基于 PRND-SAC 强化学习的耙吸式挖泥船装舱系统控制*

张云飞[1]，苏　贞[1,2]，王　伟[2]

（1. 江苏科技大学 计算机学院，江苏 镇江 212000；2. 江苏科技大学 海洋装备研究院，江苏 镇江 212003）

摘要：针对现有耙吸式挖泥船装舱控制过程人员依赖性强和效率低下的问题，融合优先经验回放(PER)和随机网络蒸馏(RND)技术，提出一种基于优先回放和随机网络蒸馏柔性动作评价(PRND-SAC)的强化学习控制算法，通过设计相应的状态空间、动作空间和奖励函数，将 PRND-SAC 控制器与传统的 SAC 控制器进行对比，并基于全过程装舱阶段仿真环境将 PRND-SAC 控制器与实际疏浚数据进行对比试验。结果表明，设计的控制器能够快速且稳定地收敛；与传统的 SAC 控制器相比，所提出的 PRND-SAC 控制器不仅有效提高了控制过程的稳定性，还显著提升了装舱效率。

关键词：强化学习；SAC；随机网络蒸馏；耙吸式挖泥船；优先经验回放

中图分类号：U616　　**文献标志码：**A　　**文章编号：**1002-4972(2025)06-0186-08

PRND-SAC reinforcement learning-based mud tank loading system control for trailing suction hopper dredger

ZHANG Yunfei[1], SU Zhen[1,2], WANG Wei[2]

(1. School of Computer, Jiangsu University of Science and Technology, Zhenjiang 212000, China;
2. Marine Equipment and Technology Institute, Jiangsu University of Science and Technology, Zhenjiang 212003, China)

Abstract: In view of the problem that current loading control processes of a trailing suction hopper dredger is high personnel dependence and low efficiency, prioritized experience replay (PER) and random network distillation (RND) are combined, and a reinforcement learning control algorithm is proposed on the basis of priority replay and random network distillation soft actor-critic (PRND-SAC). By designing appropriate state space, action space, and reward function, the PRND-SAC controller is compared with the traditional SAC controller. Furthermore, comparative experiments are conducted between the PRND-SAC controller and actual operational data on the basis of a full-process loading phase simulation environment. The results demonstrate that the proposed controller converges quickly and stably. Furthermore, compared to the traditional SAC controller, the PRND-SAC controller not only enhances the stability of the control process but also significantly increases loading efficiency.

Keywords: reinforcement learning; SAC; random network distillation; trailing suction hopper dredger; prioritized experience replay

随着现代疏浚工程对环境、工期和经济要求的不断提高，船舶智能化已成为行业内疏浚船舶发展方向的共识。对于疏浚船舶来说，疏浚作业阶段的复杂性更高，如何实现节能降耗，提高疏浚效率离不开智能船舶信息技术的支持[1]。耙吸式挖泥船作为一种重要的疏浚装备，被广泛应用

收稿日期：2024-08-23

***基金项目：**工信部高技术船舶项目(工信部装函【2019】360)；江苏科技大学校企合作项目(2175152102)

作者简介：张云飞（1998—），男，硕士研究生，研究方向为船舶智能化、人工智能。

于港口建设、航道维护和河流治理等疏浚工程[1]。随着全球对环境保护和可持续发展的日益重视，绿色疏浚成为挖泥船行业关注的焦点。

在计算机技术迅猛发展的今天，人工智能已逐步成为推动科技进步的关键力量之一。在众多人工智能技术中，强化学习凭借其独特的学习方式脱颖而出，成为构建自主学习和决策系统的核心方法。同时，随着深度强化学习技术的快速发展，深度强化学习也开始与更多学科进行交叉融合，在挖泥船的智能疏浚控制方面，张红升等[2]提出一种基于深度确定性策略梯度(deep deterministic policy gradient,DDPG)算法的耙吸式挖泥船耙头活动罩控制器，并在“新海虎 8”轮上进行算法验证，结果表明该控制器可以有效提高产量；章亮[3]则提出一种基于强化学习的耙吸式挖泥船艏吹工艺控制策略，试验表明，该策略不仅能有效排出泥舱内的泥沙，还能维持疏浚作业时泥沙浓度和流量的稳定，从而显著增强了疏浚作业的连续性和效率。

尽管现有研究取得了显著进展，但在复杂工况下，挖泥船装舱控制的鲁棒性和效率仍有待进一步提升。为此，本文提出一种融合优先经验回放(prioritized experience replay,PER)[4]和随机网络蒸馏(random network distillation,RND)[5]技术的柔性动作-评价(soft actor-critic,SAC)算法，引入一种混合动态权重机制，用以平衡外部奖励与 RND 网络产生的内部奖励，从而更精细地调优学习过程。通过基于挖泥船装舱机理模型构建的仿真环境中进行训练验证。

1　耙吸船装舱机理模型

耙吸挖泥船施工作业中，挖掘与装舱是两个独立又相互依赖的环节。其中，装舱过程的精细化控制对于提升施工效率和持续作业能力至关重要。鉴于挖掘过程的复杂性难以通过传统机理模型完全捕捉，本文聚焦于装舱过程的机理模型优化，旨在实现更加高效、精准的控制策略，从而间接提升整体作业效能。

耙吸挖泥船装舱过程[6]可分为 3 个阶段。

1）无溢流阶段：此阶段，舱内混合物(泥、沙混合形成的泥浆)高度低于溢流筒高度，舱内无溢流现象发生。

2）恒体积阶段：当舱内混合物高度达到溢流筒高度时，进入此阶段。此时，尽管舱内混合物体积保持不变，但由于低密度混合物的溢出，装载质量逐步增加。

3）恒载质量阶段：当船舶逐步达到最大吃水深度时，为进一步保证航行安全和维持施工效率，自动调节溢流筒高度，以使舱内混合物总质量保持不变。耙吸式挖泥船装舱过程见图 1。

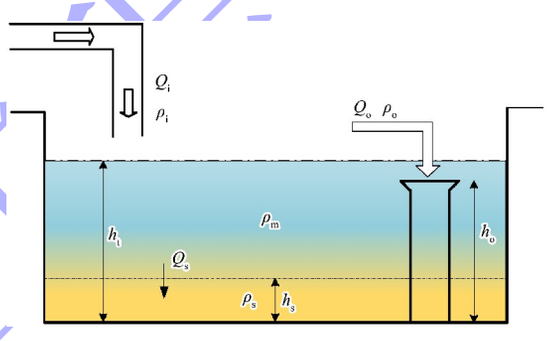


图 1　装舱过程

Fig. 1　Process of mud tank loading

因此，以泥舱中沉积泥沙质量最大化为优化目标，构建数学模型。模型具有 3 个状态变量：泥舱中的总质量 m_t，混合物在泥舱内的总体积 V_t 和沙床的质量 m_s。其沉积动力学可以由以下方程组构成：

$$\begin{cases} \dot{V}_t = Q_i - Q_o \\ \dot{m}_t = Q_i\rho_i - Q_o\rho_o \\ \dot{m}_s = Q_s\rho_s \end{cases} \tag{1}$$

式中：$\dot{V}_t$ 为体积变化率；$\dot{m}_t$ 为装载质量变化率；$\dot{m}_s$ 为沙床质量变化率；Q_i 为进舱流量；Q_o 为溢流流量；ρ_i 为进舱密度；ρ_o 为溢流密度；Q_s 为泥舱中混合物层中进入沙床层的沙的流量；ρ_s 为沙床的密度，即挖泥船施工地点自然土的密度。

沉积到沙床的流量 Q_s 主要受到重力的影响，可表示为：

$$Q_s = A(1-u)v_{s_0} \cdot \frac{\rho_m - \rho_w}{\rho_s - \rho_m}\left(\frac{\rho_q - \rho_m}{\rho_q - \rho_w}\right)^{\beta} \tag{2}$$

式中：A 为质量流入或流出控制体积的面积；u 为冲刷系数；v_{s_0} 为水平流速；ρ_m 为混合的密度；ρ_w 为水的密度；ρ_q 为沙的密度；β 为雷诺数系数。

当混合物通过管道进入泥舱后，随着泥舱装载高度不断升高，直到达到溢流高度，之后混合物以流量 Q_o 流出泥舱，该流量可表示为：

$$Q_o = k_o[\max(h_t - h_o, 0)]^{\frac{3}{2}} \tag{3}$$

式中：k_o 为取决于溢流堰的形状和周长的参数；h_t 为当前时刻装舱液面高度；h_o 为溢流筒高度。

本文采用水层模型[7]进行溢流密度 ρ_o 估计，该模型采用两段式模型，分为薄水层和汤状混合物层。对较低的溢流量 Q_o，只有纯水在溢流时从水层中流出。但是，当 $Q_o > Q_w$ 时，汤状混合物层的溢流量并非为 0，即溢流的密度由两层混合，公式为：

$$\rho_o = \frac{\rho_{ms}Q_{ms} + \rho_w Q_w}{Q_{ms} + Q_w} \tag{4}$$

式中：Q_{ms} 为汤状混合物的流量；ρ_{ms} 为汤状混合物的密度；ρ_w 为纯水的溢流密度；Q_w 为纯水的溢流流量。

此外，海底地形、船舶姿态和土质粒径等外部参数变化也会对挖泥船控制产生影响，但这些影响最终也会在进舱流量和进舱密度数值变化上有所体现。为进一步提升后续强化学习算法的动作探索效率，对挖泥船疏浚过程中部分影响装舱系统控制的外部参数进行简化，在最大限度模拟挖泥船实际施工环境的同时，提供更加连续的施工动作控制策略，基于上述原理最终可完成耙吸式挖泥船装舱机理模型构建。

2 PRND-SAC 算法设计

2.1 最大熵强化学习 SAC

SAC 算法[8-9]是一种基于最大熵的离线策略强化学习算法，它在重放缓冲区 D 中存储一组{(s_t, a_t, $r_{t,\text{external}}$, s_{t+1})}转换元组(s_t 为当前状态，a_t 为当前动作，$r_{t,\text{external}}$ 为外部奖励，s_{t+1} 为下一时刻状态)，并使用参数为 ϕ 的神经网络作为策略 π_ϕ，参数为 ω 的另一个神经网络作为 Q 值(Q_ω)。为了训练 φ 和 ω，它从重放缓冲区中随机抽取一批转换元组，并最小化价值网络损失 $L_Q(\omega)$ 和策略网络损失 $L_\pi(\phi)$ 为目标分别进行权重 ϕ 和 ω 更新：

$$L_Q(\omega) := E_{s_t, a_t \sim D}\left\{\frac{1}{2}[Q_\omega(s_t, a_t) - Q^{\text{tar}}(s_t, a_t)]^2\right\} \tag{5}$$

$$L_\pi(\phi) := E_{s_t \sim D, a_t \sim \pi_\phi}[\alpha \ln \pi_\phi(a_t | s_t) - Q_\omega(s_t, a_t)] \tag{6}$$

式中：$E_{s_t, a_t \sim D}$ 为经验回放缓冲区 D 中的状态—动作对(s_t, a_t)所求期望；$E_{s_t \sim D, a_t \sim \pi_\phi}$ 为从缓冲区 D 采样状态 s_t 经策略 π_ϕ 生成动作 a_t 的期望；Q^{tar} 为目标 Q 函数，其参数周期性的从学习到的 Q_ω 中复制而来；α 为温度系数。

2.2 PRND-SAC

RND 网络是一种用于无模型强化学习的探索机制，根据文献[5]可知，RND 网络由两部分组成：固定参数随机化目标网络 $f_{\text{target}}(s)$ 和可训练的预测网络 $f_{\text{current}}(s)$。目标网络产生随机的目标向量，而预测网络则负责学习环境状态的特征表示。RND 网络的训练目标是最小化预测目标与实际目标之间的差异，从而鼓励智能体探索未知的环境状态和动作。RND 网络的探索奖励可表示为：

$$r_{\text{internal}}(s) = \|f_{\text{target}}(s) - f_{\text{current}}(s)\|^2 \tag{7}$$

式中：$\|\cdot\|$为 L2 范数。

虽然 RND 在奖励稀疏的环境中表现出色，但在探索和利用之间缺乏有效的平衡，可能导致过度探索而忽略对已知信息的优化。

优先经验回放(prioritized experience replay，PER)作为对传统经验回放机制的一种改进，旨在通过优先考虑重要样本的回放，提升学习效率和性能。通过预测误差来衡量每个经验片段的重要性，对回放缓冲区中的样本进行加权采样。然而，PER 主要依赖于外部奖励信号来确定经验的优先级，这在探索奖励稀疏的环境中并不足够有效。

为了进一步融合和优化上述两种策略，本文提出优先回放和随机网络蒸馏柔性动作评价(prioritized replay and random network distillation soft actor-critic，

PRND-SAC)模型。该模型不仅利用了 PER 的动态优先级回放，还结合 RND 的内在奖励机制，以在复杂的连续控制任务中实现自适应优化决策。

当经验池中的数据超过一定数量时，从经验池中采集一个批次(batch)的数据，并将采集到的下一时刻状态 s_{t+1} 作为输入送入 RND 网络中进行预测，通过比较 RND 网络的预测和真实目标向量，计算出每个状态观察 s_{t+1} 对应的内部误差 L_{RND}，L_{RND} 通过线性归一化方法计算出当前批次的内部奖励 $r_{\mathrm{internal}}(s_{t+1})$，其中，训练过程中内部奖励函数为：

$$r_{\mathrm{internal}}(s_t)=\frac{L_{\mathrm{RND}}-\min(L_{\mathrm{RND}})}{\max(L_{\mathrm{RND}})-\min(L_{\mathrm{RND}})} \tag{8}$$

通过将外部奖励与内部奖励进行权重相加得到当前时刻总奖励，其中，当前时刻总奖励表达式为：

$$r_t=\lambda\cdot r(s_t,a_t)+(1-\lambda)\cdot r_{\mathrm{internal}}(s_t) \tag{9}$$

式中：λ 为一个可动态改变的权重系数，用于平衡内部奖励与外部奖励的相对重要性；$r(s_t,a_t)$ 为智能体与环境交互的外部奖励。

本策略的核心在于通过监测智能体性能的变化率动态调整内在奖励的比例。在学习初期，当智能体的性能提升迅速，内在奖励的权重相应增加，鼓励智能体进行广泛的探索；随着学习的深入，当性能提升速率下降，内在奖励的权重逐步降低时，促使智能体更多地利用已学得的知识。以下两个核心公式用于更新内在奖励的权重系数 λ：

$$\Delta P=\frac{P_t-P_{t-1}}{P_{t-1}+\varepsilon} \tag{10}$$

$$\lambda_{t+1}=\mathrm{chip}[(\lambda_t+\eta\cdot\Delta P),0,1] \tag{11}$$

式中：P_t 和 P_{t-1} 分别为当前和前一时间点的性能值，即当前回合的总奖励值；ε 为一个小正数常量，防止除零错误；ΔP 为性能变化率，它描述了性能相对于前一时间点的增长或下降比例，是调整内在奖励权重的重要依据，$\Delta P>0$ 时智能体的平均性能正在提升，特别是在学习初期，这通常意味着智能体正在快速掌握环境的规律，此时增加内在奖励的比重有助于促进更多的探索，$\Delta P<0$ 或接近于 0 时，智能体的性能提升变缓或甚至开始下降，内在奖励的比例应减少，以鼓励智能体利用自身已有的知识；λ_{t+1} 为下一时刻点的内在奖励权重；η 为学习率；chip 为裁剪函数，用于确保 λ_{t+1} 的值在[0,1]的合理范围内。通过基于性能变化率的动态权重调整策略，提供了一种简化且有效的手段，实现强化学习中环境的动态平衡探索与充分利用。

此外，传统 PER 方法在强调重要样本的同时，容易造成低优先级样本几乎不被采样，导致学习过程偏向于已知知识而忽视了探索的全面性。同时，计算高优先级样本的精确概率和执行采样操作可能涉及对整个经验池的遍历，这在大规模数据集上效率低下。本文结合加权随机树(Sum Tree)的数据结构不仅实现了对优先级的高效存储和更新，还能够有效减轻采样偏差，结合 RND 网络的最小化损失实现优先级 p_i 更新的计算公式为：

$$p_i=(|D_{\mathrm{T},i}+\lambda L_{\mathrm{RND}}+\dot{o}|)^{\varphi} \tag{12}$$

式中：$D_{\mathrm{T},i}$ 为第 i 个经验的总误差；$\dot{o}$ 为一个小的常数，用于确保优先级不为零；φ 为一个超参数，用于调节优先级的分布程度。该机制使得即使对于相对较少更新的经验，模型也能够进行有效学习，从而增加了数据的多样性，提高模型对各种情况的适应能力。

然后，对 Q_1 网络和 Q_2 网络进行训练，训练过程中网络的损失函数可表示为：

$$Q_{\mathrm{tar}}=r_t+\gamma\{\min_{j=1,2}[Q_{\bar{\theta},j}(s_{t+1},a_{t+1})]-\alpha\ln\pi_{\phi}(a_{t+1}|s_{t+1})\} \tag{13}$$

$$L_Q(\theta)=E_{(s_t,a_t,s_{t+1})\sim D}\left\{\frac{1}{2}[Q_{\theta,i}(s_t,a_t)-Q_{\mathrm{tar}}]^2\right\} \tag{14}$$

式中：Q_{tar} 为目标价值；$Q_{\theta,i}$ 为第 i 个价值函数，其中 θ 为网络参数；$Q_{\bar{\theta},j}$ 为第 j 个目标价值函数；π_{ϕ} 为策略函数；ϕ 为策略函数的网络参数；α、γ 为折扣因子。

Actor 网络的目的是在当前状态下选取价值最高的动作，即 Actor 的损失函数需要价值最大化，Actor 的损失函数为：

$$L_{\pi}(\phi)=E_{s_t,a_t\sim D}[\alpha\ln\pi_{\phi}(a_t|s_t)-\min_{i=1,2}Q_i(s_t,a_t)] \tag{15}$$

3 试验

3.1 状态空间设计

耙吸船装舱过程中，存在由 3 个核心参数构成的状态空间，即装载质量、装载体积和液面高度。这些参数直接影响船体的稳定姿态调整、装载效率波动以及装载总量衡量。因此，总体状态空间由这 3 个互相关联的状态变量共同构成：

$$s=\{V_{\mathrm{i}},h_{\mathrm{i}},m_{\mathrm{i}}\} \tag{16}$$

式中：V_i 为耙吸船的装舱体积；h_i 为耙吸船泥舱的液面高度；m_i 为耙吸船的装舱质量。

3.2 动作空间设计

进舱流量决定了混合物泥浆进入泥舱的速度，影响泥浆沉降速度和装舱效率；进舱密度(即输入泥舱的泥浆密度)影响舱内沉积物分布和装载质量；溢流筒高度直接影响沉积物堆积高度及溢流损失。通过精细调控这 3 个动作参数，可实现泥浆沉积过程优化，进而提升装载质量和整体的装舱效率。总体动作空间可表示为：

$$a=\{h_{\mathrm{o}},\rho_{\mathrm{i}},Q_{\mathrm{i}}\} \tag{17}$$

式中：h_{o} 为溢流筒高度；ρ_{i} 为进舱密度；Q_i 为进舱流量。

3.3 奖励函数设计

本文旨在制定一种能够使挖泥船在保持极限载荷和溢流损失允许范围的同时，实现最大化产量目标的泥浆沉积控制策略[10]。为提高智能体对奖励值的敏感程度，将奖励细分成产量奖励 R_m 和动作奖励 R_{a}。

1）产量奖励公式为：

$$R_{\mathrm{m}}=\begin{cases}\sum_{t}^{t+T}\Delta m_{\mathrm{s}}(t)\cdot\alpha_{\mathrm{inc}} & m_t\leqslant m_{\max}\ \text{且}\ \rho_{\mathrm{o}}\leqslant 0.95\rho_{\mathrm{i}}\\ 0 & (\text{其他})\end{cases} \tag{18}$$

式中：$\Delta m_{\mathrm{s}}(t)$ 为当前时刻的干土质量增量；t 为当前时刻；T 为累计土产量间隔时间；α_{inc} 为土增量的奖惩因子；m_t 为当前时刻装舱质量；$m_{\max}$ 为最大装舱质量；ρ_{o} 为溢流物密度；ρ_{i} 为进舱密度。

2）动作奖励公式为：

$$R_{\mathrm{a}}=-\frac{1}{m}\sqrt{\sum_{i=1}^{m}(a_t-a_{t-1})^2}\cdot\tau \tag{19}$$

式中：a_t 为当前时刻的动作；a_{t-1} 为上一时刻的动作；m 为动作个数，通过计算动作之间的欧式距离作为当前的奖励值，防止智能体出现过大的动作振荡；τ 为动作奖励平衡因子。

最终，定义每个时间步骤能获得的外部总奖励 r 表达式为：

$$r=R_m+R_{\mathrm{a}} \tag{20}$$

3.4 试验流程

PRND-SAC 算法网络结构，见图 2。首先，基于机理构建耙吸船装舱强化学习环境，获得环境的初始状态；其次，通过策略网络动作，将当前的状态 s_t 作为输入，获得执行的动作 a_t，将 a_t 放入耙吸船装舱环境进行交互得到对应的 r_t 和下一时间步的状态 s_{t+1}；然后，将得到的数据组合成四元组$\{(s_t,a_t,r_t,s_{t+1})\}$存入到经验池，当经验池中的经验累计到一定数量时，执行智能体的更新操作，依次更新两个价值网络评价(critic)，策略网络动作，RND 网络和温度系数 α，而目标价值网络的值 $Q_{\bar{\theta}_1}$、$Q_{\bar{\theta}_2}$ 延迟更新；最后，根据策略网络和价值网络的总误差 D_{T} 更新经验池中的数据优先级。

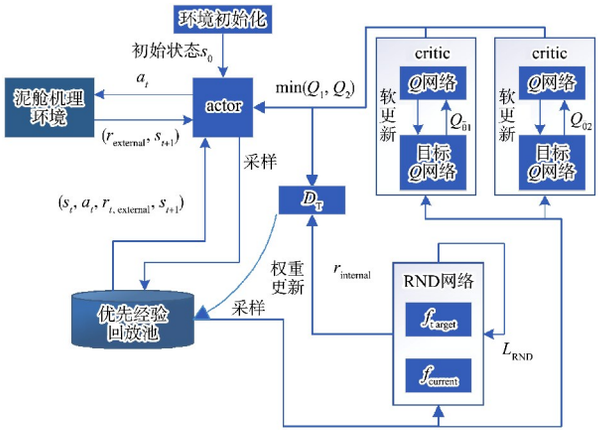


图 2 PRND-SAC 算法流程

Fig. 2 Flow of PRND-SAC algorithm

3.5 模型训练

为了验证本文算法的有效性，基于装舱机理构建了两种挖泥船装舱环境，分别为全过程装舱环境和溢流装舱阶段环境，通过观察装舱干土总量变化判断模型的训练情况。

强化学习中合理设置模型超参数对提升模型

性能至关重要。然而，由于强化学习算法通常存在多个影响收敛性和性能的超参数，手动搜索最优超参数组合往往是一项艰巨的任务。为解决这个难题，本文利用遗传算法对强化学习算法的超参数进行自动优化。遗传算法能够高效地在复杂的非线性优化空间中进行全局搜索，从而发现全局最优或近似最优的参数组合。本文构造遗传算法的适应度函数为：

$$f_{\text{itness}} = \frac{1}{M}\sum_{i=1}^{M} r_{\text{total},i} \tag{21}$$

式中：M 为控制器训练轮数；$r_{\text{total},i}$ 为第 i 轮的总奖励。

通过遗传算法获得 PRND-SAC 控制器中网络的超参数设置见表 1。

表 1　超参数设置

Tab. 1　Hyperparameters setting

折扣因子	网络最小训练轮数	critic 网络学习率	actor 网络学习率	α_{learning} 学习率	软更新系数	每轮训练数量	经验池容量	软更新轮数
0.972	389	0.001 15	0.000 206	0.000 618	0.046 7	182	9 862	2

注：软更新系数、每轮训练数量、经验池容量分别对应程序中的 tau、batch、R。

本文采用 PyTorch 深度学习框架，设置智能体的训练轮数为 1 000 次，采样频率为 10 s/次。同时，试验中还以训练轮数为横轴，以每轮训练中获得的累计奖励值为纵轴，绘制 PRND-SAC 控制器在耙吸式挖泥船装舱机理环境中训练得到的每轮总奖励变化曲线，见图 3。可以看出，初始阶段智能体在环境中表现出的学习效果并不明显，奖励值基本保持在 0 附近，这表明强化学习模型正处于随机探索阶段，此时智能体试图理解环境的基本规律。进入中间阶段后，随着模型不断从经验池中抽取样本进行学习，训练奖励逐渐提升。由于 PRND-SAC 模型结合了 RND 的奖励探索和 SAC 的熵最大化原则。这使得智能体优先探索预测误差大、新奇性高的状态，避免过早收敛于局部最优策略，因此奖励值出现较明显的波动。在训练的后期，随着 SAC 算法熵值的减小，RND 模型中预测网络与目标网络间的误差逐渐减小，PRND-SAC 网络的训练变得更为稳定，总的奖励值也趋向于一个稳定的水平，这表明智能体已学会了有效且稳健的策略。

总奖励值最优的网络模型每一时刻更新后获得的内在奖励值见图 4。可以看出，训练初期，在 300~400 回合内在奖励值波动较大，智能体正在积极探索环境中那些预测难度高、新颖性强的状态。这种高奖励驱使智能体跳出常规模式，尝试多样化的动作序列，避免陷入局部最优。

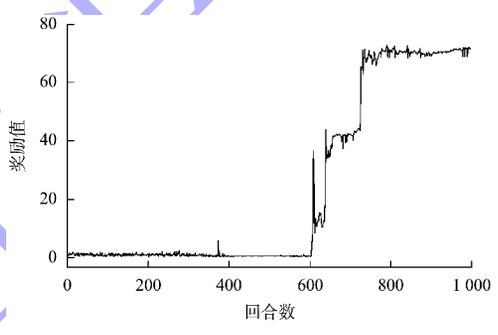


图 3　每回合总奖励值

Fig. 3　Total reward value per episode

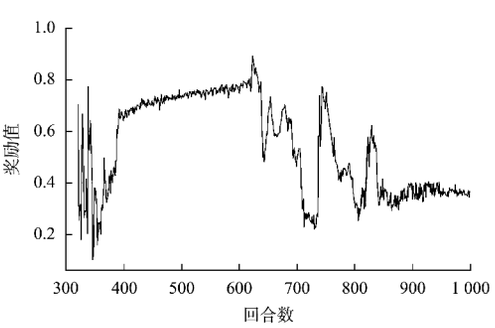


图 4　RND 网络内在奖励值

Fig. 4　Total internal reward of RND network

随着训练的深入，在 400~630 回合，RND 网络的内在奖励呈现一种稳步增长趋势，这表明本文设计的 RND 网络成功识别并量化了机理环境中的新颖性。通过将这种奖励机制与 SAC 优化策略相结合，可使智能体在保证充分探索的同时，逐步学习到更优的行为策略。

当训练进入到成熟阶段，即 640~1 000 回合，内在奖励波动逐渐减小，并逐步趋于一个稳定值，

这表明控制器已经学会了高效的控制策略，并且其性能表现稳定。

3.6　算法对比分析

PRND-SAC 和传统 SAC 算法在本文任务上的性能比较，见图 5。可以看出，两种算法的表现都随训练次数增加而改善，但在整体趋势上有明显区别。在装舱机理环境中，PRND-SAC 控制器的回合总奖励值在 600 回合左右迅速上升，明显快于 SAC 控制器的收敛速度。PRND-SAC 控制器在 750 个训练回合后逐步达到收敛并表现出较强的稳定性，而传统 SAC 控制器即使经过 1 000 个回合的训练仍未达到稳定。这表明在耙吸式挖泥船舱室环境控制任务中，PRND-SAC 控制器相较于 SAC 具有更快的参数调整能力和更优的控制稳定性。此外，尽管两种控制器在控制过程中均存在一定的轨迹振荡现象，但 PRND-SAC 控制器的振荡幅度明显小于 SAC 控制器，进一步表明其在控制精度上的优越性。

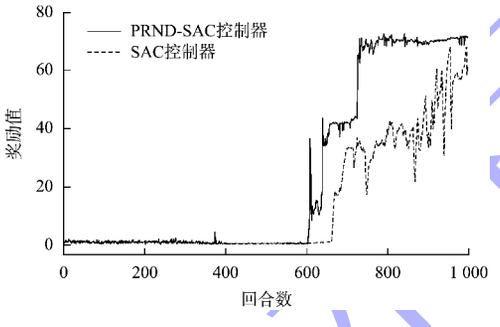


图 5　不同算法每回合总奖励值

Fig. 5　Total reward value per turn for different algorithms

3.7　产量对比分析

当强化学习模型训练结束后，选取总奖励值最优的网络模型参数作为 PRND-SAC 控制器的最终参数，为了验证 PRND-SAC 控制器的有效性，基于全过程装舱阶段仿真环境，本文通过 PRND-SAC 控制器与实际疏浚数据进行对比试验，实际数据选取“新海虎 8”耙吸式挖泥船 8 月 22 日在长江流域疏浚施工数据，该区域的自然土密度为 1 960 kg/m^3。控制器获得的土方量与该工况下的实际土方量对比情况，见图 6。

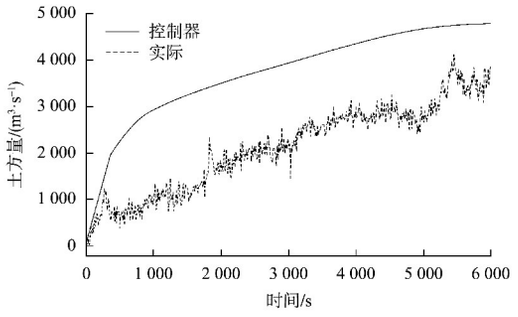
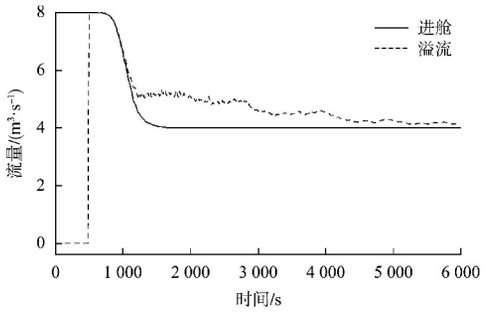


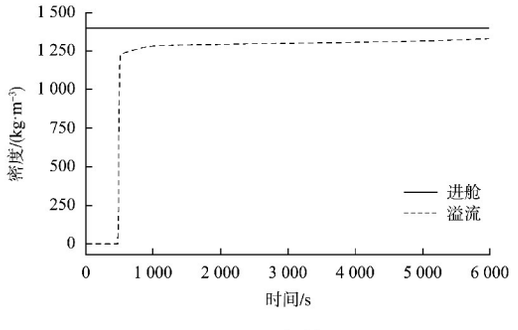
图 6　土方量对比

Fig. 6　Earthwork yield comparison

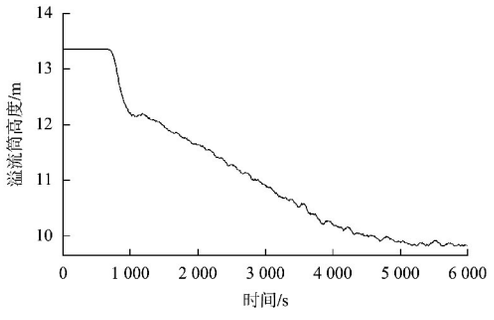
由图 6 可知，PRND-SAC 控制器在前期无溢流阶段可以有效提高挖泥船装舱效率，与人工操作相比较存在较大优势，随着挖掘的持续进行，PRND-SAC 控制器依旧保持土方量的上升趋势，无明显的波动，并在船舱载质量达到上限时，停止装舱，此时 PRND-SAC 控制效果远大于人工操作。此外，PRND-SAC 控制器为获得连续、可控的最优进舱流量、进舱密度和溢流筒高度数值，在机理环境中不断进行探索，所获得的 3 个参数最优控制曲线，见图 7。



a) 流量



b) 密度



c) 溢流筒高度

图 7　3 个参数随时间变化曲线

Fig. 7　Variation curves of three parameters over time

由图 7a)可知，在控制初期 0～900 s，为保证装舱产量最大化，进舱流量一直维持在 8 m^3/s 附近；在 900～1 500 s 随着溢流筒高度的降低和装舱质量的增加，船舱中的混合密度也随之增加，为防止进舱流量过大导致舱内的泥沙混合物出现大量溢流，需要适当减小进舱流量至合适的区间，保证泥沙混合物在船舱中稳定沉积；在控制后期 1 500～6 000 s，进舱流量在智能体控制下，逐渐稳定在 4 m^3/s 左右，表明控制策略能够根据泥舱状态的动态变化，适时调整进舱流量，智能体在装载速率与溢流损失之间实现了良好的平衡。

由图 7b)可知，进舱密度在控制周期内保持恒定，数值为 1 400 kg/m^3。这种稳定性归因于耙吸挖泥船泥舱沉积过程中溢流损失极限的判定准则，即溢流密度不得超过进舱密度的 95%。同时，随着装舱时间的增加，舱内泥浆溢流密度也逐步升高，通过维持较高的泥浆进舱密度可在兼顾装舱效率和减少溢流损失的同时，实现施工效益最大化。

由图 7c)可知，在整个控制过程中，智能体为更好地在状态空间中探索，会产生一系列随机动作，进而导致溢流筒高度短期内出现波峰、波谷现象，但长期来讲，溢流筒高度总体仍呈现下降趋势，尽管短期内这些动作并不是最优，但作为装舱控制中的一部分其仍然有效。在控制初期的 0～800 s，溢流筒高度保持在 13.35 m，随着时间的推移，在 800～6 000 s，高度逐渐稳定降低至 9.46 m。这表明本文模型能够依据泥舱的沉积状态和装载需求，灵活调整溢流筒的高度，从而在最大程度上减少了溢流损失。同时，溢流筒高度的下降轨迹呈现良好的平滑性，未出现剧烈波动，这反映出本文模型控制机制的稳定性和可靠性。

4　结论

1）由于挖泥船施工环境复杂多变，本文基于水层模型简化并构建了挖泥船装舱系统的机理环境，通过模拟舱内泥砂沉积活动，实现装载土方量预测，为后续挖泥船装舱控制优化提供环境支持。

2）结合优先经验回放和随机网络蒸馏技术的强化学习控制算法（PRND-SAC）在训练效率、控制精度和适应性等方面均优于传统的 SAC 算法，有效提高了强化学习算法的稳定性。

3）PRND-SAC 算法中，智能体能够根据舱内泥砂状态连续动态调整进舱流量、进舱密度和溢流筒高度数值，确保土方量的稳定增长。研究成果可为实现挖泥船装舱系统优化控制提供理论参考。

参考文献：

[1] 梁熙明, 刘林峰. “一键疏浚”领跑挖泥船智能化[N]. 中国交通报, 2024-05-30(6).
LIANG X M, LIU L F. “One-click Dredging” leads the intelligentization of dredgers [N]. China communications news, 2024-05-30(6).

[2] 张红升, 衣凡, 邓伍三. 基于 DDPG 算法的耙吸挖泥船耙头活动罩控制策略研究[J]. 中国港湾建设, 2022, 42 (9): 7-10, 80.
ZHANG H S, YI F, DENG W S. Research on control strategy of draghead visor of TSHD based on DDPG algorithm[J]. China harbour engineering, 2022, 42(9): 7-10, 80.

[3] 章亮. 基于强化学习的耙吸挖泥船艏吹控制策略研究[D]. 镇江: 江苏科技大学, 2022.
ZHANG L. Research on bow blowing control strategy of rake suction dredger based on reinforcement learning[D]. Zhenjiang: Jiangsu University of Science and Technology, 2022.

（下转第 210 页）