中国科技期刊卓越行动计划项目入选期刊

控制与决策

CONTROL AND DECISION

子目标驱动DQN算法的无人车狭窄转弯环境导航

耿玺钧, 崔立堃, 熊高, 刘知阳

引用本文:
耿玺钧,崔立,熊高,刘知阳. 子目标驱动DQN算法的无人车狭窄转弯环境导航[J]. 控制与决策, 2024, 39(11): 3637−3644.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2023.1513

子目标驱动DQN算法的无人车狭窄转弯环境导航

耿玺钧, 崔立堃†, 熊 高, 刘知阳

(陕西理工大学 机械工程学院，陕西 汉中 723000)

摘 要: 针对无人车在狭窄的转弯工作环境下,传统导航存在无法构建地图或所构建地图障碍物膨胀半径过大以及定位和控制存在误差,从而导致无人车与障碍物相撞,无法有效完成导航任务的问题,首先,通过将A*算法所生成的路径进行离散化,周期性选取路径点作为深度强化学习算法的目标点的方法,设计子目标驱动DQN算法,并基于此建立深度神经网络;然后,采用软件搭建狭窄的转弯环境,使用所提出子目标驱动DQN算法、无子目标驱动的DQN算法、DDPG算法、SAC算法分别对无人车进行训练,通过对比4种算法的收敛速度、执行步数以及导航成功率,验证所提出子目标驱动DQN算法在完成狭窄转弯环境导航任务时,效果最好;最后,将所提出算法的训练结果移植到全新的、空间更小、弯数更多的测试场景中进行测试,表明无人车能够顺利完成导航任务,从而验证所提出子目标驱动DQN算法的高扩展性.

关键词: A*算法；路径离散化；子目标驱动DQN算法；无人车；狭窄环境；导航

中图分类号: TP391.9 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2023.1513

引用格式: 耿玺钧, 崔立堃, 熊高, 等. 子目标驱动DQN算法的无人车狭窄转弯环境导航 [J]. 控制与决策, 2024, 39(11): 3637-3644.

Navigation in narrow turning environment of unmanned vehicle based on subgoal-driven DQN algorithm

GENG Xi-jun, CUI Li-kun†, XIONG Gao, LIU Zhi-yang

(College Mechanical Engineering，Shaanxi University of Technology，Hanzhong 723000，China)

Abstract: To address the problem of traditional navigation about unmanned vehicles in narrow turning work environments, such as the inability to construct maps or the construction of maps with excessively large obstacle expansible radii, as well as errors in positioning and control, resulting in collisions with obstacles and ineffective completion of navigation tasks, a method combining the A* algorithm and deep reinforcement learning is proposed. The path generated by the A* algorithm is discretized, and periodically selected path points are used as target points for the deep reinforcement learning algorithm. A subgoal-driven DQN algorithm is designed, and on this basic a neural network is established. The narrow turning environment is constructed using Gazebo software, and the unmanned vehicle is trained using the subgoal-driven DQN algorithm, DQN algorithm without subgoals, DDPG algorithm, and SAC algorithm. By comparing the convergence speed, execution steps, and navigation success rate, it is demonstrated that the subgoal-driven DQN algorithm performs best in completing the navigation task in narrow turning environments. The training results of the subgoal-driven DQN algorithm are transferred to a new test scenario with smaller space and more turns, and the test verifies that the unmanned vehicle can successfully complete the navigation task, proving the high scalability of the subgoal-driven DQN algorithm.

Keywords: A* algorithm；path discretization；sub-target driven DQN algorithm；unmanned vehicle；narrow environment；navigation

0 引 言

自动驾驶和无人车技术迅速发展,国内外众多学者对其进行了大量研究,并将其应用于各领域. 在应用方面,常采用传统的分层方法,即感知[1]、规划[2]和控制[3]. 然而,传统的分层方法存在一些限制,导致该方法无法单独完成任务: 首先,规划和控制的实现需要提前对工作环境进行采样,并基于采样数据构建栅格化地图[4],但是,在特殊的工作环境(如火灾现场、管

收稿日期: 2023-10-29；**录用日期:** 2024-01-26.

基金项目: 陕西省自然科学基金基础研究计划项目(2023-JC-YB-018).

责任编委: 易建强.

†通讯作者. E-mail: lekuncui@sina.com.

道作业等)中,由于无法提供地图数据,从而无法完成规划和控制任务;然后,为了避免无人车与障碍物碰撞,所构建的地图障碍物半径大于真实环境中的障碍物半径,导致地图中可访问空间减少,尤其是在狭窄的工作空间和转弯处,使得无人车的路径规划失败;最后,定位和控制等技术均存在不可忽视的误差[5],这也会导致无人车无法计算出合理路径,从而无法完成导航任务.

近年来,深度强化学习(DRL)[6-7]迅速发展,为解决上述问题提供了新的思路和方法. 基于值函数算法和策略梯度算法[8-9]的DRL将神经网络与传统强化学习相结合,成功解决了强化学习中的“维数灾难”问题[10],使得无人车能够应对复杂环境. 过去几年里,研究学者们开发了多种强化学习算法,如深度Q网络(DQN)[11]、深度确定性策略梯度(deep deterministic policy gradient, DDPG)[12]、软动作评论家(soft actor-critic, SAC)[13]以及近端策略优化(proximal policy optimization, PPO)[14]等算法,并将它们应用于虚拟游戏环境和现实世界的机器博弈[15]、无人车[16-17]、控制优化[18]、自动驾驶[19]、目标定位[20]等领域. 深度强化学习也在狭窄环境中得到了应用,如文献[21]验证了基于DRL的方法可提高智能体在高度拥挤环境中避开障碍物的性能,但是存在训练环境过于简单和训练效率低的问题;文献[22]则在狭窄、曲折的动态避障环境中,分别实现了离散动作和连续动作的静态避障仿真,但是由于DQN算法只能输出离散动作,导致无人车在避障过程中的动作和轨迹不够平滑.

为了解决上述问题,本文基于A*算法[23]与深度强化学习相结合的方法设计子目标驱动DQN算法,并构建深度神经网络,使用Gazebo[24]软件搭建实验场景,分别通过所提出子目标驱动DQN算法、无子目标驱动的DQN算法、DDPG算法、SAC算法对无人车进行训练,分析它们的训练结果,并将效果较好的所提出子目标驱动DQN算法训练结果移植到全新的更复杂的仿真环境中进行测试分析,验证所提出算法的高效性和可扩展性,为无人车在狭窄转弯环境中的导航提供一种解决方案.

1 问题描述和解决方案

狭窄工作环境下可用空间有限,使得无人车在完成路径规划任务时具有一定的挑战性. 在此工作空间下导航,易导致无人车与障碍物碰撞、无法找到最佳路径等问题. 有限的空间可能使得精确测量距离和角度变得更加困难,无法进行智能、精准、高效的路径规划[25-26]. 图1为RVIZ可视化工具[27]. 图1中:障碍点为狭窄转弯位置;红色箭头引出的位置为无人车目标位置和朝向;蓝色箭头引出的浅蓝色阴影部分为膨胀层[28],膨胀层半径设置越大,无人车可通过空间越小,膨胀层半径设置越小,无人车行驶过程中与障碍物的安全距离越小. 通常情况下,会设置较大的膨胀半径来避免无人车与障碍物发生碰撞. 这样当行驶环境为狭窄空间时,膨胀层的存在会造成没有足够的空间允许无人车通过.

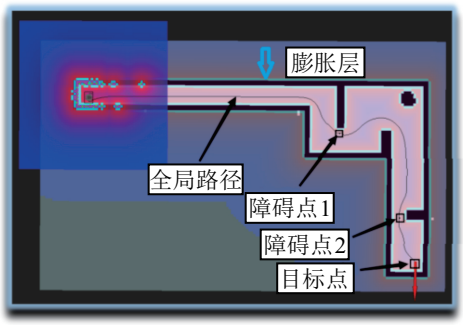


图1 无人车导航过程中地图可视化

针对上述问题,本文提出将A*算法与DQN算法相结合的方案,如图2所示. 24维激光雷达数据(DIST 0-DIST 23)、无人车当前位置与目标点的角度

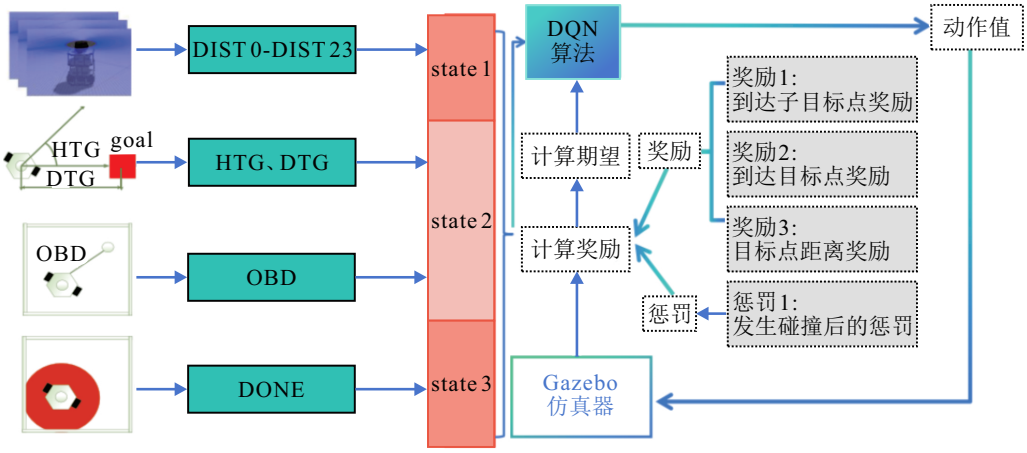


图2 无人车狭窄转弯环境导航方案

(HTG)和距离(DTG)、最近障碍物的距离(OBD)和无人车是否到达目标点(DONE)作为状态值输入DQN算法的神经网络,神经网络根据当前状态输出动作值,Gazebo仿真器根据输出的动作更新无人车位置,计算奖励值. 具体实现如下:在奖励函数中,引入A*算法规划离散的路径点作为子目标,用于驱动DQN算法,从而引导无人车朝着目标点前进,因此称之为子目标驱动的DQN算法(DQN_sub). 所提出方法中无人车沿A*算法规划路径前进,A*算法轨迹的平滑性已在文献[23]中被验证,因此所提出方法可保证导航轨迹平滑.

2　DQN算法

2.1　DQN算法的状态设计

在强化学习中,状态(state)是用来描述环境的关键信息,状态的选择对于深度强化学习任务的成功非常重要,本文设计状态包括24线激光雷达数据(DIST 0-DIST 23)、航向角(HTG)、当前位置到目标点位置的距离(DTG)、无人车与最近障碍物间的距离(OBD)、是否碰撞和到达目标点(DONE)等共28个状态量,如图2所示.

激光雷达数据(DIST 0-DIST 23)提供关于周围环境的信息. 在无地图导航的深度强化学习研究中,一般使用激光雷达扫描数据作为状态空间,但是低线数的激光雷达收集的数据较少,难以理解测量对象的特点,只适用于简单场景[29];而高线数激光雷达的数据密度较高,造成了数据处理复杂的问题[30]. 基于上述原因和本文的实验环境,这里选择使用24线激光雷达.

航向角(HTG)描述了无人车的朝向,有助于在导航过程中调整姿态,避免碰撞并正确导航至目标点,因此是状态空间中必不可少的一部分.

当前位置到目标点位置的距离(DTG)提供了无人车与目标点间的距离信息,对于路径规划和导航非常重要,无人车需要确定自身位置与目标点间的距离,以选择合适的行动和路径.

无人车与最近障碍物间的距离(OBD)提供了关于周围环境中最近障碍物的信息,可帮助无人车进行避障决策,避免与障碍物发生碰撞.

是否碰撞和到达目标点(DONE)状态量表示当前状态是否达到了终止条件,即无人车是否发生碰撞或成功到达目标点. 用于判断是否需要终止当前回合并开始下一回合的训练.

基于上述状态量可完整描述狭窄转弯中导航环境,若缺少,则会导致无人车信息不足、环境建模困难,从而对环境学习产生偏差,影响无人车的决策.

2.2　DQN算法的动作设计

首先,设计计算角速度(ang_vel)方法,即

$$\text{ang\_vel} = ((\text{action\_size} - 1)/2 - \text{action}) \times V \times 0.5. \tag{1}$$

其中:ang_vel为角速度;action_size为动作的总个数,取值为5;action为一个整数,范围为1～action_size. 通过action的取值来选择不同的角速度,即通过设定action的大小可决定角速度的正负;最后,将标准化值乘以最大角速度(V,固定为1.5 rad/s)来获得实际的角速度值,线速度恒定为0.15 m/s. 这种设计选择的合理性如下.

1) 角速度的离散化:通过将角速度的取值范围离散化为action_size个不同的动作,将连续的控制问题转化为离散的动作选择问题来简化问题的复杂性.

2) 动作空间的控制和表达能力:式(1)可灵活地控制角速度的范围和间隔,具体如下:当导航环境较为空旷时,action_size的数值可设为较大值,以增加无人车的灵活性;当导航环境较狭窄时,action_size的数值可设为较小值,以保证无人车能够安全导航. 本文设计action_size的数值为5.

3) 算法的可解释性和易实现性:式(1)可使得算法更具可解释性. 通过使用整数的action值,可直观地理解每个动作所对应的具体角速度值.

此外,这种设计选择也易于编码实现,便于后续无人车在狭窄环境中进行导航实验.

2.3　DQN算法的决策网络和目标网络设计

DQN算法是Q-learning[31]算法的改进版,使用一个决策网络和一个目标网络来设计机器学习模型[32]:决策网络用于实时决策,而目标网络用于更新决策网络的参数. 决策网络的输入为环境状态,输出为动作;而目标网络的输入为环境状态和动作,输出为期望的奖励. 通过不断更新决策网络的参数,DQN算法可学习如何在给定环境状态采取最优动作,从而获得最大的奖励. 神经网络结构如图3所示. 网络结构共有3层,分别为输入层、中间层和输出层:输入层为28个神经元,与前文所述28个状态量保持一致;中间层共有两个隐藏层,第1个隐藏层神经元个数为128,第2个隐藏层神经元个数为128,隐藏层神经元个数是一个重要的超参数,可直接影响神经网络的容量和表达能力,增加隐藏层神经元个数可增加网络的复杂多元和表达能力,但是也会增加计算负担和出现过拟合问题,根据实验需求,这里选择128个神经元个数作为隐藏层的大小;输出层神经元个

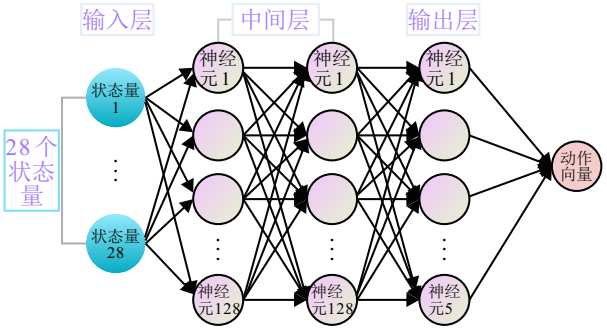


图 3 神经网络结构

数为5,与式(1)中动作数量相同. 使用Adam (adaptive moment estimation)优化器,用于在训练过程中有效地更新神经网络参数.

2.4 DQN算法的奖励函数设计

奖励设计如下所示:

$$R=\begin{cases}R_1=100/\text{dist Everstep};\\ R_2=\begin{cases}\text{Collision}\ -200,\\ \text{Subgoal 20},\\ \text{goal 2 000}.\end{cases}\end{cases}\tag{2}$$

其中: R为总奖励; R_1为每回合内每步的稀疏奖励; dist为无人车与目标点的距离,距离越近,奖励越大. R_2为最终奖励,由3部分组成: 1) 负奖励,即惩罚,当无人车发生碰撞时,获得−200的惩罚,判定条件为当无人车与最近障碍物间的距离小于0.13 m时,判定无人车与障碍物发生碰撞; 2) 到达子目标点奖励,若无人车到达A*算法所提供的子路径点,则得到20的奖励,判定条件为当无人车与子目标点间的距离小于0.2 m时,判定无人车到达子目标点位置; 3) 若无人车到达最终目标点,则得到2 000的奖励,判定条件为当前位置与最终目标点位置的距离小于0.2 m时,判定无人车到达目标点位置.

终止条件设置: 通过设置终止条件可帮助无人车避免陷入局部最优解导致的无限循环. 设计当无人车到达目标位置、与障碍物碰撞、执行该事件的步数超过2 000时的3个终止条件. 在实际应用中,需在有限的计算资源和时间内完成训练,因此,需确定一个合理的执行步数. 若执行步数过大,则会导致浪费计算资源和时间; 若执行步数过小,则会导致无人车没有足够的步数完成导航任务. 因此执行步数一般作为超参数,在实验开始前根据实验环境复杂度确定,越复杂的环境则需要越大的执行步数. 本文根据实验环境确定执行步数为2 000.

2.5 DQN算法数学过程

下式为贝尔曼方程,是DQN算法的核心思想:

$$Q(s_t,a_t)\leftarrow Q(s_t,a_t)+\alpha[r_t+\gamma\max_a Q(s_{t+1},a)-Q(s_t,a_t)].\tag{3}$$

其中: $Q(s_t,a_t)$为动作价值函数; α为学习率; γ为折扣因子; r_t为奖励; s_t为当前时刻的状态; a_t为当前时刻做出的动作; s_{t+1}为在状态s_t下,执行动作a到达的状态. 智能体与环境进行交互,使用深度神经网络模拟Q值函数(状态动作价值函数); Q值函数与神经网络的每层权重相对应,即

$$Q(s,a;\theta)=Q(s,a).\tag{4}$$

这里: θ为神经网络参数, $Q(s,a)$值函数的更新过程实质上为对神经网络参数θ的更新. 当神经网络参数θ确定时,值函数$Q(s,a)$即可确定. 设置Q目标网络,计算TD误差,使用神经网络对Q值网络进行近似时,对其中的参数θ采用梯度下降的方法进行处理,有

$$\theta_{t+1}=\theta_t+\alpha(r+\gamma\max_{a'}Q(s,a;\theta)-Q(s,a;\theta))\nabla Q(s,a;\theta),\tag{5}$$

其中$\gamma\max_{a'}Q(s,a;\theta)$为TD目标. 式(5)中目标网络参数与决策网络参数相同,导致训练结果不稳定. 为了解决此问题,通过将决策网络参数实时更新,目标网络参数经N(具体数值如表1所示: 目标网络更新频率)轮迭代后将决策网络中的参数赋值给目标网络得到,因此将式(5)改写为

$$\theta_{t+1}=\theta_t+\alpha(r+\gamma\max_{a'}Q(s,a;\theta^-)-Q(s,a;\theta))\nabla Q(s,a;\theta).\tag{6}$$

这里: θ^-为目标网络中的参数, θ为决策网络中的参数.

表 1 DQN算法的超参数

序号	主要参数	量值
1	EPISODES = 3 000	回合数
2	$\gamma=0.9$	奖励折扣因子
3	LR = 0.001	学习率
4	$N=10$	目标网络更新频率
5	MEMORY_CAPACITY = 2 000	经验池容量
6	N_ACTIONS = 5	动作空间
7	N_STATES = 28	状态空间
8	BATCH_SIZE = 128	提取样本数量

2.6 无人车运动学模型

实验环节使用的无人车为两轮差速运动无人车,其运动学模型如图4所示. 图4中: ICR (instantaneous center of rotation)为无人车绕该点旋转的中心点,其运动学模型包括正运动学模型和逆运动学模型. 这两个运动学模型分析过程如下.

根据物理学约束$v=w\times r$,求得ICR处角速度w为

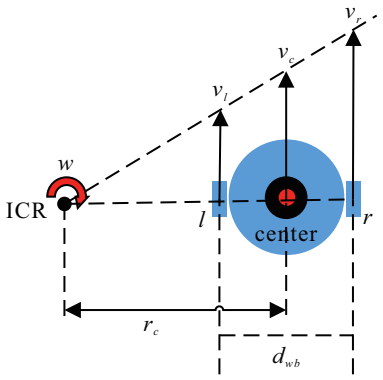


图 4　无人车运动学模型

$$w=\frac{v_c}{r_c}=\frac{v_r}{r_c+d_{wb}/2}=\frac{v_l}{r_c-d_{wb}/2}. \tag{7}$$

其中: d_{wb} 为无人车外圆直径, r_c 为当前时刻无人车中心点 center 转向半径, $[v_c, w]^{\mathrm{T}}$ 为中心点 center 的速度和角速度. 由式(7), 角速度 w 表示为

$$w=(v_r-v_l)/d_{wb}. \tag{8}$$

这里角速度 w 是有方向的: 当 $v_l<v_r$ 时, $w>0$; 反之, 则 $w<0$. 通过式(7)可得到点 center 的线速度 v_c 和左右驱动轮的速度 $[v_l, v_r]$ 的关系, 如下所示:

$$v_c=(v_l+v_r)/2. \tag{9}$$

结合式(8)和(9), 可求得点 center 的转向半径, 即

$$r_c=\frac{v_c}{w}=\frac{(v_l+v_r)d_{wb}}{2(v_r-v_l)}. \tag{10}$$

结合式(8)～(10), 得到两轮差速运动无人车正运动学模型, 具体如下所示:

$$\begin{bmatrix} v_c \\ w \end{bmatrix}=\begin{bmatrix} \dfrac{v_r+v_l}{2} \\ \dfrac{v_r-v_l}{d_{wb}} \end{bmatrix}=\begin{bmatrix} \dfrac{1}{2} & \dfrac{1}{2} \\ \dfrac{1}{d_{wb}} & -\dfrac{1}{d_{wb}} \end{bmatrix}\begin{bmatrix} v_r \\ v_l \end{bmatrix}. \tag{11}$$

结合式(7)和(10), 两轮差速无人车逆运动学模型为

$$\begin{bmatrix} v_{\mathrm{r}} \\ v_l \end{bmatrix}=\begin{bmatrix} v_c+\dfrac{d_{wb}}{2} \\ v_c-\dfrac{d_{wb}}{2} \end{bmatrix}=\begin{bmatrix} 1 & \dfrac{d_{wb}}{2} \\ 1 & -\dfrac{d_{wb}}{2} \end{bmatrix}\begin{bmatrix} v_c \\ w \end{bmatrix}. \tag{12}$$

根据上述运动学模型即可实现所提出算法对无人车的控制: 由式(1)中所描述, 所提出算法可提供线速度和角速度; 通过式(12)两轮差速运动无人车逆运动学模型, 即可求得无人车左右两驱动速度, 完成对无人车底盘的控制.

3　实验仿真测试

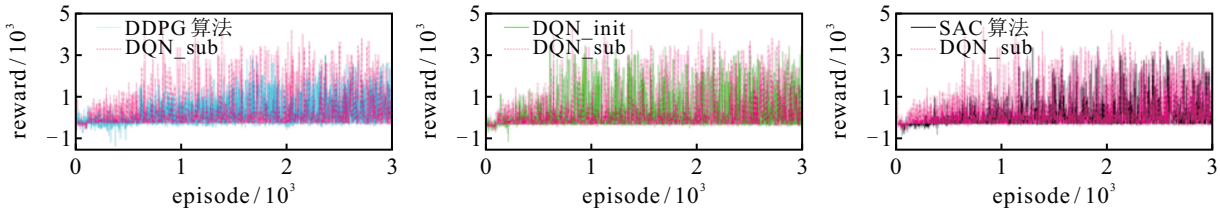
3.1　实验场景

基于 Ubuntu 18.04 系统、i5-10210 uCPU、CPU 频率 1.8 GHz、Python 3.6 实验平台, 使用 Gazebo 软件搭建实验场景用来进行实验验证, 使用 Pytorch 1.4.0 搭建神经网络结构. 首先, 在一个模拟的狭窄转弯环境中训练无人车, 命名为训练场景, 如图 1 所示, 使用所提出算法, 在训练场景中学习如何有效地避免障碍物并保存学习结果.

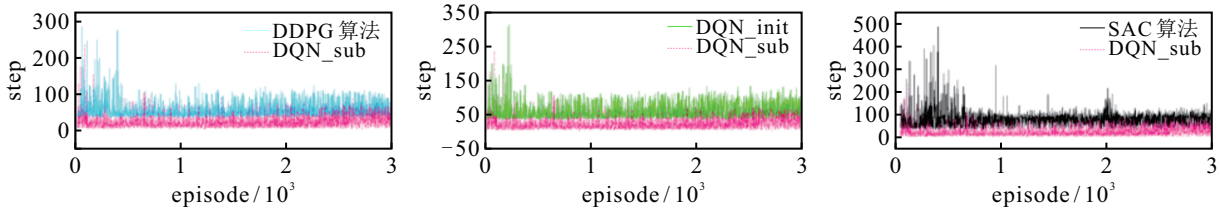
3.2　算法比较

基于上述训练场景, 使用所提算法(DQN_sub)、未使用所提出算法(DQN_init)、DDPG 算法和 SAC 算法分别在训练环境中进行训练, DQN 算法的超参数如表 1 所示.

训练的对比结果如图 5 所示. 图 5(a) 为 4 种算法每回合奖励变化趋势对比, 由图 5(a) 可知: 二者在训练初期奖励基本为负数, 这是因为无人车根据探索和利用原理, 在训练初期会进行探索, 易撞墙, 导致奖励为负. 随着训练次数的增加, 无人车可根据当前状态选择动作价值较大的动作, 从而不易撞墙, 最终到达目标点, 奖励值变大, 因此图 5(a) 中奖励值呈增大趋势. 分析图 5(a) 可知: DDPG 算法、DQN_init、SAC



(a)　奖励训练结果对比



(b)　步数训练结果对比

图 5　算法训练结果对比

算法和DQN_sub分别在1 500回合、700回合、1 250回合、600回合时奖励值趋于平稳,神经网络模型完成收敛,可见DQN_sub的模型收敛速度最快. 图5(b)为4种算法每回合执行步数变化趋势对比,执行步数为无人车从出发点到最终目标点运行的步数,执行步数越小表明无人车路径越短,因此是反映算法性能的重要指标. 由图5(b)可知: 在训练初期4类算法的执行步数均较大,这是因为在狭窄环境中使用强化学习时,墙体障碍物与无人车距离较近,无人车为了防止与墙体发生碰撞,导致无人车陷入局部最优解,因此在这个阶段的执行步数较大; 随着训练次数的增加,无人车可根据当前状态选择最优动作,算法的执行步数逐渐减少,最终完成收敛; 4种算法收敛后DQN_sub算法执步数最少,表明所提出方法在较短的步骤内完成了导航任务,具有较高的动作选择准确性,这在狭窄环境中导航是非常重要的,若动作选择产生偏差,则会与墙体发生碰撞.

经上述训练过程后得到4个算法的训练模型,将训练模型在训练场景中测试100回合,得到如表2所示的不同算法的成功率和平均步长结果. 通过导航成功率和平均步长的指标,可评估不同算法在导航任务中的性能: 较高的导航成功率表示模型在到达目标位置方面的准确性较高; 而较低的平均步长表示模型完成导航任务所需的步数较少,即效率较高. 分析表2结果可知: 所提出DQN_sub算法在此场景下导航成功率最高,平均步长最短,验证了所提出算法的高效性. 图6为所提出DQN_sub算法在训练场景下控制无人车行驶的完整过程. 图6中共有9张子图,分别表示无人车在不同时间点的位姿,该训练场景中共有两处狭窄转弯障碍点(详情如图1所示),无人车从图6的子图1所示位置出发,经子图2所示的位置后,到达子图3所示位置并进入第1个狭窄转弯环境,由子图4可知,无人车可通过子图3所示的狭窄转弯环境,表明了所提出算法的有效性. 经子图4～子图7所示的位置后,进入第2个狭窄转弯点,最后成功到达子图9所示的目标点位置.

3.3 可扩展性测试实验

为了验证所提出DQN_sub算法驱动无人车在狭窄环境中导航的可扩展性,基于上述训练实验的实验结果进行测试并搭建了如图7所示的测试场景: 该测试场景相较于训练场景有4处狭窄的转弯地点,分别位于图中“凸”字4个拐角处,导航的难度进一步提高. 将训练场景中的神经网络模型在测试场中测试100回合,得到如表3所示的测试场景测试中的成功率和平均步长结果. 通过与训练场景下测试结果(如表2所示)比较可知: DDPG导航成功率降低了1%,DQN_init算法导航成功率降低了3%,SAC算法导航成功率降低了1%,所提出DQN_sub算法导航成功率不变. 这是因为新的仿真场景与训练时的环境不同,前3种算法的神经网络模型无法准确地理解和适应新的环境特征,导致无法准确快速地理解和处理这些情况,从而导致成功率下降. 而所提出DQN_sub算法由于引进了子目标点,即使在新的环境中,所提出算法仍然只需跟随子目标点前进即可,因此即使在新的环境中,所提出算法的成功率依然不发生降低,表明所提出算法具有很高的可扩展性. 此外,通过与训练场景下测试结果比较可知: 4类算法的平均步长均有所上升,这是由于测试场景地图长度较大,算法所需完成导航的执行步长增加,导致平均步长上升. 图7为所提出DQN_sub算法在测试场景下控制无人车行

表 2 训练场景测试中的成功率和平均步长

算法	成功率/%	平均步长
DDPG	90	66
DQN_init	89	64
SAC	91	68
DQN_sub	95	57

表 3 测试场景测试中的成功率和平均步长

算法	成功率/%	平均步长
DDPG	89	78
DQN_init	86	72
SAC	90	79
DQN_sub	95	68

图 6 训练场景下无人车行驶过程

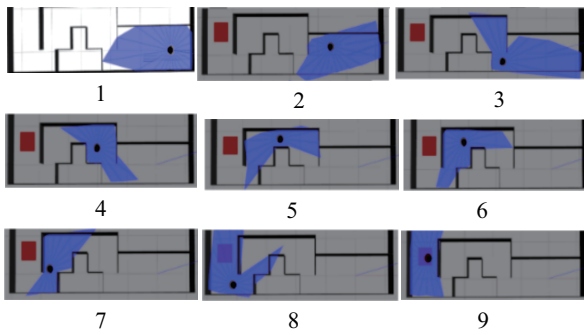


图 7 测试场景下无人车行驶过程

驶的完整过程. 图7中共有9张子图,分别表示无人车在不同时间点的位姿,无人车从图7的子图1所示位置出发,经子图2所示的位置后,到达子图3所示位置,并进入狭窄转弯环境,由子图4可知,无人车可通过子图3所示的狭窄转弯环境. 子图5～子图8情况与上述过程相似,最终到达如子图9所示的目标点. 在行驶过程中未出现碰撞或无人车无法完成导航任务而原地旋转的情况.

4 结　论

本文针对传统的路径规划算法在狭窄工作环境下完成路径规划时由于膨胀半径的存在而无法完成导航的问题,设计了子目标驱动DQN算法,并在Gazebo仿真环境中构建了狭窄转弯工作环境,与DDPG算法、SAC算法以及DQN_init进行了比较,验证了所提出算法的有效性,并将所提出算法训练结果移植到全新的更复杂的狭窄环境中进行测试分析,分析了所提出算法的可扩展性,分析结果表明:

1) DDPG算法、DQN_init、SAC算法和所提出DQN_sub分别在1 500回合、700回合、1 250回合、600回合时奖励值趋于平稳,所提出DQN_sub的模型收敛速度最快,执行步数最少;

2) SAC算法导航成功率为90 %, DDPG导航成功率为89 %, DQN_init导航成功率为91 %,所提出DQN_sub导航成功率为95 %,所提出DQN_sub的模型导航成功率最高;

3) 相较于训练场景,在空间更小、拐弯更多的测试场景中,采用所提出DQN_sub算法时,无人车在行驶过程中未出现碰撞或原地旋转的情况,顺利完成导航任务,验证了所提出DQN_sub算法的高扩展性.

参考文献(References)

[1] Nagata J J, Abad F M, Giner J R G B. Augmented reality and mobile pedestrian navigation with heritage thematic contents perception of learning[J]. RIED: Revista Iberoamericana de Educación a Distancia, 2017, 20(2): 93-118.

[2] Li X, Serlin Z, Yang G, et al. A formal methods approach to interpretable reinforcement learning for robotic planning[J]. Science Robotics, 2019, 4(37): eaay6276.

[3] Mallem A, Douak F, Walid B, et al. Firefly algorithm optimization of manipulator robotic control based on fast terminal sliding mode[J]. Journal of Automation, Mobile Robotics and Intelligent Systems, 2022,16(4): 44-52.

[4] 徐武, 高寒, 王欣达, 等. 改进ORB-SLAM2算法的关键帧选取及地图构建研究[J]. 电子测量技术, 2022, 45(20): 143-150.
(Xu W, Gao H, Wang X D, et al. Research on key frame selection and map construction of improved ORB-SLAM2 algorithm[J]. Electronic Measurement Technology, 2022, 45(20): 143-150.)

[5] Hwang Y M, Lee S Y, Sim I, et al. Positioning error reduction techniques for precision navigation by post-processing[J]. IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences, 2017, E100.(A10): 2158-2161.

[6] 闫超, 相晓嘉, 徐昕, 等. 多智能体深度强化学习及其可扩展性与可迁移性研究综述[J]. 控制与决策, 2022, 37(12): 3083-3102.
(Yan C, Xiang X J, Xu X, et al. A survey on scalability and transferability of multi-agent deep reinforcement learning[J]. Control and Decision, 2022, 37(12): 3083-3102.)

[7] Li G F, Zhou W Y, Lin S Y, et al. On-ramp merging for highway autonomous driving: An application of a new safety indicator in deep reinforcement learning[J]. Automotive Innovation, 2023, 6(3): 453-465.

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

[9] 董豪, 杨静, 李少波, 等. 基于深度强化学习的机器人运动控制研究进展[J]. 控制与决策, 2022, 37(2): 278-292.
(Dong H, Yang J, Li S B, et al. Research progress of robot motion control based on deep reinforcement learning[J]. Control and Decision, 2022, 37(2): 278-292.)

[10] 刘建伟, 高峰, 罗雄麟. 基于值函数和策略梯度的深度强化学习综述[J]. 计算机学报, 2019, 42(6): 1406-1438.
(Liu J W, Gao F, Luo X L. Survey of deep reinforcement learning based on value function and policy gradient[J]. Chinese Journal of Computers, 2019, 42(6): 1406-1438.)

[11] 刘潇, 刘书洋, 庄韫恺, 等. 强化学习可解释性基础问题探索和方法综述[J]. 软件学报, 2023, 34(5): 2300-2316.
(Liu X, Liu S Y, Zhuang Y K, et al. Explainable reinforcement learning: Basic problems exploration and method survey[J]. Journal of Software, 2023, 34(5): 2300-2316.)

[12] Mao H Y, Zhang Z C, Xiao Z, et al. Modelling the dynamic joint policy of teammates with attention multi-agent DDPG[C]. Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. Montreal, 2019: 1108-1116.

[13] Coraci D, Brandi S, Piscitelli M S, et al. Online implementation of a soft actor-critic agent to enhance indoor temperature control and energy efficiency in

buildings[J]. Energies, 2021, 14(4): 997.

[14] 孙爱红, 雷琦, 宋豫川, 等. 基于深度强化学习求解作业车间机器与AGV联合调度问题[J]. 控制与决策, 2024, 39(1): 253-262.
(Sun A H, Lei Q, Song Y C, et al. Deep reinforcement learning for solving the joint scheduling problem of machines and AGVs in job shop[J]. Control and Decision, 2024, 39(1): 253-262.)

[15] Silver D, Huang A, Maddison C J, et al. Mastering the game of go with deep neural networks and tree search[J]. Nature, 2016, 529: 484-489.

[16] Levine S, Pastor P, Krizhevsky A, et al. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection[J]. The International Journal of Robotics Research, 2018, 37(4/5): 421-436.

[17] Levine S, Finn C, Darrell T, et al. End-to-end training of deep visuomotor policies[J]. Journal of Machine Learning Research, 2016, 17(39): 1-40.

[18] Narasimhan K, Kulkarni T, Barzilay R. Language understanding for text-based games using deep reinforcement learning[C]. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, 2015: 1-11.

[19] Sallab A E, Abdou M, Perot E, et al. Deep reinforcement learning framework for autonomous driving[J]. Electronic Imaging, 2017, 29(19): 70-76.

[20] Caicedo J C, Lazebnik S. Active object localization with deep reinforcement learning[C]. IEEE International Conference on Computer Vision. Santiago, 2015: 2488-2496.

[21] Gu S X, Holly E, Lillicrap T, et al. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates[C]. IEEE International Conference on Robotics and Automation. Singapore, 2017: 3389-3396.

[22] 王大方. 基于深度强化学习的机器人导航研究[D]. 江苏: 中国矿业大学, 2019.
(Wang D F. Research on robot navigation based on deep reinforcement learning[D]. Jiangsu: China University of Mining and Technology, 2019.)

[23] Lai X, Li J H, Chambers J. Enhanced center constraint weighted A* algorithm for path planning of petrochemical inspection robot[J]. Journal of Intelligent & Robotic Systems, 2021, 102(4): 78.

[24] 孙翔龙, 梁彦刚. 基于Gazebo和PX4的强化学习训练仿真环境接口设计与实现[J]. 电子技术与软件工程, 2022(5): 68-71.
(Sun X L, Liang Y G. Design and implementation of reinforcement learning training simulation environment interface based on Gazebo and PX4[J]. Electronic Technology and Software Engineering, 2022(5): 68-71.)

[25] Lisowski J. Computer simulation of a game control model in a complex maritime traffic environment[J]. International Journal of Simulation Modelling, 2023, 22(3): 416-425.

[26] 张洪琳, 吴耀华, 胡金昌, 等. 一种基于改进冲突搜索的多机器人路径规划算法[J]. 控制与决策, 2023, 38(5): 1327-1335.
(Zhang H L, Wu Y H, Hu J C, et al. A multi-robot path finding algorithm based on improved conflict search[J]. Control and Decision, 2023, 38(5): 1327-1335.)

[27] Kam H R, Lee S H, Park T, et al. RViz: A toolkit for real domain data visualization[J]. Telecommunication Systems, 2015, 60(2): 337-345.

[28] 龚志力, 谷玉海, 朱腾腾, 等. 一种基于机器人操作系统的代价地图自适应膨胀半径设置方法[J]. 科学技术与工程, 2021, 21(9): 3662-3668.
(Gong Z L, Gu Y H, Zhu T T, et al. A method for setting costmap adaptive inflation radius based on robot operating system[J]. Science Technology and Engineering, 2021, 21(9): 3662-3668.)

[29] Gao Y, Ji Z, Wu J, et al. Hierarchical reinforcement learning-based mapless navigation with predictive exploration worthiness[C]. IEEE International Conference on Mechatronics and Automation. Harbin, 2023: 636-643.

[30] Newaz A A R, Alam T. Hierarchical task and motion planning through deep reinforcement learning[C]. Proceedings of the 5th IEEE International Conference on Robotic Computing. Taichung, 2021: 100-105.

[31] Wen S H, Lv X H, Lam H K, et al. Probability dueling DQN active visual SLAM for autonomous navigation in indoor environment[J]. Industrial Robot: The International Journal of Robotics Research and Application, 2021, 48(3): 359-365.

[32] 韩红桂, 徐子昂, 王晶晶. 基于Q学习的多任务多目标粒子群优化算法[J]. 控制与决策, 2023, 38(11): 3039-3047.
(Han H G, Xu Z A, Wang J J. A Q-learning-based multi-task multi-objective particle swarm optimization algorithm[J]. Control and Decision, 2023, 38(11): 3039-3047.)

作者简介

耿玺钧(1998−), 男, 硕士生, 从事深度强化学习路径规划的研究, E-mail: 1564237684@qq.com;

崔立堃(1976−), 男, 副教授, 博士, 从事无人驾驶汽车控制算法、无人驾驶的规划与控制等研究, E-mail: lekuncui@sina.com;

熊高(1999−), 男, 硕士生, 从事多智能体强化学习的研究, E-mail: 1971785852@qq.com;

刘知阳(1999−), 男, 硕士生, 从事深度学习目标检测的研究, E-mail: 727990720@qq.com.