工程科学学报
Chinese Journal of Engineering

一类离散动态系统基于事件的迭代神经控制

王鼎

Event-based iterative neural control for a type of discrete dynamic plant

WANG Ding

引用本文:
王鼎. 一类离散动态系统基于事件的迭代神经控制[J]. 工程科学学报, 2022, 44(3): 411–419. doi: 10.13374/j.issn2095–9389.2020.10.28.002
WANG Ding. Event–based iterative neural control for a type of discrete dynamic plant[J]. *Chinese Journal of Engineering*, 2022, 44(3): 411–419. doi: 10.13374/j.issn2095–9389.2020.10.28.002

在线阅读 View online: https://doi.org/10.13374/j.issn2095–9389.2020.10.28.002

工程科学学报, 第 44 卷, 第 3 期: 411−419, 2022 年 3 月
Chinese Journal of Engineering, Vol. 44, No. 3: 411−419, March 2022
https://doi.org/10.13374/j.issn2095-9389.2020.10.28.002; http://cje.ustb.edu.cn

一类离散动态系统基于事件的迭代神经控制

王　鼎[1,2,3,4)]✉

1) 北京工业大学信息学部, 北京 100124　2) 计算智能与智能系统北京市重点实验室, 北京 100124　3) 智慧环保北京实验室, 北京 100124
4) 北京人工智能研究院, 北京 100124
✉通信作者, E-mail: dingwang@bjut.edu.cn

摘　要　面向离散时间非线性动态系统, 提出一种基于事件的迭代神经控制框架. 主要目标是将迭代自适应评判方法与事件驱动机制结合起来, 以解决离散时间非线性系统的近似最优调节问题. 首先, 构造两个迭代序列并建立一种事件触发的值学习策略. 其次, 详细给出迭代算法的收敛性分析和新型框架的神经网络实现. 这里是在基于事件的迭代环境下实施启发式动态规划技术. 此外, 通过设计适当的阈值以确定事件驱动方法的触发条件. 最后, 借助两个仿真实例验证本文控制方案的优越性能, 尤其是在通信资源的利用方面. 本文的工作有助于构建一类事件驱动机制下的智能控制系统.

关键词　迭代自适应评判; 神经控制; 事件驱动设计; 智能控制; 非线性动态; 优化控制
分类号　TP13

Event-based iterative neural control for a type of discrete dynamic plant

WANG Ding[1,2,3,4)]✉

1) Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
2) Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China
3) Beijing Laboratory of Smart Environmental Protection, Beijing 100124, China
4) Beijing Institute of Artificial Intelligence, Beijing 100124, China
✉ Corresponding author, E-mail: dingwang@bjut.edu.cn

ABSTRACT　With the widespread popularity of network-based techniques and extension of computer control scales, more dynamical systems, particularly complex nonlinear dynamics, including increasing communication burdens, increasing difficulties in building accurate mathematical models, and different uncertain factors are encountered. Consequently, in contrast to the linear case, the optimization of the design of these uncertain complex systems is difficult to achieve. By combining reinforcement learning, neural networks, and dynamic programming, the adaptive critic method is regarded as an advanced approach to address intelligent control problems. The adaptive critic method has been currently used to solve the optimal regulation, trajectory tracking, robust control, disturbance attenuation, and zero-sum game problems. It has been considered a promising direction within the artificial intelligence field. However, many traditional design processes of the adaptive critic method are conducted based on the time-based mechanism, where the control signals are updated at each time step. Thus, the related control efficiencies are often low, which results in poor performance when considering practical updating times. Hence, more improvements are needed to enhance the control efficiency of adaptive-critic-based nonlinear control design. In this study, we developed an event-based iterative neural control framework for discrete-time nonlinear dynamics. The iterative adaptive critic method was combined with the event-driven mechanism to address the approximate optimal regulation problem in discrete-time nonlinear plants. An event-triggered value learning strategy was established with two iterative

收稿日期: 2020−10−28
基金项目: 北京市自然科学基金资助项目(JQ19013); 国家自然科学基金资助项目(61773373, 61890930-5, 62021003); 科技创新2030——“新一代人工智能”重大项目(2021ZD0112300-2); 国家重点研发计划资助项目(2018YFC1900800-5)

sequences. The convergence analysis of the iterative algorithm and the neural network implementation of the new framework were presented in detail. Therein, the heuristic dynamic programming technique was employed under the event-based iterative environment. Moreover, the triggering condition of the event-driven approach was determined with the appropriate threshold. Finally, simulation examples were provided to illustrate the excellent control performance, particularly in utilizing the communication resource. Thus, constructing a class of intelligent control systems based on the event-based mechanism will be helpful.

KEY WORDS iterative adaptive critic; neural control; event-based design; intelligent control; nonlinear dynamics; optimal control

在许多数值计算过程中, 神经网络都被视为一种能够用于参数学习和函数逼近的重要方法. 解决非线性最优反馈控制问题的关键在于如何求解复杂的 Hamilton-Jacobi-Bellman (HJB) 方程. 由于缺乏解析策略, 文献 [1] 构造了基于神经网络的自适应评判算法来获取满意的数值结果. 近年来, 基于自适应评判结构的控制系统设计受到很多关注, 在解决优化调节, 跟踪控制, 鲁棒镇定, 干扰抑制, 零和博弈等方面取得不少成果[2–11]. 当考虑实现过程时, 自适应评判有三种基本类型的技术, 包括启发式动态规划 (Heuristic dynamic programming, HDP), 二次启发式规划 (Dual HDP, DHP) 和全局二次启发式规划 (Globalized DHP, GDHP)[1]. 近年来, 离散时间情形下的迭代自适应评判结构已被分别用以处理包含 HDP[12], DHP[13] 和 GDHP[14] 结构的近似最优调节问题. 进而, 目标导向型迭代 HDP 设计的理论分析也在文献 [15] 中给出. 文献 [16] 提出一种用于离散时间未知非仿射非线性系统的在线学习最优控制方法, 并着重强调基于数据的自适应评判设计过程. 需要注意的是, 上述这些自适应评判算法是利用基于时间的更新方法来实现的, 所设计的控制器在每个时刻都进行更新, 存在着一定的资源浪费现象.

与经典的时间驱动机制相比, 基于事件的方法已经成为提高资源利用效率的先进工具. 它不仅能够用于传统的反馈镇定[17] 和容错控制[18], 而且已经在忆阻系统的脉冲控制中得到应用[19]. 针对传统时间驱动模式存在通信资源浪费的问题[20], 文献 [21] 讨论了事件驱动环境下的神经控制实现方法. 值得注意的是, 在基于事件的控制框架中, 一般根据指定的触发条件来更新控制信号. 文献 [22] 给出一种基于广义模糊双曲模型的非零和博弈事件触发设计. 另一方面, 基于文献 [23] 的工作, Dong 等[24] 针对非线性离散时间系统提出一种基于事件的 HDP 算法. 文献 [25] 则针对约束非线性系统基于事件的最优控制设计进行了扩展研究. 文献 [26] 设计一种实时事件驱动自适应评判控制器, 并将其应用于实际的电力系统中. 然而, 关于离散动态系统, 目前基于事件的迭代自适应评判控制的研究成果还比较少.

基于以上背景, 本文提出一种适用于离散时间最优调节问题的事件驱动迭代神经网络策略. 通过收敛性分析和 HDP 实现, 得到基于事件环境下的迭代自适应评判算法. 然后为基于事件的离散时间动态系统设计一个实用的触发条件. 众所周知, 迭代自适应评判方法在学习近似最优控制方面具有重要意义, 而事件驱动机制在通信资源利用方面优势明显. 因此, 将这两种机制结合起来, 可以得到一种有效的离散时间非线性系统的事件驱动迭代神经控制方法. 也就是说, 通过本文的研究, 迭代自适应评判控制和事件驱动控制的应用范围都将得到扩大.

在本文中, $\mathcal{R}$是所有实数的集合. $\mathcal{R}^n$是所有n维实向量组成的欧氏空间. 设$\boldsymbol{\Omega}$是$\mathcal{R}^n$的一个紧集并且$\boldsymbol{\Psi}(\boldsymbol{\Omega})$是上容许控制律的集合. $\mathcal{R}^{n\times m}$是所有$n\times m$维实矩阵组成的空间. $\|\cdot\|$是$\mathcal{R}^n$中向量的向量范数或$\mathcal{R}^{n\times m}$中矩阵的矩阵范数. I_n是$n\times n$维的单位矩阵. $\mathcal{N}$代表所有非负整数的集合, 即$\{0,1,2,\ldots\}$. 上标“T”代表转置操作.

1 问题描述

本文考虑由下式描述的一类离散时间非线性动态系统:

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

式中, $\boldsymbol{x}(k)\in\mathcal{R}^n$是状态变量, $\boldsymbol{u}(k)\in\mathcal{R}^m$是控制输入, $f(\cdot)$和$g(\cdot)$是可微的并且有$f(0)=0$. 通常令$\boldsymbol{x}(0)$作为初始状态. 假设$f+g\boldsymbol{u}$在包含原点的集合$\boldsymbol{\Omega}\subset\mathcal{R}^n$上是 Lipschitz 连续的. 此外, 假设系统 (1) 可以在集合$\boldsymbol{\Omega}$上借助一个状态反馈控制律$\boldsymbol{u}(k)=\boldsymbol{\mu}(\boldsymbol{x}(k))$来镇定.

为了描述基于事件的设计框架, 定义单调递增序列$\{s_j\}_{j=0}^{\infty}$, 其中, $j\in\mathcal{N}$. 这里, 基于事件的控制信号仅在采样时刻s_0, s_1, s_2,...更新. 于是, 反馈控制律可以表示为$\boldsymbol{u}(k)=\boldsymbol{\mu}(\boldsymbol{x}(s_j))$, 其中, $\boldsymbol{x}(s_j)$是关于时刻$k=s_j$的状态, $k\in[s_j,s_{j+1})$, $j\in\mathcal{N}$. 在这种结构下, 需要一个零阶保持器来保持在时刻$k=s_j$时的事件驱

动控制输入，直到下一个事件发生. 基于事件的误差信号是上述结构的基本组成部分，定义为

$$\boldsymbol{e}(k)=\boldsymbol{x}(s_j)-\boldsymbol{x}(k), k\in[s_j,s_{j+1}),\ j\in\mathcal{N} \tag{2}$$

式中，$\boldsymbol{x}(s_j)$是采样状态，$\boldsymbol{x}(k)$是当前的状态向量. 利用表达式$\boldsymbol{x}(s_j)=\boldsymbol{x}(k)+\boldsymbol{e}(k)$，反馈控制律可以改写为$\boldsymbol{u}(k)=\boldsymbol{\mu}(\boldsymbol{x}(s_j))=\boldsymbol{\mu}(\boldsymbol{x}(k)+\boldsymbol{e}(k))$. 于是，可得

$$\boldsymbol{x}(k+1)=f(\boldsymbol{x}(k))+g(\boldsymbol{x}(k))\boldsymbol{\mu}(\boldsymbol{x}(k)+\boldsymbol{e}(k)),\ k\in\mathcal{N} \tag{3}$$

这可以认为是非线性系统 (1) 的闭环形式.

本文考虑最优控制问题，需要得到一个反馈控制律$\boldsymbol{\mu}\in\boldsymbol{\Psi}(\boldsymbol{\Omega})$来最小化

$$J(\boldsymbol{x}(k))=\sum_{\ell=k}^{\infty}U(\boldsymbol{x}(\ell),\boldsymbol{\mu}(\boldsymbol{x}(s_j))) \tag{4}$$

式中，$\boldsymbol{\mu}(\boldsymbol{x}(s_j))=\boldsymbol{\mu}(\boldsymbol{x}(k)+\boldsymbol{e}(k))$，$j\in\mathcal{N}$，$U(\boldsymbol{x},\boldsymbol{u})\geqslant 0,\forall\boldsymbol{x},\boldsymbol{u}$是效用函数，且有$U(0,0)=0$成立. 在本文中，效用函数选取为二次型形式

$$\begin{aligned}U(\boldsymbol{x}(k),\boldsymbol{\mu}(\boldsymbol{x}(s_j)))=\boldsymbol{x}^{\mathrm{T}}(k)\boldsymbol{Q}\boldsymbol{x}(k)+\\ \boldsymbol{\mu}^{\mathrm{T}}(\boldsymbol{x}(s_j))\boldsymbol{P}\boldsymbol{\mu}(\boldsymbol{x}(s_j))\end{aligned} \tag{5}$$

式中涉及到的$\boldsymbol{Q}\in\mathcal{R}^{n\times n}$和$\boldsymbol{P}\in\mathcal{R}^{m\times m}$都是正定矩阵.

回顾著名的最优性原理，最优代价函数定义为

$$J^*(\boldsymbol{x}(k))=\min_{\{\boldsymbol{\mu}(\cdot)\}}\sum_{\ell=k}^{\infty}U(\boldsymbol{x}(\ell),\boldsymbol{\mu}(\boldsymbol{x}(s_j))) \tag{6}$$

且满足以下的离散时间 HJB 方程：

$$\begin{aligned}J^*(\boldsymbol{x}(k))=\min_{\boldsymbol{\mu}(\boldsymbol{x}(s_j))}\{U(\boldsymbol{x}(k),\boldsymbol{\mu}(\boldsymbol{x}(s_j)))+\\ J^*(\boldsymbol{x}(k+1))\}\end{aligned} \tag{7}$$

基于事件触发机制的最优控制策略$\boldsymbol{\mu}^*(\boldsymbol{x}(s_j))$可由下式计算：

$$\begin{aligned}\boldsymbol{\mu}^*(\boldsymbol{x}(s_j))=\arg\min_{\boldsymbol{\mu}(\boldsymbol{x}(s_j))}\{U(\boldsymbol{x}(k),\boldsymbol{\mu}(\boldsymbol{x}(s_j)))+\\ J^*(\boldsymbol{x}(k+1))\}\end{aligned} \tag{8}$$

考虑到仿射型动态系统和二次型效用函数，则有

$$\boldsymbol{\mu}^*(\boldsymbol{x}(s_j))=-\frac{1}{2}\boldsymbol{P}^{-1}g^{\mathrm{T}}(\boldsymbol{x}(k))\frac{\partial J^*(\boldsymbol{x}(k+1))}{\partial\boldsymbol{x}(k+1)} \tag{9}$$

需要注意的是，式 (7) 是动态规划过程中应该处理的关键式子.

在本文中，为了解决基于事件的最优控制设计，应该关注两个方面的问题. 一方面，需要下一个时间步的值$J^*(\boldsymbol{x}(k+1))$来获得最优代价函数$J^*(\boldsymbol{x}(k))$和最优控制$\mu^*(\boldsymbol{x}(s_j))$. 为了克服获取$J^*(\boldsymbol{x}(k+1))$和求解离散时间 HJB 方程的困难，下一节将介绍一种基于自适应评判设计的迭代结构. 另一方面，在基于事件的结构中，需要设计一个形如$\|\boldsymbol{e}(k)\|\leqslant\bar{e}$的事件触发条件，其中，$\bar{e}$是正阈值. 当一个事件满足此触发条件时，控制输入才会被更新. 基于事件控制的主要问题就是如何确定一个合适的触发阈值，这也将在下一节介绍.

2　基于事件的迭代自适应评判控制

本节重点介绍基于事件的迭代自适应评判控制框架，包括算法收敛性分析，神经网络实现和触发条件设计.

2.1　基于事件的迭代算法及其收敛性

应该指出的是，在基于事件的迭代自适应评判控制方法中，需要考虑带有触发信息的值函数学习过程. 选择一个小的正数，并构造两个迭代序列$\{J^{(i)}(\boldsymbol{x}(k))\}$和$\{\boldsymbol{\mu}^{(i)}(\boldsymbol{x}(s_j))\}$，由此开始执行算法，其中，$i$表示迭代指标且$i\in\mathcal{N}$. 令初始迭代指标$i=0$并且令初始代价函数$J^{(0)}(\cdot)=0$.

然后，迭代控制函数通过

$$\begin{aligned}\boldsymbol{\mu}^{(i)}(\boldsymbol{x}(s_j))=\arg\min_{\boldsymbol{\mu}(\boldsymbol{x}(s_j))}\{U(\boldsymbol{x}(k),\boldsymbol{\mu}(\boldsymbol{x}(s_j)))+\\ J^{(i)}(\boldsymbol{x}(k+1))\}=\\ -\frac{1}{2}\boldsymbol{P}^{-1}g^{\mathrm{T}}(\boldsymbol{x}(k))\frac{\partial J^{(i)}(\boldsymbol{x}(k+1))}{\partial\boldsymbol{x}(k+1)}\end{aligned} \tag{10}$$

进行求解. 在上述参数最小化运算中，状态向量$\boldsymbol{x}(k+1)=f(\boldsymbol{x}(k))+g(\boldsymbol{x}(k))\boldsymbol{\mu}(\boldsymbol{x}(s_j))$.

接下来，迭代代价函数通过

$$\begin{aligned}J^{(i+1)}(\boldsymbol{x}(k))=\min_{\boldsymbol{\mu}(\boldsymbol{x}(s_j))}\{U(\boldsymbol{x}(k),\boldsymbol{\mu}(\boldsymbol{x}(s_j)))+\\ J^{(i)}(\boldsymbol{x}(k+1))\}\end{aligned} \tag{11}$$

进行更新，也可以写为

$$\begin{aligned}J^{(i+1)}(\boldsymbol{x}(k))=U\left(\boldsymbol{x}(k),\boldsymbol{\mu}^{(i)}(\boldsymbol{x}(s_j))\right)+\\ J^{(i)}\left(f(\boldsymbol{x}(k))+g(\boldsymbol{x}(k))\boldsymbol{\mu}^{(i)}(\boldsymbol{x}(s_j))\right)\end{aligned} \tag{12}$$

需要注意的是，当$|J^{(i+1)}(\boldsymbol{x}(k))-J^{(i)}(\boldsymbol{x}(k))|\leqslant\epsilon$时，停止准则生效，从而获得近似最优控制律. 此外，通过令$i=i+1$来增加迭代指标，从而继续求解式 (10) 中的迭代控制函数和更新 (11) 中的迭代代价函数.

下面，根据有界性和单调性给出上面迭代算法的收敛性证明.

定理 1 迭代代价函数序列$\{J^{(i)}\}$是有上界的，即$0\leqslant J^{(i)}(\boldsymbol{x}(k))\leqslant\mathcal{J}$，$i\in\mathcal{N}$，其中，$\mathcal{J}$是一个正常数.

证明. 令$\zeta(\boldsymbol{x}(s_j))$为触发时刻$s_j$的任意容许控制输入，$\{A^{(i)}\}$是如下定义的一个序列：

$$A^{(i+1)}(\boldsymbol{x}(k))=U(\boldsymbol{x}(k),\zeta(\boldsymbol{x}(s_j)))+A^{(i)}(\boldsymbol{x}(k+1)) \tag{13}$$

式中，迭代指标取零时的初始值$A^{(0)}(\cdot)=0$. 易知，$A^{(1)}(\boldsymbol{x}(k))=U(\boldsymbol{x}(k),\zeta(\boldsymbol{x}(s_j)))$. 随着迭代指标$i$展开

$A^{(i+1)}(\boldsymbol{x}(k))-A^{(i)}(\boldsymbol{x}(k))$, 最终可以得到.

$$A^{(i+1)}(\boldsymbol{x}(k))-A^{(i)}(\boldsymbol{x}(k))=A^{(1)}(\boldsymbol{x}(k+i)) \tag{14}$$

即有

$$A^{(i+1)}(\boldsymbol{x}(k))=\sum_{\hbar=0}^{i}A^{(1)}(\boldsymbol{x}(k+\hbar)) \tag{15}$$

考虑到$\zeta(\boldsymbol{x}(s_j))$的容许性. 可知对于任意的迭代指标$i$, 都有$A^{(i+1)}(\boldsymbol{x}(k))\leqslant\mathcal{J}$成立. 由于式 (11) 中的迭代代价函数$J^{(i+1)}(\boldsymbol{x}(k))$包含了最小化运算, 可以进一步得到$J^{(i+1)}(\boldsymbol{x}(k))\leqslant A^{(i+1)}(\boldsymbol{x}(k))\leqslant\mathcal{J}$. 于是, 考虑到代价函数的非负性, 可以得到$0\leqslant J^{(i)}(\boldsymbol{x}(k))\leqslant\mathcal{J}, i\in\mathcal{N}$. 证毕.

定理 2 迭代代价函数序列$\{J^{(i)}\}$是非减的, 即$J^{(i)}(\boldsymbol{x}(k))\leqslant J^{(i+1)}(\boldsymbol{x}(k)), i\in\mathcal{N}$.

证明. 为了方便起见, 定义一个新的序列$\{B^{(i)}\}$且初始值$B^{(0)}(\cdot)=0$. 该序列中的元素更新方式如下:

$$B^{(i+1)}(\boldsymbol{x}(k))=U\left(\boldsymbol{x}(k),\boldsymbol{\mu}^{(i+1)}(\boldsymbol{x}(s_j))\right)+B^{(i)}(\boldsymbol{x}(k+1)) \tag{16}$$

利用数学归纳法, 首先因为$J^{(1)}(\boldsymbol{x}(k))-B^{(0)}(\boldsymbol{x}(k))=U(\boldsymbol{x}(k),\boldsymbol{\mu}^{(0)}(\boldsymbol{x}(s_j)))\geqslant 0$, 可以得到不等式$B^{(0)}(\boldsymbol{x}(k))\leqslant J^{(1)}(\boldsymbol{x}(k))$. 然后, 假设$B^{(i-1)}(\boldsymbol{x}(k))\leqslant J^{(i)}(\boldsymbol{x}(k))$对于任意状态向量都成立且$i=2,3,\cdots$, 注意到式 (12) 和由 (16) 推得的表达式

$$B^{(i)}(\boldsymbol{x}(k))=U(\boldsymbol{x}(k),\boldsymbol{\mu}^{(i)}(\boldsymbol{x}(s_j)))+B^{(i-1)}(\boldsymbol{x}(k+1)) \tag{17}$$

则有

$$B^{(i)}(\boldsymbol{x}(k))-J^{(i+1)}(\boldsymbol{x}(k))=B^{(i-1)}(\boldsymbol{x}(k+1))-J^{(i)}(\boldsymbol{x}(k+1))\leqslant 0 \tag{18}$$

因此, 可以得到对于任意$i\in\mathcal{N}$, 都有$B^{(i)}(\boldsymbol{x}(k))\leqslant J^{(i+1)}(\boldsymbol{x}(k))$成立, 这样就完成了数学归纳证明.

考虑到式 (11) 中代价函数$J^{(i)}(\boldsymbol{x}(k))$的导出方式, 则有$J^{(i)}(\boldsymbol{x}(k))\leqslant B^{(i)}(\boldsymbol{x}(k))$. 因此, 最终得到不等式$J^{(i)}(\boldsymbol{x}(k))\leqslant B^{(i)}(\boldsymbol{x}(k))\leqslant J^{(i+1)}(\boldsymbol{x}(k))$. 证毕.

根据定理 1 和定理 2, 迭代代价函数序列$\{J^{(i)}\}$是收敛的. 令当$i\to\infty$时的迭代代价函数为$J^{(\infty)}$. 考虑式 (11) 且根据定理 2 的结论, 则有

$$J^{(\infty)}(\boldsymbol{x}(k))\geqslant J^{(i+1)}(\boldsymbol{x}(k))=\min_{\boldsymbol{\mu}(\boldsymbol{x}(s_j))}\{U(\boldsymbol{x}(k),\boldsymbol{\mu}(\boldsymbol{x}(s_j)))+J^{(i)}(\boldsymbol{x}(k+1))\},\ i\in\mathcal{N} \tag{19}$$

当$i\to\infty$时, 进一步有

$$J^{(\infty)}(\boldsymbol{x}(k))\geqslant\min_{\boldsymbol{\mu}(\boldsymbol{x}(s_j))}\{U(\boldsymbol{x}(k),\boldsymbol{\mu}(\boldsymbol{x}(s_j)))+J^{(\infty)}(\boldsymbol{x}(k+1))\} \tag{20}$$

反之, 根据式 (11) 和定理 2, 有下式成立:

$$J^{(i+1)}(\boldsymbol{x}(k))\leqslant U(\boldsymbol{x}(k),\boldsymbol{\mu}(\boldsymbol{x}(s_j)))+J^{(i)}(\boldsymbol{x}(k+1))\leqslant U(\boldsymbol{x}(k),\boldsymbol{\mu}(\boldsymbol{x}(s_j)))+J^{(\infty)}(\boldsymbol{x}(k+1)),\ i\in\mathcal{N} \tag{21}$$

当$i\to\infty$时, 可得对于任意的$\boldsymbol{\mu}(\boldsymbol{x}(s_j))$, 都有

$$J^{(\infty)}(\boldsymbol{x}(k))\leqslant U(\boldsymbol{x}(k),\boldsymbol{\mu}(\boldsymbol{x}(s_j)))+J^{(\infty)}(\boldsymbol{x}(k+1)) \tag{22}$$

于是, 可得

$$J^{(\infty)}(\boldsymbol{x}(k))\leqslant\min_{\boldsymbol{\mu}(\boldsymbol{x}(s_j))}\{U(\boldsymbol{x}(k),\boldsymbol{\mu}(\boldsymbol{x}(s_j)))+J^{(\infty)}(\boldsymbol{x}(k+1))\} \tag{23}$$

综合式 (20) 和 (23), 最终得到

$$J^{(\infty)}(\boldsymbol{x}(k))=\min_{\boldsymbol{\mu}(\boldsymbol{x}(s_j))}\{U(\boldsymbol{x}(k),\boldsymbol{\mu}(\boldsymbol{x}(s_j)))+J^{(\infty)}(\boldsymbol{x}(k+1))\} \tag{24}$$

比较式 (7) 和 (24), 可以得到迭代序列$\{J^{(i)}\}$的极限, 即$J^{(\infty)}$, 正是代价函数的最优值. 因此, 有$J^{(i)}(\boldsymbol{x}(k))\to J^{(\infty)}(\boldsymbol{x}(k))=J^{*}(\boldsymbol{x}(k))$成立. 同理, 当$i\to\infty$时, 也有$\boldsymbol{\mu}^{(i)}(\boldsymbol{x}(s_j))\to\boldsymbol{\mu}^{*}(\boldsymbol{x}(s_j))$成立, 这可以看做一个推论.

2.2　基于神经网络的 HDP 技术实现

在实现迭代自适应评判算法时, 需要建立两个神经网络, 即评判网络和执行网络, 分别用于输出近似代价函数和近似控制律.

评判网络输出迭代代价函数的近似值, 即

$$\hat{J}^{(i+1)}(\boldsymbol{x}(k))=\boldsymbol{\omega}_{\mathrm{c}}^{(i+1)\mathrm{T}}\sigma\left(\boldsymbol{\nu}_{\mathrm{c}}^{(i+1)\mathrm{T}}\boldsymbol{x}(k)\right) \tag{25}$$

结合式 (12), 训练误差准则为

$$E_{\mathrm{c}}^{(i+1)}(k)=\frac{1}{2}\left[\hat{J}^{(i+1)}(\boldsymbol{x}(k))-J^{(i+1)}(\boldsymbol{x}(k))\right]^2 \tag{26}$$

这里涉及的权重矩阵更新方式为

$$\boldsymbol{\omega}_{\mathrm{c}}^{(i+1)}(l+1)-\boldsymbol{\omega}_{\mathrm{c}}^{(i+1)}(l)=-\eta_{\mathrm{c}}\left(\frac{\partial E_{\mathrm{c}}^{(i+1)}(k)}{\partial\boldsymbol{\omega}_{\mathrm{c}}^{(i+1)}(l)}\right) \tag{27a}$$

$$\boldsymbol{\nu}_{\mathrm{c}}^{(i+1)}(l+1)-\boldsymbol{\nu}_{\mathrm{c}}^{(i+1)}(l)=-\eta_{\mathrm{c}}\left(\frac{\partial E_{\mathrm{c}}^{(i+1)}(k)}{\partial\boldsymbol{\nu}_{\mathrm{c}}^{(i+1)}(l)}\right) \tag{27b}$$

式中, $\eta_{\mathrm{c}}>0$是评判网络的学习率, l是内循环的迭代指标. 其中, $\boldsymbol{\omega}_{\mathrm{c}}^{(i+1)}(l)$和$\boldsymbol{\nu}_{\mathrm{c}}^{(i+1)}(l)$是权重矩阵的第$l$次迭代值.

执行网络输出迭代控制函数的近似值, 即

$$\hat{\boldsymbol{\mu}}^{(i)}(\boldsymbol{x}(s_j))=\boldsymbol{\omega}_{\mathrm{a}}^{(i)\mathrm{T}}\sigma\left(\boldsymbol{\nu}_{\mathrm{a}}^{(i)\mathrm{T}}\boldsymbol{x}(s_j)\right) \tag{28}$$

值得注意的是, 执行网络的输入是基于事件的状态$\boldsymbol{x}(s_j)$, 这与传统评判网络的输入 (基于时间的状态) 不同. 学习过程的误差准则为

$$E_{\rm a}^{(i)}(s_j)=\frac{1}{2}\left(\hat{\boldsymbol{\mu}}^{(i)}(\boldsymbol{x}(s_j))-\boldsymbol{\mu}^{(i)}(\boldsymbol{x}(s_j))\right)^{\rm T}\times \left(\hat{\boldsymbol{\mu}}^{(i)}(\boldsymbol{x}(s_j))-\boldsymbol{\mu}^{(i)}(\boldsymbol{x}(s_j))\right) \tag{29}$$

其中, 根据式 (10) 可以直接计算$\boldsymbol{\mu}^{(i)}(\boldsymbol{x}(s_j))$. 相似地, 执行网络的权重更新算法为

$$\omega_{\rm a}^{(i)}(l+1)-\omega_{\rm a}^{(i)}(l)=-\eta_{\rm a}\left(\frac{\partial E_{\rm a}^{(i)}(s_j)}{\partial \omega_{\rm a}^{(i)}(l)}\right) \tag{30a}$$

$$\boldsymbol{\nu}_{\rm a}^{(i)}(l+1)-\boldsymbol{\nu}_{\rm a}^{(i)}(l)=-\eta_{\rm a}\left(\frac{\partial E_{\rm a}^{(i)}(s_j)}{\partial \boldsymbol{\nu}_{\rm a}^{(i)}(l)}\right) \tag{30b}$$

式中, $\eta_{\rm a}>0$是需要设计的学习率参数.

为清楚起见, 图 1 给出离散时间非线性系统基于事件的迭代 HDP 控制的结构简图. 其中, 实线代表信号流向, 虚线是两个神经网络的反向传播路径. 值得注意的是, 状态信息被传递到基于事件的模块用于转换信号状态, 传递到被控对象用于更新系统状态, 传递到评判网络用于计算代价函数. 因此, 系统状态组件包含三个重要角色.

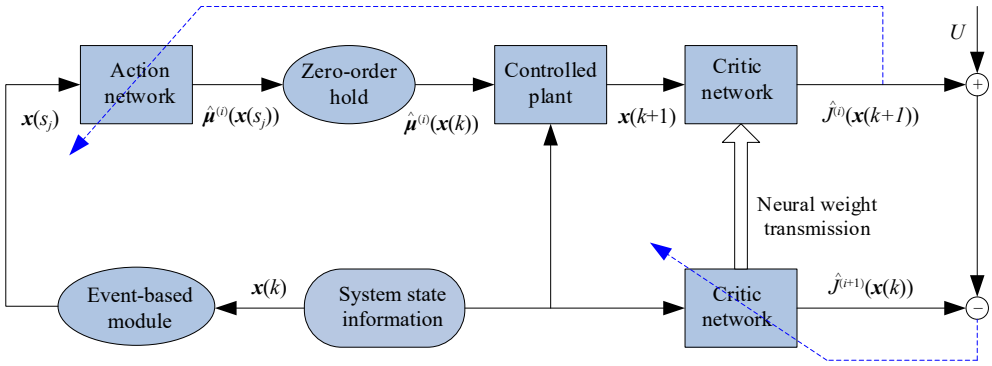

图 1　离散动态系统基于事件的迭代 HDP 框架简图

Fig.1　Simple diagram of the event-based iterative heuristic dynamic programming (HDP) framework with discrete dynamic plants

2.3　事件触发条件设计

为了确定非线性离散动态系统的具体事件触发条件, 这里给出文献 [23–25] 中使用的如下假设. 值得注意的是, 根据式 (3), $\boldsymbol{x}(k+1)$是关于$\boldsymbol{x}(k)$和$\boldsymbol{e}(k)$的函数.

假设 1 范数不等式$\|\boldsymbol{e}(k)\|\leqslant\|\boldsymbol{x}(k)\|$和$\|\boldsymbol{x}(k+1)\|\leqslant\beta\|\boldsymbol{x}(k)\|+\beta\|\boldsymbol{e}(k)\|$成立, 其中, $\boldsymbol{x}(k+1)$由式 (3) 给出, 这里的正常数$\beta\in(0,0.5)$.

定理 3 如果假设 1 成立, 则触发条件

$$\|\boldsymbol{e}(k)\|\leqslant\bar{e}=\frac{1-(2\beta)^{k-s_j}}{1-2\beta}\beta\|\boldsymbol{x}(s_j)\|,\beta\in(0,0.5) \tag{31}$$

能够保证基于事件的控制器设计的可用性.

证明. 考虑到式 (3) 给出的动态系统和假设 1, 可以得到

$$\begin{aligned}&\|\boldsymbol{e}(k)\|\leqslant\|\boldsymbol{x}(k)\|\leqslant\\&\beta\|\boldsymbol{x}(k-1)\|+\beta\|\boldsymbol{e}(k-1)\|\leqslant\\&\beta(\|\boldsymbol{e}(k-1)\|+\|\boldsymbol{x}(s_j)\|)+\beta\|\boldsymbol{e}(k-1)\|=\\&2\beta\|\boldsymbol{e}(k-1)\|+\beta\|\boldsymbol{x}(s_j)\|\end{aligned} \tag{32}$$

使用同样的方法, 易知

$$\|\boldsymbol{e}(k-1)\|\leqslant2\beta\|\boldsymbol{e}(k-2)\|+\beta\|\boldsymbol{x}(s_j)\| \tag{33}$$

然后, 结合式 (32) 和式 (33), 则有

$$\|\boldsymbol{e}(k)\|\leqslant2\beta(2\beta\|\boldsymbol{e}(k-2)\|+\beta\|\boldsymbol{x}(s_j)\|)+\beta\|\boldsymbol{x}(s_j)\| \tag{34}$$

利用$\boldsymbol{e}(s_j)=0$, 并如同式 (34) 一样扩展$\|\boldsymbol{e}(k)\|$, 最终可以得到

$$\|\boldsymbol{e}(k)\|\leqslant\beta\|\boldsymbol{x}(s_j)\|\sum_{l=0}^{\bar{l}}(2\beta)^l \tag{35}$$

式中, $\bar{l}=k-s_j-1$. 基于不等式 (35), 则有触发条件$\|\boldsymbol{e}(k)\|\leqslant\bar{e}$, 其中的阈值可以写成

$$\bar{e}=\frac{1-(2\beta)^{k-s_j}}{1-2\beta}\beta\|\boldsymbol{x}(s_j)\|,\ \beta\in(0,0.5) \tag{36}$$

证毕.

定理 3 提出的触发条件与假设 1 中的采样状态和预先指定的常数密切相关, 因此并不是唯一的. 这个条件是本文提出的事件驱动迭代自适应评判控制框架的设计基础. 为了表明触发条件的作用, 图 2 给出了执行迭代 HDP 算法之后的事件驱动控制实现, 其中, $\hat{\boldsymbol{\mu}}^*(\boldsymbol{x}(k))$是已获得的近似最优控制器, 也就是用于事件驱动设计的实际控制律. 图 2 的蓝色虚线代表下一步迭代的状态, 要与当前的状态区分. 当触发条件得以满足时 (转向“Y”), 控制信号仍然保持之前的值$\hat{\boldsymbol{\mu}}^*(\boldsymbol{x}(s_{j-1}))$. 然而, 当触发条件不被满足时 (转向“N”), 控制信号将通过执行网络更新成为$\hat{\boldsymbol{\mu}}^*(\boldsymbol{x}(s_j))$. 经过零阶保持器的作用之后, 事件驱动控制信号$\hat{\boldsymbol{\mu}}^*(\boldsymbol{x}(s_{j-1}))$或$\hat{\boldsymbol{\mu}}^*(\boldsymbol{x}(s_j))$中的一个将被转换成$\hat{\boldsymbol{\mu}}^*(\boldsymbol{x}(k))$, 最终就可以

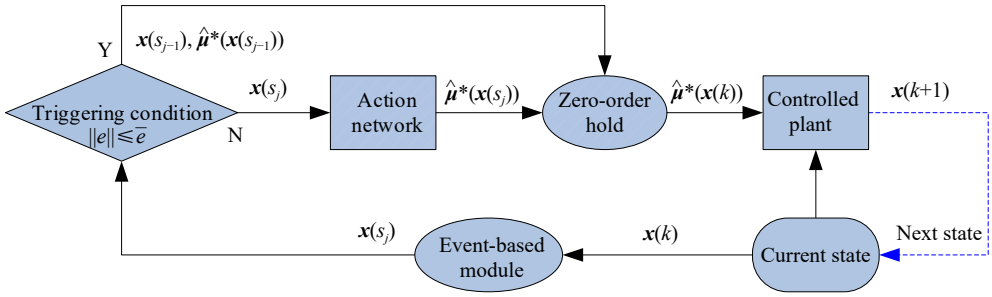

图 2　执行迭代 HDP 算法之后的事件驱动控制实现过程

Fig.2　Event-based control implementation process after conducting the iterative HDP algorithm

应用于原始被控系统.

3　仿真研究

本节给出将基于事件迭代自适应评判方法应用到一些特定动态系统的仿真研究, 以验证近似最优控制性能.

例 1 考虑质量弹簧阻尼器系统的离散化形式[24]

$$\boldsymbol{x}(k+1)=\begin{bmatrix} 0.9996\boldsymbol{x}_1(k)+0.0099\boldsymbol{x}_2(k) \\ -0.0887\boldsymbol{x}_1(k)+0.97\boldsymbol{x}_2(k) \end{bmatrix}+\begin{bmatrix} 0 \\ 0.0099 \end{bmatrix}\boldsymbol{u}(k) \tag{37}$$

式中, 状态向量为$\boldsymbol{x}(k)=[\boldsymbol{x}_1(k),\boldsymbol{x}_2(k)]^{\mathrm{T}}$, 控制变量是$\boldsymbol{u}(k)$. 为了解决基于事件的最优调节问题, 代价函数中的效用参数分别选为$\boldsymbol{Q}=0.01\boldsymbol{I}_2$和$\boldsymbol{P}=I$.

通过将网络结构预先分别设定为 2–8–1(输入层, 隐藏层, 输出层神经元的个数) 和 2–8–1, 然后根据式 (27) 和式 (30) 在迭代框架中训练评判网络和执行网络. 在训练过程中, 选择初始状态$\boldsymbol{x}(0)=[1,0.5]^{\mathrm{T}}$并且取学习率为$\eta_{\mathrm{c}}=\eta_{\mathrm{a}}=0.1$. 评判网络和执行网络的初始权重分别在$[-0.1,0.1]$和$[-0.5,0.5]$中随机选取. 特别地, 需要将基于事件的机制应用于执行网络. 采用迭代 HDP 算法进行 290 轮迭代, 每轮迭代设定 2000 次训练. 如果达到预先指定的精度$\epsilon=10^{-6}$, 就结束评判网络和执行网络的训练, 即获得满意的学习效果. 图 3 给出了迭代代价函数的收敛趋势, 也验证了定理 1 和定理 2 中的陈述.

在基于事件的控制设计中, 令$\beta=0.1$并且指定触发阈值表达式 (36) 具体如下:

$$\bar{e}=\frac{1-0.2^{k-s_j}}{8}\|\boldsymbol{x}(s_j)\| \tag{38}$$

为了与传统时间驱动方法进行比较, 执行两种情况, 即事件驱动模式和时间驱动模式下的迭代 HDP 算法, 其中情况 1(Case1) 是本文提出的事件驱动模式, 情况 2(Case2) 是文献 [12] 中提出的传统时间驱动模式. 图 4 给出了应用事件驱动迭代自适应评判方法时的状态响应, 其中也给出了应用传统迭代 HDP 算法时的状态轨迹. 这里, 可以

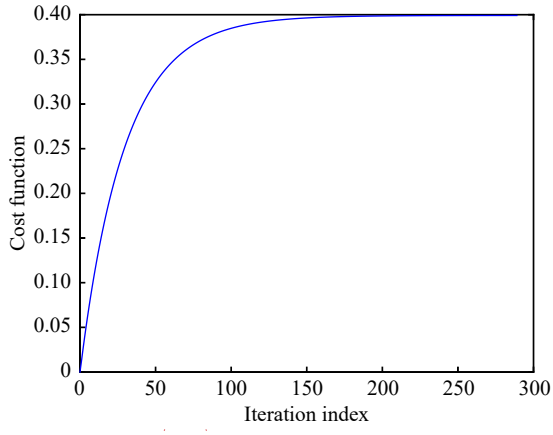

图 3　迭代代价函数的收敛性 (例 1)

Fig.3　Convergence of the iterative cost function (Example 1)

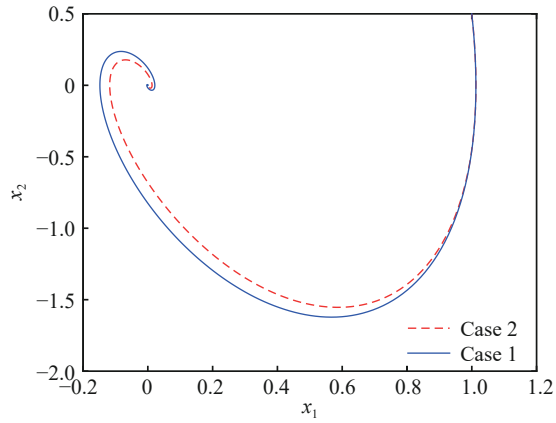

图 4　两种情况下的状态轨迹 (例 1)

Fig.4　State trajectory of the two cases (Example 1)

清楚地看到, 正如传统的迭代 HDP 算法一样, 基于事件情况下的系统状态也能够最终收敛到零向量. 顺便指出, 触发阈值的变化曲线如图 5 所示, 它随着系统状态的变化也趋于零. 此外, 与传统的迭代 HDP 算法相比, 基于事件方法的控制曲线呈阶梯状, 如图 6 所示. 在仿真中, 基于时间情形下的控制输入更新了 500 个时间步, 然而在基于事件情况下, 仅仅需要 222 个时间步, 对应的驱动时刻间隔如图 7 所示. 因此, 这就验证了基于事件的迭代自适应评判方法的优越之处, 即通信资源的利用效

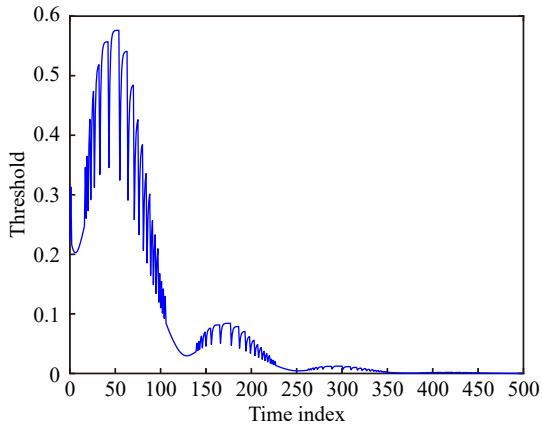

图 5　触发阈值 (例 1)

Fig.5　Triggering threshold (Example 1)

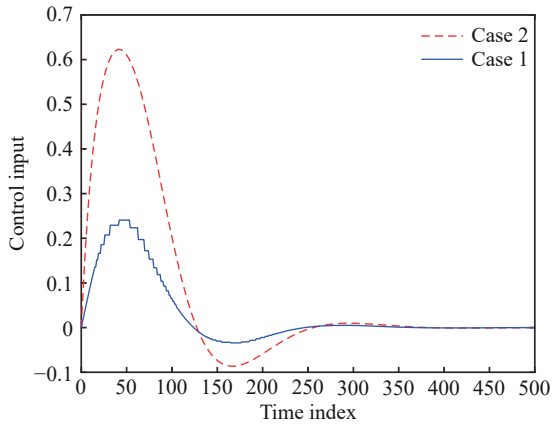

图 6　两种情况下的控制输入 (例 1)

Fig.6　Control input of the two cases (Example 1)

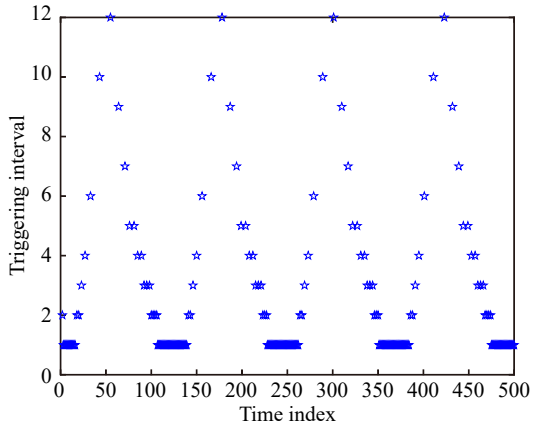

图 7　驱动时刻间隔 (例 1)

Fig.7　Triggering interval (Example 1)

率确实得以提高.

例 2 这里引入非线性因素, 考虑如下离散时间非线性系统

$$\boldsymbol{x}(k+1)=\begin{bmatrix}-0.5\cos(1.4\boldsymbol{x}_2(k))\sin(0.4\boldsymbol{x}_1(k))\\ 0.1\boldsymbol{x}_2^2(k)\end{bmatrix}+\begin{bmatrix}\boldsymbol{x}_1(k)+0.03\boldsymbol{x}_2(k)\\ -0.1\boldsymbol{x}_1(k)+\boldsymbol{x}_2(k)\end{bmatrix}+\begin{bmatrix}0\\ 0.008\end{bmatrix}\boldsymbol{u}(k)\tag{39}$$

式中, 状态向量为$\boldsymbol{x}(k)=[\boldsymbol{x}_1(k),\boldsymbol{x}_2(k)]^{\mathrm{T}}$, 控制变量是$\boldsymbol{u}(k)$. 为了解决事件驱动最优控制问题, 这里除了$\boldsymbol{P}=2\boldsymbol{I}$, $\boldsymbol{x}(0)=[1,-1]^{\mathrm{T}}$, 以及在$[-1,1]$中随机选择执行网络的初始权值之外, 其他主要参数的设置都与例 1 一样. 在进行 300 轮迭代运算之后, 代价函数的收敛性如图 8 所示. 与文献 [24] 不同的是, 本文的方法可以很好地观察迭代代价函数的收敛性. 当关注值函数学习过程时, 对收敛性能的观测就很有意义. 实际上, 这也是事件驱动环境下离散动态系统迭代自适应评判算法的优点之一.

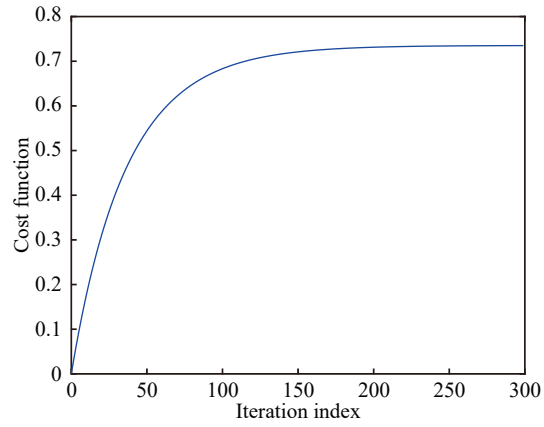

图 8　迭代代价函数的收敛性 (例 2)

Fig.8　Convergence of the iterative cost function (Example 2)

分别考虑基于事件和基于时间的控制模式, 图 9 给出两种情况下的状态轨迹. 可以看到, 图 9 中的两条轨迹非常接近, 都具有很好的稳定效果. 此外, 触发阈值和控制输入分别如图 10 和图 11 所示. 与状态曲线不同, 两种情况下的控制轨迹具有明显区别. 在这个例子中, 基于时间和基于事件框架的控制输入分别更新了 300 次和 85 次, 这里的驱动时刻间隔如图 12 所示. 也就是说, 事件驱动结构使得控制信号更新次数下降了 71.67%. 上述仿真结果表明, 基于事件的设计策略在保持较好稳定性能的前提下, 可以有效地减少控制信号的更新次数.

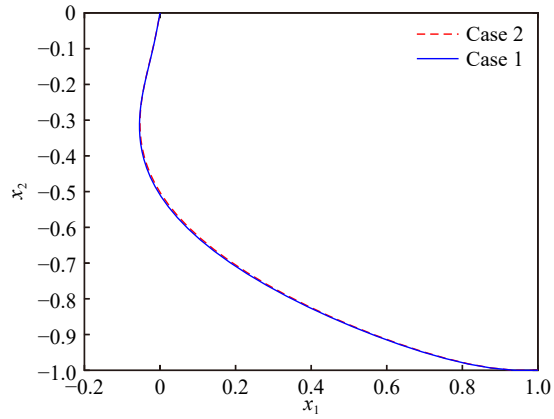

图 9　两种情况下的状态轨迹 (例 2)

Fig.9　State trajectory of the two cases (Example 2)

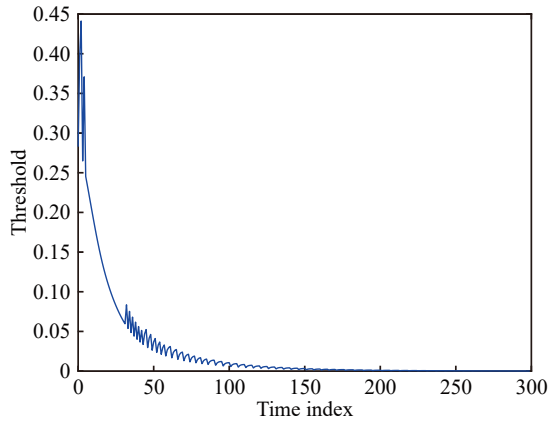

图 10　触发阈值 (例 2)

Fig.10　Triggering threshold (Example 2)

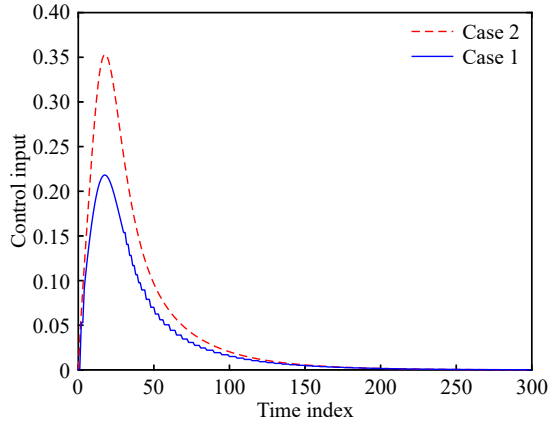

图 11　两种情况下的控制输入 (例 2)

Fig.11　Control input of the two cases (Example 2)

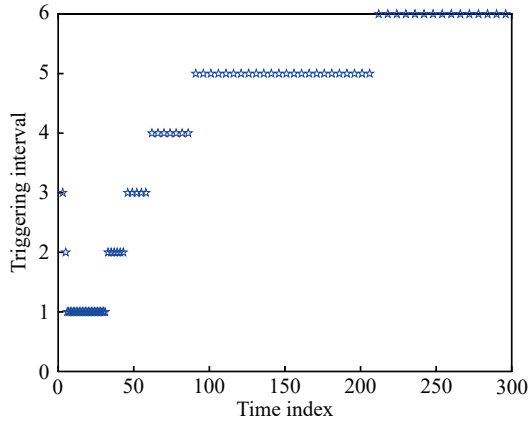

图 12　驱动时刻间隔 (例 2)

Fig.12　Triggering interval (Example 2)

4　结论

本文提出一种基于事件的迭代神经控制方法, 用以解决离散动态系统的最优调节问题. 通过收敛性分析, 神经网络实现和触发阈值设计, 构造基于事件迭代自适应评判算法的完整框架. 通过仿真研究, 验证了事件驱动迭代神经控制方法的优越性能.

参　考　文　献

[1] Werbos P J. Approximate dynamic programming for real-time control and neural modeling. In White D A and Sofge D A (Eds.) *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*. New York, NY: Van Nostrand Reinhold, 1992

[2] Li J N, Chai T Y, Lewis F L, et al. Off-policy interleaved Q-learning: Optimal control for affine nonlinear discrete-time systems. *IEEE Trans Neural Netw Learn Syst*, 2019, 30(5): 1308

[3] Zhang H G, Liu Y, Xiao G Y, et al. Data-based adaptive dynamic programming for a class of discrete-time systems with multiple delays. *IEEE Trans Syst Man Cybern*:*Syst*, 2020, 50(2): 432

[4] Zhang H G, Jiang H, Luo Y H, et al. Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method. *IEEE Trans on Ind Electron*, 2017, 64(5): 4091

[5] Ha M M, Wang D, Liu D R. Generalized value iteration for discounted optimal control with stability analysis. *Syst Control Lett*, 2021, 147: 104847

[6] Wang D, Ha M M, Qiao J F. Data-driven iterative adaptive critic control towards an urban wastewater treatment plant. *IEEE Trans Ind Electron*, 2021, 68(8): 7362

[7] Wang D, Ha M M, Qiao J F, et al. Data-based composite control design with critic intelligence for a wastewater treatment platform. *Artif Intell Rev*, 2020, 53(5): 3773

[8] Liang M M, Wang D, Liu D R. Improved value iteration for neural-network-based stochastic optimal control design. *Neural Netw*, 2020, 124: 280

[9] Liang M M, Wang D, Liu D R. Neuro-optimal control for discrete stochastic processes via a novel policy iteration algorithm. *IEEE Trans Syst Man Cybern*:*Syst*, 2020, 50(11): 3972

[10] Hou J X, Wang D, Liu D R, et al. Model-free H_∞ optimal tracking control of constrained nonlinear systems via an iterative adaptive learning algorithm. *IEEE Trans Syst Man Cybern*:*Syst*, 2020, 50(11): 4097

[11] Luo B, Liu D R, Huang T W, et al. Model-free optimal tracking control via critic-only Q-learning. *IEEE Trans Neural Netw Learn Syst*, 2016, 27(10): 2134

[12] Al-Tamimi A, Lewis F L, Abu-Khalaf M. Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof. *IEEE Trans Syst Man Cybern B*:*Cybern*, 2008, 38(4): 943

[13] 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 Trans Neural Netw*, 2009, 20(9): 1490

[14] Wang D, Liu D R, Wei Q L, et al. Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming. *Automatica*, 2012, 48(8): 1825

[15] Zhong X, Ni Z, He H. A theoretical foundation of goal representation heuristic dynamic programming. *IEEE Trans Neural Netw Learn Syst*, 2016, 27(12): 2513

[16] Yang X, Liu D R, Wang D, et al. Discrete-time online learning control for a class of unknown nonaffine nonlinear systems using reinforcement learning. *Neural Netw*, 2014, 55: 30

[17] Tabuada P. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Trans Autom Control*, 2007, 52(9): 1680

[18] Fan Q Y, Yang G H. Event-based fuzzy adaptive fault-tolerant control for a class of nonlinear systems. *IEEE Trans Fuzzy Syst*, 2018, 26(5): 2686

[19] Zhou Y, Zeng Z. Event-triggered impulsive control on quasi-synchronization of memristive neural networks with time-varying delays. *Neural Netw*, 2019, 110: 55

[20] Wang D, Zhong X N. Advanced policy learning near-optimal regulation. *IEEE/CAA J Autom Sin*, 2019, 6(3): 743

[21] Wang D. Research progress on learning-based robust adaptive critic control. *Acta Autom Sin*, 2019, 45(6): 1031
(王鼎. 基于学习的鲁棒自适应评判控制研究进展. 自动化学报, 2019, 45(6): 1031)

[22] Zhang H G, Su H G, Zhang K, et al. Event-triggered adaptive dynamic programming for non-zero-sum games of unknown nonlinear systems via generalized fuzzy hyperbolic models. *IEEE Trans Fuzzy Syst*, 2019, 27(11): 2202

[23] Eqtami A, Dimarogonas D V, Kyriakopoulos K J. Event-triggered control for discrete-time systems // *Proceedings of the* 2010 *American Control Conference*, Baltimore, 2010: 4719

[24] Dong L, Zhong X N, Sun C Y, et al. Adaptive event-triggered control based on heuristic dynamic programming for nonlinear discrete-time systems. *IEEE Trans Neural Netw Learn Syst*, 2017, 28(7): 1594

[25] Ha M M, Wang D, Liu D R. Event-triggered adaptive critic control design for discrete-time constrained nonlinear systems. *IEEE Trans Syst Man Cybern*: *Syst*, 2020, 50(9): 3158

[26] Dhar N K, Verma N K, Behera L. Adaptive critic-based event-triggered control for HVAC system. *IEEE Trans Ind Inform*, 2018, 14(1): 178