DOI:10.11918/202508035

基于 FPGA 的 DDPG 算法硬件映射解析与机器人运动技能学习

朱晓庆[1,3],毕兰越[1,3],宫婉儒[1,3],吴 通[2,3],李钟军[1,3],吴杜兴[1,3],张 川[2,3],杨晓蓬[1,3]

(1.北京工业大学 信息科学与技术学院,北京 100039;2.中核核信信息技术(北京)有限公司,北京 100091;
3.核工业智能交叉实验室(北京工业大学),北京 100124)

摘 要:为研究神经网络和强化学习算法与高等动物进化原理之间的联系,本文结合深度确定性策略梯度(deep deterministic policy gradient,DDPG)算法构建了一套可观测、可解释的轮足机器人自主运动控制系统。首先在 FPGA(field-programmable gate arrays)上部署 Actor-Critic 神经网络,并设计了一套 FPGA-ARM 机器人控制系统,通过实时导出网络权值激活信号并生成权值热力图,以可视化展示策略演化过程。实验表明,该方案单步计算时延缩减至 28 μs,5 000 步内完成收敛。同时,权值热力图揭示了策略在初期、中期及后期 3 个阶段的动态演化,定性分析表明,非关注区域对整体策略影响微弱、资源利用更趋优化。本文提出的硬件-算法协同框架为强化学习"黑箱"可观测性研究提供了新范式,展示了 FPGA 在嵌入式机器人控制中兼具低延迟、高并行和低功耗的独特优势,为多智能体协作与异构平台下的实时技能学习与硬件加速提供了潜在应用前景。

关键词:机器人;学习机理解析;技能学习;FPGA;强化学习

中图分类号:TP242 **文献标志码**:A **文章编号**:0367-6234(2026)01-0024-11

Hardware mapping analysis of DDPG algorithm based on FPGA and robot motion skill learning

ZHU Xiaoqing[1,3], BI Lanyue[1,3], GONG Wanru[1,3], WU Tong[2,3], LI Zhongjun[1,3],
WU Duxing[1,3], ZHANG Chuan[2,3], YANG Xiaopeng[1,3]

(1. School of Information Science and Technology, Beijing University of Technology, Beijing 100039, China;
2. CNNC Hexin Information Technology (Beijing) Co., LTD., Beijing 100091, China;
3. Nuclear Industry X Intelligence Laboratory (Beijing University of Technology), Beijing 100124, China)

Abstract: This paper investigates the intrinsic connection between neural networks, reinforcement learning (RL) algorithms, and the evolutionary principles of higher animals by developing an observable and interpretable autonomous control system for a wheel-legged robot. Leveraging the Deep Deterministic Policy Gradient (DDPG) algorithm, an Actor-Critic neural network has been implemented directly on Field-programmable gate arrays (FPGA). An FPGA-ARM robot control system is further designed to export weight activation signals in real time and generate weight heatmaps, thereby visualizing the strategy evolution process. Experimental results demonstrate that the proposed system has the ability of reducing the single-step computation latency to 28 μs and achieves convergence within 5 000 steps. Moreover, the weight heatmaps reveal the dynamic evolution of strategies across three phases——early, middle, and late stages. Qualitative analysis indicates that non-salient regions have minimal impact on the overall strategy, resulting in more efficient resource utilization. The proposed hardware-algorithm co-design framework establishes a novel paradigm for improving the interpretability and reducing the "black-box" nature of RL. It also showcases the unique advantages of FPGA in embedded robot control, namely low latency, high parallelism, and low power consumption. This work lays a robust foundation and presents promising prospects for real-time skill learning and hardware acceleration in scenarios involving multi-agent cooperation and heterogeneous computing platforms.

Keywords: robotics; analysis of learning mechanism; skill learning; FPGA; reinforcement learning

收稿日期:2025-08-17;**录用日期**:2025-09-11;**网络首发日期**:2025-10-11
网络首发地址:https://link.cnki.net/urlid/23.1235.T.20251010.1707.008
基金项目:国家自然科学基金(62103009);北京市自然科学基金(4202005)
作者简介:朱晓庆(1987—),男,副教授
通信作者:吴 通,wutong_cnnc@sjtu.edu.cn

在生物学中，大脑被认为是学习和记忆形成的中心器官。大脑中的神经元是相互联系的，并遵循特定的规则来产生所学技能的记忆，电子信号被用来激活神经元，部分神经网络正是以此模拟了学习的过程[1]。高等动物的运动技能学习也为机器人控制提供了重要启示：小脑通过监督学习对运动误差进行快速校正，确保运动的精准与协调[2]。这种双通路学习机制，即小脑的监督校正与基底神经节的强化学习使动物能够在复杂、多变的环境中实现快速适应和稳定运动。在机器人控制领域，通过借鉴小脑误差校正与多巴胺奖励信号的协同机制，可以将高频、低时延的误差反馈与长期、策略性更新相结合，提升系统的实时性和鲁棒性。Tejas 等[3]提出了在层次化强化学习框架下结合内在动机进而实现技能自动发现与演化，为监控和解释技能演化过程提供了新方法。Alexandar 等[4]提出了层次化强化学习结构，将长期技能学习和短期行为分解，有助于解释和监控智能体在不同层次上技能的演化，为实现复杂强化学习算法提供思路，从而显著提升机器人的自主学习能力和性能水平。

强化学习（reinforcement learning，RL）是机器学习的一个重要分支，其研究的是如何采取行动以使累积奖励最大化的决策过程。然而，目前大多强化学习应用仍局限于仿真环境和任务规划，如何将神经网络策略直接映射到硬件以实现实时运动控制的研究尚不充分[5]。FPGA 因其可定制性强与低功耗的特性成为广受关注的解决方案。2021 年 Kadokawa 等[6]设计了基于 FPGA 的机器人视觉识别与强化学习系统，其通过硬件设计将契合机器人技能学习需求的算法部署在实际的硬件平台上，从而使机器人完成相应的技能学习目标。Carlos 等[7]设计了 RobotCore，一个用于 ROS 2 的硬件加速架构，支持 FPGA 和 GPU，旨在提高机器人系统的响应速度和能效。凤雷等[8]在 ZYNQ7100 异构计算平台上完成了对 Cartpole 机器人应用的在线决策任务，实验结果表明，FPGA 在进行典型 DRL 算法训练时的计算速度和运行功耗，相对于 CPU 和 GPU 平台具有明显的优势。这些工作共同证明了基于 FPGA 的 DRL 研究，在机器人实际运动控制、计算加速、图像识别与指令执行等方面的显著优势，为未来机器人控制系统的设计和优化提供了坚实的理论基础和实践经验。

尽管现有强化学习研究在诸多应用中展现了强大的学习和决策能力，但仍存在一些不足，主要体现在以下两个方面。1）硬件-算法割裂问题[9]。目前许多研究在实现强化学习加速时，将硬件平台，如 FPGA、GPU 与算法设计相互独立地进行优化。这种割裂导致两者之间的协同效应未能充分发挥，而 FPGA 可以进行硬件和软件的协同设计，使得算法的实现更加灵活，如可以通过硬件加速关键计算模块，在软件层面上实现复杂的控制逻辑[10]。同时，FPGA 通过硬件级别的优化，在相同的功耗下实现了更高的计算性能，从而提高了能效比。Li 等[11]设计了一种基于异构计算单元的多智能体 FPGA 强化学习加速器，在三智能体的情况下，使用 CPU 73% 的功耗完成了 37 倍的计算量。Nai 等[12]设计了针对深度 Q 学习网络（deep Q learning network，DQN）的 CPU-FPGA 架构加速器，实现了 1.84 倍的速度提升。余奇[13]通过分析深度神经网络、卷积神经网络的预测过程和训练过程算法共性和特性，并以此为基础设计了专用的 FPGA 运算单元。已有研究表明，通过专用的设计使得 FPGA 能够结合硬件特点从而发挥更好的性能。2）“黑箱”训练问题。深度强化学习模型通常采用高度非线性的神经网络结构进行训练，其内部决策过程往往难以解释，形成所谓的“黑箱”。这种不透明性使得研究者难以理解智能体是如何从环境反馈中学习和演化出有效策略的，也增加了对模型鲁棒性和安全性的担忧。不仅如此，在强化学习过程中，也缺乏对智能体技能演化过程的有效监控和解释手段。

针对强化学习在机器人控制中存在的部署困难、计算开销大及解释性不足等问题，本文贡献在于提出一种结合 FPGA 硬件加速与可解释强化学习的统一框架，实现了在实际机器人上的高效部署与因果可解释性分析，开展了以下研究工作：

1）提出并实现基于 FPGA 的 DDPG 算法结构与硬件神经网络，结合功耗与资源利用率分析验证其效率，并设计了 FPGA -嵌入式一体化控制系统以实现机器人运动控制；

2）面向强化学习的“黑箱”特性，以轮足式机器人为对象构建状态与动作空间，分别对应电机力矩、机身高度、关节与机身反馈，借助实验验证算法的有效性与实时性，同时通过 Actor-Critic 结构的权值热力图揭示学习过程对控制行为的影响，探讨神经网络局部区域与长期、短期记忆的关系。

1　基于 FPGA 的反向传播神经网络

1.1　脉冲阵列乘法器

神经网络的反向传播过程与生物神经系统的学习机制具有相似性，研究其工作原理有助于理解人类和动物的学习过程，通过设计专有的神经网络，不

仅可以节约有限的硬件资源,同时可以类比神经系统的学习过程来研究神经网络学习与进化的过程[14]。其中,矩阵乘法器是完成神经网络算法与提高效果的关键,脉冲阵列乘法器(pulse array multiplier)是一种专用的硬件乘法器设计,广泛应用于数字信号处理、计算机体系结构和嵌入式系统中。其通过脉冲信号和延迟机制来实现高效的乘法操作,具有低功耗和高并行性等特点。

脉冲阵列乘法器的示意图如图 1 所示,矩阵乘法器由多个 PE(process element)单元组成,而每个 PE 单元则包含乘法器与累加器,将输入的数据 a 和 b 相乘后进行累加和存放。向量从矩阵乘法器两侧流入,按照从上至下从左至右的顺序,以时钟周期 t 为间隔逐级传递。延迟单元用于控制信号传播,实现并行乘法处理。本文设计的专用矩阵移位单元,在同步时钟驱动下生成部分积并逐步累加,保证了时序精度。根据矩阵的规模 m 和 n,以及时钟周期 t,可以得出单次计算的时间 T。通过以上逐级传播的结构,模拟了整个网络的传播过程,同时可以通过同步时钟精确获得矩阵计算完成的时间,使传播过程更具时效性,也可以一定程度地减少跨时钟域带来的时序问题。

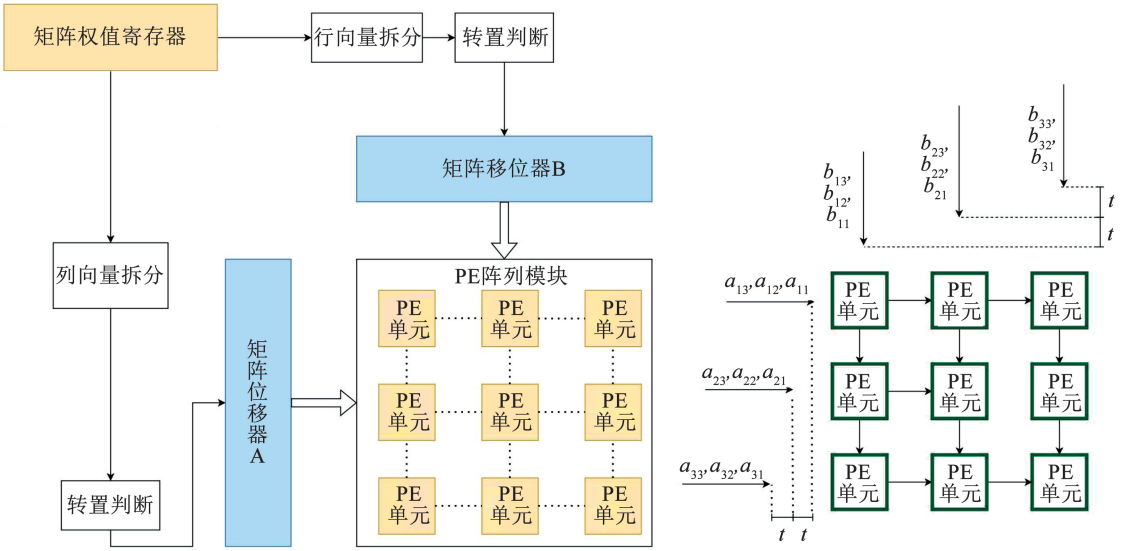


图 1　脉冲阵列乘法器

Fig. 1　Pulse array multiplier

1.2　信号跟踪与硬件映射实验

基于脉冲阵列的乘法器搭建反向传播的神经网络后,将其部署在 FPGA 上,选取正弦信号作为输入目标,开展信号的跟踪实验以验证其时序逻辑问题,以及是否能够有效完成所设计的功能。同时观察其综合情况判断资源使用量及性能,分析对非线性信号的跟踪能力和硬件映射的能力。由图 2 所示,本文涉及的神经网络初步完成了信号的跟踪学习,每个周期进行 100 次采样,采样频率为 500 Hz。搭建的神经网络在 6 个信号周期内达到基本收敛,用时 1.2 s,证明其拥有学习能力,且具有较好的速度和实时性。同时,其可以完成对非线性函数的拟合及跟踪任务,证明其可以完成策略的学习和推理,同时完成动态平衡的信号跟踪。

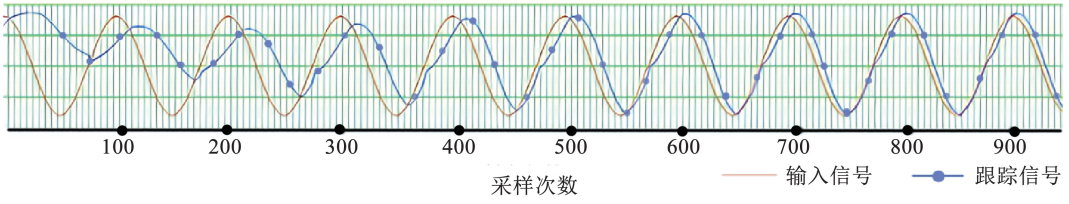
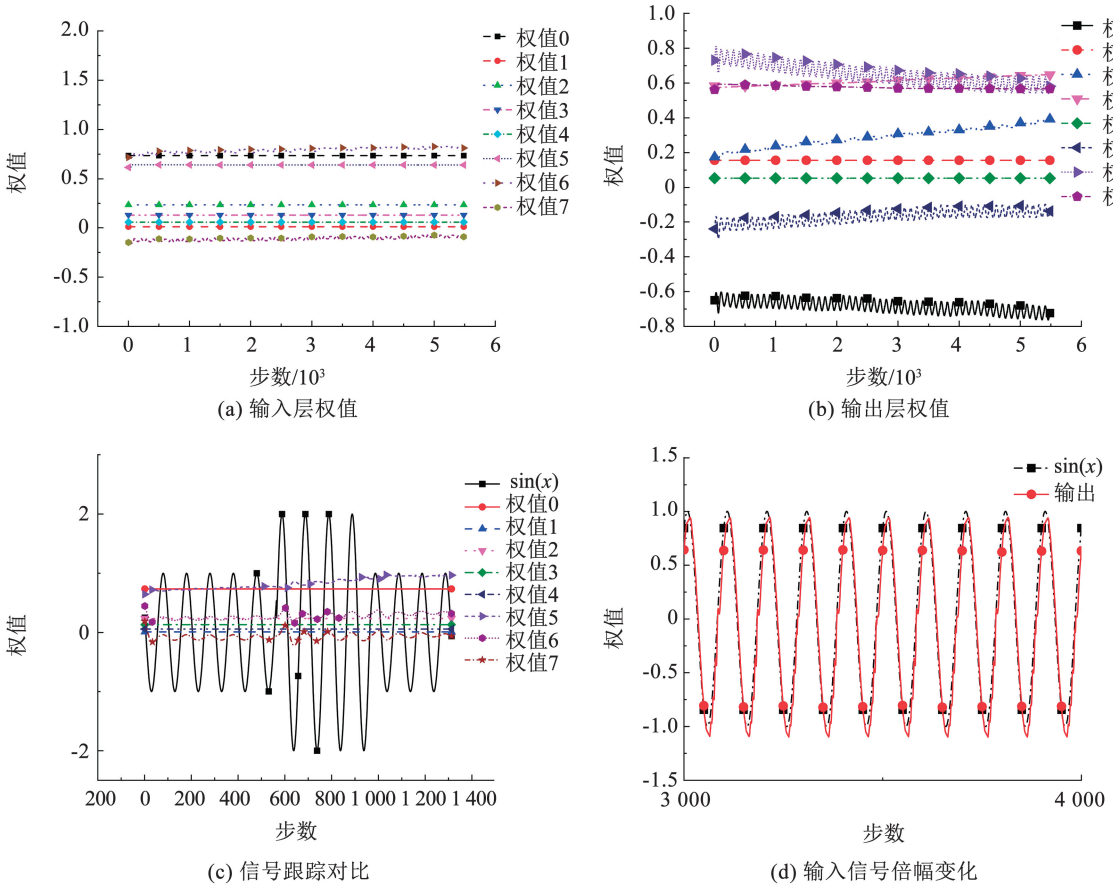


图 2　神经网络学习结果

Fig. 2　Neural network learning results

如图 3 所示,将网络权值导出后可以看到 8 个模块的网络权值快速收敛。将矩阵乘法、激活函数(ReLu)等基本操作分解为 FPGA 的流水线级操作,通过每级流水线的输入/输出数据,展示非线性变换的硬件实现细节。同时可以通过每个模块的网络激活获得整个网络被访问的情况。图 3(a)和(b)分别给出了输入输出层的权值变化与收敛趋势,图 3(c)显示了较小的跟踪误差,而图 3(d)显示了

当输入信号倍幅(由 sin(x)变为 2sin(x))的场景下权值的响应情况。该实验为验证网络在学习过程中获得有效反馈提供了证据，证明了系统在动态环境中自适应调节的能力。



(a) 输入层权值　(b) 输出层权值

(c) 信号跟踪对比　(d) 输入信号倍幅变化

图 3　神经网络权值变化

Fig. 3　Changes in neural network weights

在本研究中，将网络训练过程映射为可测量的物理信号流是核心目标之一。通过展示 BP 神经网络的实时权值演变，为整个系统提供了一个直观、可验证的硬件平台，使得强化学习模型在实际机器人运动控制中的作用可视化，进而通过硬件监测的方式进行解析和优化，为理解和解释深度强化学习算法在机器人运动控制中的内部机制提供了方法层面的支撑。

2　基于 FPGA 的 DDPG 算法

2.1　算法结构与实现

强化学习算法根据选取动作的策略不同，分为基于概率的强化学习(policy-based RL)和基于价值的强化学习(value-based RL)。在选择适用于特定应用场景的强化学习算法时，DDPG(深度确定性策略梯度)算法因其独特的特性在许多场景脱颖而出，尤其是在连续动作空间中需要输出确定、精确动作的任务中，如机械臂操作、无人机飞行控制以及自动驾驶车辆等[15-16]。DDPG 结合了值函数方法和策略搜索方法的优点，由于输出的结果在连续的动作空间内是确定的值而非概率分布函数，因此可以大幅减少随机动作以及复杂逻辑对于 FPGA 资源的消耗。

DDPG 算法采用演员-评论家(actor-critic)架构。Actor 是策略网络，输出确定性的动作。而 Critic 代表价值网络，评估当前状态-动作对的价值，通过奖励函数引导价值网络，使其输出合乎专家给定策略的评价结果。本文采用的 DDPG 算法结构如图 4 所示。其中，Actor 网络和 Critic 网络均由软更新反向传播控制引擎控制，基于 1.1 小节实现的神经网络扩展而成。Actor 网络的输入量是智能体与环境交互所反馈的状态向量 state，输出是智能体可以执行的动作空间向量 action。Critic 网络的输入是动作空间向量与状态向量的联合向量，输出为在该时刻状态下对应动作好坏的评分值，此评分值由奖励函数 reward 引导。输出之后与奖励函数根据当前状态 state 所计算出的奖励值 reward 作差得到对

应梯度,并实时反馈。奖励值与 Critic 评估值对比后反向传播进行梯度上升,来保证获得更高的奖励值。这个过程同时也会获得反馈的动作空间梯度,再将动作空间梯度反馈给 Actor 网络进行梯度下降,使其更加贴近合理的策略,最后更新当前 Actor-Critic 网络使其接近目标网络。

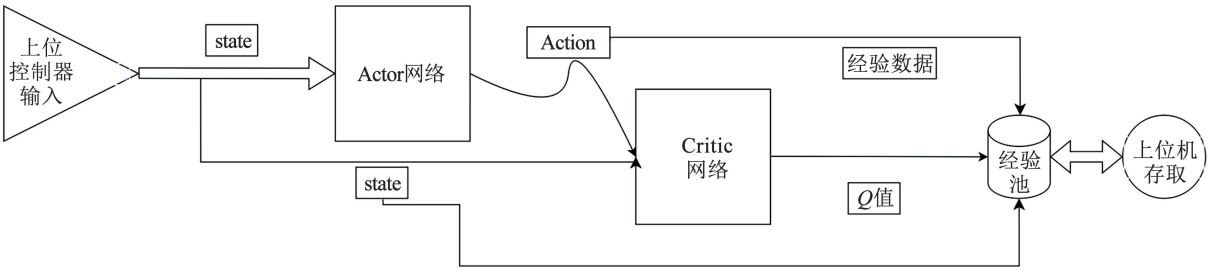


图 4 基于 FPGA 的 DDPG 算法结构图

Fig. 4 FPGA-based DDPG algorithm structure diagram

在运行过程中,DDPG 的动作价值函数 $Q_\pi(\boldsymbol{s},\boldsymbol{a})$ 表示智能体从状态 $\boldsymbol{s}$ 出发,根据策略 π 执行动作 $\boldsymbol{a}$,最终获得的回报。智能体根据状态 $\boldsymbol{s}_t$ 生成的动作 $\boldsymbol{a}_t$,表示为式(1)、式(2):

$$\begin{aligned} Q_\pi(\boldsymbol{s},\boldsymbol{a}) &= E_\pi[r_{t+1}+\eta R_{t+1} \mid \boldsymbol{s}_t=\boldsymbol{s},\boldsymbol{a}_t=\boldsymbol{a}] = \\ &\quad E_\pi[r_{t+1}+\eta Q_\pi(\boldsymbol{s}',\boldsymbol{a}') \mid \boldsymbol{s}_t=\boldsymbol{s},\boldsymbol{a}_t=\boldsymbol{a}] = \\ &\quad \sum_{\boldsymbol{s}'\in S} P(\boldsymbol{s}' \mid \boldsymbol{s},\boldsymbol{a})[R(\boldsymbol{s},\boldsymbol{a}) + \\ &\quad \eta \sum_{\boldsymbol{a}'\in A} \pi(\boldsymbol{a}' \mid \boldsymbol{s}') Q^\pi(\boldsymbol{s}',\boldsymbol{a}')] \end{aligned} \tag{1}$$

$$\boldsymbol{a}_t = \mu(\boldsymbol{s}_t \mid \theta^\mu) + N_t \tag{2}$$

式中:$\boldsymbol{a}_t$ 与 $\boldsymbol{s}_t$ 分别为智能体在 t 时刻的状态空间向量与动作空间向量,r_t 为通过奖励函数计算的奖励值,$\boldsymbol{a}'$ 与 $\boldsymbol{s}'$ 为下一时刻的状态,R_t 为 t 时刻的累计回报,η 为学习率,E_π 为在策略 π 引导下的期望,$P(\,)$ 为状态转移概率分布,N_t 为探索噪声,$\mu(\,)$ 为以 θ^μ 为参数的 Actor 网络。

对于 Critic 当前网络,DDPG 的损失函数和 DQN 是类似的,都是均方误差,即

$$J(\theta^Q) = \frac{1}{m}\sum_{j=1}^{m}(y_j - Q(\varnothing(\boldsymbol{S}_j),\boldsymbol{A}_j,\theta^Q))^2 \tag{3}$$

Actor 网络通过策略梯度来确定损失函数,Actor 的更新倾向于使得 Q 取得极大值。故 Actor 的损失与 Q 值负相关,因此对状态估计网络返回的 Q 值取负值,得到 Actor 网络梯度如式(4)所示。

$$J(\theta^\mu) = -\frac{1}{m}\sum_{j=1}^{m} Q(\boldsymbol{S}_j,\boldsymbol{A}_j,\theta^\mu) \tag{4}$$

式(3)、式(4)中:θ^Q、θ^μ 分别对应 Critic 和 Actor 网络权值,y_j 为目标网络下一步估计,$\boldsymbol{S}_j$、$\boldsymbol{A}_j$ 为动作与状态集合,$\varnothing(\boldsymbol{S})$ 为状态的特征表示。

获得网络梯度后,对网络权值进行更新,得到:

$$\theta^\mu \leftarrow \theta^\mu + \eta_\mu J(\theta^\mu) \tag{5}$$

$$\theta^Q \leftarrow \theta^Q + \eta_Q J(\theta^Q) \tag{6}$$

式中 η_Q、η_μ 为 Critic 和 Actor 网络学习率。

通过上述公式设计梯度更新引擎,但由于算法中矩阵运算的数量和规模较大,使得串行分布网络会导致严重的资源不足。为了降低设计的冗余性,设置如图 5 所示的状态机完成状态转移,以控制神经网络和矩阵乘法器的复用。当输入权值被激活时,计算开始信号置高;在时钟周期测算完成后,计算完成信号被置高,从而驱动状态机转移至输出缓存状态,以获取激活后的网络层输出。

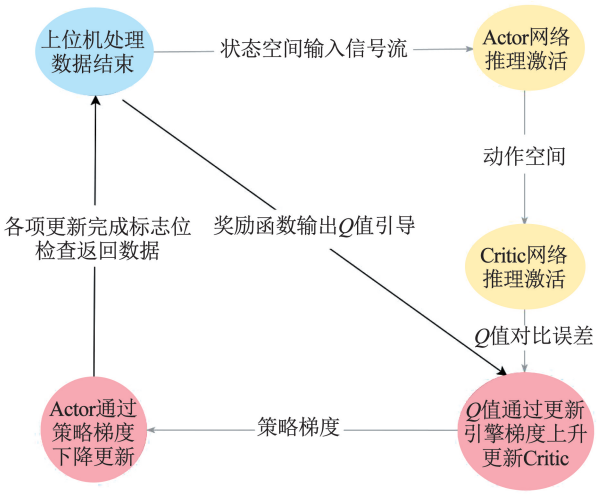


图 5 状态机转移图

Fig. 5 State machine transition diagram

2.2 学习过程可视化设置:经验池系统与信号流的映射

为深入解释机器人技能学习的内在机理并提升策略的可解释性,本文在系统设计中引入了网络内部状态的可观测性机制。经验池技术于 1992 年被提出,后经深度 Q 网络(DQN)等算法验证,通过存储智能体交互轨迹的四元组$(\boldsymbol{s},\boldsymbol{a},r,\boldsymbol{s}')$实现经验解相关与复用。在 DDPG 框架下,其不仅是稳定训练的关键组件,也为解析参数更新动力学提供了重要窗口。本研究将这一算法层概念转化为可观测的物理存储结构,进而揭示神经网络训练过程中的梯度传播规律。

基于 FPGA 的硬件特性，为构建硬件可解释性框架，本研究设计了基于块存储器（block RAM，BRAM）的嵌入式经验池系统。BRAM 的固定延迟（<10 ns）和并行访问特性使其成为研究经验复用的理想载体。系统采用分区地址策略：输入层（0x0000-0x0FFF）、隐藏层（0x1000-0x1FFF）和输出层（0x2000-0x2FFF），分别存储全连接权值矩阵（64×64）与动作输出信号。其中，隐藏层采用双端口 BRAM 支持并行读写，输出层结合高冗余存储提升抗噪性。

同时，为完成物理信号的映射，本文设计了导出激活信号与网络权值的功能，如图 6 所示。通过控制神经元权值的激活信号驱动被激活的层级，来完成相应阶段的运算和信号传播。每个神经元节点的激活信号与其权值可以直接通过存储的异步复位触发器的控制信号进行导出，类比神经元的活性值，可以获得更多的神经网络信息[17]。

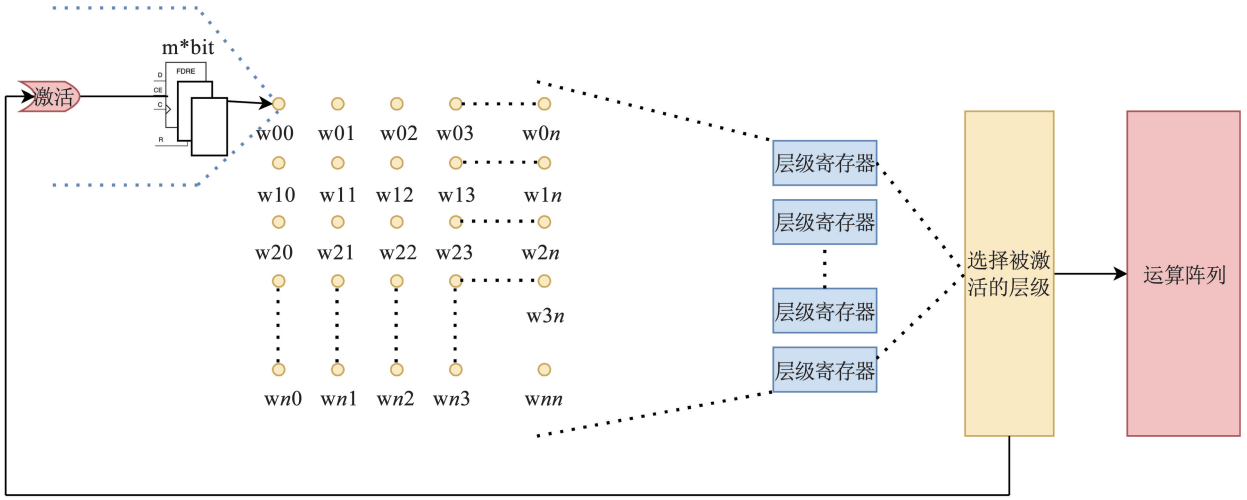


图 6　激活信号驱动结构

Fig. 6　Activation signal drive structure

本文通过导出神经网络激活信号，直观呈现网络关注区域，验证了硬件仿生学习器在实时跟踪与解释中的可靠性。不同于依赖软件后处理的方法，本文所提出的硬件内嵌式导出方案，可在迭代过程中同步记录内部状态并实现信号流物理映射，为调试与优化提供支持。该方案既增强了硬件实现的可解释性，又为揭示深度强化学习在机器人运动控制中策略演化机理，提供了直观、量化且可追溯的观测手段。通过在权值更新中附加时间戳并结合采样分析，可精确关联学习曲线与控制响应，从而解析策略演化过程。

3　实　验

3.1　实验设计

3.1.1　机器人及其控制系统设置

本实验将 Xilinx Zynq7020 FPGA 开发板作为控制核心来验证实验，主控芯片采用 ESP32-WROOM-32E 模组，采用前馈 PID 的 SimpleFOC 控制 4010 电机。传感器设备包括 INA240 电流传感器感知电机力矩，以及 jy601 惯性模块来获得机身的角度和角速度。实物机器人模型选择开源模型 LeTian-bot，机器人三维模型如图 7 所示，机器人硬件参数如表 1 所示。

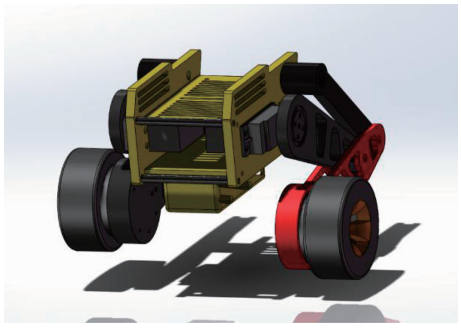


图 7　机器人三维模型

Fig. 7　Robot 3D model

表 1　机器人参数表

Tab. 1　Robot Parameters

大腿长度/mm	小腿长度/mm	机身宽度不含轮/mm	机身宽度/mm	不含电机总质量/g
120	105	44	166	345.7
轮半径/mm	**电机质量/g**	**电机减速比**	**额定扭矩/(N·m)**	**最大扭矩/(N·m)**
20	73	1:1	0.15	0.3

图 8 为机器人控制系统结构图，展示了综合的网络结构图和 FPGA 结构图，将第 2 章所述矩阵乘法器及基于 FPGA 的 DDPG 算法部署至 Zynq 系统的 PL 端，同时通过 BRAM 块建立 AXI 总线使得 PS 端可以与 PL 端交互经验池内的数据。

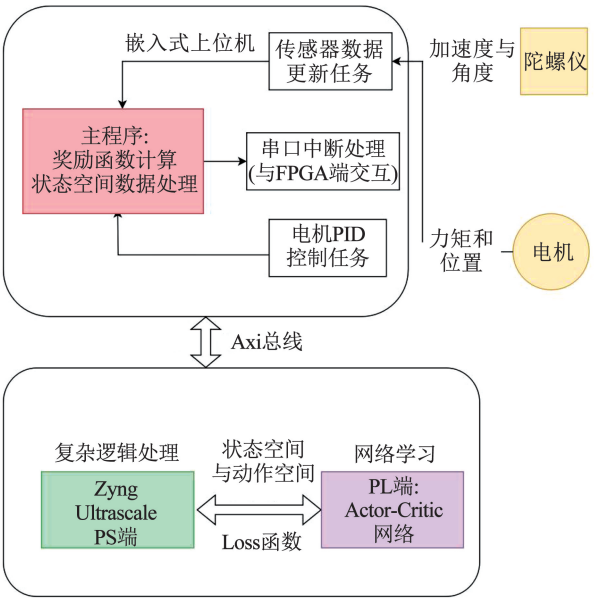


图 8 机器人控制系统结构图

Fig. 8 Robot control system structure diagram

3.1.2 算法设置与资源分析

根据第 2 章所述结构搭建基于 FPGA 的 DDPG 算法网络，设置输入的动作空间 $\boldsymbol{A}$ 和状态空间 $\boldsymbol{S}$ 如下：

$$\boldsymbol{A} = [\tau_0 \tau_1 \theta_0 \theta_1]^{\mathrm{T}} \tag{7}$$

$$\boldsymbol{S} = [\phi \ \dot{\phi} H]^{\mathrm{T}} \tag{8}$$

式中：τ_0、τ_1 为两侧电机力矩，θ_0，θ_1 为决定机身姿态的舵机角度，ϕ 为陀螺仪观测的俯仰角，H 为机身高度。

根据控制任务的需求设置奖励函数 R_t，即

$$R_t = \alpha \sum T_x - \beta |\theta| - \tau + \gamma H \tag{9}$$

式中：α 为累积误差系数，β 为俯仰角误差系数，γ 为高度奖励系数，τ 为关节电机总力矩。奖励函数设计考虑到多个方面，首先是以累计误差 $\sum T_x$ 与实时误差 $|\theta|$ 来确保机器人的平衡稳定性，通过期望机身高度来保证机器人姿态，误差的大小直接影响机器人的稳定性，因此通过这种设计，能够引导机器人在学习过程中逐渐减少不必要的偏差，最终保持平衡。其次，为了节能并减少电机负担，通过设置最小化电机力矩 τ 来控制机器人在运行时的功耗，以提高其运行能耗比。最后根据表 2 所示参数设置算法。

表 2 算法参数设置

Tab. 2 Algorithm parameters

累计误差系数 α	俯仰角误差系数 β	高度奖励系数 γ	力矩最小化系数
1.0	5.0	0.5	1.0
η_Q	η_μ	Batch size	探索噪声 N_t 标准差
2^{-10}	2^{-11}	64	0.2

基于 DDPG 算法网络完成 FPGA 端的编写后获取综合报告以分析其性能，表 3 为本文算法的使用资源情况。在能效比方面，由于 FPGA 执行计算任务时的功耗远低于传统的 CPU 和 GPU，所以适合对功耗有严格要求的应用场景[18]。余子健[19]在 Xilinx 的 Virtex-5 系列 FPGA 实现了一个 4 层卷积神经网络的前向计算加速器，其能够在功耗为 CPU 的 2.68% 的情况下实现 4 倍的处理速度。Hu 等[20]设计了基于 FPGA 的 TD3 强化学习实现，达到了 GPU 的 8 倍能效比。上述研究结果充分证明了 FPGA 相较于 GPU/CPU 在能效比和实时性上的优势，因此本文不再进行跨平台对比，而是将性能评估的重点放在 FPGA 平台上不同算法的表现差异。输入数据与权值均归一化为 16 位定点数（14 位小数），在保持计算精度的同时减少了 LUT 消耗。结合 UltraScale 系列 DSP 的乘法优化特性，实现了在 64×64 矩阵规模下的高效并行运算与参数存储。整体硬件方案在确保吞吐率的同时有效控制了功耗，满足嵌入式平台对自主学习与实时控制的需求。

表 3 算法性能对比表

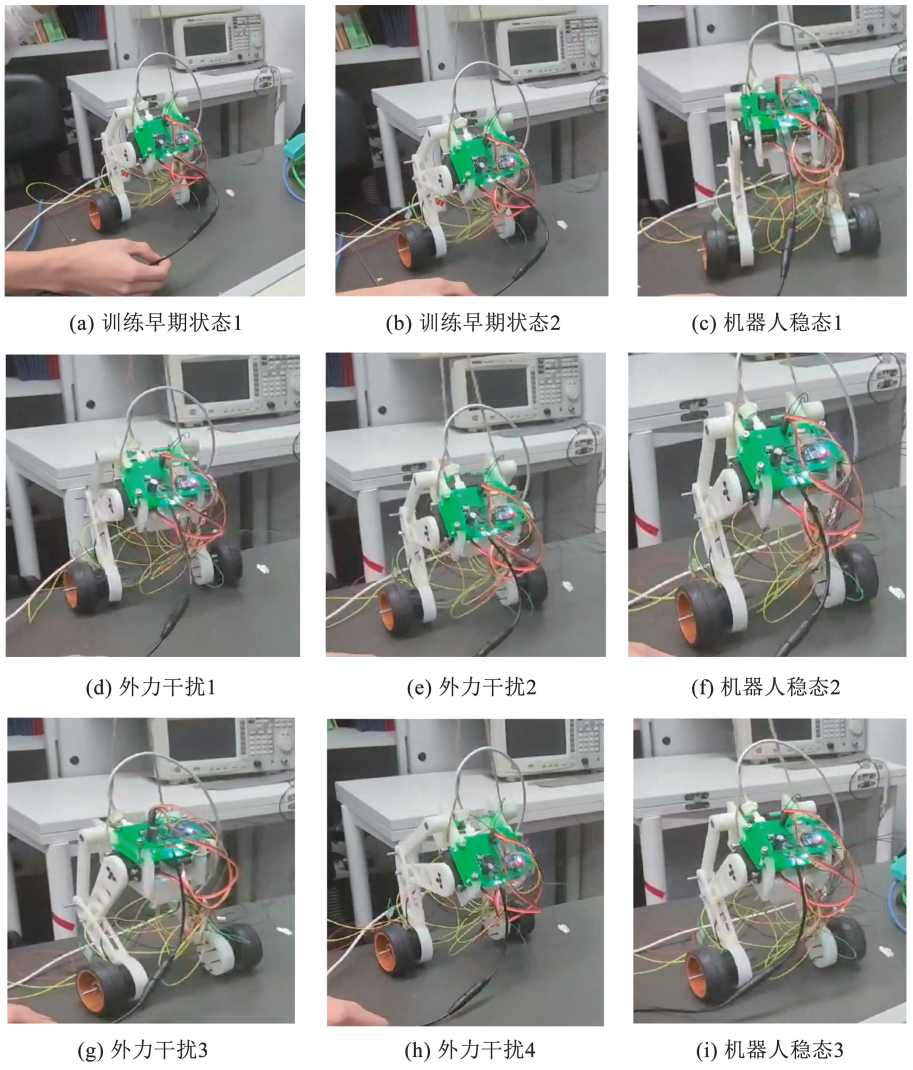
Tab. 3 Algorithm performance comparison

算法	设备	SOC	数据精度	矩阵规模	频率/MHz	LUT 数量/K		LUT 百分比/%	DSP 数量		DSP 百分比/%	使用 BRAM	功耗/W
						可用	使用		可用	使用			
FireFly[21]	Xczu3eg	Ultra-scale	INT8	144×16	300	70	15	21.4	360	288	80.0	162	2.550
GLSVLSI'19[22]	Xc7vx690t	No	FIX32	32×32	100	433	53	12.7	3 600	0	0	65	—
TCAD'22[23]	Xc7k325t	No	FIX16	16×16	200	203	16	7.8	840	0	0	220	0.982
本文	Xczu4ev	Ultra-scale	Fix16_14	64×64	100	87	19	21.8	728	260	35.7	175	3.329

3.2 实物实验

利用 3.1、3.2 节的设置搭建实物轮足机器人，完成机器人的平衡技能学习实验，如图 9 为机器人实物实验示意图。图 9(a)、(b)为初始学习区间的表现，由于早期学习策略不稳定导致倾角过大，当机器人达到一定稳定后（图 9(c)），通过外力（安全

绳）添加干扰（图 9 的（d）、（e）、（g）、（h）），而后机器人又回到稳定平衡时的状态（图 9（f）、（i））。通过导出其状态空间数据分析机器人学习过程和结果，通过线性插补法将导出状态空间数据与同一时间内的权值数据均取 5 000 步采样进行对齐，以分析数据及特征。同时将当前 Actor 网络权值梯度与网络梯度更细激活信号导出，以分析更新后的当前网络所映射的区间与不同。



(a) 训练早期状态1　(b) 训练早期状态2　(c) 机器人稳态1

(d) 外力干扰1　(e) 外力干扰2　(f) 机器人稳态2

(g) 外力干扰3　(h) 外力干扰4　(i) 机器人稳态3

图 9　轮足机器人运动技能学习过程实物实验图

Fig. 9　Actual experiment diagram of the wheeled-legged robot′s motor skill learning process

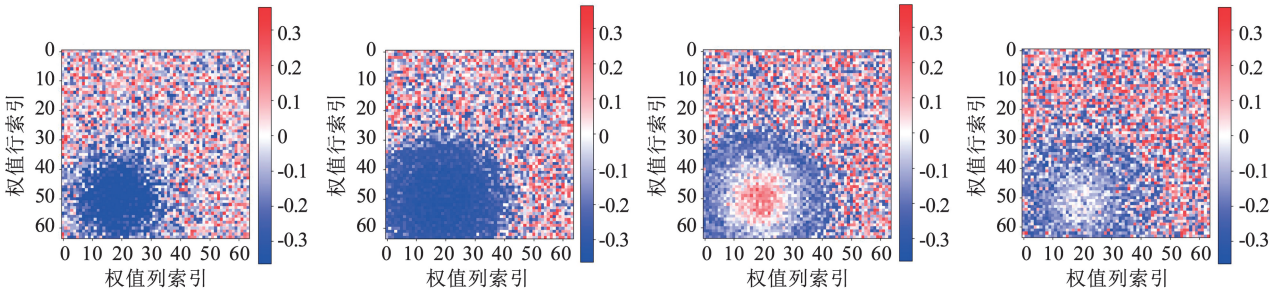
实验完成后将数据导出，利用可视化工具绘制如图 10 所示的网络权值梯度图来展示网络权值的变化过程，通过视频记录了机器人技能的逐步提升，分析了性能指标并展示了随机生成的网络权值变量逐渐收敛到固定值的过程。根据图 10 所示，生成的热力图体现了机器人进行平衡技能学习过程中应对环境变化产生的策略转变，可以通过受关注的部分分析在不同时期所产生的变化。图 10（a）显示训练 300 步左右，梯度热力图提供了早期训练时关注的区域，给出了关于平衡控制时 Actor 网络中权值变化的关注点。这一现象与小脑在动物运动学习中对误差快速修正的机制高度契合：小脑平行细胞突触可塑性使得运动误差被即时放大并修正，代表着该部分在网络中担任短时记忆部分。

图 10（b）显示，进入训练中期，1 100 步左右状态空间的急剧变化使得系统熵增加，输出力矩不停突变，所关注部分权值也呈现急剧变化，权值热力图在 1 100 步呈现多极分化特征，引发运动模式且模拟了生物神经递质的毫秒级突触传递特性，同时体现具有学习技能的控制系统内部构建的分级响应架构：即低频高精度通路处理稳态控制，高频低精度通路专司紧急避障等实时性动作。

图 10（c）同样显示 1 600 步时机器人对环境剧变冲击的响应，通过快速的调节神经元权值的更新情况对抗系统的熵增，持续维持系统稳定。从早期聚焦少数关键权值，到中期多极分化，再到后期渐趋稳定，故可以认为非关注区对平衡部分策略的影响较低。这种选择性强化机制证明：仿生架构应继承

生物系统“关键通路优先优化”的原则，通过硬件可编程阈值等方式实现资源动态分配。

由图 10(d)可以看到 3 400 步后策略趋于稳定，同时收集状态空间数据并绘制如图 11 所示的状态空间变化曲线，可以看到，在多次迭代采样后陀螺仪数据可以趋于稳定。此时机器人机身基本稳定，机身俯仰角均值为 −0.06 rad，标准差为 0.05 rad，标志着平衡技能的学习成功，输出的电机力矩可以有效地保证机器人维持机身姿态。



(a) 340步热力图 (b) 1033步热力图 (c) 1665步热力图 (d) 3481步热力图

图 10 Actor 网络梯度热力图

Fig. 10 Actor network gradient heat map

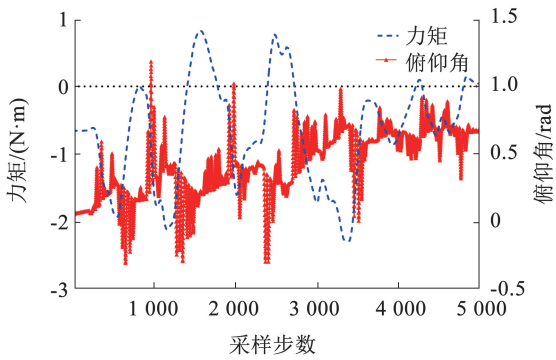
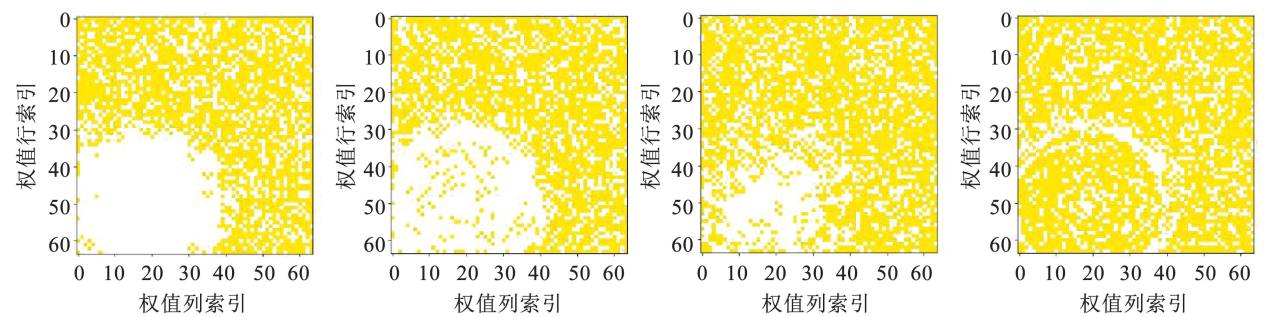


图 11 平衡实验俯仰角和轮力矩曲线

Fig. 11 Balancing experimental pitch angle and wheel torque curves

同时利用可视化工具绘制如图 12 所示的激活热力图，根据图 12 激活图表，白色部分为更新事件激活信号有效，其余部分为更新事件未激活。通过在 FPGA 上实现的脉冲阵列乘法器和权值寄存器链，研究者能够在每次权值更新时捕获激活信号，并按时序输出至上位机，而将激活阈值与更新事件结合。图 12(a)与(b)为对应环境剧变时的激活情况，分析发现，当环境出现剧烈变化时，网络关注区域对应的权值更新触发频率显著提升，这表明网络在动态环境中会对关键特征进行更频繁的自适应调整，从而维持策略的有效性和鲁棒性。而图 12(c)与(d)显示，策略稳定时矩阵更新的激活情况较为分散，代表关注长期记忆的变化而非短时记忆。这些发现为理解生物运动技能的可塑性调控提供了量化电子学模型，印证了“硬件即探针”方法论在解析智能本质层面的独特价值。



(a) 1103步激活信号 (b) 1678步激活信号 (c) 3574步激活信号 (d) 3851步激活信号

图 12 Actor 网络激活信号图

Fig. 12 Actor network activation signal diagram

为了更好地分析算法的性能，本文通过对比实验，比较了其余强化学习算法与传统控制算法结合的控制效果，数据来源为 BAEK 等[24]和 Cui 等[25]有关轮足机器人运动控制的研究，结果如表 4 所示。由于硬件平台与环境不同，本文设计了与机器人质量参数相关的归一化基准进行性能的对比。恢复时间的对比采用无量纲恢复时间 T_n，其计算公式如式(10)所示。

$$T_n = \frac{t_s}{\sqrt{\frac{l}{g}}} \tag{10}$$

式中：t_s 为实际恢复时间，l 为摆动系统的质心高度，g 为重力加速度，$\sqrt{\frac{l}{g}}$ 为摆动系统的固有时间常数。

同时，将冲击下的最大轮力矩归一化为最大力矩与机器人总质量的比，以观测其能耗水平。俯仰角本身具有直接的物理意义，可以反映机器人在实际环境中的倾斜幅度和稳定性，因此在常见情况下保留原始角度即可，无需归一化处理。

表 4　本文算法与其他轮足机器人平台算法性能对比

Tab. 4　The performance of the proposed algorithm is compared with that of other wheeled robot platforms

算法	环境	冲击俯仰角/(°)	稳定俯仰角/(°)	俯仰角标准差	恢复时间 T_n	最大力矩/质量
Learning-Based ADP[25]	并联腿足机器人	20	4.50		18.25	0.89
DDPG + LQR[24]	Mujoco SATYRR		>9.00	0.048	20.58	
SAC + LQR[24]	Mujoco SATYRR		7.00	0.049	29.58	
本文：DDPG + PID	串联腿足机器人	20	8.59	0.050	20.54	0.81

与基于强化学习引导的 ADP(adaptive dynamic programming)控制算法作对比，在存在环境影响和受到最大角度为 20° 的冲击情况下，本文算法较 ADP 控制算法恢复速度慢但最大轮力矩较低，证明了奖励函数对长期控制过程中力矩的最小化存在优化与引导作用。与基于仿真的 DDPG + LQR(linear quadratic regulator，线性二次调节器)算法及 SAC + LQR 算法做对比，在无外力干扰的稳态状态对比下，可以看到本文算法的恢复速度和稳定速度与仿真结果相仿，最大角度优于 DDPG + LQR。由于仿真环境下机器人参数不一致以及电机 PID 参数不同导致的响应时间不一致，使得部分结果与仿真结果存在差异，但总体结果依然体现了算法的有效性。

3.3　结果分析与硬件约束讨论

在本研究中，DDPG 算法的映射与可视化解析实验表明，当状态空间发生剧烈变化时，阈值触发事件显著增加。约 3 000 步后策略逐渐趋于稳定，触发频率随之下降。这一现象说明硬件映射能够提供与策略收敛相关的观测信号。结合激活图与硬件信号进行对比，可以实现策略演化过程的可视化，并在一定程度上追溯其与硬件行为之间的对应关系。

同时，实验也反映出硬件和算法耦合带来的多方面限制。在机器人层面，在线 DDPG 对学习率、折扣因子和探索策略较为敏感，传感器延时对实时控制性能会产生影响。在控制系统层面，PL 端缺乏随机探索机制，可能导致策略在奖励函数引导下出现过拟合。同时当环境发生变化时，系统需要额外步数以重新调整。硬件资源限制迫使网络结构被压缩，这与 Critic 更新延迟和策略熵下降相关。虽然高精度时钟能够保障闭环实时性，但固定时序也限制了探索噪声的动态调节，使得在未知环境下的响应存在一定滞后。

综合认为，本研究提出的硬件–算法协同框架能够将训练过程映射为可测量事件，并据此构建“电路行为–网络动力学–技能表现”的解释模型。该框架提高了训练过程的可观测性和部分可解释性，为在资源受限的嵌入式平台上进一步调试和优化强化学习控制算法提供了新的参考。

4　结　语

1)本文针对强化学习在机器人部署中的算法–硬件割裂，构建 FPGA-ARM 轮足机器人闭环控制平台，部署基于硬件神经网络的 DDPG 算法。实验证明系统在 5 000 步内收敛，冲击后 6 s 内恢复稳定，最大俯仰角 <8°，能耗降低且 FPGA 功耗仅 3.329 W。

2)本文将训练过程映射为可观测物理信号流，建立策略权重更新与机器人状态的定量关联，实现可视化验证，并通过权值热力图揭示关注区域与长短期学习机理的关系，支持关键模块定向优化。研究表明，该方法具备硬件部署可行性与可解释强化学习价值，通过网络权值热力图定性解释了机器人技能学习过程中变化的趋势，可通过硬件映射与环境反馈实现鲁棒的策略进化与在线学习能力。

3)本文所提出的硬件可观测性强化学习架构可为智能制造、服务机器人与自动驾驶等高实时性、高能效应用提供新思路，并具有一定的工程参考价值。未来将探索动态精度量化、脉冲神经网络驱动的类生物探索机制及部分可重构 FPGA 架构，以在保持可解释性的同时突破性能瓶颈，推动机器人技能学习向生物启发的自适应范式发展。

参考文献

[1]BI Lanyue, ZHU Xiaoqing, RUAN Xiaogang, et al. Hardware

implementation of on-chip learning neural network on FPGA[C]// 2023 42nd Chinese Control Conference (CCC). Tianjin, China: IEEE, 2023: 8662. DOI: 10.23919/CCC58697.2023.10240711

[2] THACH W T, GOODKIN H P, KEATING J G. The cerebellum and the adaptive coordination of movement[J]. Annu Rev Neurosci, 1992, 15: 403. DOI: 10.1146/annurev.ne.15.030192.002155. PMID:1575449

[3] KULKARNI T D, NARASIMHAN, SAEEDI A, et al. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation [C]//Proceedings of the 30th International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc, 2016: 3682

[4] VEZHNEVETS A S, OSINDERO S, SCHAUL, T, et al. Feudal networks for hierarchical reinforcement learning[C]//International conference on machine learning. [S.l.]: PMLR, 2017: 3540

[5] 李永丰, 史静平, 章卫国, 等. 深度强化学习的无人作战飞机空战机动决策[J]. 哈尔滨工业大学学报, 2021, 53(12): 33. DOI:10.11918/202005108
LI Yongfeng, SHI Jingping, ZHANG Weiguo, et al. Maneuver decision of UCAV in air combat based on deep reinforcement learning[J]. Journal of Harbin Institute of Technology, 2021, 53(12): 33. DOI: 10.11918/202005108

[6] KADOKAWA Y, TSURUMINE Y, MATSUBARA T. Binarized P-network: Deep reinforcement learning of robot control from raw images on FPGA[J]. IEEE Robotics and Automation Letters, 2021, 6(4): 8545. DOI: 10.1109/LRA.2021.3111416

[7] GUTIÉRREZ C S V, JUAN L U S, UGARTE I Z, et al. Towards a distributed and real-time framework for robots: Evaluation of ROS 2.0 communications for real-time robotic applications[EB/OL]. (2018 - 09 - 07)[2025 - 12 - 19]. https://arxiv.org/abs/1809.02595. DOI:10.48550/arXiv.1809.02595

[8] 凤雷, 王宾涛, 刘冰, 等. 基于 FPGA 的深度强化学习硬件加速技术研究[J]. 计算机测量与控制, 2022, 30(6): 242. DOI:10.16526/j.cnki.11-4762/tp.2022.06.037
FENG Lei, WANG Bintao, LIU Bing, et al. Research on hardware acceleration technology of deep reinforcement learning based on FPGA[J]. Computer Measurement & Control, 2022, 30(6): 242. DOI: 10.16526/j.cnki.11-4762/tp.2022,30(6):242-247

[9] MENG Yuan, KINSNER M, SINGH D, et al. A software-hardware Co-optimized toolkit for deep reinforcement learning on heterogeneous platforms[EB/OL]. (2023-11-15)[2025-07-30]. https://arxiv.org/abs/2311.09445v1

[10] ZHANG Weiyi, NIU Liting, ZHANG Debing, et al. HW-ADAM: FPGA-based accelerator for adaptive moment estimation [J]. Electronics, 2023, 12(2): 263. DOI: 10.3390/electronics12020263

[11] LI Ziyu, GE Fen, ZHOU Fang, et al. An A3C deep reinforcement learning FPGA accelerator based on heterogeneous compute units[C]// 2022 IEEE 22nd International Conference on Communication Technology (ICCT). Nanjing, China: IEEE, 2022: 1521. DOI: 10.1109/ICCT56141.2022.10073229

[12] NAI Yufei, FANG Zhenghan, ZHAO Limeng. A design of reinforcement learning accelerator based on deep Q-iearning network[C]//2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT). New York, USA: IEEE, 2022: 1441

[13] 余奇. 基于 FPGA 的深度学习加速器设计与实现[D]. 合肥: 中国科学技术大学, 2016
YU Qi. Design and implementation of deep learning accelerator based on FPGA[D]. Hefei: University of Science and Technology of China, 2016

[14] LILLICRAP T P, SANTORO A, MARRIS L, et al. Backpropagation and the brain[J]. Nature Reviews Neuroscience, 2020, 21(6): 335

[15] 陈恺丰, 田博睿, 李和清, 等. 基于 DDPG 算法的双轮腿机器人运动控制研究[J]. 系统工程与电子技术, 2023, 45(4): 1144
CHEN Kaifeng, TIAN Borui, LI Heqing, et al. Research on motion control of two-wheeled legged robot based on DDPG algorithm[J]. Systems Engineering & Electronics, 2023, 45(4): 1144

[16] 孙文凯. 基于深度强化学习的四足机器人运动控制方法[D]. 济南: 山东大学, 2022
SUN Wenkai. Quadruped robot motion control method based on deep reinforcement learning[D]. Ji'nan: Shandong University, 2022

[17] 陈波, 张辉, 江一鸣, 等. 基于分层仿生神经网络的多机器人协同区域搜索算法[J]. 自动化学报, 2025, 51(4): 890
CHEN Bo, ZHANG Hui, JIANG Yiming, et al. A hierarchical bio-inspired neural network based multi-robot cooperative area search algorithm[J]. Acta Automatica Sinica, 2025, 51(4): 890. DOI: 10.16383/j.aas.c240458

[18] ZHANG Yumo. Accelerating autonomous vehicles: Harnessing FPGA power for deep learning advancements[J]. Applied and Computational Engineering, 2024, 53: 157. DOI: 10.54254/2755-2721/53/20241331

[19] 余子健. 基于 FPGA 的卷积神经网络加速器[D]. 杭州: 浙江大学, 2016
YU Zijian. Convolutional neural network accelerator based on FPGA[D]. Hangzhou: Zhejiang University, 2016

[20] HU Chanwei, HU Jiang, KHATRI S P. TD3lite: FPGA acceleration of reinforcement learning with structural and representation optimizations[C]//2022 32nd International Conference on Field-Programmable Logic and Applications (FPL). Yew York, USA: IEEE, 2022: 79

[21] LI Jindong, SHEN Guobin, ZHAO Dongcheng, et al. Firefly: A high-throughput hardware accelerator for spiking neural networks with efficient dsp and memory optimization[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2023, 31(8): 1178

[22] GUO Shasha, WANG Lei, WANG Shuquan, et al. A systolic snn inference accelerator and its co-optimized software framework[C]// Proceedings of the Great Lakes Symposium on VLSI. New York, USA: Association for Computing Machinery, 2019: 63. DOI: 10.1145/3299874.3317966

[23] YE Wujian, CHEN Yuehai, LIU Yijun. The implementation and optimization of neuromorphic hardware for supporting spiking neural networks with mlp and cnn topologies[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2022, 42(2): 448

[24] BAEK D, PURUSHOTTAM A, RAMOS J. Hybrid LMC: Hybrid learning and model-based control for wheeled humanoid robot via ensemble deep reinforcement learning [C]//2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). New York, USA: IEEE, 2022: 9347

[25] CUI Leilei, WANG Shuai, ZHANG Jingfan, et al. Learning-based balance control of wheel-legged robots[J]. IEEE Robotics and Automation Letters, 2021, 6(4): 7667

(编辑 吕雪梅)