文章编号: 1001-0920(2019)11-2343-07 DOI: 10.13195/j.kzyjc.2019.0417

基于非策略Q-学习的网络控制系统最优跟踪控制

李金娜[1,2,3†], 尹子轩[1]

(1. 沈阳化工大学 信息工程学院，沈阳 110142；2. 辽宁石油化工大学 信息与控制工程学院，辽宁 抚顺 113001；3. 东北大学 流程工业综合自动化国家重点实验室，沈阳 110004)

摘 要: 针对具有数据包丢失的网络化控制系统跟踪控制问题,提出一种非策略Q-学习方法,完全利用可测数据,在系统模型参数未知并且网络通信存在数据丢失的情况下,实现系统以近似最优的方式跟踪目标. 首先,刻画具有数据包丢失的网络控制系统,提出线性离散网络控制系统跟踪控制问题;然后,设计一个Smith预测器补偿数据包丢失对网络控制系统性能的影响,构建具有数据包丢失补偿的网络控制系统最优跟踪控制问题;最后,融合动态规划和强化学习方法,提出一种非策略Q-学习算法. 算法的优点是: 不要求系统模型参数已知,利用网络控制系统可测数据,学习基于预测器状态反馈的最优跟踪控制策略;并且该算法能够保证基于Q-函数的迭代Bellman方程解的无偏性. 通过仿真验证所提方法的有效性.

关键词: 网络控制； 非策略Q-学习； 线性二次跟踪(LQT)； 数据包丢失

中图分类号: TP13 **文献标志码:** A

Off-policy Q-learning: Optimal tracking control for networked control systems

LI Jin-na[1,2,3†], *YIN Zi-xuan*[1]

(1. College of Information Engineering，Shenyang University of Chemical Technology，Shenyang 110142，China；2. School of Information and Control Engineering，Liaoning Shihua University，Fushun 113001，China；3. State Key Lab of Synthetical Automation for Process Industries，Northeastern University，Shenyang 110004，China)

Abstract: This paper develops a novel off-policy Q-learning method for solving linear quadratic tracking (LQT) problem in discrete-time networked control systems with packet dropout. The proposed method can be implemented using measured data without requiring systems dynamics to be known a priori, and it also allows bounded packet loss. First, networked control systems with packet dropout are established, thus an optimal tracking problem of linear discrete-time networked control systems is further formulated. Then, a Smith predictor is designed to predict current state based on historical data measured on the communication network. On this basis, an optimal tracking problem with packet dropout compensation is put up. Finally, a novel off-policy Q-learning algorithm is developed by integrating dynamic programming with reinforcement learning. The merit of the proposed algorithm is that the optimal tracking control law based predicted states of systems can be learned using only measured data without the need of knowing system dynamics. Moreover, the unbiasedness of solution to Q-function based Bellman equation can be guaranteed by using off-policy Q-learning approach. The simulation results show that the proposed method has good tracking performance for the network control system with unknown dynamic state and packet dropout.

Keywords: networked control system； off-policy Q-learning； linear quadratic tracking； packet dropout

0 引 言

强化学习是一种通过与环境进行“试错”交互寻找能够带来最大期望累积奖赏策略的学习方法[1-3]. 目前越来越多的研究开始将强化学习方法应用在控制领域的各个方向,可以达到最优控制的效果[4-5]. 根据学习过程中行为策略与目标策略是否一致,将强化学习分为策略(on-policy)学习和非策略(off-policy)学习. 如果在学习过程中,动作选择的行为策略与学习改进的目标策略一致,则该方法被称为策略学习,否则被称为非策略学习[6-7].

非策略强化学习相比于策略强化学习具有一些优势,并且具有预期的特性: 1)它解决了探索-开发的

收稿日期: 2019-04-07； **修回日期:** 2019-07-16.

基金项目: 国家自然科学基金项目(61673280, 61525302, 61590922, 61503257)；辽宁省高等学校创新人才项目(LR2017006)；辽宁省自然基金计划重点领域联合开放基金项目(2019-KF-03-06)；辽宁石油化工大学基金项目(2018XJJ-005).

†通讯作者. E-mail: lijinna_721@126.com.

困境. 系统采用任意行为策略来保证数据的充分挖掘,而实际学习的是最优开发策略或目标策略. 2)通常需要探测噪声来保证持续激励(PE)条件,非策略强化学习能保证贝尔曼方程解的无偏性. 对于最优控制问题,目前应用的Q-学习算法取得了很多研究成果[8-10],但是采用非策略Q-学习研究最优化控制还处于初级阶段. 文献[11]采用非策略Q-学习算法解决了离散系统H_∞控制;文献[12-13]给出了仿射非线性系统交错非策略Q-学习迭代算法,自适应评判Q-学习算法,学习最优控制策略.

随着信息技术、网络技术和计算机技术的飞速发展,基于网络的控制系统已经成为自动化领域的一个重要控制技术,网络控制系统的研究也是近年来自动控制领域的研究热点[14-16]. 对于具有数据包丢失的网络控制系统,现有的控制和优化方法主要采用基于模型的控制策略,要求系统模型参数已知,采用确定的、鲁棒或者随机控制方法镇定系统,并优化系统性能[17-19].

在网络控制系统中,网络环境千变万化,网络结构也可能随时改变,很难建立精准的系统模型. 针对系统模型参数未知的情况,文献[20]提出了一种线性网络控制的最优控制方法;文献[21]针对具有时变系统矩阵的未知网络控制问题,采用随机Q-学习方法设计了事件采样框架下的最优控制器;文献[22]将该方法推广到非线性的情况,但是当信息传输发生数据包丢失时,会给最优控制器设计带来挑战;文献[23]提出了Smith预测补偿,通过策略Q-学习算法找到最优跟踪控制器增益. 然而,采用非策略Q-学习方法,补偿数据包丢失,在系统模型参数未知的情况下,解决网络控制系统最优跟踪控制问题还未得到研究,这是本文研究的动机.

本文使用Q-学习算法,在线性离散网络控制系统的动力学方程未知的情况下,给出近似最优跟踪控制策略,优化网络控制系统性能.

本文的创新性在于:1)不同于传统的网络系统控制方法设计[20-22],本文讨论的是在系统模型存在未知参数,并存在数据包丢失的情况下利用Q-学习算法学习最优跟踪控制策略;2)本文不同于文献[23]中的策略Q-学习,本文采用完全数据驱动的非策略Q-学习方法,补偿数据包丢失,在不依赖系统模型参数的情况下,解决网络控制系统最优跟踪控制问题.

1 具有丢包补偿的优化问题描述

本节将介绍线性二次跟踪(LQT)问题和网络诱导丢包的模型,阐述具有数据包丢失的网络控制系统二次跟踪问题.

考虑如下线性离散系统:

$$\begin{cases} x(k+1)=Ax(k)+Bu(k);\\ y(k)=Cx(k). \end{cases} \tag{1}$$

其中:$x(k)$是被控对象状态,为$n_x\times 1$维;$u(k)$是被控输入,为$n_u\times 1$维;$y(k)$是被控输出,为$n_y\times 1$维;A、B和C分别为$n_x\times n_x$、$n_x\times n_u$和$n_y\times n_x$维.

参考信号如下:

$$r(k+1)=Fr(k). \tag{2}$$

其中:$r(k)$是参考输入,为$n_r\times 1$维;F是$n_r\times n_r$维. 在这个跟踪问题中,需要令系统(1)中的输出$y(k)$跟踪参考输入$r(k)$.

令$X(k)=\begin{bmatrix}x(k)\\ r(k)\end{bmatrix}$,由式(1)和(2)得到如下增广系统:

$$\begin{cases} X(k+1)=A_2X(k)+B_2u(k);\\ y(k)=C_2X(k). \end{cases} \tag{3}$$

其中:$A_2=\begin{bmatrix}A & 0\\ 0 & F\end{bmatrix}$, $B_2=\begin{bmatrix}B\\ 0\end{bmatrix}$, $C_2=[C\quad 0]$.

1.1 构建丢包补偿的Smith预测器

如图1所示,测量状态并通过通信网络传递给控制器,控制器利用获得的系统状态信息计算控制输入.

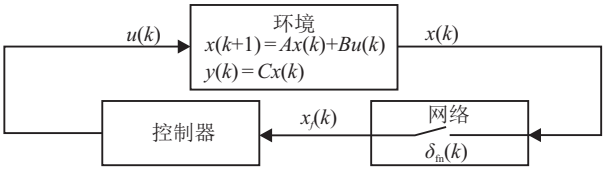

图1 具有反馈丢包的网络控制系统

假定状态信息是通过单个数据包传输的,某些数据包在传输中不可避免地丢失,称为网络诱导型的丢包. $x_f(k)$为控制器端接收的系统状态,其表达式为

$$x_f(k)=x(k-\delta_{\rm fn}(k)). \tag{4}$$

其中:$\delta_{\rm fn}(k)$是发生的连续丢包数,$0<\delta_{\rm fn}(k)<\delta_{f\max}(k)$,$\delta_{f\max}(k)$是最大连续丢包数.

由式(1)得到

$$x(k)=A^{\delta_{\rm fn}}x(k-\delta_{\rm fn}(k))+\sum_{i=1}^{\delta_{\rm fn}}A^{i-1}Bu(k-i). \tag{5}$$

在使用TCP或UDP协议的情况下,丢包数$\delta_{\rm fn}(k)$是已知的.

当$\delta_{\rm fn}(k)=0$时,有

$$\bar{z}(k)=[\underbrace{x^{\rm T}(k)\quad 0\quad\cdots\quad 0}_{\delta_{f\max}+1}\quad\underbrace{0\quad\cdots\quad 0}_{\delta_{f\max}}\quad r^{\rm T}(k)]^{\rm T}. \tag{6}$$

当$\delta_{\rm fn}(k)=1$时,有

$$\bar{z}(k)=[\underbrace{0\quad x^{\rm T}(k-1)\quad\cdots\quad 0}_{\delta_{f\max}+1}\quad\underbrace{u^{\rm T}(k-1)\quad\cdots\quad 0}_{\delta_{f\max}}\quad r^{\rm T}(k)]^{\rm T}. \tag{7}$$

当$\delta_{\rm fn}(k)=\delta_{f\max}$时,有

$$\bar{z}(k)=[\underbrace{0\quad\cdots\quad x^{\rm T}(k-\delta_{f\max})}_{\delta_{f\max}+1}\quad\underbrace{u^{\rm T}(k-1)\quad\cdots\quad u^{\rm T}(k-\delta_{f\max})}_{\delta_{f\max}}\quad r^{\rm T}(k)]^{\rm T}. \tag{8}$$

根据式(5)构建如下Smith预测器[23]:

$$X(k)=M\bar{z}(k), \tag{9}$$

其中

$$M=\begin{bmatrix} I & A & \cdots & A^{\delta_{f\max}} & AB & \cdots & (A^{\delta_{f\max}})^{-1}B & 0\\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 1\end{bmatrix}. \tag{10}$$

注1　$\bar{z}(k)$在k时刻是已知的.

由于引入了Smith预测器(9),本文可以构建如下基于预测器估计的系统状态的反馈控制器:

$$u(k)=-KX(k)=-KM\bar{z}(k). \tag{11}$$

1.2　具有丢包补偿的优化问题阐述

本文研究的目的是设计控制器(11),最小化如下性能指标,实现系统以最优的方式跟踪参考输入:

$$J=\frac{1}{2}\sum_{i=k}^{\infty}\gamma^{i-k}((r(i)-y(i))^{\rm T}Q(r(i)-y(i))+u^{\rm T}(i)Ru(i)), \tag{12}$$

其中$0<\gamma<1$是一个折现因子. 如果参考信号发生器(2)是稳定的,则可以选择$\gamma=1$;如果(2)是不稳定的,例如跟踪一个单位步长,则需要$\gamma<1$. 事实上,可取任意可镇定控制输入(11),选择折现因子γ,使$F\gamma^{0.5}$稳定,以便保证闭环系统(3)稳定[23].

由式(3)、(9)和(11)给出具有丢包补偿的网络控制系统线性二次跟踪控制(LQT)问题:

$$\begin{cases}\min\limits_{u(k)}\dfrac{1}{2}\sum\limits_{i=k}^{\infty}\gamma^{i-k}((r(i)-y(i))^{\rm T}Q(r(i)-y(i))+\\ \qquad u^{\rm T}(i)Ru(i));\\ \text{s.t.}\ \ 式(3),(9)和(10).\end{cases} \tag{13}$$

注2　在Smith预测器的帮助下,此时的LQT问题可以获取当前系统状态.

2　基于非策略Q-学习方法求解优化问题

在这一节中,主要讨论解决存在丢包的离散网络系统LQT问题的非策略Q-学习方法. 首先在文献[23]的基础上引入Q-函数矩阵设计策略Q-学习算法,以便获取不依赖模型的控制器方案;然后在此基础上,引入行为控制器,结合基于Q-函数的贝尔曼方程,提出一种非策略Q-学习算法.

使用增广系统(3),网络诱导型丢包线性二次跟踪(丢包LQT)问题性能指标为

$$J=\frac{1}{2}\sum_{i=k}^{\infty}\gamma^{i-k}[X^{\rm T}(i)Q_1X(i)+u^{\rm T}(i)Ru(i)]. \tag{14}$$

其中: $Q_1={C_1}^{\rm T}QC_1, C_1=[C\quad -I]$.

令$\bar{K}=KM$,则有

$$u(k)=\bar{K}\bar{z}(k). \tag{15}$$

根据式(14),定义值函数和Q-函数分别为

$$V(X(k))=\frac{1}{2}\min_{u_k}\sum_{i=k}^{\infty}\gamma^{i-k}[X^{\rm T}(i)Q_1X(i)+u^{\rm T}(i)Ru(i)]. \tag{16}$$

$$Q(X(k)u(k))=X^{\rm T}(i)Q_1X(i)+u^{\rm T}(i)Ru(i)+V(X(k+1)). \tag{17}$$

给出如下引理,目的是提出非策略Q-学习算法.

引理1　对于系统(3),定义的Q-函数(17)可以表示成如下二次型:

$$Q(X(k),u(k))=\frac{1}{2}\begin{bmatrix}X(k)\\ u(k)\end{bmatrix}^{\rm T}H\begin{bmatrix}X(k)\\ u(k)\end{bmatrix}=\frac{1}{2}\begin{bmatrix}X(k)\\ u(k)\end{bmatrix}^{\rm T}\begin{bmatrix}H_{xx} & H_{xu}\\ H_{ux} & H_{uu}\end{bmatrix}\begin{bmatrix}X(k)\\ u(k)\end{bmatrix}, \tag{18}$$

其中$H>0$.

基于动态规划,得到基于Q-函数的贝尔曼方程为

$$\begin{bmatrix}X(k)\\ u(k)\end{bmatrix}^{\rm T}H\begin{bmatrix}X(k)\\ u(k)\end{bmatrix}=X^{\rm T}(k)Q_1X(k)+u^{\rm T}(k)Ru(k)+\gamma\begin{bmatrix}X(k+1)\\ u(k+1)\end{bmatrix}^{\rm T}H\begin{bmatrix}X(k+1)\\ u(k+1)\end{bmatrix}=X^{\rm T}(k)Q_1X(k)+u^{\rm T}(k)Ru(k)+\gamma X^{\rm T}(k+1)\begin{bmatrix}I\\ K\end{bmatrix}^{\rm T}H\begin{bmatrix}I\\ K\end{bmatrix}X(k+1). \tag{19}$$

根据最优性的必要条件,令$\dfrac{\partial H}{\partial u}=0$,可得最优控制输入

$$u^*(k)=-H_{uu}^{-1}H_{ux}M\bar{z}(k). \tag{20}$$

由式(11)可知

$$K^* = -H_{uu}^{-1}H_{ux}. \tag{21}$$

注3 由于系统模型参数A、B未知,矩阵M也未知,控制器无法计算$u^*(k)$. 不同于文献[23],在下文Q-学习算法中引入矩阵$\bar{H}$,以便获取不依赖模型、完全数据驱动的控制器学习算法.

2.1 策略Q-学习算法设计

由Smith预测器(9)可知,Q-函数可以改写成

$$Q(X(k),u(k)) = \frac{1}{2}\begin{bmatrix}X(k)\\u(k)\end{bmatrix}^{\mathrm{T}} H \begin{bmatrix}X(k)\\u(k)\end{bmatrix} = \frac{1}{2}\begin{bmatrix}M\bar{z}(k)\\u(k)\end{bmatrix}^{\mathrm{T}} H \begin{bmatrix}M\bar{z}(k)\\u(k)\end{bmatrix} = \frac{1}{2}\begin{bmatrix}\bar{z}(k)\\u(k)\end{bmatrix}^{\mathrm{T}} \bar{H} \begin{bmatrix}\bar{z}(k)\\u(k)\end{bmatrix}, \tag{22}$$

其中

$$\bar{H} = \begin{bmatrix}M & 0\\0 & I\end{bmatrix}^{\mathrm{T}} H \begin{bmatrix}M & 0\\0 & I\end{bmatrix}. \tag{23}$$

那么,贝尔曼方程(19)可以改写为

$$\begin{bmatrix}\bar{z}(k)\\u(k)\end{bmatrix}^{\mathrm{T}} \bar{H} \begin{bmatrix}\bar{z}(k)\\u(k)\end{bmatrix} = \bar{z}^{\mathrm{T}}(k)M^{\mathrm{T}}Q_1M\bar{z}(k) + u^{\mathrm{T}}(k)Ru(k) + \gamma\begin{bmatrix}\bar{z}(k+1)\\u(k+1)\end{bmatrix}^{\mathrm{T}} \bar{H} \begin{bmatrix}\bar{z}(k+1)\\u(k+1)\end{bmatrix}. \tag{24}$$

根据最优性必要条件,由$\dfrac{\partial Q(\bar{z}(k),u(k))}{\partial u(k)} = 0$得到

$$u^*(k) = -(\bar{H}_{uu})^{-1}\bar{H}_{uz}\bar{z}(k). \tag{25}$$

定理1 贝尔曼方程(24)有唯一解$\bar{H}$,且式(25)等价于式(20).

证明 假设贝尔曼方程(24)有两个不同的解$\bar{H}$和$\bar{W}$,有

$$H = \Gamma^{-1}\begin{bmatrix}M & 0\\0 & I\end{bmatrix}\bar{H}\begin{bmatrix}M & 0\\0 & I\end{bmatrix}^{\mathrm{T}}\Gamma^{-1}, \tag{26}$$

$$H_2 = \Gamma^{-1}\begin{bmatrix}M & 0\\0 & I\end{bmatrix}\bar{W}\begin{bmatrix}M & 0\\0 & I\end{bmatrix}^{\mathrm{T}}\Gamma^{-1}, \tag{27}$$

其中$\Gamma = \begin{bmatrix}MM^{\mathrm{T}} & 0\\0 & I\end{bmatrix}$. 由于矩阵$M$为行满秩,矩阵$\Gamma$可逆. 由于$\bar{H}\neq\bar{W}$,$H\neq H_1$,那么式(19)存在两个不同解. 然而,对于优化问题(13),贝尔曼方程(19)有唯一的解H,与此产生矛盾. 原假设式(24)有两个不同的解$\bar{H}$和$\bar{W}$不成立,因而式(24)有唯一的解$\bar{H}$.

将式(23)展开,可得

$$\begin{bmatrix}M & 0\\0 & I\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}H_{xx} & H_{xu}\\H_{ux} & H_{ux}\end{bmatrix}\begin{bmatrix}M & 0\\0 & I\end{bmatrix} = \begin{bmatrix}M^{\mathrm{T}}H_{xx}M & M^{\mathrm{T}}H_{xu}\\M^{\mathrm{T}}H_{ux} & H_{uu}\end{bmatrix} = \begin{bmatrix}\bar{H}_{\bar{z}\bar{z}} & \bar{H}_{\bar{z}u}\\\bar{H}_{u\bar{z}} & \bar{H}_{uu}\end{bmatrix}. \tag{28}$$

其中:$\bar{H}_{\bar{z}u}=M^{\mathrm{T}}H_{xu}$,$\bar{H}_{uu}=H_{uu}$. 所以式(25)等价于(20). □

为了求解式(24)中的Q-函数矩阵$\bar{H}$,给出算法1.

算法1 策略Q-学习算法.

Step 1: 初始化. 给定稳定控制器增益K,并设$j=0$,其中j是迭代系数;

Step 2: 通过求解Q-函数矩阵$\bar{H}^{j+1}$进行策略评估:

$$\begin{bmatrix}\bar{z}(k)\\\bar{K}^j\bar{z}(k)\end{bmatrix}^{\mathrm{T}}\bar{H}^{j+1}\begin{bmatrix}\bar{z}(k)\\\bar{K}^j\bar{z}(k)\end{bmatrix} = (y(k)-r(k))^{\mathrm{T}}Q_1(y(k)-r(k)) + (\bar{K}^j\bar{z}(k))^{\mathrm{T}}R(\bar{K}^j\bar{z}(k)) + \gamma\begin{bmatrix}\bar{z}(k+1)\\\bar{K}^j\bar{z}(k+1)\end{bmatrix}^{\mathrm{T}}\bar{H}^{j+1}\begin{bmatrix}\bar{z}(k+1)\\\bar{K}^j\bar{z}(k+1)\end{bmatrix}. \tag{29}$$

Step 3: 策略更新.

$$u^{j+1} = -(\bar{H}_{uu}^{j+1})^{-1}\bar{H}_{u\bar{z}}^{j+1}\bar{z}(k), \tag{30}$$

$$\bar{K}^{j+1} = -(\bar{H}_{uu}^{j+1})^{-1}\bar{H}_{u\bar{z}}^{j+1}. \tag{31}$$

Step 4: 如果$\|K^{j+1}-K^j\| < l$(l是一个很小的正数),则可以停止策略迭代.

注4 为保证激励的可持续性,在算法1中需要在系统中加入探测噪声,这样会引起矩阵$\bar{H}$的偏差,导致最优跟踪控制器增益不准确. 然而,即使加入探测噪声,非策略Q-学习算法也能得到无偏的解. 本文通过研究非策略学习方法,学习最优跟踪控制器$u(k)$,解出无偏的Q-函数矩阵$\bar{H}$. 因此给出非策略Q-学习算法2.

注5 迭代矩阵$\bar{H}^{j+1}$收敛于式(24)中解$\bar{H}$,证明类似文献[9-12],此处略.

2.2 非策略Q-学习算法设计

引入目标控制策略到系统动态中,得到

$$\begin{aligned}&X(k+1) =\\&A_2X(k) + B_2u^j(k) - B_2u^j(k) + B_2u(k) =\\&A_2X(k) + B_2\bar{K}^j\bar{z}(k) + B_2(u(k)-u^j(k)) =\\&(A_2M + B_2\bar{K}^j)\bar{z}(k) + B_2(u(k)-u^j(k)).\end{aligned} \tag{32}$$

其中:$u(k)$是行为控制策略,$u^j(k)$是目标控制策略. 结合式(32),利用(29),有

$$\begin{bmatrix}\bar{z}(k)\\u^j(k)\end{bmatrix}^{\mathrm{T}}\bar{H}^{j+1}\begin{bmatrix}\bar{z}(k)\\u^j(k)\end{bmatrix} = \begin{bmatrix}\bar{z}(k)\\u^j(k)\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}M^{\mathrm{T}}Q_1M & 0\\0 & R\end{bmatrix}\begin{bmatrix}\bar{z}(k)\\u^j(k)\end{bmatrix} +$$

$$\gamma X^{\mathrm{T}}(k+1)\begin{bmatrix}I\\K\end{bmatrix}^{\mathrm{T}}\bar{H}\begin{bmatrix}I\\K\end{bmatrix}X(k+1)=$$
$$\begin{bmatrix}\bar{z}(k)\\u^j(k)\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}M^{\mathrm{T}}Q_1M & 0\\0 & R\end{bmatrix}\begin{bmatrix}\bar{z}(k)\\u^j(k)\end{bmatrix}+$$
$$\gamma\begin{bmatrix}\bar{z}(k)\\u^j(k)\end{bmatrix}^{\mathrm{T}}(A_2M+B_2)^{\mathrm{T}}\begin{bmatrix}I\\K\end{bmatrix}^{\mathrm{T}}\bar{H}\begin{bmatrix}I\\K\end{bmatrix}\times$$
$$(A_2M+B_2)\begin{bmatrix}\bar{z}(k)\\u^j(k)\end{bmatrix}. \tag{33}$$

其中

$$\bar{H}^{j+1}=$$
$$\begin{bmatrix}M^{\mathrm{T}}Q_1M & 0\\0 & R\end{bmatrix}+$$
$$\gamma(A_2M+B_2)^{\mathrm{T}}\begin{bmatrix}I\\K\end{bmatrix}^{\mathrm{T}}\bar{H}\begin{bmatrix}I\\K\end{bmatrix}(A_2M+B_2). \tag{34}$$

进一步整理,可将式(33)写成

$$[\kappa_1\ \ \kappa_2\ \ \kappa_3]\begin{bmatrix}\mathrm{vec}(\bar{H}_{\bar{z}\bar{z}}^{j+1})\\\mathrm{vec}(\bar{H}_{\bar{z}u}^{j+1})\\\mathrm{vec}(\bar{H}_{uu}^{j+1})\end{bmatrix}=\rho^j. \tag{35}$$

其中

$$\kappa_1=\bar{z}^{\mathrm{T}}(k)\otimes\bar{z}^{\mathrm{T}}(k)-\gamma\bar{z}^{\mathrm{T}}(k+1)\otimes\bar{z}^{\mathrm{T}}(k+1),$$
$$\kappa_2=2(\bar{z}^{\mathrm{T}}(k)\otimes(u^j(k)^{\mathrm{T}})-\gamma\bar{z}^{\mathrm{T}}(k+1)\otimes$$
$$(u(k+1))^{\mathrm{T}})+2\bar{z}^{\mathrm{T}}(k)\otimes(u(k)-u^j(k))^{\mathrm{T}},$$
$$\kappa_3=u^{\mathrm{T}}(k)\otimes u^{\mathrm{T}}(k)-\gamma(u^j(k+1)^{\mathrm{T}})\otimes(u^j(k+1)^{\mathrm{T}}).$$

由式(35)中的$\bar{H}_{\bar{z}\bar{z}}^{j+1}$、$\bar{H}_{\bar{z}u}^{j+1}$和$\bar{H}_{uu}^{j+1}$可得控制器迭代增益矩阵

$$\bar{K}^{j+1}=-(\bar{H}_{uu}^{j+1})^{-1}\bar{H}_{u\bar{z}}^{j+1}. \tag{36}$$

算法2　非策略Q-学习算法.

Step 1: 数据收集. 选择可镇定的行为控制策略$u(k)$作用于被控系统,收集系统数据$x(k)$、$r(k)$并将它们储存于样本集$[\kappa_1\ \ \kappa_2\ \ \kappa_3]$和$\rho^j$中.

Step 2: 初始化. 选择一个控制器增益K^0,并设定$j=0$,其中j是迭代系数.

Step 3: 执行Q-学习. 通过使用递归最小二乘(RLS)或批最小二乘(BLS)方法,计算$\mathrm{vec}(\bar{H}_{\bar{z}\bar{z}}^{j+1})$、$\mathrm{vec}(\bar{H}_{\bar{z}u}^{j+1})$和$\mathrm{vec}(\bar{H}_{\bar{z}u}^{j+1})$,并且由式(36)计算$\bar{K}^{j+1}$.

Step 4: 如果$\|K^{j+1}-K^j\|<l$(l是一个很小的正数),则可以停止策略迭代,此时最优控制策略已找到;否则,令$j=j+1$,并重复Step 3.

注6　式(35)迭代矩阵$\bar{H}^{j+1}$等价于(29)中迭代矩阵$\bar{H}^{j+1}$,证明类似文献[9-12]. 由于式(29)中$\bar{H}^{j+1}$收敛于(24)的解$\bar{H}$,则有$\lim\limits_{j\to\infty}u^{j+1}(k)=u^*(k)$.

注7　既然非策略强化学习方法在控制输入加入探测噪声时,仍然保证贝尔曼方程解的无偏性,本文不同于文献[23]采用的策略Q-学习算法,本文给出非策略Q-学习算法学习基于Smith预测器的最优状态反馈控制律.

3　仿真实验

在这一节中,通过仿真验证在发生随机有界丢包情况下非策略Q-学习算法的有效性.

首先,考虑如下开环不稳定系统[6]:

$$x(k+1)=\begin{bmatrix}-1 & 2\\2.2 & 1.7\end{bmatrix}x(k)+\begin{bmatrix}2\\1.6\end{bmatrix}u(k). \tag{37}$$

$$y(k)=[1\ \ 2]x(k). \tag{38}$$

参考信号发生器为

$$r(k+1)=-r(k). \tag{39}$$

选择$Q=6,R=1$并且连续反馈丢包的最大数目为$\delta_{f\max}$=1. 此时,丢包Smith预测器矩阵为

$$M=\begin{bmatrix}I & A & B & 0\\0 & 0 & 0 & I\end{bmatrix}. \tag{40}$$

此时,最优Q-函数矩阵H和最优跟踪控制增益K可以分别从式(18)和(21)中得到.

$$\bar{H}=\begin{bmatrix}153.621\,4 & -91.459\,5 & 37.967\,9 & -106.603\,5\\-91.459\,5 & 596.328\,6 & -47.099\,5 & 566.338\,3\\37.967\,9 & -47.099\,5 & 20.786\,0 & -35.965\,7\\-106.603\,5 & 566.338\,3 & -35.965\,7 & 561.549\,3\end{bmatrix}. \tag{41}$$

$$\bar{K}=[0.189\,8\ \ -1.008\,5\ \ 0.064\,0]. \tag{42}$$

然后执行算法2,经过10次迭代,算法收敛得到最优Q-函数矩阵和最优控制器增益.

$$\bar{H}^{10}=\begin{bmatrix}153.621\,4 & -91.459\,5 & 37.967\,9 & -106.603\,5\\-91.459\,5 & 596.328\,6 & -47.099\,5 & 566.338\,3\\37.967\,9 & -47.099\,5 & 20.786\,0 & -35.965\,7\\-106.603\,5 & 566.338\,3 & -35.965\,7 & 561.549\,3\end{bmatrix}. \tag{43}$$

$$\bar{K}^{10}=[0.189\,8\ \ -1.008\,5\ \ 0.064\,0]. \tag{44}$$

图2和图3分别展示了在学习过程中,$\bar{H}_{\bar{z}u}^{j+1}$、$\bar{H}_{uu}^{j+1}$收敛到最优值$\bar{H}_{\bar{z}u}$、$\bar{H}_{uu}$的过程. 图4和图5分别展示了非策略Q-学习算法的输出跟踪轨迹和控制输入轨迹. 仿真结果表明,在网络最大丢包数为1的情况下,采用本文不依赖模型的具有Smith预测器的状态反馈最优控制,系统跟踪性能较好.

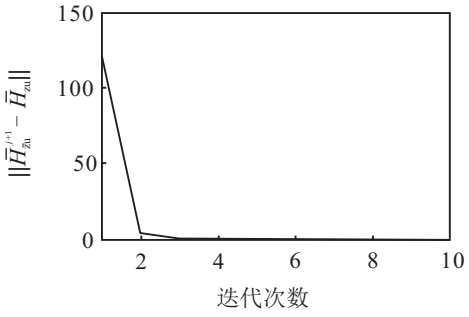

图2 学习过程中$\bar{H}_{\bar{z}u}^{j+1}$收敛到最优值$\bar{H}_{\bar{z}u}$

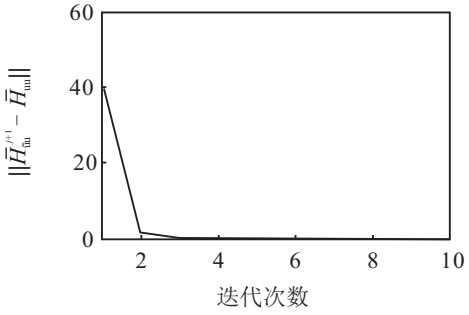

图3 学习过程中$\bar{H}_{\bar{u}u}^{j+1}$收敛到最优值$\bar{H}_{\bar{u}u}$

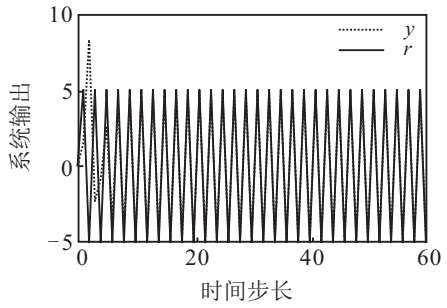

图4 非策略Q-学习算法的输出跟踪轨迹

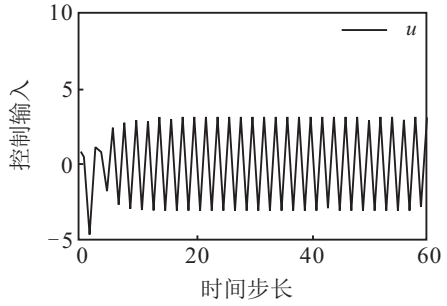

图5 非策略Q-学习算法的控制输入轨迹

图6为最大连续丢包数$\delta_{f\max}=1$时的随机丢包顺序. 接下来考虑最大连续丢包数为$\delta_{f\max}=2$时,执行算法2经过10次迭代得到最优Q-函数矩阵和最优控制器增益.

$$\bar{H}^{10}=\begin{bmatrix}144.938\,9 & -72.595\,0 & 34.323\,2 & -87.763\,6\\ -72.595\,0 & 555.361\,4 & -39.180\,7 & 525.404\,7\\ 30.762\,7 & -31.444\,7 & 17.959\,9 & -20.331\,3\\ -87.763\,6 & 525.404\,7 & -28.057\,2 & 520.669\,0\end{bmatrix}. \tag{45}$$

$$\bar{K}^{10}=[0.109\,4\quad -1.010\,7\quad 0.032\,2]. \tag{46}$$

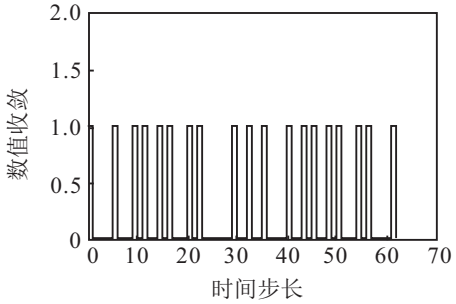

图6 随机丢包顺序

图7～图9分别给出了系统在网络最大丢包数为2时,利用算法2得到的近似最优控制作用下,系统的输出跟踪曲线、控制输入曲线和网络丢包情况.

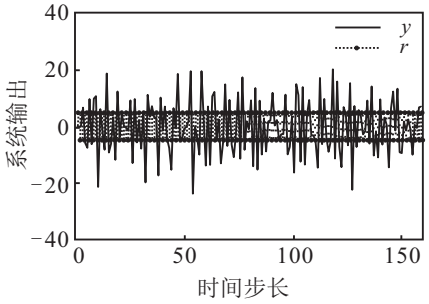

图7 非策略Q-学习算法的输出跟踪轨迹

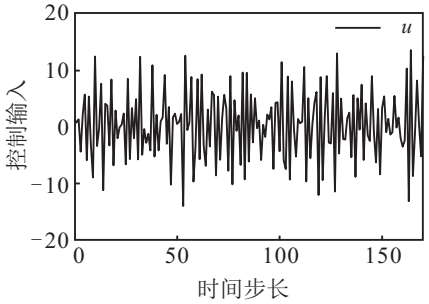

图8 非策略Q-学习算法的控制输入轨迹

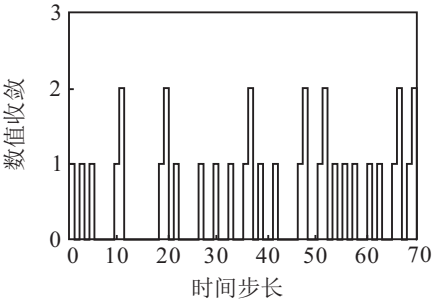

图9 随机丢包顺序

仿真结果表明,在网络最大丢包数为2的情况下,采用本文不依赖模型的具有Smith预测器的状态反馈最优控制,系统输出能够跟踪参考输入,但随着网络性能变差,跟踪性能受到一定程度影响. 可见在反馈控制的被控对象中,对丢包数的容忍范围也是有限的,如果最大连续丢包数$\delta_{f\max}$过大,则系统的稳定性无法保障.

4 结 论

本文针对系统动态未知的网络控制系统跟踪控制问题,提出了一种基于数据驱动的非策略Q-学习方法. 首先,提出了Smith丢包预测器预测系统当前状态,补偿数据丢失对网络控制系统性能的影响;然后,提出了非策略Q-学习算法,此算法可在系统动态未知的情况下,利用可测数据学习最优控制器增益矩阵. 仿真结果表明,该方法对系统动态未知的具有丢包的网络控制系统具有良好的跟踪性能. 未来研究方向是将该方法推广到非线性系统中,或考虑更多网络因素的影响,比如网络时延和数据传输率等.

参考文献(References)

[1] Liu Q, Fu Q M, Gong S R, et al. Reinforcement learning method for mean reward of minimum state variable[J]. Journal of Communications, 2011, 32(1): 66-71.

[2] Sutton R S. Learning to predict by the methods of temporal differences[J]. Machine Learning, 1988, 3(1): 9-44.

[3] Zhang H, Cui X, Luo Y , et al. Finite-horizon H_∞ tracking control for unknown nonlinear systems with saturating actuators[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(4): 1200-1212.

[4] Wang D, Liu D. Learning and guaranteed cost control with event-based adaptive critic implementation[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(12): 6004-6014.

[5] Wang D. Intelligent critic control with robustness guarantee of disturbed nonlinear plants[J]. IEEE Transactions on Cybernetics, DOI: 10.1109/TCYB.2019.2903117.

[6] Kiumarsi B, Lewis F L, Modares H, et al. Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics[J]. Automatica, 2014, 50(4): 1167-1175.

[7] Tsitsiklis J N, Roy B V. An analysis of temporal-difference learning with function approximation[J]. IEEE Transactions on Automatic Control, 2002, 42(5): 674-690.

[8] Wei Q, Liu D, Shi G. A novel dual iterative Q-learning method for optimal battery management in smart residential environments[J]. IEEE Transactions on Industrial Electronics, 2015, 62(4): 2509-2518.

[9] Al-Tamimi A, Lewis F L, Abu-Khalaf M. Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control[J]. Automatica, 2007, 43(3): 473-481.

[10] Kim J H, Lewis F L. Model-free H_∞ control design for unknown linear discrete-time systems via Q-learning with LMI[J]. Automatica, 2010, 46(8): 1320-1326.

[11] Li J, Chai T, Lewis F, et al. Off-policy Q-learning: Set-point design for optimizing dual-rate rougher flotation operational processes[J]. IEEE Transactions on Industrial Electronics, 2018, 65(5): 4092-4102.

[12] Li J, Chai T, Lewis F L, et al. Off-policy interleaved Q-learning: Optimal control for affine nonlinear discrete-time systems[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(5): 1308-1320.

[13] Luo B, Liu D, Huang T, et al. Model-free optimal tracking control via critic-only Q-learning[J]. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(10): 2134-2144.

[14] Zhang W, Branicky M S, Phillips S M. Stability of networked control systems[J]. IEEE Control Systems Magazine, 2001, 21(1): 84-99.

[15] Wang Y L, Han Q L, Peng C. Network-based modelling and dynamic output feedback control for unmanned marine vehicles in network environments[J]. Automatic, 2018, 91(3): 43-53.

[16] Wang Y L, Han Q L, Fei M R, et al. Network-based T-S fuzzy dynamic positioning controller design for unmanned marine vehicles[J]. IEEE Transations on Cybernetics, 2018, 48(9): 2750-2763.

[17] Seiler P, Sengupta R. Analysis of communication losses in vehicle control problems[C]. Proceedings of the American Control Conference. Arlington: IEEE, 2001: 1491-1496.

[18] Azimi-Sadjadi B. Stability of networked control systems in the presence of packet losses[C]. Proceeding of the 42nd IEEE Conference on Decision and Control. Maui: IEEE, 2003: 676-681.

[19] Xiong J, Lam J. Stabilization of linear systems over networks with bounded packet loss[J]. Automatica, 2007, 43(1): 80-87.

[20] Xu H, Sahoo A, Jagannathan S. Stochastic adaptive event-triggered control and network scheduling protocol co-design for distributed networked systems[J]. IET Control Theory and Applications, 2014, 8(18): 2253-2265.

[21] Xu H, Jagannathan S, Lewis F L. Stochastic optimal control of unknown linear networked control system in the presence of random delays and packet losses[J]. Automatica, 2012, 48(6): 1017-1030.

[22] Xu H, Jagannathan S. Stochastic optimal controller design for uncertain nonlinear networked control system via neuro dynamic programming[J]. IEEE Transactions Neural Networks Learning Systems, 2013, 24(3): 471-484.

[23] Jiang Y, Fan J, Chai T, et al. Tracking control for linear discrete-time networked control systems with unknown dynamics and dropout[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018: 29(10): 4607-4620.

作者简介

李金娜(1977−), 女, 教授, 博士, 从事数据驱动控制、运行优化控制、强化学习、网络控制等研究, E-mail: lijinna_721@126.com;

尹子轩(1995−), 男, 硕士生, 从事强化学习、网络控制的研究, E-mail: yinzixuan0305@foxmail.com.

(责任编辑: 闫　妍)