软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(7):2538−2561 [doi: 10.13328/j.cnki.jos.006588]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

安全强化学习算法及其在 CPS 智能控制中的应用*

赵恒军[1,3], 李权忠[1,3], 曾 霞[1,3], 刘志明[2,3]

[1](西南大学 计算机与信息科学学院 软件学院, 重庆 400715)

[2](西北工业大学 智能嵌入式软件研究中心, 陕西 西安 710129)

[3](西南大学 软件研究与创新中心, 重庆 400715)

通信作者: 曾霞, E-mail: xzeng0712@swu.edu.cn

摘 要: 信息物理系统(cyber-physical system, CPS)的安全控制器设计是一个热门研究方向, 现有基于形式化方法的安全控制器设计存在过度依赖模型、可扩展性差等问题. 基于深度强化学习的智能控制可处理高维非线性复杂系统和不确定性系统, 正成为非常有前景的 CPS 控制技术, 但是缺乏对安全性的保障. 针对强化学习控制在安全性方面的不足, 围绕一个工业油泵控制系统典型案例, 开展安全强化学习算法和智能控制应用研究. 首先, 形式化了工业油泵控制的安全强化学习问题, 搭建了工业油泵仿真环境; 随后, 通过设计输出层结构和激活函数, 构造了神经网络形式的油泵控制器, 使得油泵开关时间的线性不等式约束得到满足; 最后, 为了更好地权衡安全性和最优性控制目标, 基于增广拉格朗日乘子法设计实现了新型安全强化学习算法. 在工业油泵案例上的对比实验表明, 该算法生成的控制器在安全性和最优性上均超越了现有同类算法. 在进一步评估中, 所生成神经网络控制器以 90%的概率通过了严格形式化验证; 同时, 与理论最优控制器相比实现了低至 2%的最优目标值损失. 所提方法有望推广至更多应用场景, 实例研究的方案有望为安全智能控制和形式化验证领域其他学者提供借鉴.

关键词: 强化学习; 智能控制; 信息物理系统; 安全验证; 工业油泵

中图法分类号: TP311

中文引用格式: 赵恒军, 李权忠, 曾霞, 刘志明. 安全强化学习算法及其在 CPS 智能控制中的应用. 软件学报, 2022, 33(7): 2538–2561. http://www.jos.org.cn/1000-9825/6588.htm

英文引用格式: Zhao HJ, Li QZ, Zeng X, Liu ZM. Safe Reinforcement Learning Algorithm and Its Application in Intelligent Control for CPS. Ruan Jian Xue Bao/Journal of Software, 2022, 33(7): 2538−2561 (in Chinese). http://www.jos.org.cn/1000-9825/6588.htm

Safe Reinforcement Learning Algorithm and Its Application in Intelligent Control for CPS

ZHAO Heng-Jun[1,3], LI Quan-Zhong[1,3], ZENG Xia[1,3], LIU Zhi-Ming[2,3]

[1](College of Computer and Information Science College of Software, Southwest University, Chongqing 400715, China)

[2](Centre for Intelligent and Embedded Software, Northwestern Polytechnical University, Xi'an 710129, China)

[3](Centre for Research and Innovation in Software Engineering, Southwest University, Chongqing 400715, China)

Abstract: The problem of safe controller design for cyber-physical systems (CPS) is a hot research topic. The existing safe controller design based on formal methods has problems such as excessive reliance on system models and poor scalability. Intelligent control based on deep reinforcement learning can handle high-dimensional nonlinear complex systems and uncertain systems, and is becoming a very promising CPS control technology, but it lacks safety guarantees. This study addresses the safety issues of reinforcement learning control by focusing on a case study of a typical industrial oil pump control system, and carries out research in designing new safe reinforcement learning algorithm and applying the algorithm in intelligent control scenario. First, the safe reinforcement learning problem of the

* 基金项目: 国家自然科学基金(61902325, 62032019, 61972385, 61732019, 61702425); 西南大学国家人才建设项目(SWU116007)

本文由"智能系统的分析和验证"专题特约编辑明仲教授、张立军教授和秦胜潮教授推荐.

收稿时间: 2021-09-05; 修改时间: 2021-10-14; 采用时间: 2022-01-10; jos 在线出版时间: 2022-01-28

industrial oil pump is formulated, and simulation environment of the oil pump is built. Then, by designing the structure and activation function of the output layer, the neural network type oil pump controller is constructed to satisfy the linear inequality constraints of the oil pump switching time. Finally, in order to better balance the safety and optimality control objectives, a new safe reinforcement learning algorithm is designed based on the augmented Lagrange multiplier method. Comparative experiment on the industrial oil pump shows that the controller generated by the proposed algorithm surpasses existing algorithms in the same category, both in safety and optimality. In further evaluation, the neural network controllers generated in this study pass rigorous formal verification with probability of 90%. Meanwhile, compared with the theoretically optimal controller, neural network controllers achieve a loss of optimal objective value as low as 2%. The method proposed in this study is expected to be extended to more application scenarios, and the case study scheme is expected to be referenced by other researchers in the field of intelligent control and formal verification.

Key words: reinforcement learning; intelligent control; cyber-physical system; safety verification; industrial oil pump

信息物理系统(cyber-physical system, CPS)是一个深度结合计算机、物理硬件与网络的多维度智能化的复杂系统, 其应用领域已深入到社会生产和人民生活的各个方面. 诸如航天器控制系统、汽车自动驾驶系统之类的信息物理系统对安全性有极高的要求, 称为安全攸关系统(safety-critical system), 一旦系统发生故障, 将造成严重的经济损失和不可挽回的人员伤亡. CPS 的安全性研究是当前的热点课题, 其中, 如何设计 CPS 的安全控制器, 是尤为重要的研究问题.

目前, 针对 CPS 的安全控制器生成主要采用形式化方法: 一类方法基于可达集分析[1,2]或障碍函数生成[3], 结合传统优化方法、数值计算等生成具有安全保障的系统控制器; 另一类通过指定控制器函数模板或不变式模板, 将安全控制器生成问题编码为一阶逻辑公式, 利用符号计算方法生成满足安全约束的控制器[4]. 这两类方法在理论研究层面已经解决了部分规模有限的 CPS 安全控制问题, 但尚无法推广到一般实际问题中, 核心瓶颈源于两个方面: 形式化方法依赖刻画 CPS 物理行为的精确数学模型, 例如, 通常需要给定微分方程组或者差分方程组描述系统物理行为, 然而在实际问题中, 复杂 CPS 系统由于客观环境的不确定性等因素, 获取其准确的数学模型非常困难; 再者, 基于可达集计算、障碍函数生成或不变式生成的传统形式化方法计算成本较高, 对于复杂 CPS 的安全性分析与验证, 往往无法在有效时间内得到结果. 这导致传统形式化方法在实际 CPS 安全控制器设计中的应用较为有限. 针对这个问题, 有研究者提出, 从数据中挖掘系统的模型[5]. 本文利用人工智能技术以数据驱动的、不依赖严格数学模型的方式生成 CPS 的安全控制器, 规避形式化方法对数学模型的依赖和高计算复杂性.

随着人工智能产业的飞速发展, 人工智能的理论与技术成果被认为非常有可能在控制领域得到集成, 智能控制将是人工智能及相关前沿技术的综合体现. 例如, 文献[6]针对航天控制场景指出: “航天控制系统具有飞行环境不确定、故障模式不确定、外部干扰不确定、自身模型不确定、飞行任务不确定等特有属性”, “智能控制技术将成为实现智能航天的必然选择”. 与传统控制技术相比, 智能控制能够应用于具有高度非线性、强不确定性的复杂系统, 能够处理多重优化目标, 是一种有强大生命力的新型控制技术. 但是智能控制缺少严格的安全性、稳定性等性能保障, 这是其在安全攸关 CPS 领域取得广泛应用的主要障碍之一.

本文研究安全强化学习智能控制方法, 并运用于安全攸关 CPS 的控制器生成, 致力于保障系统的安全性质; 与此同时, 能达到系统最优控制目标. 本文围绕文献[7]中提出的一个典型 CPS 案例——工业油泵控制系统, 开展安全强化学习算法研究和智能控制应用研究. 该工业油泵控制系统已在一系列工作中被作为基准案例广泛研究[7−11], 其系统行为具有高度非线性性, 控制目标具有安全、鲁棒和最优等多重性, 给控制器的生成, 尤其是基于形式化方法的生成带来很大困难, 因而, 利用该案例探索智能控制方法的应用潜能显得十分必要; 另外, 该案例具有显式的理论最优解[9], 可以作为智能算法的评判标准, 能够为本文的对比研究提供极大便利. 本文利用深度强化学习方法为工业油泵控制系统设计一个以安全性和最优性为目标的神经网络控制器: 首先, 利用约束马尔可夫决策过程形式化了工业油泵系统的安全强化学习问题; 随后, 搭建了油泵系统的仿真环境, 设计了和环境交互过程中获得的收益值和损耗值; 然后, 搭建了用于控制油泵开关时间点的神经网络控制器, 为满足油泵开关时间的固有物理约束, 对神经网络输出层的结构和激活函数进行了巧妙设计, 使得其输出满足线性不等式约束; 最后, 通过收集控制器和仿真环境的交互行为构建训练数据集, 不断训练更

新神经网络控制器的参数, 以期获得更大的收益和更好的安全性.

为了在训练过程中更好地权衡安全性和最优性目标, 本文提出了基于增广拉格朗日乘子方法的新型安全强化学习算法. 现有的解决无模型情况下强化学习安全性的方法主要是拉格朗日乘子法[12−15], 其基本思想是将安全强化学习中带有安全约束的优化问题借助拉格朗日乘子转化为无约束优化问题. 在实践应用中, 基于拉格朗日乘子法的安全强化学习算法存在两个问题.

1) 拉格朗日乘子法中, 乘子λ按照固定的学习率线性增长, 算法效果在很大程度上受学习率大小的影响: 学习率过小, λ增长缓慢, 会导致对安全违背的惩罚不足; 学习率过大, 则λ增长过快, 会导致算法不稳定收敛困难, 对安全性和最优性的优化都达不到很好的效果. 选择合适的学习率是一个困难问题.
2) 拉格朗日乘子法中, 乘子λ的初值一般设为 0, 导致在训练初期, 优化算法更偏重提高收益而不是降低损耗, 容易陷入不安全的局部最优解, 称这种现象为惩罚“冷启动”.

本文提出结合增广拉格朗日乘子法设计新型的安全强化学习算法, 最重要的改进是在拉格朗日函数中引入二次惩罚项, 其作用主要有两点.

1) 在增广拉格朗日乘子法中, 乘子λ的学习率不再是一个常量, 而是根据当前二次惩罚项的惩罚因子大小动态调整, 乘子项和二次惩罚项协同作用, 使得对乘子λ的训练更快更稳定地收敛.
2) 增广拉格朗日函数中, 二次惩罚项的惩罚因子初值为非 0 正值, 相当于对安全违背的惩罚是“热启动”, 即通过二次惩罚项, 在训练初期就对安全违背施以较大惩罚, 从而使算法对最优策略的搜寻尽快地趋于安全区域.

以上两点作用, 使得增广拉格朗日安全强化学习算法能够更好地权衡安全性和最优性训练目标, 并且表现出更快、更稳定的收敛效果. 在油泵控制案例上与传统拉格朗日等方法的对比实验结果表明, 本文所提算法在追寻较大收益和保持安全性之间取得了非常好的平衡效果, 安全性和最优性均优于同类算法. 此外, 对所提算法生成的神经网络控制器开展了进一步的安全性和最优性评估: 在安全性评估方面, 所生成神经网络控制器在随机测试中, 以 90%的概率通过了基于 SMT 求解器的严格形式化验证; 在最优性方面, 神经网络控制器所能获得的控制目标最优值和文献[9]中理论最优值相比仅损失了 2%.

综上, 本文将强化学习应用于 CPS 的安全控制问题, 主要贡献在于:

- 完整地研究了一个 CPS 控制领域的工业油泵控制经典案例, 实验表明了智能控制器在该案例上的高安全性和最优性, 展示了智能控制在安全攸关 CPS 领域的应用前景, 为 CPS 安全控制器设计提供了相对纯形式化方法更加轻量级的、数据驱动的方法选择.
- 改进了现有基于拉格朗日乘子法的安全强化学习方法, 提出了增广拉格朗日安全强化学习算法 ALM-DDPG. 对比实验表明了所提算法在平衡安全目标和最优目标上的优越性能, 有望推广至其他约束强化学习应用场景.
- 提出了在神经网络中编码线性不等式约束的技巧, 使得神经网络的输出自然地满足线性不等式约束, 有望推广至其他含有线性约束的神经网络应用场景.
- 基于 SMT 求解器形式化验证方法对所生成神经网络控制器进行了安全性评估, 进一步验证了所提方法在安全控制器生成上的效果, 有望为智能安全控制和智能系统形式化验证领域贡献一个基准案例, 为其他研究人员所借鉴.

本文第 1 节主要介绍相关工作. 第 2 节介绍安全强化学习和工业油泵控制案例的背景知识. 第 3 节构建工业油泵控制案例的安全强化学习模型. 第 4 节重点介绍了基于增广拉格朗日方法的新型安全强化学习算法. 第 5 节通过对比实验展现了所提算法的良好性能. 第 6 节对强化学习所生成控制器进行了安全性和最优性评估. 最后, 全文在第 7 节中进行了总结和展望.

1 相关工作

本文主要利用强化学习方法开展安全攸关CPS控制器生成问题研究, 故相关工作部分将重点分析和比较基于智能方法的安全控制器生成方法. 与此同时, 作为 CPS 控制系统在实际工业界的一个典型案例, 油泵控制系统也是多年来被 CPS 和形式化方法领域广为研究的一个内容, 是本文的主要研究对象, 本节将简要介绍关于油泵控制系统的相关研究工作.

在安全攸关 CPS 的控制器设计中, 如何利用智能技术应对环境不确定性, 不依赖数学模型, 以数据驱动的方式生成具有安全保障的系统控制器, 一直是具有挑战性的热点研究课题. 其中一类重要的方法是安全强化学习, 即利用强化学习技术学习满足给定安全性约束的最优控制策略. 近5年来, 学术界在安全强化学习领域已经取得不少成果, 概括来讲, 可以分为如下几类.

- 基于安全防护层的方法[16−18]. 其基本思想是, 在强化学习框架之外套一层安全层. 作用是对强化学习产生的危险动作进行修正, 需要额外学习模型的敏感性(sensitivity), 即当前动作对安全性质的影响程度.
- 基于策略改进的方法[19−22]. 其基本思想是, 从一个初始的较为安全但是次优的策略出发, 不断地改进之前的策略, 需要给定初始的安全策略.
- 基于模仿学习的方法[23]. 其基本思想是, 通过模仿人的安全动作规避不安全行为. 模仿过程需要较多的人为干预.
- 基于定理证明的方法[24]. 其基本思想是, 通过定理证明的方式保障强化学习的安全性. 定理证明自动化程度较低, 且依赖系统模型的形式化描述.
- 基于Lyapunov函数或障碍函数的方法[25−29]. 其基本思想是, 借助经典控制中的安全性或稳定性理论, 通过安全性或稳定性凭证函数, 如 Lyapunov 函数, 或障碍函数等证明强化学习的安全性. 针对用以保障系统安全的凭证函数类型不同, 相关工作可分为控制李雅普诺夫函数(control Lyapunov function, CLF)生成以及控制障碍函数(control barrier function, CBF)生成. 对于这一类方法, Lyapunov 函数或者障碍函数生成复杂度较高, 而且依赖于准确的系统数学模型.
- 基于约束优化的方法[12−15]. 其基本思想是, 将系统的安全性需求表示为强化学习训练过程中对策略的约束条件, 即一个约束优化问题, 然后借助拉格朗日乘子法等经典的约束优化求解理论和方法进行训练迭代. 此类方法对系统的要求最少, 实现最简单, 因而具有最广的适用范围, 是本文所提算法借鉴和改进的基础. 但该方法存在两点不足, 即合适的拉格朗日乘子学习率的选择困难问题和拉格朗日乘子初值为0带来的安全惩罚“冷启动”问题. 本文提出利用增广拉格朗日乘子方法进行改进, 利用二次惩罚项对安全约束违背的惩罚进行“热启动”; 同时, 利用二次惩罚项的惩罚因子动态调节拉格朗日乘子的学习率, 非常好地平衡了安全性和最优性优化目标, 并使算法表现出更好的收敛性和稳定性.

本文的主要研究案例——工业油泵控制系统是一个高度复杂的非线性 CPS, 其控制器设计是一个非凸的多目标最优控制问题, 多年来引起许多国内外课题组的关注. 对于生成具有安全保证的最优油泵控制器问题, 主流的方法是利用形式化方法展开的. 简要来讲, 解决方法主要有两类: 第 1 类基于时间自动机(timed automata)建模和求解安全控制器, 第 2 类基于一阶逻辑的量词消去方法求解安全控制器. 在文献[7]中, 作者使用时间博弈自动机(timed game automata)对油泵系统进行建模, 并应用工具 UPPAAL-TIGA 来合成近似最优的控制器. 在文献[10,11]中, 作者进一步提出使用能量时间自动机(energy timed automata)解决油泵的最优控制问题. 在文献[9]中, 作者将最优控制器生成问题转化为量词消去问题, 接着将量词消去与离散数值计算相结合, 最终给出了在一定条件下的理论最优控制器. 该方法的结果相较于文献[7]的结果进一步提升了最优控制效果. 上述基于形式化方法的控制器生成虽然能够保证系统的安全性, 但是求解难度较高, 且依赖于系统的形式化模型, 难以推广到更一般的系统. 本文提出一种基于深度强化学习的安全控制器生成方法, 这是一个轻量级的、数据驱动的控制器合成方法, 能够处理更一般的非线性系统, 不依赖精确的系统模型, 可解决多

目标优化问题. 本文的实验结果表明, 该方法在保障系统安全性的同时, 已达到近乎文献[9]中的理论最优控制效果. 在利用智能方法生成油泵控制器方面, Jha 等人在文献[8]中做了更早的尝试, 该工作对初始状态进行随机采样, 从单个初始状态的最佳切换序列泛化到所有初始状态集合的最佳切换控制器, 其泛化过程应用了 PAC 学习思想, 但只尝试了半平面形式的学习类; 本文所提强化学习智能控制可以生成表示能力更强的神经网络控制器.

2 预备知识

本节介绍强化学习的基本概念、术语和符号, 同时简要介绍本文研究的工业油泵安全控制器设计问题.

2.1 安全强化学习问题

强化学习(reinforcement learning, RL)[30]是智能体通过与环境交互, 学习得到最大化长期期望收益的序列决策策略的机器学习技术. 如图 1 所示, 在任意整数时刻 $t\geqslant 0$, 智能体观察到系统状态 s_t, 采取动作 a_t, 获得 t 时刻的瞬时收益 r_t.

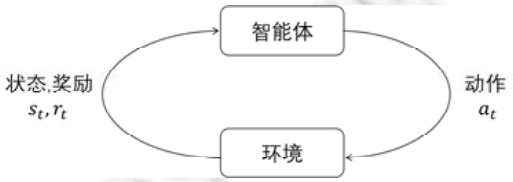


图 1 强化学习中的智能体与环境交互过程

强化学习可以通过马尔可夫决策过程[30]建模.

定义 1(马尔可夫决策过程). 一个马尔可夫决策过程(Markov decision process, MDP)可以表示为六元组 $\mathcal{M}=(S,A,P,R,\mu,\gamma)$, 其中,

- S: 表示(有穷或无穷)状态集合.
- A: 表示(有穷或无穷)动作集合.
- $P:S\times A\to\mathcal{P}(S)$: 表示状态转移函数, $\mathcal{P}(S)$表示 S 上的概率分布全体, $P(s'|s,a)$表示在状态 s 下采取动作 a 迁移到状态 s'的概率或概率密度; 在确定性系统中, $P(s'|s,a)$退化为单点分布, 此时, P 可记为函数:
$$P:S\times A\to S.$$
- $R:S\times A\times S\to\mathbb{R}$: 表示收益函数; s 状态下, 执行迁移(s,a,s')的瞬时收益为 $r=R(s,a,s')$.
- $\mu\in\mathcal{P}(S)$: 表示系统初始状态 s 的概率分布.
- $\gamma\in(0,1)$: 表示计算未来长期收益的折扣因子.

给定 MDP, 策略π定义为$\pi:S\to\mathcal{P}(A)$, $\mathcal{P}(A)$表示动作集 A 上的概率分布全体, $\pi(a|s)$表示在状态 s 下采取动作 a 的概率或概率密度. 当$\pi(a|s)$退化为单点分布时, 称π为确定性策略, 此时π可记为函数: $\pi:S\to A$. 在深度强化学习中, 策略π表示为参数化的深度神经网络, 记为π_θ, 其中, θ为神经网络的待定参数, 如权重、偏置等. 给定 MDP 和策略π, 轨迹$\tau=(s_0,a_0,s_1,a_1,\ldots,s_t,a_t,\ldots)$是一状态动作序列, 其中, $s_0\sim\mu$服从初始分布. 在任意时刻 t, a_t 由 $\pi(\cdot|s_t)$决定, s_{t+1} 由 $P(\cdot|s_t,a_t)$决定. 所有轨迹构成的集合服从由 P 和π决定的概率分布. 在不引起歧义的情况下, 在后文中, 将用π表示轨迹τ服从的概率分布.

为度量策略π的优劣, 定义轨迹τ的累积折扣收益为 $R(\tau)=\sum_{t=0}^{\infty}\gamma^t r_t$, 其中, $r_t=R(s_t,a_t,s_{t+1})$. 则策略π的期望收益为 $J_r(\pi)=E_{\tau\sim\pi}[R(\tau)]$. 强化学习的目的即为求得最优策略$\pi^*=\arg\max_\pi J_r(\pi)$. 对于参数化策略, 记$\pi^*$对应的参数为$\theta^*$.

强化学习问题中有一类安全强化学习, 其目的是寻求策略π最大化累积折扣收益, 同时使得系统的行为满足给定的安全约束. 安全强化学习可以通过约束马尔可夫决策过程[14]建模.

定义 2(约束马尔可夫决策过程). 一个约束马尔可夫决策过程(constrained MDP, CMDP)可以表示为七元组 $\mathcal{CM}=(S,A,P,R,C,\mu,\gamma)$, 其中, $C:S\times A\times S\to\mathbb{R}$表示损耗函数, 其他元素的定义同定义 1.

给定 CMDP, 定义轨迹τ的累积折扣损耗为 $C(\tau)=\sum_{t=0}^{\infty}\gamma^t c_t$, 其中, $c_t=C(s_t,a_t,s_{t+1})$, 则策略π的期望损耗为 $J_c(\pi)=E_{\tau\sim\pi}[C(\tau)]$. 从而可以给出如下的安全强化学习问题定义:

定义 3(安全强化学习). 一个安全强化学习(safe reinforcement learning, SRL)问题是指对给定的 CMDP, 求解约束优化:

$$\left.\begin{aligned}&\max_{\pi} J_r(\pi),\\ &\text{s.t. } J_c(\pi)\leqslant\alpha\end{aligned}\right\} \tag{1}$$

其中, α为容许的损耗上界. 对于参数化策略, 实际求解的是最优策略π^*对应的参数θ^*.

安全强化学习问题常用的求解方法是拉格朗日乘子法[14], 即通过引入拉格朗日乘子λ构造拉格朗日函数 L, 将公式(1)转化为如公式(2)所示的无约束优化问题.

$$\min_{\lambda\geqslant 0}\max_{\pi} L(\pi,\lambda)=\min_{\lambda\geqslant 0}\max_{\pi}[J_r(\pi)-\lambda(J_c(\pi)-\alpha)] \tag{2}$$

然后, 通过对π和λ进行交叉迭代求解. 这种方法的缺点是:

1) 在对λ进行迭代更新时, 更新的梯度为 $J_c(\pi)-\alpha$, 更新的步长为某选定的常量学习率, 步长过小, λ增长缓慢, 会导致对约束的满足性变弱; 步长过大, 则λ增长过快, 会导致算法不稳定难以收敛. 合适的乘子迭代步长的选取, 是一个困难问题.
2) λ的迭代初值一般设为 0, 从而乘子项$\lambda(J_c(\pi)-\alpha)$的初值为 0, 导致对安全约束违背的惩罚是“冷启动”的. 即训练初期惩罚力度很小, 算法容易陷入不安全的局部最优解.

在经典的约束优化方法中, 有另外一类增广拉格朗日乘子法(augmented Lagrange method, ALM)[31], 其特点是将惩罚函数法和拉格朗日乘子法的思想相结合, 构造含有二次惩罚项的增广拉格朗日函数. 对问题(1), 在损耗约束为等式型约束的特殊情况下, 据文献[31], 其增广拉格朗日函数可定义为

$$L(\pi,\lambda,\rho)=J_r(\pi)-\lambda(J_c(\pi)-\alpha)-\frac{\rho}{2}(J_c(\pi)-\alpha)^2 \tag{3}$$

其中, $\rho>0$ 为二次惩罚项$(J_c(\pi)-\alpha)^2$ 的惩罚因子. 对比公式(2)和公式(3), 可以总结增广拉格朗日乘子法的两个主要优势: 1) 在基于公式(3)的优化求解步骤中, λ的迭代更新率不再是一个常量, 而是惩罚因子ρ, 其随着优化过程的进行不断增长, 相当于对λ的学习率进行了动态调节, 乘子项和二次惩罚项协同作用, 使得算法表现出更好的收敛性和稳定性; 2) 惩罚因子ρ的初值大于 0, 结合公式(3)可知, 此时对安全约束违背的惩罚是“热启动”, 即在训练的初期就以较大的倾向去优化安全性, 避免陷入不安全的局部最优解. 鉴于增广拉格朗日乘子法的上述优势, 本文基于公式(3)设计实现新型的安全强化学习算法, 以期在安全性和最优性控制目标间取得更好的权衡.

2.2 工业油泵控制案例

本文所研究的工业油泵案例是 CPS 控制器设计领域的一个经典案例, 在一系列工作中被作为基准案例研究[7−11]. 整个油泵控制系统的结构如图 2 所示, 由机器、油库、油泵和储油器组成. 机器耗油过程表现为长度为 20 s 的消耗周期, 一个周期内的耗油率曲线如图 3 所示. 当油泵工作时, 将以 2.2 L/s 的泵油率向储油器中补充油量. 该系统的控制目标是, 通过在特定时间点打开或关闭油泵, 保障如下的安全性和最优性.

- R_s(安全性): 系统可以任意长时间运行, 同时, 对于任意时刻 t, 储油器中的油量 $v(t)$始终保持在安全区间$[V_{\min},V_{\max}]$, 其中, $V_{\min}$=4.9 L, $V_{\max}$=25.1 L.
- R_o(最优性): 最小化系统的平均累积油量, 即最小化:

$$\lim_{T\to\infty}\frac{1}{T}\int_0^T v(t)\mathrm{d}t \tag{4}$$

由于系统损耗、测量误差等因素, 上述两个目标的达成还要求满足如下限制条件.

- R_{pl}(泵延迟): 对油泵的任何两次连续控制操作(开或者关)之间, 一定存在至少 2 s 的时间延迟.
- R_r(健壮性): 同时, 应考虑系统的不确定性:
  - 耗油率波动: 当耗油率不为 0 时的波动, 最高可达 f=0.1 L/s;
  - 油量测量误差: 最高可达σ=0.06 L;
  - 开关操作响应时间误差: 最高可达δ=0.015 s.

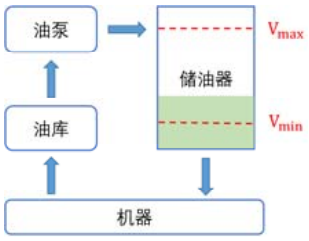


图 2 工业油泵控制系统

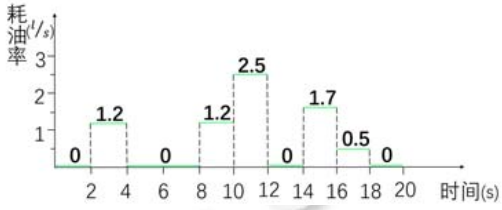


图 3 工业油泵单周期耗油率曲线

本文将利用安全强化学习技术, 在不依赖系统数学模型的基础上, 仅利用油量的仿真数据生成上述油泵系统的神经网络形式控制器, 使得 R_s, R_o, R_{pl}, R_r 需求同时得到满足. 本文的研究方法仅仅假设存在这样一个油泵仿真系统: 能够通过它, 周期性地获取每周期初始的油量值, 以及每个周期油量变化曲线对安全性的违背程度和油量积分的大小. 此处特别指出的是, 由于 R_{pl} 约束的存在, 在一个周期 20 s 内, 最多容许对油泵进行 5 次开-关操作. 在本文中, 参考文献[7,9], 将开-关次数限定为 2 次, 这意味着一个周期内存在 4 个控制时间点(t_1,t_2,t_3,t_4), 其中, t_1, t_3 为打开时间点, t_2, t_4 为关闭时间点.

3 工业油泵案例的安全强化学习问题构建

为了应用深度强化学习技术为上述工业油泵系统生成安全最优控制器, 我们首先构造其 CMDP 模型. 为此,将油泵视为强化学习问题中的环境, 将油泵控制器视为智能体; 此外, 利用油泵耗油的周期性, 将每个周期的油泵开关控制及所产生的油量变化视为智能体与环境的一次交互. 具体来说, 控制器根据每个周期开始时的油量决定该周期内的油泵开关时间点, 产生控制动作影响周期内的油量变化, 同时对系统的安全性和最优性产生影响, 完成一次交互; 周期末尾的油量作为下一次交互的基础. 如图 4 所示, 初始时刻油量为 s_0, 控制器产生控制动作, 即一个周期内两次开关的 4 个时间点(t_1,t_2,t_3,t_4), 作用于油泵使得油量在周期末迁移到 s_1, 完成一步交互.

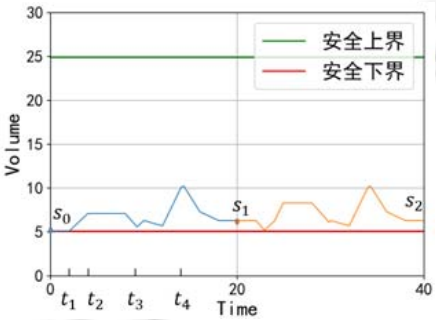


图 4 工业油泵中的智能体与环境的交互过程

3.1 工业油泵案例的CMDP模型

在上述约定下, 油泵的 CMDP 模型可以定义为$\mathcal{CM}$=(S,A,P,R,C,μ,γ), 其中,

- $S=\{s\mid V'_{\min}\leqslant s\leqslant V'_{\max}\}$. s 为每周期开始时系统储油器中的油量, $V'_{\min}=V_{\min}+0.2=5.1$, $V'_{\max}=V_{\max}-0.2=$

24.9; 此处参考文献[9], 将健壮性要求 R_r 中测量误差和响应时间误差对油量和安全性的影响, 以 0.2 的裕度硬编码到安全上、下界中.

- $A=\{a|a=(t_1,t_2,t_3,t_4),0\leqslant t_i\leqslant 20$ 且满足 R_{pl} 中 2 s 延迟的限制条件$\}$. 本文考虑每个周期中对油泵进行两次开关控制, 共需 4 个控制时间点, 故将这 4 个时间点构成的四元组作为一个控制动作 a, 满足 R_{pl} 中关于油泵开关时间延迟约束的动作 a 的全体构成动作集合 A.
- $P:S\times A\to S$. 令 s 表示某周期开始时储油器中的油量, a 表示该周期内控制油泵开关的 4 个时间点, 则 $s'=P(s,a)$表示由图 3 所示耗油率(不考虑波动)和 2.2 L/s 的泵油率所决定的该周期末时刻储油器中的油量, 详细计算方法见下文.
- 收益函数 R. 对于每周期单步交互形成的状态迁移(s,a,s'), 收益 $R(s,a,s')$由该周期内的油量积分确定, 详细计算方法见下文.
- 损耗函数 C. 对于每周期单步交互形成的状态迁移(s,a,s'), 损耗 $C(s,a,s')$由该周期内的油量对安全上界和下界的违背情况决定, 详细计算方法见下文.
- γ在本文中取为常值 0.9.

另, 本文考虑油泵的确定性控制策略, 即$\pi:S\to A$.

3.2 油泵仿真环境

为了实现控制器与油泵之间的交互, 我们搭建了油泵的仿真环境, 该仿真环境接收控制器产生的控制动作,即用于每一周期油泵开关的时间点, 根据耗油率和泵油率仿真一个周期内的油量变化, 从而获取该周期的收益、损耗以及下一周期的初始油量. 换言之, 仿真环境具体实现了上节所定义 CMDP 的迁移函数 P、收益函数 R 和损耗函数 C.

(1) 迁移函数 P 的实现

给定初始油量 s 和一个周期内的控制动作, 即 4 个开关时间点 $a=(t_1,t_2,t_3,t_4)$, 依据 20 s 内的耗油率曲线(不考虑耗油率波动, 如图 3 所示):

$$g(t)=\begin{cases}1.2, & 2\leqslant t\leqslant 4\text{或}8\leqslant t\leqslant 10\\ 2.5, & 10\leqslant t\leqslant 12\\ 1.7, & 14\leqslant t\leqslant 16\\ 0.5, & 16\leqslant t\leqslant 18\\ 0, & \text{其他}\end{cases}$$

和由 a 确定的泵油率曲线:

$$h(t)=\begin{cases}2.2, & t_1\leqslant t\leqslant t_2\text{或}t_3\leqslant t\leqslant t_4\\ 0, & \text{其他}\end{cases},$$

可得周期内的油量变化曲线 $v(t)$:

$$v(t)=s+\int_0^t(h(x)-g(x))\mathrm{d}x \tag{5}$$

则 $P(s,a)=v(20)$. 如图 5 中蓝色实线所示, 系统的初始油量为 $s=5.1$, 4 个开关时间节点为(8,10,18,20), $v(t)$曲线以 0.01 s 为仿真步长, 按公式(5)计算所得. 需要指出的是, 油量的变化曲线中出现负的油量值不符合物理实际, 可将所有的负值油量设为 0. 本文考虑到这一点对强化学习算法的效果不存在实质影响而未进行处理.

(2) 收益函数 R 的实现

根据最优需求 R_o, 需依据公式(4)、公式(5)计算并最小化油量积分. 当油量曲线过低时, 不安全的油量曲线反而产生较小的积分值. 为在油量积分计算时同时兼顾安全需求 R_s, 对 $v(t)$关于 $V'_{\min}$ 进行对称操作得到 $\hat{v}(t)$:

$$\hat{v}(t)=\begin{cases}v(t), & V'_{\min}\leqslant v(t)\\ 2V'_{\min}-v(t), & V'_{\min}>v(t)\end{cases}.$$

如图 5 的橙色虚线所示. 然后计算每次交互的周期累积油量 $\int_0^{20}\hat{v}(t)\mathrm{d}t$. 如图 5 阴影部分所示. 考虑到公式(1)中

强化学习的目标是最大化长期期望收益，和最小化长期累积油量 R_o 需求相矛盾，所以进一步将收益函数 R 变换成与累积油量成负相关的函数并进行适当的缩放:

$$R(s,a,s')=\frac{1}{50}\left(20V'_{\max}-\int_0^{20}\hat{v}(t)\mathrm{d}t\right) \tag{6}$$

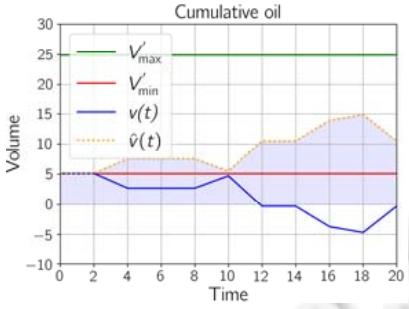


图 5　一个周期的油量曲线图和累积油量图

(3) 损耗函数 C 的实现

按照健壮性 R_r 限制条件，设带波动的耗油率为 $\tilde{g}(t)$，对于任意的 $0\leqslant t\leqslant 20$,

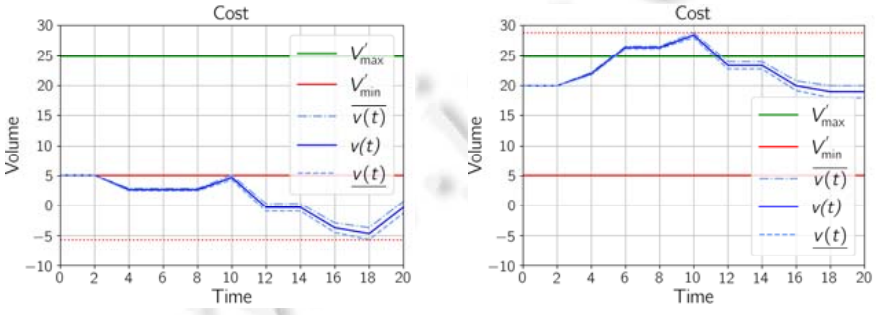
$$\tilde{g}(t)=\begin{cases}1.2\pm0.1, & 2\leqslant t\leqslant 4\text{或}8\leqslant t\leqslant 10\\ 2.5\pm0.1, & 10\leqslant t\leqslant 12\\ 1.7\pm0.1, & 14\leqslant t\leqslant 16\\ 0.5\pm0.1, & 16\leqslant t\leqslant 18\\ 0, & \text{其他}\end{cases},$$

其中，±0.1 表示 $\tilde{g}(t)$ 的取值可以在正负 0.1 的范围内任意波动. 则在初始油量为 s、控制动作为 a 且耗油率波动情况下的单周期油量变化为 $\tilde{v}(t)=s+\int_0^t(h(x)-\tilde{g}(x))\mathrm{d}x$.

对任意 $0\leqslant t\leqslant 20$，我们定义 $\overline{v(t)}=\sup\tilde{v}(t)$, $\underline{v(t)}=\inf\tilde{v}(t)$，分别表示 $\tilde{v}(t)$ 的上界和下界. 则单步交互的损耗值定义为 $\overline{v(t)}$ 超出安全上界 $V'_{\max}$ 与 $\underline{v(t)}$ 低于安全下界 $V'_{\min}$ 的部分之和，即

$$C(s,a,s')=\max(\overline{v(t)}-V'_{\max},0)+\max(V'_{\min}-\underline{v(t)},0) \tag{7}$$

显然, C 为非负函数. 在图 6(a)中，初始油量为 5.1, 4 个控制时间节点为(8,10,18,20), 3 条蓝色油量变化折线中，上方点划虚线为 $\overline{v(t)}$，下方虚线为 $\underline{v(t)}$，中间的蓝色实线为 $v(t)$，下方水平红虚线和 $V'_{\min}$ 水平线的间距为单步损耗值；在图 6(b)中，初始油量为 20，4 个控制时间节点为(2,6,8,10)，上方水平红虚线和 $V'_{\max}$ 水平线的间距为单步损耗值.



(a) 违背安全下界　　(b) 违背安全上界

图 6　损耗函数示意图

3.3 神经网络控制器的构建

为应用深度强化学习技术训练油泵的控制器, 需要根据油泵控制的时间延迟需求 R_{pl} 搭建合适的神经网络. 具体来说, 油泵控制器的动作 $a=(t_1,t_2,t_3,t_4)$需满足:

$$\begin{cases}0\leqslant t_1,t_2,t_3,t_4\leqslant 20\\ t_1\geqslant 2,\ t_2-t_1\geqslant 2,\ t_3-t_2\geqslant 2,\ t_4-t_3\geqslant 2\end{cases} \tag{8}$$

一般神经网络的输出无法天然地满足此类线性约束条件, 为此做以下变换:

$$\begin{cases}\Delta t_1=t_1-2\\ \Delta t_2=t_2-t_1-2\\ \Delta t_3=t_3-t_2-2\\ \Delta t_4=t_4-t_3-2\\ \Delta t_5=20-t_4\end{cases} \tag{9}$$

则公式(8)可以转化为

$$\begin{cases}0\leqslant \Delta t_1,\Delta t_2,\Delta t_3,\Delta t_4,\Delta t_5\leqslant 12\\ \Delta t_1+\Delta t_2+\Delta t_3+\Delta t_4+\Delta t_5=12\end{cases} \tag{10}$$

原动作(t_1,t_2,t_3,t_4)与$(\Delta t_1,\Delta t_2,\Delta t_3,\Delta t_4,\Delta t_5)$间的关系可形象地表示为图 7.

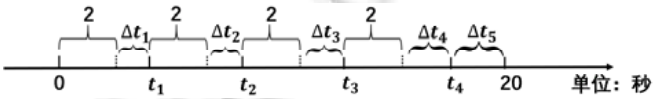


图 7 控制动作变换示意图

根据公式(10), 将油泵 CMDP 的动作空间 A 转化为动作空间 A':

$$A'=\left\{a'\mid a'=\frac{1}{12}(\Delta t_1,\Delta t_2,\Delta t_3,\Delta t_4),\Delta t_i\text{满足公式(10)}\right\}.$$

此外, 将油泵 CMDP 的状态空间 S 归一化为[−1,1], 得到状态空间 S', 从而定义如图 8 所示的油泵神经网络控制器$\pi_\theta: S'\to A'$. 其中, θ为神经网络的待定参数. 神经网络控制器π_θ的输入层置一个神经元表示初始油量 s, 中间层采用 ReLU 激活函数. 为保证π_θ的输出动作满足 A'的要求, 在其输出层置 5 个神经元, 使用 Softmax 激活函数, 用 $\pi_\theta^{(i)}(s)$ 表示π_θ的第 i 个输出, 则根据 Softmax 函数的性质有 $\sum_{i=1}^{5}\pi_\theta^{(i)}(s)=1$, $0\leqslant\pi_\theta^{(i)}(s)\leqslant 1$, i=1,2,…, 令 $\Delta t_i=12\pi_\theta^{(i)}(s)$, i=1,2,…,5, 则公式(10)显然成立, 从而根据公式(9)得到真实控制动作的表达式:

$$\begin{cases}t_1=12\pi_\theta^{(1)}+2\\ t_2=12\pi_\theta^{(2)}+t_1+2\\ t_3=12\pi_\theta^{(3)}+t_2+2\\ t_4=12\pi_\theta^{(4)}+t_3+2\end{cases} \tag{11}$$

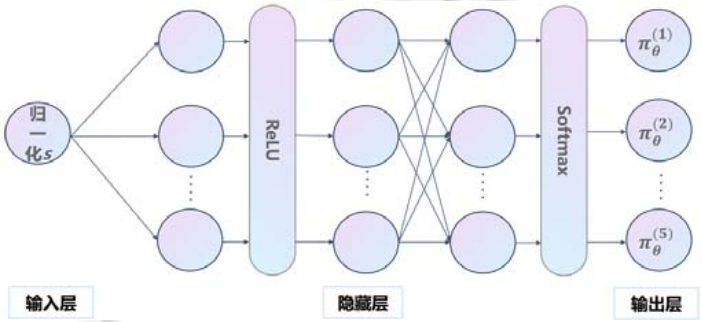


图 8 油泵的神经网络控制器

不难验证, 公式(11)定义的控制时间满足公式(8). 这样, 通过 Softmax 激活函数和一些简单变换, 巧妙地解决了神经网络输出的线性不等式约束关系. 这一重要技巧和思想是本文的创新点之一.

4 增广拉格朗日安全强化学习算法

现有 CMDP 求解的主流方法之一是拉格朗日乘子法, 在上文第 2.1 节中讨论了该方法存在的缺陷以及增广拉格朗日优化方法的对比优势. 本节将增广拉格朗日乘子法和强化学习算法(主要是 DDPG 算法)相结合, 设计基于增广拉格朗日方法的新型安全强化学习方法 ALM-DDPG.

4.1 算法设计思路

在上节所定义的工业油泵 CMDP 基础上, 学习安全最优控制器等价于求解问题(1), 即

$$\max_{\pi} J_r(\pi),$$
$$\text{s.t. } J_c(\pi) \leqslant \alpha.$$

进一步地, 根据公式(7)可知, 油泵 CMDP 中的损耗函数 C 为非负函数, 且根据安全性 R_s 的要求, 此处损耗上界 α=0. 因此, 在油泵案例中问题(1)等价于:

$$\left.\begin{aligned}&\max_{\pi} J_r(\pi),\\ &\text{s.t. } J_c(\pi)=0\end{aligned}\right\} \tag{12}$$

为了应用增广拉格朗日方法求解公式(12), 根据公式(3)可得公式(12)对应的增广拉格朗日函数:

$$L(\pi,\lambda,\rho)=J_r(\pi)-\lambda J_c(\pi)-\frac{\rho}{2}J_c(\pi)^2 \tag{13}$$

其中, ρ>0 为二次惩罚因子. 根据文献[31], 求解公式(12)的增广拉格朗日方法的基本步骤如下.

步骤 1: 选定初始策略 π, 初始拉格朗日乘子 λ=0, 初始惩罚因子 ρ>0; 选定惩罚因子扩张系数 κ>1.

步骤 2: 关于 π 最大化公式(13), 得 π'=argmax$_{\pi}L(\pi,\lambda,\rho)$.

步骤 3: 若 $|J_c(\pi')|$ 足够小, 则算法结束; 否则, 按如下方式更新 λ, ρ.

a) $\lambda=\lambda+\rho*J_c(\pi')$;

b) $\rho=\kappa*\rho$.

然后转步骤 2.

为了在强化学习算法中应用上述步骤,有 3 个难题需要解决, 分别是: (1) 如何计算期望收益 $J_r(\pi)$; (2) 如何计算期望损耗 $J_c(\pi)$; (3) 如何据公式(13)优化更新 π, 即对固定的 λ, ρ 求解 argmax$_{\pi}L(\pi,\lambda,\rho)$. 为解决这 3 个问题, 我们借鉴经典强化学习算法 DDPG (deep determinstic policy gradient)[32]的思想. DDPG 是应用于连续动作空间的确定性策略强化学习算法, 其实现基于 Actor-Critic 框架. 在 DDPG 算法的实现中, 有两个主要的神经网络, 分别是 Actor 网络即控制策略网络以及 Critic 网络即动作-收益网络. Critic 网络的目的是近似 MDP 的动作-收益函数, 从而能够起到对策略网络进行评判和改进的作用. 对给定的 MDP 和策略 π, 动作-收益函数的定义为

$$Q^{\pi}(s,a)=E_{\tau\sim\pi}[R(\tau)|s_0=s,a_0=a] \tag{14}$$

其含义是, 在状态 s 下, 采取动作 a 后再执行策略 π 所能获得的长期期望收益值; 其作用是, 对策略 π 下,在状态 s 执行特定动作 a 所产生的控制效果给出定量衡量. 如果对每一个状态-动作二元组(s,a)都能计算一个较为准确的 $Q^{\pi}(s,a)$值, 则可以评判对于策略 π 来讲, 在状态 s 下, 何种动作 a 能够带来最大的长期收益, 从而可以指导对当前策略 π 的更新. 不难看出, 对确定性策略 π 有:

$$J_r(\pi)=E_{s_0\sim\mu}[Q^{\pi}(s,\pi(s))] \tag{15}$$

因此, 可以借助动作收益函数计算期望收益 $J_r(\pi)$. 精确的 Q^{π} 函数难以获取, 然而其可以通过 Bellman 方程进行迭代逼近:

$$Q^{\pi}(s,a)=E_{s'\sim P}[R(s,a,s')+\gamma E_{a'\sim\pi}[Q^{\pi}(s',a')]] \tag{16}$$

如果将公式(14)–公式(16)中的收益函数替换为损耗函数, 则可得到动作-损耗函数和相应的 Bellman 方程.

在以上分析的基础上, 我们提出利用增广拉格朗日方法扩展 DDPG 算法构建 ALM-DDPG 算法的基本思路: 搭建 3 个神经网络, 即策略网络 π_{θ_a}、动作-收益网络 $Q_{\theta_r}(s,a)$ 和动作-损耗网络 $Q_{\theta_c}(s,a)$, 其中, θ_a, θ_r, θ_c 分别代表 3 个网络的参数; 通过 Bellman 方程(16)对 θ_r 迭代更新, 可不断逼近真实的动作-收益函数, 对 θ_c 的更新和动作-损耗函数的逼近可类似进行; 根据公式(15), 可通过 $Q_{\theta_r}(s,a)$ 和 π_{θ_a} 实现对 $J_r(\pi)$的估算, 对 $J_c(\pi)$的估算可类似进行; 在此基础上, 可根据公式(13)对 θ_a 进行梯度上升, 不断优化 π_{θ_a}. 从而, 增广拉格朗日方法的步骤 2、步骤 3 得以顺利实施.

4.2 算法实现

(1) 主体算法

本文基于增广拉格朗日乘子法实现的安全强化学习算法 ALM-DDPG 伪代码如下.

算法. ALM-DDPG.

输入: 初始化策略网络的参数 θ_a, 动作-收益网络的参数 θ_r, 动作-损耗网络的参数 θ_c, 经验池 D, 设置目标网络的参数为 $\theta'_a,\theta'_r,\theta'_c \leftarrow \theta_a,\theta_r,\theta_c$, 拉格朗日乘子 λ=0, 惩罚因子 ρ>0, 惩罚因子扩张系数 κ>1, 最大迭代回合数 *maxEpoch*, 平均损耗阈值 η.

输出: 策略网络 π_{θ_a}.

1. *epoch*←0, *maxEpochSteps*←1;
2. **while** *epoch*<*maxEpoch* **do**
3. *step*←0
4. **while** *step*<*maxEpochSteps* **do**
5. 探索, 获取一步数据(s,a,s',r,c)存入 D;
6. 随机从 D 选取批量五元组 B={(s,a,s',r,c)};
7. *UpdateQ*($B,\theta_r,\theta'_r,\theta'_a$);
8. *UpdateQ*($B,\theta_c,\theta'_c,\theta'_a$);
9. *UpdatePolicy*($B,\theta_r,\theta_c,\theta_a,\theta'_a$);
10. **if** 策略网络更新达到一定的次数 **then**
11. 计算平均动作-损耗: $Q_c=\sum_{s\in B}Q_{\theta_c}(s,\pi_{\theta_a}(s))\big/|B|$;
12. 更新拉格朗日乘子: $\lambda=\lambda+\rho*Q_c$;
13. **if** $Q_c>\eta$ **then**
14. 更新惩罚因子: $\rho\leftarrow\kappa*\rho$;
15. **else**
16. ρ←0, *maxEpochSteps*←10;
17. **end if**
18. **end if**
19. **end while**
20. **end while**
21. **return** π_{θ_a}

算法说明:

- 上述算法第 3 行–第 5 行表示探索过程, 即第 3 节提到的工业油泵中的智能体与环境的交互过程,获取一步的探索经验存入经验池 D.
- 第 7 行–第 9 行分别表示策略网络 π_{θ_a}、动作-收益网络 $Q_{\theta_r}(s,a)$ 和动作-损耗网络 $Q_{\theta_c}(s,a)$ 及其目标网络的更新过程, 具体见后文子函数 UpdateQ 和 UpdatePolicy 介绍.

- 第 10 行表示在更新若干步策略网络将其优化到一定程度之后, 进行一次增广拉格朗日乘子和惩罚因子的更新, 对应增广拉格朗日优化方法 3 个步骤中的步骤 2.
- 第 11 行、第 12 行表示按增广拉格朗日乘子法的步骤 3 更新拉格朗日乘子λ.
- 第 13 行、第 14 行表示按增广拉格朗日乘子法的步骤 3 更新惩罚因子ρ: 如果此时的损耗函数值仍比较大, 超过输入中设定的平均损耗阈值η, 则继续增大惩罚项.
- 第 15 行、第 16 行表示损耗函数降至一定程度即小于输入中设定的平均损耗阈值η时, 将惩罚因子置 0, 算法退化为一般拉格朗日乘子法; 同时, 将每回合的探索步数 *maxEpochSteps* 由 1 改为 10, 其原因随后介绍.

概况来讲, ALM-DDPG 算法在 DDPG 算法的基础上引入了如下技巧.

1) DDPG 算法只存在 1 个价值网络(和其目标网络), ALM-DDPG 算法同时训练 2 个价值网络, 分别是动作-收益网络 $Q_{\theta_r}(s,a)$ 和动作-损耗网络 $Q_{\theta_c}(s,a)$, 用来近似 CMDP 的动作-收益函数和动作损耗-函数.
2) 在 DDPG 中, 训练的目标是通过优化策略网络最大化价值网络, ALM-DDPG 的优化过程围绕增广拉格朗日函数(13)开展.
3) ALM-DDPG 通过在拉格朗日函数中引入二次惩罚项, 使得训练一开始就较倾向于降低损耗, 而传统拉格朗日乘子法的乘子λ初值为 0, 使得在训练初期对损耗的惩罚不够, 易陷入不安全的局部最优; 且在增广拉格朗日函数中, 乘子项和二次惩罚项相互协同, 实现乘子学习率的动态调节, 算法表现出更好的收敛性、稳定性.
4) ALM-DDPG 设计了两阶段探索和训练: 在训练初期, 每个回合只探索 1 个动作(*maxEpochSteps*=1), 同时, 基于增广拉格朗日函数(13)进行优化; 在训练后期, 每个回合探索 10 个动作, 同时将增广拉格朗日函数中的二次惩罚因子置为 0, 即改为采用普通拉格朗日乘子法进行训练. 这样做的目的是在训练初期增加对系统状态采样的随机性和探索性, 以更大的概率采集到不安全状态, 同时借助二次惩罚项尽快降低不安全状态的损耗值; 在训练后期, 学习到的策略已具有较高的安全性, 惩罚函数的值降到一个比较低的水平(低于阈值η), 此时不再需要借助二次惩罚项增加惩罚力度, 同时改为 10 步探索是为加大对收敛状态处的动作探索, 尽快探寻到最优动作以最大化收益函数. 简言之, 两阶段训练的第 1 阶段训练注重优化安全目标, 第 2 阶段训练注重优化最优目标.

(2) 重要子函数介绍

ALM-DDPG 算法中调用的两个子函数 UpdateQ 和 UpdatePolicy 详情如下.

子函数. UpdateQ.

输入: 批量五元组 $B=\{(s,a,s',r,c)\}$, 动作-收益网络或者动作-损耗网络参数θ及其目标网络参数θ', 策略网络的目标网络参数 θ'_a.

输出: 无.

1. 计算动作-收益函数或动作-损耗函数的目标函数:
2. $y(s')=r+\gamma Q_{\theta'}(s',\pi_{\theta'_a}(s'))$ 或者 $y(s')=c+\gamma Q_{\theta'}(s',\pi_{\theta'_a}(s'))$
3. 梯度下降法更新动作-收益网络或动作-损耗网络:
4. $\nabla_\theta \frac{1}{|B|}\sum_{(s,a,s',r)\in B}(Q_\theta(s,a)-y(s'))^2$或$\nabla_\theta \frac{1}{|B|}\sum_{(s,a,s',c)\in B}(Q_\theta(s,a)-y(s'))^2$
5. 软更新目标网络:
6. $\theta'=\tau\theta'+(1-\tau)\theta$

算法说明:

- 第 2 行表示根据 Bellman 方程(16)右端项计算动作-收益函数或动作-损耗函数的目标值.
- 第 4 行表示根据 Bellman 方程(16)构造均方误差(mean-squared Bellman error, MSBE), 用来衡量动作-

收益网络 $Q_{\theta_r}(s,a)$ 或动作-损耗网络 $Q_{\theta_c}(s,a)$ 逼近动作-收益函数和动作-损耗函数的程度, 目标是将均方误差降至 0, 所以使用梯度下降法.

- 第 6 行表示 DDPG 中的软更新方式, τ为介于 0 和 1 之间的常数, 一般选择接近 1 的常数, 如 0.995.

子函数. UpdatePolicy.

输入: 批量五元组 $B=\{(s,a,s',r,c)\}$, 动作-收益网络的参数θ_r, 动作-损耗网络的参数θ_c, 策略网络的参数θ_a及其目标网络的参数 θ'_a.

输出: 无.

1. 梯度上升法更新策略网络:
2. $\nabla_{\theta_a}\frac{1}{|B|}\sum_{s\in B}\left(Q_{\theta_r}(s,\pi_{\theta_a}(s))-\lambda Q_{\theta_c}(s,\pi_{\theta_a}(s))-\frac{\rho}{2}Q_{\theta_c}(s,\pi_{\theta_a}(s))^2\right)$
3. 软更新目标网络:
4. $\theta'_a=\tau\theta'_a+(1-\tau)\theta_a$

算法说明:

上述子算法第 2 行对策略网络的更新依据为: 策略网络 π_{θ_a} 的目标是产生 4 个时间节点 a, 使得收益尽可能大损耗尽可能小, 利用增广拉格朗日乘子法将此目标转化为公式(13), 则策略网络 π_{θ_a} 的优化目标为

$$\max_{\theta_a}\left(Q_{\theta_r}(s,\pi_{\theta_a}(s))-\lambda Q_{\theta_c}(s,\pi_{\theta_a}(s))-\frac{\rho}{2}Q_{\theta_c}(s,\pi_{\theta_a}(s))^2\right).$$

故使用梯度上升法更新θ_a.

(3) 算法重要参数设定

本文在具体实验中, 关于 ALM-DDPG 算法的一些重要参数的设定和相关依据如下.

- 最大迭代次数 *maxEpoch*=8000, 即算法迭代 8 000 个回合, 与环境交互约 80 000 次.
- 收益或损耗的折扣系数γ=0.9; 目标网络软更新系数τ=0.995.
- 网络的参数: 策略网络设置为 1×8×5 的网络, 其中, 隐藏层使用 ReLU 激活函数, 输出层使用 Softmax 激活函数; 动作-收益网络设置为 5×128×1, 激活函数都使用 ReLU 函数; 动作-损耗网络设置为 5×64×1, 激活函数都使用 ReLU 函数.

以上多为DDPG算法常见参数设置, 无须进行过多调节; 其中, 策略网络规模设定较小, 因其输入规模很小(1 维); 如下参数设定为 ALM-DDPG 算法所独有.

- 初始惩罚因子ρ=1. 理论上, ρ可取任何正数, 常见的取值小于 1, 如 0.1 等[33]; 在本文案例中, 考虑到在训练初期尽可能平衡对收益和损耗的优化, 因此保持公式(13)中 $J_r(\pi)$和 $J_c(\pi)^2$ 两项的系数在同一数量级, 故ρ取为 1.
- 惩罚因子扩张系数κ=1.05. 理论上, κ可取任何大于 1 的正数, 常见如 10 等[33]; 在本案例中, 增广拉格朗日乘子法的步骤 2 通过随机梯度下降实现, 最小化策略π的过程以梯度下降 50 次为终止标准, 这导致对策略π每更新 50 次即更新一次λ和ρ. 为防止λ和ρ增长过快, 根据初步实验分析调低了扩张系数至 1.05.
- 平均损耗阈值η=0.05. η用于判断损耗 $J_c(\pi)$是否已小至可以停止惩罚因子扩张的程度, 可通过初步训练预判损耗值最低降至何种水平来决定.

值得说明的是, 目前, 关于增广拉格朗日乘子法最重要的 3 个参数ρ, κ, η的设定已经有较为成熟的理论和实验总结, 见文献[33], 在本文实验中并未使用这些复杂的技巧. 此外, 按照文献[33]的论述, 初始惩罚因子ρ的取值要求相当松弛. 本文的实验经验也表明, 3 个参数ρ, κ, η中只有κ需要根据惩罚因子的扩张频率做适当的调整.

5 ALM-DDPG 算法在油泵控制系统上的实验

本节将 ALM-DDPG 算法应用到油泵控制案例中评估算法的性能，作为对比实现了另外两种安全强化学习算法 Lagrange-DDPG 和λ-DDPG 算法，其中，

1) Lagrange-DDPG 算法是在 DDPG 算法基础上实现的一般拉格朗日乘子方法，乘子λ会从 0 开始按照固定的学习率线性增长. 一般来讲，实现一个强化学习算法的拉格朗日版本以解决安全强化学习问题并不复杂，只需在传统强化学习的框架之外附加对拉格朗日乘子λ的更新步，实现控制策略和乘子λ的交叉迭代即可；乘子λ的更新步长固定，更新梯度可以用训练轨迹的平均损耗值估计，也可以通过额外训练一个损耗神经网络估计. 在本文的实现中, Lagrange-DDPG 自然地继承了 ALM-DDPG 算法，只是去除了拉格朗日函数中的二次项，算法同时训练收益网络和损耗网络，以损耗网络的输出作为乘子λ更新的梯度.
2) λ-DDPG 算法从实质上讲就是在一般 DDPG 算法中进行了收益函数重新设计(reward shaping)，即采用固定惩罚系数λ对损耗进行惩罚，从而修正收益函数. 这种将对约束的违背以固定的权重编码到优化目标中的思想是求解约束优化最简单直接的技巧，其关键在于选取合适的惩罚权重. Lagrange-DDPG 算法可以看作是自动学习最优惩罚系数的λ-DDPG 算法；反之, λ-DDPG 算法可以看作固定拉格朗日乘子的 Lagrange-DDPG 算法.

评价指标. 在油泵控制案例中，3 种算法的目标都是最小化长期累积油量，所以后期油量变化都趋于安全下界$V'_{\min}$附近. 为评估各种算法的训练效果，考虑初始油量为$V'_{\min}=5.1$的情况，使用训练结束后得到的神经网络控制器π_{θ_a}与油泵仿真环境交互仿真 10 个周期，得到轨迹上的总收益和总损耗作为衡量 3 种算法的指标. 另外，神经网络控制器π_{θ_a}的输出层使用了 Softmax 激活函数，导致每个输出只能理论上趋向于 0 但无法等于 0，即$\pi_{\theta_a}^{(i)}(s)>0$，$i$=1,2,…,4，所以根据公式(11)有，每周期第 1 次打开油泵的时间点$t_1=12\pi_{\theta_a}^{(1)}(s)+2>2$. 结合图 3 耗油率曲线可知，对于初始油量$V'_{\min}=5.1$，损耗必定大于 0，但可以无限趋向于 0. 换言之，对于初始油量 5.1，神经网络控制器无法保证系统对于安全边界$V'_{\min}$的绝对安全，但可以任意逼近该安全边界. 为此，引入微小的安全松弛度β，定义相对安全要求为：神经网络控制器π_{θ_a}与初始油量为 5.1 的油泵仿真环境交互 10 个周期的总损耗要小于β，称满足这个要求的控制器为相对阈值β安全的控制器.

实验环境. 本文工作基于 PyTorch 框架(https://pytorch.org/)实现，所有代码包括后文将要介绍的形式化验证相关内容均在 https://gitee.com/zhaohj2016/oil-pump-control-jos 开源. 所有实验在常规个人台式机和笔记本电脑上开展，未使用 GPU 加速. 需要指出的是，训练耗时与第 3.2 节所述油泵仿真环境的仿真步长密切相关，后文结果均为采用步长 0.01 仿真所取得；针对油泵案例，可设计基于公式推导而非固定步长离散采样的仿真方法，从而大大提高训练效率，本文对此不再详述.

5.1 算法整体对比情况

首先介绍一下对 3 种算法所采用的重要参数设置.

1) 对 ALM-DDPG 算法只采用了一组参数设置，见上文第 4.2 节的算法介绍.
2) 对 Lagrange-DDPG 算法，拉格朗日乘子λ的学习率会影响λ的增长，从而影响智能体在收益和损耗之间的权衡，对比实验为 Lagrange-DDPG 设置了 4 组不同的学习率: 10^{-4}, 2×10^{-4}, 5×10^{-4}, 10^{-3}.
3) 对λ-DDPG 算法，固定拉格朗日乘子λ的大小会影响智能体在收益和损耗中的权衡：λ很小，会导致拉格朗日函数中的损失项很小，智能体更加注重收益的优化，忽略损耗的影响；λ很大，会导致损耗项很大，智能体更加注重损耗的优化. 鉴于此，对比实验为λ-DDPG 算法设置了 4 组不同的常数: λ=1, 10, 20, 40.

对 3 种算法 9 组参数各进行了 20 次训练，每次训练 8 000 个回合，交互 80 000 步，通过最终所得神经网络控制器π_{θ_a}与初始油量为 5.1 的油泵仿真环境交互 10 个周期，得到轨迹的总收益和总损耗. 对给定的相对安全阈值β=0.1 和β=0.02，分别统计了每组参数总损耗低于β的次数、总损耗低于β且总收益高于 68 的次数，见

表 1 和表 2.

表 1 3 种算法 10 周期(初值 5.1)总损耗和总收益对比(β=0.1)

算法名称	总损耗小于 0.1 的次数	总损耗小于 0.1 且总收益大于 68 的次数
ALM-DDPG	**20**	**16**
Lagrange-DDPG (10^{-4})	6	6
Lagrange-DDPG (2×10^{-4})	9	8
Lagrange-DDPG (5×10^{-4})	15	6
Lagrange-DDPG (10^{-3})	12	4
λ-DDPG(λ=1)	1	1
λ-DDPG(λ=10)	15	8
λ-DDPG(λ=20)	14	11
λ-DDPG(λ=40)	13	6

表 2 3 种算法 10 周期(初值 5.1)总损耗和总收益对比(β=0.02)

算法名称	总损耗小于 0.02 的次数	总损耗小于 0.02 且总收益大于 68 的次数
ALM-DDPG	**17**	**15**
Lagrange-DDPG (10^{-4})	1	1
Lagrange-DDPG (2×10^{-4})	7	4
Lagrange-DDPG (5×10^{-4})	5	4
Lagrange-DDPG (10^{-3})	7	2
λ-DDPG(λ=1)	1	1
λ-DDPG(λ=10)	4	4
λ-DDPG(λ=20)	11	9
λ-DDPG(λ=40)	8	5

可以看到, ALM-DDPG 算法在 20 次实验中, 训练所得神经网络控制器 π_{θ_a} 具有如下特点.

1) 总损耗均小于 0.1, 全部达到β=0.1 的相对安全性要求; 总损耗有 17 次小于 0.02, 基本都达到更加严苛β=0.02 的安全性要求.
2) 在保证安全性要求的前提下, 基本都能够获得大于 68 的较高总收益: β=0.1 时, 16 次达到较高的总收益; β=0.02 时, 15 次达到较高的总收益.

对比 Lagrange-DDPG 算法和λ-DDPG 算法, 在总损耗方面, 二者均最多 15 次达到β=0.1 的安全性要求, λ-DDPG 算法最多 11 次达到β=0.02 的安全性要求, 明显差于 ALM-DDPG 算法; 在总收益方面, 无论是以β=0.1 还是以β=0.02 的安全性要求为前提, ALM-DDPG 算法均明显好于 Lagrange-DDPG 和λ-DDPG 这两种算法.

综上, ALM-DDPG 算法在满足安全性要求和实现最优化目标方面明显优于其他两种算法, 说明 ALM-DDPG 算法在平衡损耗和收益上取得了很好的效果.

5.2 ALM-DDPG算法与Lagrange-DDPG算法对比

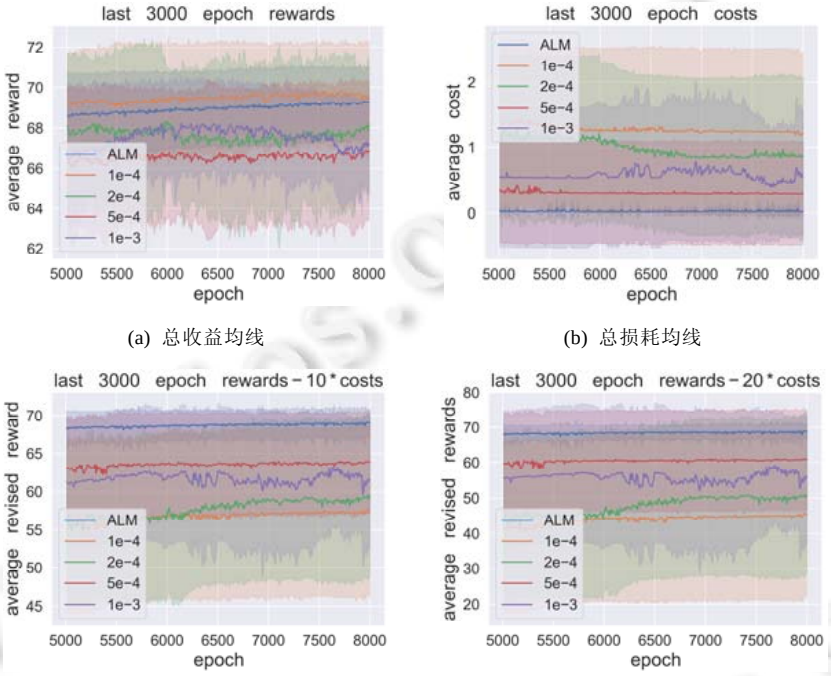
实验对比了 ALM-DDPG 算法和前面提到的 4 组不同λ更新率的 Lagrange-DDPG 算法在最后 3 000 个 epoch 的总收益和总损耗变化情况, 其均方图如图 9(a)和图 9(b)所示, 其中, 图 9(a)为总收益均方图, 图 9(b)为总损耗均方图.

(1) Lagrange-DDPG 算法之间的对比

从图 9(b)可以发现, 4 组参数的 Lagrange-DDPG 算法平均损耗都没有满足β=0.1 的相对安全性要求, 其中, 损耗最小的 5×10^{-4} 的平均总损耗也在 0.3 左右. 在图 9(a)中, 总收益为 10^{-4} 最大, 2×10^{-4} 次之, 5×10^{-4} 最小; 图 9(b)中, 总损耗为 10^{-4} 最大, 2×10^{-4} 次之, 5×10^{-4} 最小. 这印证了λ的学习率越小, λ就增长得越缓慢, 算法也就越注重优化收益项; λ的学习率越大, λ就增长得越快, λ算法也就越注重优化损耗项. 当继续增大λ的学习率至 10^{-3} 时, 其总损耗却大于 5×10^{-4} 的学习率, 而且稳定性也变差. 这意味着继续增大λ的学习率并不会带来更小的总损耗. 其原因主要是学习率较大, λ变化较快, 从而导致算法不稳定. 此外, 除了前文提到的 4 组参数以外, 还实验了λ更新率为 5×10^{-5} 总共 20 次, 发现虽均获得较大的总收益, 但总损耗也特别大, 说明总收益是以忽略安全性为代价获得的.

(2) ALM-DDPG 算法与 Lagrange-DDPG 算法的对比

观察图 9(a)可发现，ALM-DDPG 算法的总收益仅差于λ学习率为 10^{-4} 的 Lagrange-DDPG 算法，比其他学习率的 Lagrange-DDPG 算法都要好；但学习率为 10^{-4} 的总损耗明显大于 ALM-DDPG 算法的总损耗，不能保证油泵控制案例的安全要求. 从图 9(b)可发现，在总损耗方面，ALM-DDPG 算法远小于 Lagrange-DDPG 算法中的任一种，达到了油泵控制案例β=0.1(甚至更低)的相对安全性要求，且十分的稳定(方差小). 为了更明显地体现 ALM-DDPG 算法的优势，定义总修正收益=总收益−λ×总损耗. 图 9(c)和图 9(d)分别反映了λ=10 及λ=20 情况下，最后 3 000 个回合的总修正收益的变化情况. 可以看出，ALM-DDPG 算法的总修正收益最大最稳定，明显优于 Lagrange-DDPG 算法的任何一种.



(a) 总收益均线　　(b) 总损耗均线

(c) 总修正收益(λ=10)　　(d) 总修正收益(λ=20)

图 9　总收益和总损耗以及总修正收益均方图(ALM-DDPG & Lagrange-DDPG)

通过以上分析可以总结出, ALM-DDPG 算法克服了 Lagrange-DDPG 算法继续增大学习率也不能保证安全性要求的缺点，实现了油泵控制案例的安全性要求；同时保证了很好的总收益，在综合表现(总修正收益)上优于 Lagrange-DDPG 算法中的任一种. 这说明 ALM-DDPG 算法在保证安全性要求的同时，兼顾了最优总收益的目标.

5.3　ALM-DDPG算法与λ-DDPG算法对比

实验对比了 ALM-DDPG 算法和 4 组不同λ大小的λ-DDPG 算法在最后 3 000 个 epoch 总收益和总损耗的变化情况，如图 10(a)和图 10(b)所示，其中，图 10(a)为总收益均方图，图 10(b)为总损耗均方图.

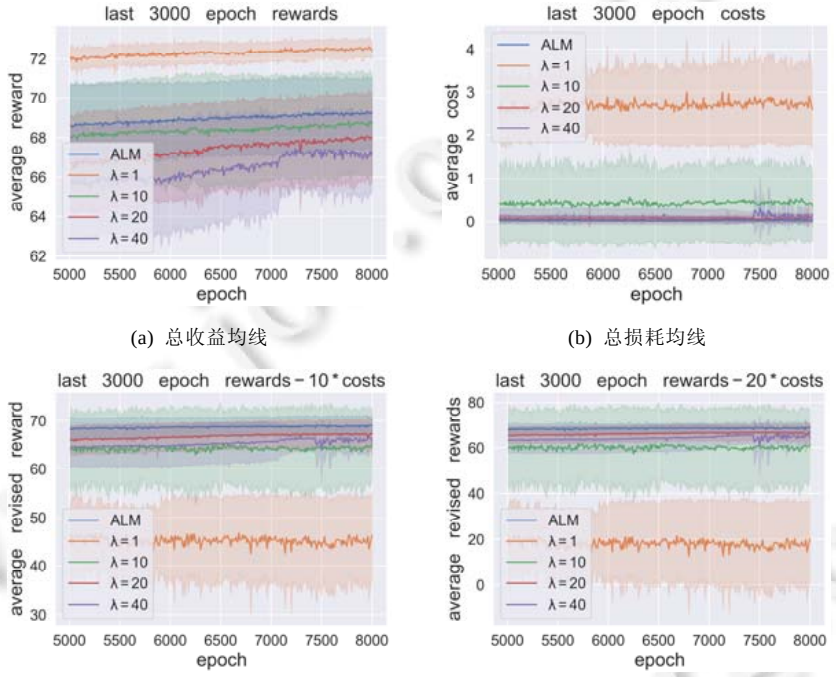
(1) 不同λ常量的λ-DDPG 算法对比

在图 10(a)中, λ越大，总收益越小；在图 10(b)中, λ越大，总损耗越小; λ=1 时，总收益最大，但总损耗也最大，最不安全. 这印证了λ越大, λ-DDPG 算法越注重优化总损耗项，越不注重优化总收益项. 观察图 10(a)、图 10(b)两图还可以发现, λ=20 和λ=40 的总损耗基本相同，但λ=40 的总收益低于λ=20，说明继续增大λ反而会降低总收益. 当λ=40 或者λ=20，总损耗接近于 0，克服了 Lagrange-DDPG 算法不能保证安全性要求的缺点，但

此时的总收益相对偏小, 说明λ=40 或者λ=20 是以减少总收益为代价而获得低的总损耗. 总之, 对于λ-DDPG 算法, λ太小, 导致安全性无法满足; λ太大, 会导致总收益偏小. 而搜寻一个合适的λ, 使得既能够满足安全性又能够最大化总收益是困难的.

(2) ALM-DDPG 算法与λ-DDPG 算法的对比

从图 10(b)可以发现, ALM-DDPG 的总损耗比λ-DDPG 更小更稳定; 从图 10(a)可以发现, 在总收益方面, ALM-DDPG 差于λ=1, 优于其他. 但上文指出, λ=1 的总损耗明显超出油泵控制案例的安全性要求, 保证不了安全性. 图 10(a)中与 ALM-DDPG 总收益最接近的是λ=10, 但据图 10(b)知, λ=10 的安全性较差; 类似地, λ=40 和λ=20 能够保证安全性要求, 但与 ALM-DDPG 算法的总收益相差甚远. 图 10(c)和图 10(d)展示了 ALM-DDPG 和λ-DDPG 的总修正收益变化情况, 其中, 图 10(c)的λ设置为 10, 图 10(d)的λ设置为 20. 易见, ALM-DDPG 算法的总修正收益最大最稳定, 好于λ-DDPG 算法的任何一种.



(a) 总收益均线 (b) 总损耗均线

(c) 总修正收益(λ=10) (d) 总修正收益(λ=20)

图 10 总收益和总损耗以及总修正收益均方图(ALM-DDPG & λ-DDPG)

通过以上分析可知, ALM-DDPG 算法克服了λ-DDPG 算法的缺点, 能够既满足安全性又尽量最大化总收益; 避免了盲目搜寻最优λ的过程, 实现了λ自动学习; 而且在总损耗方面明显小于λ-DDPG 算法, 也就是更加满足油泵控制案例的安全性要求.

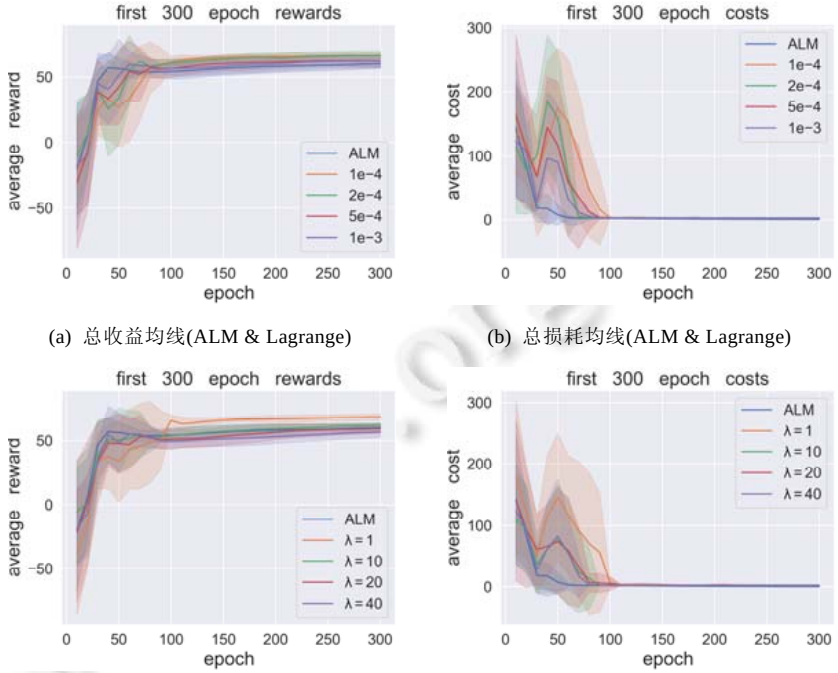
5.4 3种算法收敛情况的对比

本文第 2.1 节指出, 基于增广拉格朗日乘子法实现安全强化学习算法有望取得如下两种优势.

1) 通过惩罚因子ρ的增长对λ的学习率进行动态调节, 乘子项和二次惩罚项协同作用使算法表现出更好的收敛性和稳定性.
2) 惩罚因子ρ的初值大于 0, 借助二次惩罚项, 在训练初期以较大倾向优化安全性, 避免陷入不安全的局部最优解.

第 5.1 节–第 5.3 节的实验结果已充分说明了 ALM-DDPG 算法在平衡安全性和最优性训练目标上的优势,

本节进一步说明 ALM-DDPG 在提升算法收敛性和稳定性方面的效果. 图 11 展示了 3 种算法在训练初期 300 个回合收益和损耗均线图(各 20 次训练)的对比情况: 图 11(a)、图 11(b)分别对比了 ALM-DDPG 和 Lagrange-DDPG 这两种算法的收益均线和损耗均线, 图 11(c)、图 11(d)分别对比了 ALM-DDPG 和λ-DDPG 这两种算法的收益均线和损耗均线. 明显地, ALM-DDPG 对应曲线在 4 幅图中均以更快的速度、更小的波动收敛, 验证了本文最初的设想.



(a) 总收益均线(ALM & Lagrange)　(b) 总损耗均线(ALM & Lagrange)

(c) 总收益均线(ALM & λ-DDPG)　(d) 总损耗均线(ALM & λ-DDPG)

图 11　3 种算法训练初期收益和损耗均线对比图(ALM-DDPG、Lagrange-DDPG、λ-DDPG)

6　神经网络控制器性能评估

本节对 ALM-DDPG 所生成的油泵神经网络控制器进行性能评估, 包括安全性和最优性两个指标, 其中, 安全性评估主要借助了形式化验证方法, 最优性评估主要通过与文献[9]中的理论最优控制器进行对比.

6.1　安全性评估

(1) 基于形式化验证的安全评估总体思路

本节希望借助形式化方法证明 ALM-DDPG 生成的神经网络控制器能够严格保证系统的油量满足安全性约束, 即任意时刻的油量位于安全区间$[V'_{\min},V'_{\max}]$. 为此, 参考文献[9], 将一个周期内油量随时间的变化规律$v(v_0,a,t)$编码为一阶逻辑公式, 其中, $v_0\in[V'_{\min},V'_{\max}]$为周期初始油量, $a=\pi(v_0)$为由神经网络控制器π产生的单周期控制动作, $t\in[0,20]$为时间; 然后, 利用 SMT 约束求解器证明:

$$\forall v_0\in[V'_{\min},V'_{\max}].\ \forall t\in[0,20].\ v(v_0,a,t)\in[V'_{\min},V'_{\max}] \tag{17}$$

由于所搭建的神经网络控制器输出层采用了 Softmax 激活函数, 所以 $v(v_0,a,t)$的编码涉及到指数函数 exp,故本文采用了可求解非线性约束的 dReal 求解器[34]验证公式(17). 由于 dReal 求解器是基于区间计算的, 在安全边界处, 即 $v_0=V'_{\min}$ 或 $v_0=V'_{\max}$ 时, 由于区间计算误差公式(17)是无法严格证明的, 故在实际验证时, 将初始油量 v_0 限定在区间$[V'_{\min}+\varepsilon,V'_{\max}-\varepsilon]$, 其中, $\varepsilon=\frac{1}{200}(V'_{\max}-V'_{\min})$, 即安全区间长度的 1/200. 虽然对公式(17) 的

验证在一个很大的初值范围上开展, 但根据第 4.2 节的算法参数设置, 策略网络具有相对简单的结构, 即 1×8×5 的网络, 隐层和输出层分别采用 ReLU 激活函数和 Softmax 激活函数, 因而整个编码出的约束在 dReal 可以处理的能力范围内. 在实验中, 每一个验证任务均可在数分钟之内得到结果.

(2) 基于动作-损耗网络的控制器微调

为了提高公式(17)验证的成功率, 依据 ALM-DDPG 所返回的动作-损耗网络 $Q_{\theta_c}(s,a)$ 对策略网络 π_{θ_a} 进行了安全性微调. 具体来说, 借助状态-损耗函数 $Q_{\theta_c}(s,\pi_{\theta_a}(s))$ 检验是否存在损耗值超过某阈值的状态点, 过高的损耗值可能意味着对安全性的违背. 如图 12 所示, 初始油量 15.0 对应的状态损耗函数值出现一个尖峰, 意味着 15.0 附近的状态是潜在的不安全状态. 如果检测到这种状态, 则通过 $Q_{\theta_c}(s,\pi_{\theta_a}(s))$ 在相应的状态点上对 θ_a 进行若干次梯度下降, 从而提高策略 π_{θ_a} 在对应状态点的安全性. 本文设定阈值 0.02 对 ALM-DDPG 产生的控制器进行微调.

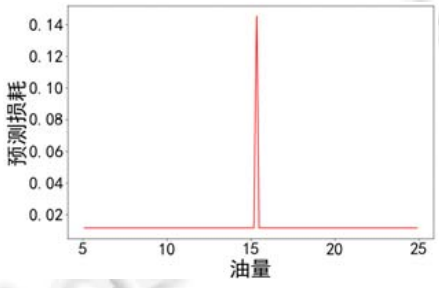


图 12 通过损耗函数识别潜在的不安全状态

(3) 微调和形式化评估结果

ALM-DDPG 实验共训练了 20 个神经网络控制器, 其中, 以 5.1 为初值连续仿真 10 周期轨迹收益值大于 70 的有 10 个, 微调和形式化验证针对这 10 个控制器进行, 详细结果见表 3. 其中, $Ctrl_i$, i=1,2,…,10 表示 ALM-DDPG 生成的 10 个控制器; *Cost*, *Reward* 行分别表示这 10 个控制器以 5.1 为初值连续仿真 10 周期的轨迹总收益和总损耗; *Reward-ft* 行表示经过微调的 10 个控制器以 5.1 为初值连续仿真 10 周期的轨迹总收益; *Verified* 和 *Verified-ft* 分别表示原始控制器和微调控制器是(✓)否(×)通过安全性形式化验证.

表 3 神经网络控制器微调和形式化验证结果

	$Ctrl_1$	$Ctrl_2$	$Ctrl_3$	$Ctrl_4$	$Ctrl_5$	$Ctrl_6$	$Ctrl_7$	$Ctrl_8$	$Ctrl_9$	$Ctrl_{10}$
Cost	0.013	0.013	0.013	0.013	0.013	0.099	0.013	0.013	0.013	0.013
Reward	70.187	70.379	70.153	70.710	70.357	70.409	70.207	70.051	70.120	70.246
Reward-ft	70.187	69.415	68.970	70.710	69.881	67.687	70.207	69.972	69.501	70.246
Verified	✓	×	×	×	×	×	✓	×	×	✓
Verified-ft	✓	✓	✓	✓	×	✓	✓	✓	✓	✓

分析表 3 的结果可以得到如下结论: 经过微调的控制器以 90%的高比例通过了形式化验证, 表示本文所提基于安全强化学习的数据驱动的控制器生成方法具有很高的安全性; 经过微调的 $Ctrl_5$ 没有通过形式化验证,通过检验 dReal 返回的反例点发现, 以 7.694 为初始油量的仿真轨迹在单周期内会产生 0.006 的损耗, 即 $Ctrl_5$ 对于 7.694 的确是不安全的, 通过设定更低的 Q_{θ_c} 阈值有望进一步提高 $Ctrl_5$ 的安全性; 对比 *Reward* 和 *Reward-ft* 发现, 安全性微调会对收益产生细微影响, 绝大多数单周期收益变化值在 0.1 以内, $Ctrl_6$ 在微调后收益受到较大影响, 因为其具有较大的损耗值, 不安全性更高, 导致调节量相对其他控制器更大.

需要强调的是, 本文的目的是探索在无模型环境下强化学习的安全性保障, 形式化验证方法在此处仅用作一种衡量强化学习所生成神经网络控制器安全性的评估手段. 本文所提方法的应用场景不受形式化方法所依赖的严格数学模型的限制.

6.2 最优性评估

为了评估 ALM-DDPG 生成控制器的最优性，本文选取表 3 中经过形式化验证的 $Ctrl_4$ 和文献[9]所提理论最优控制器 Opt 进行对比. 由于 R_o 的优化目标是长期期望收益，本文对比经过一段时间仿真之后在稳定状态下的控制器表现. 为此，从同一初始油量出发，用 $Ctrl_4$ 和 Opt 分别仿真 110 个周期，然后截取并绘制第 101–110 周期共 200 s 的油量变化，如图 13 所示，其中，蓝色实线代表 $Ctrl_4$，蓝色虚线代表 Opt. 可见，$Ctrl_4$ 和 Opt 控制器在后期都表现出稳定的周期性油量变化，且均保持在红色安全水平线 $V'_{\min}$ 之上；二者变化曲线非常吻合，在每一周期开始阶段，Opt 的曲线稍低，意味着其最优性略优于 $Ctrl_4$.

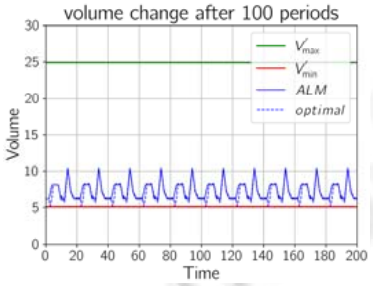


图 13 神经网络控制器和最优控制器在稳定状态下的油量曲线

为了进一步定量衡量 $Ctrl_4$ 的最优性，对其连续仿真 200 个周期计算平均周期累积油量，并结合文献[9]所报告数据给出(见表 4).

表 4 几种油泵控制器平均周期累积油量对比

	Opt	$Ctrl_4$	Uppaal-Tiga	Smart	Bang-Bang
平均周期累积油量	7.125	**7.275**	7.44	11.56	13.45

在表 4 中，Opt 和 Uppaal-Tiga 是基于形式化建模生成的控制器，其中，Opt 代表理论最优控制器，它是通过对油泵控制系统采用一阶逻辑公式建模然后进行量词消去得到的[9]，具有理论最优性和严格安全性；Uppaal-Tiga 是通过时间博弈自动机建模生成的控制器[7]，在建模过程中对时间进行了离散化，因而损失了一定的最优性，同时生成的控制器需要借助可达集计算工具 PHAVER 等进行形式化后验证; Bang-Bang 和 Smart 两个控制器来源于文献[7]的报告，由工业油泵系统所属公司根据经验设计，安全性缺乏保障且最优性较差. 可以看到，本文通过强化学习生成的神经网络控制器的最优性仅比理论最优值损失了 2%，好于其他所有控制器.

另外，本文生成的神经网络控制器相比 Opt 控制器还有一个较大的优势，即对于油量的初始值几乎没有限制，只要位于安全范围内即可，而 Opt 控制器为了保持周期性，增加了一个周期初始油量和末尾油量位于同一稳定区间的限制，从而将初值范围缩小到了[5.1,7.5]；作为对比，$Ctrl_4$ 控制器可以将 $[V'_{\min},V'_{\max}]$ 范围内的初始油量在保持安全性的前提下逐渐降至较低水平直至稳定. 图 14 对比了 $Ctrl_4$ 控制器和 Opt 控制器的初始油量-控制时间曲线图，由于存在 4 个控制时间点，所以每个控制器对应 4 条曲线，红色、绿色、黑色、蓝色线分别对应 t_1, t_2, t_3, t_4；实线和虚线分别对应 $Ctrl_4$ 控制器和 Opt 控制器. 可以看到，Opt 控制器可以应用的初值范围很小，而 $Ctrl_4$ 控制器可以适用很广的初值范围；$Ctrl_4$ 和 Opt 控制器在 t_3，t_4 这两条曲线上吻合较好，t_1，t_2 存在差异,这也解释了两者最优性的细微差距来源. 图 15 展示了 $Ctrl_4$ 从初始油量 $V'_{\max}$ 出发 10 周期的仿真效果，可以看到，$Ctrl_4$ 能够很好地将油量降低尽快趋近稳定状态，从而获取较大的长期期望收益，同时保持安全性，显示了良好的控制效果.

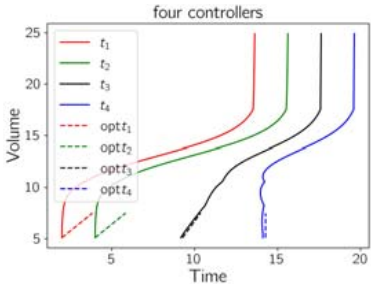


图 14 神经网络和最优控制器控制时间线

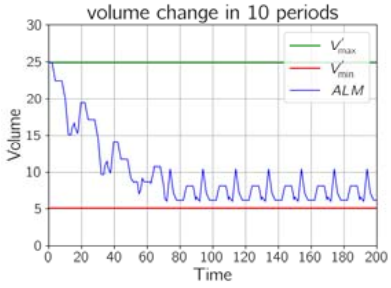


图 15 神经网络控制器 10 周期仿真(初值 24.9)

7 总 结

本文围绕工业油泵控制系统这一经典 CPS 案例, 研究了深度强化学习智能算法在安全攸关 CPS 控制器设计中的应用. 首先, 搭建了工业油泵仿真环境; 其次, 精心设计了神经网络形式的油泵控制器, 使其输出满足油泵开关时间的线性不等式约束; 最后, 为了更好地权衡安全性和最优性控制目标, 基于增广拉格朗日乘子法设计实现了新型安全强化学习算法, 改进了传统的拉格朗日安全强化学习算法. 实验部分详细评估表明, 本文所生成神经网络控制器以 90%的概率通过了严格形式化验证, 同时, 与理论最优控制器相比实现了低至 2%的最优目标值损失. 在后续工作中, 将探索本文所提算法在更广泛场景中的应用和改进, 重点将关注收益和损耗具有较强稀疏性的安全强化学习问题.

References:

[1] Kochdumper N, Gruber F, Schürmann B, Gaßmann V, Klischat M, Schürmann B, Althoff M. AROC: A toolbox for automatic arrival set optimization controller synthesis. In: Bogomolov S, Jungers R, eds. Proc. of the 24th ACM Int'l Conf. on Hybrid Systems: Computation and Control (HSCC 2021). New York: ACM, 2021. 1–6.

[2] Bai YJ, Gan T, Jiao L, Xue B, Zhan NJ. Switching controller synthesis for time-delayed hybrid systems. SCIENTIA SINICA Mathematica, 2021, 51(1): 97–114 (in Chinese with English abstract).

[3] Yang ZF, Zhang YD, Lin W, Zeng X, Tang XC, Zeng ZB, Liu ZM. An iterative scheme of safe reinforcement learning for nonlinear systems via barrier certificate generation. In: Silva A, Leino KRM, eds. Proc. of the 33rd Int'l Conf. on Computer Aided Verification (CAV 2021). Cham: Springer, 2021. 467–490.

[4] Zhao HJ, Zhan NJ, Kapur D. Synthesizing switching controllers for hybrid systems by generating invariants. In: Liu ZM, Woodcock J, Zhu HB, eds. Proc. of the Theories of Programming and Formal Methods. Berlin, Heidelberg: Springer, 2013. 354–373.

[5] Jin XY, An J, Zhan BH, Zhan NJ, Zhang MM. Inferring switched nonlinear dynamical systems. Formal Aspects of Computing, 2021, 33(3): 385–406.

[6] Bao WM, Qi ZQ, Zhang Y. Thoughts on the development of intelligent control technology. SCIENTIA SINICA Informationis, 2020, 50(8): 1267–1272 (in Chinese with English abstract).

[7] Cassez F, Jessen JJ, Larsen KG, Raskin JF, Reynier PA. Automatic synthesis of robust and optimal controllers—An industrial case study. In: Majumdar R, Tabuada P, eds. Proc. of the 12th Int'l Conf. on Hybrid Systems: Computation and Control (HSCC 2009). Berlin, Heidelberg: Springer, 2009. 90–104.

[8] Jha S, Seshia SA, Tiwari A. Synthesis of optimal switching logic for hybrid systems. In: Baruah S, Fischmeister S, eds. Proc. of the 9th ACM Int'l Conf. on Embedded Software (EMSOFT 2011). New York: ACM, 2011. 107–116.

[9] Zhao HJ, Zhan NJ, Kapur D, Larsen KG. A "hybrid" approach for synthesizing optimal controllers of hybrid systems: A case study of the oil pump industrial example. In: Giannakopoulou D, Méry D, eds. Proc. of the 18th Int'l Symp. on Formal Methods. Berlin, Heidelberg: Springer, 2012. 471–485.

[10] Bacci G, Bouyer P, Fahrenberg U, Larsen KG, Markey N, Reynier PA. Optimal and robust controller synthesis using energy timed automata with uncertainty. In: Havelund K, Peleska J, Roscoe B, Vink E, eds. Proc. of the 22nd Int'l Symp. on Formal Methods. Cham: Springer, 2018. 203−221.

[11] Bacci G, Bouyer P, Fahrenberg U, Larsen KG, Markey N, Reynier PA. Optimal and robust controller synthesis using energy timed automata with uncertainty. Formal Aspects of Computing, 2021, 33(1): 3−25.

[12] Liu YS, Halev A, Liu X. Policy learning with constraints in model-free reinforcement learning: A survey. In: Zhou ZH, ed. Proc. of the 30th Int'l Joint Conf. on Artificial Intelligence (IJCAI 2021). 2021. 4508−4515.

[13] Achiam J, Held D, Tamar A, Abbeel P. Constrained policy optimization. In: Precup D, Teh YW, eds. Proc. of the 34th Int'l Conf. on Machine Learning (ICML 2017). 2017. 22−31.

[14] Tessler C, Mankowitz DJ, Mannor S. Reward constrained policy optimization. In: Proc. of the 7th Int'l Conf. on Learning Representations (ICLR 2019). 2019. 1−15.

[15] Calian DA, Mankowitz DJ, Zahavy T, Xu ZW, Oh J, Levine N, Mann TA. Balancing constraints and rewards with meta-gradient D4PG. In: Proc. of the 9th Int'l Conf. on Learning Representations (ICLR 2021). 2021. 1−10.

[16] Alshiekh M, Bloem R, Ehlers R, Könighofer B, Niekum S, Topcu U. Safe reinforcement learning via shielding. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence (AAAI 2018). Palo Alto: AAAI, 2018. 2669−2678.

[17] Sibai H, Potok M, Mitra S. Safe reinforcement learning for control systems: A hybrid systems perspective and case study. In: Prabhakar P, Ozay N, eds. Proc. of the ACM Hybrid Systems Computation and Control (HSCC 2019). New York: ACM, 2019. 1−9.

[18] Jansen N, Könighofer B, Junges S, Serban A, Bloem R. Safe reinforcement learning using probabilistic shields. In: Konnov I, Kovács L, eds. Proc. of the 31st Int'l Conf. on Concurrency Theory (CONCUR 2020). Dagstuhl Publishing, 2020. 1−16.

[19] Turchetta M, Kolobov A, Shah S, Krause A, Agarwal A. Safe reinforcement learning via curriculum induction. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, eds. Advances in Neural Information Processing Systems 33 (NeurIPS 2020). 2020. 1−12.

[20] Simão TD, Laroche R, Combes RT. Safe policy improvement with an estimated baseline policy. In: An B, Yorke-Smith N, Seghrouchni EF, Sukthankar G, eds. Proc. of the 19th Int'l Conf. on Autonomous Agents and Multi-agent Systems (AAMAS 2020). 2020. 1269−1277.

[21] Laroche R, Trichelair P, Combes RT. Safe policy improvement with baseline bootstrapping. In: Chaudhuri K, Salakhutdinov R, eds. Proc. of the 36th Int'l Conf. on Machine Learning (ICML 2019). 2019. 1−10.

[22] Simão TD, Spaan MTJ. Structure learning for safe policy improvement. In: Kraus S, ed. Proc. of the 28th Int'l Joint Conf. on Artificial Intelligence (IJCAI 2019). 2019. 3453−3459.

[23] Saunders W, Sastry G, Stuhlmueller A, Evans O. Trial without error: towards safe reinforcement learning via human intervention. In: Dastani M, Sukthankar G, André E, Koenig S, eds. Proc. of the 17th Int'l Conf. on Autonomous Agents and Multi-agent Systems (AAMAS 2018). 2018. 2067−2069.

[24] Fulton N, Platzer A. Safe reinforcement learning via formal methods: Toward safe control through proof and learning. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence (AAAI 2018). 2018. 6485−6492.

[25] Deshmukh JV, Kapinski JP, Yamaguchi T, Prokhorov D. Learning deep neural network controllers for dynamical systems with safety guarantees. In: Pan D, ed. Proc. of the 2019 IEEE/ACM Int'l Conf. on Computer-aided Design (ICCAD 2019). IEEE, 2019. 1−7.

[26] Chow Y, Nachum O, Faust A, Dueñez-Guzman E, Ghavamzadeh M. Safe policy learning for continuous control. In: Kober J, Ramos F, Tomlin C, eds. Proc. of the 2020 Conf. on Robot Learning (CoRL 2020). 2020. 801−821.

[27] Berkenkamp F, Turchetta M, Schoellig AP, Krause A. Safe model-based reinforcement learning with stability guarantees. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, eds. Advances in Neural Information Processing Systems 30 (NIPS 2017). 2017. 1−11.

[28] Choi J, Castañeda F, Tomlin CJ, Sreenath K. Reinforcement learning for safety-critical control under model uncertainty, using control Lyapunov functions and control barrier functions. In: Proc. of the Robotics: Science and Systems 2020 (RSS 2020). 2020. 1−9.

[29] Cheng R, Orosz G, Murray RM, Burdick JW. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In: Proc. of the 33rd AAAI Conf. on Artificial Intelligence (AAAI 2019). 2019. 3387−3395.

[30] Sutton RS, Barto AG. Reinforcement Learning: An Introduction. 2nd ed., Cambridge: MIT Press, 2018. 47−71.

[31] Bertsekas DP. Constrained Optimization and Lagrange Multiplier Methods. Belmont: Athena Scientific, 1982. 96−156.

[32] Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Continuous control with deep reinforcement learning. In: Proc. of the 4th Int'l Conf. on Learning Representations (ICLR 2016). 2016. 1−14.

[33] Conn AR, Gould NIM, Toint PL. Lancelot: A Fortran Package for Large-scale Nonlinear Optimization. Berlin, Heidelberg: Springer, 1992. 128−132.

[34] Gao S, Kong S, Clarke EM. dReal: An SMT solver for nonlinear theories over the reals. In: Bonacina MP, ed. Proc. of the 24th Int'l Conf. on Automated Deduction (CADE 2013). Berlin, Heidelberg: Springer, 2013. 208−214.

附中文参考文献:

[2] 白云军, 甘庭, 焦莉, 薛白, 詹乃军. 时延混成系统的切换控制器合成. 中国科学: 数学, 2021, 51(1): 97−114.

[6] 包为民, 祁振强, 张玉. 智能控制技术发展的思考. 中国科学: 信息科学, 2020, 50(8): 1267−1272.

赵恒军(1985－), 男, 博士, 讲师, CCF 专业会员, 主要研究领域为信息物理系统, 形式化方法.

曾霞(1987－), 女, 博士, 讲师, 主要研究领域为信息物理系统, 数值符号计算.

李权忠(1995－), 男, 硕士生, 主要研究领域为强化学习, 智能控制.

刘志明(1961－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为软件理论与方法.