基于改进 DDPG 算法的船舶航迹跟随控制系统*

余凡　蒋晓明　张浩　曹立超　周勇　刘晓光

（广东省科学院智能制造研究所，广东 广州 510070）

摘要：鉴于船舶在航行时受到风、浪和流等不确定因素干扰，传统的船舶航迹控制方法难以在不确定环境且控制系统处于多输入、多输出的条件下精确建模，导致船舶容易偏离预设航迹，影响船舶行驶的安全性。为降低船舶偏航，实现船舶航迹的精准控制，将深度确定性策略梯度（DDPG）算法引入到控制系统。首先，分析船舶的运动学，详细介绍 DDPG 算法的基本原理并对算法进行改进；然后，在 Matlab/Simulink 中搭建船舶航迹跟随控制系统并进行仿真实验。实验结果表明，该系统稳定性好，能对外部干扰迅速做出响应，对船舶航迹控制具有一定的参考价值。

关键词：DDPG 算法；航迹跟随；船舶控制系统

中图分类号：U664.82　　**文献标识码：**A　　**文章编号：**1674-2605(2021)05-0004-06

DOI：10.3969/j.issn.1674-2605.2021.05.004

0　引言

在经济全球化的影响下，船舶行业的贸易占据了重要地位。随着船舶运动控制技术的不断完善，船舶行业朝着大型化、专业化、数字化和货物种类多样化方向发展。船舶相关技术的研究得到广泛关注，其中研究重点之一就是船舶运动控制自动化水平的提高[1]。船舶运动控制分为手动控制和自动控制[2-4]，手动控制对操作者的经验要求较高，不利于船舶在环境多变的海洋上航行，目前已形成自动控制代替手动控制的趋势。自动控制实现了航向和航迹保持[5]、航速控制[6-7]等功能，在提高船舶运动控制智能化[8-9]的同时，可以减少偏航次数、航向偏差；并在保证经济效益的同时，提高船舶和船员的安全性[10]。

船舶运动控制的核心问题是如何不断地改进控制策略，以保证在有干扰的环境下及船舶本身存在动态特性改变的情况下，仍能满足航运性能指标要求。由于船舶在航行中会受到风、浪和流的影响，且船舶控制系统为多输入多输出的动力学系统，在气候、水文、航道等不确定的外部因素和负载、动力等内部因素的影响下，无法建立准确的数学模型。采用端到端强化学习的方式[11]，不需要复杂的控制器，黑箱控制即可处理连续状态空间并输出连续的动作，可解决船舶控制模型难以精确建模的问题[12]。

本文在传统的船舶航迹跟随控制系统中引入深度确定性策略梯度（deep deterministic policy gradient, DDPG）算法，并把船舶航迹跟随控制系统建模成马尔可夫决策过程；改进 DDPG 算法，离线学习训练船舶航迹跟随控制系统；在 Matlab/Simulink 中搭建船舶航迹跟随控制系统并进行仿真实验，验证 DDPG 算法的有效性。

1　船舶航迹跟随控制系统设计

船舶航迹跟随控制系统主要用于船舶航向和航迹的保持。航向、航迹保持需要舵角δ克服环境干扰，将航向ψ维持在设定航向ψ_r上，将船舶的运动轨迹维持在规划轨迹上。此时，不仅需要消除航向误差，还需消除航迹误差η，如图 1 所示。

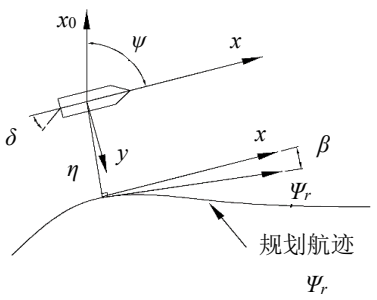

图 1　船舶航迹保持方式

* 基金项目：广东省海洋经济专项项目（GDNRC[2021]024）

航迹保持分为直接式控制和间接式控制 2 种。其中，直接式控制根据航向计划和 GPS 等定位传感器反馈的航向和船位信号直接控制舵角δ，通常用于航迹精度要求较高的场合；间接式控制将航迹和航向保持功能分开控制，由舵机控制航向误差和航迹误差η，引导船舶向消除航迹偏差的方向行驶。

船舶航行时，船舶航迹跟随控制系统会产生大量参数整定和复杂计算等问题，系统鲁棒性较差。为保证船舶航迹跟随的实时性，本文采用间接式航迹保持控制方式，并将 DDPG 算法引入控制系统，如图 2 所示。

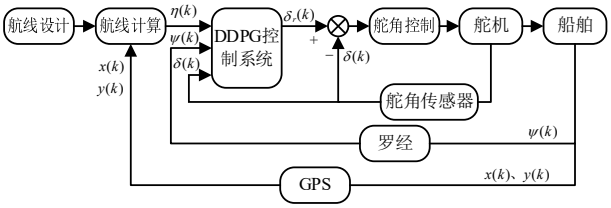

图 2　间接式航迹保持控制系统框图

图 2 中，航迹偏差$\eta(k)$是 GPS 接收船舶实际位置和规划航线之间的偏离值，是引导船舶沿规划轨迹行驶的重要参数；DDPG 控制系统通过将舵角传感器和罗经反馈的舵角$\delta(k)$和航向$\psi(k)$信号与航迹偏差$\eta(k)$进行融合，实现航迹和航向保持功能；舵角控制采用 PID 算法，可减少舵机实际转角与期望转角的误差。

2　船舶运动学分析

船舶航行时具有 6 个自由度，其中前进、横漂和起伏为 3 个平移自由度；转艏、横摇和纵摇为 3 个转动自由度。惯性坐标系与附体坐标系平面示意图如图 3 所示。

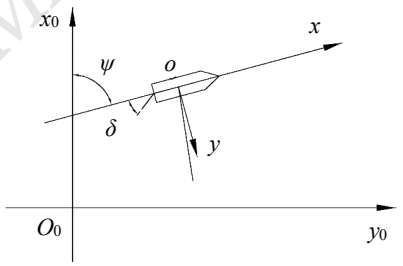

图 3　惯性坐标系与附体坐标系平面示意图

图 3 中，$x_0O_0y_0$为固定于地球表面的惯性坐标系，又称为北东坐标系；xoy为固定于船体的附体坐标系，其中 x 轴指向船首，y 轴指向右舷，符合右手定则。

船舶航迹跟随控制系统主要考虑航向角 ψ 以及航行轨迹x_0、y_0的变化；起伏、横摇和纵摇的运动状态对航迹控制的影响较小，可忽略。

由图 3 可知，船舶在 2 个坐标系间的运动学关系可表示为

$$\left.\begin{aligned}\dot{x}_0 &= u\cos\psi - v\sin\psi\\ \dot{y}_0 &= u\sin\psi + v\cos\psi\\ \dot{\psi} &= r\end{aligned}\right\} \tag{1}$$

式中，u、v、r分别为船舶在附体坐标系中的前进速度、横漂速度和转舵角速度。

对船舶舵机操作时，舵叶会受到水流的推动作用，产生绕附体坐标系原点 o 的力矩，在船舶合成速度 U（即设定航向ψ_r上）与 x 轴船首方向形成漂角β。漂角β决定船舶的航迹形状，并取逆时针方向为正，计算公式为

$$\beta = \arctan\left(-\frac{v}{u}\right) \tag{2}$$

3　DDPG 算法

3.1　马尔可夫决策过程

强化学习中智能体的动作取决于环境信息的反馈，同时受回报值的影响，朝回报最大化的方向寻找当前环境下智能体能达到预期效果的动作。其中，智能体是学习及实施决策的机器，与智能体相互作用的其他对象都被称为环境（即船舶）。智能体需具备学习能力且能够在某种程度上感知环境状态，并采取动作影响环境状态，如图 4 所示。

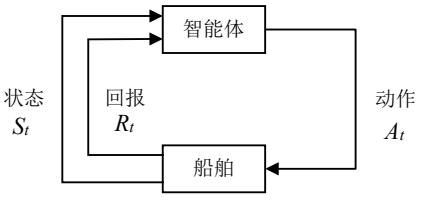

图 4　马尔可夫决策过程的智能体-船舶交互

图 4 中智能体的动作可以是任何决策，而状态则是船舶的位置、速度、外界干扰等反馈信息。在智能体与船舶进行信息交互前，不需要确定控制模型中具体参数值，可通过离线学习的方式收敛到最优值。

马尔可夫决策过程定义为一个数组(S, A, P, R, γ)，其中S为由船舶的质心位置和速度等给出的有限状态集；A 为船舶航迹跟随控制系统需调节参数的有限状态集；P^a 为状态 s 下采取动作 a 后转到状态 s_t 时的概率；$R\left(s_t=s,a_t=a\right)=E\left[r_t\middle|s_t=s,a_t=a\right]$为奖励函数；$\gamma$为折扣系数，且$\gamma\in[0,1]$。

由图 4 可知，智能体与船舶交互过程中涉及 3 个信号的传递：动作表示智能体做出的选择；状态表示做出该选择的基础；回报定义智能体的目标，并通过环境传递给智能体。智能体通过动作和状态函数计算得到的收益，且智能体的最终目标是最大化收到的长期累积收益，记为G_t。

$$G_t=R_{t+1}+\gamma R_{t+2}+\gamma^2R_{t+3}+\cdots=\sum\nolimits_{i=0}^{\infty}\gamma^iR_{i+t+1} \tag{3}$$

式中，折扣系数γ是对未来可能获得奖励的当前价值表现，当$\gamma=0$时，智能体只关心当前的最大化收益，而不考虑未来奖励，容易陷入局部最优解；当$\gamma=1$时，智能体更多地考虑未来奖励，不对未来奖励进行折扣。

价值函数是状态动作对的函数，表示在给定状态动作对的情况下未来预期的收益有多少。一个智能体的行为可由策略 π 来定义，策略 π 是指一个状态应该采取什么样的动作。策略 π 在状态 s 时的状态价值函数定义为：从状态 s 开始，智能体按照策略 π 进行决策所获得回报的概率期望值，记为$v_\pi(s)$。

$$v_\pi\left(s\right)\doteq E_\pi\left[G_t\middle|s_t\right]=E_\pi\left[\sum\nolimits_{i=0}^{\infty}\gamma^iR_{i+t+1}\middle|s_t\right] \tag{4}$$

式中，$E_\pi[\cdot]$是通过对策略 π 进行采样得到一个期望。

同样，定义一个 Q 函数（Q-function）在给定策略 π 时，从状态 s 开始，采取某一个动作 a 得到一个期望，即可计算出它的价值函数，记为$q_\pi(s,a)$，并称q_π为策略 π 的动作价值函数。

$$q_\pi(s,a)\doteq E_\pi[G_t|s_t,a_t]=E_\pi[\textstyle\sum_{i=0}^{\infty}\gamma^iR_{i+t+1}|s_t,a_t] \tag{5}$$

对 Q 函数的动作函数进行加和，即可得到价值函数：

$$v_\pi(s)=\textstyle\sum_a\pi(a|s)\,q_\pi(s,a) \tag{6}$$

上述内容说明了马尔可夫决策过程中的预测问题，即给定一个MDP < S, A, P, R, γ > 和策略 π，计算策略 π 的状态价值函数。实际上，策略 π 往往没有或者不是最优的。因此，需要确定最优的策略π_*以及在最优策略π_*时的最优状态价值函数v_*。

$$\begin{cases}v_*(s)=\max\limits_\pi v_\pi(s)\\ \pi_*(s)=\operatorname{argmax}\limits_\pi v_\pi(s)\end{cases} \tag{7}$$

最优策略共享相同的最优动作价值函数$q_*(s,a)$。

$$q_*(s,a)=\max\limits_\pi q_\pi(s,a) \tag{8}$$

由于$q_*(s,a)$是关于状态和动作的函数，如果在状态s下采取动作a，能使Q函数最大化，则采取的动作a就是最佳动作。因此，最优策略可通过对$q_*(s,a)$进行极大化来获得：

$$\pi_*(a|s)=\begin{cases}1, & \text{if } a=\operatorname{argmax}\limits_a q_*(s,a)\\ 0, & \text{otherwise}\end{cases} \tag{9}$$

贝尔曼最优方程描述的是最优策略下各个状态的价值一定等于这个状态下最优动作的期望回报。鉴于v_*不仅是策略的价值函数，也是最优的价值函数，所以v_*和$q_*(s_t,a_t)$的值可以通过贝尔曼最优方程求解得到：

$$\begin{cases}v_*(s)=\max\limits_a E[R_{t+1}+\gamma v_*(s_{t+1})|s_t,a_t]\\ q_*(s_t,a_t)=E[R_{t+1}+\gamma\max\limits_a q_*(s_{t+1},a_{t+1})\,|s_t,a_t]\end{cases} \tag{10}$$

由于使用 Q 学习在状态量较大或连续任务中，会遇到维度灾难问题，本文在强化学习中利用价值函数近似的方法可以解决该问题。本文引入 Deep Q Network（DQN）的概念，其基于 Q_learning 算法，加入价值函数近似于神经网络，采用目标网络和经验回放的方法进行网络训练，并从历史数据中随机采样，以最小化样本之间的相关性。

在强化学习算法中，智能体的策略用神经网络表示。在此引入执行器和评价器 2 个概念。其中，执行器表示在基于策略函数的学习算法中，以状态 s 为输入，以动作 a 为输出，对神经网络进行训练。此时的神经网络不仅代表智能体的策略，也称为执行器，其参数用 u 来表示，如图 5 所示。策略函数算法虽然能够处理连续的动作空间，但会出现测量噪声大，不能收敛的情况。

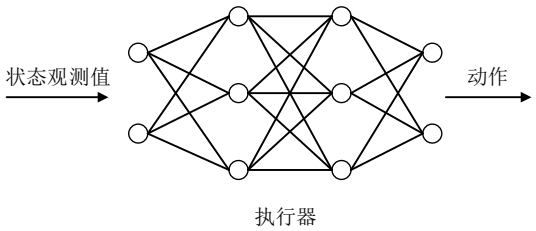

图 5　基于策略函数的学习

评价器在基于价值函数的学习算法中，以状态 s 和当前状态下的动作 a 为输入，由神经网络返回状态动作对的价值 v，此时的神经网络被称为评价器，其参数用 δ 表示，如图 6 所示。

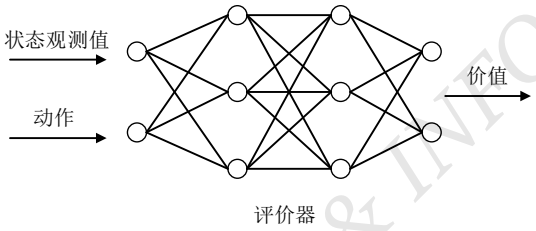

图 6　基于价值函数的学习

3.2　改进 DQN 算法

由图 6 可知，神经网络输出的是价值，并不能用来表示策略，将执行器和评价器合并成一个算法，即执行器-评价器算法，如图 7 所示。

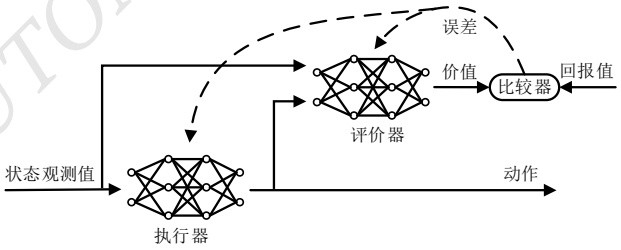

图 7　执行器-评价器算法

执行器-评价器算法中的执行器能够处理连续动作空间，评价器只需根据当前状态和执行器输出的动作来预测对应的价值，进而将此价值与环境所返回的奖励进行比较；得到的误差用来评判当前状态 s 下采取动作 a 时，环境中的奖励是否高于预测的价值。此误差也用于对评价器和执行器进行反馈，使 2 个神经网络自我更新，调整执行器输出的动作。因此，执行器-评价器算法能够处理连续的状态和动作空间，并能在环境返回奖励方差较大时加快学习速度。

3.3　DDPG 算法在控制系统中的应用

DDPG 算法是一个基于神经网络函数近似器并且可以在高维的、连续动作空间中学习策略的无模型、执行器-评价器离轨策略算法。其中，神经网络函数近似器是通过对 DDPG 算法所使用的非线性函数近似器进行修改得到的。DDPG 算法将 actor-critic 方法和 DQN 相结合。在每个批次的学习中，需要分别对评价器和执行器进行更新。

评价器更新考虑到 Q_learning 是一个无策略算法，并且使用贪婪策略$\mu(s)=\underset{a}{\operatorname{argmax}}\, q(\mathrm{s},a)$，通过最小损失函数$L$更新评价器网络：

$$\begin{cases} y_t=r_t+\gamma q(s_{t+1},\mu(s_{t+1})|\vartheta^q) \\ L=E_{s_t,a_t,r_t}[(q(s_t,a_t|\vartheta^q)-y_t)^2] \end{cases} \tag{11}$$

式中ϑ^q是评价器网络的权重。

actor-critic 方法基于 DDPG 算法，并将状态s映射到指定的动作a上，由此定义当前策略，即参数化的执行器函数$a_t=\mu(s|\vartheta^\mu)$。一个状态的回报定义为未来回报折扣的总和$R_t=\sum_{i=t}^{T}\gamma^{i-t}r(s_i,a_i)$。强化学习的目标是学习一个策略，能够从初始值分布$J=E_{r_i,s_i,a_i}[R_1]$中最大化预期回报。执行器更新也是对初始值分布$J$采用策略梯度来完成：

$$\nabla_{\vartheta^\mu}J=E_{s_t}[\nabla_a q(s,a|\vartheta^q)|_{s=s_t,a=\mu(s_t)}\nabla_{\vartheta^\mu}\mu(s|\vartheta^\mu)|_{s=s_t}] \tag{12}$$

由于大多数优化算法都假定样本是独立且恒等分布的，在线处理大批量的样本是不现实的。因此 DDPG 算法采用 DQN 算法使用的经验回放方法来解决这个问题。经验回放方法将与环境交互中采样得到

的数组(s_t,a_t,r_t,s_{t+1})储存到缓存器R中，再从缓存器R中随机选出一些样本用于网络训练，避免了评价器函数拟合的偏差，去除了样本间的相关性，使算法更容易收敛。由于缓存器R容量有限，当缓存器存满数据时，会将最先存入的数组清除。

由式(11)和式(12)可知，评价器网络$q(s,a|\vartheta^q)$在梯度频繁更新的同时，又用于执行器网络的梯度计算，使网络q在学习过程中出现不稳定的情况。为此，DDPG 算法采用soft-update的方式来更新网络，并分别建立执行器和评价器的在线网络和目标网络。其中，目标网络包含执行器目标网络$\mu'(s|\vartheta^{\mu'})$和评价器目标网络$q'(s|\vartheta^{q'})$，用于计算目标价值。

目标网络的权重ϑ通过缓慢的跟踪学习网络进行更新：$\vartheta'\leftarrow\tau\vartheta+(1-\tau)\vartheta'$。则目标网络的更新方式为

$$\begin{cases}\vartheta^{q'}\leftarrow\tau\vartheta^{q}+(1-\tau)\vartheta^{q'}\\ \vartheta^{\mu'}\leftarrow\tau\vartheta^{\mu}+(1-\tau)\vartheta^{\mu'}\end{cases}\tag{13}$$

式中τ是soft-update的速率。

DDPG 算法的整体流程如图 8 所示。

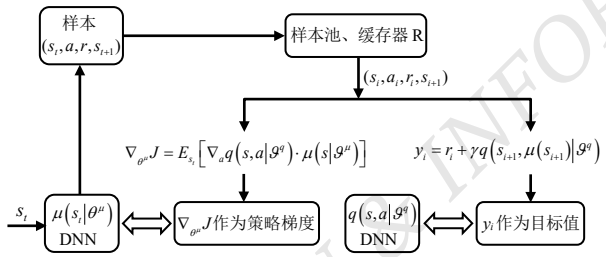

图 8　DDPG 算法流程图

4　控制系统仿真实验

为验证本系统的可行性，在 Matlab/Simulink 中搭建船舶航迹跟随控制系统，并采用常规小型船作为控制对象。为保证仿真效果更接近实际环境，在仿真环境中加入低频和高频干扰，模拟风、浪和流对船舶产生的影响；并设定操作舵的最小时间间隔为 3 ~5 s，与实际船舶航行时自动舵的调整间隔保持一致。基于 DDPG 算法的船舶航迹控制效果如图 9 所示。

由图 9 可以看出，本文提出的基于 DDPG 算法的船舶航迹跟随控制系统能够达到较好的轨迹跟踪效果，且控制效果稳定，具有良好的鲁棒性。

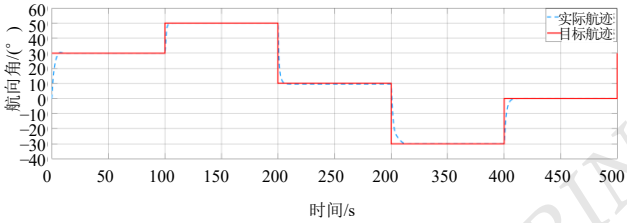

图 9　航迹控制效果

5　结论

本文针对船舶航行时，容易偏离规划的航迹路线，航迹跟踪效果较差等问题，提出一种基于 DDPG 算法的船舶航迹跟随控制系统。首先，对船舶的运动学进行分析，并给出强化学习算法的推导过程；然后，在 Matlab/Simulink 搭建船舶航迹跟随控制系统，完成船舶航迹跟踪的仿真实验。从实验结果可以看出，该控制系统稳定性好，能对外部干扰迅速做出响应。

参考文献

[1] 张显库.船舶控制系统[M].大连:大连海事大学出版社,2010.
[2] 侯志强.单片机船舶导航自动控制系统[J].舰船科学技术,2021,43(4):106-108.
[3] 韩春生,刘剑,汝福兴,等.基于 PID 算法的船舶航迹自动控制[J].自动化技术与应用,2012,31(4):9-12.
[4] 冯哲,张燕菲.基于 PID 算法的船舶航迹自动控制方法[J].舰船科学技术,2018,40(12):34-36.
[5] 祝亢,黄珍,王绪明.基于深度强化学习的智能船舶航迹跟踪控制[J].中国舰船研究,2021,16(1):105-113.
[6] 储琴,夏东青.PID 控制在船舶自动定位中的应用[J].舰船科学技术,2020,42(18):88-90.
[7] 张晓兰,王钦若,时丽丽.动力定位船舶纵向运动的反步法控制器设计[J].自动化与信息工程,2011,32(5):1-4.
[8] 刘建圻,曾碧,郑秀璋,等.基于 S3C2440 的嵌入式导航平台的设计与实现[J].自动化与信息工程,2008,29(2):1-3,13.
[9] 邹木春,曾应坚.基于机器视觉的船舶升沉检测方法[J].自动化与信息工程,2010,31(3):37-39.
[10] 潘为刚,肖海荣,周风余,等.小型船舶自动操舵控制系统的研制[J].船海工程,2009,38(1):68-70.
[11] Richard S Sutton, Andrew G Barto. Reinforcement learing: an introduction[M]. MIT Press, Bradford Books, 1998.

（下转第 32 页）