文章编号: 1001-0920 (2013) 12-1889-05

一类非线性动态系统基于强化学习的最优控制

陈学松[a], 刘富春[b]

(广东工业大学 a. 应用数学学院, b. 计算机学院, 广州 510006)

摘 要: 提出一类非线性不确定动态系统基于强化学习的最优控制方法. 该方法利用欧拉强化学习算法估计对象的未知非线性函数, 给出了强化学习中回报函数和策略函数迭代的在线学习规则. 通过采用向前欧拉差分迭代公式对学习过程中的时序误差进行离散化, 实现了对值函数的估计和控制策略的改进. 基于值函数的梯度值和时序误差指标值, 给出了该算法的步骤和误差估计定理. 小车爬山问题的仿真结果表明了所提出方法的有效性.

关键词: 非线性动态系统; 强化学习; 最优控制; 值函数; 策略函数

中图分类号: TP273 **文献标志码:** A

Optimal control of a class of nonlinear dynamic systems based on reinforcement learning

CHEN Xue-song[a], *LIU Fu-chun*[b]

(a. School of Applied Mathematics, b. School of Computers, Guangdong University of Technology, Guangzhou 510006, China. Correspondent: CHEN Xue-song, E-mail: chenxs@gdut.edu.cn)

Abstract: An optimal control based on Euler reinforcement learning(ERL) is proposed for a class of nonlinear uncertain dynamic systems. In this method, the reinforcement learning algorithm is employed to approximate unknown nonlinear functions in the plant, and the online learning rule for the reward function and the policy function is derived. The value function is estimated and the control policy is improved by using the way of implementing the temporal difference(TD) errors which are discretized by using the forward Euler approximation of time derivative. Based on the value-gradient and TD error performance index, the steps of the algorithm and error estimation theorem are given. Simulation results for the mountain-car problem show the effectiveness of the presented method.

Key words: nonlinear dynamic system; reinforcement learning; optimal control; value function; policy function

0 引 言

强化学习 (RL) 是指智能体在与未知环境进行交互时, 通过由未知环境反馈的强化信号来学习状态空间到动作空间的最优映射关系[1]. 强化学习作为一类求解序贯优化决策问题的重要机器学习方法, 已经在机器人控制[2]、人工智能[3]和多智能体系统[4]领域得到了广泛应用, 已逐步成为涉及数学、控制、计算机等多学科交叉的热点研究方向.

目前, 大多强化学习算法都是在有限的离散马尔科夫决策过程基础上进行建模, 对于大规模的连续状态空间问题, 值函数的存储将导致维数灾难[5]. 为了解决这个问题, 函数逼近已经成为一种有效方法. 根据函数逼近器的特性, 强化学习方法可以分为两类: 一类是基于非线性结构的值函数逼近, 例如, 采用一个径向基函数网络对执行器的策略函数和评价器的值函数同时进行逼近, 这样便可通过执行器-评价器学习实现 PID 参数的自适应整定[6]; 另一类是基于线性结构的逼近, 例如, 采用线性参数估计理论, 提出了基于递推最小二乘法的多步时序差分学习算法, 证明了该算法的权值将以概率 1 收敛到唯一解, 并得出了值函数估计值的误差应满足的关系式[7]. Barto 等提出的自适应启发式评价算法, 也是一种重要的强化学习算法[8], 在人工智能和智能控制等领域得到了广泛应用. 该算法同时对马尔科夫决策过程中的值函数和策略函数进行逼近, 值函数估计的精度直接影响执行器的行为选择和策略梯度的估计, 因此, 评价器的值

收稿日期: 2012-08-01; **修回日期:** 2013-03-16.

基金项目: 国家自然科学基金项目(60974019, 61273118); 广东省高等学校高层次人才项目; 广东省自然科学基金项目(S2012010010570).

作者简介: 陈学松(1977–), 男, 讲师, 博士, 从事智能控制与人工智能的研究; 刘富春(1971–), 男, 教授, 从事离散事件系统的监督控制与故障诊断等研究.

函数预测性能对于算法求解学习控制问题的效率具有关键作用. 文献[9-10]研究了基于自适应动态规划的非线性系统最优控制方法. 但是, 自适应动态规划在实际应用中有待解决的难题是算法复杂, 学习速度较慢, 难以保证收敛, 而且收敛性的理论证明及其实际验证也是一个有待解决的难题.

本文研究一类非线性不确定动态系统的最优控制问题, 提出一种新的基于欧拉强化学习(ERL)算法的最优控制方法. 该控制方法首先利用强化学习算法对系统中的未知非线性函数进行逼近, 然后采用执行器实现系统状态到控制参数的映射, 评价器则对执行器的输出结果进行评判, 同时生成时序误差信号, 该信号采用向前欧拉差分迭代公式进行离散化. 基于值函数的梯度值和时序差分误差指标值, 得到了算法的步骤和误差估计定理. 将本文所设计的控制方法与文献[1]提出的基于模糊强化学习(FRL)的控制方法, 以及文献[11]提出的基于执行器-评价器的强化学习(AC-RL)控制方法进行了对比研究, 并通过小车爬山的仿真结果验证了本文方法的有效性和优越性.

1 问题描述

考虑如下非线性动态系统:

$$\dot{\boldsymbol{x}}(t)=f(\boldsymbol{x}(t),\boldsymbol{u}(t)). \tag{1}$$

其中: $\boldsymbol{x}\in\boldsymbol{X}\subset\boldsymbol{R}^n$ 表示状态, $\boldsymbol{u}\in\boldsymbol{U}\subset\boldsymbol{R}^m$ 表示动作. 定义瞬时回报如下:

$$r(t)=r(\boldsymbol{x}(t),\boldsymbol{u}(t)). \tag{2}$$

控制目标是寻找映射关系 π, 即策略

$$\boldsymbol{u}(t)=\pi(\boldsymbol{x}(t)), \tag{3}$$

使得未来的累计回报为最大, 有

$$V^{\pi}(\boldsymbol{x}(t))=\int_t^{\infty}\mathrm{e}^{-(s-t)/\tau}r(\boldsymbol{x}(s),\boldsymbol{u}(s))\mathrm{d}s. \tag{4}$$

其中: $V^{\pi}(\boldsymbol{x})$ 表示连续可导的值函数; $r(\boldsymbol{x}(s),\boldsymbol{u}(s))$ 表示连续可导的回报函数; $\boldsymbol{x}(s)(t\leqslant s\leqslant\infty)$ 表示满足方程(1)和(3)的状态函数; τ 表示时间常数, 它满足正则化式子 $\int_t^{\infty}\frac{1}{\tau}\mathrm{e}^{-(s-t)/\tau}\mathrm{d}s=1$.

最优策略 π^* 的最优值函数 V^* 定义为

$$V^*(\boldsymbol{x}(t))=\max_{\boldsymbol{u}[t,\infty)}\Big[\int_t^{\infty}\mathrm{e}^{-(s-t)/\tau}r(\boldsymbol{x}(s),\boldsymbol{u}(s))\mathrm{d}s\Big], \tag{5}$$

其中 $\boldsymbol{u}[t,\infty)$ 表示时间在 $t\leqslant s<\infty$ 上采取的动作. 根据积分区域可加性, 可以将式(5)中的广义积分分成 $[t,t+\Delta t]$ 和 $[t+\Delta t,\infty)$ 两部进行积分, 即

$$\begin{aligned}&V^*(\boldsymbol{x}(t))=\\&\max_{\boldsymbol{u}[t,\infty)}\Big[\int_t^{t+\Delta t}\mathrm{e}^{-(s-t)/\tau}r(\boldsymbol{x}(s),\boldsymbol{u}(s))\mathrm{d}s+\\&\int_{t+\Delta t}^{\infty}\mathrm{e}^{-(s-t)/\tau}r(\boldsymbol{x}(s),\boldsymbol{u}(s))\mathrm{d}s\Big]=\\&\max_{\boldsymbol{u}[t,t+\Delta t]}\Big[\int_t^{t+\Delta t}\mathrm{e}^{-(s-t)/\tau}r(\boldsymbol{x}(s),\boldsymbol{u}(s))\mathrm{d}s\Big]+\\&\max_{\boldsymbol{u}[t+\Delta t,\infty)}[\mathrm{e}^{-\Delta t/\tau}V^*(\boldsymbol{x}(t+\Delta t))].\end{aligned} \tag{6}$$

当 Δt 充分小时, 式(6)的第1项在一定条件下等于

$$r(\boldsymbol{x}(t),\boldsymbol{u}(t))\Delta t+o(\Delta t), \tag{7}$$

式(6)的第2项用1阶Taylor近似展开为

$$V^*(\boldsymbol{x}(t+\Delta t))=V^*(\boldsymbol{x}(t))+\frac{\partial V^*}{\partial\boldsymbol{x}(t)}f(\boldsymbol{x}(t),\boldsymbol{u}(t))\Delta t+o(\Delta t). \tag{8}$$

将式(7)和(8)代入(6), 化简得

$$\begin{aligned}&(1-\mathrm{e}^{-\Delta t/\tau})V^*(\boldsymbol{x}(t))=\\&\max_{\boldsymbol{u}[t,t+\Delta t]}\Big[r(\boldsymbol{x}(t),\boldsymbol{u}(t))\Delta t+\\&\mathrm{e}^{-\Delta t/\tau}\frac{\partial V^*}{\partial\boldsymbol{x}(t)}f(\boldsymbol{x}(t),\boldsymbol{u}(t))\Delta t+o(\Delta t)\Big].\end{aligned} \tag{9}$$

对式(9)两边都除以 Δt 并令 $\Delta t\to 0$, 然后根据洛比达法则求解其极限值, 则可得最优值函数的条件为

$$\begin{aligned}&\frac{1}{\tau}V^*(\boldsymbol{x}(t))=\\&\max_{\boldsymbol{u}(t)\in U}\Big[r(\boldsymbol{x}(t),\boldsymbol{u}(t))+\frac{\partial V^*(\boldsymbol{x})}{\partial\boldsymbol{x}}f(\boldsymbol{x}(t),\boldsymbol{u}(t))\Big].\end{aligned} \tag{10}$$

式(10)就是折扣型HJB方程, 其最优策略即为最大化式(10)右边的动作函数, 该函数可以表示为

$$\begin{aligned}&\boldsymbol{u}(t)=\pi^*(\boldsymbol{x}(t))=\\&\arg\max_{\boldsymbol{u}\in U}\Big[r(\boldsymbol{x}(t),\boldsymbol{u})+\frac{\partial V^*(\boldsymbol{x})}{\partial\boldsymbol{x}}f(\boldsymbol{x}(t),\boldsymbol{u})\Big].\end{aligned} \tag{11}$$

上述连续状态动作空间下的非线性动态系统最优控制问题可以用强化学习方法求解, 基本思路分两步: 首先根据当前的策略 π 估计值函数 V; 然后根据当前估计的值函数改进策略.

2 基于ERL的最优控制方法

为了学习得到连续空间下的值函数, 需采用文献[6]中的逼近器来逼近值函数. 本文定义当前值函数的估计式近似表达为

$$V^{\pi}(\boldsymbol{x}(t))\simeq V(\boldsymbol{x}(t),\theta). \tag{12}$$

其中: θ 为函数逼近器的参数, 有时也将 $V(\boldsymbol{x}(t),\theta)$ 简单记为 $V(t)$. 例如, TD学习中的值函数就是按下式进行定义的:

$$\dot{V}^{\pi}(\boldsymbol{x}(t))=\frac{1}{\tau}V^{\pi}(\boldsymbol{x}(t))-r(t). \tag{13}$$

如果当前值函数 V 估计值能满足系统要求, 则它应满足一致性连续的条件 $\dot{V}(t)=\frac{1}{\tau}V(t)-r(t)$; 反之, 如果当前值函数 V 的估计值不能满足系统要求, 则应对它进行调整, 一般可按如下TD误差公式进行调整[1]:

$$\delta(t)\equiv r(t)-\frac{1}{\tau}V(t)+\dot{V}(t). \tag{14}$$

2.1 梯度下降法的时序差分误差

为了使得式(14)的TD误差趋向于0, 可以调整值函数 $V(t)$, 或者调整值函数的微分 $\dot{V}(t)$, 或者同时调整这两项. 如果式(14)中的 $\delta(t)\to 0$, 则以下目标函

数能任意小:

$$E(t)=\frac{1}{2}\delta^2(t). \tag{15}$$

由式 (14) 的定义和复合函数导数的法则 $\dot{V}(t)=\dfrac{\partial V}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}}(t)$, 可以得式 (15) 关于参数 θ_i 的梯度为

$$\frac{\partial E(t)}{\partial \theta_i}=$$
$$\delta(t)\frac{\partial}{\partial \theta_i}\Big[r(t)-\frac{1}{\tau}V(t)+\dot{V}(t)\Big]=$$
$$\delta(t)\Big[-\frac{1}{\tau}\frac{\partial V(\boldsymbol{x},\theta)}{\partial \theta_i}+\frac{\partial}{\partial \theta_i}\Big(\frac{\partial V(\boldsymbol{x},\theta)}{\partial \boldsymbol{x}}\Big)\dot{\boldsymbol{x}}(t)\Big]. \tag{16}$$

因此, 梯度下降法可以表示为[1]

$$\dot{\theta}_i=\alpha\frac{\partial E}{\partial \theta_i}=$$
$$\alpha\delta(t)\Big[\frac{1}{\tau}\frac{\partial V(\boldsymbol{x},\theta)}{\partial \theta_i}-\frac{\partial}{\partial \theta_i}\Big(\frac{\partial V(\boldsymbol{x},\theta)}{\partial \boldsymbol{x}}\Big)\dot{\boldsymbol{x}}(t)\Big], \tag{17}$$

其中 $0<\alpha<1$ 为学习率.

2.2　向前欧拉法的时序差分误差

在本文的时序差分计算中, 值函数的微分 $\dot{V}(t)$ 并没有采用传统的 $\dot{V}(t)=\dfrac{\partial V}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}}(t)$ 进行近似替代, 而是采用向前欧拉法, 即令 $\dot{V}(t)=\dfrac{V(t-\Delta t)-V(t)}{\Delta t}$, 并代入式 (14), 得

$$\delta(t)=r(t)+\frac{1}{\Delta t}\Big[\Big(1-\frac{\Delta t}{\tau}\Big)V(t)-V(t-\Delta t)\Big]. \tag{18}$$

将式 (18) 代入 (15), 并对其关于 θ_i 微分, 得

$$\frac{\partial E(t)}{\partial \theta_i}=$$
$$\delta(t)\frac{1}{\Delta t}\Big[\Big(1-\frac{\Delta t}{\tau}\Big)\frac{\partial V(\boldsymbol{x}(t),\theta)}{\partial \theta_i}-\frac{\partial V(\boldsymbol{x}(t-\Delta t),\theta)}{\partial \theta_i}\Big]. \tag{19}$$

因此, 式 (17) 所表示的梯度下降法可以相应地改为

$$\dot{\theta}_i=$$
$$\alpha\delta(t)\Big[-\Big(1-\frac{\Delta t}{\tau}\Big)\frac{\partial V(\boldsymbol{x}(t),\theta)}{\partial \theta_i}+\frac{\partial V(\boldsymbol{x}(t-\Delta t),\theta)}{\partial \theta_i}\Big]. \tag{20}$$

如果 Δt 充分小, 则式 (20) 可简化为

$$\dot{\theta}_i=\alpha\delta(t)\frac{\partial V(\boldsymbol{x}(t-\Delta t),\theta)}{\partial \theta_i}. \tag{21}$$

用欧拉法可以将式 (18) 表示的 TD 误差离散为

$$\delta_t=r_t+\gamma V_t-V_{t-1}, \tag{22}$$

其中折扣因子为 $\gamma=1-\dfrac{\Delta t}{\tau}\simeq \mathrm{e}^{-\Delta t/\tau}$.

2.3　值函数的估计和资格迹

考虑 $t=t_0$ 时的延时回报为 $r(t)=\delta(t-t_0)$. 按式 (4) 可以定义 $t=t_0$ 时的瞬时值函数为

$$V^{\pi}(t)=\begin{cases}\mathrm{e}^{-(t_0-t)/\tau},\ t\leqslant t_0;\\ 0,\ t>t_0.\end{cases} \tag{23}$$

因为值函数是根据回报值误差不断修正, 直到逼近最优值函数为止, 所以按本文提出的给予向前欧拉法的时序差分误差 δ_{t_0} 来修正值函数, 式 (23) 可写为

$$\hat{V}(t)=\begin{cases}\delta(t_0)\mathrm{e}^{-(t_0-t)/\tau},\ t\leqslant t_0;\\ 0,\ t>t_0.\end{cases} \tag{24}$$

其中按 $\delta(t_0)$ 进行更新的参数 θ_i 的微分为

$$\dot{\theta}_i=\alpha\int_{-\infty}^{t_0}\hat{V}(t)\frac{\partial V(\boldsymbol{x}(t),\theta)}{\partial \theta_i}\mathrm{d}t=$$
$$\alpha\delta(t_0)\int_{-\infty}^{t_0}\mathrm{e}^{-(t_0-t)/\tau}\frac{\partial V(\boldsymbol{x}(t),\theta)}{\partial \theta_i}\mathrm{d}t. \tag{25}$$

资格迹 (eligibility trace) 用来表示状态或动作的有效程度[1], 若此处用 e_i 表示参数 θ_i 的指数资格迹, 则 θ_i 和 e_i 的微分可以简单写为

$$\dot{\theta}_i=\alpha\delta(t)e_i(t), \tag{26}$$

$$\kappa\dot{e}_i(t)=-e_i(t)+\frac{\partial V(\boldsymbol{x}(t),\theta)}{\partial \theta_i}, \tag{27}$$

其中 $0<\kappa\leqslant\tau$ 为资格迹的时间常数.

如果用时间步长 Δt 离散化式 (27), 则基于向前欧拉法的 TD(λ) 学习算法为

$$e_i(t+\Delta t)=\lambda\gamma e_i(t)+\frac{\partial V_t}{\partial \theta_i}, \tag{28}$$

其中 $\lambda=(\kappa-\Delta t)/(\tau-\Delta t)$.

2.4　策略的改进

策略函数 $\boldsymbol{u}(t)=\pi(\boldsymbol{x}(t))$ 的改进一般有两类方法: 第 1 类是用贪心或 softmax 函数随机改进策略

$$\boldsymbol{u}(t)=\pi(\boldsymbol{x}(t))=$$
$$\arg\max_{\boldsymbol{u}\in U}\Big[r(\boldsymbol{x}(t),\boldsymbol{u})+\frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}}f(\boldsymbol{x}(t),\boldsymbol{u})\Big]; \tag{29}$$

第 2 类是用执行器-评价器学习实现对值函数的估计和控制策略的改进, 执行器实现系统状态到控制参数的映射, 评价器则对执行器的输出结果进行评判, 同时生成时序误差信号, 该信号采用向前欧拉差分迭代公式进行离散化. 本文主要考虑采用第 2 类方法改进策略函数 $\boldsymbol{u}(t)$, 这里采用如下的执行器改进策略:

$$\boldsymbol{u}(t)=s(A(\boldsymbol{x}(t),\theta^A)+\sigma\boldsymbol{n}(t)). \tag{30}$$

其中: $A(\boldsymbol{x}(t),\theta^A)\in R^m$ 参数向量为 θ^A 的函数逼近器, 假设 $\boldsymbol{n}(t)\in R^m$ 是高斯型随机噪声, $s(\cdot)$ 是输出函数. 参数 θ^A 根据下式进行更新:

$$\dot{\theta}_i^A=\alpha^A\delta(t)\boldsymbol{n}(t)\frac{\partial A(\boldsymbol{x}(t),\theta^A)}{\partial \theta_i^A}. \tag{31}$$

当回报函数 $r(\boldsymbol{x},\boldsymbol{u})$ 为凸函数, 且动态系统 $f(\boldsymbol{x},\boldsymbol{u})$ 为线性时, 式 (29) 的最优化问题有唯一解. 为了计算方便, 假设将回报函数 $r(\boldsymbol{x},\boldsymbol{u})$ 分成两部分: 一部分是由未知环境反馈得到的强化信号函数 $R(\boldsymbol{x})$; 另一部分是当前环境下选择动作的策略函数 $S(\boldsymbol{u})$. 因此, 根据假设, 回报函数可以写为

$$r(\boldsymbol{x},\boldsymbol{u})=R(\boldsymbol{x})-\sum_{j=1}^{m}S_j(u_j), \tag{32}$$

其中 $S_j(\cdot)$ 为第 j 个动作 u_j 的费用函数. 由无约束二元函数取得极值的条件可知, 此时式 (29) 要取到极值必须满足其偏导数等于 0, 即

$$-S_j^{'}+\frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}}\frac{\partial f(\boldsymbol{x},\boldsymbol{u})}{\partial u_j}=0,\ j=1,2,\cdots,m, \quad (33)$$

其中 $\partial f(\boldsymbol{x},\boldsymbol{u})/\partial u_j$ 为动态系统 (1) 的 $n\times m$ 输入矩阵 $\partial f(\boldsymbol{x},\boldsymbol{u})/\partial \boldsymbol{u}$ 的第 j 个列向量. 若输入矩阵 $\partial f(\boldsymbol{x},\boldsymbol{u})/\partial \boldsymbol{u}$ 不依赖于动作 $\boldsymbol{u}$, 且动作的费用函数 $S_j(\cdot)$ 是凸函数, 则上述方程 (33) 有唯一解

$$u_j=S_j^{'}\Big(\frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}}\frac{\partial f(\boldsymbol{x},\boldsymbol{u})}{\partial u_j}\Big),$$

用向量形式可表示为

$$\boldsymbol{u}=S^{'-1}\Big(\frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}}^{\mathrm{T}}\frac{\partial f(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}^{\mathrm{T}}\Big). \quad (34)$$

如果动作选择被限定为 m 个 $|u_j|\leqslant u_j^{\max}, j=1,2,\cdots,m$, 则可以定义动作的费用函数为

$$S_j(u_j)=c_j\int_0^{u_j}s^{-1}\Big(\frac{u}{u_j^{\max}}\Big)\mathrm{d}u, \quad (35)$$

其中 $s(\cdot)$ 为 sigmoid 函数. 从而其输出为

$$u_j=u_j^{\max}s\Big(\frac{1}{c_j}\frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}}^{\mathrm{T}}\frac{\partial f(\boldsymbol{x},\boldsymbol{u})}{\partial u_j}^{\mathrm{T}}\Big). \quad (36)$$

当 $c_j\to 0$ 时, 控制策略将变成开关控制律, 即

$$u_j=u_j^{\max}\mathrm{sign}\Big(\frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}}^{\mathrm{T}}\frac{\partial f(\boldsymbol{x},\boldsymbol{u})}{\partial u_j}^{\mathrm{T}}\Big). \quad (37)$$

3 ERL 算法描述及其误差估计

3.1 ERL 算法描述

根据以上分析, 给出 ERL 算法步骤如下.

Step 1: 初始化状态空间和各个参数.

Step 2: 观察当前状态 x_t.

Step 3: 对于每个时间步 $\Delta(t)$, 反复执行 Step 1 ∼ Step 6.

Step 3.1: 依据当前值函数表, 以概率 $1-\epsilon$ 按式 (29) 选择并执行一个动作 u_t;

Step 3.2: 以概率 ϵ 随机选择并执行一个动作 u_t.

Step 4: 执行完动作 u_t, 观察下一个状态 x_{t+1}, 根据式 (32) 计算瞬时回报 r_t.

Step 5: 依据向前欧拉法计算时序差分误差, 按式 (26) 更新参数 θ, 式 (28) 更新资格迹.

Step 6: $\Delta(t)\leftarrow\Delta(t)+1$.

Step 7: 按式 (24) 更新值函数 $V(t)$.

3.2 误差估计

为得到 ERL 算法的误差估计式, 给出以下定义.

定义 1 若 $x=(x_1,x_2,\cdots,x_n)\in R^n$, 则 x 的范数定义为

$$\|x\|_\infty=\max_{1\leqslant i\leqslant n}|x_i|. \quad (38)$$

定理 1 若值函数 $V^\pi(\boldsymbol{x}(t))$ 和 $V^*(\boldsymbol{x}(t))$ 分别按式 (4) 和 (5) 计算, 策略函数 $\boldsymbol{u}(t)$ 按式 (30) 进行更新, 则在不考虑随机噪声的条件下, 有

$$\|V^\pi(\boldsymbol{x}(t))-V^*(\boldsymbol{x}(t))\|_\infty\leqslant\frac{2\alpha\gamma^2}{\lambda(1-\gamma)^2}\Delta_u. \quad (39)$$

其中: γ 为折扣因子, α 为学习因子, λ 为式 (28) 中的资格迹参数, $\Delta_u=\|\boldsymbol{u}(1)-\boldsymbol{u}(0)\|_\infty$.

证明 令 $\Delta_u=\|\boldsymbol{u}(1)-\boldsymbol{u}(0)\|_\infty$, 则由文献 [1] 中 Jensen's 不等式可知

$$\|V^\pi(\boldsymbol{x}(t))-V^*(\boldsymbol{x}(t))\|_\infty=$$
$$\max_{\boldsymbol{x}\in\boldsymbol{X}}|V^\pi-V^*|\leqslant$$
$$\gamma\max_{\boldsymbol{u}(t)\in U}\Big[r(\boldsymbol{x}(t),\boldsymbol{u}(t))+\frac{\partial V^*(\boldsymbol{x})}{\partial \boldsymbol{x}}f(\boldsymbol{x}(t),\boldsymbol{u}(t))\Big]\Delta_u. \quad (40)$$

又由式 (11) 和 (28) 可得

$$\max_{\boldsymbol{u}(t)\in U}\Big[r(\boldsymbol{x}(t),\boldsymbol{u}(t))+\frac{\partial V^*(\boldsymbol{x})}{\partial \boldsymbol{x}}f(\boldsymbol{x}(t),\boldsymbol{u}(t))\Big]\leqslant\frac{2\alpha\gamma}{\lambda(1-\gamma)^2}. \quad (41)$$

将式 (41) 代入 (40) 即可得证. □

4 仿真研究

为了验证本文提出的 ERL 算法的有效性, 针对具有两维连续状态空间的小车爬山问题进行仿真研究. 小车爬山学习控制在有关强化学习的文献中通常被作为一个典型的连续状态空间强化学习问题来验证算法的学习效率和泛化性能[1]. 本文基于 ERL 的控制方法和文献 [1] 提出的基于 FRL 的控制方法以及文献 [11] 提出的基于 AC-RL 算法的控制方法进行对比研究, 通过实验结果来说明本文算法的优越性. 文献 [1] 给出了小车爬山学习控制问题的示意图, 图中曲线代表一个山谷的地形, S 为山谷最低点, G 为右端最高点. 小车的任务是从谷底以尽量短的时间运动到最高点. 系统的状态由两个连续变量 x 和 v 表示, 其中 x 为小车的水平位移, v 为小车的水平速度, 状态空间满足 $\{(x,v)|-1.2\leqslant x\leqslant 0.5,-0.07\leqslant v\leqslant 0.07\}$. 当小车位于 S 点、G 点和 A 点时, x 的取值分别为 -1.2, -0.5 和 0.5. 控制量为小车所受水平方向的力 u, 取 3 个离散值, 即 $u=\{-1,0,1\}$, 分别表示减速、匀速和加速 3 个控制行为. 设在 S 点的水平位移值为 -0.5, 则系统的动力学的动力学特性由以下方程描述[1]:

$$\begin{cases}\dot{x}=v,\\ \dot{v}=0.001u-g\cos(3x).\end{cases}$$

其中: g 为与重力有关的常数, u 为控制量, 当 $x_t=-1.2$ 时, $v_t=0$. 学习控制器的目标是调节控制器使得外力能将任意初始位置和速度的小车在最短的时间移动到山顶目标位置. 虽然小车爬山问题只有二维状态空间, 但除了系统的状态观察值以外, 没有任何有关该系统的动力学模型的先验知识, 因此采用传统的基于模型的最优控制方法仍然难以求解.

仿真采用 Matlab 7.01 软件编程实现, 计算机硬件配置为 Intel Pentium Dual/2.20 GHz/2.00 GB 内存. 在仿真中, 小车的初始状态为 $x=x_0,v=0$, 当小车到达 G 点或时间步数目超过设定值时, 结束一次学习.

仿真中的主要参数值为 $\alpha_0 = 0.05, \gamma = 0.99, T_{\max} = 0.1, T_{\min} = 0.01, \max\text{steps} = 1\,000$, 采样周期为 0.02 s. 图 1 的上半图为采用 ERL 算法在 mountain car 问题中一次典型运行学习曲线, 该曲线横坐标为学习次数 (number of episodes), 纵坐标为每次学习过程中小车从 S 点到达 G 点所需的时间步数 (number of steps). 从图中可以看出, 小车在经过 20 多次学习后即获得了有效的小车爬山控制策略. 图 1 的下半图表示仿真结束时小车已经成功到达目标点.

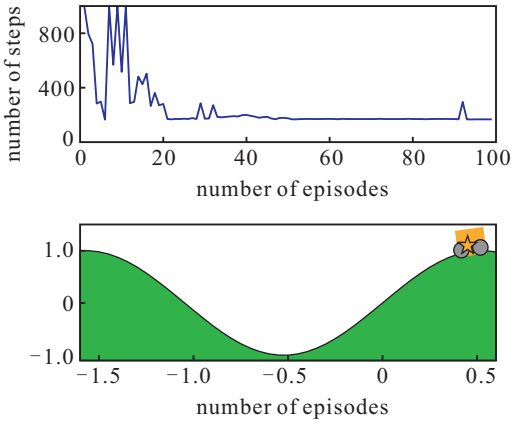

图 1　小车学习曲线

为了消除一次运行中诸多随机因数的影响, 对 FRL、ERL 和 AC-RL 算法均独立运行 20 次. 学习系统的性能由每次学习中小车从起始点 S 运动到目标点 G 的时间步数评价, 具体的系统学习性能统计结果比较如表 1 所示. 由表 1 可知, 基于 ERL 算法的学习系统平均可在 14.8 次尝试内获得最短时间的控制策略, 而基于 FRL 的学习系统则需平均 35.5 次的尝试才能获得最短时间控制, 基于 AC-RL 的学习系统则需平均 24.2 次的尝试才能获得最短的时间控制, 并且小车达到目标点所需的平均时间步数也分别多 7.4 个和 4.8 个. 因此, 与 FRL、AC-RL 算法相比, ERL 算法在小车爬山问题上具有较优的学习性能.

表 1　FRL、AC-RL 和 ERL 算法学习性能比较

算法名称	获得最优策略的尝试次数			小车达到目标点所需的时间步数		
	最小	最大	平均	最小	最大	平均
FRL 算法[1]	28	50	35.5	35	42	37.2
AC-RL 算法[11]	20	36	24.2	31	38	34.6
ERL 算法	6	23	14.8	25	31	29.8

5　结　论

本文将欧拉向前微分计算方法与强化学习算法相结合, 提出了一种新的基于 ERL 算法的非线性动态系统最优控制方法, 给出了 ERL 算法的步骤和误差估计定理. 与基于 FRL 和 AC-RL 算法的控制方法相比, 所提出的方法较好地解决了连续状态-动作空间的泛化问题, 提高了非线性系统的学习效率. 小车爬山问题的仿真实验表明, 采用 ERL 算法可以实现小车在任意时刻初始位置和速度以最短时间到达山顶目标位置, 并具有较强的鲁棒性, 在移动机器人自主越障控制领域具有一定的应用前景.

参考文献(References)

[1] Sutton R S, Barto A G. Introduction to reinforcement learning[M]. Cambridge: MIT Press, 1998: 55-68.

[2] Schaal S, Atkeson C. Learning control in robotics[J].IEEE Robotics and Automation Magazine, 2010, 17(2): 20-29.

[3] Dung L T, Komeda T, Takagi M. Reinforcement learning for pomdp using state classification[J]. Applied Artificial Intelligence, 2008, 22(7): 761-779.

[4] Lucian B, Robert B, Bart D S. A comprehension survey of multi-agent reinforcement learning[J]. IEEE Trans on Systems, Man and Cybernetics, Part C: Applications and Reviews, 2008, 68(2): 156-172.

[5] 高阳, 陈世福, 陆鑫. 强化学习研究综述[J]. 自动化学报, 2004, 30(1): 86-100.
(Gao Y, Chen S F, Lu X. Research on reingorcement learning technology: A review[J]. Acta Automatica Sinica, 2004, 30(1): 86-100.)

[6] 陈学松, 杨宜民. 基于执行器-评价器学习的自适应 PID 控制[J]. 控制理论与应用, 2011, 28(8): 1187-1193.
(Chen X S, Yang Y M. A novel adaptive PID controller based on actor-critic learning[J]. Control Theory & Applications, 2011, 28(8): 1187-1193.)

[7] 陈学松, 杨宜民. 基于递推最小二乘法的多步时序差分学习算法[J]. 计算机工程与应用, 2010, 48(8): 52-55.
(Chen X S, Yang Y M. Multi-step temporal difference learning algorithm based on recursive least-squares method[J]. Computer Engineering and Applications, 2010, 48(8): 52-55.)

[8] Barto A G, Sutton R S, Anderson C W. Neuronlike adaptive elements that can solve difficult learning control problems[J]. IEEE Trans on Systems, Man and Cybernetics, 1983, 13(5): 834-846.

[9] Zhang H, Wei Q, Liu D. An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games[J]. Automatica, 2011, 47(1): 207-214.

[10] Vamvoudakis K G, Lewis F L. Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem[J]. Automatica, 2010, 46(5): 878-888.

[11] Bhasin S, Sharma N, Patre P, et al. Asymptotic tracking by a reinforcement learning-based adaptive critic controller[J]. J of Control Theory and Application, 2011, 9(3): 400-409.