DOI:10.13878/j.cnki.jnuist.2023.02.009

马丽新[1]　刘磊[1]　刘晨[1]

安全屏障机制下基于 SAC 算法的机器人导航系统

摘要

为了提高移动机器人自主导航系统的智能化水平和安全性,设计了安全屏障机制下基于 SAC(Soft Actor-Critic)算法的自主导航系统,并构建了依赖于机器人与最近障碍物距离、目标点距离以及偏航角的回报函数.在 Gazebo 仿真平台中,搭建载有激光雷达的移动机器人以及周围环境.实验结果表明,安全屏障机制在一定程度上降低了机器人撞击障碍物的概率,提高了导航的成功率,并使得基于 SAC 算法的移动机器人自主导航系统具有更高的泛化能力.在更改起终点甚至将静态环境改为动态时,系统仍具有自主导航的能力.

关键词

移动机器人;SAC 算法;安全屏障机制;激光雷达;自主导航;Gazebo

中图分类号 TP242.6

文献标志码 A

0　引言

近几年,具有自主导航功能的无人车已应用到日常生活中,如无人公交、无人网约巴士、无人配送车等.路径规划能力是衡量无人驾驶车辆是否可以自主导航的重要标准.传统的路径规划方法通常需要人为提取特征来获知环境信息,以完成对环境地图的绘制、移动机器人的定位以及路径规划,但在复杂环境下很难实现.而强化学习[1]不依赖于环境模型以及先验知识,还可自主在线学习,近年来逐渐成为移动机器人自主导航的研究热点[2].

随着计算机硬件水平的提升,深度学习的任意逼近能力得以更大化地发挥,许多深度学习与强化学习相结合的算法被提出,如深度 Q 网络[3](Deep Q-Network,DQN)、深度确定性策略梯度[4](Deep Deterministic Policy Gradient,DDPG)等.2018 年,Haarnoja 等[5-6]针对无模型深度学习算法训练不稳定、收敛性差、调参困难等问题,提出一种基于最大熵强化学习框架的软更新行动者-评论家算法(Soft Actor-Critic,SAC).最大熵的设计使得算法在动作的选择上尽可能地随机,既避免收敛到局部最优,也提高了训练的稳定性.另外,通过在 MuJoCo 模拟器上一系列最具挑战性的连续控制任务中与 DDPG、每步梯度更新都需要一定数量新样本的近似策略优化[7]等算法做对比实验,凸显了 SAC 算法性能的高稳定性和先进性.

在路径规划领域,基于 SAC 算法的机器人自主导航相关研究已引起学者的广泛关注.Xiang 等[8]将 LSTM 网络融入到 SAC 算法中用于移动机器人导航,以 360°的 10 维激光雷达信息和目标信息为输入,输出连续空间的线速度和角速度,验证了改进后的算法在训练过程中平均回合回报(累计回报/累计回合数)的增长速度较快.de Jesus 等[9]同样基于稀疏的 10 维激光雷达数据,不过激光范围是正前方 180°,以雷达信息、目标方位、动作为网络输入,并创建了两个不同的 Gazebo 环境,在每个环境中都对 SAC、DDPG 两种深度强化学习技术在移动机器人导航中的应用效果做了比较,从导航成功率等方面验证了 SAC 算法的性能优于 DDPG 算法.

移动机器人的安全性在自主导航过程中是不可忽视的.近些年有学者通过在训练环节增加安全机制,来降低危险动作被选择的概率,进而促进机器人特定任务的完成.代珊珊等[10]针对无人车探索的安全问题,提出一种基于动作约束的软行动者-评论家算法(Constrained

收稿日期 2022-06-01

资助项目 国家自然科学基金(61773152).

作者简介

马丽新,女,硕士生,研究方向为强化学习、自主体控制.1623406486@qq.com

刘磊(通信作者),男,博士,教授,研究方向为强化学习理论研究与应用、多智能体系统分析与控制.liulei_hust@163.com

1　河海大学　理学院,南京,210098

Soft Actor-Critic,CSAC),将其用于载有摄像头的无人车车道保持任务上.动作约束具体表现为当无人车转动角度过大时,回报会相对较小;当无人车执行某动作后偏离轨道或发生碰撞时,该动作将被标记为约束动作并在之后的训练中合理约束.

基于以上启发,考虑到SAC算法在移动机器人路径规划领域的应用尚未被充分研究,本文以提高机器人自主导航系统的智能化水平和安全性为出发点,设计出一种安全屏障机制下基于SAC算法的机器人导航系统.首先对SAC算法以及仿真平台Gazebo做了简单描述.然后搭建导航系统,包括机器人状态、动作、回报函数的定义以及安全屏障机制的设计.最后在Gazebo中训练模型,通过静态环境和动态环境等5组共300回合的对比测试验证了安全屏障机制在提高导航成功率上的有效性.

1 SAC算法及仿真平台

SAC算法[5-6]是第一个基于最大熵强化学习框架的离线Actor-Critic算法,要寻找的最优策略既要满足累计期望回报最大化,也要满足熵最大化,即:

$$\pi^* = \arg\max_{\pi} \sum_t \mathrm{E}_{(s_t,a_t)\sim\rho_\pi}[r(s_t,a_t) + \alpha H(\pi(\cdot\mid s_t))], \tag{1}$$

其中α决定熵与回报的相对重要性,控制策略的随机性大小,ρ_π表示给定s_t,a_t时下一状态s_{t+1}的概率密度,H表示策略的熵.最大化策略的熵促使自主体增大对环境的探索性,避免算法收敛到局部最优,也使训练更加稳定.另外,SAC算法的软更新体现在目标值函数的参数通过Poliak-averaging方法[11]更新,即$\bar{\theta}\leftarrow\tau\theta+(1-\tau)\bar{\theta}$,其中$0<\tau\ll1$,$\theta$表示训练值函数,$\bar{\theta}$表示目标值函数.

本文选择的机器人仿真平台是一个开源的3D物理仿真平台——Gazebo.在缺乏实物机器人的情况下,Gazebo工具的存在一定程度上降低了机器人导航研究的门槛[12].在真实环境中,机器人(如树莓派小车)通过GPIO接口获取传感器的信息和发布速度指令,以驱动小车按计划行驶.但在仿真环境Gazebo中,取而代之的是机器人操作系统(Robot Operating System,ROS).ROS是一个适用于机器人的开源的元操作系统[13].针对不同的程序语言,ROS提供了不同的接口.rospy是Python版本的ROS客户端库.后续的算法实现中用到了rospy模块的众多函数,如与环境重置相关的wait_for_service(´gazebo/set_model_state´)、rospy.ServiceProxy(´gazebo/set_model_state´,SetModelState),与里程计相关的rospy.wait_for_message("/odom",Odometry),与发布速度相关的rospy.Publisher(´/cmd_vel´,Twist,queue_size=5),与雷达相关的rospy.wait_for_message("/scan",LaserScan,timeout=10),等等.

2 系统设计

2.1 状态空间

借助传感器雷达和里程计获取的信息来定义机器人的状态(图1).信息处理的具体方法如下:

1)雷达信息处理

机器人载雷达以约5次/s的频率在360°的范围内均匀发射120条射线,即相邻两条射线之间的角度为3°.另外,为了使得仿真的激光雷达更贴近真实场景中的传感器,特地在模拟的雷达数据输出前加入高斯噪声.雷达在Gazebo中发射射线的情形如图2所示.

为了方便状态的定义,将360°的雷达信息按角度平均划分为3个区域,即每个区域的角度为120°,具体划分方法如图3所示.对120维的雷达数据二次加工,筛选出“高质量”信息,即每个区域内40条射线中最短射线的长度(d_i,$i=1,2,3$),并做以下处理:

$$d_{i,\mathrm{label}} = \begin{cases} d_i, & d_i < 1.5, \\ 1.5, & d_i \geqslant 1.5. \end{cases} \tag{2}$$

注1 机器人是否会撞击到障碍物,很大程度上取决于其与最近障碍物的距离.考虑移动机器人周围360°无死角的120维障碍物信息,并从中选择最近的3个障碍物距离,与文献[8-9]中通过10维的雷达信息获取障碍物信息相比,能更有效地表示机器人周围的关键障碍物情况.

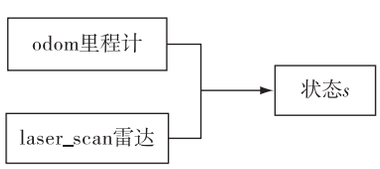

图1 状态定义

Fig. 1 Definition of state

2)里程计信息处理

机器人的行驶方向是否偏离终点是衡量导航效果好坏的重要标准.通过里程计获取机器人当前位置及行驶方向,再结合终点的方位信息,得到机器人的偏航角$0°<\varphi\leqslant180^o$.信息处理流程如图4所示.

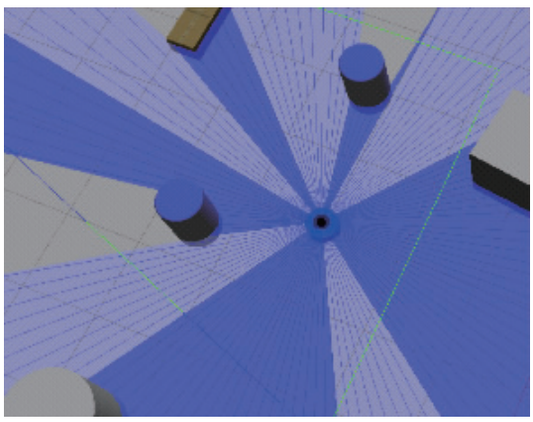

图 2　激光雷达发射射线的情形

Fig. 2　Radiation emitted by lidar

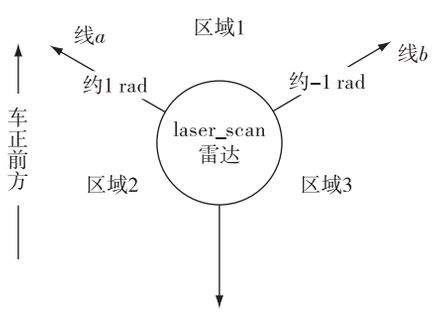

图 3　雷达射线区域划分

Fig. 3　Radar ray area division

利用处理后的雷达和里程计信息,定义机器人的状态空间为

$$S=\{(\varphi,d_{1,\text{label}},d_{2,\text{label}},d_{3,\text{label}})\mid 0°<\varphi\leqslant 180°,\ d_{i,\text{label}}\leqslant 1.5\}. \tag{3}$$

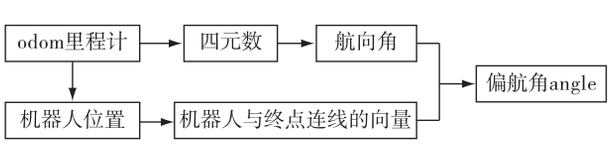

图 4　里程计信息处理

Fig. 4　Processing of odometer information

2.2　动作空间

为了降低决策的复杂性,本文机器人的线速度均大于 0,即只考虑机器人前进、左转以及右转的情形,不包括后退,具体速度大小范围如表 1 所示.考虑到太高频的发送动作指令可能会导致动作发布通道的信息堵塞,而动作执行时间(即机器人接收到动作指令后执行该动作的时长)过长有可能增加机器人撞击障碍物的概率,这里定义每个动作的执行时长为 $t=0.2$ s.因此,得到动作空间:

$$V=\{(v,w,t)\mid 0\leqslant v\leqslant 0.22,\ -2\leqslant w\leqslant 2\}. \tag{4}$$

表 1　动作空间设置

Table 1　Setting of action space

	线速度 v/(m/s)	角速度 w/(rad/s)
最小值	0	-2
最大值	0.22	2

2.3　安全屏障机制

简单地说,强化学习是试错的学习,自主体从错误的行为中吸取教训,进而做出正确的决策.其实,在自主体做决策之前适当地采取一些措施来降低下一步决策错误的概率有利于减少强化学习模型训练的回合数,甚至增强模型在其他环境中的适用性.比如,在机器人即将撞上障碍物时,若在安全范围内选择动作,则大概率会避免机器人撞击障碍物的情况,进而增大机器人成功导航到终点的可能性.后续的实验验证了该说法.因此,本文设计的机器人将在安全范围内选择动作,即当网络输出的角速度所属区域 $i(i=1,2,3)$ 的 d_i 小于安全阈值 0.5 时,重新选择角速度使得角速度所属区域在另外两个区域内,具体流程如图 5 所示.

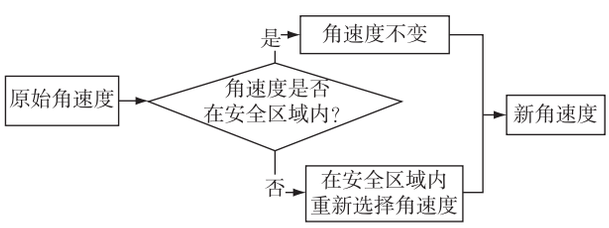

图 5　安全屏障机制工作原理

Fig. 5　Principle of security barrier mechanism

当原始角速度 w_{old} 不在安全区域内时,要重新选择角速度,表 2 描述了在安全区域内重新选择角速度的具体数值.例如:当只有区域 1 是安全时,便选择区域 1 的角度平分线,即 0 rad 为 w_{new};当只有区域 2 是安全时,选择 1 rad 和 2 rad 的均值作为 w_{new}(其中 1 rad 表示区域 1 与 2 的分隔线 a 方向(图 3),2 rad 表示最大角速度);当只有区域 2 与 3 是安全时,随机选择 1.5 rad 或-1.5 rad 作为 w_{new}(其中 1.5 rad 是线 a 方向(图 3)与最大角速度 2 rad 的均值,-1.5 rad 是线 b 方向(图 3)与最小角速度-2 rad 的均值).

注 2　在机器人距离障碍物较近时,启动安全屏障机制,使得机器人最大程度远离障碍物.与文献[10]中基于动作效果来决定是否约束该动作的设计

相比,本文的安全屏障机制主要基于机器人状态设定,使得机器人随时根据状态来决定是否启动安全屏障保护,增强了机器人的智能化水平和导航过程中的安全性.

表2　不同安全区域及其对应的 w_{new}

Table 2　Different safe areas and their corresponding w_{new}

安全区域	w_{new}
1	0
2	1.5
3	-1.5
1,2	0.5
1,3	-0.5
2,3	1.5,-1.5

注3　若网络输出的动作在安全屏障机制下有了变化,则后续存储在记忆池中的动作是新动作.这一做法有助于增大机器人在通过记忆池学习导航算法时选择安全动作的概率,使得机器人在安全屏障机制下学习最优策略.

2.4　稠密回报函数

深度强化学习算法训练过程的本质是神经网络在回报函数的引导下学习状态与动作之间的映射,因此根据任务目标来设计回报函数有助于提高算法的收敛速度.本文中机器人的任务是从起点无碰撞地自主导航到目标点.这里基于机器人是否到达终点、是否撞到障碍物、与终点距离、偏航角大小4个方面的信息来设计回报函数.

1) 对任务中的重点:导航成功(target)以及无碰撞(collision)分别设计对应的回报函数:

$$r_{\text{target}}=\begin{cases}100, & \text{到达终点},\\ 0, & \text{未到达终点}.\end{cases}\tag{5}$$

$$r_{\text{collision}}=\begin{cases}-100, & \text{撞到障碍物},\\ 0, & \text{未撞到障碍物}.\end{cases}\tag{6}$$

2) 考虑到机器人在每回合训练中大多时间处于无碰撞且未抵达终点的状况,为了让机器人在该状况下能够即时了解到动作效果的好坏,根据机器人与终点之间的距离(distance)以及偏航角(angle,φ)分别设计不同的回报函数:

$$r_{\text{distance}}=d_{\text{rt,last}}-d_{\text{rt}}+1-\frac{d_{\text{rt}}}{d_{\text{rto}}},\tag{7}$$

$$r_{\text{angle}}=1-\frac{\varphi}{90}.\tag{8}$$

其中 d_{rt} 是当前时刻机器人与终点的距离,$d_{\text{rt,last}}$ 是上一时刻机器人与终点的距离,d_{rto} 是起点与终点的距离,φ 是偏航角.

由式(5)—(8)可得回报函数式(9).无论机器人做出什么动作,环境都会给予回报.这样稠密的回报函数有助于即时向机器人反馈动作的执行效果.

$$R=\begin{cases}100, & \text{车到达终点},\\ -100, & \text{车撞到障碍物},\\ d_{\text{rt,last}}-d_{\text{rt}}+1-\dfrac{\varphi}{90}+1-\dfrac{d_{\text{rt}}}{d_{\text{rto}}}, & \text{其他情况}.\end{cases}\tag{9}$$

注4　根据状态中的元素 φ 来设计 r_{angle},增强了机器人状态与回报函数之间的相关性,为值估计和动作生成提供了可辨识依据.

2.5　导航系统整体框架

导航系统整体框架如图6所示.其中策略网络和值网络均含4个隐层,每个隐层神经元个数为256,激活函数为ReLU函数,优化器选择Adam.

3　模型训练

以下实验均在虚拟机VMware Workstation 16 Pro内的Ubuntu18.04系统中进行,Python语言版本是2.7.17.

3.1　训练场景

在Gazebo中搭建环境如图7所示,接下来将在该环境中训练SAC算法使得机器人从起点(3,3)成功走到终点(6,7).

3.2　参数设置

SAC模型的超参数设置如表3所示.

表3　SAC算法超参数设置

Table 3　Values of SAC's hyperparameters

超参数	取值
批大小	256
每回合最大步数	800
记忆池容量	50 000
折扣率	0.99
学习率	0.000 3

3.3　训练过程

在2 000个回合的训练过程中,机器人导航成功率变化情况如图8所示.其中

$$\text{成功率}=\frac{\text{累计导航成功回合数}}{\text{累计回合数}}.\tag{10}$$

由图8可以看出,刚开始训练时机器人导航成功率比较高,750个回合后,成功率还稳定在85%以上.

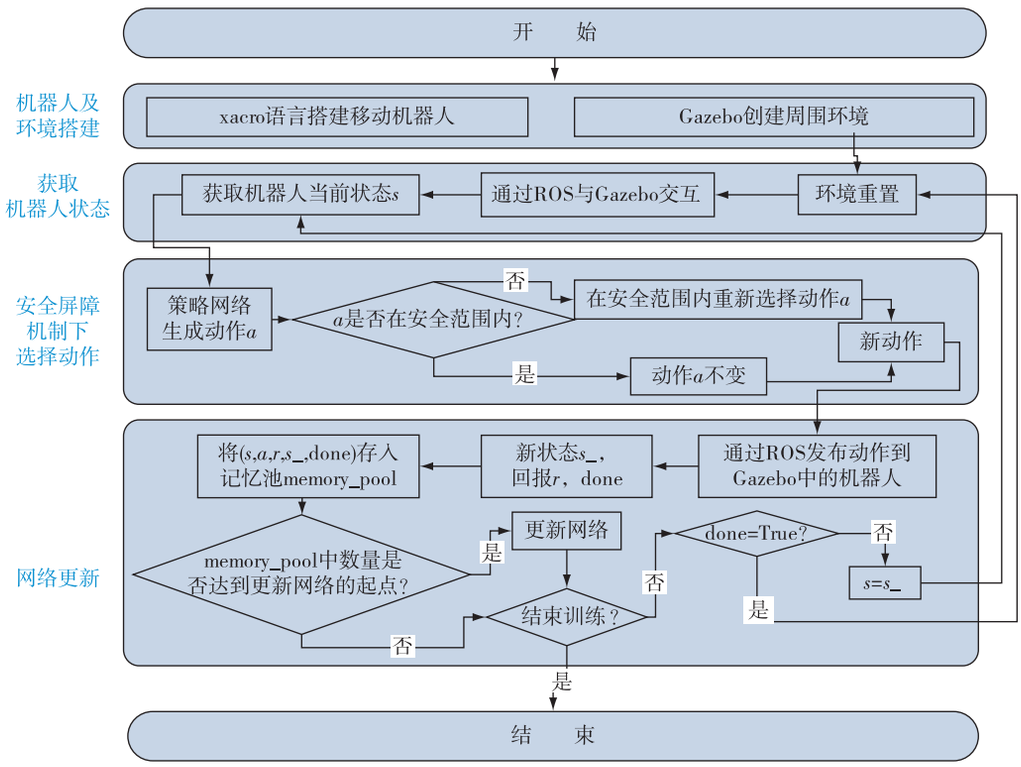

图 6　系统整体框架

Fig. 6　Framework of overall system

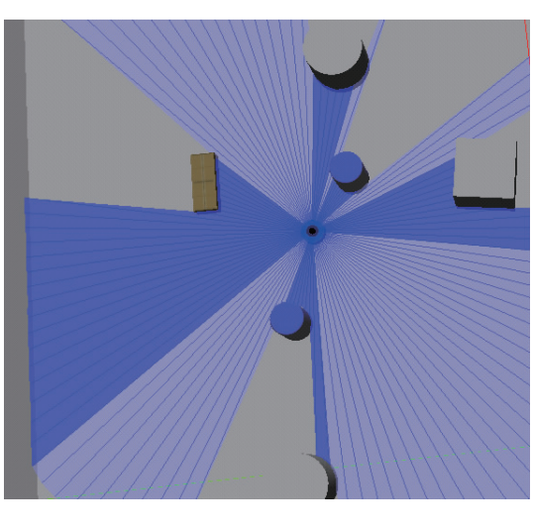

图 7　训练环境

Fig. 7　Training environment

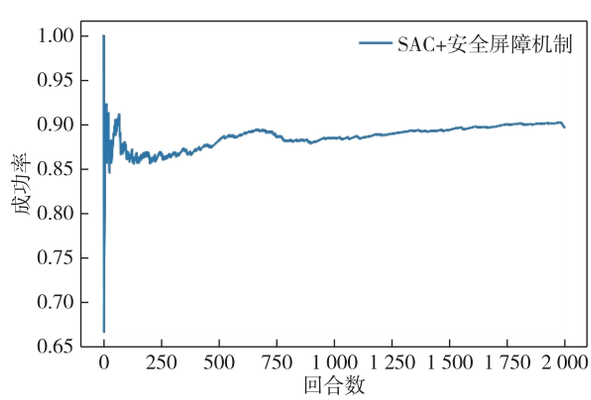

图 8　导航成功率

Fig. 8　Success rate of navigation

图 9 为每回合的平均回报变化曲线,其中

$$\text{每回合的平均回报}=\frac{\text{当前回合累计回报值}}{\text{当前回合的总步数}}. \tag{11}$$

由图 9 看出,在前 600 回合内,每回合的平均回报大致在 0 到 2 之间波动,而 600 回合后,大部分回合的平均回报的变化范围则缩小到 1 到 2 之间.

在图 10 中,500~1 000 回合是累计平均回报的快速上升期,之后累计平均回报仍在慢慢增大,表明模型正在稳中向好地演变.

$$\text{累计平均回报}=\frac{\text{累计回报值}}{\text{已训练的回合数}}. \tag{12}$$

4　模型效果测试

4.1　静态环境

为了多方位探测模型的效果,共进行 4 组不同的测试,且在每组测试中都将 SAC+安全屏障机制模

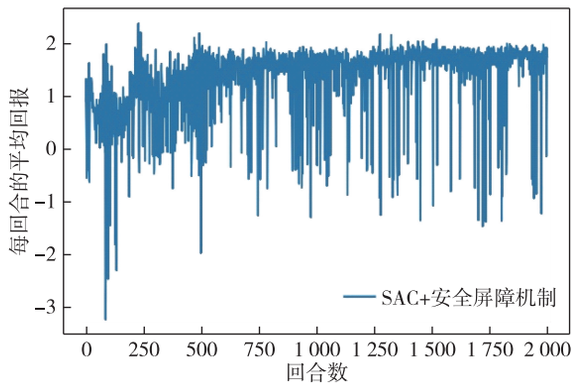

图 9　每回合的平均回报

Fig. 9　Average return per epoch

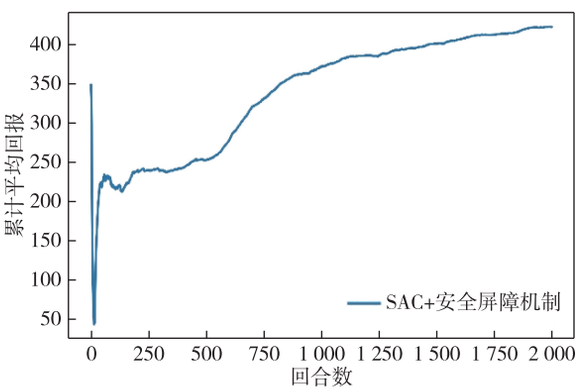

图 10　累计平均回报

Fig. 10　Cumulative average return

型(SAC+)效果和无安全屏障机制的 SAC 模型效果做对比.其中,测试 1 的环境、起点和终点与训练时的设置相同,测试 2 相对训练仅更改了终点,测试 3 相对训练更改了起点和终点,测试 4 的设置与模型训练时完全不同,不仅将环境变得相对复杂,还改变了起点和终点(图 11).详细测试条件配置及两种模型的成功率对比结果如表 4 所示.

由表 4 看出,在测试 3 中,两种模型的成功率均为 100%,在测试 1、2 中,SAC+安全屏障机制模型的成功率略高于后者,而在更改了环境的测试 4 中,SAC+安全屏障机制模型的成功率远高于 SAC 模型.

在 4 组测试中,两种模型的导航轨迹长度(即动作步数)对比如图 12—15 所示(点状表示该模型在当前回合导航失败).在测试 1 图 12 中,SAC+安全屏障机制模型的导航轨迹长度普遍低于 SAC 模型,而且 100 个回合无一失败,验证了 SAC+安全屏障机制模型的高效性和稳定性.在测试 2 图 13 中,两种模型均有导航失败的情况,但 SAC+安全屏障机制模型失败次数较少,且在轨迹长度与 SAC 模型相差不大的情况下波动相对较小,更加体现出前者的稳定性.在测试 3 图 14 中,虽然 SAC+安全屏障机制模型和 SAC 模型均无导航失败的回合,但是在大多数回合中前者导航的轨迹长度短于后者.在测试 4 图 15 中,两种模型的效果差距很大,在 SAC+安全屏障机制模型 50 次均导航成功时,SAC 模型仅成功导航 3 次,一定程度上凸显了前者在新环境的高适用度.

4.2　动态环境

根据表 4 中的模型测试结果,可以看出安全屏障机制下基于 SAC 算法的移动机器人自主导航系统在不同的静态环境中导航成功率均较高.为了更全面地探究训练模型对不同环境的泛化性以及鲁棒性,创建含有静态和动态障碍物的环境(图 16),再次测试模型的导航效果.

在动态环境图 16 中,物体 A 为动态障碍物,在点(3.5,5.5)与点(4.3,4.7)之间以约 0.062 m/s 的速度做匀速直线往返运动(图 16 中黄色虚线).模型测试条件配置及导航成功率如表 5 所示.由表 5 可知,本文设计的系统在动态环境中的导航成功率表

表 4　模型测试及结果对比 1

Table 4　Model test and result comparison 1

测试	环境	起点	终点	回合数	模型	成功次数	失败次数	导航成功率/%
1	训练环境	(3,3)	(6,7)	100	SAC	98	2	98
					SAC+安全屏障机制	100	0	100
2	训练环境	(3,3)	(9,4)	50	SAC	42	8	84
					SAC+安全屏障机制	46	4	92
3	训练环境	(2,4)	(6,3)	50	SAC	50	0	100
					SAC+安全屏障机制	50	0	100
4	新环境	(0,0)	(5,6)	50	SAC	3	47	67
					SAC+安全屏障机制	50	0	100

现虽然不及静态环境,但仍优于无安全屏障机制的导航系统,表明安全屏障机制在提高导航成功率方面具有积极作用.

图 17 为模型导航路径长度对比(点状表示该模型在当前回合导航失败).其中 SAC+安全屏障机制模型在第 1、12 回合导航的步数多于其他回合,是因为移动机器人为了躲避动态障碍物,选择了先绕过障碍物 B 再向终点前进的路径,体现了该导航系统的灵活性.

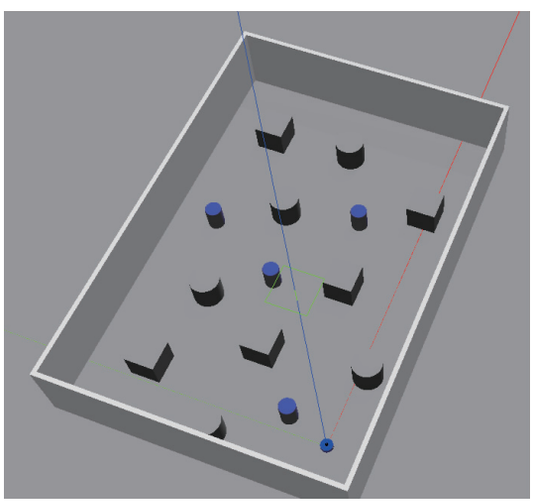

图 11　新环境

Fig. 11　New environment

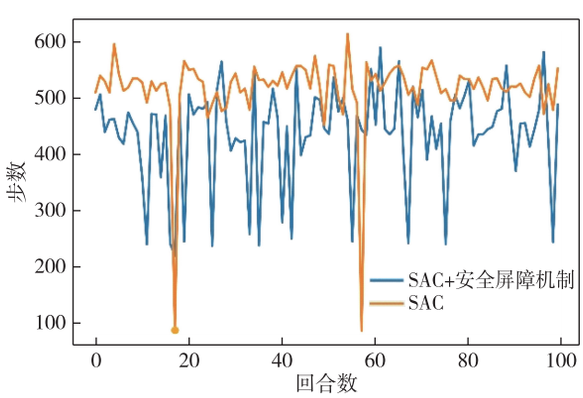

图 12　测试 1:路径长度对比

Fig. 12　Test1:comparison of path length

表 5　模型测试及结果对比 2

Table 5　Model test and result comparison 2

环境	起点	终点	回合数	模型	成功次数	失败次数	导航成功率/%
动态环境	(3,3)	(5,6)	50	SAC	28	22	56
				SAC+安全屏障机制	39	11	78

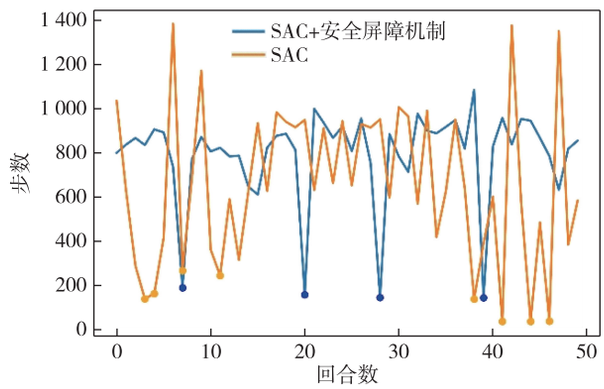

图 13　测试 2:路径长度对比

Fig. 13　Test2:comparison of path length

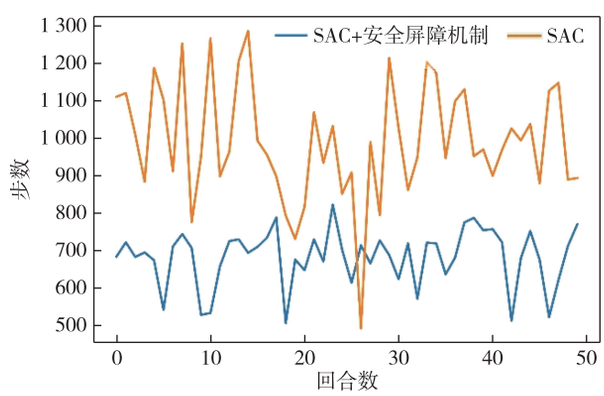

图 14　测试 3:路径长度对比

Fig. 14　Test3:comparison of path length

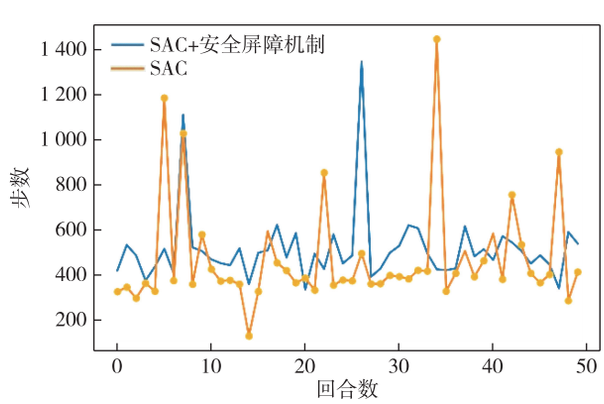

图 15　测试 4:路径长度对比

Fig. 15　Test4:comparison of path length

5　结论

本文在 Gazebo3D 仿真平台构建了基于安全屏障机制和 SAC 算法的移动机器人自主导航系统,通过静态和动态环境中的多组对比实验验证了安全屏障机制在提高机器人导航成功率方面的有效性.仿

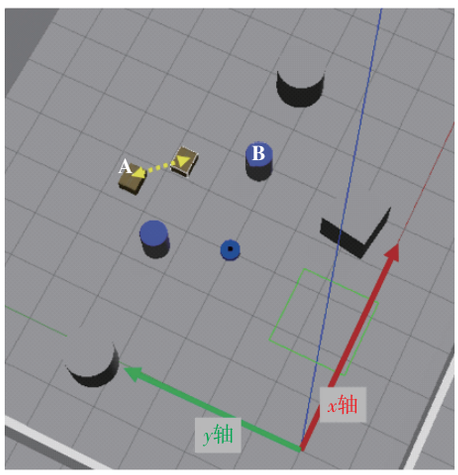

图 16　动态环境

Fig. 16 Dynamic environment

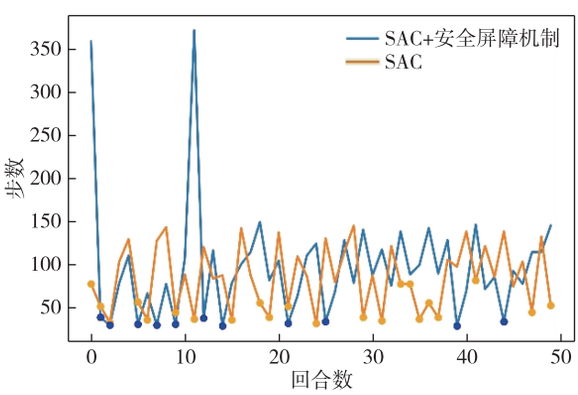

图 17　动态环境-路径长度对比

Fig. 17　Dynamic environment-comparison of path length

真使用的激光雷达只可扫描 360°的同一平面信息,因此只有当障碍物相对规则(如长方体形、圆柱形等)时才能比较准确地测出距离信息.未来可通过配置多个不同水平面的雷达或使用更高级的雷达来增大导航系统对障碍物形状的包容度,使得仿真环境更加贴近复杂的现实场景.

参考文献

References

[1] Sutton R S,Barto A G.Reinforcement learning:an introduction[J].IEEE Transactions on Neural Networks,1998,9(5):1054

[2] 刘志荣,姜树海.基于强化学习的移动机器人路径规划研究综述[J].制造业自动化,2019,41(3):90-92
LIU Zhirong,JIANG Shuhai.Review of mobile robot path planning based on reinforcement learning[J].Manufacturing Automation,2019,41(3):90-92

[3] Mnih V,Kavukcuoglu K,Silver D,et al.Playing atari with deep reinforcement learning[J].arXiv e-print,2013,arXiv:1312.5602

[4] Lillicrap T P,Hunt J J,Pritzel A,et al.Continuous control with deep reinforcement learning[J].arXiv e-print,2015,arXiv:1509.02971

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

[6] Haarnoja T,Zhou A,Hartikainen K,et al.Soft actor-critic algorithms and applications[J].arXiv e-print,2018,arXiv:1812.05905

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

[8] Xiang J Q,Li Q D,Dong X W,et al.Continuous control with deep reinforcement learning for mobile robot navigation[C]//2019 Chinese Automation Congress (CAC).November 22-24,2019,Hangzhou,China.IEEE,2019:1501-1506

[9] de Jesus J C,Kich V A,Kolling A H,et al.Soft actor-critic for navigation of mobile robots[J].Journal of Intelligent & Robotic Systems,2021,102(2):31

[10] 代珊珊,刘全.基于动作约束深度强化学习的安全自动驾驶方法[J].计算机科学,2021,48(9):235-243
DAI Shanshan,LIU Quan.Action constrained deep reinforcement learning based safe automatic driving method[J].Computer Science,2021,48(9):235-243

[11] Polyak B T,Juditsky A B.Acceleration of stochastic approximation by averaging[J].SIAM Journal on Control and Optimization,1992,30(4):838-855

[12] Koenig N,Howard A.Design and use paradigms for Gazebo,an open-source multi-robot simulator[C]//2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).September 28-October 2,2004,Sendai,Japan.IEEE,2004:2149-2154

[13] Quigley M,Gerkey B P,Conley K,et al.ROS:an open-source robot operating system[C]//ICRA Workshop on Open-Source Software,2009

Robot navigation system based on SAC with security barrier mechanism

MA Lixin[1]　LIU Lei[1]　LIU Chen[1]

1　College of Science,Hohai University,Nanjing　210098

Abstract　An autonomous navigation system was proposed based on Soft Actor-Critic under the security barrier

mechanism to improve the intelligence and security of mobile robot autonomous navigation system. The return function was designed based on distance between the robot and the nearest obstacle, the distance from the target point, and the yaw angle. On the Gazebo simulation platform, a mobile robot with lidar and its surrounding environment were built. Experiments showed that the security barrier mechanism reduced the probability of collision with obstacles to a certain extent, improved the success rate of navigation, and made the SAC-based mobile robot autonomous navigation system have high generalization ability. The system still had the ability of autonomous navigation when changing the origin and destination or even changing the environment from static to dynamic.

Key words mobile robot; soft actor-critic (SAC); security barrier mechanism; lidar; autonomous navigation; Gazebo