中国科学: 信息科学 2026 年 第 56 卷 第 5 期: 1228–1243
SCIENTIA SINICA Informationis
《中国科学》杂志社
SCIENCE CHINA PRESS
论文

基于模型缩减的输入时滞系统数据驱动控制

孟媛[1], 张颖[1*], 戚文念[2], 张杰[1]

1. 哈尔滨工业大学 (深圳) 智能科学与工程学院, 深圳 518055
2. 哈尔滨工业大学 (深圳) 机器人与先进制造学院, 深圳 518055
* 通信作者. E-mail: zhangyinghit@hit.edu.cn

收稿日期: 2025–02–08; 修回日期: 2025–05–23; 接受日期: 2025–08–21; 网络出版日期: 2026–02–11

国家自然科学基金 (批准号: 62525304, 62188101) 和国家资助博士后研究人员计划 c 档 (批准号: GZC20233461) 资助项目

摘要 为研究系统矩阵和控制输入矩阵未知的离散时间输入时滞系统的镇定问题, 本文提出了一种基于策略迭代的控制算法. 由于被控系统的系统矩阵和输入矩阵未知, 需要收集系统状态和输入数据以构造数据矩阵. 此外, 输入中存在时滞使得控制器不能立刻作用于系统, 可能导致系统不稳定. 为此, 本文采用模型缩减方法将输入时滞系统转化为无时滞系统. 针对此无时滞系统, 利用数据矩阵得到代数 Riccati 方程的解. 进而, 设计控制律使得输入时滞系统稳定. 值得注意的是, 所提出的针对带有输入时滞系统的迭代算法, 无需初始稳定的控制器. 最后, 将所提出的算法应用于两个算例以验证算法的有效性. 仿真结果显示, 被控系统的状态响应曲线收敛, 表明本文所设计的基于数据的控制律可以使得系统矩阵和输入矩阵未知的输入时滞系统稳定.

关键词 自适应动态规划, 数据驱动控制, 策略迭代算法, 离散时间输入时滞系统

1 引言

在过去的几十年中, 研究者们广泛关注各类系统的控制问题, 付出了很多努力为被控系统设计控制律, 并取得了大量的成果. 其中, 很多研究者致力于通过估计代数 Riccati 矩阵方程 (algebraic Riccati matrix equation, ARE) 的唯一正定解来设计控制律 [1,2]. 文献 [1] 设计了两种迭代算法, 通过应用当前步中变量的估计值去得到离散时间马尔可夫 (Markov) 跳跃线性系统的耦合 Lyapunov 矩阵方程的解. 文献 [2] 给出了文献 [1] 的连续时间版本, 根据最近更新的信息求解连续时间马尔可夫跳跃线性系统的耦合 Lyapunov 矩阵方程. 与文献 [1] 相同, 文献 [2] 给出了零初始条件和任意初始条件下的收敛性证明. 受反馈控制中信息传输机制的影响, 输入时滞现象在实际系统中广泛存在, 例如网络控制系统 [3]、多智能体系统 [4]. 因此, 输入时滞系统的控制问题吸引了很多研究者的关注 [5,6]. 文献 [5] 借助变量变换将时滞系统的最优跟踪控制问题等效地转化为无时滞系统的最优跟踪控制问题, 通过求解无时滞系统的 Riccati 和 Stein 方程去得到最优的跟踪控制律. 文献 [6] 通过求解均方前向和后向随机差分方程、修正的 Riccati 矩阵方程解决了输入时滞均方系统的线性二次最优控制和镇定问题. 显然, 上述文献中基于 ARE 解的控制方法, 均需要被控系统的数学模型.

引用格式: 孟媛, 张颖, 戚文念, 等. 基于模型缩减的输入时滞系统数据驱动控制. 中国科学: 信息科学, 2026, 56: 1228–1243, doi: 10.1360/SSI-2025-0049
Meng Y, Zhang Y, Qi W N, et al. Data-driven control for input-delayed systems based on model reduction. Sci Sin Inform, 2026, 56: 1228–1243, doi: 10.1360/SSI-2025-0049

© 2026《中国科学》杂志社

https://www.sciengine.com/doi/10.1360/SSI-2025-0049

然而, 在实际应用中, 被控系统的数学模型往往难以准确获得, 这使得 ARE 难以构建与求解, 进而无法通过直接求解 ARE 得到反馈增益. 为了应对这些困难, 最近发展了一类叫作自适应动态规划 (adaptive dynamic programming, ADP) 的方法去估计反馈增益 [7~10]. 这些算法都是利用收集的系统状态和/或控制输入数据去求解 ARE 的唯一正定解. 文献 [7] 针对系统矩阵未知、输入矩阵已知的连续时间线性系统的控制问题提出了一种迭代算法. 此算法利用系统状态和一个区间段内的代价函数值求解 ARE 的唯一正定解的近似值, 进而更新反馈增益. 文献 [8] 提出的算法消除了文献 [7] 中算法要求输入矩阵已知的限制, 利用系统状态和控制输入数据同时获得 ARE 的唯一解和反馈增益的近似值. 文献 [9] 利用连续时间线性周期系统的系统状态和控制输入数据得到周期 Lyapunov 方程的唯一半正定解, 然后更新反馈系数. 文献 [10] 针对文献 [9] 中的问题, 提出了两种 ADP 算法, 一种算法是在每个迭代步中将最新得到的控制律应用于系统去收集系统状态构建数据矩阵; 另一种算法是将任意的控制律应用于系统去收集系统状态构造固定的数据矩阵. 基于此数据矩阵, 求解周期代数 Riccati 矩阵方程并更新反馈增益. 在这些算法中, 均假设可以找到一个初始稳定的反馈增益. 此假设对于算法的执行是一个苛刻的前提. 为此, 文献 [11,12] 中提出了改进的控制算法. 文献 [11] 针对系统矩阵和输入矩阵未知的离散时间线性系统提出了改进的策略迭代算法. 文献 [12] 针对离散时间动态系统设计了两种策略迭代算法. 基于在线收集的序列数据, 利用定义的加权贝尔曼 (Baermann) 算子和复合贝尔曼算子求解 ARE, 进而更新控制律. 可以看出上述研究中均未对离散时间时滞系统进行研究.

本文对系统矩阵和输入矩阵未知的离散时间输入时滞系统的镇定问题进行研究. 首先收集初始反馈增益下系统的状态和控制输入数据, 然后采用模型缩减方法将输入时滞系统转化为无时滞系统, 进而针对模型缩减后的无时滞系统设计可镇定被控系统的控制律. 此外, 对所提出算法的计算复杂度进行分析. 仿真结果表明, 本文提出的基于策略迭代的数据驱动控制算法得到的控制律能够镇定输入时滞系统. 本文的主要贡献如下.

(1) 与传统基于模型的时滞系统的控制方法相比, 本文提出了一个数据驱动的控制方法. 当输入时滞系统的动态矩阵未知时, 利用系统的状态和输入数据设计控制律.

(2) 与基于状态增广的数据驱动控制方法 [13] 相比, 本文提出的策略迭代算法通过模型缩减方法将输入时滞系统转化为无时滞系统, 从而针对无时滞系统设计 Lyapunov 函数, 有效降低了因输入时滞带来的计算复杂度.

(3) 与一般的策略迭代算法 [8] 不同, 本文提出的数据驱动控制方法不依赖于初始稳定的控制器.

符号: 在本文中, $\otimes$ 表示两个矩阵的克罗内克积 (Kronecker product). $\|\cdot\|$ 代表矩阵的 2 范数. 对于两个整数 $a<b$, 用符号 $\mathbb{I}[a,b]$ 表示集合 $\{a,a+1,\ldots,b\}$. 符号 $\boldsymbol{I}$ 代表恰当维度的单位矩阵.

对于任意矩阵 $C\in\mathbb{R}^{n\times r}$, 其第 j 列向量表示为 c_j, $j\in\mathbb{I}[1,r]$. 然后, 符号 $\mathrm{vec}(C)$ 表示

$$\mathrm{vec}\,(C)=\begin{bmatrix} c_1^{\mathrm{T}} & c_2^{\mathrm{T}} & \cdots & c_r^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}.$$

对于任意的向量 $\vartheta\in\mathbb{R}^n$, $w\in\mathbb{R}^r$ 和任意矩阵 $C\in\mathbb{R}^{n\times r}$, 有下式成立:

$$\vartheta^{\mathrm{T}}Cw=(w\otimes\vartheta)^{\mathrm{T}}\,\mathrm{vec}\,(C)\,. \tag{1}$$

此外, 对于对称矩阵 $X\in\mathbb{R}^{n\times n}$, 令 x_{ij} 表示其第 i 行第 j 列的元素, $i,j\in\mathbb{I}[1,n]$, 符号 $\mathrm{vecs}\,(X)$ 表示为

$$\mathrm{vecs}\,(X)=\begin{bmatrix} x_{11} & \sqrt{2}x_{12} & \cdots & \sqrt{2}x_{1n} & x_{22} & \sqrt{2}x_{23} & \cdots & \sqrt{2}x_{2n} & \cdots & x_{n-1,n-1} & \sqrt{2}x_{n-1,n} & x_{n,n}\end{bmatrix}^{\mathrm{T}}.$$

对于如下向量 $\vartheta\in\mathbb{R}^n$:

$$\vartheta=\begin{bmatrix}\vartheta_1 & \vartheta_2 & \cdots & \vartheta_n\end{bmatrix}^{\mathrm{T}},$$

令符号 $\tilde{\vartheta}$ 表示

$$\tilde{\vartheta}=\begin{bmatrix}\vartheta_1^2 & \sqrt{2}\vartheta_1\vartheta_2 & \cdots & \sqrt{2}\vartheta_1\vartheta_n & \vartheta_2^2 & \sqrt{2}\vartheta_2\vartheta_3 & \cdots & \sqrt{2}\vartheta_2\vartheta_n & \cdots & \vartheta_{n-1}^2 & \sqrt{2}\vartheta_{n-1}\vartheta_n & \vartheta_n^2\end{bmatrix}^{\mathrm{T}}.$$

对于任意对称矩阵 X 和任意具有恰当维度的向量 ϑ, 由文献 [8] 可知

$$\vartheta^{\mathrm{T}}X\vartheta\equiv\tilde{\vartheta}^{\mathrm{T}}\mathrm{vecs}\,(X)\,. \tag{2}$$

https://www.sciengine.com/doi/10.1360/SSI-2025-0049

2　问题描述

考虑下面带有输入时滞的离散时间线性时不变系统:

$$x(t+1) = Ax(t) + Bu(t-h), \tag{3}$$

其中矩阵 $A \in \mathbb{R}^{n\times n}$ 和 $B \in \mathbb{R}^{n\times r}$ 是常数矩阵; $x(t) \in \mathbb{R}^n$ 代表状态变量, $u(t) \in \mathbb{R}^r$ 代表系统输入, h 是输入时滞. 在本文中, h 是已知的整数.

注释1　在实际系统中, 许多情况下时滞是已知整数. 例如, 在网络控制系统中, 当传感器、执行器和控制器通过网络交换数据时, 会发生通信延迟. 这些延迟被建模为输入时滞, 并且可以准确测量 [14,15]. 在搅拌罐操作的建模中 (可以看作典型的化学过程设备单元), 物料运输过程中需要考虑时滞, 其值取决于管道的长度, 并且可以测量 [16]. 在实际中, 时滞通常表现为连续时间下的非整数值. 然而, 在离散时间系统中, 所有变量的演化都是在固定采样周期下以离散步长表示的. 因此, 连续时间中的非整数时滞会被量化为其最接近的整数倍采样步长. 这种近似在工程实践中被广泛接受, 也构成了离散时间系统建模的基础. 在文献 [17~19] 中同样假设系统的输入时滞是已知整数.

针对系统 (3) 的镇定问题, 文献 [20] 推导出了模型缩减方法. 根据文献 [20] 中的结果, 当构造以下变量变换时:

$$\chi_h(t) = A^h x(t) + \sum_{j=t-h}^{t-1} A^{t-j-1} Bu(j), \tag{4}$$

变量 $\chi_h(t)$ 的动态可以描述为下面无输入时滞系统:

$$\chi_h(t+1) = A\chi_h(t) + Bu(t). \tag{5}$$

基于此简化的模型, 文献 [20] 给出了下面的结果.

引理1　如果 K 是无时滞系统 (5) 的任意稳定的反馈增益, 那么, 控制律

$$u(t) = K\left[A^h x(t) + \sum_{j=t-h}^{t-1} A^{t-j-1} Bu(j)\right] \tag{6}$$

可以镇定输入时滞系统 (3).

现有的研究中很多方法可以得到引理 1 中稳定的反馈增益 K. 如果矩阵对 (A, B) 是可稳定的, 则可以利用线性二次调节控制方法. 鉴于此, 输入时滞系统 (3) 的镇定问题可以转化为无时滞系统 (5) 的镇定问题. 具体来说, $P = P^*$ 是关于矩阵 $P = P^{\mathrm{T}} > 0$ 的 ARE 的唯一正定解,

$$P = Q + A^{\mathrm{T}}PA - A^{\mathrm{T}}PB\left(B^{\mathrm{T}}PB + R\right)^{-1} B^{\mathrm{T}}PA, \tag{7}$$

其中 Q 和 R 是正定矩阵, 即 $Q = Q^{\mathrm{T}} > 0$ 和 $R = R^{\mathrm{T}} > 0$. 从文献 [21] 中可知, 相应的反馈增益矩阵 K 为

$$K = -\left(B^{\mathrm{T}}P^*B + R\right)^{-1} B^{\mathrm{T}}P^*A. \tag{8}$$

在实际应用中, 系统矩阵 A 和输入矩阵 B 可能无法准确获得. 对于这种情况, 不能直接构造 ARE (7), 也无法求解该方程的解 P^*. 因此, 不能通过 ARE (7) 得到反馈增益矩阵 (8). 然而, 可以收集被控系统运行时的系统状态和控制输入数据. 因此, 根据收集的数据构建算法以得到控制律. 这就是本文所研究的系统矩阵和输入矩阵未知的输入时滞系统的镇定问题.

https://www.sciengine.com/doi/10.1360/SSI-2025-0049

3　控制律设计

本节目的是在系统矩阵 A 和输入矩阵 B 未知的情况下, 利用收集到的系统状态和控制输入数据, 设计可镇定被控系统的控制律.

文献 [22] 提出了下面以 $P(0)=Q$ 为初始条件的迭代算法去获得 ARE (7) 的唯一正定解:

$$P(k+1)=Q+A^{\mathrm{T}}P(k)A-A^{\mathrm{T}}P(k)B\left(B^{\mathrm{T}}P(k)B+R\right)^{-1}B^{\mathrm{T}}P(k)A. \tag{9}$$

很容易知道, ARE (7) 与模型缩减后的无时滞系统 (5) 有关, 而不是与输入时滞系统 (3) 有关. 因此, 在第 k 迭代步, 针对系统 (5), 考虑如下 Lyapunov 函数:

$$V_k(\chi_h(t))=\chi_h^{\mathrm{T}}(t)P(k)\chi_h(t). \tag{10}$$

在 Lyapunov 函数 $V_k(\chi_h(t))$ 有效的时间间隔内, 由式 (5) 可以得到

$$\chi_h^{\mathrm{T}}(t+1)P(k)\chi_h(t+1)=\chi_h^{\mathrm{T}}(t)A^{\mathrm{T}}P(k)A\chi_h(t)+2u^{\mathrm{T}}(t)B^{\mathrm{T}}P(k)A\chi_h(t)+u^{\mathrm{T}}(t)B^{\mathrm{T}}P(k)Bu(t). \tag{11}$$

首先, 定义下列符号:

$$\mathring{A}=\left[\begin{array}{cccc}A^{h-1} & A^{h-2} & \ldots & A^0\boldsymbol{I}\end{array}\right], \tag{12}$$

$$\mathcal{B}=\mathrm{diag}\left(B,B,\ldots,B\right), \tag{13}$$

$$\mathring{u}(t-h)=\left[\begin{array}{cccc}u^{\mathrm{T}}(t-h) & u^{\mathrm{T}}(t-h+1) & \ldots & u^{\mathrm{T}}(t-1)\end{array}\right]^{\mathrm{T}}. \tag{14}$$

因此, 式 (4) 中的变量变换可以写为

$$\chi_h(t)=A^hx(t)+\mathring{A}\mathcal{B}\mathring{u}(t-h), \tag{15}$$

式 (6) 中的控制律可以表示为

$$u(t)=KA^hx(t)+K\mathring{A}\mathcal{B}\mathring{u}(t-h). \tag{16}$$

利用式 (15) 中的变量变换, 可以进一步推导式 (11) 中的关系为

$$\begin{aligned}\chi_h^{\mathrm{T}}(t+1)P(k)\chi_h(t+1)=&\left[A^hx(t)+\mathring{A}\mathcal{B}\mathring{u}(t-h)\right]^{\mathrm{T}}A^{\mathrm{T}}P(k)A\left[A^hx(t)+\mathring{A}\mathcal{B}\mathring{u}(t-h)\right]\\&+2u^{\mathrm{T}}(t)B^{\mathrm{T}}P(k)A\left[A^hx(t)+\mathring{A}\mathcal{B}\mathring{u}(t-h)\right]+u^{\mathrm{T}}(t)B^{\mathrm{T}}P(k)Bu(t)\\=&x^{\mathrm{T}}(t)\left(A^h\right)^{\mathrm{T}}A^{\mathrm{T}}P(k)AA^hx(t)+2x^{\mathrm{T}}(t)\left(A^h\right)^{\mathrm{T}}A^{\mathrm{T}}P(k)A\mathring{A}\mathcal{B}\mathring{u}(t-h)\\&+\mathring{u}^{\mathrm{T}}(t-h)\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}A^{\mathrm{T}}P(k)A\mathring{A}\mathcal{B}\mathring{u}(t-h)+2u^{\mathrm{T}}(t)B^{\mathrm{T}}P(k)AA^hx(t)\\&+2u^{\mathrm{T}}(t)B^{\mathrm{T}}P(k)A\mathring{A}\mathcal{B}\mathring{u}(t-h)+u^{\mathrm{T}}(t)B^{\mathrm{T}}P(k)Bu(t).\end{aligned} \tag{17}$$

另一方面, 由式 (15) 可以得到

$$\chi_h(t+1)=A^hx(t+1)+\mathring{A}\mathcal{B}\mathring{u}(t+1-h).$$

因此, 函数 $\chi_h^{\mathrm{T}}(t+1)P(k)\chi_h(t+1)$ 可以写为

$$\begin{aligned}\chi_h^{\mathrm{T}}(t+1)P(k)\chi_h(t+1)=&x^{\mathrm{T}}(t+1)\left(A^h\right)^{\mathrm{T}}P(k)A^hx(t+1)+2x^{\mathrm{T}}(t+1)\left(A^h\right)^{\mathrm{T}}P(k)\mathring{A}\mathcal{B}\mathring{u}(t-h+1)\\&+\mathring{u}^{\mathrm{T}}(t-h+1)\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}P(k)\mathring{A}\mathcal{B}\mathring{u}(t-h+1).\end{aligned} \tag{18}$$

由于系统矩阵 A 和输入矩阵 B 未知, 因此在所提出的算法中需要避免使用这两个矩阵. 为此, 定义下面 3 个关于矩阵序列 $\{P(k)\}$ 的矩阵序列 $\{G(k)\}$, $\{F(k)\}$, 和 $\{H(k)\}$:

$$G(k)=B^{\mathrm{T}}P(k)B, \tag{19}$$

https://www.sciengine.com/doi/10.1360/SSI-2025-0049

$$F(k) = \left(A^{h+1}\right)^{\mathrm{T}} P(k) B, \tag{20}$$

$$H(k) = B^{\mathrm{T}} P(k) A \mathring{A} \mathcal{B}. \tag{21}$$

将矩阵序列 (19)∼(21) 代入到式 (17) 中, 可得

$$\begin{aligned}\chi_h^{\mathrm{T}}(t+1) P(k) \chi_h(t+1) = {} & x^{\mathrm{T}}(t)\left(A^{h+1}\right)^{\mathrm{T}} P(k) A^{h+1} x(t) + 2x^{\mathrm{T}}(t)\left(A^{h+1}\right)^{\mathrm{T}} P(k) A \mathring{A} \mathcal{B} \mathring{u}(t-h) \\ & + \mathring{u}^{\mathrm{T}}(t-h) \mathcal{B}^{\mathrm{T}} \mathring{A}^{\mathrm{T}} A^{\mathrm{T}} P(k) A \mathring{A} \mathcal{B} \mathring{u}(t-h) + 2u^{\mathrm{T}}(t) F^{\mathrm{T}}(k) x(t) \\ & + 2u^{\mathrm{T}}(t) H(k) \mathring{u}(t-h) + u^{\mathrm{T}}(t) G(k) u(t).\end{aligned} \tag{22}$$

根据 ARE 的迭代解 $P(k)$, 反馈增益矩阵 K 可以由下面的迭代值近似:

$$K(k) = -\left(B^{\mathrm{T}} P(k) B + R\right)^{-1} B^{\mathrm{T}} P(k) A. \tag{23}$$

基于此, 式 (16) 中的控制律 $u(t)$ 可以被近似为

$$u(t) = K(k) A^h x(t) + K(k) \mathring{A} \mathcal{B} \mathring{u}(t-h). \tag{24}$$

基于式 (19)∼(21) 中的矩阵序列 $\{G(k)\}$, $\{F(k)\}$, 和 $\{H(k)\}$, 式 (24) 中的控制律 $u(t)$ 可以表示为

$$u(t) = -(G(k)+R)^{-1} F^{\mathrm{T}}(k) x(t) - (G(k)+R)^{-1} H(k) \mathring{u}(t-h). \tag{25}$$

由文献 [22] 可知 $\lim_{k\to\infty} P(k) = P^*$, 则

$$\lim_{k\to\infty} K(k) = -\left(B^{\mathrm{T}} P^* B + R\right)^{-1} B^{\mathrm{T}} P^* A.$$

因此, 在第 k 迭代步中, 将式 (24) 中的控制律 $u(t)$ 应用于系统 (3). 此迭代步持续的时间区间表示为 $\mathbb{I}[t_{sk}, t_{fk}]$. 在初始迭代步 $k=0$ 时, 将开环控制信号 $u(t) = u_0(t)$ 应用于系统 (3), 并在区间 $\mathbb{I}[t_{s0}, t_{f0}]$ 内激活此控制律. 在此区间内, 可以计算得到矩阵 $P(1)$ 的值. 对于任意时间 $t \in \mathbb{I}[t_{sk}, t_{fk}]$, 可通过式 (18) 和 (22) 推导出

$$\begin{aligned} & x^{\mathrm{T}}(t+1)\left(A^h\right)^{\mathrm{T}} P(k) A^h x(t+1) + 2x^{\mathrm{T}}(t+1)\left(A^h\right)^{\mathrm{T}} P(k) \mathring{A} \mathcal{B} \mathring{u}(t-h+1) \\ & \quad + \mathring{u}^{\mathrm{T}}(t-h+1) \mathcal{B}^{\mathrm{T}} \mathring{A}^{\mathrm{T}} P(k) \mathring{A} \mathcal{B} \mathring{u}(t-h+1) \\ & = x^{\mathrm{T}}(t)\left(A^{h+1}\right)^{\mathrm{T}} P(k) A^{h+1} x(t) + 2x^{\mathrm{T}}(t)\left(A^{h+1}\right)^{\mathrm{T}} P(k) A \mathring{A} \mathcal{B} \mathring{u}(t-h) \\ & \quad + \mathring{u}^{\mathrm{T}}(t-h) \mathcal{B}^{\mathrm{T}} \mathring{A}^{\mathrm{T}} A^{\mathrm{T}} P(k) A \mathring{A} \mathcal{B} \mathring{u}(t-h) + 2u^{\mathrm{T}}(t) F^{\mathrm{T}}(k) x(t) + 2u^{\mathrm{T}}(t) H(k) \mathring{u}(t-h) + u^{\mathrm{T}}(t) G(k) u(t). \end{aligned} \tag{26}$$

从式 (24) 可知, 在第 k 迭代步中, 应用控制律

$$u(t) = K(k-1) A^h x(t) + K(k-1) \mathring{A} \mathcal{B} \mathring{u}(t-h) \tag{27}$$

于系统 (3). 将控制律 (27) 代入式 (26) 中可以得到

$$\begin{aligned} & x^{\mathrm{T}}(t+1)\left(A^h\right)^{\mathrm{T}} P(k) A^h x(t+1) + 2x^{\mathrm{T}}(t+1)\left(A^h\right)^{\mathrm{T}} P(k) \mathring{A} \mathcal{B} \mathring{u}(t-h+1) \\ & \quad + \mathring{u}^{\mathrm{T}}(t-h+1) \mathcal{B}^{\mathrm{T}} \mathring{A}^{\mathrm{T}} P(k) \mathring{A} \mathcal{B} \mathring{u}(t-h+1) \\ & = x^{\mathrm{T}}(t)\left(A^{h+1}\right)^{\mathrm{T}} P(k) A^{h+1} x(t) + 2x^{\mathrm{T}}(t)\left(A^{h+1}\right)^{\mathrm{T}} P(k) A \mathring{A} \mathcal{B} \mathring{u}(t-h) \\ & \quad + \mathring{u}^{\mathrm{T}}(t-h) \mathcal{B}^{\mathrm{T}} \mathring{A}^{\mathrm{T}} A^{\mathrm{T}} P(k) A \mathring{A} \mathcal{B} \mathring{u}(t-h) + 2\left(K(k-1) A^h x(t) + K(k-1) \mathring{A} \mathcal{B} \mathring{u}(t-h)\right)^{\mathrm{T}} F^{\mathrm{T}}(k) x(t) \\ & \quad + 2\left(K(k-1) A^h x(t) + K(k-1) \mathring{A} \mathcal{B} \mathring{u}(t-h)\right)^{\mathrm{T}} H(k) \mathring{u}(t-h) + u^{\mathrm{T}}(t) G(k) u(t) \\ & = x^{\mathrm{T}}(t)\left(A^{h+1}\right)^{\mathrm{T}} P(k) A^{h+1} x(t) + 2x^{\mathrm{T}}(t)\left(A^{h+1}\right)^{\mathrm{T}} P(k) A \mathring{A} \mathcal{B} \mathring{u}(t-h) \end{aligned}$$

https://www.sciengine.com/doi/10.1360/SSI-2025-0049

$$
\begin{aligned}
&+\mathring{u}^{\mathrm{T}}(t-h)\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}A^{\mathrm{T}}P(k)A\mathring{A}\mathcal{B}\mathring{u}(t-h)+2x^{\mathrm{T}}(t)\left(A^{h}\right)^{\mathrm{T}}K^{\mathrm{T}}(k-1)F^{\mathrm{T}}(k)x(t)\\
&+2\mathring{u}^{\mathrm{T}}(t-h)\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}K^{\mathrm{T}}(k-1)F^{\mathrm{T}}(k)x(t)+2x^{\mathrm{T}}(t)\left(A^{h}\right)^{\mathrm{T}}K^{\mathrm{T}}(k-1)H(k)\mathring{u}(t-h)\\
&+2\mathring{u}^{\mathrm{T}}(t-h)\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}K^{\mathrm{T}}(k-1)H(k)\mathring{u}(t-h)+u^{\mathrm{T}}(t)G(k)u(t).
\end{aligned}\tag{28}
$$

进一步, 式 (28) 中的关系可以等价写为

$$
\begin{aligned}
&x^{\mathrm{T}}(t+1)\left(A^{h}\right)^{\mathrm{T}}P(k)A^{h}x(t+1)+\mathring{u}^{\mathrm{T}}(t-h+1)\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}P(k)\mathring{A}\mathcal{B}\mathring{u}(t-h+1)\\
&\quad=x^{\mathrm{T}}(t)\left(A^{h+1}\right)^{\mathrm{T}}P(k)A^{h+1}x(t)+2x^{\mathrm{T}}(t)\left(A^{h+1}\right)^{\mathrm{T}}P(k)A\mathring{A}\mathcal{B}\mathring{u}(t-h)\\
&\qquad+\mathring{u}^{\mathrm{T}}(t-h)\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}A^{\mathrm{T}}P(k)A\mathring{A}\mathcal{B}\mathring{u}(t-h)+2x^{\mathrm{T}}(t)\left(A^{h}\right)^{\mathrm{T}}K^{\mathrm{T}}(k-1)F^{\mathrm{T}}(k)x(t)\\
&\qquad+4x^{\mathrm{T}}(t)\left(A^{h}\right)^{\mathrm{T}}K^{\mathrm{T}}(k-1)H(k)\mathring{u}(t-h)+2\mathring{u}^{\mathrm{T}}(t-h)\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}K^{\mathrm{T}}(k-1)H(k)\mathring{u}(t-h)\\
&\qquad+u^{\mathrm{T}}(t)G(k)u(t)-2x^{\mathrm{T}}(t+1)\left(A^{h}\right)^{\mathrm{T}}P(k)\mathring{A}\mathcal{B}\mathring{u}(t-h+1).
\end{aligned}\tag{29}
$$

由于 $P(k)$ 是对称矩阵, 根据式 (1) 和 (2) 中的结果, 式 (29) 中的关系可以等价地表示为

$$
\begin{aligned}
&\tilde{x}^{\mathrm{T}}(t+1)\mathrm{vecs}\left(\left(A^{h}\right)^{\mathrm{T}}P(k)A^{h}\right)+\tilde{\mathring{u}}^{\mathrm{T}}(t-h+1)\mathrm{vecs}\left(\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}P(k)\mathring{A}\mathcal{B}\right)\\
&\quad=\tilde{x}^{\mathrm{T}}(t)\mathrm{vecs}\left(\left(A^{h+1}\right)^{\mathrm{T}}P(k)A^{h+1}\right)+2\left(\mathring{u}(t-h)\otimes x(t)\right)^{\mathrm{T}}\mathrm{vec}\left(\left(A^{h+1}\right)^{\mathrm{T}}P(k)A\mathring{A}\mathcal{B}\right)\\
&\qquad+\tilde{\mathring{u}}^{\mathrm{T}}(t-h)\mathrm{vecs}\left(\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}A^{\mathrm{T}}P(k)A\mathring{A}\mathcal{B}\right)+2\left(x(t)\otimes K(k-1)A^{h}x(t)\right)^{\mathrm{T}}\mathrm{vec}\left(F^{\mathrm{T}}(k)\right)\\
&\qquad+4\left(\mathring{u}(t-h)\otimes x(t)\right)^{\mathrm{T}}\mathrm{vec}\left(\left(A^{h}\right)^{\mathrm{T}}K^{\mathrm{T}}(k-1)H(k)\right)\\
&\qquad+2\left(\mathring{u}(t-h)\otimes K(k-1)\mathring{A}\mathcal{B}\mathring{u}(t-h)\right)^{\mathrm{T}}\mathrm{vecs}\left(H(k)\right)+\tilde{u}^{\mathrm{T}}(t)\mathrm{vecs}\left(G(k)\right)\\
&\qquad-2\left(\mathring{u}(t-h+1)\otimes x(t+1)\right)^{\mathrm{T}}\mathrm{vec}\left(\left(A^{h}\right)^{\mathrm{T}}P(k)\mathring{A}\mathcal{B}\right).
\end{aligned}\tag{30}
$$

当 $t\in\mathbb{I}[t_{\mathrm{sk}},t_{\mathrm{fk}}]$ 时, 可以推导出附录 1 中的关系. 在式 (30) 中, 得到矩阵 $\left(A^{h}\right)^{\mathrm{T}}P(k)A^{h}$ 和 $\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}P(k)\mathring{A}\mathcal{B}$ 需要已知系统矩阵 A 和控制输入矩阵 B. 为了在所提出的迭代算法中避免使用系统矩阵 A 和输入矩阵 B, 定义下面和矩阵 $P(k)$ 相关的矩阵:

$$
\varphi(k)=\left(A^{h}\right)^{\mathrm{T}}P(k)A^{h},\quad \psi(k)=\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}P(k)\mathring{A}\mathcal{B},\tag{31}
$$

$$
\mathcal{K}_{\mathrm{A}}(k)=K(k)A^{h},\quad \mathcal{K}_{\mathrm{B}}(k)=K(k)\mathring{A}\mathcal{B},\tag{32}
$$

其中, $\varphi(0)=\left(A^{h}\right)^{\mathrm{T}}QA^{h}$ 和 $\psi(0)=\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}Q\mathring{A}\mathcal{B}$. 进一步, 定义由数据组成的矩阵

$$
\begin{aligned}
S_{\mathrm{xx}}(k)&=\left[\tilde{x}(t_{\mathrm{sk}})\ \tilde{x}(t_{\mathrm{sk}}+1)\ \cdots\ \tilde{x}(t_{\mathrm{fk}})\right]^{\mathrm{T}},\\
S_{\bar{\mathrm{u}}\mathrm{x}}(k)&=\left[\mathring{u}(t_{\mathrm{sk}}-h)\otimes x(t_{\mathrm{sk}})\ \mathring{u}(t_{\mathrm{sk}}-h+1)\otimes x(t_{\mathrm{sk}}+1)\ \cdots\ \mathring{u}(t_{\mathrm{fk}}-h)\otimes x(t_{\mathrm{fk}})\right]^{\mathrm{T}},\\
S_{\bar{\mathrm{u}}\bar{\mathrm{u}}}(k)&=\left[\tilde{\mathring{u}}(t_{\mathrm{sk}}-h)\ \tilde{\mathring{u}}(t_{\mathrm{sk}}-h+1)\ \cdots\ \tilde{\mathring{u}}(t_{\mathrm{fk}}-h)\right]^{\mathrm{T}},\\
\mathring{S}_{\mathrm{xx}}(k)&=\left[x(t_{\mathrm{sk}})\otimes\mathcal{K}_{\mathrm{A}}(k-1)x(t_{\mathrm{sk}})\ x(t_{\mathrm{sk}}+1)\otimes\mathcal{K}_{\mathrm{A}}(k-1)x(t_{\mathrm{sk}}+1)\ \cdots\ x(t_{\mathrm{fk}})\otimes\mathcal{K}_{\mathrm{A}}(k-1)x(t_{\mathrm{fk}})\right]^{\mathrm{T}},\\
\mathring{S}_{\bar{\mathrm{u}}\bar{\mathrm{u}}}(k)&=\left[\mathring{u}(t_{\mathrm{sk}}-h)\otimes\mathcal{K}_{\mathrm{B}}(k-1)\mathring{u}(t_{\mathrm{sk}}-h)\ \mathring{u}(t_{\mathrm{sk}}-h+1)\otimes\mathcal{K}_{\mathrm{B}}(k-1)\mathring{u}(t_{\mathrm{sk}}-h+1)\right.\\
&\qquad\left.\cdots\ \mathring{u}(t_{\mathrm{fk}}-h)\otimes\mathcal{K}_{\mathrm{B}}(k-1)\mathring{u}(t_{\mathrm{fk}}-h)\right]^{\mathrm{T}},\\
S_{\mathrm{uu}}(k)&=\left[\tilde{u}(t_{\mathrm{sk}})\ \tilde{u}(t_{\mathrm{sk}}+1)\ \cdots\ \tilde{u}(t_{\mathrm{fk}})\right]^{\mathrm{T}},\\
\Phi_{\bar{\mathrm{u}}\bar{\mathrm{u}}}(k)&=\left[\tilde{\mathring{u}}^{\mathrm{T}}(t_{\mathrm{sk}}-h+1)\ \tilde{\mathring{u}}^{\mathrm{T}}(t_{\mathrm{sk}}-h+2)\ \cdots\ \tilde{\mathring{u}}^{\mathrm{T}}(t_{\mathrm{fk}}-h+1)\right]^{\mathrm{T}},
\end{aligned}
$$

https://www.sciengine.com/doi/10.1360/SSI-2025-0049

$$\Gamma(k)=\left[\begin{array}{ccccccc} S_{\mathrm{xx}}(k) & 2S_{\bar{\mathrm{u}}\mathrm{x}}(k) & S_{\bar{\mathrm{u}}\bar{\mathrm{u}}}(k) & 2\mathring{S}_{\mathrm{xx}}(k) & 2\mathring{S}_{\bar{\mathrm{u}}\bar{\mathrm{u}}}(k) & S_{\mathrm{uu}}(k) & -2S_{\bar{\mathrm{u}}\mathrm{x}}(k+1) \end{array}\right]. \tag{33}$$

很明显可以看出式 (33) 中的这些矩阵由系统状态、控制输入和滞后的输入构造. 此外, 为了方便论述, 定义符号

$$D_{\mathrm{xu}}(k)=\left(A^{h+1}\right)^{\mathrm{T}} P(k) A\mathring{A}\mathcal{B}+2\left(A^{h}\right)^{\mathrm{T}} K^{\mathrm{T}}(k-1) H(k). \tag{34}$$

根据上述定义的矩阵和符号, 式 (30) 中的表达式可以更紧凑地表示为

$$\begin{aligned} & S_{\mathrm{xx}}(k+1)\operatorname{vecs}(\varphi(k))+\Phi_{\bar{\mathrm{u}}\bar{\mathrm{u}}}(k)\operatorname{vecs}(\psi(k)) \\ & \quad=S_{\mathrm{xx}}(k)\operatorname{vecs}\left(\left(A^{h+1}\right)^{\mathrm{T}} P(k) A^{h+1}\right)+2S_{\bar{\mathrm{u}}\mathrm{x}}(k)\operatorname{vec}(D_{\mathrm{xu}}(k))+S_{\mathrm{uu}}(k)\operatorname{vecs}(G(k)) \\ & \qquad+S_{\bar{\mathrm{u}}\bar{\mathrm{u}}}(k)\operatorname{vecs}\left(\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}A^{\mathrm{T}}P(k) A\mathring{A}\mathcal{B}\right)+2\mathring{S}_{\mathrm{xx}}(k)\operatorname{vec}\left(F^{\mathrm{T}}(k)\right)+2\mathring{S}_{\bar{\mathrm{u}}\bar{\mathrm{u}}}(k)\operatorname{vec}(H(k)) \\ & \qquad-2S_{\bar{\mathrm{u}}\mathrm{x}}(k+1)\operatorname{vec}\left(\left(A^{h}\right)^{\mathrm{T}} P(k)\mathring{A}\mathcal{B}\right). \end{aligned} \tag{35}$$

为了简化表达式 (35), 定义下面未知变量组成的复合矩阵:

$$\begin{aligned} \Upsilon(k)=\Big[&\operatorname{vecs}^{\mathrm{T}}\left(\left(A^{h+1}\right)^{\mathrm{T}} P(k) A^{h+1}\right)\quad \operatorname{vec}^{\mathrm{T}}(D_{\mathrm{xu}}(k))\quad \operatorname{vecs}^{\mathrm{T}}\left(\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}A^{\mathrm{T}}P(k) A\mathring{A}\mathcal{B}\right)\quad \operatorname{vec}^{\mathrm{T}}\left(F^{\mathrm{T}}(k)\right) \\ &\operatorname{vec}^{\mathrm{T}}(H(k))\quad \operatorname{vecs}^{\mathrm{T}}(G(k))\quad \operatorname{vec}^{\mathrm{T}}\left(\left(A^{h}\right)^{\mathrm{T}} P(k)\mathring{A}\mathcal{B}\right)\Big]^{\mathrm{T}}. \end{aligned} \tag{36}$$

基于式 (33) 和 (36), 式 (35) 中的关系可以进一步推导为下面的线性形式:

$$S_{\mathrm{xx}}(k+1)\operatorname{vecs}(\varphi(k))+\Phi_{\bar{\mathrm{u}}\bar{\mathrm{u}}}(k)\operatorname{vecs}(\psi(k))=\Gamma(k)\Upsilon(k). \tag{37}$$

在第 k 迭代步中, 已知矩阵 $\varphi(k)$ 和 $\psi(k)$ 的值. 基于这两个矩阵和式 (33) 中数据构造的矩阵, 可以计算矩阵 $\Upsilon(k)$ 的值. 然而, 矩阵 $\Gamma(k)$ 不是列满秩的. 这导致不能直接得到方程 (37) 的最小二乘解. 为此, 可以采用奇异值分解 (singular value decomposition, SVD)、Moore-Penrose 广义逆或岭回归近似等方法来求解复合矩阵 $\Upsilon(k)$. 具体来说, SVD 方法具有较高的计算复杂度, 不适用于大规模矩阵; Moore-Penrose 广义逆方法虽然能提供最小范数解, 但数值稳定性不如 SVD. 因此, 本文采用岭回归近似方法求解方程 (37). 定义正则化参数 λ 控制正则化项的权重. 然后, 方程 (37) 的解为

$$\Upsilon(k)=\left(\Gamma^{\mathrm{T}}(k)\Gamma(k)+\lambda(k)\boldsymbol{I}\right)^{-1}\Gamma^{\mathrm{T}}(k)\left(S_{\mathrm{xx}}(k+1)\operatorname{vecs}(\varphi(k))+\Phi_{\bar{\mathrm{u}}\bar{\mathrm{u}}}(k)\operatorname{vecs}(\psi(k))\right). \tag{38}$$

我们发现式 (38) 中包含迭代项 $\varphi(k)$ 和 $\psi(k)$. 为了计算 $\varphi(k)$ 在第 k 迭代步的值, 等式 (9) 左乘 $\left(A^{h}\right)^{\mathrm{T}}$, 右乘 A^{h}, 得到

$$\begin{aligned} & \left(A^{h}\right)^{\mathrm{T}} P(k+1) A^{h} \\ & \quad=\left(A^{h}\right)^{\mathrm{T}} Q A^{h}+\left(A^{h+1}\right)^{\mathrm{T}} P(k) A^{h+1}-\left(A^{h+1}\right)^{\mathrm{T}} P(k) B\left(B^{\mathrm{T}} P(k) B+R\right)^{-1} B^{\mathrm{T}} P(k) A^{h+1}. \end{aligned} \tag{39}$$

类似地, 为了计算第 k 迭代步中矩阵 $\psi(k)$ 的值, 将等式 (9) 左乘 $\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}$, 右乘 $\mathring{A}\mathcal{B}$, 得到

$$\begin{aligned} & \mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}P(k+1)\mathring{A}\mathcal{B} \\ & \quad=\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}Q\mathring{A}\mathcal{B}+\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}A^{\mathrm{T}}P(k) A\mathring{A}\mathcal{B}-\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}A^{\mathrm{T}}P(k) B\left(B^{\mathrm{T}}P(k) B+R\right)^{-1} B^{\mathrm{T}}P(k) A\mathring{A}\mathcal{B}. \end{aligned} \tag{40}$$

利用式 (19)~(21) 和 (31) 中的定义, 式 (39) 可以等价地写为

$$\varphi(k+1)=\varphi(0)+\left(A^{h+1}\right)^{\mathrm{T}} P(k) A^{h+1}-F(k)(G(k)+R)^{-1} F^{\mathrm{T}}(k). \tag{41}$$

类似地, 式 (40) 可以转化为如下形式:

$$\psi(k+1)=\psi(0)+\mathcal{B}^{\mathrm{T}}\mathring{A}^{\mathrm{T}}A^{\mathrm{T}}P(k) A\mathring{A}\mathcal{B}-H^{\mathrm{T}}(k)(G(k)+R)^{-1} H(k). \tag{42}$$

https://www.sciengine.com/doi/10.1360/SSI-2025-0049

很明显可以看出, 我们可以分别从初始值 $\varphi(0)$ 和 $\psi(0)$ 开始迭代求解方程 (41) 和 (42). 将初始值 $\varphi(0)$, $\psi(0)$ 和任意的增益矩阵 $\mathcal{K}_{\mathrm{A}}(0)$, $\mathcal{K}_{\mathrm{B}}(0)$ 下的数据矩阵 $\Gamma(0)$ 和 $\Phi_{\bar{u}\bar{u}}(0)$ 代入到方程 (38) 中, 可以计算得到复合矩阵 $\Upsilon(0)$. 其中, 复合矩阵 $\Upsilon(0)$ 中包含了 $\left(A^{h+1}\right)^{\mathrm{T}} P(0) A^{h+1}$, $\mathcal{B}^{\mathrm{T}} \mathring{A}^{\mathrm{T}} A^{\mathrm{T}} P(0) A \mathring{A} \mathcal{B}$, $F(0)$, $H(0)$ 和 $G(0)$. 根据这些矩阵, 可以通过式 (41) 和 (42) 更新得到矩阵 $\varphi(1)$ 和 $\psi(1)$ 的值. 根据上述分析, 如果求得第 k 迭代步中矩阵 $\varphi(k)$ 和 $\psi(k)$ 的值, 那么可以通过式 (38) 计算矩阵 $\Upsilon(k)$ 中的 $\left(A^{h+1}\right)^{\mathrm{T}} P(k) A^{h+1}$, $\mathcal{B}^{\mathrm{T}} \mathring{A}^{\mathrm{T}} A^{\mathrm{T}} P(k) A \mathring{A} \mathcal{B}$, $F(k)$, $H(k)$ 和 $G(k)$. 进而, 利用式 (41) 和 (42) 更新关于矩阵 $P(k+1)$ 的对称矩阵 $\varphi(k+1)$ 和 $\psi(k+1)$. 基于这两个对称矩阵, 类似地执行下一次迭代.

得到复合矩阵 $\Upsilon(k)$ 后, 利用式 (25) 更新控制律 $u(t)$, 并将其在下一迭代步中作用于系统 (3). 算法 1 的伪代码给出了算法主要结构的详细描述. 我们注意到迭代算法中没有使用系统矩阵 A 和输入矩阵 B. 此外, 本文所提出的迭代算法基于反馈增益. 因此, 该算法属于策略迭代算法.

算法 1 基于策略迭代的输入时滞系统的控制算法.

输入: 将任意增益矩阵 $\mathcal{K}_{\mathrm{A}}(0)$ 和 $\mathcal{K}_{\mathrm{B}}(0)$ 应用于系统 (3). 选择初始值 $\varphi(0)$, $\psi(0)$, $k=0$ 以及一个极小值 $\varepsilon>0$;
输出: 式 (25) 中的控制律 $u(t)$;
在线收集数据以构造矩阵 $\Gamma(0)$ 和 $\Phi_{\bar{u}\bar{u}}(0)$;
循环
将矩阵 $\varphi(k)$ 和 $\psi(k)$ 代入式 (38) 中, 计算得到 $\left(A^{h+1}\right)^{\mathrm{T}} P(k) A^{h+1}$, $\mathcal{B}^{\mathrm{T}} \mathring{A}^{\mathrm{T}} A^{\mathrm{T}} P(k) A \mathring{A} \mathcal{B}$, $F(k)$, $H(k)$ 和 $G(k)$;
根据式 (41) 和 (42), 更新得到矩阵 $\varphi(k)$ 和 $\psi(k)$;
根据式 (25) 推导控制律 $u(t)$;
在控制律 $u(t)$ 作用下, 收集状态和输入数据;
更新矩阵 $\Gamma(k+1)$ 和 $\Phi_{\bar{u}\bar{u}}(k+1)$;
$k \leftarrow k+1$;
直到 $\|\varphi(k)-\varphi(k-1)\| \leqslant \varepsilon$.

将第 k 迭代步中估计的控制律在第 $(k+1)$ 迭代步中应用于系统 (3) 去收集系统的状态和控制输入数据, 利用这些收集的数据去更新矩阵 $S_{\mathrm{xx}}(k+1)$, $\Phi_{\bar{u}\bar{u}}(k)$ 和 $\Gamma(k)$. 由此可知, 本文所提出的迭代算法是一个闭环控制算法.

关于本文所提出的算法的初始反馈增益矩阵, 给出以下注释.

注释2　本文所提出的算法从任意的初始反馈增益矩阵 $\mathcal{K}_{\mathrm{A}}(0)$ 和 $\mathcal{K}_{\mathrm{B}}(0)$ 开始, 去收集系统 (3) 相应的状态. 基于这些在线收集的数据, 构造矩阵 $S_{\mathrm{xx}}(1)$, $\Phi_{\bar{u}\bar{u}}(0)$ 和 $\Gamma(0)$.

接下来, 给出 ARE (7) 的解 $P(k)$ 的有界性和单调性.

引理2 (参见文献 [22])　对于初始条件为 $P(0)=Q$ 的迭代算法 (9) 得到的序列 $\{P(k)\}$, 始终有

$$P(k) \leqslant P(k+1) \leqslant P^{*},\ k \geqslant 0. \tag{43}$$

也就是说, 矩阵序列 $P(k)$ 是单调非减且有界的.

下面, 给出关于矩阵 $P(k)$ 和 $K(k)$ 的收敛性.

引理3 (参见文献 [23])　对于离散时间线性系统 (5), 在控制律 $u(t)=K\chi_h(t)$ 的作用下, 始终有

$$\lim_{k\to\infty} P(k)=P^{*},\quad \lim_{k\to\infty} K(k)=-\left(B^{\mathrm{T}} P^{*} B+R\right)^{-1} B^{\mathrm{T}} P^{*} A,$$

其中 P^{*} 是 ARE (7) 的唯一正定解.

根据引理 3, 我们可以得到如下定理.

定理1　对于离散时间线性系统 (3), 在本文所设计的迭代算法得到的控制律 $u(t)$ 的作用下, 有

$$\lim_{k\to\infty}\varphi(k)=\left(A^{h}\right)^{\mathrm{T}} P^{*} A^{h},\quad \lim_{k\to\infty}\psi(k)=\mathcal{B}^{\mathrm{T}} \mathring{A}^{\mathrm{T}} P^{*} \mathring{A} \mathcal{B},\quad \lim_{k\to\infty} G(k)=B^{\mathrm{T}} P^{*} B,$$
$$\lim_{k\to\infty} F(k)=\left(A^{h+1}\right)^{\mathrm{T}} P^{*} B,\quad \lim_{k\to\infty} H(k)=B^{\mathrm{T}} P^{*} A \mathring{A} \mathcal{B}.$$

https://www.sciengine.com/doi/10.1360/SSI-2025-0049

在所提出的算法中, 在区间 $\mathbb{I}[t_{\mathrm{s}k}, t_{\mathrm{f}k}]$ 内将控制律 $u(t)$ 应用于系统 (3) 去收集系统状态和控制输入数据. 利用这些数据的全部或者部分去构建矩阵 $S_{\mathrm{xx}}(k+1)$, $\Phi_{\bar{\mathrm{u}}\bar{\mathrm{u}}}(k)$, 和 $\Gamma(k)$. 所有区间 $\mathbb{I}[t_{\mathrm{s}k}, t_{\mathrm{f}k}]$ 的长度可以彼此相同, 也可以彼此不同.

接下来, 采用文献 [24] 中的方法分析本文所提出算法的计算复杂度, 其中, 加法、减法和乘法运算的计算复杂度均为一个浮点运算 (flop). 对于矩阵 $W \in \mathbb{R}^{N\times N}$, 转置运算 W^{T} 和求逆运算 W^{-1} 的计算复杂度分别为 0 和 N^3+N^2+N 个 flops. 定义 $v_k = t_{\mathrm{f}k} - t_{\mathrm{s}k} + 1$. 式 (38) 的计算复杂度为

$$E^3 + 4E^2 v_k + 2E^2 - E + (1+n)nv_k + (1+hr)hrv_k + Ev_k^2 - v_k,$$

其中

$$E = \frac{(1+n)n}{2} + 2hrn + \frac{(1+hr)hr}{2} + nr + hr^2 + \frac{(1+r)r}{2}.$$

当 $k \geqslant 1$ 时, 式 (41) 和 (42) 的计算复杂度分别为

$$r^3 + 2r^2 + n^2 + 2nr^2 + 2n^2 r - nr,$$

$$r^3 + 2r^2 + 2hr^3 + 2h^2 r^3 + h^2 r^2 - hr^2.$$

注释3　与文献 [25~27] 中针对输入时滞系统提出的模型缩减方法相比, 文献 [20] 中提出的模型缩减方法不要求系统矩阵 A 可逆. 然而, 文献 [20] 中的方法基于已知的系统矩阵 A 和输入矩阵 B 去设计控制律. 本文针对输入时滞系统提出了一种基于数据的控制律设计方法, 该方法既不要求系统矩阵 A 可逆, 又不需要矩阵 A 和 B 已知.

注释4　本文所考虑的输入时滞系统未包含额外的约束条件. 根据引理 1 可知, 若 K 是模型缩减后的无时滞系统 (5) 的任意稳定的反馈增益, 那么控制律 (6) 可以镇定原输入时滞系统 (3). 然而, 控制律 (6) 中包含了系统矩阵和控制输入矩阵. 为了避免使用系统矩阵和控制输入矩阵, 本文提出了控制律 (6) 的数据驱动形式 (25). 也就是说, 控制律 (25) 与 (6) 是等价的. 因此, 控制律 (25) 可以镇定带有输入时滞的系统, 保证了离散空间中时滞系统的稳定性.

4　算例

4.1　算例 1

为了验证所提出的数据驱动控制算法的有效性, 选取文献 [28] 中 F-16 飞机装置中存在通信时滞的情况在 MATLAB 平台上进行数值仿真. 具体来说, 系统状态定义为 $x = \begin{bmatrix} \alpha & q & \delta_e \end{bmatrix}^{\mathrm{T}}$, 其中 α 为迎角 (deg), q 为俯仰角速度 (deg/s), δ_e 为升降舵偏角 (deg). 系统输入表示升降舵执行器电压. 将连续时间的 F-16 飞机装置系统按照 $T_{\mathrm{s}} = 0.1$ s 的采样间隔进行离散化, 离散化系统的参数为

$$A = \begin{bmatrix} 0.9065 & 0.0816 & -0.0009 \\ 0.0741 & 0.9012 & -0.0159 \\ 0 & 0 & 0.9048 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & -0.0008 & 0.0952 \end{bmatrix}^{\mathrm{T}},$$

输入时滞 $h = 3$. 此外, 方程 (41) 和 (42) 中的初始矩阵 $\varphi(0)$, $\psi(0)$ 和 ARE (7) 中的 R 分别为

$$\varphi(0) = \begin{bmatrix} 1.5267 & 2.9672 & 0.1646 \\ 2.9672 & 5.3680 & 0.2012 \\ 0.1646 & 0.2012 & 0.0784 \end{bmatrix}, \quad \psi(0) = \begin{bmatrix} 0.0009 & 0.0011 & 0.0012 \\ 0.0011 & 0.0013 & 0.0015 \\ 0.0012 & 0.0015 & 0.0017 \end{bmatrix}, \quad R = 0.55 \times 10^{-2}.$$

https://www.sciengine.com/doi/10.1360/SSI-2025-0049

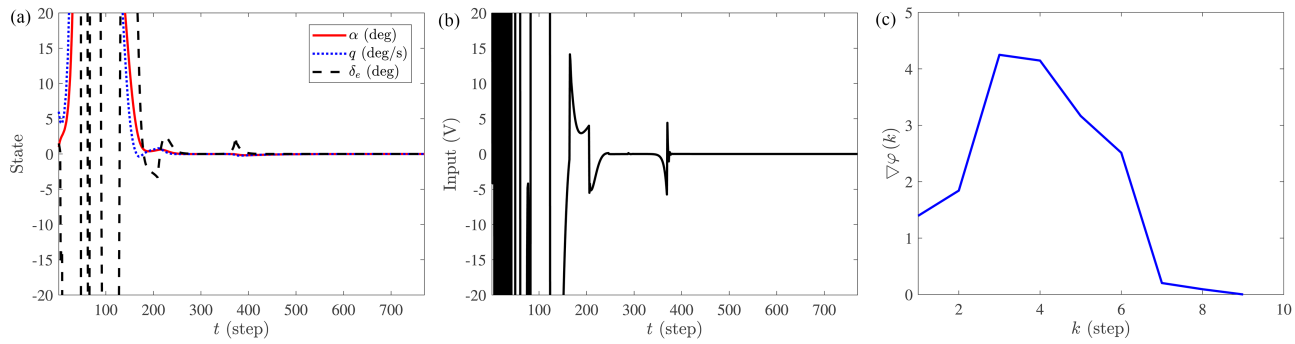

图 1　(网络版彩图) 在算法 1 得到的控制律下算例 1 中的系统的响应与梯度变化曲线. (a) 系统的状态响应曲线; (b) 系统的控制输入曲线; (c) $\varphi(k)$ 的梯度变化曲线.

Figure 1　(Color online) The response and gradient variation curves of the system in example 1 under the controller obtained by Algorithm 1. (a) State response of the system; (b) control input of the system; (c) gradient evolution of $\varphi(k)$.

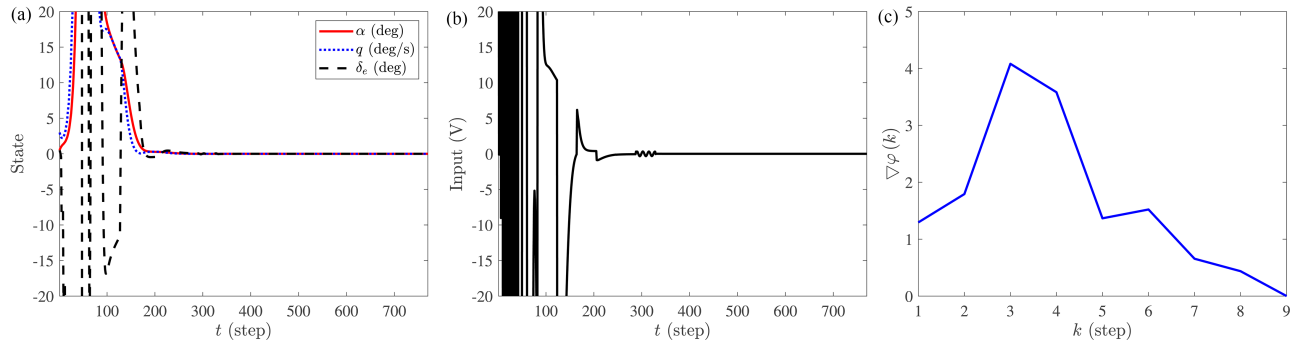

图 2　(网络版彩图) 初始状态 $x(0)=[0.5\ 3\ 0.5]^{\mathrm{T}}$ (deg, deg/s, deg) 下算法 1 得到的控制律下算例 1 中的系统的响应与梯度变化曲线. (a) 系统的状态响应曲线; (b) 系统的控制输入曲线; (c) $\varphi(k)$ 的梯度变化曲线.

Figure 2　(Color online) The response and gradient variation curves of the system in example 1 under the controller obtained by Algorithm 1 with the initial state $x(0)=[0.5\ 3\ 0.5]^{\mathrm{T}}$ (deg, deg/s, deg). (a) State response of the system; (b) control input of the system; (c) gradient evolution of $\varphi(k)$.

在此算例中, 从初始状态 $x(0)=\begin{bmatrix}1 & 6 & 1.5\end{bmatrix}^{\mathrm{T}}$ (deg, deg/s, deg) 开始, 将初始控制输入

$$u_0(t)=\begin{bmatrix}26.8917 & 27.3034 & -9.1387\end{bmatrix}x(t)+\begin{bmatrix}-0.9347 & -0.9810 & -1.0313\end{bmatrix}\mathring{u}(t-h)$$

(单位: V) 作用于系统去收集系统的状态和控制输入数据. 在第 $k>0$ 迭代步中, 将本文所提出的算法生成的控制律 $u(t)$ 应用于系统. 对于每个区间 $\mathbb{I}[t_{sk},t_{fk}]$, 利用 41 对数据计算控制增益, 即 $t_{fk}=t_{sk}+40$. 基于矩阵 $-(G(k)+R)^{-1}F^{\mathrm{T}}(k)$ 和 $-(G(k)+R)^{-1}H(k)$ 去收集数据 $\{x(t),u(t)\}$, $t\in\mathbb{I}[t_{sk},t_{fk}]$, 进而估计矩阵 G^*, F^*, 和 H^* 的迭代值. 当迭代算法的精度为 $\varepsilon=10^{-3}$ 时, 系统的状态响应和控制输入曲线分别如图 1(a) 和 (b) 所示. 此外, 定义 $\varphi(k)$ 的梯度为 $\triangledown\varphi(k)=\|\varphi(k)-\varphi(k-1)\|$. $\varphi(k)$ 的梯度如图 1(c) 所示. 从图 1 可以看出, 在每个迭代步中均使用了 41 对数据, 经过 9 个迭代步后, 矩阵 P 收敛至期望精度. 因此, 共计使用 369 对数据以更新并获得满足精度的控制律. 在该控制律的作用下, 系统进一步运行了 400 步. 从图 1 中可以得出结论, 本算例中的系统在控制律 $u(t)$ 的作用下是稳定的. 此外, 仿真时间为 0.0688 s.

为了验证系统初始状态对所提出的控制算法性能的影响, 选取了不同的初始状态进行对比分析, 其他参数保持不变. 从初始状态 $x(0)=\begin{bmatrix}0.5 & 3 & 0.5\end{bmatrix}^{\mathrm{T}}$ (deg, deg/s, deg) 开始, 系统的状态响应和控制输入曲线分别如图 2(a) 和 (b) 所示. 此外, $\varphi(k)$ 的梯度如图 2(c) 所示. 从图 2 可以看出, 系统在所设计的控制律 $u(t)$ 作用下是稳定的. 进一步, 与图 1 中的结果进行对比可知, 在不同初始状态下, 系统在本文中设计的控制律作用下均是稳定的. 结果表明, 在本文所提方法得到的控制律作用下, 初始状态只会影响系统收敛过程中的动态, 并不会影响系

https://www.sciengine.com/doi/10.1360/SSI-2025-0049

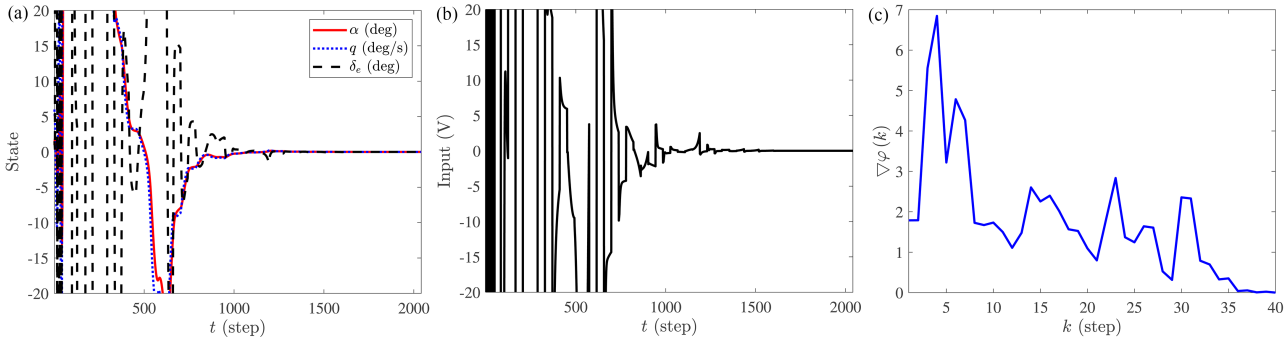

图 3　(网络版彩图) 具有不同初始控制输入的算法 1 得到的控制律下算例 1 中的系统的响应与梯度变化曲线. (a) 系统的状态响应曲线; (b) 系统的控制输入曲线; (c) $\varphi(k)$ 的梯度变化曲线.

Figure 3　(Color online) The response and gradient variation curves of the system in example 1 under the controller obtained by Algorithm 1 with different initial control inputs. (a) State response of the system; (b) control input of the system; (c) gradient evolution of $\varphi(k)$.

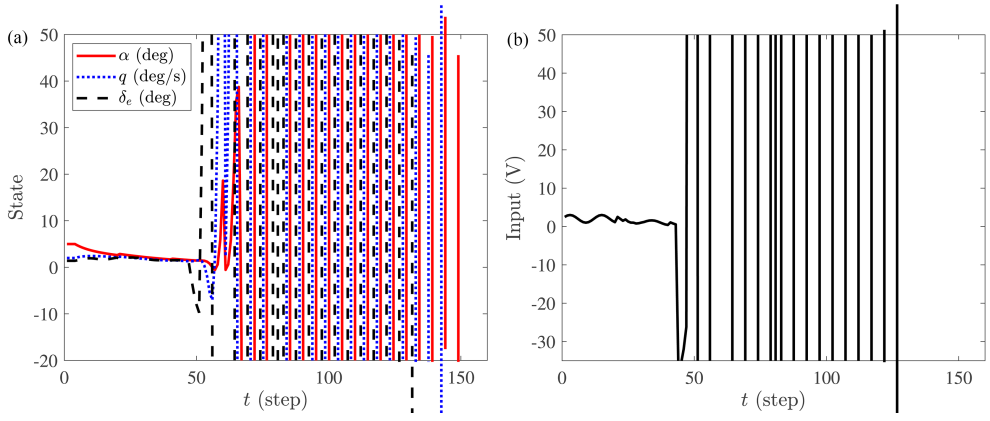

图 4　(网络版彩图) 在文献 [23] 中算法得到的控制律下算例 1 的系统响应曲线. (a) 系统状态响应曲线; (b) 系统控制输入曲线.

Figure 4　(Color online) The response curves of the system in example 1 under the controller in [23]. (a) State response of the system; (b) control input of the system.

统的收敛性. 此外, 仿真时间为 0.0612 s.

为了验证系统初始反馈增益矩阵对所提出的控制算法性能的影响, 选取了不同的反馈增益矩阵进行对比分析, 其他参数保持不变. 在 $k=0$ 时, 将初始控制输入

$$u_0(t)=\begin{bmatrix}-161.8482 & -164.0349 & -3.6645\end{bmatrix}x(t)+\begin{bmatrix}-0.5470 & -0.9168 & -1.3313\end{bmatrix}\mathring{u}(t-h)$$

(单位: V) 作用于系统去收集状态和输入数据. 在后续的迭代过程中, 将本文设计的控制律 $u(t)$ 应用于系统. 系统在该控制律作用下的状态响应和控制输入曲线分别如图 3(a) 和 (b) 所示. 此外, 序列 $\varphi(k)$ 的梯度如图 3(c) 所示. 从图 3 可以看出, 在选取不同初始反馈增益矩阵的情况下, 系统在控制律 $u(t)$ 的作用下仍然能够保持稳定. 仿真时间为 0.1255 s.

为了验证本文提出的数据驱动控制算法的有效性, 将文献 [23] 中针对无时滞系统提出的数据驱动控制算法应用于输入时滞系统 (3). 图 4(a) 展示了在文献 [23] 算法得到的控制律下, 输入时滞系统的状态响应曲线; 图 4(b) 则展示了系统的控制输入曲线. 从图 4 中可以明显观察到, 系统的状态和控制输入出现了发散现象. 与之相比, 本文提出的数据驱动控制算法是有效的.

https://www.sciengine.com/doi/10.1360/SSI-2025-0049

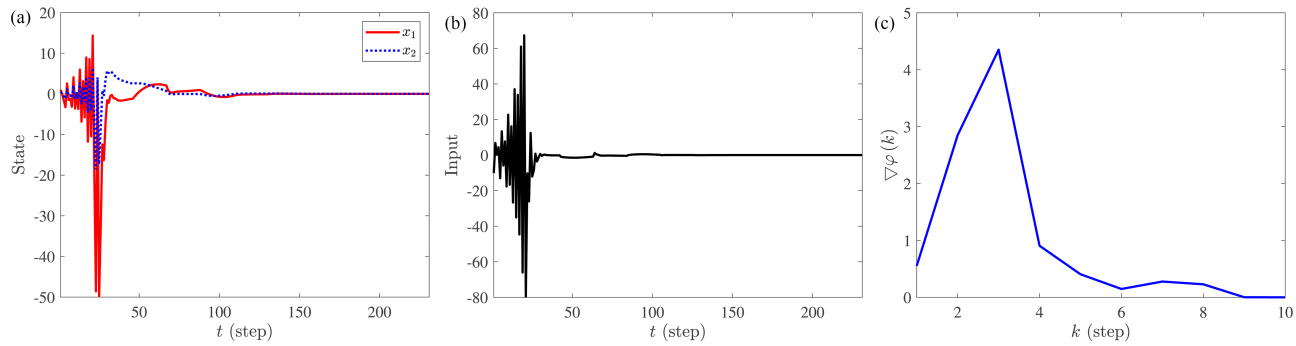

图 5　(网络版彩图) 在算法 1 得到的控制律下算例 2 中的系统的响应与梯度变化曲线. (a) 系统的状态响应曲线; (b) 系统的控制输入曲线; (c) $\varphi(k)$ 的梯度变化曲线.

Figure 5　(Color online) The response and gradient variation curves of the system in example 2 under the controller obtained by Algorithm 1. (a) State response of the system; (b) control input of the system; (c) gradient evolution of $\varphi(k)$.

4.2　算例 2

考虑一个形式为式 (3) 带有输入时滞的离散时间系统. 此系统的参数如下所示:

$$A=\begin{bmatrix}0.8 & -0.1\\ -0.1 & 0.9\end{bmatrix},\quad B=\begin{bmatrix}0.5 & 0.2\end{bmatrix}^{\mathrm{T}},$$

输入时滞 $h=3$. 此外, 方程 (41), (42) 和 ARE (7) 中的矩阵 $\varphi(0)$, $\psi(0)$ 和 R 的值分别为

$$\varphi(0)=\begin{bmatrix}-0.0735 & -0.0893\\ -0.0893 & 3.6267\end{bmatrix},\quad \psi(0)=\begin{bmatrix}0.2501 & 0.3662 & 0.5245\\ 0.3662 & 0.5356 & 0.7667\\ 0.5245 & 0.7667 & 1.0971\end{bmatrix},\quad R=0.55.$$

在此算例中, 从初始状态 $x(0)=\begin{bmatrix}1 & 0.5\end{bmatrix}^{\mathrm{T}}$ 开始, 本算例中所考虑的系统在初始控制输入

$$u_0(t)=\begin{bmatrix}-1.3466 & 0.6063\end{bmatrix}x(t)+\begin{bmatrix}-0.7121 & -0.9275 & -1.2178\end{bmatrix}\mathring{u}(t-h)$$

的作用下去收集系统的状态和控制输入数据. 在随后的迭代步中, 将本文所提出的算法生成的控制律 $u(t)$ 应用于系统. 对于每个区间 $\mathbb{I}[t_{\mathrm{s}k},t_{\mathrm{f}k}]$, 利用 21 对数据计算控制增益, 即 $t_{\mathrm{f}k}=t_{\mathrm{s}k}+20$. 当迭代算法的精度为 $\varepsilon=10^{-3}$ 时, 系统的状态响应和控制输入曲线分别如图 5(a) 和 (b) 所示. $\varphi(k)$ 的梯度如图 5(c) 所示. 在图 5(a) 中, $x_i, i=1,2$ 代表状态 x 的第 i 个分量. 从图 5 可以看出, 在每个迭代步中均使用了 21 对数据, 经过 10 个迭代步后, 矩阵 P 收敛至期望精度. 因此, 共计使用 210 对数据以更新并获得满足精度的控制律. 在该控制律的作用下, 系统进一步运行了 21 步. 从图 5 中可以得出结论, 本算例中的系统在控制律 $u(t)$ 的作用下是稳定的. 此外, 仿真时间为 0.0488 s.

为了验证系统初始状态对所提出的控制算法性能的影响, 选取了不同的初始状态进行对比分析, 其他参数保持不变. 从初始状态 $x(0)=\begin{bmatrix}2 & 5\end{bmatrix}^{\mathrm{T}}$ 开始, 系统的状态响应和控制输入曲线分别如图 6(a) 和 (b) 所示. 此外, $\varphi(k)$ 的梯度如图 6(c) 所示. 从图 6 可以看出, 系统在所设计的控制律 $u(t)$ 作用下是稳定的. 此结果与图 5 中的结果比较可知, 在本文所提出的方法得到的控制律的作用下, 初始状态只会影响系统收敛过程中的动态, 并不会影响系统的收敛性. 仿真时间为 0.0509 s.

为了验证系统初始反馈增益矩阵对所提出的控制算法性能的影响, 选取如下初始控制输入:

$$u_0(t)=\begin{bmatrix}10.3308 & -23.1095\end{bmatrix}x(t)+\begin{bmatrix}0.0035 & -0.7572 & -1.8178\end{bmatrix}\mathring{u}(t-h),$$

https://www.sciengine.com/doi/10.1360/SSI-2025-0049

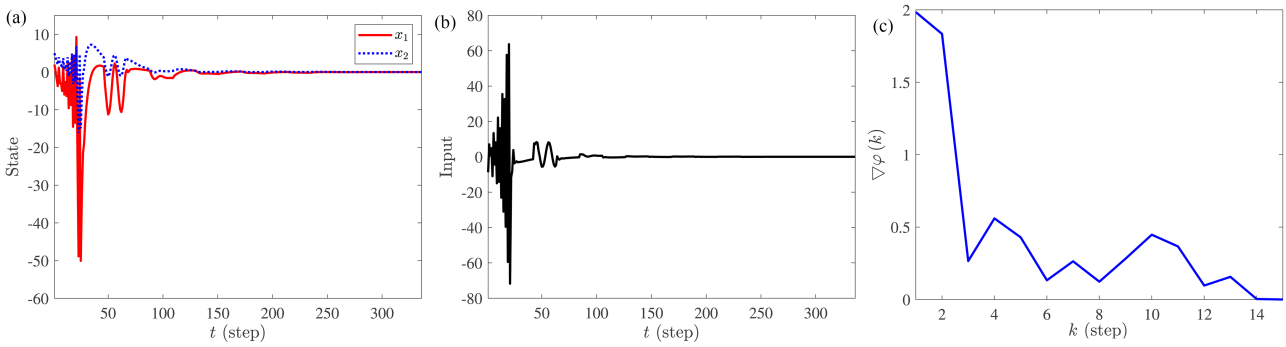

图 6　(网络版彩图) 初始状态 $x(0) = [2\ 5]^{\mathrm{T}}$ 下算法 1 得到的控制律下算例 2 中的系统的响应与梯度变化曲线. (a) 系统的状态响应曲线; (b) 系统的控制输入曲线; (c) $\varphi(k)$ 的梯度变化曲线.

Figure 6　(Color online) The response and gradient variation curves of the system in example 2 under the controller obtained by Algorithm 1 with the initial state $x(0) = [2\ 5]^{\mathrm{T}}$. (a) State response of the system; (b) control input of the system; (c) gradient evolution of $\varphi(k)$.

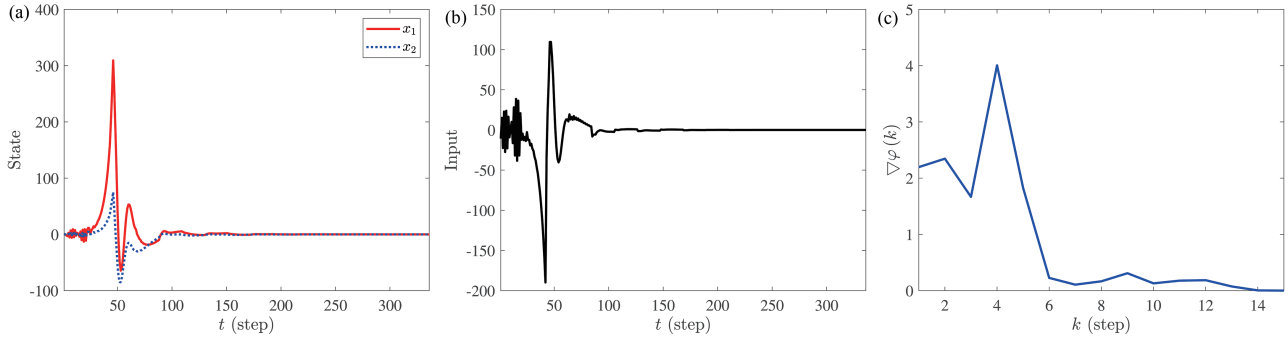

图 7　(网络版彩图) 具有不同初始控制输入的算法 1 得到的控制律下算例 2 中的系统的响应与梯度变化曲线. (a) 系统的状态响应曲线; (b) 系统的控制输入曲线; (c) $\varphi(k)$ 的梯度变化曲线.

Figure 7　(Color online) The response and gradient variation curves of the system in example 2 under the controller obtained by Algorithm 1 with different initial control inputs. (a) State response of the system; (b) control input of the system; (c) gradient evolution of $\varphi(k)$.

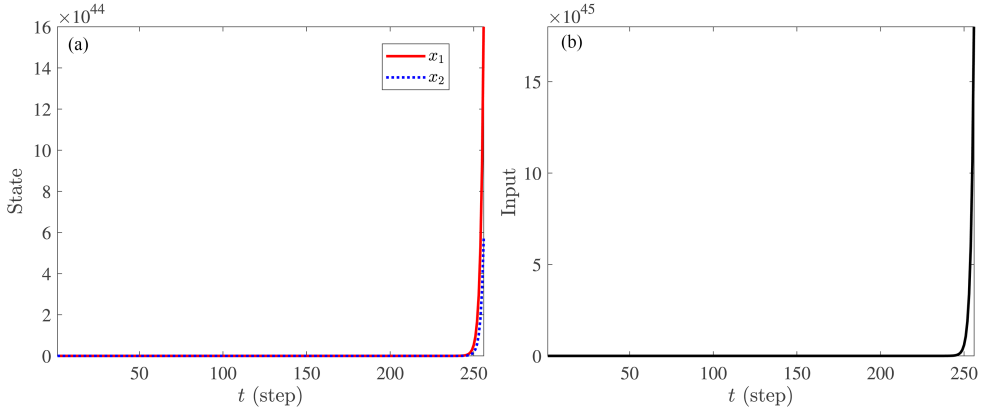

图 8　(网络版彩图) 文献 [23] 算法得到的控制律下算例 2 的系统响应曲线. (a) 系统状态响应曲线; (b) 系统控制输入曲线.

Figure 8　(Color online) The response curves of the system in example 2 under the controller in [23]. (a) State response of the system; (b) control input of the system.

将其作用于本算例中的系统去收集状态和输入数据. 在后续的迭代过程中, 将控制律 $u(t)$ 应用于被控系统. 系统在该控制律作用下的状态响应和控制输入曲线分别如图 7(a) 和 (b) 所示. 此外, 序列 $\varphi(k)$ 的梯度如图 7(c)

https://www.sciengine.com/doi/10.1360/SSI-2025-0049

所示. 从图 7 可以看出, 在选取不同初始反馈增益矩阵的情况下, 系统在控制律 $u(t)$ 的作用下均能够保持稳定. 仿真时间为 0.0708 s.

为了验证本文提出的数据驱动控制算法的有效性, 将文献 [23] 中针对无时滞系统提出的数据驱动控制算法应用于输入时滞系统 (3). 图 8(a) 展示了在文献 [23] 算法得到的控制律下, 输入时滞系统的状态响应曲线; 图 8(b) 则展示了系统的控制输入曲线. 从图 8 中可以明显观察到, 系统的状态和控制输入出现了发散现象. 与之相比, 本文提出的数据驱动控制算法是有效的.

5　结论

本文针对系统矩阵和输入矩阵未知的输入时滞系统的镇定问题, 设计了策略迭代算法以得到控制律. 首先, 利用模型缩减方法将离散时间输入时滞系统转化为无时滞系统. 针对缩减的无时滞系统, 使用系统状态和输入数据求解代数 Riccati 矩阵方程的唯一正定解的近似值, 进而得到可镇定的控制器. 仿真结果表明所设计的控制器能够镇定输入时滞系统.

参考文献

1 Wu A-G, Duan G-R. New iterative algorithms for solving coupled Markovian jump Lyapunov equations. IEEE Trans Automat Contr, 2015, 60: 289–294

2 Qian Y Y, Pang W J. An implicit sequential algorithm for solving coupled Lyapunov equations of continuous-time Markovian jump systems. Automatica, 2015, 60: 245–250

3 Barforooshan M, Derpich M S, Stavrou P A, et al. The effect of time delay on the average data rate and performance in networked control systems. IEEE Trans Automat Contr, 2020, 67: 16–31

4 Wu X, Ding S, Xu N, et al. Periodic event–triggered bipartite containment control for nonlinear multi–agent systems with input delay. Int J Syst Sci, 2024, 55: 2008–2022

5 Han S Y, Wang D, Chen Y H, et al. Optimal tracking control for discrete-time systems with multiple input delays under sinusoidal disturbances. Int J Control Autom Syst, 2015, 13: 292–301

6 Qi Q, Xie L, Zhang H. Linear-quadratic optimal control for discrete-time mean-field systems with input delay. IEEE Trans Automat Contr, 2022, 67: 3806–3821

7 Vrabie D, Pastravanu O, Abu-Khalaf M, et al. Adaptive optimal control for continuous-time linear systems based on policy iteration. Automatica, 2009, 45: 477–484

8 Jiang Y, Jiang Z P. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. Automatica, 2012, 48: 2699–2704

9 Pang B, Jiang Z-P, Mareels I. Reinforcement learning for adaptive periodic linear quadratic control. In: Proceedings of the 2019 IEEE 58th Conference on Decision and Control (CDC), 2019, 3322–3327

10 Pang B, Jiang Z P, Mareels I. Reinforcement learning for adaptive optimal control of continuous-time linear periodic systems. Automatica, 2020, 118: 109035

11 Jiang H, Zhou B, Duan G-R. Modified general policy iteration based adaptive dynamic programming for unknown discrete-time linear systems. Intl J Robust Nonlinear, 2022, 32: 7149–7173

12 Yang Y, Kiumarsi B, Modares H, et al. Model-free λ-policy iteration for discrete-time linear quadratic regulation. IEEE Trans Neural Netw Learn Syst, 2023, 34: 635–649

13 Rizvi S A A, Wei Y, Lin Z. Model-free optimal stabilization of unknown time delay systems using adaptive dynamic programming. In: Proceedings of the 2019 IEEE 58th Conference on Decision and Control (CDC), 2019. 6536–6541

14 Vatanski N, Georges J P, Aubrun C, et al. Networked control with delay measurement and estimation. Control Eng Pract, 2009, 17: 231–244

15 Zhang W, Branicky M S, Phillips S M. Stability of networked control systems. IEEE Control Syst Mag, 2001, 21: 84–99

16 Niculescu S I. Delay Effects on Stability: A Robust Control Approach. London: Springer, 2003

17 Zhou B, Lin Z. Parametric lyapunov equation approach to stabilization of discrete-time systems with input delay and saturation. IEEE Trans Circuits Syst I, 2011, 58: 2741–2754

18 Karafyllis I, Krstic M. Robust predictor feedback for discrete-time systems with input delays. Int J Control, 2013, 86: 1652–1663

19 Guo Y, Xu X, Liu L, et al. Stabilization of discrete-time linear systems with infinite distributed input delays. Automatica, 2023, 148: 110786

https://www.sciengine.com/doi/10.1360/SSI-2025-0049

20 Wu A-G, Duan G-R, Wang Y, et al. A model reduction approach for discrete-time linear time-variant systems with delayed inputs. Sci China Inf Sci, 2024, 67: 142201

21 Varga A. Periodic Lyapunov equations: some applications and new algorithms. Int J Control, 1997, 67: 69–88

22 Komaroff N. Iterative matrix bounds and computational solutions to the discrete algebraic Riccati equation. IEEE Trans Automat Contr, 1994, 39: 1676–1678

23 Wu A G, Meng Y. Data-driven adaptive optimal control for discrete-time linear time-invariant systems. Int J Syst Sci, 2024, 55: 3069–3082

24 Jha S K, Roy S B, Bhasin S. Initial excitation-based iterative algorithm for approximate optimal control of completely unknown LTI systems. IEEE Trans Automat Contr, 2019, 64: 5230–5237

25 Zhou B, Peng Z, Michiels W. Memoryless dual-observer-based output feedback stabilization of linear systems with input and output delays. IEEE Trans Cybern, 2024, 54: 5806–5817

26 Xia Y, Liu G P, Shi P, et al. Sliding mode control of uncertain linear discrete time systems with input delay. IET Control Theor Appl, 2007, 1: 1169–1175

27 Zhou B, Li Z Y, Lin Z. Stabilization of discrete-time systems with multiple actuator delays and saturations. IEEE Trans Circuits Syst I, 2013, 60: 389–400

28 Zhao Q, Xu H, Sarangapani J. Finite-horizon near optimal adaptive control of uncertain linear discrete-time systems. Optim Control Appl Meth, 2015, 36: 853–872

附录 1

当时间变量 t 在区间 $\mathbb{I}[t_{\mathrm{s}k}, t_{\mathrm{f}k}]$ 内变化时, 有

$$
\begin{aligned}
&\tilde{x}^{\mathrm{T}}(t_{\mathrm{s}k}+1)\mathrm{vecs}\left(\left(A^{h}\right)^{\mathrm{T}} P(k) A^{h}\right)+\tilde{\mathring{u}}^{\mathrm{T}}(t_{\mathrm{s}k}-h+1)\mathrm{vecs}\left(\mathcal{B}^{\mathrm{T}} \mathring{A}^{\mathrm{T}} P(k) \mathring{A}\mathcal{B}\right)\\
&\quad=\tilde{x}^{\mathrm{T}}(t_{\mathrm{s}k})\mathrm{vecs}\left(\left(A^{h+1}\right)^{\mathrm{T}} P(k) A^{h+1}\right)+2\left(\mathring{u}(t_{\mathrm{s}k}-h)\otimes x(t_{\mathrm{s}k})\right)^{\mathrm{T}}\mathrm{vec}\left(\left(A^{h+1}\right)^{\mathrm{T}} P(k) A\mathring{A}\mathcal{B}\right)\\
&\qquad+\tilde{\mathring{u}}^{\mathrm{T}}(t_{\mathrm{s}k}-h)\mathrm{vecs}\left(\mathcal{B}^{\mathrm{T}} \mathring{A}^{\mathrm{T}} A^{\mathrm{T}} P(k) A\mathring{A}\mathcal{B}\right)+2\left(x(t_{\mathrm{s}k})\otimes K(k-1) A^{h} x(t_{\mathrm{s}k})\right)^{\mathrm{T}}\mathrm{vec}\left(F^{\mathrm{T}}(k)\right)\\
&\qquad+4\left(\mathring{u}(t_{\mathrm{s}k}-h)\otimes x(t_{\mathrm{s}k})\right)^{\mathrm{T}}\mathrm{vec}\left(\left(A^{h}\right)^{\mathrm{T}} K^{\mathrm{T}}(k-1) H(k)\right)\\
&\qquad+2\left(\mathring{u}(t_{\mathrm{s}k}-h)\otimes K(k-1)\mathring{A}\mathcal{B}\mathring{u}(t_{\mathrm{s}k}-h)\right)^{\mathrm{T}}\mathrm{vecs}(H(k))+\tilde{u}^{\mathrm{T}}(t_{\mathrm{s}k})\mathrm{vecs}(G(k))\\
&\qquad-2\left(\mathring{u}(t_{\mathrm{s}k}-h+1)\otimes x(t_{\mathrm{s}k}+1)\right)^{\mathrm{T}}\mathrm{vec}\left(\left(A^{h}\right)^{\mathrm{T}} P(k)\mathring{A}\mathcal{B}\right),\\
&\tilde{x}^{\mathrm{T}}(t_{\mathrm{s}k}+2)\mathrm{vecs}\left(\left(A^{h}\right)^{\mathrm{T}} P(k) A^{h}\right)+\tilde{\mathring{u}}^{\mathrm{T}}(t_{\mathrm{s}k}-h+2)\mathrm{vecs}\left(\mathcal{B}^{\mathrm{T}} \mathring{A}^{\mathrm{T}} P(k)\mathring{A}\mathcal{B}\right)\\
&\quad=\tilde{x}^{\mathrm{T}}(t_{\mathrm{s}k}+1)\mathrm{vecs}\left(\left(A^{h+1}\right)^{\mathrm{T}} P(k) A^{h+1}\right)+2\left(\mathring{u}(t_{\mathrm{s}k}-h+1)\otimes x(t_{\mathrm{s}k}+1)\right)^{\mathrm{T}}\mathrm{vec}\left(\left(A^{h+1}\right)^{\mathrm{T}} P(k) A\mathring{A}\mathcal{B}\right)\\
&\qquad+\tilde{\mathring{u}}^{\mathrm{T}}(t_{\mathrm{s}k}-h)\mathrm{vecs}\left(\mathcal{B}^{\mathrm{T}} \mathring{A}^{\mathrm{T}} A^{\mathrm{T}} P(k) A\mathring{A}\mathcal{B}\right)+2\left(x(t_{\mathrm{s}k})\otimes K(k-1) A^{h} x(t_{\mathrm{s}k})\right)^{\mathrm{T}}\mathrm{vec}\left(F^{\mathrm{T}}(k)\right)\\
&\qquad+4\left(\mathring{u}(t_{\mathrm{s}k}-h+1)\otimes x(t_{\mathrm{s}k}+1)\right)^{\mathrm{T}}\mathrm{vec}\left(\left(A^{h}\right)^{\mathrm{T}} K^{\mathrm{T}}(k-1) H(k)\right)\\
&\qquad+2\left(\mathring{u}(t_{\mathrm{s}k}-h+1)\otimes K(k-1)\mathring{A}\mathcal{B}\mathring{u}(t_{\mathrm{s}k}-h+1)\right)^{\mathrm{T}}\mathrm{vecs}(H(k))\\
&\qquad+\tilde{u}^{\mathrm{T}}(t_{\mathrm{s}k}+1)\mathrm{vecs}(G(k))-2\left(\mathring{u}(t_{\mathrm{s}k}-h+2)\otimes x(t_{\mathrm{s}k}+2)\right)^{\mathrm{T}}\mathrm{vec}\left(\left(A^{h}\right)^{\mathrm{T}} P(k)\mathring{A}\mathcal{B}\right),\\
&\cdots\\
&\tilde{x}^{\mathrm{T}}(t_{\mathrm{f}k}+1)\mathrm{vecs}\left(\left(A^{h}\right)^{\mathrm{T}} P(k) A^{h}\right)+\tilde{\mathring{u}}^{\mathrm{T}}(t_{\mathrm{f}k}-h+1)\mathrm{vecs}\left(\mathcal{B}^{\mathrm{T}} \mathring{A}^{\mathrm{T}} P(k)\mathring{A}\mathcal{B}\right)\\
&\quad=\tilde{x}^{\mathrm{T}}(t_{\mathrm{f}k})\mathrm{vecs}\left(\left(A^{h+1}\right)^{\mathrm{T}} P(k) A^{h+1}\right)+2\left(\mathring{u}(t_{\mathrm{f}k}-h)\otimes x(t_{\mathrm{f}k})\right)^{\mathrm{T}}\mathrm{vec}\left(\left(A^{h+1}\right)^{\mathrm{T}} P(k) A\mathring{A}\mathcal{B}\right)\\
&\qquad+\tilde{\mathring{u}}^{\mathrm{T}}(t_{\mathrm{f}k}-h)\mathrm{vecs}\left(\mathcal{B}^{\mathrm{T}} \mathring{A}^{\mathrm{T}} A^{\mathrm{T}} P(k) A\mathring{A}\mathcal{B}\right)+2\left(x(t_{\mathrm{f}k})\otimes K(k-1) A^{h} x(t_{\mathrm{f}k})\right)^{\mathrm{T}}\mathrm{vec}\left(F^{\mathrm{T}}(k)\right)\\
&\qquad+4\left(\mathring{u}(t_{\mathrm{f}k}-h)\otimes x(t_{\mathrm{f}k})\right)^{\mathrm{T}}\mathrm{vec}\left(\left(A^{h}\right)^{\mathrm{T}} K^{\mathrm{T}}(k-1) H(k)\right)\\
&\qquad+2\left(\mathring{u}(t_{\mathrm{f}k}-h)\otimes K(k-1)\mathring{A}\mathcal{B}\mathring{u}(t_{\mathrm{f}k}-h)\right)^{\mathrm{T}}\mathrm{vecs}(H(k))+\tilde{u}^{\mathrm{T}}(t_{\mathrm{f}k})\mathrm{vecs}(G(k))
\end{aligned}
$$

https://www.sciengine.com/doi/10.1360/SSI-2025-0049

$$-2\left(\mathring{u}(t_{\mathrm{f}k}-h+1)\otimes x(t_{\mathrm{f}k}+1)\right)^{\mathrm{T}}\operatorname{vec}\left(\left(A^{h}\right)^{\mathrm{T}}P\left(k\right)\mathring{A}\mathcal{B}\right).$$

Data-driven control for input-delayed systems based on model reduction

Yuan MENG[1], Ying ZHANG[1*], Wennian QI[2] & Jie ZHANG[1]

1. *School of Intelligent Science and Engineering, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China*
2. *School of Robotics and Advanced Manufacturing, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China*
* Corresponding author. E-mail: zhangyinghit@hit.edu.cn

Abstract In this paper, the stabilization problem of discrete-time input-delayed systems with unknown system and input matrices is investigated. To address this, a policy iteration-based control algorithm is proposed. Since the system matrix and input matrix of the controlled system are unknown, the data of states and inputs is collected to construct data matrices. In addition, due to the presence of input delays, the controller cannot be applied to the system immediately, which may result in system instability. To this end, the model reduction method is employed to transform the input-delayed system into a time-delay-free system. For the resulting delay-free system, the algebraic Riccati matrix equation is solved by using the data matrix. Then, the control law is designed to stabilize the input-delayed system. Importantly, an initially stabilizing controller is not required for the proposed iterative algorithm. Finally, the effectiveness of the proposed algorithm is verified through two examples. The simulation results show that the state responses of systems converge, that is, the designed control laws can stabilize the considered systems.

Keywords adaptive dynamic programming, data-driven control, policy iteration algorithm, discrete-time input-delayed system

https://www.sciengine.com/doi/10.1360/SSI-2025-0049