控制与决策
Control and Decision

基于强化学习的多目标车辆跟随决策算法

邓小豪， 侯进， 谭光鸿， 万斌杨， 曹婷婷

引用本文:
邓小豪, 侯进, 谭光鸿, 等. 基于强化学习的多目标车辆跟随决策算法[J]. 控制与决策, 2021, 36(10): 2497−2503.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2020.0426

基于强化学习的多目标车辆跟随决策算法

邓小豪, 侯 进†, 谭光鸿, 万斌杨, 曹婷婷

(西南交通大学 信息科学与技术学院，成都 611756)

摘 要: 为满足自适应巡航系统跟车模式下的舒适性需求并兼顾车辆安全性和行车效率,解决已有算法泛化性和舒适性差的问题,基于深度确定性策略梯度算法(deep deterministic policy gradient, DDPG),提出一种新的多目标车辆跟随决策算法. 根据跟随车辆与领航车辆的相互纵向运动学特性,建立车辆跟随过程的马尔可夫决策过程(Markov decision process, MDP)模型. 结合最小安全距离模型,设计一个高效、舒适、安全的车辆跟随决策算法. 为提高模型收敛速度,改进了DDPG算法经验样本的存储方式和抽取策略,根据经验样本重要性的不同,对样本进行分类存储和抽取. 针对跟车过程的多目标结构,对奖赏函数进行模块化设计. 最后,在仿真环境下进行测试,当测试环境和训练环境不同时,依然能顺利完成跟随任务,且性能优于已有跟随算法.

关键词: 自主决策；车辆跟随；半自动驾驶；强化学习；深度确定性策略梯度；马尔可夫决策过程

中图分类号: TP273 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2020.0426 **开放科学(资源服务)标识码(OSID):**

引用格式: 邓小豪, 侯进, 谭光鸿, 等. 基于强化学习的多目标车辆跟随决策算法[J]. 控制与决策, 2021, 36(10): 2497-2503.

Multi-objective vehicle following decision algorithm based on reinforcement learning

DEND Xiao-hao, HOU Jin†, TAN Guang-hong, WAN Bin-yang, CAO Ting-ting

(School of Information Science and Technology，Southwest Jiaotong University，Chengdu 611756，China)

Abstract: To meet the comfort requirements of the adaptive cruise system following mode and take into account vehicle safety and driving efficiency, and solve the problem of poor generalization and comfort of existing algorithms, a new multi-target vehicle following decision is proposed based on the deep deterministic policy gradient (DDPG). According to the mutual longitudinal kinematics of the following vehicle and the pilot vehicle, a Markov decision process (MDP) model of the vehicle following process is established. Combined with the minimum safety distance model, an efficient, comfortable and safe vehicle following decision algorithm is designed. In order to improve the model convergence speed, the storage method and extraction strategy of the DDPG algorithm's experience samples are improved, and the samples are classified and stored according to the importance of the experience samples. Aiming at the multi-objective structure of the following process, the reward function is modularized. Finally, the test is performed in the simulation environment. When the test environment and the training environment are different, the following tasks can be successfully completed, and the performance is better than the existing following algorithms.

Keywords: autonomous decision; vehicle following; semi-autonomous driving; reinforcement learning; deep deterministic policy gradient; Markov decision process

0 引 言

车辆跟随是常见的行车场景,好的决策算法有助于提高驾驶安全性、舒适性和道路通行能力. 目前,常见的车辆跟随决策算法包括基于规则的算法和基于有监督学习的算法[1]. 前者主要基于一系列车辆跟随模型,结合车辆运动学特性搭建相应的速度跟随控制器. 但由于开发人员无法枚举出车辆跟随过程中可能出现的所有情况,算法的灵活性和泛化性受到限制[2]. 如文献[3]提出的基于模型预测控制(model predictive control, MPC)原理的自适应多目标鲁棒控制算法,通过采用向量约束管理法解决了模型预测控制算法硬约束造成的控制系统无优化解的问

收稿日期: 2020-04-15；**修回日期:** 2020-06-08.

基金项目: 浙江大学CAD&CG国家重点实验室开放课题(A1923)；成都市科技项目(2015-HM01-00050-SF).

责任编委: 陈虹.

†通讯作者. E-mail: jhou@swjtu.edu.cn.

题. 但在前车频繁改变速度的复杂场景下,难以同时实现高效、安全、舒适地跟随. 文献[4]提出的基于时变间距和相对角度的跟随控制算法,使用反馈控制法设计了跟随车辆速度控制器,跟随车辆能高效安全地跟随,但无法保证舒适性. 后者通常依赖于人类演示提供的数据来近似车辆跟随状态与加速度之间的关系. 文献[5]提出了一种基于深度神经网络(deep neural network, DNN)的车辆跟随模型,并使用NGSIM (next generation simulation)数据集训练该车辆跟随模型,相较于传统的模型,在复杂情景下,行车效率得到提高,但受限于数据集的不完备,舒适性较差. 基于深度神经网络的车辆跟随方法是数据驱动的,极大减少了开发人员对策略的干扰,但其本质依然是在模拟人类的驾驶行为,而非从行车效率、安全性和舒适性上进行优化,难以得到最优的驾驶策略.

强化学习(reinforcement learning, RL)是一类重要的机器学习方法,在自主决策领域有着广泛的应用. 与深度学习一样,强化学习可以极大地减少开发人员对模型的干扰;与深度学习不同,强化学习通过控制agent与环境交互,在不断地试错中,学习到最优决策策略,而非模仿人类行为[6]. 标准的agent包括状态感知器、学习器和动作选择器3个部分. 状态感知器把环境状态映射成agent内部感知;动作选择器根据当前策略选择动作并作用于环境;学习器根据环境状态的奖赏值以及内部感知更新agent的策略知识. 强化学习的目标是求解模型的最优策略π^*,使得在遵循最优策略π^*下,能获得最大累计奖赏值G_t,即

$$G_t=\sum_{t=t_0}^{T}\gamma^{t-t_0}R_t,\ 0<\gamma\leqslant 1. \tag{1}$$

其中:γ为折扣因子,R_t为agent从环境状态s_t转移到s_{t+1}所接受到的奖赏值.

文献[7]使用Q learning算法实现车辆跟随的自主决策,通过“车性化”地设计模型的状态集和动作集以及“人性化”地设计模型的奖赏函数,模型经短时间训练后,在跟车效率、安全性和舒适性上都取得了不错的效果. 但由于Q learning算法使用离散的状态集和动作集,无法准确描述车辆所处状态,难以精准地做出决策,在前车频繁改变速度的复杂场景下,跟随效果不佳.

与深度学习结合后的深度强化学习(deep reinforcement learning, DRL)广泛应用于无人驾驶领域. 文献[8]提出了一种深度Q网络(deep Q network, DQN)算法,它使用端到端强化学习直接从高维感官输入学习决策策略. 文献[9]将DQN用于自动驾驶中,通过将汽车输出动作离散化,实现了简单环境下的自动驾驶,汽车可以准确绕过静态障碍物. 文献[10]提出了一种深度确定性策略梯度(deep deterministic policy gradient, DDPG)算法,这是一种适用于连续状态空间和连续动作空间的演员-评论家(actor-critic)算法. DDPG算法在解决具有连续输入状态和连续输出动作的问题时,具有出色的表现. 文献[11]将DDPG算法用于结构化道路上的无人驾驶车辆控制,经过短时间训练后,车辆可以不偏离车道线而行驶较远的距离. 文献[12]将DDPG算法用于无人驾驶避障问题研究,取得了不错的效果. 文献[13]提出了近端策略优化(proximal policy optimization, PPO)算法,可同时用于离散控制和连续控制,在OpenAI Five上取得了巨大成功. 但是PPO算法是一种在线策略算法,需要巨量的采样才能学习,这对于无人驾驶来说是难以接受的. 文献[14]提出了一种柔性演员-评论家(soft actor-critic, SAC)算法,这是一种面向最大熵强化学习的离线策略算法. SAC使用的是随机策略,与DDPG的确定策略相比,具有更强的探索能力,但是收敛速度较慢.

本文首先建立跟车过程的马尔可夫决策过程(Markov decision process, MDP)模型;然后用DDPG算法结合最小安全距离模型求得最优策略;为了加快模型的收敛速度,对DDPG算法的经验回放机制进行了改进;设计了一个兼顾效率、舒适性和安全性的奖赏函数;最后,在两种场景下对模型的性能进行测试,车辆可在不同场景下安全、高效、舒适地完成跟车任务,且性能优于已有方法.

本文的主要贡献如下:1) 结合DDPG算法和最小安全距离模型,提出一种新的车辆跟随决策方法;2) 对DDPG算法的经验回放机制进行改进,提升了模型的收敛速度;3) 针对多目标强化学习任务奖赏函数设计困难的问题,提出一种模块化的奖赏函数设计方法.

1 跟车行为的MDP建模

MDP是序贯决策的经典形式,通常用来对强化学习问题建模[15]. MDP由五元组(S, A, P, R, γ)描述. 其中:A为动作集;S为状态集;$P:\ S\times A\times S\to(0,1)$为状态转移概率;$R$为奖赏函数;$\gamma\in(0,1)$为折扣因子,用来计算累计奖赏.

在跟车过程中,将时间作离散化处理,采样率为T_s,跟随车辆为agent,与环境进行交互. 跟随车辆与领航车辆之间的距离d_t、跟随车辆的加速度$a_{2,t}$以及两车的相对速度Δv_t,为时刻t的状态s_t,$s_t(d_t,$

$a_{2,t}, \Delta v_t) \in S$. 根据跟随车辆与领航车辆的相互纵向运动学特性,可以得到如下关系式:

$$\begin{cases} d_{t+1} = d_t + \Delta v_t T_s + 0.5 a_{1,t} T_s^2 - 0.5 a_{2,t} T_s^2, \\ \Delta v_{t+1} = \Delta v_t + a_{1,t} T_s - a_{2,t} T_s, \\ a_{2,t+1} = \left(1 - \dfrac{T_s}{\tau}\right) a_{2,t} + \dfrac{T_s}{\tau} \mu_t. \end{cases} \tag{2}$$

其中: T_s 为采样率; $a_{1,t}$ 为领航车辆加速度; $a_{2,t}$ 为跟随车辆加速度; τ 为跟车过程中采用一阶惯性环节作为理想下位控制对象的时间常数; μ_t 为跟随车辆的输出动作,即当前时刻的期望加速度值.

agent在 t 时刻接收到状态信息后,输出动作 $\mu_t \in A$,产生的 t 时刻的奖赏值 $R_t = f(s_t)$,状态变为 s_{t+1}. agent输出的动作 μ_t 由策略 π 决定,策略 π 为状态 s_t 映射到每个动作的概率: $S \to P(A)$. 跟车模型的状态集 S 如表1所示,动作集 A 如表2所示.

表1　状态集

名称	范围
车间距 d/m	$[0 \sim d_{\max}]$
相对速度 Δv/(m·s^{-1})	$[v_{\min} \sim v_{\max}]$
加速度 a_2/(m·s^{-2})	$[a_{\min} \sim a_{\max}]$

表1中: $d_{\max}$ 为传感器的最大有效探测范围,当 $d > d_{\max}$ 时,输入 $d = d_{\max}$; $v_{\max}$ 为跟随车辆最大速度与领航车辆最大速度之和; $v_{\min} = -v_{\max}$; $a_{\min}$ 为跟随车辆的最大减速度; $a_{\max}$ 为跟随车辆的最大加速度.

表2　动作集

名称	范围
期望加速度 μ/(m·s^{-2})	$[a_{\min} \sim a_{\max}]$

表2中: $a_{\min}$ 为跟随车辆的最大减速度, $a_{\max}$ 为跟随车辆的最大加速度.

2　改进的DDPG算法

DDPG算法是一种演员-评论家(actor-critic)算法,本节将从评论家、演员、奖赏函数、经验回放机制这4个方面进行介绍.

2.1　评论家

评论家用来拟合状态动作值函数,包含目标Q网络和在线Q网络,两个网络交替更新. 两者初始参数 $\theta^{Q'}$ 与 θ^{Q} 相等,从经验缓冲池中取得经验样本 (s_i, a_i, r_i, s_{i+1}) 后,通过最小化损失值 L 来更新在线Q网络. L 的计算方法如下:

$$L = \frac{1}{N} \sum_i \left(y_i - Q(s_i, a_i|\theta^Q)\right)^2. \tag{3}$$

其中: N 为单次训练样本个数, Q 为状态动作值函数, θ^Q 为在线Q网络参数, y_i 的计算方式为

$$y_i = r_i + \gamma Q'(s_{i+1}, u'(s_{i+1}|\theta^{u'})|\theta^{Q'}). \tag{4}$$

这里: r_i 为单步奖赏值, γ 为折扣因子, Q' 为目标状态动作值函数, u' 为目标策略函数, $\theta^{u'}$ 为目标策略网络参数, $\theta^{Q'}$ 为目标Q网络参数. 不同于在线Q网络的实时更新,目标Q网络每隔一段时间更新一次,其更新方法为

$$\theta^{Q'} \leftarrow \alpha\theta^Q + (1-\alpha)\theta^{Q'}, \tag{5}$$

其中 α 为软更新率.

2.2　演　员

演员用来拟合策略函数,其主要任务是针对当前状态 s_t,输出动作 μ_t,包含目标策略网络和在线策略网络,两个网络交替更新,初始参数 θ^u 与 $\theta^{u'}$ 相等. 在线策略网络参数的更新如下式所示:

$$\nabla_{\theta^u} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=u(s_i)} \nabla_{\theta^u} u(s|\theta^u)|_{s_i}, \tag{6}$$

其中 θ^u 为在线策略网络参数. 目标策略网络的更新方式与目标Q网络一样,每隔一段时间更新一次,方法如下式所示:

$$\theta^{u'} \leftarrow \alpha\theta^u + (1-\alpha)\theta^{u'}. \tag{7}$$

因为DDPG中在线策略网络将状态映射到确定动作,所以存在着缺乏探索性的问题. 为了解决这一问题,一些学者在提高agent探索能力上做了许多研究[16-17]. 本文结合车辆跟随过程的实际情况,制定新的探索方案. 跟随车辆在做出加速或减速动作时,在其输出的期望加速度上添加了噪声 N_t, $N_t \sim N(0, \sigma^2)$. 在训练前期,演员的策略较差,较大的 σ^2 有利于得到更多样的经验样本;在训练后期,演员的策略较好,较小的 σ^2 有利于得到更优质的经验样本. 经验缓冲池装满前, $\sigma^2 = 4$;开始训练后, σ^2 的取值为

$$\sigma_{t+1}^2 = \max(\sigma_t^2 \times 0.999\,9, 0.1). \tag{8}$$

其中: σ_{t+1}^2 为当前输出动作噪声方差, σ_t^2 为上一时刻输出动作噪声方差.

2.3　奖赏函数

奖赏函数在强化学习任务中有着重要的作用,它给演员和评论家的网络参数更新指明了方向[18]. 本文中算法的奖赏函数采用模块化设计,与最小安全距离模型[19]相结合,包含了车辆跟随的安全、舒适和效

率3个方面.

2.3.1 安 全

现实世界中,驾驶员的真实跟车行为体现在对两车距离的控制上,通过调节自身速度来达到理想的跟车距离. 最小安全距离模型旨在计算各个时间点两车不发生追尾的最小安全距离,要计算最小安全距离,应首先计算领航车辆以最大制动减速度进行减速时的刹停距离,即

$$D_1 = \frac{v_1^2}{2a_{1-}}. \tag{9}$$

其中: v_1 为领航车辆的速度, a_{1-} 为领航车辆的最大制动减速度. 然后,计算跟随车辆以最大制动减速度进行减速时的刹停距离 D_2,并考虑车辆在反应时间 t_0 内的行驶距离(假设车辆在 t_0 内匀速行驶,反应时间主要包括传感器的时延),即

$$D_2 = v_2 t_0 + \frac{{v_2}^2}{2a_{2-}}. \tag{10}$$

其中: v_2 为跟随车辆速度, a_{2-} 为跟随车辆最大制动减速度. 最终,根据两车的刹停距离,并考虑刹停后缓冲距离 d_0,可得出车辆跟随过程中的最小安全距离 D^*,有

$$D^* = D_2 - D_1 + d_0. \tag{11}$$

从而可得到安全性方面的奖赏函数 R_1 为

$$R_1 = \begin{cases} -10, \ d < D^*; \\ 0, \ d \geqslant D^*. \end{cases} \tag{12}$$

2.3.2 效 率

在保证安全的前提下,车间距越小,道路利用率越高. 可得到效率方面的奖赏函数 R_2 为

$$R_2 = \begin{cases} \dfrac{10}{d}, \ d \geqslant D^*; \\ 0, \ d < D^*. \end{cases} \tag{13}$$

2.3.3 舒 适

用jerk表示加速度的变化率(单位为 $\mathrm{m/s^3}$),其定义为

$$\text{jerk} = \frac{a_{t+1} - a_t}{\Delta t}, \tag{14}$$

其中 Δt 取值为0.1.

用加速度大小和jerk的大小来衡量模型的舒适性,加速度和jerk越趋近于0,舒适性越好. 可得到舒适性方面的奖赏函数 R_3 为

$$R_3 = \frac{1}{1+|a|} + \frac{1}{1+|\Delta a|}. \tag{15}$$

2.3.4 奖赏函数集成

效率、安全和舒适是3个相互矛盾的因素,追求极致的效率,势必会舍弃安全和舒适,反之亦然. 因此,奖赏函数的表达式为

$$R = \omega_1 R_1 + \omega_2 R_2 + \omega_3 R_3, \tag{16}$$

其中 ω_1、ω_2、ω_3 为各因素的权重. 权重越大,训练出来的模型越侧重于该因素,同时,某个过高的权重又可能导致模型不收敛. 多目标强化学习中,奖赏函数对策略网络的影响是复杂的,难以通过理论分析给出一个具体的最佳奖赏函数. 在后面的实验部分,将通过对 ω_1、ω_2、ω_3 取不同值,比较其跟随性能,从而筛选出最佳权重.

2.4 经验回放机制

DDPG算法采用经验回放的方法,将agent与环境交互产生的经验样本存放于经验缓冲池中,并从中随机抽取样本用来训练网络. 这种随机抽取样本的方法,既没有考虑到不同数据所具有的不同的重要性,又没有充分考虑到被抽取样本应具有的多样性,导致模型收敛较慢. 针对这一问题,本文提出一种新的样本存储和抽取策略,即按照数据的类型和重要性的不同分开存放,从而有效提高模型的收敛速度.

不同于原生DDPG算法的单个经验缓冲池,本文改进算法将经验缓冲池分为4部分,每部分容量为 M. 每次获得经验样本后,按照样本在车间距和加速度大小这两方面的表现,分别将样本存放在4个子缓冲池中,如图1所示.

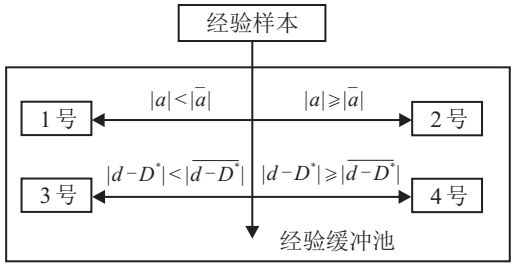


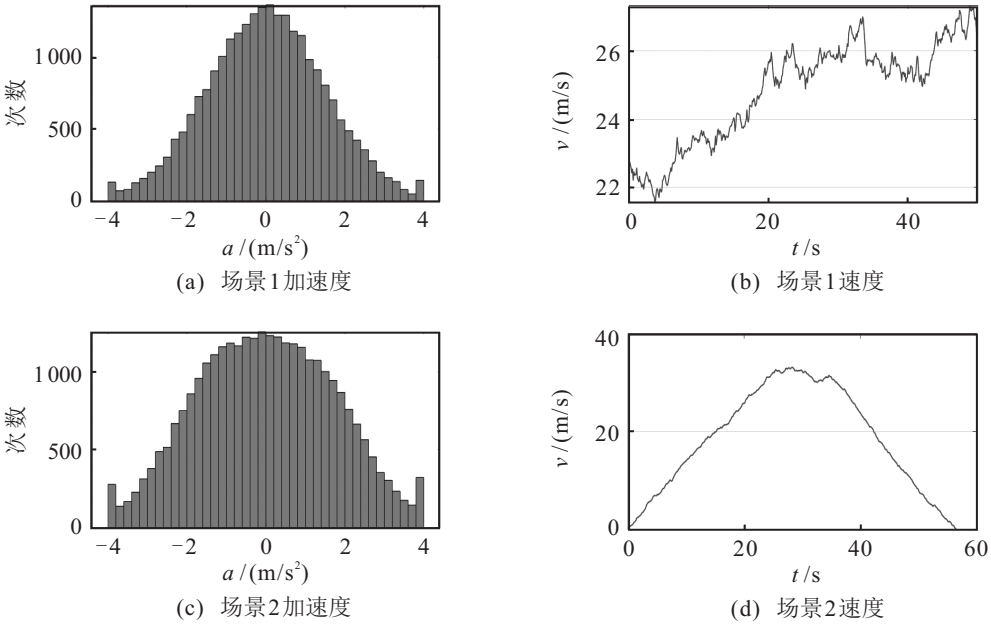
图1 样本存储

缓冲池存满后,开始抽取样本,训练网络,不同的子缓冲池具有不同的抽取量. 1号和3号子缓冲池中数据较2号和4号更优,每次抽取更多的样本用于训练. 1号和3号每次抽取batch 1组经验样本,2号和4号每次抽取batch 2组经验样本. 同时,1号中存储的样本 a 较小,3号中存储的样本 d 较小,所以1号和3号中的样本差异较大,同时从这两个子缓冲池中抽取较多数据,可以保证抽样数据的多样性.

3 实 验

为了验证模型的有效性和泛化性,本文用训练好的模型在两个不同的跟车场景下进行实验,结果表明,模型在两个场景下的表现均优于文献[3]和文献

[7]中方法. 实验中, $d_{\max}=100, v_{\min}=-80, v_{\max}=100, a_{1\min}=a_{2\min}=-4, a_{1\max}=a_{2\max}=4, T_s=0.1, \tau=0.15, t_0=0.02, d_0=3, M=2\,500, \text{batch}\,1=32, \text{batch}\,2=8$.



(a) 场景1加速度　(b) 场景1速度　(c) 场景2加速度　(d) 场景2速度

图 2　场景信息

表 3　不同权重值的跟随性能

$\omega_1:\omega_2:\omega_3$	$\lvert a\rvert/(\text{m}\cdot\text{s}^{-2})$	$\lvert\text{jerk}\rvert/(\text{m}\cdot\text{s}^{-3})$	d/m
1:1:1	4.0	0	3003
1:5:5	1.08	5.51	133.69
1:10:10	1.11	3.60	5.71
1:15:15	4.0	0	3 003
1:10:15	4.0	0	3 003
1:15:10	1.09	2.25	5.47
1:20:10	4.0	0	3 003
1:16:10	1.99	3.30	5.44
1:14:10	1.12	3.48	5.77

场景1中, 领航车辆初始速度为23 m/s, 在行驶过程中, 每0.1 s改变一次速度, $a\sim N(0,2)$. 场景2中, 领航车辆初始速度为0, 每0.1 s改变一次速度, 先加速, 然后速度保持, 最后减速到0. 加速过程中, 加速度 $a_1\sim N(1.5,1.5)$; 速度保持过程中, 加速度 $a_2\sim N(0,2)$; 减速过程中, 加速度 $a_3\sim N(-1.5,1.5)$.

从表3中可以看出, 当 $\omega_1:\omega_2:\omega_3=1{:}15{:}10$ 时, 性能最好. 此时, 奖赏函数的表达式为

$$R=R_1+15R_2+10R_3. \tag{17}$$

3.2　模型训练

共进行300回合的跟车训练, 每次训练时长50 s, 训练过程中, 总奖赏值如图3所示.

3.1　奖赏函数权重值选取

使用不同的权重值, 在场景1中训练模型, 场景2中测试模型. 各场景领航车辆的加速度与速度信息如图2所示, 跟随性能如表3所示.

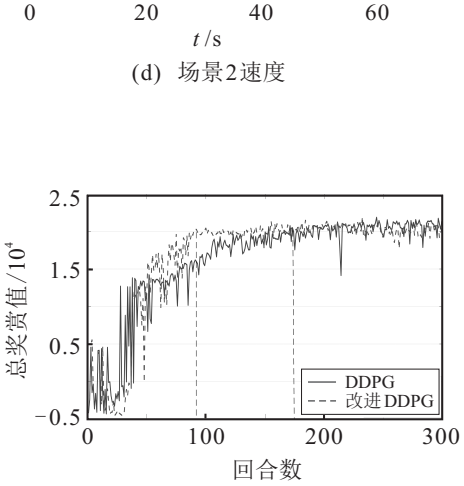


图 3　训练过程总奖赏值

从图3中可以看出, 改进后的DDPG算法相对于原生的DDPG算法, 收敛速度有了明显提高.

3.3　模型测试

3.3.1　场 景 1

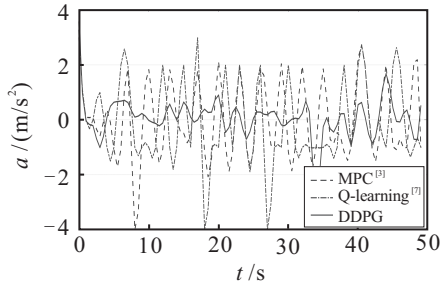
第1组测试场景为场景1. 本文方法与文献[3]和文献[7]中方法的比较结果如图4和表4所示. 其中: 文献[3]使用引入修正项的多目标MPC控制算法, 文献[7]使用Q-learning算法进行决策.

从图4和表4中可以看出, 场景1下, DDPG相对于Q-learning和MPC而言, 具有出色的舒适性、安全性和高效性.

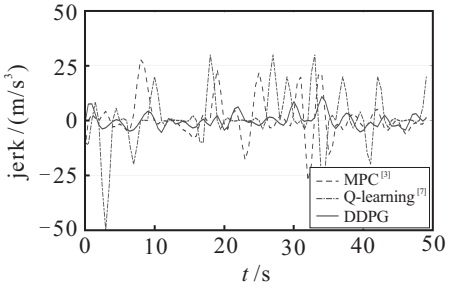
3.3.2　场 景 2

第2组测试场景为场景2, 结果如图5和表5所示.

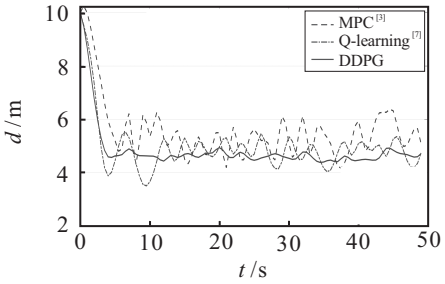
从图5和表5中可以看出, 本文提出的跟车方法在与训练环境不同的测试环境下, 依然能够安全、高



(a) 加速度

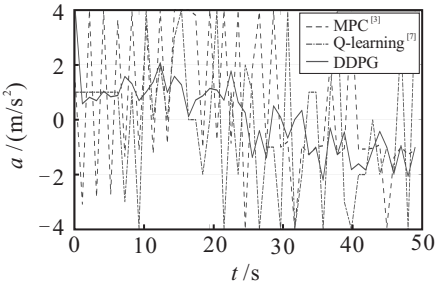


(b) 加速度变化率

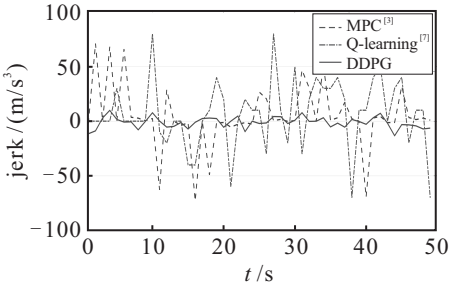


(c) 车间距

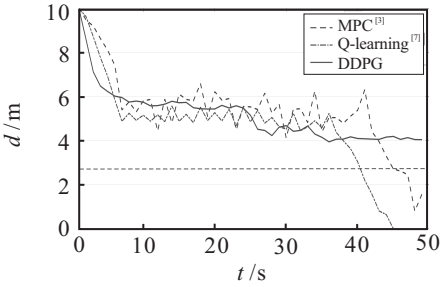
图 4 场景1下性能比较



(a) 加速度



(b) 加速度变化率



(c) 车间距

图 5 场景2下性能比较

表 4 场景1下性能比较结果

跟车方法	$\lvert a\rvert$ / (m · s^{-2})	|jerk| / (m · s^{-3})	d / m
领航车辆	1.20	16.96	
MPC[3]	1.08	5.44	5.64
Q-learning[7]	1.34	5.53	5.01
DDPG	0.51	3.34	4.80

表 5 场景2下性能比较结果

跟车方法	$\lvert a\rvert$ / (m · s^{-2})	|jerk| / (m · s^{-3})	d / m
领航车辆	1.412	14.55	
MPC[3]	1.75	13.33	5.36
Q-learning[7]	2.21	18.20	4.90
DDPG	1.09	2.25	5.47

效、舒适地跟随领航车辆,表明了该方法具有良好的泛化性. 从图5(c)可以看出: MPC在后期领航车辆制动时,车间距过小,低于缓冲距离d_0,安全性无法保障; Q-learning在后期领航车辆制动时,直接与其发生追尾,说明其泛化性差,在测试环境与训练环境不同时,性能下降明显.

4 结 论

本文首先对跟车过程进行MDP建模,然后通过对DDPG算法进行改进,得到了该模型的最优决策策略. 通过改进DDPG算法经验回放机制,有效提高了模型的收敛速度. 将奖赏函数进行模块化设计,可以较方便地解决多目标决策问题.

目前而言,本文模型只在模拟环境中具有良好表现. 由于强化学习“在试错中学习”的特点,难以在真实环境中对模型进行训练. 同时,仿真环境与真实环境存在较大差别,真实环境中存在各种复杂的干扰因素. 后期将尝试把仿真环境中训练好的模型移植到本团队的无人车上,使其能在真实环境中完成车辆跟随任务.

参考文献(References)

[1] Kuefler A, Morton J, Wheeler T, et al. Imitating driver behavior with generative adversarial networks[J]. 2017, arXiv: 1701. 06699.

[2] 刘秉政, 高松, 曹凯, 等. 车辆跟随控制策略的状态可达集建模及验证方法[J]. 控制与决策, 2021, 36(7): 1679-1685.

(Liu B Z, Gao S, Cao K, et al. A modeling and verification method of state reachable set for vehicle following control strategy[J]. Control and Decision, 2021, 36(7): 1679-1685.)

[3] 吴光强, 郭晓晓, 张亮修. 汽车自适应巡航跟车多目标鲁棒控制算法设计[J]. 哈尔滨工业大学学报, 2016, 48(1): 80-86.
(Wu G Q, Guo X X, Zhang L X. Multi-objective robust adaptive cruise control algorithm design of car following model[J]. Journal of Harbin Institute of Technology, 2016, 48(1): 80-86.)

[4] 李润梅, 张立威, 王剑. 基于时变间距和相对角度的无人车跟随控制方法研究[J]. 自动化学报, 2018, 44(11): 2031-2040.
(Li R M, Zhang L W, Wang J. A control method of unmanned car following under time-varying relative distance and angle[J]. Acta Automatica Sinica, 2018, 44(11): 2031-2040.)

[5] Wang X, Jiang R, Li L, et al. Capturing car-following behaviors by deep learning[J]. IEEE Transactions on Intelligent Transportation systems, 2018, 19(3): 910-920.

[6] 高阳, 陈世福, 陆鑫. 强化学习研究综述[J]. 自动化学报, 2004, 30(1): 86-100.
(Gao Y, Chen S F, Lu X. Research on reinforcement learning technology: A review[J]. Acta Automatica Sinica, 2004, 30(1): 86-100.)

[7] Gao Z H, Sun T J, Xiao H W. Decision-making method for vehicle longitudinal automatic driving based on reinforcement Q-learning[J]. International Journal of Advanced Robotic Systems, 2019, 16(3): 1-13.

[8] Mnih V, kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[9] Okuyama T, Gonsalves T, Upadhay J. Autonomous driving system based on deep Q learnig[C]. International Conference on Intelligent Autonomous Systems. Singapore, 2018: 201-205.

[10] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. 2016, arXiv: 1509.02971.

[11] Kendall A, Hawke J, Janz D, et al. Learning to drive in a day[J]. 2018, arXiv: 1807.00412.

[12] 徐国艳, 宗孝鹏, 余贵珍, 等. 基于DDPG的无人车智能避障方法研究[J]. 汽车工程, 2019, 41(2): 206-212.
(Xu G Y, Zong X P, Yu G Z, et al. A research on intelligent obstacle avoidance of unmanned vehicle based on DDPG algorithm[J]. Automotive Engineering, 2019, 41(2): 206-212.)

[13] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. 2017, arXiv: 1707.06347.

[14] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[J]. 2018, arXiv: 1801.01290.

[15] Wu B, Feng Y P. Policy reuse for learning and planning in partially observable markov decision processes[C]. International Conference on Information Science and Control Engineering. Changsha, 2017: 549-552.

[16] Wang X, Huang Q Y, Celikyilmaz A, et al. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation[J]. 2018, arXiv: 1811.10092.

[17] Zhu Y K, Mottaghi R, Kolve E, et al. Target-driven visual navigation in indoor seense using deep reinforcement learning[C]. IEEE International Conference on Robotics and Automation. Singapore, 2017: 3357-3364.

[18] Lowe R, Ziemke T. Exploring the relationship of reward and punishment in reinforcement learning[C]. IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning. Singapore, 2013: 140-147.

[19] 孙天骏. 基于学习控制的汽车全速自适应巡航决策与控制算法研究[D]. 长春: 吉林大学, 2019.
(Sun T J. Research on decision and control algorithm for vehicle full-speed adaptive cruise based on learning control[D]. Changchun: Jilin University, 2019.)

作者简介

邓小豪 (1996–), 男, 硕士生, 从事强化学习及无人驾驶决策方法的研究, E-mail: 2247585300@qq.com;

侯进 (1969–), 女, 副教授, 博士, 从事深度学习及无人驾驶等研究, E-mail: jhou@swjtu.edu.cn;

谭光鸿 (1994–), 男, 硕士生, 从事深度学习及机器视觉的研究, E-mail: 1424148078@qq.com;

万斌杨 (1996–), 男, 硕士生, 从事迁移学习及无人驾驶的研究, E-mail: 824050747@qq.com;

曹婷婷 (1995–), 女, 硕士生, 从事深度学习及无人驾驶的研究, E-mail: 1845832836@qq.com.

(责任编辑: 李君玲)