面向无人艇的 T-DQN 智能避障算法研究

周治国[1] 余思雨[1] 于家宝[1] 段俊伟[2] 陈 龙[3] 陈俊龙[4]

摘 要 无人艇 (Unmanned surface vehicle, USV) 作为一种具有广泛应用前景的无人系统, 其自主决策能力尤为关键. 由于水面运动环境较为开阔, 传统避障决策算法难以在量化规则下自主规划最优路线, 而一般强化学习方法在大范围复杂环境下难以快速收敛. 针对这些问题, 提出一种基于阈值的深度 Q 网络避障算法 (Threshold deep Q network, T-DQN), 在深度 Q 网络 (Deep Q network, DQN) 基础上增加长短期记忆网络 (Long short-term memory, LSTM) 来保存训练信息, 并设定经验回放池阈值加速算法的收敛. 通过在不同尺度的栅格环境中进行实验仿真, 实验结果表明, T-DQN 算法能快速地收敛到最优路径, 其整体收敛步数相比 Q-learning 算法和 DQN 算法, 分别减少 69.1% 和 24.8%, 引入的阈值筛选机制使整体收敛步数降低 41.1%. 在 Unity 3D 强化学习仿真平台, 验证了复杂地图场景下的避障任务完成情况, 实验结果表明, 该算法能实现无人艇的精细化避障和智能安全行驶.

关键词 无人艇, 强化学习, 智能避障, 深度 Q 网络

引用格式 周治国, 余思雨, 于家宝, 段俊伟, 陈龙, 陈俊龙. 面向无人艇的 T-DQN 智能避障算法研究. 自动化学报, 2023, **49**(8): 1645−1655

DOI 10.16383/j.aas.c210080

Research on T-DQN Intelligent Obstacle Avoidance Algorithm of Unmanned Surface Vehicle

ZHOU Zhi-Guo[1] YU Si-Yu[1] YU Jia-Bao[1] DUAN Jun-Wei[2] CHEN Long[3] CHEN Jun-Long[4]

Abstract Unmanned surface vehicle (USV) is a kind of unmanned system with wide application prospect, and it is important to train the autonomous decision-making ability. Due to the wide water surface motion environment, traditional obstacle avoidance algorithms are difficult to independently plan a reasonable route under quantitative rules, while the general reinforcement learning methods are difficult to converge quickly in large and complex environment. To solve these problems, we propose a threshold deep Q network (T-DQN) algorithm, by adding long short-term memory (LSTM) network on basis of deep Q network (DQN), to save training information, and setting proper threshold value of experience replay pool to accelerate convergence. We conducted simulation experiments in different sizes grid, and the results show T-DQN method can converge to optimal path quickly, compared with the Q-learning and DQN, the number of convergence episodes is reduced by 69.1%, and 24.8%, respectively. The threshold mechanism reduces overall convergence steps by 41.1%. We also verified the algorithm in Unity 3D reinforcement learning simulation platform to investigate the completion of obstacle avoidance tasks under complex maps, the experiment results show that the algorithm can realize detailed obstacle avoidance and intelligent safe navigation.

Key words Unmanned surface vehicle (USV), reinforcement learning, intelligent obstacle avoidance, deep Q network (DQN)

Citation Zhou Zhi-Guo, Yu Si-Yu, Yu Jia-Bao, Duan Jun-Wei, Chen Long, Chen Jun-Long. Research on T-DQN intelligent obstacle avoidance algorithm of unmanned surface vehicle. *Acta Automatica Sinica*, 2023, **49**(8): 1645−1655

水面无人艇 (Unmanned surface vehicle, USV) 是一种无人驾驶的水面航行器, 因其智能程度高、隐藏性高、移动能力强等特点[1−3], 成为执行搜救、侦察、监测、舰艇护航等任务的重要平台[4]. 为满足多种任务的需求, USV 的研究与设计主要包括多源信息融合、目标识别跟踪、自主路径规划等方面[5−7].

收稿日期 2021-01-25 录用日期 2021-06-25
Manuscript received January 25, 2021; accepted June 25, 2021
"十三五" 装备预研领域基金 (61403120109), 暨南大学中央高校基本科研业务费专项资金 (21619412) 资助
Supported by Equipment Pre-research Field Fund Thirteen Five-year (61403120109) and Fundamental Research Funds for the Central Universities of Jinan University (21619412)
本文责任编委 李力
Recommended by Associate Editor LI Li
1. 北京理工大学信息与电子学院 北京 100081 2. 暨南大学信息科学技术学院 广州 510532 3. 澳门大学科技学院 澳门 999078 4. 华南理工大学计算机科学与工程学院 广州 510006
1. School of Information and Electronics, Beijing Institute of Technology, Beijing 100081 2. College of Information Science and Technology, Jinan University, Guangzhou 510532 3. Faculty of Science and Technology, University of Macau, Macau 999078 4. School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006

其中, 自主决策作为衡量无人艇智能化程度的重要标准之一, 要求无人艇在静态水面环境中, 能够按照最优规划从起点行驶到终点, 同时在遇到未知危险时能够迅速地避开障碍物[3]. 避障决策包括利用已知环境的全局路径规划和遇到不明障碍物的局部避障两个部分, 其中针对全局路径规划已有较为成熟的算法 (如 A*[8]、Dijkstra[9] 等), 这些算法大多适用于无障碍或稀疏障碍等简单静态海洋环境[10]. 然而, 无人艇航行所在的环境往往存在不可预测的障碍物, 因此局部路径规划依赖于传感器获取局部环境信息, 需要重点关注面对未知环境时的适应能力和算法的避障能力. 强化学习方法在应对未知障碍物时有较好的自适应性能, 将强化学习与深度学习相结合得到的深度强化学习, 适用于处理制定决策和运动规划. 与传统避障方法相比, 基于深度强化学习的局部避障方法具有更高的智能性, 对未知环境有更强的适应性, 因此成为近年来的研究热点, 并逐渐在机器人控制和运动规划中得到广泛应用[11−12].

强化学习是机器学习方法中的一种重要方法, 主要由智能体、环境、动作、状态和奖励 5 个部分组成[13−14]. 智能体与环境不断进行交互, 其核心在于获得环境的观测值, 根据策略采取一系列动作并得到相应的奖励. 经过接连不断的交互过程, 智能体最大限度地积累奖励, 并学会在环境中采取最佳决策完成任务. 在路径规划问题中, 强化学习表现出一定程度的优越性[15−17]. Chen 等[18] 提出一种基于深度强化学习的分布式避障算法, 显著地减少智能体完成避障任务抵达目的地的时间; Tai 等[19] 提出一种深度强化学习运动规划算法, 无需借助地图信息, 利用深度确定性策略梯度算法的异步版本让智能体学习导航避障; Zhang 等[20] 提出一种基于后继特征的避障算法, 学习将先验知识从已完成的避障导航任务迁移到新的实例中, 减少试错成本. 针对深度 Q 网络 (Deep Q network, DQN) 学习算法的变式, Matthew 等[21] 用长短期记忆网络 (Long short-term memory, LSTM) 替换深度 Q 网络第 1 个后卷积全连层, 提出深度循环 Q 网络 (Deep recurrent Q network, DRQN), 解决 DQN 经验池内存限制和部分可观测马尔科夫决策过程 (Markov decision processes, MDP) 中难以获得全部可观测信息的问题. Liu 等[22] 将 DRQN 和深度双 Q 网络 (Double DQN, DDQN) 两种算法用于路径规划, 通过对比实验可以看出, 尽管 DRQN 算法具有更好的决策和路径选择能力, 但因为消耗更多的存储空间和计算资源, 收敛时间更长. Wang 等[23] 利用 LSTM 保存历史状态, 并结合 DQN 学习车道合并驾驶策略, 融合历史驾驶经验和交互驾驶行为的影响, 有助于智能体适应自动驾驶中复杂变道场景. Deshpande 等[24] 将 LSTM、DQN 与比例−积分−微分控制器纵向控制器结合, 测试算法在拥挤城市环境无信号交叉路口的自动驾驶任务完成性能, 验证算法能辅助实现安全驾驶利用 LSTM 保存历史状态并结合 DQN 学习车道合并驾驶策略. 该方法融合历史驾驶经验和交互驾驶行为的影响, 有助于智能体适应自动驾驶中复杂变道场景. Peixoto 等[25] 采用结合 LSTM 的 DQN 算法实现无人车辆驾驶, 基于过往训练状态感知实现车辆在复杂环境中的自主决策. 通过对比实验可以看出, 相同的训练次数, LSTM + DQN 成功率达到 5.12%, 而 DQN 只有 1.47%, 说明 DRQN 算法在部分可观测环境中的自适应性更强.

针对本文讨论的无人艇避障, 传统强化学习算法和现有的研究工作未能充分考虑无人艇相比其他无人系统不同的感知范围和运动特性, 存在以下问题: 1) 感知范围决定仿真粒度, 一般强化学习避障算法并未考虑到局部避障与局部感知区域的适配性, 因此仿真粒度设置存在一些不合理之处; 2) 一般强化学习算法泛化能力较差, 在面对未知环境时, 需要消耗大量时间重新规划, 在高维度空间下进行解算时容易陷入瓶颈, 因此对新环境的泛化能力还需要进一步提高.

本文提出一种基于阈值的深度 Q 网络 (Threshold deep Q network, T-DQN) 算法, 通过增加 LSTM 网络保存训练信息, 并设定合理的阈值筛选经验回放样本, 加速避障算法的收敛. 本文对航行决策过程进行详细描述, 仿真实验验证了本文算法的有效性. 根据无人艇实际感知范围, 设置不同的栅格大小和仿真粒度, 并进行对比仿真. 同时在 Unity 3D 仿真平台中进行实验验证, 考察复杂地图场景下的避障任务完成情况. 实验结果表明, 该算法能辅助实现无人艇的精细化避障和智能安全行驶, 且仿真具有较好的真实度和视觉效果.

1 算法原理

无人艇的避障决策模型是基于马尔科夫决策过程建立的. MDP 是解决强化学习相关问题的框架, 也是强化学习底层的数学模型[26−27]. 本节分析马尔科夫底层决策模型, 针对无人艇运动特性设置状态空间、动作空间和奖励函数, 并提出 T-DQN 算法用于求解策略 π, 使无人艇在开阔环境中, 在较少的训练步数内, 就能收敛到最优路径.

1.1 MDP 决策模型

一次 MDP 代表强化学习过程中的一次状态转

移过程. MDP 由五元组 (S, A, P, R, γ) 描述, 其中, S 表示有限的状态集, A 表示有限的动作集. 智能体正是通过在动作空间中选取合理的动作来得到奖赏最大化结果. P 表示状态转移概率, R 表示回报函数, $\gamma \in [0, 1)$ 是计算累积回报 G 的折扣因子, 作为参数调节每一次行为后获得回报的递减.

MDP 的核心是寻找最优策略 π. 策略 π 实际上是动作的执行概率, 取值范围为状态集 S 到动作集 A 的映射, 在每个状态 s 下, 策略 π 可表示为:

$$\pi(a|s) = p[A_t = a|S_t = s] \tag{1}$$

据此, MDP 过程可以表述为: 在每个时刻的状态下, 指定一个动作, 获取该状态的回报值, 同时根据策略转移至下一个状态; 循环上述过程, 直到累积回报最大. 每个状态的回报定义为状态值函数:

$$v_\pi(s) = \mathrm{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|s_t = s\right] \tag{2}$$

累积回报是一种评价性指标, 代表智能体从初始状态开始所获得的累积回报期望值, 其中 R 表示奖励函数, 用于对智能体在给定策略下的即时效果做出评价. 累积回报定义为:

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^n R_{t+n+1} + \cdots = \sum_{k=0}^{\infty}\gamma^k R_{t+k+1} \tag{3}$$

通过对 MDP 的定义, 强化学习可以表示为: 在给定的 MDP 中, 寻找最优策略, 使该策略作用下的累积回报期望值最大化. 在状态 s 下采取动作 a, 并遵循策略 π 得到的相应期望值定义为行动价值函数:

$$q_\pi(s, a) = \mathrm{E}_\pi(G_t|S_t = s, A_t = a) \tag{4}$$

在用强化学习解决导航、规划和控制等问题时, 目标是寻求最佳策略, 确定最佳策略后, 累计奖励值和状态值函数能够达到最大, 智能体能够在环境中进行决策, 选取最优动作直到完成任务. 确定最佳策略的常见方法包括贪婪策略、ε-greedy 策略、高斯策略、Boltzmann 分布等, 通过这些方法指导智能体在强化学习策略更新过程中如何选取动作.

航行决策问题可以看成是连续动作空间中的控制: 给定观察的状态为 S, 决策问题可以被定义为一个非线性映射 π: $S \rightarrow A$. π 为控制策略, 其中 S 为 MDP 过程中的环境状态信息, A 为决策输出的指令. 根据无人艇的运动特性, 该指令控制无人艇在复杂环境中完成路径规划和避障等任务. 本文在 MDP 基础上, 考虑到无人艇的运动特性和决策过程, 针对性地设置其状态空间、动作空间和航行时的奖励函数.

1.1.1　状态空间建模

状态空间包含环境中智能体的属性和信息, 以此为依据来选取要执行的动作. 为实现无人艇避障, 需获取无人艇当前位置、运动速度和终点目标位置. 无人艇在环境中航行时, 由激光雷达探测障碍物. 根据探测结果, 为对应位置的栅格赋值. 将栅格环境中不可通行的区域设置为 1, 可通行区域设为 0, 整个栅格环境由 0, 1 组成的二维矩阵表示. 对于 $M \times M$ 大小的栅格, $i = 1, 2, \cdots, (M \times M)/2$, 其中 i 为算法能处理的障碍物. 局部避障时, 通过计算坐标的欧氏距离, 得到智能体和障碍物的相隔距离. 设 s_{usv} 为无人艇当前位置状态函数, 表示无人艇当前位置 (x_{usv}, y_{usv}). s_{goal} 为目标点位置状态函数, 表示终点目标位置 (x_{goal}, y_{goal}). $s_{velocity}$ 为无人艇的运动速度状态函数, 表示无人艇运动速度 $(v_{usv\text{-}x}, v_{usv\text{-}y})$. $s_{obstacle}$ 为障碍物状态函数, 由 0, 1 组成的二维矩阵来表示, 对应 $(x_{obstacle}, y_{obstacle})$ 的位置值设置为 1. 综上, MDP 中的状态空间如下:

$$s = \{s_{usv}, s_{goal}, s_{obstacle}, s_{velocity}\} \tag{5}$$

1.1.2　动作空间设计

MDP 中的动作空间对应决策的输出, 指导无人艇在环境中采取动作. 将环境区域划分为相同大小的栅格, 输出动作指令定义为第一人称视角的上、下、左、右, 控制无人艇在栅格中进行垂直和水平方向移动. 将动作空间离散化的原因: 一是方便在栅格环境中进行仿真, 4 个动作符合智能体在栅格中的行进规则; 二是连续动作空间计算量较大, 收敛速度慢, 且无人艇由于受到水浪阻力, 运动轨迹无法和理想状态保持一致, 为仿真引入较多不确定因素, 难以控制. 综上, 本文采用栅格法和离散动作空间来对问题进行抽象描述.

1.1.3　奖励函数设计

奖励函数的设计决定智能体在环境中的表现水平以及能否学习到最优策略. 本文在设计奖励函数时, 考虑无人艇局部路径规划过程的多个方面, 包括向目标点航行、规避障碍物、以较小时间和路径代价到达目标点等.

当无人艇所在位置与目标点之间的距离缩短时, 代表无人艇正在接近目标点. 为使无人艇靠近目标点, 设计奖励 $R_{distance}$, 无人艇与目标点距离越近时收到的奖励越大. $R_{distance}$ 的计算公式如式 (6) 所示. 设置无人艇在栅格中移动的最小步长为 1 m, 近似无人艇自身长度. 其中 $1 < m < 100$, 为奖励参数, 在仿真实验中设为 10. 当无人艇未到达目标

点时, 根据计算公式, 距离越近, 则奖励值越大, 但小于 m 值. 当无人艇到达时, 无人艇将收到奖励 R_{end}. 在仿真实验中设置 $R_{end}=100$.

$$R_{distance}=\begin{cases}\dfrac{m}{\sqrt{(x_{usv}-x_{goal})^2+(y_{usv}-y_{goal})^2}}, \\ \qquad (x_{usv},\, y_{usv})\neq(x_{goal},\, y_{goal}) \\ N, \qquad (x_{usv},\, y_{usv})=(x_{goal},\, y_{goal})\end{cases} \tag{6}$$

无人艇通过激光雷达传感器检测局部障碍物, 无人艇获取到局部障碍物的信息后, 将采取避障动作. 通过 $R_{collision}$ 给予无人艇惩罚, 从而训练无人艇完成障碍物规避过程. 本文对 $R_{collision}$ 进行设计时, 考虑到障碍物和无人艇之间距离会造成不同程度的威胁, 当无人艇与障碍物之间的距离越近时, 无人艇收到的惩罚越大, 促使无人艇尽快避开局部障碍物, 提高安全性. 同样设置无人艇在栅格中移动的最小步长为 1 m, 近似无人艇自身长度; 无人艇和障碍物之间的距离至少为一个移动步长. $R_{collision}$ 的计算公式见式 (7), 在栅格区域内, 黑色栅格代表障碍物, 奖励值为 N, 仿真中设 $N=-100$.

$$R_{collision}=\begin{cases}\dfrac{-m}{\sqrt{(x_{usv}-x_{obstacle})^2+(y_{usv}-y_{obstacle})^2}}, \\ \qquad (x_{usv},\, y_{usv})\neq(x_{obstacle},\, y_{obstacle}) \\ N, \quad (x_{usv},\, y_{usv})=(x_{obstacle},\, y_{obstacle})\end{cases} \tag{7}$$

为缩短无人艇到达目标点的时间, 无人艇每采取一次动作后, 都将收到惩罚 R_{time}.

在无人艇局部路径规划中, 上述几种因素影响程度不同, 因此这些奖励在组合成最终的奖励函数之前, 需要增加相应的权重. 在算法中将多个权重调整至合适的大小, 以实现较优的局部路径规划结果. 整体奖励函数如下:

$$R=\lambda^{\mathrm{T}}R=\begin{bmatrix}\lambda_{distance}\\ \lambda_{collision}\\ \lambda_{time}\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}R_{distance}\\ R_{collision}\\ R_{time}\end{bmatrix} \tag{8}$$

1.2 T-DQN 算法

DQN 算法是求解 MDP 问题最优策略的一种深度强化学习方法. 利用经验回放和目标网络两个技术, 保证在使用非线性函数逼近器的情况下动作值函数的收敛[28−29]. DQN 实现一种端到端的训练方式, 仅需要很少的先验知识, 便能够在复杂任务中表现出色.

DQN 算法使用包含四元组 $(s,\, a,\, r,\, s')$ 的经验重放缓冲区, 在每一个训练周期内, 智能体与环境交互得到的数据样本存储在经验重放缓冲区. 由于神经网络的训练样本相互独立, 而强化学习中前后状态具有相关性, 通过随机选取过去的状态进行学习, 可以打乱训练样本之间的相关性, 使训练更有效率.

采用 DQN 算法实现无人艇的路径规划, 会存在下列问题: 由于 DQN 是端到端的决策方法, 在做避障决策时, 无法观测到全局环境信息, 因此无法获取完整的环境特性, 导致值函数收敛效果较差, 决策不稳定. 而且 DQN 中的经验回放策略, 对样本的选择是随机的, 没有考虑到数据的无效性, 导致环境地图变大时, 算法不能很好地收敛, 智能体与环境交互所产生的数据不能得到充分利用.

针对上述问题, 设计基于阈值筛选的 T-DQN 算法. 从两个方面进行改进: 1) 引入 LSTM 网络保存过往训练信息; 2) 加入阈值筛选机制对经验回放策略进行调整. 算法框架如图 1 所示, 其中 $Q(s,\, a)$ 和 $Q(s',\, a')$ 分别是行动价值函数的估计网络和目标网络, 根据贝尔曼方程, 两者计算公式分别为:

$$Q(s_t,\, a_t)=r+\gamma\max_{a}Q(s_{t+1},\, a_{t+1}) \tag{9}$$

$$Q(s'_t,\, a'_t)=r+\gamma\max_{a'}Q\left(s'_{t+1},\, a'_{t+1}\right) \tag{10}$$

在 T-DQN 网络训练方面, 损失函数的计算公式为:

$$L(\theta)=\mathrm{E}\left[\left(r+\gamma\max_{a'}Q(s',\, a'\,|\theta)-Q(s,\, a\,|\theta)\right)^2\right] \tag{11}$$

T-DQN 算法采用神经网络拟合表征值函数, 神经网络中的权重用 θ 表示, 值函数表示为 $Q(s,\, a\,|\,\theta)$. 在训练网络的过程中, 更新迭代参数 θ 确定时, 表示智能体在该状态下学习到动作策略. 其中 θ 的更新由小批量随机梯度下降实现:

$$\nabla L(\theta)=\mathrm{E}\Big[\Big(r+\gamma\max_{a'}Q(s',\, a'|\theta)-Q(s,\, a|\theta)\nabla Q(s,\, a|\theta)\Big)\Big] \tag{12}$$

$Q(s,\, a)$ 是在当前参数 θ 下对 $Q(s',\, a')$ 的逼近/预测, 所以更新 θ 的目标就是让 $Q(s,\, a)$ 逼近 $Q(s',\, a')$, 根据贝尔曼迭代式优化损失函数, 收敛到最优值函数上.

DQN 算法没有考虑到动作前后的相关性, 在缺失全局地图情况下, 无人艇在面对从未见过的障碍物时, 会做出大量的尝试后才学习到最优的决策. 因此, 本文为 DQN 决策模型添加 LSTM 网络, 用于处理环境状态信息的输入. LSTM 是循环神经网

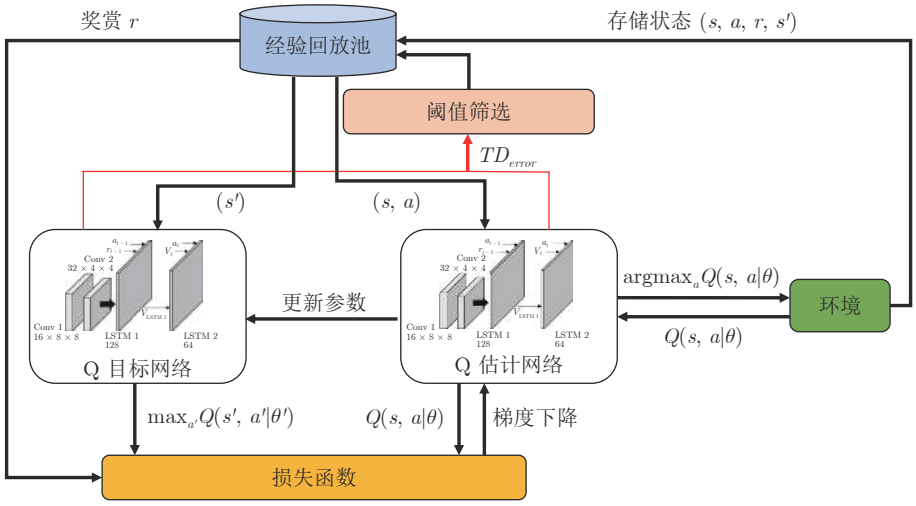


图 1　T-DQN 算法架构图

Fig. 1　T-DQN algorithm architecture

络的一种形式, 结构如图 2 所示. 通过“门”来控制丢弃或增加信息, 从而实现遗忘或记忆的功能, 适合局部可观测的强化学习问题.

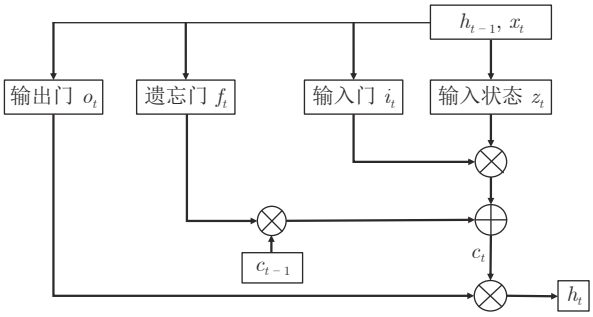


图 2　LSTM 网络结构图

Fig. 2　LSTM network structure

传统的 DQN 算法中, 在卷积结构之后使用一层全连接输出到动作空间. 本文设计 2 层 LSTM 替代全连接层, 第 1 层 LSTM 接收来自卷积的深度特征; 第 2 层接收来自上一层 LSTM 的输出 $V_{\text{LSTM 1}}$、当前执行动作和状态作为输入. 具体参数设置为: 第 1 层神经元个数为 128, 时序长度为 4, 输出为 32; 第 2 层神经元个数为 64, 时序长度为 4, 输出为 1. LSTM 网络最终输出为策略 $\pi(a|s)$, 网络层结构如图 3 所示.

DQN 中, 为解决数据分布的相关性, 引入经验回放机制. 经验池中存放一个训练周期的数据, 即智能体在从起点到终点的探索过程中的 (s, a, r, s'). 当经验池没有存储满时, 智能体随机探索环境并将训练样本回传到经验回放池中. 当经验池存储满以后, 从经验池中随机选择一定数量的历史经验, 送

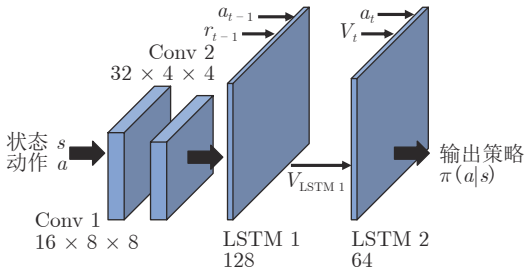


图 3　加入 LSTM 后的网络层结构

Fig. 3　Network layer structure adding LSTM

入 Q 目标网络参与训练. 开始时, 未参与回传训练的数据则被删去.

在 Q 目标网络的更新过程中, 引入时序差分误差 TD_{error} 概念. TD_{error}描述不同时间的状态估计的差异, 可表示为:

$$TD_{error} = \left| r_{s,a} + \Upsilon \max_{a'} Q(s', a') - Q(s, a) \right| \quad (13)$$

式中, $r_{s,a}$ 是当前奖励值, $Q(s', a')$ 和 $Q(s, a)$ 分别是目标网络和估计网络对应的行动价值, Υ 为行动对应的折扣因子. 一个样本即 (s, a, r, s'), 样本的 TD_{error} 越大, 说明预测精度有越大上升空间, 该部分样本作为主要起作用的训练对象被放在经验回放池中.

在每次进行更新时, 选择绝对值大的 TD_{error} 样本进行回放, 然后更新该样本的 Q 值和 TD_{error} 的权重. 本文设计一种阈值筛选机制, 不同于动作产生即送入经验池的方式, 先判断动作产生的 TD_{error} 是否足够大, 满足阈值的, 才放入经验池, 加速算法的收敛.

TD_{error} 阈值的设置由预训练决定. 将预训练样本数据按照 TD_{error} 从大到小依次排序为一序列, 序列中总样本数为 n, 设定参数 α, 代表正式训练时使用序列中样本的比例. 则选取第 $\alpha \times n$ 位置的样本所对应的 TD_{error} 值作为阈值. 为设立合理的阈值, 进行预训练, 按优先级从高到低的顺序对数据列表进行排序. 然后从高斯随机数值生成器中获取一个 0 ~ 1 之间的随机数 α, 其中 α 在 0 ~ 1 之间取值概率呈高斯分布, 避免取在接近 0 或接近 1 的极端情况.

对于正式训练的样本数据, 只有大于阈值的样本才被放入经验回放池中. 在 T-DQN 算法中, 首先采用纯粹贪婪优先方法对样本的 TD_{error} 进行排序, 确保被采样的概率在转移优先级上是单调的. 由于只进行高优先级重放会产生过拟合问题, 在排序好的样本队列中加入均匀随机采样, 抽取经验池中的不同样本进行回放. 采样概率为:

$$P(i) = \frac{r_i^{\alpha}}{\sum_{k=1}^{n} r_k^{\alpha}} r_i \tag{14}$$

式中, r_i 是第 i 个样本的 TD_{error} 在整个序列 k 中的位置排序比例.

2 算法仿真

无人艇的避障规划实现分为 2 部分: 1) 全局路径规划. 根据已知的地理信息对目标水域建模, 得到一个有利于作业、便于简化的离线环境模型, 借助搜索算法设计出一条从起点到终点的无碰路径. 2) 局部路径规划. 在无人艇传感器可探测的范围内, 根据获得的感知信息, 采取合理的实时避障策略, 在避免与障碍物发生碰撞的同时, 到达指定目标点. 判断流程如图 4 所示.

在实际训练中, 首先采用全局规划的方式找出最优通行路径. 航行过程中会遇到未知的小障碍物, 仅依靠全局路径规划无法完全避开, 因此采用局部避障辅助决策, 在感知范围内搜索区域内障碍物, 并求解最优避障通行路径. 实际场景中, 在全局规划制定的路径较为粗略的情况下, 可采用 T-DQN 局部避障算法, 避开水面小型漂浮障碍物和礁石等. 为验证算法的有效性, 本文在不同尺度大小的栅格仿真环境中进行一系列实验, 对比 Q-learning、DQN、LSTM + DQN 和 T-DQN 四种算法的避障任务完成效果.

栅格法将二维地图切分为若干矩形网格单元, 通过对栅格赋值来表征障碍物或自由空间, 利用二维矩阵完成建模. 但是环境的信息量受到栅格粒度

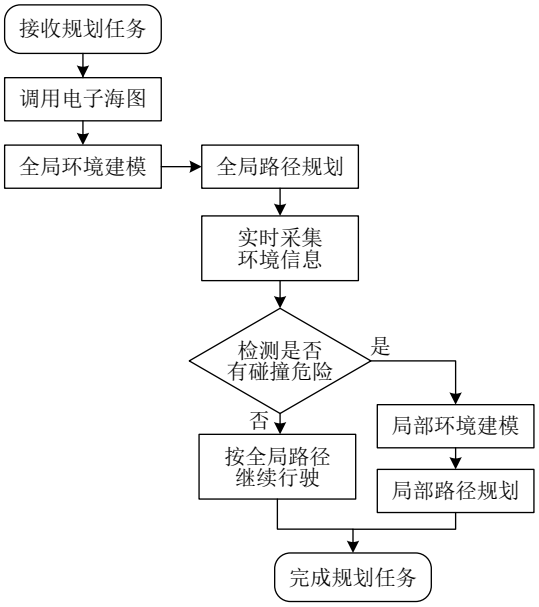
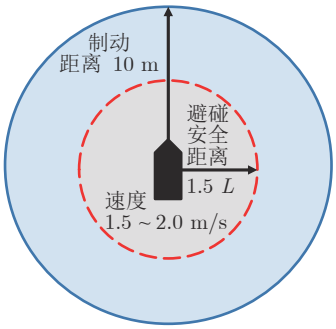


图 4 无人艇路径规划流程图

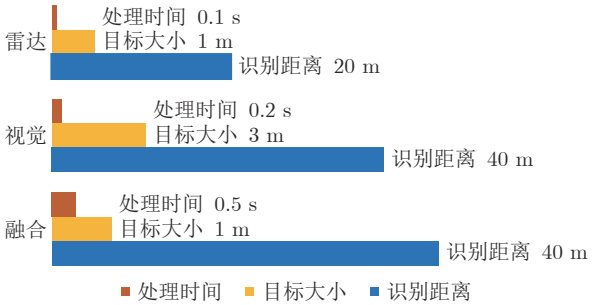
Fig. 4 Flow chart of USV path planning

大小的影响, 栅格过大, 会使得障碍物分辨率降低, 进而影响得到路径的精确度; 栅格过小, 虽然保证路径精度, 却会占据较大存储空间, 而导致运算时间过长. 图 5 展示实际无人艇的动力学和感知参数, 图 5(a) 中避碰安全距离指无人艇在经过障碍物且



(a) 运动学参数

(a) Kinematic parameters



(b) 感知范围和处理时间参数

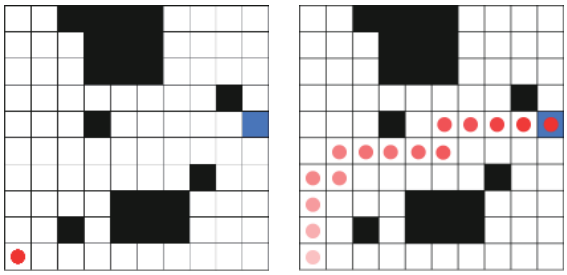
(b) Perception range and time parameter

图 5 无人艇实际参数

Fig. 5 Actual parameters of USV

不与其发生碰撞的情况下, 需要相隔的最短距离, 其中 L 为无人艇长度. 制动距离指由于水浪阻力等影响, 无人艇从失去动力到停止的经停距离. 图 5(b) 展示无人艇的感知范围和处理时间参数. 以融合数据为例: 针对尺度大小为 3 m 的目标物, 融合后的识别距离为 30 ~ 40 m, 即能识别出目标物种类和大小的最远距离为 30 ~ 40 m. 根据本文的实验测试数据, 针对尺寸为 3 m 左右的目标物, 有效感知距离范围在 30 ~ 40 m 左右, 反应距离为 10 m. 实验船长为 1.2 m, 对应栅格中智能体所占据的空间即一个格子单位. 因此在栅格实验中, 本文设置 10×10、20×20、30×30 三种不同大小的地图, 这样设置的仿真结果能涵盖实际中 50 m 范围.

基于栅格环境, 对比在三种不同大小地图下、不同算法在路径规划和避障任务上的表现, 主要对比指标为收敛速度和路径是否最优. 一次训练所需的样本数量设置为 32, 经验池大小设置为 100 000, 折扣因子为 0.99, 学习率为 0.001. 在栅格环境中, 主要对比 10×10、20×20 和 30×30 三种不同大小地图下的 T-DQN 算法避障效果, 分别如图 6、图 7 和图 8 所示. 其中圆圈为智能体的起点位置, 蓝色方格为终点位置, 其余为障碍物, 无法通行. 栅格环境中的障碍物是随机设置的. 对比算法为 Q-learning、DQN 和 LSTM + DQN. 图 9 展示的是环境地图大小分别为 10×10、20×20 和 30×30 时, 四种算法在同一地图下的收敛性能对比. 由图 9 可以看出, 在 3 种不同大小栅格环境下, T-DQN 方法能在给定的训练周期内收敛到最优路径, 在 30×30 的栅格搜索空间内, 收敛速度并未明显降低. 对比 4 种算法的训练结果可以看出, 在引入 LSTM 网络后, 相比 Q-learning 和 DQN 算法, LSTM + DQN 具有更好的决策能力, 收敛性能更好, 但消耗时间较

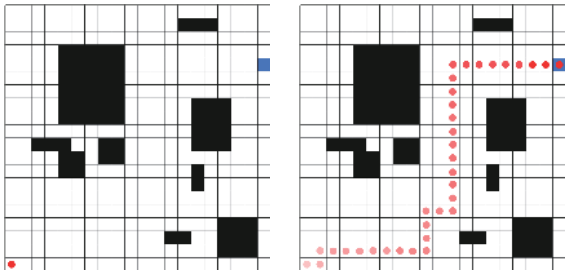


(a) 原始地图
(a) Original map

(b) T-DQN 算法训练结果
(b) T-DQN algorithm training results

图 6　10×10 栅格地图下采用 T-DQN 训练后的路径结果

Fig. 6　Path results after T-DQN training under 10×10 grid map

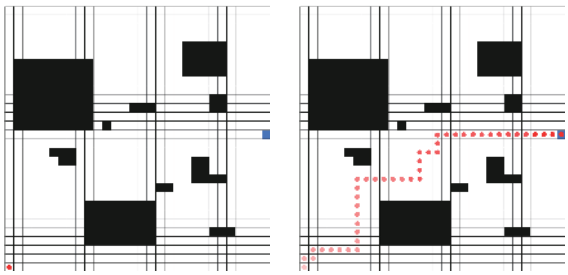


(a) 原始地图
(a) Original map

(b) T-DQN 算法训练结果
(b) T-DQN algorithm training results

图 7　20×20 栅格地图下采用 T-DQN 训练后的路径结果

Fig. 7　Path results after T-DQN training under 20×20 grid map



(a) 原始地图
(a) Original map

(b) T-DQN 算法训练结果
(b) T-DQN algorithm training results

图 8　30×30 栅格地图下采用 T-DQN 训练后的路径结果

Fig. 8　Path results after T-DQN training under 30×30 grid map

长. 而 T-DQN 采用阈值筛选机制能在一定程度上提高算法的收敛速度. T-DQN 算法能在环境地图变大时, 仍以较快速度收敛到最优路径解, 其整体收敛步数相比 Q-learning 算法和 DQN 算法, 分别减少 69.1% 和 24.8%; 相比 LSTM + DQN 算法, 减少 41.1%. 这表明 T-DQN 能够更好、更快地收敛. 训练效果对比如表 1 所示.

3　Unity 3D 平台验证

在 Unity 3D 仿真环境中进行验证, 使用的算法仿真平台 Spaitlab-unity 是一款本文课题组研发的仿真实验平台, 视觉效果逼真, 能较好地还原真实场景.

为使无人艇路径规划仿真平台的仿真结果具备较好真实性, 本文不仅需考虑无人艇的外观和结构特性, 还设计无人艇运动数学模型, 用于仿真无人艇的运动过程. 选用船舶操纵运动数学模型组作为无人艇的运动数学模型[30], 将船舶视为由船体、螺旋桨和舵构成, 并分别考虑它们各自受到的影响.

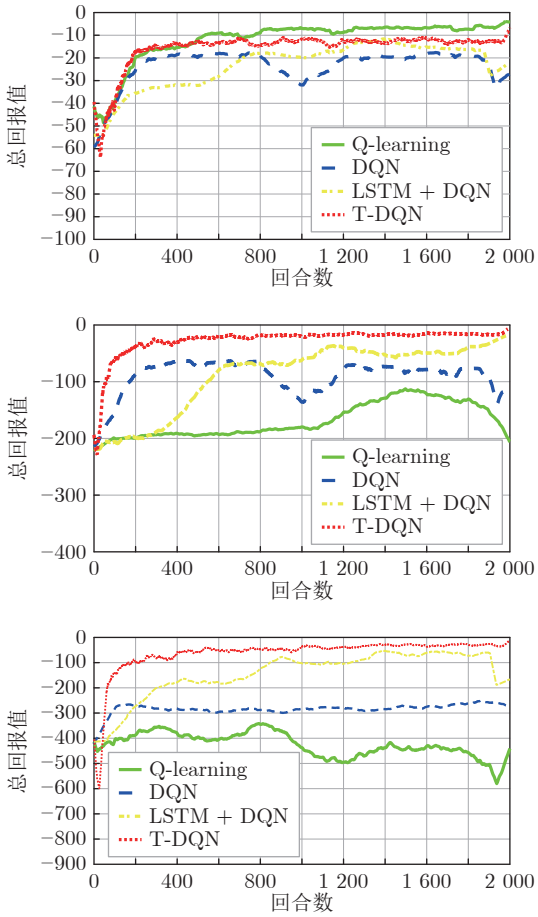


图 9 4 种算法分别在 10 × 10、20 × 20、30 × 30 栅格地图下的平均回报值对比

Fig. 9 Comparison of the average return values of 4 algorithms under 10 × 10, 20 × 20, 30 × 30 grid maps

表 1 4 种算法收敛步数对比

Table 1 Comparison of convergence steps of 4 algorithms

算法	10 × 10 栅格地图	20 × 20 栅格地图	30 × 30 栅格地图
Q-learning	888	> 2000	> 2000
DQN	317	600	> 2000
LSTM + DQN	750	705	850
T-DQN	400	442	517

本文选用的运动数学模型如下:

$$\begin{cases} (m_{USV}+m_x)\,\dot{u}-(m_{USV}+m_y)\,\dot{v}\dot{r}=X_{uh}+\\ \qquad X_{up}+X_{ur}\\ (m_{USV}+m_y)\,\dot{v}+(m_{USV}+m_x)\,\dot{u}\dot{r}=Y_{uh}+Y_{ur}\\ (I_{zz}+J_{zz})\,\dot{r}=M_{uh}+M_{ur} \end{cases} \tag{15}$$

式中, m_{USV}、m_x、m_y 分别为无人艇的质量以及 x 轴、y 轴方向上运动的附加质量. $\dot{u}$、$\dot{v}$ 分别表示无人艇的纵向速度、横向速度. $\dot{r}$ 表示无人艇的转首角加速度. X、Y 代表作用在无人艇上的力沿 x 轴、y 轴的分量; M 表示外力对无人艇绕坐标轴的转动力矩. 下标 uh、up、ur 分别代表无人艇受到的来自水、螺旋桨、舵的作用. I_{zz}、J_{zz} 代表无人艇绕 z 轴转动时的惯性矩、附加惯性矩. 在 Spaitlab-unity 平台使用 C# 编写脚本, 以完成对各个作用于无人艇的力与力矩的计算. Spaitlab-unity 环境的水域仿真如图 10 所示.

图 10 Spaitlab-unity 仿真实验平台

Fig. 10 Spaitlab-unity simulation experiment platform

基于 Spaitlab-unity 搭建高保真的水面场景, 采用地形、天气等插件构建多种训练环境, 尽量涵盖训练的边界样例. 本文在 Unity 3D 中进行虚拟仿真环境设置, 在水面环境中添加模拟静态障碍物的礁石, 无人艇搭载的虚拟激光雷达发射 180 条射线用于探测无人艇前方左、右各 90° 范围内的障碍物, 射线的探测距离范围为 50 m, 环境中包含三个视角, 可以方便直观地展示训练过程和细节. 强化学习算法和 Spaitlab-unity 仿真实验平台通过传输控制协议通信的方式实现数据传输. 在仿真训练环境中, 无人艇的位置、障碍物位置和距离等信息都可以按照一定的数据结构传输到算法端, 服务器接收这些数据进行训练. 训练完成后, 发出运动指令, 再通过通信的方式指导环境内无人艇的运动. 无人艇能根据环境信息自主决策, 按照运动指令实现路径规划和避障.

本文采用全局规划和局部规划相结合的方式, 在 Spaitlab-unity 环境中进行验证. 在水域中分别设置不同的起点和终点, 首先用 A*全局路径规划

算法规划出大致的行进路线, 选取的全局水域范围为500 × 500m, 如图 11 所示. 然后, 采用 30 × 30m 栅格, 将水域离散化成 50 × 50 m 的栅格空间, 如图 12 所示. 当无人艇沿全局路径行驶, 同时以 50 m 左右的感知范围检测周边障碍物, 确定单个栅格的起点和终点. 为做全局路径规划, 场景中相对较小的障碍物会在地图中按比例缩小, 无法在地图中有效地识别出来. 但是相对尺寸较小的无人艇而言, 这些障碍物仍要避开, 所以在全局路径规划完成后, 加入感知数据, 对可探测部分进行局部避障, 以避开地图中较小的障碍物. 因此采用 T-DQN 算法进行局部避障, 做更为精细的行驶决策. 图 13 展示无人艇接受运动指令, 从起点到终点, 避开障碍物的全局规划轨迹和局部规划轨迹的对比, 仿真结果具有较好的可视化效果. 图 11 ~ 13 中, 红色平滑曲线为全局规划轨迹, 绿色折线为局部规划轨迹.

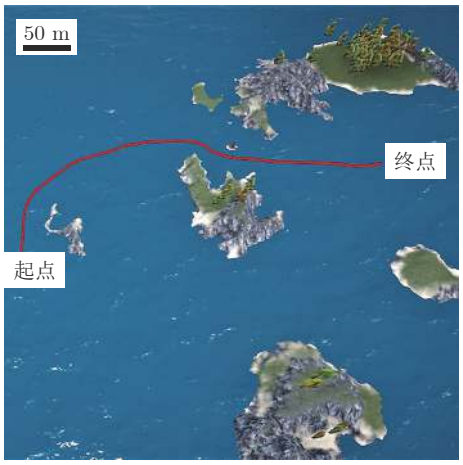


图 11　无人艇全局路径规划仿真运动轨迹

Fig. 11　Global path planning simulation trajectory of USV

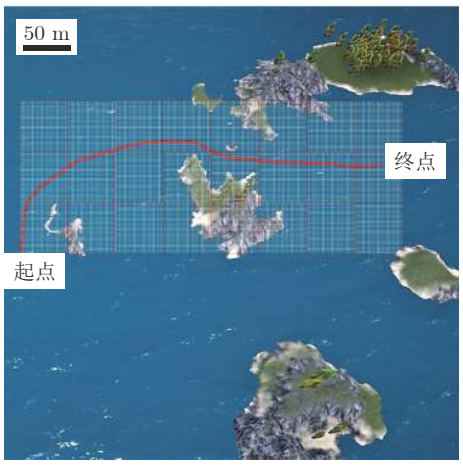


图 12　栅格化水域空间内的全局路径规划

Fig. 12　Global path planning in grid water space

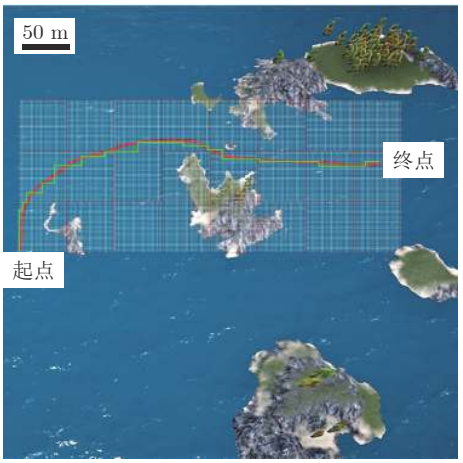


图 13　无人艇全局/局部路径规划仿真运动轨迹对比

Fig. 13　Comparison of global/local simulation trajectories of USV

4　结束语

本文针对水面环境航行决策问题, 提出一种适用于水面无人艇的避障决策新方法. 在深度强化学习算法 DQN 的基础上, 引入 LSTM 网络, 设置阈值筛选经验回放池内的样本, 提出一种基于阈值的 T-DQN 方法. 同时, 有针对性地设置马尔科夫决策过程中状态空间、动作空间和奖励函数, 并将算法分别置于三种不同大小的栅格地图中进行针对性训练. 实验结果表明, 本文提出的 T-DQN 算法能快速地收敛到最优路径, 其整体收敛步数相比 Q-learning 算法和 DQN 算法, 分别减少 69.1% 和 24.8%, 相比 LSTM + DQN 算法, 减少 41.1%. 在本课题组研发的基于 Unity 3D 构建的 Spaitlab-unity 平台对算法的有效性进行验证, 该算法能发出运动指令控制无人艇在水面环境上运动, 对陌生环境具有一定适应能力, 在复杂未知环境中的表现得到了有效验证.

References

1　Tang P P, Zhang R B, Liu D L, Huang L H, Liu G Q, Deng T Q. Local reactive obstacle avoidance approach for high-speed unmanned surface vehicle. *Ocean Engineering*, 2015, **106**: 128−140

2　Campbell S, Naeem W, Irwin G W. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annual Reviews in Control*, 2012, **36**(2): 267−283

3　Liu Z X, Zhang Y M, Yu X, Yuan C. Unmanned surface vehicles: An overview of developments and challenges. *Annual Review in Control*, 2016, **41**: 71−93

4　Zhang Wei-Dong, Liu Xiao-Cheng, Han Peng. Progress and challenges of overwater unmanned systems. *Acta Automatica Sinica*, 2020, **46**(5): 847−857

(张卫东, 刘笑成, 韩鹏. 水上无人系统研究进展及其面临的挑战. 自动化学报, 2020, **46**(5): 847−857)

5 Fan Yun-Sheng, Liu Jian, Wang Guo-Feng, Sun Yu-Tong. Dynamic path planning for unmanned surface vehicle based on heterologous information fusion. *Journal of Dalian Maritime University*, 2018, **44**(1): 9−16
(范云生, 柳健, 王国峰, 孙宇彤. 基于异源信息融合的无人水面艇动态路径规划. 大连海事大学学报, 2018, **44**(1): 9−16)

6 Zhan W Q, Xiao C S, Wen Y Q, Zhou C H, Yuan H W, Xiu S P, et al. Autonomous visual perception for unmanned surface vehicle navigation in an unknown environment. *Sensors*, 2019, **19**(10): Article No. 2216

7 Zhou C H, Gu S D, Wen Y Q, Du Z, Xiao C S, Huang L, et al. The review unmanned surface vehicle path planning: Based on multi-modality constraint. *Ocean Engineering*, 2020, **200**: Article No. 107043

8 Yang X, Cheng W. AGV path planning based on smoothing A* algorithm. *International Journal of Software Engineering and Applications*, 2015, **6**(5): 1−8

9 Lozano-Pérez T, Wesley M A. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 1979, **22**(10): 560−570

10 Yao Peng, Xie Ze-Xiao. Autonomous obstacle avoidance for AUV based on modified guidance vector field. *Acta Automatica Sinica*, 2020, **46**(8): 1670−1680
(姚鹏, 解则晓. 基于修正导航向量场的 AUV 自主避障方法. 自动化学报, 2020, **46**(8): 1670−1680)

11 Dong Yao, Ge Ying-Ying, Guo Hong-Yong, Dong Yong-Feng, Yang Chen. Path planning for mobile robot based on deep reinforcement learning. *Computer Engineering and Applications*, 2019, **55**(13): 15−19, 157
(董瑶, 葛莹莹, 郭鸿湧, 董永峰, 杨琛. 基于深度强化学习的移动机器人路径规划. 计算机工程与应用, 2019, **55**(13): 15−19, 157)

12 Wu Xiao-Guang, Liu Shao-Wei, Yang Lei, Deng Wen-Qiang, Jia Zhe-Heng. A gait control method for biped robot on slope based on deep reinforcement learning. *Acta Automatica Sinica*, 2021, **47**(8): 1976−1987
(吴晓光, 刘绍维, 杨磊, 邓文强, 贾哲恒. 基于深度强化学习的双足机器人斜坡步态控制方法. 自动化学报, 2021, **47**(8): 1976−1987)

13 Szepesvári C. Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2010, **4**(1): 1−103

14 Sigaud O, Buffet O. *Markov Decision Processes in Artificial Intelligence*. Hoboken: John Wiley & Sons, 2013. 39−44

15 Wang Zi-Qiang, Wu Ji-Gang. Mobile robot path planning based on RDC-Q learning algorithm. *Computer Engineering*, 2014, **40**(6): 211−214
(王子强, 武继刚. 基于 RDC-Q 学习算法的移动机器人路径规划. 计算机工程, 2014, **40**(6): 211−214)

16 Silva J A G D, Santos D H D, Negreiros A P F D, Vilas Boas J M, Gonalves L M G. High-level path planning for an autonomous sailboat robot using Q-learning. *Sensors*, 2020, **20**(6): Article No. 1550

17 Kim B, Kaelbling L P, Lozano-Pérez T. Adversarial actor-critic method for task and motion planning problems using planning experience. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence. Honolulu, USA: 2019. 8017−8024

18 Chen Y F, Liu M, Everett M, How J P. Decentralized non-communicating multi-agent collision avoidance with deep reinforcement learning. In: Proceedings of the IEEE international conference on robotics and automation. Singapore: IEEE, 2017. 285−292

19 Tai L, Paolo G, Liu M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Vancouver, Canada: IEEE, 2017. 31−36

20 Zhang J, Springenberg J T, Boedecker J, Burgard W. Deep reinforcement learning with successor features for navigation across similar environments. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Vancouver, Canada: IEEE, 2017. 2371−2378

21 Matthew H, Stone P. Deep recurrent Q-learning for partially observable MDPs. arXiv preprint arXiv: 1507.06527, 2015.

22 Liu F, Chen C, Li Z, Guan Z, Wang H. Research on path planning of robot based on deep reinforcement learning. In: Proceedings of the 39th Chinese Control Conference. Shenyang, China: IEEE, 2020. 3730−3734

23 Wang P, Chan C Y. Formulation of deep reinforcement learning architecture toward autonomous driving for on-ramp merge. In: Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems. Yokohama, Japan: IEEE, 2017. 1−6

24 Deshpande N, Vaufreydaz D, Spalanzani A. Behavioral decision-making for urban autonomous driving in the presence of pedestrians using deep recurrent Q-network. In: Proceedings of the 16th International Conference on Control, Automation, Robotics and Vision. Shenzhen, China: IEEE, 2020. 428−433

25 Peixoto M J P, Azim A. Context-based learning for autonomous vehicles. In: Proceedings of the IEEE 23rd International Symposium on Real-time Distributed Computing. Nashville, USA: IEEE, 2020. 150−151

26 Degris T, Pilarski P M, Sutton R S. Model-free reinforcement learning with continuous action in practice. In: Proceedings of the American Control Conference. Montreal, Canada: IEEE, 2012. 2177−2182

27 Gao N, Qin Z, Jing X, Ni Q, Jin S. Anti-intelligent UAV jamming strategy via deep Q-networks. *IEEE Transactions on Communications*, 2019, **68**(1): 569−581

28 Mnih V, Kavukcuoglu K, Silver D. Playing atari with deep reinforcement learning. arXiv preprint arXiv: 1312.5602, 2013.

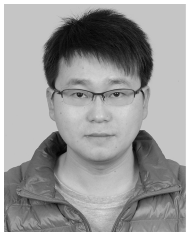
29 Mnih V, Kavukcuoglu K, Silver D. Human-level control through deep reinforcement learning. *Nature*, 2015, **518**(7540): 529−533

30 Zhang C L, Liu X J, Wan D C, Wang J B. Experimental and numerical investigations of advancing speed effects on hydrodynamic derivatives in MMG model, part I: X_(vv), Y_v, N_v. *Ocean Engineering*, 2019, **179**(5): 67−75

周治国 北京理工大学信息与电子学院副教授. 主要研究方向为智能无人系统, 信息感知与导航和机器学习. 本文通信作者.
E-mail: zhiguozhou@bit.edu.cn
(**ZHOU Zhi-Guo** Associate professor at the School of Information and Electronics, Beijing Institute of Technology. His research interest covers intelligent unmanned systems, information perception and navigation, and machine learning. Corresponding author of this paper.)

余思雨 北京理工大学信息与电子学院硕士研究生. 主要研究方向为智能无人系统信息感知与导航.
E-mail: yusiyu3408@163.com
(**YU Si-Yu** Master student at the School of Information and Electronics, Beijing Institute of Technology. Her main research interest is information perception and navigation of intelligent unmanned systems.)

于家宝 北京理工大学信息与电子学院硕士研究生. 主要研究方向为智能无人系统信息感知与导航.
E-mail: 3120200722@bit.edu.cn
(**YU Jia-Bao** Master student at the School of Information and Electronics, Beijing Institute of Technology. Her main research interest is information perception and navigation of intelligent unmanned systems.)



段俊伟 暨南大学信息科学技术学院讲师. 主要研究方向为图像融合, 机器学习和计算智能.
E-mail: jwduan@jnu.edu.cn
(**DUAN Jun-Wei** Lecturer at the College of Information Science and Technology, Jinan University. His research interest covers image fusion, machine learning, and computational intelligence.)

陈 龙 澳门大学科技学院副教授. 主要研究方向为计算智能, 贝叶斯方法和机器学习.
E-mail: longchen@um.edu.mo
(**CHEN Long** Associate professor at the Faculty of Science and Technology, University of Macau. His research interest covers computational intelligence, Bayesian methods, and machine learning.)

陈俊龙 华南理工大学计算机科学与工程学院教授. 主要研究方向为控制论, 智能系统和计算智能.
E-mail: philipchen@scut.edu.cn
(**CHEN Jun-Long** Professor at the School of Computer Science and Engineering, South China University of Technology. His research interest covers cybernetics, intelligent systems, and computational intelligence.)