中国空间技术研究院

CHINA ACADEMY OF SPACE TECHNOLOGY

研究报告

RESEARCH REPORT

报告题目 **gym 环境下的单摆控制任务**

单　　位 **503 所（航天恒星）**

专　　业 软件工程

小组成员 李源潮、智文泓、贾景琛、明阳

指导教师 赵千川

摘　要

本文以倒立摆系统（Pendulum）为研究对象，旨在探讨从传统控制方法向基于强化学习的智能控制策略的演进过程及其性能差异。首先，构建了基于 Gym 环境的倒立摆动力学模型，对系统在随机力矩作用下的非线性动态特性进行分析，为控制策略的设计提供了验证平台。随后，采用比例—积分—微分（PID）控制算法实现人工控制，通过设定角度误差为主要反馈信号，设计并优化了控制参数。利用并行自动调参算法对 P、I、D 参数进行多组组合测试与平均误差评价，获得最优参数集，从而实现了系统在平衡位置的稳定控制。

在此基础上，本文引入深度确定性策略梯度（Deep Deterministic Policy Gradient, DDPG）算法，实现了针对连续动作空间的强化学习控制。该算法采用 Actor-Critic 结构，通过经验回放与软更新机制稳定训练过程，使智能体能够在无模型条件下自主学习最优控制策略。

关键词：倒立摆，gym 环境，PID 控制，强化学习，DDPG 算法

ABSTRACT

This paper investigates the inverted pendulum system (Pendulum) with the aim of exploring the evolution from traditional control methods to intelligent control strategies based on reinforcement learning, as well as the performance differences between them. First, a pendulum dynamics model was constructed within the Gym environment to analyze the nonlinear dynamic characteristics of the system under random torque disturbances, thereby providing a validation platform for control strategy design. Subsequently, a proportional – integral – derivative (PID) control algorithm was implemented to achieve manual control. By setting the angular error as the primary feedback signal, the control parameters were designed and optimized. A parallel automatic parameter-tuning algorithm was employed to test multiple combinations of the P, I, and D parameters and evaluate their average errors, resulting in an optimal parameter set that achieved stable control of the system at the equilibrium position.

Building upon this foundation, the paper further introduces the Deep Deterministic Policy Gradient (DDPG) algorithm to implement reinforcement learning control for continuous action spaces. Utilizing an Actor – Critic architecture with experience replay and soft update mechanisms, the algorithm stabilizes the training process and enables the agent to autonomously learn optimal control strategies under model-free conditions.

Keywords: inverted pendulum, Gym environment, PID control, reinforcement learning, DDPG algorithm

目 录

第一章 绪 论

1.1 研究工作的背景与意义

在随着人工智能技术的迅速发展，强化学习（Reinforcement Learning, RL）作为其中的重要分支，已在机器人控制、自动驾驶、智能制造等连续控制任务中展现出显著优势。传统控制方法如比例－积分－微分（PID）控制、模糊控制及线性二次调节（LQR）等，尽管在结构上简单且易于实现，但通常依赖精确的系统建模与线性化假设，难以在复杂非线性系统中保持优良性能。

倒立摆系统（Inverted Pendulum）作为经典的非线性、强耦合、非稳定控制对象，广泛用于测试和验证各种控制算法的性能。其具有典型的“稳定平衡点不稳定、非线性强、动态响应快”等特性，因此成为验证智能控制算法有效性的理想平台。在实际工程中，倒立摆模型可映射到如火箭姿态控制、机器人平衡、双轮自平衡车等多种应用场景，其研究具有重要的理论价值和工程意义。

近年来，基于深度强化学习的控制策略不断取得突破。相比传统控制方法，强化学习能够在无精确模型的条件下通过与环境交互，自主学习近似最优策略，实现高维、非线性系统的智能控制。特别是深度确定性策略梯度（Deep Deterministic Policy Gradient, DDPG）算法的提出，为连续动作空间下的控制问题提供了高效的解决方案。将强化学习方法应用于倒立摆控制，不仅可以验证智能控制算法在复杂动态系统中的可行性，还能为传统控制策略的改进提供新思路。

因此，本研究以 Gym 环境下的倒立摆系统为对象，通过对比传统 PID 控制与基于 DDPG 的智能控制策略，旨在系统探讨不同控制范式在性能、稳定性及适应性方面的差异，从而为智能控制技术在非线性动态系统中的应用提供理论与实践参考。

1.2 国内外相关研究现状

在国外研究方面，倒立摆系统的控制问题自 20 世纪中期起便成为控制理论的重要试验平台。早期研究主要集中于线性控制方法，如 LQR 和 PID 控制等。美国麻省理工学院（MIT）及斯坦福大学的研究者率先将最优控制理论与倒立摆稳定化问题相结合，为后续研究奠定了基础。进入 21 世纪后，随着机器学习和神经网络的发展，强化学习逐渐被引入控制领域。Google DeepMind 提出的深度 Q 网络（DQN）和确定性策略梯度（DDPG）算法，显著提升了智能体在连续控制任务中的学习能力。国外学者如 Lillicrap 等人（2016）利用 DDPG 算法实现了机械臂与飞行器的

精确控制，验证了深度强化学习在高维动作空间下的稳定性与鲁棒性。

国内学者在该领域也进行了大量研究与实践。部分研究聚焦于 PID 控制的优化，如通过遗传算法、粒子群算法等智能优化方法自动调整 PID 参数，以提高控制精度和响应速度。近年来，随着深度学习框架（如 TensorFlow、PyTorch）的普及，强化学习在控制系统中的应用逐步深化。清华大学、浙江大学等科研团队已将深度强化学习算法应用于移动机器人、自平衡车及机械臂控制中，取得了良好效果。然而，针对倒立摆这类具有高非线性与不确定性的系统，如何在保证训练稳定性的同时提升控制性能，仍是国内研究的重点与难点。

综合来看，当前国内外研究已从传统控制向智能控制过渡，但针对强化学习算法在连续控制任务中的收敛性、泛化性及可解释性问题仍需进一步探索。本研究通过在 Gym 环境下建立标准化的仿真模型，并系统对比 PID 与 DDPG 两种控制策略，旨在为非线性控制系统的智能化研究提供可重复、可扩展的实验框架和验证依据。

第二章 研究对象数学模型

2.1 研究对象与状态/观测

倒立摆系统采用 Gym 环境作为仿真对象。其观测为：

$$o_t=(cos\theta_t,sin\theta_t,\dot{\theta}_t) \tag{1}$$

其中θ为杆相对竖直方向的角度，$\dot{\theta}$为角速度；用$\theta_t = atan2(y, x)$从$(cos\theta, sin\theta)$复原角度用于控制与评估。动作$a_t = \tau_t$为对转轴施加的力矩，实验中力矩被限幅（例如$|\tau| \le 10$）。

2.2 连续动力学模型

在以摆长 l、质点质量 m、重力加速度 g 记，忽略粘滞阻尼时的标准化模型可写为：

$$\begin{aligned}\dot{\theta}_t &= \omega_t \\ \dot{\omega}_t &= -\frac{3g}{2l}sin(\theta+\pi)+\frac{3}{ml^2}\tau_t \\ \dot{x}_t &= f(x_t,\tau_t),\ \ x_t=[\theta_t,\omega_t]^T\end{aligned} \tag{2}$$

该模型体现了系统的强非线性与单输入（力矩）特性，动作饱和由$\tau_t \in [-\tau_{max},\tau_{max}]$给出。在 Gym 的默认设定下，即时奖励通常采用二次型惩罚：

$$r_t = -\left(\theta_t^2 + c_w\omega_t^2 + c_\tau\tau_t^2\right) \tag{3}$$

用于同时鼓励“直立”（θ）、“平稳”（ω）与“省力”（τ）。

2.3 基线：PID 控制器

在以跟踪目标$\theta^* = 0$（竖直向上）为例，误差为：

$$e(t) = \theta^* - \theta(t) \tag{4}$$

PID 力矩为：

$$\tau(t) = K_P e(t) + K_I\int_0^t e(\xi)d\xi + K_D\frac{de(t)}{dt} \tag{5}$$

并在实现中加入积分限幅、力矩饱和与离散时间更新。代码给出了人工设参与并行自动化搜索两种方式对(K_P, K_I, K_D)进行网格评估与选优，以平均角度误差为稳定性指标。

2.4 强化学习：DDPG 模型

为处理连续动作空间，引入 DDPG（确定性策略梯度，Actor–Critic 架构）：

(1) 策略网络（Actor）：$\pi_\theta(o) \in [-\tau_{max}, \tau_{max}]$直接输出力矩；训练时加入 OU 噪声进行探索。

(2) 价值网络（Critic）：$Q_\phi(o,a)$评估状态–动作价值。实现采用双 Q 网络并取最小值以抑制高估：

$$y = r + \gamma(1-d)min\left\{Q_{\phi'_1}[o', \pi_{\theta'}(o')], Q_{\phi'_2}[o', \pi_{\theta'}(o')]\right\}$$

$$\mathcal{L}_Q = \frac{1}{L}\sum_i \left\{\left[Q_{\theta_1}(o_i,a_i) - y_i\right]^2 + \left[Q_{\theta_2}(o_i,a_i) - y_i\right]^2\right\} \tag{6}$$

(3) 策略更新（确定性策略梯度）：

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_i \nabla_a Q_{\phi_1}(O_i,a)\,|_{a=\pi_\theta(o_i)} \nabla_\theta \pi_\theta(o_i) \tag{7}$$

(4) 稳定化机制：经验回放缓冲区打乱相关性、目标网络软更新：

$$\psi' \leftarrow \rho\psi' + (1-p)\psi\,, \psi \in \{\theta, \phi_1, \phi_2\} \tag{8}$$

其中$\rho \approx 0.995$。

第三章 仿真与训练环境

本文的实验平台建立于 Anaconda，操作系统为 Windows 11。通过在 Anaconda 中创建独立的虚拟环境，能够有效隔离不同项目的依赖包，避免库版本冲突，并保证实验结果的可重复性与可移植性。实验中所使用的 Python 版本为 Python3.9，核心深度学习框架为 PyTorch 2.2，CUDA 版本为 12.8 用于实现深度确定性策略梯度（DDPG）算法的神经网络模型。辅助依赖库包括 gym（包括其兼容版本 gymnasium）、numpy、matplotlib、pygame、与 imageio 等。

环境搭建过程如下：

```
#创建虚拟环境
conda create -n pendulum_rl python=3.9
#激活虚拟环境
conda activate pendulum_rl
#安装 PyTorch（CUDA12.8）
pip3 install torch torchvision --index-url
https://download.pytorch.org/whl/cu128
#安装环境中的其他依赖库
pip install gym pygame matplotlib imageio numpy gymnasium
```

上述环境配置完成后，可直接运行各阶段控制与训练脚本。

第四章 控制策略设计

4.1 PID 控制器架构设计

倒立摆的核心研究任务在于构建一种高效的控制策略，使系统在受到外界扰动或初始偏差时，仍能迅速恢复并稳定保持在竖直向上的平衡状态。该系统具有显著的非线性特征和对初始条件的高度敏感性，同时易受环境噪声和不确定因素的影响，因此，提升控制器的鲁棒性成为实现稳定控制的关键环节。

PID 控制（Proportional–Integral–Derivative）是最为经典且应用最广的反馈调节机制之一。该控制器通过实时计算系统输出与目标值之间的误差，并依据比例、积分及微分三项规律生成调节信号，从而使系统输出快速逼近并保持在期望状态。倒立摆系统伪代码如下：

算法 4-1：PID 控制器设计与实现

```
Input：目标角度θ* = 0，系统观测，最大力矩约束，采样时间，步长
Result：控制力矩(用于驱动倒立摆恢复至平衡位置)
for each time step t = 1 to T do:
    (cosθ, sinθ, ω) ← observation
    θ← atan2(sinθ, cosθ)

    error ← wrap_to_pi(θ_target - θ)

    integral ← integral + error * Δt
    integral ← clip(integral, -I_LIMIT, I_LIMIT)
    derivative ← (error - prev_error) / Δt

     τ ← K_P * error + K_I * integral + K_D * derivative
     τ ← clip(τ, -τ_max, τ_max)

    observation, reward, done ← env.step([τ])

    prev_error ← error
    if done THEN break
end
```

倒立摆系统可用状态变量$(\theta, \dot{\theta})$表示，其中θ为摆杆相对竖直方向的角度，$\dot{\theta}$为角速度。控制目标是使系统角度趋近于零，即:

$$e(t) = 0 - \theta(t) \tag{9}$$

其中$e(t)$为系统偏差。PID 控制律定义为:

$$\tau(t) = K_P e(t) + K_I \int_0^t e(\xi) d\xi + K_D \frac{de(t)}{dt} \tag{10}$$

其中K_P，K_I，K_D分别为比例、积分与微分增益系数。比例环节用于快速响应误差，积分环节用于消除稳态偏差，微分环节则对系统变化率进行预测性修正，防止超调与振荡。该控制律在程序中通过连续计算误差项、积分项与微分项实现实时力矩更新。

定义最大力矩$|\tau| \leq 10$，从观测向量(x, y, z)计算当前角度与角速度：

$$\theta = arctan2(y, x), \dot{\theta} = \omega \tag{11}$$

差项$e(t)$、积分项与差分项，代入 PID 控制公式得到控制输出：

$$\tau_t = K_P e_t + K_I \sum e_t + K_D (e_t - e_{t-1}) \tag{12}$$

并对输出力矩进行限幅处理以符合环境约束。该控制力矩作为动作输入传递给环境的 step()接口，实现闭环控制循环。

4.2 自动调参功能设计

传统 PID 控制的性能依赖于参数选择。为避免人工调参带来的主观性与低效率问题，设计了一个 并行自动调参机制。其核心思想是通过多线程方式同时运行多组 PID 参数组合，计算其在完整回合内的平均角度误差，并选取误差最小的一组作为最优参数。

该自动化调参机制显著提升了控制器的搜索效率与参数稳定性。通过统计不同参数下系统的平均角误差，可以获得最优参数组(K_P, K_I, K_D)，并在主程序中对其进行验证与可视化演示。实验证明，经过优化的 PID 控制器能够在短时间内将倒立摆从任意初始角度恢复至竖直平衡位置，并保持较低的稳态误差与振荡幅度。

参数空间(K_P, K_I, K_D)中的平均控制误差分布如下图所示：

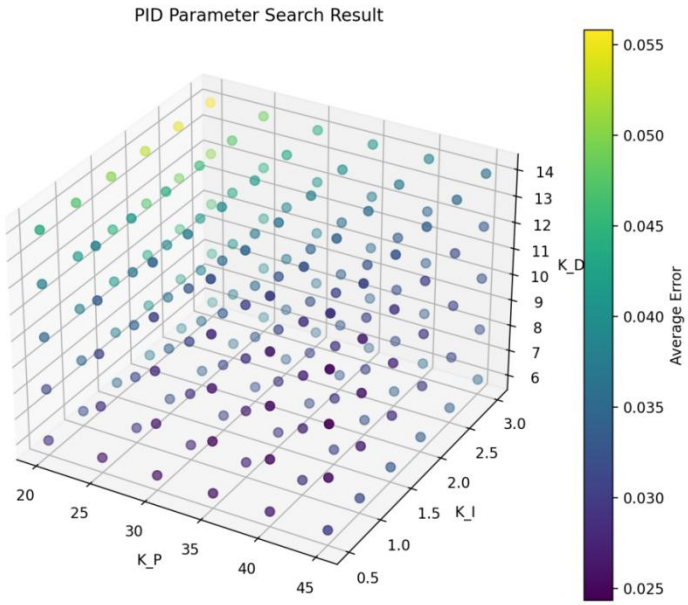

图 4-1 PID 参数搜索三维分布图

从图中可以观察到误差随参数组合的变化呈现明显的梯度分布，表明 PID 参数对控制性能具有强烈耦合关系，在较低的K_I（积分系数）与中等范围的K_P（比例系数）下，控制效果最佳，系统误差最小。下图反映了所有参数组合对应的平均误差从小到大的排序趋势，展示了调参算法的收敛特性：

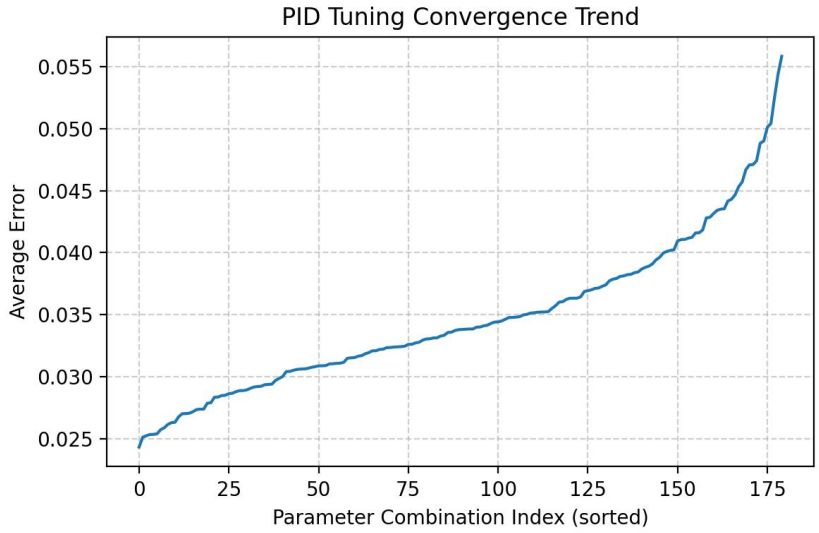

图 4-2 参数收敛趋势图

该图反映了所有参数组合对应的平均误差从小到大的排序趋势，展示了调参算法的收敛特性：随着索引增加，误差值呈平滑上升趋势，表明搜索空间中不存在明显的局部极小点，优化过程较为稳定。

第五章 训练过程

5.1 奖励函数设计

在倒立摆强化学习控制中，奖励函数的构造直接决定了智能体的行为优化方向。代码中采用了 Gym 默认奖励函数形式：

$$r_t = -\left(\theta_t^2 + 0.1\dot{\theta}_t^2 + 0.001\tau_t^2\right) \tag{13}$$

其中θ_t是摆杆相对竖直方向的偏角；$\dot{\theta}_t$是角速度；τ_t是控制力矩。

该函数的核心思想是对偏角误差、角速度以及控制输入的平方项同时进行惩罚。这样，智能体不仅学习将摆杆保持在平衡点附近，同时抑制过大的动作幅度，从而获得平稳、节能的控制策略。通过这种“负二次型奖励设计”，强化学习过程实际上等价于最小化系统能量函数的优化问题，使得控制目标更具物理可解释性。

5.2 训练算法结构

训练核心算法采用 深度确定性策略梯度（Deep Deterministic Policy Gradient，DDPG）。DDPG 是一种基于策略梯度的 Actor–Critic 架构算法，能够在连续动作空间中进行高效优化。算法结构如下图所示：

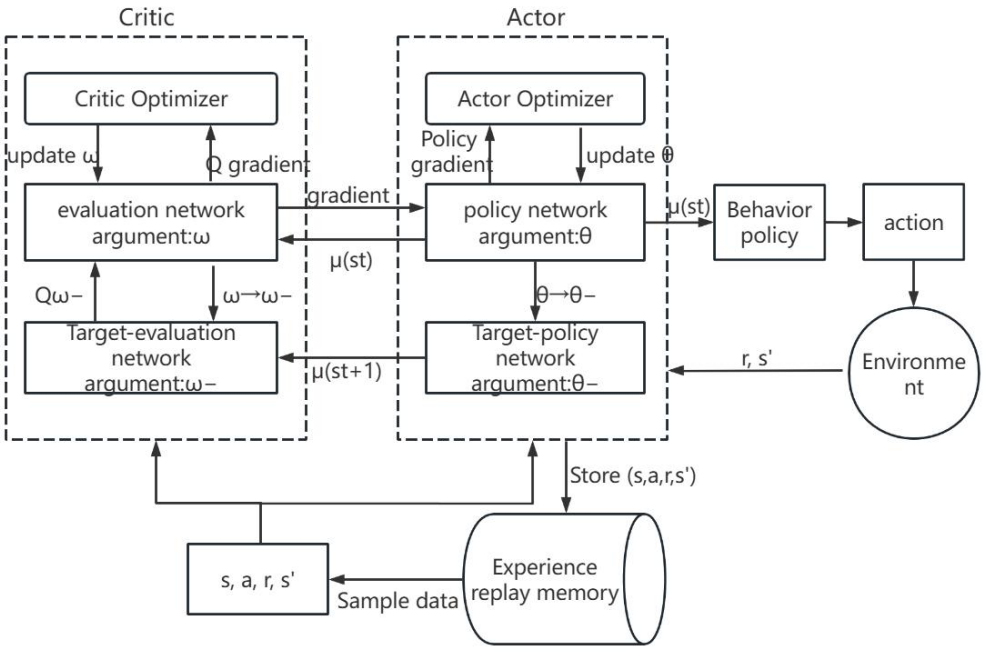

图 5-1 DDPG 算法结构图

1. Actor 网络（策略网络）

输入当前状态s_t，输出连续动作$a_t = \pi_\theta(s_t)$。网络结构为三层全连接网络[obs_dim, 256, 256, act_dim]，激活函数为 ReLU，输出层使用 Tanh 限幅以满足动作空间约束。

2. Critic 网络（价值网络）

用于评估 Actor 输出动作的状态-动作值函数$Q(s_t, a_t)$。网络输入为状态与动作的拼接向量，输出单个标量 Q 值。采用两套独立 Critic 网络（critic1, critic2）以缓解过估计问题。

3. 目标网络与软更新机制

通过设置目标网络（Target Actor / Critic），并采用 Polyak 平滑更新：

$$\theta' \leftarrow \rho\theta' + (1-\rho)\theta \tag{14}$$

其中$\rho = 0.995$，能有效降低目标值抖动，提升训练稳定性。

4. 经验回放（Replay Buffer）

存储过往的状态转移样本$(s_t, a_t, r_t, s_{t+1}, d_t)$，并随机采样批次进行训练以打破样本相关性、提高训练效率。

5. 噪声策略（探索机制）

使用 Ornstein–Uhlenbeck (OU) 噪声 进行连续空间探索：

$$n_{t+1} = n_t + \theta(\mu - n_t)\Delta t + \rho N(0,1) \tag{15}$$

该过程能生成具有时间相关性的平滑噪声，有助于物理系统中的动作连续性。

5.3 训练流程

1. 初始化环境、网络与优化器;
2. 在初期阶段（start_steps）随机探索，以填充经验缓冲区;
3. 每隔一定步数执行批量参数更新（update_every=50）;
4. Critic 更新目标:

$$y_t = r_t + \gamma(1-d_t)min\left[Q'_1(s_{t+1}, a'), Q'_2(s_{t+1}, a')\right] \tag{16}$$

并最小化均方误差损失；

5. Actor 更新目标:

$$y_t = r_t + \gamma(1-d_t)min\left[Q'_1(s_{t+1}, a'), Q'_2(s_{t+1}, a')\right] \tag{17}$$

并最小化均方误差损失；

6. 进行多次软更新，平滑同步主网络与目标网络参数；
7. 每个 Episode 结束后记录累积奖励并绘制曲线。

第六章 训练结果

6.1 训练结果展示

模型基于 DDPG（Deep Deterministic Policy Gradient）算法对连续控制环境 Pendulum-v0 进行 100 个 Episode 和 300 个 Episode 的训练。该环境的核心任务是控制倒立摆在重力作用下保持竖直向上平衡。训练过程中，模型通过与环境交互、采样状态转移(s_t, a_t, r_t, s_{t+1})，并在经验回放池中随机抽取样本进行梯度更新，实现从随机策略到最优策略的收敛。

如下图所示的 Policy Loss 曲线反映了策略网络（Actor）的学习趋势。

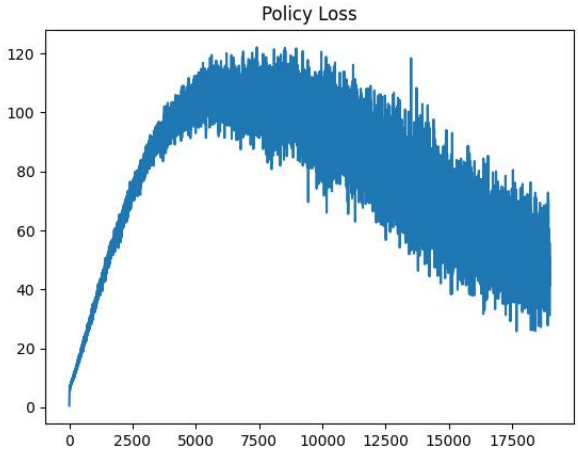

图 6-1 pi_loss 曲线图（episodes 100）

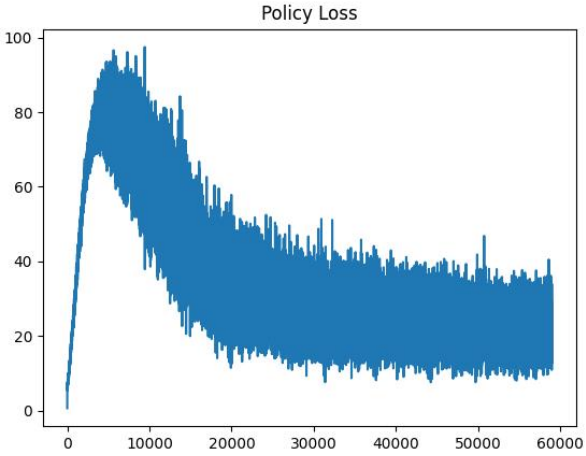

图 6-2 pi_loss 曲线图（episodes 300）

在训练早期约即 0–3000 步，损失函数迅速上升，这是由于智能体初始随机探索较多，Q 值估计尚未稳定，导致策略梯度方差较大。随后约即 3000–10000 步损失达到峰值，说明 Actor 在尝试适应 Critic 反馈时进行了激进更新。在后期阶段即 10000 步以后，损失逐渐下降并趋于平稳，说明策略网络开始收敛，输出的动作与最优控制方向一致。

这种“先上升后下降”的损失曲线符合 DDPG 的典型训练规律，即：早期策略剧烈探索 → 中期高波动学习 → 后期逐步稳定。

如下图所示，Q Loss 曲线展示了 Critic 网络对 Q 值的拟合误差变化。

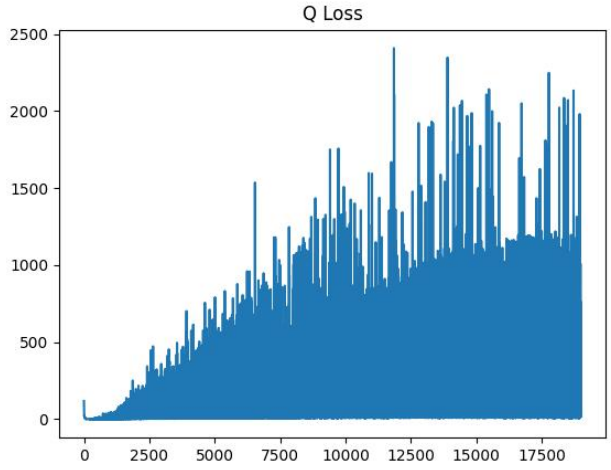

图 6-3 q_loss 曲线图（episodes 100）

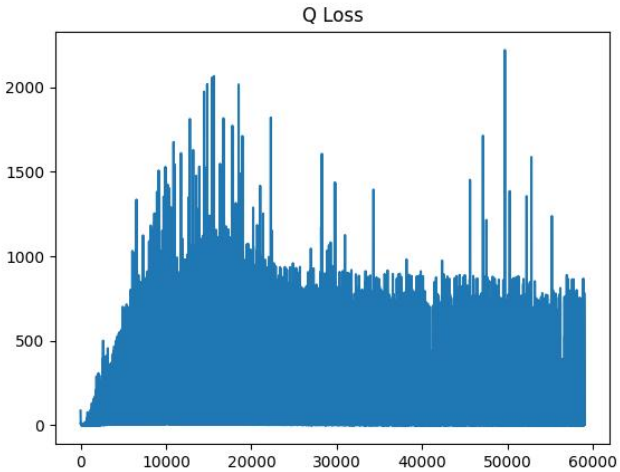

图 6-4 q_loss 曲线图（episodes 300）

可以观察到，损失在训练早期稳步上升，并在约 10000 步后出现波动。原因

主要有两点：

1. Q 网络需要在高维连续状态–动作空间中拟合复杂的值函数，早期误差上升反映出其学习难度较高。

2. 中后期波动源于 Actor 策略的持续更新，使得 Q 目标值动态变化，训练过程中存在“跟随误差”。

尽管曲线存在震荡，但整体趋势在 15000 步后趋于稳定，说明 Critic 已能稳定预测动作的长期回报值。结合算法结构可知，使用双 Q 网络（critic1, critic2）和最小值更新机制，有效抑制了过高估计问题，从而保证了策略改进的可靠性。

如下面训练回报曲线图所示，Episode Returns 曲线直接反映了智能体控制性能的提升过程。

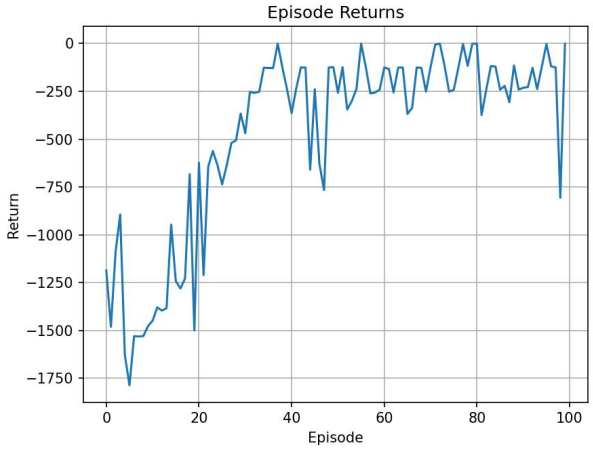

图 6-5 Episode Returns 曲线图（episodes 100）

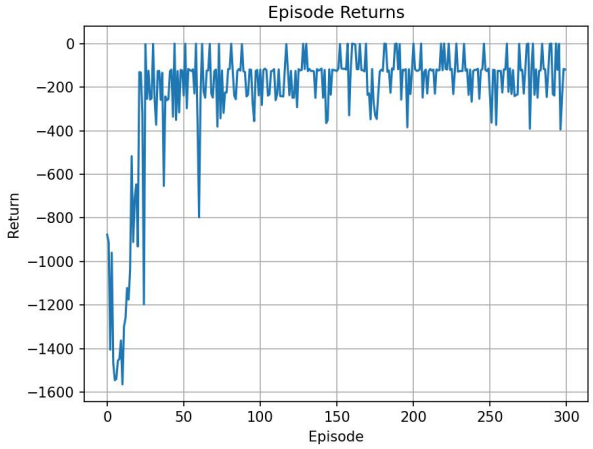

图 6-6 Episode Returns 曲线图（episodes 300）

训练初期（前 20 回合），回报值在-1500 到-1000 区间大幅波动，表明模型动

作近似随机，无法维持摆杆平衡。

随着训练的深入（20–60 回合），回报值逐渐上升至−500 以上，并在 80 回合后稳定在接近 0 的水平。此时，模型已能较稳定地将倒立摆维持在竖直状态附近，并减少不必要的能量消耗。

曲线的总体趋势说明：模型在 50 回合后进入性能收敛阶段；学到的策略具备一定的鲁棒性，能够适应不同初始状态的扰动；奖励函数的设计（惩罚角度偏差、角速度与力矩幅值）有效引导了系统学习稳定控制策略。

6.2 进训练结果总结

从整体表现来看，本次训练结果：

1. 收敛稳定性良好：策略与价值网络均在 10000 步后趋于稳定，验证了软更新（Polyak 平滑）在降低参数震荡中的有效性。

2. 探索–利用平衡适当：OU 噪声在初期提供了充分的随机性，使智能体覆盖了较大的状态空间，而在后期则逐步减弱探索幅度，提高了策略确定性。

3. 性能提升明显：最终智能体在测试阶段能够平稳地保持摆杆竖直，控制输入连续且无明显震荡。