一种基于 Off-Policy 的无模型输出数据反馈 H_∞ 控制方法

李 臻 [1] 范家璐 [1] 姜 艺 [1] 柴天佑 [1]

摘 要 针对模型未知的线性离散系统在扰动存在条件下的调节控制问题, 提出了一种基于 Off-policy 的输入输出数据反馈的 H_∞控制方法. 本文从状态反馈在线学习算法出发, 针对系统运行过程中状态数据难以测得的问题, 通过引入增广数据向量将状态反馈策略迭代在线学习算法转化为输入输出数据反馈在线学习算法. 更进一步, 通过引入辅助项的方法将输入输出数据反馈策略迭代在线学习算法转化为无模型输入输出数据反馈 Off-policy 学习算法. 该算法利用历史输入输出数据实现最优输出反馈策略的学习, 同时克服了 On-policy 算法需要频繁与实际环境进行交互这一缺点. 除此之外, 与 On-policy 算法相比, Off-policy 学习算法具有克服学习噪声的影响, 使学习结果收敛于理论最优值这一优点. 最终, 通过仿真实验验证了学习算法的收敛性.

关键词 H_∞ 控制, 强化学习, Off-policy, 数据驱动

引用格式 李臻, 范家璐, 姜艺, 柴天佑. 一种基于 Off-Policy 的无模型输出数据反馈 H_∞ 控制方法. 自动化学报, 2021, **47**(9): 2182−2193

DOI 10.16383/j.aas.c190499

A Model-Free H_∞ Control Method Based on Off-Policy With Output Data Feedback

LI Zhen[1] FAN Jia-Lu[1] JIANG Yi[1] CHAI Tian-You[1]

Abstract This paper proposes an H_∞ control method based on input and output data feedback and off-policy iteration for linear discrete-time systems with unknown model in the presence of disturbances. Aiming at the problem that it is difficult to measure the state data during the operation of the system, this paper considers the knowledge of the state feedback policy iteration (PI) algorithm and transforms the online state feedback learning algorithm into the online input-output data feedback learning algorithm by introducing augmented data vector. Furthermore, the online input-output data feedback learning algorithm is transformed into a model-free input-output data feedback off-policy learning algorithm by introducing auxiliary items. This algorithm uses historical input and output data to realize the optimal output feedback learning strategy, and overcomes the shortcoming of the on-policy algorithm that needs frequent interaction with the environment. In addition, compared with the on-policy algorithm, off-policy has the advantage of overcoming the influence of learning noise and converging to the theoretical optimal value. Finally, the convergence of this algorithm is verified by simulation experiments.

Key words H_∞ control, reinforcement learning, off-policy, data-driven

Citation Li Zhen, Fan Jia-Lu, Jiang Yi, Chai Tian-You. A model-free H_∞ control method based on off-policy with output data feedback. *Acta Automatica Sinica*, 2021, **47**(9): 2182−2193

鲁棒控制理论经过多年的完善与发展, 已经趋于成熟. 所谓的鲁棒控制是指在扰动能量一定的情况下, 仍能保证一定性能指标的控制方法. H_∞ 控制算法是鲁棒控制的一种, 通过最小化 H_∞ 指标来达到鲁棒控制的目的[1−3]. 文献 [4−5] 中指出 H_∞ 控制和零和博弈存在内在联系. 即把输入和扰动看作相互博弈的两个参与者, 一般在定义指标时使输入为指标的最小参与者, 扰动为最大参与者. 对于系统的 H_∞ 控制可以归结为求解博弈黎卡提方程 (Game algebraic Riccati equation, GARE)[5−6]. H_∞ 控制虽然能保证参数在一定波动范围内的稳定性, 但需要完整的系统动态模型. 这大大限制了 H_∞ 的应用范围.

随着信息科学技术的发展, 与生活息息相关的实际过程, 如冶金、化工、电力、物流运输等方面发生了重大变化, 复杂的生产设备与大量不同种类的传感器同时应用使工业过程趋于复杂. 因此, 这些过程在难以建立准确的数学模型的同时却产生储存着大量反映系统动态的运行数据. 数据驱动的控制方法在此基础上产生. 经过多年的发展, 数据驱动

收稿日期 2019-06-30 录用日期 2019-10-11
Manuscript received June 30, 2019; accepted October 11, 2019
国家自然科学基金 (61533015, 61304028), 兴辽英才计划 (XLYC2007135) 资助
Supported by National Natural Science Foundations of China (61533015, 61304028) and Liaoning Revitalization Talents Program (XLYC2007135)
本文责任编委 侯忠生
Recommended by Associate Editor HOU Zhong-Sheng
1. 东北大学流程工业综合自动化国家重点实验室 沈阳 110819
1. State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819

的方法主要利用这些数据实现设计控制器, 预测评估系统状态, 在线优化决策, 甚至诊断故障[7]. 而数据驱动控制是指在不使用被控过程数学模型的信息的情况下, 直接利用被控系统的数据设计控制器的控制理论和方法, 且经过严谨地数学论证后可以保证控制器满足一定的鲁棒性与收敛性[8].

强化学习通过启发机制来学习智能体与环境的交互策略, 以优化在交互过程中的长期收益的算法[6]. 在控制领域中, 强化学习常用来解决自适应最优控制问题[9−12]. 其中, Doya[13] 首次将基于强化学习的控制器应用于连续系统中. 在近期的研究中, 强化学习应用于更加有针对性的复杂工业过程中[14−19], 尤其是对难以建模的复杂工业过程进行了针对性的研究[15−17]. 文献 [12] 对数据驱动的迭代优化控制方法进行了综述性研究. 文献 [14] 将 Q-learning 应用于考虑丢包问题的网络环境中, 实现了一种数据驱动的浮选过程控制方法. 文献 [15] 考虑了输入受限情况下的数据驱动浮选控制问题, 并将强化学习应用在浮选过程的双率控制中. 文献 [20]针对非线性系统跟踪控制问题, 提出了一种基于 Q-learning 的直接求解评价函数的方法, 避免了求解更复杂的 HJBE (Hamilton-Jacobi-Bellman equation). 文献 [21] 针对输出调节控制问题提出了一种新型基于 Q-learning 的控制方法. 文献 [22] 针对离散时间非线性系统基于事件的最优调节控制问题, 提出了一种基于启发式动态规划的事件驱动方法. 文献 [23] 对非线性连续时间系统自适应评价控制问题进行了综述性研究. Al-Tamimi 等[24] 将 Q-learning 算法应用于 H_∞ 控制问题.

Off-policy 学习算法是强化学习的一种, 其主要特点是在学习过程中, Off-policy 算法定义了两个不同的策略, 一种是用来产生数据的策略, 另一种是求解得到的目标策略[25−27]. 而 On-policy 算法中, 两种策略相同, 需要将每次迭代所求得的算法代入实际环境中来产生学习所用的数据. 因此, 与 On-policy 算法相比, Off-policy 算法更具可用性. 同时与以 Q-learning 算法[24] 为代表的 On-policy 学习方法相比, Off-policy 算法可以消除在学习过程中由探测噪声所产生的误差[25]. 换句话说, Off-policy 学习算法是一种无偏的学习方法. 在基于 Off-policy 算法的连续系统 H_∞ 控制的基础上, 文献 [25] 将 Off-policy 算法引入了离散线性系统 H_∞ 控制问题中, 提出了需要状态反馈值的离散线性系统控制算法. 其通过考察状态变化与值函数变化的关系, 构造了一种与探测噪声无关的贝尔曼方程, 进而提出了一种无模型状态反馈 Off-policy 的 H_∞ 控制算法.

上述算法由于忽略了在某些应用条件下关键状态变量无法反馈这一问题, 降低了算法在实际应用环境下的可用范围. 因此, 本文提出一种基于 Off-policy 的数据驱动输出反馈 H_∞ 控制算法, 针对模型未知的离散线性系统模型, 实现无限时域上工作点附近镇定控制的同时对能量有限的噪声进行抑制. 区别与状态反馈问题, 本文的反馈量是输出, 本文通过构造了新的值函数, 避免了直接使用状态变量. 由于无法直接使用状态作为反馈, 因而构造了一种增广数据向量, 来解决输出反馈控制问题. 在线性模型依赖在线策略迭代 (Policy iteration, PI) 状态反馈算法的基础上, 将结合了系统历史输出数据与历史输入数据的增广数据向量作为反馈量, 使在线状态反馈算法转换为模型依赖在线策略迭代输入输出反馈算法. 通过引入辅助项的方法将模型依赖 On-policy 输入输出反馈算法转换为无模型 Off-policy 输入输出反馈控制算法. 由于采用了 Off-policy 算法, 该算法具有结构简单, 可以得到理论上的最优值的特点[25]. 除此之外, 与 On-policy 算法相比, Off-policy 算法的可以实现离线学习, 克服了 On-policy 算法只能在线学习需要频繁和环境交互的问题. 在本文的最后, 针对飞机短时间周期飞行姿态控制模型进行了仿真实验, 以验证算法的有效性.

1　研究背景

1.1　问题描述

对于线性离散系统

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Dw_k \\ y_k &= Cx_k \end{aligned} \tag{1}$$

其中, $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, $C \in \mathbf{R}^{l\times n}$, $D \in \mathbf{R}^{n\times r}$ 是离散线性系统动态矩阵. $x_k \in \mathbf{R}^n$ 是系统状态数据; $u_k \in \mathbf{R}^m$ 为控制输入; $w_k \in \mathbf{R}^r$ 为扰动输入; $y_k \in \mathbf{R}^l$ 为系统输出.

定义 H_∞ 控制问题的优化指标如下:

$$r(y_k, u_k, w_k) = y_k^{\mathrm{T}} Q y_k + u_k^{\mathrm{T}} R u_k - \gamma^2 w_k^{\mathrm{T}} w_k \tag{2}$$

其中, Q 和 R 为预设的正定矩阵, γ 为某一预设正值. 该指标具有实际意义: 项 $y_k^{\mathrm{T}} Q y_k$ 表示了系统在噪声作用下输出波动的大小; 项 $u_k^{\mathrm{T}} R u_k$ 为考虑到实际物理条件时, 对输入幅值的限制; 项 $\gamma^2 w_k^{\mathrm{T}} w_k$ 则是噪声相关的修正项, 保证解的存在性. 自然地, 将损失指标在时域上的累积作为待优化指标. 损失与输出、扰动噪声和控制策略相关, 可表示为

$$J(y_k,u_k,w_k)=\sum_{i=k}^{\infty}r(y_k,u_k,w_k)=\sum_{i=k}^{\infty}\left[y_k^{\mathrm{T}}Qy_k+u_k^{\mathrm{T}}Ru_k-\gamma^2w_k^{\mathrm{T}}w_k\right] \tag{3}$$

定义最优运行优化策略为 u_k^*, 最优运行优化策略为使指标最小的优化策略. 指标与扰动 w_k 同样存在相关关系, 而 w_k 是不可控的. 考虑到扰动对指标的影响, 选择一种保守的优化策略, 即使最坏情况下的指标最小. 最坏情况下的扰动与扰动对系统的影响和预设值 R , Q , γ 的选取有关. 综上所述, 可以将最优控制策略表示为

$$u_k^*=\arg\min_{u_k}\max_{w_k}J \tag{4}$$

将寻求最优控制率的问题考虑为两玩家零和博弈的问题[6], 其中, u_k 作为寻找最优控制策略的玩家 1, 而 w_k 作为寻找最坏扰动策略的玩家 2. 本文采用一种线性的决策方法, 即通过一组线性关系来描述最优控制策略和最坏扰动策略. 策略所产生的结果和由历史数据所组成的增广向量有关. 增广向量的表达式将在之后给出. 其可以定义为

$$\begin{aligned}u_k&=-K_1^*z_k\\w_k&=-K_2^*z_k\end{aligned} \tag{5}$$

其中, z_k 代表在第 k 步时系统的增广向量值, 等式中的下标 k 表示系统运行的第 k 步. K_1^* 和 K_2^* 分别代表由线性反馈策略所描述的最优控制策略和最坏扰动策略.

1.2 博弈黎卡提方程

根据动态规划中的最优性原理, 定义值函数为

$$V(x_k,u_k,w_k)=\sum_{i=k}^{\infty}\left[y_i^{\mathrm{T}}Qy_i+u_i^{\mathrm{T}}Ru_i-\gamma^2w_i^{\mathrm{T}}w_i\right] \tag{6}$$

由于值函数中突出系统的预测性和评价性, 即可以根据当前的状态推测之后的待优化指标累积值. 而系统的输出无法完全反映系统的状态, 所以这里将值函数描述为和系统状态相关的形式, 而非和系统输出相关的形式. 由最优控制策略 u_k^* 和最坏扰动策略 w_k^* 所得到的值函数可以称为目标值函数, 用 V^* 表示为

$$V^*(x_k,u_k^*,w_k^*)=\sum_{i=k}^{\infty}\left[y_i^{\mathrm{T}}Qy_i+u_i^{*\mathrm{T}}Ru_i^*-\gamma^2w_i^{\mathrm{T}*}w_i^*\right] \tag{7}$$

为保证最优策略的存在, 需要对系统进行一些一般性地假设[1−2], 如下:

1) 系统扰动能量有限, 即

$$\sum_{i=0}^{\infty}w_i^{\mathrm{T}}w_i<\infty \tag{8}$$

2) 系统存在小于等于 γ 的 L_2 增益, 即

$$\sum_{i=0}^{\infty}\left[y_i^{\mathrm{T}}Qy_i+u_i^{\mathrm{T}}Ru_i\right]\le\gamma^2\sum_{i=0}^{\infty}w_i^{\mathrm{T}}w_i \tag{9}$$

3) 系统满足条件

$$\max_{w_k}\min_{u_k}V(x_k,u_k,w_k)=\min_{u_k}\max_{w_k}V(x_k,u_k,w_k) \tag{10}$$

4) 系统可观.

如果确保了上述条件且最优策略为线性控制策略, 则值函数可以描述为关于 x_k 的二次型的形式. 其说明在引入线性控制策略表示后给出.

$$V^*(x_k)=x_k^{\mathrm{T}}Px_k \tag{11}$$

将式 (11) 代入式 (7) 中, 可以得到贝尔曼方程

$$x_k^{\mathrm{T}}Px_k=x_k^{\mathrm{T}}\bar{Q}x_k+u_k^{\mathrm{T}}Ru_k-\gamma^2w^{\mathrm{T}}w+x_{k+1}^{\mathrm{T}}Px_{k+1} \tag{12}$$

其中, $\bar{Q}=C^{\mathrm{T}}QC$, 保证了输入反馈算法和输出反馈算法值函数一致.

定义哈密顿方程

$$H(x_k,u_k,w_k)=x_k^{\mathrm{T}}\bar{Q}x_k+u_k^{\mathrm{T}}Ru_k-\gamma^2w^{\mathrm{T}}w+x_{k+1}^{\mathrm{T}}Px_{k+1}-x_k^{\mathrm{T}}Px_k \tag{13}$$

引入状态反馈中间策略 $\bar{K}_1$, $\bar{K}_2$, 以区别于根据输入输出数据进行决策的目标策略. 需要注意, 中间策略和目标策略决策结果等价.

最优决策策略 u^* 和最坏扰动策略 w^*, 应满足, $\frac{\partial H(x_k,u_k,w_k)}{\partial u_k}=0$, $\frac{\partial H(x_k,u_k,w_k)}{\partial w_k}=0$. 可以得到

$$\begin{aligned}u_k&=-\bar{K}_1^*x_k\\w_k&=-\bar{K}_2^*x_k\end{aligned} \tag{14}$$

其中,

$$\begin{aligned}\bar{K}_1^*=&\left(R+B^{\mathrm{T}}PB+B^{\mathrm{T}}PD\left(\gamma^2I-D^{\mathrm{T}}PD\right)^{-1}D^{\mathrm{T}}PB\right)^{-1}\times\\&\left(B^{\mathrm{T}}PA+B^{\mathrm{T}}PD\left(\gamma^2I-D^{\mathrm{T}}PD\right)^{-1}D^{\mathrm{T}}PA\right)\\\bar{K}_2^*=&\left(D^{\mathrm{T}}PD-\gamma^2I-D^{\mathrm{T}}PB\left(R+B^{\mathrm{T}}PB\right)^{-1}B^{\mathrm{T}}PD\right)^{-1}\times\\&\left(D^{\mathrm{T}}PA-D^{\mathrm{T}}PB\left(R+B^{\mathrm{T}}PB\right)^{-1}B^{\mathrm{T}}PA\right)\end{aligned} \tag{15}$$

至此, 可以重新讨论值函数为二次型的原因. 值函数可以拆分为多个二次型求和, 自然可以表示

为二次型的形式. 可以通过归纳法进行简单的证明: 假设当 $k=t$ 值函数为二次型的型式, 其中 t 为任意时刻, 也就是 $V^*(x_t)=x_t^{\mathrm{T}}Px_t$. 根据式 (12) 和式 (13), 当 $k=t-1$时, 有

$$V^*(x_{t-1})=x_{t-1}^{\mathrm{T}}\bar{Q}x_{t-1}+x_{t-1}^{\mathrm{T}}\bar{K}_1^{\mathrm{T}}R\bar{K}_1x_{t-1}-\gamma^2x_{t-1}^{\mathrm{T}}\bar{K}_2^{\mathrm{T}}\bar{K}_2x_{t-1}+x_t^{\mathrm{T}}Px_t$$

可知当 $k=t-1$ 时值函数同样是二次型. 则得到, 值函数应可以描述为二次型的型式.

在文献 [6] 中, 可知 P 满足博弈黎卡提方程 (GARE).

$$P=A^{\mathrm{T}}PA+\bar{Q}-\begin{bmatrix}A^{\mathrm{T}}PB & A^{\mathrm{T}}PD\end{bmatrix}\times\begin{bmatrix}R+B^{\mathrm{T}}PB & B^{\mathrm{T}}PD\\ D^{\mathrm{T}}PB & D^{\mathrm{T}}PD-\gamma^2I\end{bmatrix}^{-1}\begin{bmatrix}B^{\mathrm{T}}PA\\ D^{\mathrm{T}}PA\end{bmatrix}\tag{16}$$

至此, 一种 H_∞ 控制策略求解问题就转化为了一种最优策略求解问题, 而最优策略求解问题可以转化为博弈黎卡提方程求解问题.

2　数据驱动在线策略迭代学习算法

2.1　状态反馈在线策略迭代学习算法

在博弈黎卡提方程提出的基础上, 将描述一种 On-policy 解法, 该方法通过策略迭代算法实现在线求解最优策略. 该方法在模型已知的情况下, 通过所采集到的系统状态数据反馈来求解博弈黎卡提方程. 最后所求得的最优策略需要根据系统的状态数据进行决策, 也就是最优中间策略.

算法 1.　状态反馈在线策略迭代学习算法

1: 初始化, $j=0$, 选择初始化策略 $\bar{K}_1^0$, $\bar{K}_2^0$.

2: 将策略 $\bar{K}_1^j$, $\bar{K}_2^j$ 应用到系统中产生足够的数据.

3: 根据贝尔曼方程 (17) 更新 P .

$$x_k^{\mathrm{T}}P^{j+1}x_k-x_{k+1}^{\mathrm{T}}P^{j+1}x_{k+1}=x_k^{\mathrm{T}}\bar{Q}x_k+(u_k^j)^{\mathrm{T}}Ru_k^j-\gamma^2(w_k^j)^{\mathrm{T}}w_k^j\tag{17}$$

4: 根据式 (15) 更新策略得到 $\bar{K}_1^{j+1}$, $\bar{K}_2^{j+1}$.

5: 判断是否达到最优值. 如果

$$||\bar{K}_1^{j+1}-\bar{K}_1^j||<\epsilon,\ |\bar{K}_2^{j+1}-\bar{K}_2^j||<\epsilon$$

则认为达到了最优值, 结束计算; 否则, $j=j+1$, 回到步骤 2 进行下一次迭代. 其中, ϵ 为某一合适的极小值.

注 1. 算法 1 是一个依赖模型的算法. 在算法步骤 4 中需要用到式 (15), 由 P^{j+1} 推导 K_1^{j+1} 及 K_2^{j+1} 的过程中需要系统的模型. 算法的步骤 2 中, 需要根据矩阵 P 中未知数的个数选择合适的数据量大小, 且最小二乘法类似所采集到的数据越多每次迭代后的效果越好. 算法中步骤 3 更新 P^j, 需要运用 Kronecker 积可将二次型转化为线性相乘的性质实现求解[28].

$$a^{\mathrm{T}}Hb=(b^{\mathrm{T}}\otimes a^{\mathrm{T}})\mathrm{vec}(H)\tag{18}$$

其中, vec 表示对矩阵的向量化. $\otimes$ 表示两个向量的 Kronecker 积.

将式 (18) 代入式 (17) 中, 可以得到

$$(x_k^{\mathrm{T}}\otimes x_k^{\mathrm{T}}-x_{k+1}^{\mathrm{T}}\otimes x_{k+1}^{\mathrm{T}})\mathrm{vec}(P^j)=x_k^{\mathrm{T}}\bar{Q}x_k+(u_k^j)^{\mathrm{T}}Ru_k^j-\gamma^2(w_k^j)^{\mathrm{T}}w_k^j\tag{19}$$

通过最小二乘就可以求得 P^{j+1} 的解.

在考虑到系统的状态在实际应用中存在难以测量的问题, 所以中间最优策略在很多情况下无法直接应用于工程中. 因此, 第 2.2 节将在此算法的基础上提出一种由历史输入输出数据所组成的增广向量作为决策所需要的数据, 得到不依赖状态数据的控制策略.

2.2　一种由输入输出数据所组成的增广向量

本小节提出一种融合历史输入输出数据的增广向量. 运用增广数据向量对状态反馈算法进行改进, 实现通过输入输出进行最优策略的求解, 且此最优策略为只需要通过输入输出数据进行控制的目标控制策略.

定义 1. 增广数据向量

$$z_k=\begin{bmatrix}y_k & \cdots & y_{k-N} & u_{k-1} & w_{k-1} & \cdots & u_{k-N} & w_{k-N}\end{bmatrix}^{\mathrm{T}}\tag{20}$$

其中, N 为满足使

$$W=\begin{bmatrix}CA^N\\ \vdots\\ C\end{bmatrix}\tag{21}$$

的秩为 n 的某一预设值, n 是系统中状态变量的维数. 即满足

$$n=\mathrm{rank}(W)\tag{22}$$

为了将增广数据向量和状态反馈在线算法融合, 提出引理 1.

引理 1. 存在某一矩阵 M, 对如式 (1) 的系统使增广矩阵和系统状态存在如下关系

$$x_k=Mz_k\tag{23}$$

且对于式 (1) 所描述的系统, 可以构造

$$M = A^N W^+ \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} - D_A \Big] + [0 \cdots B \quad D \cdots A^{N-1}B \quad A^{N-1}D] \tag{24}$$

其中,

$$D_A = \begin{bmatrix} CB & CD & CAB & CAD & \cdots & CA^{N-1}D \\ 0 & 0 & CB & CD & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & CD \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix} \tag{25}$$

其中, W^+ 为 W 的广义逆.

证明. 根据系统动态可以描述为式 (1) 的系统在任意时刻的状态满足

$$y_{k-N+i} = Cx_k = CA^i x_{k-N} + C\begin{bmatrix} B & D & AB & AD & \cdots & A^{N-1}D \end{bmatrix} \begin{bmatrix} u_{k-1} \\ w_{k-1} \\ \vdots \\ w_{k-N} \end{bmatrix} \tag{26}$$

考虑 k 到 $k-N$ 时间段之内的输出, 可以表示为

$$\begin{bmatrix} y_k \\ \vdots \\ y_{k-N} \end{bmatrix} = Wx_{k-N} + D_A \begin{bmatrix} u_{k-1} \\ w_{k-1} \\ \vdots \\ u_{k-N} \\ w_{k-N} \end{bmatrix} \tag{27}$$

进一步改写为

$$\begin{bmatrix} y_k \\ \vdots \\ y_{k-N} \end{bmatrix} - D_A \begin{bmatrix} u_{k-1} \\ w_{k-1} \\ \vdots \\ u_{k-N} \\ w_{k-N} \end{bmatrix} = Wx_{k-N} \tag{28}$$

通过向量堆叠, 可以得到

$$x_{k-N} = W^+ \begin{bmatrix} I & -D_A \end{bmatrix} z_k \tag{29}$$

由于在假设中, N 可以使 $\text{rank}(W) = n$. 所以, W 的广义逆存在. 由于系统可观, 则这个 N 值一定存在, 也就是 W 一定存在. 通过式 (1) 所描述的系统状态转移关系, 可以将系统当前状态和历史状态之间的转换关系描述为

$$x_k = A^N x_{k-N} + \begin{bmatrix} B & D & AB & AD & \cdots & A^{N-1}D \end{bmatrix} \begin{bmatrix} u_{k-1} \\ w_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-N} \end{bmatrix} \tag{30}$$

根据式 (29) 和式 (30), 易得到式 (23). □

根据引理 1 可将式 (23) 代入式 (15) 中, 最优目标策略可以描述为

$$u_k = -K_1^* z_k, \quad w_k = -K_2^* z_k \tag{31}$$

其中, K_1^*, K_2^* 为目标控制策略.

$$\begin{aligned} K_1^* &= \bar{K}_1^* M = \\ &\left(R + B^{\mathrm{T}}PB + B^{\mathrm{T}}PD\left(\gamma^2 I - D^{\mathrm{T}}PD\right)^{-1}D^{\mathrm{T}}PB\right)^{-1} \times \\ &\left(B^{\mathrm{T}}PA + B^{\mathrm{T}}PD\left(\gamma^2 I - D^{\mathrm{T}}PD\right)^{-1}D^{\mathrm{T}}PA\right) M \\ K_2^* &= \bar{K}_2^* M = \\ &\left(D^{\mathrm{T}}PD - \gamma^2 I - D^{\mathrm{T}}PB\left(R + B^{\mathrm{T}}PB\right)^{-1}B^{\mathrm{T}}PD\right)^{-1} \times \\ &\left(D^{\mathrm{T}}PA - D^{\mathrm{T}}PB\left(R + B^{\mathrm{T}}PB\right)^{-1}B^{\mathrm{T}}PA\right) M \end{aligned} \tag{32}$$

结合引理 1 可以看出, 根据输入输出反馈数据与目标策略所得到的决策结果和根据状态数据与中间策略所得到的决策结果相同.

2.3 输入输出数据反馈在线策略迭代学习算法

根据上节所提出的增广数据向量可以将算法 1 转化为输入输出数据反馈在线学习方法. 该算法同样是依赖系统动态的在线学习算法, 且需要知道系统输出矩阵 C . 根据引理 1 可以将式 (12) 中的贝尔曼方程改写为

$$z_k^{\mathrm{T}} M^{\mathrm{T}} P M z_k = y_k^{\mathrm{T}} Q y_k + (u_k^j)^{\mathrm{T}} R u_k^j - \gamma^2 (w_k^j)^{\mathrm{T}} w_k^j + z_{k+1}^{\mathrm{T}} M P M z_{k+1} \tag{33}$$

与算法 1 类似, 可以得到应用输入输出数据的算法 2.

算法 2. 输入输出反馈在线策略迭代学习算法

1: 选择初始化策略 K_1^0, K_2^0.

2: 将策略 K_1^j, K_2^j 应用到系统中产生足够的数据.

3: 根据方程 (33) 更新 $M^{\mathrm{T}}PM$.

$$z_k^{\mathrm{T}}M^{\mathrm{T}}P^{j+1}Mz_k = y_k^{\mathrm{T}}Qy_k + (u_k^j)^{\mathrm{T}}Ru_k^j - \gamma^2(w_k^j)^{\mathrm{T}}w_k^j + z_{k+1}^{\mathrm{T}}M^{\mathrm{T}}P^{j+1}Mz_{k+1} \tag{34}$$

4: 根据式 (32) 更新策略得到 K_1^{j+1}, K_2^{j+1}.

5: 判断是否达到最优值. 如果

$$||K_1^{j+1} - K_1^j|| < \epsilon,\ ||K_2^{j+1} - K_2^j|| < \epsilon$$

则认为达到了最优值, 结束计算; 否则, $j = j + 1$, 回到步骤 2 进行下一次迭代. 其中, ϵ 为某一合适的极小值.

注 2. 算法 2 在步骤 3 时需要提前计算变换矩阵 M . 由于在步骤 3 时需要得到矩阵 P, 且在计算最优值的时候需要将矩阵 M 代入, 所以该算法为模型依赖算法. 如式 (32) 和式 (34) 所示, On-policy 迭代算法中每次迭代过后都需要通过更新后的控制策略来产生输出和扰动. 频繁更改控制策略是一种不经济的做法, 其难以保证在迭代过程中每一次更新的策略都是稳定的, 且策略的更改易造成决策结果的跳变从而导致边界条件大幅度变化, 除此之外对噪声进行控制并不符合实际物理条件. 算法必须频繁更换控制策略, 这是由于在式 (12) 中默认 u_k 与 w_k 是最优策略所产生的结果.

3　Off-policy 输入输出数据反馈优化算法

本节在保证所学习到的目标策略不变的前提下, 引入一种输入输出数据驱动的 Off-policy 学习算法. 该算法在实现无模型求解的同时解决了 On-policy 算法中需要频繁改变产生数据策略这一问题.

将式 (23) 和式 (31) 代入系统动态 (1) 中可得到在 On-policy 算法中前一时刻增广向量与后一时刻增广向量的关系, 即

$$Mz_{k+1} = (AM - BK_1 - DK_2)z_k \tag{35}$$

考虑 On-policy 算法必须更新产生数据所用的策略是由于式 (33) 中默认产生数据所用到的策略为每次迭代后所产生的策略. 因此, Off-policy 算法中抛弃了这一假设, 可以认为 K_1, K_2 所产数据和实际数据所用策略不相同. 用 K_1z_k, K_2z_k 代替式 (33) 中的 u_k, w_k . 同时将式 (35) 代入, 可以得到

$$z_k^{\mathrm{T}}M^{\mathrm{T}}PMz_k - z_k^{\mathrm{T}}(AM - BK_1 - DK_2)^{\mathrm{T}}P(AM - BK_1 - DK_2)z_k = y_k^{\mathrm{T}}Qy_k + z_k^{\mathrm{T}}K_1^{\mathrm{T}}RK_1z_k - \gamma^2 z_k^{\mathrm{T}}K_2^{\mathrm{T}}K_2z_k \tag{36}$$

注 3. 式 (35) 是只在 On-policy 算法中所满足的等式, 也就是在式 (31) 成立时该等式才成立. 而 Off-policy 算法中产生数据的策略可以不满足式 (31), 在此引用是为了得到一个在 Off-policy 算法下依然成立的等式. 在后续描述中认为式 (35) 和式 (33) 不成立, 只有导出结果式 (36) 成立. 此处就实现了 u_k, w_k 与 K_1z_k, K_2z_k 的分离, 式 (36) 对不同的产生数据的策略同样成立.

通过引入辅助项, 在式 (36) 的基础上进行修正. 可以得到

$$\begin{aligned} &z_k^{\mathrm{T}}M^{\mathrm{T}}PMz_k - z_{k+1}^{\mathrm{T}}M^{\mathrm{T}}PMz_{k+1} + \\ &\quad z_{k+1}^{\mathrm{T}}M^{\mathrm{T}}PMz_{k+1} - z_k^{\mathrm{T}}(AM - BK_1 - \\ &\quad DK_2)^{\mathrm{T}}P(AM - BK_1 - DK_2)z_k + \\ &\quad z_{k+1}^{\mathrm{T}}M^{\mathrm{T}}P(AM - BK_1 - DK_2)z_k - \\ &\quad z_k^{\mathrm{T}}(AM - BK_1 - DK_2)^{\mathrm{T}}PMz_{k+1} = \\ &\quad y_k^{\mathrm{T}}Qy_k + z_k^{\mathrm{T}}K_1^{\mathrm{T}}RK_1z_k - \gamma^2 z_k^{\mathrm{T}}K_2^{\mathrm{T}}K_2z_k \end{aligned} \tag{37}$$

式 (37) 能够用于 Off-policy 算法且为无模型算法主要基于如下两点考虑: 1) 式 (37) 引入 $z_{k+1}^{\mathrm{T}}M^{\mathrm{T}}\times PMz_{k+1}$ 是为了保证展开式中包含式 (32)中出现所有和系统相关的项, 而不需要再将系统动态代入式 (32) 中计算, 保证了该式最终可以实现无模型算法. 其次, 辅助项 $z_{k+1}^{\mathrm{T}}M^{\mathrm{T}}P(AM - BK_1 - DK_2)z_k$, $z_k^{\mathrm{T}}\times(AM - BK_1 - DK_2)^{\mathrm{T}}PMz_{k+1}$ 的引入并非没有意义, 而是此 Off-policy 算法能够应用的关键. 这一做法使展开后又进行同类项合并所得到的式 (39) 为最简形式, 从而减少了式 (43) 中未知项的个数, 大大减少了算法每一次迭代所需要的数据量与学习难度. 需要注意的是, 式 (37) 建立在 $z_{k+1}^{\mathrm{T}}MP(AM - BK_1 - DK_2)z_k = z_k^{\mathrm{T}}(AM - BK_1 - DK_2)^{\mathrm{T}}PMx_{k+1}$ 的基础上. 从式 (16) 中可轻易得出 P 为对称矩阵的结论, 因此, 式 (37) 显然成立.

将式 (1) 代入式 (37) 中, 展开后可得到

$$\begin{aligned} &z_k^{\mathrm{T}}M^{\mathrm{T}}PMz_k - z_{k+1}^{\mathrm{T}}M^{\mathrm{T}}PMz_{k+1} + \\ &\quad 2u_k^{\mathrm{T}}B^{\mathrm{T}}PAMz_k + 2z_k^{\mathrm{T}}K_1^{\mathrm{T}}B^{\mathrm{T}}PAMz_k - \\ &\quad u_k^{\mathrm{T}}B^{\mathrm{T}}PBK_1z_k + u_k^{\mathrm{T}}B^{\mathrm{T}}PBu_k - \\ &\quad z_k^{\mathrm{T}}K_1^{\mathrm{T}}B^{\mathrm{T}}PBK_1z_k + z_k^{\mathrm{T}}K_1^{\mathrm{T}}B^{\mathrm{T}}Pu_K + \\ &\quad 2w_k^{\mathrm{T}}D^{\mathrm{T}}PAMz_k + 2z_k^{\mathrm{T}}K_2^{\mathrm{T}}D^{\mathrm{T}}PAMz_k - \\ &\quad w_k^{\mathrm{T}}D^{\mathrm{T}}PBK_1z_k + w_k^{\mathrm{T}}D^{\mathrm{T}}PBu_k - \\ &\quad z_k^{\mathrm{T}}K_2^{\mathrm{T}}D^{\mathrm{T}}PBK_1z_k + z_k^{\mathrm{T}}K_2^{\mathrm{T}}D^{\mathrm{T}}PBu_k + \\ &\quad u_k^{\mathrm{T}}B^{\mathrm{T}}PDw_k - u_k^{\mathrm{T}}B^{\mathrm{T}}PDK_2z_k + \\ &\quad z_k^{\mathrm{T}}K_1^{\mathrm{T}}B^{\mathrm{T}}Pw_k - z_k^{\mathrm{T}}K_1^{\mathrm{T}}B^{\mathrm{T}}PDK_2z_k + \\ &\quad w_k^{\mathrm{T}}D^{\mathrm{T}}PDw_k - w_k^{\mathrm{T}}D^{\mathrm{T}}PDK_2z_k + \\ &\quad z_k^{\mathrm{T}}K_2^{\mathrm{T}}D^{\mathrm{T}}PDw_k - z_k^{\mathrm{T}}K_2^{\mathrm{T}}D^{\mathrm{T}}PDK_2z_k = \\ &\quad y_k^{\mathrm{T}}Qy_k + z_k^{\mathrm{T}}K_1^{\mathrm{T}}RK_1z_k - \gamma^2 z_k^{\mathrm{T}}K_2^{\mathrm{T}}K_2z_k \end{aligned} \tag{38}$$

合并同类项后可以得到

$$\begin{aligned}&z_k^{\mathrm{T}}M^{\mathrm{T}}PMz_k - z_{k+1}^{\mathrm{T}}M^{\mathrm{T}}PMz_{k+1} + \\&\quad 2(u_k + K_1z_k)^{\mathrm{T}}B^{\mathrm{T}}PAMz_k - \\&\quad (u_k + K_1z_k)^{\mathrm{T}}B^{\mathrm{T}}PB(K_1z_k - u_k) + \\&\quad 2(w_k + K_2z_k)^{\mathrm{T}}D^{\mathrm{T}}PAMz_k - \\&\quad (w_k + K_2z_k)^{\mathrm{T}}D^{\mathrm{T}}PB(K_1z_k - u_k) + \\&\quad (u_k + K_1z_k)^{\mathrm{T}}B^{\mathrm{T}}PD(w_k - K_2z_k) + \\&\quad (w_k + K_2z_k)^{\mathrm{T}}D^{\mathrm{T}}PD(w_k - K_2z_k) \;= \\&\quad y_k^{\mathrm{T}}Qy_k + z_k^{\mathrm{T}}K_1^{\mathrm{T}}RK_1z_k - \gamma^2z_k^{\mathrm{T}}K_2^{\mathrm{T}}K_2z_k \qquad (39)\end{aligned}$$

定义

$$\begin{aligned}&G_1 = M^{\mathrm{T}}PM, \quad G_2 = B^{\mathrm{T}}PAM, \quad G_3 = B^{\mathrm{T}}PB\\&G_4 = D^{\mathrm{T}}PAM, \quad G_5 = D^{\mathrm{T}}PB, \quad G_6 = B^{\mathrm{T}}PD\\&G_7 = D^{\mathrm{T}}PD \qquad (40)\end{aligned}$$

根据式 (32), 则目标策略可以通过 G_i $(i = 1, \cdots, 7)$推导而得.

$$\begin{aligned}K_1^* =&\Big(R + G_3 + G_2\big(\gamma^2I - G_7\big)^{-1}G_5\Big)^{-1} \times \\&\Big(G_2 + G_6\big(\gamma^2I - G_7\big)^{-1}G_4\Big)\\K_2^* =&\Big(G_7 - \gamma^2I - G_5(R + G_3)^{-1}G_6\Big)^{-1} \times \\&\Big(G_4 - G_5(R + G_3)^{-1}G_2\Big) \qquad (41)\end{aligned}$$

根据式 (18) 可将式 (39) 改写成最小二乘法可求解的迭代形式

$$\begin{aligned}&(z_k^{\mathrm{T}} \otimes z_k^{\mathrm{T}})\mathrm{vec}(M^{\mathrm{T}}P^{j+1}M) - (z_{k+1}^{\mathrm{T}} \otimes z_{k+1}^{\mathrm{T}}) \times \\&\quad \mathrm{vec}(M^{\mathrm{T}}P^{j+1}M) + 2(z_k^{\mathrm{T}} \otimes (u_k + K_1^jMz_k)^{\mathrm{T}}) \times \\&\quad \mathrm{vec}(B^{\mathrm{T}}P^{j+1}AM) - ((K_1^jMz_k - u_k)^{\mathrm{T}} \otimes \\&\quad (u_k + K_1^jMz_k))\mathrm{vec}(B^{\mathrm{T}}P^{j+1}B) + \\&\quad 2(z_k^{\mathrm{T}} \otimes (w_k + K_2^jMz_k)^{\mathrm{T}})\mathrm{vec}(D^{\mathrm{T}}P^{j+1}AM) - \\&\quad ((K_1^jMz_k - u_k)^{\mathrm{T}} \otimes (w_k + K_2^jMz_k)^{\mathrm{T}}) \times \\&\quad \mathrm{vec}(D^{\mathrm{T}}P^{j+1}B) + ((w_k - K_2^jMz_k)^{\mathrm{T}} \otimes \\&\quad (u_k + K_1^jMz_k)^{\mathrm{T}})\mathrm{vec}(B^{\mathrm{T}}P^{j+1}D) + \\&\quad ((w_k - K_2^jMz_k)^{\mathrm{T}} \otimes (w_k + K_2^jMz_k)^{\mathrm{T}}) \\&\quad \mathrm{vec}(D^{\mathrm{T}}P^{j+1}D) = y_k^{\mathrm{T}}C^{\mathrm{T}}QCy_k + z_k^{\mathrm{T}}M^{\mathrm{T}} \\&\quad (K_1^j)^{\mathrm{T}}RK_1^jMz_k - \gamma z_k^{\mathrm{T}}M^{\mathrm{T}}(K_2^j)^{\mathrm{T}}K_2^jMz_k \qquad (42)\end{aligned}$$

将历史的输入输出数据应用于最小二乘法得到 $G_{1\sim7}^j$. 根据 G^j 和式 (41) 得到策略 K_1^j 及 K_2^j. 多次迭代当策略收敛后得到最优目标策略. 至此, 得到算法 3.

算法 3. Off-policy 数据驱动输入输出反馈算法

1: 初始化策略 $j = 0$, K_1^j, K_2^j.

2: 将某一合适的策略代入系统中, 经过 L 次采样后产生满足训练需要的输入输出数据.

3: 根据式 (20), 将输入输出数据构造成增广数据向量 z_k.

4: 构造向量 Ψ_k, 计算 Φ, 如式 (43) 所示, Ψ_k 由式中各项横向堆叠而成. Φ_k 根据式 (44) 计算而得. 其构造依据为式 (42).

$$\begin{aligned}\Psi_k = [&z_k^{\mathrm{T}} \otimes z_k^{\mathrm{T}} - z_{k+1}^{\mathrm{T}} \otimes z_{k+1}^{\mathrm{T}},\\&2(z_k^{\mathrm{T}} \otimes (u_k + K_1^jMz_k)^{\mathrm{T}}),\\&-((K_1^jMz_k - u_k)^{\mathrm{T}} \otimes (u_k + K_1^jMz_k)),\\&2(z_k^{\mathrm{T}} \otimes (w_k + K_2^jMz_k)^{\mathrm{T}}),\\&-((K_1^jMz_k - u_k)^{\mathrm{T}} \otimes (w_k + K_2^jMz_k)^{\mathrm{T}}),\\&((w_k - K_2^jMz_k)^{\mathrm{T}} \otimes (u_k + K_1^jMz_k)^{\mathrm{T}}),\\&(w_k - K_2^jMz_k)^{\mathrm{T}} \otimes (w_k + K_2^jMz_k)^{\mathrm{T}}] \qquad (43)\end{aligned}$$

$$\begin{aligned}\Phi_k = &\; y_k^{\mathrm{T}}Qy_k + z_k^{\mathrm{T}}M^{\mathrm{T}}(K_1^j)^{\mathrm{T}}RK_1^jMz_k - \\&\gamma z_k^{\mathrm{T}}M^{\mathrm{T}}(K_2^j)^{\mathrm{T}}K_2^jMz_k \qquad (44)\end{aligned}$$

5: 根据式 (45) 最小二乘法, 计算可得到 $G_{1\sim7}^{j+1}$.

$$\mathrm{vec}(G^{j+1}) = (\Psi_{1\sim L}^T\Psi_{1\sim L})^{-1}\Psi_{1\sim L}\Phi_{1\sim L} \qquad (45)$$

其中,

$$\Phi_{1\sim L} = \begin{bmatrix}\Phi_1\\ \vdots \\ \Phi_L\end{bmatrix},\ \Psi_{1\sim L} = \begin{bmatrix}\Psi_1\\ \vdots \\ \Psi_L\end{bmatrix}.$$

$\Phi_{1\sim L}$ 由 Φ_1 到 Φ_L 向量纵向堆叠构成, $\Psi_{1\sim L}$ 由 Ψ_1 到 Ψ_L 向量纵向堆叠而构成, 用以通过最小二乘法求取未知项得到 $\mathrm{vec}(G^{j+1})$. 需要根据 $G_{1\sim7}^{j+1}$ 的规模, 将 $\mathrm{vec}(G^{j+1})$ 中的数据依次分配给 $G_{1\sim7}^{j+1}$.

6: 根据式 (41), 计算可得到 K_1^{j+1}, K_2^{j+1}. 判断是否达到最优值. 如果

$$||K_1^{j+1} - K_1^j|| < \epsilon \text{ 且 } ||K_2^{j+1} - K_2^j|| < \epsilon$$

则可认为达到最优值, 停止计算; 否则, $j = j + 1$, 并且回到步骤 2 继续迭代. 其中, ϵ 为某一极小值.

显而易见, 算法 3 是无模型算法, 从式 (45) 中可以看出, 推导策略所需要的 $G_{1\sim7}^j$ 完全由数据变换组合得到. 在步骤 2 中, L 的下限由式 (42) 中未知项的个数所决定, 也就是向量 Ψ 的维度. 式中的 N 可以按照式 (22) 的方法确定. 且当 $N = 1$ 时输入输出反馈算法与状态反馈算法具有相同的形式. 在步骤 6 中产生的策略 K_1^j 和 K_2^j 可以不参与实际

数据的产生, 而用于步骤 4 中 Φ_k 和 Ψ_k 向量的构造.

可以证明算法 3 收敛于式 (16) 所表示的最优解, 并得出定理 1.

定理 1. 当 $j\to\infty$ 时, 式 (42) 所得到的 P^j 收敛于式 (16) 所描述的 GARE 方程.

证明. 式 (42) 由式 (36) 所导出, 由此可以得到式 (42) 和式 (46) 等价. 等价原因与式 (42) 的导出过程一致, 在此不赘述.

$$\begin{aligned}&z_k^{\mathrm{T}}M^{\mathrm{T}}P^{j+1}Mz_k-z_k^{\mathrm{T}}(AM-BK_1^j-\\&\quad DK_2^j)^{\mathrm{T}}P^{j+1}(AM-BK_1^j-DK_2^j)z_k+\\&\quad z_{k+1}^{\mathrm{T}}M^{\mathrm{T}}P^{j+1}(AM-BK_1^j-DK_2^j)z_k-\\&\quad z_k^{\mathrm{T}}(AM-BK_1^j-DK_2^j)^{\mathrm{T}}P^{j+1}Mz_{k+1}=\\&\quad y_k^{\mathrm{T}}Qy_k+z_k^{\mathrm{T}}(K_1^j)^{\mathrm{T}}RK_1^jz_k-\gamma^2z_k^{\mathrm{T}}(K_2^j)^{\mathrm{T}}K_2^jz_k\end{aligned}\tag{46}$$

将式 (32) 代入到式 (46) 中替换 K_1^j 和 K_2^j, 可以得到式 (47). 因而, 可以看出 Off-policy 算法中, 每一次迭代的结果和产生数据所用策略无关.

$$\begin{aligned}P^{j+1}=&A^{\mathrm{T}}P^{j+1}A+\bar{Q}-\begin{bmatrix}A^{\mathrm{T}}P^jB & A^{\mathrm{T}}P^jD\end{bmatrix}\times\\&\begin{bmatrix}R+B^{\mathrm{T}}P^jB & B^{\mathrm{T}}P^jD\\D^{\mathrm{T}}P^jB & D^{\mathrm{T}}P^jD-\gamma^2I\end{bmatrix}^{-1}\begin{bmatrix}B^{\mathrm{T}}P^jA\\D^{\mathrm{T}}P^jA\end{bmatrix}\end{aligned}\tag{47}$$

在文献 [24–25] 中提及迭代式中 P 收敛于式 (16) 所示的黎卡提方程. 因此, 可知算法 3 的学习结果可收敛到和 GARE 方程等价的最优值. □

4　仿真结果及分析

在本节中, 以短时间周期飞机飞行姿态稳定控制模型为例, 对算法 3 的有效性与可靠性进行考察. H_∞ 算法在该模型下的有效性与必要性已经经过验证[29–30], 本文将仿真结果展示的重点放在收敛性的考察. 在收敛性方面, 本文从初值的选择与学习最优策略所用数据的影响两个方面来考察算法 3.

4.1　模型描述

在短时间段内飞机平稳飞行时, 主要考虑迎角、俯仰速角率和升降舵偏角三者对飞机飞行姿态造成的影响. 迎角是指飞机的速度矢量与机翼弦线之间的夹角. 俯仰角速率是指飞行器绕横轴旋转的角速度. 升降舵偏角是指飞机升降舵和平尾之间的夹角. 其中, 迎角和俯仰角速率可以直接衡量飞机的姿态变化. 升降舵偏角的变化会造成升降舵所受到的气动力矩发生变化, 进而影响飞机的飞行姿态. 因此, 将迎角、俯仰角速率和升降舵偏角三者作为系统的状态变量. 将控制升降舵变化的升降舵执行器电压作为控制变量. 考虑到飞机在平稳飞行时, 迎角易受到风向与风速的干扰, 将迎角所受到的扰动作为扰动变量. 由于飞机的俯仰速率测量比较困难, 因此可以将迎角与升降舵偏角作为输出变量. 飞机飞行示意图如图 1 所示[30].

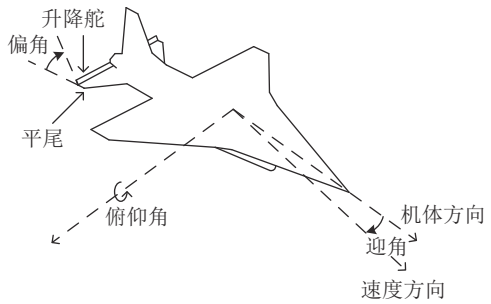


图 1　飞机飞行示意图

Fig. 1　Aircraft flight diagram

在文献 [29–30] 中描述了根据 F-16 战机实际参数构建的短周期模型, 其满足如下动态关系

$$\begin{aligned}\dot{x}(t)=&\begin{bmatrix}1.01887 & 0.90506 & 0.00215\\0.82225 & 1.07741 & 0.17555\\0 & 0 & 20.2\end{bmatrix}x(t)+\\&\begin{bmatrix}0\\0\\20.2\end{bmatrix}u(t)+\begin{bmatrix}1\\0\\0\end{bmatrix}w(t)\\y(t)=&\begin{bmatrix}1 & 0 & 0\\0 & 0 & 1\end{bmatrix}x(t)\end{aligned}\tag{48}$$

其中, $x(t)=[\alpha,q,\delta_e]^{\mathrm{T}}$ 是系统状态变量, 各分量分别表示如下: α 为飞行迎角; q 为俯仰角速率; δ_e 为升降舵偏角. 控制输入变量 $u(t)$ 为升降舵执行器电压. 扰动变量 $w(t)$ 为风速对迎角产生的扰动. $y(t)$ 为输出变量. 采用零阶保持器, 以 0.1 s 的采样时间对上述连续模型进行离散化处理, 进而得到离散系统模型.

$$\begin{aligned}x_{k+1}=&\begin{bmatrix}0.906 & 0.0816 & -5\times10^{-4}\\0.074 & 0.90121 & -7.08\times10^{-4}\\0 & 0 & 0.132\end{bmatrix}x_k+\\&\begin{bmatrix}4.44\times10^{-4}\\9.62\times10^{-3}\\8.67\times10^{-1}\end{bmatrix}u_k+\begin{bmatrix}9.51\times10^{-2}\\3.84\times10^{-3}\\0\end{bmatrix}w_k\\y_k=&\begin{bmatrix}1 & 0 & 0\\0 & 0 & 1\end{bmatrix}x_k\end{aligned}\tag{49}$$

选择一常用参数, $Q=\begin{bmatrix}1&0\\0&1\end{bmatrix}$, $R=1$, $\gamma=1$. 计算可得理论最优值

$$K_1=[-0.4722\quad 0.0286\quad 0.3828\quad 0.0048\quad 0.0322\quad 0.0040]$$

$$K_2=[-0.8558\quad -0.0002\quad 0.6788\quad -0.0003\quad -0.0002\quad 0.0072]$$

变换矩阵为

$$M=\begin{bmatrix}1&0&0&0&0&0\\11.04&0.0038&-9.9370&0.0043&0.0037&-0.1047\\0&0.4350&0&0.0750&0.4901&0\end{bmatrix}$$

4.2 仿真实验

本节从两个方面对算法 3 在迭代过程中的收敛性进行考察: 1) 对于不同的初值, 该算法皆可收敛于最优值; 2) 作为 Off-policy 算法, 对于不同的学习数据, 最后都会收敛于最优值. 为此, 设计三组仿真实验. 实验 1 作为对照组. 实验 2 和实验 3 分别改变实验 1 的初始策略与产生数据所用策略. 之后对照三组实验 K_1^j 和 K_2^j 的收敛路径.

由算法 3 中所描述, 取 N 为 2 时, 满足式 (22) 中所要求, 且需要确定 56 个未知参数, 取 L 为 90.

实验 1. 选择初始策略

$$K_1^0=\begin{bmatrix}-1&0&0&0&0&0\end{bmatrix}$$

$$K_2^0=\begin{bmatrix}-1&0&0&0&0&0\end{bmatrix}$$

产生数据所用的 u_k 与 w_k 选择为白噪声信号.

实验 2. 改变系统产生数据所用的策略, 选择 On-policy 的数据产生策略, 输入 $u_k=K_1^j z_k+n_1$, 扰动 $w_k=K_2^j z_k+n_2$, 其中, n_1 和 n_2 为探索噪声. 在此, n_1 和 n_2 选择为白噪声信号.

实验 3. 改变初始策略, 在 $[-4,4]$ 范围内随机选取 12 个数, 按顺序分配给 K_1^0 和 K_2^0, 得到

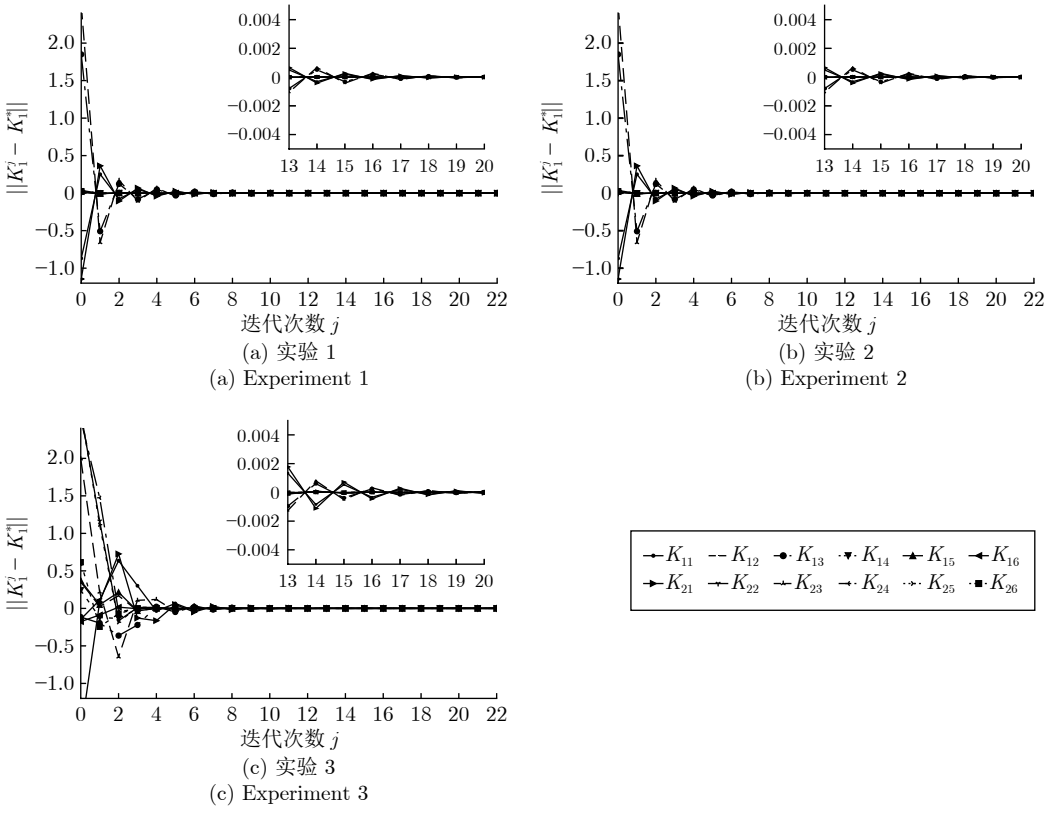
$$K_1^0=\begin{bmatrix}-1.3&3.9&0.8&1.3&-2.3&0.2\end{bmatrix}$$

$$K_2^0=\begin{bmatrix}-2.9&0.1&3.4&0.6&3.8&2.1\end{bmatrix}$$

产生数据的策略同实验 1.

实验结果如图 2 和图 3 所示. 经过仿真, 三组实验策略学习结果皆收敛于

$$K_1=[-0.4722\quad 0.0286\quad 0.3828\quad 0.0048\quad 0.0322\quad 0.004]$$



(a) 实验 1
(a) Experiment 1

(b) 实验 2
(b) Experiment 2

(c) 实验 3
(c) Experiment 3

图 2 三组实验参数收敛曲线

Fig. 2 Three groups of experimental parameters convergence curves

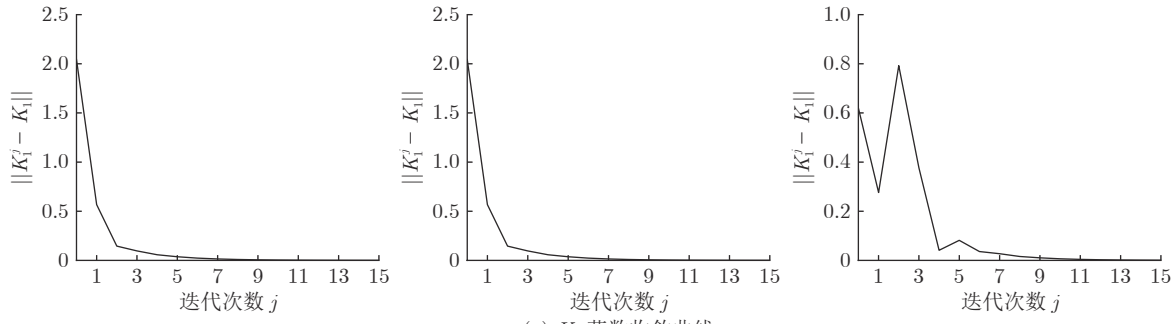
$$K_2 = [-0.86 \quad -0.0002 \quad 0.68 \quad -0.0003 \\ -0.0002 \quad 0.0072]$$

策略收敛轨迹如图 2 所示, 横坐标为系统迭代次数, 纵坐标表示策略中每个参数在迭代过程中的值与理论最优值的差值. 各子图右上角为接近最优值时收敛曲线的放大图. 可以看出三组实验皆可快速收敛于最优值附近. 从放大图中可以看出, 三组算法在第 17 次迭代之后的误差基本可以忽略不计.

为了更进一步对三组实验的策略收敛轨迹进行比较. 以最终得到的策略 K_1 和 K_2 作为基准观察每次迭代中 $||K_1^j - K_1||$, $||K_2^j - K_2||$ 的变化. 如图 3 所示, 图像的横坐标是迭代次数, 纵坐标是每次迭代的策略与最终策略之差的范数, 范数的收敛可以看作策略的收敛. 在实验 3 中, 第 3 个迭代周期中 K_1 范数收敛曲线发生了跳变. 由于该算法在迭代过程中同时学习 K_1 和 K_2 两个策略, 在第 3 个周期 K_1 发生了跳变, 而 K_2 范数收敛曲线大幅度下降, 因此, K_1 的跳变并不破坏算法的收敛效果. 从实验 1 和实验 2 的收敛轨迹可以看出, 两个算法的收敛轨迹一致, 佐证了定理 1 中, 算法 3 收敛轨迹与产生数据所用策略无关的结论. 从实验 1 和实验 3 的收敛轨迹可以看出, 选取一个和目标策略相差较大的初值最终仍然可以收敛于目标策略. 证明了该算法的收敛性与有效性, 且针对不同的初始策略, 该算法具有一定的鲁棒性.

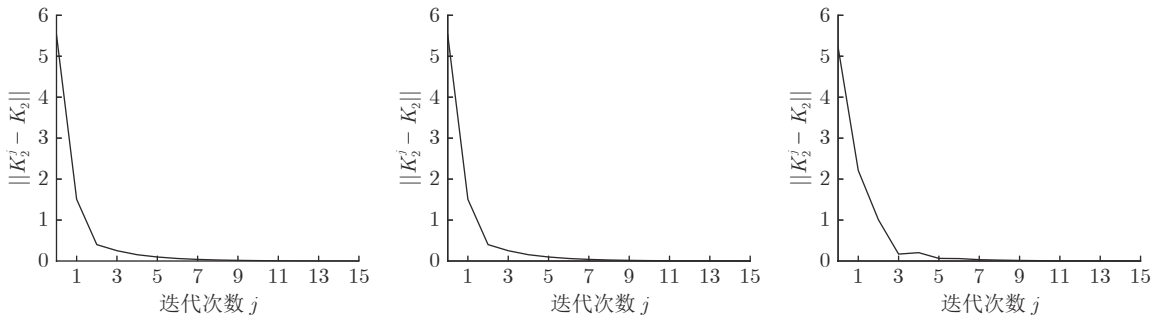
5　结束语

本文针对考虑扰动作用的线性离散系统, 提出了一种基于 Off-policy 的无模型输入输出数据反馈 H_∞ 控制方法. 该算法针对性解决了状态数据反馈算法难以应用于状态无法测得的应用环境这一问题, 通过引入一种由输入输出数据组成的增广数据向量将状态反馈在线策略迭代算法转换成输入输出反馈算法. 并通过引入辅助项的方法, 最终将输入输出反馈在线策略迭代算法转换为无模型输入输出反馈 Off-policy 算法. 该算法和 On-policy 算法所学习得出的策略相同, 且该算法在学习过程中所用的数据默认和在迭代过程中所更新的策略所产生的数据之间存在差异, 这为数据驱动的离线算法和在迭代更新策略稳定的情况下再更新策略提供了可能. 最后, 通过 F-16 飞行器仿真模型验证了该算法的收敛性与有效性. 在本篇文章的基础上, 仍有一些十分重要的问题值得进行研究, 如 输入受限情况下的控制问题. 考虑在网络控制中, 存在反馈数据



(a) K_1 范数收敛曲线

(a) The norm of K_1 convergence curve



(b) K_2 范数收敛曲线

(b) The norm of K_2 convergence curve

图 3　三组实验范数收敛曲线

Fig. 3　Three groups of experimental parameters convergence curves

丢包情况下的控制问题. 更进一步将本文所研究的内容延展到非线性系统中, 以提高本算法的应用范围与控制效果.

References

1 Zames G. Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Transactions on Automatic Control*, 1981, **26**(2): 301−320

2 Basar T, Bernhard P. *H_∞ Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Boston, MA, USA: Birkhauser. 1995.

3 Isidori A, Lin W. Global L_2-gain design for a class of nonlinear systems. *Systems and Control Letters*, 1998, **34**(5): 295−302

4 Isidori A, Kang W. H_∞ control via measurement feedback for affine nonlinear system. *IEEE Transactions on Automatic Control*, 1995, **40**(3): 466−472

5 Jacobson D H. On values and strategies for infinite-time linear quadratic games. *IEEE Transactions on Automatic Control*, 1977, **22**(3): 490−491

6 Lewis F L, Vrabie D L, Syrmos V L. *Optimal Control* (*3rd Edition*). New York: Wiley & Sons Inc, 2012.

7 Liu Qiang, Zhuo Jie, Lang Zi-Qiang, Qin S. Joe. Perspectives on data-driven operation monitoring and selfoptimization of industrial processes. *Acta Automatica Sinica*, 2018, **44**(11): 1944−1956
(刘强, 卓洁, 郎自强, 秦泗钊. 数据驱动的工业过程运行监控与自优化研究展望. 自动化学报, 2018, **44**(11): 1944−1956)

8 Hou Zhong-Sheng, Xu Jian-Xin. On data-driven control theory: the state of the art and perspective. *Acta Automatica Sinica*, 2009, **35**(6): 650−667
(侯钟生, 许建新. 数据驱动控制理论及方法的回顾和展望. 自动化学报, 2009, **35**(6): 650−667)

9 Howard R. Dynamic Programming and Markov Processes [Ph.D. dissertation], Massachusetts Institute of Technology, 1960.

10 Lewis F L, Vrabie D. Vamvoudakis K G. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine*, 2012, **32**(6): 76−105

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

12 Chi Rong-Hu, Hou Zhong-Sheng, Huang Biao. Optimal iterative learning control of batch processes: from modelbased to data-driven. *Acta Automatica Sinica*, 2017, **43**(6): 917−932
(池荣虎, 侯忠生, 黄彪. 间歇过程最优迭代学习控制的发展: 从基于模型到数据驱动. 自动化学报, 2017, **43**(6): 917−932)

13 Doya K. Reinforcement learning in continuous-time and space. *Neural Computationm*, 2000, **12**(1): 219−45

14 Wu Qian, Fan Jia-Lu, Jiang Yi, Chai Tian-You. Data-driven dual-rate control for mixed separation thickening process in a wireless network environment. *Acta Automatica Sinica*, 2019, **45**(6): 1128−1141
(吴倩, 范家璐, 姜艺, 柴天佑. 无线网络环境下数据驱动混合选别浓密过程双率控制方法. 自动化学报, 2019, **45**(6): 1128−1141)

15 Jiang Y, Fan J L, Chai T Y, Li J N, Lewis F L. Data-driven flotation industrial process operational optimal control based on reinforcement learning. *IEEE Transactions on Industrial Informatics*, 2018, **14**(5): 1974−1989

16 Jiang Y, Fan J L, Chai T Y, Lewis F L. Dualrate operational optimal control for flotation industrial process with unknown operational madel. *IEEE Transactions on Industrial Electronics*, 2019, **66**(6): 4587−4599

17 Xue W Q, Fan J L, Lopez V G, Li J N, Jiang Y, Chai T Y, Lewis F L. New methods for optimal operational control of industrial processes using reinforcement learning on multiple time-scales. *IEEE Transactions on Industrial Informatics*, 2020, **16**(5): 3085−3099

18 Jiang Y, Fan J L, Chai T Y, Lewis F L, Li J N. Tracking control for linear discrete-time networked control systems with unknown dynamics and dropout. *IEEE Transactions on Neural Networks and Learning Systems*, 2018, **29**(10): 4607−4620

19 Jiang Y, Kiumarsi B, Fan J L, Chai T Y, Li J N, Lewis F L. Optimal output regulation of linear discrete-time systems with unknown dynamics using reinforcement learning. *IEEE Transactions on Cybernatics*, 2020, **50**(7): 3147−3156

20 Luo B, Liu D R, Huang T W, Wang D. Model-free optimal tracking control via critic-only Q-learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2016, **27**(10): 2134−2144

21 Luo B, Yang Y, Liu D R. Adaptive Q-learning for data-based optimal output regulation with experience replay. *IEEE Transactions on Cybernetics*, 2018, **48**(12): 3337−3347

22 Wang D, Ha M M, Qiao J F. Self-learning optimal regulation for discrete-time nonlinear systems under event-driven formulation. *IEEE Transactions on Automatic Control*, 2020, **65**(3): 1272−1279

23 Wang D, He H B, Liu D R. Adaptive critic nonlinear robust control: A survey. *IEEE Transactions on Cybernetics*, 2017, **47**(10): 3429−3451

24 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. *Automatica*, 2007, **43**: 473−481

25 Kiumarsi B, Lewis F L, Jiang Z P. H-infinity control of linear discrete-time systems: off-policy reinforcement learning. *Automatica*, 2017, **37**(1): 144−152

26 Luo B, Wu H N, Huang T W. Off-policy reinforcement learning for H_∞ control design. *IEEE Transactions on Cybernetics*, 2014, **45**(1): 65−67

27 Li H L, Liu D R, Wang D. Integral reinforcement learning for linear continuous-time zero-sum games with completely unknown dynamics. *IEEE Transactions on Automation Science and Engineering*, 2014, **11**(3): 706−714

28 Brewer J W. Kronecker products and matrix calculus in system theory. *IEEE Transactions on Circuits and Systems*, 1978, **25**(9): 772−781

29 Jiu-Hoon K, Lewis F L. Model-free H_∞ control design for unknown linear discrete-time systems via Q-learning with LMI.

Automatica, 2010, **46**(8): 1320−1326

30 Stevens B L, Lewis F L, Johnson E N. *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, (*3rd Edition*). New York: Wiley, 2015. 516−529

李 臻 东北大学流程工业综合自动化国家重点实验室硕士研究生. 主要研究方向为工业过程运行控制, 强化学习. E-mail: alilili0131@gmail.com

(**LI Zhen** Master student at the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University. His research interest covers industrial process operational control and reinforcement learning.)

范家璐 东北大学流程工业综合自动化国家重点实验室副教授. 2011 年获得浙江大学博士学位 (与美国宾夕法尼亚州立大学联合培养). 主要研究方向为工业过程运行控制, 工业无线传感器网络与移动社会网络. 本文通信作者. E-mail: jlfan@mail.neu.edu.cn

(**FAN Jia-Lu** Associate professor at the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University. She received her Ph.D. degree in control science and engineering from Zhejiang University in 2011. She was a visiting scholar with the Pennsylvania State University during 2009 to 2010. Her research interest covers industrial process operational control and reinforcement learning. Corresponding author of this paper.)

姜 艺 东北大学流程工业综合自动化国家重点实验室博士研究生. 2016 年获得东北大学控制理论与控制工程硕士学位. 主要研究方向为工业过程运行控制, 网络控制, 自适应动态规划, 强化学习.
E-mail: JY369356904@163.com

(**JIANG Yi** Ph.D. candidate at the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University. He received his master degree in control theory and engineering from Northeastern University in 2016. His research interest covers industrial process operational control, networked control, adaptive dynamic programming, and reinforcement learning.)

柴天佑 中国工程院院士, 东北大学教授, IEEE Fellow, IFAC Fellow. 1985 年获得东北大学博士学位. 主要研究方向为自适应控制, 智能解耦控制, 流程工业综合自动化理论、方法与技术.
E-mail: tychai@mail.neu.edu.cn

(**CHAI Tian-You** Academician of Chinese Academy of Engineering, professor at Northeastern University, IEEE Fellow, IFAC Fellow. He received his Ph.D. degree from Northeastern University in 1985. His research interest covers adaptive control, intelligent decoupling control, and integrated automation theory, method and technology of industrial process.)