基于强化学习的自适应 PID 智能控制虚拟仿真平台构建

郑国权

岭南师范学院，广东省湛江市，524048；

摘要：PID 控制是工业过程控制中经典且应用广泛的策略，其参数整定是研究热点。传统整定方法依赖精确数学模型，面对复杂系统难以获理想效果。强化学习为 PID 参数自适应整定提供新思路。本文综述基于强化学习的自适应 PID 控制器研究进展，重点探讨虚拟仿真平台构建技术。文章先阐述传统 PID 控制与强化学习结合的必要性与优势；接着分析强化学习自适应 PID 控制的核心原理与典型算法；然后提出分层模块化的虚拟仿真平台总体架构，论述其核心组成模块；之后通过典型被控对象仿真案例验证平台有效性；最后总结当前领域面临的主要挑战，如训练效率等问题，并对未来发展方向，如与深度学习融合等进行展望。本文旨在为相关研究人员提供技术参考和实践指南。

关键词：强化学习；自适应 PID 控制；虚拟仿真平台；数字孪生

DOI：10.64216/3080-1516.25.04.079

引言

比例积分微分控制器诞生后，因结构简单、鲁棒性好等优点，在超 90%工业控制系统中占主导。但其控制性能依赖三个参数合理设置，传统 PID 参数整定方法，如 ZieglerNichols 法等，需基于系统阶跃响应模型或频域特性，在特定工作点附近进行，存在明显局限：一是依赖被控对象线性、时不变数学模型，而实际工业过程特性复杂，难以精确建模；二是系统工作点变化或受扰动时，固定参数的 PID 控制器难以维持最优性能，甚至致系统失稳。为克服传统 PID 不足，自适应控制理论出现，自适应PID控制器能根据系统状态自动调整参数。早期自适应 PID 控制策略对模型准确性要求高、算法复杂，工程推广有困难。近年来，强化学习发展为智能控制带来新活力。强化学习不依赖先验模型，通过智能体与环境交互“试错”学习最优策略，与自适应控制理念契合。将强化学习与 PID 控制结合，形成新型智能自适应 PID 控制器。不过，强化学习算法训练需大量数据，在真实物理系统学习成本高、风险大，所以构建虚拟仿真平台很重要，它是验证算法有效性的“试验场”和加速应用的“催化剂”。

1 强化学习自适应 PID 控制原理与算法

1.1 基本框架

将强化学习应用于PID参数自适应整定的基本框架。该框架包含两个核心部分：被控对象（环境）和 RL 智能体（控制器中的参数整定器）。环境：即被控对象（如电机、化工过程、飞行器等）及其闭环控制系统。环境的状态通常包括系统的输出误差 e(t)、误差积分、误差微分、系统输出 y(t)等。智能体采取的动作会作用与环境，环境反馈新的状态和奖励。

智能体：其核心任务是学习一个参数调整策略 π。在每一个控制时刻 t（或一个控制周期），智能体观测到当前环境状态 s_t，并根据其策略选择一個动作 a_t。这个动作就是 PID 控制器参数的调整量，即 ΔKp, ΔKi, ΔKd。随后，PID 控制器的参数更新为 Kp=Kp0+ΔKp,Ki=Ki0+ΔKi,Kd=Kd0+ΔKd。

奖励函数：是引导智能体学习的方向盘。其设计至关重要，通常需要综合考虑控制系统的多项性能指标。一个典型的奖励函数设计如下：rt=-(αet2+βut2+γ(Δut)2)其中，e_tˆ2 惩罚跟踪误差，保证系统的准确性；u_tˆ2 惩罚控制量，避免能量过大；(\Deltau_t)ˆ2 惩罚控制量的变化率，使控制过程平滑。系数 α，β，γ 用于权衡各项指标的相对重要性。通过不断交互，智能体的目标是学习到一个最优策略 π*，使得调整 PID 参数后，整个闭环系统能够获得最大的长期累积奖励，即实现综合控制性能的最优。

1.2 典型强化学习算法

应用于自适应 PID 控制的 RL 算法主要可分为三类：

1.2.1 基于值函数的方法

这类方法试图学习一个状态-动作值函数 Q(s,a)，表示在状态 s 下执行动作 a 所能获得的期望累积奖励。最具代表性的是 Q-learning 及其扩展算法。Q-learnin

g:通过更新 Q 表来学习最优策略。其优点是离线学习、简单有效。但对于连续状态和动作空间（PID 参数调整通常是连续的），需要离散化，这会引发“维度灾难”，导致学习效率低下、精度受限。

深度 Q 网络：利用深度神经网络来近似 Q 函数，成功解决了高维状态空间的问题。但在处理连续动作空间时，需要结合策略梯度方法（如 Actor-Critic）。

1.2.2 基于策略搜索的方法

这类方法直接参数化策略 π θ (a|s)，并通过优化策略参数 θ 来最大化期望回报。代表性算法是 REINFORCE。优点：天然适用于连续动作空间。

缺点：采样效率较低，方差较大，训练过程可能不稳定。

1.2.3 基于 Actor-Critic 的方法

这是目前最主流、最有效的框架，它结合了值函数和策略搜索的优点。框架包含两个部分：Actor（执行者）：负责根据当前策略选择动作（即调整 PID 参数）。

Critic（评价者）：负责评估 Actor 所执行动作的优劣，即估计状态值函数 V(s)或优势函数 A(s,a)，并指导 Actor 的策略更新。代表性算法包括：深度确定性策略梯度：专为连续控制问题设计。其 Actor 网络直接输出连续的动作值。DDPG 表现出色，但对超参数敏感。

近端策略优化：通过引入裁剪机制，保证了策略更新的稳定性，使其更易于调参和收敛，成为当前实践中的首选算法之一。

软演员-评论员：在目标函数中增加了策略的熵，鼓励探索，提高了算法的鲁棒性和采样效率。对于自适应 PID 控制问题，由于动作空间（参数调整量）是连续的，且需要学习平滑稳定的调整策略，Actor-Critic 框架下的 DDPG、PPO 和 SAC 算法通常能取得比传统方法更好的效果。

2 虚拟仿真平台的构建

一个功能完善、运行高效的虚拟仿真平台是 RL 自适应 PID 控制研究得以深入开展的关键基础设施。其构建应遵循模块化、可扩展、易用性原则。

2.1 总体架构设计

平台应采用分层模块化架构，通常包括以下四个层次：数据层：负责存储和管理所有数据，包括被控对象模型参数、训练过程中的交互数据（状态、动作、奖励）、训练好的智能体模型、仿真结果数据等。

仿真引擎层：平台的核心计算层，包含环境动力学模型求解器（如 ODE 求解器）和 RL 智能体训练算法库（如基于 PyTorch/TensorFlow 的实现）。

服务层：提供核心业务逻辑的接口，如仿真任务管理、模型训练调度、数据可视化服务、评估指标计算等。

应用层：用户交互界面，包括图形化配置界面、实时监控仪表盘、三维可视化场景、结果分析报告生成器等。

2.2 核心模块详解

2.2.1 环境模型库

这是平台真实性的基础。库中应集成多种典型被控对象的数学模型，涵盖线性与非线性、时变与定常、集中参数与分布参数等不同类型。经典控制系统：倒立摆、球杆系统、飞行器姿态动力学模型、机械臂模型、直流电机模型等。

过程控制系统：液位控制系统、温度控制系统、连续搅拌反应釜模型等。

接口开放：平台应提供标准接口（如 PythonAPI、FMI/FMU 标准），允许用户方便地导入自定义模型或利用 Simulink、Modelica 等专业建模工具建立的模型，实现与主流仿真软件的协同仿真。

2.2.2 智能体训练引擎

这是平台的“大脑”，集成了多种 RL 算法。算法库：应至少包含 DDPG,PPO,SAC,TD3 等主流连续控制算法。

训练管理：支持分布式并行训练，以加速学习过程。提供训练过程的实时监控，如奖励曲线、状态曲线、动作曲线的动态显示。

超参数配置：提供友好的图形化界面，方便用户配置网络结构、学习率、折扣因子等超参数。

2.2.3 可视化与人机交互界面

良好的可视化是理解和调试算法的关键。场景可视化：对于机械系统（如倒立摆、无人机），应提供 2D/3D 动画展示，直观呈现控制效果。

数据可视化：实时绘制系统输出、控制输入、PID 参数变化曲线、奖励值曲线等。

交互控制：允许用户在仿真过程中暂停、继续、注入扰动、切换控制器（对比 RL-PID 与常规 PID），并实时修改模型参数或控制器参数。

2.2.4 评估与分析系统

平台需提供一套科学的评估体系，用于量化比较不同算法的性能。性能指标：集成多种控制性能指标，如积分绝对误差、积分平方误差、上升时间、超调量、调节时间、控制量消耗等。

鲁棒性测试：支持对训练好的控制器进行测试，如参数摄动测试、外部扰动测试、设定值突变测试等。

对比实验：支持将 RL 自适应 PID 与 Z-N 法整定的 PID、模糊 PID 等控制器的控制效果进行同屏对比，生成详细的对比报告。

2.3 实现技术选型

现代虚拟仿真平台通常基于成熟的技术栈构建。编程语言与框架：Python 是首选，因其在科学计算和机器学习领域的丰富生态。核心仿真计算可用 NumPy/SciPy；RL 算法实现可基于 PyTorch 或 TensorFlow。

物理引擎：对于复杂刚体动力学仿真，可集成 MuJoCo、PyBullet 等专业物理引擎，以获得更高的物理真实性。

图形界面：可采用 Web 技术栈（如 React/Vue+WebGL）构建跨平台的浏览器应用，或使用 PyQt、Tkinter 等构建桌面应用。

云原生与数字孪生：未来平台可向云原生架构演进，利用容器化（Docker）和编排技术（Kubernetes）实现资源弹性调度。并可与数字孪生技术结合，实现与真实物理系统的实时数据交互和同步仿真。

3 典型应用案例仿真分析

在本节中，我们通过两个经典案例来展示基于该仿真平台的 RL 自适应 PID 控制器的应用效果。

案例一：倒立摆起摆与稳摆控制

倒立摆是一个经典的非线性、不稳定系统，是验证控制算法的“试金石”。环境状态：s=[θ, θ',x,x']，即摆杆角度、角速度、小车位置、车速。

动作：a=[ΔKp, ΔKi, ΔKd]，即调整 PID 控制器（控制对象为小车加速度/力，目标为使摆杆直立）的参数。

奖励函数：r=-(θ2+0.1θ′ 2+0.001x2+0.001u2)，重点惩罚角度偏差。

训练过程：在平台上使用 PPO 算法进行训练。智能体从随机初始状态开始，通过数百万步的交互，逐渐学习到如何调整 PID 参数，使得小车通过左右移动成功地将摆杆竖起并保持平衡。

结果分析：训练完成后，与 Z-N 法整定的固定参数 PID 进行对比。仿真结果显示，在受到外力冲击后，RL 自适应PID控制器能更快地调整参数，使摆杆恢复平衡，且超调更小，表现出更强的鲁棒性和适应性。案例二：四旋翼无人机姿态控制

无人机姿态系统是一个多变量、强耦合的非线性系统。控制设计：为俯仰、滚转、偏航三个通道分别设计一个 RL 自适应 PID 控制器。

状态与动作：每个控制器的状态包括对应通道的欧拉角误差和角速度误差，动作为该控制器 PID 参数的调整量。

挑战与策略：耦合效应是主要挑战。在奖励函数中，除了本通道的误差，还需轻微惩罚其他通道的误差，以引导智能体学习解耦控制。

仿真结果：平台上的三维可视化清晰展示了控制效果。RL 自适应 PID 控制器在面对阵风扰动时，能显著减小姿态角的波动，相比固定参数 PID，控制品质提升明显。平台的多变量数据曲线也直观揭示了 RL 控制器在抑制通道间耦合方面的优势。

4 挑战与未来展望

尽管 RL 自适应 PID 控制展现出巨大潜力，但其发展和应用仍面临诸多挑战。

4.1 当前面临的主要挑战

训练效率与稳定性：RL 算法的训练通常需要大量样本，耗时较长。训练过程可能不稳定，收敛性难以保证。

奖励函数设计：奖励函数的设计高度依赖于专家经验，被称为“奖励工程”。不合理的奖励函数可能导致智能体学习到不符合预期的“捷径”行为。

安全性保障：在探索过程中，智能体可能会执行危险动作，导致仿真中系统发散。如何引入安全约束，实现安全强化学习，是走向实际应用的关键。

仿真到现实的迁移：仿真模型与真实世界之间存在不可避免的“现实差距”。在仿真中训练好的控制器，直接部署到实物上可能性能下降。域随机化、系统辨识和在线自适应学习是解决该问题的重要方向。

4.2 未来研究方向展

望算法融合与改进：将 RL 与模型预测控制、模糊逻辑、专家系统等传统方法深度融合，利用先验知识引

导学习，提高采样效率和稳定性。

平台技术深化：发展更高保真的仿真技术，结合数字孪生，实现虚拟与现实的实时交互与迭代优化。构建云边端协同的仿真平台，支持大规模、分布式训练。

自动化与智能化：研究自动化的奖励函数学习、超参数优化技术，降低平台的使用门槛，使其成为更通用的智能控制器自动设计工具。

新兴应用场景拓展：将 RL 自适应 PID 控制应用于更复杂的场景，如能源互联网、智能交通、生物医学等领域的复杂系统控制。

5 总结

本文系统综述了基于强化学习的自适应PID控制方法及其虚拟仿真平台的构建。RL 方法为解决复杂不确定环境下的PID参数在线自整定问题提供了强有力的数据驱动工具，而虚拟仿真平台则为该技术的研发、测试和验证提供了不可或缺的支撑环境。通过对核心原理、平台架构、关键技术和应用案例的分析，可以看出，RL 自适应 PID 控制技术在提升系统控制性能、鲁棒性和自适应性方面具有显著优势。

尽管在训练效率、安全性和仿真到现实的迁移等方面仍存在挑战，但随着强化学习算法本身的不断进步、计算能力的提升以及仿真技术的日益精进，基于强化学习的自适应智能控制必将迎来更广阔的发展前景。未来，一个集成了高性能计算、高保真仿真、智能算法和友好交互的虚拟仿真平台，将成为推动智能控制从理论创新走向工程实践的核心基础设施。

参考文献

[1]陈学松,杨宜民.基于执行器-评价器学习的自适应 PID 控制[J].控制理论与应用,2011,28(8):6.DOI:10.7641/j.issn.1000-8152.2011.8.ccta100618.

[2]谢琪琦.基于强化学习的半主动悬架 PID 控制[J].汽车电器,2024(12):29-32.

[3]吕铁良,张运涵,袁凯平.一种基于强化学习的自适应 PID 温度控制算法:202510020286[P][2025-10-13].