DOI:10.3969/j.issn.1003-5060.2024.05.001

基于强化学习经验优先提取的汽车纵向多态控制

黄 鹤[1,2,3], 付梦园[1,2,3], 吴润晨[1,2,3], 黄泽辰[1,2,3], 曾 琦[1,2,3], 石 琴[1,2,3]
(1.合肥工业大学 汽车与交通工程学院,安徽 合肥 230009; 2.合肥工业大学 智能制造技术研究院,安徽 合肥 230009; 3.安徽省智慧交通车路协同工程研究中心,安徽 合肥 230009)

摘 要:文章提出一种引入经验优先提取(prioritized experience extraction,PEE)规则的深度 Q 网络(deep Q network,DQN)算法,用于解决汽车纵向行驶时的多态控制问题。首先,建立车辆纵向力矩传递模型和强化学习算法模型,在进行算法移植以及制定奖励函数时综合考虑车速、距离等相关因素的综合限制;然后,通过仿真与硬件在环实验验证强化学习算法在汽车纵向多态控制方面的有效性;最后,引入 PEE 规则提高常规 DQN 算法的计算效率,解决算法区域性过拟合问题。PEE 规则的引入有助于平滑主车的跟随车速,与相对距离相配合提升了行驶时的舒适性与安全性。

关键词:深度强化学习;纵向控制;多态控制;经验优先提取(PEE)规则
中图分类号:U461.6 **文献标志码**:A **文章编号**:1003-5060(2024)05-0577-08

Longitudinal polymorphic control of vehicle based on reinforcement learning with prioritized experience extraction

HUANG He[1,2,3], FU Mengyuan[1,2,3], WU Runchen[1,2,3],
HUANG Zechen[1,2,3], ZENG Qi[1,2,3], SHI Qin[1,2,3]
(1. School of Automobile and Traffic Engineering, Hefei University of Technology, Hefei 230009, China; 2. Intelligent Manufacturing Institute, Hefei University of Technology, Hefei 230009, China; 3. Engineering Research Center for Intelligent Transportation and Cooperative Vehicle-Infrastructure of Anhui Province, Hefei 230009, China)

Abstract: In order to solve the polymorphic control problem of the longitudinal motion of vehicle, a deep Q network(DQN) algorithm based on the rule of prioritized experience extraction(PEE) was put forward. The vehicle longitudinal torque transfer model and reinforcement learning algorithm model were analyzed and established. The transplantation of the algorithm and the formulation of reward function were made taking the comprehensive limitations of the relevant factors such as speed and distance into consideration. Through the simulation and hardware-in-the-loop experiment, the effectiveness of deep reinforcement learning algorithm in the longitudinal polymorphic control of vehicle is verified. In addition, PEE rule was introduced to improve the computational efficiency of conventional DQN algorithm and solve the overfitting problem to some extent. The PEE rule also realizes the smooth following speed of the main vehicle, which, in combination with the relative distance, improves the comfort and safety during driving.

Key words: deep reinforcement learning; longitudinal control; polymorphic control; prioritized experience extraction(PEE) rule

收稿日期:2023-05-04;**修回日期**:2023-06-20
基金项目:国家自然科学基金资助项目(71971073);2023 年度长三角科技创新共同体联合攻关资助项目(2023CSJGG1600);安徽省新能源汽车暨智能网联汽车创新工程资助项目(GXXT-2020-076)和 2019 年度合工大智能院"科技成果转化及产业化"资助项目(IMICZ2019005)
作者简介:黄 鹤(1983—),男,安徽合肥人,博士,合肥工业大学讲师,硕士生导师;
石 琴(1963—),女,安徽合肥人,博士,合肥工业大学教授,博士生导师.

0 引 言

汽车保有量的增加和有人驾驶事故率的居高不下,使得科学研究者们更加关注无人驾驶以及相关辅助驾驶技术。然而,在面对复杂道路环境和交通工况的情况下,现有技术仍然很难代替驾驶员来进行车辆操作,相关技术研究还有待深入探索[1-3]。

目前,汽车纵向控制包括自适应巡航、自动紧急制动、汽车停走等相关多态控制算法,学者们在相关方面进行了大量研究,提出了车辆预测模型[4]、车头时距控制[5]、滑模控制[6]等主流方法。文献[7-8]采用预测控制方法,在多工况下对车头时距进行有效控制,实现了可靠的自适应巡航功能,并提出多工况切换规则,使工况算法切换较为平滑;文献[9]结合高斯函数,引入非线性参考模型并设计了相应的模糊控制器,以保证车辆在较小相对距离内驾驶员的安全性与舒适性。上述传统汽车纵向控制方法都能达到对车辆的动态控制,但在多态控制方面需要进行算法切换,且会存在车辆行驶过程中的随机性问题,对于离散化动态数据的处理也有较多限制。

强化学习是一种利用马尔可夫模型[10],通过智能体与环境交互学习以达到最大奖励动作的机器学习算法[11],且能够利用单独的算法实现多态控制。针对随机性问题,强化学习给出了不同的解决思路。文献[12-13]利用 Q-learning 算法,结合深度学习的神经网络模型,将随机性离散状态数据转换为车辆执行动作,并参考经验回放,打断了庞大数据之间的相关性,达到了较好的车辆纵向控制效果;文献[14]在普通神经网络的基础上利用真实驾驶员数据进行预训练,并加入长短期记忆(long short-term memory,LSTM)网络算法,预测车辆下一状态参数信息,提升了训练效果和速度,也使神经网络的权重参数获得了更好的收敛性。但由于车辆在行驶过程中状态信息较多,且大多数状态的奖励值较小,导致迭代神经网络参数时出现区域性过拟合现象,不仅大大增加了训练时间,还得不到理想的训练效果。

为此,本文提出了一种融合深度强化学习和车辆状态参数优先提取规则的汽车纵向多态控制算法。利用深度 Q 网络(deep Q network,DQN)算法进行网络参数的迭代训练,同时为了避免过小奖励值的状态输入,在神经网络模型中加入了经验优先提取(prioritized experience extraction,PEE)规则,用来针对优先级较高的数据进行训练;最后,对常规 DQN 算法和基于 PEE 规则的 DQN 算法(PEEDQN 算法)实验结果进行比较,并得出相应结论。

1 车辆纵向力矩传递系统建模

由牛顿第二定律,可得车辆的纵向运动学方程为:

$$F_t = F_f + F_w + F_\psi + F_\zeta \tag{1}$$

其中:F_t 为汽车驱动力;F_f 为轮胎滚动阻力;F_w 为空气阻力;F_ψ 为坡度阻力;F_ζ 为加速阻力。将式(1)展开可得:

$$\frac{T_t}{l} = \frac{Wd}{l} + \frac{C_D A v^2}{21.15} + G\psi + \delta m \frac{\mathrm{d}v}{\mathrm{d}t} \tag{2}$$

其中:T_t 为驱动力矩;l 为车轮半径;W 为轮胎所受法向载荷;C_D 为空气阻力系数;A 为车辆行驶时的迎风面积;v 为车辆的行驶速度;G 为车辆重力;ψ 为坡度大小的正弦值;δ 为车辆旋转质量转换系数;m 为车辆质量。

建立车辆驱动力矩传递模型,由油门开度控制驱动力矩大小,如图 1 所示。

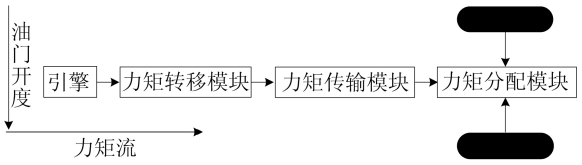

图 1 驱动力矩传递模型

针对汽车制动力,有如下表达式:

$$F_{Xb} = \begin{cases} T_\mu / l; \\ F_z \varphi, \quad p_{MC} \geqslant p_a \end{cases} \tag{3}$$

其中:F_{Xb} 为汽车制动力;T_μ 为车轮制动器摩擦力矩;F_z 为地面对轮胎的法向反作用力;φ 为地面附着系数;p_{MC} 为主缸制动压力,当 p_{MC} 达到 p_a 值时,车轮出现抱死现象,汽车制动力达到最大值不再改变。

建立车辆制动力矩传递模型,如图 2 所示。

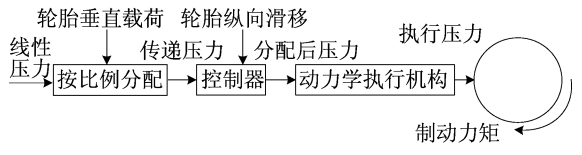

图 2 制动力矩传递模型

2 算法建模

强化学习[15]又称再励学习,具体过程如图 3

所示,智能体在环境中获得状态 S 以及此状态下环境反馈的奖励值 R,这会影响智能体的下一步动作 A 且会使其达到下一状态,经过一定次数的迭代后,智能体就会学习到在此环境下完成相应任务所需的动作策略[16]。

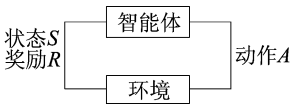

图 3　强化学习流程

2.1　马尔可夫决策过程

上述过程在任意时刻 $t=0,1,2,3,\cdots$,智能体都会与环境发生信息交换。即在任意时刻 t,智能体会观测环境并得到环境在 t 时刻的描述参数状态 $S_t\in S$ 和反馈奖励值 $R_t\in R$。根据这些数据,智能体会确定一个动作 $A_t\in A$ 并达到下一个状态 S_{t+1}。因此,智能体与环境的交互轨道可以表示如下:

$$S_0,A_0,R_1,S_1,A_1,R_2,S_2,A_2,R_3,\cdots \tag{4}$$

智能体在状态 $S_t=s$ 以及动作 $A_t=a$ 时转移到下一个状态 $S_{t+1}=s'$ 并且获得奖励 $R_{t+1}=r$ 的概率为:

$$K(s',r\mid s,a)=P\{S_{t+1}=s',R_{t+1}=r\mid S_t=s,A_t=a\} \tag{5}$$

其中:K 为马尔可夫的动态特性,上述过程即马尔可夫决策过程(Markov decision process,MDP)。

由式(5)可知,智能体达到状态 S_{t+1} 并且获得奖励 R_{t+1} 取决于上一时刻的状态 S_t 以及动作 A_t,与更早的状态和动作完全无关。

2.2　值函数迭代更新

Q-learning 是强化学习的经典算法[17],以动作值函数 $Q(s,a)$ 表示智能体在状态 s 时采取动作 a 所能获得的收益,具体以二维表格的形式来存储 Q 值,以状态值函数 $V_\pi(s)$ 表示智能体在状态 s 时的收益期望,有:

$$V_\pi(s)=E_\pi[R_{t+1}+\gamma(R_{t+2}+\gamma(\cdots))\mid S_t=s] \tag{6}$$

$$Q(s,a)=E_\pi[R_{t+1}+\gamma R_{t+2}+\gamma^2R_{t+3}+\cdots\mid A_t=a,S_t=s] \tag{7}$$

其中:γ 为奖励衰减因子。γ 趋近于 0 时,价值函数基本由当前状态的奖励值决定;γ 趋近于 1 时,价值函数由所有状态的奖励值决定。

则有:

$$Q(s,a)=R_s^a+\gamma\sum_{s_{t+1}\in S}P_{s_{t+1}}^aV_\pi(s_{t+1}) \tag{8}$$

展开得到值函数的迭代更新公式如下:

$$Q(s,a)\leftarrow Q(s,a)+\eta[R+\gamma\max_{a_{t+1}}Q(s_{t+1},a_{t+1})-Q(s,a)] \tag{9}$$

其中,η 为学习效率。

值函数在更新时会依据贪婪策略,即选择位于状态 s_{t+1} 时的最大值函数所对应的动作来执行。Q 表初始值是随机的,随着 Q 表的更新,$Q(s,a)$ 会得到越来越精确的近似解,即智能体会在相应状态选择正确的动作来执行,以达到 Q-learning 的学习目的。

2.3　DQN 算法

Q-learning 算法利用二维表格来存储 Q 值,并且状态与动作集的参数量较少,易于处理状态简单且动作单一的智能体与环境的交互行为。但在面对庞大的状态与动作集时,存储每个状态的 Q 值会需要范围更加庞大的表格,不仅会占用大量内存,还使算法的计算速度相应下降,对值函数的逼近也会造成不确定的误差。因此,DQN 算法被提出,该算法在传统强化学习的基础上做了一定的拓展,从多方面对 Q-learning 算法进行完善与改进。

首先,通过深度神经网络来获得相应的动作值函数,DQN 算法弃用了之前的表格形式来存储 Q 值,创造性地结合深度神经网络非线性拟合的优点,以虚拟的形式拟合 Q 表,只需输入状态参数即可通过神经网络计算出相对应的动作值函数;其次,加入目标神经网络,通过计算目标 Q 值来打乱状态参数之间的相关性,即

$$Q_t=R_t+\gamma\max Q(s_{t+1}),a) \tag{10}$$

其中,Q_t 为目标 Q 值。深度神经网络利用目标 Q 值来计算损失函数 L,即

$$L=\frac{1}{m}\sum_{j=1}^{m}[Q_{t_j}-Q_j(s,a,w,b)]^2 \tag{11}$$

其中:m 为神经网络从记忆库中采样的数据数;w、b 分别为神经网络的权重和偏置。神经网络的目标是利用梯度下降法通过反向误差传递来不断减小损失函数的值,从而使 w、b 的值得到收敛优化。

经验回放是 DQN 算法的重要组成部分,根据海马体记忆存储以及回放的功能,将智能体经历过的所有状态信息存放在记忆库中,并在神经网络训练时随机提取记忆库中的数据,将过去的经验与当前的经验相混合,打乱了状态数据之间

的相关性。同时,经验回放的方法还使得过去的经验得以重用,提高了智能体的学习效率。

3 算法移植

3.1 车辆 MDP 模型

对于车辆状态与动作的数据处理,本文采用离散化的数据分析,在车辆行驶过程中,定义车辆在 t 时刻的状态为:

$$S_t = \{d_t, a_{x_1,t}, \Delta v_t, \Delta e_{vt}\},$$

其中:d_t 为主车与前车的相对距离;$a_{x_1,t}$ 为主车的纵向加速度;Δv_t 为主车与前车的相对速度;Δe_{vt} 为前车车速与两车相对距离的差值。

定义车辆在 t 时刻所执行的动作 $A_t = \{T_t, B_t\}$,其中:T_t 为主车的油门开度;B_t 为主车的主缸制动压力。根据主车与前车的纵向运动学特性,有如下表达式:

$$\begin{cases} d_{t+1} = d_t + \Delta v_t \Delta t + 0.5 a_{x_2,t} (\Delta t)^2 - 0.5 a_{x_1,t} (\Delta t)^2, \\ a_{x_1,t+1} = \left(1 - \dfrac{\Delta t}{\tau}\right) a_{x_1,t} + \dfrac{\Delta t}{\tau} \alpha_t, \\ \Delta v_{t+1} = \Delta v_t + a_{x_2,t} \Delta t - a_{x_1,t} \Delta t, \\ \Delta e_{v,t+1} = v_{f,t} + a_{x_2,t} \Delta t - d_{t+1} \end{cases} \tag{12}$$

其中:Δt 为离散化时间间隔;$a_{x_2,t}$ 为前车纵向加速度;τ 为一阶惯性环节时间常数;α_t 为主车在 t 时刻的执行纵向加速度;$v_{f,t}$ 为前车车速。

主车在 t 时刻受到环境的状态反馈后,会执行相对应的动作$\{T_t, B_t\}$,并转化为一定的加速度值,在执行此动作后,环境会反馈主车在 $t+1$ 时刻的奖励值 R_{t+1} 且得到下一时刻的状态 S_{t+1}。上述状态 S 见表 1 所列,动作 A 见表 2 所列。

表 1 状态 S 参数集

参数	取值范围
两车相对距离/m	$[0, d_{\max}]$
主车纵向加速度/(m/s^2)	$[a_{\min}, a_{\max}]$
两车相对车速/(m/s)	$[\Delta v_{\min}, \Delta v_{\max}]$
前车车速与两车相对距离差	$[\Delta e_{v,\min}, \Delta e_{v,\max}]$

表 2 动作 A 参数集

参数	取值范围
油门开度	$[0,1]$
主缸制动压力/MPa	$[0, b_{\max}]$

表 1 中:$d_{\max}$ 为车辆传感器的极限探测距离,当主车与前车相对距离超过传感器极限范围时,即 $d > d_{\max}$ 时,取 $d = d_{\max}$;$\Delta v_{\max}$ 为主车与前车相对车速最大值,且有 $\Delta v_{\min} = -\Delta v_{\max}$;$a_{\max}$ 为主车加速度最大值,且有 $a_{\min} = -a_{\max}$;$\Delta e_{v,\max}$ 为前车车速与两车相对距离差值的最大值,且有 $\Delta e_{v,\min} = -\Delta e_{v,\max}$。

表 2 中,$b_{\max}$ 为车辆最大主缸压力。

3.2 奖励函数设置

奖励函数是强化学习的重要组成部分,智能体执行动作的可行性以及神经网络参数迭代收敛的好坏取决于奖励函数的设计是否成功。

本文基于车辆的纵向多态控制,对奖励函数采用两方面的设计,将车速跟随与距离跟随相结合,目的是使车辆在行驶过程中既能够在车速上完成相应的跟车行驶或启停操作,也能够保持一定的安全距离。

在车速方面,设计奖励函数如下:

$$R_1 = \begin{cases} 0, & d < 0, d > d_{\max}, |\Delta v| > \Delta v_{\max}; \\ \exp(-1.2 |\Delta v|), & 0 \leqslant d \leqslant d_{\max}, 0 \leqslant |\Delta v| \leqslant \Delta v_{\max} \end{cases} \tag{13}$$

其中,R_1 取值范围为$[0,1]$。当 $R_1 = 0$ 时,表示无奖励,即此时执行的动作不会使环境反馈奖励,这些动作在相应的状态不会被采取执行;当 $R_1 = 1$ 时,$\Delta v = 0$,表示奖励最大,即相对车速为 0,主车车速与前车车速相等,此时主车执行的动作会实现较好的车速跟随,这些动作在相应的状态会被采取执行。

在距离方面,设计奖励函数如下:

$$R_2 = \begin{cases} 0, & d < 0, d > d_{\max}, |\Delta e_v| > \Delta e_{v,\max}; \\ \exp(-1.2 |\Delta e_v|), & 0 \leqslant d \leqslant d_{\max}, 0 \leqslant |\Delta e_v| \leqslant \Delta e_{v,\max} \end{cases} \tag{14}$$

其中,R_2 取值范围为$[0,1]$。当 $R_2 = 0$ 时,表示无奖励,即此时执行的动作不会使环境反馈奖励,这些动作在相应的状态不会被采取执行;当 $R_2 = 1$ 时,$\Delta e_v = 0$,表示奖励最大,即主车与前车的相对距离等于前车车速,此时主车执行的动作会实现较好的距离跟随,这些动作在相应的状态会被采取执行。

车速与距离大多数情况下并不能同时满足大奖励条件,追求完美的车速跟随往往会导致较差的距离跟随,反之亦然,例如前后两车实现了高速的跟车行驶,但相对距离却较小,此时可能会导致追尾事故的发生。

故将奖励函数表达式写作如下形式:

$$R=\omega_1 R_1+\omega_2 R_2 \tag{15}$$

其中,ω_1、ω_2 为车速与距离奖励函数的权重实验部分。通过对 ω_1、ω_2 选取不同值,比较对应的奖励函数值,以此得出最优权重。

4　加入 PEE 规则的改进 DQN 算法

经验回放是 DQN 算法的重要优点,将智能体存储的过去的经验信息提取出来打乱数据相关性,但处于存储空间中的数据被提取的概率是相同的,即神经网络参数的迭代过程既会训练损失函数较大的数据,也会不断训练那些效果已经相对较好的数据,这样的训练使得神经网络参数收敛速度变慢,导致一定的区域性过拟合现象。

为避免区域性过拟合现象,使具有较大损失函数的数据能够被优先训练,加速算法收敛,本文提出一种 PEE 经验优先提取规则的 DQN 算法,将 SumTree 作为记忆库对智能体的状态信息进行存储。其中,定义第 j 条数据的优先级公式为:

$$p_j=\left|Q_{t_j}-Q_j(s,a,w,b)\right| \tag{16}$$

PEEDQN 算法结构如图 4 所示。将数据的优先级存储进 SumTree 的叶子节点中,所有子节点的父节点值即为此父节点 2 个子节点数值之和,可得根节点数值即为所有数据优先级之和 p。在训练网络进行数据抽取时,有区间间隔数 $c=p/m$,在每个区间内随机选取一个数并将此数从根节点按照一定的规律往下遍历,遍历到的最后一个叶子节点所对应的数据即为所要提取的数据。遍历规律为:从根节点开始,随机区间数与根节点的左子节点对比大小;若此数比左子节点大,则此数减去左子节点并向右遍历,若此数比左子节点小,则此数向左遍历;此后遍历规律同上,直到遍历到最后一个叶子节点。

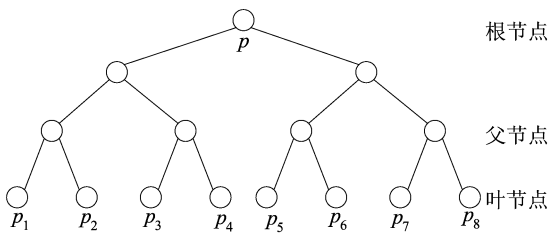

图 4　PEEDQN 算法结构示意图

在此过程中,由于数据的优先提取,产生了改变数据分布的误差,因此,引入如下优先级提取权重来修正该误差:

$$\omega_j=\left(\frac{p_j}{\min\limits_i p_i}\right)^{-\beta} \tag{17}$$

其中,i 表示训练回合数。

由此,损失函数改变为:

$$L=\frac{1}{m}\sum_{j=1}^{m}\omega_j\left[Q_{t_j}-Q_j(s,a,w,b)\right]^2 \tag{18}$$

可得 L 对 w、b 的偏导数为:

$$\frac{\partial L}{\partial w}=-\frac{2s_j}{m}\sum_{j=1}^{m}\omega_j\left[Q_{t_j}-Q_j(s,a,w,b)\right] \tag{19}$$

$$\frac{\partial L}{\partial b}=-\frac{2}{m}\sum_{j=1}^{m}\omega_j\left[Q_{t_j}-Q_j(s,a,w,b)\right] \tag{20}$$

通过神经网络利用梯度下降法更新迭代 w、b 的值如下:

$$w^n\leftarrow w^{n-1}-\left.\frac{\partial L}{\partial w}\right|_{w=w^{n-1},b=b^{n-1}} \tag{21}$$

$$b^n\leftarrow b^{n-1}-\left.\frac{\partial L}{\partial b}\right|_{w=w^{n-1},b=b^{n-1}} \tag{22}$$

训练结束后,收敛的 w、b 值参与动作参数的选取,利用贪婪策略,选取值函数大的动作作为最佳策略动作。

PEEDQN 算法流程如图 5 所示。

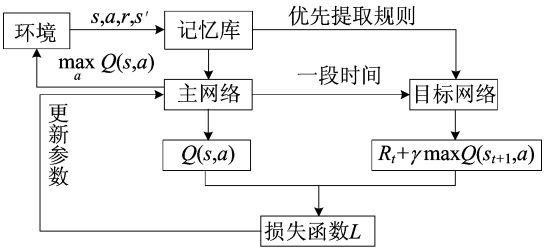

图 5　PEEDQN 算法流程

5　实验环境与结果

5.1　实验环境配置

本文利用软件仿真和硬件在环实验来验证所提算法的有效性。采用 CarSim 软件作为仿真平台进行参数迭代训练,将 Python 代码进行封装后与 CarSim 实现对接,控制指令由 Python 发出,CarSim 接收并执行,状态参数由 CarSim 发出,Python 接收并代入训练测试。本实验主机平台为 Windows10 企业版 LTSC,另外安装 Tensorflow、Cuda、Cudnn 以及相关 pip 库,所用硬件平台为 PXIe-1082 机箱、PXIe-8840RT 实时处理器、PXIe-8510 数据采集卡等相关硬件。

在 CarSim 中规定前车的多态工况,设定不规则的加减速、制动以及停走等相关操作,以达到算法进行多态控制的目的。训练初始规定前车车

速以及主车与前车相对距离,在每回合结束或不满足奖励值的情况下,初始化环境并随机设定主车车速以及主车与前车的相对距离。

5.2 奖励函数权重选取

本文实验分别对不同奖励函数权重值进行 10 000 回合跟车训练,得到不同权重的奖励函数值,具体见表 3 所列。

表 3 不同权重的奖励函数值

$\omega_1:\omega_2$	R
0.1∶0.9	568.432
0.2∶0.8	1 235.630
0.3∶0.7	1 899.756
0.4∶0.6	3 568.121
0.5∶0.5	4 023.689
0.6∶0.4	6 231.578
0.7∶0.3	8 873.638
0.8∶0.2	7 320.396
0.9∶0.1	5 783.125

从表 3 可以看出,当 $\omega_1=0.7$ 且 $\omega_2=0.3$ 时,奖励函数值最大。

继续对权重比 0.7∶0.3 附近细化权重分配,最优化结果见表 4 所列。

表 4 细化权重分配的奖励函数值

$\omega_1:\omega_2$	R
0.62∶0.38	6 735.454
0.64∶0.36	7 296.257
0.66∶0.34	7 865.298
0.68∶0.32	8 381.469
0.70∶0.30	8 873.638
0.72∶0.28	9 025.451
0.74∶0.26	8 758.267
0.76∶0.24	8 375.483
0.78∶0.22	7 628.672

从表 4 可以看出,细化权重分配后,当 $\omega_1=0.72$ 且 $\omega_2=0.28$ 时,奖励函数值最大,此时训练模型更倾向于以速度跟随为主、距离跟随为辅,得到奖励函数表达式如下:

$$R=0.72R_1+0.28R_2 \tag{23}$$

5.3 仿真实验结果与分析

本实验分别对常规 DQN 算法与加入 PEE 规则的 DQN 算法进行 10 000 回合的跟车训练,训练结果如图 6 所示。

从图 6 可以看出:常规 DQN 算法单回合训练步数呈类似指数增长,在需要高训练回合数的前提下,会导致训练陷入死循环,出现过拟合现象;改进后的 DQN 算法相较于常规 DQN 算法,单回合训练步数逐渐收敛,大大缩短了训练时间,提高了训练效率。

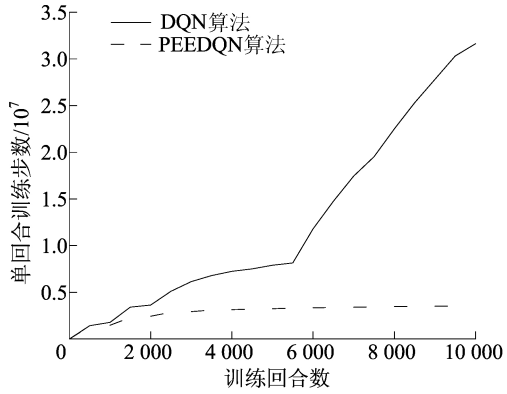

图 6 DQN 算法与 PEEDQN 算法的训练结果对比

为验证深度强化学习在汽车纵向多态控制中的有效性以及加入 PEE 规则后的高效性与稳定性,利用 CarSim 反馈数据测试 2 种算法的跟随效果。

主车速度与前车速度的对比如图 7 所示,方差的对比见表 5 所列。

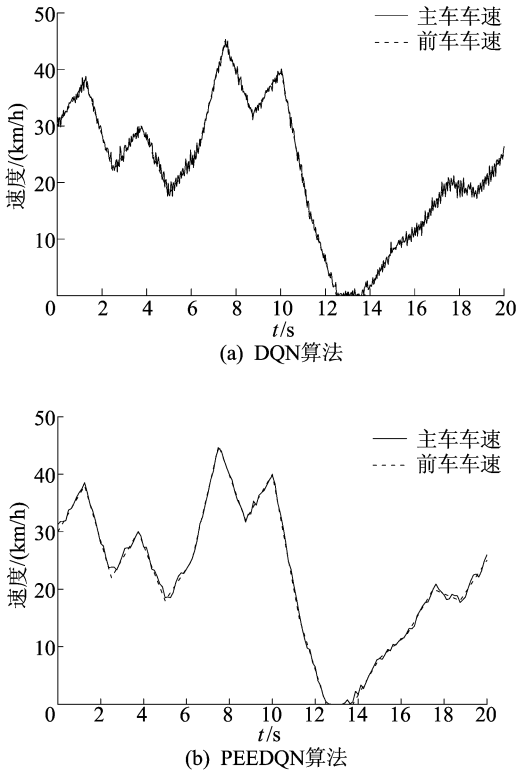

图 7 2 种算法的主车速度与前车速度对比

表 5　2 种算法的主车速度与前车速度的方差对比

算法	车速方差	主车车速与相对距离方差
DQN	0.54	3.80
PEEDQN	0.24	0.46

由图 7、表 5 可知:在车速跟随方面,常规 DQN 算法具有一定的跟随效果,能够完成初步的车辆跟随控制,主车车速与其前车车速方差为 0.54,但存在车速抖动较大,与前车车速贴合不紧密等情况;此外,在前车停止时,此算法不能够完全实现主车停止,具有一定的波动;应用 PEE 规则的 DQN 算法则能够较好地解决上述问题,通过计算可得,主车车速与前车车速方差为 0.24,相较于常规 DQN 算法降低了约 55.55%,不但实现了较好的车速跟随,体现了有效性,还实现了主车车速以平滑的方式贴合前车车速,提升了乘坐舒适性。

为验证主车与前车具有一定的安全距离,主车车速与两车相对距离的对比如图 8 所示。

由图 8 可知:在常规 DQN 算法下,主车车速与相对距离的方差为 3.80,主车车速与相对距离存在较大差异,只能在简单程度上实现两者的趋势拟合,不保证行驶安全性;而经过改进的 PEEDQN 算法则实现了主车车速与相对距离的完美配合,使车辆在行驶过程中既能实现有效跟随,也能保持一定的安全距离,大大提高了驾驶安全性。

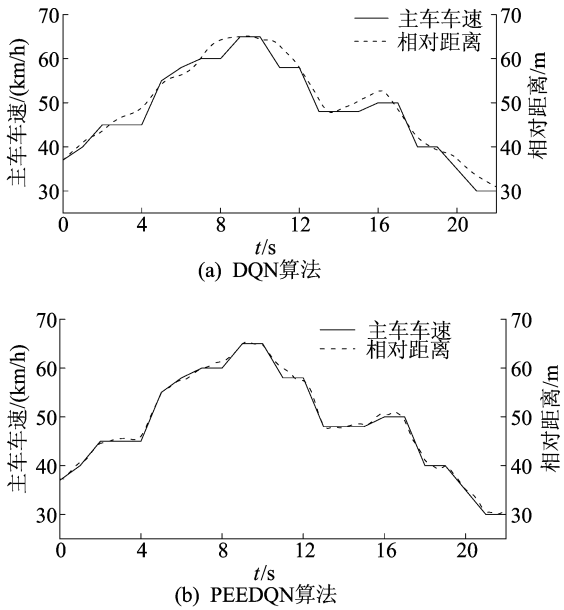

图 8　主车车速与相对距离的对比

5.4　硬件在环实验结果与分析

仿真实验验证了车辆在较低车速下的纵向跟随性能,为进一步验证本文 PEEDQN 算法在实际驾驶环境中的控制效果,本节采用硬件在环实验验证在高速工况下算法的有效性。

硬件在环实验流程图如图 9 所示。

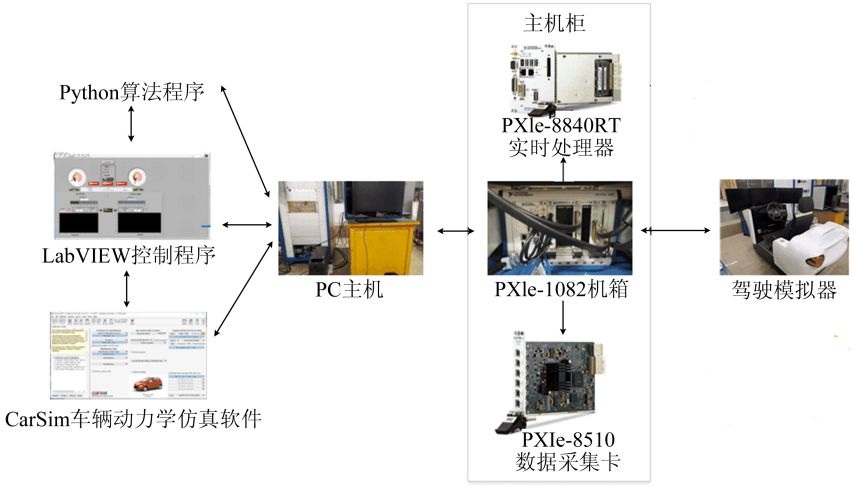

图 9　硬件在环实验流程图

通过 LabVIEW 程序实现上位机与硬件平台的实时数据通信,通过数据采集卡收集驾驶模拟器状态信息并利用 RT 实时处理器将数据反馈给上位机,上位机经过算法处理后下达控制指令,并由驾驶模拟器执行。

高速状态下车辆速度跟随图如图 10 所示。设定前车车速为 90.0～104.5 km/h 的高速状态并进行速度跟车实验。

由图 10 可知,采用 PEEDQN 算法后,主车能够实现基本的跟随性能,验证了该算法在实际

驾驶环境中的有效性,但由于硬件系统的时滞性,主车车速出现了一定的波动。

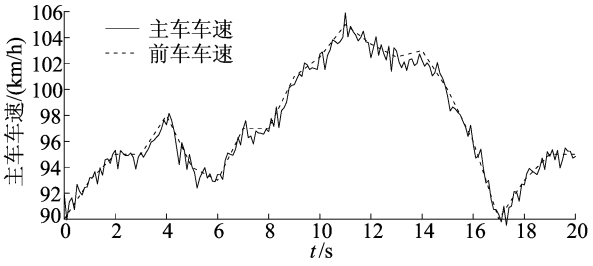

图10 高速状态下车辆速度跟随图

6 结 论

本文将PEEDQN算法用于解决汽车纵向多态控制问题。在进行车辆行为建模和制定算法的奖励函数时,考虑了车速因素(相对车速的上下限、主车车速的极限值)、距离因素(主车与前车相对距离的上下限、主车车速与相对距离的极限值)等多个目标的综合限制,此限制不仅对车速跟随效果产生影响,还可以限定主车行驶时的安全距离。

仿真实验结果显示,本文PEEDQN算法在解决汽车纵向多态控制问题时有效,能够实现主车行驶时的车速跟随、启停控制、安全驾驶等一系列功能。

硬件在环实验结果显示,PEEDQN算法在接近实车驾驶环境时具有基本的纵向控制效果,但底层算法还有待改进,用以处理硬件平台的延迟问题。

DQN算法引入PEE规则大大提高了常规DQN算法的计算效率,解决了一定程度上的过拟合问题,同时也使主车跟随前车时车速变化平缓,与相对距离相配合,实现了行驶时的舒适性与安全性。

[参 考 文 献]

[1] 姚广铮,刘小明,陈艳艳,等.城市汽车保有量极限值分析与预测[J].城市交通,2020,18(5):110-119.

[2] TOURAN A, BRACKSTONE M A, MCDONLAD M. A collision model for safety evaluation of autonomous intelligent cruise control[J]. Accident Analysis and Prevention, 1999,31(5):567-578.

[3] 刘作峰,张金友.无人驾驶汽车发展现状及发展趋势[J].河北农机,2020(1):56.

[4] NIE Z F, FARZANEH H. Adaptive cruise control for eco-driving based on model predictive control algorithm[J]. Applied Sciences,2020,10(15):5271.

[5] LIN T W, HWANG S L, GREEN P A. Effects of time-gap settings of adaptive cruise control (ACC) on driving performance and subjective acceptance in a bus driving simulator[J]. Safety Science,2008,47(5):620-625.

[6] SADAHITO K, SHIGETO O. Traction control for automobiles by model-following sliding mode control[C]//The Proceedings of the International Conference on Motion and Vibration Control. [S. l. : s. n.],2002:885-890.

[7] 段敏,刘振朋,陈天任.智能汽车可变车头时距的车间纵向控制研究[J].计算机仿真,2019,36(10):129-135.

[8] WEIMANN A, GRGES D, LIN X H. Energy-optimal adaptive cruise control combining model predictive control and dynamic programming[J]. Control Engineering Practice, 2018,72:125-137.

[9] MOHTAVIPOUR S M, MOLLAJAFARI M, NASERI A. A guaranteed-comfort and safe adaptive cruise control by considering driver's acceptance level[J]. International Journal of Dynamics and Control,2019,7(3):966-980.

[10] ODENTRANTZ J. Markov chains: gibbs fields, Monte Carlo simulation, and queues[J]. Technometrics, 2000, 42(4):53-156.

[11] ZOU F, YEN G G, TANG L X. A reinforcement learning approach for dynamic multi-objective optimization[J]. Information Sciences,2021,546:815-834.

[12] 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.

[13] 朱冰,蒋渊德,赵健,等.基于深度强化学习的车辆跟驰控制[J].中国公路学报,2019,32(6):53-60.

[14] 王丙琛,司怀伟,谭国真.基于深度强化学习的自动驾驶车控制算法研究[J].郑州大学学报(工学版),2020,41(4):41-45.

[15] SUTTON R S. Learning to predict by the methods of temporal differences[J]. Machine Learning,1988,3:9-44.

[16] HAFNER R, RIEDMILLER M. Reinforcement learning in feedback control[J]. Machine Learning, 2011, 84(1/2): 137-169.

[17] WATKINS C, DAYAN P. Technical note: Q-learning[J]. Machine Learning,1992,8(3/4):279-292.

(责任编辑 胡亚敏)