基于柔性演员-评论家算法的自适应巡航控制研究*

赵克刚[1] 石翠铎[1] 梁志豪[1] 李梓棋[1] 王玉龙[2]

（1.华南理工大学，广州 510641；2.湖南大学，汽车车身先进设计制造国家重点实验室，长沙 410082）

【摘要】针对目前自适应巡航控制技术中，深度强化学习的控制算法环境适应能力不足、模型迁移性及泛化能力较差的问题，提出一种基于最大熵原理和随机离线策略的柔性演员-评论家（SAC）控制算法。构建演员和评论家网络拟合动作值函数和动作策略函数，并使用自调节温度系数改善智能体的环境探索能力；针对奖励稀疏问题，运用奖励塑造思想设计奖励函数；此外，提出一种新的经验回放机制以提高样本利用率。将所提出的控制算法在不同场景中进行仿真及实车验证，并与深度确定性策略梯度（DDPG）算法进行比较，结果表明，该算法具有更好的模型泛化能力和实车迁移效果。

主题词：自适应巡航控制 柔性演员-评论家 可迁移性 深度强化学习

中图分类号：U461 **文献标识码：**A **DOI:** 10.19620/j.cnki.1000-3703.20220500

Research on Adaptive Cruise Control Based on Soft Actor-Critic Algorithm

Zhao Kegang[1], Shi Cuiduo[1], Liang Zhihao[1], Li Ziqi[1], Wang Yulong[2]

（1. South China University of Technology, Guangzhou 510641; 2. State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, Changsha 410082）

【Abstract】For the problems of adaptive cruise control technology, including insufficient environmental adaptability of control algorithm for Deep Reinforcement Learning (DRL), poor model mitigation and generalization ability, this paper proposed the Soft Actor-Critic (SAC) control algorithm based on the principle of maximum entropy and stochastic off-line policy. SAC network was built to fit action value function and action policy function, and auto-adjusting temperature coefficient was used to improve the environmental exploration ability of intelligent agent. For the problem of sparse reward, the reward function was designed by using the idea of reward shaping. In addition, a new experience replay mechanism was proposed to improve the utilization rate of samples. The proposed control algorithm was simulated and tested in different scenes, and compared with Deep Deterministic Policy Gradient (DDPG). The results show that the algorithm has better model generalization ability and migration effect on real vehicles.

Key words: Adaptive cruise control, Soft Actor-Critic(SAC), Mitigation, Deep Reinforcement Learning(DRL)

【引用格式】赵克刚，石翠铎，梁志豪，等. 基于柔性演员-评论家算法的自适应巡航控制研究[J]. 汽车技术，2023(3): 26-34.

ZHAO K G, SHI C D, LIANG Z H, et al. Research on Adaptive Cruise Control Based on Soft Actor-Critic Algorithm[J]. Automobile Technology, 2023(3): 26-34.

1 前言

自适应巡航控制（Adaptive Cruise Control，ACC）是重要的自动驾驶辅助技术，而目前的ACC算法依赖于大量的标定工作，并且存在复杂环境下适应性差、表现不佳的问题[1-2]。深度强化学习（Deep Reinforcement Learning，DRL）通过智能体与环境交互进行自学习最大化累计奖励值，以学习到目标任务的最优策略[3-5]，在未来有望解决自动驾驶等复杂系统的控制决策问题，已经在路径规划[6-7]、轨迹跟踪[8-9]和跟驰控制[10-11]等自动驾驶领域得到了较为广泛的研究。针对ACC算法适应复杂工况能力差的弊端，DRL可以提供新的研究思路。

目前，在自动驾驶领域应用的DRL算法主要为无模型的确定性策略和随机性策略。在确定性策略算法研究中：Fu等[12]利用深度确定性策略梯度（Deep Deterministic Policy Gradient，DDPG）算法训练紧急制动

*基金项目：广东省重点领域研发项目（2019B090912001）。

决策策略，可提高安全性；Qian等[13]利用双延迟深度确定性策略梯度（Twin Delayed Deep Deterministic Policy Gradient，TD3）算法并考虑拓扑路径的特点训练自动驾驶决策策略，解决行为决策与轨迹规划的一致性问题。上述文献所使用的确定性策略算法在训练过程中虽然能够较快地收敛到稳定状态，但是环境探索不充分且可能得到局部最优策略的缺点，使得模型的迁移性和泛化能力较差。

针对确定性策略算法探索能力差的问题，随机性策略框架提供了更全面的环境探索。Liu等[14]使用异步优势演员-评论家（Asynchronous Advantage Actor-Critic，A3C）算法并考虑节能因素，提出一种自动驾驶决策策略；He等[15]采用近端策略优化（Proximal Policy Optimization，PPO）算法提出一种自动驾驶多目标纵向决策方法，并通过熵约束加快模型训练，提高算法的稳定性。以上文献采用的随机性策略算法探索能力更强，有更好的环境适应能力，但使用的是在线策略，对历史样本数据利用率低。

因此，本文提出一种基于柔性演员-评论家（Soft Actor-Critic，SAC）的ACC算法。建立车辆自适应巡航的马尔可夫决策过程，构建合理的演员和评论家网络并加入自调节温度系数，通过设计模块化奖励函数以及新的样本训练模式进一步优化算法。将所提出的控制算法在不同仿真环境和实车环境中进行测试，验证算法的有效性。

2 自适应巡航车辆数学模型

2.1 车辆跟随模型

本文以乘用车为研究对象，车辆自适应巡航场景如图1所示。

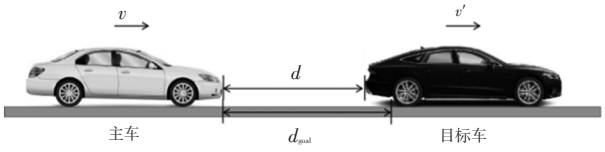


图1 车辆自适应巡航场景示意

主车跟随目标车行驶过程中，目标距离作为自适应巡航控制的重要指标，在保证行车安全和道路交通效率的同时还需兼顾驾驶员的心理预期。本文采用可变安全距离策略中的固定车头时距（Constant Time Headway，CTH）[16]作为目标距离的计算方法。车头时距τ_h定义为：

$$\tau_h=d/v \tag{1}$$

式中，d为主车与目标车的实际距离；v为主车速度。

将τ_h设置为固定值，则采用CTH计算的目标距离d_{goal}为：

$$d_{goal}=\tau_h v+d_0 \tag{2}$$

式中，d_0为目标车静止时与主车的最小安全距离。

2.2 马尔可夫决策过程

将主车作为DRL的智能体，其跟随目标车的行驶过程使用马尔可夫决策过程（Markov Decision Process，MDP）表示。MDP由4维数组$[\boldsymbol{S},\boldsymbol{A},P,R]$描述，其中，$\boldsymbol{S}$、$\boldsymbol{A}$分别为状态空间和动作空间，$P$为状态转移概率，$R$为奖励函数。本文将$t$时刻主车与目标车的实际距离与目标距离间的误差$\Delta d_t$，主车实际速度与目标速度（即目标车速度）的误差$\Delta v_t$作为状态输入，$t$时刻主车的目标速度$v_{goal}t$作为动作输出，定义$t$时刻的状态空间$\boldsymbol{s}_t$和动作空间$\boldsymbol{a}_t$为：

$$\begin{cases}\boldsymbol{s}_t=[\Delta d_t,\Delta v_t],\Delta d_t\in[\Delta D_L,\Delta D_H],\Delta v_t\in[\Delta V_{min},\Delta V_{max}]\\ \boldsymbol{a}_t=v_{goal}t,v_{goal}t\in[V_{min},V_{max}]\end{cases} \tag{3}$$

式中，ΔD_L、ΔD_H分别为距离误差的下限与上限；ΔV_{min}、ΔV_{max}分别为速度误差的最小值与最大值；V_{min}、V_{max}分别为目标速度的最小值与最大值。

t时刻自适应巡航的控制过程可以描述为：智能体接收到状态信息$\boldsymbol{s}_t$，执行DRL产生的动作$\boldsymbol{a}_t$，通过奖励函数R获得奖励值，并根据状态转移概率P将状态转移至$\boldsymbol{s}_{t+1}$。

3 自适应巡航的DRL控制算法

3.1 算法结构

本文在柔性Q学习（Soft Q-Learning，SQL）[17]基础上改进获得一种基于最大熵原理的DRL算法，其通过离线策略的方法优化一个随机性策略，在连续动作空间的复杂系统中具有较好的适用性。如图2所示为基于该算法的自适应巡航DRL过程，其学习目标是找到累计奖励与熵的和期望最大的策略π^*：

$$\pi^*=\arg\max_{\pi}E_{(s_t,a_t)\sim\pi}\left[\sum_t r(\boldsymbol{s}_t,\boldsymbol{a}_t)+\alpha H\left(\pi\left(\cdot|\boldsymbol{s}_t\right)\right)\right] \tag{4}$$

式中，$E_{(s_t,a_t)\sim\pi}$为期望；$r(\boldsymbol{s}_t,\boldsymbol{a}_t)$为$t$时刻ACC系统采取控制动作的奖励；$H(\pi(\cdot|\boldsymbol{s}_t))=-E_\alpha[\log(\pi(\boldsymbol{a}_t|\boldsymbol{s}_t))]$为在策略$\pi$下动作的熵；$\alpha$为温度系数，决定熵相对于奖励的权重。

本文使用深度神经网络拟合动作值函数和动作策略函数，分别组成评论家（Critic）和演员（Actor）网络。

3.1.1 评论家网络

对于动作值函数，使用2层隐藏层的全连接层网络对其进行拟合。在如图3所示的动作值网络中，以状态

s_t和动作a_t作为输入，线性整流函数（Rectified Linear Unit，ReLU）$f(x)=\max(0,x)$作为激活函数，a_t的估计值作为输出。

动作值网络的损失函数为动作值函数$Q_\theta(s_t,a_t)$和动作值目标函数$\bar{Q}_\theta(s_t,a_t)$的均方差：

$$J_Q(\theta)=E\left[\frac{1}{2}\left(Q_\theta(s_t,a_t)-\bar{Q}_\theta(s_t,a_t)\right)^2\right] \tag{5}$$

其中：

$$\bar{Q}_\theta(s_t,a_t)=r(s_t,a_t)+\gamma E_{(s_{t+1},a_{t+1})\sim\pi}\cdot\left(\bar{Q}_\theta(s_{t+1},a_{t+1})-\alpha\log\left(\pi\left(a_{t+1}|s_{t+1}\right)\right)\right) \tag{6}$$

式中，θ为动作值网络参数；γ为折扣因子。

具体训练中，使用双动作值网络并选取最小的$Q_\theta(s_t,a_t)$，以减少对动作值的高估。

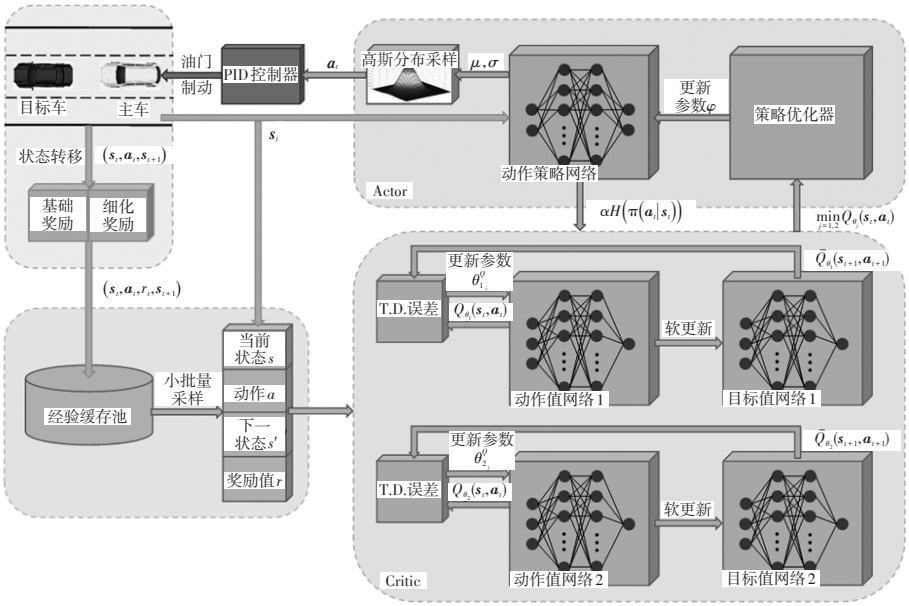


图2　自适应巡航控制DRL过程

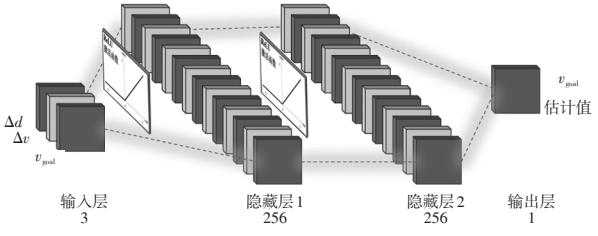


图3　动作值网络

3.1.2　演员网络

与动作值函数的拟合相同，动作策略函数同样使用2层隐藏层的全连接层网络进行拟合。如图4所示的动作策略网络中，以状态s_t作为输入，ReLU作为激活函数，高斯分布的均值μ和方差σ作为输出。但由于μ和σ采样动作并不可导，无法计算损失函数的梯度。因此，本文采用重参数的方法，将反向传播路径中的高斯分布用标准正态分布代替，从标准正态分布中获取采样值ε_t，从而获得对应均值和方差高斯分布的采样动作a_t：

$$a_t=\mu+\varepsilon_t\sigma \tag{7}$$

动作策略网络的损失函数为网络估计的高斯分布与实际基于能量分布的期望KL散度（Kullback-Leibler Divergense）：

$$J_\pi(\varphi)=E\left[D_{\mathrm{KL}}\left(\pi(\cdot|s_t)\left\|\frac{\exp\left(\frac{1}{\alpha}Q_\theta(s_t,\cdot)\right)}{Z_\theta(s_t)}\right.\right)\right] \tag{8}$$

式中，φ为动作策略网络参数；D_{KL}为KL散度；$Z_\theta(s_t)$为使分布正则化的配分函数。

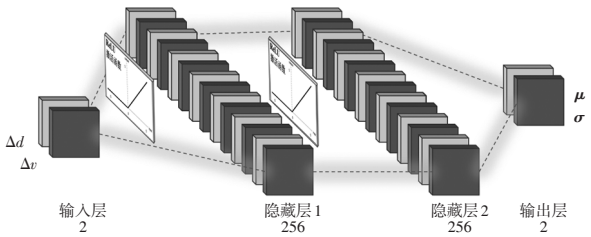


图4　动作策略网络

另外，高斯分布中采样得到的动作a_t的值域为$(-\infty,+\infty)$，但在自适应巡航场景中速度作为动作是有界的，因此需要对动作a_t进行变换。本文使用压缩高斯分布，将a_t用tanh激活函数处理，将其值域映射到$(-1,+1)$；然后进行换元计算，将激活值乘以自适应巡航限制的最高速度得到真实目标速度v_{goal}。

3.2　自调节温度系数

车辆在自适应巡航时，温度系数α作为熵的权重，

起到控制策略随机性的作用。α越大，则控制策略越随机，ACC系统对环境的探索越充分，即会尝试更多的动作。文献[18]使用依赖先验的固定α，但由于奖励值不断变化，采用固定α会导致训练不稳定，且容易收敛到局部最优。因此，本文设计自调节温度系数α，即当ACC系统探索到新的区域时，最优动作未知，将α调大鼓励探索更多动作空间，当某一区域探索比较充分时，最优动作基本确定，将α适当减小。

t时刻最优温度系数α_t^*为：

$$\alpha_t^* = \arg\min_{\alpha_t} E_{a_t \sim \pi_t^*}\left[-\alpha_t \log\left(\pi_t^*\left(\boldsymbol{a}_t|\boldsymbol{s}_t;\alpha_t\right)\right) - \alpha_t H\right] \tag{9}$$

式中，α_t为t时刻的温度系数；H为当前状态采用动作的熵。

温度系数的损失函数为：

$$J(\alpha) = E\left[-\alpha \log\left(\pi_t\left(\boldsymbol{a}_t|\boldsymbol{s}_t\right)\right) - \alpha H_0\right] \tag{10}$$

式中，H_0为熵的阈值。

3.3 奖励函数

奖励函数是DRL的重要部分，对训练效果有直接的影响，好的奖励函数可以引导智能体快速学习到有用的知识，加快训练效率、提高训练效果。本文设计的奖励函数分为基础奖励和细化奖励。

3.3.1 基础奖励

基础奖励的作用是为智能体确立学习的基础目标和具备的基本功能，其产生于训练的每一个时间步。考虑到自适应巡航汽车实现基本跟车功能的同时需兼顾乘坐舒适性，因此设计的基础奖励r_b由状态量与纵向加速度构成[19]：

$$r_b = -\left(\xi_1 \Delta d^2 + \xi_2 \Delta v^2 + \xi_3 a^2\right) \tag{11}$$

式中，a为纵向加速度；ξ_1、ξ_2、ξ_3为基础奖励各变量的权重系数，权重系数越大，训练过程中越重视该变量，本文自适应巡航DRL控制过程侧重于更快地缩小距离误差，因此确定ξ_1=8、ξ_2=2、ξ_3=1。

基础奖励的分布如图5所示。

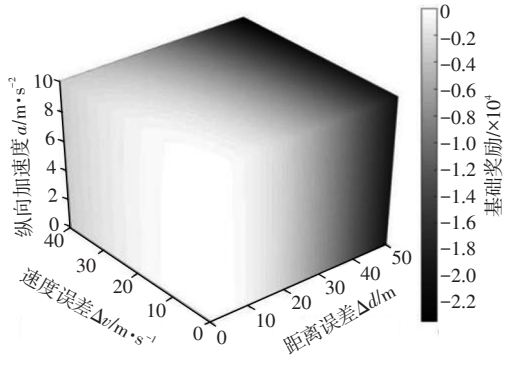


图5 基础奖励分布

3.3.2 细化奖励

为了解决奖励稀疏的问题，防止智能体学习缓慢甚至无法学习的情况出现，本文运用奖励塑造（Reward Shaping）的思想[20]设计细化奖励。如图6所示，其分为过程奖励、安全奖励和完成奖励。

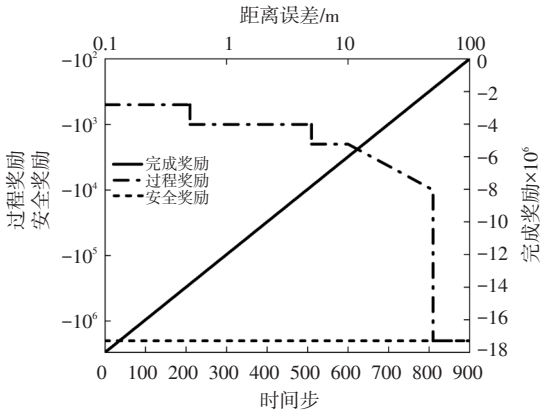


图6 细化奖励函数

过程奖励可以使智能体在训练过程中尽可能减小距离误差，其产生在训练的每一个时间步。距离误差越大，智能体获得的过程奖励越小；当距离误差超过50 m时，给予智能体最小的过程奖励并使训练终止，过程奖励r_p的表达式为：

$$r_p = \begin{cases} 0, |\Delta d| \leqslant 0.1\ \text{m} \\ -500, 0.1\ \text{m} < |\Delta d| \leqslant 0.5\ \text{m} \\ -1\,000, 0.5\ \text{m} < |\Delta d| \leqslant 5\ \text{m} \\ -2\,000, 5\ \text{m} < |\Delta d| \leqslant 10\ \text{m} \\ -200|\Delta d|, 10\ \text{m} < |\Delta d| \leqslant 50\ \text{m} \\ -2\,000\,000, 50\ \text{m} < |\Delta d| \end{cases} \tag{12}$$

安全奖励r_s的作用是防止主车与目标车发生碰撞，其产生于两车发生碰撞训练终止时：

$$r_s = -2\,000\,000 \tag{13}$$

完成奖励r_a的作用是促进智能体完整地执行自适应巡航任务。当主车走完一个回合（900个时间步）时，r_a=0，当主车与目标车碰撞或$|\Delta d|$>50 m导致训练终止时，有：

$$r_a = -20\,000 \times (900 - k) \tag{14}$$

式中，k为一个训练回合里已经走过的时间步。

3.3.3 奖励缩聚因子

在训练过程中，自适应巡航场景的环境随机因素较多，因此状态之间的差别较大，可能使得动作值网络参数变化剧烈，导致神经网络收敛变缓，甚至发散。为避免这一现象，本文在训练过程中对奖励赋予缩聚因子χ，使样本梯度的绝对数量级减小，从而使神经网络的训练更稳定。因此，集成的总奖励r为：

$$r = \chi(r_b + r_p + r_s + r_a) \tag{15}$$

3.4 经验回放机制

本文提出的算法使用离线策略，引入经验缓存池储

存智能体与环境交互产生的经验数据，从中随机抽取小批量样本用于训练动作值网络。经验缓存池保存数据的格式为四元数组$(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1},\boldsymbol{r}_t)$，即智能体与环境交互产生的当前所处的状态、当前产生的动作、下一时刻所处的状态和当前产生的奖励值。

传统的经验回放机制中，随着训练的进行，经验缓存池逐渐增大，但放入新数据的频次不变，导致缓存池中新数据的比例减少，使得训练的效果变差。针对这一问题，本文提出一种新的样本训练模式，即按照经验缓存池的大小改变模型的训练次数，随着新数据比例的下降，训练次数也逐渐增加。具体步骤为：

a. 定义经验缓存池最大容量$C_{\max}$。

b. 训练过程产生的经验数据逐组放入缓存池。

c. 当缓存池C的范围为$0.01C_{\max}\leqslant C<0.1C_{\max}$时，每增加100组新数据，训练智能体20次且每次小批量采样大小为$\bar{\omega}$的数据；当$0.1C_{\max}\leqslant C<C_{\max}$时，每增加100组新数据，训练智能体30次且每次小批量采样大小为$\bar{\omega}$的数据；当$C=C_{\max}$，即达到最大容量时，每进行100个时间步，训练智能体40次且每次小批量采样大小为$\bar{\omega}$的数据。

4 仿真与分析

4.1 训练场景设计

本文通过Python与LGSVL自动驾驶仿真器进行联合仿真训练。为了减小仿真训练得到的控制算法应用到实车中的误差，本文在训练中使用与实车上参数相近的激光雷达与轮速传感器获取状态信息。设计训练场景为τ_h=3 s、d_0=10 m[16]。仿真时，主车与目标车在同向两车道、每条车道宽为3.5 m的一条直道上行驶。主车与目标车都以10 m/s的初始速度行驶，两车的初始距离为10 m。目标车行驶工况如图7所示。

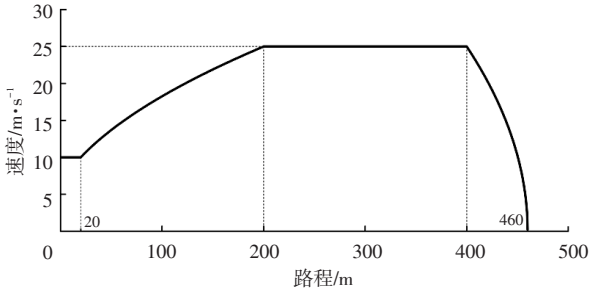


图7 目标车行驶工况

4.2 DRL模型训练

模型训练在Windows10操作系统下进行，内存为64 GB，处理器为Intel酷睿i7-8700K，显卡为NVIDIA GeForce RTX 2080 Ti，利用深度学习框架PyTorch。

本文算法的超参数对训练效果的影响：折扣因子γ用来计算累计奖励，γ越大越肯定以往的训练效果，γ越小越肯定当前回报；批量大小$\bar{\omega}$越大，训练的精度越高，但过多将使神经网络梯度变化减缓，从而无法走出局部最优点；神经网络学习率l_r过大，在梯度下降时网络参数变化过大，会使神经网络无法收敛，l_r过小，网络参数变化过于缓慢，会使神经网络学不到有效的知识。经多次训练试验，确定较优的算法超参数如表1所示。

表1 算法超参数

超参数	数值	超参数	数值
折扣因子γ	0.995	神经网络学习率l_r	0.000 1
软更新系数τ	0.02	初始温度系数α_0	0.2
批量大小$\bar{\omega}$	32	奖励缩聚因子χ	0.000 1
经验缓存池最大容量$C_{\max}$	100 000		

图8所示为训练累计奖励变化的结果，由图8可以看出，与采用相同奖励函数的基于DDPG的ACC算法相比，本文提出的基于SAC的控制算法获得的最高奖励相近，取得最高奖励的时间变长。主要原因为：本文算法采用随机性策略，在训练过程中探索更全面充分，可以学习到最优和次优策略，所以奖励曲线的波动较大、训练时间较长；DDPG算法采用确定性策略，对环境的探索不足，只能学习到最优策略，因此较快地达到最高奖励。

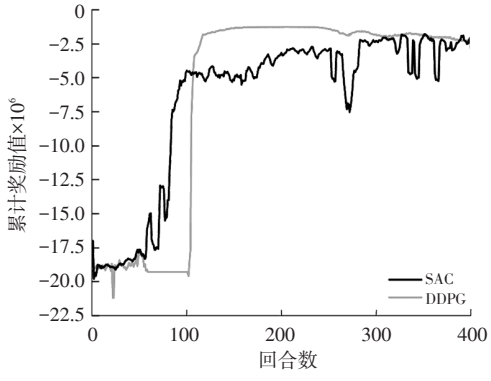


图8 DRL模型训练结果

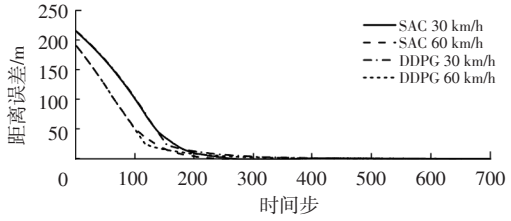
4.3 标准测试场景仿真验证

为验证本文算法的有效性和鲁棒性，选取自适应巡航标准测试场景中的目标车静止、低速和减速试验场景进行仿真验证。设定主车与目标车的初始距离为250 m，主车最高车速30 m/s，对每一个试验场景测试30回合并取数据平均值。规定标准测试场景下距离误差和速度误差分别收敛到0.8 m和0.3 m/s即认为达到自适应巡航稳定状态。

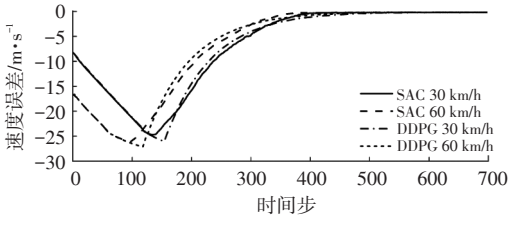
4.3.1 目标车静止场景

在目标车静止场景中，主车分别以初始速度为测试

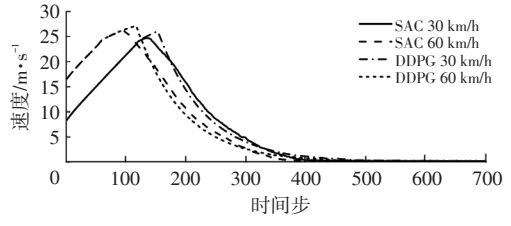
场景规定最低速30 km/h与最高速60 km/h的工况行驶。如图9所示为目标车静止场景下的主车状态曲线，从图9中可以看出，SAC和DDPG控制算法都采取了先加速缩短距离误差、后减速缩小速度误差的控制策略，以提高自适应巡航控制的效率，加快达到稳定状态。在主车初始速度为30 km/h的工况下，使用SAC和DDPG控制算法的最高速度分别为24.705 4 m/s和26.040 3 m/s；在主车初始速度为60 km/h的工况下，使用SAC和DDPG控制算法的最高速度分别为26.064 9 m/s和27.129 2 m/s。2种工况下SAC控制算法与DDPG相比最高速度分别降低了5.13%和3.92%，SAC控制算法在速度安全性上较DDPG有所提高。



(a)距离误差



(b)速度误差

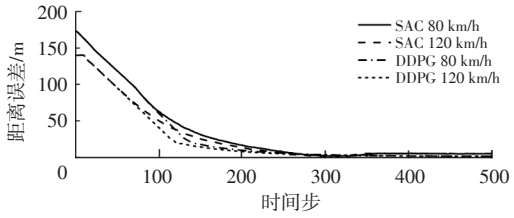


(c)速度

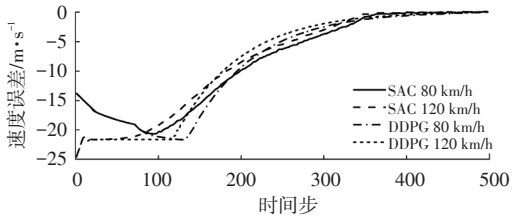
图9　目标车静止场景下主车的状态曲线

4.3.2　目标车低速场景

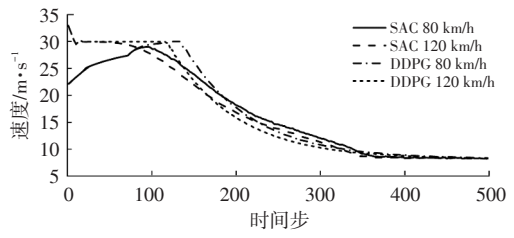
在目标车低速场景中，主车分别以初始速度为测试场景规定最低速80 km/h与最高速120 km/h的工况行驶，目标车以30 km/h的速度行驶。如图10所示为目标车低速场景下的主车状态曲线，从图10中可以看出，在主车初始速度为80 km/h的工况下，初始速度未超过最高车速30 m/s，因此SAC和DDPG控制算法都采取了先加速后减速的控制策略；在主车初始速度为120 km/h的工况下，初始速度超过最高车速，因此2种控制算法都先减速到最高车速，随后保持最高车速匀速行驶来加快减小距离误差，其中DDPG控制算法保持最高车速到第117个时间步后减速，SAC控制算法保持最高车速到第29个时间步后减速，后者保持最高车速的时间更短，具有更好的速度安全性。因此，该场景下SAC控制算法相比于DDPG采取的是更保守的控制策略。



(a)距离误差



(b)速度误差



(c)速度

图10　目标车低速场景下主车的状态曲线

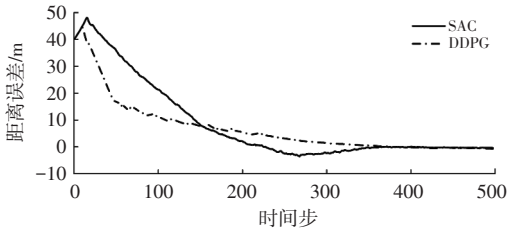
4.3.3　目标车减速场景

在目标车减速场景中，主车以120 km/h的初始速度行驶，目标车以70 km/h的初始速度、2 m/s²的减速度行驶。如图11a所示，SAC控制算法的距离误差最值较DDPG控制算法大，主要原因为SAC控制算法在跟车过程中倾向于增大距离误差来加快缩短速度误差，这与设置的奖励函数权重系数有关。图11b中速度误差曲线出现锯齿形状的原因为LGSVL仿真软件无法设置目标车匀减速行驶，因此人为对匀减速过程进行差分。如图11c所示，2种算法都采取减速的控制策略，由于主车初始速度高于最高车速，因此2种控制算法都在仿真初期短时间内将速度减小到最高车速，其中DDPG控制算法会保持最高车速行驶一段时间再减速到稳定状态，而SAC控制算法继续减速收敛到稳定状态。

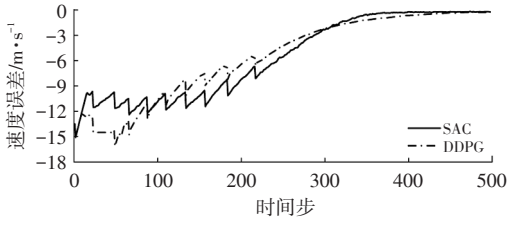
4.3.4　仿真结果总体分析

在3种场景的5个工况下，对于距离误差，2种算法收敛至稳定状态所用的时间基本相同；而对于速度

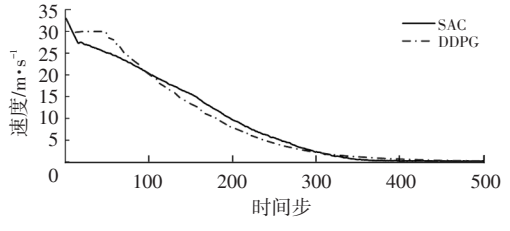
误差，如表2所示，SAC控制算法相比于DDPG收敛至稳定状态所用的时间分别减少了19.02%、22.32%、13.20%、16.97%、19.64%，明显提高了自适应巡航的控制效率。



(a)距离误差



(b)速度误差



(c)速度

图11　目标车减速场景下主车的状态曲线

表2　不同场景下速度误差收敛至稳定状态所用的时间步

场景	主车初速度/km·h⁻¹	时间步	
		SAC	DDPG
目标车静止	30	464	573
	60	421	542
目标车低速	80	388	447
	120	362	436
目标车减速	120	401	499

另外，冲击度（加速度对时间的一阶导数）反映了车辆在行驶过程中由于加、减速产生抖动颠簸的程度，可以作为乘坐舒适性的衡量指标。如表3所示，SAC控制算法的冲击度（绝对值）的均值较DDPG分别减小了25.20%、18.77%、23.92%、4.98%、46.22%，最值分别减小了57.09%、46.83%、75.20%、38.35%、57.07%，使主车行驶过程更加平稳，提高了乘员乘坐的舒适性。

从仿真结果可以看出，本文算法能够完成标准测试场景的仿真验证，与DDPG控制算法相比具有更好的速度安全性、控制效率和乘坐舒适性，对训练场景外的环境具有更好的泛化能力。

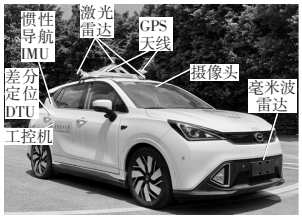
表3　不同场景下主车冲击度（绝对值）均值和最值

场景	主车初速度/km·h⁻¹	冲击度（绝对值）均值/m·s⁻³		冲击度（绝对值）最值/m·s⁻³	
		SAC	DDPG	SAC	DDPG
目标车静止	30	9.73	13.01	62.57	145.80
	60	10.59	13.04	73.87	138.92
目标车低速	80	12.00	15.77	48.45	195.31
	120	13.52	14.23	149.21	242.01
目标车减速	120	14.15	26.32	110.32	256.98

5　实车试验

为检验本文所提出的算法迁移到实车中的自适应巡航控制效果，如图12所示，在广州市番禺区内的一条长度为600 m的近似直道上进行相关测试。如图13所示，试验场景包括主车和目标车2辆车。试验主车由某品牌纯电动汽车线控改装获得，搭载激光雷达检测目标车的相对距离和速度，并利用轮速传感器获取自身速度信息。

图12　试验路线



(a)试验主车

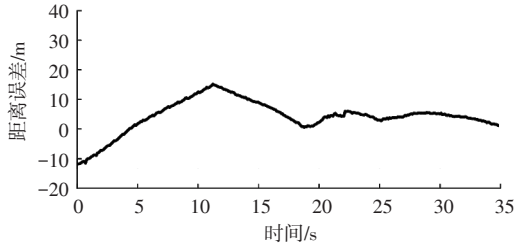
(b)试验现场

图13　试验主车和场景示意

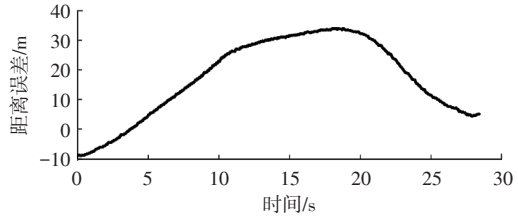
试验过程中，为提高场景的真实性及复杂度，主车与目标车的初始距离随机确定。具体工况为：主车与目标车均从静止起步，目标车按照驾驶员的驾驶习惯从静止加速到40 km/h，并在该速度附近沿道路行驶。

图14所示为实车试验主车状态曲线，从图14a、图14b中可以看出，使用SAC控制算法的距离误差在经过起步阶段的超调后可以迅速缩小到5 m内，而使用DDPG控制算法的距离误差超调时间较长，且超调时最大误差超过30 m。由图14c、图14d可以看出，使用DDPG控制算法时速度误差波动较大，幅值超过6 m/s，而使用SAC控制算法可以保持更小的速度误差。图14g、图14h中，使用SAC控制算法的冲击度（绝对值）明

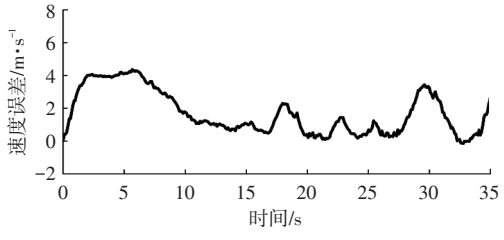
显小于DDPG，在乘车体验上更为舒适。在实车试验中，虽然存在信息传递延迟和丢包、激光雷达误检测等情况，但SAC控制算法并没有使主车出现状态急变，能够稳定地跟随目标车行驶，在实车上有较好的表现效果。



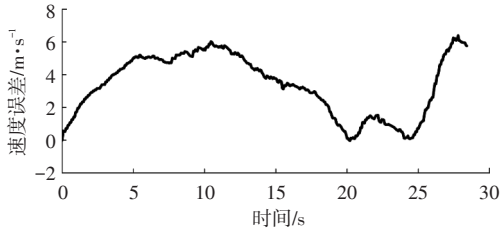
(a)SAC-距离误差



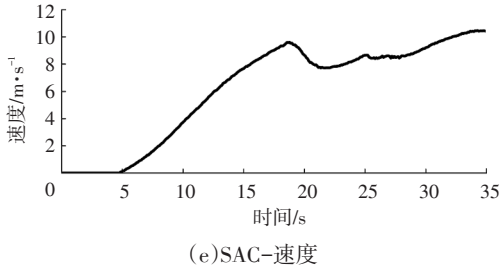
(b)DDPG-距离误差



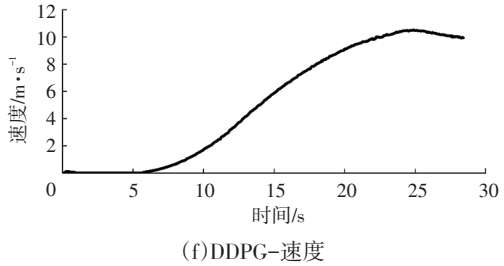
(c)SAC-速度误差



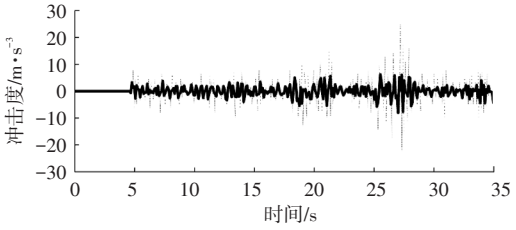
(d)DDPG-速度误差



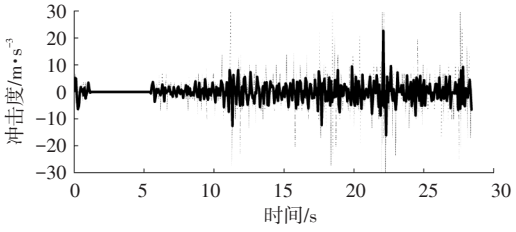
(e)SAC-速度



(f)DDPG-速度



(g)SAC-冲击度



(h)DDPG-冲击度

图14　试验主车状态曲线

6　结束语

本文研究了DRL算法在自适应巡航控制技术中的应用问题，提出一种基于SAC的控制算法。将车辆自适应巡航的学习过程描述为马尔可夫决策过程，构建评论家和演员网络拟合动作值函数和动作策略函数；使用自调节温度系数改善智能体的探索效果；构建模块化奖励函数，改善了智能体的学习效率和效果，解决了奖励稀疏的问题；提出一种新的样本训练模式，按照经验缓存池的大小改变模型的训练次数，可以进一步提高样本的利用率。

试验结果表明，相比于DDPG控制算法，本文所提出的算法能以更高的控制效率缩小速度误差，且冲击度（绝对值）更小，舒适性和安全性更好，在实车上的迁移性较好。在后续的研究中，将开展实车高速试验进一步检验算法的实车性能，并考虑路径跟踪、横纵向协同控制等自动驾驶场景的DRL控制问题。

参　考　文　献

[1] GUANETTI J, KIM Y, BORRELLI F. Control of Connected and Automated Vehicles: State of the Art and Future Challenges[J]. Annual Reviews in Control, 2018, 45: 18-40.

[2] 吴光强，张亮修，刘兆勇，等. 汽车自适应巡航控制系统研究现状与发展趋势[J]. 同济大学学报（自然科学版），2017, 45(4): 544-553.
WU G Q, ZHANG L X, LIU Z Y, et al. Research Status and Development Trend of Vehicle Adaptive Cruise Control Systems[J]. Journal of Tongji University (Natural Science), 2017, 45(4): 544-553.

[3] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-Level Control Through Deep Reinforcement Learning[J].

Nature, 2015, 518: 529-533.

[4] 赵冬斌, 邵坤, 朱圆恒, 等. 深度强化学习综述:兼论计算机围棋的发展[J]. 控制理论与应用, 2016, 33(6): 701-717.
ZHAO D B, SHAO K, ZHU Y H, et al. Review of Deep Reinforcement Learning and Discussions on the Development of Computer Go[J]. Control Theory & Applications, 2016, 33(6): 701-717.

[5] 刘全, 翟建伟, 章宗长, 等. 深度强化学习综述[J]. 计算机学报, 2018, 41(1): 1-27.
LIU Q, ZHAI J W, ZHANG Z Z, et al. A Survey on Deep Reinforcement Learning[J]. Chinese Journal of Computers, 2018, 41(1): 1-27.

[6] YOU C X, LU J B, FILEV D, et al. Advanced Planning for Autonomous Vehicles Using Reinforcement Learning and Deep Inverse Reinforcement Learning[J]. Robotics and Autonomous Systems, 2018, 114: 1-18.

[7] ZHOU X Y, WU P, ZHANG H F, et al. Learn to Navigate: Cooperative Path Planning for Unmanned Surface Vehicles Using Deep Reinforcement Learning[J]. IEEE Access, 2019, 7: 165262-165278.

[8] WASALA A, BYRNE D, MIESBAUER P, et al. Trajectory Based Lateral Control: A Reinforcement Learning Case Study [J]. Engineering Applications of Artificial Intelligence, 2020, 94.

[9] FEHÉR Á, ARADI S, BÉCSI T, et al. Proving Ground Test of a DDPG- Based Vehicle Trajectory Planner[C]// 2020 European Control Conference (ECC). St. Petersburg: IEEE, 2020: 332-337.

[10] LI G Q, GÖRGES D. Ecological Adaptive Cruise Control for Vehicles with Step- Gear Transmission Based on Reinforcement Learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 21(11): 4895-4905.

[11] GAO W N, GAO J Q, OZBAY K, et al. Reinforcement-Learning- Based Cooperative Adaptive Cruise Control of Buses in the Lincoln Tunnel Corridor with Time-Varying Topology[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(10): 3796-3805.

[12] FU Y C, LI C L, YU R, et al. A Decision-Making Strategy for Vehicle Autonomous Braking in Emergency via Deep Reinforcement Learning[J]. IEEE Transactions on Vehicular Technology, 2020, 69(6): 5876-5888.

[13] QIAN L L, XU X, ZENG Y J, et al. Deep, Consistent Behavioral Decision Making with Planning Features for Autonomous Vehicles[J]. Electronics, 2019, 8(12).

[14] LIU J, ZHAO L, ZHENG K, et al. A Distributed Driving Decision Scheme Based on Reinforcement Learning for Autonomous Driving Vehicles[C]// 2020 IEEE 91st Vehicular Technology Conference (VTC2020- Spring). Antwerp: IEEE, 2020: 1-5.

[15] HE X K, FEI C, LIU Y L, et al. Multi- Objective Longitudinal Decision- Making for Autonomous Electric Vehicle: A Entropy- Constrained Reinforcement Learning Approach[C]// 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC). Rhodes: IEEE, 2020: 1-6.

[16] 刘骝. 基于多目标决策算法和PID控制的CACC系统的优化与仿真[D]. 长春: 吉林大学, 2016.
LIU L. Optimization and Simulation of CACC System Based on Multi Objective Decision Algorithm and PID Control[D]. Changchun: Jilin University, 2016.

[17] HAARNOJA T, TANG H, ABBEEL P, et al. Reinforcement Learning with Deep Energy- Based Policies[C]// International Conference on Machine Learning. Sydney: PMLR, 2017: 1352-1361.

[18] HAARNOJA T, ZHOU A, ABBEEL P, et al. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor[C]// International Conference on Machine Learning. PMLR, 2018: 1861-1870.

[19] 李升波, 王建强, 李克强, 等. MPC实用化问题处理及在车辆ACC中的应用[J]. 清华大学学报(自然科学版), 2010, 50(5): 645-648.
LI S B, WANG J Q, LI K Q, et al. Processing of MPC Practical Problems and Its Application to Vehicular Adaptive Cruise Control Systems[J]. Journal of Tsinghua University (Science and Technology), 2010, 50(5): 645-648.

[20] NG A Y, HARADA D, RUSSELL S. Policy Invariance under Reward Transformations: Theory and Application to Reward Shaping[C]// Proceedings of the Sixteenth International Conference on Machine Learning. ICML, 1999: 278-287.

（责任编辑　斛　畔）

修改稿收到日期为2022年7月15日。