第56卷第2期
2024年4月

南京航空航天大学学报
Journal of Nanjing University of Aeronautics & Astronautics

Vol. 56 No. 2
Apr. 2024

DOI:10.16356/j.1005-2615.2024.02.020

基于Q-learning的搜救机器人自主路径规划

褚　晶，邓旭辉，岳　颀
（西安邮电大学自动化学院，西安 710121）

摘要：当人为和自然灾害突然发生时，在极端情况下快速部署搜救机器人是拯救生命的关键。为了完成救援任务，搜救机器人需要在连续动态未知环境中，自主进行路径规划以到达救援目标位置。本文提出了一种搜救机器人传感器配置方案，应用基于Q-table和神经网络的Q-learning算法，实现搜救机器人的自主控制，解决了在未知环境中如何避开静态和动态障碍物的路径规划问题。如何平衡训练过程的探索与利用是强化学习的挑战之一，本文在贪婪搜索和Boltzmann搜索的基础上，提出了对搜索策略进行动态选择的混合优化方法。并用MATLAB进行了仿真，结果表明所提出的方法是可行有效的。采用该传感器配置的搜救机器人能够有效地响应环境变化，到达目标位置的同时成功避开静态、动态障碍物。

关键词：搜救机器人；路径规划；传感器配置；Q-learning；神经网络
中图分类号：TP242　　**文献标志码：**A　　**文章编号：**1005-2615(2024)02-0364-11

Q-learning Based Autonomous Path Planning for Search and Rescue Robots

CHU Jing, *DENG Xuhui*, *YUE Qi*
(School of Automation, Xi'an University of Posts and Telecommunications, Xi'an 710121, China)

Abstract: When man-made or natural disasters occur suddenly, the rapid deployment of search and rescue (SAR) robots is crucial for saving lives. To accomplish rescue tasks, SAR robots need to autonomously plan paths in continuously dynamic and unknown environments to reach the rescue target locations. This paper proposes a sensor configuration scheme for SAR robots, applying a Q-learning algorithm based on Q-table and neural networks to achieve autonomous control of SAR robots. It addresses the challenge of path planning in unknown environments, specifically how to avoid static and dynamic obstacles. Balancing the exploration and exploitation during the training process is one of the challenges in reinforcement learning. This paper introduces a mixed optimization method for dynamically selecting search strategies, building upon greedy search and Boltzmann search. Simulations are conducted using MATLAB, and the results indicate that the proposed method is feasible and effective. SAR robots equipped with the sensor configuration can effectively respond to environmental changes, reaching target locations while successfully avoiding both static and dynamic obstacles.

Key words: search and rescue(SAR) robot; path planning; sensor configuration; Q-learning; neural network

自然灾害和人为灾害事件时有发生，且类型多变、种类多样，危害甚广。当灾害突发时，由于房屋、道路交通系统严重受损，很多人员被困受灾点，短时间内无法及时进行救援。灾害是不可避免的，但是可以通过高效的灾后应急救援，大幅降低灾害造成的损失。搜救机器人的有效应用能够提高应急救援的效率；然而，复杂、未知的灾区环境为搜救机器人的自动部署带来了极大的挑战。

基金项目：国家自然科学基金(61703336)；陕西省自然科学基金(2023-JC-QN-0727)。
收稿日期：2023-10-24；**修订日期：**2023-12-25
通信作者：褚晶，男，讲师，E-mail：jchu@xupt.edu.cn。

引用格式：褚晶，邓旭辉，岳颀. 基于Q-learning的搜救机器人自主路径规划[J]. 南京航空航天大学学报，2024，56(2)：364-374. CHU Jing, DENG Xuhui, YUE Qi. Q-learning based autonomous path planning for search and rescue robots[J]. Journal of Nanjing University of Aeronautics & Astronautics, 2024, 56(2): 364-374.

目前已经提出了许多路径规划算法来部署机器人，但大多数传统算法都是基于已知的环境模型。然而，在灾害发生之后，存在着很大的挑战，例如：(1)环境不仅是未知的，还是动态的，有时无法获得环境的模型[1]；(2)由于存在控制误差或外界因素的干扰，机器人在执行过程中所下达的指令与实际的操作结果有一定的偏差；(3)受到静态、动态两种因素的影响，规划出的路径会出现弯曲多拐点的情况，这对机器人的实际行走是不利的。

当搜救机器人在未知动态的灾后环境中工作时，传统的路径规划方法可能会因为上述挑战而失败。强化学习为解决这些挑战提供了一种很有效的方法[2-3]。Q-learning是强化学习中最具有代表性的算法，它将自学习能力引入其中，使搜救机器人在训练后能够在未知环境中实现自主避障。搜救机器人通过执行动作与环境进行信息交互，然后观察该动作的结果，结果以正奖励或负奖励的形式给出。该学习过程的目标是在一段时间内最大限度地增加累计奖励[4-5]。

在Q-learning算法中，由于机器人会在探索时出现多次的重复次优路径，从而陷入局部极值，影响算法的收敛性。近年来，也有不少学者进行了创新与改进，为加快算法的收敛性，文献[6-7]采用基于人工势场法的Q-table初始化方法，该方法能够给机器人提供先验知识，极大地降低了机器人前期探索的随机性，但是在算法的后期，当出现一些特殊的条件时，该方法就不适用了；文献[8-9]利用一种基于神经网络的Q-learning算法，采用径向基函数(Radial basis function，RBF)网络对Q-learning算法的行为值函数进行近似；文献[10]为了平衡强化学习中的探索与利用，提出了一种能够按照特定要求调整自身的行为选择策略；文献[11-12]利用前馈神经网络，将Q-learning和Sarsa算法相结合，提出了改进的Q-learning算法。

本文主要研究搜救机器人如何在复杂未知的灾区环境中进行自主路径规划实现应急救援，在实时连续的动态环境中避开静态和动态障碍物，到达目标位置。

(1)以Q-learning方法为基础，并引入神经网络方法与之结合。根据所提出的传感器配置方案，分别采用两种算法来解决未知环境中搜救机器人的路径规划问题。该方法不需要事先对环境进行建模，而是不停地进行实验，通过实验来与环境进行交互，并在反馈信息的基础上，对搜救机器人的动作进行优化。

(2)针对强化学习中的挑战，即如何平衡探索与利用问题。本文提出了一种混合优化策略，令算法能够在贪婪搜索和Boltzmann搜索策略之间进行动态切换，避免算法陷入局部最优，大大提升了搜救机器人在未知复杂的环境中寻找目标位置的能力。

(3)仿真实验表明，经过一段时间的学习后，搜救机器人已经具备了在静态和动态环境中规避障碍，寻找到目标位置的能力，并且系统性能稳定，能够稳定到达目标位置。

1　搜救机器人模型及其传感器配置

1.1　搜救机器人模型

本文所用的搜救机器人为差动驱动结构，如图1所示，由两个同轴线的驱动轮构成，搜救机器人的中心位于两车轮的中点。假设搜救机器人在二维xy平面上运动。

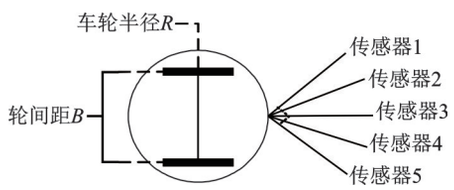

图1　搜救机器人模型

Fig.1　Search and rescue robot model

首先定义瞬时曲率中心(Instantaneous center of curvature，ICC)到搜救机器人两个车轮的中心的距离为R，左右车轮之间的距离为B。v_l和v_r分别表示左右车轮沿地面的速度，左右两轮关于ICC的旋转速度ω是相同的，通过操纵控制参数v_l和v_r可以使搜救机器人向不同的位置和方向移动。

$$v_l=\omega\left(R-\frac{B}{2}\right) \tag{1}$$

$$v_r=\omega\left(R+\frac{B}{2}\right) \tag{2}$$

两个车轮以相同的角速度转动时，搜救机器人沿直线前进；当搜救机器人需要改变方向时，两个车轮的转速都会发生改变。因此，可以得到当$v_l=v_r$时，搜救机器人沿直线向前运动；当$v_l=0$时搜救机器人绕左轮旋转；当$v_r=0$时搜救机器人绕右轮旋转。

假设搜救机器人当前位置为(x,y)，朝着相对于x轴成θ角的方向前进。那么，Δt之后(设仿真过程中$\Delta t=1$)的搜救机器人的位置坐标变化方程为

$$p(x,y)=(x,y)+v(v\cos(\theta+\Delta\theta),v\sin(\theta+\Delta\theta)) \tag{3}$$

1.2　搜救机器人的环境检测

借助传感器，搜救机器人能够直接获取环境中

障碍物的距离和角度信息，合适的传感器配置将为搜救机器人提供更高的精度。对输入数据进行简单而有效的处理同样具有重要性，然而，在基于Q-table的强化学习中，信息量过大会导致搜救机器人学习所需的时间增加，计算负荷庞大。本文对连续传感器区域进行分割和离散化，使环境空间变得更易管理，在保证精度的同时减少了计算量。

本文提出了一种由5个HC-SR04超声波传感器组成的搜救机器人，主要用来测量周围的环境以检测到障碍物。HC-SR04超声波传感器被广泛应用于避障、距离测量等场景，能够在2 cm到400 cm的范围内测量，并具有±3 mm左右的高精度。将传感器模块安装在搜救机器人的前部，使其能够有效地发送超声波脉冲并接收回波信号。

最外侧的两个传感器的夹角为90°，中间3个传感器呈对称分布。每个传感器的视野约为60°，安装时它们之间存在约15°的重叠，因此搜救机器人的总视野约为150°。针对静态障碍物避障，本研究将传感器测量的距离和角度信息离散化为144个状态。如图2所示，根据传感器的测量距离将空间划分为4个区域。

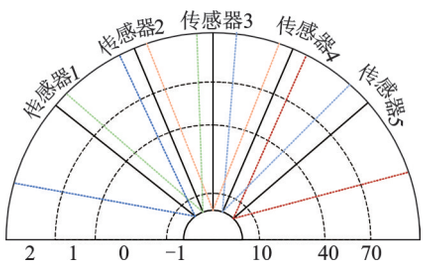

图2 机器人视野示意图

Fig.2 Schematic diagram of robot field of view

将视觉范围分为多个扇区，搜救机器人可以更容易地识别障碍物的位置和方向，并采取适当的行动来避开或处理这些障碍物。根据搜救机器人的视角范围，将150°的总视角划分为8个扇区，每个扇区覆盖一个特定的角度范围。如图3所示，这些扇区是对称分布的，两侧的扇区范围是相同的。这意味着搜救机器人可以同时感知到两个障碍物，但不能在同一侧。若同一侧有两个障碍物，算法将优先考虑扇区值编号最低的障碍物，即在一个扇区中检测到的第一个障碍物将被视为机器人在该扇区的主要障碍物。

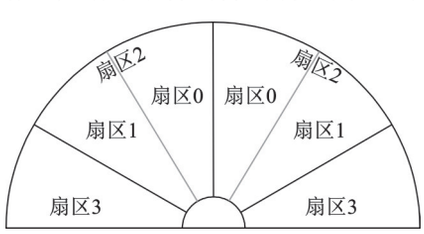

图3 区域的细分

Fig.3 Subdivision of area

用参数来表示区域和扇区，k_1、k_3分别用于表示左侧的区域和扇区，k_2、k_4则分别用于表示右侧的区域和扇区。在存在动态障碍物的仿真环境中，状态空间扩展两个附加参数k_5、k_6。k_5用值1或0表示左侧的障碍物是否为动态障碍物，同理k_6用于表示右侧的障碍物是否为动态障碍物。

用$\boldsymbol{r}$表示搜救机器人与目标的位置矢量，相对于x轴的角度记为θ，角θ可用作目标位置的指示。角θ包含在状态空间中，并被离散成6个部分，用k_j表示。当$\theta\in[0,\pi/12]$时，允许搜救机器人接近目标位置；当$\theta\in[\pi/12,\pi/2]$时，允许搜救机器人左转或右转进行空间探索；当$\theta\in[\pi/2,\pi]$时，搜救机器人只允许直线行走。同样的规则也适用于$\theta\in[0,-\pi]$。

2 基于Q-learning的学习算法

2.1 Q-table方法

Q-learning算法是一种基于值迭代的强化学习算法，它能够在离散状态和动作空间中解决最优策略问题，且不需要事先知道环境的状态转化模型；其主要思想是搜救机器人与周围的环境进行交互，搜救机器人对每个可能的状态和动作进行多次尝试，不断地学习和优化一个价值函数$Q(s,a)$来实现自主学习[13]。在使用Q-learning算法进行路径规划的过程中，有3个重要元素：状态s_t、动作a_t和奖励r_t。通过搜救机器人与环境之间不断的信息交换，算法会根据动作与状态构建Q-table，并将每个状态的状态值存储在表中，简称Q值。

在启动时，Q-table是空的，搜救机器人只知道一组可能的状态和动作。搜救机器人在当前状态s_t下，选择动作a_t，通过环境的作用，形成新的状态s_{t+1}，并产生回报或惩罚r_{t+1}。

Q-table根据式(4)更新。

$$Q(s_t,a_t)\leftarrow Q(s_t,a_t)+\alpha\left[r_{t+1}+\gamma\max_a Q(s_{s+1},a)-Q(s_t,a_t)\right] \quad (4)$$

式中：s_t为当前状态，a_t为当前状态下执行的动作，r_{t+1}为当前状态下获得的奖励；α为学习率，$\alpha\in[0,1]$，$\alpha=0$，表示搜救机器人只能学习过去的状态，$\alpha=1$表示搜救机器人可以学习所有未来的奖励；γ为折扣因子，$\gamma\in[0,1]$，当$\gamma=0$时表示搜救机器人只能接受当前的奖励。

若Q值变小，说明搜救机器人处于当前位置时选择该动作不是最优的。当搜救机器人再次处于该位置或状态时，搜救机器人可以避免再次选择

该动作。Q-table会随着移动机器人探索环境次数的增加而更新，搜救机器人会根据每个状态选择奖励最大的动作，在多次迭代后，搜救机器人最终会获得最优动作。

在本文中，基于Q-table的Q-learning算法的状态空间为这5个传感器探测到的离散化区域。在静态障碍物环境下，可以用$S_{静}=[k_1,k_2,k_3,k_4,k_j]$表示；由于动态环境中包含更多的信息，所以状态空间需要更多的元素，即$S_{动}=[k_1,k_2,k_3,k_4,k_5,k_6,k_j]$。其中$k_1$和$k_2$分别代表区域的左右侧，$k_3$和$k_4$分别代表了扇区的左右侧，$k_5$和$k_6$分别表示搜救机器人左右侧的静态或动态障碍物，$k_j$表示角度$\theta$。

搜救机器人在每个状态下可采取直行、右转、左转3种动作，表示为动作集合$a=[1,2,3]$奖励函数对搜救机器人进行激励，对表现好的行为进行正激励，对不良行为进行负面激励，以此来保障搜救机器人的安全。本文明确了搜救机器人在前进时给出正奖励值，而在转弯时给出负奖励值。为了防止搜救机器人先右后左，或先左后右，导致来回运动，应给予负奖励值。搜救机器人距离目标位置越近，奖励的正向数值就越高。最后，如果没有发生碰撞，则搜救机器人的总奖励值等于上述奖励值的总和，如果发生碰撞，则给予较大的负奖励值作为惩罚。

2.2　动作选择策略

探索可能以牺牲短期利益为代价，通过收集更多的信息来获得更准确的长期利益估计；而利用的重点是在可获得的信息的基础上使短期收益最大化。探索行为不能无休止地进行下去，否则将会以牺牲短期利益为代价，损害全局利益；同时也不能太看重短期利益，而忽略了未来的长远利益。

贪婪探索在面对问题时，总是根据目前的情况，做出当下最佳的决定。也就是说，如果不考虑全局优化，那么得到的结果只能是局部优化。a^*表示概率为$1-\varepsilon$的最优选择，a_r表示概率为ε的随机选择。

$$a=\begin{cases}a^* & P=1-\varepsilon\\ a_r & P=\varepsilon\end{cases} \tag{5}$$

对于行为选择策略，较为理想的情况是高概率选择具有高奖励的行为和低概率选择具有低或负奖励的行动。因此，不使用动作选择概率，而是使用权重来确定动作，对产生高奖励值的动作赋予高权重，反之亦然。这种策略被称为Boltzmann搜索策略，表达式如下

$$P=\frac{e^{\frac{Q(s,a)}{T}}}{\sum_a e^{\frac{Q(s,a)}{T}}} \tag{6}$$

式中：P为选择动作a的概率；$Q(s,a)$为动作a的价值估计；T为控制选择随机性的参数。

探索可以提高算法的收敛性，但是由于搜救机器人对所处的环境不够熟悉，很容易陷入局部最优；Boltzmann搜索策略允许对环境进行大范围的探索，但算法收敛缓慢。因此，本文将这两种方法结合起来，提出了混合优化策略，既可以加快算法的收敛性，又可以防止算法陷入局部极值。这种组合动作选择策略允许算法在前期对未知环境进行充分的探索，随着搜救机器人对环境的熟悉程度越来越高，通过调整算法，赋予已知环境中奖励值最大动作更高的概率，从而提高搜索效率，节约计算资源。每个状态的最优动作由式(7)生成。

$$a^*(s)=\max_a Q(s_{t+1},a) \tag{7}$$

2.3　基于Q-learning的神经网络

神经网络是指一系列受到生物学和神经科学的启发而产生的数学模型，它主要是通过抽象人脑中的神经元网络，构造出人造神经元，然后根据特定的拓扑结构将人造神经元连接起来，从而模拟出生物神经网络。

由于环境的复杂性，传统的Q-learning方法无法将变化的环境信息构建成合适状态-动作对[14]；由于每个状态都会产生Q值，因此恢复和更新Q-table会占用大量的内存，另外在维度非常大时，输入存在复杂性。为了解决这些问题，基于神经网络的Q-learning算法使用前馈神经网络代替Q-table，利用一个或多个线性Q函数近似状态和动作之间的关系，并将Q值存储在神经网络中，不需要构建离散化的状态空间[15]。

在前馈神经网络中，神经元分布在不同的层次中，如图4所示，包括输入层、隐藏层和输出层。输入层包含6个神经元，其中5个接收来自传感器的数据，这些数据代表搜救机器人周围环境的信息。另一个神经元负责接收搜救机器人与目标位置的角度信息。这些输入神经元负责接收原始数据并将其传递给神经网络的下一层。隐藏层包含18个神经元，是不可见的。最右侧是输出层，由于搜救机器人的可选动作有直行、左转和右转3个动作，因此输出层包含3个神经元，用来输出的相应动作的Q值。在搜救机器人探索环境的过程中，传感器获取的环境信息被用作神经网络的输入x_i；通过激活函数所激活的值，成为该层的输出，并将其作为下一层的输入。搜救机器人在每个状态下不断执行直行、右转和左转3种动作，每个动作对应的

Q值是神经网络的输出，这些值进一步用于确定机器人的最优动作。

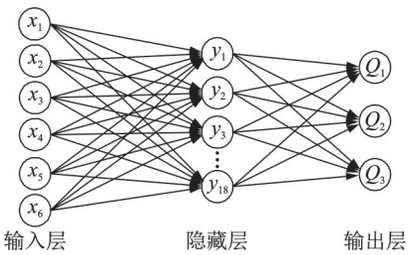

图4 基于Q-learning的神经网络

Fig.4 Neural network based on Q-learning

激活函数在神经元中扮演者极其重要的角色。选择连续可导的非线性函数作为激活函数能够显著增强网络的表示能力和学习能力。为了提高网络的效率，通常可以选择双曲正切函数，这个函数及其导函数都具有简单的形式，值域为[−1, 1]，不影响训练的效率和稳定性。

初始化后，逐层计算神经网络，直至获得最终输出。加权矩阵$\boldsymbol{\omega}_l$用于对第l层和第$l+1$层之间的连接进行加权。这些加权矩阵和每一层的输出值向量相乘，得到的乘积被传递给激活函数作为参数，输出层Q函数的激活值可以用式(8)所示。

$$Q_i=g\left(\omega^2_{(i,0)}\cdot y_0+\omega^2_{(i,1)}\cdot y_1+\omega^2_{(i,2)}\cdot y_2+\cdots+\omega^2_{(i,18)}\cdot y_{18}\right) \tag{8}$$

通过奖励函数来保障机器人的安全，对每个给定的行为进行奖惩[16]。奖励还会对Q值进行修正得到输出层的期望输出值，记为Q_{est}，表达式如式(9)所示。输出层的期望输出值Q_{est}与实际值$Q(s,a)$之间的偏差由损失函数给出，如式(10)所示。通过这种偏差修正，根据梯度下降原理，使用反向传播进行逆向求解，对权值进行优化，以此来减少神经网络的损失，提高在类似场景中重复选择相同行为的概率。

$$Q_{\text{est}}=r+\gamma\max_a Q(s',a) \tag{9}$$

$$C=\frac{1}{2}\left(Q_{\text{est}}-Q(s,a)\right)^2 \tag{10}$$

在计算偏差时，仅考虑与最后执行的动作对应的Q值，为了避免对其他Q函数进行错误估计，在第三层中将偏差向量乘以单位矩阵$\boldsymbol{I}$，$\boldsymbol{I}(a)$指单位矩阵中各种动作的每一列。设偏差为δ_l，其中参数l表示偏差针对的层数，式(11)为输出层的偏差计算公式。计算出偏差向量后，返回第二层，利用公式(12)求出该层的偏差向量，其中g'是激活函数tanh函数的导数。由于传感器值不是来自神经网络的估计值，所以不会存在偏差向量。

$$\boldsymbol{\delta}_3(a)=\left(Q_{\text{est}}-Q(s,a)\right)\cdot\boldsymbol{I}(a) \tag{11}$$

$$\boldsymbol{\delta}_2(a)=\left((\boldsymbol{\omega}_2)^{\mathrm{T}}\cdot\boldsymbol{\delta}_3\right)\cdot g'\left(\begin{bmatrix}1\\ \boldsymbol{\omega}_1\cdot\boldsymbol{x}\end{bmatrix}\right) \tag{12}$$

在利用反向传播算法找到所有的向量后，进一步利用偏差向量计算关于权重矩阵$\boldsymbol{\omega}_1$和$\boldsymbol{\omega}_2$的偏导值，即损失函数的梯度为$\nabla C=[\partial C/\partial\boldsymbol{\omega}_1,\partial C/\partial\boldsymbol{\omega}_2]$，损失函数的梯度用于更新神经网络中的权重矩阵，权重矩阵的更新公式如(14)所示。

$$\begin{cases}\dfrac{\partial C}{\partial\boldsymbol{\omega}_1}=\boldsymbol{\delta}_2\cdot(\boldsymbol{x})^{\mathrm{T}}\\ \dfrac{\partial C}{\partial\boldsymbol{\omega}_2}=\boldsymbol{\delta}_3\cdot(\boldsymbol{y})^{\mathrm{T}}\end{cases} \tag{13}$$

$$\begin{cases}\boldsymbol{\omega}_1'=\boldsymbol{\omega}_1-\alpha\dfrac{\partial C}{\partial\boldsymbol{\omega}_1}\\ \boldsymbol{\omega}_2'=\boldsymbol{\omega}_2-\alpha\dfrac{\partial C}{\partial\boldsymbol{\omega}_2}\end{cases} \tag{14}$$

3 仿真实验

3.1 仿真实验环境

本文的仿真系统简化了救灾场景，并在不同场景的地图上进行了实验，模仿了应急救援场景中可能出现的各种环境，以确保路径规划框架在不同环境规模下的通用性和鲁棒性。

搜救机器人目标到达救援任务的地图环境如图5所示，分别在存在静态和动态障碍物的环境中，使用两种不同算法求解搜救机器人的最优路径。在搜救过程中，搜救机器人要以最小时间步长规避静态和动态障碍物。图5描述了搜救机器人工作的环境，其中地图一和地图二均为30×30大

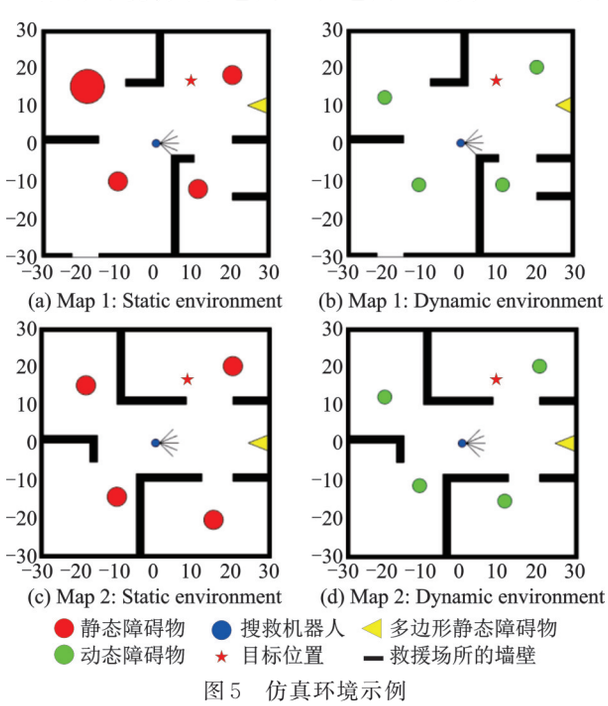

图5 仿真环境示例

Fig.5 Example of simulation environments

小，分别表示不同的环境。在图(a)和(c)中，展示了搜救机器人工作的初始场景，该场景中存在静态障碍物。其中，蓝色代表搜救机器人，黑色为灾难现场的墙壁，而红色代表静态障碍物。在图(b)和(d)中，呈现了搜救机器人所处的动态障碍物环境的初始场景。在救灾场景的房间中，还增加了 4 个绿色的动态障碍物。

这些障碍物的运动是有规律的，表现为匀速圆周运动，即每个动态障碍物都沿着一个特定的半径，以均匀的速度在环绕中心点的轨迹上移动。如图 6 所示(大小为 30×30)，编号 1、2、3 的动态障碍物沿着逆时针方向做圆周运动，编号 4 的动态障碍物沿着顺时针方向做圆周运动。这种规律的运动模式使得它们具有可预测性，但同时也增加了环境的复杂性，要求路径规划算法在考虑到障碍物运动规律的同时，能够在实时性、适应性和效率上做出有效的决策，以确保搜救机器人能够顺利完成任务。

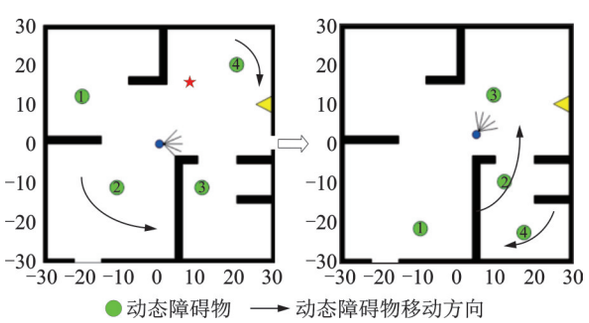

图 6　动态障碍物的位置变化

Fig.6　Position changes of dynamic obstacles

3.2　超参数设置

在进行搜救机器人避障训练时，更新 Q-table 需要确定两个关键的超参数，即折扣率 γ 和学习率 α。为了满足搜救机器人的长期需求，采取最优决策优化避障路径，因此折扣率 γ 选取较大值 0.9，表示未来决策对当前决策行为的影响较大。在学习率 α 的选择上，要得到最大的回报，须把试验的重复次数控制在可以接受的限度之内，设置学习率为 0.5。将 Boltzmann 搜索的初始参数设置为 $T=24$，贪婪搜索的初始时参数为 epsilon=0.95，以逐步减小每次尝试随机动作的概率。

在仿真的初始化阶段，将最大试验次数设置为 10 000 次，每次试验最大步长限制为 600 步。试验会在以下情况下结束：搜救机器人达到最大步长、触发碰撞条件或者到达目标位置。

3.3　仿真结果

在具有静态障碍物的环境中，搜救机器人从起始点向目标位置移动，它可以选择的动作有直行、右转和左转。起始阶段，机器人会发生碰撞，经过几次试验后，机器人学会避开墙壁和障碍物。一段时间的探索学习后，搜救机器人可以找到一条通往目标位置的路径。

在具有动态障碍物的环境中，由于无法用图捕捉到搜救机器人的所有动作，所以分别用 4 张图展示机器人的自主学习路径规划过程。

图 7(大小为 30×30)显示了动态障碍物在不断移动，阻止搜救机器人前进的路径。由于搜救机器人能够感知环境中的障碍物，因此能够寻找到正确的前进路线，但是搜救机器人无法预测动态障碍物的运动轨迹，因此在探索过程存在碰撞的风险。最后仿真表明，搜救机器人能够有效地规避静、动两类不同的障碍，并能达到预定的目标位置。

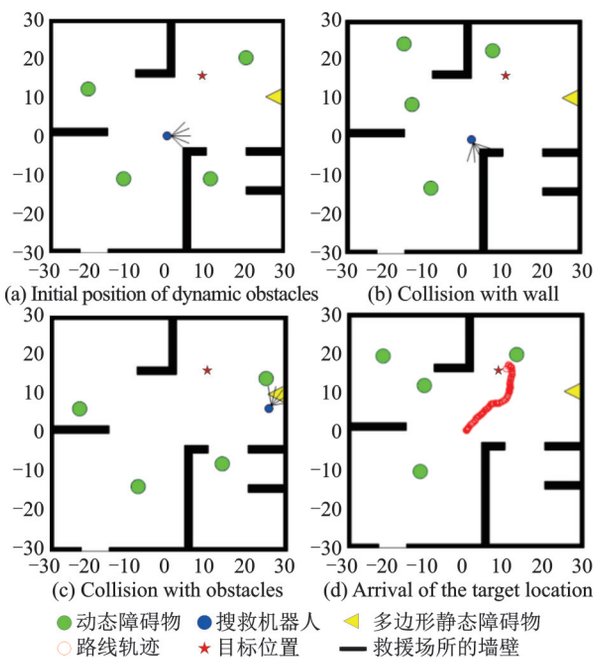

图 7　动态环境的仿真结果

Fig.7　Simulation results for dynamic environments

在实验过程中，针对不同场景的地图进行了广泛测试，并引入了不同的初始点作为对照试组，不同的初始点可以模拟搜救机器人在应急场景中可能面临的多样化启动条件，以更全面地评估提出的路径规划框架的性能。

一次试验是指搜救机器人从起点出发开始，一直到满足终止条件才结束。终止条件是搜救机器人到达终点、在环境中与墙壁或障碍物发生碰撞或到达试验设定的最大步长。试验次数表示在训练过程中，搜救机器人避开障碍到达目标位置的训练次数，碰撞次数指在稳定到达目标位置之前搜救机器人试验过程中发生碰撞的次数，碰撞率是碰撞次数和试验次数的比值。

3.3.1　基于 Q-table 的仿真结果

在基于 Q-table 的仿真实验中，通过使用两个

不同的初始点，即(0,0)和(−20,−20)，分别在地图一和地图二场景中进行了自主路径规划。在不同初始点和不同地图在平均碰撞次数、平均碰撞率、平均步长的表现如表1所示。

表1 不同地图的对比表

Table 1 Comparison table of different maps

地图一						
初始点		状态	试验次数	碰撞次数	碰撞率	步长
x	0	静态	150	28.40	16.67	51
y	0	动态	300	151.80	50.60	67
x	−20	静态	150	35.40	23.60	144
y	−20	动态	300	156.00	52.00	169
地图二						
初始点		状态	试验次数	碰撞次数	碰撞率	步长
x	0	静态	150	45.6	30.4	142
y	0	动态	300	209.8	69.93	163
x	−20	静态	150	49	32.67	230
y	−20	动态	300	214.2	71.4	259

通过图8和图9可以直观地看出，在存在静态障碍物的环境中，搜救机器人在稳定到达目标位置之前具有更低的碰撞率，而且可以通过较少的试验就稳定地到达目标位置。然而，在存在动态障碍物的环境中，不同的试验最终都可以达到稳定状态，但由于动态障碍物的干扰，即使训练稳定后，当出现动态障碍物需要躲避时依旧有可能发生碰撞。因此在动态环境中，收敛过程稳定性相对较差。

探索过程中所获得的奖励值用来表示该方案的稳定性，图10清晰地展示了上述试验过程中抽取的单次试验所获得的奖励值。该方法在初期探索阶段，先对所处环境进行试错，在探索期后，在两个环境中都选择出能够获得最佳奖励的动作。从图10可以明显观察到以下情况：在静态环境中，如图(a)所示，经过相对较少的试验，奖励值趋于稳

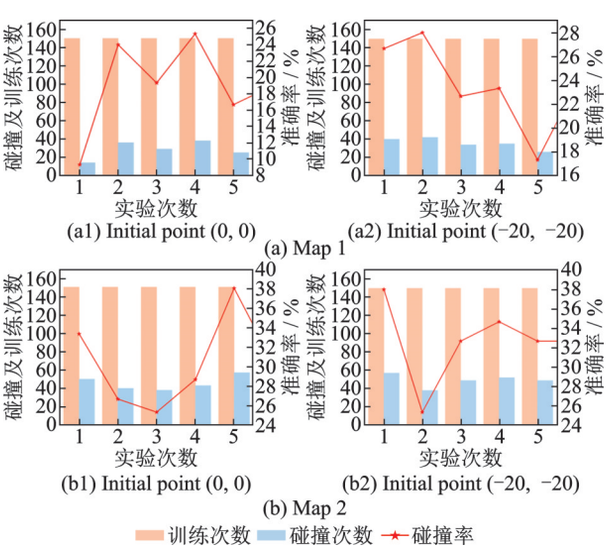

图8 静态障碍物环境中的Q-table仿真结果

Fig.8 Q-table simulation results in static obstacle environments

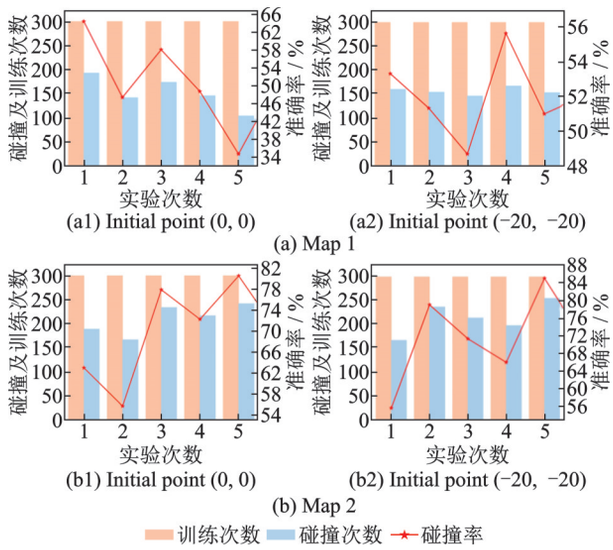

图9 动态障碍物环境中的Q-table仿真结果

Fig.9 Q-table simulation results in dynamic obstacle environments

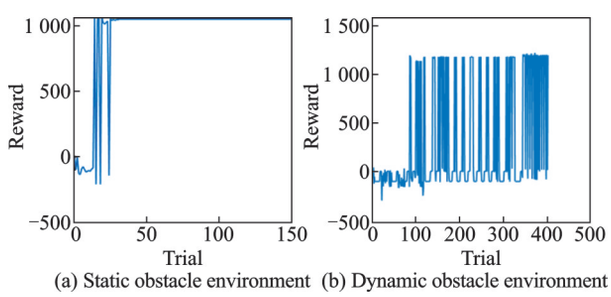

图10 Q-table方法的单次试验奖励

Fig.10 Single trial rewards for Q-table method

定，不再发生变化；而在动态环境，如图(b)所示，通过一定数量的试验，同样可以到达稳定的状态，且到达稳定状态后所获得的奖励值不改变，即不再发生碰撞。

在先前的场景中，障碍物表现出有规律的圆周运动。然而，为了模拟更真实且复杂的环境，还引入了无规则运动的动态障碍物。每个无规则运动的障碍物都被随机分配一个小的旋转角度，在平面上进行不可预测且随机性强的运动。每个障碍物的运动轨迹因此变得不规律，增加了环境的复杂性。这种引入无规则运动的障碍物的变化旨在更全面地评估搜救算法的性能，使其能够在具有挑战性和变化多样的环境中展现鲁棒性和适应性，有助于确保算法在应对真实世界中各种随机性和不确定性时能够可靠地执行任务。

如图11，实验结果表明，在引入动态障碍物进行无规则运动的情况下，搜救机器人依然能够稳定地到达目标位置，尽管试验次数相较于之前的有规律运动的场景增加了。这表明基于Q-table的算法在面对随机性更强、不可预测的动态障碍物时，仍然具备一定的稳健性和适应性。

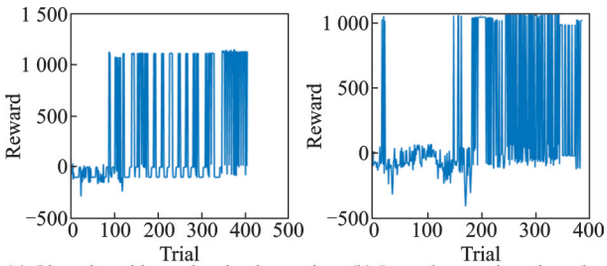

图 11　不同动态障碍物的单次试验奖励

Fig.11　Single trial rewards for different dynamic obstacles

3.3.2　基于神经网络的仿真结果

将神经网络引入 Q-learning 算法后，生成的轨迹仍然代表奖励最大的最佳路径，试验方法与 Q-table 方法一致。

表 2 展示了基于神经网络的算法在不同初始点(0，0)和(－20，－20)以及不同地图场景中进行自主路径规划时的性能表现。本文关注了平均碰撞次数、平均碰撞率和平均步长等指标，以全面评估算法在不同条件下的适应性和效果。

表 2　不同地图的对比表

Table 2　Comparison table of different maps

地图一						
初始点		状态	试验次数	碰撞次数	碰撞率	步长
x	0	静态	500	235.00	47.00	155
y	0	动态	800	314.60	39.33	168
x	－20	静态	500	306.80	65.36	251
y	－20	动态	800	376.00	52.40	275
地图二						
初始点		状态	试验次数	碰撞次数	碰撞率	步长
x	0	静态	500	316	63.20	243
y	0	动态	800	390.4	48.80	261
x	－20	静态	500	373.8	74.76	321
y	－20	动态	800	429.2	53.65	346

通过对实验结果的分析，初始点、地图结构和动态障碍物对基于神经网络的路径规划算法性能的显著影响。在地图一中，初始点为(0，0)时，算法表现出较低的碰撞次数和碰撞率，而初始点离目标位置距离变远时，碰撞次数和碰撞率上升，这突显了初始点选择对于算法性能的关键影响。在地图二中，初始点为(0，0)时，相同的试验次数内却具有较高的碰撞率，说明地图结构对路径规划的适应性也是一个关键因素。引入动态障碍物导致了更高的碰撞次数和碰撞率，验证了动态环境增加了路径规划的复杂性。

图 12 和图 13 显示了使用基于神经网络的 Q-learning 方法在静态和动态环境中，经过一段时间的学习后，搜救机器人的试验次数、碰撞次数和碰撞率。在具有静态障碍物的环境中，达到稳定时所需要的实验次数更少，且在达到稳定之前的碰撞率更低。然而，在存在动态障碍物的环境中，由于动态障碍物的干扰，搜救机器人即使训练稳定后，每次试验到达目标位置所需的步数和获得的奖励值都可能不同，甚至可能会发生碰撞。因此在一次训练过程中，如果搜救机器人在某次试验时可以从起点到达终点，然后在接下来的试验中能够多次成功到达目标位置，就可以认为搜救机器人已经达到了稳定状态。

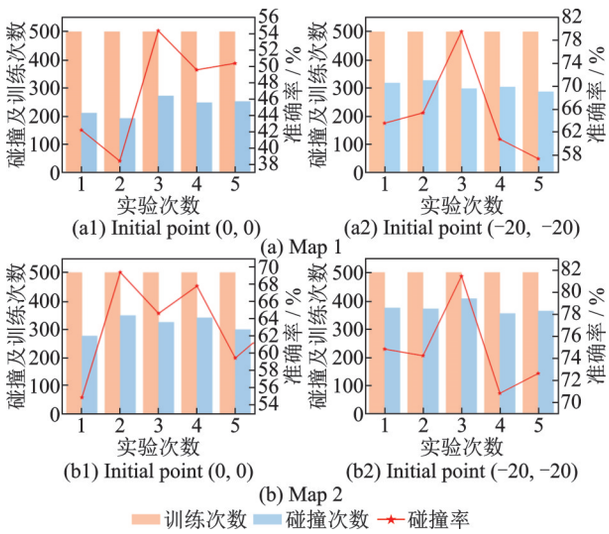

图 12　静态障碍物环境中的神经网络仿真结果

Fig.12　Neural network simulation results in static obstacle environments

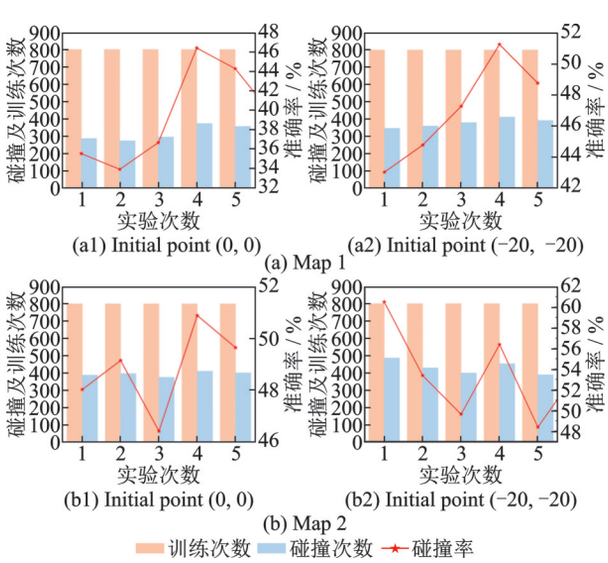

图 13　动态障碍物环境中的神经网络仿真结果

Fig.13　Neural network simulation results in dynamic obstacle environments

使用基于神经网络的 Q-learning 方法的单次试验奖励值，如图 14 所示。和 Q-table 方法类似，在静态障碍物环境中，搜救机器人经过一段时间的学习后可以稳定到达目标位置。而在动态障碍物

环境中,搜救机器人到达目标位置后会随机进行一次探索,增加了探索率,以便于可以找到更优路径。总的来说,使用基于神经网络的Q-learning方法在两个环境中,经过一段时间的学习后都可以收敛到一个解决方案,稳定地到达目标位置。

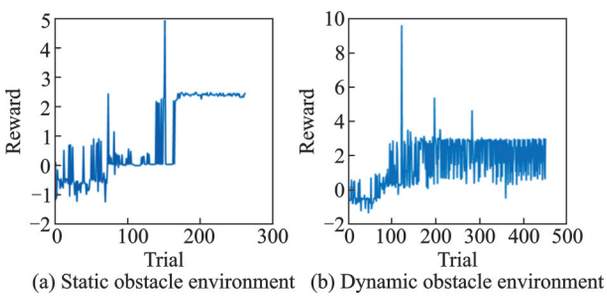

图 14 基于神经网络方法的单次试验奖励

Fig.14 Single trial rewards based on neural network method

如图 15,基于神经网络方法的实验结果同样显示,搜救机器人在面对动态无规则运动的障碍物时展现出了鲁棒性。尽管在这种具有更高随机性和不可预测性的环境下,实验次数相对增加,但神经网络算法能够有效适应这种变化,成功地导航机器人到达目标位置。

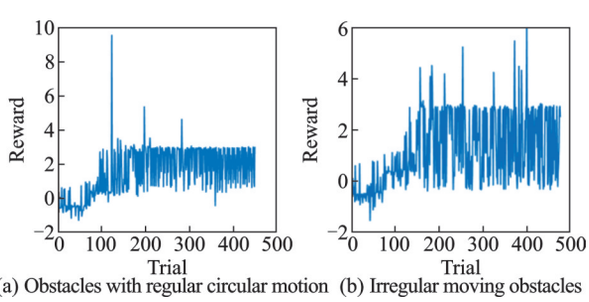

图 15 神经网络中不同动态障碍物的单次试验奖励

Fig.15 Single-trial rewards for different dynamic obstacles in neural networks

3.3.3 稀疏奖励后的仿真结果

在路径规划实验中,引入了一种稀疏奖励机制,修改奖励函数的分配方式,将奖励更加有选择性地提供,只有特定条件下才给予奖励。目的就是为了让搜救机器人更难以确定正确的动作和策略,增加学习的难度。因此奖励函数设置为只有当搜救机器人成功抵达目标位置时,系统会提供正向奖励,发生碰撞后,系统会提供惩罚函数。另外为了引导机器人逐步逼近目标,可以在机器人逼近目标位置的过程中,对 3 个动作选择列表提供小额奖励。

在地图一,初始点为(0,0)的场景中引入稀疏奖励机制的实验结果如图 16 所示。实验结果表明,搜救机器人在路径规划任务中面临更大的挑战。观察发现,引入稀疏奖励机制导致搜救机器人尝试到达目标位置的次数显著增加,而机器人在搜救刹那间中表现出不断探索的趋势,直至达到试验终止条件。

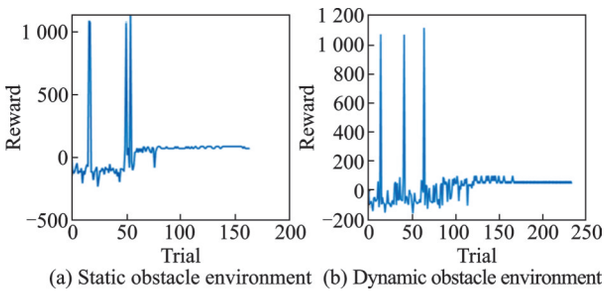

图 16 稀疏奖励机制的单次试验奖励

Fig.16 Single trial rewards for sparse reward mechanism

通过引入稀疏奖励机制,成功证明了本文提出的奖励机制更为合理和有效,更好地引导了机器人的学习过程。实验结果表明,在特定条件下提供奖励,使得机器人需要更多的尝试来成功到达目标位置。奖励值的不稳定性和探索性增加表明了稀疏奖励机制使得机器人更难以确定正确的动作和策略,反而更加注重探索环境。

3.3.4 大型环境下搜救机器人挑战与应对

在大型环境下进行搜救任务是一项极富挑战性的任务。在庞大的环境中,机器人需要面对更多复杂的地形和障碍物,以及更远距离的目标位置,这使得路径规划和决策过程变得更为复杂和耗时。

首先,地图尺寸的增加意味着机器人需要更广泛、更深入的探索环境,以了解地形、障碍物分布以及可能的目标位置。这涉及更多的试错,因此机器人需要不断调整和优化其行动策略,以适应地图的广泛范围。其次,大型地图引入了更多的动态性,增加了环境的不确定性。搜救机器人在执行任务时需要更灵活地应对这种不可预测性,可能需要更频繁地调整其路径规划和决策,以适应环境的变化。此外,大尺寸地图还可能导致搜救机器人在寻找目标位置时面临更长的路径选择,这需要强化学习算法具备更好的记忆和规划能力,以确保机器人能够有效地探索大范围的地图并最终达到目标。

基于以上挑战,可以采用一种创新的阶段目标点方法,以引导搜救机器人更有效地应对环境复杂性。这方法将大型地图划分成多个区域,并在这些区域之间引入引导点,从而形成一个阶段性的任务执行框架。

在每次试验开始时,搜救机器人通过传感器获取环境信息和当前目标点的位置信息。经过先前的训练学习后,机器人能够实时判断是否已达到预设的引导点。若搜救机器人成功到达引导点,系统

将切换至下一个阶段的目标点，从而实现任务的分阶段完成；若搜救机器人未达到预设引导点，其将持续向当前引导点靠近。这个智能化的引导系统使得搜救机器人能够根据环境变化和障碍物分布，有序地规划路径并调整决策，以更高效地实现大范围地图的探索与目标达成。

4　结　　论

为确保应急救援任务的顺利执行，搜救机器人必须在不确定的环境中进行自主的路径规划，以在避开静态和动态障碍物的前提下成功到达目标位置。为此，本文采用了基于Q-table和基于神经网络的Q-learning算法，这使得搜救机器人能够通过与外界环境的交互式反馈机制，动态地调整其动作。这种自主适应性使得机器人能够更好地适应不断变化的环境条件。引入了传感器配置方案，这意味着机器人能够通过合适的传感器感知外部环境，有助于提高环境感知的精度，进而提升路径规划的准确性。

本文还构建了小车的动力学模型以及基于强化学习的搜救机器人路径规划框架。在这个框架下，动力学模型为路径规划提供了更真实的运动模拟基础，使得路径规划更符合实际运动情况。在静态障碍物环境中的仿真结果显示，Q-table方法具有更高的学习效率，并且更快地达到稳定状态，且收敛过程较为稳定。相比之下，在动态障碍物环境下，需要考虑更多的因素，同时需要更大的状态表，这意味着系统需要更多的时间来收敛到一个解决方案。试验结果表明Q-table清晰易懂，适用简化的、输入量较小的自学系统，神经网络适用于更大和更复杂的系统。

然而，当前算法在搜救机器人需要快速做出反应以避免动态障碍物时表现出一些不足。算法采用了一种需要停止搜救机器人并进行两次测量的策略，以确定障碍物是移动的还是静止的。这导致搜救机器人在判断障碍物性质时需要额外的时间，从而减缓了其对动态环境的适应能力。这种策略会导致搜救机器人的运动模式呈现出不连续的特征，在实际场景中运动通常是连续的。未来工作的重点是改进搜救机器人的碰撞检测，提高运行速度，并优化奖励函数以减弱复杂环境的影响。增强奖励函数的任务相关性，考虑任务中的关键因素，以确保奖励函数能够有效地引导机器人学习。另外，还计划扩展机器人的动作列表，包括停止、加速、减速等动作，以更全面地模拟实际场景中的机器人行为，从而提高其适应性和灵活性。

参考文献：

[1] 郭娜，李彩虹，王迪，等. 结合预测和模糊控制的移动机器人路径规划[J]. 计算机工程与应用，2020，56(8)：104-109.
GUO Na，LI Caihong，WANG Di，et al. Path planning for mobile robots combining predictive and fuzzy control[J]. Computer Engineering and Applications，2020，56(8)：104-109.

[2] 王珂，穆朝絮，蔡光斌，等. 基于安全自适应强化学习的自主避障控制方法[J]. 中国科学：信息科学，2022，52(9)：1672-1686.
WANG Ke，MU Chaoxu，CAI Guangbin，et al. Autonomous obstacle avoidance control method based on secure adaptive reinforcement learning[J]. Chinese Science：Information Science，2022，52(9)：1672-1686.

[3] AI Bo，JIA Maoxin，XU Hanwen，et al. Coverage path planning for maritime search and rescue using reinforcement learning[J]. Ocean Engineering，2021，241(1)：110098.

[4] 段建民，陈强龙. 利用先验知识的Q-learning路径规划算法研究[J]. 电光与控制，2019，26(9)：29-33.
DUAN Jianmin，CHEN Qianglong. Research on Q-learning path planning algorithm using prior knowledge[J]. Electro Optics and Control，2019，26(9)：29-33.

[5] CHEN C，CHEN X Q，MA F，et al. A knowledge-free path planning approach for smart ships based on reinforcement learning[J]. Ocean Engineering，2019，189：106299.

[6] 王兵，吴洪亮，牛新征. 基于改进势场法的机器人路径规划[J]. 计算机科学，2022，49(7)：196-203.
WANG Bing，WU Hongliang，NIU Xinzheng. Robot path planning based on improved potential field method[J]. Computer Science，2022，49(7)：196-203.

[7] 宋勇，李贻斌，李彩虹. 移动机器人路径规划强化学习的初始化[J]. 控制理论与应用，2012，29(12)：1623-1628.
SONG Yong，LI Yibin，LI Caihong. Initialization in reinforcement learning for mobile robots path planning[J]. Control Theory & Applications，2012，29(12)：1623-1628.

[8] 卫玉梁，靳伍银. 基于神经网络Q-learning算法的智能车路径规划[J]. 火力与指挥控制，2019(2)：46-49.
WEI Yuliang，JIN Wuyin. Intelligent vehicle path planning based on neural network Q-learning algorithm[J]. Fire and Command Control，2019(2)：46-49.

[9] LI Z，LIU W，LI L，et al. Path following method for AUV based on Q-learning and RBF neural network[J]. Journal of Northwestern Polytechnical University，2021，39(3)：477-483.

[10] 徐晓苏，袁杰. 基于改进强化学习的移动机器人路径

规划方法[J]. 中国惯性技术学报, 2019, 27(3): 314-320.
XU Xiaosu, YUAN Jie. A path planning method for mobile robots based on improved reinforcement learning[J]. Chinese Journal of Inertial Technology, 2019, 27(3): 314-320.

[11] ASLI A E N, ROGHAIR J, JANNESARI A. Energy-aware goal selection and path planning of UAV systems via reinforcement learning[J]. arXiv-CS-Artificial Intelligence, 2019. DOI: arxiv-1909.12217.

[12] WANG Y H, LI T H S, LIN C J. Backward Q-learning: The combination of Sarsa algorithm and Q-learning[J]. Engineering Applications of Artificial Intelligence, 2013, 26(9): 2184-2193.

[13] 尹旷, 王红斌, 方健, 等. 基于强化学习的移动机器人路径规划优化[J]. 电子测量技术, 2021, 44(10): 91-95.
YIN Kuang, WANG Hongbin, FANG Jian, et al. Optimization of mobile robot path planning based on reinforcement learning[J]. Electronic Measurement Technology, 2021, 44(10): 91-95.

[14] 王健, 张平陆, 赵忠英, 等. 结合神经网络和Q(λ)-learning的路径规划方法[J]. 自动化与仪表, 2019, 34(9): 1-4.
WANG Jian, ZHANG Pinglu, ZHAO Zhongying, et al. Path planning method based on neural network and Q(λ)-learning[J]. Automation and Instrumentation, 2019, 34(9): 1-4.

[15] MAOUDJ A, HENTOUT A. Optimal path planning approach based on Q-learning algorithm for mobile robots[J].Applied Soft Computing, 2020, 97: 106796.

[16] 黄晓冬, 苑海涛, 毕敬, 等. 基于DQN的海战场舰船路径规划及仿真[J]. 系统仿真学报, 2021, 33(10): 2440-2448.
HUANG Xiaodong, YUAN Haitao, BI Jing, et al. DQN based ship path planning and simulationin naval battlefields[J]. Journal of System Simulation, 2021, 33(10): 2440-2448.

(编辑:胥橙庭)