基于迭代神经动态规划的数据驱动非线性近似最优调节

王 鼎 [1, 2] 穆朝絮 [2] 刘德荣 [3]

摘 要 利用数据驱动控制思想, 建立一种设计离散时间非线性系统近似最优调节器的迭代神经动态规划方法. 提出针对离散时间一般非线性系统的迭代自适应动态规划算法并且证明其收敛性与最优性. 通过构建三种神经网络, 给出全局二次启发式动态规划技术及其详细的实现过程, 其中执行网络是在神经动态规划的框架下进行训练. 这种新颖的结构可以近似代价函数及其导函数, 同时在不依赖系统动态的情况下自适应地学习近似最优控制律. 值得注意的是, 这在降低对于控制矩阵或者其神经网络表示的要求方面, 明显地改进了迭代自适应动态规划算法的现有结果, 能够促进复杂非线性系统基于数据的优化与控制设计的发展. 通过两个仿真实验, 验证本文提出的数据驱动最优调节方法的有效性.

关键词 自适应动态规划, 数据驱动控制, 迭代神经动态规划, 神经网络, 非线性近似最优调节

引用格式 王鼎, 穆朝絮, 刘德荣. 基于迭代神经动态规划的数据驱动非线性近似最优调节. 自动化学报, 2017, **43**(3): 366−375

DOI 10.16383/j.aas.2017.c160272

Data-driven Nonlinear Near-optimal Regulation Based on Iterative Neural Dynamic Programming

WANG Ding[1, 2] MU Chao-Xu[2] LIU De-Rong[3]

Abstract An iterative neural dynamic programming approach is established to design the near optimal regulator of discrete-time nonlinear systems using the data-driven control formulation. An iterative adaptive dynamic programming algorithm for discrete-time general nonlinear systems is developed and proved to guarantee the property of convergence and optimality. Then, a globalized dual heuristic programming technique is developed with detailed implementation by constructing three neural networks, where the action network is trained under the framework of neural dynamic programming. This novel architecture can approximate the cost function with its derivative, and simultaneously, adaptively learn the near-optimal control law without depending on the system dynamics. It is significant to observe that it greatly improves the existing results of iterative adaptive dynamic programming algorithm, in terms of reducing the requirement of control matrix or its neural network expression, which promotes the development of data-based optimization and control design for complex nonlinear systems. Two simulation experiments are described to illustrate the effectiveness of the data-driven optimal regulation method.

Key words Adaptive dynamic programming, data-driven control, iterative neural dynamic programming, neural networks, nonlinear near-optimal regulation

Citation Wang Ding, Mu Chao-Xu, Liu De-Rong. Data-driven nonlinear near-optimal regulation based on iterative neural dynamic programming. *Acta Automatica Sinica*, 2017, **43**(3): 366−375

最优控制研究如何设计控制器使得系统的性能指标达到最优. 它广泛存在于工程技术和社会生活中, 是现代控制理论的重要内容之一. 与线性系统的最优控制问题需要求解 Riccati 方程不同, 研究非线性系统的最优控制通常需要求解非线性 Hamilton-Jacobi-Bellman (HJB) 方程. 例如, 对于离散时间

收稿日期 2016-03-16 录用日期 2016-05-17
Manuscript received March 16, 2016; accepted May 17, 2016
国家自然科学基金 (61233001, 61273140, 61304018, 61304086, 61533017, U1501251, 61411130160), 北京市自然科学基金 (4162065), 天津市自然科学基金 (14JCQNJC05400), 中国科学院自动化研究所复杂系统管理与控制国家重点实验室优秀人才基金, 天津市过程检测与控制重点实验室开放课题基金 (TKLPMC-201612) 资助
Supported by National Natural Science Foundation of China (61233001, 61273140, 61304018, 61304086, 61533017, U1501251, 61411130160), Beijing Natural Science Foundation (4162065), Tianjin Natural Science Foundation (14JCQNJC05400), the Early Career Development Award of the State Key Laboratory of Management and Control for Complex Systems (SKL-MCCS) of the Institute of Automation, Chinese Academy of Sciences (CASIA), and Research Fund of Tianjin Key Laboratory of Process Measurement and Control (TKLPMC-201612)
本文责任编委 侯忠生
Recommended by Associate Editor HOU Zhong-Sheng
1. 中国科学院自动化研究所复杂系统管理与控制国家重点实验室 北京 100190 2. 天津市过程检测与控制重点实验室, 天津大学电气自动化与信息工程学院 天津 300072 3. 北京科技大学自动化学院 北京 100083
1. The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190 2. Tianjin Key Laboratory of Process Measurement and Control, School of Electrical and Information Engineering, Tianjin University, Tianjin 300072 3. School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083

非线性系统而言, 这一过程就包含求解非线性偏微分方程, 这在很多情况下是难以实现的. 虽然动态规划是求解最优控制问题的经典方法, 但是其后向求解的特点往往导致 "维数灾" 现象的发生[1], 同时这种后向求解模式也不利于该方法的实际应用. 于是, 基于人工神经网络良好的自适应、自学习等特性, 自适应 (或者近似) 动态规划 (Adaptive/approximate dynamic programming, ADP) 方法应运而生[2]. 文献 [3−5] 针对 ADP 方法的基本原理、实现结构和目前的发展状况, 给出了阶段性总结与研究展望, 并且指出 ADP 实际上是一种有效的数据驱动方法[5−6]. 根据文献 [2] 和文献 [7], 可以将 ADP 方法划分为三种主要结构: 1) 启发式动态规划 (Heuristic dynamic programming, HDP); 2) 二次启发式动态规划 (Dual heuristic dynamic programming, DHP); 3) 全局二次启发式动态规划 (Globalized DHP, GDHP). 在与上述内容相关的三种执行依赖结构 (Action-dependent) 中, 执行依赖 HDP 类似于机器学习领域的 Q- 学习 (Q-learning)[8]. 另外, Si 和 Wang[9] 提出的神经动态规划也是一种类似于执行依赖 HDP 的在线学习控制方法, 具有容易实现、在线优化、不依赖被控对象模型等特点, 对于 ADP 结构的发展产生了很大的影响. 但是, 值得注意的是, 上述神经动态规划方法的重点在于强调控制系统的在线学习与优化设计, 没有从理论上证明控制算法的收敛性, 因此可以看到, 实验结果的成功具有一定的概率.

近年来, 正在兴起的许多社会和工程新技术的重要特点是拥有实时海量的大数据信息[10]. 在大数据技术快速发展的背景下, 随着对数据驱动思想和类脑学习理念的深入研究, ADP 已经发展成为进行智能控制与优化设计的有效途径, 因此受到了许多学者的重视. 针对离散时间系统[11−20] 和连续时间系统[21−26], 这种基于数据的自学习控制都取得了丰硕的研究成果. Al-Tamimi 等[11] 针对离散时间仿射非线性系统 $x_{k+1} = f(x_k) + g(x_k)u_k$, 首次提出基于贪婪迭代的 HDP 算法研究无限时间最优控制设计, 创造性地将求解代数方程的迭代思想引入 ADP 方法的框架之中. 这促进了迭代 ADP 算法的快速发展, 由此涌现出大量的研究成果[12−19]. 在基本的迭代 ADP 算法中, 一般需要构建两个神经网络, 即评判网络和执行网络, 分别用以近似代价函数和控制函数. 然后利用特定的最优化算法, 通过在迭代过程中不断更新神经网络的权值矩阵, 从而自适应地学习最优权值. 值得一提的是, Wang 等[14] 针对有限时间域上的非线性最优控制问题, 提出迭代 ε-ADP 算法, 得到和文献 [11] 不同的收敛性结论, 从全新的角度诠释迭代 ADP 算法的精髓. 但是, 也应该注意到, 在现有的迭代 ADP 算法中, 针对执行网络的训练大多数依赖于控制矩阵 $g(x_k)$ 的直接信息或者其神经网络表示, 也就是在一定程度上依赖于系统动态. 于是, Zhong 等[19] 提出一种新的目标导向型 (Goal representation) ADP 结构求解非线性系统的在线优化控制, 以发展神经动态规划的结论, 放松对系统动态的要求, 但是基于 HDP 的实现结构导致评判网络不能直接输出代价函数的导函数信息, 而且 HDP 结构的控制效果也有待改进. 实际上, 已有的研究表明, 在 ADP 方法的实现结构中, DHP 和 GDHP 会在一定程度上得到比 HDP 更好的控制效果[12, 16]. 总的来说, 虽然基于 ADP 的非线性系统最优控制研究已经取得了很大的进展, 但是仍然缺少基于 GDHP 实现结构的迭代意义下神经动态规划的报道, 因此对于现有执行网络的更新方法也鲜有改进. 基于此, 本文提出一种基于迭代神经动态规划的离散时间非线性系统数据驱动近似最优控制方法, 旨在改进执行网络的训练方法, 进一步降低迭代 ADP 算法对于控制系统动态模型的依赖, 促进基于数据的复杂非线性系统优化控制的发展.

1　问题描述

考虑离散时间非线性系统

$$x_{k+1} = F(x_k, u_k), \quad k = 0, 1, 2, \cdots \tag{1}$$

其中, k 是描述系统运行轨迹的时间步骤, $x_k = [x_{1k}, x_{2k}, \cdots, x_{nk}]^{\mathrm{T}} \in \Omega_x \subset \mathbf{R}^n$ 为系统的状态向量, $u_k = [u_{1k}, u_{2k}, \cdots, u_{mk}]^{\mathrm{T}} \in \Omega_u \subset \mathbf{R}^m$ 为系统的控制向量. 我们设定时间步骤 $k = 0$ 时的状态 $x_0 = [x_{10}, x_{20}, \cdots, x_{n0}]^{\mathrm{T}}$ 为被控系统的初始状态向量. 这里, 式 (1) 描述的是一般意义下的离散时间非线性系统. 容易知道, 具有仿射形式的非线性系统, 即 $x_{k+1} = f(x_k) + g(x_k)u_k$, 其中, $g(x_k)$ 为控制矩阵, 是系统 (1) 的一种特殊情况. 这里给出下面两个基本假设[11−12, 16].

假设 1. 动态函数 $F(\cdot, \cdot)$ 在属于 $\mathbf{R}^n$ 并且包含原点的集合 Ω_x 上 Lipschitz 连续且有 $F(0, 0) = 0$, 因此, $x = 0$ 是系统 (1) 在控制 $u = 0$ 时的一个平衡状态.

假设 2. 动态系统 (1) 可控, 即在集合 Ω_u 中存在一个能够渐近镇定被控系统的连续控制律, 使得在其作用下产生的控制输入序列能够将系统从初始状态转移到平衡状态.

本文研究无限时间域上的最优调节器设计问题. 这里, 最优调节的目标是设计一个状态反馈控制律 $u(x)$, 将系统从初始状态 x_0 镇定到平衡状态, 同时使得在其作用下的 (无限时间) 代价函数

$$J(x_k)=\sum_{p=k}^{\infty}\gamma^{p-k}U(x_p,u_p) \tag{2}$$

达到最小, 其中, U 是效用函数, $U(0,0)=0$, 且对于任意的 x_p, u_p, 有 $U(x_p,u_p)\geq 0$, 折扣因子 γ 满足 $0<\gamma\leq 1$. 方便讨论起见, 选取二次型形式的效用函数 $U(x_p,u_p)=x_p^{\mathrm{T}}Qx_p+u_p^{\mathrm{T}}Ru_p$, 其中, Q 和 R 为正定矩阵. 事实上, 对于最优控制问题, 待设计的反馈控制律不仅能够在 Ω_x 上镇定被控系统, 而且使得相应的代价函数有限, 这就是容许控制的概念[11−12, 16].

根据经典的最优控制理论, 最优代价函数

$$J^*(x_k)=\min_{u_k,u_{k+1},\cdots,u_\infty}\sum_{p=k}^{\infty}\gamma^{p-k}U(x_p,u_p)$$

可以写为

$$J^*(x_k)=\min_{u_k}\Big\{U(x_k,u_k)+\gamma\min_{u_{k+1},u_{k+2},\cdots,u_\infty}\sum_{p=k+1}^{\infty}\gamma^{p-k-1}U(x_p,u_p)\Big\}$$

于是, $J^*(x_k)$ 满足离散时间 HJB 方程

$$J^*(x_k)=\min_{u_k}\{U(x_k,u_k)+\gamma J^*(x_{k+1})\} \tag{3}$$

相应的最优控制为

$$u^*(x_k)=\arg\min_{u_k}\{U(x_k,u_k)+\gamma J^*(x_{k+1})\} \tag{4}$$

注 1. 通过式 (4) 发现, 求解当前时刻 k 的最优控制 u^*, 需要得到最优代价 J^*, 但是却与系统下一时刻的状态向量 x_{k+1} 有关, 这在当前时刻是不能做到的. 因此, 在难以得到 HJB 方程解析解的情况下, 有必要研究如何获得其近似解. ADP 以及随后出现的迭代 ADP 算法, 就是为了克服这些难题而提出的近似求解方法.

2 迭代 ADP 算法及其收敛性

根据迭代 ADP 算法的基本思想[11−13, 16], 需要构建两个序列, 即代价函数序列 $\{V_i(x_k)\}$ 和控制律序列 $\{v_i(x_k)\}$, 通过迭代运算得到收敛性结论. 这里, 记 i 为迭代指标, 并初始化代价函数 $V_0(\cdot)=0$. 对于 $i=0,1,\cdots$, 迭代过程包括不断计算控制律

$$v_i(x_k)=\arg\min_{u_k}\big\{U(x_k,u_k)+\gamma V_i(x_{k+1})\big\}=\arg\min_{u_k}\big\{U(x_k,u_k)+\gamma V_i(F(x_k,u_k))\big\} \tag{5}$$

和更新代价函数

$$V_{i+1}(x_k)=\min_{u_k}\big\{U(x_k,u_k)+\gamma V_i(x_{k+1})\big\}=U(x_k,v_i(x_k))+\gamma V_i(F(x_k,v_i(x_k))) \tag{6}$$

直到算法收敛 (当 $i\to\infty$ 时, 有 $V_i\to J^*$ 和 $v_i\to u^*$).

在证明上述迭代算法的收敛性与最优性之前, 首先给出下面两个引理[11−12, 16] .

引理 1. (有界性) 定义代价函数序列 $\{V_i(x_k)\}$ 如式 (6) 所示. 如果系统可控, 则存在一个上界 Y 使得对于任意的 i, 都有 $0\leq V_i(x_k)\leq Y$ 成立.

引理 2. (单调性) 定义代价函数序列 $\{V_i(x_k)\}$ 如式 (6) 所示且有 $V_0(\cdot)=0$, 同时定义控制律序列 $\{v_i(x_k)\}$ 如式 (5) 所示. 那么, $\{V_i(x_k)\}$ 是一个单调非减序列, 即 $0\leq V_i(x_k)\leq V_{i+1}(x_k)$, $\forall i$.

定理 1. 定义代价函数序列 $\{V_i(x_k)\}$ 如式 (6) 所示, 且 $V_0(\cdot)=0$, 控制律序列 $\{v_i(x_k)\}$ 如式 (5) 所示. 执行迭代 ADP 算法, 代价函数序列 $\{V_i(x_k)\}$ 收敛于离散时间 HJB 方程中的最优代价函数 $J^*(x_k)$, 即当 $i\to\infty$ 时, 有 $V_i(x_k)\to J^*(x_k)$. 相应地, 当 $i\to\infty$ 时, $\{v_i(x_k)\}$ 收敛于最优控制律 $u^*(x_k)$, 即 $\lim_{i\to\infty}v_i(x_k)=u^*(x_k)$.

证明. 根据引理 1 和引理 2, 代价函数序列 $\{V_i(x_k)\}$ 单调非减且有上界, 所以, 它的极限存在. 定义 $\lim_{i\to\infty}V_i(x_k)=V_\infty(x_k)$ 为其极限.

一方面, 对于任意的 u_k 和 i, 根据式 (6), 可得

$$V_i(x_k)\leq U(x_k,u_k)+\gamma V_{i-1}(x_{k+1}) \tag{7}$$

由引理 2, 对于任意的 i, 都有 $V_i(x_k)\leq V_\infty(x_k)$ 成立. 因此, 式 (7) 变为

$$V_i(x_k)\leq U(x_k,u_k)+\gamma V_\infty(x_{k+1}),\ \forall i$$

令 $i\to\infty$, 则

$$V_\infty(x_k)\leq U(x_k,u_k)+\gamma V_\infty(x_{k+1}) \tag{8}$$

考虑到式 (8) 中的控制向量 u_k 是任意的, 可以得到

$$V_\infty(x_k)\leq\min_{u_k}\{U(x_k,u_k)+\gamma V_\infty(x_{k+1})\} \tag{9}$$

另一方面, 由于对任意的 i, 迭代过程中的代价函数满足

$$V_i(x_k)=\min_{u_k}\big\{U(x_k,u_k)+\gamma V_{i-1}(x_{k+1})\big\}$$

再次考虑 $V_i(x_k)\leq V_\infty(x_k)$, 我们有

$$V_\infty(x_k)\geq\min_{u_k}\big\{U(x_k,u_k)+\gamma V_{i-1}(x_{k+1})\big\},\ \forall i$$

令 $i\to\infty$, 则

$$V_\infty(x_k) \geq \min_{u_k}\left\{U(x_k,u_k)+\gamma V_\infty(x_{k+1})\right\} \quad (10)$$

结合式 (9) 和式 (10), 可以得到

$$V_\infty(x_k) = \min_{u_k}\left\{U(x_k,u_k)+\gamma V_\infty(x_{k+1})\right\}$$

同样地, 记 $\lim_{i\to\infty} v_i(x_k) = v_\infty(x_k)$ 为控制律序列 $\{v_i(x_k)\}$ 的极限. 根据式 (5) 和式 (6), 有

$$V_\infty(x_k) = \min_{u_k}\left\{U(x_k,u_k)+\gamma V_\infty(x_{k+1})\right\} = U(x_k,v_\infty(x_k))+\gamma V_\infty(F(x_k,v_\infty(x_k))) \quad (11)$$

其中,

$$v_\infty(x_k) = \arg\min_{u_k}\left\{U(x_k,u_k)+\gamma V_\infty(x_{k+1})\right\} \quad (12)$$

注意式 (11) 和式 (3), 同时注意式 (12) 和式 (4), 可以得到, $V_\infty(x_k) = J^*(x_k)$ 和 $v_\infty(x_k) = u^*(x_k)$, 即, $\lim_{i\to\infty} V_i(x_k) = J^*(x_k)$ 且 $\lim_{i\to\infty} v_i(x_k) = u^*(x_k)$. 由此验证了迭代算法的收敛性和最终得到的控制律的最优性. □

注 2. 利用迭代代价函数的表达式 (6), 依据迭代指标 i 逐次进行递推, 我们有

$$\begin{aligned}
&V_{i+1}(x_k) = U(x_k,v_i(x_k))+\gamma V_i(x_{k+1})\\
&V_i(x_{k+1}) = U(x_{k+1},v_{i-1}(x_{k+1}))+\gamma V_{i-1}(x_{k+2})\\
&\qquad\vdots\\
&V_1(x_{k+i}) = U(x_{k+i},v_0(x_{k+i}))+\gamma V_0(x_{k+i+1})
\end{aligned}$$

进而, 考虑到 $V_0(x_{k+i+1}) = 0$ 这一事实, 可以将迭代代价函数 $V_{i+1}(x_k)$ 写成关于效用函数加和的形式

$$V_{i+1}(x_k) = \sum_{l=0}^{i}\gamma^l U(x_{k+l},v_{i-l}(x_{k+l})) \quad (13)$$

观察式 (13) 可以发现, 在迭代代价函数 $V_{i+1}(x_k)$ 中, 构成效用函数的控制输入序列是由一个控制律组 $(v_i, v_{i-1}, \cdots, v_0)$ 产生的, 即其中的每一个控制输入都依赖于不同的控制律, 因此控制输入是 $v_{i-l}(x_{k+l})$ 的形式, 其中, $l = 0, 1, \cdots, i$. 尽管如此, 最终作用到被控对象的控制律, 是经过上述迭代算法之后得到的收敛的 (状态反馈) 控制律. 事实上, 根据定理 1 和容许控制的概念, 最终得到的 $v_\infty = u^*$ 是一个可以镇定系统的稳定控制. 在其作用下, 将会产生一个控制输入序列, 实现被控非线性系统的最优调节.

3　迭代神经动态规划及其实现

由于这里研究的被控对象是一般的非线性系统, 难以直接求解 HJB 方程. 虽然通过执行迭代 ADP 算法 (5) 和 (6), 可以从理论上得到最优控制律和最优代价函数, 但是迭代控制律和代价函数的信息是不能精确获得的, 而且进行迭代运算需要被控系统的近似动态信息. 所以, 利用函数近似结构 (例如神经网络) 来重构系统动态以及 $v_i(x_k)$ 和 $V_i(x_k)$. 这里, 将基于神经动态规划思想的迭代 ADP 算法称为迭代神经动态规划方法. 本节给出基于 GDHP 技术的迭代神经动态规划实现方案, 包含构建三种神经网络, 即模型网络、评判网络和执行网络.

3.1　模型网络

为了不依赖被控系统的动态信息 $F(x_k,u_k)$, 在执行主要的迭代过程之前, 首先构建一个模型网络并记隐藏层神经元个数为 N_m, 输入层到隐藏层的权值矩阵为 $\nu_m \in \mathbf{R}^{(n+m)\times N_m}$, 隐藏层到输出层的权值矩阵为 $\omega_m \in \mathbf{R}^{N_m\times n}$. 输入状态向量 x_k 和近似的控制向量 $\hat{v}_i(x_k)$ 如下文所示, 模型网络的输出为

$$\hat{x}_{k+1} = \omega_m^{\mathrm{T}}\sigma\left(\nu_m^{\mathrm{T}}\left[x_k^{\mathrm{T}}, \hat{v}_i^{\mathrm{T}}(x_k)\right]^{\mathrm{T}}\right)$$

其中, $\sigma(\cdot) \in \mathbf{R}^{N_m}$ 为激活函数 (下同). 模型网络的误差函数为 $e_{mk} = \hat{x}_{k+1} - x_{k+1}$, 训练目标函数为 $E_{mk} = (1/2)e_{mk}^{\mathrm{T}}e_{mk}$. 利用梯度下降法更新模型网络的权值矩阵

$$\omega_m^{(j+1)} = \omega_m^{(j)} - \alpha_m\left[\frac{\partial E_{mk}}{\partial \omega_m^{(j)}}\right]$$

$$\nu_m^{(j+1)} = \nu_m^{(j)} - \alpha_m\left[\frac{\partial E_{mk}}{\partial \nu_m^{(j)}}\right]$$

其中, $\alpha_m > 0$ 是模型网络的学习率且 j 是训练权值参数的迭代指标. 当模型网络经过充分学习之后, 保持其权值不再改变, 并开始执行迭代神经动态规划的主要步骤, 即训练评判网络和执行网络.

3.2　评判网络

评判网络的作用是近似代价函数 $V_i(x_k)$ 及其偏导数 $\frac{\partial V_i(x_k)}{\partial x_k}$ (称为协函数, 记为 $\lambda_i(x_k)$, 即 $\lambda_i(x_k) := \frac{\partial V_i(x_k)}{\partial x_k}$. 根据定理 1, 当 $i\to\infty$ 时, $V_i(x_k)\to J^*(x_k)$. 由于 $\lambda_i(x_k) = \frac{\partial V_i(x_k)}{\partial x_k}$, 则相应的协函数序列 $\{\lambda_i(x_k)\}$ 在 $i\to\infty$ 时也是收敛的, 即 $\lambda_i(x_k)\to\lambda^*(x_k)$. 这在仿真研究中也会得到验证.

设评判网络的隐藏层神经元个数为 N_c, 输入层到隐藏层的权值矩阵为 $\nu_c \in \mathbf{R}^{n\times N_c}$, 隐藏层到输出层的权值矩阵为 $\omega_c \in \mathbf{R}^{N_c\times(n+m)}$. 在进行第 i 次迭代时, 可以将权值矩阵写为 ν_{ci} 和 ω_{ci}, 于是, 评判网

络的输出为

$$\begin{bmatrix} \hat{V}_i(x_k) \\ \hat{\lambda}_i(x_k) \end{bmatrix} = \begin{bmatrix} \omega_{ci}^{V\mathrm{T}} \\ \omega_{ci}^{\lambda\mathrm{T}} \end{bmatrix} \sigma\left(\nu_{ci}^{\mathrm{T}} x_k\right) = \omega_{ci}^{\mathrm{T}} \sigma\left(\nu_{ci}^{\mathrm{T}} x_k\right)$$

其中, $\omega_{ci} = \left[\omega_{ci}^{V}, \omega_{ci}^{\lambda}\right]$. 展开来写, 有

$$\hat{V}_i(x_k) = \omega_{ci}^{V\mathrm{T}} \sigma\left(\nu_{ci}^{\mathrm{T}} x_k\right)$$

$$\hat{\lambda}_i(x_k) = \omega_{ci}^{\lambda\mathrm{T}} \sigma\left(\nu_{ci}^{\mathrm{T}} x_k\right)$$

这里, GDHP 技术中评判网络的结构如图 1 所示. 可以看出, 它将 HDP 和 DHP 技术中的评判网络进行了融合.

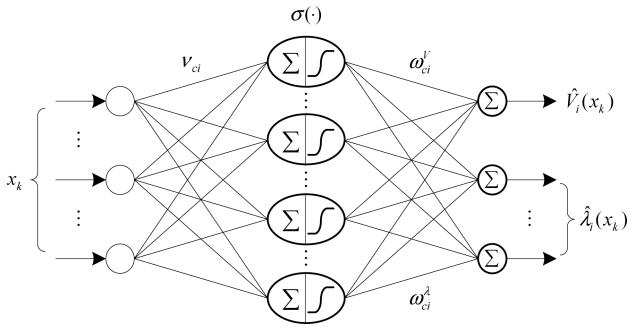

图 1 评判网络结构

Fig. 1 The architecture of critic network

在 GDHP 实现结构中, 评判网络的训练目标由代价函数和协函数两部分组成, 即

$$V_i(x_k) = U(x_k, \hat{v}_{i-1}(x_k)) + \gamma \hat{V}_{i-1}(\hat{x}_{k+1})$$

$$\lambda_i(x_k) = 2Qx_k + 2\left(\frac{\partial \hat{v}_{i-1}(x_k)}{\partial x_k}\right)^{\mathrm{T}} R\hat{v}_{i-1}(x_k) + \gamma\left(\frac{\partial \hat{x}_{k+1}}{\partial x_k} + \frac{\partial \hat{x}_{k+1}}{\partial \hat{v}_{i-1}(x_k)} \frac{\partial \hat{v}_{i-1}(x_k)}{\partial x_k}\right)^{\mathrm{T}} \times \hat{\lambda}_{i-1}(\hat{x}_{k+1})$$

训练过程的误差函数包括两项, 即 $e_{cik}^{V} = \hat{V}_i(x_k) - V_i(x_k)$, $e_{cik}^{\lambda} = \hat{\lambda}_i(x_k) - \lambda_i(x_k)$, 而需要最小化的目标函数为

$$E_{cik} = (1-\beta)E_{cik}^{V} + \beta E_{cik}^{\lambda}$$

其中, $E_{cik}^{V} = (1/2)e_{cik}^{V\mathrm{T}} e_{cik}^{V}$, $E_{cik}^{\lambda} = (1/2)e_{cik}^{\lambda\mathrm{T}} e_{cik}^{\lambda}$. 利用梯度下降法更新评判网络的权值矩阵, 即

$$\omega_{ci}^{(j+1)} = \omega_{ci}^{(j)} - \alpha_c \left[(1-\beta)\frac{\partial E_{cik}^{V}}{\partial \omega_{ci}^{(j)}} + \beta \frac{\partial E_{cik}^{\lambda}}{\partial \omega_{ci}^{(j)}}\right]$$

$$\nu_{ci}^{(j+1)} = \nu_{ci}^{(j)} - \alpha_c \left[(1-\beta)\frac{\partial E_{cik}^{V}}{\partial \nu_{ci}^{(j)}} + \beta \frac{\partial E_{cik}^{\lambda}}{\partial \nu_{ci}^{(j)}}\right]$$

其中, $\alpha_c > 0$ 为评判网络的学习率, j 为更新权值参数的迭代指标, $0 \le \beta \le 1$ 是一个常数, 反映 HDP 和 DHP 在 GDHP 技术中相结合的权重大小.

注 3. 这里采用的 GDHP 技术综合了 HDP 能够直接输出代价函数和 DHP 控制效果好的优点. 虽然引入协函数会在一定程度上增加计算复杂度, 但是可以获得比初等的 ADP 方法 (例如 HDP) 更好的运行效果.

3.3 执行网络

构建执行网络的作用是近似控制律, 设其隐藏层神经元个数为 N_a, 输入层到隐藏层的权值矩阵为 $\nu_a \in \mathbf{R}^{n\times N_a}$, 隐藏层到输出层的权值矩阵为 $\omega_a \in \mathbf{R}^{N_a\times m}$. 在上述的迭代环境下, 我们将权值矩阵写成 $\nu_{a(i-1)}$ 和 $\omega_{a(i-1)}$ 的形式, 则执行网络的输出为

$$\hat{v}_{i-1}(x_k) = \omega_{a(i-1)}^{\mathrm{T}} \sigma\left(\nu_{a(i-1)}^{\mathrm{T}} x_k\right)$$

这里, 误差函数定义为 $e_{a(i-1)k} = \hat{V}_{i-1}(x_{k+1}) - S_k$, 其中, $S_k = 0$ 是 $\hat{V}_{i-1}(x_{k+1})$ 的目标值, 需要最小化的目标函数为 $E_{a(i-1)k} = (1/2)e_{a(i-1)k}^{\mathrm{T}} e_{a(i-1)k}$. 在这种设置下, 执行网络输出合适的控制律, 使得系统的代价函数达到最小. 执行网络的权值更新算法仍然为梯度下降法, 即

$$\omega_{a(i-1)}^{(j+1)} = \omega_{a(i-1)}^{(j)} - \alpha_a \left[\frac{\partial E_{a(i-1)k}}{\partial \omega_{a(i-1)}^{(j)}}\right]$$

$$\nu_{a(i-1)}^{(j+1)} = \nu_{a(i-1)}^{(j)} - \alpha_a \left[\frac{\partial E_{a(i-1)k}}{\partial \nu_{a(i-1)}^{(j)}}\right]$$

其中, $\alpha_a > 0$ 是执行网络的学习率, j 是更新权值参数的迭代指标.

总的来说, 本文提出的迭代神经动态规划的结构如图 2 所示, 其中, 模块 γDX 表示 $\hat{x}_{k+1}$ 关于 x_k 的偏导数计算结果 $n \times n$ 方阵的 γ 倍.

注 4. 传统的迭代 ADP 算法, 例如文献 [11−18], 在训练执行网络时需要利用控制矩阵的直接信息或者其神经网络表示. 其中, 针对仿射系统[11−13, 15, 17], 需要系统控制矩阵的直接信息 $g(x_k)$[11, 12, 17], 或者辨识控制矩阵得到其近似表示 $\hat{g}(x_k)$[13, 15]; 针对非仿射系统[14, 16, 18], 也需要神经网络表示. 那样, 执行网络的训练目标为

$$v_{i-1}(x_k) = -\frac{\gamma}{2} R^{-1} \hat{g}^{\mathrm{T}}(x_k) \hat{\lambda}_{i-1}(\hat{x}_{k+1})$$

误差函数定义为 $\bar{e}_{a(i-1)k} = \hat{v}_{i-1}(x_k) - v_{i-1}(x_k)$, 在此基础上训练执行网络. 这样的实现方法, 很大程度上依赖于控制系统的动态信息, 尤其是控制矩阵的信息. 这里提出的迭代神经动态规划方法, 不仅沿用迭代 ADP 算法的基本框架, 能够保证迭代算法的收敛性; 而且引入神经动态规划的思想, 放松对系统动态的要求, 所以更利于达到数据驱动控制的目的.

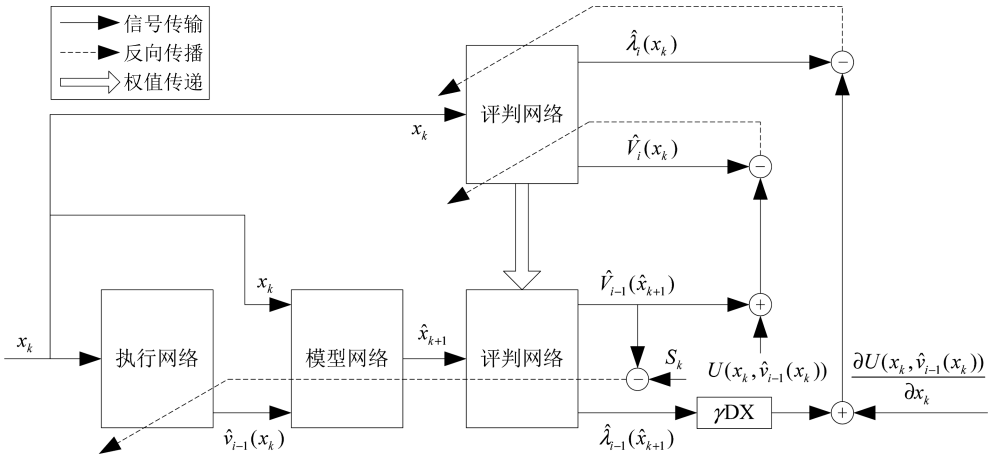

图 2　迭代神经动态规划结构

Fig. 2　The architecture of iterative neural dynamic programming

3.4　设计步骤

设 x_k 为任意可控状态, $J^*(x_k)$ 为最优代价函数. 根据定理 1 中的收敛性结论, 当迭代指标 $i \to \infty$ 时, $V_i(x_k) \to J^*(x_k)$. 但是, 在计算机实现中, 不可能无限地执行迭代算法. 从工程应用角度来看, 我们更关心是否存在一个有限的 i, 使得

$$|J^*(x_k) - V_i(x_k)| \leq \varepsilon \tag{14}$$

成立. 因此, 将 $J^*(x_k)$ 和 $V_i(x_k)$ 之间的误差 ε 引入迭代 ADP 算法, 使得代价函数序列 $\{V_i(x_k)\}$ 能够在经过有限次迭代之后收敛. 从这个角度来看, 这里设计的控制器实现了对被控系统近似最优调节的目的. 实际上, 这种近似意义上的收敛, 能够满足一般的设计需求; 也是 ADP 方法在无法精确求解 HJB 方程的背景下, 进行近似最优控制设计的体现.

但是, 也应该看到, 在一般情况下, 最优代价函数 $J^*(x_k)$ 事先未知, 难以利用停止准则 (14) 来验证迭代算法是否达到要求. 因此, 这里提出一种相对容易判定的算法停止准则, 即

$$|V_{i+1}(x_k) - V_i(x_k)| \leq \varepsilon \tag{15}$$

定理 2. 对于非线性系统 (1) 和代价函数 (2), 在使用迭代神经动态规划方法时, 由式 (14) 和式 (15) 描述的两种收敛性准则是等价的.

证明. 一方面, 若 $|J^*(x_k) - V_i(x_k)| \leq \varepsilon$ 成立, 则有 $J^*(x_k) \leq V_i(x_k) + \varepsilon$. 根据引理 2 和定理 1 可知 $V_i(x_k) \leq V_{i+1}(x_k) \leq J^*(x_k)$ 成立. 于是, 有 $V_i(x_k) \leq V_{i+1}(x_k) \leq V_i(x_k) + \varepsilon$. 即, $0 \leq V_{i+1}(x_k) - V_i(x_k) \leq \varepsilon$, 也即式 (15) 成立.

另一方面, 根据定理 1, $|V_{i+1}(x_k) - V_i(x_k)| \to 0$ 意味着 $V_i(x_k) \to J^*(x_k)$. 这样, 如果对于任意小的 ε 都有 $|V_{i+1}(x_k) - V_i(x_k)| \leq \varepsilon$ 成立, 则当 i 相当大时, $|J^*(x_k) - V_i(x_k)| \leq \varepsilon$ 成立. 由此证明了两种准则的等价性. □

考虑到神经网络的近似作用, 在具体的实现过程中, 采用近似的代价函数构建停止准则, 即 $|\hat{V}_{i+1}(x_k) - \hat{V}_i(x_k)| \leq \varepsilon$. 这里给出利用迭代神经动态规划方法设计非线性系统近似最优调节器的具体步骤, 如算法 1 所示.

算法 1. 迭代神经动态规划方法

步骤 1. 设置算法的最大迭代次数 $i_{\max}$ 和计算精度 ε. 选取被控系统的初始状态 x_0 和效用函数中的权值矩阵 Q 和 R. 初始化三种神经网络的权值矩阵.

步骤 2. 基于系统输入/输出数据, 构建并训练模型网络, 充分学习系统动态, 固定并输出最终权值.

步骤 3. 令 $i = 0$, 选取初始代价函数 $\hat{V}_0(x_k) = 0$, 并直接计算初始控制律 $\hat{v}_0(x_k)$.

步骤 4. 构建评判网络并更新其权值矩阵, 输出代价函数 $\hat{V}_1(x_k)$ 及其偏导数 $\hat{\lambda}_1(x_k)$. 如果 $|\hat{V}_1(x_k)| \leq \varepsilon$, 停止迭代; 否则转到步骤 5.

步骤 5. 更新迭代指标, 令 $i = i + 1$.

步骤 6. 构建并训练执行网络, 计算近似控制律 $\hat{v}_i(x_k)$.

步骤 7. 进一步训练评判网络, 近似代价函数 $\hat{V}_{i+1}(x_k)$ 及其偏导数 $\hat{\lambda}_{i+1}(x_k)$.

步骤 8. 如果 $|\hat{V}_{i+1}(x_k) - \hat{V}_i(x_k)| \leq \varepsilon$, 停止迭代, 输出执行网络的最终权值, 转到步骤 10; 否则, 转到步骤 9.

步骤 9. 如果 $i > i_{\max}$, 停止迭代, 输出执行网络的最终权值, 转到步骤 10; 否则, 转到步骤 5.

步骤 10. 利用执行网络的最终权值, 得到实现被控系统近似最优调节的反馈控制律.

注 5. 定理 2 的重要作用在于, 它提供了利用迭代神经动态规划方法实现离散时间非线性系统近似最优调节的具有实用意义的设计准则. 因此, 在实际应用中, 我们可以运行算法 1 得到合理可行的结果.

4　仿真实验

本节开展两个仿真实验: 1) 针对仿射非线性系

统; 2) 针对非仿射形式的一般非线性系统.

例 1. 考虑离散时间 (仿射) 非线性系统

$$x_{k+1}=\begin{bmatrix}0.2x_{1k}\mathrm{e}^{x_{2k}^2}\\0.3x_{2k}^3\end{bmatrix}+\begin{bmatrix}0\\1\end{bmatrix}u_k \tag{16}$$

这是对文献 [14] 和文献 [20] 中仿真例子的修改, 其中, $x_k=[x_{1k},x_{2k}]^{\mathrm{T}}\in\mathbf{R}^2$ 和 $u_k\in\mathbf{R}$ 分别是被控系统的状态向量和控制向量. 选取二次型形式的效用函数 $U(x_k,u_k)=x_k^{\mathrm{T}}x_k+u_k^{\mathrm{T}}u_k$.

利用三层反向传播 (Back propagation) 神经网络来构建模型网络、评判网络和执行网络, 且三者的结构分别为 3-8-2、2-8-3 和 2-8-1. 激活函数通常选取为

$$[\sigma(\xi)]_j=\frac{\mathrm{e}^{\xi_j}-\mathrm{e}^{-\xi_j}}{\mathrm{e}^{\xi_j}+\mathrm{e}^{-\xi_j}}$$

其中, ξ 是一个列向量且维数与隐藏层神经元个数相同, ξ_j 代表该向量的第 j 个分量.

注 6. 这里对隐藏层神经元个数的设定主要是凭借工程经验, 同时在计算精度要求和计算复杂度之间取得一个折衷方案.

利用迭代神经动态规划方法, 运行算法 1, 首先需要训练模型网络: 输入层和隐藏层、隐藏层和输出层之间的权值分别在区间 $[-0.5,0.5]$ 和 $[-0.1,0.1]$ 中随机初始化. 参数设置 (如学习率) 会在一定程度上影响算法的收敛速度. 我们通过实验选取合适的学习率 $\alpha_m=0.1$, 采集 500 组数据进行学习, 并在训练结束之后保持其权值不再变化. 其次, 评判网络和执行网络的初始权值都在区间 $[-0.1,0.1]$ 中随机选取. 然后, 选取折扣因子 $\gamma=1$, GDHP 技术的调节参数 $\beta=0.5$, 在 $k=0$ 时刻执行神经动态规划方法完成 59 次迭代 (即 $i=1,2,\cdots,59$), 使得计算误差达到预先定义的精度 10^{-6}. 在每次迭代中, 都对评判网络和执行网络分别进行 2 000 次训练, 并且学习率参数取为 $\alpha_c=\alpha_a=0.05$. 评判网络和执行网络的权值矩阵范数的收敛结果如图 3 所示. 这里, 我们对比两种不同的实现方法的收敛效果. 这种不同主要体现在对执行网络的训练方法上 (如第 3.3 节和注 4 所述). 对于 $k=0$ 和 $x_0=[0.5,-1]^{\mathrm{T}}$, 代价函数及其偏导数序列的收敛过程如图 4 所示 (清楚起见, 只刻画前 15 次迭代的结果), 其中, 星线代表本文提出的迭代神经动态规划方法, 点线代表传统的迭代 ADP 算法[12−18] (下同). 可以发现, 迭代神经动态规划方法在不利用系统动态信息的情况下, 也基本达到了和传统迭代 ADP 算法一样的收敛效果, 这验证了迭代神经动态规划方法的有效性.

最后, 对于给定的初始状态 $x_0=[0.5,-1]^{\mathrm{T}}$, 我们将基于两种不同实现方法的 GDHP 近似最优控制律运用于被控对象 (16). 在运行 15 个时间步后得到的系统状态响应曲线及相应的控制曲线分别如图 5 和图 6 所示. 由此可以清楚地看到, 采用两种不同的实现方法得到的控制效果是很相近的. 这再次验证了融合迭代 ADP 算法, 神经动态规划思想, 和 GDHP 技术的优点.

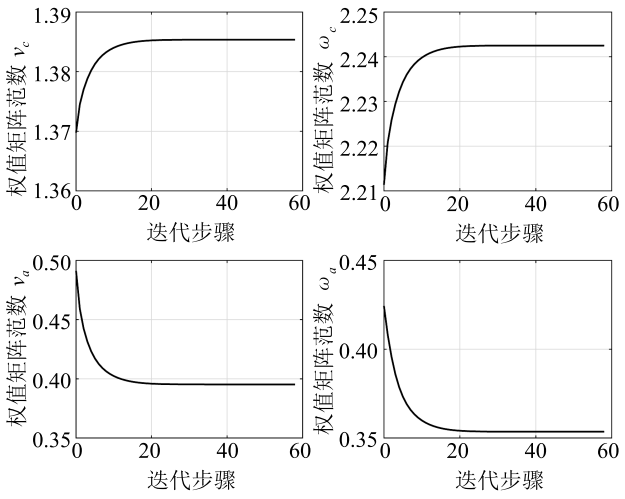

图 3 权值矩阵范数的收敛过程

Fig. 3 The convergence process of the norm of weight matrices

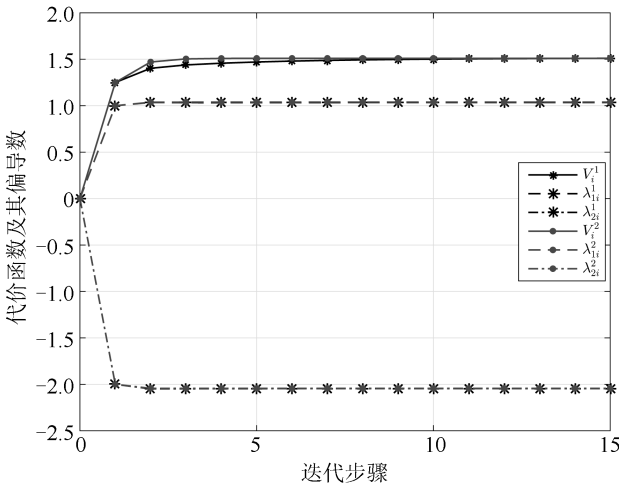

图 4 代价函数及其偏导数的收敛过程

Fig. 4 The convergence process of the cost function and its derivative

例 2. 考虑离散时间 (非仿射) 非线性系统

$$x_{k+1}=-0.5x_k^2+\sin(x_k+\tanh(u_k)) \tag{17}$$

其中, $x_k\in\mathbf{R}$ 和 $u_k\in\mathbf{R}$ 分别是被控系统的状态向量和控制向量. 构建模型网络、评判网络和执行网络, 且三者的结构分别为 2-6-1、1-6-2 和 1-6-1. 首先训练模型网络, 得到的最终权值为

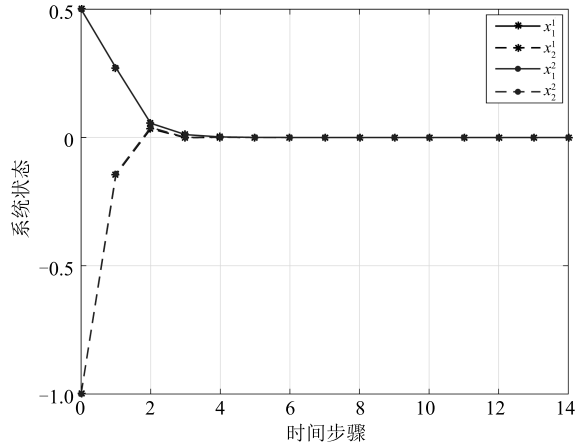

图 5　系统状态轨迹 x

Fig. 5　The system state trajectory x

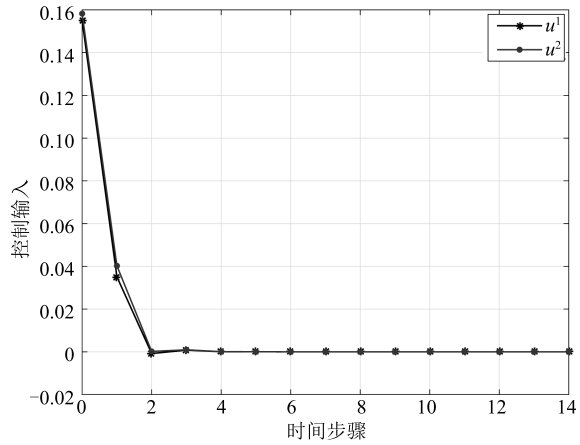

图 6　控制输入轨迹 u

Fig. 6　The control input trajectory u

$$\nu_m = \begin{bmatrix} -0.3190 & 0.0704 \\ 0.0644 & 0.0348 \\ 0.6628 & 0.5020 \\ 0.8737 & 1.1051 \\ -0.6330 & -0.4545 \\ -0.3225 & -0.0963 \end{bmatrix}^{\mathrm{T}}, \quad \omega_m = \begin{bmatrix} 0.2029 \\ -0.0089 \\ 0.2647 \\ 0.8784 \\ -0.2180 \\ 0.0828 \end{bmatrix}$$

对于评判网络和执行网络, 选取初始的权值矩阵分别为

$$\nu_c = \begin{bmatrix} 0.0888 \\ 0.0646 \\ 0.0606 \\ 0.0653 \\ -0.0849 \\ 0.0697 \end{bmatrix}^{\mathrm{T}}, \quad \omega_c = \begin{bmatrix} -0.0601 & -0.0741 \\ -0.0443 & -0.0398 \\ 0.0768 & 0.0221 \\ 0.0806 & 0.0302 \\ -0.0395 & -0.0133 \\ 0.0168 & 0.0798 \end{bmatrix}$$

$$\nu_a = \begin{bmatrix} -0.0723 \\ 0.0565 \\ -0.0909 \\ -0.0311 \\ 0.0490 \\ 0.0734 \end{bmatrix}^{\mathrm{T}}, \quad \omega_a = \begin{bmatrix} 0.0990 \\ -0.0476 \\ 0.0479 \\ -0.0581 \\ 0.0101 \\ -0.0772 \end{bmatrix}$$

其他参数设置同例 1. 在 $k = 0$ 时刻执行算法 1 并完成 19 次迭代, 使得计算误差达到预先定义的精度 10^{-5}. 评判网络和执行网络的权值矩阵范数的收敛结果如图 7 所示. 对于 $k = 0$ 和 $x_0 = 0.8$, 代价函数及其偏导数序列的收敛过程如图 8 所示. 最后, 对于给定的初始状态 $x_0 = 0.8$, 利用 GDHP 技术和迭代神经动态规划方法得到的最优控制律运用于被控对象 (17), 在运行 60 个时间步后得到的系统状态响应曲线及相应的控制曲线如图 9 所示. 这些仿真结果验证了迭代神经动态规划设计方法的有效性.

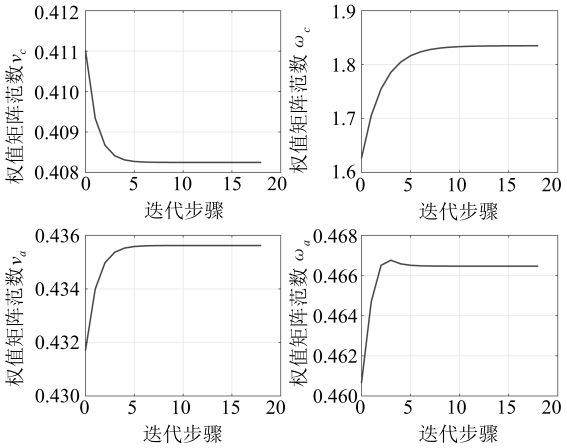

图 7　权值矩阵范数的收敛过程

Fig. 7　The convergence process of the norm of weight matrices

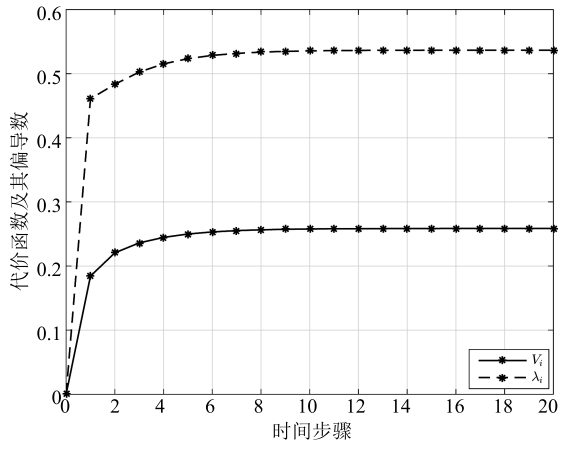

图 8　代价函数及其偏导数的收敛过程

Fig. 8　The convergence process of the cost function and its derivative

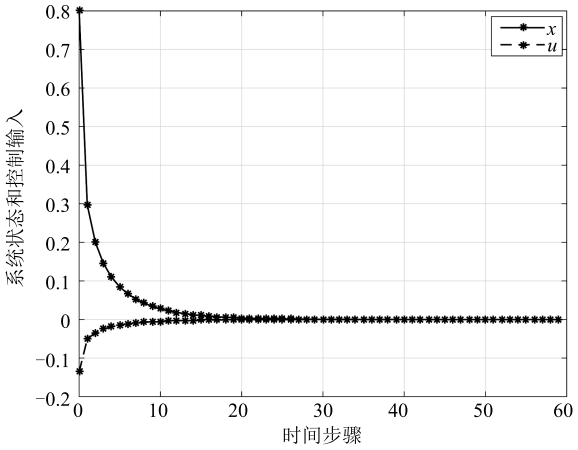

图 9 系统状态轨迹 x 和控制输入轨迹 u

Fig. 9 The system state trajectory x and control input trajectory u

5 结论

本文利用基于数据的思想, 建立针对离散时间非线性系统近似最优调节的迭代神经动态规划方法. 提出离散时间非仿射非线性系统的迭代 ADP 算法并且证明其满足收敛性与最优性. 通过构建三种神经网络 (模型网络、评判网络和执行网络), 结合 GDHP 技术, 给出迭代算法的具体实现步骤. 在这种新颖的迭代神经动态规划结构中, 训练执行网络不需要利用系统动态信息, 尤其是仿射非线性系统 $x_{k+1} = f(x_k) + g(x_k)u_k$ 中的控制矩阵 $g(x_k)$. 这在很大程度上减少了迭代算法对系统动态的依赖, 改进了以往的实现结构. 通过仿真研究, 验证了本文建立的数据驱动最优调节器设计策略的有效性. 值得注意的是, 本文研究的是无限时间近似最优控制问题. 如何将神经动态规划思想与有限时间迭代 ADP 算法[14] 相结合, 改进执行网络的训练方法, 从而将迭代神经动态规划方法推广到有限时间近似最优调节器设计是值得深入研究的主题之一. 另外, 本文目前的研究侧重于理论方面的收敛性分析和具体的算法实现，如何将提出的方法应用于实际系统也有待于进一步讨论.

References

1 Bellman R E. *Dynamic Programming*. Princeton, NJ: Princeton University Press, 1957.

2 Werbos P J. Approximate dynamic programming for real-time control and neural modeling. *Handbook of Intelligent Control*. New York: Van Nostrand Reinhold, 1992.

3 Lewis F L, Vrabie D, Vamvoudakis K G. Reinforcement learning and feedback control: using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems*, 2012, **32**(6): 76−105

4 Zhang Hua-Guang, Zhang Xin, Luo Yan-Hong, Yang Jun. An overview of research on adaptive dynamic programming. *Acta Automatica Sinica*, 2013, **39**(4): 303−311
(张化光, 张欣, 罗艳红, 杨珺. 自适应动态规划综述. 自动化学报, 2013, **39**(4): 303−311)

5 Liu De-Rong, Li Hong-Liang, Wang Ding. Data-based self-learning optimal control: research progress and prospects. *Acta Automatica Sinica*, 2013, **39**(11): 1858−1870
(刘德荣, 李宏亮, 王鼎. 基于数据的自学习优化控制: 研究进展与展望. 自动化学报, 2013, **39**(11): 1858−1870)

6 Hou Z S, Wang Z. From model-based control to data-driven control: survey, classification and perspective. *Information Sciences*, 2013, **235**: 3−35

7 Prokhorov D V, Wunsch D C. Adaptive critic designs. *IEEE Transactions on Neural Networks*, 1997, **8**(5): 997−1007

8 Sutton R S, Barto A G. *Reinforcement Learning — An Introduction*. Cambridge, MA: MIT Press, 1998.

9 Si J, Wang Y T. Online learning control by association and reinforcement. *IEEE Transactions on Neural Networks*, 2001, **12**(2): 264−276

10 Wang Fei-Yue. Parallel control: a method for data-driven and computational control. *Acta Automatica Sinica*, 2013, **39**(4): 293−302
(王飞跃. 平行控制: 数据驱动的计算控制方法. 自动化学报, 2013, **39**(4): 293−302)

11 Al-Tamimi A, Lewis F L, Abu-Khalaf M. Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof. *IEEE Transactions on Systems, Man, Cybernetics, Part B, Cybernetics*, 2008, **38**(4): 943−949

12 Zhang H G, Luo Y H, Liu D R. Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints. *IEEE Transactions on Neural Networks*, 2009, **20**(9): 1490−1503

13 Dierks T, Thumati B T, Jagannathan S. Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence. *Neural Networks*, 2009, **22**(5−6): 851−860

14 Wang F Y, Jin N, Liu D R, Wei Q L. Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ε-error bound. *IEEE Transactions on Neural Networks*, 2011, **22**(1): 24−36

15 Liu D R, Wang D, Zhao D B, Wei Q L, Jin N. Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming. *IEEE Transactions on Automation Science and Engineering*, 2012, **9**(3): 628−634

16 Wang D, Liu D R, Wei Q L, Zhao D B, Jin N. Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming. *Automatica*, 2012, **48**(8): 1825−1832

17 Zhang H G, Qin C B, Luo Y H. Neural-network-based constrained optimal control scheme for discrete-time switched nonlinear system using dual heuristic programming. *IEEE Transactions on Automation Science and Engineering*, 2014, **11**(3): 839−849

18 Liu D R, Li H L, Wang D. Error bounds of adaptive dynamic programming algorithms for solving undiscounted optimal control problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2015, **26**(6): 1323−1334

19 Zhong X N, Ni Z, He H B. A theoretical foundation of goal representation heuristic dynamic programming. *IEEE Transactions on Neural Networks and Learning Systems*, 2016, **27**(12): 2513−2525

20 Heydari A, Balakrishnan S N. Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics. *IEEE Transactions on Neural Networks and Learning Systems*, 2013, **24**(1): 145−157

21 Jiang Y, Jiang Z P. Robust adaptive dynamic programming and feedback stabilization of nonlinear systems. *IEEE Transactions on Neural Networks and Learning Systems*, 2014, **25**(5): 882−893

22 Na J, Herrmann G. Online adaptive approximate optimal tracking control with simplified dual approximation structure for continuous-time unknown nonlinear systems. *IEEE/CAA Journal of Automatica Sinica*, 2014, **1**(4): 412−422

23 Liu D R, Yang X, Wang D, Wei Q L. Reinforcement-learning-based robust controller design for continuous-time uncertain nonlinear systems subject to input constraints. *IEEE Transactions on Cybernetics*, 2015, **45**(7): 1372−1385

24 Luo B, Wu H N, Huang T W. Off-policy reinforcement learning for H_∞ control design. *IEEE Transactions on Cybernetics*, 2015, **45**(1): 65−76

25 Mu C X, Ni Z, Sun C Y, He H B. Air-breathing hypersonic vehicle tracking control based on adaptive dynamic programming. *IEEE Transactions on Neural Networks and Learning Systems*, 2017, **28**(3): 584−598

26 Wang D, Liu D R, Zhang Q C, Zhao D B. Data-based adaptive critic designs for nonlinear robust optimal control with uncertain dynamics. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2016, **46**(11): 1544−1555

王 鼎 中国科学院自动化研究所副研究员. 2009 年获得东北大学理学硕士学位, 2012 年获得中国科学院自动化研究所工学博士学位. 主要研究方向为自适应与学习系统, 智能控制, 神经网络. 本文通信作者.
E-mail: ding.wang@ia.ac.cn
(**WANG Ding** Associate professor at the Institute of Automation, Chinese Academy of Sciences. He received his master degree in operations research and cybernetics from Northeastern University, Shenyang, China and his Ph. D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2009 and 2012, respectively. His research interest covers adaptive and learning systems, intelligent control, and neural networks. Corresponding author of this paper.)

穆朝絮 天津大学电气自动化与信息工程学院副教授. 2012 年获得东南大学工学博士学位. 主要研究方向为非线性控制理论与应用, 智能控制与优化, 智能电网. E-mail: cxmu@tju.edu.cn
(**MU Chao-Xu** Associate professor at the School of Electrical and Information Engineering, Tianjin University. She received her Ph. D. degree in control science and engineering from Southeast University, Nanjing, China, in 2012. Her research interest covers nonlinear control and application, intelligent control and optimization, and smart grid.)

刘德荣 北京科技大学教授. 主要研究方向为自适应动态规划, 计算智能, 智能控制与信息处理, 复杂工业系统建模与控制. E-mail: derong@ustb.edu.cn
(**LIU De-Rong** Professor at University of Science and Technology Beijing. His research interest covers adaptive dynamic programming, computational intelligence, intelligent control and information processing, and modeling and control for complex industrial systems.)