Full-range Adaptive Cruise Control Based on Supervised Adaptive Dynamic Programming☆

Dongbin Zhao[a], Zhaohui Hu[a,b], Zhongpu Xia[a,*], Cesare Alippi[a,c], Yuanheng Zhu[a], Ding Wang[a]

[a]*State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*

[b]*Electric Power Research Institute of Guangdong Power Grid Corporation, Guangzhou 510080, China*

[c]*Dipartimento di Elettronica e Informazione, Politecnico, di Milano, 20133 Milano, Italy*

Abstract

The paper proposes a Supervised Adaptive Dynamic Programming (SADP) algorithm for a full-range Adaptive Cruise Control (ACC) system, which can be formulated as a dynamic programming problem with stochastic demands. The suggested ACC system has been designed to allow the host vehicle to drive both in highways and in Stop and Go (SG) urban scenarios. The ACC system can autonomously drive the host vehicle to a desired speed and/or a given distance from the target vehicle in both operational cases. Traditional adaptive dynamic programming (ADP) is a suitable tool to address the problem but training usually suffers from low convergence rates and hardly achieves an effective controller. A supervised ADP algorithm which introduces the concept of Inducing Region is here introduced to overcome such training drawbacks. The SADP algorithm performs very well in all simulation scenarios and always better than more traditional controllers. The conclusion is that the proposed SADP algorithm is an effective control methodology able to effectively address the full-range ACC problem.

Keywords: adaptive dynamic programming, supervised reinforcement learning, neural networks, adaptive cruise control, stop and go

1. Introduction

Nowadays, driving safety and driver-assistance systems are of paramount importance: by implementing these techniques accidents reduce and driving safety significantly improves [1]. There are many applications derived from this concept, e.g., Anti-lock Braking Systems (ABS), Electronic Braking Systems (EBS), Electronic Brake-force Distribution systems (EBD), Traction Control Systems (TCS), Electronic Stability Program (ESP) [1].

1.1. Adaptive cruise control

Adaptive cruise control is surely another issue going in the direction of safe driving and, as such, of particular relevance. Nowadays, ACC is mounted in some luxury vehicles to increase both comfort and safety [2]. The system differentiates from the Cruise Control (CC) system mostly used in highway driving, which controls the throttle position to maintain the constant speed as set by the driver (eventually adjusted manually to adapt to environmental changes). However, the driver has always to brake when approaching the target vehicle proceeding at a lower speed. Differently, an ACC system equipped with a proximity radar [3] or sensors detecting the distance and the relative speed between the host vehicle and the one in front of it, proceeding in the same lane (target vehicle), can operate either on brake or the engine throttle valve to keep a safe distance.

As a consequence, the ACC does not only free the driver from frequent accelerations and decelerations but

☆This work was supported partly by National Natural Science Foundation of China under Grant Nos. 61273136, 61034002, and 60621001), Beijing Natural Science Foundation under Grant No. 4122083, and Visiting Professorship of Chinese Academy of Sciences.

*Corresponding author at: State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, PR China. Tel.: +8613683277856, fax:8610-8261-9580.

Email addresses: dongbin.zhao@ia.ac.cn (Dongbin Zhao), huzhaohui27@foxmail.com (Zhaohui Hu), zhongpu.xia@gmail.com (Zhongpu Xia), alippi@elet.polimi.it (Cesare Alippi), zyh7716155@163.com (Yuanheng Zhu), ding.wang@ia.ac.cn (Ding Wang)

also reduce the stress of the driver as pointed out in [4]. Interestingly, [5] showed that if 25% vehicles driving in a highway were equipped with the ACC system, congestions could be avoided. The ACC problem could be solved by considering different techniques, e.g., a PID controller [12], a fuzzy controller as pointed out in [11], a sliding mode approach [9] or a neural network [18].

ACC systems suggested in the literature, and currently implemented in vehicles, work nicely at a vehicle speed over 40 km/h and in highways [1], but always fail at a lower speed hence requiring accelerations (action on the throttle) and decelerations (mostly breaking) to keep a safe clearance to the target vehicle in urban areas. In this case, the driving activity increases significantly, even more within an urban traffic with an obvious impact on fuel consumption and pollutant emissions. To address the problem the literature suggested solutions like stop and go, collision warning and collision avoidance [22]. When the ACC and the SG solutions are considered together, we speak about a full-range ACC. A full-range ACC system with collision avoidance was proposed in [16]. There, driving situations were classified in three control modes based on the warning index and the time-to-collision: comfort, large deceleration and severe braking. Three controllers were proposed and combined to provide the ultimate control strategy. [16] pointed out how the full-range ACC problem was a nonlinear process requesting a nonlinear controller, for instance designed with reinforcement learning.

1.2. Reinforcement learning and adaptive dynamic programming

Reinforcement Learning (RL) [21] is suited for the ACC problem, because it can grant quasi-optimal control performance through a trial and error mechanism in a changing environment. However, the convergence rate of RL might be a problem [23] also leading to some inefficiency. Most of the time, the agent (the software implementing the controller) will learn the optimal policy after a relatively long training, especially when the model is characterized by a large state space. This inefficiency can be fatal in some real time control systems.

Supervised Reinforcement Learning (SRL) can be introduced to mitigate the RL problem, by combining Supervised Learning (SL) and RL and, hence, taking advantage of both algorithms. Pioneering work has been done in Rosenstein and Barto's [7, 19] where SRL was applied to solve the ship steering task and the manipulator control and the peg insertion task. All results clearly showed how SRL outperforms RL. In [17], a potential function was introduced to construct the shaping reward function; they proved that an optimal control policy could be gained. The results showed that such shaping method could be used also in dynamic models by dramatically shortening the learning time.

Our team applied the SRL control strategy to the ACC problem first in [14]. There, we showed that the speed and the distance control had enough accuracy and was robust with respect to different drivers [14]. However, since the state and the action needed to be discretized, there are some drawbacks. Firstly, the discretization of the distance, speed, and acceleration, introduces some fluctuations in the continuous control problem. Secondly, the higher number of discretized states cause the larger state and the action spaces. As a consequence, there always exists a conflict between control accuracy and required training time.

For continuous reinforcement learning problem, ADP was proposed in [8, 25] with neural networks mapping the relationships between states and actions, and the relationships between states, actions and performance index. More in detail, the algorithm uses a single step computation of the neural network to approximate the performance index which will be obtained by iterating the dynamic programming algorithm. The method provides us with a feasible and effective way to address many optimal control problems; examples can be found in the cart-pole control [13, 20], pendulum robot upswing control [26], urban intersection traffic signal control [15], freeway ramp metering [6, 27], play Go-Moku [28], and so on. However, the learning inefficiency of RL is also inherited in ADP but can also be remedied with a supervisor to formulate SADP.

1.3. The idea

In this paper we propose a novel effective SADP algorithm able to deal with the full-range ACC problem. The considered framework is as follows:

(1) There are two neural networks in SADP, the Action and the Critic networks. The Action network is used to map the continuous state space to the control signal; the Critic network is used to evaluate the goodness of the action signals generated by the Action network and provides advice while training both networks. In this way we avoid the curse of dimensionality caused by the large dimension of the discrete state-action pairs.
(2) The supervisor can always provide information for RL, hence speeding up the learning process.

In this paper, the ACC problem is described as a Markov decision process. The main contributions are as follows:

(1) A simple single neural network controller is proposed and optimized to solve the full-range adaptive cruise control problem.
(2) An inducing region scheme is introduced as a supervisor, which is combined with ADP, provides an effective learning algorithm.
(3) An extensive experimental campaign is provided to show the effectiveness and robustness of the proposed algorithm.

The paper is organized as follows. Section 2 formalizes the full-range ACC problem. Section 3 proposes the SADP algorithm based on the Inducing Region concept and presents design details. Section 4 provides experimental results based on typical driving scenarios. Section 5 summarizes the paper.

2. The adaptive cruise control

2.1. The ACC model

The ACC model is shown in Figure 1 with the nomenclature give in Table 1.

During driving, the ACC system assists (or replaces) the driver to control the host vehicle. In other words, ACC will control the throttle and the brake to drive the vehicle safely despite the uncertainty scenarios we might encounter. More in detail, there are two controllers in the ACC system: the upper and the bottom ones. The upper controller generates the desired acceleration control signal according to the current driving profile; the bottom controller transfers the desired acceleration signal to the brake or the throttle control action according to the current acceleration of the host vehicle.

Denote as $d_r(t)$ the distance at step t between the host and the target vehicles. Such a distance can be detected by radar or other sensing devices, and it is used to compute the instant speed of the target vehicle $v^T(t)$ (refer to Figure 1); the desired distance $d_d(t)$ between these vehicles is always set by the driver while the host vehicle speed $v^H(t)$ can be read from the speed encoder.

The control goal is to keep the host vehicle within a safety distance and maintain the safe relative speed $\Delta v(t)$

$$\Delta v(t) = v^H(t) - v^T(t). \tag{1}$$

Similarly, the relative distance $\Delta d(t)$ at step t is

$$\Delta d(t) = d_r(t) - d_d(t). \tag{2}$$

The upper controller goal is to simultaneously drive variables $(\Delta v(t), \Delta d(t))$ to zero by enforcing the most appropriate acceleration control action, more in detail, by taking into account the different driving habits.

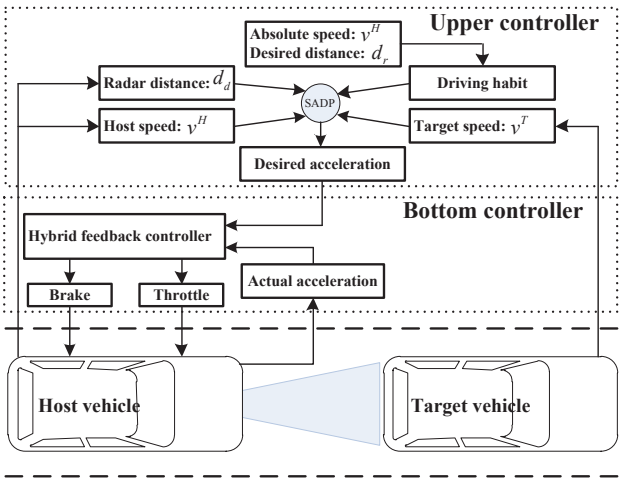


Figure 1: The SADP framework for the full-range ACC. The radar detects the distance between the two vehicles and the target vehicle's speed. The host vehicle speed and the current acceleration come from the mounted sensors. The upper controller generates the desired acceleration signal by combining the relative speed and the relative distance information. The bottom controller maps the acceleration to the brake or the throttle control signals.

The bottom controller manages both the throttle and the brake. A fuzzy gain scheduling scheme based on a PID control is used to control the throttle. A hybrid feed-forward & feedback control is applied to control the brake. The throttle and the brake controllers are coordinated by use of a proper switch logic. The control actions transfer the desired acceleration signal to the corresponding throttle position or braking strength [10].

2.2. The driving habit function

As previously discussed, different drivers have different driving habits: an intelligent ACC controller should learn the driving habit [29]. The host speed $v^H(t)$, the desired distance d_0 between the motionless host and target vehicles and the headway time index τ is adopted to characterize the driving habit

$$d_d(t) = d_0 + v^H(t)\tau \tag{3}$$

It comes out that the headway time is high for conservative drivers, and low for sportive drivers.

2.3. Driving scenarios

In a full-range ACC the host vehicle driving conditions can be cast into five scenarios, as shown in Figure 2.

(1) The CC scenario: the host vehicle travels at a constant speed without any target vehicle in front of it.

Table 1: ACC nomenclatures

Parameter	Description
$v^H(t)$	The speed of the host at step t
$v^T(t)$	The speed of the target at step t
$d_r(t)$	The distance between the host and the target vehicles at step t
$d_d(t)$	The distance the host driver desires to maintain at step t
$\Delta v(t)$	The relative speed at step t
$\Delta d(t)$	The relative distance at step t
$d^H(\Delta t)$	The distance the host vehicle travels in time interval Δt
$\Delta d_g(t)$	The maximum tolerable relative distance at step t
$\Delta v_g(t)$	The maximum tolerable relative speed at step t
d_0	The zero-speed clearance between the two vehicles
τ	The headway time

(2) The ACC scenario: both the target and host vehicles are running at high speed and the host vehicle needs to keep pace with the target vehicle or slow down to keep a safe distance to a slower forerunner.
(3) The SG scenario: this case simulates the frequent stop and go situations of the city traffic. The target vehicle stops at first, then moves again; this profile repeats frequently.
(4) The emergency braking scenario: the target vehicle stops suddenly with a large abnormal deceleration, the host vehicle must take an prompt braking action.
(5) The cut-in scenario: while the host car is operating in a normal ACC or SG mode, another vehicle interferes with it. More in detail, the third vehicle, coming from the neighboring lane, enters a position between the host and the target vehicles. The entering vehicle becomes the new target vehicle.

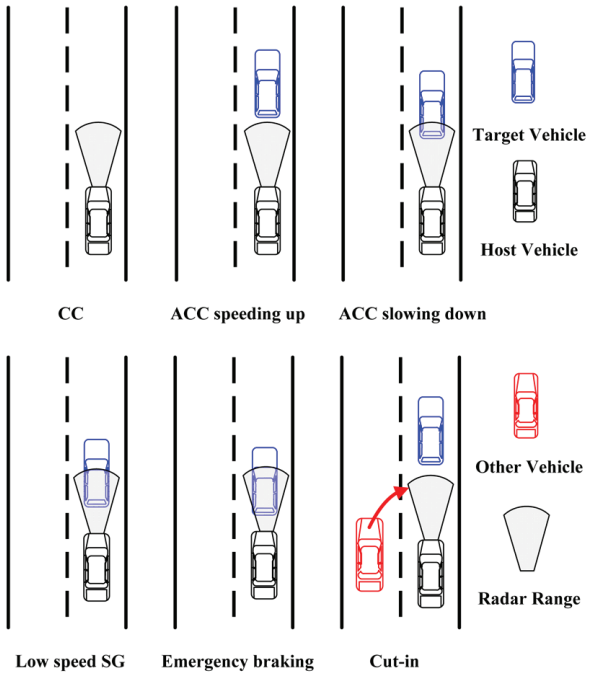


Figure 2: Different driving scenarios for the full-range ACC.

3. The SADP control strategy

3.1. The ADP framework

The structure of the SADP system is shown in Figure 3. The system includes a basic ADP and a supervisor (blue shadowed line). The Action and the Critic neural networks are present to generate the ADP framework. We recall that the Action network is used to model the relationship between the state and the control signal. Instead, the Critic network is used to evaluate the performance of the control signal as coming from the Action network. The Plant responds to the action and presents new state to the agent; afterwards, the reward is given. The dash lines represent the training process involving the two neural networks. Some major notations are listed in Table 2.

The training process can be summarized by the following procedure: At first, the agent takes action $u(t)$

Table 2: SADP nomenclatures

Parameter	Description
$x(t)$	The current state
$u(t)$	The control signal
$r(t)$	The reward
$J(t)$	The Critic network output
$R(t)$	The return or the rewards-to-go
$U_c(t)$	The desired objective
γ	The discount factor
N_{ah}	Number of hidden nodes, Action network
N_{ch}	Number of hidden nodes, Critic network
$E_a(t)$	Objective training function, Action network
$E_c(t)$	Objective training function, Critic network
$w_a(t)$	Weights matrix, Action network
$w_c(t)$	Weights matrix, Critic network
$l_a(t)$	Learning rate, Action network
$l_c(t)$	Learning rate, Critic network

following the input state $x(t)$ according to the Action network indication; the plant moves then to the next state $x(t+1)$ and the environment gives the agent a reward $r(t)$; then the Critic network output $J(t)$ provides an approximate performance index (or return); the Critic and the Action networks are then trained with error back-propagation based on the obtained reward [25]. These procedures iterate until the networks weights converging.

The ADP control strategy is stronger than a procedure based solely on RL. In fact, ADP possesses the common basic features of RL: state, action, transition, and reward. However, in ADP the state and the action are continuous values rather than discrete, and the method used to gain the action and the state values is rather different.

3.1.1. The reward and the return

The return $R(t)$, defined as "how good the situation is", is defined as the cumulated discounted rewards-to-go

$$\begin{aligned} R(t) &= r(t+1) + \gamma r(t+2) + \gamma^2 r(t+3) + \cdots \\ &= \sum_{k=0}^{T} \gamma^k r(t+k+1) \end{aligned} \quad (4)$$

where $0 \leq \gamma \leq 1$ represents the discount factor, t the step, $r(t)$ the gained reward and T the terminal step.

The higher the cumulated discounted future rewards-to-go is, the better the agent performs. However, the above definition needs the forward-in-time computation, hardly available. Therefore, in discrete RL, the

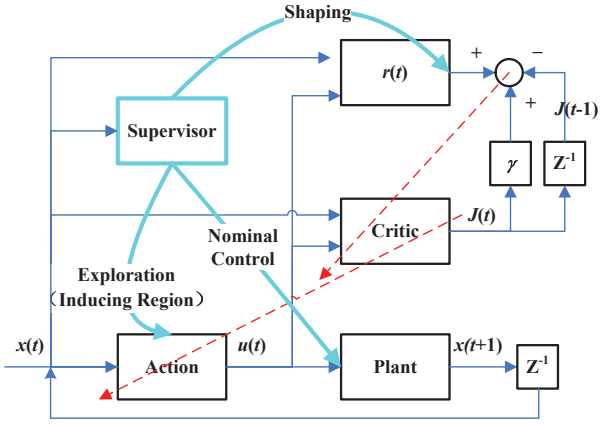


Figure 3: The schematic diagram of the SADP framework: The Action network is used to generate the control signal; the Critic network is used to evaluate the goodness of the control signal as generated by the Action network. The dash lines represent the training of those neural networks. There are three types of supervisors: shaping, nominal control, and exploration.

state value function $V(s)$ or the state-action value function $Q(s,a)$ is used to estimate $R(t)$. The final goal is to have a converged look-up Q-table in Q-learning [24]

$$Q(s,u) = Q(s,u) + \alpha[r(t) + \gamma \max_{u'} Q(s',u') - Q(s,u)]. \quad (5)$$

where α is the step size parameter, u and s is the current action and state, u' and s' is the next action and state, respectively.

There are many strategies for action selection, e.g., those based on the Boltzmann action selection strategy, the Softmax strategy and epsilon greedy strategy [21].

In ADP, the Critic network output $J(t)$ is used to approximate the state-action value function $Q(s,a)$. The Critic network embeds the gained experience (through trial and error) in the weights of the neural networks instead of relying on a look-up Q-table.

The definition of reward is somehow a tricky concept, as it happens with human learning. A wrong definition of reward will lead, with a high probability, to scarce learning results.

3.1.2. The Action network

The structures of the Action and the Critic networks are shown in Figure 4. Based on [20], simple three layered feed-forward neural networks with hyperbolic tangent activation function

$$Th(y) = \frac{1 - exp(-y)}{1 + exp(-y)}$$

is considered to solve the full-range ACC problem.

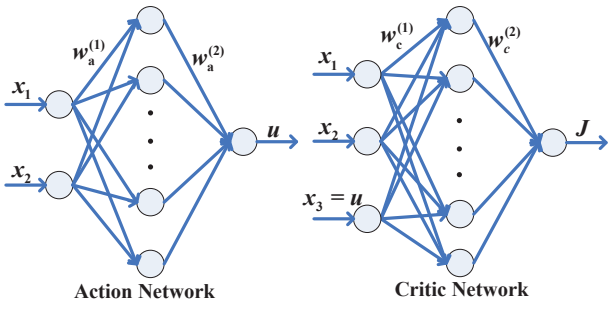


Figure 4: The structure of the Action and the Critic networks. The Action network has two inputs, namely, the relative distance and the relative speed; the output is the acceleration control signal. The Critic network has three inputs: the acceleration control signal, the relative distance and the relative speed; its output is the rewarding value $J(t)$.

The Action network's input is state $x(t) = (\Delta d(t),\ \Delta v(t))$. The output is $u(t)$ which can be derived from

$$u(t) = Th(m(t)), \tag{6}$$

$$m(t) = \sum_{i=1}^{N_{ah}} w_{a_i}^{(2)}(t) g_i(t), \tag{7}$$

$$g_i(t) = Th(h_i(t)),\ i = 1, 2, \cdots, N_{ah}, \tag{8}$$

$$h_i(t) = \sum_{j=1}^{2} w_{a_{ij}}^{(1)}(t) x_j(t),\ i = 1, 2, \cdots, N_{ah}, \tag{9}$$

where N_{ah} is the number of neurons in the hidden layer, $w_{a_{ij}}^{(1)}$ is the generic input weight of the Action network and $w_{a_i}^{(2)}$ is the generic output weight.

The Action network is trained to minimize the objective function

$$E_a(t) = \frac{1}{2} e_a^2(t), \tag{10}$$

$$e_a(t) = J(t) - U_c, \tag{11}$$

where U_c is the desired objective.

Training is performed with error back propagation

$$w_a(t+1) = w_a(t) + \Delta w_a(t), \tag{12}$$

$$\Delta w_a(t) = -l_a(t) \frac{\partial E_a(t)}{\partial J(t)} \frac{\partial J(t)}{\partial u(t)} \frac{\partial u(t)}{\partial w_a(t)}, \tag{13}$$

where $l_a(t)$ is the learning rate for the Action network.

3.1.3. The Critic network

The network receives as inputs both the state and the control signal, and outputs the estimated return $J(t)$,

$$J(t) = \sum_{i=1}^{N_{ch}} w_{c_i}^{(2)}(t) p_i(t), \tag{14}$$

$$p_i(t) = Th(q_i(t)),\ i = 1, 2, \cdots, N_{ch}, \tag{15}$$

$$q_i(t) = \sum_{j=1}^{3} w_{c_{ij}}^{(1)}(t) x_j(t),\ i = 1, 2, \cdots, N_{ch}, \tag{16}$$

where N_{ch} is the number of neurons in the hidden layer, $w_{c_{ij}}^{(1)}$ is the generic input weight of the Critic network to be learned, and $w_{c_i}^{(2)}$ the generic output weight.

The Critic network is trained by minimizing the objective function

$$E_c(t) = \frac{1}{2} e_c^2(t), \tag{17}$$

$$e_c(t) = \gamma J(t) - J(t-1) + r(t). \tag{18}$$

When the objective function $E_c(t)$ approaches zero, $J(t-1)$ can be derived from Eq. (18) as

$$J(t-1) = r(t) + \gamma r(t+1) + \gamma^2 r(t+2) + \cdots, \tag{19}$$

which is the same of return $R(t)$. Therefore, the convergence of the Critic network output $J(t)$ can be used to evaluate the goodness of the control signal.

Again, training is modeled as

$$w_c(t+1) = w_c(t) + \Delta w_c(t), \tag{20}$$

$$\Delta w_c(t) = -l_c(t) \frac{\partial E_c(t)}{\partial J(t)} \frac{\partial J(t)}{\partial w_c(t)}, \tag{21}$$

where $l_c(t)$ is the learning rate for the Critic network.

3.2. The disadvantages of the ADP

As mentioned above, ADP proposes a simple, feasible, and effective solution for the RL problem with continuous states and actions. Higher storage demand for the Q-table in Q-learning can be avoided and the "curse of dimensionality" problem in Dynamic Programming (DP) can be solved with a single step computation by using the above equations.

However, there are still some problems to be solved with ADP. The first is associated with the choice of the initial values of the network weights. Inappropriate configurations lead to poor Action and Critic networks (and then it becomes interesting to know how likely we will end in a good performing algorithm).

The second comes from U_c. This reward value is critical to the training phase. Usually, the reward is set 0 for encouragement and -1 for punishment and the return $R(t)$ is zero if the action is an optimal one. Hence the output $J(t)$ of the Critic network converges to 0 if optimal actions are always taken (and the induced value of U_c is 0). But in some complex cases a continuous reward would be a better choice. With error back propagation, a large discrepancy on U_c might lead to a

large training error which will affect negatively the performance of the controller.

The above problems can be solved if we consider a supervisor to guide the learning process.

3.3. The supervisor: Inducing Region

As shown in Figure 3, SADP combines the structure of ADP and SL. Therefore, the agent learns from the interaction with the environment as well as benefits from a feedback coming from the supervisor.

There are three ways to implement the supervisor in SADP [7]: (1) shaping: the supervisor gives additional reward, hence simplifying the learning process for the agent; (2) nominal control: the supervisor gives additional direct control signal to the agent; (3) exploration: the supervisor gives hints that indicate which action should be taken.

The exploration way gives the smallest supervisor information and is adopted here. Since the goal of the control system is to drive the relative speed and the relative distance to zero, the desired target requires that both $v(t)$ and $d(t)$ satisfy

$$\begin{cases} |\Delta v(t)| < \epsilon_v \\ |\Delta d(t)| < \epsilon_d \end{cases}, \tag{22}$$

where ϵ_v and ϵ_d are feasible tolerable small positive values for $\Delta v(t)$ and $\Delta d(t)$, respectively.

The aim of the full-range ACC is to satisfy the above inequalities or "goal state" as soon as possible (promptness in action) and stay there during the operational driving of the vehicle. However, at the beginning, the agent is far away from the goal state, especially when no priors are available. If the goal state region is too small the agent will always be penalized during learning and the training process will hardly converge. Even if a high number of training episodes are given there it is not guaranteed that the ADP will learn an effective control strategy. On the contrary, if the goal state area is too large, then the learning task might converge at the expenses of a poor control performance.

It makes sense to have a large goal state area at the beginning to ease the agent entering into a feasible region and reduce gradually afterwards the area, as learning proceeds, to drive the learning towards the desired final state configuration. In other terms, it means that the supervisor will guide the agent towards its goal through a rewarding mechanism. This concept is at the base of the Inducing Region where ϵ_v and ϵ_d evolve with time.

3.4. SADP for the full-range ACC

There are five components in the SADP framework: the state, the action, the state transmission matrix, the reward and the supervisor.

3.4.1. The state

The relative speed $\Delta v(t)$ and the relative distance $\Delta d(t)$ are the state variables $x(t) = (\Delta v(t), \Delta d(t))$. The aim of the full-range ACC is to achieve the final goal state with the minimum amount of time and an Inducing Region characterized as

$$\begin{cases} |\Delta v(t)| < 0.072\ km/h \\ |\Delta d(t)| < 0.2\ m \end{cases}, \tag{23}$$

Besides the goal state, a special "bump" state is introduced and reached when the host vehicle collides with the target one, namely,

$$\Delta d(t) + d_d(t) < 0. \tag{24}$$

3.4.2. Acceleration: the control variable

The full-range ACC problem can be intended as mapping different states to corresponding actions. Here, the action is the acceleration of the host vehicle. In view of the comfort of the driver and passengers, the acceleration should be bounded in the $[-2, 2]\ m/s^2$ interval in normal driving conditions, and to $[-8, -2]\ m/s^2$ in severe and emergency braking situations [16]. It is required to transfer $u(t)$ which is within the $[-1, 1]$ range into the range $[-8, 2]\ m/s^2$, namely,

$$a = \begin{cases} |a_{min}| \cdot u & u < a_{max}/|a_{min}| \\ a_{max} & u \geq a_{max}/|a_{min}| \end{cases}, \tag{25}$$

where a_{min} is $-8\ m/s^2$ and a_{max} is $2\ m/s^2$ here.

3.4.3. The state transition

When the vehicle is in state $x(t) = (\Delta v(t), \Delta d(t))$, and takes action $a = a^H$, the next state $x(t+1)$ is updated as

$$\begin{cases} v^H(t+1) = v^H(t) + a^H(t)\Delta t \\ d^H(\Delta t) = v^H(t) + a^H \Delta t^2/2 \\ \Delta v(t+1) = v^H(t+1) - v^T(t+1) \\ \Delta d(t+1) = \Delta d(t) - (d^H(\Delta t) - (v^T(t) \\ \qquad\qquad + v^T(t+1))\Delta t/2) \end{cases}, \tag{26}$$

where Δt represents the sampling time. It can be seen that the next state $x(t+1)$ cannot be computed after taking an action, since the target speed of the next step or the acceleration is unknown.

3.4.4. The reward

The reward is 0 when the agent reaches the goal state, -2 when it reaches the bump state, -1 otherwise. The reward provides an encouragement for achieving the goal, heavy penalty for collision, and a slight punishment for having not reached the target state.

3.4.5. Inducing Region

The updating rule for the Inducing Region is given by

$$\begin{cases} \Delta d_g(t) = \Delta d_g(t-1) - C_d, \\ \qquad 0.2 < \Delta d_g(t) < \Delta d_g(0); \\ \Delta d_g(t) = 0.2, \quad 0.2 \geq \Delta d_g(t); \\ \Delta v_g(t) = \Delta v_g(t-1) - C_v, \\ \qquad 0.072 < \Delta v_g(t) < \Delta v_g(0); \\ \Delta v_g(t) = 0.072, \quad 0.072 \geq \Delta v_g(t); \end{cases} . \tag{27}$$

where the $\Delta d_g(t)$ and $\Delta v_g(t)$ characterize the goal state area for $\Delta d(t)$ and $\Delta v(t)$, respectively. C_d and C_v are the constant shrinking length at each step for the goal distance and the goal speed, set to 0.3 m and 0.36 km/h. $\Delta d(0)$ and $\Delta v(0)$ are the initial goal state ranges, set to 18 m and 18 km/h, respectively. As presented above, the goal state area gradually shrinks to guide the Action network towards the final goal.

3.5. Learnability vs. stability

It is very hard to prove the stability of the suggested full-range ACC in a close form. However, we can make a strong statement in probability by inspecting the learnability properties of the suggested full-range ACC problem. Since the suggested SADP algorithm is Lebesgue measurable with respect to the weight spaces of the action and critical networks we can use Randomized Algorithms [30, 31] to assess the difficult of learning problem.

To do this, we define at first the "performance satisfaction" criterion $P_s(S)$ and say that the performance provided by the full-range ACC system S is satisfying when:

(1) convergence: the Action and Critic networks converge, i.e., they reach a fixed configuration for the weights at the end of the training process.
(2) comfortable: the acceleration of the host vehicle is mostly within $[-2, 2]m/s^2$ range and comes out of that range only in emergency braking situations.
(3) accurate: the suggested full-range ACC system can effectively control the host vehicle to achieve the final goal state defined in Eq.(23) and, then, stay there.

When the performance is satisfied, $P_s(S)$ assumes value 1, 0 otherwise.

As SADP learns through a trial and error mechanism, it will explore exhaustively the state space provided that the number of experiments is large enough. At the end of the training process we can then test whether the performance of the full-range ACC system $P_s(S)$ is 1 or not. Of course, the performance satisfaction criterion must be evaluated on a significant test set containing all those operational modalities the host vehicle might encounter during its driving life. It is implicit that if the full-range ACC system satisfies the performance satisfaction criterion it is also stable. The opposite does not necessarily hold.

We observe that the unique randomization in the training phase is associated with the process providing the initial values for the network weights. Afterwards, SADP is a deterministic process that, given the same initial configuration of weights and the fixed training data (experiment) provides the same final networks (not necessarily satisfying the performance criterion).

Train now a generic system S_i and compute the indicator function $I_d(S_i)$ defined as

$$I_d(S_i) = \begin{cases} 1, & if\ P_s(S_i) = 1 \\ 0, & otherwise \end{cases} . \tag{28}$$

In fact, the indicator function $I_d(S_i) = P_s(S_i)$ and states whether the generic system satisfies the performance criterion or not for the full-range ACC for the i-th training process.

Let ρ be the probability that a trained system S satisfies the performance criterion for the full-range ACC. ρ is unknown but can be estimated with a randomization process as suggested in [30, 31]. More specifically, we can evaluate the estimate $\hat{\rho}_N$ of ρ by drawing N initial configurations for the Action and Critic networks, hence leading to the N systems

$$\hat{\rho}_N = \frac{1}{N} \sum_{i=1}^{N} I_d(S_i). \tag{29}$$

To be able to estimate ρ we wish the discrepancy between $\hat{\rho}_N$ and ρ to be small, say below a positive ε value, i.e., $|\rho - \hat{p}_N| < \varepsilon$. However, the satisfaction of the inequality is a random variable, which depends on the particular realization of the N systems. We then request the inequality to be satisfied with high confidence and require

$$Pr(|\rho - \hat{p}_N| < \varepsilon) \geq 1 - \delta, \tag{30}$$

where $1-\delta$ represents the confidence value. The above equation holds for any value of ε and δ provided that N satisfies the Chernoff's bound [32].

$$N \geq \frac{\ln(\frac{2}{\delta})}{2\varepsilon^2} \tag{31}$$

If we now select a high confidence, say $1-\delta$ then, with probability at least $1-\delta$ inequality $|\rho - \hat{p}_N| < \varepsilon$ holds. In turn, that means that the unknown probability ρ is bounded as

$$\hat{p}_N - \varepsilon \leq \rho \leq 1 \tag{32}$$

Eq (32) must then be intended as follow: designed a generic system S with the SADP method and the above hypotheses, the system will satisfy the performance satisfaction criterion with at least probability $\hat{p}_N - \varepsilon$; the statement holds with confidence $1-\delta$. In other terms, if $\hat{p}_N$ assumes high values the learnability for a generic system is granted with high probability and, as a consequence, the stability for the system satisfying the performance criterion is implicitly granted as well.

4. Experimental results

4.1. Longitudinal vehicle dynamic model

We adopt the complete all-wheel-drive vehicle model present in the SimDriveline software of Simulink/Matlab. The vehicle model is shown in Figure 5. It combines the Gasoline Engine, the Torque Convertor, the Differential, the Tire, the Longitudinal Vehicle Dynamics and the Brake blocks. The throttle position, the brake pressure and road slope act as input signals, the acceleration and the velocity as output signals. Such a model has been used to validate the performance of the suggested controllers [10].

4.2. Training process

In the SADP model, the discount factor γ is 0.9, the initial learning rates for the Action and the Critic networks are set to 0.3, and decrease to 0.001 by 0.05 at each step. Both the Action and the Critic networks are three layered feed-forward neural networks with 8 hidden neurons. The network weights are randomly generated initially, to test the SADP learning efficiency and drawn from section 3.5. Here, we set $\tau = 2\ s$, $d_0 = 1.64\ m$ and $\Delta t = 1\ s$.

An experiment, e.g., a full training of a controller requires presentation of the same episode (training profile) 1000 times. Each episode is as follows:

(1) The host speed and the initial distance between the two vehicles are 90 km/h and 60 m, respectively. The target speed is 72 km/h and fixed in time interval $[0, 90)$ s;
(2) The target speed then increases to 90 km/h in time interval $[90, 100)$ s with fixed acceleration;
(3) The target maintains the speed at 90 km/h in the time interval $[100, 150)$ s.

In this case, $\Delta v(0) = 18\ km/h$ and $\Delta d(0) = 18.36\ m$, hence the agent starts from the initial state $x(0) = (18, 18.36)$, takes continuous action at each time instance and either ends in the bump state or in the goal state. We have seen that if a collision occurs, a heavy penalty is given and the training episode will restart. Although the agent is trained in a simple scenario, the training process is not trivial. SADP, through trial and error, will force the agent to undergo many different states. The training phase is then exhaustive and the trained SADP controller shows a good generalization performance.

For comparison we also carried out training experiments with ADP which has the same final goal as SADP. The training episodes are increased until 3000 to give the agent more time to learn. Table 3 shows the performance comparison between SADP and ADP. We say that one experiment is successful when both the Action and Critic networks weights keep fixed for the last 300 episodes and the performance of the system evaluated on the test set satisfies the performance criterion defined in section 3.5. As expected, the presence of the supervisor guarantees the training process convergence so that the full-range ACC is always achieved.

Table 3: Convergence comparison between SADP and ADP

	Training episodes	Number of experiments	Number of success
SADP	1000	1000	999
ADP	1000	1000	0
ADP	3000	1000	0

Analyzing the only one failed experiment from SADP, we obtain that the Action and Critic networks weights keep fixed for the last 224 episodes. If the number of episodes defining the success is smaller, e.g., 200, then this experiment can also be thought of as a success one.

4.3. Generalization test with different scenarios

The effectiveness of the obtained SADP control strategy is tested in the driving scenarios of Section 2. The driving habit parameters are changed as follows:

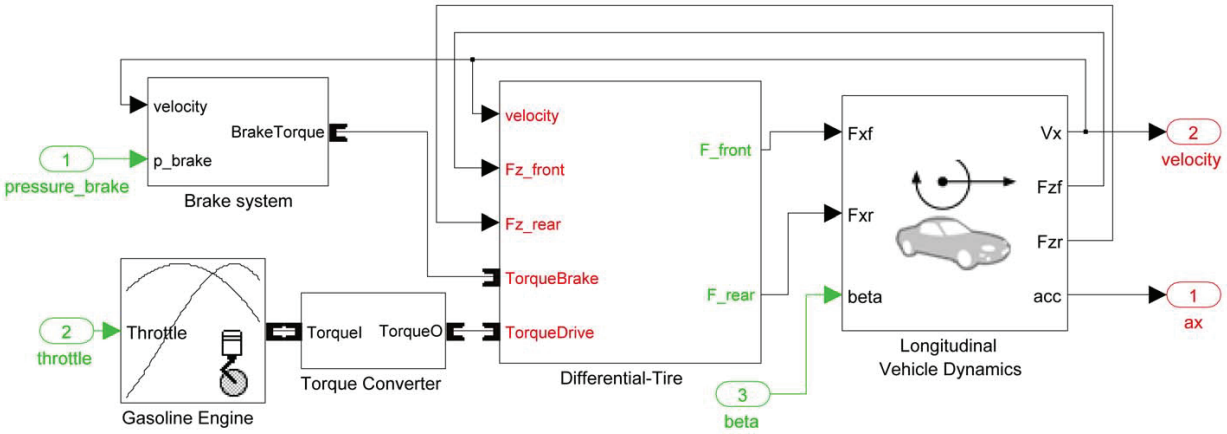


Figure 5: Longitudinal vehicle dynamic model suggested within Matlab/Simulink [10].

$\tau = 1.25\ s$ and $d_0 = 4.3\ m$. Here, the CC scenario is omitted for its simplicity. The test scenarios include **the normal ACC driving scenario, the SG scenario, the emergency braking scenario, the cut-in scenario and the changing driving habit scenario.**

[16] proposed three different control strategies for the full-range ACC problem, namely, the safe, the warning, and the dangerous modes as the function of the warning index and the time-to-collision. The outcome controller provides an effective control strategy that we consider here for comparison.

In this paper, only a single trained nonlinear controller is used to deal with the full-range ACC problem.

4.3.1. The normal ACC scenario

The target vehicle runs with varied speeds and the host vehicle has to either keep a safe distance or a relative speed with respect to the target.

Results are shown in Figure 6. We comment that speed and distance requests are nicely satisfied. Moreover, the requested acceleration is more than acceptable. More in detail, at time 20 s the host vehicle reaches the goal state, and stays there. Whenever the target vehicle slows down or increases its speed, the host vehicle reacts to the change by imposing the corresponding acceleration action. The normal ACC problem can be thought as a linear process, while the mixed control strategy [16] provides a near-optimal control. Experiments show that the obtained SADP behaves as well as the mixed control strategy.

4.3.2. The SG scenario

Starting from 20 km/h the target vehicle accelerates to reach a speed of about 40 km/h and, then, decelerates to a full stop. Results are shown in Figure 7. We appreciate the fact that the host vehicle performs well both in distance and speed control. In the first 10 s, the host vehicle decelerates to a stop, then the host vehicle accelerates (constant acceleration) until time 80 s. Afterwards, it keeps a constant speed for a period and, finally, goes to a full stop. As in the case of the normal ACC scenario, the mixed control strategy [16] and SADP both provide near-optimal control performance, indicating the good learning ability of SADP.

4.3.3. The emergency braking scenario

This scenario is designed to test the control performance under extreme conditions to ensure that driving safety is achieved. The target vehicle brakes suddenly at time instant 60 s and passes from 80 km/h to 0 km/h in 5 s.

Figure 8 shows the experimental results, clearly indicating that both the methods stop the vehicle successfully with similar clearances to the target vehicle, but the SADP control strategy outperforms the mixed control strategy [16] with a smoother acceleration (e.g., see the deceleration peak requested by the mixed approach). In [16] the control signal was a combination of two control strategies; as such it introduced frequent spikes in the acceleration signal when prompt actions were requested.

4.3.4. The cut-in scenario

The host and target vehicles proceed at high speed, A vehicle from the neighboring lane interferes and inserts between the target and the host vehicle, which needs to the host one brake. The distance to the new target vehicle abruptly reduces up to 50%.

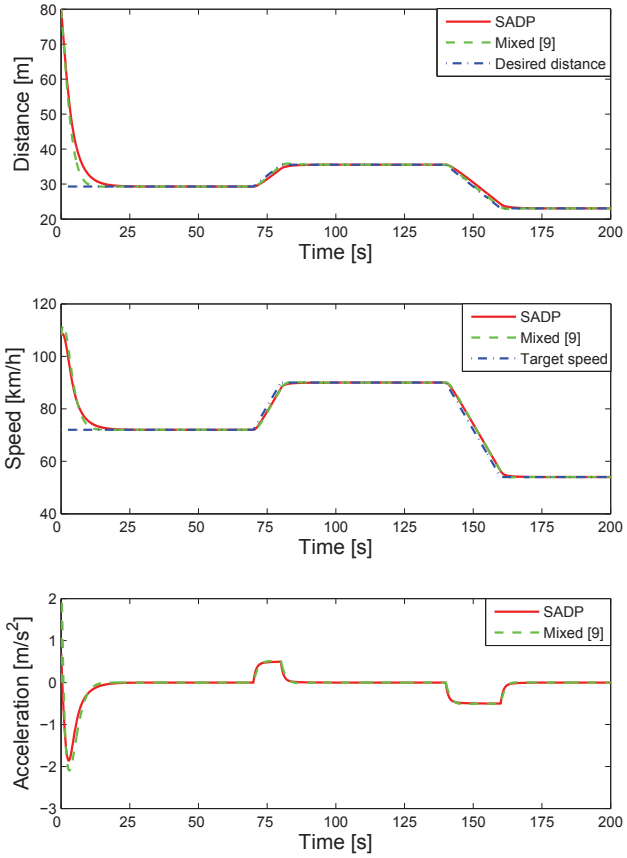


Figure 6: Experimental results with SADP and the mixed control strategies in the normal ACC scenario: (a) distance; (b) speed; (c) acceleration.

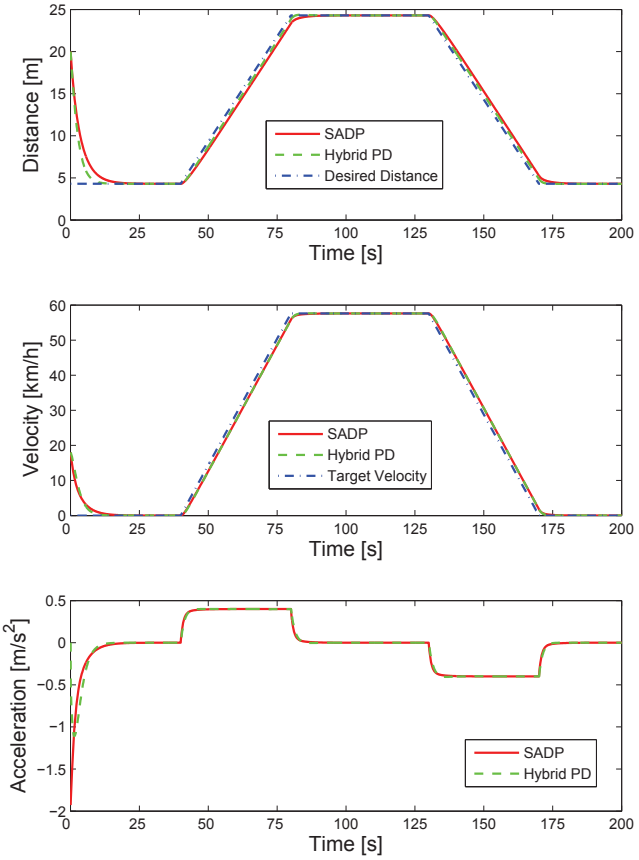


Figure 7: Experimental results with SADP and the mixed control strategies. SG scenario: (a) distance; (b) speed; (c) acceleration.

Figure 9 shows that both algorithms perform well. Since there is a significant reduction in the safety distance, the host brakes to avoid the crash. This is a normal action in current ACC systems. In our algorithm, small driving habit parameters must be set to emulate the behavior of a sportive driver, which might leave a very small and safety distance for the neighboring vehicle to cut in.

4.3.5. The changing driving habit scenario

The above four scenarios are set with parameters $d_0 = 4.3\ m$ and $\tau = 1.25\ s$. In practical implementations, there could be several driving habits for the human driver to choose from. We verify the proposed algorithm and it always meets the driver expectation.

4.4. Robustness

In real vehicles, measurement errors introduce uncertainties on the relative distance and the relative speed measurements. Such uncertainties affect the controller performances. In the following we consider sensing uncertainties by adding noise to the real values. Figure 10 shows an emergency braking situation. A random 2% in magnitude uniform noise is added to the target speed. Since the relative distance is derived from the speed uncertainty propagates. We see that SADP outperforms the mixed control strategy [16], with a higher accuracy in the distance control and a smoother acceleration requirements. We verified that SADP provides satisfactory performances when the noise increases up to 5% in magnitude.

Other uncertainties may include the changing load of the vehicle and the friction between the vehicle and the road. They can be solved with the aforementioned bottom controller.

4.5. Discussions

We can conclude that the SADP control strategy is robust and effective in different driving scenarios.

Furthermore, the changing driving habit scenario immediately shows the generalization performance of the

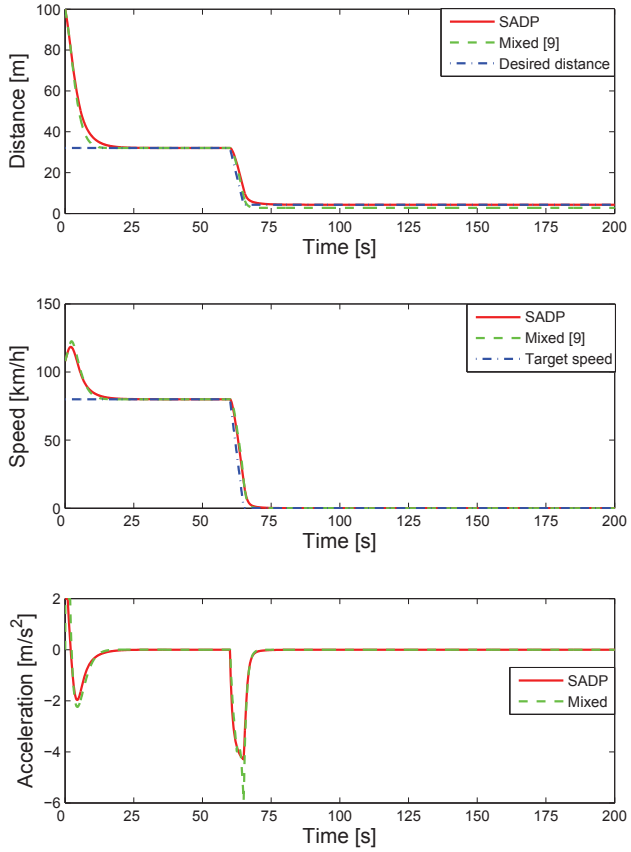


Figure 8: Experimental results with SADP and the mixed control strategies. The emergency braking scenario: (a) distance; (b) speed; (c) acceleration.

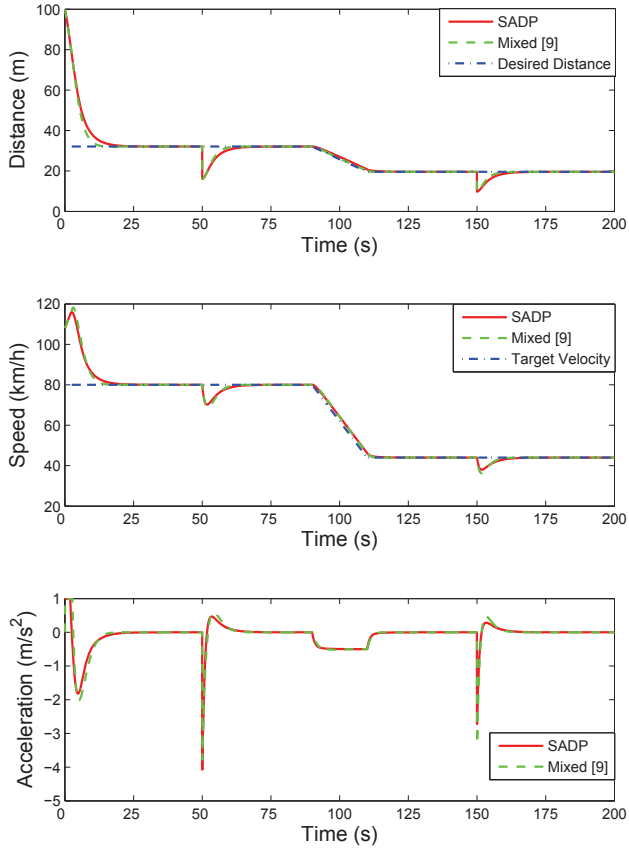


Figure 9: Experimental results with SADP and the mixed control strategies. The cut-in scenario: (a) distance; (b) speed; (c) acceleration.

control strategy: the controller performs well, especially in its distance control, when the driving habit changes.

There are two reasons for the good performance of the SADP control strategy:

(1) The training scenario only consists of changing the speed in time. However, due to the trial and error mechanism of SADP, the state space is exhaustively explored during the training process. As a result, most typical states are excited and used during the training phases.

(2) The state of SADP is $(\Delta v, \Delta d)$ and not $(\Delta v, d_r)$. As such, different driving habits will solely lead to different states of SADP, which means that the Action network will provide corresponding action strategies.

The Demonstration of stability for the obtained controller in a close form is not an easy task. However, as shown in section 3.5. We can estimate how the learning process is difficult. Such a complexity can be intended in terms of learnability, namely, the probability that given a training experiment the outcome controller is effective. By having considered 1000 experiments (i.e., we have generated $N = 1000$ controllers) we discover that only 1 out of 1000 does not provide the requested performances.

Following the derivation given in section 3.5 and the Chernoff's bound, let $\delta = 0.01$ and $\varepsilon = 0.05$, then $N \geq 1060$ is obtained. With $\hat{p_N} = 0.999$ obtained from the experiment results, we can state that the probability that our controller satisfies the performance criterion is above 0.95: the statement holds with confidence 0.99. In other terms, the learning process is particularly efficient. Since performance validation is carried out on a significant test set covering the functional driving conditions for our vehicle, stability is implicitly granted, at least for the considered conditions.

Future analysis might consider a double form of randomization where driving habits are also drawn randomly and provided to the vehicle so as to emulate its lifetime behavior.

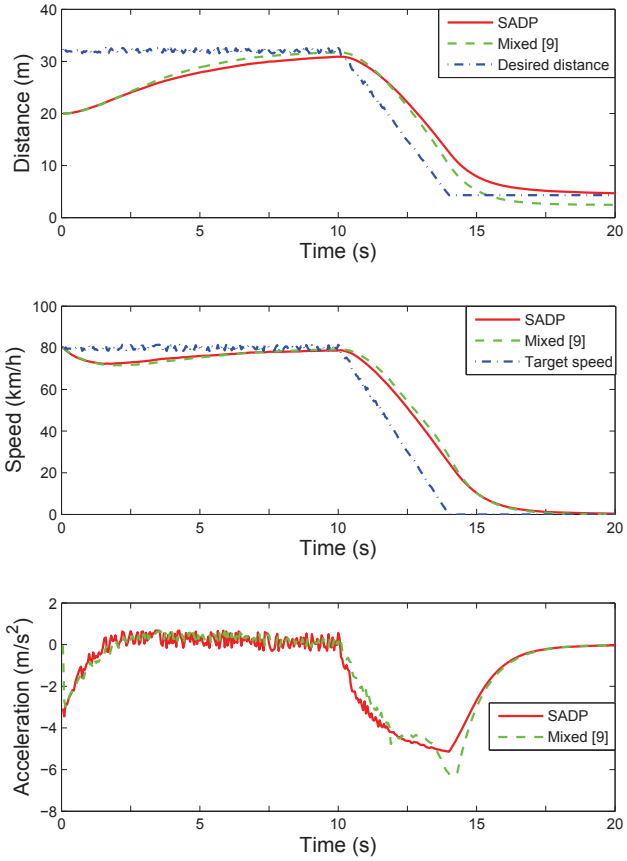


Figure 10: Robust experiments with SADP and the mixed control strategies in an emergency braking scenario (Moon et al., 2009): (a) distance; (b) speed; (c) acceleration.

5. Conclusions

The major contribution of this paper is the suggestion of a simple and effective learning control strategy for the full-range ACC problem. The control action is based on SADP and introduces the concept of Inducing Region to speed up the learning efficiency.

The trained SADP is applied to different driving scenarios including normal ACC, SG, emergency braking, cut-in and driver habits changing. The SADP control strategy performs well in all encountered scenarios. The method shows to be particularly effective in the emergency braking case.

We also show, by using randomized algorithms, how the proposed SADP is particularly effective to provide good control performance on our test scenarios at least with probability 0.95 and confidence 0.99.

6. Acknowledgments

We strongly acknowledge Prof. Derong Liu for valuable discussions and Mr. Yongsheng Su for the assistance with the experimental campaign.

References

[1] B. Siciliano, O. khatib, Springer Handbook of Robotics, Chapter 51 Intelligent Vehicles, Springer-Verlag Berlin Heidelberg, 2008, pp. 1175-1198.

[2] G.N. Bifulco, F. Simonelli, R.D. Pace, Experiments toward an human-like adaptive cruise control, Proc. IEEE Intelligent Vehicles Symposium, Eindhoven, 2008, pp.919-924.

[3] S.K. Park, J.P. Hwang, E. Kim, H.J. Kang,Vehicle tracking using a microwave radar for situation awareness, Control Engineering Practice, 18(4) (2010) 383-395.

[4] K. Yi, I. Moon, A driver-adaptive stop-and-go cruise control strategy, Proc. IEEE International Conference on Networking, Sensing and Control, 2004, pp. 601-606.

[5] A. Kesting, M. Treiber, M. Schonhof, D. Helbing, Adaptive cruise control design for active congestion avoidance, Transportation Research Part C, 16 (2008) 668-683.

[6] X.R. Bai, D.B. Zhao, J.Q. Yi, Coordinated control of multiple ramps metering based on ADHDP(λ) controller, International Journal of Innovative Computing, Information and Control, 5(10(B)) (2009) 3471-3481.

[7] A.G. Barto, T.G. Dietterich, Reinforcement learning and its relationship to supervised learning, In J. Si, A. Barto, W. Powell, D. Wunsch(Eds.), Handbook of Learning and Approximate Dynamic Programming, IEEE Press, John Wiley Sons, Inc. 2004, pp. 47-63.

[8] D.P. Bertsekas, J.N. Tsitsiklis, Neuro-Dynamic Programming, Belmont Massachusetts: Athena Scientific, 1996.

[9] M. Won, S.S. Kim, B.B. Kang, H.J. Jung, Test bed for vehicle longitudinal control using chassis dynamometer and virtual reality: an application to adaptive cruise control, KSME International Journal, 15(9) (2001) 1248-1256.

[10] Z.P. Xia, D.B. Zhao, Hybrid feedback control of vehicle longitudinal acceleration, Chinese Control Conference, Hefei, 2012, pp.7292-7297.

[11] P.S. Fancher, H. Peng, Z. Bareket, Comparative analyses of three types of headway control systems for heavy commercial vehicles, Vehicle System Dynamics, 25 (1996) 139-151.

[12] B.A. Guvenc, E. Kural, Adaptive cruise control simulator: a low-cost, multiple-driver-in-the-loop simulator, IEEE Control Systems Magazine, 26(3) (2006) 42-55.

[13] H.B. He, Z. Ni, J. Fu, A three-network architecture for on-line learning and optimization based on adaptive dynamic programming, Neurocomputing, 78(1) (2012) 3-13.

[14] Z.H. Hu, D.B. Zhao, Supervised reinforcement learning for adaptive cruise control, Proc. 4th International Symposium on Computational Intelligence and Industrial Application, 2010, pp. 239-248.

[15] T. Li, D.B. Zhao, J.Q. Yi, Adaptive dynamic neurofuzzy system for traffic signal control, Proc. IEEE International Joint Conference on Neural Networks, Hong Kong, 2008, 1841-1847.

[16] I. Moon, K. Yi, Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance, Control Engineering Practice, 17(4) (2009) 442-455.

[17] A.Y. Ng, D. Harada, S.J. Russell, Policy invariance under reward transformations: theory and application to reward shaping, Proc. Sixteenth International Conference on Machine Learning, 1999, pp. 278-287.

[18] H. Ohno, Analysis and modeling of human driving behaviors using adaptive cruise control, IECON 2000-26th Annual Conference of the IEEE-Industrial-Electronics-Society, 1-4, 2000, pp. 2803-2808.

[19] M.T. Rosenstein, A.G. Barto, Supervised actor-critic reinforcement learning, In J. Si, A. Barto, W. Powell, D. Wunsch (Eds.), Handbook of Learning and Approximate Dynamic Programming, IEEE Press, John Wiley Sons, Inc, 2004, pp. 359-380.

[20] J. Si, Y.T. Wang, On-line learning control by association and reinforcement, IEEE Transactions on Neural Networks, 12(2) (2001) 264-276.

[21] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, Cambridge MA: The MIT Press, 1998.

[22] C.C. Tsai, S.M. Hsieh, C.T. Chen, Fuzzy longitudinal controller design and experimentation for adaptive cruise control and stop go, Journal of Intelligent Robotic Systems, 59(2) (2010) 167-189.

[23] D. Wang, D.R. Liu, Q.L. Wei, Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach, Neurocomputing, 78(1) (2012) 14-22.

[24] C. Watkins, P. Dayan, Q-learning, Machine Learning, 8 (1992) 279-292.

[25] P.J. Werbos, Advanced forecasting methods for global crisis warning and models of intelligence, General Systems Yearbook, 38 (1997) 22-25.

[26] D.B. Zhao, J.Q. Yi, D.R. Liu, Particle swarm optimized adaptive dynamic programming, Proc. IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning, Honolulu, HI, 2007, pp. 32-37.

[27] D.B. Zhao, X.R. Bai, F.Y. Wang, J. Xu, W.S. Yu, DHP for coordinated freeway ramp metering, IEEE Transactions on Intelligent Transportation Systems, 12(4) (2011) 990-999.

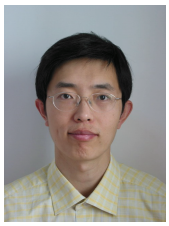
[28] D.B. Zhao, Z. Zhang, Y.J. Dai, Self-teaching adaptive dynamic programming for Go-Moku, Neurocomputing, 78(1) (2012) 23-29.

[29] P.J. Zheng, M. McDonald, Manual vs. adaptive cruise control - can driver's expectation be matched? Transportation Research Part C, 13(5-6) (2005) 421-431.

[30] R.Tempo, G.Calafiore, F. Dabbene. Randomized algorithms for analysis and control of uncertain systems, Springer, 2005.

[31] M.Vidyasagar. A Theory of Learning and Generalization, Springer, 1997.

[32] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations, Annals of Mathematical Statistics, 23(4) (1952) 493-507.



Dongbin Zhao (M'06, SM'10): received the B.S., M.S., Ph.D. degrees in Aug. 1994, Aug. 1996, and Apr. 2000 respectively, in materials processing engineering from Harbin Institute of Technology, China. Dr. Zhao was a postdoctoral fellow in humanoid robot at the Department of Mechanical Engineering, Tsinghua University, China, from May 2000 to Jan. 2002.

He is currently an associate professor at the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, China. He has published one book and over thirty international journal papers. His current research interests lies in the area of computational intelligence, adaptive dynamic programming, robotics, intelligent transportation systems, and process simulation.

Dr. Zhao is an Associate Editor of the IEEE Transactions on Neural Networks and Learning Systems, and Cognitive Computation.

Zhaohui Hu received the B.S. degree in mechanical engineering from the University of Science & Technology Beijing, Beijing, China and the M.S. degree in Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2008 and 2010, respectively. He is now with the Electric Power Research Institute of Guangdong Power Grid Corporation, Guangzhou, China. His main research interests include the area of computational intelligence, adaptive dynamic programming, power grids, and intelligent transportation systems.

Zhongpu Xia received the B.S. degree in automation control from China University of Geosciences, Wuhan, China in 2011. He is currently working toward the M.S. degree in the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include computational intelligence, adaptive dynamic programming and intelligent transportation systems.

Cesare Alippi received the degree in electronic engineering cum laude in 1990 and the PhD in 1995 from Politecnico di Milano, Italy. Currently, he is a Full Professor of information processing systems with the Politecnico di Milano. He has been a visiting researcher at UCL (UK), MIT (USA), ESPCI (F), CASIA (CN). Alippi is an IEEE Fellow, Vice-President education of the IEEE Computational Intelligence Society (CIS), Associate editor (AE) of the IEEE Computational Intelligence Magazine, past AE of the IEEE-Tran. Neural Networks, IEEE-Trans Instrumentation and Measurements (2003-09) and member and chair of other IEEE committees including the IEEE Rosenblatt award.

In 2004 he received the IEEE Instrumentation and Measurement Society Young Engineer Award; in 2011 has been awarded Knight of the Order of Merit of the Italian Republic.Current research activity addresses adaptation and learning in non-stationary environments and Intelligent embedded systems.

He holds 5 patents and has published about 200 papers in international journals and conference proceedings.

Yuanheng Zhu received the B.S. degree in school of management and engineering from Nanjing University, Nanjing, China, in July 2010. He is currently a PhD candidate at the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, China. His current research interests lies in the area of adaptive dynamic programming and fuzzy system.

Ding Wang received the B.S. degree in mathematics from Zhengzhou University of Light Industry, Zhengzhou, China, the M.S. degree in operational research and cybernetics from Northeastern University, Shenyang, China, and the Ph.D. degree in control theory and control engineering from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2007, 2009, and 2012, respectively. He is currently an assistant professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include adaptive dynamic programming, neural networks, and intelligent control.