doi:10.19306/j.cnki.2095-8110.2025.01.004

# 倒立摆模糊确定性策略梯度控制方法研究

李霖翔[1],刘开南[1],班晓军[1],冯志超[2]

(1. 哈尔滨工业大学控制理论与制导技术研究中心,哈尔滨 150001;
2. 火箭军工程大学导弹工程学院,西安 710025)

**摘　要**:倒立摆系统作为一类典型的非最小相位系统,具有显著的非线性和不稳定性特点,使其控制问题具有一定挑战性。针对传统基于深度强化学习的倒立摆控制方法中存在的神经网络可解释性不足、状态量难以收敛到期望值的问题,提出了一种基于确定性策略梯度的模糊强化学习(FDPG)控制算法。该算法将确定性策略梯度方法与 T-S 模糊模型相结合,利用 T-S 模糊模型良好的函数拟合能力,逼近 Actor-Critic 框架中的 Actor 结构,进而将控制策略用模糊规则直观地表达出来,使控制器的实际意义更加明确。同时,基于 T-S 模糊模型良好的可解释性优势,将线性二次型调节器(LQR)推导的最优控制律作为先验知识融入 T-S 模型中,保证了控制器局部稳定性。最后,通过与传统的深度确定性策略梯度(DDPG)算法以及模糊控制方法进行对比分析,验证了所提算法在倒立摆系统的控制中具有更好的控制效果与泛化能力。

**关键词**:模糊强化学习;模糊 T-S 模型;倒立摆控制;确定性策略梯度;DDPG 算法
**中图分类号**:TP273　　**文献标志码**:A　　**文章编号**:2095-8110(2025)01-0038-12

# Research on fuzzy deterministic policy gradient control method for inverted pendulum system

LI Linxiang[1], LIU Kainan[1], BAN Xiaojun[1], FENG Zhichao[2]

(1. Center for Control Theory and Guidance Technology, Harbin Institute of Technology, Harbin 150001, China;
2. School of Missile Engineering, Rocket Force University of Engineering, Xi'an 710025, China)

**Abstract**: As a typical non-minimum phase system, the inverted pendulum system exhibits significant nonlinear and unstable characteristics, making it challenging to control. In response to the problems of insufficient interpretability of neural networks and difficulty in converging state variables to expected values in traditional deep reinforcement learning-based control methods for the inverted pendulum, a fuzzy deterministic policy gradient (FDPG) control algorithm is proposed. This algorithm integrates the deterministic policy gradient method with a Takagi-Sugeno (T-S) fuzzy model, exploiting the excellent function approximation capabilities of the T-S fuzzy model to approximate the Actor structure within the Actor-Critic framework, thereby expressing control strategies intuitively through fuzzy rules and enhancing the practical significance of the controller. In addition, by exploiting the interpretability of the T-S fuzzy model, the optimal control law derived from the linear quadratic regulator (LQR) is incorporated into the T-S model as prior knowl-

**收稿日期**: 2024-09-21;**修订日期**:2024-11-28
**基金项目**:国家自然科学基金青年基金(62203461)
**作者简介**:李霖翔(2001—),男,硕士研究生,主要从事强化学习、飞行器制导与控制方面的研究。
**通信作者**:班晓军(1978—),男,教授,博士生导师,主要从事经典伺服控制理论、模糊逻辑控制理论、自适应自学习控制理论、系统参数状态估计及在运动体制导控制中的应用等方面的研究。

edge, which ensures the local stability of the controller. Finally, through comparative analysis with the traditional deep deterministic policy gradient (DDPG) algorithm and the piecewise fuzzy control method, the proposed algorithm is shown to offer superior control performance and generalization ability in controlling the inverted pendulum system.

**Key words**: Fuzzy reinforcement learning; Fuzzy T-S model; Inverted pendulum control; Deterministic policy gradient; Deep deterministic policy gradient (DDPG) algorithm

## 0　引言

倒立摆系统作为一类典型的非最小相位系统，具有显著的非线性和不稳定性特点。许多实际系统的控制与倒立摆系统的控制具有类似的特性，因此，研究倒立摆系统的控制问题对航天器控制领域、机械设计领域等有重要的参考价值[1-2]。该系统的控制过程包括起摆与稳摆两部分，这一过程中的强非线性以及不稳定性给控制器设计带来了巨大挑战，也引起了研究者的广泛关注。

截至目前，研究者们已针对该系统的起摆与稳摆过程提出了大量有效的控制方法。对于稳摆过程，有相关研究采用了基于模糊比例-积分-微分控制器（fuzzy proportion-integration-differentiation controller, Fuzzy PID)[3]的方法、基于线性二次型调节器（linear quadratic regulator, LQR）的方法[4]，以及鲁棒控制的方法[5]等。这些方法大多采用线性化的思路，在稳摆过程中表现优异，但并未考虑起摆过程中的非线性控制问题。为了解决这一问题，Astrom 等[6]提出了能量法，通过加大摆杆能量，以弥补对小车位移及速度的控制，解决了倒立摆的起摆与平衡问题。Hou 等[7]设计了一种 Bang-Bang-Adjust 控制算法，对开环控制参数进行分析，有效完成了倒立摆的起摆。顾杰等[8]针对旋转二级倒立摆的起摆与平衡问题，提出了运动规划和跟踪控制相结合的方案，设计了一个非线性前馈控制器和一个最优反馈控制器，并通过数值仿真验证了算法的有效性。然而，这些方法往往受到小车导轨长度的限制，并且控制效果难以达到最优。

随着人工智能技术的迅速发展，深度强化学习领域取得了重大突破[9-11]。Mnih 等[12]率先设计了一种基于价值的深度强化学习算法——深度 Q 网络（deep Q-network ，DQN)算法，但该算法只能处理离散的动作空间。在此之后，确定性策略梯度（deterministic policy gradient, DPG)算法[13]被提出，在 DPG 算法中，控制器的输出是确定的，不再依赖于概率分布，而是依赖于交互环境中的状态量。Lillicrap 等[14]在此基础上设计了深度确定性策略梯度（deep deterministic policy gradient, DDPG)算法，该算法将深度神经网络和 DPG 算法进行融合，构建了 Actor-Critic 框架，分别用不同的神经网络逼近 Actor 部分和 Critic 部分，已被应用于解决具有连续动作空间的问题。

在控制领域中，强化学习算法在非线性系统控制方面展现出巨大的潜力[15-16]。研究者们利用深度强化学习的奖惩机制去逼近非线性系统控制问题中的最优控制律[17-18]。近年来，已有部分研究者将深度强化学习算法应用于倒立摆的起摆与平衡问题[19-20]，通过设计代价函数对某些状态量进行优化，取得了较好的控制效果，并实现了对控制性能的优化。

然而，这些传统的深度强化学习方法将神经网络作为“黑盒”使用，在实际控制系统的应用中仍存在一些问题，如可解释性较弱、稳定性分析困难，状态量可能无法收敛到期望值。相比之下，模糊逻辑系统(fuzzy logic system, FLS)作为一种模拟人类大脑思考逻辑的方法，在描述复杂且难以准确建模的系统动态时展现出了更好的可解释性。其中，T-S 模型是一种重要的模糊系统模型，凭借其卓越的非线性逼近能力和出色的可解释性，在许多领域都有广泛应用[21]。利用模糊规则和人类语言之间固有的相似性以及它们基于经验知识的适应性设计，FLS 已成为强化学习设计中的一个重要选项[22]。Xie 等[23]针对空间机械臂的控制问题，提出了一种基于强化学习的模糊自适应滑模控制器，通过一种改进的强化学习机制对模糊逻辑规则进行调整，从而实现模糊逻辑规则的优化，以达到更好的跟踪性能。Shi 等[24]提出了一种基于模糊贝叶斯强化学习的方法，利用模糊机制提取系统的经验特征。上述方法均考虑将模糊系统作为数据的预处理部分，以提升算法效果。另外一些学者[25-26]将模糊系统直接与 Q-learning 算法结合，利用模糊系统拟合价值网络。在这些研究中，控制器设计是基于经典的基于值函数方法进行调整的，因此，仅适用于解决离

散问题。

为了解决连续动作空间的问题，并获得参数具有实际物理意义的控制器，本文将 T-S 模糊模型与文献[16]中的 DDPG 深度强化学习算法相结合，搭建了基于模糊确定性策略梯度(fuzzy deterministic policy gradient，FDPG)的强化学习算法。利用 T-S 模糊模型良好的可解释性，引入先验知识融合机制，保证控制系统的局部稳定性，提高控制性能。同时，以 Actor-Critic 框架为基础，设计了 T-S 模糊模型的参数自适应更新律与 Critic 网络的参数自适应更新律，完成了对相关控制参数的优化，以逼近非线性系统最优控制的解析解，完成对倒立摆系统的控制。

## 1 系统建模与分析

倒立摆系统由导轨、小车及摆杆组成。摆杆通过铰链连接在小车上，小车可沿导轨水平运动。通过在小车上施加力可以控制倒立摆的角度，实现从自由状态到起摆，并最终稳定控制在竖直的位置，完整结构如图 1 所示。

定义小车水平位移为 $X$，施加在小车上的推力为 $F$，选择水平向右为正方向，摆杆脱离竖直向上平衡位置的角度为 $\theta$，以竖直向上方向为基准，逆时针方向为正。小车与地面为光滑接触，摆杆角度期望值 $\theta_d = 0$，小车位移期望值 $X_d = 0$。其中，摆杆角度约束为：$-\pi \leqslant \theta \leqslant \pi$，小车位移约束为：$-5\ \mathrm{m} \leqslant X \leqslant 5\ \mathrm{m}$。扰动信号大小有固定上界。系统相关物理参数如表 1 所示。

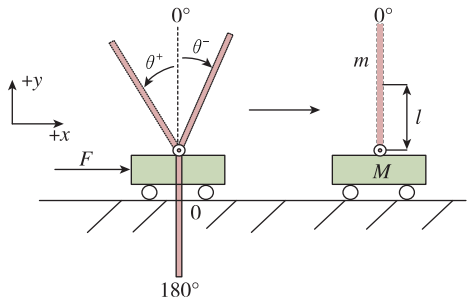



**图 1 倒立摆系统简化模型**

**Fig. 1 Simplified model of inverted pendulum system**

**表 1 系统相关物理参数**

**Tab. 1 Physical parameters of the system**

| 符号 | 物理意义 | 数值 |
|---|---|---|
| $M$ | 小车质量/kg | 0.5 |
| $m$ | 摆杆质量/kg | 0.5 |
| $g$ | 重力加速度/(m/s²) | 9.82 |
| $l$ | 杆长/m | 0.6 |
| $b$ | 摩擦系数 | 0.1 |

选取状态变量：$x_1 = X$，$x_2 = \dot{x}$，$x_3 = \theta$，$x_4 = \dot{\theta}$，参考文献[27]，系统非线性状态空间模型为

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \dfrac{-2mlx_4^2\sin\theta + 3mg\sin\theta\cos\theta - 4bx_2}{4(M+m) - 3m\cos^2\theta} \\ x_4 \\ \dfrac{-3mlx_4^2\sin\theta\cos\theta + 6(M+m)g\sin\theta - 6b\dot{x}\cos\theta}{4l(M+m) - 3ml\cos^2\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{4}{4(M+m) - 3m\cos^2\theta} \\ 0 \\ \dfrac{6\cos\theta}{4l(M+m) - 3ml\cos^2\theta} \end{bmatrix} F \tag{1}$$

## 2 基于 FDPG 的强化学习控制方法

由式(1)可以看出，倒立摆系统具有明显的非线性，在起摆控制器设计环节存在一定困难。然而，人可以很容易地将类似的倒摆机构用手“摇起来”，并将摆杆平衡在竖直向上的位置。更为重要的是，我们可以将该起摆过程用“语言规则”描述出来，形成专家经验。基于上述原因，本章提出了一种结合 T-S 模糊模型与 DPG 的强化学习算法。接下来，将从经典的 DDPG 算法与 FDPG 控制器设计两方面详细介绍这种基于模糊确定性策略的强化学习算法的具体实现。

### 2.1 DDPG 算法

DDPG 算法是一类基于 Actor-Critic 框架的深度强化学习算法，由 DQN 算法[14]和 DPG 算法[15]发展而来，是一种无模型的深度强化学习算法，用于解决连续动作控制问题。DDPG 算法中有 4 个神经网络：Critic 网络 $Q(\cdot|\omega^c)$，Target-Critic 网络 $Q'(\cdot|\omega^{c'})$，Actor 网络 $\nu(\cdot|\theta^a)$，Target-Actor 网络 $\nu'(\cdot|\theta^{a'})$，引入 Target 网络的目的是保证参数更新过程的平稳。其中，Critic 以及 Target-Critic 网络输入为状态量 $x$ 和控制量 $u$，输出为评价值，网络参数分别为 $\omega^c$ 和 $\omega^{c'}$；Actor 以及 Target-Actor 网络输入为状态量 $x$，输出为控制量 $u$，网络参数分

别为 $\theta^{a}$ 和 $\theta^{a'}$。

在训练过程中，将与环境交互过程中产生的数据依次存入经验池，通过对过往数据进行批量采样，以提高参数学习效率。假设每次从经验池中采样数据容量为 $N$，对于第 $n$ 条采样序列 $<x_k,u_k,r_k,x_{k+1}>_n$，其中 $x_k$ 表示当前时刻状态量，$r_k$ 表示当前动作的奖励值，$x_{k+1}$ 表示下一时刻状态量，$u_k$ 表示当前时刻控制量，具体实现过程包括以下 3 个核心部分。

1)更新价值函数

利用 Target-Actor 网络计算出下一时刻状态量 $x_{k+1}$ 所对应的控制量 $u_{k+1}$

$$u_{k+1}=\nu'(x_{k+1}\mid\theta'_{a}) \tag{2}$$

利用 Target-Critic 网络计算当前状态量与控制量序列 $<x_k,u_k>$ 下对应的目标价值函数 $Q_{target}$ 为

$$Q_{target}=r_k+\gamma Q'(x_{k+1},u_{k+1}\mid\omega^{c'}) \tag{3}$$

其中，$r_k$ 为奖励函数。对于 $N$ 条采样数据，求解误差 $L$ 为

$$L=\frac{1}{N}\sum_{i=1}^{N}[r_k+\gamma Q'(x_{k+1},u_{k+1}\mid\omega^{c'})-Q(x_k,u_k\mid\omega^{c})]^2 \tag{4}$$

采用梯度下降算法对该 Critic 网络参数 $\omega_c$ 进行更新

$$\omega_{new}^{c}=\omega^{c}-\alpha\times\frac{\partial L}{\partial\omega_c} \tag{5}$$

其中，$\alpha$ 为 Critic 网络参数更新的学习率；$\omega_{new}^{c}$ 为 Critic 更新后的网络参数。

2)更新策略函数

针对 Actor-Critic 的架构，期望通过更新策略网络使价值函数增大，对策略网络进行更新如下。

利用 Actor 网络计算出当前状态 $x_k$ 下输出的控制量 $u_k$

$$u_k=\nu(x_k\mid\theta^{a}) \tag{6}$$

计算价值函数对策略网络参数 $\theta^{a}$ 的梯度

$$\nabla J_{\theta^{a}}=\frac{\partial Q(x_k,u_k)}{\partial u_k}\times\frac{\partial u_k}{\partial\theta^{a}} \tag{7}$$

采用梯度上升算法对该网络进行更新

$$\theta_{new}^{a}=\theta^{a}+\beta\times\nabla J_{\theta^{a}} \tag{8}$$

其中，$\beta$ 为参数更新的学习率；$\theta_{new}^{a}$ 为 Actor 网络更新后的网络参数。

3)以软更新的形式实现目标网络的更新

引入学习率 $\rho$，将未更新的目标网络参数与对应的网络参数进行加权，获得新的目标网络参数，其中 $\rho\in(0,1)$。具体算法如下。

Target Actor 网络更新

$$\theta^{a'}=\rho\theta^{a}+(1-\rho)\theta^{a'} \tag{9}$$

Target Critic 网络更新

$$\omega^{c'}=\rho\omega^{c}+(1-\rho)\omega^{c'} \tag{10}$$

### 2.2　FDPG 控制器设计

尽管经典的 DDPG 算法已相对成熟，但仍存在可解释性不足的问题，因此，本文提出了一种基于 T-S 模型的模糊强化学习控制算法。该算法通过搭建 Actor-Critic 框架，利用 Critic 网络的梯度信息，优化 T-S 模糊控制器参数，从而逼近最优控制策略，并基于模糊系统的良好可解释性优势，在控制器中融入先验知识，以提升控制性能。接下来，将从评价网络(Critic)设计、策略网络(Actor)设计以及先验知识融合 3 个方面介绍所提出控制器的详细设计方案。

#### 2.2.1　评价网络设计

Critic 网络主要利用生成奖励或者惩罚作为自适应控制器的反馈指标，以实现对当前状态动作序列 $<x,u>$ 价值的衡量。现引入惩罚函数为

$$\psi_0(t)=(\boldsymbol{z}-\boldsymbol{z}_d)^{T}\boldsymbol{P}(\boldsymbol{z}-\boldsymbol{z}_d)+\boldsymbol{u}^{T}\boldsymbol{G}\boldsymbol{u} \tag{11}$$

其中，$\boldsymbol{z}$ 为系统状态量的某种函数形式，$\boldsymbol{z}_d$ 为 $\boldsymbol{z}$ 的期望输出向量。以倒立摆系统为例，定义 $\boldsymbol{u}$ 为控制量，$\boldsymbol{z}=[X,\sin\theta,\cos\theta]^{T}$，$\boldsymbol{z}_d=[0,0,1]^{T}$。$\boldsymbol{P}$，$\boldsymbol{G}$ 分别为状态量以及控制量的权重矩阵。为保证训练稳定性，对式(11)中代价函数进行标准化处理，将其缩放到[0,1]区间。

$$\psi(t)=1-e^{-0.5\psi_0(t)} \tag{12}$$

定义价值函数为

$$J(t)=\int_{t}^{\infty}e^{-\gamma(\tau-t)}\psi(\tau)d\tau \tag{13}$$

当该价值函数取最小时，可获得该指标下最优的控制效果，对该价值函数进行离散化近似

$$J(k)=\sum_{i=k}^{\infty}\gamma^{i-k+1}\psi_k \tag{14}$$

定义 Critic 网络形式为：$f_c(z,u\mid\delta^{c})$，其中 $\delta^{c}$ 为 Critic 网络的权重参数，$f_c(\cdot)$ 为经典的全连接网络，该网络模型理论上可以实现对价值函数的任意精度的逼近。参考 DDPG 算法中的目标网络设计，定义 $f'_c(z,u\mid\delta^{c'})$ 为 Target Critic 网络，则对当前状态动作序列对下对应的价值函数估计值为 $\hat{J}=f_c(z,u\mid\delta^{c})$。假定经验池采样大小为 $N$，对于某一组采样序列 $<x_k,\psi_k,x_{k+1},u_k>$ 而言，期望的价

值函数为

$$\hat{J}_{\text{target}}=\psi_k+\gamma f'_{\text{c}}(z_{k+1},u_{k+1}\mid\delta^{\text{c}'}) \tag{15}$$

Critic 的估计误差为

$$\zeta_{\delta^{\text{c}}}=\hat{J}_{\text{target}}-\hat{J} \tag{16}$$

对于 $N$ 条采样数据而言，误差描述为

$$\zeta_{\delta^{\text{c}}}=\frac{1}{N}\sum_{i=1}^{N}\{f_{\text{c}}(z_k,u_k\mid\delta^{\text{c}})-[\psi_k+\gamma f'_{\text{c}}(z_{k+1},u_{k+1}\mid\delta^{\text{c}'})]\}^2 \tag{17}$$

根据梯度下降算法设计 Critic 网络的更新方式为

$$\delta^{\text{c}}_{\text{new}}=\delta^{\text{c}}-lr_{\text{c}}\times\frac{\partial\zeta_{\delta^{\text{c}}}}{\partial\delta^{\text{c}}} \tag{18}$$

其中，$lr_{\text{c}}$ 为价值网络的参数学习率；$\delta^{\text{c}}$ 为价值网络参数；$\delta^{\text{c}}_{\text{new}}$ 为更新后的价值网络参数。同样参考式(10)完成对目标网络 $f'_{\text{c}}(z,u\mid\delta^{\text{c}'})$ 的软更新

$$\delta^{\text{c}'}=\rho\delta^{\text{c}}+(1-\rho)\delta^{\text{c}'} \tag{19}$$

2.2.2 执行网络设计

经典的强化学习算法往往使用神经网络模型作为非线性逼近器近似策略函数，为增强其可解释性，本节利用 T-S 模糊模型作为执行器，通过模糊规则所对应的条件隶属度表示其对应的控制律适用范围，从而在保证执行器逼近最优的同时，使执行器具有可解释性，进而利用人工先验知识对执行器的迭代过程和结果进行优化。在 T-S 模糊模型中，模糊规则所对应的前提条件称为前件，可以通过各个单一变量的隶属度函数相乘获得，其对应的控制律则被称为后件。图 2 所示为本文所使用的单维度的高斯型隶属度函数示例。

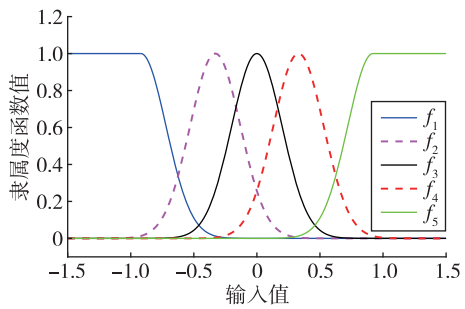



**图 2 单一维度的隶属度函数**

**Fig. 2 Membership function of a single dimension**

在此基础上，确定前件对应的后件部分，即可建立 T-S 模糊模型。模糊 T-S 的数学解析表达式一般可以由式(20)表示

$$\Gamma(x)=\frac{\sum_{l=1}^{M}\bar{y}_l\left[\prod_{i=1}^{a}c_i^l\exp\left(-\frac{1}{2}\left(\frac{\tilde{x}_i-\bar{x}_i^l}{\sigma_i^l}\right)^2\right)\right]}{\sum_{l=1}^{M}\left[\prod_{i=1}^{a}c_i^l\exp\left(-\frac{1}{2}\left(\frac{\tilde{x}_i-\bar{x}_i^l}{\sigma_i^l}\right)^2\right)\right]} \tag{20}$$

其中，$M$ 为规则数；$a$ 为输入量维数；$\tilde{x}_i$ 为输入量 $x$ 中的第 $i$ 维分量的归一化形式；$c_i^l,\bar{x}_i^l,\sigma_i^l$ 为规则 $l$ 的第 $i$ 维分量的前件参数；$\bar{y}_l$ 为后件的表达形式。前件和后件参数均可以通过自适应或学习方法确定，也可以通过经验事先确定。在本文中，为提高拟合精度，针对目标 4 阶模型，对每个维度的输入变量选择 5 个隶属度函数，对应的隶属度函数中心值 $\bar{x}_i^l$ 为 $[-1,-0.5,0,0.5,1]$，共 625 条规则，此外，设 $c_i^l=1,\sigma_i^l=0.25$，后件使用线性模型，即 $\bar{y}_l=\eta_l x$，其中 $\eta_l$ 为线性控制律。在此基础上，设 $h_l(x)$ 为第 $l$ 条规则下对应的归一化后的条件隶属度，即

$$h_l(x)=\frac{\left[\prod_{i=1}^{a}c_i^l\exp\left(-\frac{1}{2}\left(\frac{\tilde{x}_i-\bar{x}_i^l}{\sigma_i^l}\right)^2\right)\right]}{\sum_{l=1}^{M}\left[\prod_{i=1}^{a}c_i^l\exp\left(-\frac{1}{2}\left(\frac{\tilde{x}_i-\bar{x}_i^l}{\sigma_i^l}\right)^2\right)\right]} \tag{21}$$

因此，在估计过程中，对于线性控制律，式(20)可以表示为

$$\Gamma(x)=\sum_{l}^{M}h_l(x)\hat{\eta}_l x \tag{22}$$

其中，$\Gamma(x)$ 为执行器输出，即控制量 $u_k$；$\hat{\eta}_l$ 为第 $l$ 条规则的后件参数估计。在 $h_l(x)$ 参数固定的情况下，根据输入变量 $x$，获取参数 $\hat{\eta}_l$ 即可完成对策略函数的拟合。

为保证 T-S 模型的训练效率，采用小批量梯度下降算法，假设采样大小为 $N$，在对成功价值函数进行拟合后，评价值 $\hat{J}=f_{\text{c}}(z,u\mid\omega^{\text{c}})$ 对 T-S 系统中参数 $\hat{\eta}_l$ 的平均梯度为

$$I_{\hat{\eta}_l}=\frac{1}{N}\sum_{k=1}^{N}\frac{\partial\hat{f}_{\text{c}}(z_k,u_k\mid\omega^{\text{c}})}{\partial u_k}\times\frac{\partial u_k}{\partial\hat{\eta}_l} \tag{23}$$

对于神经网络，式(23)的后半部分需要复杂的迭代求解。而在 T-S 模糊模型中，根据式(22)，求解较为简便。将式(22)代入式(23)可得

$$I_{\hat{\eta}_l}=\frac{1}{N}\sum_{k=1}^{N}\frac{\partial\hat{f}_{\text{c}}(z_k,u_k\mid\omega^{\text{c}})}{\partial u_k}h_l(x_k)x_k \tag{24}$$

则 T-S 模糊系统中参数 $\eta$ 的更新方式为

$$\hat{\eta}_l^{\text{new}}=\hat{\eta}_l-lr_{\text{a}}\times I_{\hat{\theta}_l} \tag{25}$$

其中，$lr_{\text{a}}$ 为 T-S 模型参数的学习率；$\hat{\eta}_l^{\text{new}}$ 为更新后的 T-S 模型参数。

2.2.3　先验知识融合

对于经典的最优控制而言，若目标为线性模型，则有一套成熟的理论可以针对目标价值函数计算出最优控制律。由于平衡位置附近可以进行线性化处理，可依据线性模型求取最优控制的解析解。故利用模糊 T-S 模型的可解释性优势，在初始化过程中，将最优控制律作为先验知识融入 T-S 模型，但不让其参与后续的参数迭代，以保证该控制系统的局部稳定性，完整的设计流程如下。

在平衡点 $\boldsymbol{x}_0=[0,0,0,0]^{\mathrm{T}}$ 附近，对式(1)进行线性化处理，定义

$$\boldsymbol{A}=\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & -\dfrac{4b}{4(M+m)-3m} & \dfrac{3mg}{4(M+m)-3m} & 0\\ 0 & 0 & 0 & 1\\ 0 & -\dfrac{6b}{4l(M+m)-3ml} & \dfrac{6(M+m)g}{4l(M+m)-3ml} & 0\end{bmatrix},\boldsymbol{B}=\begin{bmatrix}0\\ \dfrac{4}{4(M+m)-3m}\\ 0\\ \dfrac{6}{4l(M+m)-3ml}\end{bmatrix}$$

则系统平衡点附近的状态空间方程可近似为

$$\dot{\boldsymbol{x}}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{F} \tag{26}$$

设最优控制指标为

$$J=\int_t^{\infty}[\boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{Q}\boldsymbol{x}(t)+\boldsymbol{u}^{\mathrm{T}}(t)\boldsymbol{R}\boldsymbol{u}(t)]\mathrm{d}t \tag{27}$$

其中，$\boldsymbol{Q}$ 为半正定矩阵，$\boldsymbol{R}$ 为正定矩阵。可以依据最优控制理论推导最优状态反馈控制律 $\boldsymbol{K}$，并将在稳定点附近的 T-S 模型参数固定。对于指标式(27)选择的参数矩阵为：$\boldsymbol{Q}=10\boldsymbol{I}_4$，$\boldsymbol{R}=10$，其中 $\boldsymbol{I}_4$ 为单位矩阵。依据最优控制理论，对应黎卡提(Riccati)方程为

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{P}+\boldsymbol{P}\boldsymbol{A}+\boldsymbol{Q}-\boldsymbol{P}\boldsymbol{B}(\boldsymbol{R}^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P})=0 \tag{28}$$

其对应的状态反馈矩阵及控制量解分别为

$$\boldsymbol{K}=\boldsymbol{R}^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P} \tag{29}$$

$$\boldsymbol{u}=-\boldsymbol{K}\boldsymbol{x} \tag{30}$$

将模型参数(23)代入式(25)和式(26)中，解得的状态负反馈控制律系数如下

$$\boldsymbol{K}=[-0.999\ 9,-2.182\ 4,26.152\ 7,4.579\ 7] \tag{31}$$

本文中，将 LQR 最优控制律在初始化阶段固定在 T-S 模糊模型中，考虑到模型线性化时只对角度 $\theta$ 作要求，定义满足 $\theta$ 隶属度函数中心为 0，而状态变量隶属度函数中心为[−0.5，0，0.5]，所对应的规则为 $l'$。这些规则所对应的后件信息为 $\bar{y}_{l'}$，将这些规则对应的后件参数 $\eta_{l'}$ 固定为上述推导的 LQR 最优控制律，即令

$$\begin{aligned}\bar{y}_{l'}&=\eta_{l'}x=-Kx\\&=0.999\ 9X+2.182\ 4\dot{X}-26.152\ 7\theta-4.579\ 7\dot{\theta}\end{aligned} \tag{32}$$

## 2.3　算法流程介绍

依据 2.2 节算法设计的内容，完整的算法结构如图 3 所示。

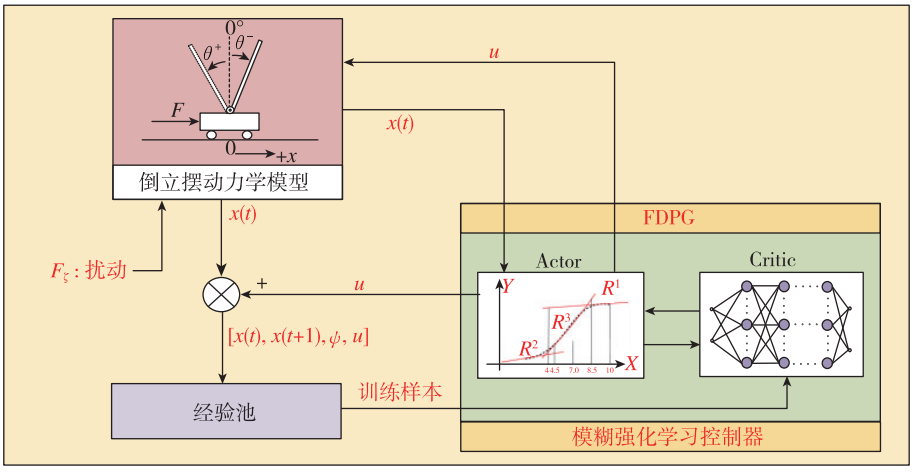



**图 3　模糊强化学习训练框架**

**Fig. 3　Fuzzy reinforcement learning training framework**

图 3 中，FDPG 模块为本文提出的算法，Actor 为 T-S 模糊模型。Actor 输出作为倒立摆模型的输入，二者交互产生经验数据并存入经验池中。在 Actor 与环境交互的过程中，算法不断从经验池中采样数据，这些数据分别输入到 Actor 与 Critic。其中，Critic 网络用于计算经验数据所对应的 Q 值，利用 2.2.2 节中推导的 T-S 模糊模型的梯度算法，计算出 Q 值对 T-S 模型参数的梯度，从而使得参数沿着 Q 值增大的方向更新。在与环境交互过程中，T-S 模型参数也在不断进行梯度更新，从而逼近最优的 T-S 控制器。

完整算法流程介绍如下。

步骤 1　根据式(19)预训练 Critic 网络，保存 Critic 网络参数。

步骤 2　由已知的先验知识，根据式(32)预先固定模糊 T-S 模型部分参数，其余参数在[−1,1]之间随机取值。

步骤 3　初始化环境状态，获取模型初始状态 $x_{\text{reset}}$。

步骤 4　依据当前状态量，利用当前的 T-S 模型输出控制量 $u_k=\sum_{l}^{M}h_l(x)\hat{\theta}_l x_k$。

步骤 5　执行当前的控制量 $u_k=\sum_{l}^{M}h_l(x)\hat{\theta}_l x_k$，到达系统的下一状态 $x_k \xrightarrow{u_k} x_{k+1}$。

步骤 6　将当前与环境交互的数据存放到经验池：$R \leftarrow R\langle x_k,u_k,\psi_k,x_{k+1}\rangle$。

步骤 7　从经验池中进行小批量采样($n$ 条数据)：$R \xrightarrow{\text{采样}} n\times R\langle x_k,u_k,\psi_k,x_{k+1}\rangle$，并依据式(24)计算 T-S 模糊模型的参数更新梯度。

步骤 8　根据式(25)更新 T-S 模糊模型参数。

步骤 9　重复执行步骤 4～8 直至达到每回合最大时间步长。

步骤 10　重复执行步骤 1～9 直至达到最大训练回合数。

## 3　仿真验证

T-S 模型内部结构参数已在 2.2.2 节中给出，表 2 所示为算法训练中需要的超参数，其中归一化均值与方差均由人为经验设计，为保证训练稳定，网络学习率不宜选择过大。

**表 2　FDPG 算法超参数选择**

**Tab. 2　FDPG algorithm hyperparameter selection**

| 相关参数 | 数值大小 | 相关参数 | 数值大小 |
|---|---|---|---|
| 归一化均值 $\mu$ | [0,0,0,0] | 状态维度 | 4 |
| 归一化方差 $\sigma$ | [4,2,π,15] | 回合步数 | 500 |
| 经验池容量 | 100 000 | 折扣因子 $\gamma$ | 0.98 |
| 采样大小 | 64 | Critic 学习率 | $10^{-4}$ |
| 训练回合数 | 1 200 | Actor 学习率 | $10^{-3}$ |

本实验的仿真环境为在 Pygame 框架下搭建的倒立摆系统模型，在 Pytorch 框架下对所提出算法进行设计和训练。每回合开始时，摆杆初始角度服从均值为 π，方差为 0.1 的正态分布，即 $\theta_0 \sim N(\mu=\pi,\sigma^2=0.1)$，选取式(11)中状态量为 $\boldsymbol{z}=[X,\sin\theta,\cos\theta]^{\mathrm{T}}$，其中期望值为 $\boldsymbol{z}=[0,0,1]^{\mathrm{T}}$。由于起摆过程中，需要较大力矩，故在该过程中不对能量消耗进行限制，故令式(11)中的 $\boldsymbol{G}=0$。定义式(11)中矩阵 $\boldsymbol{P}=\begin{bmatrix}1 & l & 0\\ l & l^2 & 0\\ 0 & 0 & l^2\end{bmatrix}$，定义 FDPG 算法中奖励函数为惩罚函数 $\psi(t)$ 的相反数，即 $Reward=-\psi(t)=-(1-\mathrm{e}^{-0.5\psi_0(t)})$。在 DDPG 算法中使用与 FDPG 算法中相同的 Critic 网络逼近代价函数，DDPG 算法中的 Actor 网络结构如表 3 所示。

**表 3　DDPG 算法中 Actor 网络模型**

**Tab. 3　Actor network model in DDPG algorithm**

| 层名称 | 层类型 | 输入、输出维度 | 激活函数 |
|---|---|---|---|
| Input | Linear Layer | (4,64) | Relu |
| Hidden1 | Linear Layer | (64,128) | Relu |
| Hidden2 | Linear Layer | (128,128) | Relu |
| Output | Linear Layer | (128,1) | Tanh |

由表 3 可以计算出该神经网络的参数量为 25 281。为了验证所提算法的性能，将从算法可行性、抗扰性及泛化性三方面进行实验评估，并与经典深度强化学习算法 DDPG，以及一种传统的模糊控制器(后文用 FUZZY 指代)进行对比分析，其中模糊规则设计参考文献[27]。

### 3.1　算法可行性验证

在训练过程中，累计奖励函数变化情况如图 4 所示。可以发现，0～400 回合之间，两种控制方法

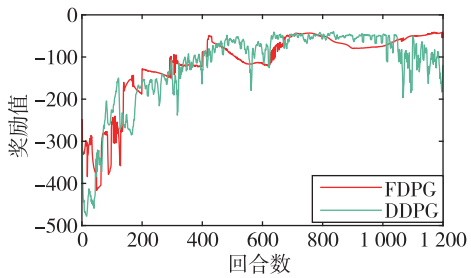



**图 4　训练过程中的奖励函数变化**

**Fig. 4　The change of reward function during training**

对应的奖励函数均呈现明显上升趋势，400 回合后都已达到较大值。二者在训练速度上基本一致，尽管二者在后续阶段均有波动，但基本可以认为已经达到收敛状态。可以发现，在训练后期，DDPG 算法的振荡较大，出现了奖励值下降的情况。

图 5 和图 6 所示分别为控制器输出曲线以及系统的状态量响应曲线，仿真过程中的相关性能指标如表4所示。图 6(a)和(b)分别为小车位置和速度响应。结合表 4 可以看出，DDPG 算法虽然保证了小车位移曲线达到收敛，但是并未收敛到 0 处。而基于模糊控制器的方法能收敛到 0，但是具有较大的超调。相反，本文提出的模糊强化学习算法，在保证收敛的同时，还可实现收敛到最优值。图 6(c)和(d)分别表示摆杆角度变化曲线和摆杆角速度变化曲线，可以发现，FDPG 控制器相较于 DDPG 控制器以及 FUZZY 控制器，可更快收敛到平衡位置。

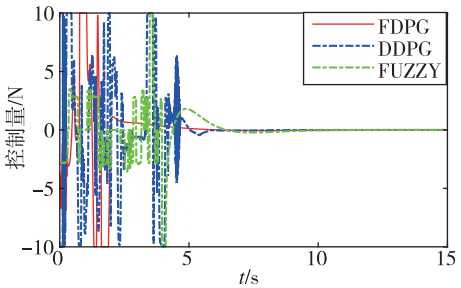
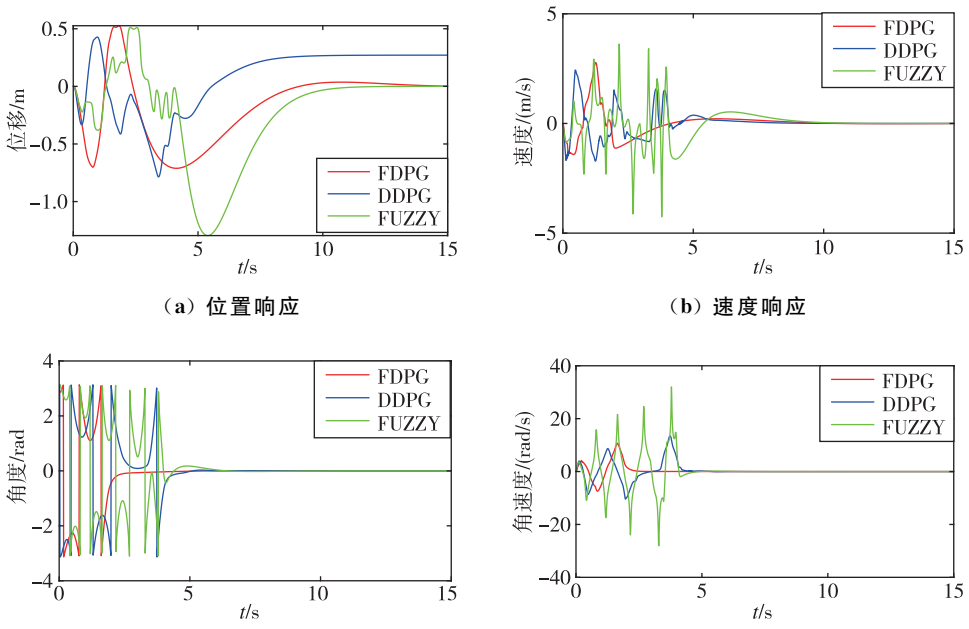



**图 5　倒立摆系统仿真的控制输出**

**Fig. 5　Control output of inverted pendulum system simulation**




（a）位置响应　　（b）速度响应

（c）角度响应　　（d）角速度响应

**图 6　倒立摆系统仿真结果**

**Fig. 6　Simulation results of inverted pendulum system**

**表 4　FDPG 算法控制性能测试**

**Tab. 4　Control performance test of the FDPG algorithm**

| 控制器 | 可训练参数数量 | 位置误差/m | 角度误差/(°) | 调节时间/s（±2%） | 耗能/J |
|---|---|---|---|---|---|
| FUZZY | — | $-3.2\times10^{-5}$ | $2.5\times10^{-5}$ | 5.25 | 12.730 |
| DDPG | 25 281 | $2.7\times10^{-5}$ | $3.9\times10^{-5}$ | 4.57 | 18.271 |
| FDPG | 2 500 | $-6.2\times10^{-4}$ | $1.9\times10^{-5}$ | 2.28 | 11.512 |

### 3.2 引入扰动后控制性能验证

与 3.1 节仿真条件保持一致，为验证控制器的抗扰动能力，对被控对象施加一个干扰信号 $d$，干扰信号为幅值 0.2 N，周期 5 s 的正弦信号，即：$d=0.2\sin\left(\frac{2\pi}{T}t\right)$。

其中，图 7 所示为控制器的输出曲线，表 5 所示为添加扰动后的仿真性能指标。结合表 5 可以看出，在起摆过程中，FDPG 控制器消耗的能量更少，在系统稳定后三者输出基本一致。

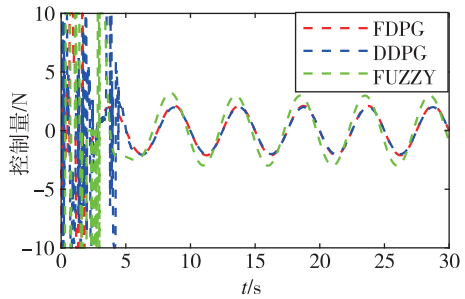



图 7 引入扰动信号后的控制输出

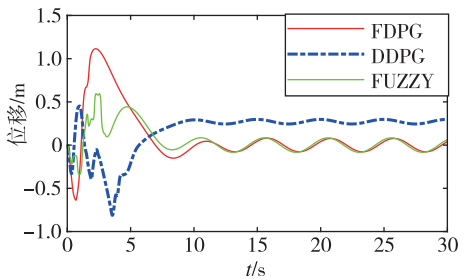
Fig. 7 Control output after the introduction of disturbance signal

表 5 添加扰动信号后控制性能测试

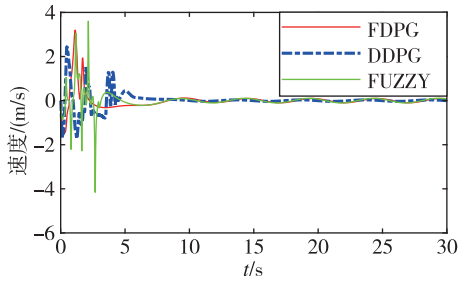
Tab. 5 Control performance testing with disturbed signals

| 控制器 | 位置误差/m | 角度误差/(°) | 调节时间/s | 耗能/J |
|---|---|---|---|---|
| FUZZY | 0.03 | 0.004 | 3.59 | 18.71 |
| DDPG | 0.30 | 0.004 | 5.26 | 19.12 |
| FDPG | 0.03 | 0.004 | 2.23 | 17.72 |

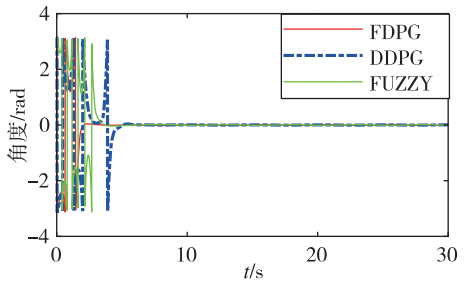
图 8 所示为系统状态量的响应曲线，其中(a)和(b)分别为小车的位移及速度曲线，两者均能达到稳定状态。结合图 7 的控制量曲线可以看出，FUZZY 控制器在起摆阶段控制量的方向切换频率较高，有效减小了位移超调。但是图 8(d)表明，控制量方向的高频切换导致了角速率的大幅振荡。同时，FDPG 控制器下的小车位移曲线相较于 DDPG 控制器收敛较慢，原因在于小车为了保证位置收敛到 0，以及损耗较小能量，导致小车位移收敛较慢。从图 8(b)可以看出，在前期，FDPG 控制器的振荡程度略小于 DDPG 和 FUZZY 控制器。图 8(c)和(d)分别为摆杆的角度曲线及速度曲线，可以发现，增加干扰信号之后，3 种框架算法下摆杆都能达到平衡点附近，并保持高性能稳摆。但 FDPG 控制器能更快地将摆杆平衡在竖直位置，响应速度更快，并且在起摆过程中其振荡情况略小于 DDPG 控制器和 FUZZY 控制器。



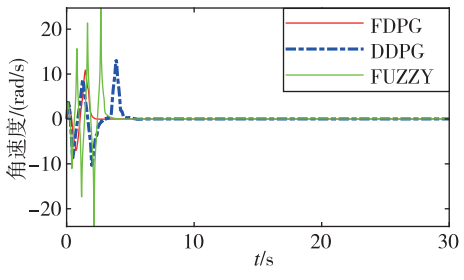

(a) 位移响应




(b) 速度响应




(c) 角度响应




(d) 角速度响应

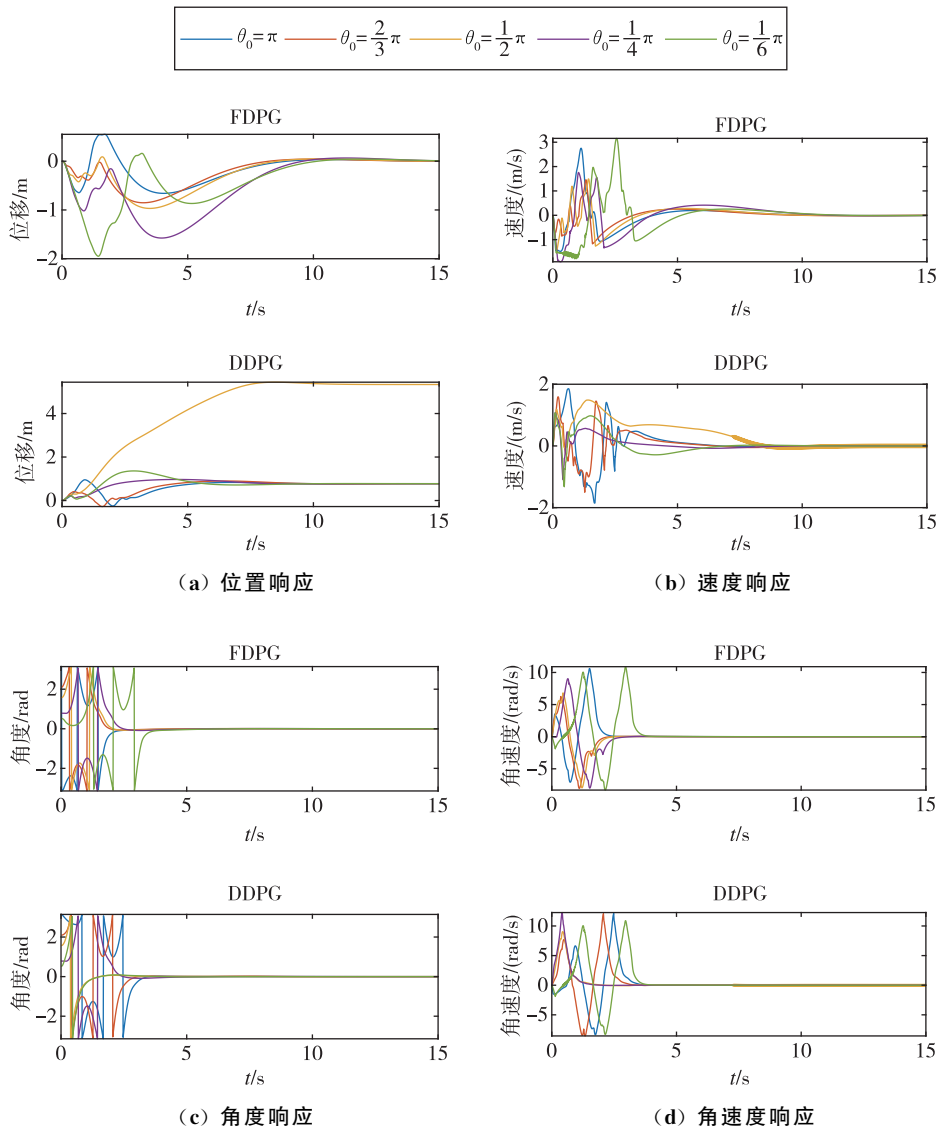
图 8 抗扰性能验证仿真结果

Fig. 8 Simulation results of disturbance-resistant function verification

### 3.3 泛化性能验证

引入先验知识的 FDPG 控制器，在保留基于数据训练的前提下，同时保证了控制系统在平衡点附近的局部稳定性，一定程度上提高了该控制器的泛

化能力。为验证 FDPG 控制器的泛化性能，将 DDPG 算法与本文提出的 FDPG 算法进行对比，改变摆杆释放的初始位置，选取初始角度分别为 $\theta_0=\pi,\frac{2}{3}\pi,\frac{1}{2}\pi,\frac{1}{4}\pi,\frac{1}{6}\pi$，其对应的状态响应如图 9 所示。尽管在参数训练时，初始状态均是在 $\theta_0=\pi$ 附近，但是训练好的 FDPG 控制器在其他初始状态下同样也能保证控制性能，各个状态变量均在 10 s 内收敛至期望值，具有较好的控制效果。相反，从图 9(a)可以看出，在 $\theta_0=\frac{1}{2}\pi$ 的场景下，DDPG 算法虽然也达到稳态，但是位移曲线最终收敛到了 5.3 m 附近，具有较大的稳态误差。




（a）位置响应　　（b）速度响应

（c）角度响应　　（d）角速度响应

**图 9　泛化性能验证仿真结果**

**Fig. 9　Simulation results of generalized performance verification**

## 4　结论

本文针对传统 DDPG 算法存在的可解释性不足、难以结合人工先验知识的问题，提出了一种 FDPG 控制方法。通过理论研究和实验结果得出结论如下：

1）对于倒立摆系统而言，本文提出的 FDPG 模糊强化学习方法能够融入已有的先验知识，在控制器学习过程中具有更好的收敛特性，奖励曲线相比 DDPG 算法更加稳定。

2）仿真实验表明，本文所述方法解决了 DDPG 控制器下某些状态量难以收敛到期望值的问题，同

时，相较于经典的模糊控制方法无法保证性能指标的局限性，FDPG 控制器利用奖励函数机制提升了控制性能，避免了系统状态量的大幅振荡。此外，本文在 T-S 模糊模型的后件部分使用了线性模型，并在局部应用了线性最优控制律。因此，本文所提方法有利于稳定性分析，并保证了平衡点附近的局部稳定性，具有良好的泛化性能。

3)在该方法中，控制器的参数已具有实际的物理意义。在这种情况下，本文以先验知识的形式保证了系统的局部稳定性。拟在后续的工作中，基于模糊 T-S 模型确定的结构形式，完成对该控制方法的全局稳定性分析与证明。

## 参考文献


[1] RONQUILLO-LOMELI G, RÍOS-MORENO G J, GÓMEZ-ESPINOSA A, et al. Nonlinear identification of inverted pendulum system using Volterra polynomials [J]. Journal of Structural Mechanics, 2016, 44(1-2): 5-15.

[2] LI Z, ZHANG Y. Robust adaptive motion force control for wheeled inverted pendulums[J]. Automatica, 2010, 46(8): 1346-1353.

[3] ABUT T, SOYGUDER S. Two-loop controller design and implementations for an inverted pendulum system with optimal self-adaptive fuzzy proportional integral derivative control[J]. Transactions of the Institute of Measurement and Control, 2022, 44(2): 468-483.

[4] 王仲民，孙建军，何永利，等. 基于遗传算法的倒立摆 LQR 控制系统权值优化[C]// 2006 中国控制与决策学术年会. 天津：IEEE，2006：432.
WANG Zhongmin, SUN Jianjun, HE Yongli, et al. Weight optimization of LQR control system for inverted pendulum based on genetic algorithm[C]// Proceedings of 2006 Chinese Control and Decision Conference. Tianjin: IEEE, 2006: 432(in Chinese).

[5] DAS S K, PAUL K K. Robust compensation of a Cart-Inverted Pendulum system using a periodic controller: experimental results[J]. Automatica, 2011, 47(11): 2543-2547.

[6] ÅSTRÖM K J, FURUTA K. Swinging up a pendulum by energy control[J]. Automatica, 2000, 36(2): 287-295.

[7] HOU X, YU H, CHEN C. The bang-bang-adjust control algorithm and simulation during the swing-up process of circular rail inverted pendulum[C]// Proceedings of 2013 25th Chinese Control and Decision Conference. Guiyang: IEEE, 2013: 373-378.

[8] 顾杰，忻欣，李玥，等. 基于轨迹规划和跟踪的旋转二级倒立摆的摆起控制[C]// 2022 中国自动化大会. 厦门：IEEE，2022：6.
GU Jie, XIN Xin, LI Yue, et al. Swing control of a rotating two-stage inverted pendulum based on trajectory planning and tracking[C]// Proceedings of 2022 China Automation Congress. Xiamen: IEEE, 2022: 6(in Chinese).

[9] VRABIE D, PASTRAVANU O, ABU-KHALAF M, et al. Adaptive optimal control for continuous-time linear systems based on policy iteration[J]. Automatica, 2009, 45(2): 477-484.

[10] LEWIS F L, ZHANG H, HENGSTER-MOVRIC K, et al. Cooperative control of multi-agent systems: optimal and adaptive design approaches[M]. Berlin: Springer Science Business Media, 2013.

[11] XU X, HUANG Z, GRAVES D, et al. A clustering-based graph Laplacian framework for value function approximation in reinforcement learning[J]. IEEE Transactions on Cybernetics, 2014, 44(12): 2613-2625.

[12] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing Atari with deep reinforcement learning[J]. Computer Science, 2013, 21: 351-362.

[13] SILVER D, LEVER G, HEESS N, et al. Deterministic policy gradient algorithms[C]// Proceedings of 31st International Conference on Machine Learning. Beijing: IMLS, 2014: 387-395.

[14] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv: 1509.02971, 2015.

[15] ZHAO Z, HE W, MU C, et al. Reinforcement learning control for a 2-DOF helicopter with state constraints: theory and experiments[J]. IEEE Transactions on Automation Science and Engineering, 2022, 21(1): 157-167.

[16] 王金强，苏日新，刘莉，等. Q-learning 强化学习协同拦截制导律[J]. 导航定位与授时，2022，9(5)：84-90.
WANG Jinqiang, SU Rixin, LIU Li, et al. Q-learning collaborative interception guidance law for reinforcement learning[J]. Navigation Positioning and Timing, 2022, 9(5): 84-90(in Chinese).

[17] GRAVELL B, ESFAHANI P M, SUMMERS T. Learning optimal controllers for linear systems with multiplicative noise via policy gradient[J]. IEEE Transactions on Automatic Control, 2020, 66(11): 5283-5298.

[18] HAN M, TIAN Y, ZHANG L, et al. Reinforcement

learning control of constrained dynamic systems with uniformly ultimate boundedness stability guarantee [J]. Automatica, 2021, 129: 109689.

[19] YAMADA S, NAKASHIMA M, SHIONO S. Reinforcement learning to train a cooperative network with both discrete and continuous output neurons [J]. IEEE Transactions on Neural Networks, 1998, 9(6): 1502-1508.

[20] WANG L, LIU Y, ZHAI X. Design of reinforce learning control algorithm and verified in inverted pendulum [C]// Proceedings of 2015 34th Chinese Control Conference (CCC). Hangzhou: IEEE, 2015: 3164-3168.

[21] LIAN J, LI S, LIU J. T-S fuzzy control of positive Markov jump nonlinear systems[J]. IEEE Transactions on Fuzzy Systems, 2018, 26(4): 2374-2383.

[22] WEN G, LI B, NIU B. Optimized backstepping control using reinforcement learning of observer-critic-actor architecture based on fuzzy system for a class of nonlinear strict-feedback systems[J]. IEEE Transactions on Fuzzy Systems, 2022, 30(10): 4322-4335.

[23] XIE Z, SUN T, KWAN T, et al. Motion control of a space manipulator using fuzzy sliding mode control with reinforcement learning[J]. Acta Astronautica, 2020, 176: 156-172.

[24] SHI H, LIN Z, ZHANG S, et al. An adaptive decision-making method with fuzzy Bayesian reinforcement learning for robot soccer[J]. Information Sciences, 2018, 436: 268-281.

[25] DESOUKY S F, SCHWARTZ H M. Self-learning fuzzy logic controllers for pursuit-evasion differential games [J]. Robotics and Autonomous Systems, 2011, 59(1): 22-33.

[26] AWHEDA M D, SCHWARTZ H M. A residual gradient fuzzy reinforcement learning algorithm for differential games[J]. International Journal of Fuzzy Systems, 2017, 19(4): 1058-1076.

[27] 班晓军，李士勇．倒立摆的一种 FUZZY-PD 复合控制器设计[J]. 哈尔滨工业大学学报，2003，35(11)：1290-1293.
BAN Xiaojun, LI Shiyong. A FUZZY-PD compound inverted pendulum controller design[J]. Journal of Harbin Institute of Technology, 2003, 35 (11): 1290-1293(in Chinese).

（编辑：孟彬）