一类控制受约束非线性系统的基于单网络贪婪迭代 DHP 算法的近似最优镇定

罗艳红[1,2] 张化光[1,2] 曹 宁[2] 陈 兵[3]

摘 要 提出一种贪婪迭代 DHP (Dual heuristic programming) 算法, 解决了一类控制受约束非线性系统的近似最优镇定问题. 针对系统的控制约束, 首先引入一个非二次泛函把约束问题转换为无约束问题, 然后基于协状态函数提出一种贪婪迭代 DHP 算法以求解系统的 HJB (Hamilton-Jacobi-Bellman) 方程. 在算法的每个迭代步, 利用一个神经网络来近似系统的协状态函数, 而后根据协状态函数直接计算系统的最优控制策略, 从而消除了常规近似动态规划方法中的控制网络. 最后通过两个仿真例子证明了本文提出的最优控制方案的有效性和可行性.

关键词 贪婪迭代, 约束, 非二次泛函, 最优控制, 神经网络
中图分类号 TP183

Near-optimal Stabilization for a Class of Nonlinear Systems with Control Constraint Based on Single Network Greedy Iterative DHP Algorithm

LUO Yan-Hong[1,2] ZHANG Hua-Guang[1,2] CAO Ning[2] CHEN Bing[3]

Abstract The near-optimal stabilization problem for nonlinear constrained systems is solved by greedy iterative DHP (Dual heuristic programming) algorithm. Considering the control constraint of the system, a nonquadratic functional is first introduced in order to transform the constrained problem into a unconstrained problem. Then based on the costate function, the greedy iterative DHP algorithm is proposed to solve the Hamilton-Jacobi-Bellman (HJB) equation of the system. At each step of the iterative algorithm, a neural network is utilized to approximate the costate function, and then the optimal control policy of the system can be computed directly according to the costate function, which removes the action network appearing in the ordinary approximate dynamic programming (ADP) method. Finally, two examples are given to demonstrate the validity and feasibility of the proposed optimal control scheme.

Key words Greedy iterative, constraint, nonquadratic functional, optimal control, neural network

美国学者 Bellman 于 1957 年提出了求解多级决策过程最优化的动态规划方法. 从理论上看, 动态规划方法特别适用于目标函数的极小化或极大化问题, 如能量极小化问题、效益极大化问题等. 但该方法随着问题规模的增大, 计算量也迅速增大乃至出现 "维数灾" 问题, 故而其只适用于小规模简单非线性系统的最优控制问题. 70 年代开始有人着手研究动态规划问题的近似解法[1−3], 这类近似解法可用于传统动态规划方法无法求解的目标函数极小化或极大化最优控制问题中, 被称为近似动态规划 (Approximate dynamic programming, ADP) 方法. 其主要原理为通过训练一个神经网络来逐渐逼近动态规划问题中的目标函数并且同时训练另一个神经网络来产生最优控制信号. 近几年, 这类近似动态规划方法引起了不少研究者的注意并取得了一些成果, 如文献 [4 – 19]. 在文献 [14] 中, 一种贪婪的启发式动态规划 (Heuristic dynamic programming, HDP) 迭代方案被提出以求解非线性离散系统的最优控制问题, 其首次提出了求解非线性系统的近似最优控制可以从任意控制策略开始迭代的方案.

然而, 尽管近似动态规划方法已经在最优控制领域取得了较大的进步, 但是如何利用近似动态规划方法求解带约束系统的最优控制问题仍然是个难题. 对于带有输入约束的系统, 其最优控制解特别

收稿日期 2008-09-10 收修改稿日期 2009-06-12
Received September 10, 2008; in revised form June 12, 2009
国家自然科学基金 (60534010, 60774048, 60728307), 国家高技术研究发展计划 (863 计划) (2006AA04Z183), 长江学者和创新团队发展计划 (60521003) 和高等学校学科创新引智计划 (B08015) 资助
Supported by National Natural Science Foundation of China (60534010, 60774048, 60728307), National High Technology Research and Development Program of China (863 Program) (2006AA04Z183), the Program for Changjiang Scholars and Innovative Research Groups of China (60521003), and the Programme of Introducing Talents of Discipline to Universities (B08015)
1. 东北大学流程工业综合自动化教育部重点实验室 沈阳 110004 2. 东北大学信息科学与工程学院 沈阳 110004 3. 青岛大学复杂性科学研究所 青岛 266071
1. Key Laboratory of Integrated Automation for the Process Industry, Ministry of Education, Northeastern University, Shenyang 110004 2. School of Information Science and Engineering, Northeastern University, Shenyang 110004 3. Institute of Complexity Science, Qingdao University, Qingdao 266071
DOI: 10.3724/SP.J.1004.2009.01436

是光滑的最优控制解, 很难解析地获得. 为了解决这个难题, 本文通过引入非二次泛函来克服被控系统控制受约束的问题[20]. 通过使用非二次泛函可以得到对应的哈密顿–雅可比–贝尔曼 (Hamilton-Jacobi-Bellman, HJB) 方程, 通过求解这个 HJB 方程可以获得光滑有界的控制策略. 然而, 如何求解 HJB 方程以得到最优的值函数仍旧是个难题. 因此, 本文进一步提出一种新型的近似动态规划算法即贪婪迭代的二次启发式动态规划 (Dual heuristic programming, DHP) 算法以近似求解控制受约束系统的 HJB 方程, 并给出了严格的收敛性证明. 另外, 为了便于贪婪迭代 DHP 算法的实现, 首先引入一个神经网络来近似逼近协状态函数, 然后通过这个协状态函数来直接求取最优的控制策略. 这里的控制策略是根据协状态函数直接求得的一个解析式, 因而无需像以往的近似动态规划方法一样引入另一个控制网络来近似控制策略, 从而消除了控制网络的使用, 在降低计算复杂度的同时还大大提高了计算精度.

1　问题陈述

考虑如下一类非线性系统

$$\boldsymbol{x}(k+1)=\boldsymbol{f}(\boldsymbol{x}(k))+g(\boldsymbol{x}(k))\boldsymbol{u}(k) \tag{1}$$

其中, $\boldsymbol{x}(k)\in\mathbf{R}^n$ 是系统的状态向量, 函数 $\boldsymbol{f}:\mathbf{R}^n\to\mathbf{R}^n$ 和 $g:\mathbf{R}^n\to\mathbf{R}^{n\times m}$ 分别对各自的自变量是可微的, 并且满足 $\boldsymbol{f}(0)=\mathbf{0}$. 假设 $\boldsymbol{f}+g\boldsymbol{u}$ 在一个 $\mathbf{R}^n$ 上包含原点的区域 Ω_x 里是李普希兹连续的, 并且假设系统 (1) 是可控的, 即在 Ω_x 上至少存在一个能够渐近镇定系统的连续的控制策略. 控制向量 $\boldsymbol{u}(k)\in\Omega_u$, $\Omega_u=\{\boldsymbol{u}(k)=[u_1(k),u_2(k),\cdots,u_m(k)]^{\mathrm{T}}\in\mathbf{R}^m:|u_i(k)|\le\bar{u}_i, i=1,\cdots,m\}$, 这里 $\bar{u}_i$ 代表第 i 个执行器的饱和上界. 令 $\bar{U}\in\mathbf{R}^{m\times m}$ 为由各个执行器饱和上界组成的常对角矩阵, 即 $\bar{U}=\mathrm{diag}\{\bar{u}_1,\bar{u}_2,\cdots,\bar{u}_m\}$.

本文主要讨论如何设计系统 (1) 的最优镇定控制器问题, 因此, 我们的目标是寻找控制输入 $\boldsymbol{u}(\cdot)$ 来最小化如下的一般形式的非二次泛函

$$J(\boldsymbol{x}(k),\boldsymbol{u})=\sum_{i=k}^{\infty}\left\{\boldsymbol{x}(i)^{\mathrm{T}}Q\boldsymbol{x}(i)+W(\boldsymbol{u}(i))\right\} \tag{2}$$

其中, $W(\boldsymbol{u}(i))$ 和 Q 都是正定的.

对于最优镇定控制问题, 控制输入 $\boldsymbol{u}(\cdot)$ 不但要在 Ω_x 上镇定系统, 而且要保证性能指标 (2) 是有限值, 即容许控制[9].

定义 1. 容许控制: 如果控制输入 $\boldsymbol{u}(\boldsymbol{x})$ 在 Ω_x 上镇定系统 (1), $\boldsymbol{u}(0)=\mathbf{0}$, 而且对于 $\forall\boldsymbol{x}(0)\in\Omega_x$, 性能指标值 $J(\boldsymbol{x}(0),\boldsymbol{u})$ 是有限值, 则控制输入 $\boldsymbol{u}(\boldsymbol{x})$ 被称为 Ω_x 上相对于性能指标 (2) 的容许控制.

根据贝尔曼最优性原理, 假设最优代价函数用 V^* 表示, 则可以得到

$$V^*(\boldsymbol{x}(k))=\min_{\boldsymbol{u}(k)}\Big\{\boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+W(\boldsymbol{u}(k))+V^*(\boldsymbol{x}(k+1))\Big\} \tag{3}$$

对于无约束控制问题, $W(\boldsymbol{u}(i))$ 通常选择为二次型形式 $W(\boldsymbol{u}(i))=\boldsymbol{u}(i)^{\mathrm{T}}R\boldsymbol{u}(i)$, 其中 $R\in\mathbf{R}^{m\times m}$ 是半正定的. 假设值函数光滑, 则根据最优控制的一阶必要条件, 可以得到

$$\boldsymbol{u}^*(k)=-\frac{1}{2}R^{-1}g^{\mathrm{T}}(\boldsymbol{x}(k))\frac{\partial V^*(\boldsymbol{x}(k+1))}{\partial\boldsymbol{x}(k+1)} \tag{4}$$

其中, V^* 是对应于最优控制策略 $\boldsymbol{u}^*$ 的值函数.

然而, 对于约束控制问题, 上面的推导不可行. 为了解决这个有界控制问题, 受文献 [20] 的启发, 首先引入如下的非二次泛函

$$W(\boldsymbol{u}(i))=2\int_0^{\boldsymbol{u}(i)}\boldsymbol{\varphi}^{-\mathrm{T}}(\bar{U}^{-1}\boldsymbol{s})\bar{U}R\mathrm{d}\boldsymbol{s} \tag{5}$$

$$\boldsymbol{\varphi}^{-1}(\boldsymbol{u}(i))=\left[\varphi^{-1}(u_1(i)),\cdots,\varphi^{-1}(u_m(i))\right]$$

其中, R 是正定的, $\boldsymbol{s}\in\mathbf{R}^m$, $\boldsymbol{\varphi}\in\mathbf{R}^m$, $\varphi(\cdot)$ 为一个属于 $C^p(p\ge1)$ 和 $L_2(\Omega_x)$ 的一对一的有界函数, 并且满足 $|\varphi(\cdot)|\le1$. 另外, 它是一个单调递增的奇函数, 并且其一阶导数具有常数上界 M. 满足这些条件的函数很容易找到, 例如双曲正切函数 $\varphi(\cdot)=\tanh(\cdot)$. 应该指出的是, 通过上面的定义, 可以保证非二次泛函 $W(\boldsymbol{u}(i))$ 是正定的, 因为 $\boldsymbol{\varphi}^{-1}(\cdot)$ 为单调奇函数, 而且 R 是正定矩阵.

把式 (5) 代入式 (3), 可以得到 HJB 方程为

$$V^*(\boldsymbol{x}(k))=\min_{\boldsymbol{u}(k)}\Big\{\boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+2\int_0^{\boldsymbol{u}(k)}\boldsymbol{\varphi}^{-\mathrm{T}}(\bar{U}^{-1}\boldsymbol{s})\bar{U}R\mathrm{d}\boldsymbol{s}+V^*(\boldsymbol{x}(k+1))\Big\} \tag{6}$$

若假设值函数光滑, 则根据最优控制的一阶必要条件, 可以得到

$$\frac{\partial V^*(\boldsymbol{x}(k))}{\partial\boldsymbol{u}(k)}=2\bar{U}R\boldsymbol{\varphi}^{-1}(\bar{U}^{-1}\boldsymbol{u}(k))+\Big(\frac{\partial\boldsymbol{x}(k+1)}{\partial\boldsymbol{u}(k)}\Big)^{\mathrm{T}}\frac{\partial V^*(\boldsymbol{x}(k+1))}{\partial\boldsymbol{x}(k+1)}=0 \tag{7}$$

因此, 最优控制可以求解为

$$\boldsymbol{u}^*(k)=\bar{U}\boldsymbol{\varphi}\left(-\frac{1}{2}(\bar{U}R)^{-1}g^{\mathrm{T}}(\boldsymbol{x}(k))V_x^*(\boldsymbol{x}(k+1))\right) \tag{8}$$

如果值函数 V^* 已知, 则最优控制 $\boldsymbol{u}^*(k)$ 可以通过式 (8) 直接计算得到. 然而, 当前还没有方法能够解析地求得这个约束最优控制问题的值函数. 因此, 在下一节我们着重讨论如何利用贪婪迭代的 DHP 算法来求取近似最优的控制解.

2 贪婪迭代 DHP 算法的推导和实现

2.1 贪婪迭代 DHP 算法的推导

为了便于分析, 以下用 $W(\boldsymbol{u}(k))$ 表示非二次泛函 $2\int_0^{\boldsymbol{u}(k)}\boldsymbol{\varphi}^{-\mathrm{T}}(\bar{U}^{-1}\boldsymbol{s})\bar{U}R\mathrm{d}\boldsymbol{s}$.

定义协状态 $\boldsymbol{\lambda}(\boldsymbol{x}(k))=\dfrac{\partial V(\boldsymbol{x}(k))}{\partial \boldsymbol{x}(k)}$, 则式 (8) 可以写为

$$\boldsymbol{u}^*(k)=\bar{U}\boldsymbol{\varphi}\Big(-\frac{1}{2}(\bar{U}R)^{-1}g^{\mathrm{T}}(\boldsymbol{x}(k))\boldsymbol{\lambda}^*(\boldsymbol{x}(k+1))\Big) \tag{9}$$

结合式 (7), 则协状态函数 $\boldsymbol{\lambda}^*$ 满足下式

$$\begin{aligned}
&\boldsymbol{\lambda}^*(\boldsymbol{x}(k))=\frac{\partial V^*(\boldsymbol{x}(k))}{\partial \boldsymbol{x}(k)}=\\
&\frac{\partial(\boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+W(\boldsymbol{u}(k)))}{\partial \boldsymbol{x}(k)}+\\
&\left(\frac{\partial \boldsymbol{u}(\boldsymbol{x}(k))}{\partial \boldsymbol{x}(k)}\right)^{\mathrm{T}}\frac{\partial(\boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+W(\boldsymbol{u}(k)))}{\partial \boldsymbol{u}(\boldsymbol{x}(k))}+\\
&\left(\frac{\partial \boldsymbol{x}(k+1)}{\partial \boldsymbol{x}(k)}\right)^{\mathrm{T}}\frac{\partial V^*(\boldsymbol{x}(k+1))}{\partial \boldsymbol{x}(k+1)}+\\
&\left(\frac{\partial \boldsymbol{u}(\boldsymbol{x}(k))}{\partial \boldsymbol{x}(k)}\right)^{\mathrm{T}}\left(\frac{\partial \boldsymbol{x}(k+1)}{\partial \boldsymbol{u}(\boldsymbol{x}(k))}\right)^{\mathrm{T}}\frac{\partial V^*(\boldsymbol{x}(k+1))}{\partial \boldsymbol{x}(k+1)}=\\
&\frac{\partial(\boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+W(\boldsymbol{u}(k)))}{\partial \boldsymbol{x}(k)}+\\
&\left(\frac{\partial \boldsymbol{u}(\boldsymbol{x}(k))}{\partial \boldsymbol{x}(k)}\right)^{\mathrm{T}}\left[\frac{\partial(\boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+W(\boldsymbol{u}(k)))}{\partial u(\boldsymbol{x}(k))}+\right.\\
&\left.\left(\frac{\partial \boldsymbol{x}(k+1)}{\partial \boldsymbol{u}(\boldsymbol{x}(k))}\right)^{\mathrm{T}}\frac{\partial V^*(\boldsymbol{x}(k+1))}{\partial \boldsymbol{x}(k+1)}\right]+\\
&\left(\frac{\partial \boldsymbol{x}(k+1)}{\partial \boldsymbol{x}(k)}\right)^{\mathrm{T}}\frac{\partial V^*(\boldsymbol{x}(k+1))}{\partial \boldsymbol{x}(k+1)}=\\
&2Q\boldsymbol{x}(k)+\left(\frac{\partial \boldsymbol{x}(k+1)}{\partial \boldsymbol{x}(k)}\right)^{\mathrm{T}}\boldsymbol{\lambda}^*(\boldsymbol{x}(k+1))
\end{aligned} \tag{10}$$

如果协状态函数 $\boldsymbol{\lambda}^*$ 能够从式 (10) 中解得, 则最优控制 $\boldsymbol{u}^*(k)$ 可以通过式 (9) 直接计算得到. 然而, 由于偏差分方程的两点边值问题, 很难解析地求解式 (10). 因此, 下面我们提出一种贪婪迭代 DHP 算法来计算协状态函数 $\boldsymbol{\lambda}^*$ 和最优的控制策略 $\boldsymbol{u}^*$.

首先设初始的代价函数 $V_0(\cdot)=0$, 初始的协状态函数 $\boldsymbol{\lambda}_0(\cdot)=0$, 然后计算相应的单步最优控制 $\boldsymbol{u}_0$ 如下

$$\begin{aligned}
\boldsymbol{u}_0(\boldsymbol{x}(k))=\arg\min_{\boldsymbol{u}(k)}\big\{&\boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+W(\boldsymbol{u}(k))+\\
&V_0(\boldsymbol{x}(k+1))\big\}
\end{aligned} \tag{11}$$

根据最优性条件, 可以通过式 (11) 的右半部分对 $\boldsymbol{u}(k)$ 的导数等于零来求解 $\boldsymbol{u}_0(x)$, 即

$$\begin{aligned}
&\boldsymbol{u}_0(\boldsymbol{x}(k))=\\
&\bar{U}\boldsymbol{\varphi}\Big(-\frac{1}{2}(\bar{U}R)^{-1}g^{\mathrm{T}}(\boldsymbol{x}(k))\boldsymbol{\lambda}_0(\boldsymbol{x}(k+1))\Big)
\end{aligned} \tag{12}$$

因此可以得到

$$\boldsymbol{x}(k+1)=\boldsymbol{f}(\boldsymbol{x}(k))+g(\boldsymbol{x}(k))u_0(\boldsymbol{x}(k)) \tag{13}$$

和

$$\begin{aligned}
\boldsymbol{u}_0(\boldsymbol{x}(k+1))=\arg\min_{\boldsymbol{u}(k+1)}\big\{&\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+\\
&W(\boldsymbol{u}(k+1))+V_0(\boldsymbol{x}(k+2))\big\}
\end{aligned} \tag{14}$$

根据 $\boldsymbol{\lambda}_0(\boldsymbol{x}(k+2))=\dfrac{\partial V_0(\boldsymbol{x}(k+2))}{\partial \boldsymbol{x}(k+2)}$, 进一步求解式 (14) 可以得到

$$\begin{aligned}
\boldsymbol{u}_0(\boldsymbol{x}(k+1))=&\bar{U}\boldsymbol{\varphi}\Big(-\frac{1}{2}(\bar{U}R)^{-1}g^{\mathrm{T}}(\boldsymbol{x}(k+1))\times\\
&\boldsymbol{\lambda}_0(\boldsymbol{x}(k+2))\Big)
\end{aligned} \tag{15}$$

然后可以更新代价函数为

$$\begin{aligned}
V_1(\boldsymbol{x}(k+1))=&\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+\\
&W(\boldsymbol{u}_0(k+1))+V_0(\boldsymbol{x}(k+2))
\end{aligned} \tag{16}$$

因此, 对于 $i=0,1,\cdots$, 我们可以在

$$\begin{aligned}
\boldsymbol{u}_i(\boldsymbol{x}(k+1))=\arg\min_{\boldsymbol{u}(k+1)}\big\{&\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+\\
&W(\boldsymbol{u}(k+1))+V_i(\boldsymbol{x}(k+2))\big\}
\end{aligned} \tag{17}$$

和

$$\begin{aligned}
&V_{i+1}(\boldsymbol{x}(k+1))=\min_{\boldsymbol{u}(k+1)}\big\{\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+\\
&W(\boldsymbol{u}(k+1))+V_i(\boldsymbol{x}(k+2))\big\}=\\
&\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+\\
&W(\boldsymbol{u}_i(k+1))+V_i(\boldsymbol{x}(k+2))
\end{aligned} \tag{18}$$

之间进行反复迭代直到代价函数序列 $V_i(\boldsymbol{x})$ 收敛.

显然, 若假设 V_i 光滑, 则式 (17) 中的最优控制 $\boldsymbol{u}_i(k+1)$ 能够进一步计算如下

$$\begin{aligned}&\frac{\partial V_{i+1}(\boldsymbol{x}(k+1))}{\partial \boldsymbol{u}(k+1)}=\\&\frac{\partial(\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+W(\boldsymbol{u}_i(k+1))}{\partial \boldsymbol{u}(k+1)}+\\&g^{\mathrm{T}}(\boldsymbol{x}(k+1))\frac{\partial V_i(\boldsymbol{x}(k+2))}{\partial \boldsymbol{x}(k+2)}=0\end{aligned}\tag{19}$$

即

$$\begin{aligned}\boldsymbol{u}_i(k+1)=&\bar{U}\boldsymbol{\varphi}\Big(-\frac{1}{2}(\bar{U}R)^{-1}g^{\mathrm{T}}(\boldsymbol{x}(k+1))\times\\&\boldsymbol{\lambda}_i(\boldsymbol{x}(k+2))\Big)\end{aligned}\tag{20}$$

因为 $\boldsymbol{\lambda}_{i+1}(\boldsymbol{x}(k+1))=\dfrac{\partial V_{i+1}(\boldsymbol{x}(k+1))}{\partial \boldsymbol{x}(k+1)}$, 所以类似式 (10) 的推导可以得到

$$\begin{aligned}&\boldsymbol{\lambda}_{i+1}(\boldsymbol{x}(k+1))=\\&\frac{\partial(\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+W(\boldsymbol{u}_i(k+1)))}{\partial \boldsymbol{x}(k+1)}+\\&\left(\frac{\partial \boldsymbol{u}_i(\boldsymbol{x}(k+1))}{\partial \boldsymbol{x}(k+1)}\right)^{\mathrm{T}}\times\\&\frac{\partial(\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+W(\boldsymbol{u}_i(k+1)))}{\partial \boldsymbol{u}_i(\boldsymbol{x}(k+1))}+\\&\left(\frac{\partial \boldsymbol{x}(k+2)}{\partial \boldsymbol{x}(k+1)}\right)^{\mathrm{T}}\frac{\partial V_i(\boldsymbol{x}(k+2))}{\partial \boldsymbol{x}(k+1)}+\\&\left(\frac{\partial \boldsymbol{u}_i(\boldsymbol{x}(k+1))}{\partial \boldsymbol{x}(k+1)}\right)^{\mathrm{T}}\left(\frac{\partial \boldsymbol{x}(k+2)}{\partial \boldsymbol{u}_i(\boldsymbol{x}(k+1))}\right)^{\mathrm{T}}\times\\&\frac{\partial V_i(\boldsymbol{x}(k+2))}{\partial \boldsymbol{x}(k+2)}=\\&\frac{\partial(\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+W(\boldsymbol{u}_i(k+1)))}{\partial \boldsymbol{x}(k+1)}+\\&\left(\frac{\partial \boldsymbol{x}(k+2)}{\partial \boldsymbol{x}(k+1)}\right)^{\mathrm{T}}\frac{\partial V_i(\boldsymbol{x}(k+2))}{\partial \boldsymbol{x}(k+2)}\end{aligned}\tag{21}$$

即

$$\begin{aligned}\boldsymbol{\lambda}_{i+1}(\boldsymbol{x}(k+1))=&2Q\boldsymbol{x}(k+1)+\\&\left(\frac{\partial \boldsymbol{x}(k+2)}{\partial \boldsymbol{x}(k+1)}\right)^{\mathrm{T}}\boldsymbol{\lambda}_i(\boldsymbol{x}(k+2))\end{aligned}\tag{22}$$

因此最优控制可以计算如下:

$$\begin{aligned}&\boldsymbol{u}_i(\boldsymbol{x}(k))=\\&\bar{U}\boldsymbol{\varphi}\left(-\frac{1}{2}(\bar{U}R)^{-1}g^{\mathrm{T}}(\boldsymbol{x}(k))\boldsymbol{\lambda}_i(\boldsymbol{x}(k+1))\right)\end{aligned}\tag{23}$$

因此, 式 (17) 和 (18) 之间的迭代可以直接通过式 (22) 和式 (23) 之间的迭代实现.

2.2　贪婪迭代 DHP 算法的收敛性分析

本节给出式 (17) 和 (18) 之间的迭代过程的收敛性证明. 首先给出两个引理.

引理 1. 令 $\boldsymbol{\mu}_i$ 为任意的控制策略序列, $\boldsymbol{u}_i$ 为式 (17) 定义的策略. 同时令 V_i 为式 (18) 定义的序列, P_i 定义为

$$\begin{aligned}&P_{i+1}(\boldsymbol{x}(k+1))=\\&\quad\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+\\&\quad W(\boldsymbol{\mu}_i(k+1))+P_i(\boldsymbol{x}(k+2))\end{aligned}\tag{24}$$

如果 $V_0=P_0=0$, 那么 $V_i\le P_i,\ \forall i$ 成立.

证明. V_{i+1} 是对于控制输入 $\boldsymbol{u}$ 最小化方程 (18) 的右边得到的代价函数, 而 P_{i+1} 是任意输入得到的代价函数, 所以 V_{i+1} 很明显的比 P_{i+1} 小. □

引理 2. 令序列 $\{V_i\}$ 如式 (18) 定义. 如果系统是可控的, 那么存在一个上界 Y 满足 $0\le V_i\le Y, \forall i$.

证明. 令 $\boldsymbol{\eta}(\boldsymbol{x}(k+1))$ 为任意镇定容许的控制输入, $V_0(\cdot)=Z_0(\cdot)=0$, 其中, V_i 通过式 (18) 更新, Z_i 通过下式更新

$$\begin{aligned}&Z_{i+1}(\boldsymbol{x}(k+1))=\\&\quad\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+\\&\quad W(\boldsymbol{\eta}(\boldsymbol{x}(k+1)))+Z_i(\boldsymbol{x}(k+2))\end{aligned}\tag{25}$$

那么可以得到

$$\begin{aligned}&Z_{i+1}(\boldsymbol{x}(k+1))-Z_i(\boldsymbol{x}(k+1))=\\&\quad Z_i(\boldsymbol{x}(k+2))-Z_{i-1}(\boldsymbol{x}(k+2))=\\&\quad Z_{i-1}(\boldsymbol{x}(k+3))-Z_{i-2}(\boldsymbol{x}(k+3))=\\&\quad Z_{i-2}(\boldsymbol{x}(k+4))-Z_{i-3}(\boldsymbol{x}(k+4))=\\&\qquad\qquad\vdots\\&\quad Z_1(\boldsymbol{x}(k+i+1))-Z_0(\boldsymbol{x}(k+i+1))\end{aligned}\tag{26}$$

因此可以得到下面的关系

$$\begin{aligned}Z_{i+1}(\boldsymbol{x}(k+1))=&Z_1(\boldsymbol{x}(k+i+1))+\\&Z_i(\boldsymbol{x}(k+1))-Z_0(\boldsymbol{x}(k+i+1))\end{aligned}\tag{27}$$

因为 $Z_0(\cdot)=0$, 所以有

$$\begin{aligned}&Z_{i+1}(\boldsymbol{x}(k+1))=\\&\quad Z_1(\boldsymbol{x}(k+i+1))+Z_i(\boldsymbol{x}(k+1))=\\&\quad Z_1(\boldsymbol{x}(k+i+1))+Z_1(\boldsymbol{x}(k+i))+\\&\quad Z_{i-1}(\boldsymbol{x}(k+1))=\\&\quad Z_1(\boldsymbol{x}(k+i+1))+Z_1(\boldsymbol{x}(k+i))+\\&\quad Z_1(\boldsymbol{x}(k+i-1))+\cdots+\\&\quad Z_1(\boldsymbol{x}(k+1))\end{aligned}\tag{28}$$

从而式 (28) 可以重写为

$$\begin{aligned}&Z_{i+1}(\boldsymbol{x}(k+1))=\sum_{j=0}^{i}Z_1(\boldsymbol{x}(k+j+1))=\\&\quad\sum_{j=0}^{i}\big\{\boldsymbol{x}(k+j+1)^{\mathrm{T}}Q\boldsymbol{x}(k+j+1)+\\&\quad W(\boldsymbol{\eta}(\boldsymbol{x}(k+j+1)))\big\}\le\\&\quad\sum_{j=0}^{\infty}\big\{\boldsymbol{x}(k+j+1)^{\mathrm{T}}Q\boldsymbol{x}(k+j+1)+\\&\quad W(\boldsymbol{\eta}(\boldsymbol{x}(k+j+1)))\big\}\end{aligned}\tag{29}$$

因为 $\boldsymbol{\eta}(\boldsymbol{x}(k+1))$ 是镇定容许的控制输入, 即当 $k\to\infty$ 时 $\boldsymbol{x}(k+1)\to\boldsymbol{0}$, 所以有

$$\forall i:\ \ Z_{i+1}(\boldsymbol{x}(k+1))\le\sum_{i=0}^{\infty}Z_1(\boldsymbol{x}(k+i+1))\le Y\tag{30}$$

结合引理 1, 可以得到

$$\forall i:\ \ V_{i+1}(\boldsymbol{x}(k+1))\le Z_{i+1}(\boldsymbol{x}(k+1))\le Y\tag{31}$$

□

引理 1 和 2 将用于主要定理的证明中.

定理 1. 定义序列 $\{V_i\}$ 如式 (18), $V_0=0$, 定义序列 $\{\boldsymbol{\lambda}_i\}$ 如式 (22). 那么 $\{V_i\}$ 为一个满足 $V_{i+1}(\boldsymbol{x}(k+1))\ge V_i(\boldsymbol{x}(k+1)),\forall i$ 的非减序列, 而且收敛于离散 HJB 方程的值函数, 即当 $i\Rightarrow\infty$ 时 $V_i\Rightarrow V^*$. 同时协状态序列 $\{\boldsymbol{\lambda}_i\}$ 和控制序列 $\{\boldsymbol{u}_i\}$ 也是收敛的, 即当 $i\Rightarrow\infty$ 时, $\boldsymbol{\lambda}_i\Rightarrow\boldsymbol{\lambda}^*$ 和 $\boldsymbol{u}_i\Rightarrow\boldsymbol{u}^*$.

证明. 为了便于分析, 定义一个新序列如下:

$$\begin{aligned}\Phi_{i+1}(\boldsymbol{x}(k+1))=\ &\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+\\&W(\boldsymbol{u}_{i+1}(k+1))+\Phi_i(\boldsymbol{x}(k+2))\end{aligned}\tag{32}$$

其中, $\Phi_0=V_0=0$, 控制策略 $\boldsymbol{u}_i$ 由式 (17) 定义, 代价函数 V_i 由式 (18) 给出.

在下面的部分, 我们通过数学归纳法证明 $\Phi_i(\boldsymbol{x}(k+1))\le V_{i+1}(\boldsymbol{x}(k+1))$.

首先, 证明当 $i=0$ 时结论成立. 因为

$$\begin{aligned}&V_1(\boldsymbol{x}(k+1))-\Phi_0(\boldsymbol{x}(k+1))=\\&\quad\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+W(\boldsymbol{u}_0(k+1))\ge0\end{aligned}\tag{33}$$

因此对于 $i=0$, 有

$$V_1(\boldsymbol{x}(k+1))\ge\Phi_0(\boldsymbol{x}(k+1))\tag{34}$$

其次, 假设结论对于 $i-1$ 成立, 即 $V_i(\boldsymbol{x}(k+1))\ge\Phi_{i-1}(\boldsymbol{x}(k+1))$, $\forall\boldsymbol{x}(k+1)$. 那么对于 i, 因为

$$\begin{aligned}\Phi_i(\boldsymbol{x}(k+1))=\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+\\W(\boldsymbol{u}_i(k+1))+\Phi_{i-1}(\boldsymbol{x}(k+2))\end{aligned}\tag{35}$$

和

$$\begin{aligned}V_{i+1}(\boldsymbol{x}(k+1))=\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+\\W(\boldsymbol{u}_i(k+1))+V_i(\boldsymbol{x}(k+2))\end{aligned}\tag{36}$$

成立, 因此有

$$\begin{aligned}&V_{i+1}(\boldsymbol{x}(k+1))-\Phi_i(\boldsymbol{x}(k+1))=\\&\quad V_i(\boldsymbol{x}(k+2))-\Phi_{i-1}(\boldsymbol{x}(k+2))\ge0\end{aligned}\tag{37}$$

即下式成立

$$\Phi_i(\boldsymbol{x}(k+1))\le V_{i+1}(\boldsymbol{x}(k+1))\tag{38}$$

到此, 数学归纳法证明结束.

另外, 由引理 1 有 $V_i(\boldsymbol{x}(k+1))\le\Phi_i(\boldsymbol{x}(k+1))$, 因此可以得到

$$V_i(\boldsymbol{x}(k+1))\le\Phi_i(\boldsymbol{x}(k+1))\le V_{i+1}(\boldsymbol{x}(k+1))\tag{39}$$

因此, 代价函数序列 $\{V_i\}$ 是一个满足 $V_{i+1}(\boldsymbol{x}(k+1))\ge V_i(\boldsymbol{x}(k+1)),\forall i$ 的非减序列, 而且最终收敛于离散 HJB 方程的值函数, 即当 $i\Rightarrow\infty$ 时 $V_i\Rightarrow V^*$, 同时协状态函数序列 $\{\boldsymbol{\lambda}_i\}$ 也是收敛的, 即当 $i\Rightarrow\infty$ 时 $\boldsymbol{\lambda}_i\Rightarrow\boldsymbol{\lambda}^*$.

因为代价函数序列和协状态函数序列收敛, 根据式 (9) 和 (20), 当 $i\Rightarrow\infty$, 控制序列 $\{\boldsymbol{u}_i\}$ 收敛于最优的控制策略 $\boldsymbol{u}^*$. □

2.3 单网络贪婪迭代 DHP 算法的神经网络实现

对于线性系统, 如果性能指标是二次型的, 则最优控制策略是线性的. 而对于非线性系统, 这一点并不一定成立. 因此需要引入一个参数结构如神经网络等来近似协状态函数 $\boldsymbol{\lambda}_i(\boldsymbol{x}(k))$, 进而求得最优的控制策略.

在本节中, 我们利用单神经网络 GI-DHP 算法来求解最优的协状态函数及最优控制策略. 单神经网络自适应评价技术[12] 可以消除常规自适应评价系统中的控制网络的使用. 这种做法使得整个系统

的结构得到了简化, 从而节省了大量的存储空间, 有效地减少了计算量. 除此之外, 它还消除了控制网络所可能带来的神经网络近似误差, 因而提高了计算精度.

从式 (23) 可以看出, 控制策略是该方程的一个隐式解, 其求解仍然是个问题. 因此, 首先要对协状态函数进一步进行变换. 在上一节, 我们定义 $\boldsymbol{\lambda}(\boldsymbol{x}(k+1))=\dfrac{\partial V(\boldsymbol{x}(k+1))}{\partial \boldsymbol{x}(k+1)}$, 它表明 $\boldsymbol{\lambda}(\boldsymbol{x}(k+1))$ 是 $\boldsymbol{x}(k+1)$ 的函数. 实际上, 根据式 (1), $\boldsymbol{x}(k+1)$ 可以表示为 $\boldsymbol{x}(k)$ 和 $\boldsymbol{u}(k)$ 的函数. 同时, 由于 $\boldsymbol{u}(k)$ 是时刻 k 的最优控制, 它具有式 (9) 所示的形式, 即它也是 $\boldsymbol{x}(k)$ 的函数. 因此, $\boldsymbol{\lambda}(\boldsymbol{x}(k+1))$ 可以用 $\boldsymbol{x}(k)$ 唯一表示. 令 $\bar{\boldsymbol{\lambda}}(\boldsymbol{x}(k))=\boldsymbol{\lambda}(\boldsymbol{x}(k+1))$, 则评价网络被用来近似逼近函数 $\bar{\boldsymbol{\lambda}}(\boldsymbol{x}(k))$.

因此, 式 (22) 变换为

$$\bar{\boldsymbol{\lambda}}_{i+1}(\boldsymbol{x}(k))=2Q\boldsymbol{x}(k+1)+\left(\frac{\partial \boldsymbol{x}(k+2)}{\partial \boldsymbol{x}(k+1)}\right)^{\mathrm{T}}\bar{\boldsymbol{\lambda}}_i(\boldsymbol{x}(k+1)) \quad (40)$$

最优控制策略可以计算如下:

$$\boldsymbol{u}_i(\boldsymbol{x}(k))=\bar{U}\boldsymbol{\varphi}\left(-\frac{1}{2}(\bar{U}R)^{-1}g^{\mathrm{T}}(\boldsymbol{x}(k))\bar{\boldsymbol{\lambda}}_i(\boldsymbol{x}(k))\right) \quad (41)$$

一旦通过评价网络计算得到协状态函数 $\bar{\boldsymbol{\lambda}}_i(\boldsymbol{x}(k))$, 则根据式 (41) 可以直接获得最优控制策略. 因此, 令 $\bar{\boldsymbol{\lambda}}_0(\cdot)=\boldsymbol{0}$, 式 (22) 和 (23) 之间的迭代转换为 (40) 和 (41) 之间的迭代.

基于上面的分析, 可给出整个结构图如图 1 所示.

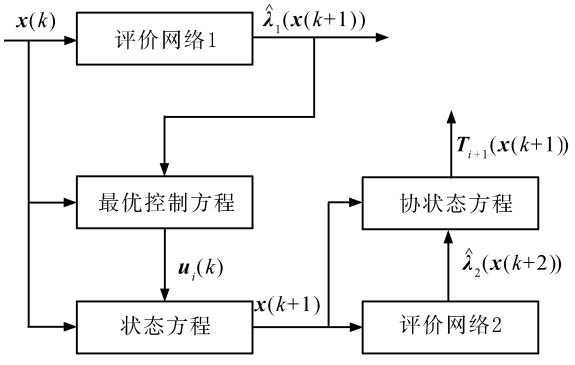


图 1　单网络贪婪迭代 DHP 算法的结构图

Fig. 1　The structure diagram of the single network GI-DHP algorithm

在每一个迭代步, 我们应用一个 3 层前向网络作为评价网络来近似逼近协状态函数 $\boldsymbol{\lambda}_i(\boldsymbol{x}(k+1))$ 如下:

$$\hat{\boldsymbol{\lambda}}_i(\boldsymbol{x}(k+1))=\hat{\bar{\boldsymbol{\lambda}}}_i(\boldsymbol{x}(k))=w_i^{\mathrm{T}}\boldsymbol{\sigma}(v_i^{\mathrm{T}}\boldsymbol{x}(k)) \quad (42)$$

其中, w_i 和 v_i 分别代表第 i 步时输出层和隐层的权矩阵, $\boldsymbol{\sigma}(\cdot)\in\mathbf{R}^l$, $[\boldsymbol{\sigma}(\boldsymbol{z})]_p=\dfrac{\mathrm{e}^{z_p}-\mathrm{e}^{-z_p}}{\mathrm{e}^{z_p}+\mathrm{e}^{-z_p}}, p=1,\cdots,l$ 是隐层的激活函数, l 是隐层节点数.

根据式 (22), 目标协状态函数可以表示为

$$\begin{aligned}\boldsymbol{T}_{i+1}(\boldsymbol{x}(k+1))&=2Q\boldsymbol{x}(k+1)+\left(\frac{\partial \boldsymbol{x}(k+2)}{\partial \boldsymbol{x}(k+1)}\right)^{\mathrm{T}}\times\\ &\hat{\boldsymbol{\lambda}}_i(\boldsymbol{x}(k+2))=2Q\boldsymbol{x}(k+1)+\\ &\left(\frac{\partial \boldsymbol{x}(k+2)}{\partial \boldsymbol{x}(k+1)}\right)^{\mathrm{T}}w_i^{\mathrm{T}}\boldsymbol{\sigma}(v_i^{\mathrm{T}}\boldsymbol{x}(k+1)) \quad (43)\end{aligned}$$

其中 $\boldsymbol{x}(k+1)$ 可由 $\boldsymbol{x}(k)$ 和 $\boldsymbol{u}_i(\boldsymbol{x}(k))$ 计算得到.

定义评价网的误差函数为

$$\boldsymbol{e}_j(k+1)=\hat{\boldsymbol{\lambda}}_{i(j)}(\boldsymbol{x}(k),w_{i(j)},v_{i(j)})-\boldsymbol{T}_{i+1}(\boldsymbol{x}(k+1)) \quad (44)$$

评价网络欲最小化的目标函数为

$$E_j(k+1)=\frac{1}{2}\boldsymbol{e}_j^{\mathrm{T}}(k+1)\boldsymbol{e}_j(k+1) \quad (45)$$

因此, 根据梯度下降法, 评价网的权值更新规则为

$$w_{i(j+1)}(k+1)=w_{i(j)}(k+1)-\alpha\left[\frac{\partial E_j(k+1)}{\partial w_{i(j)}(k+1)}\right] \quad (46)$$

$$v_{i(j+1)}(k+1)=v_{i(j)}(k+1)-\alpha\left[\frac{\partial E_j(k+1)}{\partial v_{i(j)}(k+1)}\right] \quad (47)$$

其中, $\alpha>0$ 为学习率, j 为权值更新的迭代次数.

在评价网络实现对协状态函数 $\boldsymbol{\lambda}_i(\boldsymbol{x}(k+1))$ (即函数 $\bar{\boldsymbol{\lambda}}_i(\boldsymbol{x}(k))$) 的近似之后, 我们可以直接通过式 (41) 获得最优的控制策略.

3　仿真例子

本节提供两个例子来说明本文提出的控制方案的有效性.

例 1. 考虑文献 [12] 中的 Van der Pol 振荡器:

$$\begin{aligned}\dot{x}_1&=x_2 \quad (48)\\ \dot{x}_2&=\alpha(1-x_1^2)x_2-x_1+(1+x_1^2+x_2^2)u\end{aligned}$$

其中, $\alpha=0.05$, 控制约束为 $|u|\le 0.2$.

首先利用欧拉法对系统进行离散化处理, 其中 $t=kT$, T 是采样周期, k 是采样步数. 如果采样周期 T 能够选择到相对于系统的时间常数而言充分小

的数, 那么由离散化方法获得的系统响应也是足够准确的[19].

因此, 令 $T = 0.1\,\mathrm{s}$, 可得到如下的离散系统:

$$\begin{aligned} x_1(k+1) =& x_1(k) + 0.1x_2(k) \\ x_2(k+1) =& -x_1(k) + 0.005(1 - x_1^2(k))x_2(k) + \\ & x_2(k) + (1 + x_1^2(k) + x_2^2(k))u(k) \end{aligned} \tag{49}$$

定义性能指标为

$$\begin{aligned} J(\boldsymbol{x}(k), u) = \sum_{i=k}^{\infty} \{ & \boldsymbol{x}(i)^{\mathrm{T}} Q \boldsymbol{x}(i) + \\ & 2\bar{U} \int_0^{\boldsymbol{u}(i)} \tanh^{-\mathrm{T}}(\bar{U}s) R \mathrm{d}s \} \end{aligned} \tag{50}$$

其中, $\bar{U} = 0.2$, 而权矩阵选择为 $Q = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$, $R = [\,2\,]$.

评价网络的结构选为 2-9-2, 包含 2 个输入神经元和 9 个隐层神经元. 输出层的初始权值全被设置为零以保证评价网络的初始输出为零, 即 $\boldsymbol{\lambda}_0 = \boldsymbol{0}$, 而隐层的初始权值在 $[-1, 1]$ 之间随机取值. 评价网络训练 1 200 个训练环, 每个训练环在学习率 $\alpha = 0.1$ 下运行 2 000 个时间步. 评价网络的收敛性判定的容限值选为 10^{-10}. 迭代完成之后得到协状态函数的收敛过程曲线如图 2 所示. 另外从初始状态 $x_1(0) = 0.1$, $x_2(0) = 0.3$ 开始, 我们应用得到的饱和最优控制策略于系统运行 200 个时间步, 得到系统的状态曲线如图 3 所示, 控制曲线如图 4 所示.

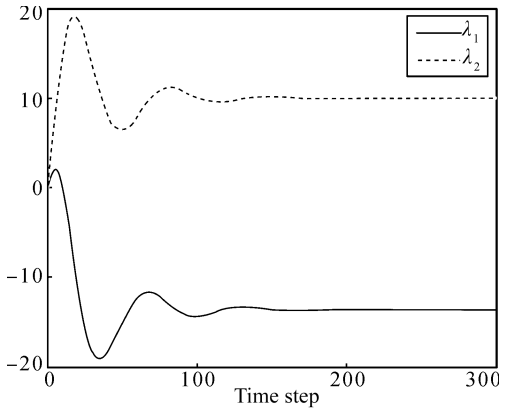


图 2 协状态函数的收敛过程曲线

Fig. 2 The convergence process of the costate function

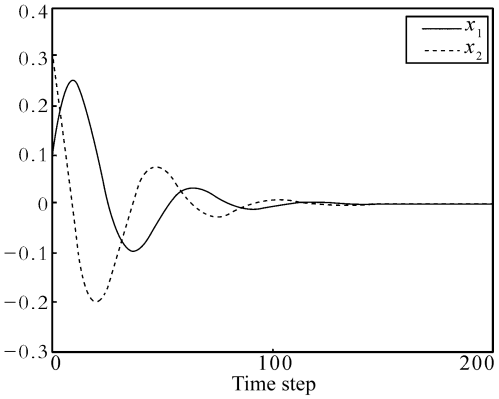


图 3 状态变量曲线

Fig. 3 The state variables curves

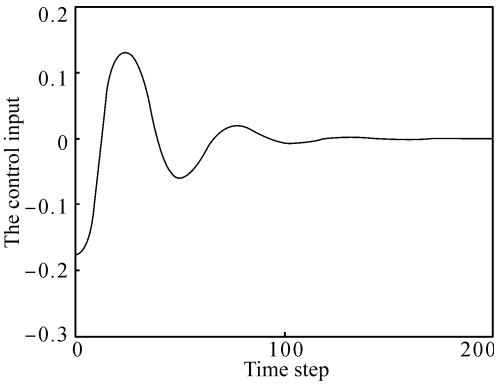


图 4 控制输入曲线

Fig. 4 The control input curve

另外, 为了与未考虑执行器饱和而设计的控制器进行对比, 我们把由相似算法设计但不考虑饱和约束得到的控制器应用于被控系统, 得到仿真结果如下: 状态曲线如图 5 所示, 控制曲线如图 6 所示.

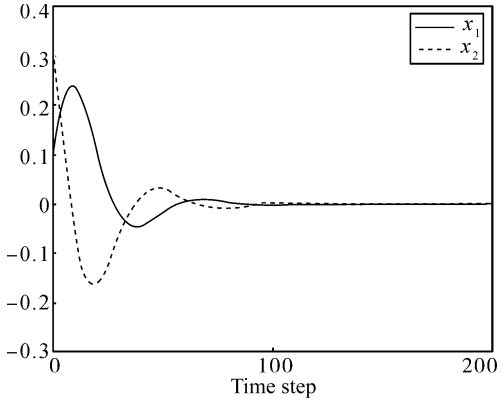


图 5 控制器设计时未考虑控制约束得到的状态变量曲线

Fig. 5 The state variables curves obtained by the controller designed without considering control constraints

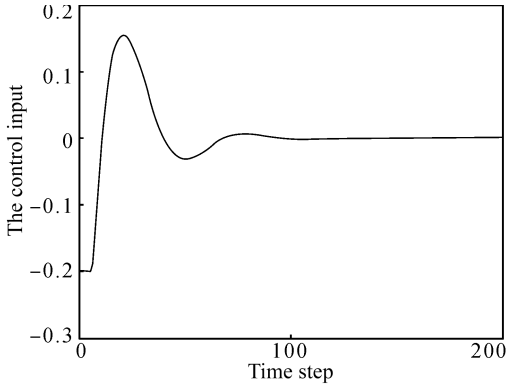


图 6　控制器设计时未考虑控制约束得到的控制输入曲线

Fig. 6　The control input curve obtained by the controller designed without considering control constraints

从图 4 和图 6 的对比中可以看出, 通过使用非二次泛函, 系统的控制约束问题得以解决, 获得了光滑的最优控制律. 而设计过程中未考虑饱和限制的控制器则出现了饱和现象, 无法获得系统的最优性能. 因此进一步说明本文提出的贪婪迭代 DHP 算法确实能够有效地解决控制带约束系统的最优控制问题.

例 2.　连续搅拌反应釜 (Continuous stirred tank reactor, CSTR) 系统

CSTR 系统里面进行的是一阶不可逆的放热反应过程. 根据质量守恒和能量守恒的原理, 原料的质量 C 和反应器的温度 T 的变化关系可由下列式子给出[21]:

$$\begin{aligned} V_{ol}\frac{\mathrm{d}C}{\mathrm{d}t} =& \zeta(C_0 - C) - V_{ol}R_a \\ V_{ol}C_p\frac{\mathrm{d}T}{\mathrm{d}t} =& \zeta(T_0 - T) + (\Delta H)V_{ol}R_a - \\ & U(T - T_w) \end{aligned} \tag{51}$$

其中, V_{ol} 表示反应器的体积, ζ 表示原料进出的流动速度, C_0 表示期望的原料供给量, R_a 表示单位体积的反应率, C_p 表示单位体积的原料所产生的热能, T_0 表示期望的温度, ΔH 表示反应的摩尔热, 如果是放热反应系统, 应取正值. U 表示反应器表面的热传导系数, T_w 表示冷却剂的平均温度.

假设反应率的一阶动力学方程为

$$R_a = \kappa_0 C \exp\left(\frac{-Q}{T}\right)$$

其中, κ_0 表示反应速度常数, Q 表示 Arrhenius 激活能量与气体常数的比值. 通过如下的变量代换:

$$x_1 = \frac{C_0 - C}{C_0},\ \ x_2 = \frac{T - T_0}{Q},$$

$$\nu = \frac{t}{\frac{V_{ol}}{\zeta}},\ \ u = \frac{T_w - T_0}{Q}$$

式 (51) 可以改写为

$$\begin{aligned} \frac{\mathrm{d}x_1}{\mathrm{d}\nu} =& -x_1 + D_a(1 - x_1)\exp\left(-\frac{1}{x_2 + \rho}\right) \\ \frac{\mathrm{d}x_2}{\mathrm{d}\nu} =& -(1 + \varrho)x_2 + HD_a(1 - x_1)\times \\ & \exp\left(-\frac{1}{x_2 + \rho}\right) + \varrho u \end{aligned} \tag{52}$$

其中

$$D_a = \frac{\kappa_0 V_{ol}}{\zeta},\ H = \frac{(\Delta H)C_0}{QC_p},\ \rho = \frac{T_0}{Q},\ \varrho = \frac{U}{\zeta C_p}$$

对于给定的一组参数: D_a, H, ρ, 和 ϱ, 式 (52) 可能有 1 个, 2 个, 或者 3 个平衡点. 这取决于对应的稳态输入 $u = u_e$ 的不同取值. 若选取参数为

$$D_a = 0.072,\ \ H = 8,\ \ \rho = 20,\ \ \varrho = 0.3$$

并设 $u_e = 0$, 可知系统 (52) 仅有一个平衡点, 即

$$\boldsymbol{x}_e = \begin{bmatrix} 0.0642 \\ 0.3948 \end{bmatrix}$$

下面定义新的状态变量:

$$\begin{aligned} \iota x_1 &= x_1 - x_{e1} \\ \iota x_2 &= x_2 - x_{e2} \end{aligned}$$

则式 (52) 的动态模型可化为:

$$\begin{aligned} \frac{\mathrm{d}\iota x_1}{\mathrm{d}\nu} =& -\iota x_1 - x_{e1} + D_a(1 - \iota x_1 - x_{e1})\times \\ & \exp\left(-\frac{1}{\iota x_2 + x_{e2} + \rho}\right) \\ \frac{\mathrm{d}\iota x_2}{\mathrm{d}\nu} =& -(1 + \varrho)(\iota x_2 + x_{e2}) + HD_a(1 - \iota x_1 - x_{e1})\times \\ & \exp\left(-\frac{1}{\iota x_2 + x_{e2} + \rho}\right) + \varrho u \end{aligned} \tag{53}$$

假设控制约束为 $|u| \le 0.05$. 首先类似于例 1 的做法, 令 $T = 0.01\,\mathrm{s}$, 利用欧拉法对系统 (53) 进行离散化后有

$$\begin{aligned} &\iota x_1(k+1) = -\iota x_1(k)T - x_{e1}T + D_a\times \\ &(1 - \iota x_1(k) - x_{e1})\exp\left(-\frac{1}{\iota x_2(k) + x_{e2} + \rho}\right)T + \\ &\iota x_1(k) \\ &\iota x_2(k+1) = -(1 + \varrho)(\iota x_2(k) + x_{e2})T + HD_a\times \end{aligned}$$

$$(1-\iota x_1(k)-x_{e1})\exp\left(-\frac{1}{\iota x_2(k)+x_{e2}+\rho}\right)T+\iota x_2(k)+\varrho Tu \tag{54}$$

假设性能指标的形式与例 1 相同, 其中 $\bar{U}=0.05$, 而权矩阵选择为 $Q=\begin{bmatrix}2&0\\0&2\end{bmatrix}$, $R=[\,0.5\,]$.

选择评价网络的结构为 2-11-2, 包含 2 个输入神经元和 11 个隐层神经元. 输出层的初始权值同样被设置为零, 隐层的初始权值在 $[-1,\ 1]$ 之间随机取值. 评价网络训练 1 600 个训练环, 每个训练环在学习率 $\alpha=0.1$ 下运行 2 000 个时间步. 其他条件设置与例 1 相同, 迭代完成之后得到协状态函数的收敛过程曲线如图 7 所示. 从初始状态 $\iota x_1(0)=1$, $\iota x_2(0)=-1$ 开始, 我们应用得到的饱和最优控制策略于系统运行 100 个时间步, 得到系统的状态曲线如图 8 所示, 控制曲线如图 9 所示.

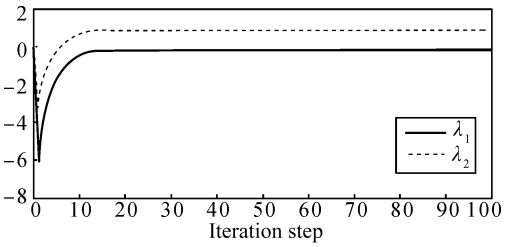


图 7 协状态函数的收敛过程曲线

Fig. 7 The convergence process of the costate function

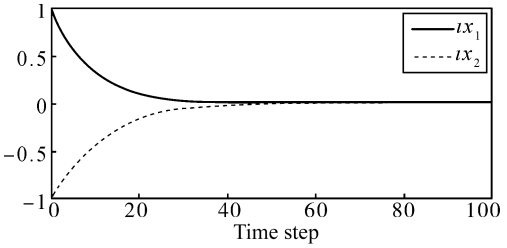


图 8 状态变量曲线

Fig. 8 The state variables curves

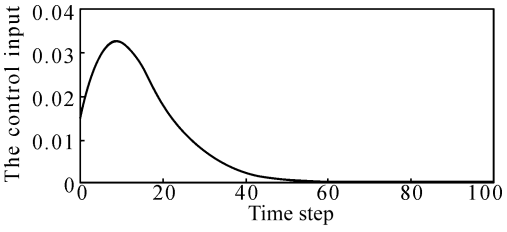


图 9 控制输入曲线

Fig. 9 The control input curve

同样为了与未考虑执行器饱和而设计的控制器进行对比, 我们把由相似算法设计但不考虑饱和约束得到的控制器应用于被控系统, 可得到仿真结果如下: 状态曲线如图 10 所示, 控制曲线如图 11 所示.

把图 9 和图 11 进行对比可以看出, 通过使用非二次泛函, 避免了执行器饱和现象的出现, 并且获得了光滑的最优控制律, 从而再次表明本文提出的贪婪迭代 DHP 算法的有效性.

4 结论

本文提出了一种贪婪迭代算法来寻求一类离散时间系统的最优控制策略. 首先引入一个非二次泛函来处理非线性系统的控制约束问题, 然后提出贪婪迭代的 DHP 算法来求解近似最优的协状态函数和最优控制策略, 同时还严格证明了这种迭代算法的收敛性. 为了便于实现, 引入了一个神经网络来近似协状态函数, 从而通过这个近似的协状态函数直接求取最优的控制策略. 仿真研究中通过两个例子表明了本文提出的针对离散约束非线性系统的最优控制方案的有效性.

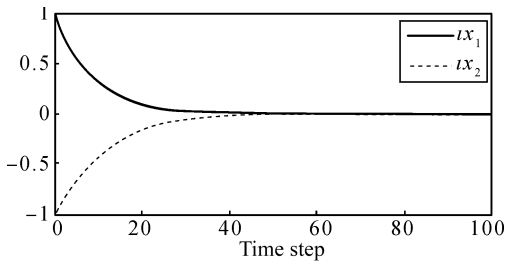


图 10 控制器设计时未考虑控制约束得到的状态变量曲线

Fig. 10 The state variables curves obtained by the controller designed without considering control constraints

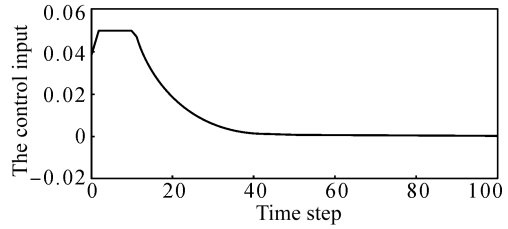


图 11 控制器设计时未考虑控制约束得到的控制输入曲线

Fig. 11 The control input curve obtained by the controller designed without considering control constraints

References

1 Widrow B, Gupta N K, Maitra S. Punish/reward: learning with a critic in adaptive threshold systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 1973, **3**(5): 455–465

2 Barto A G, Sutton R S, Anderson C W. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 1983, **13**(5): 835−846

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

4 Bertsekas D P, Tsitsiklis J N. *Neuro-Dynamic Programming*. MA: Athena Scientific, 1996

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

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

7 Liu D R, Xiong X X, Zhang Y. Action-dependent adaptive critic designs. In: Proceedings of the International Joint Conference on Neural Networks. Washington D. C., USA: IEEE, 2001. 990−995

8 Murray J J, Cox C J, Lendaris G G, Saeks R. Adaptive dynamic programming. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2002, **32**(2): 140−153

9 Abu-Khalaf M, Lewis F L. Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. *Automatica*, 2005, **41**(5): 779−791

10 Liu De-Rong. Approximate dynamic programming for self-learning control. *Acta Automatica Sinica*, 2005, **31**(1): 13−18

11 Liu D R, Zhang H G. A neural dynamic programming approach for learning control of failure avoidance problems. *International Journal of Intelligent Control and Systems*, 2005, **10**(1): 21−32

12 Padhi R, Unnikrishnan N, Wang X H, Balakrishnan S N. A single network adaptive critic (SNAC) architecture for optimal contrl synthesis for a class of nonliner systems. *Neural Networks*, 2006, **19**(10): 1648−1660

13 Zhang H G, Wei Q L, Liu D R. On-line learning control for discrete nonlinear systems via an improved ADDHP method. In: Proceedings of the 4th International Symposium on Neural Networks. Nanjing, China: Springer-Verlag, 2007. 387−396

14 Al-Tamimi A, Lewis F L. Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof. In: Proceedings of the IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning. Hawaii, USA: IEEE, 2007. 38−43

15 Cheng T, Lewis F L, Abu-Khalaf M. Fixed-final-time-constrained optimal control of nonlinear systems using neural network HJB approach. *IEEE Transactions on Neural Networks*, 2007, **18**(6): 1725−1736

16 Zhang H G,Wei Q L, Luo Y H. A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear system based on greedy HDP iteration algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2008, **38**(4): 937−942

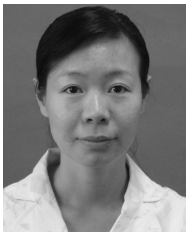
17 Baddeley B. Reinforcement learning in continuous time and space: interference and not ill conditioning is the main problem when using distributed function approximators. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2008, **38**(4): 950−956

18 Wang F Y, Zhang H G, Liu D R. Adaptive dynamic programming: an introduction. *IEEE Computational Intelligence Magazine*, 2009, **4**(2): 39−47

19 Chen Z, Jagannathan S. Generalized Hamilton-Jacobi-Bellman formulation-based neural network control of affine nonlinear discrete-time systems. *IEEE Transactions on Neural Networks*, 2008, **19**(1): 90−106

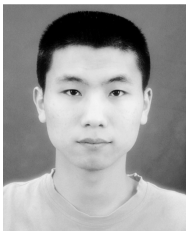
20 Lyshevski S E. Nonlinear discrete-time systems: constrained optimization and application of nonquadratic costs. In: Proceedings of the American Control Conference. Philadelphia, USA: IEEE, 1998. 3699−3703

21 Zhang H G, Yang J, Su C Y. T-S fuzzy-model-based robust H_∞ design for networked control systems with uncertainties. *IEEE Transactions on Industrial Informatics*, 2007, **3**(4): 289−301



罗艳红 东北大学信息科学与工程学院副教授. 主要研究方向为近似最优控制, 神经网络控制. 本文通信作者.
E-mail: neuluo@gmail.com
(**LUO Yan-Hong** Associate professor at the School of Information Science and Engineering, Northeastern University. Her research interest covers approximate optimal control and neural network control. Corresponding author of this paper.)

张化光 东北大学信息科学与工程学院教授. 主要研究方向为模糊控制, 网络控制及混沌控制. E-mail: z_hg@tom.com
(**ZHANG Hua-Guang** Professor at the School of Information Science and Engineering, Northeastern University. His research interest covers fuzzy control, network control, and chaos control.)



曹 宁 东北大学控制理论与控制工程专业硕士研究生. 主要研究方向为自适应动态规划, 最优控制.
E-mail: nii_c@163.com
(**CAO Ning** Master student at the School of Control Theory and Control Engineering, Northeastern University. His research interest covers adaptive dynamic programming and optimal control.)

陈 兵 青岛大学复杂性科学研究所教授. 主要研究方向为非线性系统鲁棒控制, 模糊控制.
E-mail: bing1958@eyou.com
(**CHEN Bing** Professor at the Institute of Complexity Science, Qingdao University. His research interest covers robust control, fuzzy control of nonlinear systems.)