机器人
Robot
ISSN 1002-0446,CN 21-1137/TP

《机器人》网络首发论文

题目：水下滑翔蛇形机器人滑翔控制的强化学习方法
作者：张晓路，李斌，常健，唐敬阁
DOI：10.13973/j.cnki.robot.180398
收稿日期：2018-06-07
网络首发日期：2019-03-26
引用格式：张晓路，李斌，常健，唐敬阁．水下滑翔蛇形机器人滑翔控制的强化学习方法[J/OL]．机器人. https://doi.org/10.13973/j.cnki.robot.180398

网络首发时间：2019-03-26 18:20:43
网络首发地址：http://kns.cnki.net/kcms/detail/21.1137.TP.20190326.1110.001.html

DOI：10.13973/j.cnki.robot.180398

水下滑翔蛇形机器人滑翔控制的强化学习方法

张晓路[1,2,3]，李 斌[2,3]，常 健[2,3]，唐敬阁[2,3,4]

（1. 东北大学信息科学与工程学院，辽宁 沈阳 110819；
2. 中国科学院沈阳自动化研究所机器人学国家重点实验室，辽宁 沈阳 110016；
3. 中国科学院机器人与智能制造创新研究院，辽宁 沈阳 110016； 4. 中国科学院大学，北京 100049）

摘 要：研究了一种强化学习算法，用于水下滑翔蛇形机器人的滑翔运动控制．针对水动力环境难以建模的问题，使用强化学习方法使水下滑翔蛇形机器人自适应复杂的水环境，并自动学习仅通过调节浮力来控制滑翔运动．对此，提出了循环神经网络蒙特卡洛策略梯度算法，改善了由于机器人的状态难以完全观测而导致的算法难以训练的问题，并将水下滑翔蛇形机器人的基本滑翔动作控制问题近似为马尔可夫决策过程，从而得到有效的滑翔控制策略．通过仿真和实验证明了所提出方法的有效性．

关键词：强化学习；水下滑翔蛇形机器人；马尔可夫决策过程；循环神经网络

中图分类号：TP24 **文献标识码：**A

A Reinforcement Learning Method for Gliding Control of Underwater Gliding Snake-like Robot

ZHANG Xiaolu[1,2,3]，LI Bin[2,3]，CHANG Jian[2,3]，TANG Jingge[2,3,4]

(1. *College of Information Science and Engineering*, *Northeastern University*, *Shenyang* 110819, *China*;
2. *The State Key Laboratory of Robotics*, *Shenyang Institute of Automation*, *Chinese Academy of Sciences*, *Shenyang* 110016, *China*;
3. *Institutes for Robotics and Intelligent Manufacturing*, *Chinese Academy of Sciences*, *Shenyang* 110016, *China*;
4. *University of Chinese Academy of Sciences*, *Beijing* 100049, *China*)

Abstract: A reinforcement learning algorithm for gliding control of underwater gliding snake-like robot is studied. To solve the problem that the hydrodynamic environment is hard to be modeled, a reinforcement learning method is adopted so that the underwater gliding snake-like robot can adapt to the complex water environment and automatically learn the gliding actions only by adjusting buoyancy. A Monte Carlo policy gradient algorithm using recurrent neural network is proposed to solve the problem that the algorithm is difficult to train because the robot state can't be fully observed. The gliding action control of the underwater gliding snake-like robot is approximated as Markov decision processes (MDPs), so as to obtain an effective gliding control policy. Simulation and experiment results show the effectiveness of the proposed method.

Keywords: reinforcement learning; underwater gliding snake-like robot; Markov decision process; recurrent neural network

1 引言（Introduction）

随着复杂水下环境中的探索需求日益增长，对相应探测设备的续航能力和机动性能有了更高的要求．水下滑翔机[1-2]具有功耗低、噪声小和续航能力强的优点，适用于大江和海洋等宽阔水环境，但是其固定型的结构和仅仅依靠净浮力来驱动的特点导致水下滑翔机机动性差，使其难以有效转向及避障；蛇形和鱼形等仿生水下机器人[3-4]具有高机动性的特点，可以灵活应用于水下建筑裂纹检测、水下设备维修和水下救援等复杂水下环境任务中，但存在能效低、续航差的缺点．因此，文 [5] 提出一种将水下滑翔机和水下蛇形机器人有效结合的新型水下滑翔蛇形机器人（underwater gliding snake-like robot，UGSR），其兼具高机动性和长续航的优点．本文基于此模型展开研究，并对 UGSR 通过控制浮力有效前进的滑翔动作进行研究．

蛇形机器人的机身形状不规则，水下环境具有高度复杂性，所以 UGSR 在水中与水流的相互作用使得水动力环境有着极高的不确定性，难以使用传统的水动力分析对环境建立精确的数学模型．强化学习[6]（RL）方法中“无模型（model-free）”[6-7]

基金项目：国家重点研发计划（2017YFB1300101）；国家自然科学基金青年基金（61803365）．
通信作者：常健，changjian@sia.cn 收稿/录用/修回：2018-06-07/2019-01-17/2019-01-25

的一类算法有着不必对环境进行精确的数学建模、不需要先验知识以及对环境适应性强等特点，因此强化学习算法为 UGSR 的控制策略提供了新的思路．强化学习是指智能体通过与外部环境进行交互来学习如何把状态映射到动作来达到某个长期目标从而获取最大化奖励的学习范式[6]．强化学习算法具有很强的环境自适应能力和对复杂系统的自学能力，将机器人控制问题转化为马尔可夫决策过程（MDP）[8]，在机器人领域有着越来越广泛的研究和应用．基于表格的传统强化学习算法往往是对状态或者状态－动作对求取值函数并建立表格进行存储，以供之后的搜索和更新．但是在 UGSR 的应用中，包含位置、关节角度、姿态、速度和加速度等特征的状态都是连续特征，多自由度的机器人也经常有着较高维数的连续动作，单纯地对状态和动作进行离散化往往造成“维数灾难”问题，因此需要采取适用于连续状态、动作空间的函数近似方法．神经网络能以任意精度拟合任意的非线性连续函数，因此本文以此来逼近强化学习中 UGSR 控制策略的非线性特性．

由于在实际应用中机器人状态难以完全观测，因此使用强化学习的 UGSR 的滑翔控制问题实际上属于部分可观测马尔可夫决策过程（partially observable Markov decision process，POMDP）[9-10]．本文在经典的蒙特卡洛策略梯度（Monte Carlo policy gradient，MCPG）算法[5]基础上作了改进，提出了一种采用循环神经网络（recurrent neural network，RNN）的蒙特卡洛策略梯度算法，称之为循环蒙特卡洛策略梯度法（recurrent Monte Carlo policy gradient，RMCPG）．使用一种称为长短期记忆[11]（long short-term memory，LSTM）的循环神经网络来记忆机器人经历的观测信息，以此来拟合机器人的真实状态，有效改善了状态的部分可观测问题．

本文首先对循环神经网络 LSTM 单元和强化学习算法 MCPG 分别作了介绍，然后通过对两者结合提出了 RMCPG 算法，并针对 UGSR 的滑翔动作分别使用经典的 MCPG 算法和 RMCPG 算法作了仿真对比，验证了 RMCPG 算法的高效性，最后对 UGSR 的滑翔动作进行实验验证了算法的有效性．

2 水下滑翔蛇形机器人的设计（Design of the underwater gliding snake-like robot）

本文采用的水下滑翔蛇形机器人是在中国科学院沈阳自动化研究所郁树梅等设计的水下蛇形机器人“探索者”[12]基础上进行的改进．UGSR 包含中间 5 个 2 自由度关节和头尾 2 个伸缩关节，在蛇身两侧安装 2 个平板的滑翔翼．UGSR 结构示意如图 1 所示．

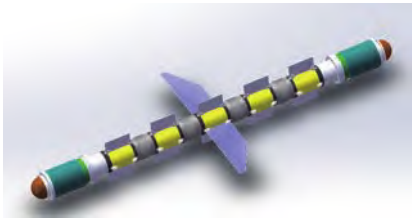

图 1 水下滑翔蛇形机器人结构示意图

Fig.1 Structure diagram of the underwater gliding snake-like robot

中间 5 个关节用于调节机器人的姿态，两头的伸缩关节通过丝杠单元控制伸缩量来改变浮力大小，从而控制上浮下潜，而滑翔翼的存在就是配合各个关节的角度和伸长量来实现 UGSR 的滑翔运动．UGSR 中间 5 个 2 自由度关节装有角度舵机用于控制关节角度，从而通过关节的摆动来实现驱动前进和改变方向等运动．按照 UGSR 的设计，在 2 个伸缩关节的伸缩量为 0 时，可以悬浮在水中．参考文 [3] 对 UGSR 的结构设计及样机模型进行了详细描述．

3 UGSR 滑翔控制算法（Gliding control algorithm of UGSR）

强化学习及其控制问题可以看成一个智能体在随机环境中序贯地选择动作串，从而与环境交互来获取最大的奖励．把机器人的控制问题转化成一个 MDP，包括一个状态空间 $\mathcal{S}$、一个动作空间 $\mathcal{A}$、一个初始状态分布 $p_1(s_1)$ 和一个状态转移概率 $p(s_{t+1}|s_t,a_t)$．根据马尔可夫性质，强化学习中状态转移到下一个状态的情况只跟当前的状态和动作有关，而与经历过的状态动作无关．即：

$$p(s_{t+1}|s_1,a_1,\cdots,s_t,a_t)=p(s_{t+1}|s_t,a_t) \tag{1}$$

其中 s_t 代表 t 时刻的状态，a_t 代表 t 时刻所选的动作．对于任意状态动作序列 $s_1,a_1,r_1,s_2,a_2,r_2,\cdots,s_T$，奖励函数 $r:\mathcal{S}\times\mathcal{A}\to\mathbb{R}$．强化学习就是要找到一个从状态空间映射到动作空间的策略，使智能体可以根据策略解决 MDP 问题．UGSR 的伸缩关节伸长量、身体的倾斜角等都是连续的特征，因此状态也是连续值．本文利用神经网络对连续非线性函数的拟合能力，设计用循环神经网络来拟合具有连续状态的 UGSR 控制问题．

UGSR 在与水环境进行交互时，比较容易获得其头尾关节在水中的位置以及倾斜角等观测结果，

难以得到其速度、加速度、角速度和角加速度等动力学相关的信息．如果不考虑寻找先进的传感器来获取这些参数，而仅使用可以直接得到的观测结果，将使该滑翔控制不满足马尔可夫性．

文 [13] 通过对深度 Q 网络（deep Q network，DQN）算法添加循环神经网络层提出了 DRQN 算法，对部分可观测马尔科夫决策过程取得了良好的试验效果．为了感知 UGSR 的动力学，本文对简单实用的策略梯度算法作类似的改进，提出了使用 LSTM 记忆单元的 RMCPG 算法．

3.1 循环神经网络和 LSTM

为了解决 UGSR 的控制转化为 MDP 时遇到的 POMDP 问题，本文使用循环神经网络来记忆长期的观测信息，以此来近似出机器人的实际状态．

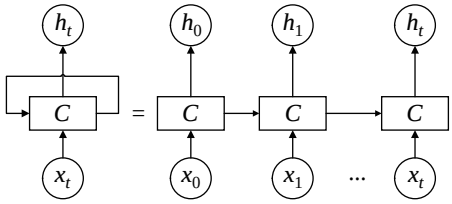

图 2 循环神经网络

Fig.2 Recurrent neural network

循环神经网络示意图如图 2 所示，C 为神经网络结构，也可称之为循环神经网络单元．x_t 和 h_t 分别为 C 的输入和输出．RNN 的隐藏层不仅与当前的输入有关，还和之前经历过的输入特征有关．网络只用一套神经网络结构 C，参数通过穿越时间反向传播（backpropagation through time，BPTT）来更新．普通的 RNN 存在梯度爆炸和梯度消失问题，并不能真正地处理好长距离的依赖，本文选用一种特殊的 RNN——长短时记忆（LSTM）网络——来拟合强化学习算法．LSTM 网络指的是以 LSTM 单元来替换普通 RNN 的全连接隐藏层．LSTM 单元示意图如图 3 所示．

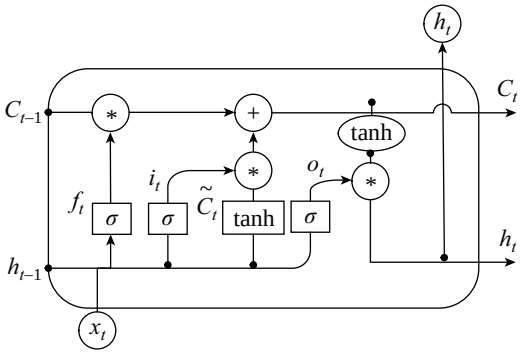

图 3 长短期记忆单元

Fig.3 LSTM

LSTM 对普通 RNN 无法处理长期依赖的问题作了改进，增加了长期依赖项 C_t，通过 3 个 Sigmoid 函数 σ 形成的输入门 i_t、遗忘门 f_t 和输出门 o_t 来控制过往的状态信息的记忆．

3.2 蒙特卡洛策略梯度法

由于水下环境极其复杂、UGSR 结构不规则，因此对水环境难以精确地建立模型．本文采用一种不依赖模型的方法来研究 UGSR 在水中滑翔运动的控制问题．蒙特卡洛策略梯度法作为一种典型的不依赖模型的实用算法，通过经验平均来代替随机变量的期望，并摒弃值函数更新的方法，可直接拟合强化学习策略．

UGSR 在未知水环境模型的情况下，设定滑翔控制目标，按一个初始化的策略 π 产生多回合状态动作轨迹，每回合都是从一个初始状态开始滑翔，在终止状态结束（2 种状态的选取参见 4.1 节），然后计算实验回合中每个状态的折扣累计奖赏返回值：

$$G_t(s) = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{T-1-t} r_T \tag{2}$$

其中 $\gamma \in [0,1]$ 是折扣因子，r_t 为 t 时刻的即时奖赏．

本文 UGSR 在学习滑翔运动实验中，采用循环神经网络来拟合当前的控制策略 $\pi_{\boldsymbol{\theta}}$，然后通过优化一个目标函数来更新策略的参数，从而得到最佳策略．这里选取目标函数为策略 $\pi_{\boldsymbol{\theta}}$ 下的累计奖赏期望：

$$J(\pi_{\boldsymbol{\theta}}) = \sum_s \mu_\pi(s) \sum_a \pi_{\boldsymbol{\theta}}(a|s,\boldsymbol{\theta}) q_\pi(s,a) \tag{3}$$

其中 $\mu_\pi(s)$ 代表状态 s 的概率分布；$\pi_{\boldsymbol{\theta}}(a|s,\boldsymbol{\theta})$ 代表 UGSR 根据当前策略 $\pi_{\boldsymbol{\theta}}$ 在状态 s 选择动作 a 的概率；$q_\pi(s,a)$ 为状态动作值函数，代表 UGSR 在状态 s 下选择动作 a 后直到终止状态得到的折扣累计奖赏．策略梯度法将滑翔运动的策略学习转化为优化问题，求解该优化问题可用梯度上升的方法：

$$\boldsymbol{\theta}_{\text{new}} = \boldsymbol{\theta}_{\text{old}} + \alpha \nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) \tag{4}$$

其中 $0 < \alpha < 1$ 称为学习率或者更新步长．

使用蒙特卡洛方法的采样实验得到的返回值 $G_t(s)$ 代替 $q_\pi(s,a)$．则目标函数梯度可求得：

$$\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) = E_\pi[G_t \nabla_{\boldsymbol{\theta}} \log \pi(a_t|s_t,\boldsymbol{\theta})] \tag{5}$$

这里选用代价函数为负的目标函数：

$$L(\pi_{\boldsymbol{\theta}}) = -E_\pi[G_t \log(\pi(a_t|s_t,\boldsymbol{\theta}) + \varepsilon^+)] \tag{6}$$

其中 ε^+ 为一个正的极小量，以防出现对 0 取对数的错误．最小化该损失函数就是最大化目标函数 $J(\pi_{\boldsymbol{\theta}})$．重复实验和更新直到得到一个可以接受的策略 $\pi_{\boldsymbol{\theta}}$ 为止．

3.3 循环蒙特卡洛策略梯度法

普通的蒙特卡洛策略梯度法以观测到的状态作为输入，而对于 UGSR 这种所处环境复杂、运动形态比较复杂的机器人将遇到状态不完全观测的问题，如涉及水环境动力学的各关节速度、加速度、角速度、角加速度等无法直接观测，以所获取的状态来训练强化学习算法将无法转化为马尔可夫决策过程．UGSR 的动作不仅与当前的观测信息相关，还与之前一定时间步的观测信息相关，因此利用 LSTM 的记忆特性，以滑动窗口的方法将近期一定数量时间步的观测作为 LSTM 的输入来认知机器人的状态．为了加强神经网络的拟合能力，在输入层后和输出层前分别添加全连接层，中间的记忆层采用多个 LSTM 单元．图 4 为本文采用的神经网络的拓扑图．

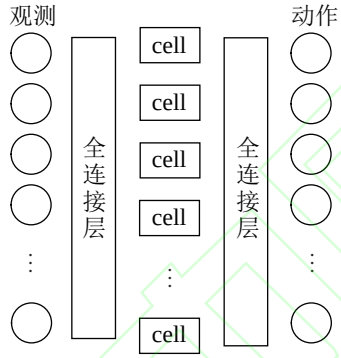

图 4 循环蒙特卡洛策略梯度法的神经网络拓扑图

Fig.4 Neural network topology of RMCPG

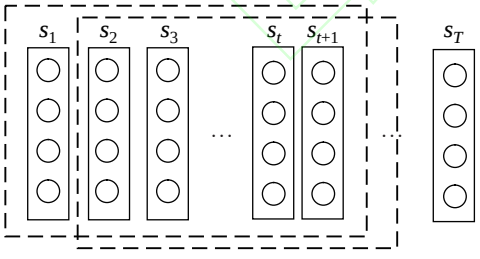

图 5 LSTM 滑动窗口生成

Fig.5 Sliding window generation of LSTM

为了简化计算，采用固定周期的 LSTM，即 RNN 每次循环采用固定的步数 L．通过滑窗的方式，将蒙特卡洛实验得到的状态组合成相应的 RNN 输入，图 5 为蒙特卡洛实验的一个完整回合中相邻 2 次滑动窗口的生成示意．

将每个滑动窗口中的所有状态按时间序列输入 RNN，滑动窗口的最后一个状态对应的输出作为该周期 RNN 的输出，并通过 softmax 的方法得到每个动作对应的概率，以此概率选择动作．当每个蒙特卡洛实验回合结束后，通过式 (2) ～ (5) 来更新 RNN 的网络连接权重，从而不断优化机器人的控制策略．

为了便于理解该修改后算法的实现，表 1 给出 RMCPG 算法的伪代码．

表 1 RMCPG 算法

Tab.1 RMCPG algorithm

RMCPG 算法
1 输入：神经网络代表的参数化策略 $\pi(a\|s,\boldsymbol{\theta})$，$\forall a\in\mathcal{A}$，　$s\in\mathcal{S}$，$\boldsymbol{\theta}\in\mathbb{R}^n$
2 初始化策略 π 的循环神经网络参数 $\boldsymbol{\theta}$
3 循环直到条件满足：
4　按 softmax 策略得到的概率生成一回合实验：$s_1,a_1,r_1,s_2,a_2,r_2,\cdots,s_T$
5　对本回合状态观测进行均值为 0、方差为 1 的回合标准化
6　填充 L 个初始状态作为 LSTM 首个窗口
7　**for** $t=1,2,\cdots,T$ **do**:
8　　对当前窗口的状态按时序输入 LSTM，接收最后一个状态的输出动作和即时奖赏
9　　求窗口末端状态 S_t 之后的折扣累计奖赏：$G_t(S_t)\leftarrow r_{t+1}+\gamma r_{t+2}+\cdots+\gamma^{T-1-t}r_T$
10　　$\boldsymbol{\theta}_{\text{new}}=\boldsymbol{\theta}_{\text{old}}+\alpha\nabla\log\pi(a_t\|S_t,\boldsymbol{\theta})G_t$
11　以步长 n 滑动窗口产生新的 RNN 输入
12　**end for**
13 输出：策略网络 $\pi(\boldsymbol{\theta})$

4 强化学习在滑翔控制中的应用（Application of the reinforcement learning to gliding control）

为了缩短训练实验消耗的大量时间并减小反复训练过程对 UGSR 造成的机械损伤，机器人应先在虚拟环境中做仿真实验，等得到较好的效果后，再把训练得到的参数网络保存下来移植到实体机器人上作为初始化参数网络．为此搭建了一个专用于强化学习算法训练的水动力虚拟环境，在虚拟环境中产生 UGSR 和水的相互作用，使得 UGSR 有相应的状态改变，而不获取虚拟环境计算得到的相互作用力、速度、加速度、角速度和角加速度等水动力相关的数据，仅利用可以观测的各关节的坐标和伸缩量生成状态，从而更加真实地模拟在真实水环境中

的实验.

4.1 滑翔控制强化学习算法的要素设计

UGSR 主要有 2 类运动方式：第 1 类是像水下蛇形机器人一样，通过 2 自由度关节的联合摆动来产生推动力，从而驱动 UGSR 灵活地前进；第 2 类是像水下滑翔机一样靠控制浮力的变化来更加节省能耗地实现滑翔运动. 本文对第 2 类运动进行研究，通过控制 UGSR 头尾两个伸缩关节的伸缩量来改变浮力，驱动自身滑翔. 通过强化学习，UGSR 可以学会以滑翔运动的方式实现最大程度的水平分位移.

4.1.1 状态 $\boldsymbol{s}$ 的选取

根据 RMCPG 算法，神经网络结构中 LSTM 单元的循环输入周期为 L. 对于每个观测到的状态，其特征选取为 UGSR 中间关节中心点坐标 (x,z)、头关节伸缩量 d_{h}、尾关节伸缩量 d_{t} 和身体的倾斜角 θ. 每个观测状态为 $\boldsymbol{s}=(x,z,d_{\mathrm{h}},d_{\mathrm{t}},\theta)$，其中横坐标 x 以水平向右为正方向，纵坐标 z 以垂直向下为正方向. 以向下滑翔为例，初始状态选为机器人在水面某位置的平衡状态 $\boldsymbol{s}=(0,0,0,0,0)^{\mathrm{T}}$.

4.1.2 动作 a 的选取

机器人运动的动作往往是连续的，若 UGSR 选取连续动作，伸缩关节的最大可伸缩量为 $d_{\mathrm{threshold}}$，那么头尾关节的动作均为 $[-d_{\mathrm{threshold}},d_{\mathrm{threshold}}]$ 区间内的连续值. 根据文 [3]，关节的伸缩是舵机经由丝杠转化而来，难以实现机器人实时运动控制时相邻动作较大的突变，因此应摒弃绝对伸缩量，选择相邻动作的相对变化量 Δd 作为动作，这样就有 3 个可能动作：伸长 Δd，缩短 Δd，不伸缩. 2 个伸缩关节就可以组合出 9 个可能的动作，每个 t_{step} 的时间选择一次动作. 这种离散化与对 $[-d_{\mathrm{threshold}},d_{\mathrm{threshold}}]$ 的范围直接离散化相比，大大降低了维度，避免了维数灾难的发生.

4.1.3 初始、终止状态的选取

强化学习算法中智能体运行起始于一个初始状态. 以向下滑翔为例，初始状态可选为机器人在水面某位置的平衡状态 $\boldsymbol{s}=(0,0,0,0,0)^{\mathrm{T}}$.

强化学习中算法每回合的实验都要有一个终止状态或说吸收状态. 当到达该状态时，本回合实验就终止，下回合实验就要从初始状态开始重新训练. 为了实现高效的滑翔，节省训练成本，本文设置了一个倾角阈值 $\theta_{\mathrm{threshold}}$，$\theta\in[-\theta_{\mathrm{threshold}},\theta_{\mathrm{threshold}}]$. 设定水环境的范围为 $x\in[0,x_{\mathrm{threshold}}]$，$z\in[0,z_{\mathrm{threshold}}]$. 当倾斜角或 UGSR 的位置达到或者超过设定范围时，即进入终止状态，一个回合结束. 为防止永远到达不了终止状态，设置了一个最大时间步限制：$T\leqslant T_{\mathrm{threshold}}$.

4.1.4 即时奖励 r 的选取

由于采取蒙特卡洛方法，每次策略更新都需要从起始状态到终止状态完成一次完整实验. 机器人不像仿真环境中的智能体那样可以轻易获取海量的实验数据，因此要求所需实验次数尽可能少. 由于开始训练时采取初始化的随机策略，前期的很多次实验都达不到目标.

为了使 UGSR 通过滑翔运动达到设定的水平分位移，设置每个时间步的奖励 r 正比于当前水平坐标与上一时间步水平坐标的差值. 本文以水平向右为横轴 ox 正方向，垂直向下为纵轴 oz 正方向. 为了保证方向性，要实现向上滑翔运动时，当前纵坐标 z_2 应小于上个时间步的纵坐标 z_1. 因此向上滑翔的即时奖励函数可以设置为

$$r=k_x(x_2-x_1)-k_zI(z_2>z_1) \tag{7}$$

向下滑翔时：

$$r=k_x(x_2-x_1)-k_zI(z_2<z_1) \tag{8}$$

其中 $I()$ 为指示函数，满足括号内条件时函数值为 1，否则为 0，k_x 和 k_z 为权重系数. 为了减少收敛到局部最优的情况，当本回合实验 UGSR 终止且状态未达到预定目标时，得到一个负奖励 $r=-1$.

4.2 强化学习的应用方法

针对 UGSR 的结构特点设计好强化学习算法的各个要素之后，便可对多个滑翔动作使用强化学习算法进行训练.

以本文采用的直接策略迭代算法为例，无论使用该类型何种算法，都需要根据 4.1.1 节的方法选择 UGSR 的状态，然后对状态信息作出对应的处理，并输入到强化学习策略函数中. 从策略函数的输出中获取下一步动作. 在 UGSR 的滑翔控制中，策略函数的输出对应于头尾两个伸缩关节下一步的伸缩情况. 动作作用于环境之后，头尾关节的伸长量分别产生了对应大小的浮力，从而调整了转矩，调节了 UGSR 的姿态. 针对不同的姿态，滑翔翼的存在可保证 UGSR 产生滑翔运动. 动作完成后，根据奖赏函数得到一个反馈奖赏值，并转移到新的状态. 以该顺序反复运行，根据得到的 (s,a,r,s') 数据跟随相应的算法来更新策略函数的参数，直到策略函数参数收敛，可以得到奖赏最大化的最优策略或者近似最优策略. UGSR 则根据收敛后的策略函数对每个状态选择动作，从而完成目标滑翔运动.

5 滑翔动作的仿真（Gliding action simulation）

本节进行 UGSR 基本滑翔动作的仿真，分别使用 MCPG 和 RMCPG 进行对比分析．一个完整的滑翔运动由向下滑翔和向上滑翔 2 个部分构成，其中向上滑翔运动的初始状态为向下滑翔运动的终止状态．5.1 节和 5.2 节分别对 2 种滑翔动作做了仿真，仿真参数接近真实的 UGSR 参数．设置 $\theta_{\text{threshold}}=0.8$ rad，$x_{\text{threshold}}=5$ m，$z_{\text{threshold}}=10$ m，$d_{\text{threshold}}=3$ cm，$\Delta d=0.5$ cm，更新时间 $t_{\text{step}}=0.2$，$T_{\text{threshold}}=3000$，$k_x=2$，$k_z=1$．本文 RNN 隐藏层采用了 12 个 LSTM 单元，状态输入的循环周期 $L=15$．除了这些设定值，还需要根据实验过程手动调整学习率、折扣因子和神经网络的结构等超参数．其余的参数由神经网络自己学习更新．

UGSR 的滑翔角 γ_{gliding}、攻角 α_{attack} 和俯仰角 θ_{pitch} 等方向和坐标如图 6 所示．

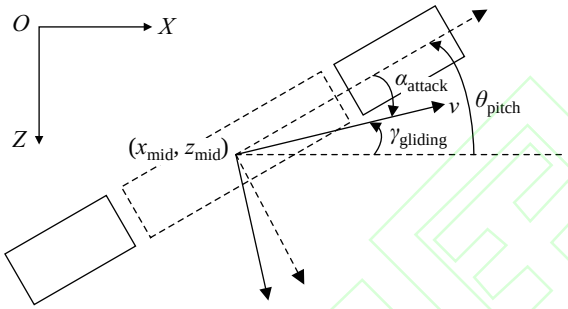

图 6 UGSR 的方向和坐标示意图

Fig.6 Direction and coordinate diagram of UGSR

5.1 向下滑翔运动的仿真

该仿真选择神经网络的学习率 α 是自衰减的：

$$\alpha=\alpha_{\text{start}}\times(\gamma_{\text{decay}})^{\frac{N_{\text{learn}}}{n_{\text{learn}}}} \tag{9}$$

其中 α_{start} 代表初始学习率，γ_{decay} 代表衰减率，N_{learn} 代表已经训练的总学习次数，n_{learn} 代表每次衰减所需学习次数．式 (9) 说明每经过 n_{learn} 次学习，α 都要按衰减率衰减 1 次．本文初始值 $\alpha_{\text{start}}=0.001$，$n_{\text{learn}}=50$，衰减率 $\gamma_{\text{decay}}=0.995$．

5.1.1 普通 MCPG 算法

设置即时奖励函数为式 (8)，根据 MCPG 算法，以普通全连接神经网络来拟合策略，每个时间步的状态输入都为该时间步的观测．仿真的代价函数和平均返回值的曲线如图 7 和图 8 所示．

可以看到，开始的前面部分回合由于策略的随机初始化，得到的返回值 $G_t(S_t)$ 均值为负．因为 $\pi(a_t|s_t,\boldsymbol{\theta})\in[0,1]$，$\log(\pi(a_t|s_t,\boldsymbol{\theta})+\varepsilon)<0$，所以损失函数式 (6) 也为负值．随着训练的进行，UGSR 通过学习逐渐可以得到正的奖励，返回值均值逐渐大于 0，代价函数值也变成正数．大约 2500 回合之后，返回值均值就比较稳定地收敛到一个正值．而代价函数则在大约 3000 回合后逐渐收敛到 0 附近．直观的理解是，累计奖励返回值稳定代表着 UGSR 能按相对固定的路线达到目标位置．而参数网络对于每个输入状态产生的输出动作的确定性在慢慢增高，因此 $\pi(a_t|s_t,\boldsymbol{\theta})$ 在慢慢趋向于 1，$\log(\pi(a_t|s_t,\boldsymbol{\theta})+\varepsilon)$ 逐步增大到 0，代价函数 $L(\pi_{\boldsymbol{\theta}})$ 也逐渐减小到 0．

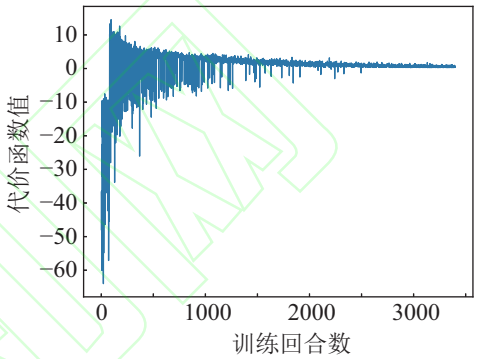

图 7 向下滑翔运动的 MCPG 代价函数仿真曲线

Fig.7 Simulation curve of MCPG cost function in downward gliding motion

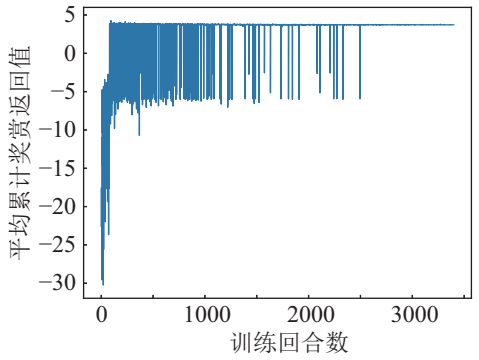

图 8 向下滑翔运动的 MCPG 平均返回值曲线

Fig.8 Average return curve of MCPG in downward gliding motion

5.1.2 RMCPG 算法的仿真

按 3.3 节的描述，采用循环神经网络拟合的 RMCPG 算法来训练 UGSR 的向下滑翔动作，则仿真结果如图 9 和图 10 所示．

从代价函数曲线可以看到，RMCPG 算法的折扣累计奖赏平均返回值在 300 回合左右就可以稳定到一个正的常数，即可以稳定到达预设的水平位移．代价函数也几乎同时收敛到 0 附近，训练速度极快．由此可见，采用循环神经网络的蒙特卡洛策略梯度法比采用普通全连接神经网络的蒙特卡洛策

略梯度法收敛更快，说明对于部分可观测马尔可夫决策过程，循环神经网络可以起到显著的作用.

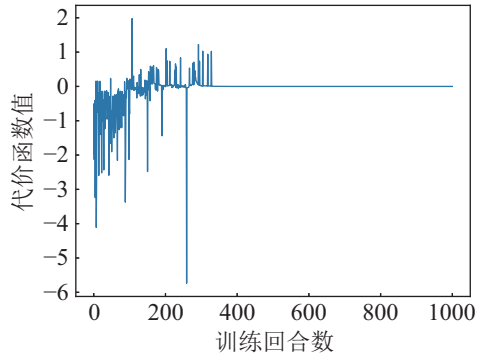

图 9　向下滑翔运动的 RMCPG 代价函数仿真曲线

Fig.9　Simulation curve of RMCPG cost function in downward gliding motion

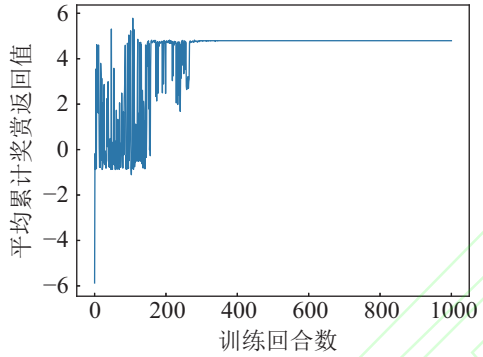

图 10　向下滑翔运动的 RMCPG 平均返回值曲线

Fig.10　RMCPG average return value in downward gliding motion

5.2　向上滑翔运动仿真

若仅从水下一个静止状态开始向上滑翔，则与向下滑翔类似，只需把即时奖励函数改为式 (7)，其余参数同向下滑翔. 但是考虑到实际情况下，向上滑翔运动都承接着向下滑翔运动，因此向上滑翔运动的起始状态应该是向下滑翔运动的末端状态，包括速度、加速度和伸缩量等. “转向”的速度方向和非零的状态观测无疑给训练增加了难度.

向上滑翔运动的训练，仅使用普通全连接神经网络拟合的 MCPG 算法，得到训练结果如图 11 和图 12 所示. 图 11 中，后期代价函数非常不稳定，并且多为负值. 由图 12 可见，平均奖赏返回值多为负值，即没能达到预期的向上滑翔设定，从而受到了惩罚，即得到负的奖赏. 根据式 (6)，以当前策略在某状态下产生错误的动作时产生负的奖赏值，代价函数也相应地为负值. 则由此可以判定算法无法收敛到一个合适的极值点，无法捕捉前期速度向下的趋势，没能有效地完成“转向”，没得到有效的控制策略.

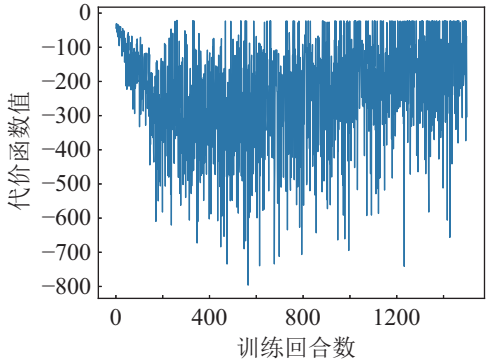

图 11　向上滑翔运动的 MCPG 代价函数仿真曲线

Fig.11　Simulation curve of MCPG cost function in upward gliding motion

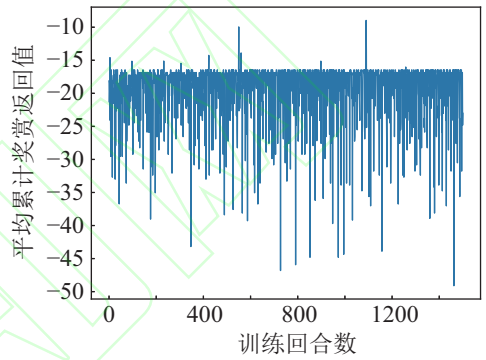

图 12　向上滑翔运动的 MCPG 平均返回值曲线

Fig.12　MCPG average return value in upward gliding motion

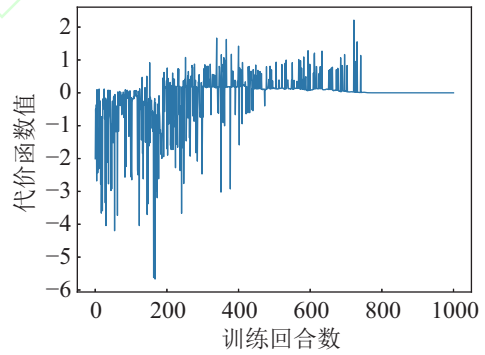

图 13　向上滑翔运动的 RMCPG 代价函数仿真曲线

Fig.13　Simulation curve of RMCPG cost function in upward gliding motion

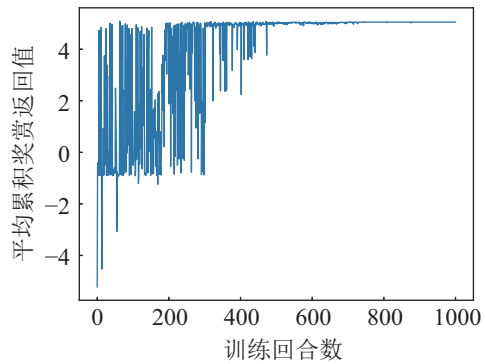

图 14　向上滑翔运动的 RMCPG 平均返回值曲线

Fig.14　RMCPG average return value in upward gliding motion

使用循环神经网络拟合的蒙特卡洛策略梯度法，得到仿真结果如图 13 和图 14 所示.

受向下的初速度影响，收敛速度要比向下滑翔时慢一些，但是还是较快地完成了向上滑翔动作的仿真，得到了有效的滑翔策略.

5.3 仿真效果

图 15 给出了 UGSR 使用 RMCPG 算法结合 2 种滑翔动作先向下后向上周期性滑翔时一回合实验的仿真示意图，其中机身姿态采样间隔为 1 s.

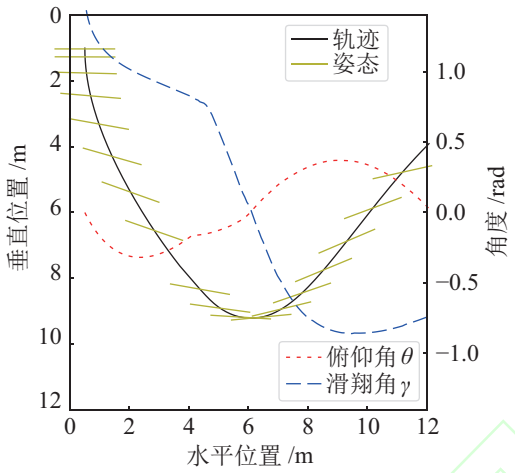

图 15 UGSR 姿态、轨迹和角度仿真示意图

Fig.15 Simulation diagram of the attitude, trajectory and angle of UGSR

定义纵坐标向下为正方向. 由图 15 可见，当 UGSR 向下滑翔到中点前时，俯仰角为负值，滑翔角为正值；当向上滑翔时，俯仰角增大到正值，滑翔角减小到负值. 包括滑翔角、俯仰角、轨迹和姿态的滑翔形态合乎常理，达到了预期的滑翔效果.

6 滑翔动作实验（Gliding action experiment）

从随机初始化参数开始反复地进行实体机器人的动作训练会非常耗时、耗力并且会带来大量的机械磨耗和机体损伤，因此本文在仿真训练收敛后的参数的基础上进行实验. 当前的机械实验平台搭建地尚缺乏适当的传感器，因此也未在仿真的基础上做后续的实体训练，仅将 RMCPG 算法仿真得到的动作序列按顺序输入给 UGSR，如图 16 所示，实现了 2 个基本的滑翔动作组合成的周期性滑翔运动.

图 16 表示实验中 UGSR 所采用的动作时间序列. 以下实验在中国科学院沈阳自动化研究所的深水实验池中完成.

该实验前期为向下滑翔运动，然后以向下滑翔运动的末端状态作为向上滑翔动作的初始状态，从而形成完整的周期性滑翔运动.

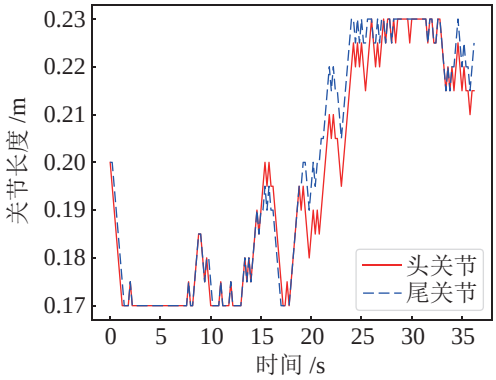

图 16 UGSR 周期性滑翔动作的时间序列

Fig.16 Time series of UGSR periodic gliding action

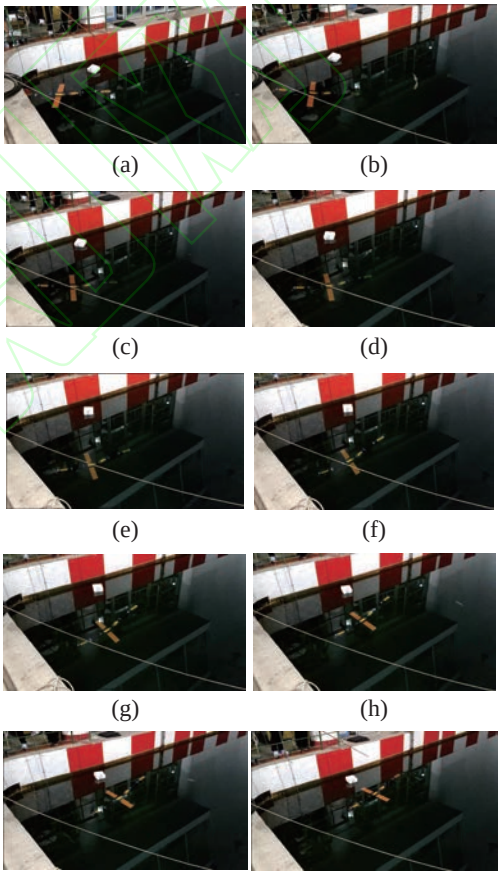

图 17 UGSR 周期性滑翔运动实验

Fig.17 Periodic gliding experiment of UGSR

初始状态时 UGSR 的伸缩关节处于原始长度，机身水平悬浮于水面. 实验开始时，UGSR 头部关节先开始缩短，而后尾部关节开始缩短，如图 16 中前 2 s 所示，头部关节始终要稍短于尾部关节，因此机身产生俯仰力矩，机器人的头部关节开始缓慢下沉，如图 17(a)(b) 所示. 头尾关节持续缩短到最短长度并保持 8 s 左右，水下滑翔蛇形机器人总体积减小、整体开始下沉并向前滑翔，如图 17(c)

所示．在图 17(a) ～ (c) 中产生了向下的俯仰角并且绝对值逐渐增大，但是从第 14 s ～ 18 s 中，头关节的长度始终大于尾关节，2 个伸缩关节的伸缩量将抑制俯仰角绝对值持续增大，从而使 UGSR 可以持续向前，不至于造成垂直向下运动而无法滑翔的现象．在图 17(c) ～ (e) 中俯仰角逐渐减小，并于图 17(e) 处于近似水平状态．待俯仰角基本为 0 后，UGSR 保持水平方向的分速度，仍继续向前滑翔．

之后头尾关节均开始伸长，俯仰角正向增大，开始向上滑翔，俯仰角在图 17(e) ～ (g) 中逐渐增大并向上滑翔运动．在俯仰角足够大时，俯仰角于图 (g) ～ (j) 中再逐渐减小，整个过程都保持着向上滑翔动作，并于图 (j) 后到达水平面．

整个实验向下滑翔最大深度约为 1.5 m，向前滑翔的最大水平分位移约为 1.5 m．该实验完成了周期性滑翔动作，验证了 RMCPG 算法的有效性．但是实验效果却与仿真效果存在一定的差异，尤其是滑翔距离并未达到仿真中设定的位移量．

究其原因主要有两方面．一是仿真采用的动力学与实际水环境动力学存在较大差距，且由于平台条件限制，并未反复从真实环境中进行动作学习，导致仿真中的动作与真实环境有一定的不匹配性．二是该样机尚处于简单试验阶段，仿真中设定的每个时间步间隔时间所需要的伸缩量未必可以在样机实验中实现．而且，目前缺少足够的传感器无法完成闭环控制，导致位移量和速度均比仿真中小．

7 总结（Conclusions）

研究了强化学习方法在水下滑翔蛇形机器人的滑翔控制中的应用．提出了使用循环神经网络的蒙特卡洛策略梯度算法 RMCPG 来进行 UGSR 的滑翔动作学习．采用了 RNN 的 LSTM 单元对强化学习策略进行拟合，记忆了过往的状态观测，可以更好地解决部分可观测马尔可夫性，加速了算法的收敛．先通过仿真验证了算法的收敛性，又通过实体实验采用 RMCPG 算法仿真得到的动作序列完成了周期性滑翔动作，但是由于实验条件的限制并未进行大量的实体训练，导致实体效果与仿真效果存在较大差异．

为了实现效果更佳的实体机器人的滑翔动作，后续将在改进机器人结构后，安装合适的传感器进行实体机器人训练．

参考文献（References）

[1] Javaid M Y, Ovinis M, Nagarajan T, et al. Underwater gliders: A review[C]//4th International Conference on Production, Energy and Reliability. Paris, France: EDP Sciences, 2014: No.02020.

[2] 俞建成，张奇峰，吴利红，等．水下滑翔机器人运动调节机构设计与运动性能分析 [J]．机器人，2005，27(5)：390-395.
Yu J C, Zhang Q F, Wu L H, et al. Movement mechanism design and motion performance analysis of an underwater glider [J]. Robot, 2005, 27(5): 390-395.

[3] Ming A G, Ichikawa T, Zhao W J, et al. Development of a sea snake-like underwater robot[C]//IEEE International Conference on Robotics and Biomimetics. Piscataway, USA: IEEE, 2014: 761-766.

[4] 李立，王明辉，李斌，等．蛇形机器人水下 3D 运动建模与仿真 [J]．机器人，2015，37(3)：336-342.
Li L, Wang M H, Li B, et al. Modeling and simulation of snake robot in 3D underwater locomotion[J]. Robot, 2015, 37(3): 336-342.

[5] 唐敬阁，李斌，李志强，等．水下蛇形机器人的滑翔运动性能研究 [J]．高技术通讯，2017，27(3)：269-276.
Tang J G, Li B, Li Z Q, et al. Research on the gliding performance of underwater snake-like robots[J]. High Technology Letters, 2017, 27(3): 269-276.

[6] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. Cambridge, USA: MIT Press, 1998.

[7] Gläscher J, Daw N, Dayan P, et al. States versus rewards: Dissociable neural prediction error signals underlying model-based and model-free reinforcement learning[J]. Neuron, 2010, 66(4): 585-595.

[8] Puterman M L. Markov decision processes: Discrete stochastic dynamic programming[M]. Hoboken, USA: John Wiley & Sons, Inc., 1994.

[9] Monahan G E. State of the art – A survey of partially observable Markov decision processes – Theory, models, and algorithms[J]. Management Science, 1982, 28(1): 1-16.

[10] Wang X N, He H G, Xu X. Reinforcement learning algorithm for partially observable Markov decision processes[J]. Control and Decision, 2004, 19(11): 1263-1266.

[11] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[12] 郁树梅，马书根，李斌，等．水陆两栖蛇形机器人的上浮和下潜步态研究 [J]．仪器仪表学报，2011，32(S1)：276-279.
Yu S M, Ma S G, Li B, et al. Study on the floatation and submergence gait of amphibious snake-like robot [J]. Chinese Journal of Scientific Instrument, 2011, 32(S1): 276-279.

[13] Hausknecht M, Stone P. Deep recurrent Q-learning for partially observable MDPs[C]//AAAI Fall Symposium. El Segundo, USA: AI Access Foundation, Computer Science, 2015: 29-37.

作者简介：

张晓路（1995 –），男，硕士．研究领域：机器人运动控制，强化学习算法．

李　斌（1963 –），男，硕士，研究员．研究领域：仿生机器人，可重构机器人，月球探测机器人．

常　健（1983 –），男，博士，副研究员．研究领域：仿生机器人自主控制，智能化机器人行为控制，机械臂运动控制．