energies

MDPI

Article

Real-Time HEV Energy Management Strategy Considering Road Congestion Based on Deep Reinforcement Learning

Shota Inuzuka *, Bo Zhang and Tielong Shen

Faculty of Science and Technology, Sophia University, Tokyo 102-8554, Japan; zhangbo@eagle.sophia.ac.jp (B.Z.); tetu-sin@sophia.ac.jp (T.S.)

* Correspondence: shota526@eagle.sophia.ac.jp; Tel.: +81-3-3238-3874

Abstract: This paper deals with the HEV real-time energy management problem using deep reinforcement learning with connected technologies such as Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I). In the HEV energy management problem, it is important to run the engine efficiently in order to minimize its total energy cost. This research proposes a policy model that takes into account road congestion and aims to learn the optimal system mode selection and power distribution when considering the far future by policy-based reinforcement learning. In the simulation, a traffic environment is generated in a virtual space by IPG CarMaker and a HEV model is prepared in MATLAB/Simulink to calculate the energy cost while driving on the road environment. The simulation validation shows the versatility of the proposed method for the test data, and in addition, it shows that considering road congestion reduces the total cost and improves the learning speed. Furthermore, we compare the proposed method with model predictive control (MPC) under the same conditions and show that the proposed method obtains more global optimal solutions.

Keywords: HEV energy management; connected technology; deep reinforcement learning

Citation: Inuzuka, S.; Zhang, B.; Shen, T. Real-Time HEV Energy Management Strategy Considering Road Congestion Based on Deep Reinforcement Learning. *Energies* **2021**, *14*, 5270. https://doi.org/10.3390/en14175270

Academic Editor: Nicu Bizon

Received: 23 July 2021
Accepted: 17 August 2021
Published: 25 August 2021

1. Introduction

In recent years, the electrification of vehicles such as hybrid electric vehicles (HEVs), fuel cell electric vehicles (FCVs), and electric vehicles (EVs), which emit less carbon dioxide than vehicles powered by internal combustion engines, has been spreading rapidly in the market due to regulations on carbon dioxide emissions in various countries. Since HEVs use fossil fuels as their energy source, they do not require any new infrastructure, their popularity has already increased mainly in developed countries, and they are expected to remain in the global market for the next few decades.

The main feature of HEVs is that they are equipped with an engine and a motor for power, and the total energy consumption varies depending on the power distribution. The problem of determining the optimal power distribution to minimize the total energy consumption is called the HEV energy management problem. The HEV energy management problem is an important issue when global environmental problems are becoming more serious and the HEV market is expected to expand in the future.

In general, optimization performance can be improved by using high-performance engines, motors, and batteries and by expanding the feasible region. However, as automated driving becomes more widespread in the future, it will be necessary to install cameras and communication devices, which will further increase the cost of automobiles. Therefore, it is impossible to allocate high costs to the system, and control technology is necessary to maintain high performance even with inexpensive systems. It should be noted that most of the conventional real-time energy management strategies for HEV are rule-based control [1], where the optimality is not directly targeted.

In this HEV energy management problem, off-line optimization problems where the vehicle speed or the demand torque are given in advance based on dynamic programming have been solved [2]. It has been shown that the optimized solution achieves better results

than heuristic rule-based control strategy and indicates further potential optimization, such as distributing the power to the engine when it can be operated most efficiently over the entire vehicle speed. However, since the off-line optimization problem requires the vehicle speed or the demand torque in advance, it cannot be applied to real-time control and is limited to finding new optimization rules.

For real-time energy management, model-based optimization has been investigated using MPC [3]. In this method, the system modeling is important for optimization, and the demand torque constraint is needed to determine the torque of each plant at each time step, thus a prediction model is needed to deal with it as an external constraint. Since these optimization calculations are performed at each step, the computational load is extremely high. Since MPC is an optimization of a finite evaluation interval, it is theoretically impossible to plan the future power distribution beyond the finite evaluation interval.

In recent years, connected technologies such as V2V, which connects vehicles and vehicles, and V2I, which connects vehicles and infrastructure, have been attracting attention, and they can be used as useful information for future prediction for HEV energy management, which needs to be optimized sequentially with predicting the future. For this reason, there are many studies using these technologies not only for automated driving but also for HEV energy management. In [4], a model-based optimization using MPC is performed using a traffic flow model based on Gaussian process using the acceleration of surrounding vehicles and traffic lights as V2V and V2I information. In [5], a proposed controller optimizes the vehicle acceleration, power distribution, and engine operating point based on information such as vehicle speed, vehicle position, road terrain, and speed limit. In [6], the demand torque is predicted based on a Gaussian process using the dynamics of surrounding vehicles, information on traffic lights, and the distance to the intersection, and the optimization is performed by MPC based on these predictions. In this way, connectivity such as V2V and V2I has the potential to further improve the HEV energy management problem.

On the other hand, reinforcement learning has been increasingly researched [7–16] and used in HEV energy management in recent years [17]. Reinforcement learning is based on dynamic programming, which has been used in offline optimization, and uses deep learning to improve policies by learning value functions or state-action value functions from trajectory data. In [18,19], each rotational speed, tilt angle, and State of Charge (*SoC*), which is the charge rate of the battery, etc. are defined as states and a model-free approach using Deep Q-network (DQN) is applied, which shows better performance than traditional rule-based optimization in off-line optimization. Reference [20] compares the performance of Deterministic DP, Stochastic DP, and reinforcement learning on several driving cycles and shows the stability of transfer learning with parameter initialization using Stochastic DP in advance. In [21], optimization is performed by using Proximal Policy Optimization (PPO) [7], which is policy-based reinforcement learning, with vehicle speed, acceleration, and battery *SoC* as states. The paper shows that the parameters can be initialized using the training driving cycle and the optimization for another driving cycle can be learned quickly by transfer learning.

In this research, we assume a connected environment such as V2V and V2I, and deal with the HEV real-time energy management problem using such information. Since the energy cost of HEVs is affected by the behavior of each energy plant, it is necessary to predict the axle speed of the system in the future and optimize the power distribution sequentially with prediction to optimize the total energy cost over the entire time series. We construct a policy model that takes into account the behavior of the vehicle in front and the traffic lights, as well as the congestion of vehicles on the planned route, which has not been taken into account in previous studies, and adapt deep reinforcement learning to search for the optimal solution and optimize whether the vehicle should run in EV mode or HEV mode, and if in HEV mode, how much power should be distributed to each plant with respect to the demand torque. The algorithm adopts PPO, which is a policy-based reinforcement learning, as the objective function for learning and performs

model-free optimization. There are various types of policy-based reinforcement learning, such as Trust Region Policy Optimization (TRPO) [8], however PPO is an algorithm that is easy to implement and has high performance. In Section 4, we construct a simulation environment that enables real-time control and show the versatility of the proposed method, the effectiveness of considering road congestion, and the comparison with an MPC method through simulation.

2. Preliminaries

2.1. System Modeling

In this research, a parallel HEV system with a gearbox, as shown in Figure 1, is targeted. The gearbox is located between the motor and the wheel axle, and there is a clutch between the engine and the motor. The control inputs of this system are the engine torque (τ_e), motor torque (τ_m), and gear numbers of the gearbox (i_g: 6 gears), and the control inputs are denoted as $u = [\tau_e, \tau_m, i_g]$. The clutch is designed to be automatically connected when the engine is activated. Therefore, when the clutch is disconnected, the vehicle runs in EV mode where only the motor transmits power to the wheel axle, and when the clutch is connected, the vehicle runs in HEV mode where not only the motor but also the engine transmits power to the wheel axle. The speed of engine (ω_e) is equal to 0 when the clutch is disconnected as EV mode, and ω_e and the speed of motor (ω_m) are equal when the clutch is connected as HEV mode, and the ratio between them and the speed of the wheel axle is determined by the gear number. The vehicle dynamics model is formulated as shown in Equations (1) and (2) below:

$$M\dot{v} = \frac{\tau_d}{R_t} - \left(\mu Mg + \frac{1}{2}\rho_a A C_d v^2\right) \tag{1}$$

$$\tau_d = (\tau_e + \tau_m) r_{i_g}(i_g) r_{i_0} \tag{2}$$

where M is the weight of the vehicle, v is the vehicle speed, τ_d is the demand torque, R_t is the radius of the tire, μ is the coefficient of rolling resistance, g is the acceleration of gravity, ρ_a is the air density, A is the frontal area of the vehicle, C_d is the coefficient of air resistance, r_{i_g} is the gear ratio of the gears, and r_{i_0} is the differential gear ratio. r_{i_g} is a function of i_g. In Equation (2), τ_e is equal to 0 when the HEV system runs in EV mode, and $\tau_e \geq 0$ when in HEV mode. For the dynamics of the battery, the *SoC* is simply formulated as follows:

$$\dot{SoC} = \frac{-V_{oc} + \sqrt{{V_{oc}}^2 - 4R_b P_b}}{2Q_b R_b} \tag{3}$$

where V_{oc} and R_b are the open-circuit voltage and internal resistance of the battery, respectively, and are functions of *SoC* based on experimental values. Q_b represents the battery capacity, and P_b represents the electrical power of battery and is formulated as follows:

$$P_b = \tau_m \omega_m + P_m^{loss}(\tau_m, \omega_m) \tag{4}$$

where P_m^{loss} is the energy lost in the conversion of electrical energy to mechanical energy and is interpolated based on experimental values as a function of τ_m and ω_m, and P_b is defined as sum of the mechanical energy of motor and P_m^{loss}. In addition, the instantaneous fuel consumption of engine ($\dot{m}_f$) is also an interpolated value based on experimental values, which is a function of ω_e and τ_e.

The system model in this research assumes a connected car that can receive traffic information, such as state of the traffic light, distance to the next traffic light, distance to the vehicle in front, speed and acceleration of the vehicle in front, and road congestion on the planned route.

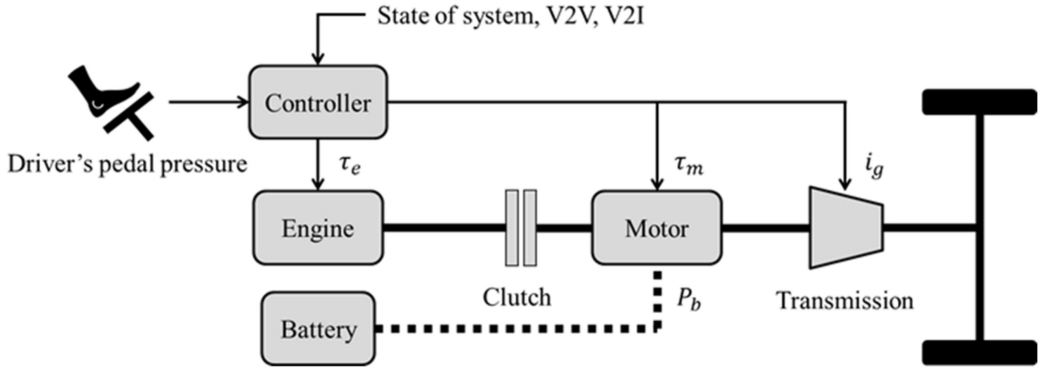

Figure 1. Structure of HEV system.

2.2. Problem Formulation

In this research, the power distribution to the engine and motor is optimized under the demand torque from the driver and the traffic information as connectivity to minimize the fuel consumption of the engine and the electrical energy consumption of the battery as the energy cost of the HEV system. Therefore, the objective function is defined as follows:

$$\min_{u_{1:T}} \sum_{k=1}^{T} \left(\dot{m}_{f_k} - \Gamma \dot{SoC}_k \right) \Delta t \tag{5}$$

where $u_{1:T}$ represents the control inputs from $k = 1$ to T. Γ represents the weight coefficient and can be set freely. Δt is sampling time. For this objective function, there are several constrains as follows, subject to Equations (1)–(4) and (6)–(12):

$$v_0 = v_{init},\ \omega_{e_0} = \omega_{e_{init}},\ \omega_{m_0} = \omega_{m_{init}},\ SoC_0 = SoC_{init} \tag{6}$$

$$SoC_{min} \leq SoC_k \leq SoC_{max} \tag{7}$$

$$P_{b_{min}} \leq P_{b_k} \leq P_{b_{max}} \tag{8}$$

$$0 \leq \tau_{e_k} \leq \tau_{e_{max}}(\omega_{e_k}) \tag{9}$$

$$\tau_{m_{min}}(\omega_{m_k}) \leq \tau_{m_k} \leq \tau_{m_{max}}(\omega_{m_k}) \tag{10}$$

$$\omega_{e_k} \leq \omega_{max} \tag{11}$$

$$\omega_{m_k} \leq \omega_{max} \tag{12}$$

where Equation (6) is the boundary conditions, and v_{init}, $\omega_{e_{init}}$, $\omega_{m_{init}}$, SoC_{init} represent the initial values of vehicle speed, engine speed, motor speed, and SoC. Equation (7) is the upper and lower limits of the SoC, which range from 0(%) to 100(%). Equation (8) is the upper and lower limits of the battery power. Equations (9) and (10) are the upper and lower limits of the engine torque and motor torque. Equations (11) and (12) are the maximum speed constraints for the engine and motor. In this research, τ_d is given from the driver at each time step k, but all τ_d for time series is not known in advance because it is assumed that this research is real-time control. Since the driver demands τ_d considering the traffic situation such as the vehicle speed and acceleration of vehicle in front, the color of traffic light, and the road congestion, it is necessary to plan the optimal power distribution considering V2V and V2I information.

3. Control Design

3.1. Overview

In this research, deep reinforcement learning is applied to the problem formulated in Section 2. Reinforcement learning considers Markov decision process and defines the finite set of states as S, the finite set of actions as A, the transition probability distribution

as $P(s_{k+1}|s_k, a_k)$ $(s \in S, a \in A)$, the stochastic policy as $\pi(a_k|s_k)$, the reward function as $r : S \times A \times S \to \mathbb{R}$, the distribution of the initial state s_0 as $\rho_0(s_0)$, and the discount factor as $\gamma \in (0, 1)$. In policy-based reinforcement learning, the policy is represented as $\pi(a_k|s_k, \theta)$ with parameter θ, and θ is learned from trajectory data to maximize the following expected discounted reward.

$$\max_{\theta} \mathbb{E}_{s_0, a_0, \ldots \sim \rho_0, \pi, P} \left[\sum_{k=0}^{\infty} \gamma^k r_k \right] \tag{13}$$

The structure of control system in this research is as shown in Figure 2. The policy in Figure 2 receives defined states, and the mode selection distribution, engine torque distribution, and gear number distribution of the system are modeled by the neural network. These distributions are used to first sample the mode of the system, and then to select the engine torque and gear number according to the mode. The local controller considers the sampled engine torque and gear number, the state of the system, and the pedal pressure of the driver, and decides the final control input to the system by a rule-based control. The HEV system receives this control input, transitions to the next state, and calculates the reward. The reward is defined as follows to minimize the total energy cost of the HEV system:

$$r_k = -\left(\dot{m}_{f_k} - \Gamma \dot{SoC}_k \right) \Delta t \tag{14}$$

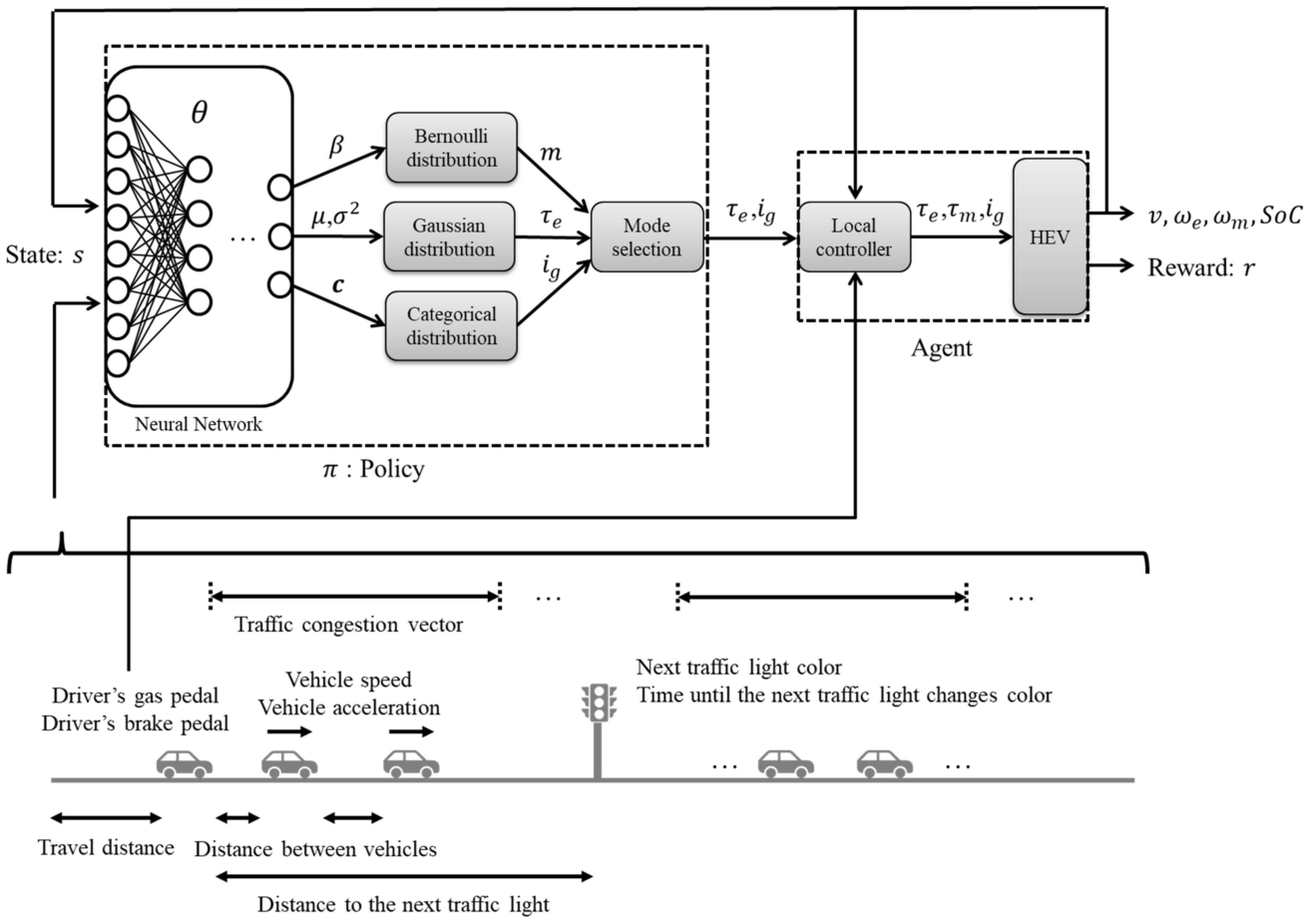

Figure 2. Structure of control system.

The policy learns to maximize this sum of expected discounted reward. In the learning phase, the parameter θ of the policy is trained according to the objective function proposed

in [7], and the advantage function is approximated using the Generalized Advantage Estimator (GAE) proposed in [9]. GAE is modeled using the value function, thus the value function is estimated from the trajectory data by a neural network with parameter ϕ.

3.2. State

The state is summarized as a vector s and inputted into the policy model as shown in Figure 2. Each element of state vector is summarized in Table 1. In addition to the driver's pedal pressure and the state of the system, V2V information such as the distance between vehicles, the behavior of the vehicle in front, the behavior of two vehicles ahead, and the road congestion on the planned route are defined as state, which are related to the future vehicle speed. The i-th element of the traffic congestion vector is defined as the number of vehicles located between $(i-1) \times 100$ (m) and $i \times 100$ (m) from the vehicle. How far ahead to consider can be chosen, and if it is set to 5 [km] ahead, the dimension of the vector is 50. If the remaining distance to run is shorter than the distance considered, the corresponding vector element is set to 0. In addition, the color of the next traffic light, the time until the color of next traffic light changes, and the distance are defined as V2I information as one of the state. Each element is normalized before being input to the policy model.

Table 1. Each element of state vector s.

State: s	Driver's gas pedal pressure [−]
	Driver's brake pedal pressure [−]
	Vehicle speed (v) [m/s]
	Engine speed (ω_e) [rpm]
	Motor speed (ω_m) [rpm]
	Battery charge rate (SoC) [−]
	Distance to the vehicle in front [m]
	Vehicle speed in front [m/s]
	Vehicle acceleration in front [m/s^2]
	Distance between the vehicle in front and two vehicles ahead [m]
	Speed of two vehicles ahead [m/s]
	Acceleration of two vehicles ahead [m/s^2]
	Traffic congestion vector [−]
	Next traffic light color [−]
	Time until the color of next traffic light changes [s]
	Distance to the next traffic light [m]

3.3. Policy Model

The policy model uses the neural network with parameter θ to model the mode selection of system as a Bernoulli distribution, the engine torque selection as a Gaussian distribution, and the gear number selection as a categorical distribution from the state vector s. The model equations for the input layer and hidden layer of the neural network are as follows:

$$h_1 = \tanh(sW_1 + b_1) \tag{15}$$

$$h_k = \tanh(h_{k-1}W_k + b_k) \tag{16}$$

where W is the parameter matrix to be learned, and b is the parameter vector to be learned. Equation (15) represents the equation of input layer, and Equation (16) represents the equation of the k-th hidden layer, and this calculation is repeated from $k = 2$ to $k = N$. The tanh function is adapted as an activation function.

The probability β of the Bernoulli distribution for mode selection is modeled as follows:

$$\beta = \frac{1}{1 + \exp\{-(h_N W_\beta + b_\beta)\}} \tag{17}$$

where W_β is the parameter matrix to be learned, and b_β is the parameter vector to be learned. From the output h_N of the last hidden layer, β is calculated using the sigmoid

function. If this probability β is the probability of driving in the HEV mode, the probability of driving in the EV mode can be modeled as $1 - \beta$.

The equation for the output layer of the Gaussian distribution of engine torque is as follows:

$$\mu = \frac{\tau_{e_{max}}}{1 + \exp\{-(h_N W_\mu + b_\mu)\}} \tag{18}$$

$$\sigma = \frac{\sigma_{e_{max}}}{1 + \exp\{-(h_N W_\sigma + b_\sigma)\}} \tag{19}$$

where μ is the mean value of Gaussian distribution of engine torque and σ is the standard deviation of Gaussian distribution of engine torque. $\tau_{e_{max}}$ represents the maximum value of engine torque and $\sigma_{e_{max}}$ represents the maximum standard deviation. W_μ, W_σ are the parameter matrix to be learned, and b_μ, b_σ are the parameter vector to be learned. From the h_N from the last hidden layer, μ and σ are output using the sigmoid function.

Each probability of the gear number is modeled as following categorical distribution:

$$f_c = h_N W_c + b_c \tag{20}$$

$$c_{i_{g_j}} = \frac{\exp\left(f_{c_j}\right)}{\sum_{j \in J} \exp\left(f_{c_j}\right)} \tag{21}$$

where W_c is the parameter matrix to be learned, and b_c is the parameter vector to be learned. J represents the set of gear number. The f_c corresponding to each gear number is output by Equation (20), and j-th gear number probability $c_{i_{g_j}}$ is modeled using the SoftMax function in Equation (21).

The action decisions in the policy are selected according to the following probability distribution:

$$m \sim \pi_m(m|s) = \beta^m (1-\beta)^{1-m} \tag{22}$$

$$\tau_e \sim \pi_{\tau_e}(\tau_e|s) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left\{ -\frac{(\tau_e - \mu)^2}{2\sigma^2} \right\} \tag{23}$$

$$i_{g_j} \sim \pi_{i_{g_j}}\left(i_{g_j}\middle|s\right) = c_{i_{g_j}} \tag{24}$$

In Equation (22), $m = 0$ represents the EV mode, and $m = 1$ represents the HEV mode. When the selected driving mode is the HEV mode, the selected τ_e is inputted to the local controller, but when the selected driving mode is the EV mode, $\tau_e = 0$ is inputted to the local controller. From the above, the model of the policy is as follows:

$$\pi_\theta(a|s) = \begin{cases} \pi_m(m=0|s)\pi_{i_{g_j}}\left(i_{g_j}\middle|s\right) \\ \pi_m(m=1|s)\pi_{\tau_e}(\tau_e|s)\pi_{i_{g_j}}\left(i_{g_j}\middle|s\right) \end{cases} \tag{25}$$

where in the above case, that is the selected driving mode is the EV mode, $\tau_e = 0$, thus $\pi_{\tau_e}(\tau_e = 0|s) = 1$.

3.4. Local Controller

The local controller receives the engine torque and gear number from the policy, and receives the state of the system and the driver's pedal pressure as inputs to determine the final control inputs to the HEV system. Since deep reinforcement learning sometimes selects actions that are clearly not optimal or do not satisfy the constraints, these actions are adjusted independently by the local controller. The policy considers the local controller and the HEV system as an agent and learns by its input and output.

Rule-based algorithms can be set up appropriately, and in this research, the local controller implements as shown in Figure 3. First, if the vehicle speed is equal to 0 and the demand torque is equal to 0, then the engine torque and motor torque are corrected

to 0 and the gear number is shifted to low. Next, if the vehicle is decelerating, the motor provides all the torque due to braking, and if the speed of each plant exceeds ω_{max}, the gear number is adjusted, where f_{τ_m} is the remaining motor torque when τ_d, τ_e, i_g and v are given based on Equations (1) and (2), and is as follows:

$$f_{\tau_m}(\tau_d, \tau_e, i_g, v) = \frac{\left(M\dot{v} + \mu Mg + \frac{1}{2}\rho_a A C_d v^2\right) R_t}{r_{i_g}(i_g) r_{i_0}} - \tau_e \tag{26}$$

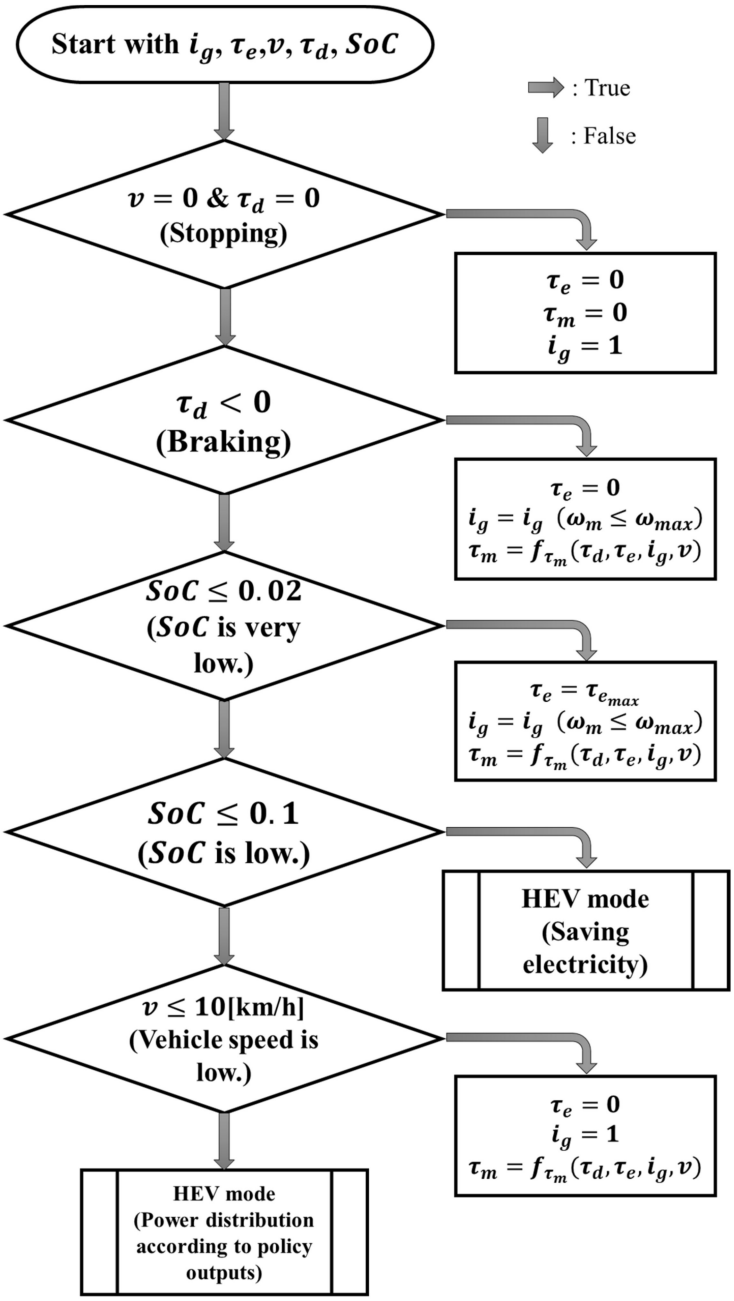

Figure 3. Flowchart in local controller.

Next, if the *SoC* is very low, below 0.02, the engine runs at maximum torque and the power distribution is managed to prevent the *SoC* from decreasing. Next, if the *SoC* is low at 0.1 or less, the system applies rule-based control to conserve electrical energy in HEV mode. Finally, if the vehicle speed is lower than 10 (km/h), the system runs in

EV mode and shifts the gear number to low in order to prevent the engine from running inefficiently, otherwise it manages the power distribution according to τ_e and i_g determined by the policy.

3.5. Proximal Policy Optimization

In the learning phase, the parameter θ of the policy is updated using the gradient method according to PPO proposed in [7]. In PPO, the parameter θ is updated by formulating the expected discounted reward as a surrogate function with trust region as follows:

$$\max_{\theta} \mathbb{E}_{s_0,a_0,\ldots\sim\rho_0,\pi_{\theta_{old}},P}\left[\min\left\{\zeta_k A_{\phi_{old}}(s_k,a_k), clip(\zeta_k, 1-\varepsilon, 1+\varepsilon)A_{\phi_{old}}(s_k,a_k)\right\}\right] \tag{27}$$

$$\zeta_k = \frac{\pi_\theta(a_k|s_k)}{\pi_{\theta_{old}}(a_k|s_k)} \tag{28}$$

where θ_{old} and ϕ_{old} are the parameters θ and ϕ before updating. The probability ratio of the policy to the action a_k before and after updating is clipped to prevent θ from updating too much. $A_{\phi_{old}}(s_k,a_k)$ is the advantage function as a critic, which is modeled by the value function with parameter ϕ_{old} as GAE and can determine whether the state-action value function due to action a_k is better than the value function for s_k. The sign of this advantage function determines the learning direction of the policy for action a_k.

3.6. Generalized Advantage Estimator

As described in [9], the advantage function $A_\phi(s_k,a_k)$ is modeled as GAE based on the estimate of the value function $V_\phi(s_k)$ as follows:

$$A_\phi(s_k,a_k) = \sum_{l=0}^{\infty}(\gamma\lambda)^l \delta_{k+l}^V \tag{29}$$

$$\delta_k^V = r_k + \gamma V_\phi(s_{k+1}) - V_\phi(s_k) \tag{30}$$

where $\lambda \in (0,1)$ is a hyperparameter that adjusts valance of variance and bias. $V_\phi(s_k)$ is the estimate of the value function for s_k by the neural network with parameter ϕ. ϕ is updated from the trajectory data by the following regression problem:

$$\min_{\phi} \sum_{k=0}^{T} ||\hat{V}_k - V_\phi(s_k)||^2 \tag{31}$$

$$\hat{V}_k = \sum_{l=0}^{\infty}\gamma^l r_{k+l} \tag{32}$$

where $\hat{V}_k$ is the sum of the discounted rewards after step k. ϕ is updated based on the gradient method to minimize the squared error.

4. Simulation

4.1. Overview

In the simulation, the traffic environment is defined by IPG CarMaker and the HEV system is modeled by MATLAB/Simulink as shown in Figure 4. These are linked and simulated in the environment where the HEV model runs on the traffic environment defined by CarMaker and the energy cost can be calculated. In this research, we did not use a library to implement the neural network, but made our own program on MATLAB.

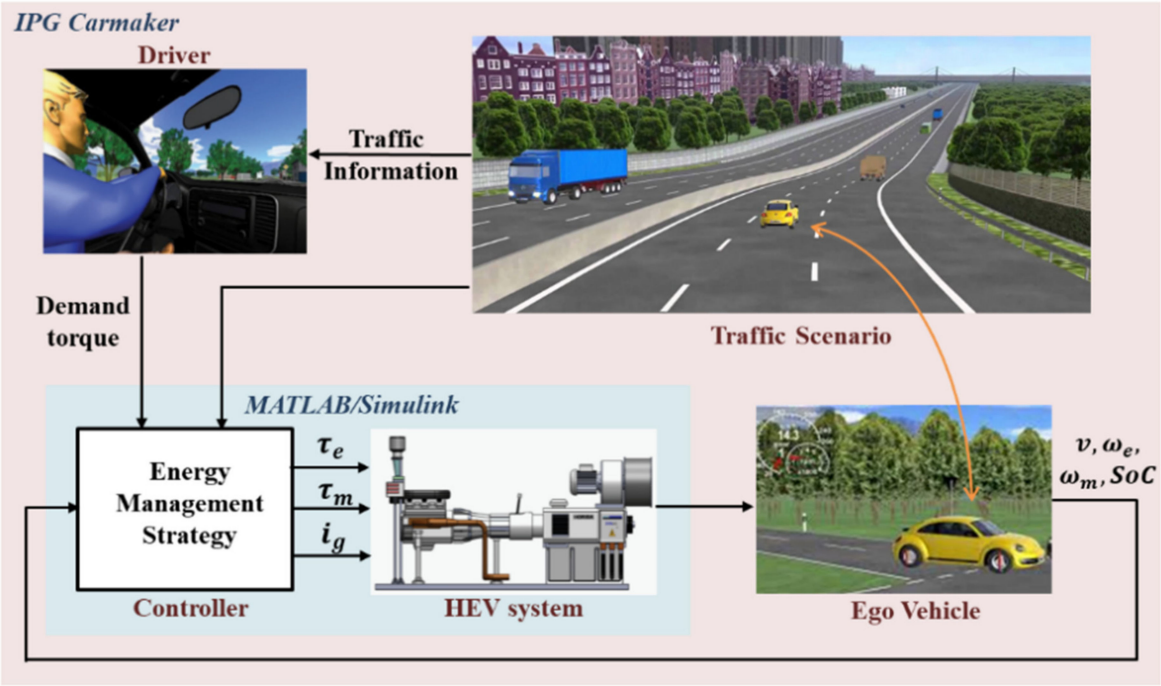

Figure 4. Simulation structure on IPG CarMaker and MATLAB/Simulink.

For the road environment, two road types, Road A and Road B, are prepared as shown in Figures 5 and 6. Both roads have straight lines with a total length of 6 (km), and they are flat without any gradient. Since the number of vehicles in CarMaker is limited, 6 km is the maximum length that can be set. In addition, we set up two roads, one of which has a speed limit of 60 (km/h) for city roads and the other which has a speed limit of 100 (km/h) for country roads and set up one congestion area in each of them. Traffic lights are placed on the city road every 200 to 300 (m), assuming urban areas in Tokyo. Road A is the road that first runs on a city road, then on a country road, and finally on a city road again, while Road B is the road that first runs on a country road, then on a city road. In the simulation, five types of Road A are prepared as training data, and one type each of Road A and Road B are prepared as test data. For the parameters shown in Figures 5 and 6, each road is defined with the setting values as shown in Table 2.

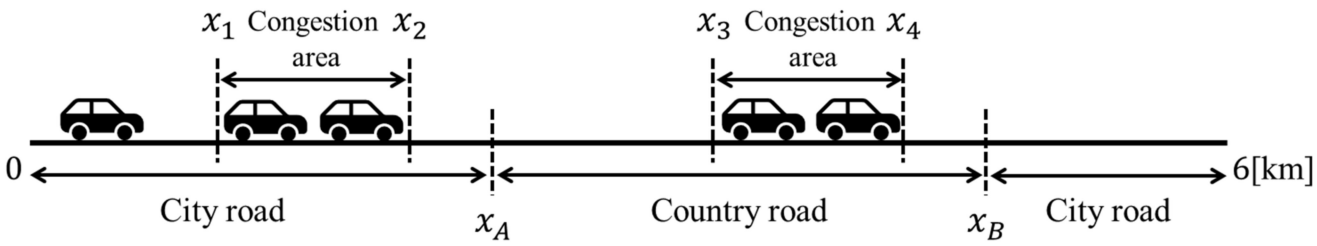

Figure 5. The figure of Road A.

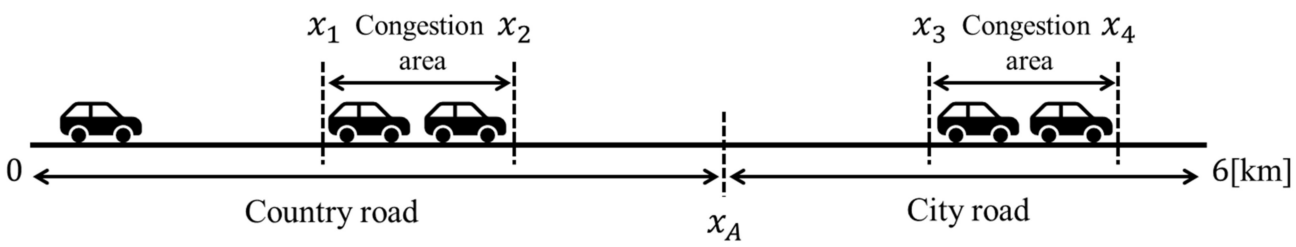

Figure 6. The figure of Road B.

Table 2. The setting values of each road environment.

		x_A [m]	x_B [m]	x_1 [m]	x_2 [m]	x_3 [m]	x_4 [m]
Training data	Road A	1700	5400	800	1200	4100	4700
		2200	5500	1200	1700	4200	4800
		2200	5300	1400	1800	3700	4200
		1800	5300	900	1400	3700	4400
		2000	5700	1100	1500	4500	5000
Test data	Road A	2000	5500	1000	1400	4000	4500
	Road B	3500	-	2000	2600	5000	5400

For the learning simulations, we prepared a total of 50 driving cycles and traffic information patterns as training data in advance, 10 patterns each from the five Road A patterns defined by CarMaker. From these 50 data types, we randomly selected and simulated them during each episode of simulation and used these data for training. In addition, one pattern from each of the defined Road A and Road B was prepared in advance as test data using CarMaker, and these data were used to check the versatility of the trained controller. Figures 7 and 8 show the vehicle speed in the test data for Road A and Road B, respectively. During training, actions are sampled according to the probability distribution of the policy to search for the optimal solution, but when checking the versatility for test data, the maximum action for each probability value is selected for the system mode selection and gear number, and the mean value of the Gaussian distribution is selected for engine torque as the control input.

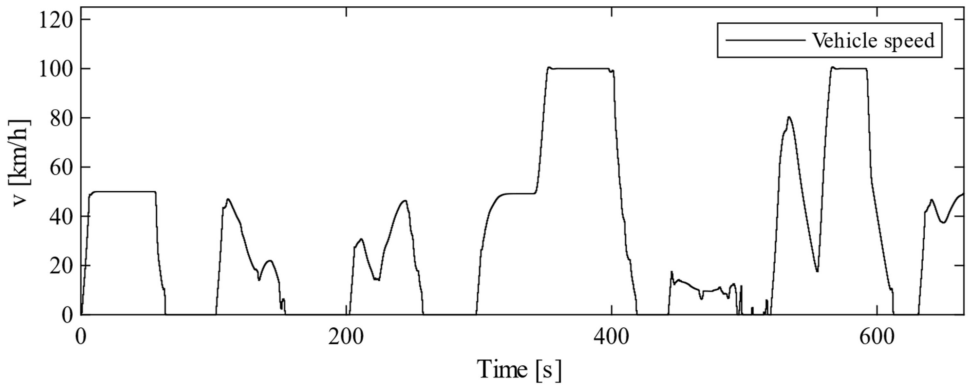

Figure 7. Vehicle speed in the test data for Road A.

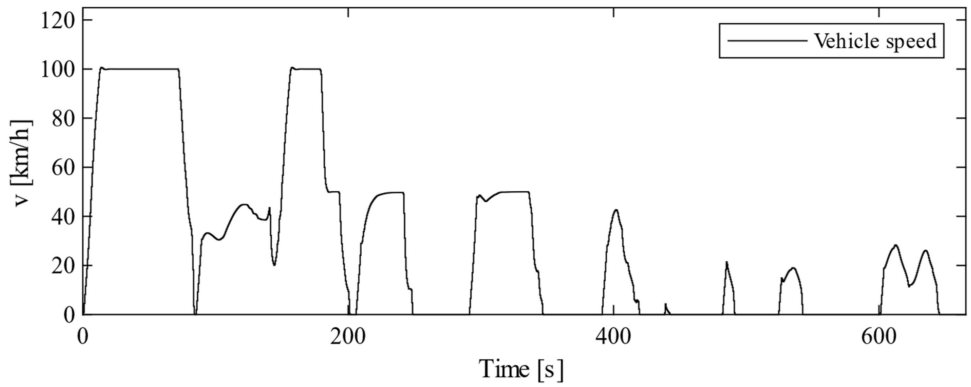

Figure 8. Vehicle speed in the test data for Road B.

In this research, one episode is defined as the completion of driving one road, and the parameters θ and ϕ are updated by Adam [22] every N_{epi} episodes. For the parameter update, minibatch learning is applied, where the number of epochs is K and the number of

minibatches is M. This updating is continued until the learning converges, where N_{iter} is the number of updates. Each hyperparameter during training is defined as in Table 3. In all episodes, the initial speed of the vehicle is equal to 0 (km/h). The number of hidden layers and units are settings where various combinations are tried and the best results are obtained. Γ can be set freely, but in this research, it is set as the value that converts the amount of variation in SoC to the mass of gasoline based on gasoline and electricity prices in Japan. The initial value of SoC is set low so that the vehicle cannot run all 6 (km) of road in EV mode, because the more electrical energy is used in this setting Γ, the more the total cost decreases. Traffic congestion vector in the state s is defined as a vector of dimension 60 that considers 6 (km/h) ahead.

Table 3. The initial values and hyperparameters in the simulation.

Initial Value and Hyperparameter	Value
Δt [s]	0.5
v_{init} [km/h], $\omega_{e_{init}}$ [rpm], $\omega_{m_{init}}$ [rpm]	0
SoC_{init} [%]	15
Number of hidden layers in the policy [−]	2
Number of hidden layers in the value function [−]	2
Number of units in the policy [−]	[64, 64]
Number of units in the value function [−]	[64, 32]
Γ [−]	363.09
$\tau_{e_{max}}$ [Nm]	150
$\sigma_{e_{max}}$ [−]	30
γ [−]	0.9999
λ [−]	0.95
Learning rate [−]	3×10^{-8}
ε [−]	0.2
N_{epi} [−]	20
M [−]	128
K [−]	10

Section 4.2 shows the simulation results of the proposed method. 4.3 shows a comparison with the case where congestion is excluded from state s to demonstrate the effectiveness of considering congestion. In Section 4.4, the performance of the proposed method is compared with that of the method proposed in [6], which is based on real-time optimal control by MPC under the prediction of the demand torque by the Gaussian process on the same system and under the same conditions.

4.2. Simulation Results of the Proposed Method

The simulation results for the total cost at convergence are shown in Table 4, and the changes in the total cost during training and testing for each episode are shown in Figure 9. As shown in Figure 9, the total cost of both the training and test data decreases as the number of episodes increases. Convergence of the total cost requires $N_{iter} = 600$ iterations, and the total cost for the test data on Road A, which has similar features to the training data, is lower than that for the test data on Road B, as shown in Table 4. Although no training is performed on the test data, its total cost decreases. Therefore, the versatility of the proposed method can be shown. The behavior of engine, SoC, and total cost before and after learning for the test data on Road A are shown in Figure 10. From the engine torque and engine speed in Figure 10, it can be seen that when the HEV system runs the HEV mode changes before and after learning. As for the engine speed, the engine operates at around 4000 [rpm], but by changing the gear number through learning, the engine is operated around 2000 [rpm] where the thermal efficiency of the engine is efficient. The terminal value of the SoC after learning in Figure 10 is lower than the initial value, which can be adjusted by Γ.

Table 4. The total cost of proposed method at convergence.

Road Type	Total Cost [g]	Distance Per Liter [km/ℓ]
Test: Road A	149.34	30.13
Test: Road B	158.04	28.47

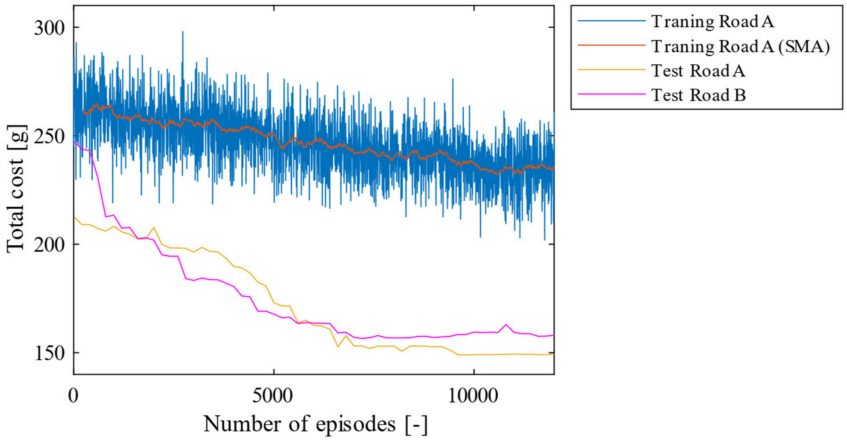

Figure 9. Total cost transition for each episode for the training/testing data by the proposed method. (SMA: Simple Moving Average).

4.3. Simulation Results of the Proposed Method without Traffic Congestion Vector

Section 4.3 shows the effectiveness of considering road congestion on a planned route. We compare the proposed method with the case where the traffic congestion vector is excluded from the state s to show this. The simulation results of the total cost at convergence when the traffic congestion vector is excluded from state s are shown in Table 5, and the changes in the total cost during training and testing for each episode are shown in Figure 11. As shown in Figure 11, the total cost decreases for training and test data as in Figure 9, but the learning speed is slow and it takes $N_{iter} = 4000$ iterations to converge. Comparing Tables 4 and 5, the total cost at learning convergence is lower when road congestion is considered. Similar to 4.2, the total cost for the test data on Road A is lower than that for the test data on Road B. Figure 12 shows the solution for the test data on Road A at the convergence of the proposed method and the solution at the convergence when traffic congestion vector is excluded from the state s. The engine speed and engine torque in Figure 12 show that the solution of the proposed method runs in HEV mode less frequently, activates the engine when the vehicle speed is high, and runs in EV mode when the vehicle speed is low, such as in congestion areas. This indicates that it is possible to determine when the thermal efficiency of the engine becomes efficient and when it is appropriate to run the engine for the entire route based on the road congestion on the planned route, which is related to the vehicle speed in the far future. On the other hand, when the road congestion is not considered, only the behavior of the vehicle in front and the next traffic light, which are related to the vehicle speed in the relatively near future, are considered, indicating that it is not possible to determine when the engine should be run for the entire route.

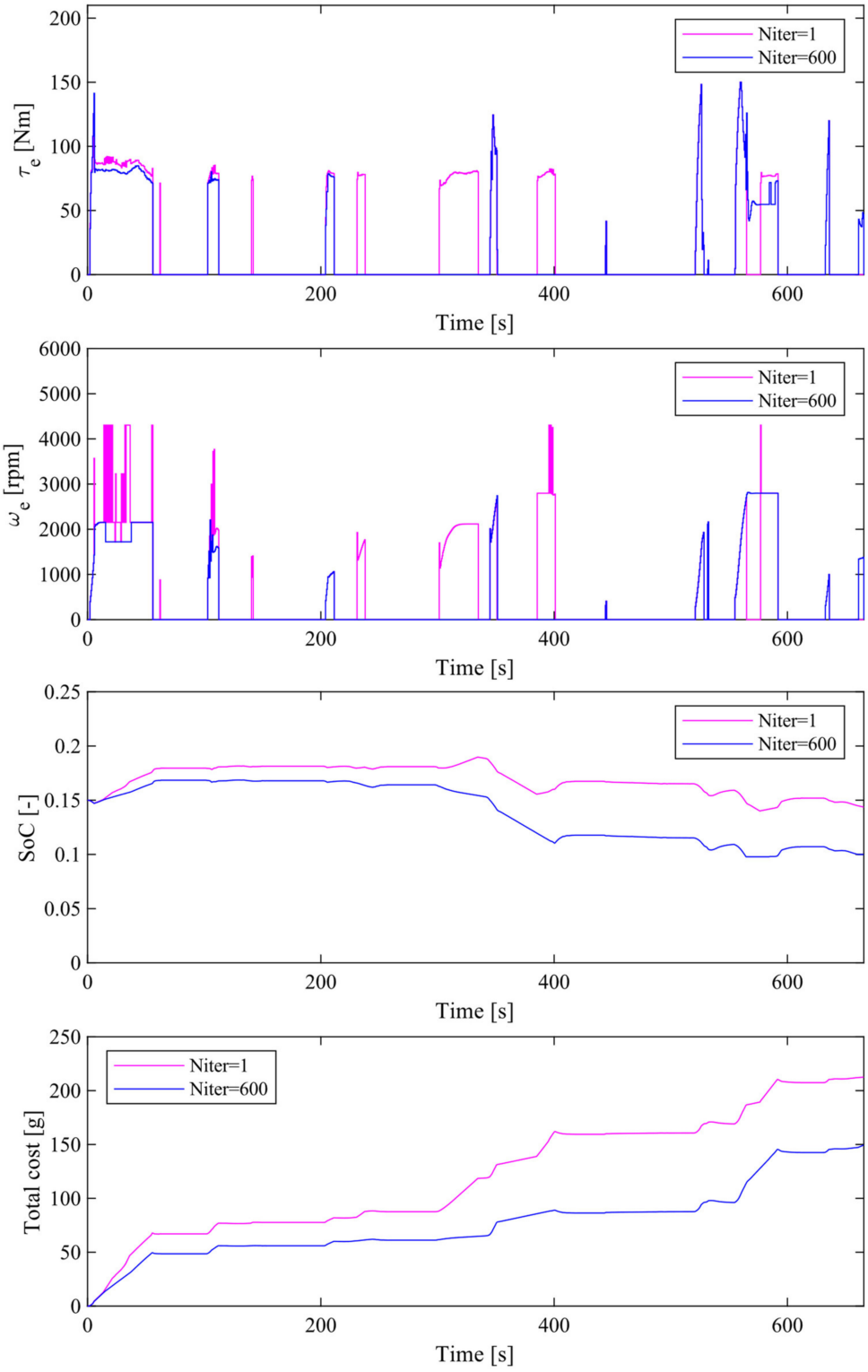

Figure 10. Simulation results of the proposed method at $N_{iter} = 1$ and $N_{iter} = 600$ for the test data on Road A (engine torque, engine speed, *SoC*, and total cost).

Table 5. The total cost of proposed method without the traffic congestion vector at convergence.

Road Type	Total Cost [g]	Distance per Liter [km/ℓ]
Test: Road A	191.71	23.47
Test: Road B	205.41	21.91

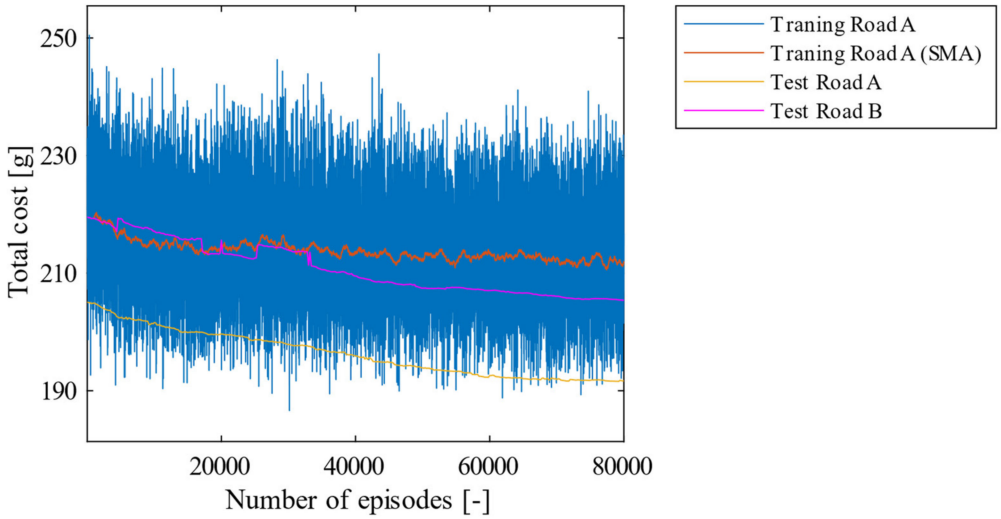

Figure 11. Total cost transition for each episode for the training/testing data by the proposed method without the traffic congestion vector. (SMA: Simple Moving Average).

4.4. Comparison of Simulation Results between the Proposed Method and MPC

In Section 4.4, we compare the performance of our method with that of MPC, which has recently attracted attention as a solution method for HEV real-time energy management problems. The MPC method is proposed in [6] and is a solution that uses the prediction of the driver's demand torque as an external condition based on a Gaussian process using V2V and V2I information such as the behavior of the vehicle in front and information from traffic lights. The system model, problem setting, constraint conditions, training data, and test data are all the same to ensure accurate comparison. The prediction model of the driver's demand torque based on the Gaussian process is trained using the training data, and the real-time optimization problem is solved using the trained prediction model and MPC algorithm for the test data. Table 6 shows the total cost simulation results of the MPC method, and Figure 13 shows the comparison of the simulation results of the proposed method and the MPC method for the test data on Road A and Figure 14 on Road B. Comparing Tables 4 and 6, the total cost for the test data on Road A, which has similar characteristics to the training data, show that the performance of the proposed method outperforms the MPC method by 1.1 (km/h) and the total cost for the test data on Road B, which has different characteristics from the training data, show that the performance of the proposed method outperforms the MPC method by 2.45 (km/h). Figures 13 and 14 show that the proposed method activates the engine when the vehicle speed around the start is relatively high and stable. This is because the initial value of *SoC* starts at a relatively low value of 15(%), and the electrical energy is saved around the start to run in EV mode in areas where the thermal efficiency of the engine deteriorates, such as congestion areas. Since the proposed method mainly activates the engine around the start, the total cost of the proposed method is larger than that of the MPC method in the first half, but instead, the *SoC* of the proposed method is larger than the initial value, as shown in Figures 13 and 14. Then, the electrical energy charged in the first half is used to select the EV mode in areas where the vehicle speed is relatively low, such as congestion areas after that. This means that the information on when the engine can be activated efficiently along the entire route can be obtained while considering the road congestion on the planned route and the *SoC*, and the global optimal solution can be selected while considering the value function based on dynamic programming rather than minimizing the instantaneous cost. On the other hand, the MPC method selects the EV mode when the vehicle speed is low, and the engine is activated at high speed and high power when large acceleration is expected.

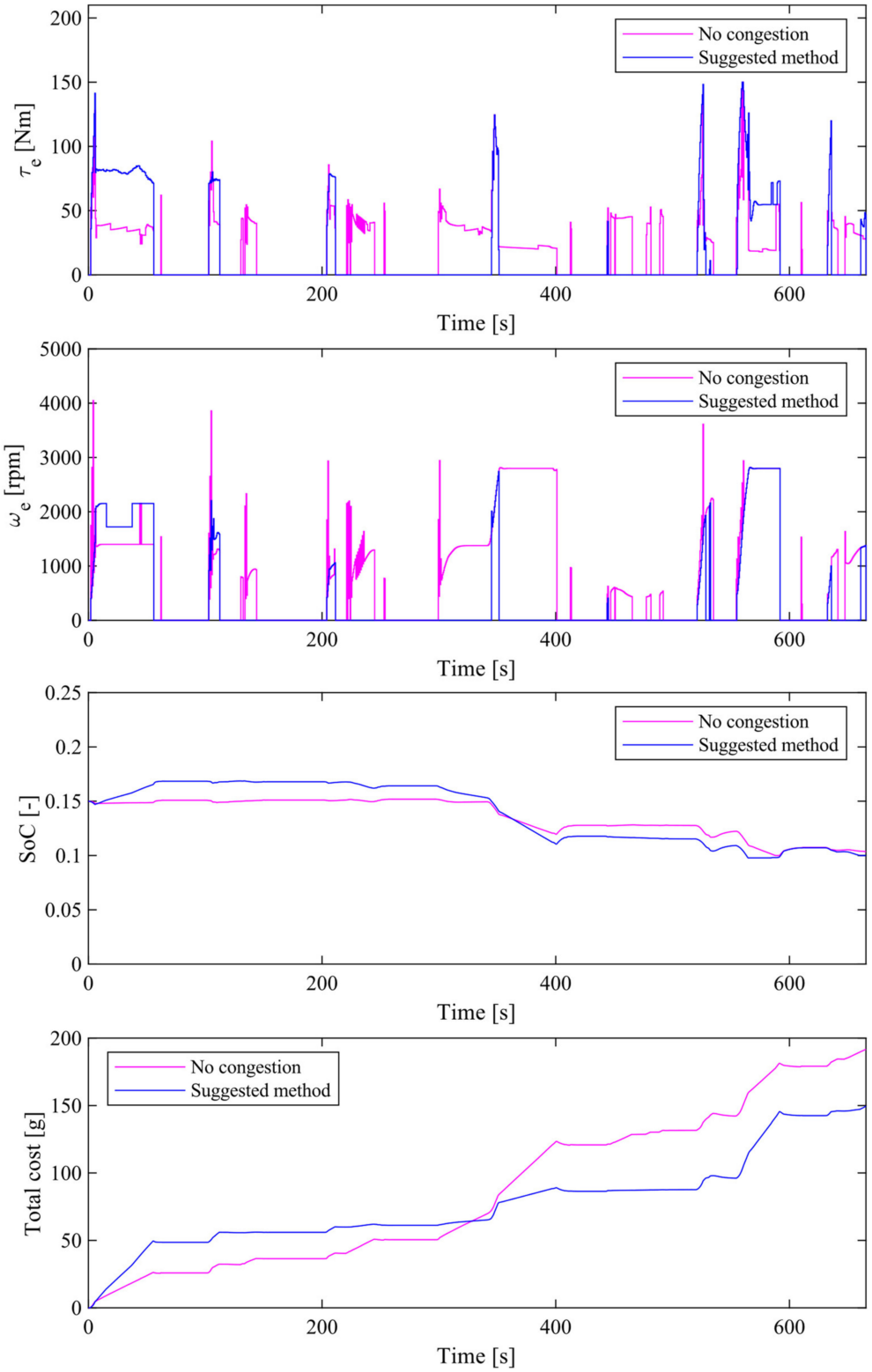

Figure 12. Simulation results of the proposed method and without the traffic congestion vector for the test data on Road A (engine torque, engine speed, *SoC*, and total cost).

Table 6. The total cost of MPC method at convergence.

Road Type	Total Cost [g]	Distance per Liter [km/ℓ]
Test: Road A	155.01	29.03
Test: Road B	172.93	26.02

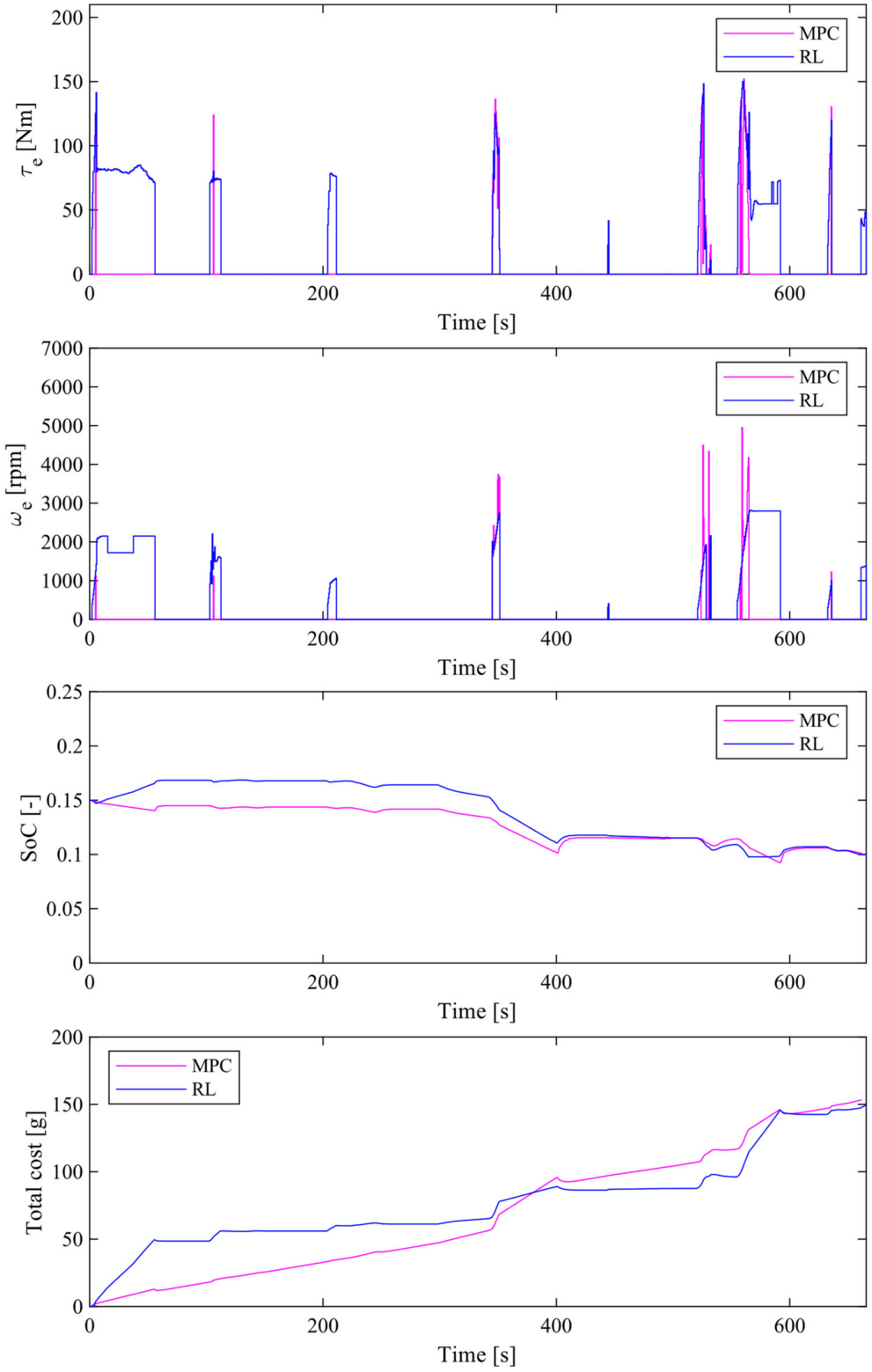

Figure 13. Simulation results of the proposed method and the MPC method for the test data on Road A (engine torque, engine speed, *SoC*, and total cost).

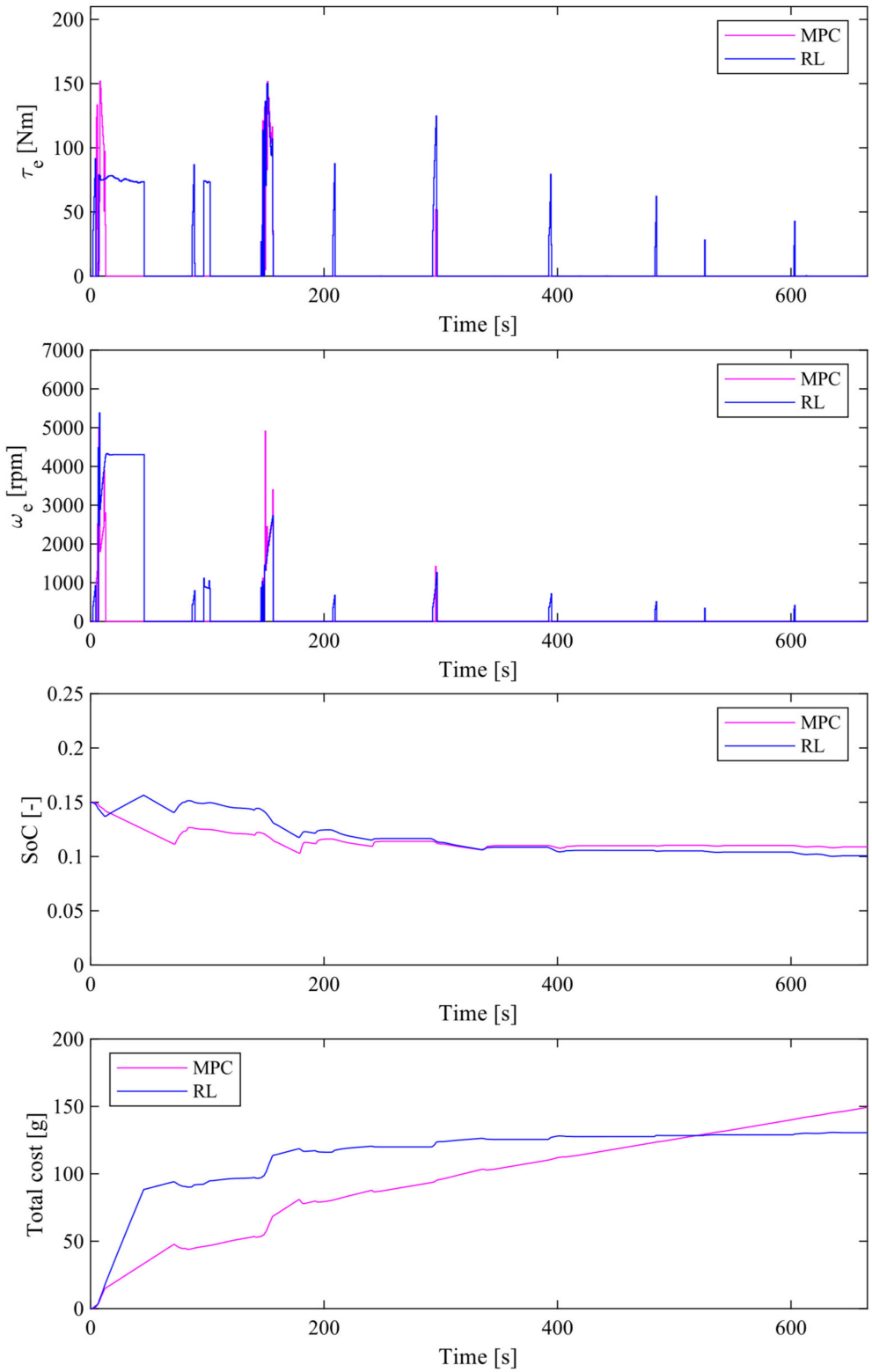

Figure 14. Simulation results of the proposed method and the MPC method for the test data on Road B (engine torque, engine speed, *SoC*, and total cost).

5. Conclusions

In this research, we proposed the policy-based deep reinforcement learning method for the HEV energy management problem that considers the road congestion on the planned route in the environment of V2V and V2I, which are connected technologies, and adopts the neural network to learn the system mode, engine torque, and gear number selection by using Bernoulli distribution, Gaussian distribution, and categorical distribution, respectively. The local controller is placed between these policy models and the system model, and the policy model is trained by eliminating solutions that do not satisfy the constraints or that are clearly not optimal.

In the simulation validation, we prepared 50 kinds of training data and tested data with similar features to the training data and test data without similar features to the training data to show the versatility of the proposed method. The simulation results show that the total cost is minimized for both test data, but the simulation for the test data with similar features to the training data has better performance. The versatility of the policy is expected to be further improved as more training data with various characteristics are collected. Therefore, it is necessary to increase the number of training data, and although only one pattern from each of Road A and Road B is used as test data to accurately compare the performance of the controller before and after learning in this research, it is necessary to simulate various test data as future work.

By considering road congestion, this research aims to control the engine to operate relatively efficiently over the entire route, not only in the relatively near future but also in the far future, while considering the *SoC*. The results are somewhat dependent on the initial values of the parameters, but in most cases the learning convergence is faster and the total cost at convergence is minimized when the road congestion is considered. In addition, the optimal solution does not minimize the instantaneous cost, but minimizes it globally over the entire route, considering the value function at the transition states. However, for the future demand of HEVs for mid-to-long distances, the simulation has not been sufficiently verified for long distance roads due to the limitation of the number of vehicles in CarMaker, and this remains a future work.

In Section 4.4, the proposed method is compared with the MPC method, which has been attracting attention as a solution for HEV real-time energy management problems. The results show that the proposed method equals or outperforms the MPC method in terms of total energy cost. Since MPC solves the optimization for each sample period, its evaluation interval cannot be set long enough for problems with short sample periods and nonlinearities such as HEV energy management in terms of computational load. On the other hand, in the proposed method we define factors that are related to vehicle speed in the far future such as road congestion as state and estimate the value function in each state from trajectory data. By using the value functions, the policy can be trained considering the far future.

Author Contributions: Conceptualization, S.I.; validation, B.Z.; supervision, T.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hofman, T.; Steinbuch, M.; van Druten, R.M.; Serrarens, A.F.A. Rule-Based Equivalent Fuel Consumption Minimization Strategies for Hybrid Vehicles. In Proceedings of the 17th IFAC World Congress, Seoul, Korea, 6–11 June 2008; Volume 41, pp. 5652–5657. [CrossRef]
2. Wang, R.; Lukic, S.M. Dynamic programming technique in hybrid electric vehicle optimization. In Proceedings of the 2012 IEEE International Electric Vehicle Conference, Greenville, SC, USA, 4–8 March 2012. [CrossRef]

3. Borhan, H.; Vahidi, A.; Phillips, A.M.; Ming, L.; Kuang, I.; Kolmanovsky, V.; Di Cairano, S. MPC-based energy management of a power-split hybrid electric vehicle. *IEEE Trans. Control Syst. Technol.* **2012**, *20*, 593–603. [CrossRef]
4. Xu, F.; Shen, T. Look-Ahead Prediction-Based Real-Time Optimal Energy Management for Connected HEVs. *IEEE Trans. Veh. Technol.* **2020**, *69*, 2537–2551. [CrossRef]
5. Hu, J.; Shao, Y.; Sun, Z.; Wang, M.; Bared, J.; Huang, P. Integrated optimal eco-driving on rolling terrain for hybrid electric vehicle with vehicle-infrastructure communication. *Transp. Res. Part C Emerg. Technol.* **2016**, *68*, 228–244. [CrossRef]
6. Zhang, B.; Zhang, J.; Xu, F.; Shen, T. Optimal control of power-split hybrid electric powertrains with minimization of energy consumption. *Appl. Energy* **2020**, *266*, 114873. [CrossRef]
7. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
8. Schulman, J.; Levine, S.; Moritz, P.; Jordan, M.; Abbeel, P. Trust Region Policy Optimization. *arXiv* **2015**, arXiv:1502.05477.
9. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; Abbeel, P. High-Dimensional Continuous Control Using Generalized Advantage Estimation. *arXiv* **2015**, arXiv:1506.02438.
10. Heess, N.; Dhruva, T.B.; Sriram, S.; Lemmon, J.; Merel, J.; Wayne, G.; Tassa, Y.; Erez, T.; Ziyu Wang, S.M.; Eslami, A.; et al. Emergence of Locomotion Behaviours in Rich Environments. *arXiv* **2017**, arXiv:1707.02286.
11. Mnih, V.; Puigdomenech Badia, A.; Mirza, M.; Alex Graves, L.T.P.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. *arXiv* **2016**, arXiv:1602.01783.
12. Martens, J.; Sutskever, I. Training deep and recurrent networks with hessian-free optimization. In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 479–535. [CrossRef]
13. Pascanu, R.; Bengio, Y. Revisiting Natural Gradient for Deep Networks. *arXiv* **2013**, arXiv:1301.3584.
14. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous Control with Deep Reinforcement Learning. *arXiv* **2015**, arXiv:1509.02971.
15. Van Hasselt, H.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-Learning. *arXiv* **2016**, arXiv:1509.06461.
16. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized Experience Replay. *arXiv* **2016**, arXiv:1511.05952.
17. Hu, Y.; Li, W.; Xu, K.; Zahid, T.; Qin, F.; Li, C. Energy Management Strategy for a Hybrid Electric Vehicle Based on Deep Reinforcement Learning. *Appl. Sci.* **2018**, *8*, 187. [CrossRef]
18. Liessner, R.; Schroer, C.; Dietermann, A.; Baker, B. Deep Reinforcement Learning for Advanced Energy Management of Hybrid Electric Vehicles. In Proceedings of the 10th International Conference on Agents and Artificial Intelligence, Madeira, Portugal, 16–18 January 2018. [CrossRef]
19. Yang, C.; Zhou, C. An Energy Management Strategy of Hybrid Electric Vehicles based on Deep Reinforcement Learning. *Int. J. Eng. Adv. Res. Technol.* **2019**, *5*, 1–4.
20. Lee, H.; Song, C.; Kim, N.; Cha, S.W. Comparative Analysis of Energy Management Strategies for HEV: Dynamic Programming and Reinforcement Learning. *IEEE Access* **2020**, *8*, 67112–67123. [CrossRef]
21. Liu, T.; Wang, B.; Tan, W.; Lu, S.; Yang, Y. Data-Driven Transferred Energy Management Strategy for Hybrid Electric Vehicles via Deep Reinforcement Learning. *arXiv* **2020**, arXiv:2009.03289.
22. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.