中国科学 : 信息科学　2022 年　第 52 卷　第 9 期: 1672–1686
SCIENTIA SINICA Informationis
《中国科学》杂志社
SCIENCE CHINA PRESS
面向特殊应用场景的无人机智能决策与控制专刊 · 论文

# 基于安全自适应强化学习的自主避障控制方法

王珂[1], 穆朝絮[1*], 蔡光斌[2], 汪韧[3], 孙长银[4]

1. 天津大学电气自动化与信息工程学院, 天津 300072
2. 火箭军工程大学导弹工程学院, 西安 710025
3. 中国运载火箭技术研究院研究发展部, 北京 100076
4. 东南大学自动化学院, 南京 210096
* 通信作者. E-mail: cxmu@tju.edu.cn

收稿日期: 2022–01–31; 修回日期: 2022–04–06; 接受日期: 2022–05–15; 网络出版日期: 2022–09–14

国家重点研究发展计划 (批准号: 2021YFB1714700) 和国家自然科学基金 (批准号: 62022061) 资助项目

**摘要** 障碍规避是无人机等自主无人系统运动规划的重要环节, 其核心是设计有效的避障控制方法. 为了进一步提高决策优化性和控制效果, 本文在最优控制的设定下, 提出一种基于强化学习的自主避障控制方法, 以自适应方式在线生成安全运行轨迹. 首先, 利用障碍函数法在代价函数中设计了一个光滑的奖惩函数, 从而将避障问题转换为一个无约束的最优控制问题. 然后, 利用行为 – 评价神经网络和策略迭代法实现了自适应强化学习, 其中评价网络利用状态跟随核函数逼近代价函数, 行为网络给出近似最优的控制策略; 同时, 通过状态外推法获得模拟经验, 使得评价网络能利用经验回放实现可靠的局部探索. 最后, 在简化的无人机系统和非线性数值系统上进行了仿真实验与方法对比, 结果表明, 提出的避障控制方法能实时生成较优的安全运行轨迹.

**关键词** 自主无人系统, 避障控制, 强化学习, 神经网络, 经验回放

## 1 引言

以无人机、无人车、机器人等为代表的自主无人系统已经得到普遍关注和初步应用, 为了执行复杂任务, 需要这类运动对象能够进行快速决策并实现精准控制 [1]. 特别是面向巡逻、搜寻、侦察等特殊应用场景, 工作环境中存在着各种障碍物, 要求无人系统完成既定任务的同时避免与障碍物发生碰撞. 因此, 设计自主避障控制策略以生成安全轨迹, 是运动规划与决策的重要环节 [2].

根据是否完全利用模型信息, 避障方法大体上可以分为两类: 第 1 类是预生成方法, 典型的如 A* 算法和 Dijkstra 算法, 这种方法利用整个环境模型来寻求最佳安全轨迹, 是全局规划但不具备实时调整性 [3]; 第 2 类是反应式方法, 典型的如人工势场法, 此时智能体仅利用有限的检测数据寻找安全轨

**引用格式:** 王珂, 穆朝絮, 蔡光斌, 等. 基于安全自适应强化学习的自主避障控制方法. 中国科学: 信息科学, 2022, 52: 1672–1686, doi: 10.1360/SSI-2022-0054
Wang K, Mu C X, Cai G B, et al. Autonomous obstacle avoidance control method based on safe adaptive reinforcement learning (in Chinese). Sci Sin Inform, 2022, 52: 1672–1686, doi: 10.1360/SSI-2022-0054

迹, 是局部规划但易于在线实施 [4~6]. 例如文献 [4] 通过构建势场环境模型实现了无人车的避障规划, 文献 [5] 则提出了多约束模型预测控制方法以解决障碍规避问题. 显然, 在特殊应用场景中, 反应式方法具备更大优势, 因为它提供了实现最优实时反馈的可能性. 然而, 已报道的大多数实时方法要么没有考虑策略的优化性, 要么仅能考虑几个时间步上的优化性. 特别是涉及复杂非线性的时候, 最优避障控制器的设计更具挑战性.

近年来, 强化学习与最优控制的融合发展为解决这个问题提供了新颖思路, 典型的一种方法便是自适应动态规划, 它利用神经网络实现函数逼近并借助强化学习过程实现最优控制策略的近似求解 [7]. 通过使用自适应权值更新律, 自适应动态规划本质上对应着一个自适应强化学习过程. 本文考虑策略迭代法实现这个强化学习过程. 根据控制策略表现不同, 可以分为同策略 (on-policy) 学习和异策略 (off-policy) 学习. 前者进行策略评估后立即执行, 具有良好的自适应能力, 例如文献 [8, 9] 在线获得了扰动系统的鲁棒控制策略; 后者依据行为策略产生的数据进行离线学习, 易于与经验回放结合、数据利用率较高 [10], 例如文献 [11] 实现了多个异构旋翼无人机的无模型姿态同步. 值得强调的是, 这些典型的学习方法通常需要大量的基函数进行大范围探索, 以获得一个近似最优的控制策略; 同时, 需要给系统注入一个微小探测噪声以满足激励条件, 这无可避免地为系统带来了震荡现象 [12]. 这些条件对大多数控制任务影响有限, 但由于潜在的安全性威胁, 与避障控制是背道而驰的. 最近, 文献 [13] 提出了一种基于状态跟随 (state-following, StaF) 核函数的自适应动态规划方法, 该方法在当前状态的邻域内对代价函数进行局部逼近, 有效提高了计算效率. 可以发现, StaF 为避障控制提供了一种可能性, 因为智能体仅仅通过局部探索就可以完成策略优化.

关于自适应动态规划实现最优避障控制, 已经有一些方案被提出, 见文献 [14~16] 等. 例如文献 [15] 基于文献 [17] 中的避让函数, 在代价函数中设计了一个障碍惩罚项, 保证了自适应学习过程中控制策略的安全性. 但是这些方案的避障策略偏于保守、数学定义不够清晰, 优化性能仍然有待提升. 另一方面, 障碍环境意味着无人系统或智能体的状态空间是受到限制的, 从数学角度来说, 避障问题就是一个受约束的策略优化问题 [18]. 因此, 可以借助障碍函数 (barrier function, BF) 法设计更具解释性的障碍惩罚项, 进而增加学习过程的可预测性和可控性.

基于上述介绍和问题讨论, 本文在最优控制意义下针对自主无人系统的避障控制问题展开研究, 基于 StaF 和 BF 提出一种安全自适应强化学习 (safe adaptive reinforcement learning, SARL) 方法, 通过状态外推和经验回放实现对障碍物的有效规避. 主要贡献在于: (1) 结合障碍函数, 设计了一种新颖的奖惩函数, 从而将避障问题转换为一个无约束的最优控制问题. (2) 基于行为 – 评价神经网络 (actor-critic neural network, ACNN) 结构实现了自适应强化学习, 其中评价网络利用 StaF 核函数逼近代价函数, 行为网络通过投影算子法给出安全的避障控制策略. (3) 在简化的无人机系统上进行了仿真测试, 并与其他控制方法进行了对比评估; 结果表明, 提出方法能够实时生成安全轨迹并具有明显的相对优势.

本文后续部分的安排为: 第 2 节对自主系统的避障问题进行描述, 并完成最优控制问题转换. 第 3 节介绍 SARL 的设计原理, 包括神经网络、状态外推与经验回放、自适应权值律等. 第 4 节讨论系统稳定性与方法特点. 第 5 节通过 3 个算例验证避障控制方法的有效性并进行对比评估. 最后, 第 6 节总结全文并给出研究展望.

## 2 避障问题描述与最优控制

本文涉及的自主系统利用一类仿射非线性系统表示, 因其自主性和决策控制一体性, 后续亦将其

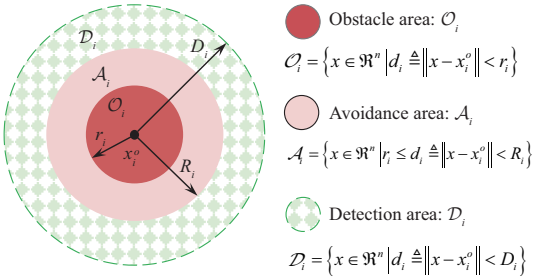


图 1 (网络版彩图) 障碍区、避让区和检测区示意

**Figure 1** (Color online) Obstacle area, avoidance area, and detection area

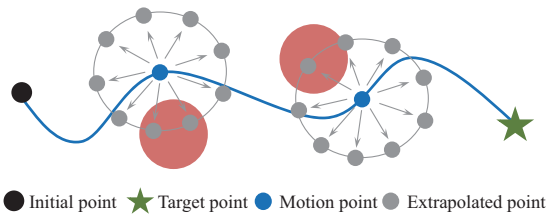


图 2 (网络版彩图) 外推获得模拟经验的示意图

**Figure 2** (Color online) Obtaining simulated experience by extrapolation

称之为智能体, 在此基础上开发的方案亦适用于线性情形. 此外, 不失一般性地认为系统具备自主探测能力且探测范围是有限的, 因此只有障碍物进入智能体的检测半径内才会被发现.

### 2.1 系统模型与障碍环境

考虑这样的一类自主系统或智能体, 由如下仿射非线性模型表征:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \tag{1}$$

其中 $x \in \mathbb{R}^n$ 是系统状态, $f(x) : \mathbb{R}^n \to \mathbb{R}^n$ 表示内部动态信息, $g(x) : \mathbb{R}^n \to \mathbb{R}^{n\times q}$ 是控制耦合矩阵, $u(t) \in \mathbb{R}^q$ 是控制输入[1)]. 在运动规划问题中, 系统状态一般由位置表示, 此时系统模型便是运动学模型. 系统的操作域或状态空间定义为一个包含原点的紧集 $x \in \Omega \in \mathbb{R}^n$, 并假设 $f(x)$ 和 $g(x)$ 在这个紧集上是 Lipschitz 连续的. 在本文中, 原点被标定为目标点 (target point) 并且满足 $f(0) = 0$.

本文对障碍问题作如下的设定.

**假设1** 为便于说明, 障碍物被认为是静态物体, 障碍物的数量记为 $O_m$; 此外, 暂不考虑障碍物互相重叠的情形, 一旦重合则将其视为一个更大的障碍物.

**假设2** 以最大半径作为避障条件, 将障碍物看成一个圆形区域或球形区域, 其位置中心点记为 $x_i^o \in \mathbb{R}^n$. 因此, 第 $i$ 个障碍和智能体之间的距离用欧氏距离表示为 $d_i \triangleq d_i(x) = \|x - x_i^o\|$.

我们希望智能体只有在检测到障碍物后才采取规避动作, 同时能有效保证安全性. 基于此, 围绕障碍物定义 3 个区域, 如图 1 所示. 由外到内, 3 个区域分别为:

- 检测区域 (detection area). 进入此区域, 智能体将发现障碍物并应该采取避障动作.
- 避让区域 (avoidance area). 可以理解为缓冲区, 用以保证系统的安全性; 不鼓励智能体进入此区域, 一旦进入应采取较大的规避操作.
- 障碍区域 (obstacle area). 也就是危险区, 智能体一旦进入或发生任何碰撞行为则任务失败.

因此, 在多障碍环境中, 智能体对应的 3 个区域应为 $\mathcal{D} = \cup_{i\in O_m}\mathcal{D}_i$, $\mathcal{A} = \cup_{i\in O_m}\mathcal{A}_i$, $\mathcal{O} = \cup_{i\in O_m}\mathcal{O}_i$. 此外, 在后续分析中智能体将被视为一个运动的质点, 因此为了保证绝对安全性, 避让区的半径应大于无人系统的最大尺寸. 例如, 对一个半径为 0.2 m 的旋翼无人机, 对应避让区的宽度应满足 $(R_i - r_i) > 0.2$ m.

1) 注意, 为简洁起见, 在不引起混淆的情况下与时间有关的变量或函数中的 $t$ 将被省略, 例如, $g(x(t))$ 被简写为 $g(x)$.

紧接着, 用一个光滑函数 $h_i(x)$ 表征障碍区域, 即 $\mathcal{O}_i=\{x\in\mathbb{R}^n|h_i(x)=\|x-x_i^o\|-r_i<0\}$, 整个障碍区域可以进一步表示为 $\mathcal{O}=\{x\in\mathbb{R}^n|h(x)=\cup_{i\in O_m}h_i(x)<0\}$. 因此系统的安全域 $\mathcal{S}$ 可以定义为 $\mathcal{O}$ 在操作域 $\Omega$ 上的补集, 即 $\mathcal{S}=\Omega\setminus\mathcal{O}$, 进而有定义 1.

**定义1** 对于系统 (1) 而言, 如果存在一个控制策略 $u$ 总能使得 $x_0\triangleq x(0)\in\mathcal{S}\to x(t)\in\mathcal{S}$, $t>0$ 成立, 那么就说这个控制策略是系统 (1) 的安全控制策略, 它可以使得运行轨迹始终保持在障碍区域外部, 由此生成的轨迹称之为安全运行轨迹[2)].

### 2.2 最优避障控制

接下来, 展示如何将障碍问题和控制优化联系起来. 首先, 智能体的代价函数定义如下:

$$J(x,u)=\int_t^\infty\big(x(\tau),u(\tau)\big)\mathrm{d}\tau;\quad r(x,u)=x^{\mathrm{T}}Qx+u^{\mathrm{T}}Ru+\mathcal{B}(x),\tag{2}$$

其中第 1 项 $x^{\mathrm{T}}Qx$, $Q=Q^{\mathrm{T}}>0$ 是系统运行成本, 第 2 项 $u^{\mathrm{T}}Ru$, $R=R^{\mathrm{T}}>0$ 是控制成本, 第 3 项 $\mathcal{B}(x)$ 是设计的障碍函数. 基于上面讨论和文献 [18] 中的倒数型障碍函数方法, $\mathcal{B}(x)$ 被设计为光滑函数 $h(x)$ 的倒数形式, 满足如下性质:

$$\mathcal{B}(x)=\frac{K_s s(x)}{h(x)};\quad\lim_{x\to\mathcal{O}}\mathcal{B}(x)=\infty,\tag{3}$$

其中 $s:\mathbb{R}^n\to[0,1]$ 是一个自定义的调度函数[3)], 它可以确保障碍函数仅仅在智能体检测障碍物后才被激活, 这是与智能体的局部检测能力相契合的. 另外, $K_s>0$ 是一个常数增益, 用以调整障碍函数在代价函数中的分量. 至此, 控制目标可以表述如下.

**控制目标.** 对于状态反馈形式的控制策略 $u(t)=u(x)$, 希望通过最小化包含障碍函数 $\mathcal{B}(x)$ 的代价函数 (2), 将智能体驱动至目标点同时规避障碍物. 此时, 最优代价函数可以表示为

$$J^*(x)=\min_{u(\tau)}\int_t^\infty r\big(x(\tau),u(\tau)\big)\mathrm{d}\tau.\tag{4}$$

**注释1** 依据 3 个区域, $s(x)$ 被设计为一个光滑变换的函数, 满足下面的条件:

$$s(x)=\sum_{i=1}^{O_m}s_i(x),\quad s_i(x)=\begin{cases}0, & d_i>D_i,\\ l_1+l_1\cos\left(\pi\dfrac{d_i^2-R_i^2}{D_i^2-R_i^2}\right), & R_i<d_i\leqslant D_i,\\ l_2+l_3\cos\left(\pi\dfrac{d_i^2-r_i^2}{R_i^2-r_i^2}\right), & r_i<d_i\leqslant R_i,\\ 1, & d_i\leqslant r_i,\end{cases}\quad\begin{cases}l_2+l_3=1,\\ l_2-l_3=2l_1.\end{cases}\tag{5}$$

在调度函数中, $l_1,l_2,l_3\in[0,1]$ 3 个参数用以调整 $s(x)$ 在区间 $[r_i,D_i]$ 上的稀疏程度, 进而影响障碍函数的稀疏程度[4)]. 同时注意到, 当 $s(x)=0$ 的时候, 获得的控制策略便不会包含避障行为.

**注释2** 避障问题转化为一个无约束最优控制问题的关键在于障碍函数 $\mathcal{B}(x)$, 其本质上是一个奖惩函数. 在 $s(x)$ 的作用下, 只有智能体检测到障碍物后它才会被激活, 并且随着智能体靠近障碍区域会逐渐增大. 这种设置鼓励智能体远离障碍物以获得更小的惩罚, 从而遂行控制目标.

2) 本文涉及的安全性描述本质上与文献 [18] 中的前向不变集是类似的.

3) 关于这个函数具体性质, 见注释 1.

4) 当代价函数对应到强化学习中时, 这 3 个参数影响的其实就是奖励信号的稀疏程度.

接下来, 定义如下的 Hamilton 方程:

$$H(x,u,\nabla J^*)=r(x,u)+(\nabla J^*)^{\mathrm{T}}\big(f(x)+g(x)u\big), \tag{6}$$

其中 $\nabla J^*\triangleq\partial J^*(x)/\partial x$ 是最优代价函数关于状态的偏导数. 依据最优控制知识, 可以获得如下的最优控制策略:

$$u^*(x)=-\frac{1}{2}R^{-1}g^{\mathrm{T}}(x)\nabla J^*. \tag{7}$$

进一步地, 将最优控制策略 (7) 带入式 (6) 就可以得到 HJB (Hamilton-Jacobi-Bellman) 方程:

$$0=x^{\mathrm{T}}Qx+(\nabla J^*)^{\mathrm{T}}f(x)-\frac{1}{4}(\nabla J^*)^{\mathrm{T}}g(x)R^{-1}g^{\mathrm{T}}(x)\nabla J^*,\quad J^*(0)=0. \tag{8}$$

不难发现, 求解这个方程就可获得最优代价函数, 进而利用式 (7) 计算控制信号即可. 但当涉及非线性项时, 很难获得其解析解, 因此下面我们将利用神经网络和策略迭代法通过自适应学习的形式获得其近似解. 因为学习设计中考虑障碍惩罚以保证安全性, 故而称之为安全自适应强化学习.

## 3 自适应强化学习设计

本节主要介绍 SARL 的设计过程, 首先是 ACNN 近似器构建, 其次是外推探索与模拟经验获取, 最后给出自适应权值更新律.

### 3.1 行为 – 评价神经网络

本文提出的 SARL 算法利用策略迭代法实施, 其中评价 (critic) 网络近似代价函数, 其更新过程对应于策略评估; 行为 (actor) 网络给出近似最优控制策略, 其更新过程便是策略改进.

首先, 基于 StaF 核函数和 BF 方法, 最优代价函数和最优控制策略可以参数化为

$$J^*(x)=w_c^{\mathrm{T}}\varphi(x,c(x))+\mathcal{B}_o(x)+\varepsilon(x), \tag{9}$$

$$u^*(x)=-\frac{1}{2}R^{-1}g^{\mathrm{T}}(x)\Big(\nabla\varphi^{\mathrm{T}}(x,c(x))\,w_c+\nabla\mathcal{B}_o(x)+\nabla\varepsilon(x)\Big), \tag{10}$$

其中 $w_c\in\mathbb{R}^{\mathcal{L}}$ 是评价神经网络的理想权值. $\varphi(x,c(x)):\mathbb{R}^n\to\mathbb{R}^{\mathcal{L}}$ 是 StaF 核函数[5], 其中 $c(x)\in(B_r(x))^{\mathcal{L}}$ 是与当前状态相关的核, 以闭包 $B_r(x)$ 的形式包围当前的状态向量 [13]. 此外, $\varepsilon(x):\mathbb{R}^n\to\mathbb{R}$ 代表网络构造误差. 另外, 为了更好地在学习过程中体现避障属性, 在代价函数近似中加上、减去一个有界项 $\mathcal{B}_o(x)$[6]. 在障碍函数 $\mathcal{B}(x)$ 的基础上, $\mathcal{B}_o(x)$ 被设置为

$$\mathcal{B}_o(x)=\frac{K_s s(x)}{h(x)+\alpha}, \tag{11}$$

其中 $\alpha>0$ 是一个小常数, 用以防止零分母情形, 进而使得障碍惩罚项 $\mathcal{B}_o(x)$ 是有界的. 考虑到代价函数中包含了一项间歇性引入的奖惩函数, 因此在评价网络近似中引入这一项可以更好地刻画代价函数变化的间歇性, 使得代价函数的逼近更加准确.

但是, 在实际学习过程中理想权值不可得的, 因此可以使用下式对代价函数进行逼近:

$$\hat{J}(x)=\hat{w}_c^{\mathrm{T}}\varphi(x,c(x))+\mathcal{B}_o(x), \tag{12}$$

5) 这个核函数相当于经典自适应动态规划中的激活函数或基函数, 不同的是, StaF 核函数更关注当前状态的邻域.
6) 在重构误差 $\varepsilon(x)$ 中, 其实已经隐含了 $-\mathcal{B}_o(x)$ 这一项.

其中 $\hat{w}_c \in \mathbb{R}^{\mathcal{L}}$ 是理想权值 $w_c$ 的近似值, 亦是真实的评价网络权值.

接着, 在评价网络的基础上构建行为网络, 利用文献 [7] 中的设计, 控制器进一步参数化为

$$\hat{u}(t) \triangleq \hat{u}(x, c(x), \hat{w}_a) = -\frac{1}{2} R^{-1} g^{\mathrm{T}}(x) \Big( \nabla \varphi^{\mathrm{T}}(x, c(x)) \hat{w}_a + \nabla \mathcal{B}_o(x) \Big), \tag{13}$$

其中 $\hat{w}_a \in \mathbb{R}^{\mathcal{L}}$ 是行为网络权值, 同样逼近理想权值 $w_c$. 这个近似最优的控制策略将被用来执行策略迭代中的策略改进环节. 同时注意到, 这个控制策略耦合了障碍函数项的偏导数, 这种形式使得控制器仅在检测到障碍后才采取避让行为, 其余时刻进行轨迹优化.

在得到近似代价函数 $\hat{J}(x)$ 和近似控制策略 $\hat{u}(t)$ 后, 将它们带入 HJB 方程可以得到实时学习误差

$$\delta_e(t) \triangleq \delta_e(x, c(x), \hat{w}_c, \hat{w}_a) = \Big( \nabla \varphi^{\mathrm{T}}(x, c(x)) \hat{w}_c + \nabla \mathcal{B}_o(x) \Big)^{\mathrm{T}} \big( f(x) + g(x) \hat{u}(t) \big) + r(x, \hat{u}(t)). \tag{14}$$

这个误差亦称为贝尔曼误差 (Bellman error, BE), 通过将其最小化至零便可实现对神经网络的训练. 这个误差是在真实的系统轨迹上生成的, 是 on-policy 数据, 可以理解为智能体的实时经验. 为了促进学习效果, 下面介绍如何通过状态外推以获得模拟经验.

### 3.2 基于状态跟随的外推探索

从障碍惩罚项的设计看出, 智能体接近障碍物时会获得较大的负面奖励 (即惩罚), 但是在真实的避障中, 这种带有风险性的试错行为是不可行的. 因此本文考虑利用状态外推的方式来获取模拟经验, 这个过程的简单示意如图 2 所示. 可以看出, 通过以当前状态为核心进行外推, 可以得到许多虚拟状态, 虚拟状态可能会接触到障碍物, 相应地产生较大的负面奖励, 使得智能体可以提前调整控制策略, 进而安全地经过障碍物. 不难理解, 外推状态对应的是虚拟轨迹, 所模拟得到的经验是 off-policy 数据.

在具体操作中, 让智能体在当前状态的闭包邻域内通过外推生成 off-policy 轨迹 $\{x_k \in B_r(x(t))\}_{k=1}^N$, 并且在这些轨迹对应的采样点 $x_k$ 对贝尔曼误差进行外推评估. 于是, 外推贝尔曼误差可以计算为

$$\delta_{e,k}(t) = \Big( \nabla \varphi^{\mathrm{T}}(x_k, c(x)) \hat{w}_c + \nabla \mathcal{B}_o(x_k) \Big)^{\mathrm{T}} \big( f(x_k) + g(x_k) \hat{u}_k(t) \big) + r(x_k, \hat{u}_k(t)), \tag{15}$$

其中外推控制策略 $\hat{u}_k(t)$ 为

$$\hat{u}_k(t) = -\frac{1}{2} R^{-1} g^{\mathrm{T}}(x_k) \Big( \nabla \varphi^{\mathrm{T}}(x_k, c(x)) \hat{w}_a + \nabla \mathcal{B}_o(x_k) \Big). \tag{16}$$

**注释3** 如图 2 所示, 外推的虚拟状态 $x_k$ 可能会碰到障碍甚至进入障碍区域内部, 因此外推的障碍惩罚项可能会出现 $h(x) = 0$ 的情况, 这也就是为什么要在神经网络近似中引入非零正常数 $\alpha$.

**注释4** 从式 (15) 中可以看出, 核向量 $c(x)$ 没有实行外推, 因为其与当前状态紧密相关. 如果 StaF 核函数的核也被外推的话, 这种局部探索便失去了参考性, 由此模拟得到的经验也是无效的.

### 3.3 自适应权值更新律设计

获得模拟经验后, 接下来至关重要的事便是设计自适应权值更新律以最小化下面的学习误差总和:

$$\delta_{e,\mathrm{sum}} = \frac{1}{2} \delta_e^{\mathrm{T}} \delta_e + \frac{1}{2} \sum_{k=1}^{N} \delta_{e,k}^{\mathrm{T}} \delta_{e,k}. \tag{17}$$

这个误差和可以理解为一般强化学习中的损失函数, 由实时误差和外推误差的平方和组成.

依据上述平方误差和, 采用梯度自适应下降的方式, 可以推导得到评价网络权值更新律为

$$\dot{\hat{w}}_c = -k_{c1}F_c\frac{\phi_\varphi(t)}{(\phi_\varphi^{\mathrm{T}}(t)\phi_\varphi(t)+1)^2}\delta_e(t) - \frac{k_{c2}}{N}F_c\sum_{k=1}^{N}\frac{\phi_{\varphi,k}(t)}{(\phi_{\varphi,k}^{\mathrm{T}}(t)\phi_{\varphi,k}(t)+1)^2}\delta_{e,k}(t), \tag{18}$$

其中常数 $k_{c1}, k_{c2} > 0$ 是评价网络学习率, 亦可用于调整实时数据和经验数据在权值更新律中的分量; $F_c \in \mathbb{R}^{\mathcal{L}\times\mathcal{L}}$ 是常正定矩阵. 此外, $\phi_\varphi(t) = \nabla\varphi^{\mathrm{T}}(x, c(x))(f(x)+g(x)\hat{u}(t))$ 代表回归向量; 相应地, 外推回归向量为 $\phi_{\varphi,k}(t) = \nabla\varphi^{\mathrm{T}}(x_k, c(x))(f(x_k)+g(x_k)\hat{u}_k(t))$. 从式 (18) 亦不难看出, 由于既包括真实轨迹上产生的实时数据, 也考虑了虚拟轨迹上获得的模拟经验数据, 因此评价网络的更新过程可以看作是 on-policy 和 off-policy 的融合.

接下来, 设计行为网络更新律. 考虑到行为网络权值与控制信号的大小紧密相关, 为了防止过大的控制输入, 希望将行为权值约束到一个较小的凸集内. 因此使用文献 [19] 中的梯度投影法, 行为网络权值更新律可以设计为

$$\dot{\hat{w}}_a \triangleq \mathrm{proj}\{-k_aF_a(\hat{w}_a - \hat{w}_c)\} = \mathrm{proj}\{-k_a\Gamma_a\}, \tag{19}$$

其中 $k_a > 0$ 是行为网络学习率, $F_a$ 是一个正定常值矩阵; 此外, 'proj{·}' 是一个投影算子, 代表一种映射规则. 这个投影规则是可以自定义的, 不同的规则会获得不同的权值更新律. 例如, 在本文设计中, 希望将行为权值约束到这样的一个凸集 $\{\hat{w}_a \in \mathbb{R}^{\mathcal{L}} \mid \gamma(\hat{w}_a) \leqslant 0\}$ 内, 其中 $\gamma(\hat{w}_a)$ 是一个光滑函数. 此时, 就可以按照如下规则进行投影:

$$\dot{\hat{w}}_a = \mathrm{proj}\{-k_a\Gamma_a\} = \begin{cases} -k_a\Gamma_a, & \text{if } \big(-k_a\Gamma_a\big)^{\mathrm{T}}\nabla\gamma_w \leqslant 0, \\ -k_a\Gamma_a + k_a\dfrac{\nabla\gamma_w\nabla\gamma_w^{\mathrm{T}}}{\nabla\gamma_w^{\mathrm{T}}k_a\nabla\gamma_w}k_a\Gamma_a, & \text{others}, \end{cases} \tag{20}$$

其中 $\nabla\gamma_w \triangleq \partial\gamma(\hat{w}_a)/\partial\hat{w}_a$ 为偏导数.

至此, SARL 的设计全部完成, 算法的迭代过程主要体现为式 (18) 和 (20) 的自适应更新过程. 学习过程中的安全性由障碍惩罚项保证, 智能体围绕当前状态进行局部探索并通过状态外推获取经验数据.

**注释5** 值得注意的是, 本文涉及的经验回放既不同于文献 [20] 中的 on-policy 经验回放, 即经验数据在真实轨迹上获得; 亦不同于文献 [21] 中的 off-policy 经验回放, 即不需要利用一个行为策略 (behavior policy) 生成数据. 两种方式各有利弊, 本文考虑到它们的互补优势, 在避障控制问题中实现了两种方法的良好融合, 使得 SARL 方法可以在线执行的同时获得大量的经验数据. 这种经验数据是外推产生的, 因此也保证了计算效率. 简言之, SARL 算法的内核是 on-policy 迭代, 但是会结合 off-policy 数据.

## 4 稳定性说明与讨论

本文提出的自适应强化学习与避障控制方法可以在最终一致有界 (uniformly ultimately bounded, UUB) 的意义下建立闭环系统的稳定性和学习过程的收敛性, 需要满足下面两个假设或条件.

**假设3** 与 ACNN 网络相关的理想权值、核函数、构造误差及其关于状态的偏导数都是有界的.

**假设4** 两个回归向量 $\phi_\varphi(t)$ 和 $\phi_{\varphi,k}(t)$ 应该满足如下条件:

$$\vartheta_1 I_{\mathcal{L}} \leqslant \int_t^{t+T}\left(\frac{\phi_\varphi(\tau)\phi_\varphi^{\mathrm{T}}(\tau)}{\eta^2(\tau)}\right)\mathrm{d}\tau, \quad \vartheta_2 I_{\mathcal{L}} \leqslant \frac{1}{N}\sum_{k=1}^{N}\frac{\phi_{\varphi,k}(t)\phi_{\varphi,k}^{\mathrm{T}}(t)}{\eta_k^2(t)}, \tag{21}$$

其中 $\vartheta_1, \vartheta_2$ 是非负常数并且至少一个严格为正; 此外, $\eta(t) \triangleq \phi_{\varphi}^{\mathrm{T}}(t)\phi_{\varphi}(t)+1$, $\eta_k(t) \triangleq \phi_{\varphi,k}^{\mathrm{T}}(t)\phi_{\varphi,k}(t)+1$.

然后, 将网络权值误差记为 $\tilde{w}_c = w_c - \hat{w}_c$, $\tilde{w}_a = w_c - \hat{w}_a$, 选取如下的 Lyapunov 函数:

$$L(t) = J^*(x) + \frac{1}{2}\tilde{w}_c^{\mathrm{T}} F_c^{-1}\tilde{w}_c + \frac{1}{2}\tilde{w}_a^{\mathrm{T}} k_a^{-1}\tilde{w}_a. \tag{22}$$

通过对其取时间导数, 并执行相关的数学操作即可获得系统稳定所需满足的 UUB 条件, 整个过程类似于文献 [13] 的做法. 因篇幅所限, 本文不再作详细展示.

需要强调的是, 假设 4 本质上是一种持续激励 (persistently exciting, PE) 条件. 在一般自适应动态规划方法中, 满足该条件的经典做法是在控制信号中注入探测噪声, 但这会带来震荡和安全隐患, 实际在线运行时也是很难实现的. 在本文中, 通过选择足够多 ($N \gg \mathcal{L}$) 且合适的 off-policy 轨迹 (例如均匀分布采样、正态分布采样) 就可以满足这个条件. 换言之, 虚拟探索本身就是一个持续激励过程.

## 5 数值仿真实验与分析

本节通过在 3 个无人系统上分别进行仿真实验说明提出 SARL 方法的可靠性与优势, 以证明智能体能够自主运动至目标点, 并能规避多个障碍物. 仿真环境基于 MATLAB 2020a, 仿真步长为 0.01 s.

第 1 组仿真实验在一个简化的二维旋翼无人机系统上进行, 此时认为无人机做平飞运动, 不涉及高度变化. 这组仿真实验考虑了 3 种情形, 包括方法对比、奖励参数变化、控制约束情形.

第 2 组仿真实验在一个非线性数值系统上进行, 用以测试提出方法对复杂非线性的表现.

第 3 组仿真实验在一个简化的三维旋翼无人机系统上进行, 此时障碍对应为三维球体.

### 5.1 二维坐标系下的无人机避障

这组实验的仿真对象为式 (1) 表示的简化无人机系统, 其中 $f(x) = 0_{2\times 1}$, $g(x) = I_2$, 即 $\dot{x} = u$ [15]. 其中 $x \in \mathbb{R}^2$ 是二维坐标系下的位置状态向量, $u \in \mathbb{R}^2$ 是无人机的速度控制指令. 我们考虑了 3 种情形, 情形一中测试多障碍环境下的避障控制性能, 并通过与其他控制方法对比以评估提出方法的优势; 情形二中对调度函数中的参数进行仿真分析, 以观察不同奖励设置对避障结果的影响; 情形三中考虑控制约束, 用以说明提出方法的扩展性.

这一组实验的主要参数为: $R = 5I_2$, $Q = I_2$, $K_s = 5$, $\alpha = 1$, $l_1 = 1/8$, $l_2 = 5/8$, $l_3 = 3/8$, $k_{c1} = 0.1$, $k_{c2} = 0.75$, $k_a = 0.75$, $F_a = 0.08I_3$, $F_c = 0.01 \times \mathbf{1}_{3\times 3}$. 对于评价网络来说, 激活函数设置为 $\varphi(x, c(x)) = [x^{\mathrm{T}}c_1(x), x^{\mathrm{T}}c_2(x), x^{\mathrm{T}}c_3(x)]^{\mathrm{T}}$, 其中 3 个核函数 $c_i(x) = x + \rho_i$ 满足

$$\rho_1 = 0.005 \times [0, 1]^{\mathrm{T}},\ \rho_2 = 0.005 \times [0.866, -0.5]^{\mathrm{T}},\ \rho_3 = 0.005 \times [-0.866, -0.5]^{\mathrm{T}}.$$

外推选在 $0.05 \times 0.05$ 上生成 25 个均匀分布的采样点. 对于行为网络, $\gamma(\hat{w}_a) = (\hat{w}_a + 2)(\hat{w}_a - 3)$, 这意味着行为权值将被约束到区间 $[-2, 3]$ 内. 此外, 初始点为 $x_0 = (7, 6.5)$, 目标点为 $x_e = (0, 0)$.

多障碍环境的设置如表 1 所示, 无人机的检测半径设置为 1 m, 即 $D_i = 1$ m.

#### 5.1.1 情形一: 多障碍环境与方法对比

在上述配置下, 智能体在多障碍环境中生成的安全轨迹如图 3(a) 所示, 可以看出智能体成功到达了目标点并且有效规避了所有障碍物; 并且基本不会进入避让区, 这得益于基于状态外推的虚拟探索和障碍惩罚项. 图 3(b) 中分别给出了评价网络权值 $\hat{w}_c(t)$、行为网络权值 $\hat{w}_a(t)$、调度函数 $s(x)$ 和与

**表 1　二维坐标系下的障碍设置 (单位: m)**

**Table 1**　Obstacle setting in the 2D-coordinate system (m)

| Obstacle location | Obstacle radius | Avoidance radius | Obstacle location | Obstacle radius | Avoidance radius |
|---|---|---|---|---|---|
| $x_1^o = (6, 5.1)$ | $r_1 = 0.4$ | $R_1 = 0.6$ | $x_2^o = (4.85, 4.2)$ | $r_2 = 0.5$ | $R_2 = 0.7$ |
| $x_3^o = (1.7, 1.85)$ | $r_3 = 0.45$ | $R_3 = 0.65$ | $x_4^o = (3.2, 3.5)$ | $r_4 = 0.55$ | $R_4 = 0.75$ |
| $x_5^o = (4.5, 2.6)$ | $r_5 = 0.45$ | $R_5 = 0.65$ | $x_6^o = (2, 4.5)$ | $r_6 = 0.5$ | $R_6 = 0.7$ |

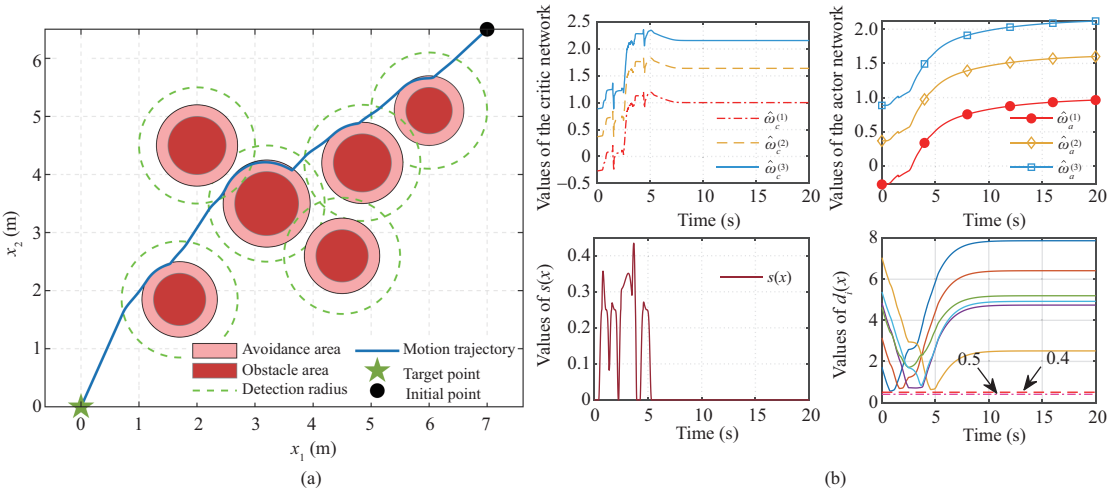


**图 3　(网络版彩图) 多障碍环境下无人机避障结果**

**Figure 3**　(Color online) Obstacle avoidance results of the drone in the multi-obstacle environment. (a) Safe motion trajectory; (b) main signals during the learning process

障碍物的实时距离 $d_i(t)$. 值得说明的是, 由于行为网络采用了梯度投影法, 行为权值的收敛过程更为平滑, 这使得生成的避障控制策略不会陡然变化.

接下来, 为了验证提出方法的优势, 我们考虑了两种对比方法. 对比方法 1 (comparison method 1) 采取与本文相同的避障设计, 但算法主体利用文献 [9] 中的 ADP (adaptive dynamic programming) 方法. 需要说明的是, 文献 [9] 提出了一种基于经验回放的并行学习方法, 但需要探测噪声以激励系统, 促进权值收敛; 因此对比方法 1 要在控制信号中注入微小的探测噪声 (正弦波和指数衰减信号的组合). 对比方法 2 (comparison method 2) 来自于文献 [15], 作者在代价函数中设计了一种类似于势场法的避障惩罚项, 其保守性较高、优化性不足. 同时, 为了进行量化评估, 定义 3 个评估指标, 分别为: (1) 路径长度 (path length, PL); (2) 距离目标点的误差 (error from target point, ETP): $\|x - x_e\|$; (3) 距第 $i$ 个障碍物边缘的最小距离 (minimum distance to the $i$-th obstacle edge, MDO$i$): $\min(d_i - r_i)$. PL 用以衡量控制策略在整个轨迹上的优化性能, 其值越小, 表明控制策略的优化性更好; ETP 用以衡量算法的收敛性能, 其值越小, 说明算法的局部优化性能较佳; MDO$i$ 用以衡量系统的安全性, 其值越大, 表明智能体离障碍物越远, 但并非越大越好, 面对障碍时该值保持在避让区宽度 $(R_i - r_i)$ 的附近最佳. 3 种方法的结果对比如图 4(a) 和表 2 所示.

不难看出, 提出方法能在保证安全性的前提下生成最短的运行轨迹, 同时距目标点的收敛误差最小. 这是因为本文提出方法是全阶段的策略优化, 并且利用障碍函数对安全域作出了清晰描述, 辅以虚拟探索, 使得智能体不必进入避让区 “太深”. 对比方法 2 确保了安全性, 但过于保守, 使得智能体

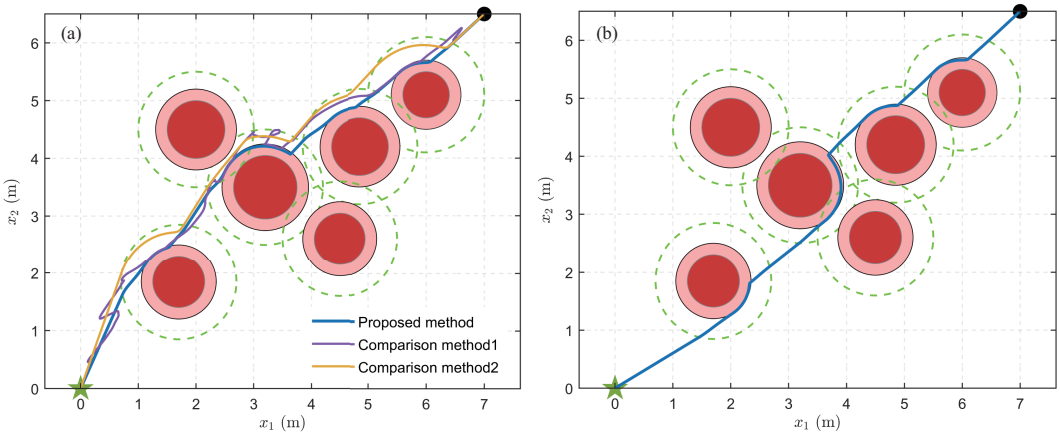


**图 4** (网络版彩图) (a) 不同方法生成的运行轨迹; (b) 不同参数下的安全轨迹

**Figure 4** (Color online) (a) Motion trajectories of different methods; (b) safe trajectory under different parameters

**表 2** 不同控制方法的对比结果 (单位: m)

**Table 2** Comparison results of different control methods (m)

| Method | PL | ETP | MDO1 | MDO2 | MDO3 | MDO4 | MDO5 | MDO6 |
|---|---|---|---|---|---|---|---|---|
| Proposed method | 10.1599 | 0.0015 | 0.1575 | 0.1820 | 0.1808 | 0.1559 | 1.2551 | 0.3264 |
| Comparison method 1 | 12.5701 | 0.0857 | 0.1795 | 0.3474 | 0.1902 | 0.1744 | 1.4214 | 0.3011 |
| Comparison method 2 | 10.3920 | 0.0148 | 0.4573 | 0.4130 | 0.4223 | 0.2405 | 1.4556 | 0.2716 |

面对障碍时会 “绕更大的圈子”. 此外, 由于探测噪声的缘故, 对比方法 1 的轨迹会发生微小震荡; 尽管利用本文的安全设计实现了避障功能, 但还是会出现走 “回头路” 的现象.

### 5.1.2 情形二: 调度函数 $s(x)$ 中的参数分析

在第 2 节中提到, 调度函数虽然在 0 到 1 之间平滑变换, 但通过增益 $K_s$, 就可以明显地影响障碍函数在代价函数中的分量, 进而改变强化学习中的奖励设置. 直观地说, 如果 $l_1$ 较小, 则避让区到检测区间的障碍奖惩会比较稀疏. 在这个情形中, 我们将调度函数 $s(x)$ 中的 3 个参数设置为 $l_1 = 1/20$, $l_2 = 21/40$, $l_3 = 19/40$, 其余参数与情形一保持一致.

这种情形下得到的安全轨迹如图 4(b) 所示. 可以看出, 此时智能体选择了一条完全不同的路径, 对应的运动路径长度为 10.2342 m. 在这种情形下, 尽管智能体依然可以安全绕过每个障碍, 但明显路径更长, 因此过于稀疏的奖励设置对优化性能还是会产生一定的负面影响.

### 5.1.3 情形三: 控制输入受到约束

另外, 在实际应用中, 由于电机转速、机构限制等客观条件, 使得速度指令不能过大, 即控制策略是受到约束的. 此情形中假设控制器的输出限幅为 $u_c = 1$, 即 $|u| \leqslant u_c$. 此时, 代价函数中控制成本项 $u^{\mathrm{T}}Ru$ 可以替换为

$$\Phi(u) = \int_0^u 2u_c \tanh^{-1}(\nu/u_c)^{\mathrm{T}} R \mathrm{d}\nu. \tag{23}$$

进一步地, 约束的学习控制策略按照下式计算:

$$\hat{u} = -u_c \tanh\left(\frac{1}{2u_c} R^{-1} g^{\mathrm{T}}(x)\left(\nabla\varphi^{\mathrm{T}}(x, c(x))\hat{w}_a + \nabla\mathcal{B}_o(x)\right)\right). \tag{24}$$

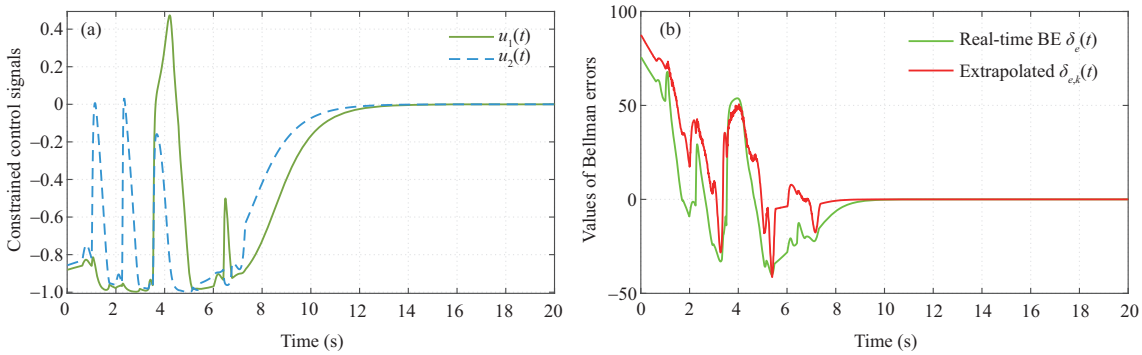


**图 5　(网络版彩图) (a) 受到约束的控制信号; (b) 实时贝尔曼误差和外推贝尔曼误差**

**Figure 5**　(Color online) (a) Constrained control input signals; (b) real-time BE and extrapolated BE

相应的仿真结果如图 5(a) 和 (b) 所示, 其中图 5(a) 给出了无人机的控制信号, 可以看出, 控制器的输出信号已经被约束到了 $[-1,1]$ 以内. 图 5(b) 中给出了贝尔曼误差的收敛过程, 其中实时 BE 和外推 BE 都能收敛至 0, 这说明自适应学习过程和外推探索过程都能稳定收敛.

### 5.2　非线性系统测试

此节中我们考虑一个带有复杂非线性的数值系统, 主要参数为: $R=10I_2$, $Q=5I_2$, $K_s=20$, $F_a=0.3I_3$, $F_c=0.001\times\mathbf{1}_{3\times3}$, 其余参数和障碍环境设置与 5.1 小节中的情形一保持一致.

该非线性系统为

$$\begin{aligned} f(x)&=\begin{bmatrix} -x_1+x_2 \\ -0.5x_1-0.5x_2\big(1-(\cos(2x_1)+2)^2\big)\end{bmatrix},\\ g(x)&=\begin{bmatrix}\sin(2x_1)+2 & 0\\ 0 & \cos(2x_1)+2\end{bmatrix},\\ x&=[x_1,x_2]^{\mathrm{T}}. \end{aligned}\tag{25}$$

相应的仿真结果如图 6 所示. 从图 6(a) 中可以看出, 提出方法依然可以为非线性系统找到一条安全的运行轨迹. SARL 的评价网络和行为网络的收敛结果分别如图 6(b) 和 (c) 所示, 可以看出, 行为权值的变化依旧比较平滑.

### 5.3　三维坐标系下的无人机避障

这组实验的仿真对象为式 (1) 表示的简化无人机系统, 其中 $f(x)=0_{3\times1}$, $g(x)=I_3$, 即 $\dot{x}=u$. 此时 $x\in\mathbb{R}^3$ 是三维坐标系下的位置状态向量, $u\in\mathbb{R}^3$ 是无人机的速度控制指令.

这一组实验的主要参数为: $R=5I_3$, $Q=I_3$, $K_s=40$, $\alpha=0.8$, $l_1=1/8$, $l_2=5/8$, $l_3=3/8$, $k_{c1}=0.1$, $k_{c2}=0.75$, $k_a=0.75$, $F_a=0.8I_4$, $F_c=0.1\times\mathbf{1}_{4\times4}$. 对于评价网络来说, 激活函数设置为 $\varphi(x,c(x))=[x^{\mathrm{T}}c_1(x),x^{\mathrm{T}}c_2(x),x^{\mathrm{T}}c_3(x),x^{\mathrm{T}}c_4(x)]^{\mathrm{T}}$, 其中 4 个核函数 $c_i(x)=x+\rho_i$ 满足 $\rho_1=0.05\times[1,0,0]^{\mathrm{T}}$, $\rho_2=0.05\times[-0.333,0,943,0]^{\mathrm{T}}$, $\rho_3=0.05\times[-0.333,-0.471,0.471]^{\mathrm{T}}$, $\rho_4=0.05\times[-0.333,-0.471,-0.471]^{\mathrm{T}}$. 外推选择在 $0.5\times0.5\times0.5$ 上生成 25 个均匀分布的采样点. 初始点为 $x_0=(9,8,9)$, 目标点为 $x_e=(0,0,0)$, 这个运动过程类似于一个自动降落过程.

多障碍环境的设置如表 3 所示, 无人机的检测半径设置为 1.4 m, 即 $D_i=1.4$ m.

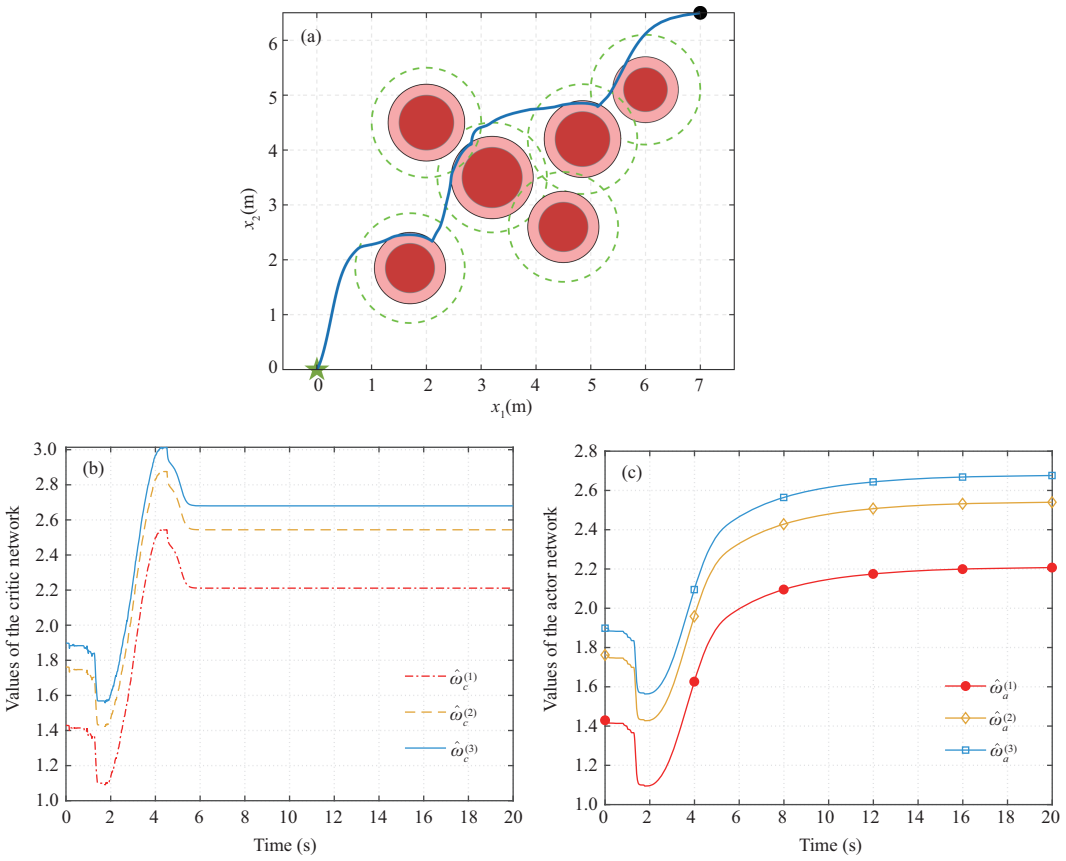


**图 6　(网络版彩图) 非线性系统的仿真结果**

**Figure 6**　(Color online) Simulation results of nonlinear systems. (a) Motion trajectory; (b) critic weights; (c) actor weights

**表 3　三维坐标系下的障碍设置 (单位: m)**

**Table 3**　Obstacle setting in the 3D-coordinate system (m)

| Obstacle location | Obstacle radius | Avoidance radius | Obstacle location | Obstacle radius | Avoidance radius |
|---|---|---|---|---|---|
| $x_1^o = (6, 5.5, 6)$ | $r_1 = 0.75$ | $R_1 = 1.05$ | $x_2^o = (3, 2, 3)$ | $r_2 = 0.8$ | $R_2 = 1.1$ |
| $x_3^o = (3, 6, 2)$ | $r_3 = 0.75$ | $R_3 = 1.05$ | $x_4^o = (5, 3.5, 4.4)$ | $r_4 = 0.85$ | $R_4 = 1.15$ |

三维环境下生成的运动轨迹如图 7 所示, 从两个不同的视角可以看出, 无人机可以成功地避开每个障碍物成功到达目标点; 另外可以看出, 无人机不会深入避让区太多, 从而有效保证了安全性. 图 8 分别给出了 AC 网络的权值信号、调度函数、外推的调度函数. 注意到, 针对这个三维系统, SARL 中每个网络仅需 4 个权值, 对应于 StaF 核函数拥有 4 个基函数分量; 相比之下, 经典自适应动态规划方法一般需要使用 6 个以上 [22], 若涉及到复杂非线性, 权值个数还将进一步增加. 另外还需注意的是, 从图 8(b) 中可以发现实时调度函数 $s(x)$ 并没有达到最大值 1, 这说明无人机本体不会碰到障碍物; 但是外推的调度函数 $s(x_k)$ 则会达到最大值 1 (如图 8(c) 所示), 这意味着外推的虚拟体确实会碰到障碍物, 这种对比反差恰好说明了提出方法中虚拟探索的可靠性.

**注释6**　从上述仿真案例中可以发现, StaF 核函数 $\varphi(x, c(x))$ 的选择至关重要. 这个核函数或基函

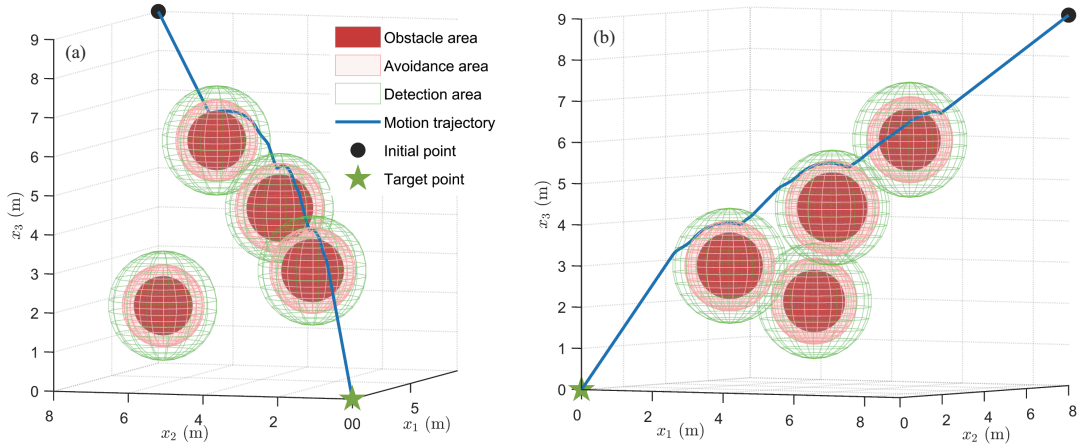


图 7 (网络版彩图) 三维坐标系下无人机避障过程

**Figure 7** (Color online) Obstacle avoidance process of the drone in the 3D-coordinate system. (a) First-view; (b) second-view

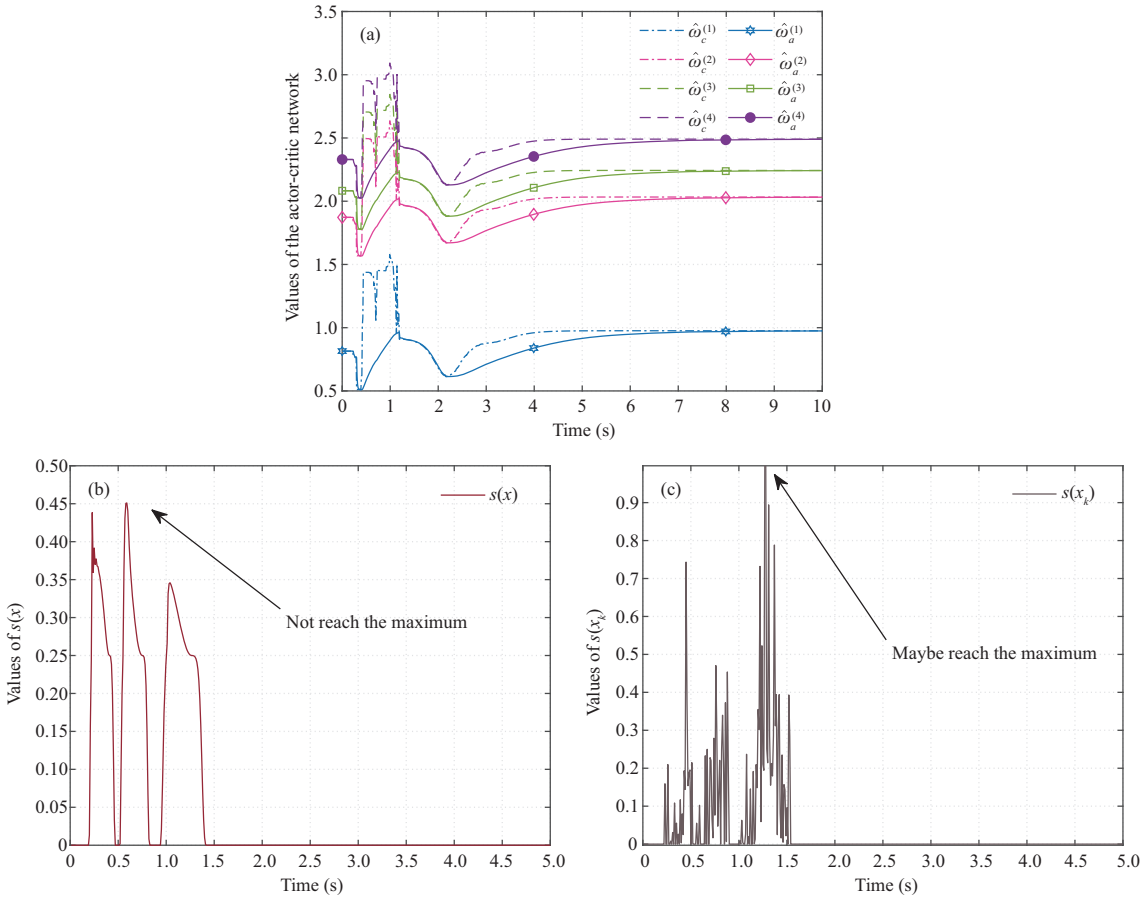


图 8 (网络版彩图) SARL 的主要运行结果

**Figure 8** (Color online) Main results generated by SARL. (a) AC network weights; (b) scheduling function $s(x)$; (c) extrapolated scheduling function $s(x_k)$

数被称为状态跟随的, 是因为它必须仅仅围绕当前状态, 且在当前状态的邻域附近. 为了实现这个目的, 建议选择为 $\varphi(x,c(x))=[x^{\mathrm{T}}c_1(x),\ldots,x^{\mathrm{T}}c_i(x),\ldots]^{\mathrm{T}}$, 其中每个核 $c_i(x)=x+\rho_i$. 此时, 只要让 $\rho_i$ 位于以原点为质心的 $n$- 单纯形 ($n$-simplex) 的顶点处即可. 因此, 对于一个 $n$ 维状态的系统, 至少需要使用 $n+1$ 个核函数, 即神经网络隐层的数量为 $\mathcal{L}=n+1$. 相比经典 ADP 方法, 提出方法所需的基函数的数量是更少的、形式也更容易确定; 再加之利用状态外推进行探索, 因而其计算效率亦相对较高.

综上, 我们可以得出结论: 提出的自主避障控制方法可以为无人系统在线生成安全轨迹, 其中状态外推可以确保探索的安全性; 方法的扩展性较强, 也可以处理复杂非线性情形; 障碍惩罚项易于数学解释, 通过参数配比可以实现灵活的奖励调整; 此外, 对比已有方法, 提出方法在避障效果、优化性能等方面均可表现出相对优势.

## 6 总结

本文主要提出了一种基于安全自适应强化学习的自主避障控制方法. 该方法基于最优控制思想设计, 在代价函数中引入了障碍惩罚项, 使得智能体可以在规避障碍物的同时获得近似最优的控制策略. 利用梯度自适应法和梯度投影法分别为评价网络和行为网络设计了自适应权值更新律, 并通过虚拟外推实现了可靠的局部探索. 在 3 个自主无人系统上的仿真结果验证了提出方法的有效性和优势. 后续工作将考虑移动障碍物、系统模型未知、复杂约束条件等问题; 同时, 还将在无人机硬件平台上对所提方法进行测试和改进.

## 参考文献


1 Lyu Y, Kang T N, Pan Q, et al. UAV sense and avoidance: concepts, technologies, and systems. Sci Sin Inform, 2019, 49: 520–537 [吕洋, 康童娜, 潘泉, 等. 无人机感知与规避: 概念、技术与系统. 中国科学: 信息科学, 2019, 49: 520–537]

2 Mkiramweni M E, Yang C, Li J, et al. A survey of game theory in unmanned aerial vehicles communications. IEEE Commun Surv Tut, 2019, 21: 3386–3416

3 Gonzalez D, Perez J, Milanes V, et al. A review of motion planning techniques for automated vehicles. IEEE Trans Intell Transp Syst, 2016, 17: 1135–1145

4 Gao L, Lu L P, Chu D F, et al. Multi-lane convoy control based on graph and potential field. Acta Autom Sin, 2020, 46: 117–126 [高力, 陆丽萍, 褚端峰, 等. 基于图与势场法的多车道编队控制. 自动化学报, 2020, 46: 117–126]

5 Ji J, Khajepour A, Melek W W, et al. Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints. IEEE Trans Veh Technol, 2017, 66: 952–964

6 Wu J F, Wang H L, Wang Y X, et al. UAV reactive interfered fluid path planning. Acta Autom Sin, 2021, 47: 1–16 [吴健发, 王宏伦, 王延祥, 等. 无人机反应式扰动流体路径规划. 自动化学报, 2021, 47: 1–16]

7 Khan S G, Herrmann G, Lewis F L, et al. Reinforcement learning and optimal adaptive control: an overview and implementation examples. Annu Rev Control, 2012, 36: 42–59

8 Mu C X, Wang K, Sun C Y. Learning control supported by dynamic event communication applying to industrial systems. IEEE Trans Ind Inf, 2021, 17: 2325–2335

9 Xue S, Luo B, Liu D R, et al. Constrained event-triggered $H_\infty$ control based on adaptive dynamic programming with concurrent learning. IEEE Trans Syst Man Cybern Syst, 2022, 52: 357–369

10 Li J, Chai T, Lewis F L, et al. Off-policy interleaved Q-learning: optimal control for affine nonlinear discrete-time systems. IEEE Trans Neural Netw Learn Syst, 2019, 30: 1308–1320

11 Zhao W B, Liu H, Wang B H. Model-free attitude synchronization for multiple heterogeneous quadrotors via reinforcement learning. Int J Intell Syst, 2021, 36: 2528–2547

12 Kamalapurkar R, Rosenfeld J A, Dixon W E. State following (StaF) kernel functions for function approximation part II: adaptive dynamic programming. In: Proceedings of American Control Conference (ACC), Chicago, 2015. 521–526

13 Rosenfeld J A, Kamalapurkar R, Dixon W E. The state following approximation method. IEEE Trans Neural Netw Learn Syst, 2019, 30: 1716–1730
14 Lan X J, Liu L, Wang Y. ADP-based intelligent decentralized control for multi-agent systems moving in obstacle environment. IEEE Access, 2019, 7: 59624–59630
15 Deptula P, Chen H Y, Licitra R A, et al. Approximate optimal motion planning to avoid unknown moving avoidance regions. IEEE Trans Robot, 2020, 36: 414–430
16 Zhu P, Liu C, Ferrari S. Adaptive online distributed optimal control of very-large-scale robotic systems. IEEE Trans Control Netw Syst, 2021, 8: 678–689
17 Rodríguez-Seda E J, Tang C, Spong M W, et al. Trajectory tracking with collision avoidance for nonholonomic vehicles with acceleration constraints and limited sensing. Int J Robot Res, 2014, 33: 1569–1592
18 Ames A D, Xu X, Grizzle J W, et al. Control barrier function based quadratic programs for safety critical systems. IEEE Trans Autom Control, 2017, 62: 3861–3876
19 Ioannou P A, Sun J. Robust Adaptive Control. New York: Dover Publications, Inc., 2012
20 Chowdhary G, Yucelen T, Mühlegg M, et al. Concurrent learning adaptive control of linear systems with exponentially convergent bounds. Int J Adapt Control Signal Process, 2013, 27: 280–301
21 Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay. In: Proceedings of the 4th International Conference on Learning Representations (ICLR), San Juan, 2016. 1–21
22 Mu C X, Zhang Y. Learning-based robust tracking control of quadrotor with time-varying and coupling uncertainties. IEEE Trans Neural Netw Learn Syst, 2020, 31: 259–273

# Autonomous obstacle avoidance control method based on safe adaptive reinforcement learning

Ke WANG[1], Chaoxu MU[1*], Guangbin CAI[2], Ren WANG[3] & Changyin SUN[4]

1. *School of Electrical and Information Engineering, Tianjin University, Tianjin* 300072*, China*;
2. *College of Missile Engineering, Rocket Force University of Engineering, Xi'an* 710025*, China*;
3. *R&D Center, China Academy of Launch Vehicle Technology, Beijing* 100076*, China*;
4. *School of Automation, Southeast University, Nanjing* 210096*, China*
* Corresponding author. E-mail: cxmu@tju.edu.cn

**Abstract** Obstacle avoidance is an important issue in the motion planning of autonomous unmanned systems. Therefore, designing an effective avoidance control method is crucial. For further improving the decision-making process, this paper presents a novel autonomous obstacle avoidance control method based on reinforcement learning that generates a safe motion trajectory in an adaptive manner. First, the barrier function is utilized to design a smooth penalty function in the cost function, thereby transforming the avoidance problem into an unconstrained optimal control problem. Then, adaptive reinforcement learning is implemented by using an actor-critic neural network architecture and policy iteration, in which the critic network uses the state-following kernel function to approximate the cost function while the actor network provides an approximate optimal control policy. During this learning process, the simulated experience is obtained through state extrapolation such that the critic network can use experience replay for reliable local exploration. Finally, simulation experiments on simplified drone systems and a nonlinear numerical system are provided. The proposed method can generate a safe motion trajectory in real time with comparable performance.

**Keywords** autonomous unmanned systems, obstacle avoidance control, reinforcement learning, neural networks, experience replay