# Online Optimization of Gear Shift and Velocity for Eco-Driving using Adaptive Dynamic Programming

Li, Guoqiang; Gorges, Daniel; Wang, Meng

**DOI**
10.1109/TIV.2021.3111037

**Publication date**
2022

**Document Version**
Final published version

**Published in**
IEEE Transactions on Intelligent Vehicles

**Citation (APA)**
Li, G., Gorges, D., & Wang, M. (2022). Online Optimization of Gear Shift and Velocity for Eco-Driving using Adaptive Dynamic Programming. *IEEE Transactions on Intelligent Vehicles*, *7*(1), 123-132. https://doi.org/10.1109/TIV.2021.3111037

# Online Optimization of Gear Shift and Velocity for Eco-Driving using Adaptive Dynamic Programming

Li, Guoqiang; Gorges, Daniel; Wang, Meng

**DOI**
10.1109/TIV.2021.3111037

**Publication date**
2022

**Document Version**
Accepted author manuscript

**Published in**
IEEE Transactions on Intelligent Vehicles

# Online Optimization of Gear Shift and Velocity for Eco-Driving using Adaptive Dynamic Programming

Guoqiang Li, Daniel Görges, *Member*, *IEEE* and Meng Wang, *Member*, *IEEE*

***Abstract*—In this paper a learning-based optimization method for online gear shift and velocity control is presented to reduce the fuel consumption and improve the driving comfort in a car-following process. The continuous traction force and the discrete gear shift are optimized jointly to improve both the powertrain operation and the longitudinal motion. The problem is formulated as a nonlinear mixed-integer optimization problem and solved based on adaptive dynamic programming. A major difference compared to existing approaches is that the developed control method is model-free, i.e. it does not rely on vehicle models. It can address system nonlinearities and adapt to changes in engine characteristics (e.g. consumption map) during vehicle driving. The computation is efficient and enables possible real-time implementation. The proposed control method is studied for an urban driving cycle to evaluate the control performance with respect to the fuel economy and the driving comfort. Simulations indicate that the host vehicle can reduce the fuel consumption by 5.03% and 1.12% for two consumption maps in comparison to the preceding while keeping a desired inter-vehicle distance. The results further show a decrease of 1.59% and 2.32% in fuel consumption compared to a linear quadratic controller with the same gear shift schedule.**

***Index Terms*—Eco-driving, gear shift schedule, velocity optimization, adaptive cruise control, adaptive dynamic programming, reinforcement learning**

## I. Introduction

Recently the reduction of the fuel consumption and pollutant emissions has attracted a lot of attention throughout the world. Advanced driver assistance systems for a fuel-economic driving have received considerable interest in both academia and industry [1]. Studies in [2], [3] have shown that the on-board eco-driving device for drivers which provides instantaneous fuel economy feedback can reduce the fuel consumption by 6% and 1% on city streets and highways respectively.

During eco-driving the vehicle velocity is controlled either by a human driver or a cruise controller, which is the focus of this paper to minimize the fuel consumption and ensure the driving safety. Generally eco-driving can be classified into two categories: the free-flow driving and the car-following driving. The free-flow driving only takes a host vehicle as objective without consideration of surrounding vehicles. The pulse-and-glide method has been proposed to improve the fuel economy in [4], [5]. It makes use of the vehicle kinematic energy and operates the engine in a highly efficient region. The road and the traffic condition can influence the driving behavior and the fuel consumption. With the preview of the road grade and the trip distance information, the velocity profile is optimized to reduce the fuel consumption in [6]–[8]. A probabilistic traffic-signal phase and timing information as well as the mixed traffic on arterial roads is studied for the energy-efficient velocity planning in [9], [10]. The optimal velocity trajectory which maximizes the chance of going through green lights is designed to save the fuel by reducing idling at red lights. In [11] the velocity is optimized by a power-based optimal longitudinal controller considering the traffic condition and the traffic signal status. Furthermore, in order to reduce the online computational burden, a cloud-based method using a spatial-domain dynamic programming (DP) is developed to improve the fuel economy in [12]. The real-time application is achieved through a two-way communication system between the vehicle and the cloud. A Pontryagin's Maximum Principle based solution to determine the optimal velocity profile for energy saving by incorporating the gear shifting, speed limit and road grade constraints is proposed in [13]. However, the driving safety in terms of the required inter-vehicle space particularly in congested urban areas is not addressed in the aforementioned papers.

In the car-following driving, a host vehicle follows a preceding vehicle within a safe distance. The velocity of the host vehicle is selected with adaptive cruise control to improve the fuel economy. Model predictive control (MPC) methods have been applied for the ecological driving in [14]–[16]. A robust eco-adaptive cruise controller is developed in [17] to control the vehicle acceleration and the gear shift schedule. The gear shift control policy is, however, designed offline without an online optimization for the fuel optimality. An energy-optimal adaptive cruise controller combining MPC and DP is introduced in [18]. DP provides the energy-optimal speed trajectory using information like speed limits, road slope, travel time, etc., and MPC is used to calculate the traction force of the vehicle to follow the speed trajectory. In the methods described above a fixed system model is used. Changes in the system model are not considered. Reinforcement learning has been used to design discrete control inputs for adaptive cruise control in [19], [20]. The discretized control inputs may influence the performance for the real driving environment which contains continuous states.

The joint control of the powertrain operation and the longitudinal dynamic motion has a promising potential to reduce the fuel consumption. The gear ratio in the step-gear transmission can adjust the engine working points for fuel economy. A fuel-

The work of Guoqiang Li was supported by the China Scholarship Council.

Guoqiang Li is with the School of Mechanical Engineering, Beijing Institute of Technology, 10081 Bijing, China (guoqiangli19@outlook.com). Daniel Görges is with the Institute of Electromobility, Dept. of Electrical and Computer Engineering, University of Kaiserslautern, 67663 Kaiserslautern, Germany (goerges@eit.uni-kl.de). Meng Wang is with the Faculty of Civil Engineering and Geosciences, Delft University of Technology, Stevinweg 1, 2600 GA Delft, Netherlands (m.wang@tudelft.nl).

efficient velocity profile can be realized through optimizing the engine torque, the brake force, and the gear shift while guaranteeing a safe inter-vehicle space. This problem usually relies on a mixed-integer nonlinear optimization with a considerable computation burden. It is challenging for an online calculation. Pontryagin's Minimum Principle (PMP) has been applied in an integrated optimization of the velocity and the powertrain for a real-time eco-driving system in [21], [22]. However these controllers are developed using a fixed system model and a velocity preview. They can therefore not adapt to parameter changes and varying driving behaviors in the real urban traffic.

Throughout the literature, an online optimization method for the gear shift schedule and the velocity which can both improve the fuel economy and the driving comfort has not been well studied. The joint optimization of the continuous traction force and the discrete gear shift control is highly desirable. Furthermore the adaptivity with respect to the changes in engine characteristics during the vehicle operation is still open.

The objective of this paper is to address the issues indicated above and realize the joint optimization of the powertrain operation and the longitudinal dynamic motion for the fuel economy and the driving comfort. A learning-based control algorithm using adaptive dynamic programming (ADP) is applied to design the gear shift schedule and the velocity trajectory. The gear shift control is solved based on enumeration to adjust the engine working points and improve the engine fuel efficiency. The velocity is optimized through controlling the traction force with ADP to guarantee a desired driving space from preceding vehicles. Preliminary results to the proposed controller were presented in [23]. In this paper the results are extended in the following directions. First, a more in-depth investigation of the developed method is provided. Second, the optimal control performance particularly regarding the fuel economy is further compared with a linear quadratic controller (LQR). Third, the real-time capability is discussed. Finally, the adaptivity with respect to the changes in the engine fuel consumption map is studied.

In summary, the major contribution of this paper lies in providing a real-time optimal control method for eco-driving in a car-following process. It differs from the current adaptive cruise controllers in the following aspects: i) the joint optimization of the powertrain operation and the longitudinal dynamic motion is realized by controlling the gear shift and the traction force for a fuel-efficient and comfortable driving; ii) it enables an online learning of the control policy based on interaction with the environment; iii) it does not require a system model and can adapt to the changes in engine characteristics.

This paper is organized as follows. The problem formulation for eco-driving and the control objective are introduced in Section II. The ADP principle and the controller design are presented in Section III. In Section IV the proposed controller is investigated for an urban driving scenario to evaluate the control performance and adaptivity. Conclusions are finally given in Section V.

## II. Problem Formulation

The eco-driving studied in this paper is illustrated in Fig. 1. It is formulated into a car-following framework with a host vehicle and a preceding vehicle. The preceding vehicle drives with velocity $v_{\mathrm{p}}$ based on the traffic condition and is followed by the host vehicle. The velocity of the host vehicle $v_{\mathrm{h}}$ is controlled to keep a desired inter-vehicle distance $L_{\mathrm{des}}$ from the preceding vehicle for safe driving, and at the same time to minimize the fuel consumption. The actual inter-vehicle distance denoted by $L$ can be measured by a radar sensor without vehicular communication or velocity prediction. Furthermore, the host vehicle is equipped with a multi-step gear transmission to adjust the engine operation and output power.

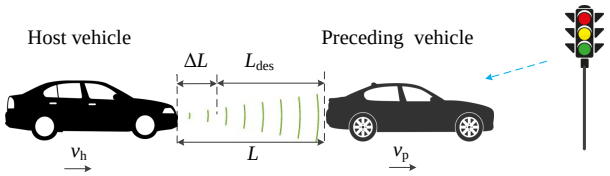



Fig. 1. Car-following scenario

Figure 2 presents an overall framework of the proposed ADP method for ecological driving. The ADP-based controller receives the velocities and distance, i.e. $v_{\mathrm{p}}$, $v_{\mathrm{h}}$ and $L$. The gear position and the traction force are calculated by ADP and applied to the host vehicle. The fuel rate at each step is used to update the control policy in ADP to improve the fuel economy.

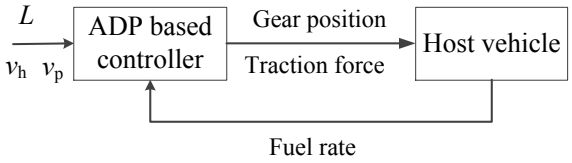



Fig. 2. Overall framework of the ADP-based controller for ACC

### A. Vehicle dynamics model

The system dynamics model of the car-following framework is expressed as

$$\begin{aligned} \dot{\Delta L} &= v_{\mathrm{p}} - v_{\mathrm{h}} - \tau_{\mathrm{h}} a_{\mathrm{h}} \\ \dot{\Delta v} &= a_{\mathrm{p}} - a_{\mathrm{h}} \end{aligned} \tag{1}$$

where $\Delta L$ is the inter-vehicle distance deviation defined by $\Delta L = L - L_{\mathrm{des}}$, $\Delta v$ is the relative velocity deviation with $\Delta v = v_{\mathrm{p}} - v_{\mathrm{h}}$, and $a_{\mathrm{h}}$ and $a_{\mathrm{p}}$ are the acceleration of the host and preceding vehicle, respectively. $\tau_{\mathrm{h}}$ is the nominal time headway which is used to calculate the desired inter-vehicle distance $L_{\mathrm{des}}$ for the driving safety following a constant time headway policy, i.e.

$$L_{\mathrm{des}} = \tau_{\mathrm{h}} v_{\mathrm{h}} + d_0 \tag{2}$$

where $d_0$ is the standstill distance. According to (2) the desired inter-vehicle distance $L_{\mathrm{des}}$ increases linearly with the host

vehicle velocity $v_\mathrm{h}$ to take the influence of the velocity on the braking distance into account where the time headway $\tau_\mathrm{h}$ as linearity factor is selected based on legal requirements and driver preferences.

The longitudinal dynamics model of the host vehicle is formulated as

$$\begin{aligned} \dot{x}_\mathrm{h} &= v_\mathrm{h} \\ \dot{v}_\mathrm{h} &= a_\mathrm{h} \\ m a_\mathrm{h} &= F_\mathrm{t} - \frac{\rho A c_\mathrm{d} v_\mathrm{h}^2}{2} - mgf\cos\alpha - mg\sin\alpha \end{aligned} \tag{3}$$

where $F_\mathrm{t}$ is the traction force at the wheels, $\rho$ is the air density, $A$ is the equivalent area of the vehicle body, $c_\mathrm{d}$ is the aerodynamic resistance coefficient, $m$ is the vehicle mass, $f$ is the rolling resistance coefficient, and $\alpha$ is the road angle. The resistance forces consist of the aerodynamic force, rolling force and grading force.

The internal combustion engine in the host vehicle is described by a static fuel consumption map as shown in Fig. 3 from [24]. The fuel rate $\dot{m}_\mathrm{f}$ is a nonlinear function of the engine torque $T_\mathrm{e}$ and the engine speed $\omega_\mathrm{e}$, i.e.

$$\dot{m}_\mathrm{f} = f(T_\mathrm{e}, \omega_\mathrm{e}). \tag{4}$$

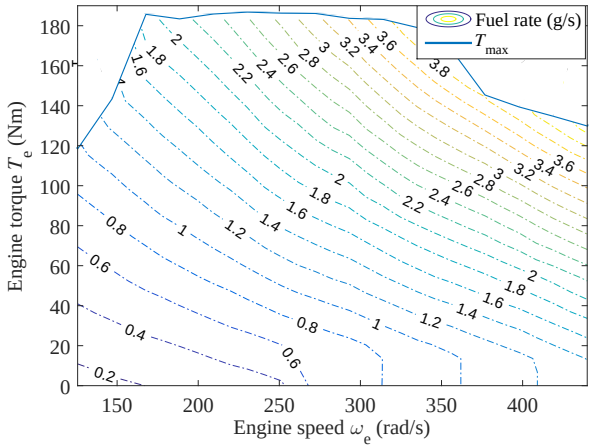



Fig. 3. Fuel consumption map

It can be observed that under the same output torque the engine working in the low speed region can lead to less fuel consumption than the one in the high speed region. Therefore a gear shift schedule optimization is required to adjust the engine working points for improving the fuel economy.

The host vehicle is equipped with a multi-step gearbox. It transfers the engine power to the wheels and provides the required power for driving. The simplified structure is presented in Fig. 4. The relationship of the speed and torque between the engine and the wheels is determined by

$$\begin{aligned} \omega_\mathrm{e} &= \frac{v_\mathrm{h} i_\mathrm{g}(g)}{r_\mathrm{w}} \\ T_\mathrm{e} &= \frac{F_\mathrm{t} r_\mathrm{w}}{i_\mathrm{g}(g)} \text{ for } F_\mathrm{t} > 0 \end{aligned} \tag{5}$$

where $i_\mathrm{g}(g)$ is the transmission ratio corresponding to the gear position $g$, and $r_\mathrm{w}$ is the wheel radius.

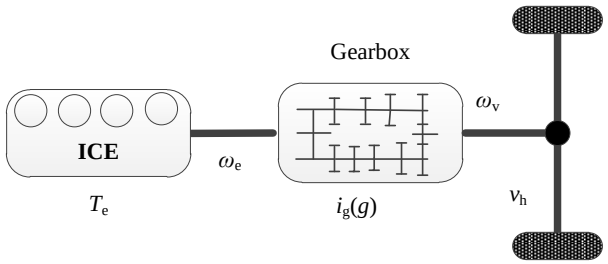



Fig. 4. Simplified structure of the host vehicle

The gear shift strategy of the gearbox determines the gear ratio and in this way adjusts the engine speed and torque by (5). In order to avoid shift jumps which yield a poor driving comfort, only a sequential gear shift is allowed. The gear shift schedule is designed to control the gear position $g(s)$ at the current time step based on the gear position $g(s-1)$ at the last time step and the gear shift command $u_\mathrm{g}(s)$ according to the dynamic model

$$g(s) = g(s-1) + u_\mathrm{g}(s) \tag{6}$$

where $s$ is the sampling interval for the gear shift, and $u_\mathrm{g}$ belongs to the set $\{-1, 0, 1\}$. -1 means downshift, 1 represents upshift, and 0 is sustainment.

### B. Control objective for eco-driving

The control objective for eco-driving in the car-following scenario consists in minimizing the fuel consumption and maintaining a desired inter-vehicle distance $L_\mathrm{des}$ from the preceding vehicle. The tracking performance of the host vehicle can be evaluated by the inter-vehicle distance deviation $\Delta L$ and the velocity deviation $\Delta v$. Therefore the objective in the optimal control problem is to minimize $\Delta L$, $\Delta v$, and the fuel consumption, which can be formulated as

$$J = \int_0^{T_\mathrm{cyc}} \left(\Delta L^2 + \Delta v^2 + \dot{m}_\mathrm{f}\right) dt \tag{7}$$

where $T_\mathrm{cyc}$ is the total length of the driving trip.

The optimal problem behind the eco-driving is to optimize the gear shift schedule and the traction force, i.e. $\boldsymbol{u} = [F_\mathrm{t}, u_\mathrm{g}]^\mathrm{T}$ such that the cost function (7) is minimized.

## III. Controller Design

In this section ADP based on an actor-critic structure is introduced as well as the controller design for the eco-driving.

### A. Introduction of ADP

ADP as a major variant of reinforcement learning is a learning-based control method which takes decisions based on interactions with the environment. The algorithm can realize online learning of nonlinear optimal control problems without the system model. It overcomes the curse of dimensionality within DP and facilitates the real-time implementation.

*1) Problem description:* Consider a general nonlinear discrete-time system

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t), \quad t = 0, 1, 2, ... \tag{8}$$

where $\boldsymbol{x}_t \in \mathbb{R}^m$ is the state vector, $\boldsymbol{u}_t \in \mathbb{R}^n$ is the input vector and $f(\boldsymbol{x}_t, \boldsymbol{u}_t)$ is the system dynamics function. A value function for the system (8) is defined as

$$\begin{aligned} V(\boldsymbol{x}_t) =& \sum_{i=t}^{\infty} \beta^{i-t} r(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ =& \sum_{i=t+1}^{\infty} \beta^{i-(t+1)} r(\boldsymbol{x}_i, \boldsymbol{u}_i) + r(\boldsymbol{x}_t, \boldsymbol{u}_t) \\ =& \beta V(\boldsymbol{x}_{t+1}) + r(\boldsymbol{x}_t, \boldsymbol{u}_t) \end{aligned} \tag{9}$$

where $\beta$ is a discount factor with $0 < \beta < 1$. $r(\boldsymbol{x}_i, \boldsymbol{u}_i)$ is the instantaneous cost depending on the control input $\boldsymbol{u}_i$ and the state $\boldsymbol{x}_i$.

This equation is also denoted as Bellman equation. Based on the Bellman optimality principle, the optimal value function for the policy $\boldsymbol{u}_t = \boldsymbol{h}(\boldsymbol{x}_t)$ is given by

$$V^*(\boldsymbol{x}_t) = \min_{\boldsymbol{h}(\cdot)} \left[ r\left(\boldsymbol{x}_t, \boldsymbol{h}(\boldsymbol{x}_t)\right) + \beta V^*\left(\boldsymbol{x}_{t+1}\right) \right] \tag{10}$$

which is referred to the Bellman optimality function.

The optimal policy results from (10) as

$$\boldsymbol{h}^*(\boldsymbol{x}_t) = \arg\min_{\boldsymbol{h}(\cdot)} \left[ r\left(\boldsymbol{x}_t, h(\boldsymbol{x}_t)\right) + \beta V^*\left(\boldsymbol{x}_{t+1}\right) \right]. \tag{11}$$

The optimal control problem in (11) can be solved by ADP through learning by trial-and-error interactions with the dynamic environment.

*2) ADP with iterative learning:* The iterative learning algorithm for ADP is processed as follows. Given a control policy $\boldsymbol{h}(\boldsymbol{x}_t)$, the value function can be updated iteratively using

$$V^{(i+1)}(\boldsymbol{x}_t) = r(\boldsymbol{x}_t, \boldsymbol{h}^{(i)}(\boldsymbol{x}_t)) + \beta V^{(i)}(\boldsymbol{x}_{t+1}) \tag{12}$$

where $i$ is the iteration step.

After the value iteration, the control policy $\boldsymbol{h}(\boldsymbol{x}_t)$ can be improved by

$$\boldsymbol{h}^{(i+1)}(\boldsymbol{x}_t) = \arg\min(r(\boldsymbol{x}_t, \boldsymbol{h}^{(i)}(\boldsymbol{x}_t)) + \beta V^{(i+1)}(\boldsymbol{x}_{t+1})). \tag{13}$$

The action dependent heuristic dynamic programming is one kind of the ADP design family with an actor-critic structure shown in Fig. 5. It can be used for systems with continuous space and has simple implementation. Here the neural networks are used for a smooth and nonlinear function approximation [25]. A critic network is used to approximate the value function $V(\boldsymbol{x}_t)$. The reward $r(\boldsymbol{x}_t, \boldsymbol{u}_t)$ resulting from the current action is the reinforcement signal. The action network provides the control variable $\boldsymbol{u}_t$.

*3) Critic network and online learning:* The critic network has input variables consisting of the $m$-dimensional state vector $\boldsymbol{x}_t$ and the $n$-dimensional action vector $\boldsymbol{u}_t$. There are $N_{\mathrm{ch}}$ neurons in the hidden layer and one output neuron which is presented in Fig. 6. $\boldsymbol{w}_{\mathrm{c}}^{(1)}(t)$ is the weighting matrix from the input neurons to the hidden neurons. The weighted sum of all inputs to each hidden neuron $k$ is given as

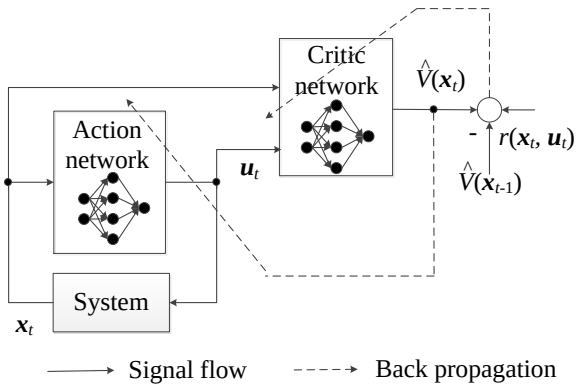



Fig. 5. Schematic diagram of the actor-critic structure

$\sigma_{\mathrm{c}k}(t) = \sum_{i=1}^{m} w_{\mathrm{c}ki}^{(1)}(t) x_{it} + \sum_{j=1}^{n} w_{\mathrm{c}kj}^{(1)}(t) u_{jt}$. The hyperbolic tangent transfer function $\phi(x) = \frac{1-e^{-x}}{1+e^{-x}}$ is utilized as activation function in the hidden neurons. It can approximate smooth nonlinear functions more accurately than a linear basis function. The output of each hidden neuron is then given as $q_{\mathrm{c}k}(t) = \phi(\sigma_{\mathrm{c}k}(t)) = \frac{1-e^{-\sigma_{\mathrm{c}k}(t)}}{1+e^{-\sigma_{\mathrm{c}k}(t)}}$. The weighting vector from the hidden layer to the output neuron is denoted as $\boldsymbol{w}_{\mathrm{c}}^{(2)}$. The output of the critic network $\hat{V}(\boldsymbol{x}_t)$ can finally be calculated from $\hat{V}(\boldsymbol{x}_t) = \boldsymbol{w}_{\mathrm{c}}^{(2)}(t)\boldsymbol{q}_{\mathrm{c}}(t) = \sum_{k=1}^{N_{\mathrm{ch}}} w_{\mathrm{c}k}^{(2)}(t) q_{\mathrm{c}k}(t)$.

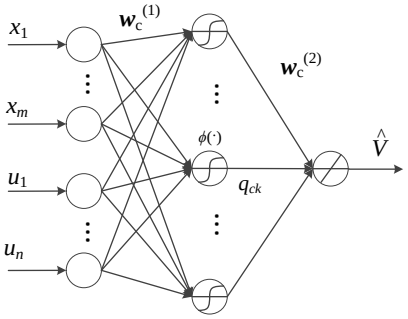



Fig. 6. Structure of the critic network

The error function of the critic network can be defined as the temporal difference with

$$e_{\mathrm{c}}(t) = \beta \hat{V}(\boldsymbol{x}_t) + r(\boldsymbol{x}_t, \boldsymbol{u}_t) - \hat{V}(\boldsymbol{x}_{t-1}) \tag{14}$$

where $r(\boldsymbol{x}_t, \boldsymbol{u}_t)$ is the external reward. Note that a shifted value function is considered here to simplify the implementation as proposed in [26]. The learning objective for the critic network is to minimize the error function $e_{\mathrm{c}}(t)$ by updating the parameters $\boldsymbol{w}_{\mathrm{c}}$, therefore the objective function for the critic network is introduced, i.e.

$$E_{\mathrm{c}}(t) = \frac{1}{2} e_{\mathrm{c}}^2(t). \tag{15}$$

A gradient descent adaptation algorithm is used to update the weights as

$$\boldsymbol{w}_{\mathrm{c}}^{p+1}(t) = \boldsymbol{w}_{\mathrm{c}}^{p}(t) + \Delta \boldsymbol{w}_{\mathrm{c}}^{p}(t) \tag{16}$$

where $p$ denotes the iteration index.

With a chain rule and backpropagation procedure, the weights adaption of the critic network from the hidden layer to the output layer results from

$$\begin{aligned}\Delta w_{ck}^{(2)}(t) &= \eta_{\rm c}(t)\left[-\frac{\partial E_{\rm c}(t)}{\partial w_{ck}^{(2)}(t)}\right],\ k=1,2,\ldots,N_{\rm ch},\\ \frac{\partial E_{\rm c}(t)}{\partial w_{ck}^{(2)}(t)} &= \frac{\partial E_{\rm c}(t)}{\partial e_{\rm c}(t)}\frac{\partial e_{\rm c}(t)}{\partial \hat{V}(\boldsymbol{x}_t)}\frac{\partial \hat{V}(\boldsymbol{x}_t)}{\partial w_{ck}^{(2)}(t)} = e_{\rm c}(t)q_{ck}(t)\end{aligned} \tag{17}$$

where $\eta_{\rm c}(t)$ is the learning rate of the critic network.

The weights adaption from the input layer to the hidden layer is calculated by

$$\begin{aligned}\Delta w_{ckj}^{(1)}(t) &= \eta_{\rm c}(t)\left[-\frac{\partial E_{\rm c}(t)}{\partial w_{ckj}^{(1)}(t)}\right],\ j=1,2,\ldots,m+n,\\ \frac{\partial E_{\rm c}(t)}{\partial w_{ckj}^{(1)}(t)} &= \frac{\partial E_{\rm c}(t)}{\partial e_{\rm c}(t)}\frac{\partial e_{\rm c}(t)}{\partial \hat{V}(\boldsymbol{x}_t)}\frac{\partial \hat{V}(\boldsymbol{x}_t)}{\partial q_{\rm ck}(t)}\frac{\partial q_{\rm ck}(t)}{\partial \sigma_{\rm ck}(t)}\frac{\partial \sigma_{\rm ck}(t)}{\partial w_{ckj}^{(1)}(t)}\\ &= e_{\rm c}(t)w_{ck}^{(2)}(t)\left[\frac{1}{2}\left(1-q_{ck}^2(t)\right)\right]x_{jt}.\end{aligned} \tag{18}$$

*4) Action network and online learning:* The action network has a similar structure with the critic network for the control policy. The input is the state vector $\boldsymbol{x}_t$ and the output is the action vector $\boldsymbol{u}_t$. The hidden layer contains $N_{\rm ah}$ neurons with the hyperbolic tangent transfer function. Defining $\boldsymbol{w}_{\rm a}^{(1)}$ as the weighting matrix from the input neurons to the hidden neurons, the value to each hidden neuron $k$ is calculated as $\sigma_{\rm ak}(t) = \sum_{i=1}^{m} w_{aki}^{(1)}(t)x_{it}$ and the output of each hidden neuron is $q_{\rm ak}(t) = \phi(\sigma_{\rm ak}(t)) = \frac{1-e^{-\sigma_{\rm ak}(t)}}{1+e^{-\sigma_{\rm ak}(t)}}$. The input to each output node $j$ is given as $\mu_j(t) = \sum_{k=1}^{N_{\rm ah}} w_{akj}^{(2)}(t)q_{akj}(t)$ where $\boldsymbol{w}_{\rm a}^{(2)}(t)$ is the weighting vector from the hidden neurons to the output neurons. Finally the output of the action network is the $n$-dimensional vector of action variables with $u_{jt} = \frac{1-e^{-\mu_j(t)}}{1+e^{-\mu_j(t)}}$, $j=1,2,\ldots,n$.

With the system states acting as the input variables, the optimal control action can be generated by the action network through adapting the weights $\boldsymbol{w}_a(t)$ so as to minimize $\hat{V}(t)$. The learning objective function can thus be defined as

$$E_{\rm a}(t) = \hat{V}(t). \tag{19}$$

The training of the action network is similar to the one used in the critic network. With a gradient descent rule, the weights can be adapted using

$$\boldsymbol{w}_{\rm a}^{p+1}(t) = \boldsymbol{w}_{\rm a}^{p}(t) + \Delta\boldsymbol{w}_{\rm a}^{p}(t) \tag{20}$$

where $p$ is the iteration index.

The weights adaption from the hidden layer to the output layer is

$$\begin{aligned}\Delta w_{akj}^{(2)}(t) &= \eta_{\rm a}(t)\left[-\frac{\partial E_{\rm a}(t)}{\partial w_{akj}^{(2)}(t)}\right],\ k=1,2,\ldots,N_{\rm ah},\\ \frac{\partial E_{\rm a}(t)}{\partial w_{akj}^{(2)}(t)} &= \frac{\partial \hat{V}(\boldsymbol{x}_t)}{\partial u_j(t)}\frac{\partial u_j(t)}{\partial \mu_j(t)}\frac{\partial \mu_j(t)}{\partial w_{akj}^{(2)}(t)}\\ &= e_{\rm a}(t)\frac{1}{2}\left(1-u_j^2(t)\right)\sum_{r=1}^{N_{\rm ch}}\left[w_{cr}^{(2)}(t)\frac{1}{2}\left(1-q_{cr}^2(t)\right)w_{cr(m+j)}^{(1)}(t)\right]q_{ak}(t)\end{aligned} \tag{21}$$

where $\eta_{\rm a}(t)$ is the learning rate of the action network.

The weights adaption from the input layer to the hidden layer is calculated from

$$\begin{aligned}\Delta w_{aki}^{(1)}(t) =& \eta_{\rm a}(t)\left[-\frac{\partial E_{\rm a}(t)}{\partial w_{aki}^{(1)}(t)}\right]\\ \frac{\partial E_{\rm a}(t)}{\partial w_{aki}^{(1)}(t)} =& \frac{\partial E_{\rm a}(t)}{\partial \hat{V}(\boldsymbol{x}_t)}\left[\frac{\partial \hat{V}(\boldsymbol{x}_t)}{\partial \boldsymbol{u}_t}\right]^{\rm T}\frac{\partial \boldsymbol{u}_t}{\partial \boldsymbol{\mu}(t)}\frac{\partial \boldsymbol{\mu}(t)}{\partial q_{\rm ak}(t)}\frac{\partial q_{\rm ak}(t)}{\partial \sigma_{\rm ak}(t)}\frac{\partial \sigma_{\rm ak}(t)}{\partial w_{aki}^{(1)}(t)}\\ =& \frac{\partial E_{\rm a}(t)}{\partial \hat{V}(\boldsymbol{x}_t)}\sum_{j=1}^{n}\frac{\partial \hat{V}(\boldsymbol{x}_t)}{\partial u_{jt}}\frac{\partial u_{jt}}{\partial \mu_j(t)}\frac{\partial \mu_j(t)}{\partial q_{\rm ak}(t)}\frac{\partial q_{\rm ak}(t)}{\partial \sigma_{\rm ak}(t)}\frac{\partial \sigma_{\rm ak}(t)}{\partial w_{aki}^{(1)}(t)}\\ =& e_{\rm a}(t)\sum_{j=1}^{n}\sum_{r=1}^{N_{\rm ch}}\left[w_{cr}^{(2)}(t)\frac{1}{2}\left(1-q_{cr}^2(t)\right)w_{cr(m+j)}^{(1)}(t)\right]\\ &\cdot\frac{1}{2}\left(1-u_{jt}^2\right)w_{ajk}^{(2)}(t)\frac{1}{2}\left(1-q_{ak}^2(t)\right)x_{it}.\end{aligned} \tag{22}$$

### *B. Controller design based on ADP*

The purpose of the eco-driving in the car-following process is to keep a desired inter-vehicle distance from the preceding vehicle and at the same time minimize the fuel consumption. The traction force $F_{\rm t}$ of the host vehicle is optimized to control the velocity to follow the preceding vehicle for comfortable driving. The optimal gear shift control $u_{\rm g}$ is derived to adjust the engine operating points for fuel economy. The overall control algorithm is presented in Fig. 7. It is assumed that the inter-vehicle distance deviation $\Delta L(t)$ and velocity deviation $\Delta v(t)$ can be measured. At each time step, the gear shift command $u_{\rm g}$ is chosen from the set $\{-1,0,1\}$. The engine torque and engine speed for each $u_{\rm g}$ can be obtained from (5) and (6), i.e.

$$\begin{aligned}\omega_{\rm e}(u_{\rm g}^j(t)) &= \frac{v_{\rm h}(t)}{r_{\rm w}}i_{\rm g}\left(g(t-1)+u_{\rm g}^j(t)\right)\\ T_{\rm e}(u_{\rm g}^j(t)) &= \frac{F_{\rm t}(t)r_{\rm w}}{i_{\rm g}\left(g(t-1)+u_{\rm g}^j(t)\right)},\ j\in\{1,2,3\}.\end{aligned} \tag{23}$$

The traction force $F_{\rm t}$ is then calculated from the action network within the ADP. The state variables in ADP is chosen as $\boldsymbol{x}_t = [\Delta L(t), \Delta v(t)]^{\rm T}$. The reward is defined by $r = \Delta L^2(t) + \Delta v^2(t) + \dot{m}_{\rm f}(t)$ to minimize the cost function in (7). During learning, the critic and action networks are adapted to derive the optimal control policy for $F_{\rm t}$. By comparing the fuel rate for each gear shift command $u_{\rm g}^j$, the optimal traction

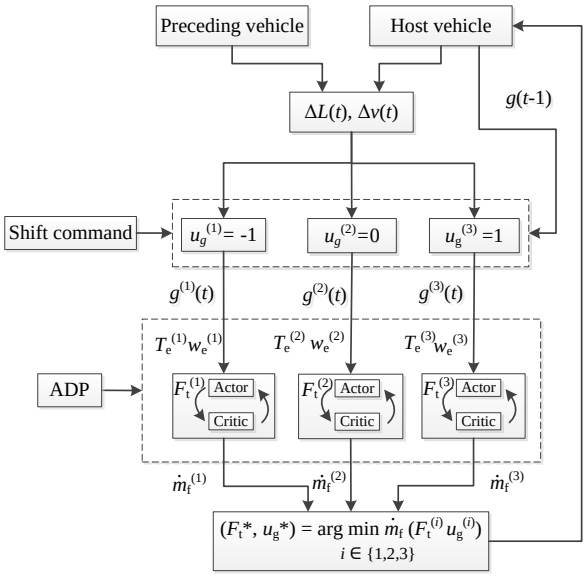



Fig. 7. Control algorithm for the eco-driving based on ADP

force $F_{\mathrm{t}}^*$ and the optimal gear shift $u_{\mathrm{g}}^*$ yielding the minimum fuel consumption can be obtained, i.e.

$$\left(F_{\mathrm{t}}^*, u_{\mathrm{g}}^*\right) = \arg\min_{F_{\mathrm{t}}, u_{\mathrm{g}}} \dot{m}_{\mathrm{f}}\left(F_{\mathrm{t}}^j, u_{\mathrm{g}}^j\right),\ j \in \{1,2,3\}. \tag{24}$$

This calculation can be processed parallelly for each $u_{\mathrm{g}}$ to improve the computational efficiency.

During the learning process for ADP, in each time step the critic network is adapted iteratively for maximal iteration number $N_{\mathrm{c}}$ or when $E_{\mathrm{c}}$ reaches the tolerance $T_{\mathrm{c}}$. If either of them is satisfied, the adaptation is stopped and the approximated value function is derived from the critic network. Then the parameters of the action network are adapted based on similar stopping criteria $N_{\mathrm{a}}$ and $T_{\mathrm{a}}$, where $N_{\mathrm{a}}$ is the maximal iteration number and $T_{\mathrm{a}}$ is the tolerance for $E_{\mathrm{a}}$ . After adaptation, the optimal traction force $F_{\mathrm{t}}$ and the gear shift command $u_{\mathrm{g}}$ are obtained and applied on the host vehicle. It is necessary to mention that the system model for the controller design is not required. The nonlinear system model in (1)-(5) is used for simulation. The learning procedure is summarized in the Algorithm 1.

**Algorithm 1** Learning procedure of the controller

1: Define the state variables : $\boldsymbol{x} = [\Delta L(t), \Delta v(t)]^{\mathrm{T}}$
2: Define the control variables : $\boldsymbol{u} = [F_{\mathrm{t}}, u_{\mathrm{g}}]^{\mathrm{T}}$
3: Initialize the parameters of the critic and action networks
4: Start from $t = 1$
5: Choose the gear shift command $u_{\mathrm{g}}$ from $\{-1, 0, 1\}$
6: Calculate the engine speed and engine torque with (23)
7: Calculate $E_{\mathrm{c}}(t)$, set $p = 0$
8: **while** $E_{\mathrm{c}}(t) > T_{\mathrm{c}}$ & $p < N_{\mathrm{c}}$ **do**
9: Update $\boldsymbol{w}_{\mathrm{c}}^{p+1}(t) = \boldsymbol{w}_{\mathrm{c}}^{p}(t) + \Delta \boldsymbol{w}_{\mathrm{c}}^{p}(t)$ with (16)-(18)
10: Set $p = p + 1$
11: **end while**
12: Calculate $E_{\mathrm{a}}(t)$, set $p = 0$
13: **while** $E_{\mathrm{a}}(t) > T_{\mathrm{a}}$ & $p < N_{\mathrm{a}}$ **do**
14: Update $\boldsymbol{w}_{\mathrm{a}}^{p+1}(t) = \boldsymbol{w}_{\mathrm{a}}^{p}(t) + \Delta \boldsymbol{w}_{\mathrm{a}}^{p}(t)$ with (20)-(22)
15: Set $p = p + 1$
16: **end while**
17: Choose optimal control variables yielding minimum $\dot{m}_{\mathrm{f}}$
18: Update state variables with the derived control variables
19: Return to step 5 for the next time instance

## IV. Simulation Study

In this section, the control performance of the ADP method is studied and the comparisons with different control methods are discussed. Particularly the adaptivity with respect to changes in the engine fuel consumption map is presented.

### A. Analysis of the control performance

The proposed controller for the eco-driving is evaluated with an urban driving cycle. A fixed preceding vehicle drives along the velocity trajectory from the Urban Dynamometer Driving Schedule (UDDS) and is followed by the host vehicle. The simulation parameters are given in Table I.

TABLE I
Parameters for online learning

| Parameter | Value |
|---|---|
| $\eta_a$: learning rate of actor network | $5 \cdot 10^{-5}$ |
| $\eta_c$: learning rate of critic network | $10^{-3}$ |
| $N_{\mathrm{ah}}$: number of neurons in actor network | 20 |
| $N_{\mathrm{ch}}$: number of neurons in critic network | 20 |
| $N_a$: maximum iteration step of actor network | 40 |
| $N_c$: maximum iteration step of critic network | 40 |
| $T_a$: error tolerance of actor network | $10^{-8}$ |
| $T_c$: error tolerance of critic network | $10^{-6}$ |

Simulation results for the first 300 seconds are shown in Fig. 8 for clear illustration. The velocity profile of the host vehicle is quite close to the one of the preceding vehicle and the distance deviation is kept within the range from -2.2 m to 2.2 m. This indicates a good tracking performance which is crucial for seamless traffic integration. The acceleration of the host vehicle is smoother than the one of the preceding vehicle. The magnitude is below $2\,\mathrm{m/s^2}$ to realize a comfortable driving. Higher gear positions are obtained compared to the one resulting from a rule-based strategy given in [24] where the gear shift is controlled based on the current driving velocity. The engine working points are adjusted to improve the fuel economy. Since the control objective is to minimize the fuel consumption, the gear shift frequency is not considered here.

The changing profiles for a part of weights in the action network during the vehicle driving process are presented in Fig. 9. At each time step, the weights are adapted online based on the reward $r$ containing the fuel consumption and the tracking error. The control policy from the action network is learned to minimize the reward and improve the fuel economy for different driving behaviors.

The fuel consumption comparison between the host and the preceding vehicles with different gear shift schedules is presented in Table II. Based on the gear shift strategy by the proposed ADP controller, the host vehicle requires the minimum fuel consumption to finish the driving cycle. The fuel consumption of the preceding vehicle is 5.03%

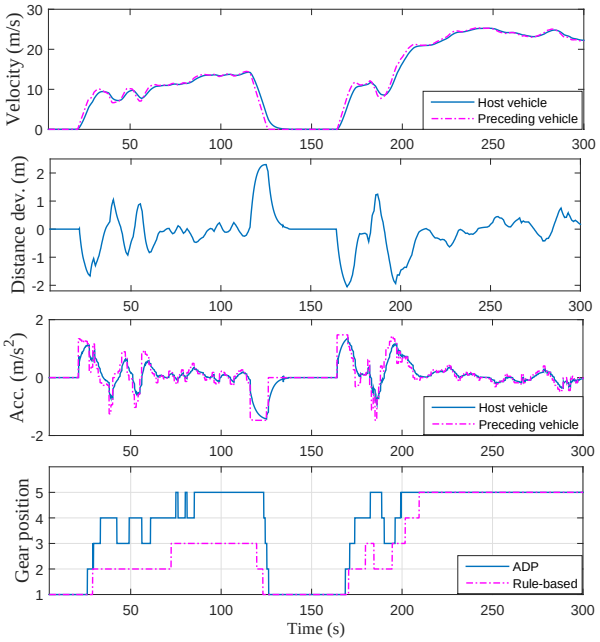



Fig. 8. Simulation results for UDDS: the velocity profiles of the host and preceding vehicles, the inter-vehicle distance deviation profile, the acceleration trajectories of the host and preceding vehicles, and the gear shift schedule comparison

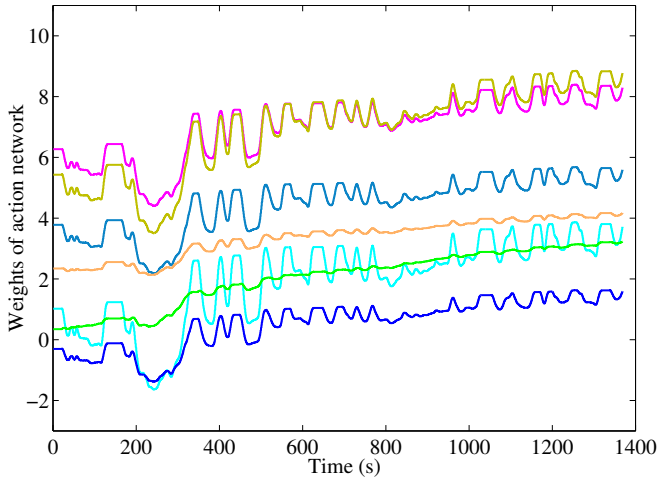



Fig. 9. Weights adaption of the action network

higher compared to the one from the host vehicle under the same gear shift schedule. This can be explained from Fig. 8 which shows that the host vehicle has smoother velocity and lower acceleration/deceleration than the preceding vehicle at each time step. The fuel consumption is mainly influenced by the vehicle acceleration. Higher acceleration will lead to more fuel consumption and requires a strong braking subsequently to maintain a desired inter-vehicle distance. The high deceleration for braking not only deteriorates the driving comfort, but also contributes to a lot of energy lost during braking and increases the total fuel consumption. Therefore the unnecessary acceleration and deceleration should be prevented by the host vehicle to improve the fuel economy. Furthermore, with the rule-based gear shift schedule, 14.95% more fuel is consumed by the host vehicle compared to the proposed gear shift strategy within the ADP controller. Since the ADP controller is adapted by the reward $r$ which incorporates the fuel rate at each time step, the control policy executes the gear shift command to adjust the engine working points in a fuel efficient region to reduce the reward during the learning process of ADP. Consequently the fuel economic gear shift policy is derived by ADP to reduce the total energy cost.

TABLE II
FUEL CONSUMPTION COMPARISON FOR UDDS

| Vehicle | Gear shift | Fuel (g) | Change rate |
|---|---|---|---|
| Host vehicle | ADP | 385.4 | - |
| Host vehicle | Rule-based | 443.0 | +14.95% |
| Preceding vehicle | ADP | 404.8 | + 5.03% |

To evaluate the influence by the gear shift schedule, comparisons of engine working points between two gear shift strategies are given in Fig. 10. The blue circle points represent the results by the ADP controller from Fig. 8 and the engine working points with the rule-based strategy in [24] are indicated by pink stars. It is obvious that the ADP controller adjusts the engine to work in the low-speed-high-load region with smaller fuel rate, while a large number of engine points by the rule-based gear shift strategy fall in the high load region. This will lead to more fuel consumption. Therefore the derived gear shift strategy can improve the fuel economy.

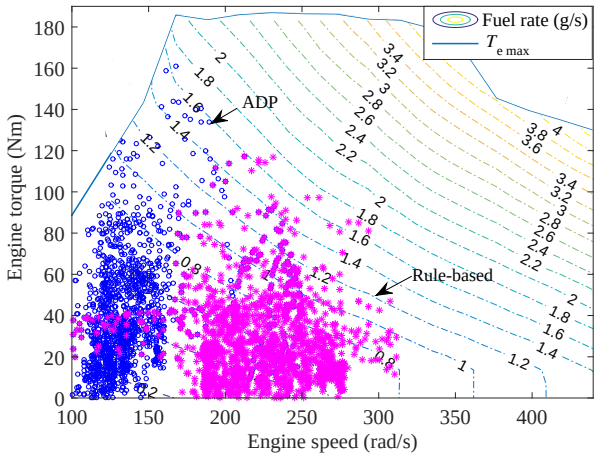



Fig. 10. Engine working points with different gear shift schedules for UDDS

In order to verify the optimal control performance particularly regarding the fuel economy for the eco-driving behavior, LQR is used as a benchmark to adjust the vehicle velocity for fuel efficiency. In the LQR only either the acceleration or a quadratic approximation of the fuel consumption map can be used. Here LQR is based on a linear model and the cost function relies on the acceleration $a_{\mathrm{h}}$, the inter-vehicle distance deviation $\Delta L$ and the velocity deviation $\Delta v$. The same nonlinear model and gear shift schedule are used for simulation. The fuel consumption comparisons between ADP and LQR are presented in Table III. LQR leads to 1.59% more fuel consumption than the ADP controller. The average engine efficiency resulting from ADP is 25.73%, which is higher than

the one based on LQR with 24.77%. The reason for this is that LQR is designed based on a linear system model. The control objective for fuel economy is realized by reducing the vehicle acceleration, but the nonlinearity from the engine fuel map is not regarded. However, the ADP controller is model-free, and the control policy is adapted by the reward depending on the fuel rate $\dot{m}_\mathrm{f}$, $\Delta L$ and $\Delta v$. The nonlinearity from the engine can be addressed during the actor-critic interaction. The control policy is updated which contributes to the fuel economy. The velocity profiles from the two controllers shown in Fig. 11 are very close, following the preceding vehicle well for good tracking performance. It indicates that both methods could provide good driving performance. Therefore the proposed ADP controller respecting the nonlinearity can improve the engine efficiency and reduce the fuel consumption.

TABLE III
FUEL CONSUMPTION COMPARISON BETWEEN ADP AND LQR FOR THE HOST VEHICLE

| Controller | Fuel (g) | Change | Avg. Engine efficiency |
|---|---|---|---|
| ADP | 385.4 | - | 25.73% |
| LQR | 391.0 | +1.59% | 24.77% |

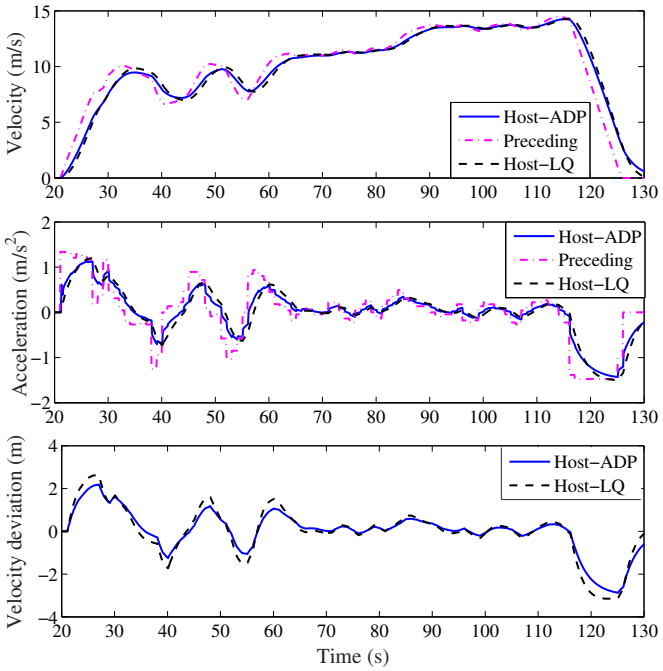



Fig. 11. Simulation comparison between ADP and LQ

To assess the real-time capability of the ADP-based controller, the computation time for each step is calculated. The simulation is executed on a desktop PC with MATLAB® R2014b. The computer is equipped with Intel® Core™ i7-4790 3.60 GHz CPU and 16 GB RAM. The sampling time for simulation is 100 ms to realize fast reaction for driving safety. Table IV presents the average and maximum execution time, which are all below the sampling time for the simulation. A major benefit from the online learning of the ADP controller is that the weight adaptation for the control policy is based on a gradient descent backpropagation rule. It has an explicit expression with low computation burden and can improve the calculation efficiency. Moreover, the parallel computation might be processed for the calculation of the gear shift control. The simulation study provides a first indication that the ADP-based controller can be applied in real time. Further studies with compiled code on an automotive electronic control unit are, however, needed.

TABLE IV
COMPUTATIONAL TIME FOR THE ADP-BASED CONTROLLER

| Cycle | Average execution time | Maximum execution time |
|---|---|---|
| UDDS | 5.8 ms | 31.2 ms |

### *B. Adaptivity with respect to the changes in the fuel consumption map*

The vehicle is a complex dynamic system which includes thousands of components working under sophisticated conditions. In the normal daily driving, the engine characteristic is affected by environment factors (air temperature, air pressure, fuel properties) or aging factors (friction, wear). Consequently the fuel consumption map along the speed and torque changes a lot. For model-based controllers, the control policy is usually designed based on a fixed system model, which is difficult to adapt during the vehicle operation. This will influence the control performance and the parameter turning for the controller to operate under different conditions is very elaborate. Figure 12 shows a modified fuel consumption map based on Fig. 3. It represents a change in the engine characteristics resulting in a different engine model. The economic region with low fuel rate moves to a different area. This can indicate a general change for the engine model.

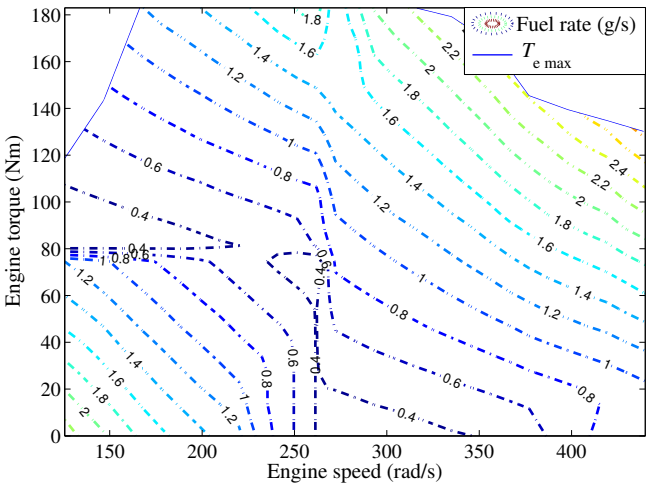



Fig. 12. Modified fuel consumption map

The proposed ADP algorithm does not require the system model. It can therefore adapt to the changes in the fuel consumption map due to the reward $r$ consisting in the fuel consumption at each time step. For a model-based method

like LQR, the gear shift schedule is designed based on a determined fuel consumption map in advance, and cannot adapt to the changes during the vehicle operation.

In the following, simulation is studied for the same driving cycle with the modified fuel consumption map in Fig. 12. The trained weights in Fig. 9 for the old fuel consumption map are used as the initial parameters in the networks. The proposed method with ADP is then learned online to adapt the changes and improve the control performance during the vehicle driving. Figure 13 presents the engine working points comparison between ADP and LQR. The blue circle points represent the results by the proposed ADP method, and the engine working points from LQR is shown by pink stars. For the results with LQR, the gear shift schedule is selected as the one in Section IV-A which realizes the optimal fuel economy for the original fuel consumption map. It can be observed that with ADP the engine works in the minimum fuel rate region, while LQR with the original gear shift schedule leads the vehicle to operate in the relative higher fuel rate area. This will definitely bring additional fuel consumption. Furthermore, through comparing the engine working points with ADP between Fig. 10 and Fig. 13, the ADP method can adjust the engine to work in the low fuel rate region, reacting on the changes in the fuel consumption map.

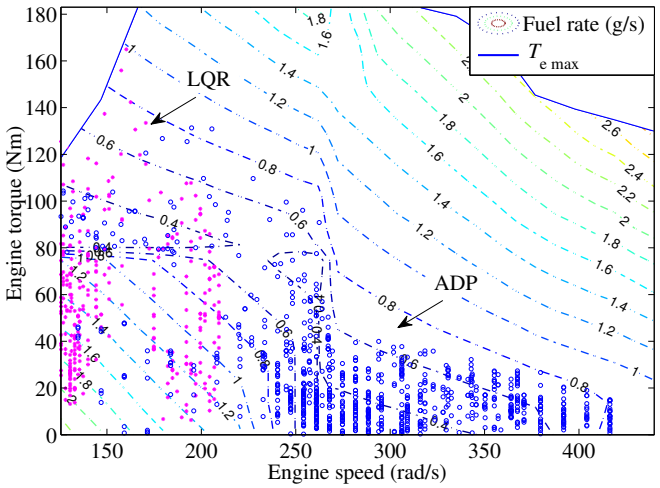



Fig. 13. Engine working points comparison between ADP and LQR

The weights adaption of the action network for the modified fuel consumption map is shown in Fig. 14. The online learning starts with weights at the end of the driving cycle in Fig. 9 for the old fuel consumption map and continues during the vehicle driving. It can be seen that when the fuel consumption map changes the weights are adapted again with the reward $r$ to improve the fuel economy and converge in the end.

The comparison of fuel consumption for the host and preceding vehicles with ADP and LQR is indicated in Table V. Here the same gear shift schedule derived from the modified fuel consumption map is applied on the vehicles. The host vehicle with the ADP controller requires the minimum fuel consumption. Around 1.12% more fuel is utilized by the preceding vehicle to drive for the same cycle. The fuel

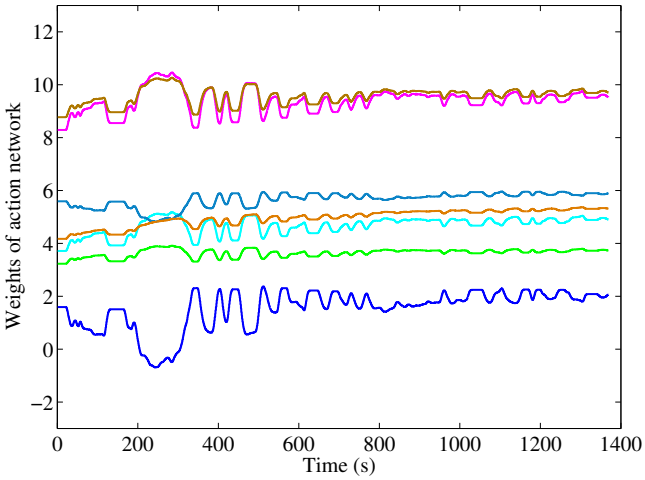



Fig. 14. Weights adaption of the action network for the new fuel consumption map

consumed by the host vehicle with LQR is increased by 2.32%. Therefore, the developed ADP method can adapt to the vehicle model change and improve the fuel economy.

TABLE V
FUEL CONSUMPTION COMPARISON FOR UDDS WITH MODIFIED FUEL CONSUMPTION MAP

| Vehicle | Controller | Fuel (g) | Change |
|---|---|---|---|
| Host vehicle | ADP | 473.9 | - |
| Preceding vehicle | ADP | 479.2 | + 1.12% |
| Host vehicle | LQR | 484.9 | + 2.32% |

## V. CONCLUSIONS

In this paper an online optimization method for the gear shift schedule and the velocity control in eco-driving to improve the fuel economy and driving comfort in a car-following process has been presented. The traction force of the host vehicle has been derived with ADP to follow the preceding vehicle and guarantee the desired inter-vehicle distance. The online gear shift schedule has been designed by enumeration to adjust the engine operating points for fuel economy. The controller is model-free and can adapt online to different driving situations. Simulation analysis for the urban driving cycle has indicated that the host vehicle can follow the preceding vehicle with a smooth acceleration and small inter-vehicle distance deviation. The host vehicle saves fuel by 5.03% and 1.12% for two engine fuel consumption maps compared to the preceding vehicle under the same gear shift schedule. With a rule-based gear shift schedule, 14.95% more fuel is consumed by the host vehicle than the one based on the proposed gear shift strategy within the ADP controller. Furthermore, LQR has been used as a benchmark which requires 1.59% and 2.32% more fuel consumption than the proposed control method. Finally, simulation studies under changes in the fuel consumption map have indicated the adaptivity of the proposed approach. In the future, the vehicle longitudinal motion control for ecological driving systems at the road intersection based

on ADP will be considered to reduce the fuel consumption and improve the traffic flow.

## REFERENCES


[1] J. Han, A. Sciarretta, L. L. Ojeda, G. De Nunzio, and L. Thibault, "Safe- and eco-driving control for connected and automated electric vehicles using analytical state-constrained optimal solution," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 163–172, 2018.

[2] K. Boriboonsomsin, A. Vu, and M. Barth, "Eco-driving: pilot evaluation of driving behavior changes among us drivers," *University of California Transportation Center*, 2010.

[3] M. Sivak and B. Schoettle, "Eco-driving: Strategic, tactical, and operational decisions of the driver that influence vehicle fuel economy," *Transport Policy*, vol. 22, pp. 96–99, 2012.

[4] S. E. Li, H. Peng, K. Li, and J. Wang, "Minimum fuel control strategy in automated car-following scenarios," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 998–1007, 2012.

[5] S. Xu, S. E. Li, X. Zhang, B. Cheng, and H. Peng, "Fuel-optimal cruising strategy for road vehicles with step-gear mechanical transmission," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3496–3507, 2015.

[6] E. Hellström, M. Ivarsson, J. Åslund, and L. Nielsen, "Look-ahead control for heavy trucks to minimize trip time and fuel consumption," *Control Engineering Practice*, vol. 17, no. 2, pp. 245–254, 2009.

[7] M. A. S. Kamal, M. Mukai, J. Murata, and T. Kawabe, "Ecological vehicle control on roads with up-down slopes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 783–794, 2011.

[8] H. Lim, W. Su, and C. C. Mi, "Distance-based ecological driving scheme using a two-stage hierarchy for long-term optimization and short-term adaptation," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 1940–1949, 2017.

[9] G. Mahler and A. Vahidi, "An optimal velocity-planning scheme for vehicle energy efficiency through probabilistic prediction of traffic-signal timing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2516–2523, 2014.

[10] N. Wan, A. Vahidi, and A. Luckow, "Optimal speed advisory for connected vehicles in arterial roads and the impact on mixed traffic," *Transportation Research Part C: Emerging Technologies*, vol. 69, pp. 548–563, 2016.

[11] Q. Jin, G. Wu, K. Boriboonsomsin, and M. J. Barth, "Power-based optimal longitudinal control for a connected eco-driving system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2900–2910, 2016.

[12] E. Ozatay, S. Onori, J. Wollaeger, U. Ozguner, G. Rizzoni, D. Filev, J. Michelini, and S. Di Cairano, "Cloud-based velocity profile optimization for everyday driving: A dynamic-programming-based solution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2491–2505, 2014.

[13] E. Ozatay, U. Ozguner, and D. Filev, "Velocity profile optimization of on road vehicles: Pontryagin's maximum principle based approach," *Control Engineering Practice*, vol. 61, pp. 244–254, 2017.

[14] S. Li, K. Li, R. Rajamani, and J. Wang, "Model predictive multi-objective vehicular adaptive cruise control," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 556–566, 2011.

[15] M. Wang, W. Daamen, S. P. Hoogendoorn, and B. van Arem, "Rolling horizon control framework for driver assistance systems. part I: Mathematical formulation and non-cooperative systems," *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 271–289, 2014.

[16] S. G. Dehkordi, G. S. Larue, M. E. Cholette, A. Rakotonirainy, and H. A. Rakha, "Ecological and safe driving: A model predictive control approach considering spatial and temporal constraints," *Transportation Research Part D: Transport and Environment*, vol. 67, pp. 208–222, 2019.

[17] Y. Shao and Z. Sun, "Robust eco-cooperative adaptive cruise control with gear shifting," in *American Control Conference (ACC), 2017*. IEEE, 2017, pp. 4958–4963.

[18] A. Weißmann, D. Görges, and X. Lin, "Energy-optimal adaptive cruise control combining model predictive control and dynamic programming," *Control Engineering Practice*, vol. 72, pp. 125–137, 2018.

[19] C. Desjardins and B. Chaib-draa, "Cooperative adaptive cruise control: A reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1248–1260, 2011.

[20] D. Zhao, Z. Hu, Z. Xia, C. Alippi, Y. Zhu, and D. Wang, "Full-range adaptive cruise control based on supervised adaptive dynamic programming," *Neurocomputing*, vol. 125, pp. 57–67, 2014.

[21] H. Chen, L. Guo, H. Ding, Y. Li, and B. Gao, "Real-time predictive cruise control for eco-driving taking into account traffic constraints," *IEEE Transactions on Intelligent Transportation Systems*, 2018.

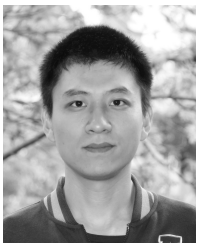
[22] J. Hu, Y. Shao, Z. Sun, and J. Bared, "Integrated vehicle and powertrain optimization for passenger vehicles with vehicle-infrastructure communication," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 85–102, 2017.

[23] G. Li and D. Görges, "Learning-based ecological adaptive cruise control for vehicles with step-gear transmissions," in *Proceedings of the 4th International Symposium on Advanced Vehicle Control, Beijing, China, AVEC2018*, 2018.

[24] T. Markel, A. Brooker, T. Hendricks, V. Johnson, K. Kelly, and et al., "ADVISOR: a systems analysis tool for advanced vehicle modeling," *Journal of Power Sources*, vol. 110, no. 2, pp. 255–266, 2002.

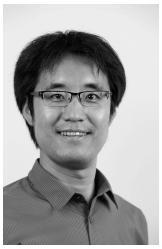
[25] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming*. Athena Scientific Belmont, MA, 1996, vol. 5.

[26] J. Si and Y.-T. Wang, "Online learning control by association and reinforcement," *IEEE Transactions on Neural networks*, vol. 12, no. 2, pp. 264–276, 2001.

**Guoqiang Li** received the B.S. and the M.S. degrees from the School of Mechanical Engineering, Beijing Institute of Technology, Beijing, China, in 2011 and 2014, the Dr.-Ing. degree from the Department of Electrical and Computer Engineering, University of Kaiserslautern, Kaiserslautern, Germany, in 2019. His research interests include design and control of vehicle transmission systems, optimal energy management of electrified vehicles, and motion planning of autonomous driving.

**Daniel Görges** (S'07–M'10) received the Dipl.-Ing. and Dr.-Ing. degrees from the Department of Electrical and Computer Engineering, University of Kaiserslautern, Kaiserslautern, Germany, in 2005 and 2011, respectively. He has held positions as Researcher and Juniorprofessor at the University of Kaiserslautern and as Research Group Leader at the German Research Center for Artificial Intelligence (DFKI). Since 2021, he has been Professor and the Head of the Institute of Electromobility, Department of Electrical and Computer Engineering, University of Kaiserslautern. His research interests include methods for model predictive control, distributed control, and learning control and their application in vehicular systems, transportation systems, and power systems.



**Meng Wang** received his PhD cum laude in Transportation Engineering from Delft University of Technology (TU Delft), The Netherlands. He is currently an Assistant Professor and Co-Director of the Electric and Automated Transport Research Lab of TU Delft. He has won IEEE ITS Society Best PhD Dissertation Award (2015) and IEEE ITSC Best Paper Award (2013). He is Associate Editor of IEEE Transactions on Intelligent Transportation Systems and has been the Associate Editors/IPC Members of the IEEE ITSS's flagship conferences of ITSC since 2013 and IV' since 2016. He serves as a member of IEEE ITSS Technical Committee on Cooperative Driving and TRB Subcommittee on Traffic Flow Modelling for Connected and Automated Vehicles. His main research interests are modeling, control design & impact assessment of automated and cooperative driving systems for safe and efficient traffic flow operations.