

基于神经网络近似的自适应优化控制

林小峰, 黄元君

(广西大学 电气工程学院, 广西 南宁 530004)

摘要:为了解决初始和终端确定的一类离散时间非线性系统有限时间优化控制, 利用动态规划原理求解过程中遇到维数灾的问题, 提出了基于神经网络的自适应动态规划近似优化控制。在分析动态规划求解遇到维数灾的基础上, 进而给出了迭代ADP算法, 并采用神经网络近似代价函数和控制律来实现迭代ADP算法, 设计近似优化控制器。通过matlab实验仿真结果表明, 采用迭代ADP算法能够避免求解中遇到的维数灾, 从而有效地实现了一类离散时间非线性系统的有限时间近似优化控制。

关键词:非线性系统; 迭代ADP; 神经网络; 优化控制

中图分类号: TP183

文献标识码: A

文章编号: 1673-629X(2011)11-0100-05

Adaptive Optimal Control Based on Neural Network Approximation

LIN Xiao-feng, HUANG Yuan-jun

(School of Electrical Engineering, Guangxi University, Nanning 530004, China)

Abstract: In order to solve the finite optimal control for a class of discrete-time nonlinear system with the initial and terminal certain, meeting the problem of dimension disaster in solving by dynamic programming, an adaptive iterative approximate control method based on neural network is proposed. Analyze the problem of the dynamic programming solving meets dimension disaster, then give the iterative ADP algorithms, which is realized by neural network approximate cost function and control law, by the iterative ADP algorithm design the approximate optimal controller. Through the matlab experimental simulation, the results show that the iteration ADP algorithm can avoid the problem of the dimension disaster, effectively realized the finite approximation optimal control for a class of discrete-time nonlinear system.

Key words: nonlinear systems; iterative ADP; neural networks; optimal control

0 引言

自适应动态规划(Adaptive Dynamic Programming, ADP)是把自适应评价设计和强化学习的思想与动态规划原理相结合而提出的一种方法^[1]。由于它采用了前向迭代求解, 可避免动态规划中的“维数灾”问题, 并获得Hamilton-Jacobi-Bellman(HJB)方程的近似解, 成为解决最优控制问题的有效工具, 吸引了大批学者的关注^[1-7]。文献[2]中提出了贪婪的迭代方案以求解非线性离散系统的最优控制问题。文献[3, 4]研究了自适应动态规划算法的稳定性和最优性, 从数学上严格证明了从一个初始稳定的控制策略开始迭

代, 通过给出的迭代算法能够使得系统最终求得所需的近似最优控制, 并基于HJB方程给出了算法的收敛性分析。文献[5]针对有限时间最优控制问题, 提出了带误差界 ϵ 的自适应动态规划算法。然而, 对有限时间最优控制问题的迭代ADP算法, 如何有效的近似求解研究较少。

在迭代ADP算法中, 最优控制很难解析地求解, 因此需要寻找一种近似求解。文中用迭代ADP方法, 通过神经网络来近似逼近非线性函数^[6-10]得到近似最优解, 从而实现了有限时间的最优控制。首先, 分析了动态规划求解有限时间最优控制, 出现维数灾问题; 其次, 给出了迭代ADP算法及性质, 推导了神经网络近似函数公式; 最后, 实验仿真验证了该算法的有效性。

1 问题描述

考虑如下一类离散时间非线性系统:

收稿日期: 2011-04-02; 修回日期: 2011-07-10

基金项目: 国家自然科学基金(60964002); 国家自然科学基金重点项目(61034002); 广西研究生教育创新计划资助项目(105930003009)

作者简介: 林小峰(1955-), 男, 教授, 研究方向为智能优化控制、过程控制; 黄元君(1983-), 男, 硕士研究生, 研究方向为智能优化控制。

$$x(k+1) = F(x(k), u(k)), k = 0, 1, 2, \dots, N \quad (1)$$

其中 $x(k) \in \mathbb{R}^n$ 是状态向量, $u(k) \in \mathbb{R}^m$ 是控制向量, $x(0)$ 是初始状态向量。对 $\forall x(k), u(k)$ 系统函数 $F(x(k), u(k))$ 是离散的并且有 $F(0, 0) = 0$, 最优控制问题即系统(1)从状态 $x(0)$ 开始在有限容许控制序列 $\underline{u}(\cdot) = (u(0), u(1), \dots, u(N-1))$ 作用下使下面的性能指标函数极小化

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

其中 $U(x, u)$ 称为效用函数, 对任意 (x, u) , $U(x, u) \geq 0$ 且 $U(0, 0) = 0$ 。如果存在一个控制序列 $\underline{u}(\cdot)$ 使得 $F(x(0), \underline{u}(\cdot)) = 0$, 则称初始状态 $x(0)$ 是可控的, 同时控制序列 $\underline{u}(\cdot)$ 称为容许控制序列。

根据以上定义有限终端时间不确定的最优性能指标可写成

$$J^*(x) = \min\{J(x(0), \underline{u}(\cdot))\} \quad (3)$$

根据(2)式和贝尔曼最优性原理 $J^*(x)$ 的离散时间 HJB 方程可写成

$$J_{k+1}^*(x) = \min\{U(x(k), u(k)) + J_k^*(F(x(k), u(k)))\} \quad (4)$$

其中 $u(k) = u(x(k))$

最优控制向量 $u^*(x)$ 表示为

$$u^*(x) = \arg \min_{u(k)} \{U(x(k), u(k)) + J_k^*(F(x(k), u(k)))\} \quad (5)$$

当利用动态规划后向求解有限时间最优控制, 其第一步是由(6)式决定函数 $u^*(x(N-1))$

$$u^*(x(N-1)) = \arg \min_{u(N-1)} \{U(x(N-1), u(N-1))\} \quad (6)$$

s. t. $F(x(N-1), u(N-1)) = 0$

再求出其相关的性能指标函数

$$J^*(x(N-1)) = U(x(N-1), u^*(x(N-1))) \quad (7)$$

对函数 $u^*(x(N-1))$ 和 $J^*(x(N-1))$ 定出后, 可利用方程(4)和(5), 定出所有的函数 $u^*(x(k))$ 和性能指标函数 $J^*(x(k))$, $k = N-2, N-3, \dots, 1$, 再根据已知的 $x(0)$, 解出 $u^*(0) = u^*(x(0))$, 再把 $u^*(0)$ 代入系统方程(1)得到 $x(1) = F(x(0), u^*(0))$ 。然后利用 $x(1)$ 和函数 $u^*(x(1))$ 解出 $u^*(1)$, 再用 $u^*(1)$ 代入系统方程(1)得到 $x(2) = F(x(1), u^*(1))$, 重复这个过程, 可得到最优控制序列 $\underline{u}^*(x(0)) = \{u^*(0), u^*(1), \dots, u^*(N-1)\}$ 。

离散动态规划的思想是后向求解^[11], 注意到这里必须计算和保存所有次数 $k = 0, 1, 2, \dots, N$ 的 $J^*(x(k))$ 和 $u^*(x(k))$, 因此必须考虑“维数灾”, 利用动态规划通常在计算上无法获得最优解。

2 迭代 ADP 算法

2.1 迭代 ADP 算法公式推导

为了区别 HJB 中的性能指标函数 $J(x(k))$ 在迭代 ADP 算法中与之对应的 $V(x(k))$ 称为代价函数。迭代从 $i=0$ 初始化 $V_0(x(k))=0$ 开始, 这里 $x(k)$ 是初始状态。

当 $i=1$ 时, 代价值根据下式计算

$$\begin{aligned} V_1(x(k)) &= \min\{U(x(k), u(k)) + V_0(F(x(k), u(k)))\} \quad \text{s. t. } F(x(k), u(k)) = 0 \\ &= \min\{U(x(k), u(k))\} \quad \text{s. t. } F(x(k), u(k)) = 0 \\ &= U(x(k), u_1(k)) \end{aligned} \quad (8)$$

上式中 $V_0(F(x(k), u(k))) = 0$ 且满足 $F(x(k), u_1(k)) = 0$, 其中

$$\begin{aligned} u_1(x(k)) &= \arg \min_{u_1(k)} U(x(k), u_1(k)) \\ \text{s. t. } F(x(k), u_1(k)) &= 0 \end{aligned} \quad (9)$$

对 $i=2, 3, 4, \dots$ 迭代 ADP 算法在下面(10)式和(11)式之间迭代更新

$$\begin{aligned} V_i(x(k)) &= \min\{U(x(k), u(k)) + V_{i-1}(F(x(k), u(k)))\} \\ &= U(x(k), u_i(k)) + V_{i-1}(F(x(k), u_i(k))) \end{aligned} \quad (10)$$

其中

$$\begin{aligned} u_i(x(k)) &= \arg \min_{u_i(k)} \{U(x(k), u_i(k)) + V_{i-1}(F(x(k), u_i(k)))\} \end{aligned} \quad (11)$$

方程(8)~(11)为迭代 ADP 算法, 类似于 HJB 方程, 其主要思想是通过式(10)和(11)之间根据当前递归迭代的值来更新代价函数 $V(x(k))$ 和控制律 $u(x(k))$ 。但与 HJB 方程不同在于 HJB 中求解出的最优性能指标 $J^*(x(k))$ 和最优控制律 $u^*(x(k))$ 是唯一的, 而迭代 ADP 算法得到的代价函数值和控制律对 $\forall i \neq j, V_i(x(k)) \neq V_j(x(k)), u_i(x(k)) \neq u_j(x(k))$, 即迭代过程中, 不同的迭代步数, 得到不同的控制律。

2.2 迭代算法的性质

1) 对于给定的状态, 此算法在每个循环中改进代价函数和控制律函数, 每一个后继的代价总比前一个代价要小, 即有 $V_i(x(k)) \geq V_{i+1}(x(k))$ 。

2) 当系统状态 $x(k)$ 是可控并且代价函数 $V_i(x)$ 收敛于最优性能指标函数 $J^*(x)$, 则迭代控制律 $u_i(x)$ 收敛于最优控制律 $u^*(x)$ 。

以上性质证明见文献[5]。

3 基于神经网络的迭代 ADP 算法实现

由于在迭代 ADP 算法中的控制律函数 $u_i(x(k))$ 和代价函数 $V_i(x(k))$ 一般为非线性的, 通常无法得到显示的表达式, 因此文中采用函数近似的方法来获得。

3.1 神经网络近似代价函数和控制律

文中用神经网络近似迭代 ADP 算法中的代价函数 $V_i(x)$ 和控制律 $u_i(x)$ 。

$$\hat{V}_i(x) = \sum_{j=1}^L w_{vi}^j \varphi_j(x) = W_{vi}^T \varphi(x) \quad (12)$$

$$\hat{u}_i(x) = \sum_{j=1}^L w_{ui}^j \sigma_j(x) = W_{ui}^T \sigma(x) \quad (13)$$

其中, $\varphi_j(x), \sigma_j(x) \in C^1(\Omega)$ 是激励函数, 且 $\varphi_j(0) = \sigma_j(0) = 0$ 。 w_{vi}^j, w_{ui}^j 为神经网的权值, L 是隐层中的神经元个数。

激励函数向量矩阵为

$$\varphi(x) = [\varphi_1(x) \varphi_2(x) \cdots \varphi_L(x)]^T, \sigma(x) = [\sigma_1(x) \sigma_2(x) \cdots \sigma_L(x)]^T$$

权值向量

$$W_{vi} = [w_{vi}^1 w_{vi}^2 \cdots w_{vi}^L]^T, W_{ui} = [w_{ui}^1 w_{ui}^2 \cdots w_{ui}^L]^T$$

(1) 代价函数的近似。

每一次迭代中, 满足最小方差要求来近似下面的目标函数(14)式

$$\begin{aligned} d(\varphi(x), W_{vi}^T, W_{ui}^T) &= U(x(k), u_i(k)) + V_i(x(k+1)) \\ &= U(x(k), u_i(k)) + W_{vi}^T \varphi(x) \end{aligned} \quad (14)$$

根据(12)式中的 W_{vi} 和目标函数(14)式的关系, 利用最小方差来调整 W_{vi} 。

$$W_{vi+1} = \arg \min_{W_{vi+1}} \int_{\Omega} \|W_{vi+1}^T \varphi(x) - d(\varphi(x), W_{vi}^T, W_{ui}^T)\|^2 dx \quad (15)$$

其中 $W_{vi+1}^T \varphi(x) - d(\varphi(x), W_{vi}^T, W_{ui}^T) = e_L(x)$ 为误差, 求解最小方差使用权残值法调整^[12], 即权 W_{vi+1} 修正根据投影 $\frac{de_L(x)}{dW_{vi+1}}$, 对 $\forall x \in \Omega$ 满足下式内积。

$$\left\langle \frac{de_L(x)}{dW_{vi+1}}, e_L(x) \right\rangle = 0 \quad (16)$$

$\langle f, g \rangle = \int_{\Omega} fg^T dx$ 是 Lebesgue 积分, 于是式(16)可写成

成

$$0 = \int_{\Omega} \varphi(x(k)) (\varphi^T(x(k)) W_{vi+1} - d^T(\varphi(x(k)), W_{vi}^T, W_{ui}^T)) dx \quad (17)$$

$$\begin{aligned} W_{vi+1} &= \left(\int_{\Omega} \varphi(x(k)) \varphi^T(x(k)) dx \right)^{-1} \\ &\times \int_{\Omega} \varphi(x(k)) d^T(\varphi(x(k)), W_{vi}^T, W_{ui}^T) dx \end{aligned} \quad (18)$$

假设 1 $\varphi_j(x(k))$ 在集 Ω 上是线性独立的。

根据假设 1 可得 $\left(\int_{\Omega} \varphi(x(k)) \varphi^T(x(k)) dx \right)^{-1}$ 是满秩, 这意味着它是可逆的, 于是方程(18)存在唯一解。

(2) 控制律的近似。

每一次迭代中对控制律(11)式, 用权值调整来近似, 根据(13)式的神经网络函数表达式, 若(13)式用 $\hat{u}_i(x(k), W_{ui})$ 表示, 则(11)式可以重写成

$$\begin{aligned} W_{ui} &= \arg \min_{W_{ui}} \{ U(x(k), \hat{u}_i(x(k), W_{ui})) \\ &+ \hat{V}_{i-1}(x(k+1)) \} \end{aligned} \quad (19)$$

其中 $x(k+1) = F(x(k), u_i(x(k), W_{ui}))$ 。注意到(19)式中的 W_{ui} 是隐式, 因为它决定 $x(k+1)$, 所以很难得到显式的解。因此可用最小方差法来更新权值

$$\begin{aligned} W_{ui} |_{m+1} &= W_{ui} |_m - \alpha \frac{\partial (U(x(k), \hat{u}_i(x(k), W_{ui} |_m)) + \hat{V}_{i-1}(x(k+1)))}{\partial W_{ui}} \\ &= W_{ui} |_m - \alpha \frac{\partial (U(x(k), \hat{u}_i(x(k), W_{ui} |_m)) + W_{vi}^T \varphi(x(k+1)))}{\partial W_{ui}} \end{aligned} \quad (20)$$

其中 α 表示正步长, m 是迭代步数, 于是当 $m \rightarrow \infty$ 有 $W_{ui} |_m \rightarrow W_{ui}$, 式(20)得到求解。

3.2 基于神经网络近似的迭代 ADP 结构

图 1 给出了基于神经网络近似函数的迭代 ADP 算法结构图。图中实线表示状态前向计算, 两条细虚线表示根据误差调整网络的权值。条形箭头表示代价网络的权值更新。

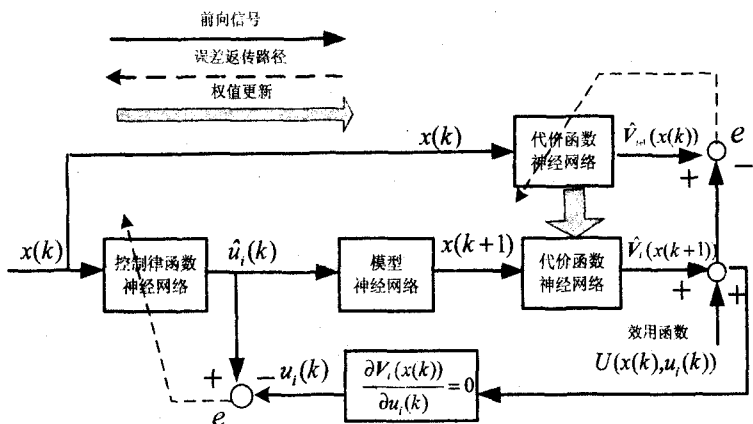


图 1 基于神经网络的迭代 ADP 算法结构图

3.3 迭代 ADP 算法步骤

迭代 ADP 算法步骤如下：

(1) 选择迭代最大次数 i_{\max} , 每次迭代中函数近似循环步数为 j_{\max} 和近似误差界限为 ε 。

(2) 若模型表达式未知, 则可用神经网络近似模型, 先训练好模型神经网络。

(3) 初始化各参数, 设定代价神经网络的初始权值为 W_{vi} , 控制律神经网络的初始权值为 W_{ui} 。

(4) 从对象区间内选择状态向量 $x(k)$ 通过(11)式(当 $i=1$ 时, 利用式(9))计算其对应的输出目标为 $u_i(x(k))$, 其中下一时刻的状态向量 $x(k+1)$ 由方程(1)(或模型网)计算得到。

(5) 利用数据 $(x(k), u_i(x(k)))$ 根据式(20)来

更新控制网络的权参数 W_{ui} 一共 j_{\max} 次来获得近似的最优控制律 $\hat{u}_i(x)$ 。

(6) 根据向量 $x(k)$ 通过(10)式(当 $i=1$ 时, 利用式(8)) 计算其对应的代价函数输出目标为 $V_i(x(k))$ 。

(7) 令 $W_{vi+1} = W_{vi}$, 利用数据集 $(x(k), V_i(x(k)))$ 根据式(18)更新神经网络权值 W_{vi} 获得近似代价函数 $\hat{V}_i(x)$ 。

(8) 如果 $|\hat{V}_{i+1}(x(k)) - \hat{V}_i(x(k))| \leq \varepsilon$

转步骤(10); 否则, 转步骤(9)。

(9) 如果 $i \geq i_{\max}$, 转步骤(10); 否则, 令 $i = i + 1$ 转步骤(4)。

(10) 结束。获得近似的最优控制律 $\hat{u}_i(x(k))$, 令最优的近似控制律为 $\hat{u}_i = u_i = u^*$ 。

4 仿真分析

考虑如下非线性系统

$$x(k+1) = f(x(k)) + g(x(k))u(k) \quad (21)$$

其中

$$f(x(k)) = \begin{bmatrix} -0.5x_1(k)\exp(x_2(k)^4) \\ 0.2\sin(x_1(k)^4) \end{bmatrix}$$

$$g(x(k)) = \begin{bmatrix} x_1(k)x_2(k) & 0.1 \\ x_2(k) & 0.5x_2(k) \end{bmatrix}, Q = R = I_{2 \times 2}$$

代价函数为二次型

$$V(x(k)) = \sum_{i=k}^{N-1} (x(i)^T Q x(i) + u(i)^T R u(i)) \quad (22)$$

选初始状态 $x(k) = (1, -1)$ 。根据迭代 ADP 算法和结构图 1, 控制律和代价函数近似网络分别选用 2-10-2, 2-8-1 的 BP 神经网络参数结构。学习率 $\beta_u = \alpha_v = 0.01$, 内循环 $j_{\max} = 100$, 外循环迭代次数 $i_{\max} = 30$, $\varepsilon = 10^{-3}$ 。

4.1 迭代 ADP 算法收敛性的验证

图 2 为运行算法得到(8)式对应的 $V_i(x(k))$ 值。图 3 为整个迭代过程中 $V_i(x(k))$ 的值, 可以看到代价函数值随着迭代步数 i 增大, 代价值在一步一步减小,

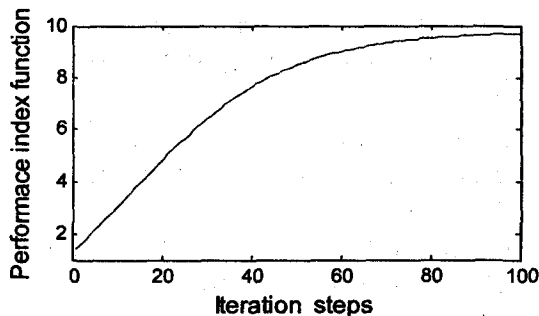


图 2 $i=0$ 时 $V_i(x(k))$ 的值

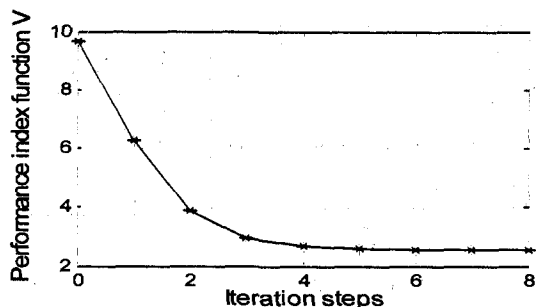


图 3 $i=1, 2, \dots$ 时 $V_i(x(k))$ 的序列值

验证了性质 1, $V_i(x(k)) \geq V_{i+1}(x(k))$, 即 $V_i(x(k))$ 是非增序列。当 $i=8$ 时, 代价函数值不再减小, 于是 $V_i(x(k))$ 近似收敛于 $J^*(x(k))$, 验证了性质 2。根据性质 2, 迭代得到控制律 $u_i(x(k))$ 也将近似地收敛最优 $u^*(x)$ 。

4.2 函数近似的收敛性

图 4 和图 5 给出了在每一次迭代中, 内层循环 100 步的误差收敛情况, 说明了用以上推导的神经网络近似方法能有效地逼近控制律和代价函数。

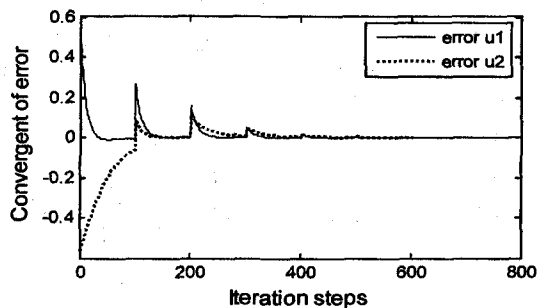


图 4 控制律近似的误差收敛过程

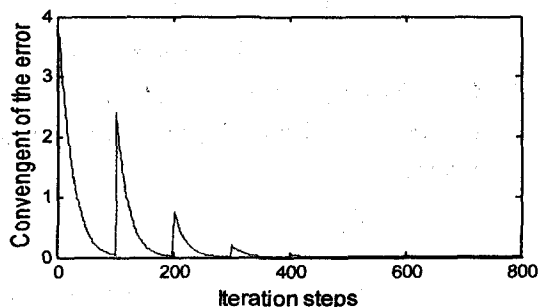


图 5 代价函数近似的误差收敛过程

4.3 控制效果

为验证神经网络训练得到的控制效果, 取任意状态 $x_A = (1, -1)$, $x_B = (-1, 1)$, $x_C = (0.5, -0.5)$, $x_D = (-0.5, 0.5)$ 进行控制测试。图 6 给出了对应的控制轨迹和状态轨迹。可以看到, 神经网络控制器有效地控制以上各状态, 并且在 8 步有限的时间内达到稳定点。

5 结束语

文中分析了迭代算法与 HJB 方程之间的关系, 设计了神经网络控制器。利用迭代 ADP 算法, 通过神经

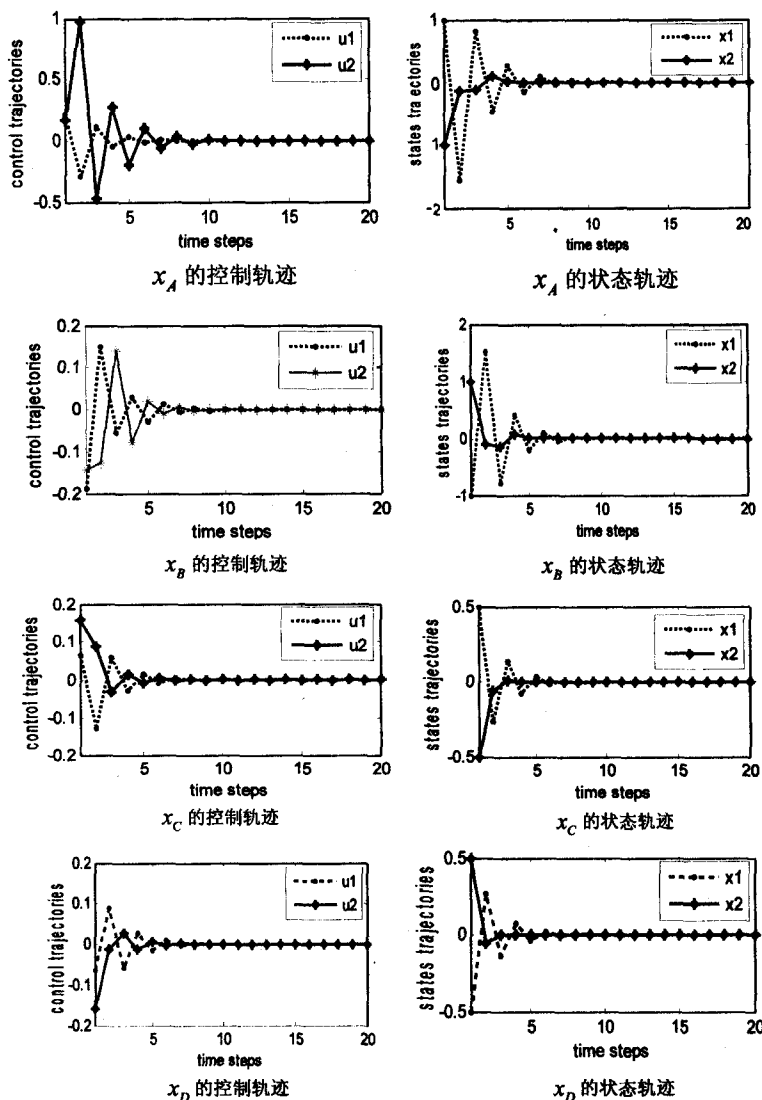


图 6 控制轨迹和状态轨迹

网络近似非线性函数,得到代价函数和最优控制律的近似,实现对离散非线性系统优化控制,实验仿真验证了该算法的有效性。

参考文献:

- [1] Werbos P J. Approximate dynamic programming for real control and neural modeling [M]//Handbook of Intelligent Control: Neural Fuzzy, and Adaptive Approaches. New York: Van Nostrand Reinhold, 1992.
- [2] Al-Tamimi A, Lewis F L, Abu-Khalaf M. Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2008, 38(4): 943-949.
- [3] Liu D, Xiong X, Zhang Y. Action-dependent adaptive critic designs [C]//Proceedings of the INNS-IEEE International Joint Conference on Neural Networks. Washington, DC: [s. n.], 2001: 990-995.
- [4] Murray J J, Cox C J, Lendaris G G, et al. Adaptive dynamic programming [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2002, 32(2): 140-153.
- [5] Wang Feiyue, Jin Ning, Liu Derong, et al. Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ε -error bound [J]. IEEE Transactions on Neural Networks, 2011, 22(1): 24-36.
- [6] 韩力群. 神经网络教程 [M]. 北京: 北京邮电大学出版社, 2006.
- [7] 王蓓, 刘桥. 优化 BP 神经网络的可靠性预测模型 [J]. 计算机技术与发展, 2007, 17(9): 102-105.
- [8] 洪素惠, 吴发成, 米红. 神经网络自适应 PID 在吹瓶机中的应用 [J]. 计算机技术与发展, 2009, 19(9): 177-180.
- [9] 王忠, 孙钰. 基于神经网络的自适应彩色图像盲水印算法 [J]. 计算机技术与发展, 2006, 16(12): 108-113.
- [10] 梁久祯, 何新贵, 周家庆. 神经网络 BP 学习算法动力学分析 [J]. 自动化学报, 2002, 28(5): 729-735.
- [11] 胡寿松, 王执铨, 胡维礼. 最优控制理论与系统 [M]. 第 2 版. 北京: 科学出版社, 2005.
- [12] Abu-Khalaf M, Lewis F L. Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach [J]. Automatica, 2005, 41(5): 779-791.
- [10] 葛志辉, 李陶深, 张继成. 无线 Mesh 网络逐层信道分配策略研究 [J]. 广西大学学报 (自然科学版), 2010, 35(6): 13-17.
- [11] 李陶深, 韦燕霞, 葛志辉. 基于跨层负载感知的无线 mesh 网络拥塞控制算法 [J]. 北京邮电大学学报 (自然科学版), 2011, 34(1): 50-54.
- [12] 郝建民. 信道带宽与功率互换原理在扩频体制中不成立 [J]. 遥测遥控, 1997, 19(4): 57-60.
- [13] 徐雷鸣, 庞博, 赵耀. NS 与网络仿真 [M]. 北京: 人民邮电出版社, 2003.

(上接第 99 页)

(APBE) in IEEE 802.11 Wireless LANs [C]//Ubiquitous Information Technologies and Applications (CUTE). International Conference Center, Sany, Hainan: [s. n.], 2010: 1-11.

- [8] Oyman, Paulraj A J. Power-Bandwidth Tradeoff in Dense Multi-Antenna Relay Networks [J]. IEEE Transaction on Wireless Communications, 2007, 6(7): 2282-2293.
- [9] Rodas J, Escudero C J. Dynamic path-loss estimation using a particle filter [J]. International Journal of Computer Science Issues, 2010, 7(4): 1-5.