doi:10.19562/j.chinasae.qcgc.2019.02.013

基于 DDPG 的无人车智能避障方法研究*

徐国艳,宗孝鹏,余贵珍,苏鸿杰
(北京航空航天大学交通科学与工程学院,北京 100191)

[摘要] 本文中提出一种基于强化学习的无人车智能避障方法。鉴于无人车运动必须满足内外约束,包括汽车动力学约束和交通规则约束,且动作输出必须连续,而传统强化学习无法应对连续动作空间问题,提出了一种改进的 DDPG 算法,解决连续动作空间问题,实现转向盘转角和加速度的连续输出;采取多源传感器数据融合,满足无人车避障算法的状态输入;增加车辆内外约束条件,使输出动作更合理有效。最后,在开源仿真平台 TORCS 进行仿真,验证了算法的有效性和鲁棒性。

关键词:无人车;避障;强化学习;TORCS

A Research on Intelligent Obstacle Avoidance of Unmanned Vehicle Based on DDPG Algorithm

Xu Guoyan, Zong Xiaopeng, Yu Guizhen & Su Hongjie
School of Transportation Science and Engineering, Beihang University, Beijing 100191

[**Abstract**] An intelligent obstacle avoidance scheme for unmanned vehicle based on reinforcement learning is proposed in this paper. In view of that the movement of unmanned vehicle must meet both interior and exterior constraints, including vehicle dynamics constraints and traffic rule constraints and its output must be continuous, which the traditional reinforcement learning cannot assure, an improved deep deterministic policy gradient algorithm is proposed to tackle continuous motion space issue and achieve the continuous output of steering wheel angle and acceleration. Multi-source sensor data fusion is adopted to fulfill the state input of unmanned vehicle obstacle avoidance algorithm and both interior and exterior constraints are added to make output motion more reasonable and effective. Finally a simulation is conducted on the open-source simulation platform TORCS and the effectiveness and robustness of the algorithm verified.

Keywords: unmanned vehicle; obstacle avoidance; reinforcement learning; TORCS

前言

在未知环境中,无人车运行须躲避任意形状的静态和动态障碍物,为提高无人车的智能性,控制算法须考虑一系列环境状态。现有的控制算法大都是基于规则的,但这种人工经验编程很难应对其它突发情况,因此有必要提出一种更加智能的算法来解决这个问题[1]。

随着强化学习的发展,越来越多的研究者将其应用在无人车控制中。强化学习的目的是通过与环境的交互学习最优的行为。与传统的机器学习相比,强化学习有以下优势:第一,由于不需要样本标注过程,它能更有效地解决环境中存在的特殊情况;第二,可把整个系统作为一个整体,从而使其中的一些模块更加鲁棒;第三,强化学习可比较容易地学习到一系列行为。这些特点,对于无人车决策控制都很适用。

* 国家自然科学基金(51775016)资助。
原稿收到日期为 2017 年 11 月 17 日,修改稿收到日期为 2018 年 2 月 8 日。
通信作者:徐国艳,副教授,博士,E-mail:xuguoyan@ buaa.edu.cn。

深度强化学习(deep Q network, DQN)将深度学习与强化学习相结合,能更容易实现人类水平的控制。在 DQN 模型中,深度学习用来处理传感器数据,强化学习用来做出决策。这种模型已在 Atari 游戏中成功实现,并且通过像素输出,做出人类玩家水平的决策[2]。然而,DQN 模型只能处理离散低维动作空间的问题,而无人车的控制是一种连续高维动作空间问题,须输出连续的转向盘转角值和加速度值,DQN 算法无法解决。

一种典型的解决方法是将动作空间离散化,但离散程度过高,会引起"维数灾难";离散程度过低,会降低控制精度;因此,离散化动作空间不是解决连续动作空间问题的最优选择。针对这个问题,谷歌 DeepMind 团队提出深度确定性策略梯度算法(deep deterministic policy gradient, DDPG)[3],在仿真环境中实现了连续动作空间的控制。

本文中提出一种无人车控制模型,首先定义无人车的输入和输出状态,讨论车辆内外约束条件;然后设计奖赏函数与探索策略,提出改进的 DDPG 算法。为了评估避障策略的有效性,在 TORCS(the open racing car simulator)仿真软件中搭建各种仿真环境,包含不同赛道和不同车辆。通过设计静态和动态障碍环境来验证算法的有效性。结果表明,通过一段时间的自学习,无人车能够学习到优秀的行为,并且在新的测试环境中表现良好。

1　无人车状态定义

1.1　传感器数据融合

DDPG 算法中的环境状态信息通过传感器数据获得,输出动作包括无人车的转向、加速、制动和挡位值。因此首先要设计无人车传感器数据融合方法,作为 DDPG 算法的环境状态输入。

1.1.1　传感器类型

无人车通过各种传感器感知环境信息,常用的传感器包括摄像头、GPS、激光雷达和超声波雷达。摄像头可识别车道线信息,判断车辆与车道线的相对位置;GPS 提供实时的位置信息和车辆行驶状态,包括航向角、车速等;激光雷达检测到车身周围障碍物的距离信息;超声波雷达布置在车身周围,实现道路边缘检测。

1.1.2　多源传感器数据融合

无人车上的传感器可提供关于车辆状态和车辆周围环境的有用信息。避障算法中使用的输入变量名称和定义如表 1 所示。

表 1　输入变量定义

名称	范围	描述
Angle	$[-\pi,\pi]$	车辆航向角与道路轴线的夹角
SpeedX	$(-\infty,+\infty)$ km/h	车辆纵向速度
SpeedY	$(-\infty,+\infty)$ km/h	车辆横向速度
TrackPos	$(-\infty,+\infty)$	车辆与道路轴线的归一化距离,当车辆在轴线上时为 0,在道路右边缘时为-1,左边缘时为 1,大于 1 或小于-1 意味着车辆在道路外
Opponents	[0,100] m	返回 360°范围内障碍物距离

在真实运行环境中,通过摄像头进行车道线识别,获得车辆运行方向与道路轴线的夹角,用弧度表示,同时获得车辆质心至路面投影点与道路轴线的距离,并将此距离按道路半宽归一化为[-1,1]。通过 GPS 数据获得车辆纵向和横向速度,方向遵循汽车坐标系标准。通过激光雷达与超声波雷达,获得车辆周围 360°范围内障碍物距离,以及车辆与道路边缘的距离。

由于无人车获得的是多个不同类型的传感器观测数据,信息具有多样性和复杂性,因此须进行合理有效的融合,作为无人车状态输入。

多源传感器数据融合的过程如下:

(1) 收集多个不同类型传感器的观测数据;

(2) 对传感器的输出数据进行特征提取,得到代表观测数据的特征值;

(3) 对特征值进行数据关联,完成对相同目标的共同描述;

(4) 对不同目标的特征值进行组合,以字典的格式传入,作为强化学习模型中的状态输入。

至此,完成了多源传感器的数据融合和无人车的状态描述。

1.2　车辆约束条件

1.2.1　车辆动力学约束

无人车通过控制转向盘转角和加速/制动踏板来躲避障碍物。当速度过高时,车辆可能发生滑移甚至侧翻,因此为提高行车安全,车辆行驶时要求侧向加速度不大于 $0.4g$[4]。

$$a_y \leqslant 0.4g \tag{1}$$

式中:a_y 为侧向加速度;g 为重力加速度。

侧向加速度与前轮转角和车速的关系[5]为

$$a_y = u\frac{u/L}{1+Ku^2}\delta \tag{2}$$

式中:u 为车速;δ 为前轮转角;K 为稳定性因数。K 通过下式来计算:

$$K = \frac{m}{L}\left(\frac{a}{k_1}-\frac{b}{k_2}\right) \tag{3}$$

式中:m 为汽车质量;L 为车辆轴距;a 为前轴到车辆质心的距离;b 为后轴到车辆质心的距离;k_1 和 k_2 分别为前后轮侧偏刚度。根据约束条件:$a_y \leqslant 0.4g$,得

$$u\frac{u/L}{1+Ku^2}\delta \leqslant 0.4g \tag{4}$$

从而前轮转角满足:

$$\delta \leqslant \delta_{\max} = 0.4g\frac{1+Ku^2}{u^2/L} \tag{5}$$

1.2.2 交通规则约束

除了车辆动力学约束外,还须考虑交通规则约束。典型的交通规则约束包括交通信号灯、车道线和速度限制等。当无人车在避障过程中换道时,必须遵守相关的交通规则约束。在仿真环境 TORCS 中主要考虑车道线限制与速度限制。车道线分为实线和虚线,在换道过程中实线不可穿越,而虚线可以。速度限制指最高速度不得超过 120km/h,读取无人车实时车速,通过反馈调节限制车辆速度。

1.3 控制变量

无人车的控制通过一组典型的执行器实现,即转向盘、加速踏板、制动踏板和变速器,变量定义如表 2 所示。

表 2 输出变量定义

名称	范围	描述
加速	[0,1]	加速踏板值(0 表示开度为 0,1 表示开度为 100%)
制动	[0,1]	制动踏板值(0 表示无制动,1 表示全制动)
挡位	-1,0,1,…,6	挡位值
转向	[-1,1]	转向盘转角值(-1 和+1 分别表示全右和全左)

2 无人车避障策略设计

2.1 奖赏设计

强化学习的奖赏函数将感知的状态映射为增强信号,用来评估动作的好坏。奖赏信号通常是标量,正值表示奖励,负值表示惩罚。奖赏函数的奖赏值与每一时刻车辆纵向速度呈正相关,当车辆发生碰撞或驶出车道线时给予额外的惩罚。本文中设计奖赏函数如下:

$$r_t = \begin{cases} -10, & \text{如果发生碰撞} \\ -20, & \text{如果驶出轨道} \\ v_x\cos\varphi - v_x\sin\varphi - v_x|\text{trackPos}|, & \text{其它} \end{cases} \tag{6}$$

当发生碰撞时,奖赏值设为-10,如果车辆行驶出道路,奖赏值设为-20。其它情况下,奖赏值的目的是最大化车辆纵向速度,最小化侧向速度。其中,φ 为车辆纵向与道路轴线的夹角,v_x 为车辆纵向速度,因此 $v_x\cos\varphi$ 表示车辆沿道路轴向方向的速度,$v_x\sin\varphi$ 表示车辆沿垂直于道路轴线方向的速度。考虑到交通规则约束,即无人车不能穿过道路边缘的实线,在公式中添加第 3 项 $v_x|\text{trackPos}|$,其中 $|\text{trackPos}|$ 表示车辆质心至路面投影点与道路轴线的归一化距离。

每次实验都包含许多学习回合,当车辆出界或者陷入局部最小值即速度小于设定的最小值时,结束该回合的学习。

2.2 探索策略

在强化学习中,适当的探索策略必不可少,尝试更多新的动作可避免陷入局部最优,即在某些特定的场景中总是采取相同的行动。奥恩斯坦-乌伦贝克过程是一种具有平均回归特性的随机过程,本文中用它来实现连续空间中的探索[6]:

$$\mathrm{d}x_t = \varepsilon(\mu - x_t)\,\mathrm{d}t + \sigma\mathrm{d}W_t \tag{7}$$

式中:ε 为变量趋于平均值的速度;μ 为均值;σ 为过程的波动程度。将该过程分别添加到转向、制动和加速中,其中加速的 μ 值需合理设置,避免出现车辆一直踩制动踏板不踩加速踏板的极限情况。

2.3 DDPG 算法改进

DDPG 结合了 DQN、确定性策略梯度算法 DPG(deterministic policy gradient)和演员-评论家算法(actor-critic methods),可解决强化学习中连续动作空间问题[3]。DQN 利用神经网络来逼近值函数,其参数是每层网络的权重,对值函数进行更新其实就是更新权重参数。DPG 算法采用异策略学习方法,行动策略采用随机策略,以保证足够的探索,评估策略采用确定策略,以减少动作空间的采样数量。DPG 采用演员-评论家算法框架,它通过分离策略函数和价值函数来降低学习难度,策略函数被称为演员,价值函数被称为评论家,演员根据当前的环境状

态产生一个动作,而评论家则对演员采取的动作进行评价。在本文中,评论家网络模型选择 SARSA(state action reward state action)算法,演员网络模型选择策略梯度算法。

在常规 DDPG 算法中,网络从回放缓冲区中随机采样进行离线训练。回放缓冲区是一个有限大小的缓冲区 R,元祖(s_t,a_t,r_t,s_{t+1})储存在缓冲区中并且根据探索策略随机采样。然而,由于有限的采样空间大小,且前期的样本学习效果一般,导致后期学习速率将变慢,且行为无法明显改善。因此,在学习的第二阶段增大样本空间,增加后期行为较好的样本,改进后的算法如表 3 所示。

表 3 改进的 DDPG 算法流程[3]

算法 1:改进的 DDPG 算法

随机初始化评论家网络 $Q(s,a|\theta^Q)$ 和演员网络 $\mu(s|\theta^\mu)$ 的权值 θ^Q 和 θ^μ

初始化目标网络 Q'和μ'的权值 $\theta^{Q'}\leftarrow\theta^Q,\theta^{\mu'}\leftarrow\theta^\mu$

初始化回放缓冲区 R

for 回合数=1,M do

在动作探索策略中初始化随机过程 N

接收初始观测状态 s_1

for $t=1,T$ do

根据当前策略和随机噪声选择动作 $a_t=\mu(s_t|\theta^\mu)+N_t$

执行动作 a_t 并且观察奖赏值 r_t,得到新的状态 s_{t+1}

保存元祖(s_t,a_t,r_t,s_{t+1})到缓冲区 R

从缓冲区 R 中随机采样生成 N 维数据库(s_i,a_i,r_i,s_{i+1})

if $t<T/2$ 则 $N\leftarrow N_0$(N_1 前期样本空间)

否则 $N\leftarrow N_1$(N_1 后期样本空间,且 $N_1>N_0$)

设 $y_i=r_i+\gamma Q'(s_{i+1},\mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$

通过最小化损失函数更新评论家网络:

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

使用样本的策略梯度更新演员网络:

$$\nabla_{\theta^\mu}J\approx\frac{1}{N}\sum_i\nabla_\alpha Q(s,a|\theta^Q)|_{s=s_i,a=\mu(s_i)}\nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$$

目标网络参数更新:

$$\theta^{Q'}\leftarrow\tau\theta^Q+(1-\tau)\theta^{Q'}$$

$$\theta^{\mu'}\leftarrow\tau\theta^\mu+(1-\tau)\theta^{\mu'}$$

式中,$\tau(0<\tau<1)$为参数更新速率

end for

end for

在迭代更新过程中,先积累经验回放缓冲区直到达到数据库指定个数,然后根据样本分别更新两个网络,先更新评论家网络,通过 loss 函数 L 更新参数 θ^Q。再通过评论家得到的 Q 函数相对于动作的梯度,然后应用演员网络更新公式更新参数 θ^μ。更新得到的参数 θ^Q 和 θ^μ 按照比例(通过参数 τ)更新到目标网络,这个目标网络会在下一步的训练中用于预测策略和 Q 函数值。

3 仿真实验

在 TORCS 中实现避障算法的仿真,TORCS 是一款高度可移植、跨平台的多车竞技、开源游戏平台,它拥有多玩家、多智能体、多赛道和多模式(练习模式、快速比赛、冠军赛等)[7]。TORCS 中含有不同的赛道,这些道路都包括静态障碍物和动态障碍物。静态障碍物包括道路边缘、树木和建筑物,动态障碍物是指移动的竞争车辆,无人车的目的是躲避这些障碍物,并尽快完成比赛。

根据障碍物种类,将无人车任务分为两类情况:情景一只包含静态障碍物,情景二包含静态和动态障碍物。无人车躲避这两种情景的障碍物,设定不同的参数,使车辆可学习到更好的策略。

3.1 情景一:静态障碍物

3.1.1 参数设定

首先,在没有其它车辆的道路上进行网络训练,无人车躲避的静态障碍物包括路边沿、树木和建筑物等。

演员网络和评论家网络均通过 Keras 构建。演员神经网络由两个隐含层组成,分别有 300 和 600 个单元。输出层根据变量的值域选择不同的激活函数:tanh 激活函数的输出范围是[-1,1],用于实现转向指令;sigmoid 激活函数的输出范围是[0,1],用于实现加速和制动指令。策略网络的学习速率是 10^{-4}。评论家网络包含两个隐藏层,分别有 300 和 600 个单元,学习速率是 10^{-3},神经网络训练大约 600 个回合。

3 个输出变量的随机噪声采用奥恩斯坦-乌伦贝克过程,作为适当的探索策略,且噪声随训练过程的增多逐渐减小,具体参数如表 4 所示。

表 4 随机噪声参数

动作	ε	μ	σ
转向	0.6	0.0	0.30
加速	1.0	0.6	0.10
制动	1.0	-0.1	0.05

ε 代表变量趋于平均值的速度,由于转向动作数量多,需提高转向的探索次数,将转向的 ε 设置为

0.6,加速和制动的 ε 设为 1.0。μ 代表噪声的平均值,转向有正有负,因此均值为 0;加速均值为 0.6,使车辆拥有初始速度;为了避免频繁制动,将制动 μ 值设为-0.1。σ 为噪声的波动程度,为提高转向的探索动作数量,将转向的 σ 值设为 0.30,同样地,为避免频繁制动,加速的 σ 值需大于制动的 σ 值,分别设为 0.10 和 0.05。

3.1.2 实验结果

选择 CG Speedway number 1 作为训练赛道,如图 1 所示。其长 2 057.56m,宽 15m,拥有道路的典型特征,包含静态障碍物和车道线,中途有 20 个坑洼障碍。

图 1 训练赛道

图 2 为每回合的训练步数。由图可知,在大约前 800 个回合,无人车的训练步数均小于 500 步,表明无人车触发训练终止条件,即无人车驶出道路或者陷入局部最小值(速度为 0)。大约在第 805 个回合,训练步数开始增加,普遍超过 500 步,甚至超过 2 000步,表明无人车学习到较好的策略,能完整地跑完整条赛道,并重复行驶多圈。在第 900 个回合左右,有几个回合的训练步数减小,是由于无人车尝试随机探索动作的原因。

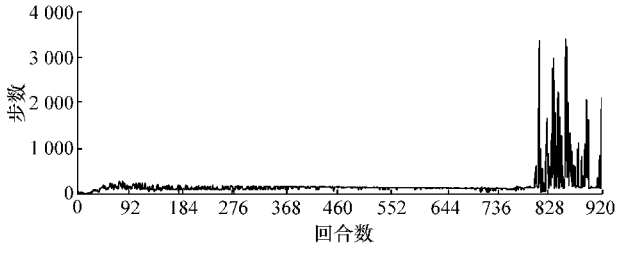

图 2 每回合步数

无人车每回合的累计奖赏值如图 3 所示,无人车的目的是通过不断学习来提高环境奖赏值,从而获得最大的奖赏值。因此,奖赏值越大,表明学习效果越好。对比可知,累计奖赏值的变化趋势与每回合步数变化趋势保持一致。

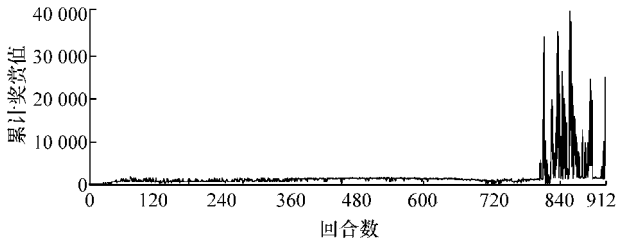

图 3 每回合累计奖赏值

每一回合的每一步平均奖赏值反映了学习进程的效果。学习过程如图 4 所示,平均奖赏值呈逐渐增加的趋势。大约经过 400 个回合,平均奖赏值大于 50,表明无人车学习到较好策略。在 720 个回合左右,平均奖赏值有一定的降低,且直到第 840 个回合左右才趋于稳定,保持一个较大的值。

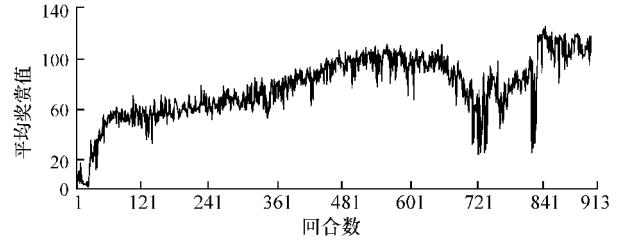

图 4 每回合平均奖赏值

在赛道 CG Speedway number 1 学习完成后,在另一赛道 CG track 2 上进行算法验证,验证赛道要比学习赛道更长,更复杂,长3 185.83m,宽 15m,在赛道中含有 16 个坑洼障碍,如图 5 所示。

图 6 为在验证赛道上每步的奖赏值。由图可以

图 5 验证赛道

看出,所有的奖赏值均为正值,表明无人车能成功跑完整条赛道,每一步都没有碰撞发生。

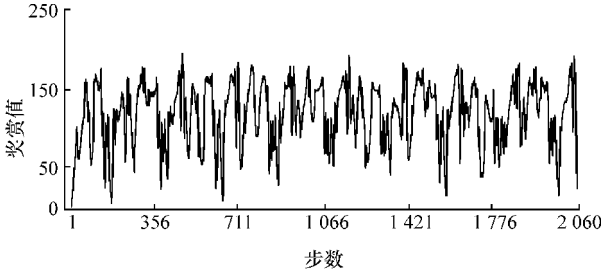

图 6　每步奖赏值

3.2　情景二:动态障碍物

3.2.1　参数设定

在静态障碍环境训练后,添加其它竞争车辆作为动态障碍物,无人车不仅要躲避静态障碍物,还要躲避动态障碍物。

演员网络和评论家网络的构建方法与情景一类似,有两个隐藏层,分别有 300 和 600 个神经单元。3 个输出变量的随机噪声参数如表 5 所示。

表 5　随机噪声参数

动作	ε	μ	σ
转向	0.6	0.0	0.30
加速	1.0	0.4	0.10
制动	1.0	0.1	0.05

参数的具体含义同表 4,区别是将加速度的 μ 值从 0.6 减到 0.4,制动的 μ 值从-0.1 变到 0.1,这是因为无人车需要更多的制动来躲避其它车辆。

3.2.2　实验结果

同样选择 CG Speedway number 1 赛道作为训练赛道,如图 7 所示。添加其它 5 辆车作为移动障碍物,由 AI 控制完成整条赛道的比赛,由于一些车辆比较极端,可能会撞到路边围栏,因此无人车应学习如何躲避这些事故车辆,不发生碰撞,获得最大奖励。在训练开始阶段,无人车无法有效躲避其它车辆,会发生碰撞,有时也会撞到道路边缘。经过多回合训练,逐渐提升性能,最终学会躲避静态障碍物和动态障碍物。

每回合的步数如图 8 所示。在实验开始阶段,每回合的步数小于 100,无人车发生碰撞,重新开始训练。大约经过 500 个回合后,步数显著增加,表明无人车学到较好策略。

图 7　训练赛道

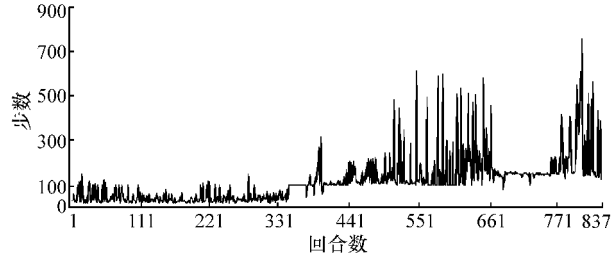

图 8　每回合步数

每回合的累计奖赏值如图 9 所示。在前 400 个回合,累计奖赏值较小,这与回合中的学习步数少有关。在 500 个回合左右,学习步数增多,累计奖赏值也变大。在 700 个回合左右,由于学习步数没有明显变化,累计奖赏值也保持一定值,随后随着步数的增加,累计奖赏值也增加。可以看出,累计奖赏值的变化趋势与学习步数的变化趋势保持一致。

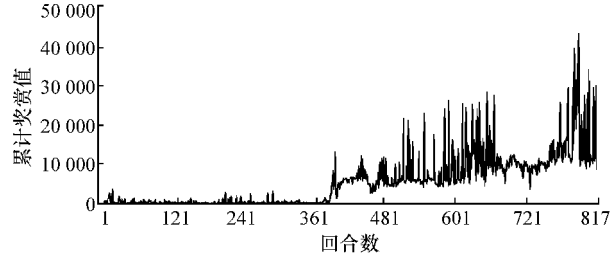

图 9　每回合累计奖赏值

每回合的平均奖赏值如图 10 所示。前 400 个回合,平均奖赏值都比较低,无人车处于学习状态。大约 400 个回合后,平均奖赏值开始增加,并且逐渐趋于稳定,表明无人车学习到较好策略。

当无人车在训练赛道上表现优异时,在赛道 CG Track 2 上进行验证,同样添加其它 5 辆车作为移动障碍物,如图 11 所示。

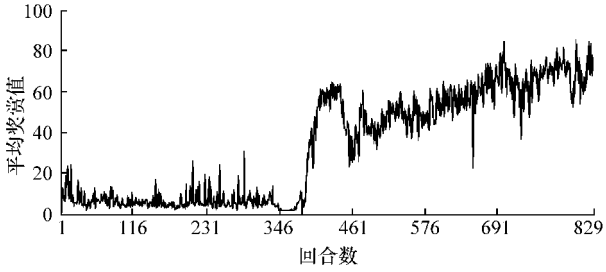

图 10 每回合平均奖赏值

图 11 验证赛道

每一步的奖赏值如图 12 所示,所有的奖赏值均为正值,表明没有碰撞发生。

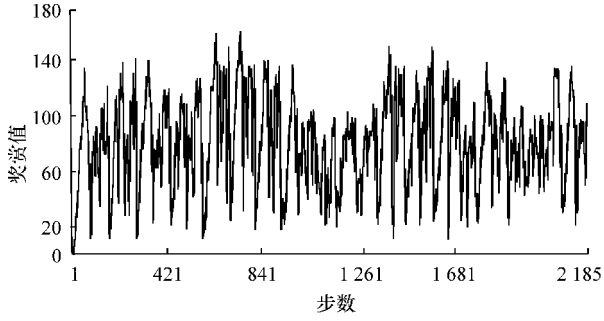

图 12 每步奖赏值

4 结论

本文中将深度确定策略梯度算法应用于无人车避障策略,使无人车执行连续动作。算法中考虑车辆动力学约束和交通规则约束,使输出动作更合理有效。将障碍物分为静态障碍物和动态障碍物两大类,分别通过学习过程和测试过程来验证算法的有效性。首先在学习轨道上训练,经过足够多的训练回合后,在其它赛道进行验证学习效果。结果表明,该算法在无人车仿真平台 TORCS 中表现良好。

参考文献

[1] SALLAB A E, ABDOU M, PEROT E, et al. Deep reinforcement learning framework for autonomous driving[J]. Electronic Imaging, 2017(19):70-76.

[2] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540):529-533.

[3] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[J]. Computer Science, 2015, 8(6):A187.

[4] XU Liangzheng, UIAO Chengyong, ZHANG J W. Analysis of dynamics stability of the tractor/full trailer combination vehicle and simulation for its control system[J]. Computer Simulation, 2003, 20(12):107-100.

[5] 王树凤,张大伟.车速与前轮转角的极限关系分析[J].机械设计与制造,2017(s1).

[6] 杨会会,宁丽娟.非线性漂移的 Fokker-Planck 方程的近似非定态解[J].物理学报,2013,62(18):38-45.

[7] XIONG X, WANG J, ZHANG F, et al. Combining deep reinforcement learning and safety based control for autonomous driving [J/OL]. https://arxiv.org/ftp/arxiv/papers/1612/1612.00147.pdf.