

doi:10.12068/j.issn.1005-3026.2025.12.20240094

## 基于深度强化学习的风险导向人群导航策略

姜杨<sup>1</sup>, 赵天祥<sup>2</sup>, 孙若怀<sup>2,3</sup>, 王雷<sup>4,5</sup>

(1. 东北大学 机器人科学与工程学院, 辽宁 沈阳 110169; 2. 东北大学 信息科学与工程学院, 辽宁 沈阳 110819;  
3. 沈阳新松机器人自动化股份有限公司, 辽宁 沈阳 110168; 4. 中煤科工(辽宁)具身智能科技有限公司, 辽宁  
沈阳 110168; 5. 中煤科工机器人科技有限公司, 辽宁 沈阳 110168)

**摘要:** 针对传统导航方法在遇到动态障碍物时出现的机器人冻结问题及动态避障效果不佳的问题, 提出一种基于深度强化学习的导航方法. 该方法的核心为风险感知模块和路径选择模块. 风险感知模块实时计算机器人与附近动态障碍物的碰撞概率, 从而使机器人优先规避高风险的障碍物. 同时, 路径选择模块实时计算机器人附近区域的“可通过性”, 引导机器人选择较安全区域通过. 与未引入这两个模块的深度强化学习方法相比, 所提出的方法在所有仿真测试环境中均取得最高的导航成功率, 最高提升达11%.

**关键词:** 深度强化学习; 人群导航; 动态避障; 机器人冻结; 风险感知; 路径选择

中图分类号: TP 181

文献标志码: A

文章编号: 1005-3026(2025)12-0001-08

## Risk-Oriented Crowd Navigation Strategy Based on Deep Reinforcement Learning

JIANG Yang<sup>1</sup>, ZHAO Tian-xiang<sup>2</sup>, SUN Ruo-huai<sup>2,3</sup>, WANG Lei<sup>4,5</sup>

(1. School of Robot Science & Engineering, Northeastern University, Shenyang 110169, China; 2. School of Information Science & Engineering, Northeastern University, Shenyang 110819, China; 3. SIASUN Robot & Automation Co., Ltd., Shenyang 110168, China; 4. China Coal Science and Engineering (Liaoning) Embodied Intelligent Technology Co., Ltd., Shenyang 110168, China; 5. China Coal Technology & Engineering Group Robotics Co., Ltd., Shenyang 110168, China. Corresponding author: JIANG Yang, E-mail: jiangyang@mail.neu.edu.cn)

**Abstract:** To improve robot freezing and suboptimal performance of traditional navigation methods in the presence of dynamic obstacles, a navigation method based on deep reinforcement learning was proposed. The core of this method lies in its risk perception module and path selection module. The risk perception module calculated the collision probability between the robot and nearby dynamic obstacles in real time, allowing the robot to prioritize avoiding more hazardous obstacles. Concurrently, the path selection module evaluated the “passing ability” of the robot in surrounding areas in real time, guiding the robot to choose safer paths. In comparison experiments with a deep reinforcement learning method that lacks these modules, the proposed method achieved the highest navigation success rate in all simulation test environments, with an improvement rate of up to 11%.

**Key words:** deep reinforcement learning; crowd navigation; dynamic obstacle avoidance; robot freezing; risk perception; path selection

近年来,随着人工智能和机器人技术的迅猛发展,移动机器人已经成为各行各业中不可或缺的组成部分.不仅可以替代体力劳动者完成各类搬运任务,还在银行、餐饮、医院等行业中承担引导和服务的角色.然而,随着移动机器人技术的不断

进步,其应用场景也变得更加复杂,外部障碍物的未知性也随之增加<sup>[1]</sup>.传统的路径规划算法,例如A\*(A-Star)算法、Dijkstra算法、人工势场法、蚁群算法、DWA(dynamic window approach)算法等,在面对已知环境和静态障碍物时表现出较好

收稿日期: 2024-04-19

基金项目: 辽宁省重大科技计划项目(2023020703-JH26/101).

作者简介: 姜杨(1982—),男,辽宁沈阳人,东北大学副教授.

的避障效果,但人类行为通常具有复杂性和不可预测性,因此在应对复杂环境或人群动态障碍物时<sup>[2]</sup>,这些算法的避障策略却相对不足.这给传统路径规划算法带来了巨大的挑战.

在面对动态环境时,由于传统算法往往基于静态地图或先验信息进行规划,难以适应快速变化的场景.人类行为通常受到多种因素影响,包括情绪、社交互动和环境变化等,这进一步增加了移动机器人在真实世界中的导航难度.为解决这些问题,研究者们正在积极探索新的路径规划与避障方法,包括基于深度学习、强化学习等人工智能的方法.深度学习可通过学习大量实际环境数据提升机器人的感知与决策能力,从而更灵活地适应不同的场景.强化学习允许机器人在与环境的交互中学习最优行为策略,使其能够更好地应对复杂、动态的现实环境,许多学者在此方面开展了研究.Zhou等<sup>[3]</sup>使用社会注意力机制对环境进行特征提取,并通过整合无模型强化学习和在线规划,提出一种类人的决策过程,使机器人更高效地理解人群行为.Everett等<sup>[4]</sup>提出了一种无需假设智能体遵循特定行为规则的算法,并引入了LSTM(long short-term memory),使算法能够处理任意数量智能体的观察结果.Tai等<sup>[5]</sup>提出了一种基于深度强化学习(deep reinforcement learning, DRL)的运动规划器,并在使用DDPG(deep deterministic policy gradient)算法时使用异步采样以提升采样效率.该方法在静态环境及少量动态障碍物场景中,通常能有效地实施避障策略.然而,当机器人需同时与多个障碍物交互时,由于对所有外部障碍物采取统一的规避策略,而不考虑障碍物间差异,往往导致各避障动作相互冲突,致使机器人频繁陷入停滞状态<sup>[6]</sup>.

本文提出一种基于深度强化学习的方法,利用二维激光雷达观测外部障碍物信息,进行全局导航和局部避障,同时引入风险感知模块和路径选择模块辅助避障.在多个动态障碍物同时存在的情况下,综合考虑障碍物的距离、速度以及机器人自身速度,评估不同动态障碍物与机器人发生碰撞的概率,进而选择碰撞概率较高的障碍物进行避障<sup>[7-8]</sup>.简化了避障过程,提高了机器人避障效率.同时,路径选择模块的引入使得机器人能够选择交并比(intersection over union, IOU)较小的区域作为路径选择的依据,从而迅速做出路径决策<sup>[9]</sup>.本文方法综合了风险感知和路径选择,通过计算外部障碍物的碰撞风险,将风险较高的

障碍物纳入观测空间,进而使机器人在不预测外部智能体行为的情况下实现避障.

## 1 路径规划技术

路径规划技术是移动机器人领域的关键技术之一.路径规划算法分为全局路径规划算法和局部路径规划算法.其中,全局路径规划算法需要一张全局地图,并依照该地图进行导航.常见的全局路径规划算法包括A\*算法、Dijkstra算法等.局部路径规划算法通常依托于全局路径规划算法进行大致的路径规划,通过感知自身附近的环境信息,实现障碍物规避.常用的局部路径规划算法为动态窗口法(dynamic window approach, DWA),该算法具有计算量小、实时性高的特点,同时在障碍物规避方面表现良好,因此被研究人员广泛使用.然而DWA算法仅在静态、简单的环境中表现出良好的导航效果,当环境变得复杂,特别是出现一些动态障碍物时,DWA算法容易陷入局部极小值而无法保持良好的导航性能.

强化学习从本质上讲是一种决策过程,定义了一个代理作为决策者,代理之外的一切都定义为环境.代理不断地与周围环境进行交互,代理的一切行为都会对应一定的奖励,奖励有正有负.如果代理获得正向奖励,它就会倾向于执行类似的动作.如果代理获得负向奖励,则会避免重复此类动作.强化学习以最大化系统的累计奖励为目标.

深度强化学习自提出以来就获得了研究人员的广泛关注,其结合了深度学习的图像处理能力和强化学习“交互式学习”与“试错学习”的特点,在决策问题中成为了路径规划和移动机器人控制领域的一个研究热点.

## 2 动态避障问题建模与DRL导航

在本节中,将介绍用于动态避障的方法.机器人应与障碍物保持安全距离,为此,本文采用基于深度强化学习的方法,使用环境障碍物的信息作为输入,通过学习生成机器人动作.

### 2.1 问题描述

移动机器人动态避障的问题可以转化为部分可观测的马尔可夫决策过程(POMDP).将POMDP定义为 $(S, A, T, R, \Omega, O, \gamma)$ 元组,其中: $S$ 为机器人的状态空间,定义了所有可能的系统状态的集合; $A$ 为机器人的动作空间,定义了决策者

采取的所有可能动作的集合;  $T$  为转移模型, 描述了在给定一个状态和行动的情况下, 系统将转移到其他状态的概率分布;  $R$  是机器人动作的奖励函数, 定义了在不同状态下执行不同行动所获得的即时奖励;  $\Omega$  是机器人的观测空间, 定义了所有可能的观测结果;  $O$  是机器人的观测函数, 描述了在给定一个状态和行动的情况下, 系统观测到特定结果的概率分布;  $\gamma \in (0, 1)$  是未来奖励的折扣因子, 用于权衡当前奖励和未来奖励.

$$\left. \begin{array}{l} |S_t - S_p| < d_g, \text{目标约束;} \\ \min(O_{L,t}) > d_m, \text{碰撞约束;} \\ \operatorname{argmin}[t|s, \pi], \text{时间约束.} \end{array} \right\} \quad (1)$$

其中:  $S_t$  为机器人的位置;  $S_p$  为目标位置;  $d_g$  为目标区域半径;  $O_{L,t}$  为机器人与障碍物之间距离的观测值;  $d_m$  为机器人距离障碍物的安全区域半径;  $s$  为机器人当前状态;  $\pi$  为机器人的行动策略. 算法目标是在不发生碰撞的情况下使机器人到达目标点的时间尽可能短.

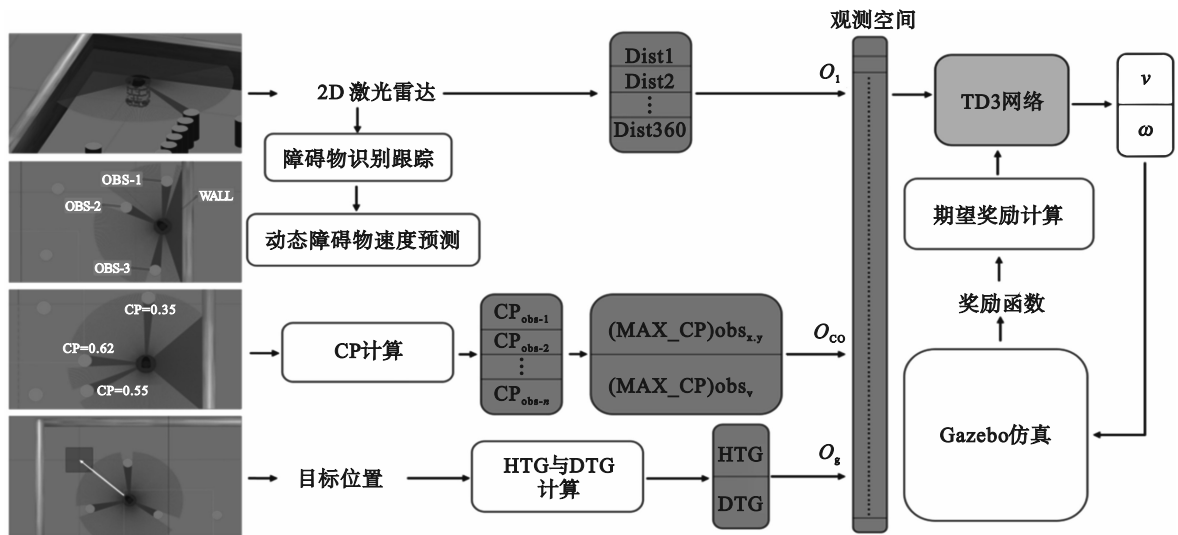
## 2.2 DRL 网络架构

系统设计的网络结构如图 1 和图 2 所示. 网络结构图详细揭示了利用二维激光雷达进行外部障碍物信息检测的完整流程, 并进一步展示了

动态障碍物碰撞概率 (collision probability, CP) 的计算机制<sup>[10]</sup>. 此外, 图中还包含了 TD3 (twin delayed deep deterministic) 策略梯度网络架构, 该架构用于实现障碍物识别、跟踪以及动态速度预测的高效学习与决策<sup>[11]</sup>. 观测空间涵盖了 TD3 网络在学习和执行导航任务时所需的输入特征, 以支持全局导航和局部避障<sup>[12]</sup>. 在全局导航方面, 通过设定目标点, 计算机器人与目标点之间的相对距离和方向, 将其作为观测信息. 其中, 朝向目标 (heading to goal, HTG) 和到目标的距离 (distance to goal, DTG) 用于量化机器人与目标的位置关系.  $v$  为机器人沿前进方向的移动速度,  $\omega$  为机器人绕自身旋转的角速度, 二者共同决定机器人的运动状态. 二维激光雷达在观测空间中提供外部障碍物的感知信息, 通过计算机器人与不同障碍物之间的碰撞概率来解决局部避障问题.

## 2.3 风险感知模块

在面对大量动态障碍物时, 机器人在同时避开所有障碍物方面面临挑战. 因此, 本文采用计算动态障碍物与机器人碰撞概率的方法, 反映不同障碍物的风险程度. 通过比较 CP 值的大小, 机器人能够有效感知周围环境的风险水平, 并相应调整行动, 以提高安全性和导航效率.



$O_1$  为激光雷达点云观测空间;  $O_{co}$  为距离障碍物 CP 值观测空间;  $O_g$  为目标位置观测空间.

图 1 系统结构图

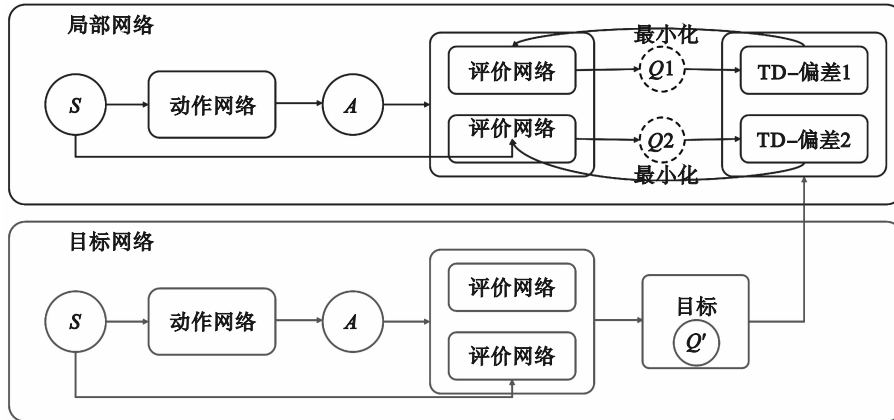
Fig. 1 System structure diagram

通过二维 (2D) 激光扫描技术<sup>[13]</sup>, 机器人能够区分动态和静态障碍物, 并实现对动态障碍物的跟踪. 首先, 根据激光雷达扫描得到的障碍物相对机器人的方向, 结合机器人的位置和方向, 将障碍物的极坐标位置信息 (OL) 转换为笛卡尔坐

标系. 随后, 利用 Kuhn-Munkres 算法将转换后的障碍物坐标信息划分为  $N$  组, 与机器人可感知到的障碍物进行匹配. 进一步, 通过相邻的 2 根雷达线反馈的距离信息, 计算障碍物的梯度以确定其类型 (墙壁或其他障碍物)<sup>[14]</sup>. 当障碍物的梯度为

0时,判定为墙壁.对不同类型的障碍物分别进行处理,利用每组的中心扫描数据进行障碍物的识

别、跟踪和类型判断.最后,计算每个动态障碍物与机器人的碰撞概率.



$Q$ 为动作价值函数,简称 $Q$ 值; $Q'$ 为目标 $Q$ 值; $Q_1, Q_2$ 为评价网络输出的2个 $Q$ 值.

图2 TD3网络架构

Fig. 2 TD3 network architecture

不同障碍物的危险程度不仅与机器人和障碍物之间的距离有关,还与它们之间的相对速度密切相关<sup>[15]</sup>.因此,本文定义的碰撞概率由两部分构成:基于时间的碰撞概率和基于距离的碰撞概率.通过对这两者进行加权处理,得到总的碰撞概率.

$$CP = \alpha P_t + (1 - \alpha) P_d \quad (2)$$

其中: $\alpha \in (0, 1)$ 是基于时间的碰撞概率和基于距离的碰撞概率的计算权重,在本文实验中, $\alpha$ 取值为0.5; $P_t$ 为基于时间的碰撞概率; $P_d$ 为基于距离的碰撞概率.

基于时间的碰撞概率如图3所示,在机器人和障碍物附近划分出危险区域,分别为 $r_1$ 和 $r_2$ .通过坐标变换,将机器人附近的危险区域叠加到障碍物上,此时机器人附近无危险区域,而障碍物附近的危险区域扩展为 $r_1+r_2$ .以障碍物为参考系计算机器人相对速度 $v$ ,即机器人速度 $v_1$ 减去障碍物速度 $v_2$ .机器人到障碍物危险区域的最短距离 $d$ 与速度 $v$ 的比值即为基于时间的碰撞概率,最大值为1.

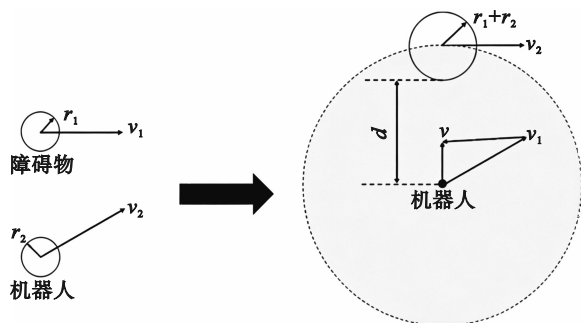


图3 基于时间的碰撞概率

Fig. 3 Time-based collision probability

图3中,如果 $v$ 方向的延长线与障碍物有交点,则 $t$ 表示按当前速度行进时机器人与障碍物发生碰撞的时间.0.15 s是机器人每个动作所需的时间,即,如果计算出的 $t$ 小于0.15 s,则机器人必将与障碍物发生碰撞.如果 $v$ 的方向与障碍物无交点,则 $d$ 为无穷大, $P_t$ 为0.如式(3)所示:

$$P_t = \min \left\{ 1, \frac{0.15}{t} \right\}, \quad t = \frac{d}{v} \quad (3)$$

基于距离的碰撞概率如图4所示, $r$ 为机器人与障碍物之间的距离.在机器人附近划分出两个区域,分别为 $r_{min}$ 和 $r_{max}$ .如果障碍物与机器人的距离大于 $r_{max}$ ,则认为碰撞概率为0;如果距离小于 $r_{min}$ ,则认为碰撞概率为1.在 $r_{min}$ 与 $r_{max}$ 之间的区域,采用线性插值法计算碰撞概率.

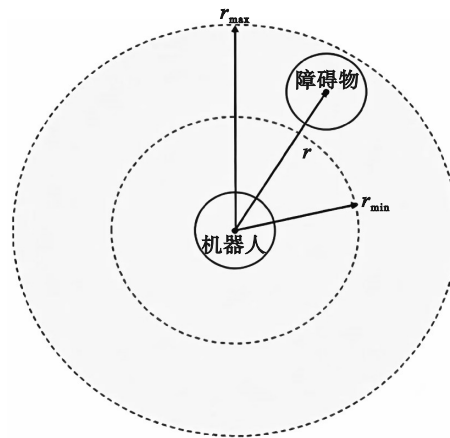


图4 基于距离的碰撞概率

Fig. 4 Distance-based collision probability

$$P_d = \frac{r_{\max} - r}{r_{\max} - r_{\min}}. \quad (4)$$

对障碍物列表中的每个障碍物的碰撞概率进行计算,并将碰撞概率较高的障碍物位置和速度作为  $O_{co}$  输入观察空间,进而对该障碍物进行重点避障.

## 2.4 路径选择模块

在面对多个动态障碍物时,仅依赖 TD3 网络进行导航,机器人往往难以有效地选择适当的路径,容易陷入死区.因此,本文设计了路径选择模块,帮助机器人在应对多个动态障碍物时作出更明智的路径选择.

两个障碍物之间的空间越大,路径的可通过性越高.因此,本文设计了路径选择模块,通过计算不同障碍物之间的交并比来评估它们之间的可通过性.

$$I = \frac{S_0}{S_1 + S_2 - S_0}. \quad (5)$$

式中: $I$ 为交并比; $S_1$ 为障碍物1的危险区域; $S_2$ 为障碍物2的危险区域; $S_0$ 为  $S_1$ 与  $S_2$ 的交集.通过  $I$ 值来衡量2个障碍物之间交集的大小. $I$ 值越小,表示2个障碍物之间的间隙越大,该区域的可通过性越好.

如图5所示,在障碍物周围划定危险区域.相邻障碍物之间的危险区域存在一定的交集.

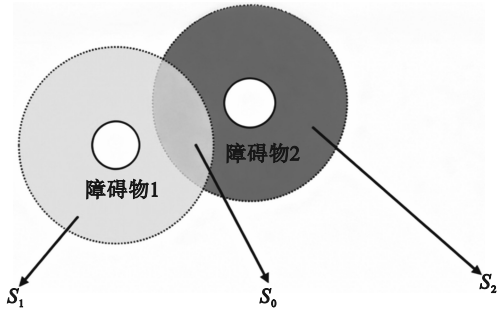


图5 交并比

Fig. 5 Intersection over union

## 2.5 奖励函数

为塑造机器人的行动策略,本文设计了奖励函数系统.

导航目标是使机器人在密集人群环境中无碰撞地到达设定的目标点.因此,在机器人训练过程中,对机器人的不同行为设置了不同的奖励.经过训练后,机器人的行为最终会收敛到奖励值最高的方向上.奖励函数为

$$R = R_{\text{step}} + R_{\text{dtg}} + R_{\text{htg}} + R_{\text{action}} + R_{\text{goal}} + R_{\text{collision}}. \quad (6)$$

式中: $R_{\text{step}} = -2$ ,表示机器人每一步给予一个负的

奖励,这样可以防止机器人为了获取  $R_{\text{dtg}}$  和  $R_{\text{htg}}$  而原地转圈或在目标点附近振荡,无法到达目标点;同时也可以鼓励机器人改进路径,从而更快地到达目标点.

$$R_{\text{dtg}} = \begin{cases} 1, & d(m, g)_t < d(m, g)_{t-1}; \\ 0, & \text{其他.} \end{cases} \quad (7)$$

其中:下标  $t$  表示当前时刻,  $t-1$  表示上一个时刻;  $m$  代表机器人的当前位置;  $g$  表示目标点;  $d(m, g)$  表示机器人到目标点的欧氏距离;  $R_{\text{dtg}}$  表示一个动作前后机器人距离目标点的距离,如果缩小,就给机器人一个正的奖励,否则不给予机器人奖励.通过该奖励的设置,可以鼓励机器人缩短与目标点之间的距离.

$$R_{\text{htg}} = \begin{cases} 1, & \theta(m, g)_t < \theta(m, g)_{t-1}; \\ 0, & \text{其他.} \end{cases} \quad (8)$$

其中,  $R_{\text{htg}}$  表示一个动作前后机器人的朝向和机器人同目标点之间的方向夹角.如果缩小,就给机器人一个正的奖励,否则不给予机器人奖励.通过该奖励设置,可以鼓励机器人缩短与目标点之间的航向角,朝着目标点的方向前进.

$$R_{\text{action}} = \begin{cases} 5, & \text{前进;} \\ 1, & \text{其他.} \end{cases} \quad (9)$$

其中,  $R_{\text{action}}$  表示机器人执行每个动作时获得的奖励.机器人的动作空间包含两个参数:线速度  $v$  和角速度  $\omega$ .当线速度  $v > 0$  且角速度  $\omega = 0$  时,判定机器人为直行;当  $v > 0$  且角速度  $\omega > 0$  时,判定机器人为左转;当  $v > 0$ ,角速度  $\omega < 0$  时,判定机器人为右转;当  $v = 0$  时,判定为停止.针对机器人不同的动作设置差异化奖励:如果判断机器人直行,给予一个较大的奖励;如果是其余动作,给予一个较小的奖励.通过该奖励的设置,可以鼓励机器人直行,在一定程度上可有效抑制机器人摆动现象.

$R_{\text{goal}}$  为机器人到达目标位置时给予的奖励,  $R_{\text{collision}}$  为机器人发生碰撞时给予的惩罚.  $R_{\text{goal}} = +200$  是机器人到达目标位置时给予的最大正奖励;如果发生碰撞,则给予  $R_{\text{collision}} = -200$  的惩罚.

## 3 实 验

### 3.1 实验设置

在 Gazebo 仿真环境中进行实验验证,实验服务器运行环境如下:

操作系统:Ubuntu MATE20.04; CPU: Intel(R) Core (TM) i7-9700 CPU @ 3.00 GHz; GPU:

Intel (R) UHD Graphics 630; Python 版本 : 3.8;  
Torch 版本 : 1.10.1+cu11.1; Torchvision 版本 :  
0.11.2++cu11.1.

Actor 网络和 Critic 网络的学习率均设置为 0.001. 训练在一张 NVIDIA GeForce GTX 1050 显卡上进行.

本文在 Gazebo 仿真平台中搭建实验环境进行算法训练. 环境配置如图 6 所示. 算法的设计目标为实现复杂动态环境下机器人的自主导航与避障功能. 鉴于最终测试环境的复杂性, 可能导致算法难以收敛, 本研究采用渐进式的训练策略. 首先在图 6a 所示的无障碍物环境中进行训练, 使机器人掌握基础导航能力; 待算法收敛后, 将该模型迁移到有静态障碍物的环境(图 6b), 进而训练机器人躲避静态障碍物的能力; 最终在图 6c 所示的动态障碍物环境中进行训练(障碍物运动模拟真实人群移动模式), 使机器人获得动态避障的能力.

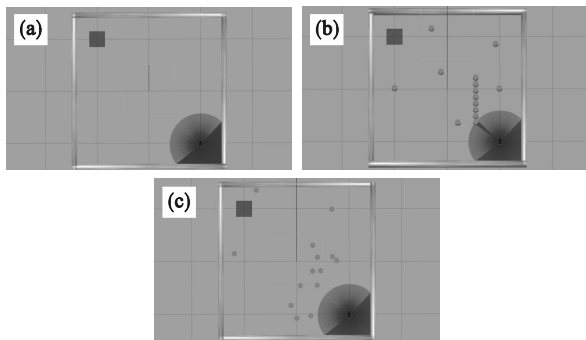


图 6 算法训练环境

Fig. 6 Algorithm training environment

(a)一无障碍物; (b)一有静态障碍物; (c)一有动态障碍物.

为了防止陷入循环的机器人浪费训练时间和计算资源, 设置每个轮次中机器人最多采集 1 000 次环境数据. 当达到最大采集次数或机器人发生碰撞或到达终点时, 机器人将停止行动并重新开始新的训练轮次.

### 3.2 算法训练

算法分别在无障碍物环境、有静态障碍物环境以及有动态障碍物环境中进行训练. 在无障碍物环境下, 算法经过 350 轮次训练后逐渐收敛, 如图 7 所示. 以无障碍物环境下收敛的模型为基础, 在有静态障碍物的环境下进行训练, 算法经过 450 轮次训练后逐渐收敛, 如图 8 所示. 以静态障碍物环境下收敛的模型为基础, 在含有动态障碍物的环境下训练, 经过 300 轮次训练后模型收敛, 如图 9 所示.

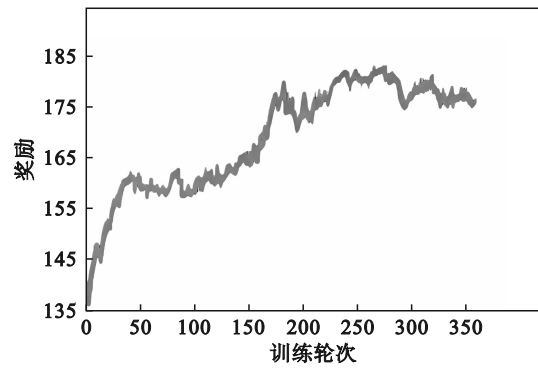


图 7 无障碍物训练过程

Fig. 7 Obstacle-free training process

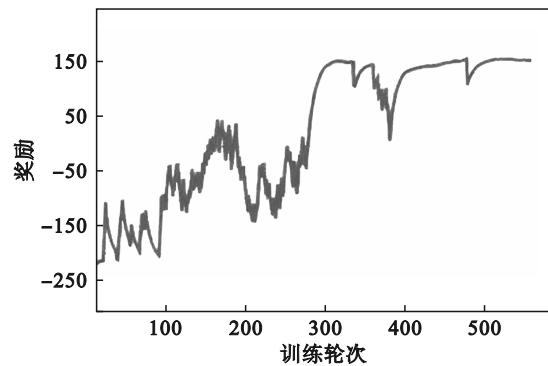


图 8 有静态障碍物训练过程

Fig. 8 Training process with static obstacles

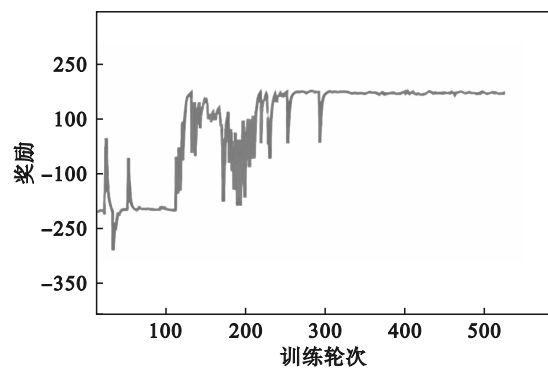


图 9 有动态障碍物训练过程

Fig. 9 Training process with dynamic obstacles

### 3.3 仿真实验验证

为了验证算法的动态避障能力, 本文将在上述环境中训练好的模型应用到更大范围的场景中, 分别测试了动态障碍物数为 4, 6, 8, 14 个时算法的性能表现. 测试环境如图 10 所示. 针对每种环境, 还测试了机器人在低速 (0.3 m/s) 和高速 (1.5 m/s) 状态下的避障效果. 在每个环境中分别进行 100 次测试, 并将本文算法与未加入风险感知模块的 TD3 算法 (原算法) 进行对比. 低速和高速状态下的测试结果如图 11~图 16 所示.

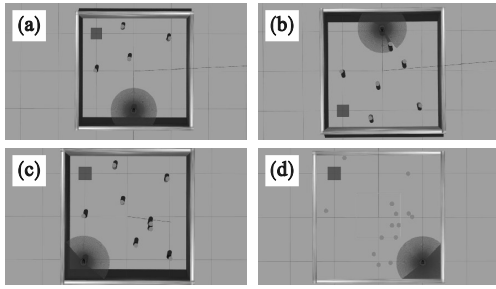


图 10 动态障碍物测试环境

Fig. 10 Testing environment with dynamic obstacles

(a) — 动态障碍物数为 4; (b) — 动态障碍物数为 6;  
(c) — 动态障碍物数为 8; (d) — 动态障碍物数为 14.

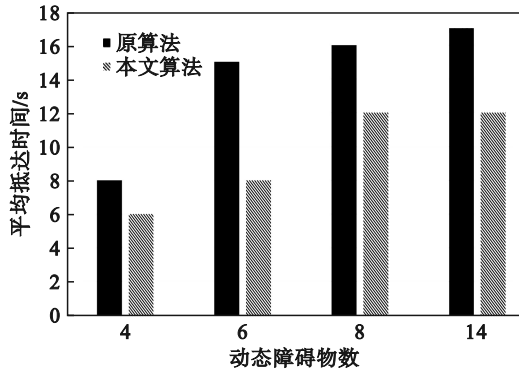


图 11 低速环境下平均抵达时间对比

Fig. 11 Average time to arrival comparison in low-speed environments

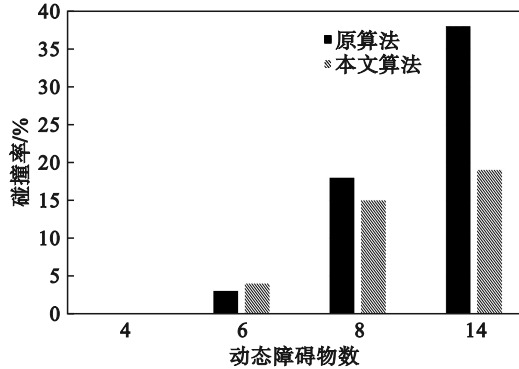


图 12 低速环境下碰撞率对比

Fig. 12 Collision rate comparison in low-speed environments

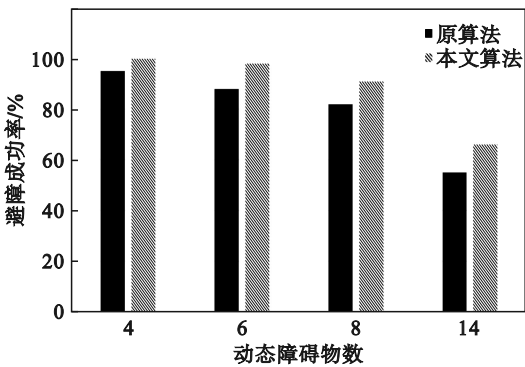


图 13 低速环境下避障成功率对比

Fig. 13 Obstacle avoidance success rate comparison in low-speed environments

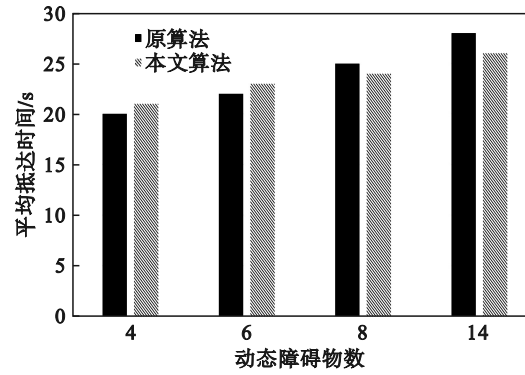


图 14 高速环境下平均抵达时间对比

Fig. 14 Average time to arrival comparison in high-speed environments

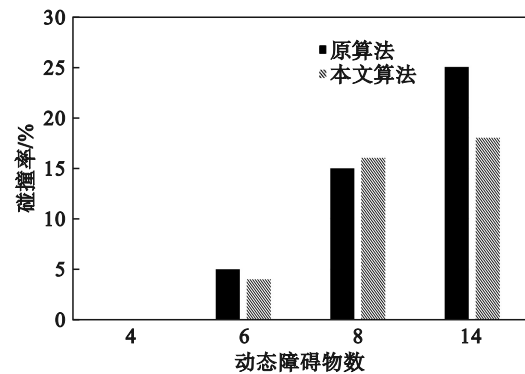


图 15 高速环境下碰撞率对比

Fig. 15 Average collision rate comparison in high-speed environments

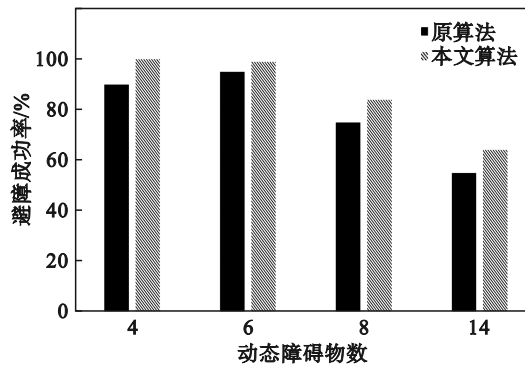


图 16 高速环境下避障成功率对比

Fig. 16 Obstacle avoidance success rate comparison in high-speed environments

从图 11~图 16 可见,在动态障碍物较少的情况下,本文所提出的算法并未展现出显著优势.然而,随着动态障碍物数的增加,算法的优越性逐渐显现.例如,在低速(0.3 m/s)条件下,面对 8 个动态障碍物时,本文算法导航成功率提高了 9%;在 14 个动态障碍物的环境中,提升达到了最高 11%.即便在高速(1.5 m/s)条件下,避障成功率最大提升也达到了 9%.

## 4 结 语

1) 为解决复杂人群环境下的移动机器人导航问题,本文提出了一种融合风险感知与路径选择的深度强化学习导航方法.

2) 本文提出了障碍物风险程度评估方法,使机器人能在移动过程中实时定量评估附近障碍物的危险程度,将机器人的避障效率在低速环境下最高提升 11%,在高速环境下最高提升 9%.

3) 提出了适用于复杂动态环境的机器人路径选择方法,使机器人能在复杂动态环境中快速选择较安全的路径通过,将低速障碍物环境下机器人抵达终点的平均时间降低 5 s. 仿真实验结果表明,本文方法能有效改善机器人在复杂动态环境下避障效果不佳及难以抵达终点的问题,并具有较好的泛化能力.

### 参考文献:

- [1] 谢洪乐, 陈卫东, 范亚娟, 等. 月面特征稀疏环境下的视觉惯性 SLAM 方法[J]. 航空学报, 2021, 42(1): 524169. (Xie Hong-le, Chen Wei-dong, Fan Ya-xian, et al. Visual-inertial SLAM in featureless environments on lunar surface [J]. *Acta Aeronautica et Astronautica Sinica*, 2021, 42(1): 524169.)
- [2] Xing Z W, Zhu X R, Dong D C. DE-SLAM: SLAM for highly dynamic environment[J]. *Journal of Field Robotics*, 2022, 39(5): 528-542.
- [3] Zhou Z Q, Zhu P M, Zeng Z W, et al. Robot navigation in a crowd by integrating deep reinforcement learning and online planning [J]. *Applied Intelligence*, 2022, 52(13): 15600-15616.
- [4] Everett M, Chen Y F, How J P. Motion planning among dynamic, decision-making agents with deep reinforcement learning [C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Madrid, 2019: 3052-3059.
- [5] Tai L, Paolo G, Liu M. Virtual-to-real deep reinforcement learning: continuous control of mobile robots for mapless navigation [C]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Vancouver, 2017: 31-36.
- [6] 孙喜亮, 关宏灿, 苏艳军, 等. 面向高精度城市测绘的激光紧耦合 SLAM 方法[J]. 测绘学报, 2021, 50(11): 1585-1593. (Sun Xi-liang, Guan Hong-can, Su Yan-jun, et al. A tightly coupled SLAM method for precise urban mapping [J]. *Acta Geodaetica et Cartographica Sinica*, 2021, 50(11): 1585-1593.)
- [7] Tian Y, Chen H W, Wang F S, et al. Overview of SLAM algorithms for mobile robots[J]. *Computer Science*, 2021, 48(9): 223-234.
- [8] 张宇桐, 宋建梅, 丁艳, 等. 基于鱼眼相机和 RGBD 相机的异构协同 SLAM[J]. 航空学报, 2023, 44(10): 327621. (Zhang Yu-tong, Song Jian-mei, Ding Yan, et al. Heterogeneous collaborative SLAM based on fisheye and RGBD cameras [J]. *Acta Aeronautica et Astronautica Sinica*, 2023, 44(10): 327621.)
- [9] 陶交, 范馨月, 周非. 点线特征融合的双目视觉 SLAM 算法[J]. 小型微型计算机系统, 2022, 43(6): 1191-1196. (Tao Jiao, Fan Xin-yue, Zhou Fei. Point-line feature fusion in stereo visual SLAM algorithm [J]. *Journal of Chinese Computer Systems*, 2022, 43(6): 1191-1196.)
- [10] Chen W Q, Wang Y, Chen H Y, et al. EIL-SLAM: depth-enhanced edge-based infrared-LiDAR SLAM [J]. *Journal of Field Robotics*, 2022, 39(2): 117-130.
- [11] 蒋林, 刘林锐, 周安娜, 等. 基于运动预测的改进 ORB-SLAM 算法[J]. 浙江大学学报(工学版), 2023, 57(1): 170-177. (Jiang Lin, Liu Lin-rui, Zhou An-na, et al. Improved ORB-SLAM algorithm based on motion prediction [J]. *Journal of Zhejiang University (Engineering Science)*, 2023, 57(1): 170-177.)
- [12] 李荣华, 祁宇峰, 谢辉, 等. 面向未知环境的紧耦合激光 SLAM 方法[J]. 红外与激光工程, 2023, 52(9): 135-144. (Li Rong-hua, Qi Yu-feng, Xie Hui, et al. Tightly coupled LiDAR SLAM method for unknown environment [J]. *Infrared and Laser Engineering*, 2023, 52(9): 135-144.)
- [13] Li J Q, Pei L, Zou D P, et al. Attention-SLAM: a visual monocular SLAM learning from human gaze [J]. *IEEE Sensors Journal*, 2021, 21(5): 6408-6420.
- [14] 沈晔湖, 陈嘉皓, 李星, 等. 基于几何-语义联合约束的动态环境视觉 SLAM 算法[J]. 数据采集与处理, 2022, 37(3): 597-608. (Shen Ye-hu, Chen Jia-hao, Li Xing, et al. Dynamic visual SLAM based on unified geometric-semantic constraints [J]. *Journal of Data Acquisition and Processing*, 2022, 37(3): 597-608.)
- [15] 刘少哲, 刘作军, 胡超芳, 等. 点线特征自适应融合室内 SLAM 算法[J]. 小型微型计算机系统, 2023, 44(5): 1015-1022. (Liu Shao-zhe, Liu Zuo-jun, Hu Chao-fang, et al. Point-line feature adaptive fusion indoor SLAM algorithm [J]. *Journal of Chinese Computer Systems*, 2023, 44(5): 1015-1022.)