文章编号: 1001-0920 (2011) 10-1586-05

非线性离散时间系统带 ε 误差限的自适应动态规划

林小峰, 张 衡, 宋绍剑, 宋春宁

(广西大学 电气工程学院，南宁 530004)

摘 要: 为了获得非线性离散时间系统的最优控制策略, 基于自适应动态规划的原理, 提出了一种带误差限的自适应动态规划方法. 对于一个任意的状态, 用一个有限长度的控制序列近似最优控制序列, 使性能指标与最优性能指标的误差在一个较小的范围内. 选取一个非线性离散时间系统对算法的性能进行数值实验, 结果验证了该算法的有效性, 用较少的计算代价获得了近似最优的控制策略.

关键词: 最优控制；离散时间系统；自适应动态规划；神经网络；ε 误差限

中图分类号: TP183 **文献标识码:** A

Adaptive dynamic programming with ε-error bound for nonlinear discrete-time systems

LIN Xiao-feng, *ZHANG Heng*, *SONG Shao-jian*, *SONG Chun-ning*

(School of Electrical Engineering，Guangxi University，Nanning 530004，China．Correspondent：LIN Xiao-feng, E-mail：gxulinxf@163.com)

Abstract: In order to obtain the optimal control strategy of nonlinear discrete-time systems, based on the principle of adaptive dynamic programming, an adaptive dynamic programming method with error bound is proposed. For any arbitrary state, a length-limited control sequence is used to approximate the optimal control sequence, which makes the error between the performance index and optimal performance index in a smaller range. A nonlinear discrete-time system is chosen for the numerical experiments about the performance of algorithm, and the results show the effectiveness of the algorithm, and a near optimal control strategy is obtained with less computational cost.

Key words: optimal control；discrete-time systems；adaptive dynamic programming；neural network；ε-error bound

1 引 言

动态规划是解决最优化和最优控制问题的有效工具. 但由于“维数灾”, 高维问题的动态规划在计算上往往是不可行的. 自适应动态规划 (ADP) 采用非线性函数拟合方法逼近动态规划的性能指标, 有效地解决了动态规划“维数灾”的难题, 为高维复杂非线性系统的最优控制提供了一种切实可行的理论和方法[1-2].

近年来, 国际控制界和智能计算领域的许多学者对自适应动态规划进行了研究[3-5], 但多数是关于算法和仿真研究, 也有一些文献谈到了自适应动态规划理论问题, 如收敛性和最优性等, 但主要是针对线性系统和特定的非线性系统[6-7]. 文献 [8] 研究了有限范围的离散时间自适应动态规划问题, 论证了不同步长的控制策略与其代价间的关系. [9] 引入误差限的概念, 提出了 ε-最优代价和 ε-最优控制, 并对相关问题给出了详细的数学证明. [10] 在贝尔曼方程中增加了折扣因子, 进一步阐述了 ε-最优代价和 ε-自适应动态规划算法. [11] 将 ε-最优控制算法思想与迭代自适应动态规划算法原理相结合, 对任意状态求其 ε-最优控制律, 极大地提高了算法的效率.

本文针对确定性非线性离散时间系统的动态规划, 研究了有限范围的终止时间未定的最优控制问题. 对于任意的可控状态 x, 用长度为 $K_\varepsilon(x)$ ($K_\varepsilon(x) = k$) 的最优控制序列给出性能指标 $J_k^*(x)$ 来近似理论上的最优性能指标 $J_\infty^*(x)$, $J_k^*(x)$ 不会大于 $J_\infty^*(x)+$

收稿日期: 2010-06-30；**修回日期:** 2010-12-22.

基金项目: 国家自然科学基金项目(60964002)；中科院自动化研究所复杂系统与智能科学重点实验室开放基金项目(20080101).

作者简介: 林小峰(1955−), 男, 教授, 从事智能优化控制、过程控制等研究；张衡(1983−), 男, 硕士生, 从事智能优化控制的研究.

ε, ε 为规定的误差限 $\varepsilon > 0$. 系统地介绍了该算法的基本原理和方法, 选取非线性离散时间系统对算法的性能进行数值实验, 仿真算法表明了该算法的有效性, 用较少的计算代价获得了近似最优的控制策略.

2　离散时间系统的 ε-自适应动态规划

2.1　问题的提出

对于离散时间确定性系统

$$x_{k+1} = F(x_k, u_k),\ k = 0, 1, \cdots. \tag{1}$$

其中: $x_k \in R^n$ 为状态向量, $u_k \in R^m$ 为控制向量. 系统函数 F 是连续的, 原点是系统的一个平衡点, 系统在原点附近是可控的. 控制的目标是使状态 x 到达原点, 即 $x = 0$. 尝试驱使系统在有限但未知的时间步内到达该目标, 即研究有限范围的终止时间未定的最优控制问题.

系统 (1) 从状态 x 开始, 在控制序列 $\underline{u} = (u_0, u_1, \cdots, u_{N-1})$ 控制下的性能指标定义为

$$J(x, \underline{u}) = \sum_{k=0}^{N-1} U(x_k, u_k), \tag{2}$$

其中 $U(x, u)$ 为效用函数, 对于任意 (x, u), 有 $U(x, u) \geqslant 0$, $U(0, 0) = 0$. 令 $x_0 = x$, $x_k = F(x_{k-1}, u_{k-1}), k = 1, 2, \cdots, N$. 在 $\underline{u}$ 控制下, 状态轨迹的最后一个状态用 $x^{(f)}(x, \underline{u})$ 表示. 若存在一个控制序列 $\underline{u}$ 使得 $x^{(f)}(x, \underline{u}) = 0$, 则称初始状态 $x_0 = x$ 是可控的, 同时控制序列 $\underline{u}$ 称为 x 的容许控制序列.

对于任意给定的初始状态 x, 最优控制的目标是找到一个容许控制序列 $\underline{u}$ 以最小化性能指标 $J(x, \underline{u})$. 控制序列 $\underline{u} = (u_0, u_1, \cdots, u_{N-1})$ 的长度用 $|\underline{u}|$ 来表示, 但在最优控制序列确定之前, 不知道其长度. 这类最优化问题称为终止时间未定的有限范围问题, 在研究这类问题之前, 先考虑固定终止时间为 k 的有限范围问题, 其中 $k = 1, 2, \cdots$.

令 $J_k^*(x)$ 表示状态 x 的所有长度为 k 的容许控制序列最优代价, 由贝尔曼最优性原理有

$$J_k^*(x) = \min_u \{U(x, u) + J_{k-1}^*(F(x, u))\}, \tag{3}$$

最优控制向量为

$$\nu_k^*(x) = \arg\min_u \{U(x, u) + J_{k-1}^*(F(x, u))\}. \tag{4}$$

动态规划的第 1 步是确定函数 $\nu_1^*(\cdot)$. 对于任意给定的状态向量 x, 控制向量 $\nu_1^*(x)$ 是下述问题的最优解:

$$\min_u U(x, u),\ \text{s.t. } F(x, u) = 0. \tag{5}$$

相应的性能指标为

$$J_1^*(x) = U(x, \nu_1^*(x)). \tag{6}$$

确定函数 $\nu_1^*(\cdot)$ 和 $J_1^*(\cdot)$ 后, 可应用式 (3) 和 (4) 递推求得 $\nu_j^*(\cdot)$ 和 $J_j^*(\cdot)$, $j = 2, 3, \cdots, k$, 从而得到最优控制序列

$$\underline{\nu}_k^*(x_0) = (\nu_k^*(x_0), \nu_{k-1}^*(x_1), \cdots, \nu_1^*(x_{k-1})).$$

这时便产生了“维数灾”问题, 因为必须计算并记录所有的 $J_j^*(\cdot)$ 和 $\nu_j^*(\cdot), j = 1, 2, \cdots, k$. 在大多数实际应用中, k 值往往较大, 这样对计算和存储的要求是巨大的.

为了避免“维数灾”, 可以应用无限范围问题的方法. 无限范围问题研究了具有无限长度的控制序列, 并重新定义容许控制序列为使状态在限定的性能指标内渐近到达目标的控制序列. 无限范围问题仅有一个最优性能指标, 它不依赖于控制序列的长度, 因为所有的控制序列均有一个相同的长度 ∞. 但性能指标是无限项的总和, 它是一个无穷级数的极限, 因此必须考虑级数 $J(x, \underline{u}) = \sum_{k=0}^{\infty} U(x_k, u_k)$ 的收敛性. 如果该极限存在, 则其能否达到最大下界 $\inf J(x, \underline{u})$, 且 $\inf J(x, \underline{u})$ 是否满足 HJB 方程.

对于很多系统, 特别是非线性系统, 很难确定是否存在一个容许控制序列 $\widehat{\underline{u}}$, 使得性能指标 $J(x_0, \widehat{\underline{u}})$ 等于最大下界 $\inf_{\underline{u}} J(x_0, \underline{u})$. 另一方面, 若 $\underline{u}^*$ 是使用 $\inf_{\underline{u}} J(x_0, \underline{u})$ 作为李雅普洛夫函数而确定的控制序列, 则很难确定 $\underline{u}^*$ 就是一个容许控制序列. 因此, 当试图找到一个最优代价和一个最优控制器来避免“维数灾”时, 将面对这样的问题, 即能否用一个容许控制序列来实现这个最优代价, 以及该最优控制器是否为容许的控制器. 但 $\inf_{\underline{u}} J(x_0, \underline{u})$ 不是一个合适的值, 虽然它是所有性能指标的最大下界, 但如果试图通过近似 $\inf_{\underline{u}} J(x_0, \underline{u})$ 来控制一个系统时, 则可能会得到一个非容许的控制器.

如果令 $J_\infty^*(x) = \inf_{\underline{u}} \{J(x, \underline{u}), \underline{u} \in \vartheta_x\}$, 其中 ϑ_x 表示 x 的所有容许控制序列的集合, 则可以证明, 对于每个可控的状态 x, $J_\infty^*(x)$ 均是有意义的, 显然 $J_\infty^*(x)$ 是理论上的最优代价. 进一步可以证明, 不同控制步数的最优性能指标 J_k^* 形成了一个非增的序列, 极限 $\lim\limits_{k\to\infty} J_k^*$ 等于所有性能指标的最大下界 J_∞^*, 且 J_∞^* 满足贝尔曼原理, 即

$$J_\infty^*(x) = \inf_u \{U(x, u) + J_\infty^*(F(x, u))\}. \tag{7}$$

因此, 可以将 $J_\infty^*(x)$ 作为最优性能指标. 那么, 最优控制的一个自然的策略是, 找到一个控制序列, 使得相应的性能指标恰好是 $J_\infty^*(x)$. 但在许多情况下, 不能找到使等式 $J_k^*(x) = J_\infty^*(x)$ 成立的 k 值, 即对于任意有限长度的控制序列 $\underline{u}$, 从状态 x 开始的, 在 $\underline{u}$ 控制下的性能指标要大于 $J_\infty^*(x)$, 而不是等于 $J_\infty^*(x)$. 另一方面, 通过求解式 (7), 能得到状态 x 的一个控制向量 $\nu_\infty^*(x)$, 进而得到一个控制序列

$$\underline{\nu}_\infty^*(x) = (\nu_\infty^*(x_0), \nu_\infty^*(x_1), \cdots, \nu_\infty^*(x_k), \cdots),$$

其中

$$x_0 = x,\ x_1 = F(x_0, \nu^*_\infty(x_0)),\ \cdots,$$
$$x_k = F(x_{k-1}, \nu^*_\infty(x_{k-1})), \cdots.$$

一般而言, $\underline{\nu}^*_\infty(x)$ 具有无限长度, 即通过求解式 (7) 得到的控制器不能在有限时间步数内控制状态到达目标.

在某些情况中, $\underline{\nu}^*_\infty(x)$ 不是容许控制序列, 即使是容许控制序列也可能被滥用, 因为必须从 J^*_∞ 的一个近似 $\widehat{J}$ 来获得 $\underline{\nu}^*_\infty(x)$, 而不是从其本身. 如果由计算误差引起 $\widehat{J} < J^*_\infty(x)$, 则从 $\widehat{J}$ 获得的控制将不再是容许的, 由于 J^*_∞ 是关于所有容许控制序列的所有性能指标的最大下界, $\widehat{J}$ 不再是性能指标函数.

2.2 ε-最优性能指标和 ε-最优控制

解决上述问题的方法是固定控制序列的长度. 预先设置一个正整数 K, 用 $J^*_K(x)$ 作为 $J^*_\infty(x)$ 的一个近似. 当 K 足够大时, $J^*_K(x)$ 是对 $J^*_\infty(x)$ 的一个较好的近似, 但 $\lim\limits_{k\to\infty} J^*_\infty(x) = J^*_\infty(x)$ 有可能不是一致收敛的. 对于一个固定的 K, 当 x 较大时, $J^*_K(x) - J^*_\infty(x)$ 的误差可能很大. 因此, 可以将控制序列的长度也考虑进去, 对于不同的 x, 将采用不同的 K 作为最优控制序列的长度. 对于一个给定的误差限 $\varepsilon > 0$, 选择 K 使 $J^*_\infty(x)$ 和 $J^*_K(x)$ 之间的误差不会大于 ε.

对于任意可控的状态向量 x, 定义

$$K_\varepsilon(x) = \min\{k : |J^*_k(x) - J^*_\infty(x)| \leqslant \varepsilon\}, \tag{8}$$

其中 ε 是一个大于零的正数. $K_\varepsilon(x)$ 的意义在于, 对于任意的可控状态 x, 长度为 $K_\varepsilon(x)(K_\varepsilon(x) = k)$ 的最优控制序列将给出一个性能指标 $J^*_k(x)$, 它将充分地接近"理论的最优代价" $J^*_\infty(x)$. $J^*_k(x)$ 比 $J^*_\infty(x)$ 稍微大一点, 但不会大于 $J^*_\infty(x)+\varepsilon$, 这样即可考虑用 $J^*_k(x)$ 来近似 $J^*_\infty(x)$.

再定义一个状态集合 $T^{(\varepsilon)}_k$, 当 $x \in T^{(\varepsilon)}_k$ 时, 要找到性能指标不大于 $J^*_\infty(x) + \varepsilon$ 的最优控制序列, 只需考虑长度 $|\underline{u}| \leqslant k$ 的控制序列 $\underline{u}$ 即可. 其实 $T^{(\varepsilon)}_k$ 是一个区域, 在该区域内, J^*_k 与 J^*_∞ 之间的误差不超过 ε. 当 k 较大时, 集合 $T^{(\varepsilon)}_k$ 也较大. 集合 $T^{(\varepsilon)}_k$ 的大小也依赖于 ε 的值, ε 的值越小, $T^{(\varepsilon)}_k$ 集合也越小. 对于每个可控状态 x, 总能找到一个合适的 k 值, 并用长度为 k 的控制序列来近似最优控制.

如果 $x \in T^{(\varepsilon)}_k$ 且 $u^* = \nu^*_k(x)$, 则 $F(x, u^*) \in T^{(\varepsilon)}_{k-1}$. 因此, 再定义一个控制向量的集合为

$$\prod_x^{\varepsilon,k} = \{u : F(x,u) \in T^{(\varepsilon)}_{k-1}\},\ x \in T^{(\varepsilon)}_k. \tag{9}$$

这样对于任意可控状态 x, 假如 $K_\varepsilon(x) = k$, 则 $J^*_k(x)$ 是 $J^*_\infty(x)$ 的误差不大于 ε 的一个近似. 于是定义 ε-最优性能指标为

$$V^*_\varepsilon(x) = \begin{cases} J^*_k(x),\ x \neq 0, K_\varepsilon(x) = k, k = 1, 2, \cdots; \\ 0,\ x = 0. \end{cases} \tag{10}$$

根据 $V^*_\varepsilon(x)$, 定义 ε-最优控制 $\mu^*_\varepsilon(\cdot)$ 为

$$\mu^*_\varepsilon(x) = \begin{cases} \nu^*_k(x),\ x \neq 0, K_\varepsilon(x) = k, k = 1, 2, \cdots; \\ 0,\ x = 0. \end{cases} \tag{11}$$

这样, 对于任意可控状态 x, 如果 $k = K_\varepsilon(x) > 0$, 则有

$$V^*_\varepsilon(x) = \min_{u \in \prod_x^{\varepsilon,k}} \{U(x,u) + V^*_\varepsilon(F(x,u))\}, \tag{12}$$

且

$$\mu^*_\varepsilon(x) = \arg\min_{u \in \prod_x^{\varepsilon,k}} \{U(x,u) + V^*_\varepsilon(F(x,u))\}. \tag{13}$$

特别地, 如果 $K_\varepsilon(x) = 1$, 则有

$$\prod_x^{\varepsilon,1} = \{u : F(x,u) = 0\}, \tag{14}$$

$$V^*_\varepsilon(x) = \min_{u \in \prod_x^{\varepsilon,k}} \{U(x,u)\} = J^*_1(x), \tag{15}$$

$$\mu^*_\varepsilon(x) = \arg\min_{u \in \prod_x^{\varepsilon,k}} \{U(x,u)\} = \nu^*_1(x). \tag{16}$$

式 (12) 与 (7) 较为相似, 但式 (7) 中, 右侧是下确界; 式 (12) 中, 右侧是最小值. 下确界不是总能够实现的, 一方面, 对于任意容许控制序列 $\underline{u}$, $J(x,\underline{u}) = J^*_\infty(x)$ 可能不成立; 另一方面, 如果 $\widetilde{\underline{u}}$ 是满足 $J(x,\widetilde{\underline{u}}) = J^*_\infty(x)$ 的一个控制序列, 则 $\widetilde{\underline{u}}$ 可能不是容许的. 在 $\widetilde{\underline{u}}$ 控制下的轨迹也许不能收敛到目标, 然而, 最小值意味着它是能实现的.

2.3 离散时间系统的 ε-自适应动态规划算法

ε-自适应动态规划算法用 3 个网络来分别近似 ε-最优性能指标 $V^*_\varepsilon(x)$, ε-最优控制器 $\mu^*_\varepsilon(x)$ 和 K_ε, 3 个网络对应的输出分别为 $\widehat{J}$, $\widehat{\mu}$ 和 $\widehat{K}$. 算法开始时考虑问题

$$J^*_1(x) = \min_u U(x,u),\ \text{s.t. } F(x,u) = 0, \tag{17}$$

使得式 (17) 有解的 x 的集合即为 T_1. 当 $X \in T_1$ 时, 考察方程 $F(x,u) = 0$. 若该方程只有一个解, 则该解就是 $\nu^*_1(x)$, 因为它是唯一的选择. 若有不止一个解, 则选择能最小化代价的那个解. 根据下式初始化网络:

$$\widehat{J}(x) = \begin{cases} J^*_1(x),\ x \in T_1; \\ 0,\ \text{otherwise}. \end{cases} \tag{18}$$

$$\widehat{K}(x) = \begin{cases} 1,\ x \in T_1, x \neq 0; \\ 0,\ \text{otherwise}. \end{cases} \tag{19}$$

$$\widehat{\mu}(x) = \begin{cases} \nu^*_1(x),\ x \in T_1; \\ 0,\ \text{otherwise}. \end{cases} \tag{20}$$

网络 $\widehat{J}$ 和 $\widehat{\mu}$ 将依据测量的数据按照式 (12) 和 (13) 进行更新, $\widehat{K}$ 更新为

$$\widehat{K}(x)=\widehat{K}(F(x,\widehat{\mu}(x)))+1. \tag{21}$$

对于每个状态 $x\in R^n$, 整数 $K_\varepsilon(x)=k$ 指出, 最优性能指标将用 k 步来实现 (误差限为 ε). 若 $K_\varepsilon(x)$ 是无限的, 则 x 是不可控的. 即若 $K_\varepsilon(x)=k$, 则 $J_k^*(x)$ 是最优性能指标 $J_\infty^*(x)$ 的一个近似. 而若 $K_\varepsilon(x)=+\infty$, 则最优代价在有限的时间步内是不能实现的. 作为一个数值算法, 仅考虑有界区域和有限的控制步数. 用 $\max K$ 来代表控制步数的上界, 系统从任何初始状态到达目标的总的控制步数限制在 $k\leqslant\max K$ 中, 即对于一个初始状态 x, 如果控制系统不能从 x 开始在 $\max K$ 步内到达目标, 则认为 x 是不可控的.

2.4 ε-自适应动态规划算法步骤

Step 1: 初始化. 求解问题 (17) 以确定集合 T_1 和函数 $J_1^*(x)$ 和 $\nu_1^*(x)$, 根据式 (18)∼(21) 训练 $\widehat{K}(x)$, $\widehat{J}(x)$ 和 $\widehat{\mu}(x)$.

Step 2: 从整体状态空间中随机选择一列初始状态 $x_0=(x_0^{(1)},x_0^{(2)},\cdots,x_0^{(p)})$.

Step 3: 对 $k=1,2,\cdots,\max K$ 执行Step 3∼Step 6.

Step 4: 从初始状态 x_0 开始, 在 $x_1=(x_1^{(1)},x_1^{(2)},\cdots,x_1^{(p)})$ 的控制下运行系统, 记录由此得到的状态, 并计算对应的代价 $C_1^{(i)}=U(x_0^{(i)},\widehat{\mu}(x_0^{(i)}))+\widehat{J}(x_1^{(i)})$. 记录每一个 $k_0^{(i)}=\widehat{K}(x_0^{(i)})$ 和 $k_1^{(i)}=\widehat{K}(x_1^{(i)})$.

Step 5: 更新 $\widehat{J}$ 和 $\widehat{K}$. 对每一个 $i=1,2,\cdots,p$, 如果 $x_0^{(i)}\in T_1$ 且 $J_1^*(x_0^{(i)})\leqslant C_1^{(i)}+\varepsilon$, 则有

$$\widehat{J}(x_0^{(i)})=J_1^*(x_0^{(i)}).$$

$$\widehat{K}(x_0^{(i)})=\begin{cases}0,\ x_0^i=0;\\ 1,\ x_0^i\neq 0.\end{cases} \tag{22}$$

若 $k_1^{(i)}\leqslant\max(k_0^{(i)}-1,1)$, 则有

$$\widehat{J}(x_0^{(i)})=C_1^{(i)},\ \widehat{K}(k_0^{(i)})=k_1^{(i)}+1. \tag{23}$$

Step 6: 更新 $\widehat{\mu}$. 如果 $x\in T_1$ 且 $J_1^*(x)\leqslant\widehat{J}(x)+\varepsilon$, 则 $\widehat{\mu}(x)=\nu_1^*(x)$, 且有

$$\widehat{\mu}(x)=\arg\min_u\{U(x,u)+\widehat{J}(F(x,u)),$$
$$\widehat{K}(F(x,u))\leqslant\max(\widehat{K}(x)-1,1)\}. \tag{24}$$

Step 7: 令 $x_0=x_1$.

Step 8: 返回 Step 1, 直到该过程收敛为止.

3 实验分析

为了评估 ε-自适应动态规划算法的性能, 选择一个非二次型效用函数的非线性系统来进行数值实验. 程序用 Matlab 编写, 在 Lenovo 个人电脑上运行, 配置为 Windows XP 操作系统和 Pentium IV 处理器. 考虑系统

$$x_{k+1}=F(x_k,u_k)=x_k+\sin(2u_k),$$

其中 $x_k,u_k\in R$, 且 $-5\leqslant x_k\leqslant 5$, $k=0,1,\cdots$. 效用函数为 $U(x,u)=|x|+u^2$, 由于 $F(0,0)=0$, $x=0$ 是系统的一个平衡状态, 且有 $F(x,u)=x+\sin(2u)$, 根据 $F(x,u)=0$, 隐函数为

$$u=f^{(i)}(x)=0.5\sin^{-1}(-x)+i\pi,$$
$$u=g^{(i)}(x)=-0.5\sin^{-1}(-x)+(i+0.5)\pi.$$

其中: $-1\leqslant x\leqslant 1; i=0,\pm1,\pm2,\cdots$. 容易发现 $T_1=[-1,1]$, 对于 $x\in T_1$, 有

$$J_1^*(x)=\min\{U(x,u):F(x,u)=0\}=$$
$$|x|+(0.5\sin^{-1}(-x))^2,$$
$$\nu_1^*(x)=0.5\sin^{-1}(-x). \tag{25}$$

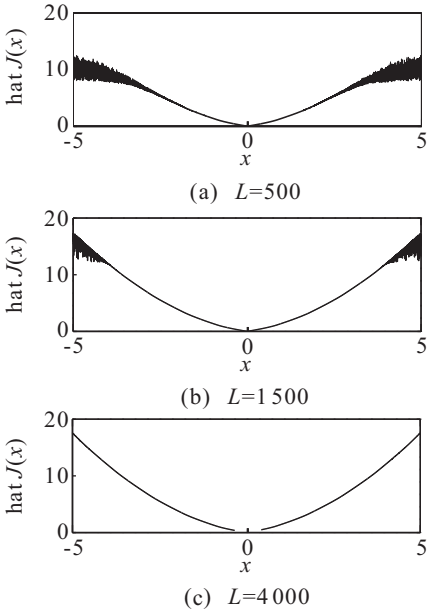
因为 $\sin(2u)$ 的值在 -1 和 1 之间, 可以发现, 对于任何 $k=1,2,\cdots$, 有 $T_k=[-k,k]$.

在离散时间系统的 ε-自适应动态规划算法的 Step 6, 需要找到 $\arg\min_u[U(x,u)+\widehat{J}(F(x,u))]$. 因为 $U(x,u)=|x|+u^2$, $F(x,u)=x+\sin(2u)$, 所以最小化问题变为

$$\min_u\{|x|+u^2+\widehat{J}(\omega)\}, \tag{26}$$

其中 $\omega=x+\sin(2u)$. 因为 $\sin(\cdot)$ 是周期为 2π 的周期函数, 所以式 (26) 的最小值仅在 $u\in(-\pi/2,\pi/2)$ 时取得. 选择误差限的值为 $\varepsilon=0.01$. 状态的区间是 $|x|\leqslant 5$, 控制步数限制在 10 步. 初始状态集 x_0 为每步 50, 即在每次内部循环迭代开始时, 随机选择 50 个初始状态. 执行 ε-自适应动态规划算法, 每次内部循环进行 10 次迭代控制, 意味着如果某个状态在 10 个控制步内没有被控制到平衡点, 则该状态是不可控的. 外部循环迭代的次数用 L 表示, 运行该算法直到 $L=4\,000$. 数值实验的结果如图 1∼图 4 所示.

图 1 和图 2 分别为 $\widehat{J}(x)$ 和 $\widehat{\mu}(x)$, 可以发现, 函数 $\widehat{J}(x)$ 和 $\widehat{\mu}(x)$ 从原点附近开始收敛, 随着外部循环次

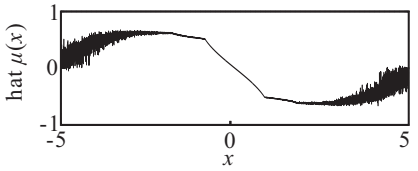


(a) L=500

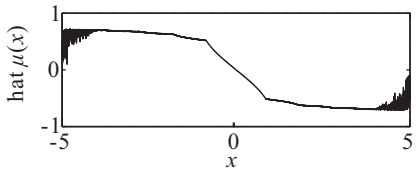
(b) L=1 500

(c) L=4 000

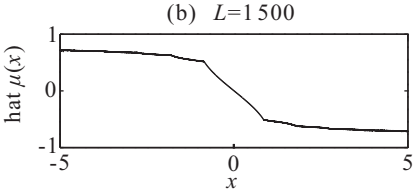
图 1　ε-最优代价



(a) L=500

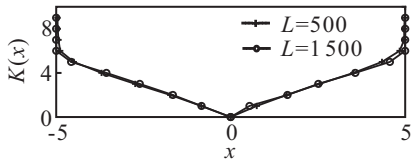


(b) L=1 500

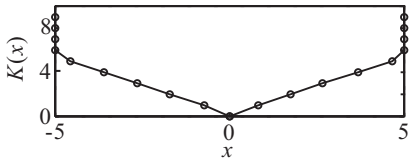


(c) L=4 000

图 2 ε-最优控制

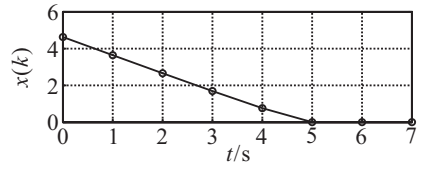


(a) L=500,L=1 500

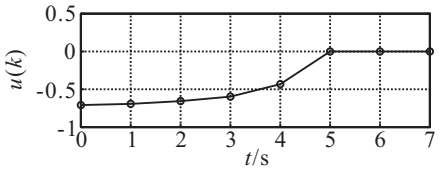


(b) L=4 000

图 3 ε-最优控制步数



(a) trajectory of state



(b) control sequence

图 4 状态轨迹和控制序列

数的增加, 收敛区域逐渐向整个状态空间扩展, 并最终在整个状态空间内均能以一个不大于ε的误差近似最优代价函数$J^*(x)$及相应的最优控制$u^*(x)$, 即是本文定义的ε-最优性能指标$V_\varepsilon^*(x)$和ε-最优控制$\mu_\varepsilon^*(x)$. 虽然控制函数$\widehat{\mu}(x)$始终可以为任意状态x提供一个控制序列, 但在$\widehat{\mu}(x)$于整个状态空间内完全收敛之前, 它所提供的控制序列能否使代价函数满足误差要求和控制步数要求则是不能确定的, 显然, $V_\varepsilon^*(x)$和$\mu_\varepsilon^*(x)$都是分段函数. 图3所示为$\widehat{K}$, 当$|x|$很小时, $K_\varepsilon(x)$也很小, 即只需较少的控制步数即可驱使x到达目标, 而当$|x|$较大时, 需要更多的控制步数才能使状态到达目标. 图4是任意选取的一个状态在已经训练好的控制器控制下的状态轨迹和控制序列. 由图4可知, 仅用5个控制步即可将该任意选取的状态控制到目标.

4 结 论

本文研究非线性离散时间系统的动态规划, 介绍了一种带有ε误差限的自适应动态规划算法, 即离散时间系统的ε-自适应动态规划算法. 该方法定义了一个ε-最优性能指标函数$V_c^*(\cdot)$, 由它确定的控制器称为ε-最优控制器$\mu_c^*(\cdot)$. 该控制器总是能够控制状态以使其接近平衡状态, 同时, 性能指标与理论上最优的性能指标误差不大于ε. 利用非线性离散时间系统对算法的性能进行数值实验, 结果验证了该算法的可行性和有效性.

参考文献(References)

[1] Werbos P J. A menu of designs for reinforcement learning over time, in neural networks for control[M]. Cambridge: MIT Press, 1990: 67-95.

[2] Werbos P J. Approximate dynamic programming for real-time control and neural modeling, in handbook of intelligent control: Neural, fuzzy and adaptive approaches[M]. New York: Van Nostrand Reinhold, 1992: 493-525.

[3] Danil V P, Donald C W. Adaptive critic designs[J]. IEEE Trans on Neural Networks, 1997, 8(5): 997-1007.

[4] George G L, Christian P. Training strategies for critic and action neural networks in dual heuristic programming method[C]. Proc of the 1997 IEEE Int Conf on Neural Networks. Houston, 1997: 712-717.

[5] Si J, Wang Y T. On-line learning control by association and reinforcement[J]. IEEE Trans on Neural Networks, 2001, 12(2): 264-276.

[6] Zhang H G, Wei Q L, Luo Y H. A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm[J]. IEEE Trans on Systems, Man and Cybernetics, 2008, 38(4): 937-942.

[7] Asma A T, Frank L L, Murad A K. Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof[J]. IEEE Trans on Systems, Man and Cybernetics, 2008, 38(4): 943-949.

(下转第1595页)