【引用格式】万骏，张钰竹. 半潜式航行体回收辅助操控技术应用[J]. 数字海洋与水下攻防，2024，7（2）：186-194.

半潜式航行体回收辅助操控技术应用

万　骏[1, 2]，张钰竹[1, 2]

（1. 中国船舶集团有限公司第七一〇研究所，湖北 宜昌 443003；2. 清江创新中心，湖北 武汉 430076）

摘　要　半潜式航行体回收过程耗时较长，对操纵者经验、技术门槛高，试错成本大。针对此问题提出 VH-PPO 算法，从收敛性、期望收敛时间的上下界、时间复杂度 3 个方面分析其性能。通过人工操控成功的历史数据给予初始概率分布并在此基础上进行训练，省去了自由探索不断试错的过程，可有效减小期望收敛时间，使训练模型更快地收敛，从而降低算法的时间复杂度。针对训练时不同的阶段，选择更优的超参数，防止超调和欠调现象，可帮助训练模型更好地收敛，降低期望收敛时间的上下界，从而降低算法的时间复杂度。在 OpenAI Gym 上通过该算法进行强化学习，训练完成后应用于操控软件，在某海域进行验证并进一步调整模型。实验结果表明：随着试验次数增加，智能体在真实环境中的适应能力越来越好，辅助操控指令在总操控指令中占比超过 50%，有效地减缓了操纵者的疲劳，降低了新手训练难度及替换操纵者的门槛。

关键词　半潜式航行体；算法复杂度；可视化界面；OpenAI Gym；强化学习；辅助操控

中图分类号　TJ67　　**文献标识码**　A　　**文章编号**　2096-5753(2024)02-0186-09

DOI　10.19838/j.issn.2096-5753.2024.02.007

Application of Auxiliary Control Technology for Semi-submersible Vehicle Recovery

WAN Jun[1, 2]，ZHANG Yuzhu[1, 2]

（1. No.710 R&D Institute，CSSC，Yichang 443003，China；
2. Qingjiang Innovation Center，Wuhan 430076，China）

Abstract　The recovery process of semi-submersible vehicles takes a long time，requires a high level of experience and technology of the operator，and the cost of trial and error is high. The VH-PPO algorithm is proposed to address this issue，and its performance is analyzed from three aspects，which are convergence，upper and lower bounds of expected convergence time，and time complexity. Using historical data successfully manipulated by human，the initial probability distribution is given and trained on this basis. The process of free exploration and continuous trial and error is eliminated and the expected convergence time can be effectively reduced，so that the training model can converge faster，and time complexity of the algorithm can be reduced. Choosing better hyperparameters for different stages of training to prevent overshoot and undershooting can help the training model converge better，reduce the upper and lower bounds of the expected convergence time，and thus reduce the time complexity of the algorithm. The algorithm is used for reinforcement learning in OpenAI Gym. After training，it is applied to the control software. The model is validated and further adjusted in a certain sea area. The experimental results show that as the number of experiment increases，the adaptability of the intelligent agent in the real environment gets better and better，and auxiliary control commands account for more than 50% of the total control commands，which effectively relieves the fatigue of the operator，and reduces the difficulty of novice training and the

收稿日期：2023-10-23

作者简介：万骏（1990-），男，硕士，工程师，主要从事控制算法研究。

threshold for replacing operators.

Key words　semi-submersible vehicle；algorithm complexity；visual interface；OpenAI Gym；reinforcement learning；auxiliary control

0　引言

相比于传统的水下机器人，半潜式航行体由于增加了裸露在空气中的桅杆装置，既可用于无线电通信、定位，也可为柴油机引擎提供进排气系统，用途更加广泛[1-2]。但也正因为桅杆的存在，深度受限、防碰撞等一系列风险随之而来，如何安全回收成为了一大难题。

其中一种可行的办法是利用牵引装置，母船拖曳半潜式航行体使两者同速同向，再由捕捉装置将其回收至母船上，这样做有 3 个好处：1）牵引时母船保持一定航速匀速拖曳，有利于稳定半潜式航行体姿态[3]。若出现意外使两者脱离，母船与半潜式航行体距离会越来越远，有效地降低了意外撞船风险。2）牵引绳伸出舷外或布放在母船正后方，牵引时母船与半潜式航行体保持一定距离，既方便回收也能保证其安全性。3）易实现，对于整套流程的各个岗位，非专业人员经过一定时间训练也可以熟练掌握其技巧[4]。

在风、流、浪等外界干扰下，半潜式航行体需不断调整垂直舵[5]来保证其按照预期的航线追逐牵引绳，并寻找合适的时机上浮，完成拖曳。决定回收成败的关键技术之一就是如何操控半潜式航行体按预期航线航行，并寻找恰当的时机上浮[6]。操纵者通过肉眼观测，凭借经验、技术虽能很好地完成任务，但长时间的注意力高度集中难免会带来疲惫感，到了最后准备上浮停机时稍有不慎就可能导致回收失败，功亏一篑。且经验、技术均需要通过大量的练习来获得，大大地提高了替换操纵者的门槛。若使用自动控制技术，仅需牵引绳与半潜式航行体的位置、速度、航向等信息，加上合适的控制策略便可实现航迹的控制[7]，且在精确数值计算方面，电脑比人脑更快、更准确，但海上的环境本身复杂多变，仍无法完全离开人工操控。本文主要讨论如何获得合适的控制策略来辅助人工操控，并寻找到最佳上浮时机，提高回收成功率。

获得控制策略有多种途径，其中使用最广泛的就是 A2C 框架下的强化学习算法[8]。近年来，汽车自动驾驶技术从辅助驾驶已逐渐走向无人驾驶研究[9]，其中，L2 级辅助驾驶功能已成为多数现售车型的标配[10]。近端策略优化算法（Proximal Policy Optimization，PPO）基于 A2C 框架，其广泛应用于定速巡航、车道保持，在高速公路上行驶时开启可有效减缓驾驶疲劳，深受广大司机们的喜爱[11]。借鉴于汽车的自动驾驶，水下设备的自动控制技术也开始逐步探索，鲍轩使用 PPO 对水下机器人目标抓取进行了仿真试验，取得了一定效果，但发现在复杂环境中算法过早收敛，出现了局部最优而非全局最优[12]。颜承昊等人通过 PPO 学习到的策略控制 AUV 进行网箱巡检，由于该问题的特殊性及马尔科夫性，只需求解局部最优即可，但训练速度缓慢[13]。鉴于以上传统 PPO 算法的缺点，不少改进型 PPO 算法相继问世，胡致远等人通过改进 PPO 算法的网络结构，通过最大值池化处理方法有效降低了系统维数，避免陷入局部最优且具有更快的训练速度，但同时牺牲了计算精度[14]。李沐阳提出一种基于优秀经验集的 PPO 算法，通过增加一种优秀经验采集机制，并动态调节裁剪因子等超参数来寻求训练速度更快、收敛更稳定、效果更好的控制策略，但人工调整超参数需要丰富的经验及反复试错[15]。因此，本文提出基于初始概率的变超参数 PPO 算法（Variable Hyperparamenter based on initial probability Proximal Policy Optimization，VH-PPO），按照 stable-baselines3 接口规范自定义仿真环境，在 OpenAI Gym 上仿真验证后进行海上实验，验证了真实环境下的有效性。

1　基于初始概率的变超参数 PPO 算法

1.1　算法流程

根据大量人工操作的试验数据，可拟合出各状态下的策略概率分布，替代原初始条件下的均匀分布。基于初始概率的变超参数 PPO 算法流程如图 1 所示。

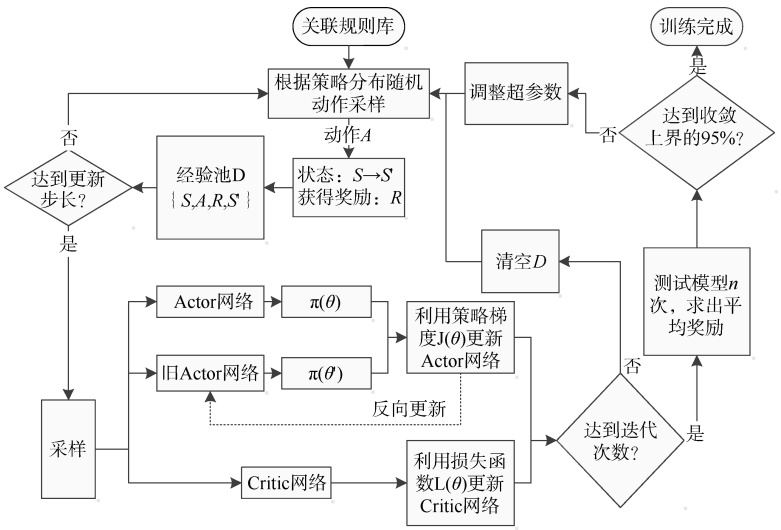

图 1　基于初始概率的变超参数 PPO 算法流程图

Fig. 1　Flow chart of VH-PPO algorithm based on initial probability

基于初始概率的变超参数 PPO 算法流程描述如下。

输入：Actor 网络参数 θ，旧 Actor 网络参数 θ'

输出：Actor 网络参数 θ

1. 初始化：以大量历史数据拟合出的概率分布覆盖原始均匀分布

2. for i from 1 to T 进行迭代

2-1. 根据当前状态 S 对应的动作概率分布，随机采样得到动作 A，智能体执行该动作与环境交互，的到新状态 S'和奖励 R

2-2. 将四元组{S，A，R，S'}存入经验池 D 中

2-3. 若 i=T，执行步骤 3，否则 $S'\rightarrow S$，i=i+1，然后跳转至步骤 2-1

3. for j from 1 to K 进行迭代

3-1. 从经验池中随机采样 M 条数据，分别输入至 Actor 网络得到策略 $\pi(\theta)$，输入至旧 Actor 网络得到策略 $\pi(\theta')$

3-2. 利用策略梯度 $J(\theta)$ 更新 Actor 网络，利用损失函数 $L(\theta)$ 更新 Critic 网络

3-3. 若 j=K，执行步骤 4，否则 j=j+1，然后用更新后的 Actor 网络权重反向更新旧 Actor 的网络参数，清空经验池 D 并跳转至步骤 2

4. 测试模型 n 次，求出平均奖励值。若平均奖励值未达收敛上界的 95%，则调整超参数接着训练，跳转至步骤 2。否则，训练完成

1.2　算法复杂度分析

1.2.1　收敛性

分析最极端情况，将结束条件平均奖励值达收敛上界的 95%改为奖励值达收敛上限，且此为最优解。由于实际问题的复杂性，需要提出一些必要的假设。

假设 1：经验池随机变量序列记为 D_n。且每一轮迭代中，经验池存在最优解，记为 D；

假设 2：奖励函数存在收敛上界；

假设 3：智能体与任意环境交互 1 个步长周期 T 时，经验池 D 存在唯一的最优解。

当假设 1、2、3 成立时，只需证明该算法找到一个最优解的概率为 1，就能证明该算法收敛，且 D_n 依概率收敛于 D。

证明如下。

假设在迭代次数 j 时找到最优解，其概率为

$$P(A_1, A_2, A_3, \cdots, A_T) = \prod_{i=1}^{T} P_{ij} \tag{1}$$

式中，A_1，A_2，A_3，…，A_T 为每次采样的最优动作。

只要有 1 个动作不是最优解，那么便不能得到最优解，则至少有 1 个动作不是最优解的概率为

$$\overline{P_j} = 1 - \prod_{i=1}^{T} P_{ij} \tag{2}$$

在 n 个迭代周期 K 后找到最优解的概率为

$$P^* = 1 - \prod_{j=1}^{nK} (1 - \prod_{i=1}^{T} P_{ij}) \tag{3}$$

其中 $0 < P_{ij} \leq 1$，因此可得

$$\lim_{n\to\infty} P^* = \lim_{n\to\infty}(1-\prod_{j=1}^{nK}(1-\prod_{i=1}^{T}P_{ij})\ ) = 1 \qquad (4)$$

$$\forall\varepsilon>0, \lim_{n\to\infty} P(\left|D_n - D\right| < \varepsilon) = 1 \qquad (5)$$

则该算法收敛，且 $D_n \xrightarrow{P} D$ 。

1.2.2　期望收敛时间的上下界

定义 1：该随机过程为一个吸收态 Markov 链

$$X(t)\Big|_{t=0}^{+\infty}\left(\forall X(t)=\left(S(t),\ A(t)\right)\in Y\right) \qquad (6)$$

其最优状态空间为 $Y^*\in Y$ 。

定义 2：$\gamma>0$，若满足

①当 $t>\gamma$ 时，$P\{X(t)\in Y^*\}=1$；

②当 $0\le t<\gamma$ 时，$P\{X(t)\in Y^*\}<1$；

则称 γ 为该算法首次找到最优解的收敛时间。

定义 3：$\lambda(t_m)=P\{X(t_m)\in Y^*\}$ 为该算法在 t_m 时刻找到最优解而进入最优状态空间的概率。

由定义 1、2、3，记 $p_{\min}\le p_{ij}\le p_{\max}$，可知：

$$E\gamma=\int_0^{+\infty} t_m P(t=t_m)\mathrm{d}t_m=\int_0^{+\infty} t(\lambda(t_m)-\lambda(t_{m\text{-}1}))\mathrm{d}t_m=$$

$$\int_0^{+\infty}\int_t^{+\infty}(\lambda(t_m)-\lambda(t_{m\text{-}1}))\mathrm{d}t_m\mathrm{d}t=$$

$$\int_0^{+\infty}(\lim_{t_m\to\infty}\lambda(t_m)-\lambda(t))\mathrm{d}t=$$

$$\int_0^{+\infty}(1-\lambda(t))\mathrm{d}t \qquad (7)$$

通过给予初始概率分布可有效减小 t_m 值，从而有效减小 $E\gamma$ 值。

由积分中值定理和介质定理可知：

$$(1-\lambda(0))\sum_{m=0}^{+\infty}(1-{P_{\max}}^T)^{K_m}\le E\gamma\le(1-\lambda(0))\sum_{m=0}^{+\infty}(1-{P_{\min}}^T)^{K_m}$$

由 $\lambda(0)=0$，$\dfrac{1}{1-x}=\sum_{n=0}^{+\infty}x^n$ 可确定 $E\gamma$ 上下界：

$$\frac{1}{1-(1-{P_{\max}}^T)^K}\le E\gamma\le\frac{1}{1-(1-{P_{\min}}^T)^K} \qquad (8)$$

针对训练时不同的阶段，选择更优的超参数，防止超调和欠调现象，可有效降低迭代次数 T、K，从而使 $P_{\min}$ 和 $P_{\max}$ 均减小，降低了 $E\gamma$ 的收敛上界与收敛下界。

1.2.3　时间复杂度

由表 1 及式（8）可知，算法的时间复杂度为

$$O(T\cdot K)=O(n^2) \qquad (9)$$

因此，在强化学习训练及设计仿真模型时，T、K 不宜过大。通过提供初始概率分布和超参数调优，均能有效的减小 T、K 值，从而降低时间复杂度。

2　仿真分析

2.1　仿真环境建模

OpenAI Gym 为开发者提供了丰富的环境，帮助开发者们研究和开发强化学习算法，是该领域最广泛使用的工具之一[16]。Stable-baselines3 提供了 PPO 基础算法及自定义环境接口，能够快速完成强化学习算法的搭建和评估。

根据文献[3]，建立半潜式航行体空间运动数学模型。通过历史试验数据，可通过采集的经纬度信息描绘出航线，并结合各个时刻的航向、舵角更加精准的修正模型[17]。仿真中的上浮时间根据文献[18]计算。

2.2　半潜式航行体空间运动数学模型

2.2.1　坐标系

已知半潜式航行体的基本结构多为回转形式，并结合动力、运动学理论为值构建空间运动方程[12]。在半潜式航行体浮心横截面交轴线中心坐标系 o_{xyz} 原点，通过地面、流体坐标系的相对角度可得到半潜式航行体的横滚角 φ、偏航角 ψ、俯仰角 θ，根据原点位置 x_0、y_0、z_0 可确定其空间位置[13]。

地面坐标系转换为流体坐标系的矩阵为

$$\boldsymbol{C}_b^0=\begin{pmatrix}\cos\theta\cos\psi & \sin\theta & -\cos\theta\sin\psi\\ -\sin\theta\cos\psi\cos\varphi+\sin\psi\sin\varphi & \cos\theta\cos\varphi & \sin\theta\sin\psi\cos\varphi+\cos\psi\sin\varphi\\ \sin\theta\cos\psi\cos\varphi+\sin\psi\cos\varphi & -\cos\theta\sin\varphi & -\sin\theta\sin\psi\sin\varphi+\cos\psi\sin\varphi\end{pmatrix} \qquad (10)$$

流体坐标系转换为地面坐标系的矩阵为式（10）的转置矩阵：

$$\boldsymbol{C}_0^b=\begin{pmatrix}\cos\theta\cos\psi & -\sin\theta\cos\psi\cos\varphi+\sin\psi\cos\varphi & \sin\theta\cos\psi\sin\varphi+\sin\psi\cos\varphi\\ \sin\theta & \cos\theta\cos\varphi & -\cos\theta\sin\varphi\\ -\sin\psi\cos\theta & \sin\theta\sin\psi\cos\varphi+\cos\psi\sin\varphi & -\sin\theta\sin\psi\sin\varphi+\cos\psi\cos\varphi\end{pmatrix} \qquad (11)$$

2.2.2 空间运动方程组

分析航行体的动力学特性，结合动量、动量矩定理在流体坐标系中构建的空间运动方程组[14]，得到：

$$\begin{aligned}&(m+\lambda_{11})\dot{v}_x - my_c\dot{\omega}_z + mz_c\dot{\omega}_y + \\&m\left[v_z\omega_y - v_y\omega_z - x_c\left(\omega_y^2+\omega_z^2\right)+y_c\omega_x\omega_y+z_c\omega_z\omega_x\right]= \\&-(G-B)\sin\theta+T+X(m+\lambda_{22})\dot{v}_y+(mx_c+\lambda_{26})\dot{\omega}_z- \\&mz_c\dot{\omega}_x+m\left[v_x\omega_z-v_z\omega_x+x_c\omega_x\omega_y+z_c\dot{\omega}_y\omega_z-y_c\left(\omega_x^2+\omega_z^2\right)\right]= \\&-(G-B)\cos\theta\cos\varphi+Y\end{aligned} \tag{12}$$

$$\begin{aligned}&(m+\lambda_{33})\dot{v}_z-(mx_c-\lambda_{35})\dot{\omega}_y+my_c\dot{\omega}_x+ \\&m\left[v_y\omega_x-v_x\omega_y-x_c\omega_z\omega_x+y_c\omega_y\omega_z+z_c\left(\omega_x^2+\omega_z^2\right)\right]= \\&(G-B)\cos\theta\sin\varphi+Z(J_{xx}+\lambda_{44})\dot{\omega}_x+my_c\dot{v}_z- \\&mz_c\dot{v}_y+my_c\left(v_y\omega_x-v_x\omega_y\right)+mz_c\left(v_z\omega_x-v_x\omega_z\right)+ \\&\left(J_{zz}-J_{yy}\right)\omega_y\omega_z=G\cos\theta\left(y_c\sin\varphi+z_c\cos\varphi\right)- \\&B\cos\theta\left(y_b\sin\varphi+z_b\cos\varphi\right)+M_x\end{aligned} \tag{13}$$

$$\begin{aligned}&\left(J_{yy}+\lambda_{55}\right)\dot{\omega}_y+mz_c\dot{v}_x-(mx_c-\lambda_{35})\dot{v}_z+ \\&mz_c\left(v_z\omega_y-v_y\omega_z\right)+mx_c\left(v_x\omega_y-v_y\omega_x\right)+ \\&\left(J_{xx}-J_{zz}\right)\omega_z\omega_x=-G\left(x_c\cos\theta\sin\varphi+z_c\sin\varphi\right)+ \\&B\left(x_b\cos\theta\sin\varphi+z_b\sin\varphi\right)+M_y\end{aligned} \tag{14}$$

$$\begin{aligned}&\left(J_{zz}+\lambda_{66}\right)\dot{\omega}_z+my_c\dot{v}_x-(mx_c-\lambda_{26})\dot{v}_y+ \\&mx_c\left(v_x\omega_z-v_z\omega_x\right)+my_c\left(v_y\omega_z-v_z\omega_y\right)+ \\&\left(J_{zz}-J_{yy}\right)\omega_x\omega_y=G\left(y_c\sin\theta-+x_c\cos\theta\cos\varphi\right)- \\&B\left(y_b\sin\theta-+x_b\cos\theta\cos\varphi\right)+M_z\end{aligned} \tag{15}$$

式中：m、G 分别表示航行体的质量、重力，（v_x、v_y、v_z）（ω_x、ω_y、ω_z）（J_{xx}、J_{yy}、J_{zz}）分别代表速度、角速度、转动惯量分量；x_c、y_c、z_c 与 x_b、y_b、z_b 分别为航行体质心、浮心在流体坐标系内的坐标；X、Y、Z 与 M_x、M_y、M_z 各自描述了流体动力主矢量、主力矩在流体坐标系中的阻力、升力、侧力分量以及横滚力矩、偏航力矩、俯仰力矩；λ、B、T 分别为附加质量、浮力、推力；参数上方“·”为其对应变化率。

ω_x、ω_y、ω_z 与姿态角变化率 $\dot{\psi}$、$\dot{\theta}$、$\dot{\varphi}$ 的关系为

$$\begin{bmatrix}\dot{\psi}\\ \dot{\theta}\\ \dot{\varphi}\end{bmatrix}=\begin{bmatrix}0 & \sec\theta\cos\varphi & \sec\theta\sin\varphi\\ 0 & \sin\varphi & \cos\varphi\\ 1 & -\tan\theta\cos\varphi & \tan\theta\sin\varphi\end{bmatrix}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} \tag{16}$$

半潜式航行体的空间位置取决于：

$$\begin{pmatrix}\mathrm{d}x_0/\mathrm{d}t\\ \mathrm{d}y_0/\mathrm{d}t\\ \mathrm{d}z_0/\mathrm{d}t\end{pmatrix}=\boldsymbol{C}_0^b\begin{pmatrix}v_x\\ v_y\\ v_z\end{pmatrix} \tag{17}$$

攻角 α、侧滑角 β 和速度 v 的定义如下：

$$\alpha=-\arctan\left(v_y/v_x\right) \tag{18}$$

$$\beta=\arctan\left(v_z/\sqrt{v_x^2+v_y^2}\right) \tag{19}$$

$$v=\sqrt{v_x^2+v_y^2+v_z^2} \tag{20}$$

2.2.3 OpenAI Gym 及 stable-baselines3 相关设置

在操控过程中，智能体根据具体策略随机采样得到动作 A 与环境交互，当前环境 S 在动作 A 的影响下变为 S'。其中，S 可以通过传感器采样获得，A 为半潜式航行体的操控指令。其状态空间如表 1 所示，其中下标为 1 表示母船状态，下标为 2 表示半潜式航行体状态，下标为 3 表示操控动作。

表 1 状态空间与动作空间设置

Table 1 Settings for state space and action space

类别	序号	符号	含义
状态空间	1	lon_1	经度
	2	lat_1	纬度
	3	v_1	航速
	4	head_2	航向
	5	lon_2	经度
	6	lat_2	纬度
	7	s_2	柴油机转速
	8	v_2	航速
	9	a_2	舵角
	10	d_2	深度
	11	t	UTC 时间
动作空间	1	s_3	柴油机转速设置
	2	a_3	舵角设置

根据 UTC 时间 t 和 lon_2、lat_2，可绘制出半潜式航行体的运动轨迹。

其中，数据可视化过程使用 render 函数[19]。每一帧的图像对应于每一时刻的状态 S[20]。

2.3　最佳上浮时机分析

在母船直线行驶过程中，牵引绳在海水表面的阻力下呈抛物线状态[21]。上浮过早或过晚都会导致失败，在仿真环境中可分析理想条件下最佳上浮时机，其上浮态势抽象表示如图 2 所示。

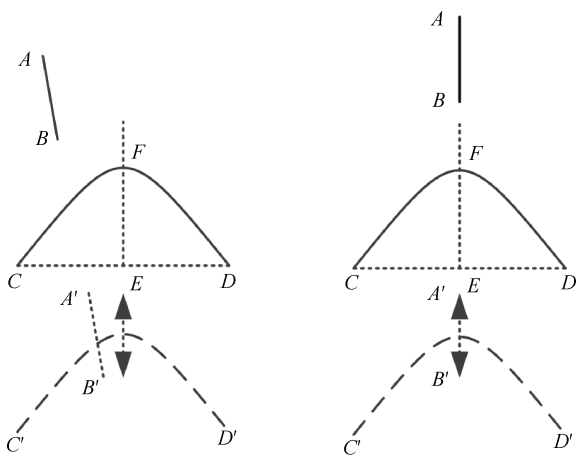

图 2　半潜式航行上浮态势抽象表示

Fig. 2　Abstract representation of buoyancy situation in semi-submersible navigation

记半潜式航行体可挂绳区域为线段 AB，牵引绳两端点分别为 C、D，过线段 CD 的中点 E，作线段 CD 的垂线交抛物线于 F，即：EF 是线段 CD 的中垂线。根据文献[14]计算出上浮时间后，可预估 AB 浮出水面时与抛物线 CD 的虚像 $A'B'$、$C'D'$，若 $A'B' \cap C'D' \neq \varnothing$，则挂绳成功。

理论上 $A'B'$、$C'D'$在任意处相交都可成功。但实际操控中这一过程由人工操控，需要一定的时间训练才可以在脑海中构建出虚像 $A'B'$、$C'D'$，且在风、流、浪等外界干扰下，AB 经历完上浮过程后并不可能完全与 $A'B'$重合，因此，需要留出一定的余量，余量越大，则成功率更高，且更安全。假设人工实际操作与期望上浮时机的时间偏差为 $t \in (-\Delta t,\ \Delta t)$，母船与半潜式航行体航速差为 Δv，记 $A'B'$的中点为 G，下面来证明 $A'B'$在线段 $C'D'$的中垂线上，且 $A'B'$与抛物线 $C'D'$的交点 F 重合于 A'B'的中点 G 时为上浮最佳时机。

①由对称性可知，当人工实际操作与期望上浮时机的时间存在偏差时，其容错距离为 $L = 2\Delta v \cdot t \le L_{\max}$，当 $A'B'$的中点 G 与抛物线 $C'D'$上的点重合时容错性最高。

②假设 $A'B'$与 EF 的夹角为 α，则最大容错距离为 $L_{\max} = L_{AB}\cos\alpha$。在 $\alpha = 0$ 时取得最大值，此时 $L_{\max} = L_{AB}$，即：$A'B' /\!/ EF$ 时容错距离最大。

③在风、流、浪等外界干扰下，$A'B'$实际位置存在偏差，当满足①和②时，可最大程度的抵抗前后偏差。为了抵抗左右偏差，需选择牵引绳对称轴位置，即抛物线 $C'D'$中点 F。

综合①–③可知：$A'B'$在线段 $C'D'$的中垂线上，且 $A'B'$与抛物线 $C'D'$的交点 F 重合于 $A'B'$的中点 G 时为上浮最佳时机。

2.4　奖励函数设置及其收敛上界的存在性

奖励函数的设置是强化学习环境搭建过程中最重要的部分，奖励函数往往很大程度上影响算法的收敛性能以及模型的训练速度。本文中主要设置了以下几个奖励函数。

1）航迹偏差奖励。

航迹偏差奖励主要目的是为了不断激励半潜式航行体靠近牵引绳两端中垂线，为其创造最佳上浮时机条件。记当前时刻半潜式航行体 $A'B'$的中点

G 所在的位置离 EF 的距离为 d，则航迹偏差奖励为

$$r_d = 1 - \mathrm{clip}(\log(\frac{\max(d,\varepsilon)}{d_{\max}}),0,1) \qquad (21)$$

式中：clip 是一个截断函数，主要是为了将 r_d 的大小限制在[0，1]内；$d_{\max}$ 为 G 距离 EF 的最大允许偏差；为防止对数函数中的真数为 0，此处设置保护系数 $\varepsilon = 0.001$。

2）航向偏差奖励。

航向偏差奖励主要目的是为了不断激励半潜式航行体尽量与牵引绳两端中垂线平行，为其创造最佳上浮时机条件。记 $A'B'$与 EF 的夹角为 α，则航向偏差奖励为

$$r_\alpha = 1 - \mathrm{clip}(\log(\frac{\max(\alpha,\varepsilon)}{\alpha_{\max}}),0,1) \qquad (22)$$

式中，$\alpha_{\max}$ 为 $A'B'$与 EF 的夹角最大允许偏差。

3）回合结束奖励。

$$r_{\text{end}}\begin{cases}100, & \text{半潜式航行体正常上浮}\\ -100, & \text{半潜式航行体到达仍未上浮}\\ -200, & \text{半潜式航行体触碰边界}\\ 0, & \text{手动终止}\end{cases}$$

因此，总奖励可以表达为

$$R_{\text{sum}}=\sum_{i=1}^{T}\gamma^{T-i}(r_{di}+r_{\alpha i})+r_{\text{end}} \tag{23}$$

式中：γ为折扣因子；T为式（1）中的迭代次数，是有限值，因此奖励函数必然存在收敛上界。

2.5 仿真结果分析

通过 register 注册自定义环境，并对相关参数进行设置。本文中使用的部分参数设置如表 2 所示。

表 2 部分参数设置

Table 2 Partial parameter settings

类别	序号	符号及单位	数值
初始值	1	v_1 / kn	4
	2	v_2 / kn	5
	3	s_2 /（r/s）	700
	4	a_2 /（°）	0
	5	d_2 / m	1.2
	6	L_{AB} / m	2
	7	L_{CD} / m	6
	8	$d_{\max}$ / m	0.5
	9	$\alpha_{\max}$ /（°）	2
	10	γ	0.99
	11	T	300
	12	K	1 000
	13	N	5
动作空间	1	s_3 /（r/s）	[700，1 000]
	2	a_3 /（°）	[–10，10]

本文中仿真平台配置为 Windows11 系统，i7-10875H 八核处理器，16 GB 内存，CUDA 12.2，装载的 Python 版本为 3.8.17，以 stable_baselines3 库函数中对应版本的 PPO 算法为基础，使用 Optuna 超参数调优库[22]。通过从给定范围中不断测试择优来替代人工调优，对其改进前后进行训练，由 $A'B'$的中点 G 点不断靠近 F 使其重合，并通过奖励函数的设置让其尽可能学习到沿 CD 的中垂线路径行驶的控制策略。由于 K=1 000，因此每训练 1 000 次进行 1 次超参数调优并保存当前训练的模型。最终，得到训练模型过程中的奖励值，如图 3 所示。

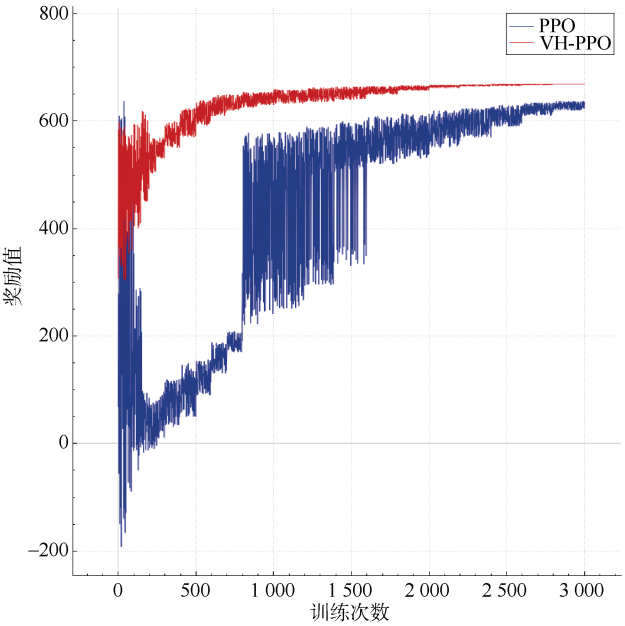

图 3 改进前后奖励值比较

Fig. 3 Comparison of reward values before and after improvement

由图 3 可知，由于 VH-PPO 有初始动作空间的概率分布，失败次数很少，大约训练 200 次左右就能一直成功上浮；每 1 000 次训练经过超参数调优，收敛性更好；在训练 2 000 次后偶尔能达到 95%收敛上界；训练 3 000 次后稳定达到 95%收敛上界。基础 PPO 算法在最初自由探索时，基本上大概率失败，小概率成功；学习到一定经验后基本稳定失败；在训练 700~800 次左右开始偶尔成功、偶尔失败，因为失败后奖励值–200，成功奖励值+100，因此从奖励值图像上来看差距非常大；在训练 1 600~1 700 次后基本稳定成功，但是奖励函数不如 VH-PPO，需要更多的训练次数才能达到收敛上界的 95%。分别加载 VH-PPO 训练 1 000、2 000、3 000 次的模型，提取半潜式航行体的轨迹，如图 4 所示。

由图 4 可知，随着训练次数的增加，半潜式航行体的运动轨迹更容易趋向于一条直线，使操纵者更容易抓住最佳上浮时间从而挂绳成功。

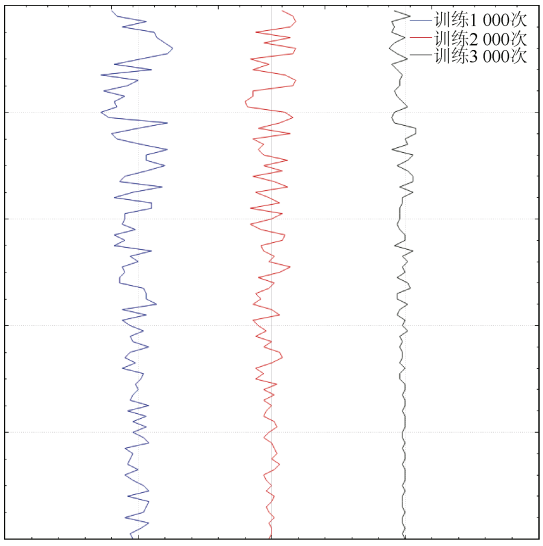

图 4　训练 1 000、2 000、3 000 次的运动轨迹
Fig. 4　Movement trajectories for trainings of 1 000, 2 000 and 3 000

3　海上试验分析

由于海上试验为真实环境，无法获取 F 处的经纬度，而半潜式航行体的经纬度为桅杆位置，与 G 有一定偏差，但其也在 CD 的中垂线上，影响不大。因此，将训练好的模型加载于便携式操控仪时，仅作为辅助决策功能使用，当人工操控时小幅修改 F 点的定位将其重新迭代回模型并修正。在某海域的回收试验中，半潜式航行体的运动轨迹如图 5 所示。统计辅助决策操作数与总操作数，其操作占比如图 6 所示。

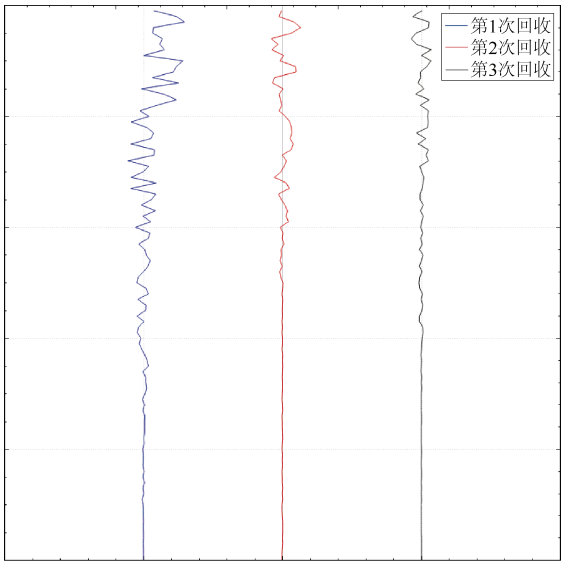

图 5　海上试验的运动轨迹
Fig. 5　Movement trajectories of sea trials

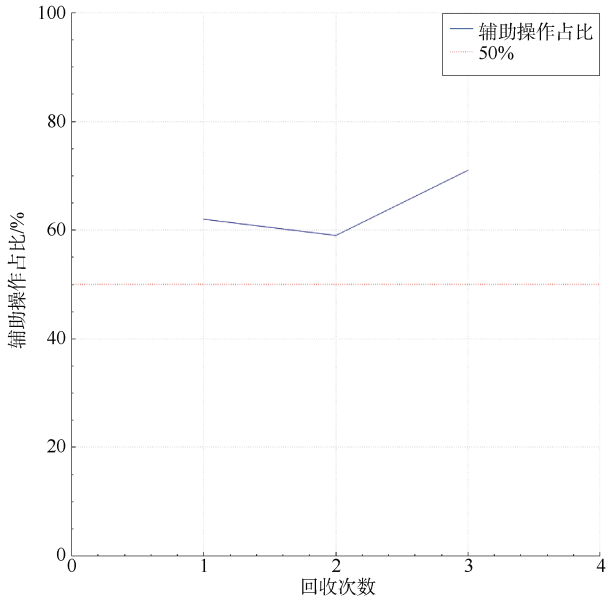

图 6　辅助操作占比
Fig. 6　Proportion of auxiliary operation

从图 5 可以看出，由于在途中半潜式航行体已成功上浮挂绳，在牵引绳的牵引作用下半潜式航行体几乎保持与母船同速同向，且在人工干预下，上浮前一刻基本已保持直线行驶。分析图 6 可知，平均辅助操控指令在总操控指令中占比超过 50%，可有效地减缓操纵者长期操控时的疲劳，降低了新手训练难度及替换操纵者的门槛。

4　结束语

根据历史数据，为 PPO 算法提供初始概率分布，应用超参数调优库 Optuna 进行超参数调优，设计基于初始概率的变超参数 PPO 算法，选用合适参数训练出此框架下较为优秀的模型，完成半潜式航行体的辅助操控功能，并在真实试验中不断修正模型。最后，通过海上试验测试其实际效果，结果表明：辅助决策操作在总操作数中占比超过 50%，有效地减缓了操纵者的疲劳，降低了新手训练难度及替换操纵者的门槛。但面对更恶劣的海况，可能仍需不断修正改正模型，甚至修改奖励函数，重新训练新的策略。

参考文献

[1] 易谷丰. 半潜式航行器运动特性研究[J]. 舰船电子工程，2015，35（6）：128-132.

[2] 易谷丰. 半潜式航行器安全控制策略研究[J]. 舰船电子工程，2013，33（1）：128-130.
[3] 欧阳凌浩，师子锋. 半潜式航行体横滚调整方式分析[J]. 舰船科学技术，2013，35（2）：63-67.
[4] 刘栋. 基于高海况条件下水面收放技术的研究与设计[J]. 机械管理开发，2012（5）：47-48.
[5] 欧阳凌浩，田振华. 半潜式航行体拖曳系统收放过程动态响应[J]. 水雷战与舰船防护，2014，22（4）：41-45.
[6] 龚喜，于亦凡，刘诗玉. 基于 PID 控制的半潜式航行器缩比模型耐波性分析[J]. 水雷战与舰船防护，2017，25（4）：20-24.
[7] 董校成. UUV 水下自主回收路径规划与运动控制研究[D]. 大连：大连海事大学，2022.
[8] 王日中，李慧平，崔迪，等. 基于深度强化学习算法的自主式水下航行器深度控制[J]. 智能科学与技术学报，2020，2（4）：354-360.
[9] 李浩. 基于元强化学习的无人驾驶车辆行为决策研究[D]. 大连：大连理工大学，2021.
[10] 韩胜明，肖芳，程纬森. 深度强化学习在自动驾驶系统中的应用综述[J]. 西华大学学报：自然科学版，2023，42（4）：25-31.
[11] 王兆维. 基于 PPO 算法的智能汽车端到端深度强化学习控制研究[D]. 长春：吉林大学，2021.
[12] 鲍轩. 基于近端策略优化算法的水下机器人目标抓取仿真验证[J]. 舰船科学技术，2020，42（23）：121-128.
[13] 颜承昊，林远山，李然，等. 一种基于 PPO 的 AUV 网箱巡检方法[J]. 计算机与数字工程，2023，51（1）：93-97.
[14] 胡致远，王征，杨洋，等. 改进 PPO 算法的 AUV 路径规划研究[J]. 电光与控制，2023，30（1）：87-91，102.
[15] 李沐阳. 基于 EER-PPO 算法的自主水下机器人路径跟踪及自主避障研究[D]. 济南：山东大学，2022.
[16] BROCKMAN G，CHEUNG V，PETTERSSON L，et al. OpenAI Gym[EB/OL]. [2016-06-05]. https：//arxiv.org/pdf/1606.01540.pdf.
[17] 熊玮. 不完全时间序列与纵向数据的建模研究[D]. 长春：吉林大学，2023.
[18] 陈佳华，吕海宁. 环境对航行体上浮速度和出水姿态的影响研究[J]. 装备制造技术，2023（2）：24-30.
[19] 刘广泽. 基于人机协作的深度强化学习电子游戏算法的研究与实践[D]. 北京：北京邮电大学，2020.
[20] 姜文翼. 基于神经网络的角色动画运动控制器优化研究[D]. 厦门：厦门大学，2019.
[21] 毛磊. 高海况水下设备的回收技术研究[D]. 北京：中国舰船研究院，2015.
[22] 张颖. 深度学习模型超参数优化的研究[D]. 北京：首都经济贸易大学，2020.

（责任编辑：曹晓霖）