_{mg}^{(t)} \geq 0$, $T_b^{(t+1)} = 0$
6	Increase Generator torque	$T_{mg}^{(t+1)} = T_{mg}^{(t)} - \Delta T_{mg}$,if $T_{mg}^{(t)} < 0$, $T_b^{(t+1)} = 0$
7	Decrease Generator torque	$T_{mg}^{(t+1)} = T_{mg}^{(t)} + \Delta T_{mg}$,if $T_{mg}^{(t)} < 0$, $T_b^{(t+1)} = 0$
8	Shift up one gear	$N_g^{(t+1)} = N_g^{(t)} + 1$, if $N_g^{(t+1)} < \max(N_g)$
9	Shift down one gear	$N_g^{(t+1)} = N_g^{(t)} - 1$, if $N_g^{(t+1)} > \min(N_g)$
10	Increase brake torque	$T_b^{(t+1)} = T_b^{(t)} + \Delta T_b$
11	Decrease brake torque	$T_b^{(t+1)} = T_b^{(t)} - \Delta T_b$

After a certain action is chosen, mechanical and electric constraints of the hybrid system in Equations (3)-(7) are applied to ensure the operation of the selected action is reasonable. The speed and torque range of the ICE and MG are checked, and the state of charge (SOC) of the battery is ensured to maintain in a reasonable range. If the selected action conflicts the constraints, no action will be executed for the specific instant and a punishment term in the instant reward will be applied.

$$\omega_{ice\_idle} \leq \omega_{ice}^{(t)} \leq \omega_{ice\_\max} \tag{3}$$

$$0 \leq T_{ice}^{(t)} \leq T_{ice\_\max} \tag{4}$$

$$0 \leq \omega_{mg}^{(t)} \leq \omega_{mg\_\max} \tag{5}$$

$$T_{mg\_\min} \leq T_{mg}^{(t)} \leq T_{mg\_\max} \tag{6}$$

$$0.2 \leq SOC^{(t)} \leq 0.8 \tag{7}$$

To evaluate the performance of the IPA policy, an instant reward function is proposed in this study as (8). The reward considers the speed chasing ability, instant fuel consumption, battery SOC management, and constraint punishment. (9)-(12), which show the normalization of vehicle speed deviation, ICE fuel consumption, battery SOC and total punishment, respectively. The values of V_n, F_n, SOC_n and PN_n fall in the range between 0 and 1. The exponential function is used to further encourage the DQN to achieve better performance. With the base of each term set to be 0.1, the boundary of each term is between 0.1 and 1, a value close to 1 corresponds to a better performance while a value close to 0.1 implies a worse one. Coefficients c_V, c_F, c_{SOC}, c_{PN} in (8) are the weights to decide the importance of each term, in this study, the weights are set as $c_V = 70$, $c_F = 10$, $c_{SOC} = 10$, $c_{PN} = 10$. Therefore, the total instant reward falls between the range of 0 and 100; by this means, a transparent evaluation can be obtained where a reward close to 100 implies a better performance of the IPA.

$$\begin{aligned} r_{ss'}^{a} &= c_V \times 0.1^{V_n} + c_F \times 0.1^{F_n} \\ &+ c_{SOC} \times 0.1^{SOC_n} + c_{PN} \times 0.1^{PN_n} \end{aligned} \tag{8}$$

$$V_n = \frac{\left| V_{vehicle}^{(t)} - V_{t\arg et}^{(t)} \right|}{\max(V_{vehicle})} \tag{9}$$

$$F_n = \frac{F_{ice}^{(t)}}{\max(F_{ice})} \tag{10}$$

$$SOC_n = \frac{\left| SOC^{(t)} - SOC_{ideal} \right|}{\max(SOC)} \tag{11}$$

$$PN_n = \frac{PN_{total}^{(t)}}{\max(PN_{total})} \tag{12}$$

C. Hyperparameters

The NN structure of the IPA was tuned and a structure of 2 hidden layers with 50 neurons in each layer is found to conduct decent performance and is chosen for further analysis. The learning rate of the DQN is set to be 0.001 through the whole training process and reward decay γ is set to be 0.9. ε-greedy is 0.9 at the beginning of the training to ensure well exploration of all possible states and is gradually decreased to exploit the well-trained policy at the later stage of training. In this case, memory size is set to be 2000 for experience replay and minibatch size is set to be 32. The target network is updated every 100 time steps. RMSProp optimizer is used to optimize the loss function of the DQN.

D. Baseline Rule-Based PID Controller

A rule-based PID controller is established as the benchmark to evaluate the IPA in this study. This comparative controller uses a PID algorithm to generates the driving torque demand T_{demand} for the vehicle to chase the target speed. After the driving torque command is derived, a set of powertrain management rules is applied to decide the operations of each system unit. When $T_{demand} \geq 0$, the motor torque is decided via (13) according the battery SOC, the ICE torque is then decided via (14). When $T_{demand} < 0$, the generator torque is decided via (15). The gear shift rule is shown in (16), the ω_{ice_shift} defined here is set to be 1334 rpm.

$$T_{mg}^{(t+1)} = \begin{cases} \dfrac{T_{demand}^{(t)}}{N_f}, \text{if } SOC^{(t)} \geq 0.4 \\ \dfrac{1}{2}\dfrac{T_{demand}^{(t)}}{N_f}, \text{if } 0.3 \leq SOC^{(t)} < 0.4 \\ \dfrac{1}{3}\dfrac{T_{demand}^{(t)}}{N_f}, \text{if } SOC^{(t)} < 0.3 \end{cases} \tag{13}$$

$$T_{ice}^{(t+1)} = \frac{T_{demand}^{(t)} - N_f T_{mg}^{(t+1)}}{N_t N_f} \tag{14}$$

$$T_{mg}^{(t+1)} = \begin{cases} 0, \text{ if } SOC^{(t)} = 1.0 \\ \dfrac{T_{demand}^{(t)}}{N_f}, \text{ if } SOC^{(t)} < 1.0 \end{cases} \tag{15}$$

$$N_g^{(t+1)} = \begin{cases} N_g^{(t)} - 1, \text{ if } \omega_{ice}^{(t)} \le \omega_{ice\_idle} \\ N_g^{(t)} + 1, \text{ if } \omega_{ice}^{(t)} \ge \omega_{ice\_shift} \\ N_g^{(t)}, \text{otherwise} \end{cases} \quad (16)$$

IV. Simulation and Results

The DQN model of the IPA is developed with Tensorflow script, and the vehicle powertrain environment is built with Python script. For the simulation, the time interval of each time step is 0.5 seconds, initial vehicle speed is 0 km/h, the ideal battery SOC is set to be 0.6. The system states listed in TABLE II are input to the IPA, and the IPA generates an action which will be input to the powertrain environment. Then the system states at the next time step and the reward at the current time step will be calculated. The process repeats and the IPA updates its policy along with the training process.

To acquire states 9 and 10, the current and next target speed, listed in TABLE II, a pre-defined speed profile which represents the speed of a vehicle versus time will be used to pick up the target speed at each time step. Various drive cycles defined by different countries and organizations are used as the speed profile for the simulation. The adopted drive cycles include ECE, EUDC, Japan 1015, UDDS, HD-UDDS, LA92 and HWFET. The ECE cycle simulates a low-speed driving scenario, while EUDC is for middle- to high-speed driving. Japan 1015, UDDS, HD-UDDS and LA92 have combinations of low speed, middle and high speed, and include some stop-and-go scenarios, which are similar to an urban driving scenario. HWFET imitates a highway driving scenario. With various training cycles, different driving environments can be simulated to train the IPA to adapt to varied driving scenarios.

In the training process, the ECE cycle was firstly selected due to its simplicity, after the IPA learned how to operate the powertrain to follow the ECE cycle, other drive cycles were then sequentially input to the Python program. It is found that the training of a previous cycle may contribute to a better performance of the IPA in the next drive cycle. For example, compared to no previous training, the IPA could learn relatively faster how to follow the EUDC cycle when it was previously trained with the ECE cycle. As the IPA was trained with more different cycles, its speed chasing ability while facing a new drive cycle became more accurate and stable. More detail learning results of the IPA for the HD-UDDS cycle is explained as follows.

A. Learning and Convergence

The learning curve of the IPA for the HD-UDDS cycle is shown in Fig. 7. In Fig. 7(a), the reward gradually increases from 50 and converges to 80 after cycle iteration 500. Fig. 7(b) shows the speed chasing ability of the IPA which refers to the speed difference between the target speed and vehicle speed, the decreasing tendency in the first 500 cycle iterations can be observed, indicating the IPA adjusts its control policy to operate the powertrain so that the vehicle can meet the driving speed requirement. Fig. 7(c) illustrates the fuel consumption of each cycle iteration, a decreasing trend indicates the IPA in a process of learning how to save fuel along the training process, which contributes to a more efficient driving behavior.

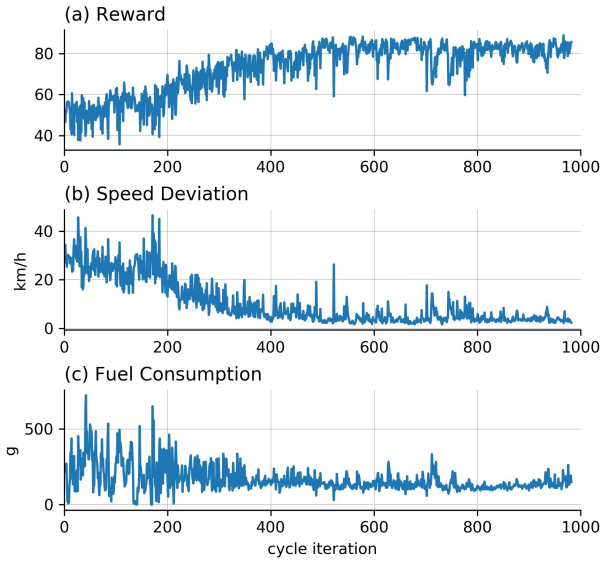

Figure. 7 Learning curve of IPA for HD-UDDS cycle

B. Powertrain Operations

The simulation results of the IPA using DQN algorithm and the comparative rule-based PID controller for the HD-UDDS cycle is shown in Fig. 8 and Fig. 9. It is observed that both the DQN and rule-based PID can chase the target speed accurately, but their operation of the transmission, the engine and the electric machine can be quite different. Fig. 8(b) shows that the rule-based PID controller follows the gear shift rule and switches gear from 1 to 5 strictly according to the vehicle speed. On the other side, the DQN uses a high gear strategy to reduce the ICE speed so that the ICE can operate more in its high efficiency area.

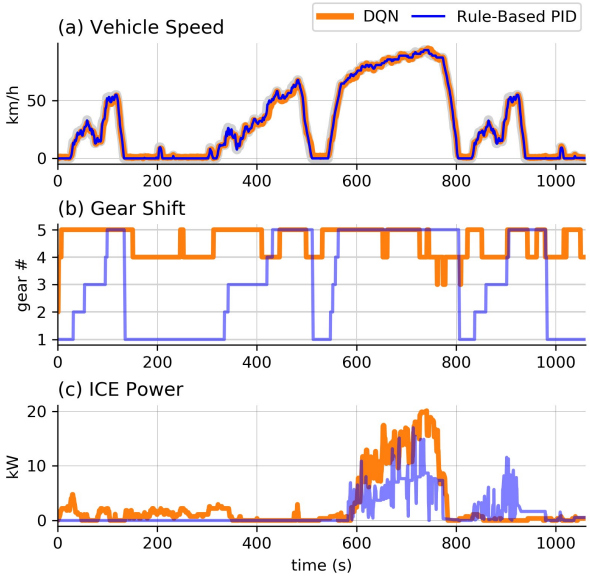

Figure. 8 Vehicle speed and operations of transmission and engine

For power management, according to Fig. 9(a), the rule-based PID controller uses MG power to fulfill the driving requirement when the battery SOC is sufficient and no ICE power is used before 580s as depicted in Fig. 8(c). For the DQN, the ICE outputs power most of the time during the cycle.

It is observed that at 600-800s, the DQN consumes much more ICE power than the rule-based PID, this is because the vehicle is driving at high speed and the ICE is getting the chance to enter its high efficiency area. The DQN seizes this chance and switches the electric machine into generator mode and uses this extra power to charge the battery as shown in Fig. 9(b)(c). During 800-920s, the rule-based PID controller runs out of battery so that it can only use ICE power mainly to drive the rest of the route. On the other side, the DQN has already charged its battery previously, therefore, it can still use MG power to propel the vehicle at the following low speed driving, which is known as the high efficiency operation area for electric machine. The simulation demonstrates that the IPA with DQN algorithm is able to adjust its control strategy through the training process to adapt to different driving environments, and achieve a better power management than the conventional rule-based PID controller.

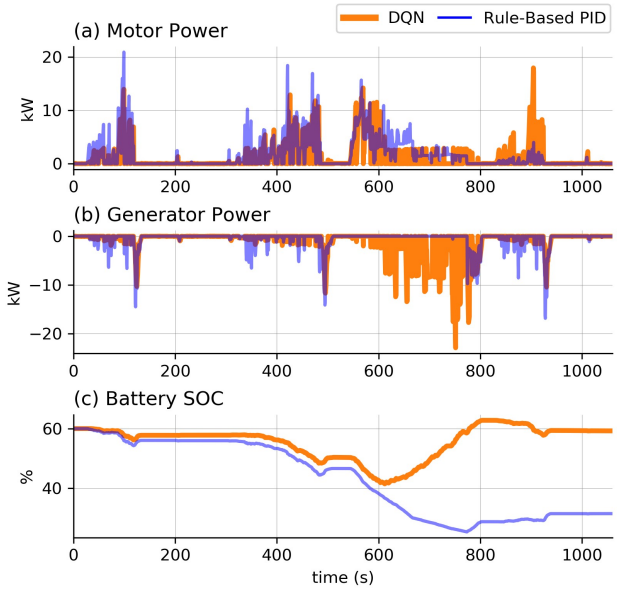

Figure. 9 Operations of motor/generator and battery

V. Conclusion

The development of an intelligent powertrain controller named IPA for ADS using DQN is described in this paper. The IPA receives driving requirement and vehicle information as the states input to the DQN, and the actions taken by the IPA are defined as the powertrain operations. An instant reward is designed to have a transparent view of the performance of the IPA in speed chasing ability and power management. The IPA is applied to a parallel HEV model as a case study to demonstrate its feasibility. The simulation results with various drive cycles show that the IPA possesses the ability to learn from its driving experience to develop reasonable control strategy, and the IPA can evolve itself to fit different driving conditions and be adaptive to change of environment.

Three main contributions of this study are summarized as follows: (1) The fundamental framework of an integrated and intelligent powertrain control strategist IPA for ADS based on DQN is established. (2) An instant reward function is proposed to achieve a transparent evaluation of the learning process of IPA. (3) The simulation results of this study verify the potential of using RL algorithm to take the task of vehicle powertrain control.

Several improvements of deep RL algorithm has been published in recent years, which can be utilized to improve from the original DQN framework adopted in this study. To further improve the learning efficiency and system performance of the IPA, more advanced deep RL techniques such as dueling DDQN[8] will be applied in the following study. On the other hand, safety issue is also a main concern of the applications of the machine learning technology, approaches in literatures on the topic of safe RL, e.g. [9], will be surveyed and applied to the IPA for improving system reliability and stability. An experiment is also planned to test the IPA for practical deployment. These remain topics for future studies.

Acknowledgment

This study is sponsored by Ministry of Science and Technology of Taiwan under the Project No. 107-2917-I-564-039. The authors also wish to thank the support of Berkeley DeepDrive.

References

[1] G. Paganelli, S. Delprat, T. M. Guerra, J. Rimaux, and J. J. Santin, "Equivalent consumption minimization strategy for parallel hybrid powertrains," in *Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No.02CH37367)*, 2002, vol. 4, pp. 2076 - 2081 vol.4.

[2] H. P. Jinming Liu, "Modeling and Control of a Power-Split Hybrid Vehicle," *Control Systems Technology, IEEE Transactions on*, no. 6, pp. 1242 - 1251, 2008.

[3] Y. L. Murphey, J. Park, Z. Chen, M. L. Kuang, M. A. Masrur, and A. M. Phillips, "Intelligent Hybrid Vehicle Power Control #x2014;Part I: Machine Learning of Optimal Vehicle Power," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 8, pp. 3519 - 3530, Oct. 2012.

[4] Y. L. Murphey *et al.*, "Intelligent Hybrid Vehicle Power Control #x2014;Part II: Online Intelligent Energy Management," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 69 - 79, Jan. 2013.

[5] X. Qi, G. Wu, K. Boriboonsomsin, M. J. Barth, and J. Gonder, "Data-Driven Reinforcement Learning - Based Real-Time Energy Management System for Plug-In Hybrid Electric Vehicles," *Transportation Research Record*, vol. 2572, no. 1, pp. 1 - 8, Jan. 2016.

[6] X. Qi, Y. Luo, G. Wu, K. Boriboonsomsin, and M. Barth, "Deep reinforcement learning enabled self-learning control for energy efficient driving," *Transportation Research Part C: Emerging Technologies*, vol. 99, pp. 67 - 81, Feb. 2019.

[7] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529 - 533, Feb. 2015.

[8] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling Network Architectures for Deep Reinforcement Learning," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, 2016, pp. 1995 - 2003.

[9] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe Reinforcement Learning via Shielding," *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.