Energies **2015**, *8*, 7243-7260; doi:10.3390/en8077243

OPEN ACCESS

energies

ISSN 1996-1073

www.mdpi.com/journal/energies

Article

Reinforcement Learning–Based Energy Management Strategy for a Hybrid Electric Tracked Vehicle

Teng Liu, Yuan Zou *, Dexing Liu and Fengchun Sun

Collaborative Innovation Center of Electric Vehicles in Beijing, School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China; E-Mails: liuteg123@bit.edu.cn (T.L.); liudex@bit.edu.cn (D.L.); sunfch@bit.edu.cn (F.S.)

* Author to whom correspondence should be addressed; E-Mail: zouyuan@bit.edu.cn; Tel./Fax: +86-10-6894-4115.

Academic Editors: Joeri Van Mierlo, Ming Cheng, Omar Hegazy and Wei Hua

Received: 14 January 2015 / Accepted: 29 June 2015 / Published: 16 July 2015

Abstract: This paper presents a reinforcement learning (RL)–based energy management strategy for a hybrid electric tracked vehicle. A control-oriented model of the powertrain and vehicle dynamics is first established. According to the sample information of the experimental driving schedule, statistical characteristics at various velocities are determined by extracting the transition probability matrix of the power request. Two RL-based algorithms, namely *Q*-learning and *Dyna* algorithms, are applied to generate optimal control solutions. The two algorithms are simulated on the same driving schedule, and the simulation results are compared to clarify the merits and demerits of these algorithms. Although the *Q*-learning algorithm is faster (3 h) than the *Dyna* algorithm (7 h), its fuel consumption is 1.7% higher than that of the *Dyna* algorithm. Furthermore, the *Dyna* algorithm registers approximately the same fuel consumption as the dynamic programming–based global optimal solution. The computational cost of the *Dyna* algorithm is substantially lower than that of the stochastic dynamic programming.

Keywords: reinforcement learning (RL); hybrid electric tracked vehicle (HETV); *Q*-learning algorithm; *Dyna* algorithm; dynamic programming (DP); stochastic dynamic programming (SDP)

---

1. Introduction

In recent years, hybrid electric vehicles (HEVs) are being widely used for reducing fuel consumption and emissions. In these vehicles, an energy management strategy controls the power distribution among multiple energy storage systems [1,2]. This strategy realizes several control objectives, such as the driver's power demand, optimal gear shifting, and battery state-of-charge (SOC) regulation. Many optimal control methods have been proposed for designing energy management strategies in HEVs. For instance, because vehicles follow a certain driving cycle, the deterministic dynamic programming (DDP) approach can be used to obtain global optimal results [3–5]. In addition, previous studies have applied the stochastic dynamic programming (SDP) approach to utilize the probabilistic statistics of the power request [6,7]. Pontryagin's minimum principle was introduced in [8,9] and an equivalent consumption minimization strategy was suggested in [10–12] to obtain optimal control solutions. Furthermore, a model predictive control was introduced in [13] and convex optimization was presented in [14]. Recently, game theory [15] and reinforcement learning (RL) [16] have attracted research attention for HEV energy management. RL is a heuristic learning method applied in numerous areas, such as robotic control, traffic improvement, and energy management. For example, previous studies have applied RL approaches for robotic control and for enabling robots to learn and adapt to situations online [17,18]. Furthermore, [19] proposed an RL approach for enabling a set of unmanned aerial vehicles to automatically determine patrolling patterns in a dynamic environment.

The aforementioned RL studies have not evaluated energy management strategies for HEVs. A power management strategy for an electric hybrid bicycle was presented in [20]; however, the powertrain is simpler than that in HEVs and the power is not distributed among multiple power sources. In the current study, RL was applied to solve an energy management problem of a hybrid electric tracked vehicle (HETV). Statistical characteristics of an experimental driving schedule were extracted as a transition probability matrix of the power request. The energy management problem was formulated as a stochastic nonlinear optimal control problem with two state variables, namely the battery SOC and rotational speed of the generator, and one control variable, namely the engine throttle signal. Subsequently, the *Q*-learning and *Dyna* algorithms were applied to determine an energy management strategy for improving the fuel economy performance and achieving battery charge sustenance. Furthermore, the RL-based energy management strategy was compared with the dynamic programming (DP)–based energy management strategy. The simulation results indicated that the *Q*-learning algorithm entailed a lower computational cost (3 h) compared with the *Dyna* algorithm (7 h); nevertheless, the fuel consumption of the *Q*-learning algorithm was 1.7% higher than that of the *Dyna* algorithm. The *Dyna* algorithm registered almost the same fuel consumption as the DP-based global optimal solution. The *Dyna* algorithm is computationally more effective than SDP. However, because of their computational burdens, the *Q*-learning and *D*yna algorithms cannot be used in current online operations, and further research on real-time applications is required.

The remainder of this paper is organized as follows: in Section 2, a hybrid powertrain is modeled and the optimal control problem is formulated. In Section 3, a statistical information model that is based on the experimental driving schedule is developed, and the *Q*-learning and *Dyna* algorithms are presented. The RL-based energy management strategy is compared with the DP-, and SDP-based energy management strategies in Section 4. Section 5 concludes this paper.

2. Hybrid Powertrain Modeling

Figure 1 shows a heavy-duty HETV with a dual-motor drive structure. The powertrain comprises two main power sources: an engine-generator set (EGS) and a battery pack. The dashed arrow lines in the figure indicate the directions of power flows. To guarantee a quick and adequately precise simulation, a quasi-static modeling methodology [21] was used to model the power request of the hybrid powertrain. Table 1 lists the vehicle parameters used in the model.

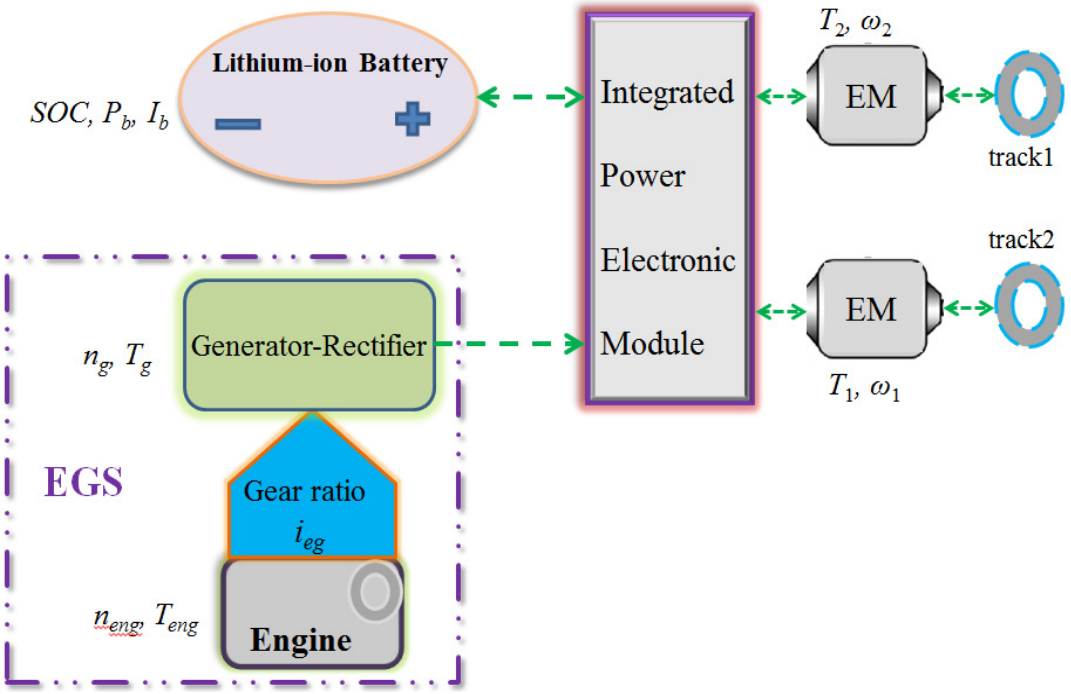


Figure 1. Powertrain configuration of the HETV.

Table 1. HETV Parameters.

Parameter	Symbol	Value
Sprocket radius	r	0.313 m
Inertial yaw moment	I_z	55,000 kg·m^2
Motor shafts efficiency	η	0.965
Gear ratio param.	i_0	13.2
Vehicle tread	B	2.55 m
Curb weight	m_v	15,200 kg
Gravit. constant	g	9.81 m/s^2
Rolling resis. coefficient	f	0.0494
Contacting track width	L	3.57 m
Motor efficiency	η_{em}	0.9
Electromotive force param.	K_e	1.65 Vsrad^{-2}
Electromotive force param.	K_x	0.00037 NmA^{-2}
Generator inertia	J_g	2.0 kg·m^2

Table 1. *Cont.*

Parameter	Symbol	Value
Engine inertia	J_e	3.2 kg·m^2
Gear ratio param.	i_{eg}	1.6
Battery capacity	Q_b	50 Ah
Min. engine speed	$n_{\text{eng,min}}$	650 rpm
Max. engine speed	$n_{\text{eng,max}}$	2100 rpm
Min. *SOC*	$SOC_{\min}$	0.2
Max. *SOC*	$SOC_{\max}$	0.8

2.1. Power Request Model

Assume that only longitudinal motions are considered [4]; the torque of the two motors is calculated as follows:

$$T_1 = (\frac{F_1 r}{i_0 \eta} - \frac{Mr}{Bi_0 \eta}) + [\frac{m_v r^2}{i_0^2 \eta} \frac{R}{R - B/2} - \frac{I_z r^2}{i_0^2 \eta B(R - B/2)}]\dot{\omega}_1 \quad (1)$$

$$T_2 = (\frac{F_2 r}{i_0 \eta} + \frac{Mr}{Bi_0 \eta}) + [\frac{m_v r^2}{i_0^2 \eta} \frac{R}{R + B/2} + \frac{I_z r^2}{i_0^2 \eta B(R - B/2)}]\dot{\omega}_2 \quad (2)$$

where T_1 and T_2 are the torque of the inside and outside motors, respectively, and ω_1 and ω_2 are the rotational speed of the inside and outside sprockets, respectively; r is the radius of the sprocket, I_z is the yaw moment of inertial, η is the efficiency from the motor shafts to the tracks, i_0 is the fixed gear ratio between motors and sprockets, B is the vehicle tread, R is the turning radius of the vehicle, m_v is the curb weight, and F_1 and F_2 are the rolling resistance forces of the two tracks. The yaw moment from the ground M is evaluated as follows:

$$M = \frac{1}{4} u_t m_v g L \quad (3)$$

where g is the acceleration of gravity and L is the track contact length. The lateral resistance coefficient u_t is computed empirically [22]:

$$u_t = u_{\max}(0.925 + 0.15R/B)^{-1} \quad (4)$$

where u_{max} is the maximum value of the lateral resistance coefficient. The turning radius R is expressed as:

$$R = \frac{B}{2} \frac{\omega_2 + \omega_1}{\omega_2 - \omega_1} \quad (5)$$

The rotational speed of the inside and outside sprockets (ω_1 and ω_2, respectively) is calculated as follows:

$$\omega_{2,1} = \frac{30}{\pi} \frac{v_{2,1} \cdot i_0}{r} \quad (6)$$

where v_1 and v_2 are the speed of the two tracks. The rolling resistance forces acting on the two tracks are obtained using the following expression:

$$F_1 = F_2 = \frac{1}{2} f m_v g \tag{7}$$

where f is the rolling resistance coefficient. The power request P_{req} should be balanced by the two motors anytime as follows:

$$P_{req} = T_1 \omega_1 \eta_{em}^{\pm 1} + T_2 \omega_2 \eta_{em}^{\pm 1} \tag{8}$$

where η_{em} is the efficiency of the motor. When the power request is positive, electric power is delivered to propel the vehicle and a positive efficiency sign is returned, and vice versa; however, when the powertrain absorbs the electric power (e.g., regenerative braking [23]), a negative efficiency sign is returned.

2.2. EGS Model

Figure 2 illustrates the equivalent electric circuit of the engine, permanent magnet, directive generator, and rectifier, where ω_g is the rotational speed of the generator, T_g is the electromagnetic torque, K_e is the coefficient of the electromotive force, and $K_x\omega_g$ is the electromotive force; K_x is calculated as follows:

$$K_x = \frac{3}{\pi} K L_g \tag{9}$$

where K is the number of poles and L_g is the synchronous inductance of the armature. The output voltage and current of the generator, U_g and I_g, respectively, are computed as follows [4]:

$$\frac{T_{eng}}{i_{eg}} - T_g = 0.1047 i_{eg} \left(\frac{J_{eng}}{i_{eg}^2} + J_g \right) \frac{dn_{eng}}{dt} \tag{10}$$

$$K_e I_g - K_x I_g^2 = T_g \tag{11}$$

$$U_g = K_e \omega_g - K_x \omega_g I_g \tag{12}$$

$$n_{eng} = 30 \omega_g / \pi i_{eg} \tag{13}$$

where n_{eng} and T_{eng} are the rotational speed and torque of the engine, respectively. Furthermore, J_e and J_g are the moments of inertia; i_{eg} is the fixed gear ratio connecting the engine and generator. The power request is balanced at any time by the EGS and battery as follows:

$$P_{req} = (U_g \cdot I_g + U_b \cdot I_b) \cdot \eta_{em}^{\pm 1} \tag{14}$$

where U_b and I_b are the voltage and current, respectively, of the battery. Figure 3 depicts the results of the EGS test and simulation run to validate the effectiveness of the equivalent electric circuit model, in which U_g and n_{eng} are predicted at an acceptable accuracy during the pulse transient current load.

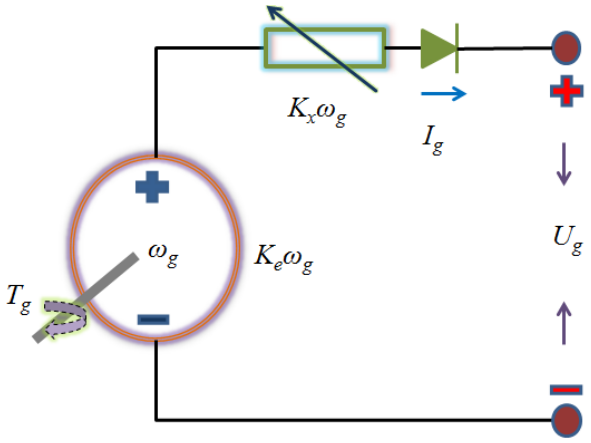


Figure 2. Equivalent circuit of the engine-generator set.

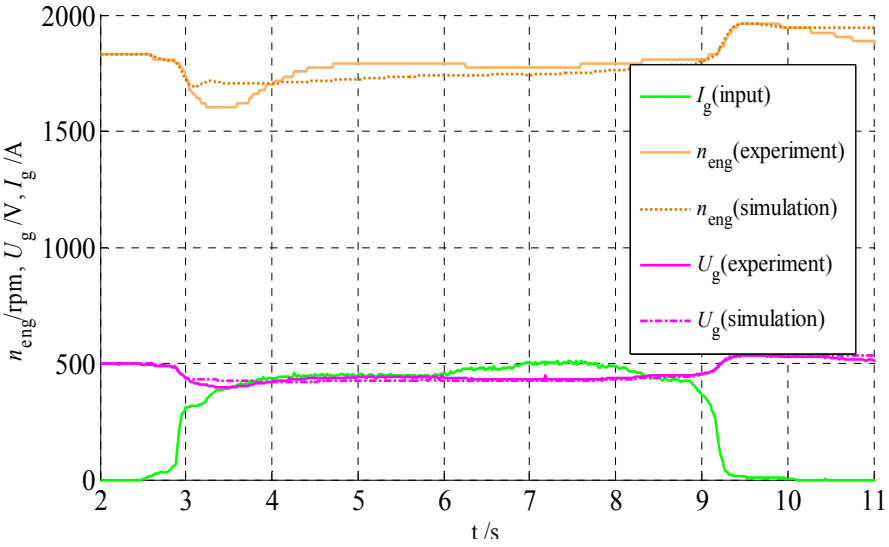


Figure 3. Test and simulation results of the equivalent circuit.

The engine must be limited to the specific work area to ensure safety and reliability:

$$n_{eng,\min} \le n_{eng} \le n_{eng,\max} \tag{15}$$

$$0 \le T_{eng} \le T_{eng,\max} \tag{16}$$

The fuel mass flow rate $\dot{m}_f$ (g/s) was determined according to the engine torque T_{eng} and speed n_{eng} by using a brake specific fuel consumption map, which is typically obtained through a bench test. The control variable, engine throttle signal $u\_th(t)$, was normalized in the range [0,1], and the engine's torque was optimally regulated to control the power split between the EGS and battery to achieve minimum fuel consumption.

2.3. Battery Model

The *SOC* in a battery is a second state variable and is calculated as follows:

$$\frac{d(\mathrm{SOC}(t))}{dt} = -\frac{I_b(t)}{Q_b} \tag{17}$$

$$I_b(t) = \frac{V_{oc} - \sqrt{V_{oc}^2 - 4R_{\mathrm{int}}P_b(t)}}{2R_{\mathrm{int}}} \tag{18}$$

where Q_b is the battery capacity, I_b is the battery current, V_{oc} is the open circuit voltage, R_i is the internal resistance, and P_b is the output power of the battery. To ensure reliability and safety, the current and SOC are constrained as:

$$I_{b,\min} \le I_b(t) \le I_{b,\max} \tag{19}$$

$$\mathrm{SOC}_{\min} \le \mathrm{SOC}(t) \le \mathrm{SOC}_{\max} \tag{20}$$

Figure 4 shows the V_{oc} and R_{int} parameters [4].

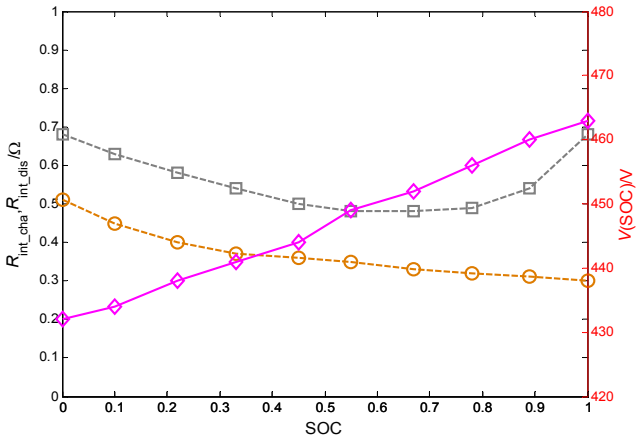


Figure 4. Parameters of Voc and R_{int}.

Cost function minimization is a trade-off between fuel consumption and charge sustainability in the battery and is expressed as follows:

$$J = \int_{t_0}^{t_f} [\dot{m}_f(t) + \beta(\mathrm{SOC}(t_f) - \mathrm{SOC}(t_0))^2]dt \tag{21}$$

where β is a positive weighting factor, which is normally identified through multiple simulation iterations, and $[t_0, t_f]$ is the entire time span.

3. RL-Based Energy Management Strategy

RL is a machine learning approach in which an agent senses an environment through its state and responds to the environment through its action under a control policy. In the proposed model, the control policy is improved iteratively by RL algorithms called *Q*-learning and *Dyna* algorithms. The environment provides numerical feedback called a reward and supplies a transition probability matrix for the agent. According to the driving schedule statistical model, a transition probability matrix is extracted from the

sample information. Subsequently, the RL algorithm is adopted to optimize fuel consumption in another driving schedule by using the transition probability matrix.

3.1. Statistic Information of the Driving Schedule

A long natural driving schedule, including significant accelerations, braking, and steering (Figure 5), was obtained through a field experiment. The power request corresponding to the driving schedule is calculated according to Equations (1)–(8) (Figure 6).

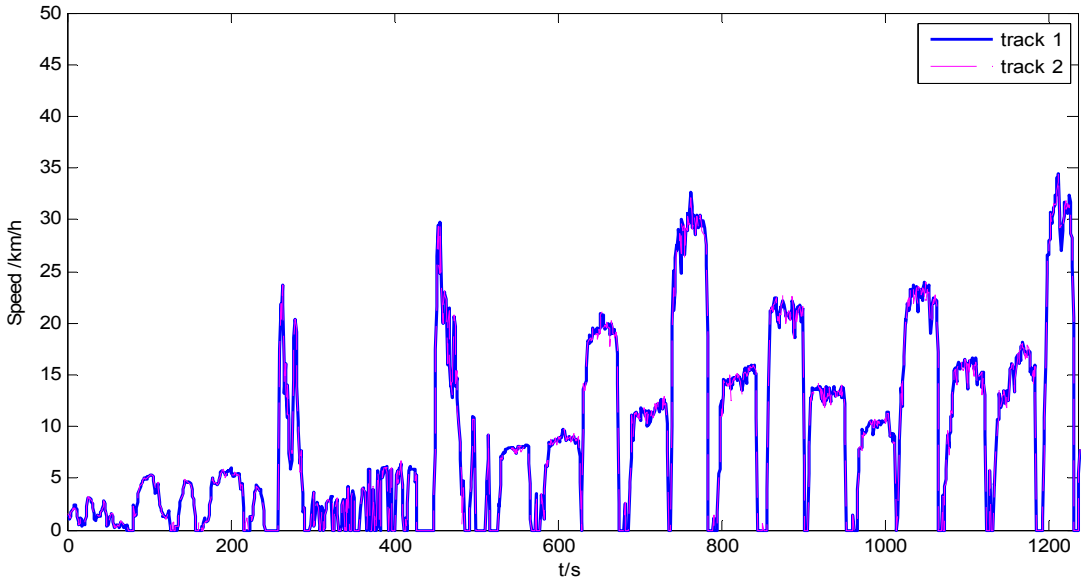


Figure 5. Long driving schedule of the tracked vehicle.

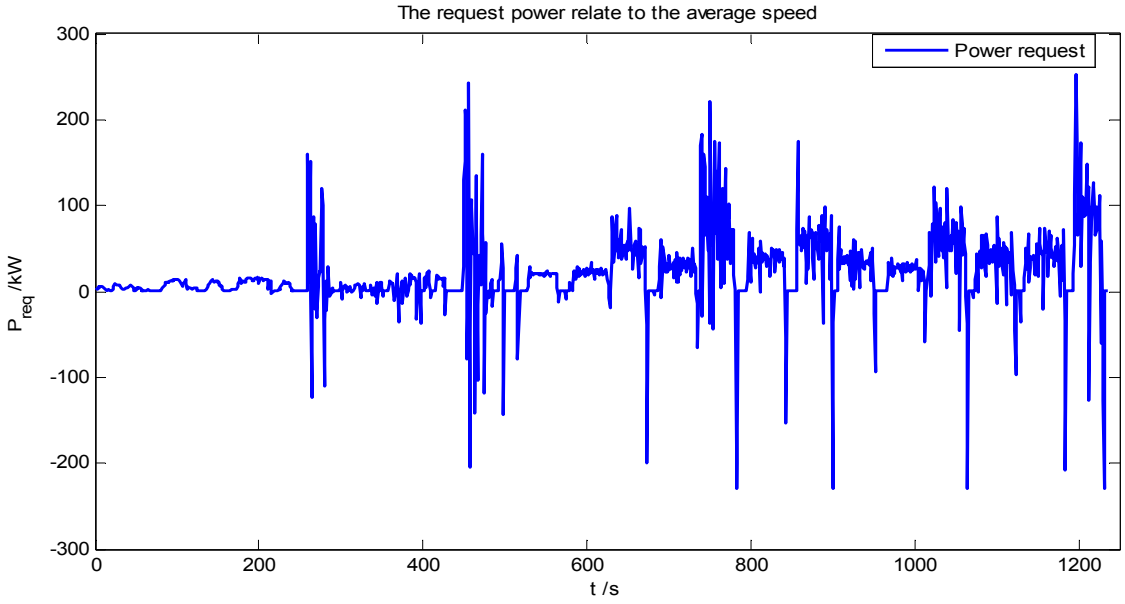


Figure 6. Power request of the long driving schedule.

Maximum likelihood estimation and nearest neighbor method were employed to compute the transition probability of the power request [24]:

$$p_{ik,j} = \frac{N_{ik,j}}{N_{ik}} \qquad N_{ik} \neq 0 \tag{22}$$

where $N_{ik,j}$ is the number of times the transition from P^i_{req} to P^j_{req} has occurred at a vehicle average velocity of $\overline{v}_k$, and N_{ik} is the total event counts of the P^i_{req} occurrence at an average velocity of $\overline{v}_k$. A smoothing technique was applied to the estimated parameters [25]. Figure 7 illustrates the transition probability map at a velocity of 25 km/h.

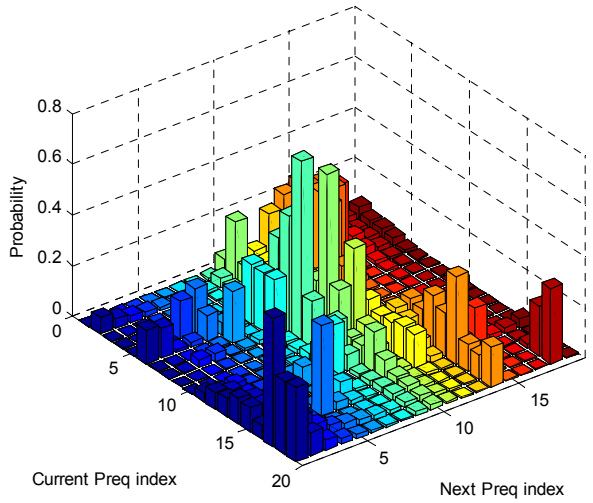


Figure 7. Power request transition probability map at 25 km/h.

In this study, according to the Markov decision processes (MDPs) introduced in [26], the driving schedule was considered a finite MDP. The MDP comprises a set of state variables $S = \{(SOC(t), n_{eng}(t))|\ 0.2 \leq SOC(t) \leq 0.8,\ n_{eng,min} \leq n_{eng}(t) \leq n_{eng,max}\}$, set of actions $a = \{u\_th(t)\}$, reward function $r = \dot{m}_f(s,a)$, and transition function $p_{sa,s'}$, where $p_{sa,s'}$ represents the probability of making a transition from state s to state s' using action a.

3.2. Q-Learning and Dyna Algorithms

When π is used as a complete decision policy, the optimal value of a state s is defined as the expected finite discounted sum of the rewards [27], which is represented as follows:

$$V^*(s) = \min_{\pi} E\left(\sum_{t=t_0}^{t=t_f} \gamma^t r_t\right) \tag{23}$$

where $\gamma \in [0,1]$ is the discount factor. The optimal value function is unique and can be reformulated as follows:

$$V^*(s) = \min_{a}\left(r(s,a) + \gamma \sum_{s' \in S} p_{sa,s'} V^*(s')\right) \quad \forall s \in S \tag{24}$$

Given the optimal value function, the optimal policy is specified as follows:

$$\pi^*(s) = \arg\min_a (r(s,a) + \gamma \sum_{s' \in S} p_{sa,s'} V^*(s')) \tag{25}$$

Subsequently, the Q value and optimal Q value corresponding to the state s and action a are defined recursively as follows:

$$Q(s,a) = r(s,a) + \gamma \sum_{s' \in S} p_{sa,s'} Q(s', a') \tag{26}$$

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s' \in S} p_{sa,s'} \min_{a'} Q^*(s', a') \tag{27}$$

The variable $V^*(s)$ is the value of s assuming an optimal action is taken initially; therefore, $V^*(s) = Q^*(s, a)$ and $\pi^*(s) = arg\ \min_a Q^*(s, a)$. The Q-learning updated rule is expressed as follows:

$$Q(s,a) := Q(s,a) + \alpha(r + \gamma \min_{a'} Q(s', a') - Q(s,a)) \tag{28}$$

where $\alpha \in [0,1]$ is a decayed factor in Q-learning. Unlike the Q-learning algorithm, the *Dyna* algorithm operates by iteratively interacting with the environment. For a tracked vehicle, the *Dyna* algorithm records the sample information as the vehicle operates on a new driving schedule. Then, incremental statistical information is used to update the reward and transition functions. The *Dyna* algorithm updated rule is as follows:

$$Q(s,a) = \bar{r}(s,a) + \gamma \sum_{s' \in S} \bar{p}_{sa,s'} Q(s', a') \tag{29}$$

$$Q(s,a) := Q(s,a) + \alpha(r + \gamma \min_{a'} Q(s', a') - Q(s,a)) \tag{30}$$

where $\bar{r}$ and $\bar{p}_{sa,s'}$ are time variant and change as the driving schedule is updated. The *Dyna* algorithm clearly entails a heavier computational burden compared with the Q-learning algorithm. Section 4 compares the optimality between the two algorithms. Figure 8 depicts the computational flowchart of the two algorithms.

4. Results and Discussion

4.1. Comparison between the Q-Learning and Dyna Algorithm

Figure 9 shows the experimental driving schedule used in the simulation. Figure 10 illustrates the mean discrepancy of the two algorithms at v = 25 km/h, where the mean discrepancy is the deviation of two Q values per 100 iterations. The mean discrepancy declined with iterative computations, indicating the convergence of the Q-learning and *Dyna* algorithms. Figure 10 also shows that the rate of convergence of the *Dyna* algorithm is faster than that of the Q-learning algorithm. A possible conclusion is that the time-variant reward function and the transition function in the *Dyna* algorithm accelerates the convergence [28].

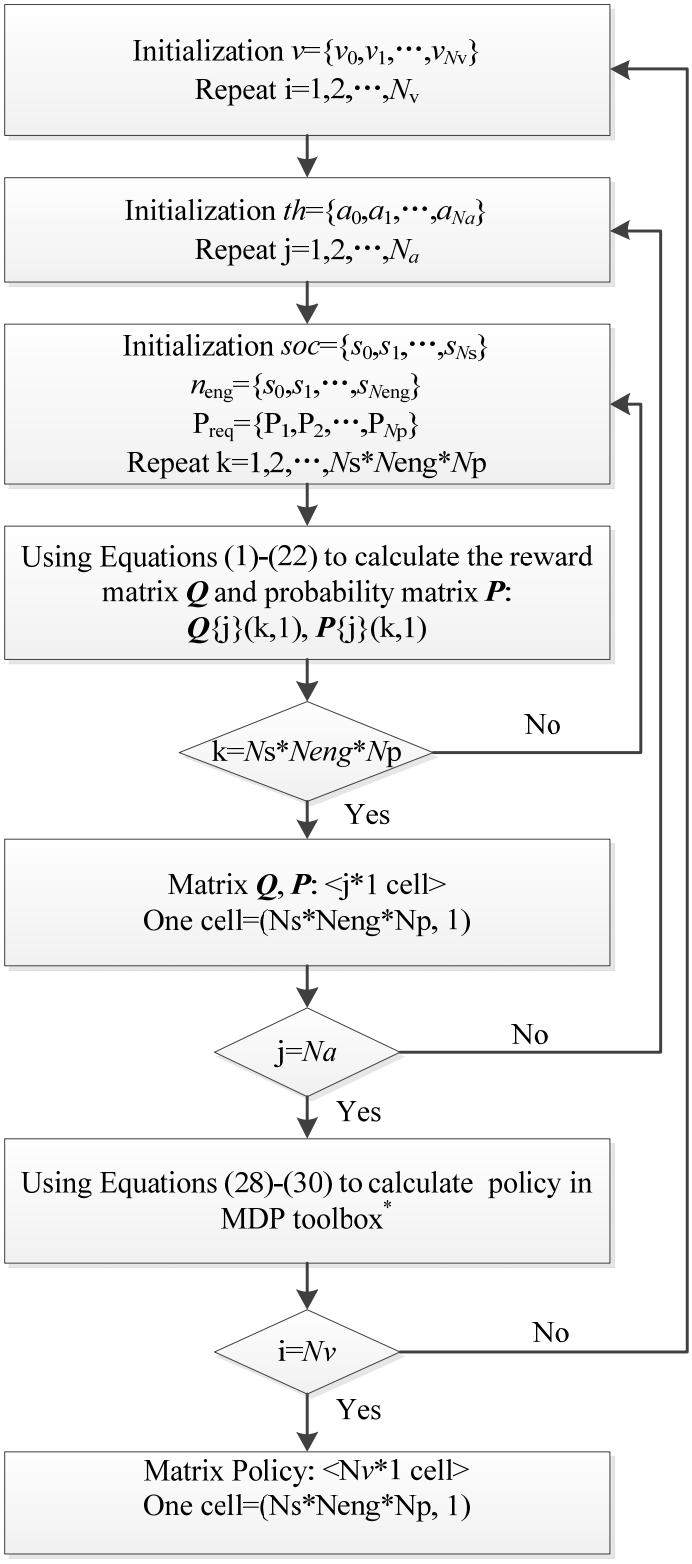


Figure 8. Computational flowchart of the *Q*-learning and *Dyna* algorithms. * The MDP toolbox is introduced in [26].

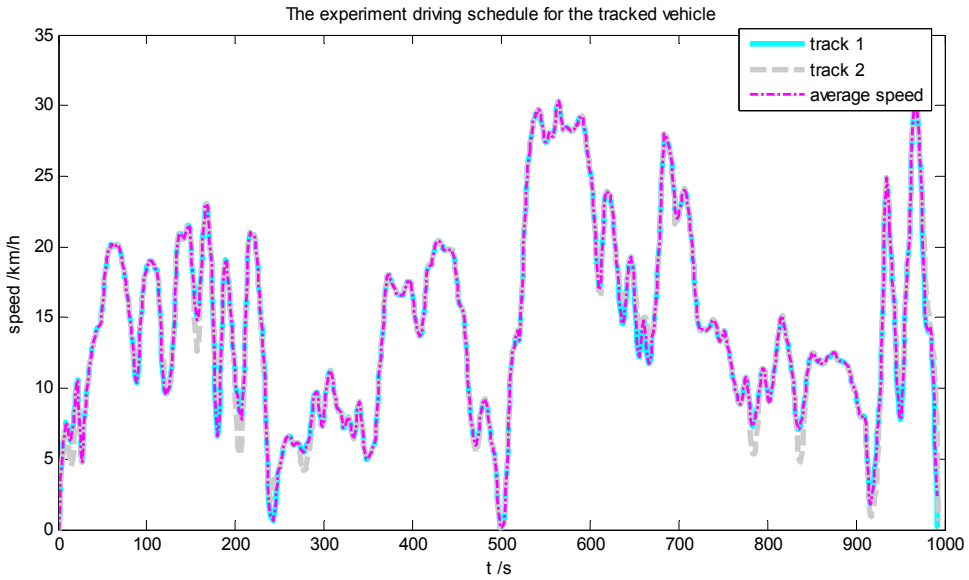


Figure 9. Experimental driving schedule used in the simulation.

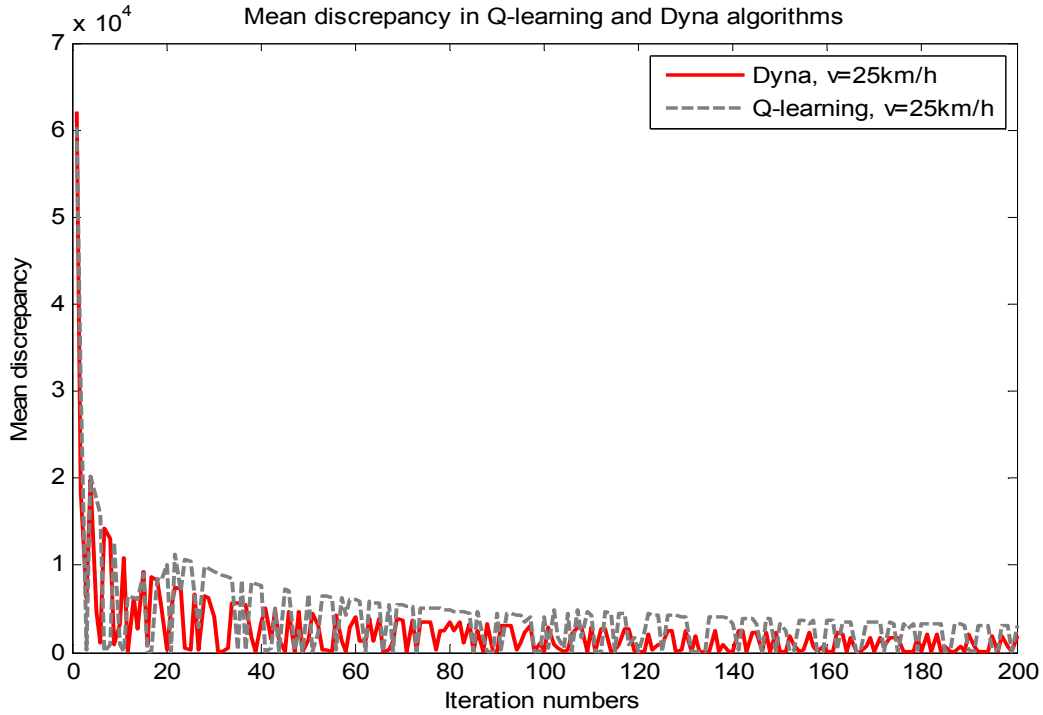
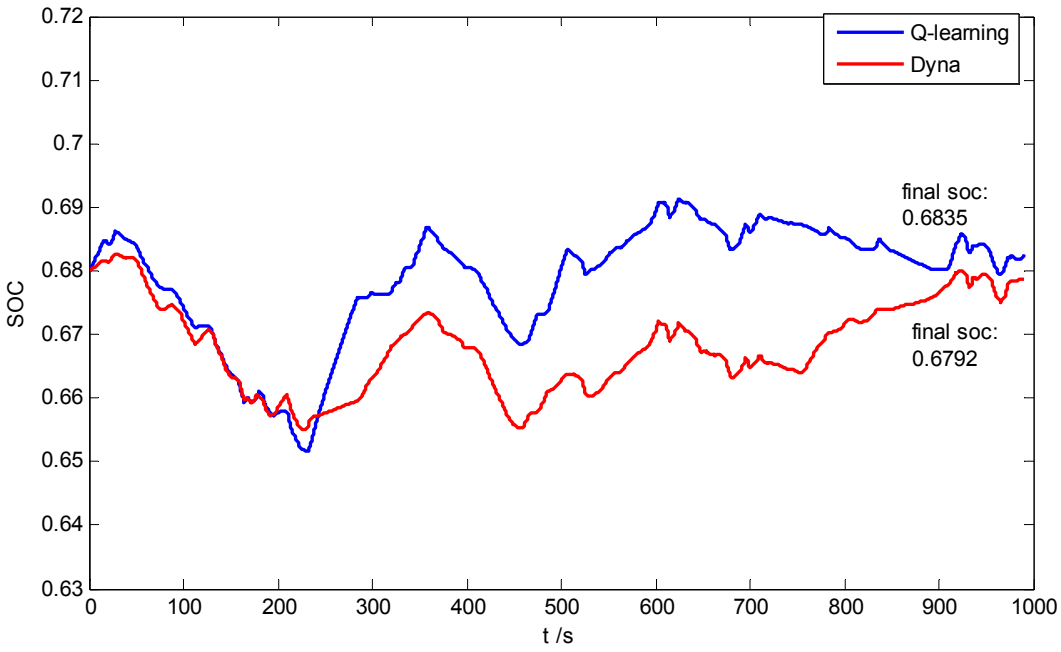
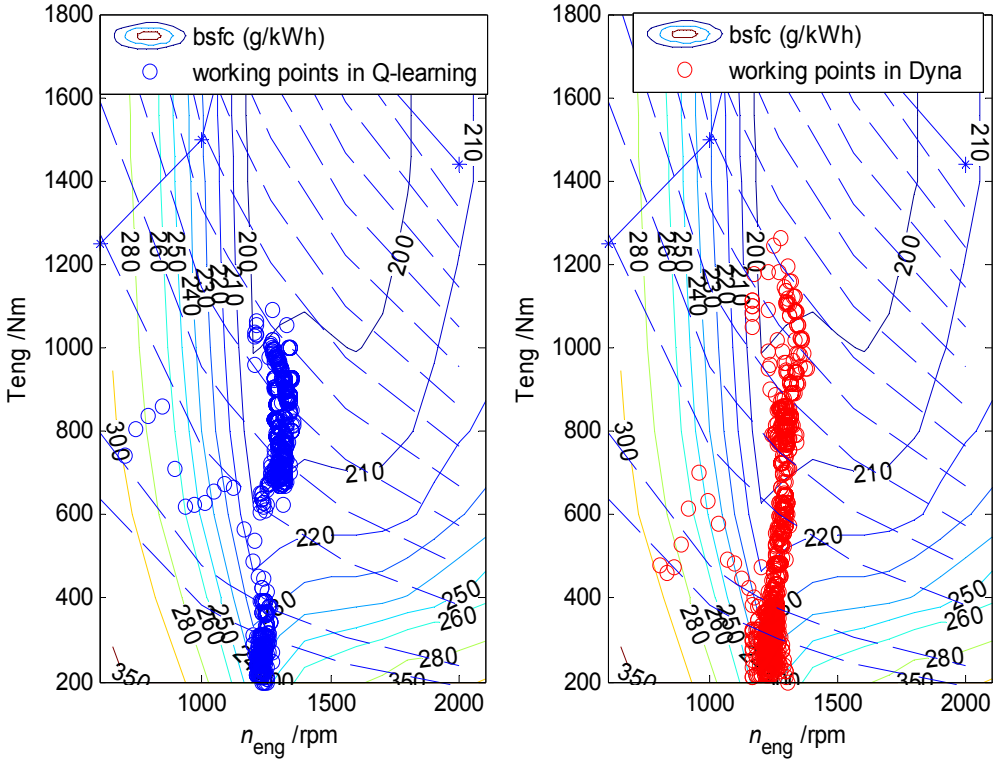


Figure 10. Mean discrepancy of the value function in the *Q*-learning and *Dyna* algorithms.

Figure 11 depicts the simulation results of the *Q*-learning and *Dyna* algorithms. Because of the charge sustenance in the cost function, the final SOC values were close to the initial SOC value. Figure 11b shows the fuel consumption and working points of the engine. An SOC-correction method [29] was applied to compensate for the fuel consumption caused by the various SOC final values. Figure 12 illustrates the performance of the two algorithms. Table 2 lists the fuel consumption; the fuel consumption of the *Dyna* algorithm is lower than that of the *Q*-learning algorithm, which is attributable to the difference in the time-variant reward function and the transition function between the *Dyna* and *Q*-learning algorithms.



(**a**) SOC trajectories in two algorithms.



(**b**) Engine operation area in the two algorithms.

Figure 11. SOC trajectories and engine operation area in the *Q*-learning and *Dyna* algorithms.

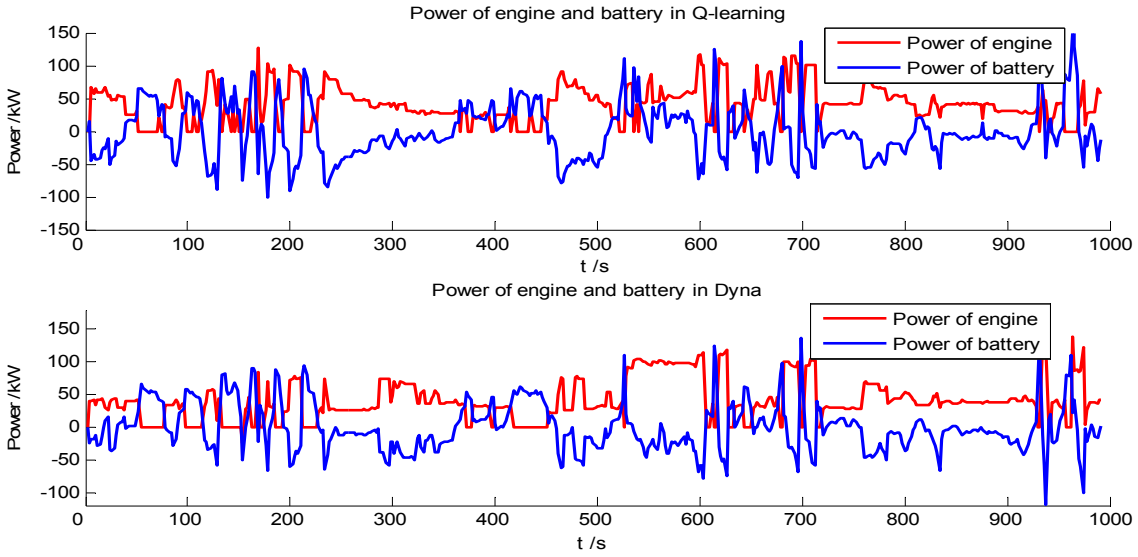


Figure 12. Battery and engine power in the *Q*-learning and *Dyna* algorithms.

Table 2. Fuel consumption in the *Q*-learning and *Dyna* algorithms.

Algorithm	Fuel Consumption (g)	Relative Increase (%)
Dyna	2847	–
Q-learning	2896	1.72

Table 3 shows the computation times of the two algorithms; the *Dyna* algorithm has a longer computation time compared with the *Q*-learning algorithm. This is caused by the updated rule of the *Dyna* algorithm, in which the reward function and the transition probability are updated at a certain step size [28]. Thus, the updated transition probability and reward function of the *Dyna* algorithm resulted in lower fuel consumption but longer computation time.

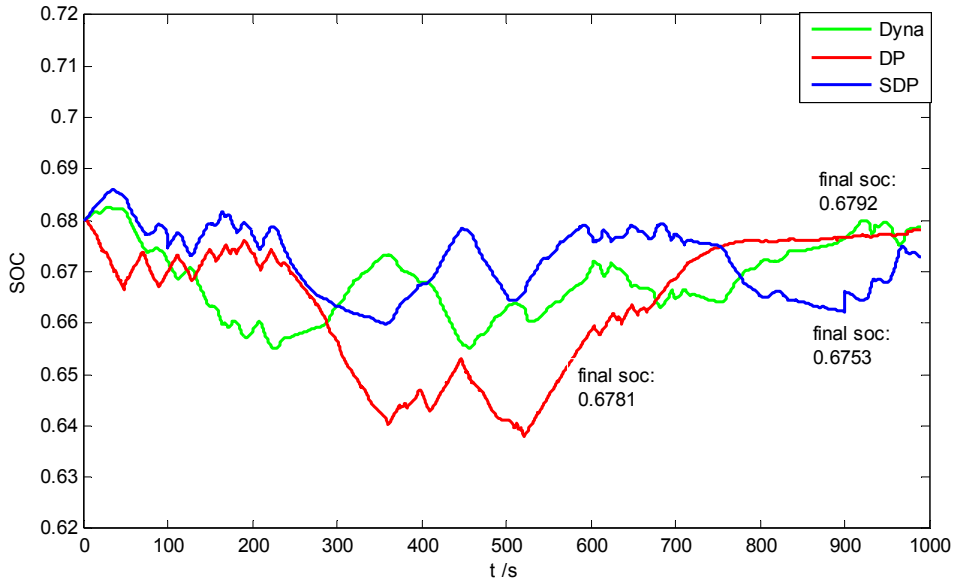
Table 3. Computation times of the *Q*-learning and *Dyna* algorithms.

Algorithms	*Q*-learning	*Dyna*
Time [a] (h)	3	7

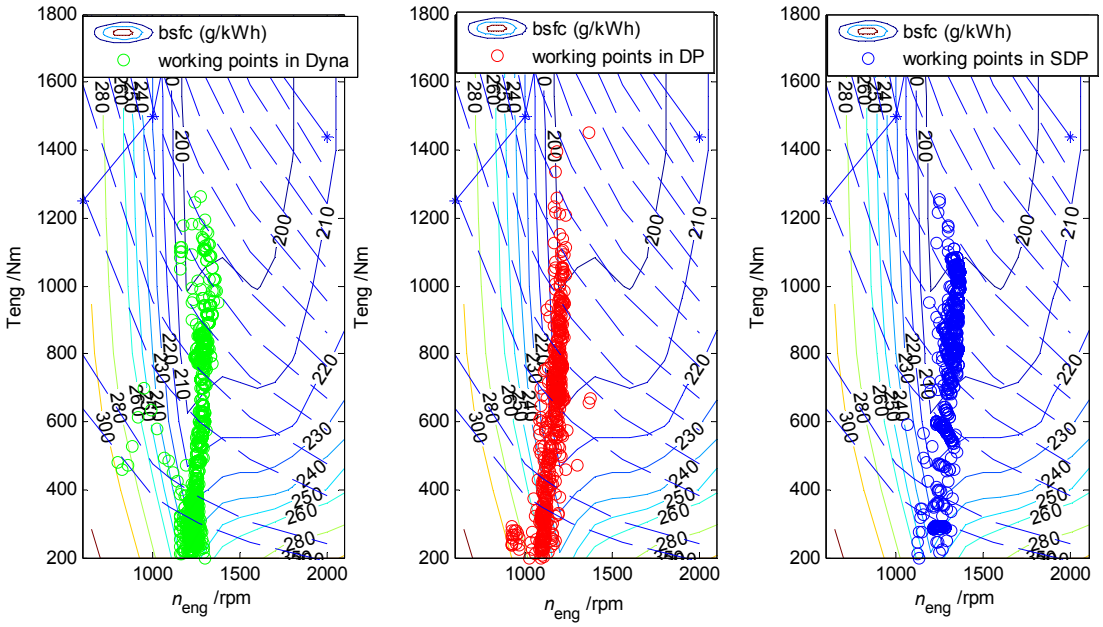
[a] A 2.4 GHz microprocessor with 12 GB RAM was used.

4.2. Comparative Analysis of the Results of Dyna Algorithm, SDP, and DP

To validate the optimality of the RL technique, the *Dyna* algorithm, SDP [24], and DP [30] were controlled on the experimental driving schedule shown in Figure 10; Figure 13 presents the simulation results. The SOC terminal values were close to the initial values because of the final constraint in the cost function. Figure 13b illustrates the engine work area, indicating that the engine frequently works in a low fuel consumption field to ensure optimal fuel economy. Table 4 lists the fuel consumption after SOC correction. The *Dyna*-based fuel consumption was lower than the SDP-based fuel consumption and extremely close to the DP-based fuel consumption. Table 5 shows the computation time of the three algorithms. Because of the policy iteration process in SDP, the SDP-based computation time was considerably longer than the *Dyna*- and DP-based computation times.



(**a**) SOC trajectories in the *Dyna* algorithm, SDP, and DP.



(**b**) Engine operation area in three algorithms.

Figure 13. SOC trajectories and engine operation area in the *Dyna* algorithm, SDP, and DP.

Table 4. Fuel consumption in the *Dyna* algorithm, SDP, and DP.

Algorithm	Fuel Consumption (g)	Relative Increase (%)
DP	2847	—
Dyna	2853	0.21
SDP	2925	2.74

Table 5. Computation times of the *Dyna* algorithm, SDP, and DP.

Algorithms	***DP***	***Dyna***	***SDP***
Time [a] (h)	2	7	12

[a] A 2.4 GHz microprocessor with 12 GB RAM was used.

Because the *Dyna*-based control policy is extremely close to the DP-based optimal control policy, the *Dyna* algorithm has the potential to realize a real-time control strategy in the future. When the present power request is considered a continuous system, the next power request of a vehicle can be predicted accurately using the method introduced in [31,32]. Subsequently, when the power request is combined with the *Dyna* algorithm, the reward function and transition probability matrix can be updated. Furthermore, the computation time can be reduced when the transition probability matrix is updated as the reference [31]. Finally, the power split at the next time can be determined and a real-time control can be implemented.

5. Conclusions

In this study, the RL method was employed to derive an optimal energy management policy for an HETV. The updated rules of the *Q*-learning and *Dyna* algorithms were elucidated. The two algorithms were applied to the same experimental driving schedule to compare their optimality and computation times. The simulation results indicated that the *Dyna* algorithm registers more efficient fuel economy than the *Q*-learning algorithm does. However, the computation time of the *Dyna* algorithm is considerably longer than that of the *Q*-learning algorithm. The global optimality of the *Dyna* algorithm was validated by comparing it with the DP and SDP methods. The results showed that the *Dyna*-based control policy is more effective than the SDP-based control policy and close to the DP-based optimal control policy. In future studies, the *Dyna* algorithm will be used to realize a real-time control by predicting the next power request in a stationary Markov chain–based transition probability model.

Acknowledgments

The authors appreciate the scrupulous reviewers for their valuable comments and suggestions. This research was supported by the National Nature Science Foundation, China (Grant 51375044), National Defense Basic Research, China (Grant B2220132010) and University Talent Introduction Program of China (Grant B12022).

Author Contributions

Teng Liu, is writing and revising this manuscript integrally. Yuan Zou is in charge of deciding how to modify the reinforcement learning algorithm in this manuscript and contacting the editor. The task of Dexing Liu is computing and recomputing the transition probability matrix of different driving schedules in this manuscript. Fengchun Sun is responsible for revising all figures according to reviewers' suggestions and the English editing in this manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Serrao, L.; Onori, S.; Rizzoni, G. A Comparative Analysis of Energy Management Strategies for Hybrid Electric Vehicles. *J. Dyn. Syst. Meas. Control* **2011**, *133*, 031012:1–031012:9.

2. Lin, C.C.; Kang, J.M.; Grizzle, J.W.; Peng, H. Energy Management Strategy for a Parallel Hybrid Electric Truck. In Proceedings of the American Control Conference 2001, Arlington, VA, USA, 25–27 June 2001; Volume 4, pp. 2878–2883.
3. Zou, Y.; Liu, T.; Sun, F.C.; Peng, H. Comparative study of dynamic programming and pontryagin's minimum principle on energy management for a parallel hybrid electric vehicle. *Energies* **2013**, *6*, 2305–2318.
4. Zou, Y.; Sun, F.C.; Hu, X.S.; Guzzella, L.; Peng, H. Combined optimal sizing and control for a hybrid tracked vehicle. *Energies* **2012**, *5*, 4697–4710.
5. Sundstrom, O.; Ambuhl, D.; Guzzella, L. On implementation of dynamic programming for optimal control problems with final state constraints. *Oil Gas Sci. Technol*. **2009**, *65*, 91–102.
6. Johannesson, L.; Åsbogård, M.; Egardt, B. Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming. *IEEE Trans. Intell. Transp. Syst.* **2007**, *8*, 71–83.
7. Tate, E.; Grizzle, J.; Peng, H. Shortest path stochastic control for hybrid electric vehicles. *Int. J. Robust Nonlinear Control* **2008**, *18*, 1409–1429.
8. Kim, N.; Cha, S.; Peng, H. Optimal control of hybrid electric vehicles based on Pontryagin's minimum principle. *IEEE Trans. Control Syst. Technol.* **2011**, *19*, 1279–1287.
9. Delprat, S.; Lauber, J.; Marie, T.; Rimaux, J. Control of a paralleled hybrid powertrain: Optimal control. *IEEE Trans. Veh. Technol*. **2004**, *53*, 872–881.
10. Nüesch, T.; Cerofolini, A.; Mancini, G.; Guzzella, L. Equivalent consumption minimization strategy for the control of real driving NO*x* emissions of a diesel hybrid electric vehicle. *Energies* **2014**, *7*, 3148–3178.
11. Musardo, C.; Rizzoni, G.; Guezennec, Y.; Staccia, B. A-ECMS: An adaptive algorithm for hybrid electric vehicle energy management. *Eur. J. Control* **2005**, *11*, 509–524.
12. Sciarretta, A.; Back, M.; Guzzella, L. Optimal control of paralleled hybrid electric vehicles. *IEEE Trans. Control Syst. Technol.* **2004**, *12*, 352–363.
13. Vu, T.V.; Chen, C.K.; Hung, C.W. A model predictive control approach for fuel economy improvement of a series hydraulic hybrid vehicle. *Energies* **2014**, *7*, 7017–7040.
14. Nüesch, T.; Elbert, P.; Guzzella, L. Convex optimization for the energy management of hybrid electric vehicles considering engine start and gearshift costs. *Energies* **2014**, *7*, 834–856.
15. Gao, B.T.; Zhang, W.H.; Tang, Y.; Hu, M.J.; Zhu, M.C.; Zhan, H.Y. Game-theoretic energy management for residential users with dischargeable plug-in electric vehicles. *Energies* **2014**, *7*, 7499–7518.
16. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; The MIT Press: Cambridge, MA, USA; London, UK, 2005; pp. 140–300.
17. Hester, T.; Quinlan, M.; Stone, P. RTMBA: A real-time model-based reinforcement learning architecture for robot control. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 85–90.
18. Degris, T.; Pilarski, P.M.; Sutton, R.S. Model-free reinforcement learning with continuous action in practice. In Proceedings of the 2012 American Control Conference, Montreal, QC, Canada, 27–29 June 2012; pp. 2177–2182.

19. Perron, J.; Moulin, B.; Berger, J. A hybrid approach based on multi-agent geo simulation and reinforcement learning to solve a UAV patrolling problem. In Proceedings of the Winter Simulation Conference, Austin, TX, USA, 7–10 December 2008; pp. 1259–1267.
20. Hsu, R.C.; Liu, C.T.; Chan, D.Y. A reinforcement-learning-based assisted power management with QoR provisioning for human–electric hybrid bicycle. *IEEE Trans. Ind. Electron.* **2012**, *59*, 3350–3359.
21. Abdelsalam, A.A.; Cui, S.M. A fuzzy logic global power management strategy for hybrid electric vehicles based on a permanent magnet electric variable transmission. *Energies* **2012**, *5*, 1175–1198.
22. Langari, R.; Won, J.S. Intelligent energy management agent for a parallel hybrid vehicle—Part I: System architecture and design of the driving situation identification process. *IEEE Trans. Veh. Technol.* **2005**, *54*, 925–934.
23. Guo, J.G.; Jian, X.P.; Lin, G.Y. Performance evaluation of an anti-lock braking system for electric vehicles with a fuzzy sliding mode controller. *Energies* **2014**, *5*, 6459–6476.
24. Lin, C.C.; Peng, H.; Grizzle, J.W. A stochastic control strategy for hybrid electric vehicles. In Proceedings of the American Control Conference, Boston, MA, USA, 30 June–2 July 2004; pp. 4710–4715.
25. Dai, J. Isolated word recognition using Markov chain models. *IEEE Trans. Speech Audio Proc.* **1995**, *3*, 458–463.
26. Brazdil, T.; Chatterjee, K.; Chmelik, M.; Forejt, V.; Kretinsky, J.; Kwiatkowsha, M.; Parker, D.; Ujma, M. Verification of markov decision processes using learning algorithms. *Logic Comput. Sci.* **2014**, *2*, 4–18.
27. Chades, I.; Chapron, G.; Cros, M.J. Markov Decision Processes Toolbox, Version 4.0.2. Available online: http://cran.r-project.org/web/packages/MDPtoolbox/ (accessed on 22 July 2014).
28. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement learning: A survey. *Artif. Intell. Res.* **1996**, *4*, 237–285.
29. Zou, Z.Y.; Xu, J.; Mi, C.; Cao, B.G. Evaluation of model based state of charge estimation methods for lithium-Ion batteries. *Energies* **2014**, *7*, 5065–5082.
30. Jimenez, F.; Cabrera-Montiel, W. System for Road Vehicle Energy Optimization Using Real Time Road and Traffic Information. *Energies* **2014**, *7*, 3576–3598.
31. Filev, D.P.; Kolmanovsky, I. Generalized markov models for real-time modeling of continuous systems. *IEEE Trans. Fuzzy Syst.* **2014**, *22*, 983–998.
32. Di Cairano, S.; Bernardini, D.; Bernardini, D.; Bemporad, A.; Kolmanovsky, I.V. Stochastic MPC with Learning for Driver-Predictive Vehicle Control and its Application to HEV Energy Management. *IEEE Trans. Cont. Syst. Technol.* **2015**, *22*, 1018–1030.