江南大学

硕 士 学 位 论 文

（学 术 学 位）

题　　目：基于强化学习的迭代学习控制与优化方法研究

英文并列题目：Research on Reinforcement Learning Based Iterative Learning Control and Optimization Methods

研 究 生 姓 名：王　瑞

专　　业：控制科学与工程

研 究 方 向：控制理论与控制工程

导 师 姓 名：陶洪峰 教授

指导小组成员：

学位授予日期：2024 年 6 月

答辩委员会主席：陈　莹

江 南 大 学

地址：无锡市蠡湖大道 1800 号

二〇二四 年 六 月

摘 要

迭代学习控制是一种适用于在有限时间内多次执行相同任务系统的智能控制算法，其核心思想是利用先前批次储存的控制输入与跟踪误差等系统信息，来不断地修正输入信号，以驱动控制系统跟踪上期望参考轨迹。将优化理论和迭代学习控制结合，可以使得迭代学习控制器的性能进一步提升，从而实现更快速精确的跟踪。

与迭代学习控制的特点类似，从历史信息中学习并调整动作也是强化学习的主要思想。强化学习是机器学习的一个分支，旨在让智能体与环境交互并不断调整其策略，学习特定任务中的最优策略，从而在一系列的步骤中最大化累计奖励。将强化学习处理复杂决策的优势与迭代学习控制结合，有助于提升迭代学习控制器的性能并拓宽迭代学习控制的应用场景。

本文探索了强化学习与迭代学习控制结合的可能性，开展了基于强化学习的迭代学习控制与优化方法研究，具体工作内容和创新点包括：

1. 针对一类线性离散系统的跟踪问题，在非提升范数优化框架下，提出了一种基于值迭代的迭代学习控制方法。首先，将迭代学习控制过程描述为马尔可夫决策过程，引入强化学习的未来收益指导当下动作的思想。通过求解状态值函数的优化问题，得到迭代学习控制更新律，提出了系统在该学习律下渐近稳定和跟踪误差单调收敛的条件并给出了相应证明。进一步地，分析了所提方法在计算复杂度方面的优势。最后，通过直流电动机的仿真，验证了所提方法的有效性和优势。

2. 针对一类模型信息未知的线性离散系统的跟踪问题，在非提升范数优化框架下，提出了一种基于 Q 学习的无模型迭代学习控制方法。首先，将迭代学习控制过程描述为马尔可夫决策过程。接着，引入 Q 学习算法，通过求解 Q 函数的优化问题，得到包含模型信息的迭代学习控制更新律，并利用可测数据通过最小二乘方法求解更新律所需的模型信息，从而实现无需模型参数的迭代学习控制方法，证明了该方法的收敛性。进一步地，分析了所提方法在计算复杂度和求解模型所需实验批次数量方面的优势。最后，通过直流电动机的仿真，验证了所提方法的有效性和优势。

3. 针对一类随时间和批次同时变化的执行器故障下的线性离散系统的跟踪问题，提出了一种基于 Q 学习的故障估计迭代学习容错控制方案。沿着时间和批次同时变化的未知故障会影响迭代学习控制器的跟踪性能，针对以上问题，引入了 Q 学习算法，将故障估计过程转化为马尔可夫决策过程，通过持续调整设计的故障估计器以适应变化的故障。同时，采用范数优化理论设计迭代学习容错控制器，通过 Q 学习的故障估计结果来调整控制器，以抵消故障的影响，提出了系统在该学习律下跟踪误差有界收敛的条件并给出了相应证明。最后，通过移动机器人的仿真，验证了所提方法的有效性和优势。

关键词：迭代学习控制；强化学习；范数优化；Q 学习；容错控制

Abstract

Iterative learning control is an intelligent control algorithm for systems executing the same task multiple times within a finite interval, whose core idea is to continuously modify the input signals by incorporating system information such as control input and tracking error from the previous trials with the purpose of driving the systems to follow the desired reference. By integrating of the optimization theory with iterative learning control, an iterative learning controller with further enhanced performance can be obtained to realize faster and more precisely trajectory tracking.

Similar to iterative learning control, learning from historical information and adjusting actions is also the main idea of reinforcement learning. Reinforcement learning is a branch of machine learning, aiming at enabling the agent to interact with the environment and continuously adjust their policies. Over a series of steps, a maximized cumulative reward can be obtained by learning the optimal policy for specific tasks. Reinforcement learning possesses the ability to handle complex decision-making tasks. Combining this advantage of reinforcement learning and iterative learning control problem contributes to enhancing the performance of iterative learning controller and broadening the application scenarios of iterative learning control to accommodate various task requirements.

This paper explores the possibility of combining reinforcement learning with iterative learning control, and carries out the research on reinforcement learning based iterative learning control and optimization methods. The main research contents and innovations are as follows:

1. For the tracking problem of a class of linear discrete-time systems, under the framework of non-lifted norm-optimal technique, a value iteration-based iterative learning control algorithm is proposed. Firstly, the iterative learning control process is described as a Markov decision process, and the concept of the future reward guiding the present action from reinforcement learning is introduced. By solving the optimization problem of the value function, iterative learning control update law is derived. Conditions for the asymptotic stability of the system and the monotonic convergence of tracking error are proposed and correspondingly proved. Moreover, the computational complexity is analyzed. Finally, the effectiveness and advantage of the proposed algorithm are verified through the simulation on the plant of a DC motor.

2. For the tracking problem of a class of linear discrete-time systems with unknown system information, under the framework of non-lifted norm-optimal technique, a Q-learning based model-free iterative learning control algorithm is proposed. Firstly, the iterative learning control process is described as a Markov decision process, and Q-learning algorithm is introduced, and by solving the optimization problem of the Q-function, iterative learning control update law containing model information is derived. Utilizing measurable data and applying the least square method yield the model information required for the update laws, thus achieving a model-free method without the need of model parameters. The convergence of the proposed algorithm is proven. Moreover, the computational complexity and the

required experimental trial number for solving the model information are analyzed. Finally, the effectiveness and advantage of the proposed algorithm are verified through the simulation on the plant of a DC motor.

3. For the tracking problem of a class of linear discrete-time systems under actuator faults varying with both time and trial axes, a Q-learning based fault estimation and iterative learning fault-tolerant control scheme is proposed. Unknown faults varying with both time and trial axes pose a challenge to the tracking performance of iterative learning controller. To address this issue, Q-learning algorithm is introduced. The fault estimation process is described as a Markov decision process. The designed fault estimator is continuously adjusted to adapt the changing fault. Moreover, an iterative learning fault-tolerant controller is designed using norm optimization theory, where the controller is adjusted based on the fault estimation results from Q-learning to counteract the influence of faults. Conditions for the bounded convergence of tracking error under the proposed iterative learning control update law is provided and proved. Finally, the effectiveness and advantages of the proposed algorithm are verified through the simulation on the plant of a mobile robot.

Keywords: iterative learning control; reinforcement learning; norm-optimization; Q-learning; fault-tolerant control

主要符号表

符号	意义
$\mathbb{N}$	自然数集合
$\mathbb{R}$	实数集合
$\mathbb{R}^n$	n 维实数列向量的集合
$\mathbb{R}^{n\times m}$	$n\times m$ 维实数矩阵的集合
I_n	$n\times n$ 维单位矩阵
$\left\|A\right\|$	矩阵 A 的欧几里得（Euclidean）范数
A^{-1}	矩阵 A 的逆
$A^{\dagger}$	矩阵 A 的伪逆
A^{T}	矩阵 A 的转置
$\rho\left(A\right)$	矩阵 A 的谱半径
$\mathrm{vec}\left(A\right)$	将矩阵 A 按列堆叠成的列向量
$X\otimes Y$	矩阵 X 与矩阵 Y 的克罗内克积（Kronecker Product）
$\mathrm{diag}\left\{A_1,\ldots,A_n\right\}$	分块对角矩阵
$\left\|x\right\|_R^2=x^{\mathrm{T}}Rx$	在希尔伯特（Hilbert）空间中权重矩阵 R 定义的 x 的诱导范数
$\left\langle x,y\right\rangle_R=x^{\mathrm{T}}Ry$	在希尔伯特（Hilbert）空间中权重矩阵 R 定义的 x,y 的内积
$\ell_2^n\left[a,\ b\right]$	定义在区间 $\left[a,\ b\right]$ 上的在 $\mathbb{R}^n$ 取值的勒贝格（Lebesgue）平方可和序列空间

目 录

第一章　绪论

1.1 研究背景及意义

迭代学习控制（Iterative Learning Control）旨在模仿人类“循序渐进”的学习过程，其核心思想是利用先前批次储存的控制输入与跟踪误差等系统信息，来不断修正当前批次的输入信号，以减小当前批次的跟踪误差，使得跟踪误差在一定批次后收敛至一个极小的阈值，从而驱动控制系统逐步快速地跟踪上给定的参考轨迹[1-3]。作为智能控制方法之一，迭代学习控制适用于在有限时间内执行给定重复任务的系统，在结构简单与性能高效方面有着明显优势。将优化理论和迭代学习控制结合，可以使得迭代学习控制器的性能进一步提高，从而实现更快速精确的跟踪。目前在实际工程中已经有广泛应用，如机器人操作系统[4]、闭环胰岛素泵系统[5]、电机控制系统[6]、高速公路交通控制系统[7]、工业注塑成型系统[8]等。

与迭代学习控制的特点类似，从历史信息中学习并调整当前动作同样是强化学习（Reinforcement Learning）的主要思想。强化学习的产生源自控制论、学习心理学、统计学等多学科的交叉，是一种重要的机器学习算法[9,10]。强化学习旨在模仿人类“通过与环境交互”的学习过程，其中，学习者或决策者定义为“智能体”，智能体之外与其交互的事物定义为“环境”，以试错和延迟收益为特征，让智能体与环境交互，智能体根据环境反馈的奖励调整其动作，并权衡试探和开发，从而逐步地学习并优化其决策策略，最终得到使数值化的累计奖励最大化的最优策略[11]。区别于迭代学习控制适用于具有重复性质的系统，由于强化学习面对复杂问题时拥有出众的决策能力，目前，强化学习在需要策略指导的复杂非确定系统已经得到广泛应用，如水下机器人寻迹与避障[12]、自然语言处理[13]、自动驾驶[14]、游戏策略[15]、金融量化交易等[16]。

迭代学习控制和强化学习有许多相似之处，均受启发于自然界中生物的学习行为，并拥有从获取的历史信息中学习并修正行动以达到目标的能力。二者又各有其特点，迭代学习控制适用于存在重复运行特性的学习目标，具有结构简单、性能高效的特点，强化学习适用于复杂的动态学习目标，具有适应性强的特点和自主学习的优势。因此，将二者结合是一个有潜力且具有挑战性的研究问题，本文主要研究使用强化学习方法解决迭代学习控制问题，而将强化学习方法应用于迭代学习控制存在一个不可避免的讨论点，即如何建立二者的联系[17,18]。将强化学习的无需先验知识与处理复杂决策的优势与迭代学习控制问题相结合，有助于提升迭代学习控制的性能和拓宽迭代学习控制的应用场景，但如何发挥强化学习的优势，同时不与迭代学习控制在思想或目标层面发生冲突，是使用强化学习方法解决迭代学习控制问题的关键点。

使用强化学习解决迭代学习控制问题，一方面，可以考虑将强化学习与迭代学习控制在控制算法设计层面结合，即使用强化学习方法直接设计迭代学习控制器，目的在于将强化学习未来收益指导当前动作的思想直接引入迭代学习控制更新律，并且在某些方法中可以将算法拓展为无模型形式[19]。这种结合方式既能充分利用强化学习的适应性和

学习能力，又能借助控制算法对系统动力学的建模和分析，从而同时发挥二者优势，进一步提升迭代学习控制的性能潜能。

另一方面，可以考虑使用强化学习方法协助解决迭代学习控制的固有局限性，即改进迭代学习控制的某一部分。迭代学习控制适用于具有重复运行性质的系统，一些非重复因素，如随着批次变化的故障或扰动等，会对迭代学习控制器的性能带来不利影响[20]。强化学习对于复杂非确定系统有适应性与学习能力，具有解决动态非重复因素的能力。因此，使用强化学习方法解决迭代学习控制的固有局限性，既能充分利用强化学习的对于复杂不确定性的适应能力，又可以充分发挥迭代学习控制对于重复跟踪问题的高效性能，从而发挥二者的综合优势，最终在复杂情况下实现更好的控制效果，并拓宽迭代学习控制的应用边界。

本文研究基于强化学习的迭代学习控制与优化方法，意义在于探索强化学习和迭代学习控制结合的可能性，从而使用强化学习方法提升迭代学习控制的性能潜能与价值创造，以拓宽迭代学习控制的应用场景。因此，本课题在理论方面有较大的研究发展空间与深远的研究意义，同时在实际应用方面也有广阔的应用前景。

1.2 迭代学习控制与优化方法研究现状

作为智能控制算法的主要分支之一，迭代学习控制有着悠久的发展历史，追溯回1978 年，Uchiyama 第一次提出其控制思想[21]，但是受限于语言没有得到广泛关注。之后，在 1984 年，Arimoto 等对迭代学习控制理论进一步完善，将其应用于机器人控制系统的跟踪任务，并得到了高精度的控制结果[22]，这使得迭代学习控制开始进入国际视野，相关的理论研究逐步发展成熟，并得到广泛应用，成为多个领域具有重复运行性质的控制系统的重要控制方法之一[23,24]。

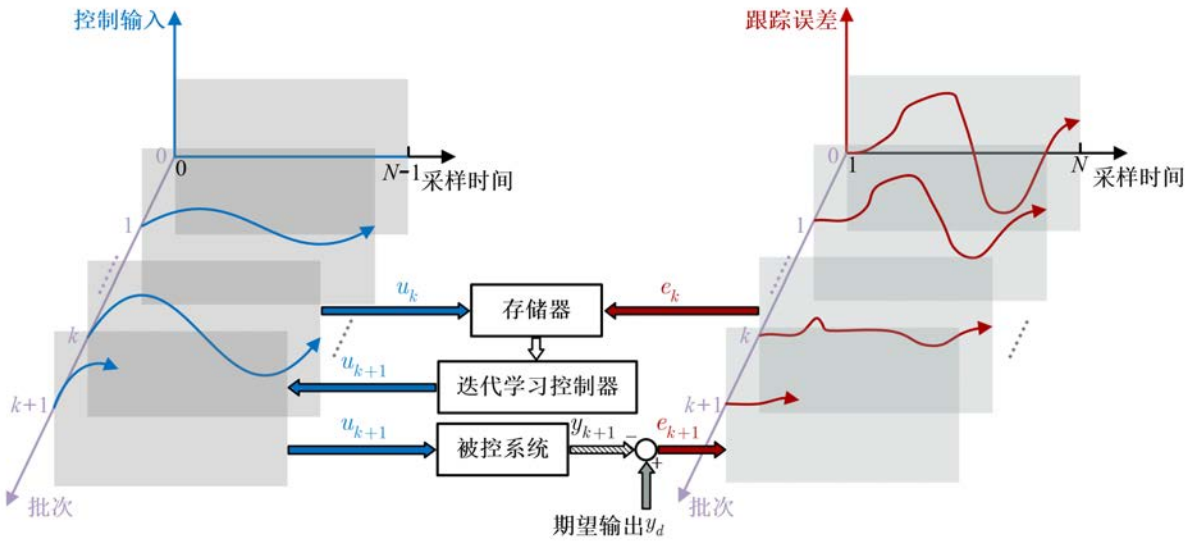

图 1-1 迭代学习控制的控制过程图

迭代学习控制的控制过程如图 1-1，其中，u_k 为第 k 批次的控制输入信号，y_k 为第 k 批次的系统输出信号，y_d 为给定的控制系统期望输出，定义 $e_k = y_d - y_k$ 为第 k 批次的跟踪误差。迭代学习控制的控制过程可以描述为：拥有重复运行特性的控制系统的控制

过程可以分为沿时间轴与沿批次轴两个方向。在每一个迭代批次，系统的控制输入信号u_k与跟踪误差信号e_k会储存至存储器，通过预先设计的迭代学习控制器进行运算，并生成下一批次的控制输入信号u_{k+1}，在第$k+1$批次，控制输入信号u_{k+1}作用于被控系统得到输出y_{k+1}，并与系统期望输出y_d作差得到跟踪误差e_{k+1}，如此循环往复，最终，随着迭代批次k的增加，控制输入信号u_k被不断修正，使得跟踪误差e_k逐渐减小，直至收敛到一个极小的阈值，即系统输出信号y_k跟踪上给定的系统期望输出y_d。可以看出，与反馈控制不同，迭代学习控制的本质是前馈控制，但由于迭代学习控制律中引入了系统历史批次的信息，因此可以实现良好的跟踪效果。

基于可重复性的要求，在使用传统的迭代学习控制算法解决轨迹跟踪问题时，通常假设控制系统满足以下条件[25]：

（1）控制系统在每个批次的运行时间是有限且固定的，即$t \in \left[0,\ N\right]$，$N > 0$。

（2）给定的系统期望输出y_d在每个迭代批次是固定的。

（3）系统初始状态$x_k(0)$在每个迭代批次是恒定的，即$\forall k$，$x_k(0) = x_0$。

（4）系统的动力学在每个迭代批次是不变的。

（5）系统输出信号y_k在每个迭代批次是可测的，以获取跟踪误差e_k，从而与控制输入u_k共同参与下一批次的控制输入u_{k+1}的更新。

（6）存在期望控制输入u_d，使得系统在u_d的作用下可以完成给定的跟踪任务。

迭代学习控制研究的核心是迭代学习控制更新律的设计。与传统的时间维度的 PID 控制算法类似，学者们利用跟踪误差的不同形式，设计了不同的迭代学习控制律结构，如 P 型、D 型、PD 型和 PID 型迭代学习控制律[26-30]，以提高系统的跟踪效果。为了引入更多批次的系统历史信息，学者们设计了高阶迭代学习控制律[31,32]，以进一步提升跟踪性能。此类传统的迭代学习控制算法结构简单、参数易调节，因此，相关理论仍在持续发展，并得到广泛应用。

跟踪性能的提升是迭代学习控制的一个研究焦点，有学者将优化理论与迭代学习控制相结合，转化控制任务为优化问题，通过优化算法求解最优控制输入，以获得更好的跟踪效果。其中，典型的优化迭代学习控制算法包括梯度下降法[33]、牛顿法[34]、范数优化迭代学习控制算法[35]、参数优化迭代学习控制算法[36]等。梯度下降法常用于解决线性优化问题，文献[37]采用梯度下降法设计了一个分布式迭代学习控制算法，使跟踪误差沿批次方向不断减小，并且误差减小的速率与选取的学习步长有关。梯度下降法的收敛速度呈线性变化，收敛速度较慢，而牛顿法可以进一步解决非线性优化问题并提高跟踪误差的收敛速度，文献[38]使用牛顿法设计了一个用于非线性双轴控制系统的动态轮廓误差估计的迭代学习控制算法。范数优化迭代学习控制算法由英国学者 Amann 提出，核心思想是设计跟踪误差、输入变化等多控制目标的二次性能指标函数并将其最小化以得到每批次最优的输入信号[39]，该算法具有良好的跟踪精度和鲁棒性，是目前应用较为广泛的优化迭代学习控制方法。文献[40]通过在每一个迭代批次使用跟踪误差对参考信

号进行修改，设计了一个加速收敛的低增益范数优化迭代学习控制算法。在范数优化迭代学习控制框架的基础上，Owens 进一步提出了参数优化迭代学习控制算法，通过设计控制增益和跟踪误差的二次性能指标函数并将其最小化得到每批次最优的输入信号，虽然该方法有较优的跟踪性能，但存在系统矩阵与本身的转置之和必须是正定的限制[41]。由于范数优化框架下设计的迭代学习控制算法相对于其他优化迭代学习控制算法具有较优的跟踪性能和较强的鲁棒性，本文将在范数优化的框架下使用强化学习方法解决迭代学习控制问题。

同时，从以上工作可以看出优化迭代学习控制算法多需要系统准确的模型信息，通过优化基于已知模型信息设计的性能指标，完成优化控制任务。如果系统存在未知信息，如未知的系统参数或未知的故障，则未知信息会将不确定因素带入确定的系统动力学中，从而影响算法的跟踪性能，这也是本文将考虑解决的问题。

进一步地，为了克服单一控制方法的固有缺陷或提升控制性能来拓展迭代学习控制的应用范围，将迭代学习控制与其他控制方法和智能方法结合，如预测控制[42]、自适应控制[43]、神经网络[44]、强化学习[45]等，也是当前的一个研究热点。

1.3 基于强化学习的迭代学习控制研究现状

迭代学习控制与强化学习的共同优点是对于既定目标的学习能力，区别在于迭代学习控制适用于学习具有重复运行特性的既定目标，强化学习适用于在复杂动态环境下的探索和学习既定目标。同时发挥强化学习和迭代学习控制的优势，并让二者保持在学习目标和思想上的一致性，是目前使用强化学习方法解决迭代学习控制问题的关键点。

目前，强化学习与迭代学习控制的结合的工作仍处于起步和探索阶段，其中，一类工作注重强化学习与迭代学习控制在控制算法设计方面的结合，如何在控制算法对系统动态的建模和分析过程中，找到强化学习动作和状态的映射关系是算法设计层面结合的关键点。Zhang 等将强化学习思想引入范数优化框架下的迭代学习控制，基于值迭代方法设计了迭代学习控制律，并进一步引入 Q 学习，通过一定的实验批次进行系统模型求解以将所提算法拓展为无模型算法[46]。Song 针对线性离散系统，提出了一种基于 Q 学习的数据驱动迭代学习控制，随着迭代批次的进行，更新控制增益，设计了一种无模型的变增益迭代学习控制[47]。Shi 等提出了一种基于深度确定性策略梯度算法设计的非线性系统迭代学习控制算法，在无需系统模型信息的情况下通过训练得到最优控制律[48]。后续，Liu 等进一步提出了使用 SAC（Soft Actor-Critic）算法设计的适用于非线性系统的迭代学习控制算法，在无需系统模型信息并且存在非重复扰动的情况下，训练得到的优化控制律仍能较好地跟踪上给定的控制任务[49]。Meindl 等尝试将强化学习思想用于未知非线性系统的重复性轨迹跟踪迭代学习控制的设计，并应用于两轮倒立摆机器人，赋予了该机器人在学习上的自主性[50]。Poot 等提出了一个基于行动者-评判家的迭代学习控制框架，在无需模型信息的情况下更新控制输入[51]。此类结合方式一般随着迭代批次的增加，可以达到无需模型信息并且近似基于模型算法的跟踪性能，但是这类方法需要将一定数量的迭代批次用于强化学习的探索，这通常会导致迭代的初期算法性能的下降

和跟踪误差的收敛速度过慢等问题。因此，如何引入一定机制以降低强化学习的固有的探索与利用特性对迭代学习控制的快速收敛优势的影响，是此类结合方式的重要问题。

另一类结合方法尝试间接发挥强化学习的复杂决策优势来协助处理迭代学习控制的局限性，如调节参数、补偿非重复因素的影响等。Liu 等针对非线性强耦合的四旋翼无人机系统，提出了使用强化学习方法调节 PD 型迭代学习控制律的学习增益参数的强化迭代学习控制方法[52]。Xu 等使用强化学习方法补偿 P 型迭代学习控制的控制输入，提升了 P 型迭代学习控制算法在非重复扰动下的鲁棒性[53]。Liu 等针对具有模型不匹配和非重复特性的复杂间歇过程，设计了由迭代学习控制器和深度强化学习补偿器组成的控制方案，并在传统的在线训练方法的基础上提出了一种实时实现方案[54]。Ruan 等提出了一种运动控制方案，将运动控制分为基于强化学习的轨迹优化和基于迭代学习控制的定位控制，在有效减少过程时间的同时，保证了系统的稳定性[55]。Vuga 等尝试结合迭代学习控制的快速收敛特性和强化学习的鲁棒性，使两种方法均在优点上得到保留，同时又克服各自的局限性，并将算法应用于上半身人形机器人系统[56]。Nemec 等尝试使用强化学习方法改进自适应迭代学习控制中的自适应过程，以提高未知或部分已知环境的迭代学习控制性能[57]。此类结合方式大多可以利用强化学习在复杂动态系统的适应性，协助迭代学习控制保持良好的跟踪性能。因此，如何找到二者的有效结合点，并保持学习目标的一致性是需要注意的问题。

强化学习在参数估计方面有少量研究，准确估计迭代学习控制过程中的未知不利影响（如未知的故障）并进行补偿，可以进一步提升迭代学习控制在复杂情形下的控制性能。Jin 等针对网络安全系统使用强化学习方法设计安全状态估计算法[58]；Li 等在系统模型存在未知参数情况下，仅利用可测量数据，使用强化学习方法寻求最优观测器增益以实现状态估计[59]。Herman 等提出了一种基于梯度的逆强化学习方法，通过解决组合优化问题，来同时估计奖赏与系统动态[60]。Shahrabi 等结合强化学习方法，进行动态作业车间调度工作中的参数估计，以提升调度方法的性能[61]。如何将使用强化学习方法进行参数估计并应用于迭代学习控制，在估计信息的基础上进一步提升控制性能是一个有意义的探索方向。

总体而言，从以上工作可以看出基于强化学习的迭代学习控制方法的关键点在于发挥二者优势并降低二者的相互负面影响，本文将从以上两类结合方式出发，在保持学习目标一致性的同时，解决现有结合方式存在的问题，以及探索强化学习与迭代学习控制新的结合点。

1.4 本文主要研究内容

为探索使用强化学习解决迭代学习控制问题的可能性，本文将围绕基于强化学习的迭代学习控制与优化方法开展研究，研究主要分为两个方面，分别为结合强化学习和迭代学习控制设计控制算法和使用强化学习协助解决迭代学习固有局限性，本文具体思路与结构如图 1-2 所示。

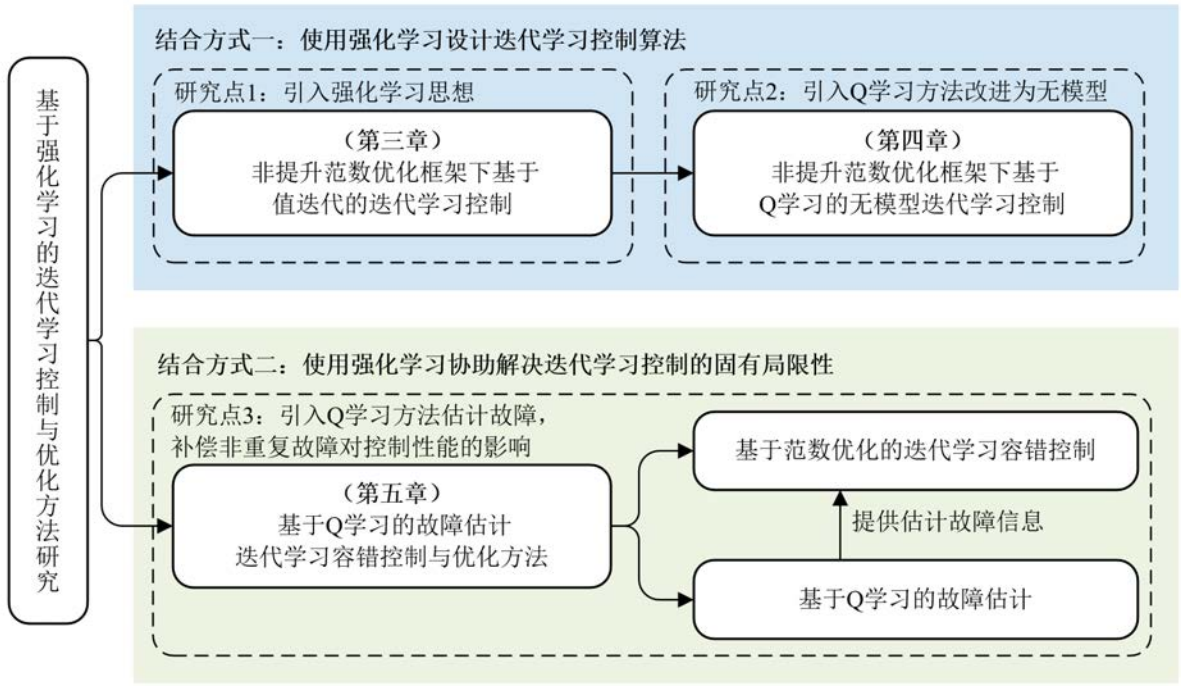

图 1-2 论文研究思路与结构

本文共有六个章节，具体内容如下：

第一章，绪论部分，阐释了本课题研究背景及意义，介绍了迭代学习控制研究现状和迭代学习控制与优化方法研究现状，并且，从强化学习与迭代学习控制两类结合方式出发分别对现有的基于强化学习的迭代学习控制工作进行梳理，给出了本文的研究思路和具体内容安排。

第二章，预备知识部分，介绍了迭代学习控制的范数优化理论，以及强化学习的基础框架，为后续研究的展开提供理论基础。

第三章，研究非提升范数优化框架下基于值迭代的迭代学习控制问题。针对一类线性离散系统的跟踪问题，在非提升范数优化框架下，首先，将迭代学习控制过程描述为马尔可夫决策过程，引入强化学习的未来收益指导当下动作的思想，通过求解状态值函数的优化问题，得到迭代学习控制更新律，提出了系统在该学习律下渐近稳定和跟踪误差单调收敛的条件并给出了相应证明，进一步地，分析了所提方法在计算复杂度方面的优势。最后，通过直流电动机的仿真，验证了所提方法的有效性和优势。

第四章，研究非提升范数优化框架下基于 Q 学习的无模型迭代学习控制问题。针对一类模型信息未知的线性离散系统的跟踪问题，在非提升范数优化框架下，首先，将迭代学习控制过程描述为马尔可夫决策过程。引入 Q 学习算法，通过求解 Q 函数的优化问题，得到包含模型信息的迭代学习控制更新律，并利用可测数据通过最小二乘方法求解更新律所需的模型信息，从而实现无需模型参数的迭代学习控制方法，证明了该方法的收敛性。进一步地，分析了所提方法在计算复杂度和求解模型所需实验批次数量方面的优势。最后，通过直流电动机的仿真，验证了所提方法的有效性和优势。

第五章，研究基于 Q 学习的故障估计迭代学习容错控制与优化问题。针对一类随时间和批次同时变化的执行器故障下的线性离散系统的跟踪问题，提出了一种基于 Q 学习的故障估计迭代学习容错控制方案。沿着时间和批次同时变化的未知故障会影响迭代学

习控制器的跟踪性能，针对以上问题，将故障估计过程转化为马尔可夫决策过程，通过持续调整设计的故障估计器以适应变化的故障。并且，采用范数优化理论设计迭代学习容错控制器，通过 Q 学习的故障估计结果来调整控制器，以抵消故障的影响，提出了系统在该学习律下跟踪误差有界收敛的条件并给出了相应证明。最后，通过移动机器人的仿真，验证了所提方法的有效性和优势。

第六章，结论与展望部分，对全文的研究内容进行总结，并对后续可进一步探索与研究的工作进行展望。

第二章　预备知识

本章为后续研究的开展提供理论基础，包括迭代学习控制的范数优化理论、强化学习的基础理论相关知识。

2.1 迭代学习控制的范数优化理论

考虑如下一类线性时不变离散系统

$$\begin{cases} x_k(t+1) = Ax_k(t) + Bu_k(t), \\ y_k(t) = Cx_k(t), \end{cases} \tag{2.1}$$

其中，下标k表示迭代批次；$t \in \left[0,\ N\right]$表示一个重复运行周期T内的采样时刻，N为一个批次的采样点个数；$x_k(t) \in \mathbb{R}^n$，$u_k(t) \in \mathbb{R}^m$，$y_k(t) \in \mathbb{R}^l$分别代表系统在第k批次的状态变量、控制输入以及输出信号；A，B，C分别表示具有相应维数的系统参数矩阵；为保证系统输出可控，需要满足CB满秩；$x_k(0)$表示系统在第k批次运行时的初始状态值，假设系统在不同批次的初始状态值相同，即$\forall k$，$x_k(0) = x_0$。接下来将围绕系统(2.1)分别介绍提升范数优化框架与非提升范数优化框架下的迭代学习控制的基础理论与区别。

2.1.1 提升范数优化框架下的迭代学习控制

迭代学习控制关注迭代域的跟踪性能，因此，利用提升技术将系统(2.1)转换成超向量形式的提升模型

$$y_k = Gu_k + d, \tag{2.2}$$

其中，系统矩阵$G \in \mathbb{R}^{lN \times mN}$和初始状态响应$d \in \mathbb{R}^{lN}$定义为

$$G = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \cdots & CB \end{bmatrix},$$

$$d = \begin{bmatrix} (CA)^{\mathrm{T}} & (CA^2)^{\mathrm{T}} & (CA^3)^{\mathrm{T}} & \cdots & (CA^N)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} x_0.$$

输入信号向量$u_k \in \ell_2^m\left[0,\ N-1\right]$和输出信号向量$y_k \in \ell_2^l\left[1,\ N\right]$分别定义为

$$u_k = \left[u_k^{\mathrm{T}}(0),\ u_k^{\mathrm{T}}(1),\ \cdots,\ u_k^{\mathrm{T}}(N-1)\right]^{\mathrm{T}},$$

$$y_k = \left[y_k^{\mathrm{T}}(1),\ y_k^{\mathrm{T}}(2),\ \cdots,\ y_k^{\mathrm{T}}(N)\right]^{\mathrm{T}}.$$

输入 Hilbert 空间$\ell_2^m\left[0,\ N-1\right]$和输出 Hilbert 空间$\ell_2^l\left[1,\ N\right]$分别由下述内积和对应的诱导范数定义为

$$\left\langle u,\ v\right\rangle_{R_1} = u^{\mathrm{T}} R_1 v,\ \left\|u\right\|_{R_1} = \sqrt{\left\langle u,\ u\right\rangle_{R_1}}, \tag{2.3}$$

$$\left\langle y,\ z\right\rangle_{Q_1} = y^{\mathrm{T}} Q_1 z,\ \left\|y\right\|_{Q_1} = \sqrt{\left\langle y,\ y\right\rangle_{Q_1}}, \tag{2.4}$$

其中，$R_1 \in \mathbb{R}^{mN\times mN}$ 和 $Q_1 \in \mathbb{R}^{lN\times lN}$ 是对称正定权重矩阵。

迭代学习控制的传统控制目标是在重复的任务中逐步修正控制输入信号 u_k，最终，输入信号 u_k 会收敛到唯一的期望输入 u_d，输出信号 y_k 会跟踪上给定的期望输出信号 y_d，这意味着跟踪误差 e_k 会收敛到 0，即

$$\lim_{k\to\infty} u_k = u_d,\ \lim_{k\to\infty} e_k = 0, \tag{2.5}$$

其中，跟踪误差 e_k 定义为

$$e_k = y_d - y_k. \tag{2.6}$$

同时，根据式(2.2)和式(2.6)，给定的期望输出信号 y_d 可以用 u_d 表示为

$$y_d = Gu_d + d. \tag{2.7}$$

提升技术用超向量的形式在迭代域中重新描述被控系统，这简化了时间域中的计算，并为引入优化理论到迭代学习控制框架中提供了许多便利。为了实现迭代学习控制任务目标，引入范数优化方法来优化每个批次的多目标性能指标[62]。性能指标定义为

$$J_{k+1} \triangleq \left\|y_d - Gu_{k+1} - d\right\|_{Q_1}^2 + \left\|u_{k+1} - u_k\right\|_{R_1}^2, \tag{2.8}$$

其中，性能指标由两部分组成：跟踪误差和批次间的输入信号变化。最小化跟踪误差的目的是完成迭代学习控制的任务目标，即跟踪上期望输出信号。减小批次间输入信号变化，即使得批次间输入信号变化平滑是为了增加算法的鲁棒性。$Q_1 \in \mathbb{R}^{lN\times lN}$ 和 $R_1 \in \mathbb{R}^{mN\times mN}$ 分别为跟踪误差和批次间输入信号变化的对称正定权重矩阵，即 $Q_1 = {Q_1}^{\mathrm{T}} > 0$，$R_1 = {R_1}^{\mathrm{T}} > 0$，用以表示优化过程中跟踪误差减小和鲁棒性的优先级，不失一般性，可以取权重矩阵 $Q_1 = q_1 I_{lN}$，$R_1 = r_1 I_{mN}$。

通过最小化性能指标，即 $\partial J_{k+1} / \partial u_{k+1} = 0$，可得提升框架下迭代学习控制更新律为

$$u_{k+1} = u_k + \left(G^{\mathrm{T}} Q_1 G + R_1\right)^{-1} G^{\mathrm{T}} Q_1 e_k. \tag{2.9}$$

提升范数优化框架下方法求解得到的学习律包含矩阵 $\left(G^{\mathrm{T}} Q_1 G + R_1\right) \in \mathbb{R}^{mN\times mN}$ 的逆，这使得迭代学习控制更新律的计算复杂度为 $O(N^3)$，也就是说，学习律的计算复杂度通常会随着采样点个数 N 呈 3 次方阶趋势增加，因此，会存在高计算复杂度导致的采样点个数限制问题。目前，有一些降低使用提升技术的范数优化迭代学习算法的计算复杂度的工作，如使用非提升范数优化框架[63]，接下来将介绍非提升范数优化框架下的迭代学习控制。

2.1.2 非提升范数优化框架下的迭代学习控制

非提升范数优化框架下的迭代学习控制不将系统沿时间轴提升，而是分开考虑每个时间点的性能优化，因此，跟踪误差$e_k(t)$可以定义为

$$e_k(t) = y_d(t) - y_k(t). \tag{2.10}$$

与此同时，非提升范数优化框架下的迭代学习控制的目标与传统的迭代学习控制一致，即在重复的任务中逐步修正控制输入信号$u_k(t)$，最终，输入信号$u_k(t)$会收敛到唯一的期望输入$u_d(t)$，输出信号$y_k(t)$会跟踪上给定的期望输出信号$y_d(t)$，这意味着跟踪误差$e_k(t)$会收敛到0，即

$$\lim_{k\to\infty} u_k(t) = u_d(t), \ \lim_{k\to\infty} e_k(t) = 0. \tag{2.11}$$

为了实现迭代学习控制任务目标，引入范数优化方法来优化每个批次每个时间点的多目标性能指标。性能指标定义为

$$J_{k+1}(t) \triangleq \left\| y_d(t+1) - CAx_{k+1}(t) - CBu_{k+1}(t) \right\|_{Q_2}^2 + \left\| u_{k+1}(t) - u_k(t) \right\|_{R_2}^2, \tag{2.12}$$

其中，性能指标由两部分组成：跟踪误差和批次间的输入信号变化。$Q_2 \in \mathbb{R}^{l\times l}$和$R_2 \in \mathbb{R}^{m\times m}$分别为跟踪误差和批次间输入信号变化的对称正定权重矩阵，以表示优化过程中误差减小和鲁棒性的优先级，即$Q_2 = Q_2^{\mathrm{T}} > 0$，$R_2 = R_2^{\mathrm{T}} > 0$。不失一般性，可以取权重矩阵$Q_2 = q_2 I_l$，$R_2 = r_2 I_m$。

通过最小化性能指标，即$\partial J_{k+1}(t) / \partial u_{k+1}(t) = 0$，可得非提升范数优化框架下迭代学习控制更新律为

$$\begin{aligned} u_{k+1}(t) = u_k(t) &+ \left(B^{\mathrm{T}}C^{\mathrm{T}}Q_2CB + R_2 \right)^{-1} B^{\mathrm{T}}C^{\mathrm{T}}Q_2 e_k(t+1) \\ &- \left(B^{\mathrm{T}}C^{\mathrm{T}}Q_2CB + R_2 \right)^{-1} B^{\mathrm{T}}C^{\mathrm{T}}Q_2CA\Delta x_{k+1}(t), \end{aligned} \tag{2.13}$$

其中，$\Delta x_{k+1}(t) = x_{k+1}(t) - x_k(t)$为批次间的状态变化。由此可见，通过非提升范数优化框架下求解得到的迭代学习控制更新律包含矩阵$\left(B^{\mathrm{T}}C^{\mathrm{T}}Q_2BC + R_2 \right) \in \mathbb{R}^{m\times m}$的逆，而该矩阵的维度是不随着采样点个数$N$的增加而变化的常数，因此，非提升范数优化框架下迭代学习控制更新律在一个批次内的计算复杂度$O(N)$，相较于提升范数优化框架下的计算复杂度$O(N^3)$，有计算负担小的优势。关于提升范数优化框架与非提升范数优化框架区别的进一步的讨论请参考文献[63]。

除了迭代学习控制更新律的计算负担小的优势，本文引入非提升范数优化框架，目的在于给使用强化学习方法直接设计的迭代学习控制算法带来额外优势，即降低强化学习的固有的探索与利用特性对迭代学习控制的快速收敛优势的影响，具体内容将在后续第三章和第四章展开。

2.2 强化学习基础理论

强化学习的框架结构图如图 2-1 所示，强化学习问题由智能体（Agent）、环境（Environment）、状态（State）、动作（Action）和奖励（Reward）组成。智能体是学习者和决策者，环境是与智能体进行交互的事物，包含除智能体之外的一切。随着交互的进行，对于每一步交互 $n \in 0,\ 1,\ 2,\ \cdots$，智能体在状态 $\mathcal{S}_n$ 选择动作 $\mathcal{A}_n$。而后，智能体转移到状态 $\mathcal{S}_{n+1}$，并在 $(n+1)$ 步从环境中获得奖励 $\mathcal{R}_{n+1}$。最终，智能体通过学习最优策略来最大化累计奖励。

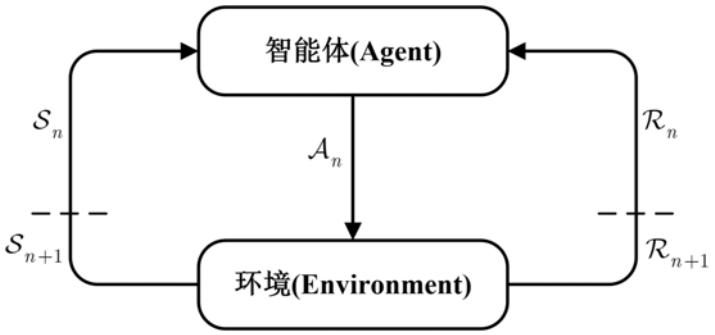

图 2-1 强化学习框架结构图

强化学习任务通常由马尔可夫决策过程（Markov Decision Process）描述，典型的有限马尔可夫决策过程通常由如下所示五元组表示 $(\mathcal{S}, \mathcal{A}, p, \mathcal{R}, \gamma)$：

- $\mathcal{S}$ 为状态空间，其中状态 $s \in \mathcal{S}$，第 n 步的状态定义为 $\mathcal{S}_n \in \mathcal{S}$；
- $\mathcal{A}$ 为动作空间，其中动作 $a \in \mathcal{A}$，在第 n 步状态 s 选择的动作定义为 $\mathcal{A}_n \in \mathcal{A}(s)$；
- p 为状态转移函数，在传统强化学习算法中通常定义为状态 $s \in \mathcal{S}$ 转移到下一状态 $s' \in \mathcal{S}$ 的概率，定义为

$$p(s', r \mid s, a) = Pr\{\mathcal{S}_{n+1} = s', \mathcal{R}_{n+1} = r \mid \mathcal{S}_n = s, \mathcal{A}_n = a\},$$

其中，p 是一个由四个参数决定的方程，满足 $p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to [0,\ 1]$。而在使用强化学习解决控制问题时，状态转移函数通常为被控系统动态的建模和分析，可以具体到当前状态、选择动作和下一状态的关系等式；

- $\mathcal{R}$ 为收益函数，第 $(n+1)$ 步的即时收益定义为 $\mathcal{R}_{n+1}$；
- γ 为折扣因子，决定了未来收益的现在价值，并且 $\gamma \in (0,1]$。

马尔可夫过程可以用于描述智能体与环境的交互过程，即智能体在某状态 s 下执行某动作 a 后可能获得的收益值 r 和下一个状态 s'，具体过程如图 2-2 所示。

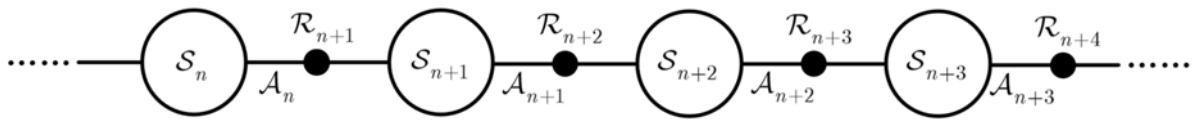

图 2-2 马尔可夫决策过程示意图

图 2-2 的交互过程中，累计回报定义为

$$G_n = R_{n+1} + \gamma R_{n+2} + \cdots = \sum_{i=0}^{\infty} \gamma^i R_{n+i+1}. \tag{2.14}$$

智能体的学习目标是，在与环境的交互中更新累计回报(2.14)，逐步优化选择的动作，最终得到最优策略，可以表示为

$$\pi\big(a \mid s\big) = P\big(a_t = a \mid s_t = s\big), \tag{2.15}$$

最优策略用于指导特定状态下选择最优动作，以实现累计回报最大化的学习目标。

本小节为强化学习的基础理论，后续第三章、第四章与第五章将涉及到一些强化学习算法，会以此框架为基础展开。其中，第三章和第四章将迭代学习控制过程转化为马尔可夫决策过程，使用强化学习方法设计迭代学习控制律，第五章将故障估计过程转化为马尔可夫决策过程，使用强化学习方法设计故障估计算法，为迭代学习容错控制律提供估计故障信息。

第三章　非提升范数优化框架下基于值迭代的迭代学习控制

3.1 引言

为解决工业过程中的具有重复运行特性系统的控制问题，传统的范数优化框架通常使用历史系统信息设计性能指标函数[64,65]，为引入强化学习未来收益指导当前动作的思想，本章首先在范数优化框架下引入值迭代方法设计迭代学习控制更新律。同时，目前使用强化学习方法直接设计迭代学习控制器的结合的工作，一般在提升范数优化框架下完成，此类算法在基于模型信息的情形下，会存在高计算复杂度带来的采样点个数限制问题。因此，本章在提升范数优化框架下基于值迭代的迭代学习控制算法[46]的基础上，进一步引入了文献[63]中的非提升范数优化框架，以降低算法在采样点数量多的情形下的计算复杂度。

本章针对一类具有重复运行特性的线性时不变离散系统进行研究，主要内容和结构如下：第 3.2 节介绍了非提升范数优化框架下迭代学习控制算法设计问题；第 3.3 节在非提升范数优化框架下，首先将迭代学习控制过程描述为马尔可夫决策过程，引入值迭代方法，通过求解状态值函数的优化问题得到迭代学习控制更新律，接着提出了系统在该更新律下渐近稳定和跟踪误差单调收敛的条件并给出了相应证明，最后介绍了提升范数优化框架下基于值迭代的迭代学习控制算法，给出了本章所提方法与该算法的计算复杂度对比分析；第 3.4 节通过直流电动机的仿真，验证了所提方法的有效性和优势；第 3.5 节对本章内容进行小结。

3.2 问题描述

考虑如下一类线性时不变离散系统

$$\begin{cases} x_k(t+1) = Ax_k(t) + Bu_k(t), \\ y_k(t) = Cx_k(t), \end{cases} \tag{3.1}$$

其中，下标 k 表示迭代批次；$t \in \left[0,\ N\right]$ 表示一个重复运行周期 T 内的采样时刻，N 为一个批次的采样点个数；$x_k(t) \in \mathbb{R}^n$，$u_k(t) \in \mathbb{R}^m$，$y_k(t) \in \mathbb{R}^l$ 分别代表系统在第 k 批次的状态变量、控制输入以及输出信号；A，B，C 分别表示具有相应维数的系统参数矩阵；为保证系统输出可控，需要满足 CB 满秩；$x_k(0)$ 表示系统在第 k 批次运行时的初始状态值，假设系统在不同批次的初始状态值相同，即 $\forall k$，$x_k(0) = x_0$。

假设 $y_d(t)$ 为期望参考轨迹，那么系统存在一个期望控制输入 $u_d(t) \in \mathbb{R}^m$，并且期望控制输入 $u_d(t)$ 可以驱动系统在有限时间 $t \in \left[0,\ N\right]$ 内跟踪上期望参考轨迹，即

$$\begin{cases} x_d(t+1) = Ax_d(t) + Bu_d(t), \\ y_d(t) = Cx_d(t), \end{cases} \tag{3.2}$$

其中，$x_d(t)$ 代表了系统的期望状态。迭代学习控制律的设计目标是调节下一个迭代批次的输入 $u_{k+1}(t)$，并使其逐渐收敛于 $u_d(t)$。

定义跟踪误差为

$$e_k(t) = y_d(t) - y_k(t). \tag{3.3}$$

定义 3.1（非提升范数优化框架下迭代学习控制算法设计问题） 设计一个多目标性能指标函数并将其优化，得到如下类似式(2.13)形式的基于模型信息的迭代学习控制更新律

$$u_{k+1}(t) = f\big(u_k(t),\ e_k(t+1),\ \Delta x_{k+1}(t),\ A,\ B,\ C\big), \tag{3.4}$$

其中，下一批次的输入信号 $u_{k+1}(t)$ 由当前批次的输入信号 $u_k(t)$、当前批次的跟踪误差 $e_k(t+1)$ 和批次间状态变化 $\Delta x_{k+1}(t)$ 组成。系统在式(3.4)所示的迭代学习控制更新律作用下，可以实现迭代学习控制的跟踪目标，即

$$\lim_{k\to\infty} u_k(t) = u_d(t),\ \ \lim_{k\to\infty} e_k(t) = 0. \tag{3.5}$$

3.3 基于值迭代的迭代学习控制

3.3.1 优化控制算法设计

根据跟踪误差的定义式(3.3)，可得

$$e_{k+1}(t+1) = e_k(t+1) - CA\Delta x_{k+1}(t) - CB\Delta u_{k+1}(t), \tag{3.6}$$

其中，批次间输入变化定义为 $\Delta u_{k+1}(t) = u_{k+1}(t) - u_k(t)$。

与文献[46]中的描述过程类似，将迭代学习控制过程中的误差动态方程(3.6)描述为马尔可夫决策过程，从而使得迭代学习控制问题可以用强化学习方法求解。定义如下五元组 $\big(\mathcal{S},\mathcal{A},f,\mathcal{R},\gamma\big)$：

- $\mathcal{S}$ 为状态空间，状态 $s\in\mathcal{S}$ 定义为跟踪误差 $e_k(t+1)\in\mathbb{R}^l$；
- $\mathcal{A}$ 为动作空间，动作 $a\in\mathcal{A}$ 定义为动作向量 $\Delta w_{k+1}(t)\in\mathbb{R}^n$，其中，动作向量定义为

$$w_k(t) \triangleq Ax_k(t) + Bu_k(t), \tag{3.7}$$

$$\Delta w_{k+1}(t) \triangleq A\Delta x_{k+1}(t) + B\Delta u_{k+1}(t); \tag{3.8}$$

- f 为状态转移函数，定义为 $s' = f(s,a)$，即

$$e_{k+1}(t+1) = f(e_k(t+1),\ \Delta w_{k+1}(t)) = e_k(t+1) - C\Delta w_{k+1}(t); \tag{3.9}$$

- $\mathcal{R}$ 为收益函数，定义为

$$\mathcal{R}\big(e_k(t+1),\ \Delta w_{k+1}(t)\big) = \big\|e_k(t+1)\big\|_Q^2 + \big\|\Delta w_{k+1}(t)\big\|_R^2, \tag{3.10}$$

其中，$Q\in\mathbb{R}^{l\times l}$ 和 $R\in\mathbb{R}^{n\times n}$ 为对称正定权重矩阵；

- γ 为折扣因子，决定了智能体的目光长远程度，即未来收益的现在价值，并且 $\gamma\in\big(0,1\big]$。

当前批次的状态值函数 $V\big(e_k(t+1)\big)$ 定义为

$$V\left(e_k(t+1)\right)=\sum_{j=0}^{\infty}\gamma^j\mathcal{R}\left(e_{k+j}(t+1),\ \Delta w_{k+1+j}(t)\right). \tag{3.11}$$

值得注意的是，在本章的设计问题中，状态值函数$V\left(e_k(t+1)\right)$可以看作迭代学习控制的性能指标函数$J_{k+1}(t)$，因此，定义$J_{k+1}(t)=V\left(e_k(t+1)\right)$。

迭代学习控制算法设计目标是求解使性能指标$J_{k+1}(t)$最小化的最优输入$\Delta u_{k+1}^{*}(t)$，强化学习的设计目标是通过最小化状态值函数$V\left(e_k(t+1)\right)$，为下一动作找到最佳策略即最佳动作$\Delta w_{k+1}^{*}(t)$。在第$k+1$批次的第t时刻，$\Delta x_{k+1}(t)$为已知常量，因此，

$$\Delta w_{k+1}^{*}(t)=A\Delta x_{k+1}(t)+B\Delta u_{k+1}^{*}(t),$$

即迭代学习控制的算法设计目标与强化学习的设计目标一致。

因此，由马尔可夫决策过程描述的迭代学习控制算法设计目标是通过最小化式(3.11)，从而为下一动作$\Delta w_{k+1}(t)$找到最佳策略，同时得到最优输入，即

$$\min_{\Delta u_{k+1}(t)}\left\{J_{k+1}(t)\right\}\Leftrightarrow\min_{\Delta w_{k+1}(t)}\left\{V\left(e_k(t+1)\right)\right\}. \tag{3.12}$$

注释 3.1 收益函数式(3.10)参考非提升范数优化框架下的迭代学习控制的性能指标(2.12)定义，由两部分组成，分别为跟踪误差$e_k(t+1)$和动作向量$\Delta w_{k+1}(t)$，其中，动作向量$\Delta w_{k+1}(t)$包含了批次间控制输入变化$\Delta u_{k+1}(t)$和批次间状态变化$\Delta x_{k+1}(t)$，由于批次间状态变化$\Delta x_{k+1}(t)$为常量，因此，性能指标中动作向量$\Delta w_{k+1}(t)$的部分主要是为了引入批次间控制输入变化$\Delta u_{k+1}(t)$来提高算法鲁棒性。分别用对称正定阵$Q\in\mathbb{R}^{l\times l}$和$R\in\mathbb{R}^{n\times n}$来表示其优先级，即$Q=Q^{\mathrm{T}}>0$，$R=R^{\mathrm{T}}>0$。不失一般性，可以取权重矩阵$Q=qI_l$，$R=rI_n$。并且，$e_k(t+1)$和$\Delta w_{k+1}(t)$的诱导范数定义为：

$$\left\|e_k(t+1)\right\|_Q^2=e_k^{\mathrm{T}}(t+1)Qe_k(t+1), \tag{3.13}$$

$$\left\|\Delta w_{k+1}(t)\right\|_R^2=\Delta w_{k+1}^{\mathrm{T}}(t)R\Delta w_{k+1}(t). \tag{3.14}$$

注释 3.2 强化学习的学习目标是为了最大化累计回报，因此，通常将背离学习目标的收益定义为负值，靠近学习目标的收益定义为正值。而在强化学习与优化控制算法结合时，收益函数通常只依靠传统代价函数衡量，因此，为了保证二者的优化目标一致，并简化推导过程，通常将由代价函数定义的背离学习目标的收益定义为正值，从而修改强化学习的学习目标为最小化累计回报即最小化值函数，来和优化控制的最小化性能指标的目标保持一致[66]。

以上马尔可夫决策过程可以看作一个 LQR（Linear Quadratic Regulator）问题，因此状态值函数是二次型的，即

$$V\left(e_k(t+1)\right)=e_k^{\mathrm{T}}(t+1)Pe_k(t+1), \tag{3.15}$$

其中，$P\in\mathbb{R}^{l\times l}$是一个对称正定矩阵。

定理 3.1（非提升范数优化框架下基于值迭代的迭代学习控制算法）求解式(3.12)的最优解，可以得到迭代学习控制更新律

$$u_{k+1}(t)=u_k(t)+L_e e_k(t+1)-L_x\Delta x_{k+1}(t), \tag{3.16}$$

其中，控制增益 L_e 和 L_x 分别为

$$L_e=\left(\gamma B^{\mathrm{T}}C^{\mathrm{T}}PCB+B^{\mathrm{T}}RB\right)^{-1}\gamma B^{\mathrm{T}}C^{\mathrm{T}}P, \tag{3.17}$$

$$L_x=\left(\gamma B^{\mathrm{T}}C^{\mathrm{T}}PCB+B^{\mathrm{T}}RB\right)^{-1}\left(B^{\mathrm{T}}RA+\gamma B^{\mathrm{T}}C^{\mathrm{T}}PCA\right), \tag{3.18}$$

式(3.17)和式(3.18)的 P 是离散代数Riccati方程(Discrete Algebra Riccati Equation)的解

$$P=Q+\gamma P-\gamma^2 PC\left(\gamma C^{\mathrm{T}}PC+R\right)^{-1}C^{\mathrm{T}}P. \tag{3.19}$$

证明 基于近似动态规划，由式(3.11)可得

$$V\left(e_k(t+1)\right)=\mathcal{R}\left(e_k(t+1),\ \Delta w_{k+1}(t)\right)+\gamma V\left(e_{k+1}(t+1)\right). \tag{3.20}$$

根据式(3.10)，式(3.15)和式(3.20)，可得

$$V\left(e_k(t+1)\right)=\left\|e_k(t+1)\right\|_Q^2+\left\|\Delta w_{k+1}(t)\right\|_R^2+\gamma e_{k+1}^{\mathrm{T}}(t+1)Pe_{k+1}(t+1). \tag{3.21}$$

进一步代入式(3.13)和式(3.14)，可得

$$\begin{aligned}V\left(e_k(t+1)\right)\ &=e_k^{\mathrm{T}}(t+1)Qe_k(t+1)+\Delta w_{k+1}^{\mathrm{T}}(t)R\Delta w_{k+1}(t)\\&\quad+\gamma e_{k+1}^{\mathrm{T}}(t+1)Pe_{k+1}(t+1).\end{aligned} \tag{3.22}$$

根据式(3.12)的优化目标，将状态值函数 $V\left(e_k(t+1)\right)$ 对 $\Delta u_{k+1}(t)$ 求微分，并令 $\partial V\left(e_k(t+1)\right)\big/\partial\Delta u_{k+1}(t)=0$，代入式(3.9)可得

$$\frac{\partial V\left(e_k(t+1)\right)}{\partial\Delta u_{k+1}(t)}=B^{\mathrm{T}}R\Delta w_{k+1}(t)-\gamma B^{\mathrm{T}}C^{\mathrm{T}}P\left(e_k(t+1)-C\Delta w_{k+1}(t)\right)=0. \tag{3.23}$$

将式(3.8)代入上式，合并同类项，可得

$$\begin{aligned}\Delta u_{k+1}(t)=&\left(\gamma B^{\mathrm{T}}C^{\mathrm{T}}PCB+B^{\mathrm{T}}RB\right)^{-1}\gamma B^{\mathrm{T}}C^{\mathrm{T}}Pe_k(t+1)\\&-\left(\gamma B^{\mathrm{T}}C^{\mathrm{T}}PCB+B^{\mathrm{T}}RB\right)^{-1}\left(B^{\mathrm{T}}RA+\gamma B^{\mathrm{T}}C^{\mathrm{T}}PCA\right)\Delta x_{k+1}(t).\end{aligned} \tag{3.24}$$

令

$$L_e=\left(\gamma B^{\mathrm{T}}C^{\mathrm{T}}PCB+B^{\mathrm{T}}RB\right)^{-1}\gamma B^{\mathrm{T}}C^{\mathrm{T}}P,$$

$$L_x=\left(\gamma B^{\mathrm{T}}C^{\mathrm{T}}PCB+B^{\mathrm{T}}RB\right)^{-1}\left(B^{\mathrm{T}}RA+\gamma B^{\mathrm{T}}C^{\mathrm{T}}PCA\right).$$

可得迭代学习控制更新律式(3.16)。

接下来推导 P 的离散代数 Riccati 方程，根据式(3.12)的优化目标，将状态值函数 $V\left(e_k(t+1)\right)$ 对 $\Delta w_{k+1}(t)$ 求微分，并令 $\partial V\left(e_k(t+1)\right)\big/\partial\Delta w_{k+1}(t)=0$，代入式(3.9)可得

$$\frac{\partial V\left(e_k(t+1)\right)}{\partial\Delta w_{k+1}(t)}=R\Delta w_{k+1}(t)-\gamma C^{\mathrm{T}}P\left(e_k(t+1)-C\Delta w_{k+1}(t)\right)=0, \tag{3.25}$$

即

$$\Delta w_{k+1}(t)=\left(\gamma C^{\mathrm{T}}PC+R\right)^{-1}\gamma C^{\mathrm{T}}Pe_k(t+1), \tag{3.26}$$

其中，定义

$$L_w=\left(\gamma C^{\mathrm{T}}PC+R\right)^{-1}\gamma C^{\mathrm{T}}P. \tag{3.27}$$

将式(3.15)代入式(3.22)，得

$$\begin{aligned} e_k^{\mathrm{T}}(t+1)Pe_k(t+1) &= e_k^{\mathrm{T}}(t+1)Qe_k(t+1)+\Delta w_{k+1}^{\mathrm{T}}(t)R\Delta w_{k+1}(t)\\ &\quad+\gamma e_{k+1}^{\mathrm{T}}(t+1)Pe_{k+1}(t+1). \end{aligned} \tag{3.28}$$

将式(3.9)和式(3.26)代入式定理 3.1可得

$$\begin{aligned} &e_k^{\mathrm{T}}(t+1)Pe_k(t+1)\\ &=e_k^{\mathrm{T}}(t+1)\big[Q+\gamma P\\ &\quad+\gamma^2PC\left(\gamma C^{\mathrm{T}}PC+R\right)^{-1}R\left(\gamma C^{\mathrm{T}}PC+R\right)^{-1}C^{\mathrm{T}}P\\ &\quad+\gamma^2PC\left(\gamma C^{\mathrm{T}}PC+R\right)^{-1}\gamma C^{\mathrm{T}}PC\left(\gamma C^{\mathrm{T}}PC+R\right)^{-1}C^{\mathrm{T}}P\\ &\quad-2\gamma^2PC\left(\gamma C^{\mathrm{T}}PC+R\right)^{-1}C^{\mathrm{T}}P\Big]e_k(t+1), \end{aligned}$$

即

$$P=Q+\gamma P-\gamma^2PC\left(\gamma C^{\mathrm{T}}PC+R\right)^{-1}C^{\mathrm{T}}P.$$

至此，定理 3.1 证毕。 ■

注释 3.3 传统的迭代学习控制仅利用历史批次的跟踪误差信息更新控制输入信号，这被认为是一种开环前馈控制方法，可能导致系统没有良好的暂态性能。为改进这一不足，学者们在迭代学习控制律的设计中引入当前批次的状态信息，构造基于状态误差反馈的 P 型学习律[67]，其中状态误差指批次间状态变化，非提升范数优化框架下的迭代学习控制律的形式满足基于状态误差反馈的 P 型学习律。

由于矩阵 P 的非线性关系，直接求出离散代数 Riccati 方程(3.19)的解是有难度的，因此，接下来推导 P 的递归方程以求解其值。

通过将式(3.9)和式(3.26)代入定理 3.1，得到下式

$$\begin{aligned} e_k^{\mathrm{T}}(t+1)Pe_k(t+1) &= e_k^{\mathrm{T}}(t+1)Qe_k(t+1)+e_k^{\mathrm{T}}(t+1)L_w^{\mathrm{T}}RL_we_k^{\mathrm{T}}(t+1)\\ &\quad+\gamma e_k^{\mathrm{T}}(t+1)\left(I-CL_w\right)^{\mathrm{T}}P\left(I-CL_w\right)e_k(t+1). \end{aligned} \tag{3.29}$$

由此可得 P 的递归方程

$$P=Q+L_w^{\mathrm{T}}RL_w+\gamma\left(I-CL_w\right)^{\mathrm{T}}P\left(I-CL_w\right). \tag{3.30}$$

综上，本章所提的非提升范数优化框架下基于值迭代的迭代学习控制算法流程如算法 3-1 所示。

算法 3-1 非提升范数优化框架下基于值迭代的迭代学习控制算法

Input：给定系统参数矩阵 A， B 和 C；初始控制输入 $u_0(t)$， $t \in \left[0, N-1\right]$；期望参考轨迹 y_d；初始 P_0， $L_{e,0}$， $L_{x,0}$ 和 $L_{w,0}$；最大的迭代学习控制迭代次数 $k_{\max}$

Output：下一批次的输入 $u_{k+1}(t)$

Initialization：设置 $k=0$， $t=0$

repeat

  repeat

    将控制输入 $u_k(t)$ 作用于系统，并获取跟踪误差 $e_k(t+1)$ 与状态变化 $\Delta x_{k+1}(t)$

    策略评估，根据式(3.30)得到更新后的 P

$$P_{k+1} = Q + {L_{w,k}}^{\mathrm{T}} R L_{w,k} + \left(I - CL_{w,k}\right)^{\mathrm{T}} P_k \left(I - CL_{w,k}\right)$$

    值迭代，根据式(3.17)，式(3.18)和式(3.27)得更新后的 $L_{e,k+1}$， $L_{x,k+1}$ 和 $L_{w,k+1}$

$$L_{e,k+1} = \left(\gamma B^{\mathrm{T}} C^{\mathrm{T}} P_{k+1} CB + B^{\mathrm{T}} RB\right)^{-1} \gamma B^{\mathrm{T}} C^{\mathrm{T}} P_{k+1}$$

$$L_{x,k+1} = \left(\gamma B^{\mathrm{T}} C^{\mathrm{T}} P_{k+1} CB + B^{\mathrm{T}} RB\right)^{-1} \left(B^{\mathrm{T}} RA + \gamma B^{\mathrm{T}} C^{\mathrm{T}} P_{k+1} CA\right)$$

$$L_{w,k+1} = \left(\gamma C^{\mathrm{T}} P_{k+1} C + R\right)^{-1} \gamma C^{\mathrm{T}} P_{k+1}$$

    根据迭代学习控制更新律(3.16)更新下一批次输入 $u_{k+1}(t)$

$$u_{k+1}(t) = u_k(t) + L_{e,k+1} e_k(t+1) - L_{x,k+1} \Delta x_{k+1}(t)$$

    设置 $t = t+1$

  until $t = N-1$

  设置 $k = k+1$

until $k = k_{\max}$

3.3.2 收敛性分析

基于值迭代的迭代学习控制算法可以映射到式(3.30)的离散代数 Riccati 方程，其收敛性可以参考文献[68, 69]。接下来，将对称正定矩阵 P 看作收敛后的常值，对本章所提非提升范数优化框架下基于值迭代的迭代学习控制算法进行收敛性分析。为了后续定理的证明，首先介绍所需的引理。

引理 3.1[70] 对于系统(3.1)以及迭代学习控制更新律 $u_{k+1} = \mathcal{K}_u u_k + \mathcal{K}_r r$，当且仅当条件

$$\rho\left(\mathcal{K}_u\right) < 1 \tag{3.31}$$

满足，则系统沿迭代轴渐近稳定，其中 $\rho\left(\bullet\right)$ 为该矩阵的谱半径。

引理 3.2[71] 对于系统(3.1)以及迭代学习控制更新律 $u_{k+1} = \mathcal{K}_u u_k + \mathcal{K}_r r$，当且仅当条件

$$\left\|\mathcal{K}_u\right\| < 1 \tag{3.32}$$

满足，则系统的跟踪误差沿迭代轴单调收敛。

定理 3.2 对于系统(3.1)，使用非提升范数优化框架下基于值迭代的迭代学习控制更

新律(3.16)，若条件

$$\lambda\left[\left(\gamma B^{\mathrm{T}}C^{\mathrm{T}}PCB+B^{\mathrm{T}}RB\right)^{-1}B^{\mathrm{T}}RB\right]<1 \tag{3.33}$$

满足，则系统渐近稳定。

证明 将式(3.1)代入式(3.3)，得

$$e_k(t+1)=y_d(t+1)-CAx_k(t)-CBu_k(t). \tag{3.34}$$

将式(3.34)代入学习律(3.16)得

$$u_{k+1}(t)=L_u u_k(t)+L_e y_d(t+1)-L_e CAx_{k+1}(t)-L_x\Delta x_{k+1}(t), \tag{3.35}$$

其中，$L_u=I-L_eCB=\left(\gamma B^{\mathrm{T}}C^{\mathrm{T}}PCB+B^{\mathrm{T}}RB\right)^{-1}B^{\mathrm{T}}RB$。

由系统(3.1)得

$$x_{k+1}(t)=A^t x_{k+1}(0)+\sum_{l=0}^{t-1}A^{t-1-l}Bu_{k+1}(l), \tag{3.36}$$

$$\Delta x_{k+1}(t)=A^t\Delta x_{k+1}(0)+\sum_{l=0}^{t-1}A^{t-1-l}B\left(u_{k+1}(l)-u_k(l)\right). \tag{3.37}$$

将式(3.36)和(3.37)代入式(3.35)，得

$$u_{k+1}(t)+\sum_{l=0}^{t-1}\eta_{t,l}u_{k+1}(l)=L_u u_k(t)+\sum_{l=0}^{t-1}\varsigma_{t,l}u_k(l)+\kappa(t), \tag{3.38}$$

其中，

$$\eta_{t,l}=\left(L_x+L_eCA\right)A^{t-1-l}B,$$

$$\varsigma_{t,l}=L_xA^{t-1-l}B,$$

$$\kappa(t)=L_e y_d(t+1)-L_eCA^{t+1}x_{k+1}(0)-L_xA^t\Delta x_{k+1}(0).$$

为表述的简明，使用提升系统表示式(3.35)，定义如下超向量

$$u_k=\left[u_k^{\mathrm{T}}(0),u_k^{\mathrm{T}}(1),\ldots,u_k^{\mathrm{T}}(N-1)\right]^{\mathrm{T}},$$

$$y_d=\left[y_d^{\mathrm{T}}(1),y_d^{\mathrm{T}}(2),\ldots,y_d^{\mathrm{T}}(N)\right]^{\mathrm{T}},$$

$$\kappa=\left[\kappa^{\mathrm{T}}(0),\kappa^{\mathrm{T}}(1),\ldots,\kappa^{\mathrm{T}}(N-1)\right]^{\mathrm{T}},$$

$$H=\begin{bmatrix} I & & & \\ \eta_{1,0} & I & & \\ \vdots & \ddots & \ddots & \\ \eta_{N-1,0} & \cdots & \eta_{N-1,N-2} & I \end{bmatrix},$$

$$\overline{L}_u=\begin{bmatrix} L_u & & & 0 \\ \varsigma_{1,0} & L_u & & \\ \vdots & \ddots & \ddots & \\ \varsigma_{N-1,0} & \cdots & \varsigma_{N-1,N-2} & L_u \end{bmatrix},$$

由此可得式(3.38)的提升形式

$$u_{k+1} = H^{-1}\overline{L}_u u_k + H^{-1}\kappa, \tag{3.39}$$

根据引理 3.1，系统沿迭代轴渐近稳定的条件为$\rho\left(H^{-1}\overline{L}_u\right)<1$。又因为$H$是一个对角线元素为单位阵的下三角矩阵，因此$H^{-1}\overline{L}_u$的特征值和$\overline{L}_u$的特征值相同。因此，系统沿迭代轴渐近稳定的条件可以表示为$\lambda\left(L_u\right)<1$，即

$$\lambda\left[\left(\gamma B^{\mathrm{T}}C^{\mathrm{T}}PCB + B^{\mathrm{T}}RB\right)^{-1}B^{\mathrm{T}}RB\right]<1.$$

由此得到式(3.33)，证毕。 ■

定理 3.3 对于系统(3.1)，使用非提升范数优化框架下基于值迭代的迭代学习控制更新律式(3.16)，若条件

$$\left\|H^{-1}\overline{L}_u\right\|<1 \tag{3.40}$$

满足，则被控系统跟踪误差单调收敛。其中，

$$\begin{aligned}H^{-1}\overline{L}_u =& \left(\overline{B}^{\mathrm{T}}\overline{R}\overline{A}\Phi + 2\gamma\overline{B}^{\mathrm{T}}\overline{C}^{\mathrm{T}}\overline{P}\overline{C}\overline{A}\Phi + \overline{B}^{\mathrm{T}}\overline{R}\overline{B} + \gamma\overline{B}^{\mathrm{T}}\overline{C}^{\mathrm{T}}\overline{P}\overline{C}\overline{B}\right)^{-1}\\ &\times\left(\overline{B}^{\mathrm{T}}\overline{R}\overline{B} - \overline{B}^{\mathrm{T}}\overline{R}\overline{A}\Phi - \gamma\overline{B}^{\mathrm{T}}\overline{C}^{\mathrm{T}}\overline{P}\overline{C}\overline{A}\Phi\right),\end{aligned}$$

其中，分别定义如下提升矩阵

$$\overline{A}=\begin{bmatrix}A & & \\ & \ddots & \\ & & A\end{bmatrix},\ \overline{B}=\begin{bmatrix}B & & \\ & \ddots & \\ & & B\end{bmatrix},\ \overline{C}=\begin{bmatrix}C & & \\ & \ddots & \\ & & C\end{bmatrix},$$

$$\overline{P}=\begin{bmatrix}P & & \\ & \ddots & \\ & & P\end{bmatrix},\ \overline{R}=\begin{bmatrix}R & & \\ & \ddots & \\ & & R\end{bmatrix},\ \Phi=\begin{bmatrix}0 & & & \\ AB & \ddots & & \\ \vdots & \ddots & \ddots & \\ A_{N-1}B & \cdots & AB & 0\end{bmatrix}.$$

证明 根据定理 3.3 中的定义，式(3.39)中的$\mathcal{H}$和$\overline{\mathcal{L}}_u$可以改写为

$$H = \left(\gamma\overline{B}^{\mathrm{T}}\overline{C}^{\mathrm{T}}\overline{P}\overline{C}\overline{B} + \overline{B}^{\mathrm{T}}\overline{R}\overline{B}\right)^{-1}\left(\overline{B}^{\mathrm{T}}\overline{R}\overline{A} + 2\gamma\overline{B}^{\mathrm{T}}\overline{C}^{\mathrm{T}}\overline{P}\overline{C}\overline{A}\right)\Phi + I, \tag{3.41}$$

$$\begin{aligned}\overline{L}_u =& -\left(\gamma\overline{B}^{\mathrm{T}}\overline{C}^{\mathrm{T}}\overline{P}\overline{C}\overline{B} + \overline{B}^{\mathrm{T}}\overline{R}\overline{B}\right)^{-1}\left(\overline{B}^{\mathrm{T}}\overline{R}\overline{A} + \gamma\overline{B}^{\mathrm{T}}\overline{C}^{\mathrm{T}}\overline{P}\overline{C}\overline{A}\right)\Phi\\ &+\left(\gamma\overline{B}^{\mathrm{T}}\overline{C}^{\mathrm{T}}\overline{P}\overline{C}\overline{B} + \overline{B}^{\mathrm{T}}\overline{R}\overline{B}\right)^{-1}\overline{B}^{\mathrm{T}}\overline{R}\overline{B}.\end{aligned} \tag{3.42}$$

由式(3.41)和式(3.42)可得，

$$\begin{aligned}H^{-1}\overline{L}_u =& \left(\overline{B}^{\mathrm{T}}\overline{R}\overline{A}\Phi + 2\gamma\overline{B}^{\mathrm{T}}\overline{C}^{\mathrm{T}}\overline{P}\overline{C}\overline{A}\Phi + \overline{B}^{\mathrm{T}}\overline{R}\overline{B} + \gamma\overline{B}^{\mathrm{T}}\overline{C}^{\mathrm{T}}\overline{P}\overline{C}\overline{B}\right)^{-1}\\ &\times\left(\overline{B}^{\mathrm{T}}\overline{R}\overline{B} - \overline{B}^{\mathrm{T}}\overline{R}\overline{A}\Phi - \gamma\overline{B}^{\mathrm{T}}\overline{C}^{\mathrm{T}}\overline{P}\overline{C}\overline{A}\Phi\right).\end{aligned} \tag{3.43}$$

因此，由引理 3.2 可得，当条件$\left\|H^{-1}\overline{L}_u\right\|<1$，即条件

$$\left\| \begin{matrix} \left(\overline{B}^{\mathrm{T}}\overline{R}\overline{A}\Phi + 2\gamma \overline{B}^{\mathrm{T}}\overline{C}^{\mathrm{T}}\overline{P}\overline{C}\overline{A}\Phi + \overline{B}^{\mathrm{T}}\overline{R}\overline{B} + \gamma \overline{B}^{\mathrm{T}}\overline{C}^{\mathrm{T}}\overline{P}\overline{C}\overline{B} \right)^{-1} \\ \times \left(\overline{B}^{\mathrm{T}}\overline{R}\overline{B} - \overline{B}^{\mathrm{T}}\overline{R}\overline{A}\Phi - \gamma \overline{B}^{\mathrm{T}}\overline{C}^{\mathrm{T}}\overline{P}\overline{C}\overline{A}\Phi \right) \end{matrix} \right\| < 1 \tag{3.44}$$

满足时，系统的跟踪误差沿迭代轴单调收敛，证毕。 ■

注释 3.4 如果应用算法时已知系统的模型信息 A，B 和 C，则可以采用基于值迭代的迭代学习控制算法，但是在实际应用中，不能保证精确的系统模型已知，因此第四章考虑在本章所提算法的基础上，进一步提出非提升范数优化框架下基于 Q 学习的迭代学习控制算法，从而将算法的应用情形拓展到无模型。

3.3.3 计算复杂度对比分析

为更清晰论述本章所提方法在计算复杂度方面的优势，本小节首先简明介绍目前使用强化学习直接设计迭代学习控制器的已有工作，提升范数优化框架下基于值迭代的迭代学习控制算法[46]。

将迭代学习控制过程转化为马尔可夫决策过程，定义如下五元组 $\left(\mathcal{S},\mathcal{A},f,\mathcal{R},\gamma \right)$：$\mathcal{S}$ 为状态空间，状态定义为跟踪误差向量 $e_k \in \mathbb{R}^{lN}$；$\mathcal{A}$ 为动作空间，动作定义为输入变化 $\Delta u_{k+1} \in \mathbb{R}^{mN}$；$f$ 为状态转移函数，定义为 $e_{k+1} = e_k - G\Delta u_{k+1}$；$\mathcal{R}$ 为收益函数，定义为 $\mathcal{R}\left(e_k,\ \Delta u_{k+1} \right) = \left\| e_k \right\|_{Q_{\mathrm{lift}}}^2 + \left\| \Delta u_{k+1} \right\|_{R_{\mathrm{lift}}}^2$，其中，$Q_{\mathrm{lift}} \in \mathbb{R}^{lN\times lN}$ 和 $R_{\mathrm{lift}} \in \mathbb{R}^{mN\times mN}$ 为对称正定权重矩阵；

以上马尔可夫决策过程可看作 LQR 问题，因此值函数可以表示为 $V\left(e_k \right) = e_k^{\mathrm{T}}P_{\mathrm{lift}}e_k$，其中，$P_{\mathrm{lift}} \in \mathbb{R}^{lN\times lN}$ 是一个对称正定矩阵。根据优化目标，求解 $\partial V\left(e_k \right)\big/\partial \Delta u_{k+1} = 0$，可得提升范数优化框架下基于值迭代的迭代学习控制更新律写作

$$u_{k+1} = u_k + L_{\mathrm{lift}}e_k, \tag{3.45}$$

其中，控制增益 L_{lift} 为

$$L_{\mathrm{lift}} = \left(R_{\mathrm{lift}} + \gamma G^{\mathrm{T}}P_{\mathrm{lift}}G \right)^{-1} G^{\mathrm{T}}P_{\mathrm{lift}}, \tag{3.46}$$

且 P_{lift} 的递归方程为

$$P_{\mathrm{lift}} = Q_{\mathrm{lift}} + L_{\mathrm{lift}}^{\mathrm{T}}R_{\mathrm{lift}}L_{\mathrm{lift}} + \gamma \left(I - GL_{\mathrm{lift}} \right)^{\mathrm{T}} P_{\mathrm{lift}} \left(I - GL_{\mathrm{lift}} \right). \tag{3.47}$$

计算量，即 flop 数，是评价算法性能的一项指标，通常由算法中乘法运算与加法运算的次数表示[72]，这里引入计算量大小来评价算法的计算复杂度。接下来，将分别从迭代学习控制更新律的计算复杂度与 P 的递归方程的计算复杂度两方面，说明所提的算法 3-1 非提升范数优化框架下基于值迭代的迭代学习控制算法相对于提升范数优化框架下基于值迭代的迭代学习控制算法的计算效率的提升。

（1）迭代学习控制更新律的计算复杂度对比分析

本章所提算法 3-1 中非提升范数优化框架下基于值迭代的迭代学习控制更新律(3.16)共包含 1 个数与矩阵的乘法、10 个矩阵与矩阵的乘法、2 个矩阵与向量的乘法、2 个矩

阵与矩阵的加法、2 个向量与向量的加法（减法）和 1 个矩阵的逆运算。因此，可得如表 3-1 所示的每个迭代批次的算法 3-1 的更新律的计算量。

表 3-1 非提升范数优化框架下基于值迭代的迭代学习控制更新律的计算量

算法	非提升范数优化框架下 基于值迭代的迭代学习控制
乘法次数	$\left(m^3+3m^2n+m^2l+2mnl+3mn^2+ml^2+ml+mn\right)N$
加法次数	$\left(m^3+3m^2n+m^2l-2m^2+2mnl+3mn^2+ml^2-3mn-2ml\right)N$
总计算量	$\left(2m^3+6m^2n+2m^2l-2m^2+4mnl+6mn^2+2ml^2-2mn-ml\right)N$

文献[46]中所提的提升范数优化框架下基于值迭代的迭代学习控制更新律(3.45)共包含 1 个数与矩阵的乘法、3 个矩阵与矩阵的乘法、1 个矩阵与向量的乘法、1 个矩阵与矩阵的加法、1 个向量与向量的加法和 1 个矩阵的逆运算。因此，可得如表 3-2 所示的每个迭代批次的提升范数优化框架下基于值迭代的迭代学习控制更新律的计算量。

表 3-2 提升范数优化框架下基于值迭代的迭代学习控制更新律的计算量

算法	提升范数优化框架下 基于值迭代的迭代学习控制
乘法次数	$\left(m^3+2m^2l+ml^2\right)N^3+2mlN^2$
加法次数	$\left(m^3+2m^2l+ml^2\right)N^3+\left(-m^2-ml\right)N^2$
总计算量	$\left(2m^3+4m^2l+2ml^2\right)N^3+\left(ml-m^2\right)N^2$

综上，由表 3-1 和表 3-2 可以看出，本章所提算法 3-1 的迭代学习控制更新律的计算复杂度与采样点个数 N 呈线性关系变化，即 $O(N)$；文献[46]中使用提升技术基于值迭代的算法的计算复杂度与采样点个数 N 呈 3 次方阶趋势增加，即 $O(N^3)$。因此，本章所提算法 3-1 对于采样点个数 N 较多的情形更有优势，计算效率更高。

（2）值迭代更新式的计算复杂度对比分析

本章所提算法 3-1 中 P 的递归方程式(3.30)共包含 1 个数与矩阵的乘法、5 个矩阵与矩阵的乘法、3 个矩阵与矩阵的加法（减法）。因此，可得如表 3-3 所示的本章所提算法 3-1 中每个迭代批次的 P 的递归方程的计算量。

表 3-3 非提升范数优化框架下基于值迭代的迭代学习控制中 P 的递归方程的计算量

算法	非提升范数优化框架下 基于值迭代的迭代学习控制
乘法次数	$\left(n^2l+2nl^2+2l^3+l^2\right)N$
加法次数	$\left(n^2l+2nl^2+2l^3-nl-l^2\right)N$
总计算量	$\left(2n^2l+4nl^2+4l^3-nl\right)N$

文献[46]中所提的提升范数优化框架下基于值迭代的迭代学习控制更新律(3.47)共包含 1 个数与矩阵的乘法、5 个矩阵与矩阵的乘法、3 个矩阵与矩阵的加法（减法）。因此，可得如表 3-4 所示的提升范数优化框架下基于值迭代的迭代学习控制的每个迭代批次的 P 的递归方程的计算量。

表 3-4 提升范数优化框架下基于值迭代的迭代学习控制中 P_{lift} 的递归方程的计算量

算法	提升范数优化框架下 基于值迭代的迭代学习控制
乘法次数	$\left(m^2l+2ml^2+2l^3\right)N^3+l^2N^2$
加法次数	$\left(m^2l+2ml^2+2l^3\right)N^3+\left(-ml-l^2\right)N^2$
总计算量	$\left(2m^2l+4ml^2+4l^3\right)N^3-mlN^2$

综上，由表 3-3 和表 3-4 可以看出，本章所提算法 3-1 中每个迭代批次 P 的递归方程的计算复杂度与采样点个数 N 呈线性关系变化，即 $O(N)$；文献[46]中提升范数优化框架下基于值迭代的迭代学习控制中每个迭代批次的 P_{lift} 的递归方程的计算复杂度与采样点个数 N 呈 3 次方阶趋势增加，即 $O(N^3)$。因此，本章所提算法 3-1 对于采样点个数 N 较多的情形更有优势，计算效率更高。

3.4 仿真实例

为验证本章所提算法 3-1 的有效性，本节考虑如图 3-1 所示的直流电动机系统[73]作为仿真对象。

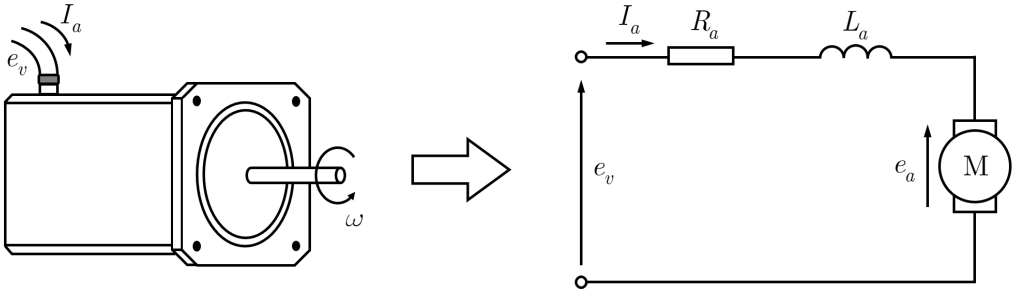

图 3-1 直流电动机示意图

图 3-1 中，左侧为直流电动机物理结构图，右侧为直流电动机等效电路图。直流电动机的动态模型如下所示

$$\begin{cases} R_a i_a + L_a \dfrac{di_a}{dt} + C_e\omega = e_v, \\ J_r \dfrac{d\omega}{dt} + C_f\omega = C_M i_a, \end{cases}$$

其中，e_v 和 i_a 分别表示输入电压和电枢电流，ω 表示电机转速。直流电机系统按表 3-5 进行参数设置：

表 3-5 直流电机系统参数的定义与值

变量	定义	值
R_a	电枢电阻	2.1Ω
L_a	电枢电感	$0.8\,\mathrm{H}$
C_e	反电动势系数	$0.18\,\mathrm{V/(rad/s)}$
C_M	转矩系数	$0.646\,\mathrm{Nm/A}$
C_f	电机机械阻尼	$1.07\times10^{-3}\,\mathrm{Nm/(rad/s)}$
J_r	转子转动惯量	$1\mathrm{kgm}^2$

定义输出变量为电机转速 $u=\omega$，输入变量为电枢电流 $y=i_a$，状态变量为 $x=[i_a \quad \omega]^{\mathrm{T}}$，设置系统仿真时间 $T=20\mathrm{s}$，采样时间 $T_s=0.1\mathrm{s}$，即采样点总个数 $N=200$，使用零阶保持器的方法将直流电机系统离散化，则系统的状态空间方程的参数矩阵分别为

$$A=\begin{bmatrix}0.7684 & -0.0198\\ 0.0710 & 0.9990\end{bmatrix},\ B=\begin{bmatrix}0.1099\\ 0.0046\end{bmatrix},\ C=\begin{bmatrix}0 & 1\end{bmatrix}.$$

给定直流电机系统的控制任务为：120 个批次内（$k_{\max}=120$），直流电机的转速跟踪上给定参考轨迹：

$$y_d=2\left(\sin\left(2\pi t/20\right)+\sin(2\pi t/30)\right).$$

期望轨迹的选取主要从工业实际应用角度考虑，如机器人使用的含有变速机的直流电动机的工况，需求为此类小速度的正反转。为评价系统的跟踪控制性能，引入均方根误差（Root Mean Square Error, RMSE）作为系统的评价指标：

$$\mathrm{RMSE}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}e_k^2\left(i\right)}.$$

（1）非提升范数优化框架下基于值迭代的迭代学习控制算法

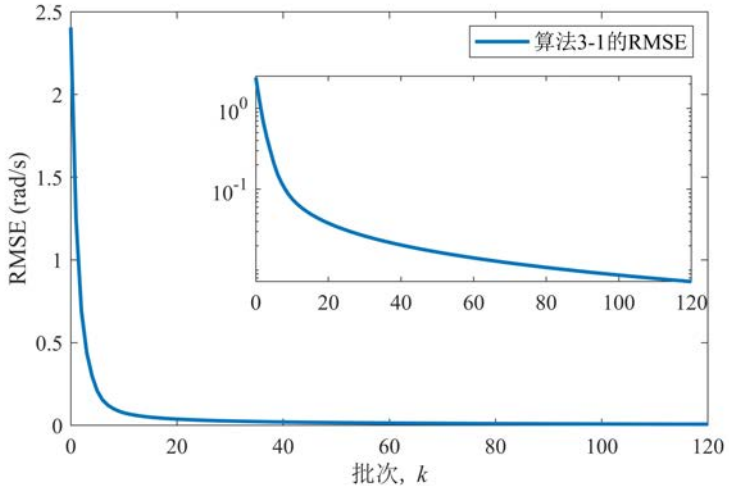

图 3-2 直流电机系统在算法 3-1 下的电机转速的跟踪误差 2-范数收敛图和对数收敛图

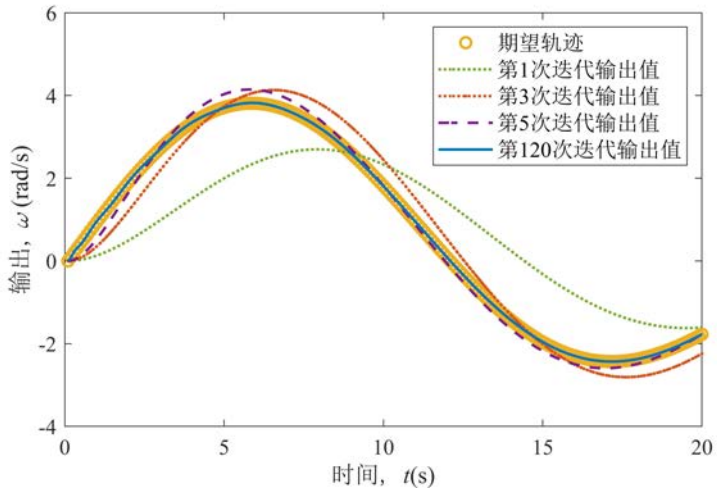

图 3-3 直流电机系统在算法 3-1 下的电机转速期望值轨迹与实际输出值图

设置控制器参数$Q = I$，$R = 0.1I$，$\gamma = 0.85$，则算法 3-1 的仿真结果如图 3-2 和图 3-3 所示。从图 3-2 可以看出随着迭代批次的增加，直流电机转速的跟踪误差可以逐渐收敛。从图 3-3 可以看出，随着迭代批次的增加，电机转速可以逐渐跟踪上给定的参考轨迹，因此算法 3-1 能完成给定的控制任务。

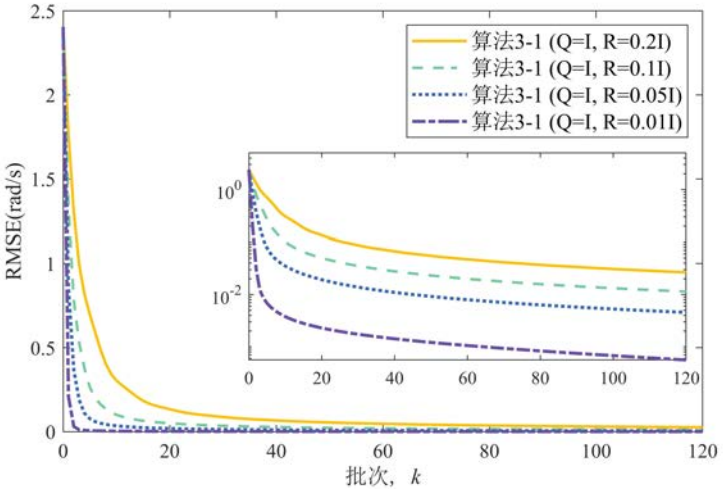

图 3-4 算法 3-1 在不同Q和R下的电机转速的跟踪误差 2-范数收敛图和对数收敛图

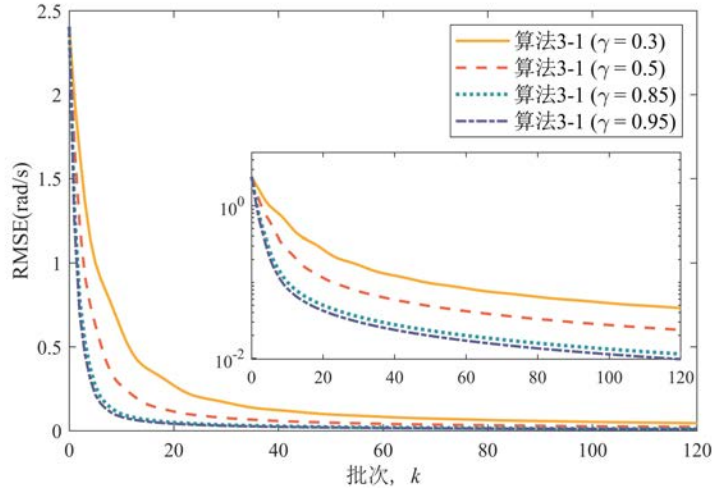

图 3-5 算法 3-1 在不同γ下的电机转速的跟踪误差 2-范数收敛图和对数收敛图

图 3-4 展示了控制器参数$\gamma = 0.85$时，不同的Q和R的选择下的跟踪误差收敛对比图，可以看出，增加Q的值或减小R的值会提高跟踪误差的收敛速度和收敛精度。图 3-5 展示了控制器参数$Q = I$，$R = 0.1I$时，不同的γ选择下的跟踪误差收敛对比图，可以看出，增加γ的值会提高跟踪误差的收敛速度和收敛精度。

（2）算法对比

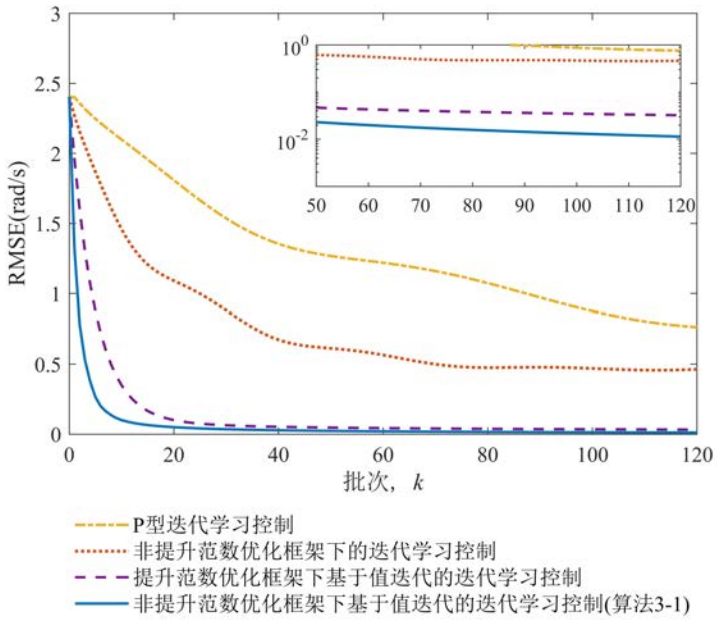

图 3-6 直流电机系统在 Case3-1 下的电机转速的跟踪误差 2-范数收敛对比图

Case3-1：选取 P 型迭代学习控制算法[22]、非提升范数优化框架下的迭代学习控制算法[63]和提升范数优化框架下基于值迭代的迭代学习控制算法[46]作为对比算法与算法 3-1 进行对比实验。P 型迭代学习控制算法的控制增益为$k_P = 0.03$，非提升范数优化框架下的迭代学习控制算法的控制律如式(2.13)所示，控制器参数$Q_2 = I$，$R_2 = 0.5I$，提升范数优化框架下基于值迭代的迭代学习控制算法的控制律如式(3.45)，控制器参数为$Q_{\text{lift}} = I$，$R_{\text{lift}} = 30I$，$\gamma = 0.85$，本章所提算法 3-1 控制器参数为$Q = I$，$R = 0.1I$，$\gamma = 0.85$。从图 3-6 可以看出，本章所提算法 3-1 的跟踪误差的收敛速度与收敛精度均优于 P 型迭代学习控制算法和非提升范数优化框架下的迭代学习控制算法，优于 P 型迭代学习控制算法的原因是因为引入了基于模型的优化理论，优于非提升范数优化框架下的迭代学习控制算法的原因是因为本章所提算法 3-1 引入了强化学习思想，在利用模型信息的基础上，进一步通过未来收益指导当前动作，即下一批次的控制输入信号。同时本章所提算法 3-1 的跟踪误差的收敛速度与收敛精度均优于提升范数优化框架下基于值迭代的迭代学习控制算法，本章所提算法引入了状态信息，因此有更多用于控制的系统信息。

Case3-2：选取提升范数优化框架下基于值迭代的迭代学习控制算法[46]与本章所提算法 3-1 进行计算时间对比实验，提升范数优化框架下基于值迭代的迭代学习控制算法的控制器参数为$Q_{\text{lift}} = I$，$R_{\text{lift}} = 30I$，$\gamma = 0.85$，本章所提算法 3-1 控制器参数为

$Q=I$，$R=0.1I$，$\gamma=0.85$，两种算法均进行 120 次试验（$k_{\max}=120$）。程序计算在一台 3.3GHz AMD 锐龙 9 5900HX 处理器和 32GB RAM 的笔记本电脑上使用 MATLAB R2020b 进行。由图 3-7 可以看出，提升范数优化框架下基于值迭代的迭代学习控制算法，一方面，其计算时间与采样点个数 N 呈 3 次方阶增长关系，即 $O(N^3)$，另一方面，由于需要求解的系统模型矩阵过大，受平台的硬件内存限制，采样点个数 N 最大为 20000。这意味着，如果采样率为 1kHz，则批次长度需要限制在小于等于 20s。对于批次长度更大的情形，本计算平台无法分配更多的内存用于矩阵计算，在实际应用场景中也有类似的局限性，这些结论与第 3.3.3 节的计算复杂度对比分析以及文献[74]所得结论一致。相比之下，本章所提算法 3-1 的计算时间与采样点个数 N 呈线性增长关系，即 $O(N)$，且不会占用过多硬件内存。

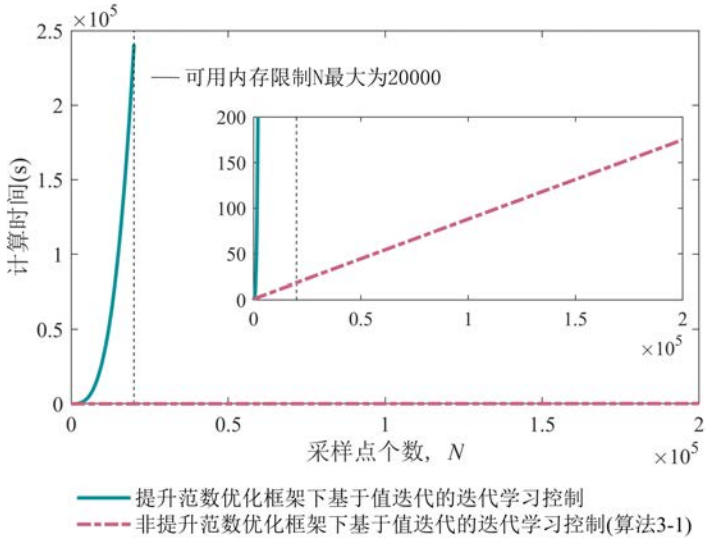

图 3-7 Case3-2 下不同算法的计算时间随采样点个数增长对比图

3.5 小结

本章研究了一类线性离散系统的跟踪问题，在非提升范数优化框架下，提出了一种基于值迭代的迭代学习控制方法。将迭代学习控制过程描述为马尔可夫决策过程，引入强化学习的未来收益指导当下动作的思想。通过最小化状态值函数得到迭代学习控制更新律，提出了系统在该学习律下系统渐近稳定的条件和跟踪误差单调收敛的条件并给出了相应证明，分析了所提方法在计算复杂度方面的优势。最后，通过直流电动机的仿真，验证了所提方法的有效性和优势。

本章所提的非提升范数优化框架下基于值迭代的迭代学习控制算法是基于模型的方法，然而实际应用中，不能保证精确的系统模型已知，因此第四章考虑在本章所提算法 3-1 的基础上，进一步提出非提升范数优化框架下基于 Q 学习的迭代学习控制算法，从而将算法的应用情形拓展到无模型。

第四章 非提升范数优化框架下基于Q学习的无模型迭代学习控制

4.1 引言

使用优化理论解决工业过程中的具有重复运行特性系统的控制问题时，通常需要系统参数信息来求解最优控制增益，然而在实际应用中，不一定能获得精确的系统模型信息，因此本章在第三章的所提算法的基础上，引入 Q 学习方法，将算法从基于模型拓展到无需模型信息。同时，目前使用强化学习方法直接设计无模型迭代学习控制器的结合的工作，一般在提升范数优化框架下完成，此类算法会存在高计算复杂度导致的采样点个数限制问题以及用于获取求解系统模型的实验批次数量过多的问题。因此，本章在提升范数优化框架下基于 Q 学习的迭代学习控制算法[46]的基础上，进一步引入了文献[63]中的非提升范数优化框架，一方面，可以降低采样点数量多的情形下算法的计算复杂度，另一方面，可以减少用于获取求解系统模型的实验批次数量。

本章针对一类具有重复运行特性的模型信息未知的线性时不变离散系统进行研究，主要内容和结构如下：第 4.2 节介绍了非提升范数优化框架下迭代学习控制算法设计问题；第 4.3 节在非提升范数优化框架下，首先将迭代学习控制过程描述为马尔可夫决策过程，引入 Q 学习方法，使用系统信息表示 Q 函数，并通过求解 Q 函数的优化问题得到迭代学习控制更新律，推导出利用可测数据求解更新律所需的模型信息的最小二乘方法，然后证明了本章所提算法的收敛性，最后介绍了提升范数优化框架下基于 Q 学习的迭代学习控制算法，给出了本章所提方法与该算法的计算复杂度对比分析；第 4.4 节通过直流电动机的仿真，验证了所提方法的有效性和优势；第 4.5 节对本章内容进行小结。

4.2 问题描述

考虑如下一类具有未知模型参数的线性时不变离散系统

$$\begin{cases} x_k(t+1) = A(\theta)x_k(t) + B(\theta)u_k(t), \\ \quad y_k(t) = C(\theta)x_k(t), \end{cases} \tag{4.1}$$

其中，下标 k 表示迭代批次；$t \in \left[0,\ N\right]$ 表示一个重复运行周期 T 内的采样时刻，N 为一个批次的采样点个数；$x_k(t) \in \mathbb{R}^n$，$u_k(t) \in \mathbb{R}^m$，$y_k(t) \in \mathbb{R}^l$ 分别代表系统在第 k 批次的状态变量、控制输入以及输出信号；$A(\theta)$，$B(\theta)$，$C(\theta)$ 分别表示具有相应维数的系统参数矩阵；θ 表示未知的系统参数常量；为保证系统输出可控，需要满足 $C(\theta)B(\theta)$ 满秩；$x_k(0)$ 表示系统在第 k 批次运行时的初始状态值，假设系统在不同批次的初始状态值相同，即 $\forall k$，$x_k(0) = x_0$。

假设 $y_d(t)$ 为期望参考轨迹，那么系统存在一个的期望控制输入 $u_d(t) \in \mathbb{R}^m$ 并且期望控制输入 $u_d(t)$ 可以驱动系统在有限时间 $t \in \left[0,\ N\right]$ 内跟踪上期望参考轨迹，即

$$\begin{cases} x_d(t+1) = A(\theta)x_d(t) + B(\theta)u_d(t), \\ \quad y_d(t) = C(\theta)x_d(t), \end{cases} \tag{4.2}$$

其中，$x_d(t)$ 代表了系统的期望状态。迭代学习控制律的设计目标是调节下一个迭代批次的输入 $u_{k+1}(t)$，并使其逐渐收敛于 $u_d(t)$。

定义 4.1（非提升范数优化框架下无模型迭代学习控制算法设计问题） 设计一个多目标性能指标函数并将其优化，得到如下类似式(2.13)形式的无需模型信息的迭代学习控制更新律

$$u_{k+1}(t) = f\big(u_k(t),\ e_k(t+1),\ \Delta x_{k+1}(t)\big), \tag{4.3}$$

其中，下一批次的输入信号 $u_{k+1}(t)$ 由当前批次的输入信号 $u_k(t)$、当前批次的跟踪误差 $e_k(t+1)$ 和批次间状态变化 $\Delta x_{k+1}(t)$ 组成。系统在式(3.4)所示的迭代学习控制更新律作用下，可以实现迭代学习控制的跟踪目标，即

$$\lim_{k\to\infty} u_k(t) = u_d(t),\ \ \lim_{k\to\infty} e_k(t) = 0. \tag{4.4}$$

4.3 基于Q学习的无模型迭代学习控制

4.3.1 Q 函数的表示

根据跟踪误差的定义式(3.3)，可得

$$e_{k+1}(t+1) = e_k(t+1) - C(\theta)A(\theta)\Delta x_{k+1}(t) - C(\theta)B(\theta)\Delta u_{k+1}(t). \tag{4.5}$$

将迭代学习控制过程中的误差动态方程(4.5)描述为马尔可夫决策过程，从而使得迭代学习控制问题可以用强化学习方法求解。定义如下五元组 $\big(\mathcal{S},\mathcal{A},f,\mathcal{R},\gamma\big)$：

- $\mathcal{S}$ 为状态空间，状态 $s\in\mathcal{S}$ 定义为跟踪误差 $e_k(t+1)\in\mathbb{R}^l$;
- $\mathcal{A}$ 为动作空间，动作 $a\in\mathcal{A}$ 定义为动作向量 $\Delta w_{k+1}(t)\in\mathbb{R}^n$，即

$$w_k(t) \triangleq A(\theta)x_k(t) + B(\theta)u_k(t), \tag{4.6}$$

$$\Delta w_{k+1}(t) \triangleq A(\theta)\Delta x_{k+1}(t) + B(\theta)\Delta u_{k+1}(t); \tag{4.7}$$

- f 为状态转移函数，定义为 $s' = f(s,a)$，即

$$e_{k+1}(t+1) = f(e_k(t+1),\ \Delta w_{k+1}(t)) = e_k(t+1) - C(\theta)\Delta w_{k+1}(t); \tag{4.8}$$

- $\mathcal{R}$ 为收益函数，定义为

$$\mathcal{R}\big(e_k(t+1),\ \Delta w_{k+1}(t)\big) = \big\|e_k(t+1)\big\|_Q^2 + \big\|\Delta w_{k+1}(t)\big\|_R^2; \tag{4.9}$$

- γ 为折扣因子，决定了未来收益的现在价值，并且 $\gamma\in\big(0,1\big]$。

当前批次的状态值函数 $V\big(e_k(t+1)\big)$ 及迭代学习控制性能指标 $J_{k+1}(t)$ 定义为

$$V\big(e_k(t+1)\big) = J_{k+1}(t) = \sum_{j=0}^{\infty}\gamma^j\mathcal{R}\big(e_{k+j}(t+1),\ \Delta w_{k+1+j}(t)\big). \tag{4.10}$$

迭代学习控制算法设计目标是求解使性能指标 $J_{k+1}(t)$ 最小化的最优输入 $\Delta u_{k+1}^*(t)$，强化学习的设计目标是通过最小化状态值函数 $V\big(e_k(t+1)\big)$，为下一动作找到最佳策略即最佳动作 $\Delta w_{k+1}^*(t)$。在第 $k+1$ 批次的第 t 时刻，$\Delta x_{k+1}(t)$ 为已知常量，因此，

$$\Delta w_{k+1}^{*}(t)=A(\theta)\Delta x_{k}(t)+B(\theta)\Delta u_{k+1}^{*}(t),$$

即迭代学习控制的算法设计目标与强化学习的设计目标一致。

因此，由马尔可夫决策过程描述的迭代学习控制算法设计目标是通过最小化式(4.10)，从而为下一动作 $\Delta w_{k+1}(t)$ 找到最佳策略，同时得到最优输入，即

$$\min_{\Delta u_{k+1}(t)}\left\{J_{k+1}(t)\right\}\Leftrightarrow\min_{\Delta w_{k+1}(t)}\left\{V\left(e_{k}(t+1)\right)\right\}. \tag{4.11}$$

Q 学习是一种强化学习方法，有基于模型或无模型的应用方式，并在一定的假设下保证自身收敛性[75]。Q 函数是状态动作值函数，其定义如下

$$\begin{aligned}&Q\left(e_{k}(t+1),\ \Delta w_{k+1}(t)\right)\\&\quad=\mathcal{R}\left(e_{k}(t+1),\Delta w_{k+1}(t)\right)+\gamma Q\left(e_{k+1}(t+1),\ \Delta w_{k+1}(t)\right).\end{aligned} \tag{4.12}$$

需要说明的是，状态值函数是在策略 $\Delta w_{k+1}(t)$ 下的状态 $e_{k}(t+1)$ 的值函数，因此，在这里与 Q 函数有着相同的数值，即

$$V\left(e_{k}(t+1)\right)=Q\left(e_{k}(t+1),\ \Delta w_{k+1}(t)\right). \tag{4.13}$$

针对最优迭代学习控制问题，可以将 Q 函数写作如下二次形式

$$\begin{aligned}Q\left(e_{k}(t+1),\ \Delta w_{k+1}(t)\right)&=e_{k}^{\mathrm{T}}(t+1)Qe_{k}(t+1)+\Delta w_{k+1}^{\mathrm{T}}(t)R\Delta w_{k+1}(t)\\&\quad+\gamma e_{k+1}^{\mathrm{T}}(t+1)Pe_{k+1}(t+1)\\&=\begin{bmatrix}e_{k}(t+1)\\\Delta w_{k+1}(t)\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}Q+\gamma P & -\gamma PC(\theta)\\-\gamma C(\theta)^{\mathrm{T}}P & R+\gamma C(\theta)^{\mathrm{T}}PC(\theta)\end{bmatrix}\begin{bmatrix}e_{k}(t+1)\\\Delta w_{k+1}(t)\end{bmatrix}\\&=Z_{1,k+1}^{\mathrm{T}}(t+1)\mathcal{H}Z_{1,k+1}(t+1),\end{aligned} \tag{4.14}$$

其中，

$$Z_{1,k+1}(t+1)=\begin{bmatrix}e_{k}^{\mathrm{T}}(t+1) & \Delta w_{k+1}^{\mathrm{T}}(t)\end{bmatrix}^{\mathrm{T}}=\begin{bmatrix}e_{k}^{\mathrm{T}}(t+1) & \Delta x_{k+1}^{\mathrm{T}}(t+1)\end{bmatrix}^{\mathrm{T}}\in\mathbb{R}^{p_{1}},p_{1}=n+l.$$

系统参数矩阵 $\mathcal{H}=\mathcal{H}^{\mathrm{T}}\in\mathbb{R}^{p_{1}\times p_{1}}$ 被分割定义为

$$\mathcal{H}=\begin{bmatrix}H_{ee} & H_{ew}\\H_{we} & H_{ww}\end{bmatrix},$$

并且

$$\begin{aligned}H_{ee}&=Q+\gamma P,\qquad H_{ew}=-\gamma PC(\theta),\\H_{we}&=-\gamma C(\theta)^{\mathrm{T}}P,\ H_{ww}=R+\gamma C(\theta)^{\mathrm{T}}PC(\theta).\end{aligned}$$

又因为

$$\begin{bmatrix}e_{k}(t+1)\\\Delta w_{k+1}(t)\end{bmatrix}=\begin{bmatrix}I & 0 & 0\\0 & A(\theta) & B(\theta)\end{bmatrix}\begin{bmatrix}e_{k}(t+1)\\\Delta x_{k+1}(t)\\\Delta u_{k+1}(t)\end{bmatrix}, \tag{4.15}$$

故式(4.14)可改写为

$$\begin{aligned} Q\big(e_k(t+1),\ \Delta w_{k+1}(t)\big) &= Z_{2,k+1}^{\mathrm{T}}(t+1)\begin{bmatrix} I & 0 & 0 \\ 0 & A(\theta) & B(\theta) \end{bmatrix}^{\mathrm{T}} \mathcal{H} \\ &\quad \times \begin{bmatrix} I & 0 & 0 \\ 0 & A(\theta) & B(\theta) \end{bmatrix} Z_{2,k+1}(t+1) \\ &= Z_{2,k+1}^{\mathrm{T}}(t+1)\mathcal{F} Z_{2,k+1}(t+1), \end{aligned} \tag{4.16}$$

其中，

$$Z_{2,k+1}(t+1) = \begin{bmatrix} e_k^{\mathrm{T}}(t+1) & \Delta x_{k+1}^{\mathrm{T}}(t) & \Delta u_{k+1}^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{p_2}, p_2 = n+l+m.$$

系统参数矩阵 $\mathcal{F} = \mathcal{F}^{\mathrm{T}} \in \mathbb{R}^{p_2 \times p_2}$ 被分割定义为

$$\mathcal{F} = \begin{bmatrix} F_{ee} & F_{ex} & F_{eu} \\ F_{xe} & F_{xx} & F_{xu} \\ F_{ue} & F_{ux} & F_{uu} \end{bmatrix},$$

并且

$$\begin{aligned} F_{ee} &= Q + \gamma P \in \mathbb{R}^{l \times l}, \\ F_{ex} &= F_{xe}^{\mathrm{T}} = -\gamma P C(\theta) A(\theta) \in \mathbb{R}^{n \times l}, \\ F_{eu} &= F_{ue}^{\mathrm{T}} = -\gamma P C(\theta) B(\theta) \in \mathbb{R}^{l \times m}, \\ F_{xx} &= A(\theta)^{\mathrm{T}} \Big(R + \gamma C(\theta)^{\mathrm{T}} P C(\theta) \Big) A(\theta) \in \mathbb{R}^{n \times n}, \\ F_{xu} &= F_{xu}^{\mathrm{T}} = A(\theta)^{\mathrm{T}} \Big(R + \gamma C(\theta)^{\mathrm{T}} P C(\theta) \Big) B(\theta) \in \mathbb{R}^{n \times m}, \\ F_{uu} &= B(\theta)^{\mathrm{T}} \Big(R + \gamma C(\theta)^{\mathrm{T}} P C(\theta) \Big) B(\theta) \in \mathbb{R}^{m \times m}. \end{aligned}$$

4.3.2 无模型优化控制算法设计

根据优化目标式(4.4)与 Q 函数定义式(4.12)，将 Q 函数 $Q\big(e_k(t+1),\ \Delta w_{k+1}(t)\big)$ 对 $\Delta u_{k+1}(t)$ 求微分，同时令 $\partial Q\big(e_k(t+1),\ \Delta w_{k+1}(t)\big) \big/ \partial \Delta u_{k+1}(t) = 0$，可得

$$\begin{aligned} \frac{1}{2}\frac{\partial Q\big(e_k(t+1),\ \Delta w_{k+1}(t)\big)}{\partial \Delta u_{k+1}(t)} &= F_{ue} e_k(t+1) + F_{ux}\Delta x_{k+1}(t) + F_{uu}\Delta u_{k+1}(t) \\ &= 0, \end{aligned} \tag{4.17}$$

由此，基于 Q 学习的无模型优化迭代学习控制更新律可以表示为

$$\Delta u_{k+1}(t) = -\Big(F_{uu}\Big)^{-1} F_{ue} e_k(t+1) - \Big(F_{uu}\Big)^{-1} F_{ux}\Delta x_{k+1}(t), \tag{4.18}$$

即控制增益可以定义为

$$L_e^q = -\Big(F_{uu}\Big)^{-1} F_{ue},\ L_x^q = -\Big(F_{uu}\Big)^{-1} F_{ux}. \tag{4.19}$$

迭代学习控制更新律的控制增益(4.19)是基于模型信息的，但对于模型信息未知的系统，参数矩阵 $\mathcal{F}$ 可以使用一定批次数量的实验数据通过最小二乘方法(4.24)进行估计，具体的求解过程将本节后续部分叙述。

为解决连续状态和动作空间问题，使用实验数据求解参数矩阵 $\mathcal{F}$，式(4.14)可参数

化为

$$Z_{2,k+1}^{\mathrm{T}}(t+1)\mathcal{F}Z=\overline{Z}_{2,k+1}^{\mathrm{T}}(t+1)\overline{\mathcal{F}}, \tag{4.20}$$

其中，

$$\overline{\mathcal{F}}=\mathrm{vec}(\mathcal{F})=\left[T_1^{\mathrm{T}},\ T_2^{\mathrm{T}},\ T_3^{\mathrm{T}},\ T_4^{\mathrm{T}},\ T_5^{\mathrm{T}},\ T_6^{\mathrm{T}},\ T_7^{\mathrm{T}},\ T_8^{\mathrm{T}},\ T_9^{\mathrm{T}}\right]^{\mathrm{T}},$$

$$T_1=\mathrm{vec}(F_{ee}^{\mathrm{T}}),\ T_2=\mathrm{vec}(F_{ex}^{\mathrm{T}}),\ T_3=\mathrm{vec}(F_{xu}^{\mathrm{T}}),$$

$$T_4=\mathrm{vec}(F_{xe}^{\mathrm{T}}),\ T_5=\mathrm{vec}(F_{xx}^{\mathrm{T}}),\ T_6=\mathrm{vec}(F_{xu}^{\mathrm{T}}),$$

$$T_7=\mathrm{vec}(F_{ue}^{\mathrm{T}}),\ T_8=\mathrm{vec}(F_{ux}^{\mathrm{T}}),\ T_9=\mathrm{vec}(F_{uu}^{\mathrm{T}}).$$

回归向量$\overline{Z}_{2,k+1}$定义为Z_{k+1}的克罗内克积，即

$$\begin{aligned}\overline{Z}_{2,k+1}(t+1)&=Z_{2,k+1}(t+1)\otimes Z_{2,k+1}(t+1)\\&=\left[z_1^2;\ z_1z_2;\ \cdots\ z_1z_{p_2};\ z_2z_1;\ z_2^2;\ \cdots\ z_2z_{p_2};\ \cdots\ z_{p_2}^2\right],\end{aligned}$$

其中，z_i为向量$Z_{2,k+1}(t+1)\in\mathbb{R}^{p_2}$的第$i$个元素。

根据式(4.20)，式(4.14)可以改写为

$$\overline{Z}_{2,k+1}^{\mathrm{T}}(t+1)\overline{\mathcal{F}}=\mathcal{R}\left(e_k(t+1),\Delta w_{k+1}(t)\right)+\gamma\overline{Z}_{2,k+2}^{\mathrm{T}}(t+1)\overline{\mathcal{F}}. \tag{4.21}$$

由于未知参数矩阵$\mathcal{F}$为对称阵，未知参数向量$\overline{\mathcal{F}}$储存着个独立元素，需要$L\geq p_2(p_2+1)\big/2$个数据样本，才足够组成一定秩的数据矩阵，从而使用最小二乘法（Least Square）求解系统参数向量。

定义数据矩阵为

$$\Phi=\begin{bmatrix}\overline{Z}_{2,k+1}^{\mathrm{T}}(t+1)-\gamma\overline{Z}_{2,k+2}^{\mathrm{T}}(t+1))\\ \overline{Z}_{2,k+2}^{\mathrm{T}}(t+1)-\gamma\overline{Z}_{2,k+3}^{\mathrm{T}}(t+1))\\ \vdots\\ \overline{Z}_{2,k+L}^{\mathrm{T}}(t+1)-\gamma\overline{Z}_{2,k+L+1}^{\mathrm{T}}(t+1))\end{bmatrix}, \tag{4.22}$$

$$\Upsilon=\begin{bmatrix}\mathcal{R}\left(e_k(t+1),\Delta w_{k+1}(t)\right)\\ \mathcal{R}\left(e_{k+1}(t+1),\Delta w_{k+2}(t)\right)\\ \vdots\\ \mathcal{R}\left(e_{k+L-1}(t+1),\Delta w_{k+L}(t)\right)\end{bmatrix}, \tag{4.23}$$

则未知参数向量$\overline{\mathcal{F}}$的最小二乘法的解可计算为

$$\overline{\mathcal{F}}=\left[\Phi^{\mathrm{T}}\Phi\right]^{-1}\Phi^{\mathrm{T}}\Upsilon. \tag{4.24}$$

值得注意的是，需要在迭代学习控制更新律式(4.18)中引入探测噪声，一方面以满足最小二乘法的求解的持续激励条件[75,76]，另一方面以增加对系统参数$\mathcal{F}$的探索。本工作中选用常见的交替随机频率的正弦信号之和[77]，其中交替随机频率服从$\left[a,\ b\right]$范围的均匀分布。因此，控制律可以改写为

$$\Delta u_{k+1}(t)=L_e^qe_k(t+1)+L_x^qF_{ux}\Delta x_{k+1}(t)+n_{pb}, \tag{4.25}$$

其中，n_{pb} 为探测噪声。

综上，非提升范数优化框架下基于 Q 学习的无模型迭代学习控制如算法 4-1 所示。

算法 4-1 提升范数优化框架下基于 Q 学习的无模型迭代学习控制算法

Input： 初始控制输入 $u_0(t)$，$t \in \left[0, N-1\right]$；期望参考轨迹 y_d；初始 $L_e^{q[0]}$ 和 $L_x^{q[0]}$；最大的迭代学习控制迭代次数 $k_{\max}$

Output： 下一批次的输入 $u_{k+1}(t)$

Initialization： 设置 $k=0$，$t=0$，$j=0$

repeat

  repeat

    将控制输入 $u_k(t)$ 作用于系统，以采集 $Z_{2,k+1}(t+1)$ 和 $\mathcal{R}\left(e_k(t+1), \Delta w_{k+1}(t)\right)$ 的数据

    根据迭代学习更新律(4.25)更新下一批次输入 $u_{k+1}(t)$

    if 采集到足够数据，即 $L \geq p(p+1)/2$，**then**

      设置 $j = j+1$

      根据式(4.21)与式(4.24)得到 $\bar{\mathcal{F}}^{[j]}$，并将 $\bar{\mathcal{F}}^{[j]}$ 复原为 $\mathcal{F}^{[j]}$

      根据式(4.19)得更新后的 $L_e^{q[j]}$ 和 $L_x^{q[j]}$

$$L_e^{q[j]} = -\left(F_{uu}^{[j]}\right)^{-1} F_{ue}^{[j]}$$

$$L_x^{q[j]} = -\left(F_{uu}^{[j]}\right)^{-1} F_{ux}^{[j]}$$

    end if

    设置 $t = t+1$

  until $t = N-1$

  设置 $k = k+1$

until $k = k_{\max}$

注释 4.1 相较于第三章的算法 3-1，本章所提算法 4-1 增加了最小二乘法估计参数矩阵的过程。除该点以外，对比算法 3-1 的控制律式(3.16)与算法 4-1 的控制律式(4.18)可知，此时算法 4-1 与算法 3-1 等同。因此，算法 4-1 的迭代学习控制更新律的收敛条件与收敛性分析可参考算法 3-1 的分析过程，接下来主要分析所提算法 4-1 中系统参数 $\mathcal{F}$ 的收敛过程。

4.3.3 收敛性分析

为证明所提算法 4-1 中系统参数 $\mathcal{F}$ 的收敛过程，给出以下引理及证明过程。

引理 4.1 式(4.12)等同于如下式(4.26)的迭代过程：

$$\mathcal{H}^{[j+1]} = \begin{bmatrix} Q & \\ & R \end{bmatrix} + \gamma \begin{bmatrix} I & -C(\theta) \\ L^{q[j]} & -L^{q[j]}C(\theta) \end{bmatrix}^{\mathrm{T}} \mathcal{H}^{[j]} \begin{bmatrix} I & -C(\theta) \\ L^{q[j]} & -L^{q[j]}C(\theta) \end{bmatrix}. \tag{4.26}$$

证明 根据优化目标(4.11)与 Q 函数定义式(4.12)，将 Q 函数$Q\big(e_k(t+1),\ \Delta w_{k+1}(t)\big)$对$\Delta w_{k+1}(t)$求微分，同时令$\partial Q\big(e_k(t+1),\ \Delta w_{k+1}(t)\big)\big/\partial\,\Delta w_{k+1}(t)=0$，可得

$$\begin{aligned}\frac{1}{2}\frac{\partial Q\big(e_k(t+1),\ \Delta w_{k+1}(t)\big)}{\partial \Delta w_{k+1}(t)} &= H_{we}e_k(t+1)+H_{ww}\Delta w_{k+1}(t)\\ &= 0,\end{aligned}$$

$$\Delta w_{k+1}(t)=L^q e_k(t+1), \tag{4.27}$$

其中，$L^q=-H_{ww}^{-1}H_{we}$。由此，式(4.12)可以写作

$$\begin{aligned}Q\big(e_k(t+1),\ \Delta w_{k+1}(t)\big)=&\begin{bmatrix}e_k(t+1)\\ \Delta w_{k+1}(t)\end{bmatrix}^{\mathrm{T}}\Bigg(\begin{bmatrix}Q & \\ & R\end{bmatrix}+\gamma\begin{bmatrix}I & -C(\theta)\\ L^{q[j]} & -L^{q[j]}C(\theta)\end{bmatrix}^{\mathrm{T}}\\ &\times\mathcal{H}^{[j]}\begin{bmatrix}I & -C(\theta)\\ L^{q[j]} & -L^{q[j]}C(\theta)\end{bmatrix}\Bigg)\begin{bmatrix}e_k(t+1)\\ \Delta w_{k+1}(t)\end{bmatrix}.\end{aligned} \tag{4.28}$$

又

$$Q\big(e_k(t+1),\ \Delta w_{k+1}(t)\big)=Z_{1,k+1}^{\mathrm{T}}(t+1)\mathcal{H}^{[j+1]}Z_{1,k+1}(t+1),$$

因此，可得

$$\mathcal{H}^{[j+1]}=\begin{bmatrix}Q & \\ & R\end{bmatrix}+\gamma\begin{bmatrix}I & -C(\theta)\\ L^{q[j]} & -L^{q[j]}C(\theta)\end{bmatrix}^{\mathrm{T}}\mathcal{H}^{j}\begin{bmatrix}I & -C(\theta)\\ L^{q[j]} & -L^{q[j]}C(\theta)\end{bmatrix},$$

即式(4.12)等同于如下式(4.26)的迭代过程，证毕。 ■

引理 4.2 可以分别描述为式

$$\mathcal{H}^{[j+1]}=\begin{bmatrix}Q+\gamma P^{[j]} & -\gamma P^{[j]}C(\theta)\\ -\gamma C(\theta)^{\mathrm{T}}P^{[j]} & R+\gamma C(\theta)^{\mathrm{T}}P^{[j]}C(\theta)\end{bmatrix}, \tag{4.29}$$

$$\begin{aligned}&\mathcal{F}^{[j+1]}\\ &=\begin{bmatrix}Q+\gamma P^{[j]} & -\gamma P^{[j]}C(\theta)A(\theta) & -\gamma P^{[j]}C(\theta)B(\theta)\\ -\gamma A(\theta)^{\mathrm{T}}C(\theta)^{\mathrm{T}}P^{[j]} & A(\theta)^{\mathrm{T}}\Big(R+\gamma C(\theta)^{\mathrm{T}}P^{[j]}C(\theta)\Big)A(\theta) & A(\theta)^{\mathrm{T}}\Big(R+\gamma C(\theta)^{\mathrm{T}}P^{[j]}C(\theta)\Big)B(\theta)\\ -\gamma B(\theta)^{\mathrm{T}}C(\theta)^{\mathrm{T}}P^{[j]} & B(\theta)^{\mathrm{T}}\Big(R+\gamma C(\theta)^{\mathrm{T}}P^{[j]}C(\theta)\Big)A(\theta) & \gamma B(\theta)^{\mathrm{T}}\Big(R+\gamma C(\theta)^{\mathrm{T}}P^{[j]}C(\theta)\Big)B(\theta)\end{bmatrix},\end{aligned} \tag{4.30}$$

$$L^{q[j]}=\Big(R+\gamma C(\theta)^{\mathrm{T}}P^{[j]}C(\theta)\Big)^{-1}\gamma C(\theta)^{\mathrm{T}}P^{[j]}, \tag{4.31}$$

其中，

$$P^{[j]}=\begin{bmatrix}I\\ L^{q[j]}\end{bmatrix}^{\mathrm{T}}\mathcal{H}^{[j]}\begin{bmatrix}I\\ L^{q[j]}\end{bmatrix}. \tag{4.32}$$

证明 根据引理 4.1，可得

$$\begin{aligned}\mathcal{H}^{[j+1]} &= \begin{bmatrix} Q & \\ & R \end{bmatrix} + \gamma \begin{bmatrix} I & -C(\theta) \\ L^{q[j]} & -L^{q[j]}C(\theta) \end{bmatrix}^{\mathrm{T}} \mathcal{H}^{[j]} \begin{bmatrix} I & -C(\theta) \\ L^{q[j]} & -L^{q[j]}C(\theta) \end{bmatrix} \\ &= \begin{bmatrix} Q & \\ & R \end{bmatrix} + \gamma \begin{bmatrix} I & -C(\theta) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} I \\ L^{q[j]} \end{bmatrix}^{\mathrm{T}} \mathcal{H}^{[j]} \begin{bmatrix} I \\ L^{q[j]} \end{bmatrix} \begin{bmatrix} I & -C(\theta) \end{bmatrix}.\end{aligned}$$

根据 $P^{[j]}$ 和 $\mathcal{F}^{[j]}$ 的关系，可得

$$\begin{aligned}\mathcal{H}^{[j+1]} &= \begin{bmatrix} Q & \\ & R \end{bmatrix} + \gamma \begin{bmatrix} I & -C(\theta) \end{bmatrix}^{\mathrm{T}} P^{[j]} \begin{bmatrix} I & -C(\theta) \end{bmatrix} \\ &= \begin{bmatrix} Q + \gamma P^{[j]} & -\gamma P^{[j]}C(\theta) \\ -\gamma C(\theta)^{\mathrm{T}} P^{[j]} & R + \gamma C(\theta)^{\mathrm{T}} P^{[j]} C(\theta) \end{bmatrix}.\end{aligned} \tag{4.33}$$

又因为式中 $\mathcal{H}^{[j]}$ 和 $\mathcal{F}^{[j]}$ 的关系

$$\mathcal{F}^{[j]} = \begin{bmatrix} I & 0 & 0 \\ 0 & A(\theta) & B(\theta) \end{bmatrix}^{\mathrm{T}} \mathcal{H}^{[j]} \begin{bmatrix} I & 0 & 0 \\ 0 & A(\theta) & B(\theta) \end{bmatrix},$$

可得

$$\begin{aligned}&\mathcal{F}^{[j+1]} \\ &= \begin{bmatrix} Q + \gamma P^{[j]} & -\gamma P^{[j]}C(\theta)A(\theta) & -\gamma P^{[j]}C(\theta)B(\theta) \\ -\gamma A(\theta)^{\mathrm{T}} C(\theta)^{\mathrm{T}} P^{[j]} & A(\theta)^{\mathrm{T}} \Big(R + \gamma C(\theta)^{\mathrm{T}} P^{[j]} C(\theta) \Big) A(\theta) & A(\theta)^{\mathrm{T}} \Big(R + \gamma C(\theta)^{\mathrm{T}} P^{[j]} C(\theta) \Big) B(\theta) \\ -\gamma B(\theta)^{\mathrm{T}} C(\theta)^{\mathrm{T}} P^{[j]} & B(\theta)^{\mathrm{T}} \Big(R + \gamma C(\theta)^{\mathrm{T}} P^{[j]} C(\theta) \Big) A(\theta) & \gamma B(\theta)^{\mathrm{T}} \Big(R + \gamma C(\theta)^{\mathrm{T}} P^{[j]} C(\theta) \Big) B(\theta) \end{bmatrix}.\end{aligned}$$

将式(4.33)代入式(4.27)，可得

$$L^{q[j]} = \Big(R + \gamma C(\theta)^{\mathrm{T}} P^{[j]} C(\theta) \Big)^{-1} \gamma C(\theta)^{\mathrm{T}} P^{[j]}.$$

至此，引理 4.2 证毕。 ■

引理 4.3　$P^{[j]}$ 与 $\mathcal{H}^{[j]}$ 的迭代关系是近似的，可以表示为

$$P^{[j+1]} = Q + \gamma P^{[j]} - \gamma^2 P^{[j]} C(\theta) \Big(\gamma C(\theta)^{\mathrm{T}} P^{[j]} C(\theta) + R \Big)^{-1} C(\theta)^{\mathrm{T}} P^{[j]}. \tag{4.34}$$

证明　将引理 4.2 中的式(4.29)代入 $P^{[j]}$ 和 $\mathcal{H}^{[j]}$ 的关系式(4.32)，可得

$$P^{[j+1]} = \begin{bmatrix} I \\ L^{q[j]} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} Q + \gamma P^{[j]} & -\gamma P^{[j]}C(\theta) \\ -\gamma C(\theta)^{\mathrm{T}} P^{[j]} & R + \gamma C(\theta)^{\mathrm{T}} P^{[j]} C(\theta) \end{bmatrix} \begin{bmatrix} I \\ L^{q[j]} \end{bmatrix}. \tag{4.35}$$

将式(4.31)代入式(4.35)，可得式(4.34)，证毕。 ■

定理 4.1　在 $P_0 = 0$，$\mathcal{H}_0 = 0$，$\mathcal{F}_0 = 0$ 的初始条件下，当迭代次数 j 趋向无穷大时，系统参数收敛。

证明　根据文献[68, 69]，当且 $P_0 = 0$ 迭代次数 j 趋向于无穷大时，$P^{[j]}$ 会趋向于离散代数 Riccati 的解 P。由引理 4.3 可知，$P^{[j]}$ 与 $\mathcal{H}^{[j]}$ 的迭代关系是近似的，根据引理 4.2 中的式(4.29)，当 $\mathcal{H}_0 = 0$，$\mathcal{F}_0 = 0$ 且 $j \to \infty$ 时，有

$$\mathcal{H}^{[j]} \to \mathcal{H} = \begin{bmatrix} Q+\gamma P & -\gamma PC(\theta) \\ -\gamma C(\theta)^{\mathrm{T}} P & R+\gamma C(\theta)^{\mathrm{T}} PC(\theta) \end{bmatrix}.$$

又式(4.30)，有

$\mathcal{F}^{[j]} \to$

$$\mathcal{F} = \begin{bmatrix} Q+\gamma P^{[j]} & -\gamma P^{[j]}C(\theta)A(\theta) & -\gamma P^{[j]}C(\theta)B(\theta) \\ -\gamma A(\theta)^{\mathrm{T}}C(\theta)^{\mathrm{T}}P^{[j]} & A(\theta)^{\mathrm{T}}\Big(R+\gamma C(\theta)^{\mathrm{T}}P^{[j]}C(\theta)\Big)A(\theta) & A(\theta)^{\mathrm{T}}\Big(R+\gamma C(\theta)^{\mathrm{T}}P^{[j]}C(\theta)\Big)B(\theta) \\ -\gamma B(\theta)^{\mathrm{T}}C(\theta)^{\mathrm{T}}P^{[j]} & B(\theta)^{\mathrm{T}}\Big(R+\gamma C(\theta)^{\mathrm{T}}P^{[j]}C(\theta)\Big)A(\theta) & \gamma B(\theta)^{\mathrm{T}}\Big(R+\gamma C(\theta)^{\mathrm{T}}P^{[j]}C(\theta)\Big)B(\theta) \end{bmatrix}.$$

因此，在算法 4-1 过程中，系统参数 $\mathcal{F}$ 收敛，证毕。 ■

4.3.4 计算复杂度对比分析

为更清晰论述本章所提方法在计算复杂度方面的优势，本小节首先在第 3.3.3 节介绍的基础上，简明介绍目前使用强化学习直接设计无模型迭代学习控制器的已有工作，提升范数优化框架下基于 Q 学习的无模型迭代学习控制算法[46]。

同样将将迭代学习控制过程转化为马尔可夫决策过程，五元组定义过程详见第 3.3.3 节，针对最优迭代学习控制问题，可以将 Q 函数写作如下二次型形式

$$\begin{aligned} Q\big(e_k,\ \Delta u_{k+1}\big) &= e_k^{\mathrm{T}} Q_{\mathrm{lift}} e_k + \Delta u_{k+1}^{\mathrm{T}} R_{\mathrm{lift}} \Delta u_{k+1} + \gamma e_{k+1}^{\mathrm{T}} P_{\mathrm{lift}} e_{k+1} \\ &= \begin{bmatrix} e_k \\ u_{k+1} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} Q_{\mathrm{lift}}+\gamma P_{\mathrm{lift}} & -\gamma P_{\mathrm{lift}} G(\theta) \\ -\gamma G(\theta)^{\mathrm{T}} P_{\mathrm{lift}} & R_{\mathrm{lift}}+\gamma G(\theta)^{\mathrm{T}} P_{\mathrm{lift}} G(\theta) \end{bmatrix} \begin{bmatrix} e_k \\ u_{k+1} \end{bmatrix}. \end{aligned} \tag{4.36}$$

根据优化目标，求解 $\partial Q\big(e_k,\ \Delta u_{k+1}\big)\big/\partial \Delta u_{k+1} = 0$，可得提升范数优化框架下基于 Q 学习的迭代学习控制算法

$$u_{k+1} = u_k - \big(M_{uu}\big)^{-1} M_{ue} e_k, \tag{4.37}$$

其中，$M_{uu} = R_{\mathrm{lift}} + \gamma G(\theta)^{\mathrm{T}} P_{\mathrm{lift}} G(\theta) \in \mathbb{R}^{mN\times mN}$，$M_{ue} = -\gamma G(\theta)^{\mathrm{T}} P_{\mathrm{lift}} \in \mathbb{R}^{mN\times lN}$。

未知参数向量 $\bar{\mathcal{M}}$ 的最小二乘法的解为

$$\bar{\mathcal{M}} = \Big[\Xi^{\mathrm{T}}\Xi\Big]^{-1}\Xi^{\mathrm{T}}\Psi, \tag{4.38}$$

其中，L_{lift} 为需要的实验批次个数，$\Xi \in \mathbb{R}^{L_{\mathrm{lift}}\times p_{\mathrm{lift}}}$，$\Psi \in \mathbb{R}^{L_{\mathrm{lift}}}$，$p_{\mathrm{lift}} = \big(l+m\big)N$。

接下来，将分别从迭代学习控制更新律的计算复杂度与最小二乘法求解的计算复杂度两方面，说明所提的算法 4-1 非提升范数优化框架下基于 Q 学习的无模型迭代学习控制算法相对于提升范数优化框架下基于 Q 学习的无模型迭代学习控制算法在计算复杂度和获取实验数据的效率方面的提升。

（1）迭代学习控制更新律的计算复杂度对比分析

本章所提算法 4-1 的迭代学习控制律更新律(4.18)共包含 2 个矩阵与矩阵的乘法、2 个矩阵与向量的乘法、2 个向量与向量的加法和 1 个矩阵的逆运算。因此，可得如表 4-1 所示的每个迭代批次的算法 4-1 的迭代学习控制更新律的计算量。

表 4-1 非提升范数优化框架下基于 Q 学习的无模型迭代学习控制更新律的计算量

算法	非提升范数优化框架下基于 Q 学习的 无模型迭代学习控制
乘法次数	$\left(m^3+m^2l+m^2n+ml+mn\right)N$
加法次数	$\left(m^3+m^2l+m^2n-m^2\right)N$
总计算量	$\left(2m^3+2m^2l+2m^2n+ml+mn-m^2\right)N$

文献[46]中所提的提升范数优化框架下基于 Q 学习的无模型迭代学习控制更新律(4.37)共包含 1 个矩阵与矩阵的乘法、1 个矩阵与向量的乘法、1 个向量与向量的加法和 1 个矩阵的逆运算。因此，可得如表 4-2 所示的每个迭代批次的提升范数优化框架下基于 Q 学习的无模型迭代学习控制更新律的计算量。

表 4-2 提升范数优化框架下基于 Q 学习的无模型迭代学习控制更新律的计算量

算法	提升范数优化框架下基于 Q 学习的 无模型迭代学习控制
乘法次数	$\left(m^3+m^2l\right)N^3+mlN^2$
加法次数	$\left(m^3+m^2l\right)N^3-m^2N^2$
总计算量	$\left(2m^3+2m^2l\right)N^3+\left(ml-m^2\right)N^2$

综上，由表 4-1 和表 4-2 可以看出，本章所提算法 4-1 的迭代学习控制更新律的计算复杂度与采样点个数 N 呈线性关系变化，即 $O(N)$；文献[46]中提升范数优化框架下基于 Q 学习的无模型迭代学习控制更新律的计算复杂度与采样点个数 N 呈 3 次方阶趋势增加，即 $O(N^3)$。因此，本章所提算法 4-1 对于采样点个数 N 较大的情形更有优势，计算效率更高。

（2）最小二乘法的计算复杂度对比分析

定义最小二乘法的观测矩阵为 $X\in\mathbb{R}^{n_{\text{samples}}\times n_{\text{features}}}$，观测向量为 $Y\in\mathbb{R}^{n_{\text{features}}}$，最小二乘法中求解参数向量 $\theta=\left(X^{\mathrm{T}}X\right)^{-1}X^{\mathrm{T}}Y$ 的一步包含 1 个矩阵与矩阵的乘法、2 个矩阵与向量的乘法和 1 个矩阵的逆运算，所需的计算量如表 4-3 所示。

表 4-3 最小二乘法的计算量

算法	最小二乘法
乘法次数	$n_{\text{features}}^3+n_{\text{samples}}n_{\text{features}}^2+n_{\text{samples}}n_{\text{features}}+n_{\text{features}}^2$
加法次数	$n_{\text{features}}^3+n_{\text{samples}}n_{\text{features}}^2+n_{\text{samples}}n_{\text{features}}-n_{\text{features}}^2-2n_{\text{features}}$
总计算量	$2n_{\text{features}}^3+2n_{\text{samples}}n_{\text{features}}^2+2n_{\text{samples}}n_{\text{features}}-2n_{\text{features}}$

本算法 4-1 的最小二乘法步骤中，n_{features} 和 n_{samples} 分别为

$$n_{\text{features,算法4-1}} = p_2(p_2+1)\big/2 = \big(n+l+m\big)\big(n+l+m+1\big)\big/2,$$

$$n_{\text{samples,算法4-1}} = L \geq n_{\text{features,算法4-1.}}$$

文献[46]中基于 Q 学习的算法的最小二乘法步骤中，n_{samples} 和 n_{features} 分别为

$$n_{\text{features,lift}} = p_{\text{lift}}(p_{\text{lift}}+1)\big/2 = \big[\big(l+m\big)N\big]\big[\big(l+m\big)N+1\big]\big/2,$$

$$n_{\text{samples,lift}} = L_{\text{lift}} \geq n_{\text{features,lift.}}$$

综上，由于本章所提算法 4-1 的最小二乘法步骤中 n_{features} 和 n_{samples} 均不随着与采样点个数 N 变化；提升范数优化框架下基于 Q 学习的无模型迭代学习控制算法的最小二乘法步骤中 n_{features} 和 n_{samples} 与采样点个数 N 呈 2 次方阶趋势增加，即 $O(N^2)$；因此，在采样点个数 N 大的情形，算法 4-1 对应表 4-3 的乘法次数，加法次数与总计算量的增长趋势均更小。因此，本章所提的算法 4-1 在一方面，有助于降低最小二乘方法的计算复杂度，提升了计算效率提升了实验效率；另一方面，有助于减少收集用于最小二乘法数据的实验批次数量。

4.4 仿真实例

为有效对比算法 3-1 与算法 4-1，继续选取第三章的直流电动机系统[73]作为仿真对象，建模过程与物理量的取值详见第 3.4 节，同样定义输出变量为电机转速 $u=\omega$，输入变量为电枢电流 $y=i_a$，状态变量为 $x=[i_a \quad \omega]^{\mathrm{T}}$，设置系统仿真时间 $T=20\text{s}$，采样时间 $T_s=0.1\text{s}$，即采样点总个数 $N=200$，使用零阶保持器的方法将直流电机系统离散化，则系统的状态空间方程的参数矩阵分别为

$$A=\begin{bmatrix}0.7684 & -0.0198\\ 0.0710 & 0.9990\end{bmatrix},\ B=\begin{bmatrix}0.1099\\ 0.0046\end{bmatrix},\ C=\begin{bmatrix}0 & 1\end{bmatrix}.$$

为实现无模型控制，这里的系统参数矩阵主要用于生成求解迭代学习控制律所需的实验数据矩阵(4.22)和(4.23)，而非直接用于求解迭代学习控制律。θ 表示参数矩阵为常量，因此不需要额外为其赋值。

给定直流电机系统的控制任务为：120 个批次内（$k_{\max}=120$），直流电机的转速跟踪上给定参考轨迹：

$$y_d = 2\Big(\sin\big(2\pi t\big/20\big)+\sin(2\pi t\big/30)\Big).$$

（1） 非提升范数优化框架下基于 Q 学习的无模型迭代学习控制

设置控制器参数 $Q=I$，$R=0.1I$，$\gamma=0.85$，实验数据数量 $L=21$，则算法 4-1 的仿真结果如图 4-1 和图 4-2 所示。从图 4-1 可以看出随着迭代批次的增加，跟踪误差可以逐渐收敛，并且每 L 批次，控制增益进行依次更新，随着迭代批次的增加，控制效果逐渐稳定。从图 4-2 可以看出，随着迭代批次的增加，电机转速可以逐渐跟踪上给定

的参考轨迹，因此算法 4-1 能完成给定的控制任务。

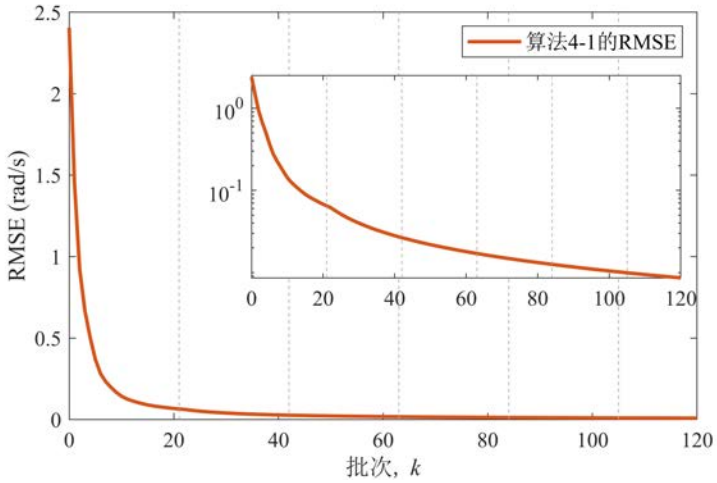

图 4-1 直流电机系统在算法 4-1 下的电机转速的跟踪误差 2-范数收敛图和对数收敛图

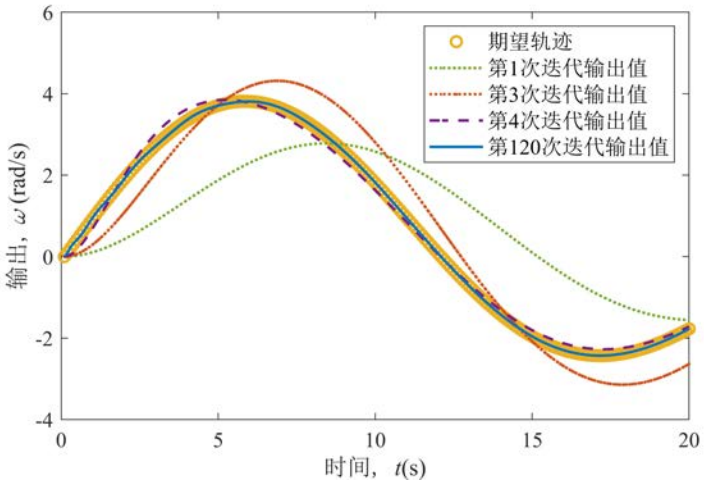

图 4-2 直流电机系统在算法 4-1 下的电机转速期望值轨迹与实际输出值图

（2）算法对比

Case4-1：选取 P 型迭代学习控制算法[22]、非提升范数优化框架下的迭代学习控制算法[63]和第三章所提算法 3-1 作为对比算法与本章所提算法 4-1 进行对比实验。P 型迭代学习控制算法的控制增益为$k=0.03$，非提升范数优化框架下的迭代学习控制算法的控制律如式(2.13)所示，控制器参数$Q_2=I$，$R_2=0.5I$，第三章所提算法 3-1 控制器参数为$Q=I$，$R=0.1I$，$\gamma=0.85$，本章所提算法 4-1 控制器参数为$Q=I$，$R=0.1I$，$\gamma=0.85$，实验数据数量为$L=21$。从图 4-3 可以看出，第三章所提算法 3-1 和本章所提算法 4-1 的跟踪误差的收敛速度与收敛精度均优于 P 型迭代学习控制算法和非提升范数优化框架下的迭代学习控制算法，优于 P 型迭代学习控制算法的原因是因为引入了基于模型的优化理论，优于非提升范数优化框架下的迭代学习控制算法的原因是因为第三章所提算法 3-1 和本章所提算法 4-1 引入了强化学习思想，在利用模型信息的基础上，进一步通过未来收益指导当前动作，即下一批次的控制输入信号。同时，本章所提算法 4-1 通过引入 Q 学习方法，将算法 3-1 改进成了无模型的算法 4-1。因此，随着迭代批次的增加，算法 4-1 可以在无需模型信息的情况下，达到近似算法 3-1 的控制效果。

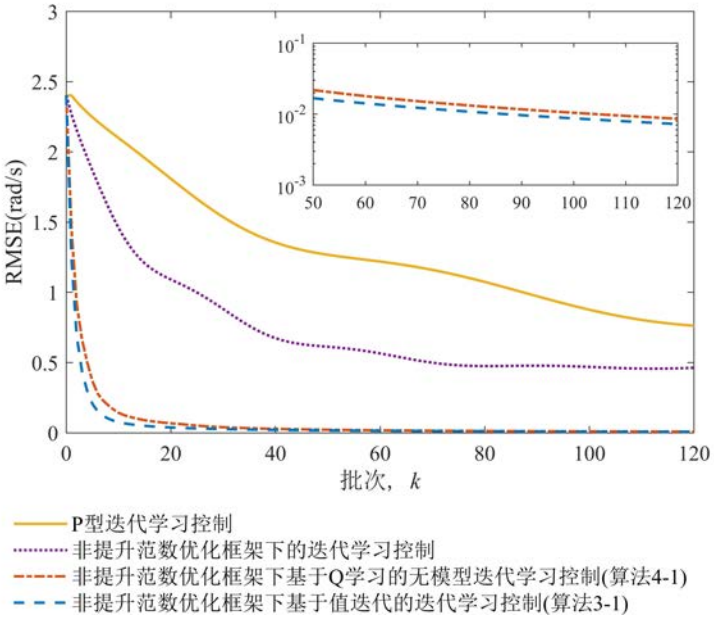

图 4-3 直流电机系统在 Case4-1 下的电机转速的跟踪误差 2-范数收敛对比图

Case4-2：选取提升范数优化框架下基于 Q 学习的无模型迭代学习控制算法[46]与本章所提算法 4-1 进行对比实验。受硬件内存限制，设置系统仿真时间$T = 5\text{s}$，采样时间$T_s = 0.2\text{s}$，即采样点总个数$N = 25$。本章所提算法 4-1 的控制器参数设置为$Q = I$，$R = 0.5I$，$\gamma = 0.85$，提升范数优化框架下基于 Q 学习的迭代学习控制算法的控制器参数设置为$Q_{\text{lift}} = I$，$R_{\text{lift}} = 0.5I$，$\gamma = 0.85$，实验数据数量为$L = 21$，$L_{\text{lift}} = 500$，分别进行 1600 次试验（$k_{\max} = 1600$）。图 4-4 为提升范数优化框架下基于 Q 学习的迭代学习控制算法与本章所提算法 4-1 的误差收敛对比图，可以看出本章所提算法 4-1 在收敛速度与收敛精度方面均更优。同时，灰色实线为本章所提算法 4-1 进行最小二乘求解系统模型的批次，灰色虚线为提升范数优化框架下基于 Q 学习的迭代学习控制算法进行最小二乘求解系统模型的批次，可以看出本章所提算法在提升最小二乘求解系统模型的效率上有所提升。

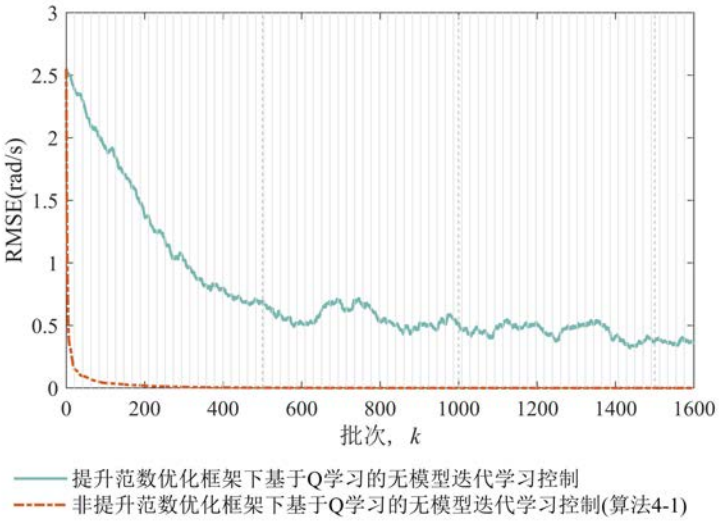

图 4-4 直流电机系统在 Case4-2 下的电机转速的跟踪误差 2-范数收敛对比图

Case4-3：选取提升范数优化框架下基于 Q 学习的无模型迭代学习控制算法[46]与本章所提算法 4-1 进行计算复杂度对比实验，本章所提算法 4-1 的控制器参数设置为 $Q=I$，$R=0.5I$，$\gamma=0.85$，提升范数优化框架下基于 Q 学习的迭代学习控制算法的控制器参数设置为$Q_{\text{lift}}=I$，$R_{\text{lift}}=0.5I$，$\gamma=0.85$，为保障提升范数优化框架下基于 Q 学习的迭代学习控制算法的可以收集足够批次的实验批次数量来进行最小二乘法求解，两种算法均进行 10000 次试验（$k_{\max}=10000$），仿真程序计算在一台 3.3GHz AMD 锐龙 9 5900HX 处理器和 32GB RAM 的笔记本电脑上使用 MATLAB R2020b 进行。

由图 4-5 可以看出，提升范数优化框架下基于 Q 学习的无模型迭代学习控制算法，一方面，其计算时间与采样点个数 N 呈 3 次方阶增长关系，即 $O(N^3)$，另一方面，由于需要求解的系统模型矩阵过大，受平台的硬件内存限制，采样点个 N 最大仅为 285。这意味着，如果采样率为 0.1kHz，则批次长度需要限制在小于等于 2.85s。对于批次长度更大的情形，本计算平台无法分配更多的内存用于矩阵计算，在实际应用场景中也有类似的局限性，这些结论与文献[74]所得结论一致。本章所提算法 4-1 的计算时间与采样点个数 N 呈线性增长关系，即 $O(N)$，并且不会占用过多内存。

图 4-6 给出了两种算法的最小实验批次数量随着采样点个数 N 增长的变化对比图，从图 4-6 可以看出，提升范数优化框架下基于 Q 学习的无模型迭代学习控制算法所需的最小实验批次数量与采样点个数 N 呈 2 次方阶增长关系，即 $O(N^2)$，并且在图中的点 A 位置，$N=285$ 时，最小实验批次数量会达到 162735，这意味着，如果采样率为 0.1kHz，则系统需要进行 1627.35s，即 27.12min 的实验才可以更新一次迭代学习控制更新律的控制增益，这样的计算效率不利于算法在采样点个数多情形下的实际应用。相比之下，本章所提算法 4-1 所需的最小实验批次数量，如图中的点 B 和点 C 位置，始终为常数 10，不会随着采样点个数 N 的增长而增长，更符合实际应用场景。

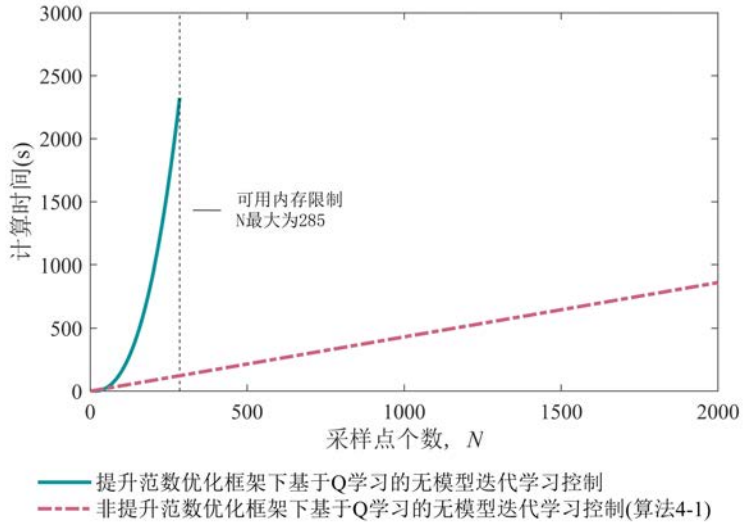

图 4-5 Case4-3 下不同算法的计算时间随采样点个数增长对比图

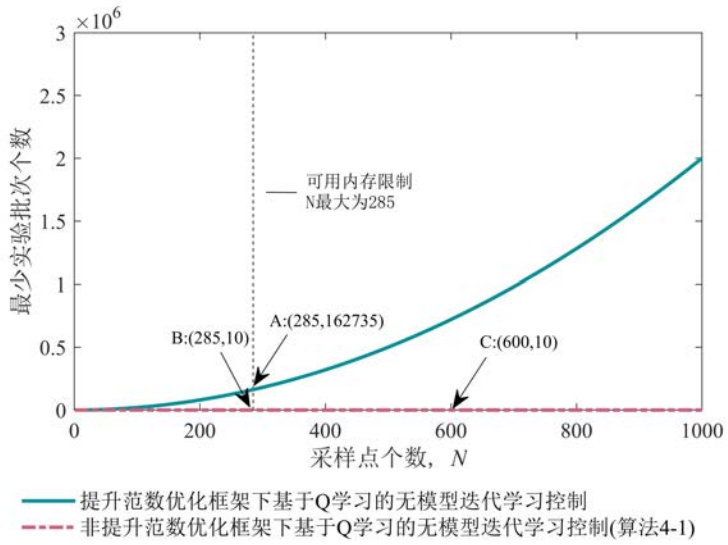

图 4-6 Case4-3 下不同算法的最少实验批次个数随采样点个数增长对比图

4.5 小结

本章研究了一类模型信息未知的线性离散系统的跟踪问题，在非提升框架范数优化下，提出了一种基于 Q 学习的无模型迭代学习控制方法。将迭代学习控制过程描述为马尔可夫决策过程，引入 Q 学习算法，通过最小化 Q 函数得到包含模型信息的迭代学习控制更新律，并收集系统数据，通过最小二乘方法估计更新律所需的模型信息，从而实现无需模型参数的迭代学习控制方法，并证明了该无模型方法的收敛性。进一步地，分析了所提方法在计算复杂度和求解模型所需实验批次数量方面的优势。最后，通过直流电动机的仿真，验证了所提方法的有效性和优势。

第五章　基于Q学习的故障估计迭代学习容错控制与优化方法

5.1 引言

第三章与第四章研究了使用强化学习直接设计迭代学习控制设计控制算法的结合方式，本章对另一类结合方法，即间接发挥强化学习的复杂决策优势来协助处理迭代学习控制的局限性开展研究。沿着时间和批次同时变化的未知故障会影响迭代学习控制器的跟踪性能，目前已有的故障估计方法与迭代学习控制容错方法，多使用固定结构的故障观测器[78-80]或固定结构的容错控制器[81-83]应对故障的不利影响。为了更好地估计动态的故障，提升随时间和批次同时变化的执行器故障下的跟踪性能，本章引入 Q 学习算法设计了一个动态调整的故障估计器，并使用故障估计结果，主动调节基于范数优化方法设计的迭代学习容错控制器的参数。

本章针对一类随时间和批次同时变化的执行器故障下的线性离散系统进行研究，主要内容和结构如下：第 5.2 节介绍了迭代学习容错控制算法的设计问题；第 5.3 节采用范数优化理论设计迭代学习容错控制器，通过估计故障来调整控制器，以抵消故障的影响，引入了 Q 学习算法，通过不断地调整设计的故障估计器以适应随时间和批次同时变化的故障，给出了整体的算法描述，提出了系统在该学习律下跟踪误差有界收敛的条件并给出了相应证明；第 5.4 节通过移动机器人的仿真，验证了所提方法的有效性和优势；第 5.5 节对本章内容进行小结。

5.2 问题描述

考虑如下一类随时间和批次同时变化的执行器故障下线性离散系统

$$\begin{cases} x_k(t+1) = Ax_k(t) + B\Gamma_k(t)u_k(t), \\ \quad y_k(t) = Cx_k(t), \end{cases} \tag{5.1}$$

其中，下标 k 表示迭代批次；$t \in \left[0,\ N\right]$ 表示一个重复运行周期 T 内的采样时刻，N 为一个批次的采样点个数；$x_k(t) \in \mathbb{R}^n$，$u_k(t) \in \mathbb{R}^m$，$y_k(t) \in \mathbb{R}^l$ 分别代表系统在第 k 批次的状态变量、控制输入以及输出信号；A，B，C 分别表示具有相应维数的系统参数矩阵 $\Gamma_k(t)$ 表示随时间和批次同时变化的故障参数矩阵；为保证系统输出可控，需要满足 CB 满秩；$x_k(0)$ 表示系统在第 k 批次运行时的初始状态值，假设系统在不同批次的初始状态值相同，即 $\forall k$，$x_k(0) = x_0$。

执行器故障下的输入信号定义为 $u_k^F(t) = \Gamma_k(t)u_k(t)$ 和 $u_{i,k}^F(t) = \Gamma_{i,k}(t)u_k(t)$，即

$$u_k^F(t) = \left[u_{1,k}^F(t),\ u_{2,k}^F(t),\ \cdots,\ u_{m,k}^F(t)\right]^{\mathrm{T}}. \tag{5.2}$$

故障参数矩阵代表执行器有效因子，定义为

$$\Gamma_k(t) = \mathrm{diag}\left\{\Gamma_{1,k}(t),\ \Gamma_{2,k}(t),\ \cdots,\ \Gamma_{m,k}(t)\right\}, \tag{5.3}$$

其中，每个元素有相应的范围

$$\underline{\Gamma}_i \le \Gamma_{i,k}(t) \le \overline{\Gamma}_i,\ i = \{1,\ 2,\ \cdots,\ m\}. \tag{5.4}$$

与上述定义类似，将估计故障定义为

$$\tilde{\Gamma}_k(t) = \mathrm{diag}\{\tilde{\Gamma}_{1,k}(t),\ \tilde{\Gamma}_{2,k}(t),\ \cdots,\ \tilde{\Gamma}_{m,k}(t)\}, \tag{5.5}$$

$$\underline{\Gamma}_i \le \tilde{\Gamma}_{i,k}(t) \le \overline{\Gamma}_i,\ i = \{1,\ 2,\ \cdots,\ m\}. \tag{5.6}$$

将故障参数的下界矩阵和上界矩阵分别定义为

$$\underline{\Gamma} = \mathrm{diag}\{\underline{\Gamma}_1,\ \underline{\Gamma}_2,\ \ldots,\ \underline{\Gamma}_m\}, \tag{5.7}$$

$$\overline{\Gamma} = \mathrm{diag}\{\overline{\Gamma}_1,\ \overline{\Gamma}_2,\ \ldots,\ \overline{\Gamma}_m\}. \tag{5.8}$$

将故障参数的最小值和最大值分别定义为

$$\underline{\Gamma}_{\min} = \min_{1\le i\le m} \underline{\Gamma}_i, \tag{5.9}$$

$$\overline{\Gamma}_{\max} = \max_{1\le i\le m} \overline{\Gamma}_i. \tag{5.10}$$

上述故障参数的上下界矩阵元素$\underline{\Gamma}_i\left(0 \le \underline{\Gamma}_i \le 1\right)$和$\overline{\Gamma}_i\left(\overline{\Gamma}_i \ge \underline{\Gamma}_i\right)$是已知的，即故障参数矩阵$\Gamma_k(t)$的元素$\Gamma_{i,k}(t)$和估计故障矩阵$\tilde{\Gamma}_k(t)$的元素$\tilde{\Gamma}_{i,k}(t)$是未知的但在已知范围内变化。执行器故障的类型，主要分为以下几种情况[84]：$\Gamma_{i,k}(t)=0$代表第i个执行器完全故障；$\Gamma_{i,k}(t)=1$代表第i个执行器正常工作；$0<\Gamma_{i,k}(t)<1$代表第i个执行器有部分驱动力；$\Gamma_{i,k}(t)>1$代表第i个执行器过载，驱动力大于正常的执行器水平。

使用提升技术将执行器故障下系统(5.1)的离散状态空间模型转化为时间序列形式的提升模型

$$y_k = G\Gamma_k u_k + d, \tag{5.11}$$

其中，G和d分别代表系统输入输出映射矩阵和初始状态响应，定义为

$$G = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \cdots & CB \end{bmatrix}, \tag{5.12}$$

$$d = [(CA)^{\mathrm{T}},(CA^2)^{\mathrm{T}},\ldots,(CA^N)^{\mathrm{T}}]^{\mathrm{T}} x_0. \tag{5.13}$$

Γ_k是一个对角阵，定义为

$$\Gamma_k = \begin{bmatrix} \Gamma_k(0) & & & & \\ & \Gamma_k(1) & & & \\ & & \Gamma_k(2) & & \\ & & & \ddots & \\ & & & & \Gamma_k(N-1) \end{bmatrix}. \tag{5.14}$$

根据式(5.11)可以得出执行器故障下系统(5.1)的名义提升模型

$$y_k = G\tilde{\Gamma}_k u_k + d, \tag{5.15}$$

其中，$\tilde{\Gamma}_k$ 定义为

$$\tilde{\Gamma}_k = \begin{bmatrix} \tilde{\Gamma}_k(0) & & & & \\ & \tilde{\Gamma}_k(1) & & & \\ & & \tilde{\Gamma}_k(2) & & \\ & & & \ddots & \\ & & & & \tilde{\Gamma}_k(N-1) \end{bmatrix}. \tag{5.16}$$

执行器故障下系统(5.11)第 k 批次的实际跟踪误差定义为

$$e_k = y_d - G\Gamma_k u_k - d, \tag{5.17}$$

其中，y_d 为期望参考轨迹。

执行器故障下名义系统第 k 批次的数值跟踪误差定义为

$$\tilde{e}_k = y_d - G\tilde{\Gamma}_k u_k - d. \tag{5.18}$$

定义 5.1（迭代学习容错控制算法设计问题） 本章节考虑的执行器故障下迭代学习容错控制算法的设计问题是设计一个引入故障估计信息主动调节的迭代学习容错控制算法

$$u_{k+1} = f(u_k,\ e_k,\ \tilde{\Gamma}_k,\ \tilde{\Gamma}_{k+1}). \tag{5.19}$$

迭代学习控制容错算法(5.19)中的故障估计信息由第 5.3.2 节中的故障估计过程提供。随着迭代学习控制更新过程的进行，对输入信号进行迭代修正，使得输出信号可以在执行器故障下保持良好的跟踪性能，即

$$\lim_{k\to\infty} \left\| e_k \right\| \le \epsilon_e, \tag{5.20}$$

其中，一个相对小的正常数 ϵ_e 是跟踪误差 e_k 的范数的上界。需要注意的是，传统的迭代学习控制算法是为在迭代域内重复执行的任务设计的，但本章中考虑的是随时间和批次同时变化的故障，这会对控制器造成可以用一些方法抑制但不能完全消除的影响。因此本章设计的迭代学习容错控制算法不能完成传统迭代学习控制任务的控制目标，即跟踪误差 e_k 不能收敛到 0，跟踪误差 e_k 有界收敛在实际应用中也是合理的。

本章所设计的迭代学习容错控制算法的设计目标是在故障存在的情形下保持控制系统相对高的控制性能，而不要求跟踪误差收敛到 0，即跟踪误差允许有界收敛以取得容错控制的可靠性和迭代学习控制的跟踪性能之间的平衡。

5.3 基于范数优化的迭代学习容错控制

本节首先介绍了基于范数优化的迭代学习容错控制算法设计，旨在解决第 5.2 节中提出的定义 5.1 迭代学习容错控制设计问题；随后给出了基于 Q 学习的故障估计算法设计。接下来，详细描述了迭代学习控制框架下的故障估计与容错控制算法流程；最后分析了所提迭代学习容错控制算法的收敛性。

5.3.1 迭代学习容错控制算法设计

为了解决定义 5.1 中的迭代学习容错控制设计问题，引入范数优化框架来优化每个

批次的多目标性能指标[85]。类似预备知识范数优化框架中式(2.8)的形式，性能指标定义为

$$J_{k+1} \triangleq \left\| y_d - G\tilde{\Gamma}_{k+1}u_{k+1} - d \right\|_Q^2 + \left\| u_{k+1} - u_k \right\|_R^2, \tag{5.21}$$

其中，性能指标由两部分组成：数值跟踪误差和批次间的输入信号变化。最小化跟踪误差的目的是完成迭代学习控制的任务，即跟踪上期望目标。减小批次间输入信号变化，即使得批次间输入信号变化平滑是为了增加算法的鲁棒性。Q 和 R 分别为数值跟踪误差和批次间输入信号变化的对称正定权重矩阵，以表示优化过程中误差减小和鲁棒性的优先级，即 $Q = Q^{\mathrm{T}} > 0$，$R = R^{\mathrm{T}} > 0$。不失一般性，可以取权重矩阵 $Q = qI$，$R = rI$。

最优控制输入通过最小化性能指标得到，表示为

$$u_{k+1} = \arg\min_{u_{k+1}} \left\{ J_{k+1} \right\}. \tag{5.22}$$

定理 5.1（基于范数优化的迭代学习控制容错算法）求解式(5.22)的最优解，可以得到迭代学习控制更新律

$$u_{k+1} = L_{k+1}^u u_k + L_{k+1}^e e_k, \tag{5.23}$$

其中，L_{k+1}^u 和 L_{k+1}^e 分别为第 k 批次的输入学习增益和跟踪误差学习增益，定义为

$$L_{k+1}^u = \left(\tilde{\Gamma}_{k+1}^{\mathrm{T}} G^{\mathrm{T}} Q G \tilde{\Gamma}_{k+1} + R \right)^{-1} \left(\tilde{\Gamma}_{k+1}^{\mathrm{T}} G^{\mathrm{T}} Q G \tilde{\Gamma}_{k} + R \right), \tag{5.24}$$

$$L_{k+1}^e = \left(\tilde{\Gamma}_{k+1}^{\mathrm{T}} G^{\mathrm{T}} Q G \tilde{\Gamma}_{k+1} + R \right)^{-1} \tilde{\Gamma}_{k+1}^{\mathrm{T}} G^{\mathrm{T}} Q, \tag{5.25}$$

证明 根据(2.3)与式(2.4)的诱导范数定义，将式(5.17)代入式(5.21)可得

$$\begin{aligned} J_{k+1} = & \left(y_d - G\tilde{\Gamma}_{k+1}u_{k+1} - d \right)^{\mathrm{T}} Q \left(y_d - G\tilde{\Gamma}_{k+1}u_{k+1} - d \right) \\ & + \left(u_{k+1} - u_k \right)^{\mathrm{T}} R \left(u_{k+1} - u_k \right). \end{aligned} \tag{5.26}$$

根据式(5.22)的优化目标，将性能指标 J_{k+1} 对 u_{k+1} 求微分，并令 $\partial J_{k+1} / \partial u_{k+1} = 0$，可得

$$R\left(u_{k+1} - u_k \right) - \tilde{\Gamma}_{k+1}^{\mathrm{T}} G^{\mathrm{T}} Q \left(\tilde{e}_k + G\tilde{\Gamma}_k u_k + d - G\tilde{\Gamma}_{k+1}u_{k+1} - d \right) = 0. \tag{5.27}$$

将式(5.27)合并同类项得到

$$\left(\tilde{\Gamma}_{k+1}^{\mathrm{T}} G^{\mathrm{T}} Q G \tilde{\Gamma}_{k+1} + R \right) u_{k+1} = \left(\tilde{\Gamma}_{k+1}^{\mathrm{T}} G^{\mathrm{T}} Q G \tilde{\Gamma}_{k} + R \right) u_k + \tilde{\Gamma}_{k+1}^{\mathrm{T}} G^{\mathrm{T}} Q \tilde{e}_k. \tag{5.28}$$

因为权重矩阵 Q 和 R 是对称正定矩阵，故矩阵 $\tilde{\Gamma}_{k+1}^{\mathrm{T}} G^{\mathrm{T}} Q G \tilde{\Gamma}_{k+1}$ 是对称非负定矩阵，所以矩阵 $\left(\tilde{\Gamma}_{k+1}^{\mathrm{T}} G^{\mathrm{T}} Q G \tilde{\Gamma}_{k+1} + R \right)$ 是对称正定矩阵，即 $\left(\tilde{\Gamma}_{k+1}^{\mathrm{T}} G^{\mathrm{T}} Q G \tilde{\Gamma}_{k+1} + R \right)$ 可逆。同时，将名义提升模型的数值跟踪误差 $\tilde{e}_k$ 替换为实际系统的跟踪误差 e_k，式(5.28)可改写为

$$\begin{aligned} u_{k+1} = & \left(\tilde{\Gamma}_{k+1}^{\mathrm{T}} G^{\mathrm{T}} Q G \tilde{\Gamma}_{k+1} + R \right)^{-1} \left(\tilde{\Gamma}_{k+1}^{\mathrm{T}} G^{\mathrm{T}} Q G \tilde{\Gamma}_{k} + R \right) u_k \\ & + \left(\tilde{\Gamma}_{k+1}^{\mathrm{T}} G^{\mathrm{T}} Q G \tilde{\Gamma}_{k+1} + R \right)^{-1} \tilde{\Gamma}_{k+1}^{\mathrm{T}} G^{\mathrm{T}} Q e_k. \end{aligned} \tag{5.29}$$

由此可得迭代学习容错控制更新律式(5.23)，证毕。 ■

名义提升模型的数值跟踪误差$\tilde{e}_k$依赖估计故障信息$\tilde{\Gamma}_k$，并且缺乏真实的故障信息Γ_k，会减弱算法的鲁棒性。因此，将名义提升模型的数值跟踪误差$\tilde{e}_k$替换为实际系统测量所得的跟踪误差e_k，从而引入真实的故障信息Γ_k，增强本章所提迭代学习容错控制算法的鲁棒性。

注释 5.1 为了保证本章所提迭代学习容错控制算法的收敛条件，Q和R的选择需要满足收敛条件式(5.39)。除此之外，Q和R的选择虽然没有严格的规则，但仍有一定的模式可以遵循：（1）增大Q中元素的数值，可以提高所提算法的收敛速度；（2）减小R中元素的数值，可以减小输入能量变化以及提高所提算法的鲁棒性；（3）Q和R的选择的是相互制约的。关于Q和R的选择带来的影响的探讨具体可以参考文献[86]，选择效果的验证可以参考第 5.4 节的仿真部分。

5.3.2 基于 Q 学习的故障估计算法设计

故障估计的目的在于为迭代学习容错控制更新律(5.23)提供估计故障信息$\tilde{\Gamma}_{k+1}$和$\tilde{\Gamma}_k$。由于故障的上下界信息是已知的，在已知范围内寻找未知的故障信息可以看作在固定范围的网格世界寻找终点，故本章采用用于解决类似上述问题的 Q 学习算法，故障估计的任务是在第k批次的第t时刻估计故障矩阵$\Gamma_{k+1}(t)$。

在基于 Q 学习算法的故障估计过程中，将智能体看作故障估计器，环境看作控制系统，将故障估计过程描述为马尔可夫决策过程，从而使得故障估计问题可以用强化学习方法求解。定义如下五元组$\left(\mathcal{S},\mathcal{A},f,\mathcal{R},\gamma\right)$：

- $\mathcal{S}$为状态空间，状态$s\in\mathcal{S}$定义为

$$s=\left[\tilde{\Gamma}_{1,k}(t),\ \tilde{\Gamma}_{2,k}(t),\ \cdots,\ \tilde{\Gamma}_{m,k}(t)\right]; \tag{5.30}$$

- $\mathcal{A}$为动作空间，$\Delta\tilde{\Gamma}_{i,k}(t)$为动作改变的故障数值偏差，动作$a\in\mathcal{A}$定义为

$$a=\left[\Delta\tilde{\Gamma}_{1,k}(t),\Delta\tilde{\Gamma}_{2,k}(t),\cdots,\Delta\tilde{\Gamma}_{m,k}(t)\right]; \tag{5.31}$$

- f为状态转移函数，定义为

$$\begin{aligned} s' &= s+a \\ &=\left[\tilde{\Gamma}_{1,k}(t)+\Delta\tilde{\Gamma}_{1,k}(t),\ \tilde{\Gamma}_{2,k}(t)+\Delta\tilde{\Gamma}_{2,k}(t),\ \cdots,\ \tilde{\Gamma}_{m,k}(t)+\Delta\tilde{\Gamma}_{m,k}(t)\right]; \end{aligned} \tag{5.32}$$

- $\mathcal{R}$为收益函数，定义为

$$\mathcal{R}=\begin{cases}\mathcal{R}_c,\ \text{if } \mathcal{L}\le\varepsilon_{\mathcal{L}},\\ -1,\ \text{if } \mathcal{L}>\varepsilon_{\mathcal{L}},\end{cases} \tag{5.33}$$

其中，$\mathcal{R}_c=n_\Gamma^m$是一个有关状态数量n_Γ的常数，是$\varepsilon_{\mathcal{L}}$代表故障估计准确度的损失函数阈值，$\mathcal{L}$为损失函数，旨在评估故障矩阵估计的准确度，设计为

$$\mathcal{L}=\left\|x_k(t+1)-Ax_k(t)-B\tilde{\Gamma}_k(t)u_k(t)\right\|^2; \tag{5.34}$$

- γ为折扣因子，决定了未来收益的现在价值，并且$\gamma\in\left(0,1\right]$。

Q 学习算法采用ϵ-贪心算法作为动作选择策略$\pi(s)$：

$$\mathcal{A} = \pi(s) = \begin{cases} \arg\max\limits_{a} Q(s,a), \text{if } p < (1-\epsilon), \\ \text{random action, if } p \leq \epsilon, \end{cases} \tag{5.35}$$

其中，ϵ为贪心概率，p为动作选择概率。式(5.35)表示智能体有ϵ的概率随机选择一个动作，并且有$(1-\epsilon)$的概率选择返回当前状态动作值函数最大值的动作。状态动作值函数的更新式为

$$Q^{\pi}(s,a) \leftarrow Q^{\pi}(s,a) + \alpha\Big[\mathcal{R}^{a}_{s\to s'} + \gamma \max_{a'} Q^{\pi}(s',a') - Q^{\pi}(s,a)\Big], \tag{5.36}$$

其中，$\alpha \in [0,\ 1]$是学习率。

本章提出的故障估计算法的一个优势在于引入了 Q 学习算法，将每个批次随采样时间不断变化的故障估计任务分解为在每个采样时间进行故障值估计的子任务。因此，这种分解可以减少 Q 学习中状态数据的数量，从而降低强化学习算法的计算负担。同时，这种分解还能为容错控制提供每个采样时间估计故障值，以调节控制器参数。

注释 5.2 由于时间因果关系的存在，本章估计的故障矩阵$\tilde{\Gamma}_{k+1}$实际上是实际故障矩阵Γ_k的直接近似。然而，估计的$\tilde{\Gamma}_{k+1}$总是能够在每次试验中跟上实际的Γ_k变化。因此，基于故障估计结果的迭代学习容错控制仍然能够获得良好的控制性能。

5.3.3 算法描述

在本小节中，描述了所提算法 5-1 和算法 5-2 的详细流程。其中，如下的算法 5-1 展示了基于范数优化的迭代学习容错控制算法的具体流程。

算法 5-1 基于范数优化的迭代学习容错控制算法

Input： 系统参数矩阵A，B和C；期望参考轨迹y_d；权重矩阵Q和R；总采样点个数N；最大的迭代学习控制迭代次数$k_{\max}$。

Output： 下一批次的输入u_{k+1}。

1： **Initialization：** 设置$k=0$，$i=0$；初始控制输入u_0；初始状态x_0；初始估计故障矩阵$\tilde{\Gamma}_1$。

2： 将控制输入u_0作用于系统(5.1)获得y_0，并计算跟踪误差e_0；通过迭代学习容错控制更新律式(5.23)得到控制输入u_1。

3： **repeat**

4： **repeat**

5： 执行算法 5-2 以获得估计故障$\tilde{\Gamma}_{k+1}(i)$。

6： 设置$i=i+1$。

7： **until** $i=N-1$

8： 将$\tilde{\Gamma}_{k+1}(t)$，$t\in[0,\ N]$组成$\tilde{\Gamma}_{k+1}$，根据u_k，e_k $\tilde{\Gamma}_k$和$\tilde{\Gamma}_{k+1}$通过迭代学习容错控制更新律(5.23)得到下一批次的控制输入u_{k+1}。

9： 将控制输入u_{k+1}作用于系统，获得y_{k+1}并计算得到e_{k+1}。

10： 设置$k=k+1$。

11： **until** $k=k_{\max}$

算法 5-1 中包含的算法 5-2 基于 Q 学习的故障估计算法过程如下所示。

算法 5-2 基于 Q 学习的故障估计算法

Input：状态空间 $\mathcal{S}$，其中状态 $s \in \mathcal{S}$ 并且 $s = \left[\tilde{\Gamma}_{1,k}(t), \tilde{\Gamma}_{2,k}(t),, \cdots, \tilde{\Gamma}_{m,k}(t)\right]$；动作空间 $\mathcal{A}$，其中动作 $a \in \mathcal{A}$ 并且 $a = \left[\Delta\tilde{\Gamma}_{1,k}(t),\ \Delta\tilde{\Gamma}_{2,k}(t),\ \cdots,\ \Delta\tilde{\Gamma}_{m,k}(t)\right]$；学习率 α；折扣因子 γ；贪心概率 ϵ；损失函数阈值 $\varepsilon_{\mathcal{L}}$；状态 x_k 和输入 u_k。

Output：估计故障矩阵 $\tilde{\Gamma}_k(t)$。

1：**Initialization**：初始化状态动作值函数和初始状态 s_0。

2：通过 ϵ-贪心算法在初始状态 s_0 选择动作 a_0。

3：**repeat**

4：　执行动作 a，得到对应的奖赏 $\mathcal{R}$，跳转到状态 s'。

5：　通过 ϵ-贪心算法在初始状态 s' 选择动作 a'。

6：　通过式(5.36)更新状态动作值函数 $Q^{\pi}(s,a)$。

7：　设置 $s \leftarrow s'$，$a \leftarrow a'$。

8：**until** $\mathcal{L} \le \varepsilon_{\mathcal{L}}$

此外，具体的迭代学习容错控制的流程在图 5-1 中进行了说明。

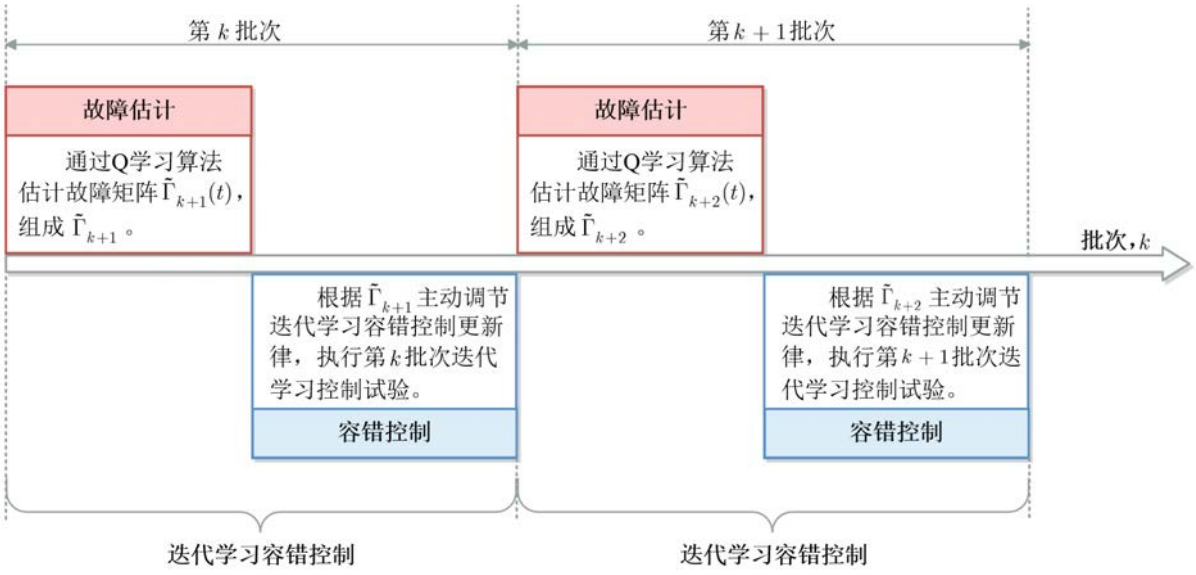

图 5-1 故障估计与迭代学习容错控制流程图

如图 5-1 所示，在第 k 次批次，首先进行故障估计，即通过 Q 学习算法估计在采样时间 $t \in [0, N-1]$ 内的估计故障矩阵 $\tilde{\Gamma}_{k+1}(t)$，以组成 $\tilde{\Gamma}_{k+1}$。接着进行容错控制，即使用估计的故障矩阵 $\tilde{\Gamma}_{k+1}$ 调节迭代学习容错控制器。通过控制器的调节，故障的负面影响可以一定程度上被抵消，从而保持系统在执行器故障下的良好性能。

注释 5.3 在每个迭代批次中，Q 学习算法的计算复杂度为 $O(N_Q N)$，其中 N_Q 是 Q 学习算法的迭代次数，并且不随着采样点个数 N 增加而增加。迭代学习容错控制算法的计算复杂度为 $O(N^3)$，由此可以看出，采样点个数 N 是影响所提迭代学习容错控制算法计算复杂度的主要因素，随着 N 的增加，迭代学习容错控制算法中控制增益矩阵的维度

将增加，从而增加计算复杂度。有关提升范数优化框架下的迭代学习控制计算复杂度的更多讨论，请参考文献[63]。

5.3.4 收敛性分析

本小节主要对迭代学习容错控制算法式(5.23)的收敛性进行了分析，Q 学习算法的收敛性分析可参考文献[74]。由于故障和估计的故障都是不确定的，并且它们都有上下界，因此，将实际故障矩阵和估计故障矩阵视为随着批次k变化的有界矩阵，给出了迭代学习容错控制更新律(5.23)收敛条件和证明，证明过程中将使用以下引理。

引理 5.1[87] 对于任意给定矩阵$A\in\mathbb{R}^{m\times n}$，满足

$$\rho\big(A\big)<1, \tag{5.37}$$

其中$\rho\big(A\big)$是矩阵A的谱半径，那么，至少存在一种矩阵的范数使得

$$\lim_{k\to\infty}\big\|A\big\|_S^k=0. \tag{5.38}$$

定理 5.2 对于执行器故障下系统(5.1)使用迭代学习容错控制更新律(5.23)，若条件

$$\Big\|\Gamma_{k+1}L_{k+1}^u\Gamma_k^\dagger-\Gamma_{k+1}L_{k+1}^eG\Big\|\le\rho<1, \tag{5.39}$$

满足，则当迭代批次$k\to\infty$时，跟踪误差的范数有界收敛，即

$$\lim_{k\to\infty}\big\|e_{k+1}\big\|\le\frac{b_uc}{1-\rho}, \tag{5.40}$$

其中，正常数$b_u=b\big\|u_d\big\|$，$c=\big\|G\big\|$。同时，b是一个正标量，定义为

$$\Big\|I-\tilde{\Gamma}_{k+1}L_{k+1}^u\tilde{\Gamma}_k^\dagger\Big\|\leqslant b<\frac{\overline{\Gamma}_{\max}}{\underline{\Gamma}_{\min}}\Big(\overline{\Gamma}_{\max}^2\big\|R^{-1}\big\|\big\|G^{\mathrm{T}}\big\|\big\|Q\big\|\big\|G\big\|+1\Big)+1.$$

证明 根据式(2.7)和式(5.17)，第k批次的跟踪误差可以改写为

$$\begin{aligned}e_k&=y_d-y_k\\&=Gu_d+d-G\Gamma_ku_k-d\\&=G(u_d-\Gamma_ku_k).\end{aligned} \tag{5.41}$$

定义输入误差δu_k为

$$\delta u_k=u_d-\Gamma_ku_k. \tag{5.42}$$

则式(5.41)可以改写为

$$e_k=G\delta u_k. \tag{5.43}$$

根据式(5.42)和迭代学习容错控制更新律(5.23)，可得第$k+1$批次的输入误差δu_{k+1}为

$$\begin{aligned}\delta u_{k+1}&=u_d-\Gamma_{k+1}u_{k+1}\\&=u_d-\Gamma_{k+1}L_{k+1}^uu_k-\Gamma_{k+1}L_{k+1}^ee_k\\&=u_d-\Gamma_{k+1}L_{k+1}^u(\Gamma_k^\dagger u_d-\Gamma_k^\dagger\delta u_k)-\Gamma_{k+1}L_{k+1}^eG\delta u_k\\&=\Big[\Gamma_{k+1}L_{k+1}^u\Gamma_k^\dagger-\Gamma_{k+1}L_{k+1}^eG\Big]\delta u_k+\Big[I-\Gamma_{k+1}L_{k+1}^u\Gamma_k^\dagger\Big]u_d.\end{aligned} \tag{5.44}$$

对等式两边取范数，可得

$$\left\|\delta u_{k+1}\right\| \leq \left\|\Gamma_{k+1}L_{k+1}^{u}\Gamma_{k}^{\dagger} - \Gamma_{k+1}L_{k+1}^{e}G\right\|\left\|\delta u_{k}\right\| + \left\|I - \Gamma_{k+1}L_{k+1}^{u}\Gamma_{k}^{\dagger}\right\|\left\|u_{d}\right\|. \tag{5.45}$$

接下来，证明$\left\|I - \Gamma_{k+1}L_{k+1}^{u}\Gamma_{k}^{\dagger}\right\|$存在上界。根据范数的相容性和三角不等式，可得

$$\left\|I - \Gamma_{k+1}L_{k+1}^{u}\Gamma_{k}^{\dagger}\right\| \leq \left\|\Gamma_{k+1}\right\|\left\|L_{k+1}^{u}\right\|\left\|\Gamma_{k}^{\dagger}\right\| + 1. \tag{5.46}$$

由于Γ_{k}和$\tilde{\Gamma}_{k}$均为对角阵，因此，$\left\|\Gamma_{k}\right\|$，$\left\|\Gamma_{k}^{\dagger}\right\|$和$\left\|\tilde{\Gamma}_{k}\right\|$的上界可表示为

$$\left\|\Gamma_{k}\right\| \leq \overline{\Gamma}_{\max},\ \left\|\Gamma_{k}^{\dagger}\right\| \leq \frac{1}{\underline{\Gamma}_{\min}},\ \left\|\tilde{\Gamma}_{k}\right\| \leq \overline{\Gamma}_{\max}. \tag{5.47}$$

根据式(5.24)，可得

$$\left\|L_{k+1}^{u}\right\| = \left\|\left(R^{-1}\tilde{\Gamma}_{k+1}^{\mathrm{T}}G^{\mathrm{T}}QG\tilde{\Gamma}_{k+1} + I\right)^{-1}\left(R^{-1}\tilde{\Gamma}_{k+1}^{\mathrm{T}}G^{\mathrm{T}}QG\tilde{\Gamma}_{k} + I\right)\right\|. \tag{5.48}$$

由于矩阵R^{-1}和Q均为正定对称权重矩阵，$\left(R^{-1}\tilde{\Gamma}_{k+1}^{\mathrm{T}}G^{\mathrm{T}}QG\tilde{\Gamma}_{k+1}\right)$是对称的非负定矩阵。因此，有$\left\|\left(R^{-1}\tilde{\Gamma}_{k+1}^{\mathrm{T}}G^{\mathrm{T}}QG\tilde{\Gamma}_{k+1} + I\right)^{-1}\right\| < 1$。故根据式(5.47)和式(5.48)，可得

$$\begin{aligned}\left\|L_{k+1}^{u}\right\| &< u_{d} - \Gamma_{k+1}u_{k+1}\\ &< \left\|R^{-1}\right\|\left\|\tilde{\Gamma}_{k+1}^{\mathrm{T}}\right\|\left\|G^{\mathrm{T}}\right\|\left\|Q\right\|\left\|G\right\|\left\|\tilde{\Gamma}_{k}\right\| + 1\\ &< \overline{\Gamma}_{\max}^{2}\left\|R^{-1}\right\|\left\|G^{\mathrm{T}}\right\|\left\|Q\right\|\left\|G\right\| + 1.\end{aligned} \tag{5.49}$$

将式(5.47)代入式(5.46)可得

$$\left\|I - \tilde{\Gamma}_{k+1}L_{k+1}^{u}\tilde{\Gamma}_{k}^{\dagger}\right\| < \frac{\overline{\Gamma}_{\max}}{\underline{\Gamma}_{\min}}\left(\overline{\Gamma}_{\max}^{2}\left\|R^{-1}\right\|\left\|G^{\mathrm{T}}\right\|\left\|Q\right\|\left\|G\right\| + 1\right) + 1. \tag{5.50}$$

接下来，定义一个正标量b，且满足

$$\left\|I - \tilde{\Gamma}_{k+1}L_{k+1}^{u}\tilde{\Gamma}_{k}^{\dagger}\right\| \leqslant b < \frac{\overline{\Gamma}_{\max}}{\underline{\Gamma}_{\min}}\left(\overline{\Gamma}_{\max}^{2}\left\|R^{-1}\right\|\left\|G^{\mathrm{T}}\right\|\left\|Q\right\|\left\|G\right\| + 1\right) + 1. \tag{5.51}$$

根据式(5.51)，式(5.45)可改写为

$$\left\|\delta u_{k+1}\right\| \leqslant \left\|\Gamma_{k+1}L_{k+1}^{u}\Gamma_{k}^{\dagger} - \Gamma_{k+1}L_{k+1}^{e}G\right\|\left\|\delta u_{k}\right\| + b\left\|u_{d}\right\|, \tag{5.52}$$

其中，定义$b_{u} = b\left\|u_{d}\right\|$。在$k$个迭代批次后，可得

$$\begin{aligned}\left\|\delta u_{k+1}\right\| &< \left\|\Gamma_{k+1}L_{k+1}^{u}\Gamma_{k}^{\dagger} - \Gamma_{k+1}L_{k+1}^{e}G\right\|^{k}\left\|\delta u_{0}\right\|\\ &+ \frac{1 - \left\|\Gamma_{k+1}L_{k+1}^{u}\Gamma_{k}^{\dagger} - \Gamma_{k+1}L_{k+1}^{e}G\right\|^{k}}{1 - \left\|\Gamma_{k+1}L_{k+1}^{u}\Gamma_{k}^{\dagger} - \Gamma_{k+1}L_{k+1}^{e}G\right\|}b_{u}.\end{aligned} \tag{5.53}$$

根据引理 5.1，如果条件(5.39)满足，那么可得$\lim\limits_{k\to\infty}\left\|\Gamma_{k+1}L_{k+1}^{u}\Gamma_{k}^{\dagger} - \Gamma_{k+1}L_{k+1}^{e}G\right\|^{k} = 0$。当$k \to \infty$时，跟踪误差的范数可以表示为

$$\lim_{k\to\infty}\left\|\delta u_{k+1}\right\| \leqslant \frac{b_{u}}{1-\rho}. \tag{5.54}$$

根据式(5.43)和式(5.54)可得

$$\begin{aligned}\lim_{k\to\infty}\left\|e_{k+1}\right\| &= \lim_{k\to\infty}\left\|G\right\|\left\|\delta u_{k+1}\right\| \\ &\leqslant \frac{b_u\left\|G\right\|}{1-\rho}.\end{aligned} \tag{5.55}$$

最后，令$c=\left\|G\right\|$，可得

$$\lim_{k\to\infty}\left\|e_{k+1}\right\|\leqslant\frac{b_u c}{1-\rho}. \tag{5.56}$$

由此，跟踪误差的范数收敛于一个有界值，证毕。 ■

如果条件(5.39)满足，那么跟踪误差的范数可以有界收敛，因此，满足第 5.2 节中所提的定义 5.1 迭代学习容错控制算法设计问题。

注释 5.4 由于随时间和批次同时变化的执行器故障的影响，跟踪误差在最初的几个迭代批次不能单调收敛，而是在有界范围内随着故障的变化而振荡，这一点可以通过第 5.4 节的仿真直观地观察到。

注释 5.5 本章所提的故障估计与容错控制算法主要考虑在重复控制操作期间发生的部分执行器故障。当执行器完全故障或过载时时，意味着至少有一个执行器完全失效或过度控制，在这种情况下系统可能是不可控的，尤其是对于解耦系统。因此，本章主要关注的情况为部分执行器故障，完全故障或过载的情况将在未来的研究工作中考虑。

5.4 仿真实例

为了验证所提故障估计与迭代学习容错控制算法的有效性，本节考虑双轮独立的移动机器人系统[88]作为仿真对象。将多输入多输出系统进行解耦，然后将其离散化，以通过驱动电压u_v和u_ϕ分别控制线速度v和方位角ϕ，从而使移动机器人在绝对坐标系$O-XY$中进行刚性运动。

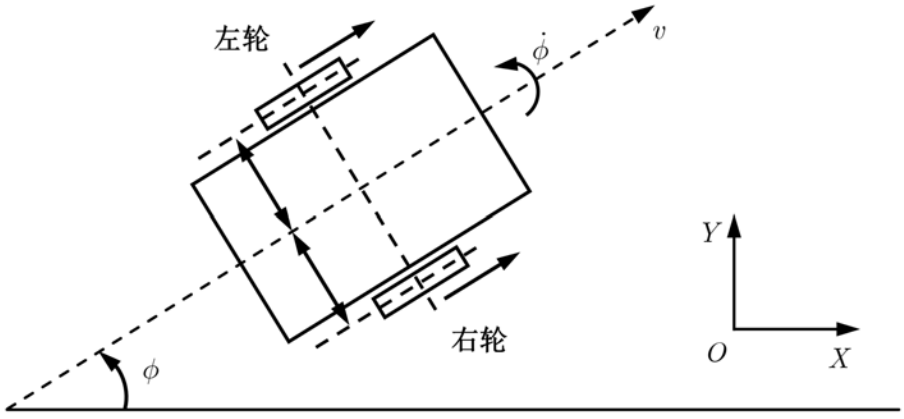

图 5-2 双轮独立移动机器人示意图

5.4.1 控制任务描述

双轮独立的双输入双输出移动机器人系统的示意图如图 5-2 所示，定义状态变量为 $x=\begin{bmatrix} v & \phi & \dot{\phi} \end{bmatrix}^{\mathrm{T}}$，控制输入为 $u=\begin{bmatrix} u_r & u_l \end{bmatrix}^{\mathrm{T}}$，输出变量为 $y=\begin{bmatrix} v & \phi \end{bmatrix}^{\mathrm{T}}$，则状态空间模型矩阵可以表示为

$$A=\begin{bmatrix} a_1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & a_2 \end{bmatrix},\ B=\begin{bmatrix} b_1 & b_1 \\ 0 & 0 \\ b_2 & -b_2 \end{bmatrix},\ C=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \tag{5.57}$$

其中，a_1，a_2，b_1 和 b_2 分别定义为

$$a_1=-\frac{2c_m}{Mr^2+2I_w},\quad a_2=-\frac{2c_m l_m^2}{I_v r^2+2I_w l_m^2},$$
$$b_1=\frac{k_m r}{Mr^2+2I_w},\quad b_2=\frac{k_m r l_m}{I_v r^2+2I_w l_m^2}.$$

具体的系统参数定义和取值详见表 5-1。

表 5-1 双轮独立的移动机器人系统参数的定义与取值

变量	定义	值
I_v	移动机器人车轮的转动惯量	$10\ \mathrm{kgm}^2$
M	移动机器人的质量	$200\ \mathrm{kg}$
l_m	移动机器人的左轮或右轮到移动机器人重心的距离	$0.3\ \mathrm{m}$
I_w	围绕移动机器人重心的转动惯量	$0.005\ \mathrm{kgm}^2$
c_m	粘性摩擦系数	$0.05\ \mathrm{kgm}^2/\mathrm{s}$
r	移动机器人的车轮半径	$0.1\ \mathrm{m}$
k_m	驱动增益因子	5

式(5.57)的移动机器人状态空间是一个线性耦合模型，接下来将系统解耦，以更好地观察控制效果，使用如下的解耦矩阵[89]

$$\begin{bmatrix} u_r \\ u_l \end{bmatrix}=\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} u_v \\ u_\phi \end{bmatrix},$$

其中，解耦后的控制输入变量定义为 $\widehat{u}=\begin{bmatrix} u_v & u_\phi \end{bmatrix}^{\mathrm{T}}$，并且 u_v 是直接控制机器人线速度的驱动电压，而 u_ϕ 也是直接控制机器人方位角的驱动电压。

在离散化过程中，考虑使用零阶保持器，并将采样时间 T_s 设置为 $0.05\,\mathrm{s}$。与此同时，将一个批次的时间 T 设置为 $2\,\mathrm{s}$，即每次迭代过程中的采样点个数 $N=40$。由此，解耦后的离散状态空间模型可以表示为

$$\begin{cases} x_k(t+1) = \widehat{A}x_k(t) + \widehat{B}\widehat{u}_k(t), \\ y_k(t) = \widehat{C}x_k(t), \end{cases}$$

其中，状态空间模型的矩阵为

$$\widehat{A} = \begin{bmatrix} 0.9975 & 0 & 0 \\ 0 & 1 & 0.0499 \\ 0 & 0 & 0.9956 \end{bmatrix},\ \widehat{B} = \begin{bmatrix} 0.0248 & 0 \\ 0 & 0.0037 \\ 0 & 0.1483 \end{bmatrix},\ \widehat{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

迭代学习控制的控制任务是使得移动机器人的系统输出 $y = \begin{bmatrix} v & \phi \end{bmatrix}^{\mathrm{T}}$ 跟踪上期望的线速度和方位角，定义为

$$v_d = 2\ \mathrm{m/s},\ \phi_d = \pi t\ \mathrm{rad},$$

即移动机器人的期望运动轨迹是一个圆。

在具有重复性质的控制任务的执行期间，部分退化和磨损可能导致执行器的部分失效故障[90]。因此，在仿真中考虑如下执行器故障，其中执行器的故障矩阵表示为：

$$\Gamma_k(t) = \begin{bmatrix} \Gamma_{1,k} & \\ & \Gamma_{2,k}(t) \end{bmatrix},$$

其中，矩阵元素定义为

$$\begin{aligned} \Gamma_{1,k} &= 0.15\sin\left(\pi k/10 - \pi/2\right) + 0.7, \\ \Gamma_{2,k}(t) &= 0.1\sin\left(\pi k/8 - \pi/2\right) + 0.75 + 0.1\sin\left(2\pi t\right),\ t \in \left[0, N-1\right], \end{aligned}$$

并且矩阵元素有上下界

$$0.55 \le \underline{\Gamma}_i \le \Gamma_{i,k}(t) \le \overline{\Gamma}_i \le 0.95,\ i = 1,2.$$

为了更好地观察算法 5.1 基于 Q 学习的故障估计算法在沿时间轴和沿迭代轴方向的性能，仿真中设置第1个执行器的故障矩阵 $\Gamma_{1,k}$ 仅随迭代批次变化，第2个执行器的故障矩阵 $\Gamma_{2,k}(t)$ 随时间和迭代批次同时变化。关于算法 5.2 基于 Q 学习的故障估计算法，设置学习率 $\alpha = 0.1$，折扣因子 $\gamma = 1$，贪心概率 $\epsilon = 0.1$，损失函数阈值 $\varepsilon_{\mathcal{L}} = 10^{-11}$。

5.4.2 仿真结果

（1）基于范数优化的迭代学习容错控制算法

将所提算法应用于第 5.4.1 节描述的控制任务，进行 30 次试验（$k_{\max} = 30$），设置权重矩阵分别为 $Q = I$ 和 $R = 0.001I$。图 5-3 和图 5-4 分别展示了移动机器人系统在所提算法 5.1 和 5.2 作用下的线速度和方位角的期望参考信号，以及前几个批次与最后一个批次的线速度和方位角输出信号曲线。图 5-5 展示了移动机器人在绝对坐标系 $O-XY$ 的期望运动轨迹，以及前几个批次与最后一个批次的运动轨迹。从图 5-3，图 5-4 和图 5-5 可以看出，输出信号和运动轨迹在前几个批次迅速跟踪上参考信号和参考运动轨迹，并且在最后一个批次基本跟踪上期望参考信号和参考运动轨迹，基本跟踪上参考信号和参考运动轨迹是因为随着迭代批次的增加，跟踪误差可以逐渐达到有界收

敛，但由于随批次和时间同时变化故障的影响，跟踪误差不能收敛到0，这与第 5.2 节中所提的定义 5.1 迭代学习容错控制算法设计问题描述一致。图 5-6 和图 5-7 展示了对应 u_v 和 u_ϕ 的输出信号曲线。可以看出，输入信号会逐渐被修正以驱使输出信号完成控制任务目标。

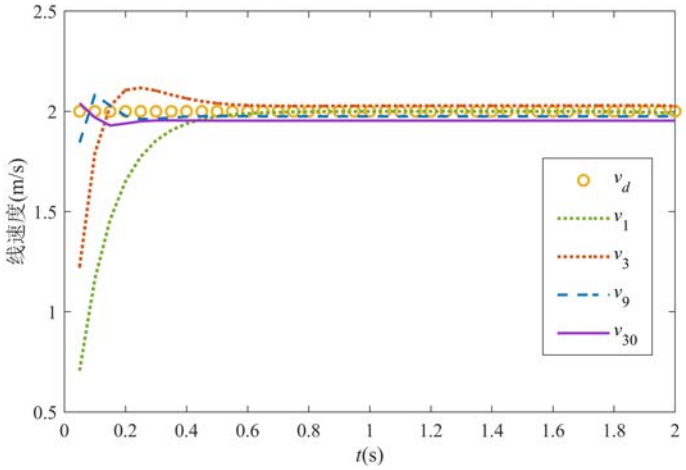

图 5-3 移动机器人在前几个批次和最后批次的线速度输出

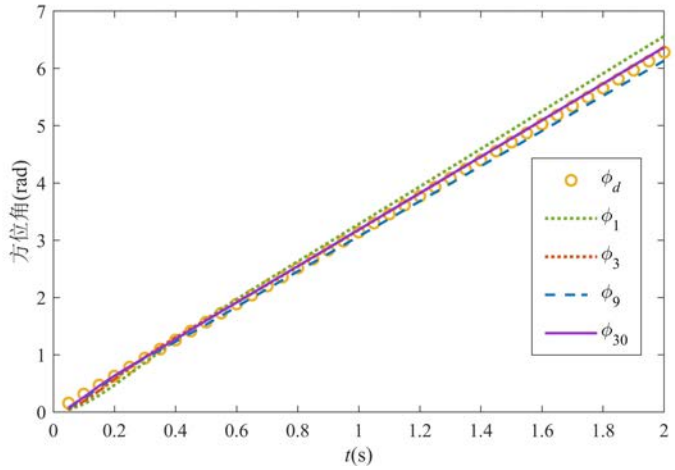

图 5-4 移动机器人在前几个批次和最后批次的方位角输出

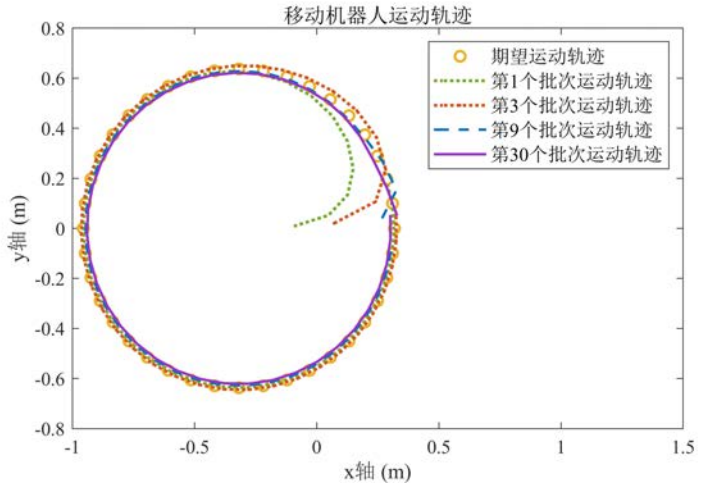

图 5-5 移动机器人在前几个批次和最后批次的运动轨迹

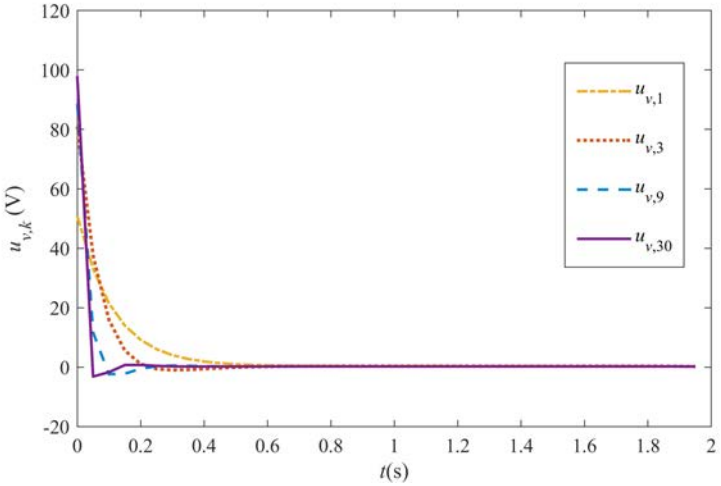

图 5-6 移动机器人在前几个批次和最后批次的线速度输入

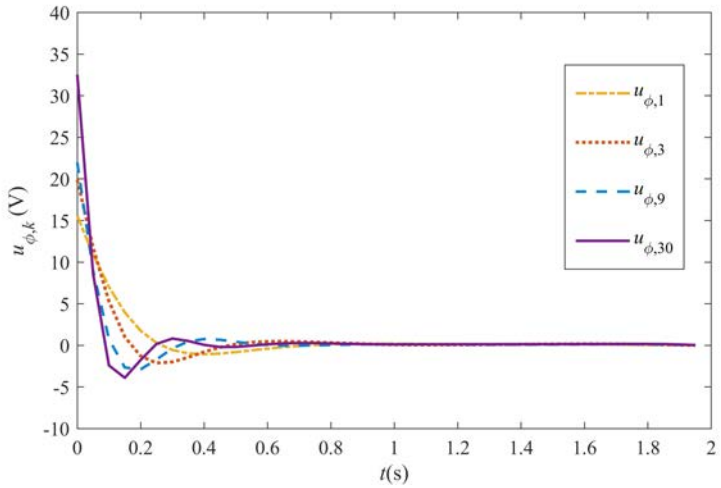

图 5-7 移动机器人在前几个批次和最后批次的方位角输入

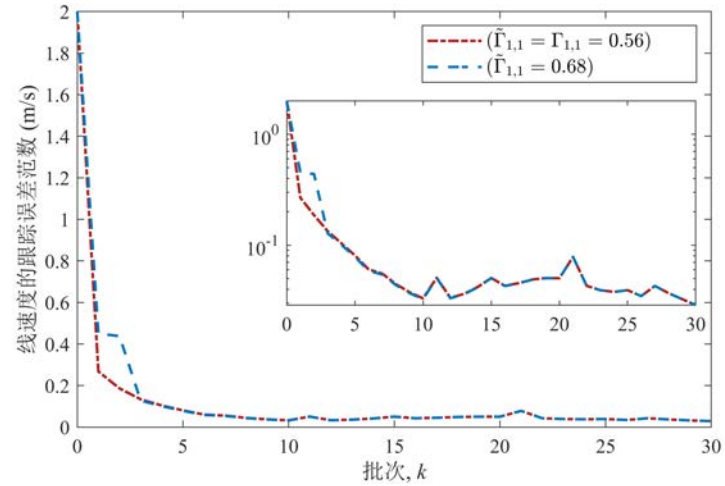

图 5-8 线速度的跟踪误差的范数收敛图

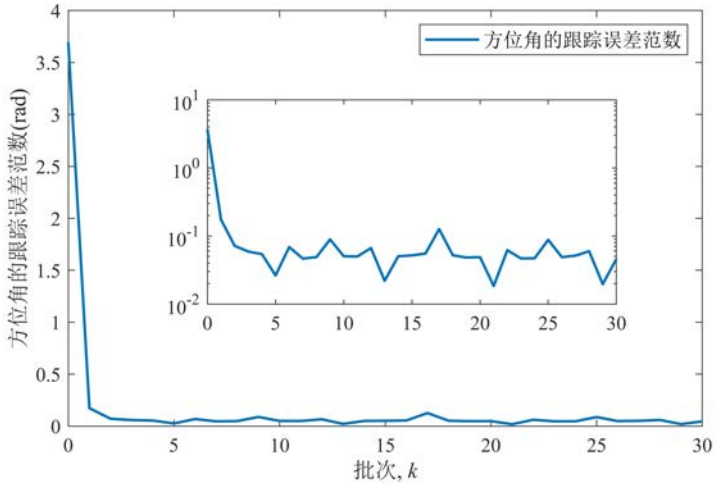

图 5-9 方位角的跟踪误差的范数收敛图

图 5-8 和图 5-9 展示了线速度和方向角的跟踪误差的范数在标准坐标系和对数坐标系下的收敛图，可以看出，线速度和方位角的跟踪误差随着迭代批次的增加逐渐减小，而后在一个有界范围内相对平稳地振荡，验证了注释 5.4 的描述。图 5-10 展示了权重矩阵 Q 和 R 的不同选择下的跟踪误差的范数收敛图，可以看出，增加或减小 Q 和 R 的值可以增加跟踪误差收敛速度并且提高控制性能，证实了注释 5.1 的描述。

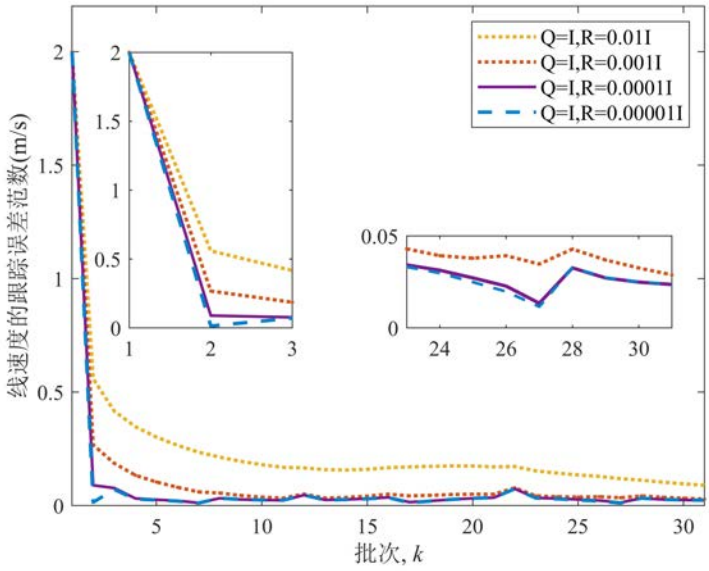

图 5-10 不同 Q 和 R 选择下线速度的跟踪误差范数收敛图

图 5-11 和图 5-12 描述了估计故障矩阵 $\tilde{\Gamma}_{1,k}$ 和 $\tilde{\Gamma}_{2,k}(t)$ 的估计误差的范数随批次变化图，可以看出估计误差在较小的范围内振荡，这说明了基于 Q 学习的故障估计算法的有效性。图 5-13 和图 5-14 展示了具体的故障估计过程。从图 5-13 可以看出，估计过程中存在一个迭代批次的延迟，这与第 5.3.2 节的注释 5.2 的描述一致。图 5-14 说明在每个采样时间，当前批次的估计故障值与前一批次的实际故障一致，这个延迟的产生是迭代学习控制的更新过程产生的，即仅能从前一批次提供的信息来估计当前批次的故障。这个延迟也导致了图 5-11 和图 5-12 中的估计误差。

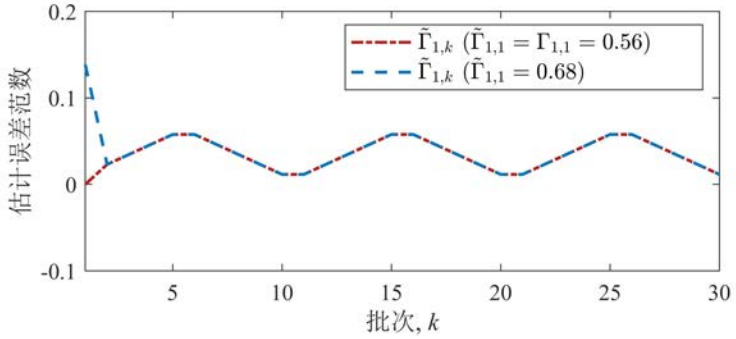

图 5-11 $\tilde{\Gamma}_{1,k}$ 的估计误差的范数

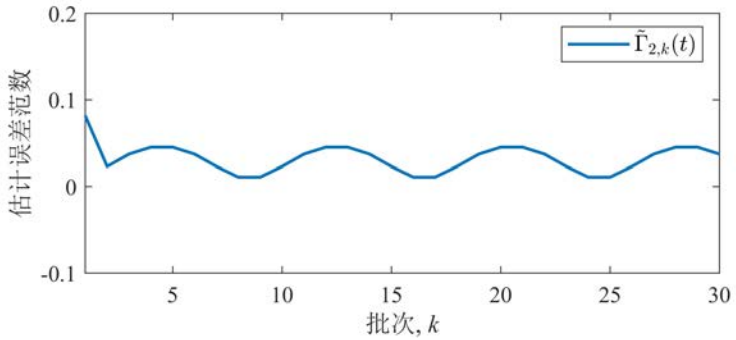

图 5-12 $\tilde{\Gamma}_{2,k}(t)$ 的估计误差的范数

从图 5-13 和图 5-14 可以看出，初始批次的 $\tilde{\Gamma}_1$ 是人为选择的，原因是为了保证控制系统在初始迭代批次中的稳定性，迭代学习控制的初始信号通常被设置为 0，即 $u_0 = 0$，因此，基于 Q 学习的故障估计没有可以参考的信息用于估计。但是从图 5-11 和图 5-12 可以看出，从 $\tilde{\Gamma}_2$ 开始，故障估计的结果不会被人为选择的 $\tilde{\Gamma}_1$ 影响。这是因为从 $\tilde{\Gamma}_2$ 开始，迭代学习控制器开始有输入信号并且可以提供足够的信息用于故障估计。从图 5-8 和图 5-9 可以看出，人为选择的估计故障矩阵 $\tilde{\Gamma}_1$ 和实际故障矩阵 Γ_1 之间的差值会影响前几个批次的跟踪误差收敛速度，但随着迭代学习容错控制算法开始引入相对准确的故障估计信息来修正控制输入，所提迭代学习容错控制算法会很快克服人为选择的 $\tilde{\Gamma}_1$ 的影响。

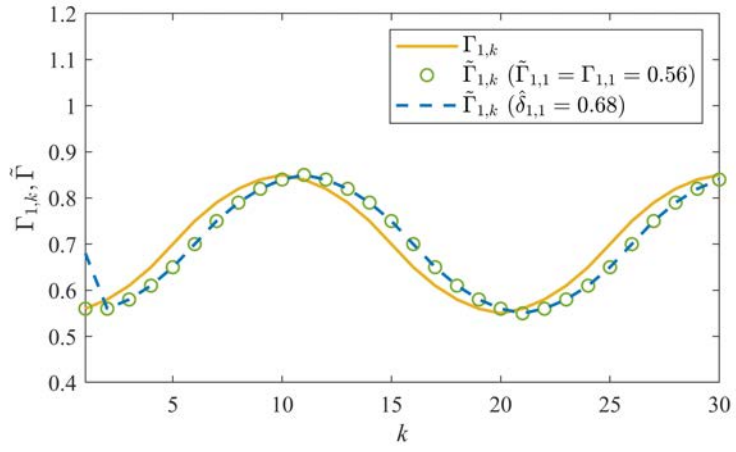

图 5-13 实际故障 $\Gamma_{1,k}$ 和估计故障 $\tilde{\Gamma}_{1,k}$ 对比图

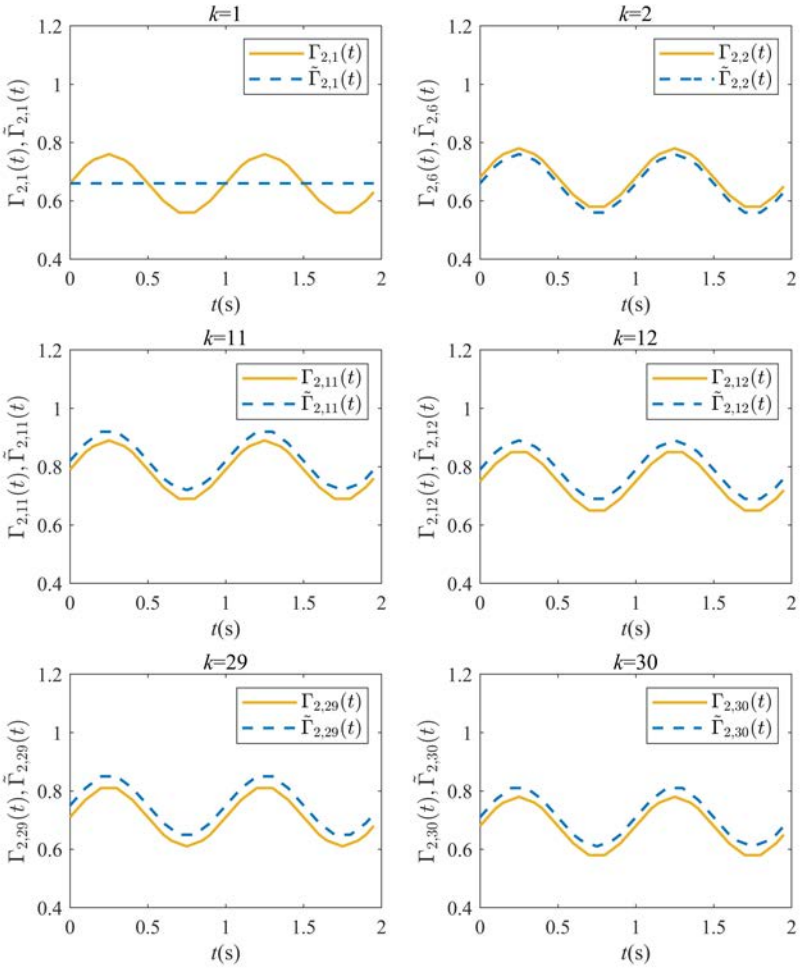

图 5-14 实际故障 $\Gamma_{2,k}(t)$ 和估计故障 $\tilde{\Gamma}_{2,k}(t)$ 对比图

（2）算法对比

为了更好地验证所提算法 5.1 和算法 5.2 的有效性和优势，选择已有的容错控制算法和故障估计算法作对比。选取文献[83]中的传统可靠控制（Traditional Reliable Control）算法，其中，控制增益设置为

$$K_{11}=\begin{bmatrix}-85.4117 & 0 & 0\\ 0 & -70.2263 & -10.1736\end{bmatrix},\ K_{12}=\begin{bmatrix}33.5987 & 0\\ 0 & 9.2374\end{bmatrix}.$$

根据文献[91]，乘性故障可以转化为加性故障，即 $\tilde{f}_k(t)=(\tilde{\Gamma}_k(t)-I)u_k(t)$，因此选取文献[80]中基于观测器的故障估计算法

$$\begin{cases}\tilde{x}_k(t+1)=A\tilde{x}_k(t)+Bu_k(t)+B\tilde{f}_k(t)+L(y_k(t)-\tilde{y}_k(t)),\\ \tilde{y}_k(t)=C\tilde{x}_k(t),\\ \tilde{f}_k(t+1)=\tilde{f}_k(t)+F(y_k(t)-\tilde{y}_k(t)),\end{cases}$$

其中，观测器增益设置为

$$L=\begin{bmatrix}-1.0615 & -0.0008\\ -0.0004 & -1.1630\\ -0.0079 & -3.4223\end{bmatrix},\ F=\begin{bmatrix}-2.5798 & -0.0315\\ 0.0227 & -2.2791\end{bmatrix}.$$

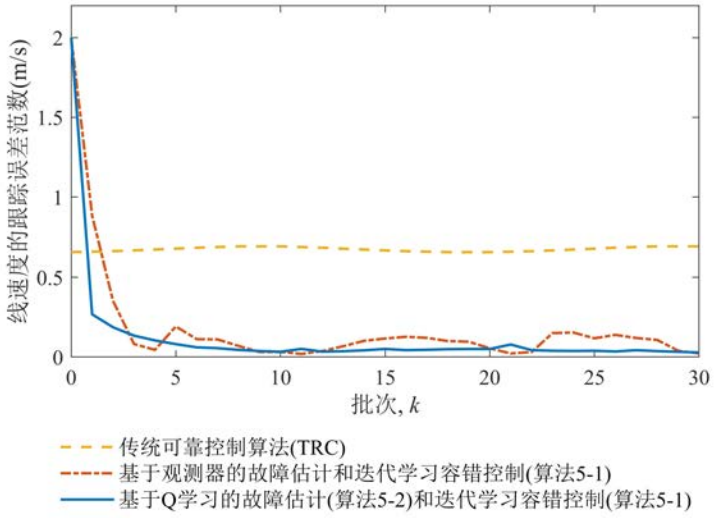

图 5-15 不同方法下的线速度跟踪误差的范数对比图

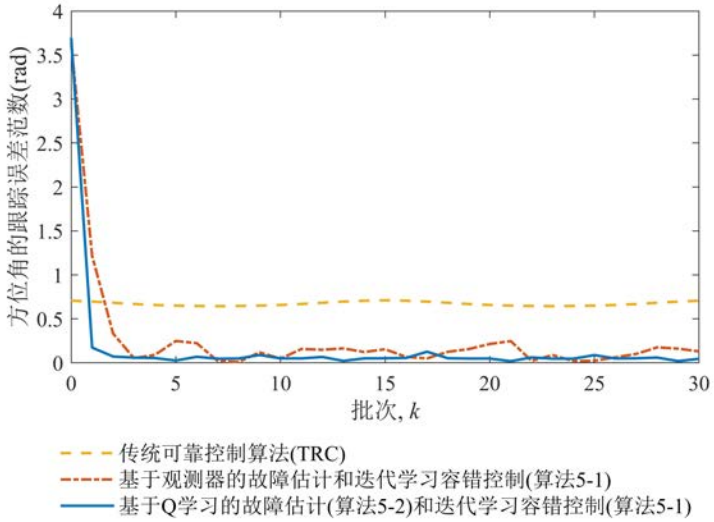

图 5-16 不同方法下的方位角跟踪误差的范数对比图

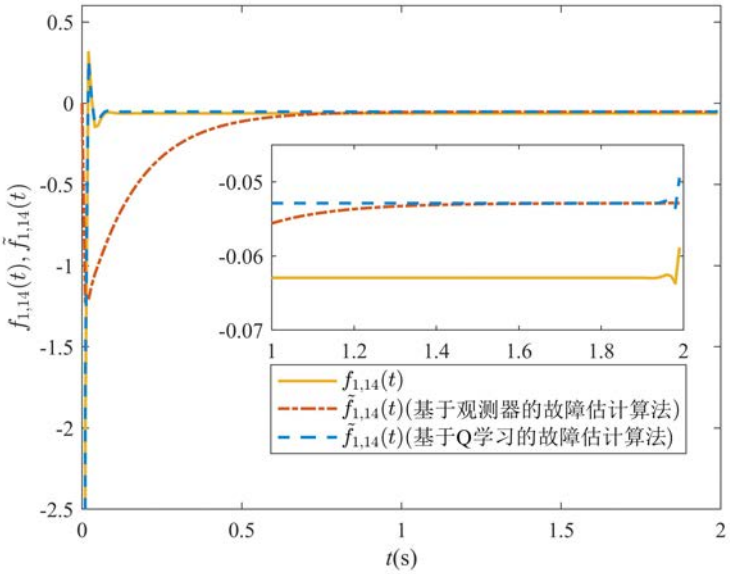

图 5-17 估计故障 $\tilde{f}_{1,14}(t)$ 和实际故障 $f_{1,14}(t)$ 的对比

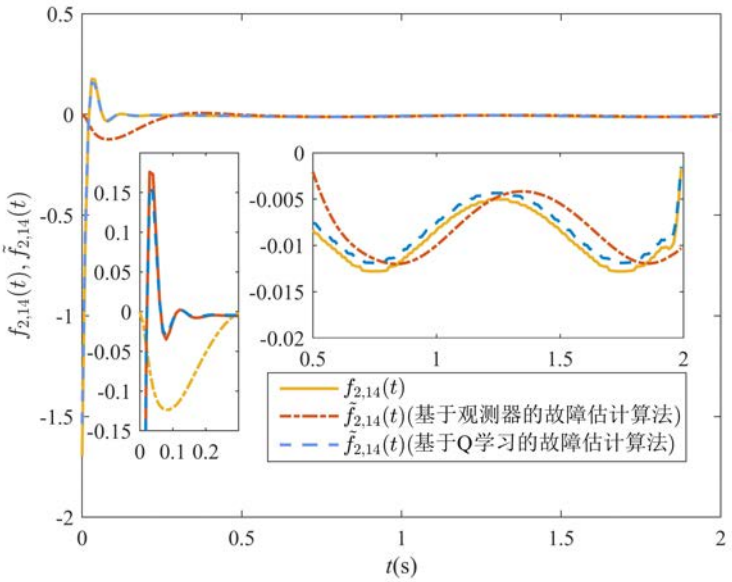

图 5-18　估计故障 $\tilde{f}_{2,14}(t)$ 和实际故障 $f_{2,14}(t)$ 的对比

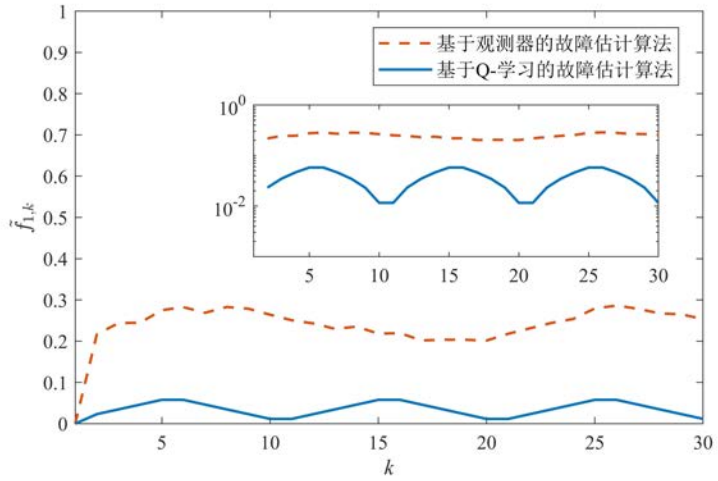

图 5-19　不同故障估计算法的 $\tilde{f}_{1,k}$ 估计误差的范数

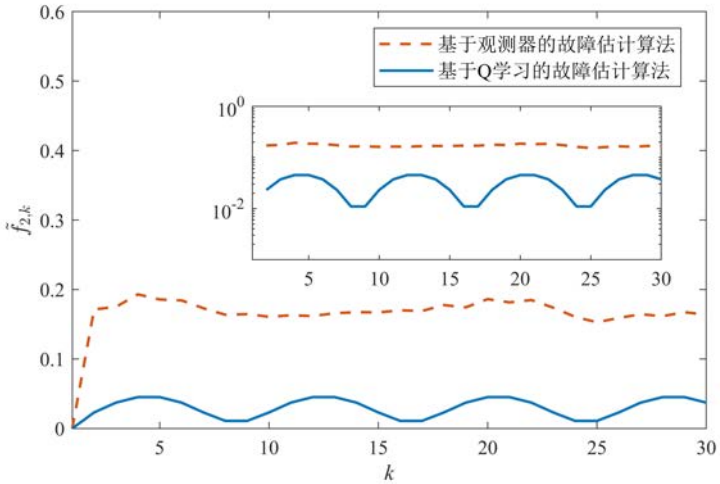

图 5-20　不同故障估计算法的 $\tilde{f}_{2,k}$ 估计误差的范数

将基于观测器的故障估计算法应用于本章所提迭代学习容错控制算法，以进一步和本章所提基于 Q 学习的故障估计算法在估计性能上作对比。图 5-15 和图 5-16 分别展示了不同方法下的线速度和方位角的跟踪误差收敛对比图。可以看出，在随时间和批次同时变化的执行器故障下，本章所提迭代学习容错控制算法有着更好的控制性能。原因在于，传统可靠控制算法保障了沿着时间轴的控制性能，而本章所提迭代学习容错控制算法同时保障了沿着时间轴和批次轴的控制性能。并且，传统可靠控制算法的控制器结构是不变的，而本章所提迭代学习容错控制器会根据故障估计的结果动态调节。与此同时，可以看出使用本章所提基于 Q 学习的故障估计结果的迭代学习容错控制算法相比使用基于观测器的故障估计结果的迭代学习容错控制算法，在收敛速度和收敛精度方面有着更好的控制性能。这是因为基于 Q 学习的故障估计算法在随着时间和批次同时变化的执行器故障下有着更好的估计性能，这一点从图 5-17 至图 5-20 也可以看出。相比于有着固定结构的基于观测器的故障估计算法，基于 Q 学习的故障估计算法在应对随时间和批次变化的故障方面，有着更好的动态特性。综上，本章所提算法的有效性得以验证。

5.5 小结

本章研究了一类随时间和批次同时变化的执行器故障下线性离散系统的跟踪问题，提出了一种基于 Q 学习的故障估计和基于范数优化的迭代学习容错控制方案。为了减少沿着时间和批次同时变化的未知故障对迭代学习控制器跟踪性能的影响，引入了 Q 学习算法，通过不断地调整故障估计器以适应变化的故障。并且，在范数优化框架下设计了多目标性能指标下的迭代学习容错控制器，通过 Q 学习的故障估计结果来调整控制器，以抵消故障的影响，给出了算法的流程描述，提出了系统在该学习律下跟踪误差有界收敛的条件并给出了相应证明。最后，通过移动机器人的仿真，验证了所提方法在故障估计与跟踪性能方面的有效性和优势。

第六章　结论与展望

6.1 结论

本文开展了对基于强化学习的迭代学习控制与优化方法研究，进行了一些探索性工作，其主要研究工作总结如下：

（1）针对一类线性离散系统的跟踪问题，在非提升范数优化框架下，提出了一种基于值迭代的迭代学习控制方法。将迭代学习控制过程描述为马尔可夫决策过程，引入强化学习的未来收益指导当下动作的思想。通过最小化状态值函数得到迭代学习控制更新律，提出了系统在该学习律下渐近稳定和跟踪误差单调收敛的条件并给出了相应证明。进一步地，分析了所提方法在计算复杂度方面的优势。最后，通过直流电动机的仿真，验证了所提方法的有效性和优势。

（2）针对一类模型信息未知的线性离散系统的跟踪问题，在非提升范数优化框架下，提出了一种基于 Q 学习的无模型迭代学习控制方法。将迭代学习控制过程描述为马尔可夫决策过程。引入 Q 学习算法，通过最小化 Q 函数得到包含模型信息的迭代学习控制更新律，并利用可测数据通过最小二乘方法求解 Bellman 方程得到更新律所需的模型信息，从而实现无需模型参数的迭代学习控制方法，证明了该方法的收敛性。进一步地，分析了所提方法在计算复杂度和求解模型所需实验批次数量方面的优势。最后，通过直流电动机的仿真，验证了所提方法的有效性和优势。

（3）针对一类随时间和批次同时变化的执行器故障下的线性离散系统的跟踪问题，提出了一种基于 Q 学习的故障估计和迭代学习容错控制方案。沿着时间和批次同时变化的未知故障会影响迭代学习控制器的跟踪性能，针对以上问题，引入了 Q 学习算法，将故障估计过程转化为马尔可夫决策过程，通过持续调整设计的故障估计器以适应变化的故障。并且，采用范数优化理论设计迭代学习容错控制器，通过 Q 学习的故障估计结果来调整控制器，以抵消故障的影响，提出了系统在该学习律下跟踪误差有界收敛的条件并给出了相应证明。最后，通过移动机器人的仿真，验证了所提方法的有效性和优势。

6.2 展望

本文针对基于强化学习的迭代学习控制与优化方法问题，尽管取得了也一些学术成果，但是受本人学术水平与时间的限制，仍有一些问题值得进一步研究和探索：

（1）强化学习与迭代学习控制的第一种结合方式，即使用强化学习方法直接设计迭代学习控制器，会将迭代学习控制过程转化为马尔可夫决策过程，由于需要找到准确合适的状态转移方程，本文的第三章与第四章内容仅考虑了情形较为理想的线性时不变离散系统，之后可以考虑将算法进一步拓展，引入博弈代数 Riccati 方程（Game Algebra Riccati Equation），将算法拓展到存在扰动或时滞的情形。

（2）关于强化学习与迭代学习控制的第二种结合方式，即间接发挥强化学习的复杂决策优势来协助处理迭代学习控制的局限性，本文目前仅讨论了非重复执行器故障的

情形，可以将处理的非重复对象，进一步拓展到非重复模型、非重复批次长度、非重复的初始状态等情形，以进一步拓宽迭代学习控制的应用场景。

（3）目前本文考虑的强化学习方法较为单一，可以进一步考虑其他更为复杂的强化学习方法如行动者-评判家、深度确定性策略梯度算法等方法，通过发挥不同强化学习方法的复杂决策优势以应对不同复杂情形的控制场景。

参考文献

[1] Bristow D A, Tharayil M, Alleyne A G. A survey of iterative learning control[J]. IEEE Control Systems Magazine, 2006, 26(3): 96-114.

[2] 李仁俊, 韩正之. 迭代学习控制综述[J]. 控制与决策, 2005, 20(09): 961-966.

[3] Shen D, Wang Y Q. Survey on stochastic iterative learning control[J]. Journal of Process Control, 2014, 24(12): 64-77.

[4] 陈强, 陈凯杰, 施卉辉, 等. 机械臂变长度误差跟踪迭代学习控制[J]. 自动化学报, 2023, 49(12): 2594-2604.

[5] Wang Y Q, Dassau E, Doyle F J. Closed-loop control of artificial pancreatic β-cell in type 1 diabetes mellitus using model predictive iterative learning control[J]. IEEE Transactions on Biomedical Engineering, 2009, 57(2): 211-219.

[6] 郑鑫鑫, 曹荣敏, 侯忠生. 基于 RBF 的直线电机二维平台无模型自适应迭代学习控制[J]. 控制工程, 2023, 30(10): 1881-1890.

[7] Hou Z S, Yan J W, Xu J X, et al. Modified iterative-learning-control-based ramp metering strategies for freeway traffic control with iteration-dependent factors[J]. IEEE Transactions on Intelligent Transportation Systems, 2011, 13(2): 606-618.

[8] Tao H F, Zheng J H, Wei J Y, et al. Repetitive process based indirect-type iterative learning control for batch processes with model uncertainty and input delay[J]. Journal of Process Control, 2023, 132: 103112.

[9] Kaelbling L P, Littman M L, Moore A W. Reinforcement learning: A survey[J]. Journal of Artificial Intelligence Research, 1996, 4: 237-285.

[10] Arulkumaran K, Deisenroth M P, Brundage M, et al. Deep reinforcement learning: A brief survey[J]. IEEE Signal Processing Magazine, 2017, 34(6): 26-38.

[11] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. Cambridge: MIT press, 2018.

[12] Singh B, Kumar R, Singh V P. Reinforcement learning in robotic applications: A comprehensive survey[J]. Artificial Intelligence Review, 2022, 55(2): 945-990.

[13] Uc-Cetina V, Navarro-Guerrero N, Martin-Gonzalez A, et al. Survey on reinforcement learning for language processing[J]. Artificial Intelligence Review, 2023, 56(2): 1543-1575.

[14] Kiran B R, Sobh I, Talpaert V, et al. Deep reinforcement learning for autonomous driving: A survey[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(6): 4909-4926.

[15] Perolat J, De Vylder B, Hennes D, et al. Mastering the game of Stratego with model-free multiagent reinforcement learning[J]. Science, 2022, 378(6623): 990-996.

[16] Taghian M, Asadi A, Safabakhsh R. Learning financial asset-specific trading rules via deep reinforcement learning[J]. Expert Systems with Applications, 2022, 195: 116523.

[17] Ahn H S. Reinforcement learning and iterative learning control: Similarity and difference[C]. Proceedings of the International Conference on Mechatronics and Information Technology (ICMIT). Gwangju, Korea: IEEE, 2009: 3-5.

[18] Nian R, Liu J F, Huang B. A review on reinforcement learning: Introduction and applications in industrial process control[J]. Computers & Chemical Engineering, 2020, 139: 106886.

[19] Zhang Y Q, Chu B, Shu Z. A preliminary study on the relationship between iterative learning control and reinforcement learning[J]. IFAC-PapersOnLine, 2019, 52(29): 314-319.

[20] 王玉刚. 非严格重复系统的迭代学习控制方法研究[D]:[博士学位论文]. 济南: 山东大学, 2021.

[21] Uchiyama M. Formation of high-speed motion pattern of a mechanical arm by trial[J]. Transactions of the Society of Instrument and Control Engineers, 1978, 14(6): 706-712.

[22] Arimoto S, Kawamura S, Miyazaki F. Bettering operation of robots by learning[J]. Journal of Robotic

Systems, 1984, 1(2): 123-140.
[23] Shen D. A technical overview of recent progresses on stochastic iterative learning control[J]. Unmanned Systems, 2018, 6(03): 147-164.
[24] Rogers E, Chu B, Freeman C, et al. Iterative learning control algorithms and experimental benchmarking[M]. Hoboken, John Wiley & Sons, 2023.
[25] 于少娟, 齐向东, 吴聚华. 迭代学习控制理论及应用[M]. 北京, 机械工业出版社, 2005.
[26] Chi R H, Li H Y, Shen D, et al. Enhanced P-type control: Indirect adaptive learning from set-point updates[J]. IEEE Transactions on Automatic Control, 2022, 68(3): 1600-1613.
[27] Gu P P, Tian S P. D-type iterative learning control for one-sided Lipschitz nonlinear systems[J]. International Journal of Robust and Nonlinear Control, 2019, 29(9): 2546-2560.
[28] 王晶, 周楠, 王森, 等. 随机变批次长度的反馈辅助 PD 型量化迭代学习控制[J]. 控制与决策, 2021, 36 (10): 2569-2576.
[29] Tao H F, Zhou L H, Hao S L, et al. Output feedback based PD-type robust iterative learning control for uncertain spatially interconnected systems[J]. International Journal of Robust and Nonlinear Control, 2021, 31(12): 5962-5983.
[30] 刘艳, 阮小娥. 线性时不变系统 PID-型迭代学习控制律的单调收敛形态[J]. 控制理论与应用, 2020, 37(09): 1873-1879.
[31] 刘保彬, 周伟. 基于高阶内模的鲁棒自适应迭代学习控制[J]. 控制工程, 2018, 25(05): 770-776.
[32] Wan K, Li X D. High-order internal model-based iterative learning control for 2-D linear FMMI systems with iteration-varying trajectory tracking[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, 51(3): 1462-1472.
[33] Freeman C T, Tan Y. Iterative learning control with mixed constraints for point-to-point tracking[J]. IEEE Transactions on Control Systems Technology, 2012, 21(3): 604-616.
[34] Tao H F, Paszke W, Rogers E, et al. Modified Newton method based iterative learning control design for discrete nonlinear systems with constraints[J]. Systems & Control Letters, 2018, 118: 35-43.
[35] Arnold F, King R. Norm-optimal iterative learning control in an integer-valued control domain[J]. International Journal of Control, 2023, 96(1): 170-181.
[36] Liu Y, Ruan X E. Linearly monotonic convergence of nonlinear parameter-optimal iterative learning control to linear discrete-time-invariant systems[J]. International Journal of Robust and Nonlinear Control, 2021, 31(9): 3955-3981.
[37] Zhao X D, Wang Y Q. Distributed point-to-point iterative learning control for multi-agent systems with quantization[J]. Journal of the Franklin Institute, 2021, 358(13): 6508-6525.
[38] Yuan H, Zhao X M. Advanced contouring compensation approach via Newton-ILC and adaptive jerk control for biaxial motion system[J]. IEEE Transactions on Industrial Electronics, 2021, 69(5): 5081-5090.
[39] Amann N, Owens D H, Rogers E. Iterative learning control using optimal feedback and feedforward actions[J]. International Journal of Control, 1996, 65(2): 277-293.
[40] Owens D H, Chu B. Error corrected references for accelerated convergence of low gain norm optimal iterative learning control[J]. IEEE Transactions on Automatic Control, 2024. DOI: 10.1109/TAC. 2024.3362857.
[41] Owens D H, Chu B, Songjun M. Parameter-optimal iterative learning control using polynomial representations of the inverse plant[J]. International Journal of Control, 2012, 85(5): 533-544.
[42] Zhang X B, Wang B F, Gamage D, et al. Model predictive and iterative learning control based hybrid control method for hybrid energy storage system[J]. IEEE Transactions on Sustainable Energy, 2021, 12(4): 2146-2158.
[43] 施卉辉, 陈强. 一类不确定系统的自适应滑模迭代学习控制[J]. 控制理论与应用, 2023, 40(07): 1162-1171.

[44] Xu K C, Meng B, Wang Z. Generalized regression neural networks-based data-driven iterative learning control for nonlinear non-affine discrete-time systems[J]. Expert Systems with Applications, 2024, 248: 123339.

[45] Dai M K, Li H X, Wang S W. A reinforcement learning-enabled iterative learning control strategy of air-conditioning systems for building energy saving by shortening the morning start period[J]. Applied Energy, 2023, 334: 120650.

[46] Zhang Y Q, Chu B, Shu Z. Model-free predictive optimal iterative learning control using reinforcement learning[C]. 2022 American Control Conference (ACC). Atlanta, USA: IEEE, 2022: 3279-3284.

[47] Song B. From model-based to data-driven discrete-time iterative learning control[D]: [PhD thesis]. New York: Columbia University, 2019.

[48] Shi J, Wen K C, Xu X H, et al. Design of nonlinear iterative learning control based on deep reinforcement learning algorithm[C]. 2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS). Suzhou, China: IEEE, 2021: 722-727.

[49] Liu J N, Hong W J, Shi J. Two dimensional (2D) feedback control scheme based on deep reinforcement learning algorithm for nonlinear non-repetitive batch processes[C]. 2022 IEEE 11th Data Driven Control and Learning Systems Conference (DDCLS). Emeishan, China: IEEE, 2022: 262-267.

[50] Meindl M, Lehmann D, Seel T. Bridging reinforcement learning and iterative learning control: Autonomous motion learning for unknown, nonlinear dynamics[J]. Frontiers in Robotics and AI, 2022, 9: 793512.

[51] Poot M, Portegies J, Oomen T. On the role of models in learning control: Actor-critic iterative learning control[J]. IFAC-PapersOnLine, 2020, 53(2): 1450-1455.

[52] 刘旭光，杜昌平，郑耀．基于强化迭代学习的四旋翼无人机轨迹控制[J]．计算机应用，2022, 42(12): 3950-3956.

[53] Xu X H, Xie H M, Shi J. Iterative learning control (ILC) guided reinforcement learning control (RLC) scheme for batch processes[C]. 2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS). Liuzhou, China: IEEE, 2020: 241-246.

[54] Liu J N, Zhou Z K, Hong W J, et al. Two-dimensional iterative learning control with deep reinforcement learning compensation for the non-repetitive uncertain batch processes[J]. Journal of Process Control, 2023, 131: 103106.

[55] Ruan Y F, Zhang Y, Mao T, et al. Trajectory optimization and positioning control for batch process using learning control[J]. Control Engineering Practice, 2019, 85: 1-10.

[56] Vuga R, Nemec B, Ude A. Enhanced policy adaptation through directed explorative learning[J]. International Journal of Humanoid Robotics, 2015, 12(03): 1550028.

[57] Nemec B, Simonič M, Likar N, et al. Enhancing the performance of adaptive iterative learning control with reinforcement learning[C]. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Vancouver, Canada: IEEE, 2017: 2192-2199.

[58] Jin Z W, Ma M L, Zhang S T, et al. Secure state estimation of cyber-physical system under cyber attacks: Q-learning vs. SARSA[J]. Electronics, 2022, 11(19): 3161.

[59] Li J N, Xiao Z F, Li P, et al. Networked controller and observer design of discrete-time systems with inaccurate model parameters[J]. ISA Transactions, 2020, 98: 75-86.

[60] Herman M, Gindele T, Wagner J, et al. Inverse reinforcement learning with simultaneous estimation of rewards and dynamics[C]. Artificial Intelligence and Statistics(AISTATS). Cadiz, Spain: W&CP, 2016: 102-110.

[61] Shahrabi J, Adibi M A, Mahootchi M. A reinforcement learning approach to parameter estimation in dynamic job shop scheduling[J]. Computers & Industrial Engineering, 2017, 110: 75-82.

[62] Owens D H, Hätönen J. Iterative learning control—An optimization paradigm[J]. Annual Reviews in

Control, 2005, 29(1): 57-70.
[63] Sun H Q, Alleyne A G. A computationally efficient norm optimal iterative learning control approach for LTV systems[J]. Automatica, 2014, 50(1): 141-148.
[64] Chu B, Rauh A, Aschemann H, et al. Constrained iterative learning control for linear time-varying systems with experimental validation on a high-speed rack feeder[J]. IEEE Transactions on Control Systems Technology, 2021, 30(5): 1834-1846.
[65] Zhou C H, Tao H F, Chen Y Y, et al. Robust point-to-point iterative learning control for constrained systems: A minimum energy approach[J]. International Journal of Robust and Nonlinear Control, 2022, 32(18): 10139-10161.
[66] Lewis F L, Vrabie D, Vamvoudakis K G. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers[J]. IEEE Control Systems Magazine, 2012, 32(6): 76-105.
[67] Paszke W, Rogers E, Gałkowski K, et al. Robust finite frequency range iterative learning control design and experimental verification[J]. Control Engineering Practice, 2013, 21(10): 1310-1320.
[68] Stoorvogel A A, Weeren A J T M. The discrete-time Riccati equation related to the H∞ control problem[J]. IEEE Transactions on Automatic Control, 1994, 39(3): 686-691.
[69] Lancaster P, Rodman L. Algebraic Riccati equations[M]. New York: Clarendon press, 1995.
[70] Norrlöf M, Gunnarsson S. Time and frequency domain convergence properties in iterative learning control[J]. International Journal of Control, 2002, 75(14): 1114-1126.
[71] 陶洪峰, 李健, 杨慧中. 输入约束不确定系统的点对点迭代学习控制与优化[J]. 控制与决策, 2021, 36(06): 1435-1441.
[72] Golub G H, Van Loan C F. Matrix Computations[M]. Baltimore: JohnsHopkins University Press, 2013.
[73] Tao H F, Chen D P, Yang H Z. Iterative learning fault diagnosis algorithm for non-uniform sampling hybrid system[J]. IEEE/CAA Journal of Automatica Sinica, 2017, 4(3): 534-542.
[74] Barton K L, Bristow D A, Alleyne A G. A numerical method for determining monotonicity and convergence rate in iterative learning control[J]. International Journal of Control, 2010, 83(2): 219-226.
[75] Bradtke S J, Ydstie B E, Barto A G. Adaptive linear quadratic control using policy iteration[C]. Proceedings of American Control Conference (ACC). Baltimore, USA: IEEE, 1994, 3: 3475-3479.
[76] Jiang Y, Kiumarsi B, Fan J L, et al. Optimal output regulation of linear discrete-time systems with unknown dynamics using reinforcement learning[J]. IEEE Transactions on Cybernetics, 2019, 50(7): 3147-3156.
[77] Chai Y, Luo J J, Ma W H. Data-driven game-based control of microsatellites for attitude takeover of target spacecraft with disturbance[J]. ISA Transactions, 2022, 119: 93-105.
[78] 刘秀华, 韩建, 魏新江. 基于中间观测器的多智能体系统分布式故障估计[J]. 自动化学报, 2020, 46(01): 142-152.
[79] Zhu F L, Fu Y H, Dinh T N. Asymptotic convergence unknown input observer design via interval observer[J]. Automatica, 2023, 147: 110744.
[80] Tabatabaeipour S M, Bak T. Robust observer-based fault estimation and accommodation of discrete-time piecewise linear systems[J]. Journal of the Franklin Institute, 2014, 351(1): 277-295.
[81] Lan Z M. Iterative learning control algorithm for sensor fault nonlinear systems[J]. Journal of Intelligent & Fuzzy Systems, 2021, 40(4): 5927-5934.
[82] Wang L M, Liu F F, Yu J X, et al. Iterative learning fault-tolerant control for injection molding processes against actuator faults[J]. Journal of Process Control, 2017, 59: 59-72.
[83] Wang Y Q, Shi J, Zhou D H, et al. Iterative learning fault-tolerant control for batch processes[J]. Industrial & Engineering Chemistry Research, 2006, 45(26): 9050-9060.
[84] Tao H F, Paszke W, Rogers E, et al. Iterative learning fault-tolerant control for differential time-delay

batch processes in finite frequency domains[J]. Journal of Process Control, 2017, 56: 112-128.
[85] Tao H F, Li J, Chen Y Y, et al. Robust point-to-point iterative learning control with trial-varying initial conditions[J]. IET Control Theory & Applications, 2020, 14(19): 3344-3350.
[86] Owens D H, Freeman C T, Van Dinh T. Norm-optimal iterative learning control with intermediate point weighting: Theory, algorithms, and experimental evaluation[J]. IEEE Transactions on Control Systems Technology, 2012, 21(3): 999-1007.
[87] Yang S P, Xu J X, Huang D Q, et al. Optimal iterative learning control design for multi-agent systems consensus tracking[J]. Systems & Control Letters, 2014, 69: 80-89.
[88] Zhuang Z H, Tao H F, Chen Y Y, et al. An optimal iterative learning control approach for linear systems with nonuniform trial lengths under input constraints[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2023, 53(6): 3461-3473.
[89] Watanabe K, Tang J, Nakamura M, et al. A fuzzy-Gaussian neural network and its application to mobile robot control[J]. IEEE Transactions on Control Systems Technology, 1996, 4(2): 193-199.
[90] Jin X. Fault-tolerant iterative learning control for mobile robots non-repetitive trajectory tracking with output constraints[J]. Automatica, 2018, 94: 63-71.
[91] Ding S X. Model-based fault diagnosis techniques: Design schemes, algorithms, and tools[M]. Berlin, Springer, 2008.