

摘要

在多智能体系统(Multi-agent System, MAS)中,智能体之间通过协调与合作,能够实现单个智能体无法完成的复杂任务,因此多智能体系统的协调与控制已应用到众多领域中,例如在移动机器人、传感器网络和卫星编队等领域。一致性问题是多智能体系统控制中的基本问题,是指智能体依靠局部信息来实现对全局的控制,最终使智能体的状态变量趋于一致,通常这个问题的假设条件包括智能体的稳定性、拓扑结构的固定性以及网络通信的可靠性。

对于具有周期性运行特性的多智能体系统,迭代学习控制(Iterative Learning Control, ILC)因其固有的重复机制而逐渐应用于执行重复跟踪任务的多智能体系统中。它通过使用之前批次的系统输入输出信息,不断修正和更新当前批次的控制输入信号,最终能够在有限时间内实现对参考轨迹的完全跟踪。在实际的工业过程中,往往并不需要跟踪完整的参考轨迹,只需要在特定时间跟踪上给定的目标值,这种跟踪控制问题被称为点对点跟踪问题。由于点对点跟踪任务放宽了对跟踪的要求,所以极大地提高了算法设计的自由度。因此,在保证跟踪精度的同时,提高系统整体的性能,本文研究了多智能体系统点对点迭代学习控制及其性能优化。主要研究内容和创新点包括:

1. 在固定拓扑结构下,对于具有周期性运动特性的线性离散多智能体系统,针对其点对点跟踪问题,设计了分布式点对点迭代学习控制方法,并且给出其收敛的充分必要条件。由于实际存在的设备老化、测量偏差等原因,这使得实际的系统模型与标称系统模型往往存在偏差,因此进一步讨论了在模型不确定性下的鲁棒性分析。最后,通过仿真结果验证了所提方法的有效性与合理性。

2. 针对多智能体系统点对点迭代学习控制问题,考虑到实际工程应用中减少能量损耗的需求,将优化理论与迭代学习控制技术相结合,提出了一种集中式范数优化点对点迭代学习控制方法,并且给出其收敛的充分必要条件,进一步讨论了在模型不确定性下的系统鲁棒性分析。由于所提算法是一种集中式的控制形式,因此存在诸多限制,利用交替方向乘子法(Alternating Direction Method of Multipliers, ADMM)实现对集中式控制的分布式解决方法,在此基础上分析其鲁棒性。最后,通过仿真结果验证了所提方法的有效性与合理性。

3. 针对多智能体系统点对点迭代学习控制问题,当点对点跟踪任务中的跟踪时间不同时,系统输出轨迹不同,所对应的系统输入能量也不同。因此将跟踪时间作为变量,利用坐标下降法,设计了跟踪时间优化点对点迭代学习控制方法,通过迭代的方式寻找最优的跟踪时间,从而进一步减小系统的能量损耗,并且给出其收敛的充分条件,在此基础上讨论了在模型不确定性下的系统鲁棒性分析。最后,通过仿真结果验证了所提方法的有效性与合理性。

关键词: 迭代学习控制; 多智能体系统; 点对点跟踪; 最小能量

Abstract

In Multi-agent Systems (MAS), agents can accomplish complex tasks that cannot be completed by a single agent through coordination and cooperation among agents. Therefore, the coordination and control of MAS have been applied to many fields, such as mobile robots, sensor networks, and satellite formations. Consensus problem is a fundamental issue in MAS control, which refers to the use of local information by agents to achieve global control and ultimately make the state variables of the agents tend to consensus. The assumptions for this problem usually include the stability of agents, the fixed topology of the network, and the reliability of network communication.

For multi-agent systems with periodic motion characteristics, Iterative Learning Control (ILC) has gradually been applied to multi-agent systems performing repetitive tracking tasks due to its inherent repetition mechanism. By using the input-output information of the previous trials of the system, it continuously corrects and updates the current trail of control input signals, and can finally achieve complete tracking of the reference trajectory in a finite time. In practical industrial processes, it is often not necessary to track the complete reference trajectory, but only to track the given target value at specific times. This tracking control problem is called point-to-point tracking problem. Since point-to-point tracking tasks relax the requirements for tracking, it greatly improves the freedom of design. Therefore, this paper studies the point-to-point iterative learning control of multi-agent systems and its performance optimization while ensuring tracking accuracy and improving the overall performance of the system.

The main research contents and innovations include:

1. A distributed point-to-point iterative learning control method is designed for linear discrete multi-agent systems with periodic motion characteristics under a fixed topology structure, and its sufficient and necessary condition for convergence is provided. Due to the existence of factors such as equipment aging and measurement deviation in practical applications, there are deviations between the actual system model and the nominal system model. Therefore, the robustness analysis under model uncertainty is further discussed. Finally, the effectiveness and rationality of the proposed method are verified through simulation results.

2. A centralized norm-optimal point-to-point iterative learning control method is proposed by combining optimization theory with iterative learning control for the point-to-point tracking problem of multi-agent systems. Its sufficient and necessary condition for convergence is provided, and the robustness analysis of the system under model uncertainty is further discussed. As the proposed algorithm is a centralized control form, there are many restrictions. To address this, the Alternating Direction Method of Multipliers (ADMM) is used to achieve a distributed solution for centralized control. On this basis, the robustness of the approach is analyzed. Finally, the effectiveness and rationality of the proposed method are verified through simulation results.

3. For the point-to-point iterative learning control problem of multi-agent systems, when

the tracking time instants in the point-to-point tracking task are different, the system output trajectory and the corresponding system input energy are also different. Therefore, by treating tracking time instants as variables and using coordinate descent method, a point-to-point iterative learning control method with optimized tracking time instants is designed, which searches for the optimal tracking time instants iteratively in order to further reduce the energy loss of the system. Its sufficient condition for convergence is provided, and the robustness analysis of the system under model uncertainty is further discussed. Finally, the effectiveness and rationality of the proposed method are verified through simulation results.

Keywords: iterative learning control; multi-agent systems; point-to-point tracking problem; energy optimization

主要符号表

符号	代表意义
$\mathbb{N}$	自然数集合
$\mathbb{R}$	实数集合
$\mathbb{R}^n$	n 维实数列向量集合
$\mathbb{R}^{n \times n}$	n 阶实数矩阵集合
I_n	$n \times n$ 阶单位矩阵
$0_{n \times m}$	$n \times m$ 阶零矩阵
$\ A\ $	矩阵 A 的欧几里得范数
A^{-1}	矩阵 A 的逆
A^T	矩阵 A 的转置
$diag\{A_1, \dots, A_n\}$	分块对角矩阵
$X \otimes Y$	矩阵 X 和矩阵 Y 的克罗内克积
$\ x\ _R^2 = x^T R x$	范数空间 R 内的 x 范数
$\langle x, y \rangle_R = x^T R y$	范数空间 R 内的 x, y 内积
mod	取模运算符

目 录

第一章 绪论.....	1
1.1 课题背景与意义.....	1
1.2 迭代学习控制的研究现状	2
1.3 多智能体系统迭代学习控制的研究现状	4
1.4 本文主要研究工作.....	5
第二章 预备知识	7
2.1 多智能体系统的拓扑结构	7
2.2 多智能体系统常见控制结构	9
2.3 交替方向乘子法.....	10
2.4 本章小结.....	11
第三章 分布式多智能体系统点对点迭代学习控制	12
3.1 引言.....	12
3.2 问题描述.....	12
3.2.1 系统描述.....	12
3.2.2 点对点跟踪任务中的一致性问题.....	13
3.3 分布式点对点算法设计.....	15
3.4 鲁棒性分析.....	17
3.5 仿真结果与分析.....	18
3.6 本章小结.....	24
第四章 最小能量约束多智能体系统点对点迭代学习控制	25
4.1 引言.....	25
4.2 问题描述.....	25
4.3 集中式范数优化点对点算法设计	26
4.4 分布式范数优化点对点算法设计	29
4.5 鲁棒性分析.....	31
4.6 仿真结果与分析.....	32
4.7 本章小结.....	35
第五章 跟踪时间优化多智能体系统点对点迭代学习控制	36
5.1 引言.....	36
5.2 问题描述.....	36
5.3 跟踪时间优化点对点算法设计	37
5.3.1 最小能量约束.....	37
5.3.2 跟踪时间优化.....	38
5.4 鲁棒性分析.....	40

5.5 仿真结果与分析.....	41
5.6 本章小结.....	48
第六章 结论与展望	49
6.1 结论.....	49
6.2 展望.....	49
参考文献.....	51

第一章 绪论

1.1 课题背景与意义

在自然界中,群体之间的协调与合作现象非常普遍,比如候鸟迁徙、蜂群采蜜、鱼群捕食等。这些群体是由大量相似的个体组成,虽然个体能力具有局限性,但是通过个体之间的协调与合作,可以使群体完成个体难以企及的任务。学者们受到自然界中生物集群合作的启发,利用数学公式建立了多智能体系统模型^[1]。它是由一系列具有自治性、反应性、社会性、主动性等特征的个体(智能体)组成,通过相互协作实现单个智能体无法完成的复杂任务,并且具有自适应、协同性强、可扩展的优点,因此多智能体系统控制逐渐成为智能控制领域中的重要分支之一。

随着计算机技术与网络通信技术的飞速发展,多智能体系统的协调控制在实际应用领域中取得了显著成果,如航天器的编队优化^[2]、微电网控制^[3]、无人机编队控制^[4,5]、多机械臂协调控制^[6]等。目前,多智能体系统协调控制问题可以分为一致性问题^[7]、群集控制问题^[8]、编队控制问题^[9]、蜂拥控制问题^[10]等。其中,一致性问题是多智能体系统协调控制中最基础的,也是最关键的问题,并且在一阶系统^[11]、二阶系统^[12,13]、高阶系统^[14,15]、非线性系统^[16]等方面中均取得了很大的进展。

实际应用中,通常需要多个智能体重复执行某项任务,比如卫星编队控制^[17]、多列车轨迹跟踪控制^[18]等。并且为了保证智能体执行任务的同步性,需要在智能体之间进行通信,保持协调。但是由于通信拓扑结构具有局限性,导致了只有有限个数的智能体能够直接获取期望跟踪任务的信息,而其他智能体只能从邻接智能体处获取信息。那么对于算法设计来说,就是尽可能利用能够获取的局部信息以实现期望轨迹的高精度跟踪,迭代学习控制因其特殊的重复控制机制在处理有限时间内不断执行周期性任务的多智能体系统具有很大优势。

迭代学习控制的核心思想是通过学习已完成批次历史信息,不断修正当前批次的控制输入信号,从而使得系统跟踪性能随着迭代批次的增加而不断改善。迭代学习控制的设计依赖于历史信息,即使在系统信息不精确的情况下,也能够实现高精度跟踪,体现了其具有鲁棒性。因此,迭代学习控制在需要高精度的工业应用中具有非常可观的前景,例如喷墨打印打印机^[19,20]、化工批次过程^[21,22]、中风康复^[23]等。

在传统的跟踪任务中,通常都需要使系统的实际输出跟踪上给定的完整参考轨迹。然而对于有些系统,只需要在特定时间跟踪上某些关键点的位置即可,不需要跟踪上完整的参考轨迹,这类跟踪问题被称为点对点跟踪问题,如文献[24]中的机械臂执行“拾取”和“放置”操作任务,文献[25]中的中风康复系统在某些特定的时刻人工刺激肌肉进行功能性训练。由于点对点跟踪任务放宽了跟踪需求,所以极大地提高了算法设计的自由度,也提升了系统整体性能。在点对点迭代学习控制中,当所需要跟踪的关键点数量足够多时,点对点迭代学习控制跟踪问题也就逐渐变成了全轨迹迭代学习控制跟踪问题。因此,可以把全轨迹迭代学习控制看成是点对点迭代学习控制的一种特例,将迭代学习控制框架应用在点对点跟踪问题上更具有普遍性。

多智能体系统能够实现单个智能体无法完成的复杂任务，所以有必要将点对点迭代学习控制框架扩展到多智能体系统^[26-28]。本文针对具有周期性运行特性的多智能体系统，使用迭代学习控制方法对点对点跟踪任务中的一致性问题的研究展开研究工作，并且采用优化的思想进一步提升系统的性能，该课题的理论研究对于实际生产过程具有较为可观的应用前景。

1.2 迭代学习控制的研究现状

迭代学习控制思想的提出最早可以追溯到 1978 年，由日本学者 M.Uchiyama 用日文撰写的一篇论文^[29]，但是因为语言的限制导致在日本以外的地方没有受到很大的关注。直到 Arimoto 等人^[30]将迭代学习控制应用于机器人控制系统中，实现了较高的跟踪性能，相关的理论研究才进入高速发展阶段。从迭代学习控制思想的提出至今的四十多年时间里，其理论体系已经发展得相当完善，如今迭代学习控制已经是控制领域中重要的研究方法，适用于具有重复运行特性的系统。与其他控制方法不同的是，在迭代学习控制中除了需要考虑时间轴，还需要考虑迭代轴，而迭代学习控制的目的就是要实现沿批次轴的收敛，换句话说迭代学习控制是一个典型的二维(2D)过程。

迭代学习控制的基本结构如图 1-1，其中 u_k 为第 k 批次的系统输入信号， y_k 为第 k 批次的系统输出轨迹， r 为期望的跟踪轨迹，定义 $e_k = r - y_k$ 为第 k 批次的系统跟踪误差。迭代学习控制的原理可以描述为：利用上一批次产生的输入信号 u_k 和跟踪误差 e_k ，更新当前批次的控制输入信号 u_{k+1} ，保存在存储器中并且再次作用于被控系统，从而产生相应的输出轨迹 y_{k+1} 。如此循环往复，随着迭代批次 k 的增加，使得跟踪误差 e_k 不断减小，直至系统输出轨迹最终完全跟踪上期望轨迹，即 $e_k = 0$ 。可以看到，迭代学习控制的本质是前馈控制，但在迭代学习控制中控制律的设计运用了之前批次的信息，因此具有较好的跟踪性能。

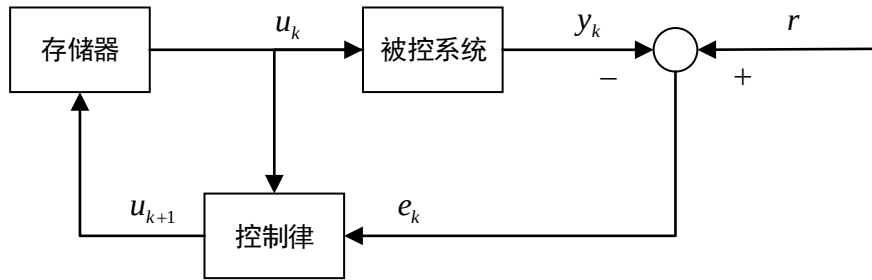


图 1-1 迭代学习控制的基本结构图

在传统的迭代学习控制研究中，通常假设系统满足以下条件：

- (1) 系统运行在每个迭代批次的时间是有限且固定的，即 $t \in [0, T], T > 0$ ；
- (2) 每个批次结束后，系统的初始状态会重置为一个恒值，即 $x_k(0) = x_0, \forall k$ ；
- (3) 期望轨迹 r 是提前给定的跟踪控制任务，不会随批次变化；
- (4) 每个批次的跟踪误差 e_k 可测，并且基于当前批次的 u_k ，可以用来更新下一批次的 u_{k+1} ；

- (5) 整个迭代过程中保证系统动力学的不变性，并且系统动力学是可逆的，即存在一个可以反转给定系统动力学的因果稳定系统。

在迭代学习控制研究中，控制算法(控制律)的设计就显得尤为重要，其决定了系统的收敛性、收敛速度以及鲁棒性等性能。自 Arimoto 提出 D 型控制律后，学者们根据对系统跟踪误差的不同利用方式设计了许多形式不同的控制律，如 P 型^[31]、PD 型^[32]、PID 型^[33]以及高阶控制律^[34]等。因为控制律固有的重复特性，类似于沿批次方向上有积分作用^[35]，故而通常不会单独将 I 型(积分型)单独使用。传统的迭代学习控制算法因为其控制律具有结构简单、便于实现的优点，是早期迭代学习控制领域研究和应用最多的算法。学者们后续又将优化理论与迭代学习控制结合，其中具有代表型的优化算法有梯度下降法^[36]、牛顿法^[37]、范数优化迭代学习控制^[38]等。此外，随着智能控制的发展，学者们将迭代学习控制与鲁棒控制^[39]、预测控制^[40]、自适应控制^[41]相结合，解决了系统控制存在模型不确定性、多变量约束、非周期性随机扰动等问题。

根据跟踪控制任务的不同，可以将迭代学习控制设计分为全轨迹迭代学习控制^[31-34,38-41]和点对点迭代学习控制^[35-37,45-47]。大多数情况下，迭代学习控制解决了有限时间范围内跟踪完整参考轨迹的问题。然而，在点对点迭代学习控制中，当所需要跟踪的关键点数量足够多时，点对点迭代学习控制跟踪问题也就逐渐变成了全轨迹迭代学习控制跟踪问题。因此，可以把全轨迹迭代学习控制看成是点对点迭代学习控制的一种特例，将迭代学习控制框架应用在点对点跟踪问题上更具有普遍性。点对点迭代学习控制首先被用来解决控制系统终端的问题，也被称为终端迭代学习控制(Terminal Iterative Learning Control, TILC)，如火车进站控制^[42]、弹道控制^[43]、快速热反应过程^[44]等。在这些研究中，终端迭代学习控制就是利用终端的跟踪误差信息来更新控制输入，显然终端迭代学习控制也是点对点迭代学习控制的一种特例。由于点对点跟踪问题中所需要跟踪的关键点数量是任意的，所以终端跟踪问题并不能直接应用于一般的点对点跟踪问题。作为终端迭代学习控制的扩展，Freeman 等人开发了点对点迭代学习控制算法^[36,37]，由于点对点迭代学习控制放宽了跟踪需求，因此极大地提高了算法设计的自由度，也提升了系统整体性能。

现有的点对点迭代学习控制框架总是将多个目标位置的跟踪时间视为先验知识，通常会嵌入在需要经过多个批次优化的性能指标函数中。将这类点对点迭代学习控制框架进行扩展，通过将几个关键点处的跟踪时间作为额外变量来优化系统性能，例如最小化能量、提高生产效率和减少机器部件损坏，从而实现实际效益。文献[45,46]中分别提出了连续时间系统和离散时间系统中最优跟踪时间分配方法，使点对点任务的系统能量最小。文献[47]在此基础上，将迭代算法扩展到约束系统中，提出了鲁棒点对点最小能量迭代学习控制算法。

为了能够较为清晰地阐述上述问题，以图 1-2 实现点对点跟踪任务的机械臂为例。控制任务可以描述为：机械臂在有限时间 $[0, T]$ 内能够进行重复性跟踪任务，从每个批次开始($t=0$)从初始点(黑色点)出发，在指定时间 t_1 移动到第一个跟踪点(粉色点)，然后依次在指定时间 t_2 和指定时间 t_3 移动到第二个跟踪点(红色点)和最后的跟踪点(蓝色点)，

最终在每个批次结束时($t=T$)复位到初始点。虽然实现了相同的点对点跟踪任务时,但是由于跟踪时间(t_1, t_2, t_3)的不同,两条输出轨迹存在差异,因此系统的跟踪轨迹及其相应的性能取决于跟踪时间。

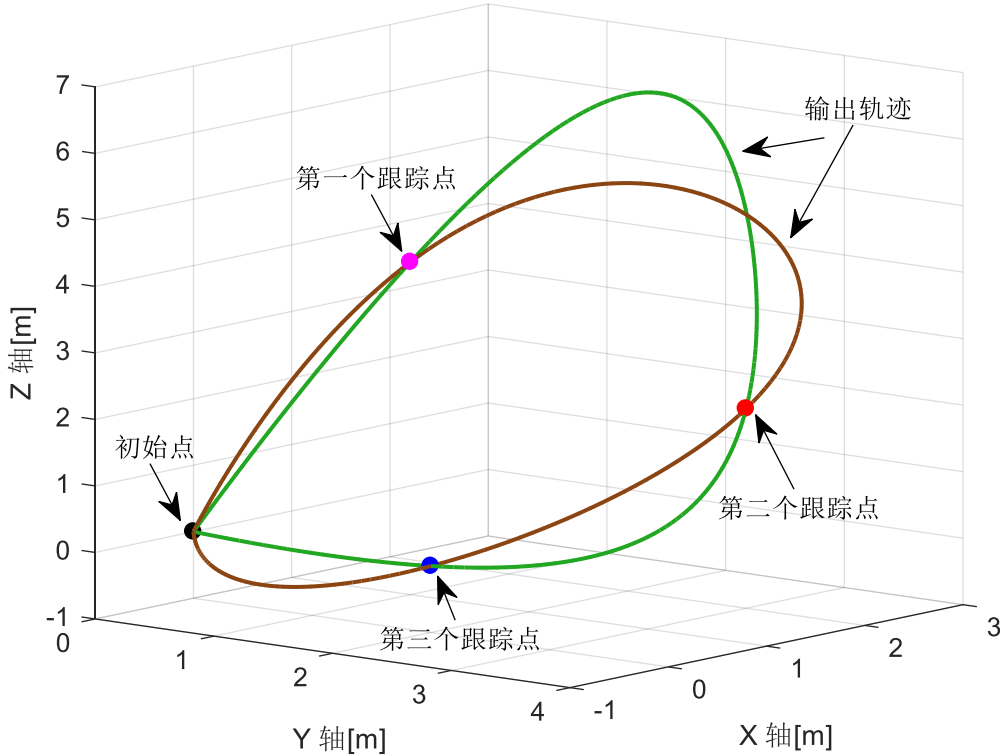


图 1-2 实现点对点跟踪任务的机械臂

1.3 多智能体系统迭代学习控制的研究现状

近年来随着迭代学习控制理论的不完善,对于一些特定的应用场景,学者们采用迭代学习控制的方法来处理。Ahn 等人^[17]提出了一种迭代学习控制方案来保持卫星编队飞行过程中的轨迹,并且以重复补偿外部干扰的方法实现鲁棒控制。Sun 等人^[18]用迭代学习控制设计多列车轨迹跟踪算法,以满足高速铁路的高安全性要求。Meng 等人研究了多智能体系统的群体控制问题,提出了分布式迭代学习控制算法,分别解决了非线性动力学和交换网络拓扑的编队控制问题^[48]、受随机扰动影响的鲁棒性问题^[49]、高精度控制问题^[50]。Xiao 等人^[51]将迭代学习控制应用到多智能体系统的编队控制中,分别包括时不变编队、时变编队和轨迹跟踪,体现了迭代学习控制算法的有效性。

编队控制问题依赖于一致性问题,学者们针对具有周期性运行特性的多智能体系统,深入研究其一致性问题,并且取得了一定的研究成果。针对一类单积分多智能体系统的一致性问题,Li 等人^[52]提出了分布式迭代学习控制算法,给出了相应收敛性的证明,并扩展到一般的线性多智能体系统中。针对一类双积分多智能体系统,Shi 等人^[53]通过提出分布式迭代学习控制算法,解决了一致性问题。Meng 等人^[54]从二维系统的角度分别以时间步长和迭代次数为自变量构造了两类分布式迭代学习控制协议,实现了对在任意期望的终端输出上实现一致性跟踪,后来又将研究工作延伸至对时变参考轨迹的跟踪

[55], 针对初始状态偏差、干扰和切换拓扑的情况解决了鲁棒性问题[56]。也有学者将自适应控制思想引入分布式迭代学习控制算法, 在文献[57]中开发了一种具有自适应机制的连续滑模跟踪协议解决了具有扰动和执行器故障的二阶非线性多智能体系统的一致性跟踪问题, 在文献[58]中解决了非线性的领导者-追随者多智能体系统的一致性跟踪问题, 并且基于李雅普诺夫稳定性理论进行分析。

针对多智能体系统点对点跟踪问题, 目前研究成果并不多。在文献[26]中提出了一种基于自适应估计且具有恒定控制输入的非线性离散时间多智能体系统终端迭代学习控制, 使用数据驱动的方法解决终端跟踪控制任务。文献[27]中提出了一种用于多智能体系统跟踪误差量化的分布式点对点迭代学习控制来解决点对点跟踪问题。文献[28]中开发了针对点对点一致性跟踪问题的范数优化 ILC 算法, 并且为这种算法提供了分布式实现方法, 使所提算法能够应用于大规模的网络动力系统。虽然以上文献解决了多智能体系统中点对点一致性跟踪控制问题, 但却把关键点处的跟踪时间作为先验知识, 从而减少了系统额外设计的自由度。因此, 本文将点对点迭代学习控制框架扩展到多智能体系统中, 通过将几个关键点处的跟踪时间作为额外变量来优化系统性能。

1.4 本文主要研究工作

本文针对具有周期性运行特性的多智能体系统, 使用迭代学习控制方法对点对点跟踪任务中的一致性问题的展开研究工作, 并在系统运行过程中对其性能进行优化。

本文各章节的内容安排如下:

第一章为绪论部分, 交代了本课题的研究背景与意义, 分别阐述了迭代学习控制的研究现状以及多智能体系统迭代学习控制的研究现状, 最后总结了全文主要研究工作。

第二章为预备知识部分, 包括多智能体系统的拓扑结构、迭代学习控制方法、多智能体系统常见控制结构以及交替方向乘子法, 为后续的研究提供了理论基础。

第三章研究了在固定拓扑结构下, 对于具有周期性运动特性的线性离散多智能体系统, 针对其点对点跟踪问题, 设计了分布式点对点迭代学习控制方法, 并且给出其收敛的充分必要条件。由于实际存在的设备老化、测量偏差等原因, 这使得实际的系统模型与标称系统模型往往存在偏差, 因此进一步讨论了在模型不确定性下的鲁棒性分析。最后, 通过仿真结果验证了所提方法的有效性与合理性。

第四章针对多智能体系统点对点迭代学习控制问题, 考虑到实际工程应用中减少能量损耗的需求, 将优化理论与迭代学习控制技术相结合提出了一种集中式范数优化点对点迭代学习控制方法, 并且给出其收敛的充分必要条件, 进一步讨论了在模型不确定性下的系统鲁棒性分析。由于所提算法是一种集中式的控制形式, 因此存在诸多限制, 利用 ADMM 方法实现对集中式控制的分布式解决方法, 在此基础上分析其鲁棒性。最后, 通过仿真结果验证了所提方法的有效性与合理性。

第五章针对多智能体系统点对点迭代学习控制问题, 当点对点跟踪任务中的跟踪时间不同时, 系统输出轨迹不同, 所对应的系统输入能量也不同。因此将跟踪时间作为变量, 利用坐标下降法, 设计了跟踪时间优化点对点迭代学习控制方法, 通过迭代的方式寻找最优的跟踪时间, 从而进一步减小系统的能量损耗, 并且给出其收敛的充分条件,

在此基础上讨论了在模型不确定性下的系统鲁棒性分析。最后，通过仿真结果验证了所提方法的有效性与合理性。

第六章为结论与展望部分，对本文的研究内容进行了总结，并且对后续研究问题工作进行展望。

第二章 预备知识

本章为后续章节要用的一些基础理论，包括多智能体系统的拓扑结构、迭代学习控制方法、多智能体系统常见控制结构以及交替方向乘子法的相关知识。

2.1 多智能体系统的拓扑结构

在多智能体系统的协调控制中，智能体之间的通信关系通常会使用代数图论进行描述。将每个智能体抽象为一个节点，则所有智能体的集合可以表示为节点的集合 $\mathcal{V} = \{1, 2, \dots, S\}$ ；智能体之间的通信关系抽象为连接节点的边，则智能体之间边的集合表示为 $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ ；智能体之间的通信情况可以描述为一个加权邻接矩阵 $\mathcal{A} = (a_{ij}) \in \mathbb{R}^{S \times S}$ ，权重表示信息的影响力。因此，使用拓扑结构图 $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ 来描述多智能体系统的网络通信情况。

在邻接矩阵 $\mathcal{A} = (a_{ij}) \in \mathbb{R}^{S \times S}$ 中，用 i 和 j 分别表示两个不同的节点， a_{ij} 表示节点 i 和节点 j 之间的连接权值，本文中不考虑权值大小对拓扑结构的影响，赋值关系如下：

$$a_{ij} = \begin{cases} 1, (j, i) \in \mathcal{E} \\ 0, (j, i) \notin \mathcal{E} \end{cases}$$

$a_{ij} = 1$ 表示节点 i 可从节点 j 处获取信息，将节点 j 定义为节点 i 的父节点，也就是说节点 i 为节点 j 的子节点， $a_{ij} = 0$ 则表示节点 i 与节点 j 之间没有通信关系。在本文的拓扑结构中，不考虑自反馈的影响，即 $(i, i) \notin \mathcal{E}$ 。

图 2-1 所示的是两种常见拓扑结构，若 $(i, j) \in \mathcal{E}$ 且 $(j, i) \in \mathcal{E}$ 同时存在，则图被称为无向拓扑图，如图左所示。否则该图为有向拓扑图，如图右所示。图中带数字编号的圆圈就是节点，节点之间的线就是边，边上的权重表示为信息的影响力，默认为 1。如图 2-1(a) 所示的无向图，任意两个不同的节点之间都存在着一一条路径，则该图被称为连通图。如图 2-1(b) 所示的有向图，任意两个不同的节点之间都存在着一一条有向路径，则该图被称为有向强连通图。如果拓扑结构图 $\mathcal{G}$ 中存在一个节点沿图中边的方向到其他任意节点都有一条有向路径，则该节点是全局可达的。

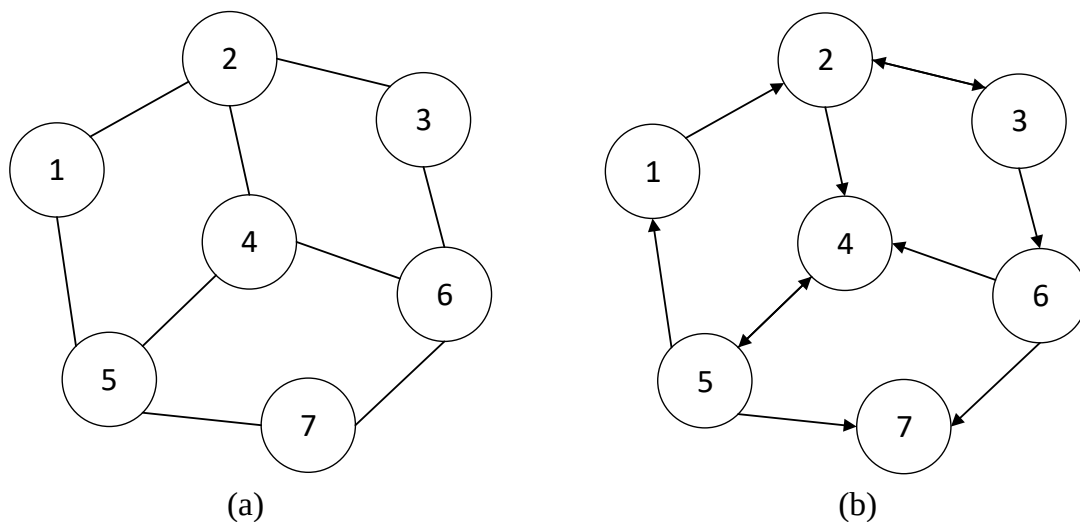


图 2-1 无向图与有向图

无向图与有向图的区别在于，有向图的边是带有方向信息的。因此，描述无向图与有向图的邻接矩阵分别为

$$\mathcal{A}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}, \mathcal{A}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

在代数图论中，通常使用拉普拉斯矩阵 L 对拓扑结构 $\mathcal{G}$ 进行理论分析，其定义为：

$$L = \mathcal{D} - \mathcal{A}$$

其中， $\mathcal{D} = \text{diag}(d_1^{\text{in}}, \dots, d_s^{\text{in}})$ 为拓扑图的入度矩阵，矩阵节点 i 的入度定义为 $d_i^{\text{in}} = \sum_{j=1}^s a_{ij}$ 。

则上述无向图与有向图的拉普拉斯矩阵分别表示为

$$L_1 = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 3 & -1 & -1 & 0 \\ -1 & 0 & 0 & -1 & 3 & 0 & -1 \\ 0 & 0 & -1 & -1 & 0 & 3 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}, L_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 3 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

显然，拉普拉斯矩阵包含了拓扑结构的所有信息，并且拉普拉斯矩阵的每行元素之和为 0。为了便于在后续章节中对拉普拉斯矩阵的应用，提前定义矩阵的克罗内克积及部分运算性质。

定义 2.1 给定任意矩阵 $X \in \mathbb{R}^{m \times n}$, $Y \in \mathbb{R}^{p \times q}$ ，则矩阵 X 和矩阵 Y 的克罗内克积定义为：

$$X \otimes Y = \begin{bmatrix} x_{11}Y & x_{12}Y & \dots & x_{1n}Y \\ x_{21}Y & x_{22}Y & \dots & x_{2n}Y \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1}Y & x_{m2}Y & \dots & x_{mn}Y \end{bmatrix} \in \mathbb{R}^{mp \times nq}$$

满足以下运算性质：

- (1) 结合律： $(X \otimes Y) \otimes Z = X \otimes (Y \otimes Z)$
- (2) 分配律： $X \otimes (Y + Z) = X \otimes Y + X \otimes Z, (X + Y) \otimes Z = X \otimes Z + Y \otimes Z$
- (3) 双线性： $\lambda(X \otimes Y) = (\lambda X) \otimes Y$
- (4) 转置： $(X \otimes Y)^T = X^T \otimes Y^T$
- (5) 若矩阵 X, Y 非奇异， $(X \otimes Y)^{-1} = X^{-1} \otimes Y^{-1}$

2.2 多智能体系统常见控制结构

常见的三种控制系统结构如图 2-2 所示，可以分为集中式控制、分散式控制和分布式控制，它们之间的区别主要体现在控制决策和执行方式上^[59]。

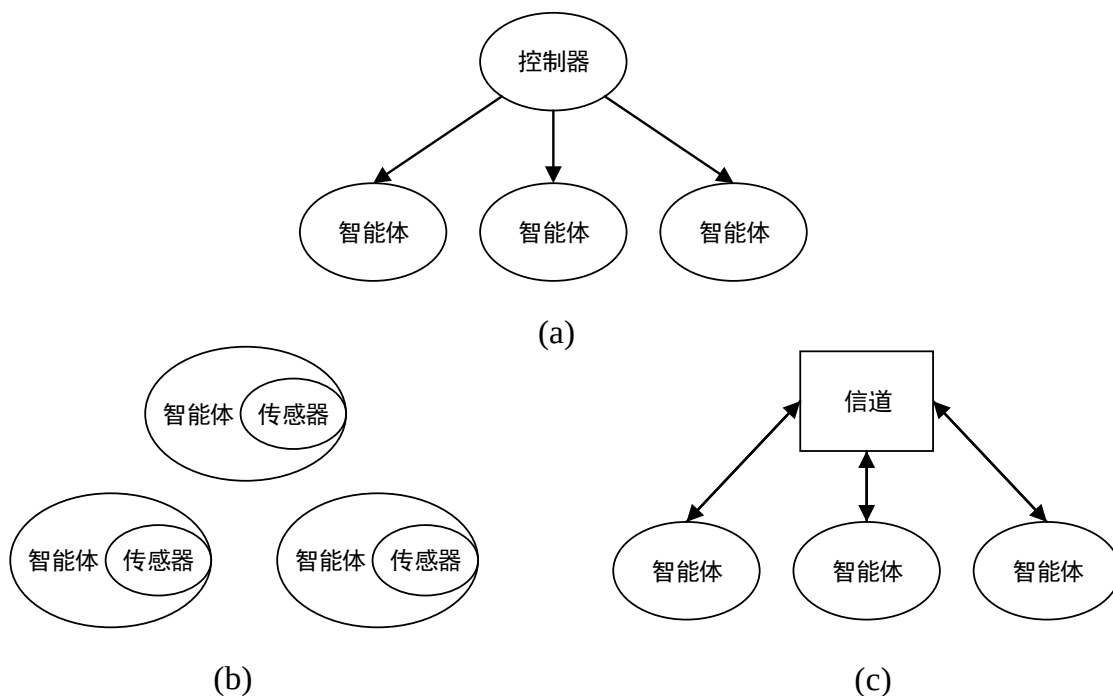


图 2-2 常见控制结构

(1) 集中式控制(Centralized Control)

集中式控制是指将控制系统中的决策和执行权集中在一个中央控制器中，中央控制器通过与各个智能体的通信进行控制和协调，控制结构如图 2-2(a)所示。在这种结构下，中央控制器负责所有决策和执行，因此对中央控制器的可靠性要求较高，同时也存在单点故障的风险。由于决策和执行都由中央控制器完成，因此集中式控制通信集中，易造成拥塞，从而影响效率，不适用于大规模的系统。

(2) 分散式控制(Decentralized Control)

分散式控制是指将控制系统中的决策和执行权分散到各个智能体上，控制结构如图 2-2(b)所示。在这种结构下，每个智能体具有相同的环境感知能力，可以独立地完成一部分任务，相互之间没有太多的交互。因此，尽管智能体自主性强，但是只能完成相对简单的任务。

(3) 分布式控制(Distributed Control)

分布式控制是介于分散式控制和集中式控制之间的一种控制结构，它将控制系统中的决策和执行权分散到多个控制器中，这些控制器之间通过通信进行协调和控制，控制结构如图 2-2(c)所示。在这种结构下，各个控制器之间具有一定的自治性，可以独立地完成一部分任务，同时也可以通过通信协调完成整个系统的任务。

综上所述，分布式控制结构简单，信息量少，可以提高整个系统的可靠性和灵活性，同时也可以降低单个控制器的负载。因此，在多智能体系统的协调与控制中，使用分布式控制结构更具优势。

2.3 交替方向乘子法

交替方向乘子法(Alternating Direction Method of Multipliers, ADMM)适用于求解分布式凸优化问题,通过分解协调过程,将一个复杂、计算量大的全局问题分解为多个较小、较容易求解的局部子问题。为了后续章节中 ADMM 方法的应用,将 ADMM 方法作为一种分布式优化的方法,进行说明:

$$\begin{aligned} \min_{x_i} \quad & \sum_{i=1}^S f_i(x_i) \\ \text{s.t.} \quad & x_i - E_i z = 0, \quad i=1, \dots, S \end{aligned}$$

其中 $f_i(x_i)$ 都是凸函数, x_i 是局部变量, $z = [z_1^T, \dots, z_S^T]^T$ 是公共全局变量, E_i 为全局变量与局部变量之间的映射矩阵。约束条件是使局部变量与对应的全局变量趋于一致,局部变量与全局变量的映射关系如图 2-3 所示。

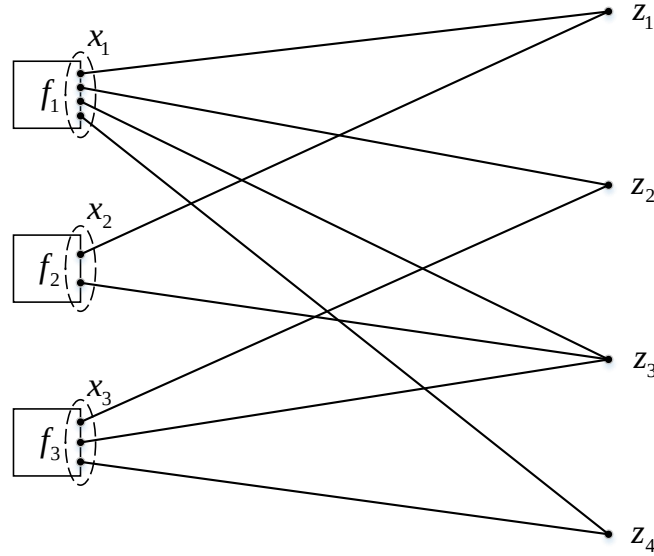


图 2-3 映射关系

构造上述问题的增广拉格朗日函数:

$$L_{\rho}(x_1, \dots, x_N, z, \mu) = \sum_{i=1}^N \left(f_i(x_i) + \mu_i^T (x_i - E_i z) + \frac{\rho}{2} \|x_i - E_i z\|^2 \right)$$

得到的 ADMM 方法如下:

$$\begin{aligned} x_i^{g+1} &= \arg \min_{x_i} \left(f_i(x_i) + (\mu_i^g)^T (x_i - E_i z^g) + \frac{\rho}{2} \|x_i - E_i z^g\|^2 \right) \\ z^{g+1} &= \arg \min_z \left((\mu_i^g)^T (x_i^{g+1} - E_i z) + \frac{\rho}{2} \|x_i^{g+1} - E_i z\|^2 \right) \\ \mu_i^{g+1} &= \mu_i^g + \rho (x_i^{g+1} - E_i z^{g+1}) \end{aligned}$$

按照上述三步迭代的步骤,重复直至变量不再发生变化,即最终收敛。

具体有关 ADMM 方法的收敛性分析详见^[60],本文不多介绍。

2.4 本章小结

本章介绍了论文中所要用到的多智能体系统的拓扑结构、迭代学习控制方法、多智能体系统常见控制结构以及交替方向乘子法，为后续的研究提供了理论基础。

第三章 分布式多智能体系统点对点迭代学习控制

3.1 引言

近几十年来,多智能体系统在理论研究与应用领域都获得了广泛关注,并且取得了许多进展。到目前为止,使用迭代学习控制解决多智能体系统一致性问题的研究成果并不多,并且大多考虑的是连续系统^[17,51-53,57-58]。在连续时间控制系统中,智能体之间的信息发送与接受是不受限制的,但在实际系统中,采用连续时间控制会给网络通信带来较大的负担。从实现角度来看,使用离散时间控制算法在实际应用中更容易实现,智能体在固定时间间隔内发送和接收信息,这样可以减轻网络通信的压力。由于实际情况中存在的设备老化、测量偏差等原因,系统往往存在不确定性,即实际的系统模型与标称系统模型存在偏差,这些模型不确定性可能会导致被控系统在实际运行中不可预测的行为,从而造成系统稳定性降低、控制精度下降、安全性等问题,因此鲁棒性分析必不可少。

3.2 问题描述

3.2.1 系统描述

考虑一类周期性运行特性的线性离散多智能体系统,是由 S 个具有相同输入输出数量的智能体系统构成

$$\begin{cases} x_{i,k}(t+1) = A_i x_{i,k}(t) + B_i u_{i,k}(t) \\ y_{i,k}(t) = C_i x_{i,k}(t) \end{cases} \quad (3.1)$$

其中 $1 \leq i \leq S$ 表示为第 i 个智能体, $k \in \mathbb{N}$ 表示迭代批次, $t \in [0, N]$ 表示一个运行周期时间 T 内的采样时刻, N 表示采样点个数; $x_{i,k}(t) \in \mathbb{R}^n$, $u_{i,k}(t) \in \mathbb{R}^m$, $y_{i,k}(t) \in \mathbb{R}^l$ 分别为第 i 个智能体在第 k 批次时的状态量、输入信号以及输出信号。 A_i, B_i, C_i 分别为第 i 个智能体的系统参数矩阵,并且具有相应维度。系统(3.1)的解可以表示为

$$y_{i,k}(t) = \sum_{\sigma=0}^{t-1} C_i A_i^{t-1-\sigma} B_i u_{i,k}(\sigma) + C_i A_i^t x_{i,k}(0) \quad (3.2)$$

为保证系统输出的可控性,需要满足 $C_i B_i \neq 0$ 。 $x_{i,k}(0)$ 表示第 i 个智能体在第 k 批次时的初始状态值。假设每个智能体在不同批次的初始状态值相同,即 $x_{i,k}(0) = x_{i,0}$,不失一般性,设置 $x_{i,0} = 0$ 。因为 ILC 是在迭代域进行设计制定的,因此可以利用提升技术对系统进行提升并描述为超向量的形式。针对式(3.1)形式的线性离散多智能体系统,可用提升矩阵模型表示:

$$\begin{cases} y_{i,k} = G_i u_{i,k} \\ y_{i,k} = [y_{i,k}(1)^T, y_{i,k}(2)^T, \dots, y_{i,k}(N)^T]^T \in \mathbb{R}^{lN} \\ u_{i,k} = [u_{i,k}(0)^T, u_{i,k}(1)^T, \dots, u_{i,k}(N-1)^T]^T \in \mathbb{R}^{mN} \end{cases} \quad (3.3)$$

其中, $y_{i,k}, u_{i,k}$ 分别为第 i 个智能体的输出信号与输入信号, $G_i \in \mathbb{R}^{lN \times mN}$ 为对应的映射矩阵, 且满足:

$$G_i = \begin{bmatrix} C_i B_i & 0 & 0 & \cdots & 0 \\ C_i A_i B_i & C_i B_i & 0 & \cdots & 0 \\ C_i A_i^2 B_i & C_i A_i B_i & C_i B_i & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ C_i A_i^{N-1} B_i & C_i A_i^{N-2} B_i & C_i A_i^{N-3} B_i & \cdots & C_i B_i \end{bmatrix} \quad (3.4)$$

给定多智能体系统的期望轨迹为 $r = [r(1)^T, r(2)^T, \dots, r(N)^T]^T \in \mathbb{R}^{lN}$, 并将期望轨迹看作是领导者智能体。将领导者智能体的编号标记为 0, 系统中的其他智能体分别标记为 $1, 2, \dots, S$ 。在通信拓扑结构中, 只有有限个智能体可以直接从领导者智能体获取信息, 用对角矩阵 $P = \text{diag}\{p_{11}, \dots, p_{SS}\}$ 来表示, $p_{ii} = 1$ 表示第 i 个智能体可以直接从领导者智能体获取信息, $p_{ii} = 0$ 则表示不行。其他智能体只能从邻接智能体处获取信息, 网络通信拓扑结构如图 3-1 所示。

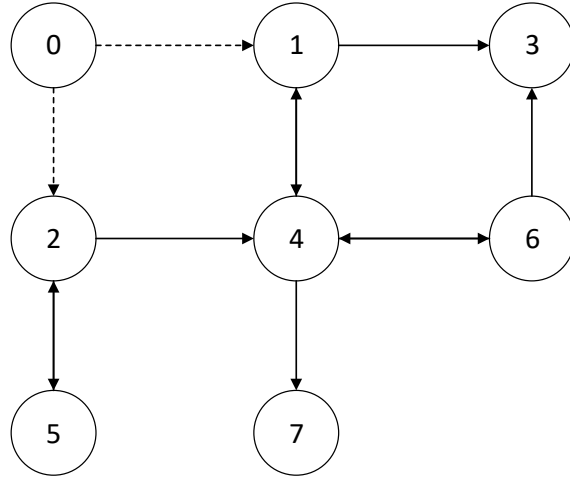


图 3-1 通信拓扑结构图

3.2.2 点对点跟踪任务中的一致性问题

在传统的多智能体系统迭代学习控制中, 考虑到智能体之间的拓扑结构, 只有有限个智能体可以直接获取参考轨迹信息, 因此需要定义其一致性误差为:

$$\xi_{i,k} = \sum_{j=1}^S a_{ij} (y_{j,k} - y_{i,k}) + p_{ii} (r - y_{i,k}) \quad (3.5)$$

其中, $\mathcal{A} = (a_{ij}) \in \mathbb{R}^{S \times S}$ 为拓扑图 $\mathcal{G}$ 的邻接矩阵。

定义 3.1 (多智能体系统 ILC 算法设计问题): 对单个智能体的输入信号 $u_{i,k}$ 进行修正, 使其收敛到唯一的期望值 u_i^* , 智能体系统的一致性误差 $\xi_{i,k}$ 收敛到 0, 那么相应的输出轨迹 $y_{i,k}$ 最终在有限时间范围 $[1, N]$ 内跟踪上给定的参考轨迹 r , 即

$$\begin{aligned} u_{i,k+1} &= f(u_{i,k}, \xi_{i,k}) \\ \lim_{k \rightarrow \infty} u_{i,k} &= u_i^*, \lim_{k \rightarrow \infty} \xi_{i,k} = 0, \lim_{k \rightarrow \infty} y_{i,k} = r \end{aligned} \quad (3.6)$$

注释 3.1 单个智能体实际的跟踪误差可以定义为 $e_{i,k} = r - y_{i,k}$ ，但是根据网络通信拓扑结构，只有有限个智能体可以直接获取参考轨迹信息，所以对于单个智能体来说，并不能将实际的跟踪误差 $e_{i,k}$ 作为控制律设计中的误差更新项。

在点对点跟踪任务中，只需要在特定时间跟踪上某些关键点的位置即可，对于完整参考轨迹 r 的剩余部分并没有跟踪的必要，因此传统的 ILC 设计就显得很多余。为了提高 ILC 设计的灵活度，本文提出了具体的多智能体系统点对点迭代学习控制框架。定义 M 个关键点处的跟踪时间为 $t_l, l=1, \dots, M$ ，满足以下条件：

$$0 < t_1 < t_2 < \dots < t_M \leq N \quad (3.7)$$

定义选择矩阵 $\Phi = \{\phi_{\alpha\beta}\}_{M \times N} \in \mathbb{R}^{M \times N}$ ，其每个单元可以定义为：

$$\phi_{\alpha\beta} = \begin{cases} 0_{l \times l}, \beta \neq t_\alpha \\ I_{l \times l}, \beta = t_\alpha \end{cases} \quad (3.8)$$

通过选择矩阵 Φ ，可以把信号转化为对应的点对点形式。在式(3.3)的基础上，可以推导出智能体的点对点系统动力学方程为：

$$\begin{cases} y_{i,k}^p = G_i^p u_{i,k} \\ y_{i,k}^p = [y_{i,k}(t_1)^T, y_{i,k}(t_2)^T, \dots, y_{i,k}(t_M)^T]^T \in \mathbb{R}^{IM} \\ u_{i,k} = [u_{i,k}(0)^T, u_{i,k}(1)^T, \dots, u_{i,k}(N-1)^T]^T \in \mathbb{R}^{mN} \end{cases} \quad (3.9)$$

其中， $G_i^p = \Phi G_i$ ， $y_{i,k}^p$ 为第 i 个智能体的点对点输出信号。点对点参考轨迹 r^p 是从完整参考轨迹 r 中提取的，如下所示：

$$r^p = [r(t_1)^T, r(t_2)^T, \dots, r(t_M)^T]^T \in \mathbb{R}^{IM} \quad (3.10)$$

点对点任务中的一致性误差为：

$$\xi_{i,k}^p = \sum_{j=1}^S a_{ij} (y_{j,k}^p - y_{i,k}^p) + p_{ii} (r^p - y_{i,k}^p) \quad (3.11)$$

定义 3.2 (多智能体系统点对点 ILC 算法设计问题)：针对点对点系统(3.9)，对单个智能体的输入信号 $u_{i,k}$ 进行修正，使其收敛到唯一的期望值 u_i^* ，智能体系统的点对点一致性误差 $\xi_{i,k}^p$ 收敛到 0，那么相应的点对点输出轨迹 $y_{i,k}^p$ 最终在有限时间范围 $[1, N]$ 内跟踪上给定的点对点参考轨迹 r^p ，即

$$\begin{aligned} u_{i,k+1} &= f(u_{i,k}, \xi_{i,k}^p) \\ \lim_{k \rightarrow \infty} u_{i,k} &= u_i^*, \lim_{k \rightarrow \infty} \xi_{i,k}^p = 0, \lim_{k \rightarrow \infty} y_{i,k}^p = r^p \end{aligned} \quad (3.12)$$

注释 3.2 对注释 3.1 进行补充说明，单个智能体实际的点对点跟踪误差 $e_{i,k}^p = r^p - y_{i,k}^p$ 被点对点一致性误差 $\xi_{i,k}^p$ 代替作为控制律(3.12)设计中的误差更新项。

为了便于接下来内容的分析，针对系统中含有的 S 个智能体，需要提前对一些变量进行定义。将 U_k, Y_k^p, R^p, E_k^p 分别记为多智能体系统整体的控制输入、点对点输出轨迹、期望的点对点输出轨迹以及对应的点对点跟踪误差：

$$\begin{cases} U_k = [u_{1,k}^T, u_{2,k}^T, \dots, u_{S,k}^T]^T \\ Y_k^p = [(y_{1,k}^p)^T, (y_{2,k}^p)^T, \dots, (y_{S,k}^p)^T]^T \\ R^p = [(r^p)^T, (r^p)^T, \dots, (r^p)^T]^T \\ E_k^p = [(e_{1,k}^p)^T, (e_{2,k}^p)^T, \dots, (e_{S,k}^p)^T]^T \\ G^p = \text{diag}(G_1^p, G_2^p, \dots, G_S^p) \end{cases} \quad (3.13)$$

G^p 是分块对角矩阵，包含了每个智能体的系统矩阵信息，并且也是一个输入到输出的映射矩阵，存在对应关系 $Y_k^p = G^p U_k$ 。

3.3 分布式点对点算法设计

针对多智能体系统点对点跟踪任务，本文提出一种基于梯度下降的分布式 ILC 控制算法：

$$u_{i,k+1} = u_{i,k} + \gamma_i (G_i^p)^T \xi_{i,k}^p \quad (3.14)$$

其中 γ_i 为控制增益。

算法 3.1 分布式点对点 ILC 算法

给定系统参数矩阵 A_i, B_i, C_i ，期望的点对点参考轨迹 r^p ，关键跟踪点处的时间为 $t_i, i=1, \dots, M$ ，拉普拉斯矩阵 L ，对角矩阵 P ，控制增益 γ_i ，最大的 ILC 迭代次数 $k_{\max}$ 。

- 1: **Initialization:** 设置 $k=0$
 - 2: **repeat**
 - 3: 设置 $k=k+1$
 - 4: 用分布式 ILC 控制算法(3.14)优化 $u_{i,k+1}$
 - 5: 将优化控制输入 $u_{i,k+1}$ 分别作用于单个智能体
 - 6: **until** $k=k_{\max}$
 - 7: **return** 每个智能体的优化控制输入 $u_{i,k_{\max}}$
-

注释 3.3 定义一个有关误差项的性能指标函数 $\hat{f}(u_{i,k}) = \|\xi_{i,k}^p\|^2$ ，对 $u_{i,k}$ 进行优化，可以得到表达式 $\nabla_{u_{i,k}} \hat{f}(u_{i,k}) = -2 \left(\sum_{j=1}^S a_{ij} + p_{ii} \right) (G_i^p)^T \xi_{i,k}^p$ 。那么基于梯度下降的分布式 ILC 控制律可以写成 $u_{i,k+1} = u_{i,k} - \frac{\hat{\gamma}_i}{2} \nabla_{u_{i,k}} \hat{f}(u_{i,k})$ 的形式，对其进行简化，得到了控制算法(3.14)。

定理 3.1 对于系统(3.9)，使用分布式点对点算法(算法 3.1)，以下充分必要的收敛条件成立：

$$\left\| \left(I_{IMS} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{IM}) \right)^{-1} \right\| > 1 \Leftrightarrow \lim_{k \rightarrow \infty} E_k^p = 0 \quad (3.15)$$

其中, $\Upsilon = \text{diag}(\gamma_1, \gamma_2, \dots, \gamma_S)$ 。当 Υ 满足上述条件时, 由算法 3.1 生成的实际点对点跟踪误差 $\{E_k^p\}$ 收敛到 0。

证明: 根据注释 3.2 中智能体实际点对点跟踪误差 $e_{i,k}^p$ 的定义, 可以将智能体系统的点对点一致性误差(3.11)描述为

$$\xi_{i,k}^p = \sum_{j=1}^S a_{ij} (e_{i,k}^p - e_{j,k}^p) + p_{ii} e_{i,k}^p \quad (3.16)$$

由此可以得到

$$\xi_k^p = ((L+P) \otimes I_{lm}) E_k^p \quad (3.17)$$

其中, $\xi_k^p = \left[(\xi_{1,k}^p)^T, (\xi_{2,k}^p)^T, \dots, (\xi_{S,k}^p)^T \right]^T$ 。那么由式(3.17)结合 ILC 控制律(3.14), 可以将控制输入描述为

$$U_{k+1} = U_k + (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{lm}) E_k^p \quad (3.18)$$

因此多智能体系统整体点对点的输出轨迹为

$$Y_{k+1}^p = Y_k^p + G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{lm}) E_k^p \quad (3.19)$$

■ 充分性:

通过考虑第 $k+1$ 批次时的整体点对点跟踪误差, 进一步分析系统的收敛性

$$\begin{aligned} E_{k+1}^p &= R^p - Y_{k+1}^p \\ &= \left(I_{lms} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{lm}) \right) E_k^p \end{aligned} \quad (3.20)$$

两边取范数

$$\|E_{k+1}^p\| \leq \|I_{lms} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{lm})\| \|E_k^p\| \quad (3.21)$$

于是, 得到了充分性的收敛条件

$$\|I_{lms} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{lm})\| < 1 \quad (3.22)$$

如果矩阵 $(I_{lms} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{lm}))$ 是满秩的, 则

$$\begin{aligned} \|I_{lms}\| &\leq \|I_{lms} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{lm})\| \left\| \left(I_{lms} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{lm}) \right)^{-1} \right\| \\ &< \left\| \left(I_{lms} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{lm}) \right)^{-1} \right\| \end{aligned}$$

因此, 进一步得到了收敛条件

$$\left\| \left(I_{lms} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{lm}) \right)^{-1} \right\| > 1 \quad (3.23)$$

■ 必要性:

收敛条件(3.23)的必要性证明是通过反证法, 假设 Υ 满足以下条件

$$\left\| \left(I_{IMS} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{IM}) \right)^{-1} \right\| \leq 1 \quad (3.24)$$

通过改写式(3.20)得

$$E_k^p = \left(I_{IMS} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{IM}) \right)^{-1} E_{k+1}^p \quad (3.25)$$

可以得到

$$\begin{aligned} \|E_k^p - E_{k-1}^p\| &= \left\| \left(I_{IMS} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{IM}) \right)^{-1} (E_{k+1}^p - E_k^p) \right\| \\ &\leq \left\| \left(I_{IMS} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{IM}) \right)^{-1} \right\| \|E_{k+1}^p - E_k^p\| \end{aligned} \quad (3.26)$$

上式(3.26)通过归纳法可以得到

$$\begin{aligned} \|E_1^p - E_0^p\| &\leq \left\| \left(I_{IMS} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{IM}) \right)^{-1} \right\|^k \|E_{k+1}^p - E_k^p\| \\ &\leq \|E_{k+1}^p - E_k^p\| \end{aligned} \quad (3.27)$$

显然 $\|E_{k+1}^p - E_k^p\| \geq \|E_1^p - E_0^p\| \geq 0$ 。所以，当不满足收敛条件(3.23)时，实际的点对点跟踪误差 E_k^p 会随着 k 的增加而发散，证毕。

注释 3.4 实际情况中会发现充分性的收敛条件(3.22)并不够严格，因为在很多情况下，即使不满足充分性的收敛条件，系统仍然能够保证收敛，所以进一步研究其充分必要的收敛条件是非常有必要的。

3.4 鲁棒性分析

由于实际情况中存在的设备老化、测量偏差等原因，系统往往存在不确定性，即实际的系统模型 $\tilde{G}_i$ 与标称系统模型 G_i 存在偏差，这些模型不确定性可能会导致被控系统在实际运行中不可预测的行为，从而造成系统稳定性降低、控制精度下降、安全性等问题，因此鲁棒性分析必不可少。假设实际的系统模型 $\tilde{G}_i$ 具有如文献[47]中的乘性不确定性

$$\tilde{G}_i = G_i (I_{mN} + \Delta_i) \quad (3.28)$$

其中未知矩阵 $\Delta_i \in \mathbb{R}^{mN \times mN}$ 表示模型不确定性，并且满足 $\|\Delta_i\| \leq 1$ 。通过选择矩阵 Φ 可以得到实际的点对点输出与控制输入的关系为 $\tilde{y}_{i,k}^p = \tilde{G}_i^p u_{i,k}$ ，其中 $\tilde{G}_i^p = \Phi \tilde{G}_i$ 。则实际的多智能体系统整体点对点的输出轨迹与控制输入的关系可以表示为 $\tilde{Y}_k^p = \tilde{G}^p U_k$ ，映射关系矩阵 $\tilde{G}^p = \text{diag}(\tilde{G}_1^p, \tilde{G}_2^p, \dots, \tilde{G}_S^p)$ 。

定理 3.2 对于具有乘性不确定性的系统(3.28)，使用分布式点对点算法(算法 3.1)，以下充分必要的鲁棒收敛条件成立：

$$\left\| \left(I_{IMS} - G^p (I_{mNS} + \Delta) (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{IM}) \right)^{-1} \right\| > 1 \Leftrightarrow \lim_{k \rightarrow \infty} E_k^p = 0 \quad (3.29)$$

其中, $\Delta = \text{diag}(\Delta_1, \Delta_2, \dots, \Delta_s)$ 。当 Δ 满足上述条件时, 由算法 3.1 生成的实际点对点跟踪误差 $\{E_k^p\}$ 收敛到 0。

证明: 将算法 3.1 应用到实际系统模型 $\tilde{G}_i$ 中, 可以将控制输入描述为

$$U_{k+1} = U_k + (\Upsilon \otimes I_{mN})(G^p)^T ((L+P) \otimes I_{IM}) E_k^p \quad (3.30)$$

此时, 实际的多智能体系统整体点对点误差 $E_k^p = R^p - \tilde{Y}_k^p$, 因此实际系统的整体点对点输出轨迹为

$$\tilde{Y}_{k+1}^p = \tilde{Y}_k^p + \tilde{G}^p (\Upsilon \otimes I_{mN})(G^p)^T ((L+P) \otimes I_{IM}) E_k^p \quad (3.31)$$

通过考虑系统第 $k+1$ 批次时的实际系统点对点跟踪误差, 进一步分析系统的收敛性

$$E_{k+1}^p = (I_{IMS} - \tilde{G}^p (\Upsilon \otimes I_{mN})(G^p)^T ((L+P) \otimes I_{IM})) E_k^p \quad (3.32)$$

两边取范数

$$\|E_{k+1}^p\| \leq \|I_{IMS} - \tilde{G}^p (\Upsilon \otimes I_{mN})(G^p)^T ((L+P) \otimes I_{IM})\| \|E_k^p\| \quad (3.33)$$

于是, 得到了充分性的收敛条件

$$\|I_{IMS} - G^p (I_{mNS} + \Delta)(\Upsilon \otimes I_{mN})(G^p)^T ((L+P) \otimes I_{IM})\| < 1 \quad (3.34)$$

接下来必要性的证明步骤同定理 3.1, 可以得到充分必要的鲁棒收敛条件(3.29), 证毕。

3.5 仿真结果与分析

为验证本章分布式点对点 ILC 算法的可行性, 以具有周期性运行特性的鸚鵡 AR 无人机^[61]为仿真对象, 考虑其在 z 轴上的性能。如图 3-2 所示, 鸚鵡 AR 无人机实验测试台由一个动作捕捉系统、四个无人机、无线和有线通信通道以及一台地面站计算机组成。

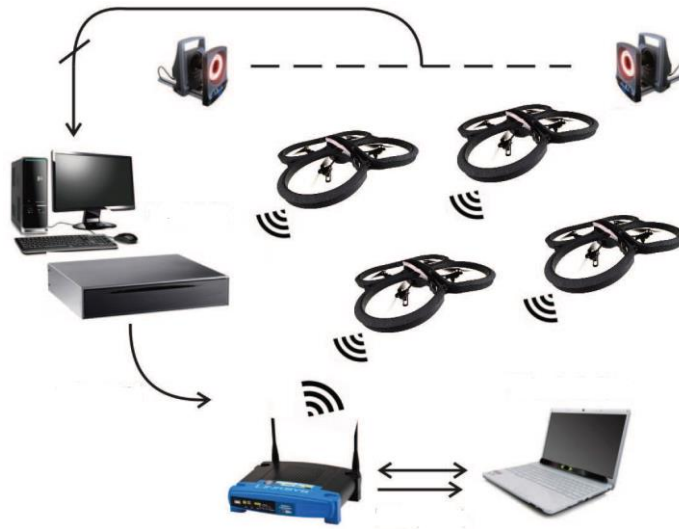


图 3-2 鸚鵡 AR 无人机实验测试台

单个无人机的控制输入 U_z 与 z 轴上的速度 $\dot{z}$ 直接相关, 传递函数表示的数学模型如

下

$$\frac{z(s)}{U_z(s)} = \frac{k_n \omega_n^2}{s^3 + 2\zeta \omega_n s^2 + \omega_n^2 s} \quad (3.35)$$

按照用于偏航模型识别的类似方法, 分别设置参数 $k_n = 775.3207$, $\omega_n = 4.5994$, $\zeta = 0.7218$ 。因此, 状态空间中 z 轴上高度的动态模型如下

$$\begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -21.1544 & -6.6396 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ \ddot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 16401.4442 \end{bmatrix} U_z \quad (3.36)$$

使用零阶保持器进行离散化, 并且设置采样时间 $T_s = 0.1s$, 那么每个无人机的离散状态空间模型矩阵可以表示如下

$$A = \begin{bmatrix} 1 & 0.0970 & 0.0040 \\ 0 & 0.9156 & 0.0705 \\ 0 & -1.4923 & 0.4473 \end{bmatrix}, B = \begin{bmatrix} 2.3108 \\ 65.3999 \\ 1.1570 \end{bmatrix}, C = [1 \ 0 \ 0] \quad (3.37)$$

不失一般性, 设置每个无人机的初始控制输入为 0, 即 $u_{i,0} = 0$, 单位为伏特(V)。

给定系统的跟踪控制任务为: 运行周期 $T = 10s$ 内, 即 $N = 100$, 使用模型的 z 轴跟踪 5 个关键点($M = 5$), 其中点对点参考轨迹设置为

$$r^p = [1, 7, -4, -6, 3]^T \quad (3.38)$$

单位为米(m), 并且跟踪时间设置为

$$t_1 = 5, t_2 = 25, t_3 = 45, t_4 = 65, t_5 = 85 \quad (3.39)$$

单位为秒(s)。

无人机之间的通信拓扑结构如图 3-3 所示, 将期望轨迹的信息 r^p 记为编号为 0 的领导者智能体。

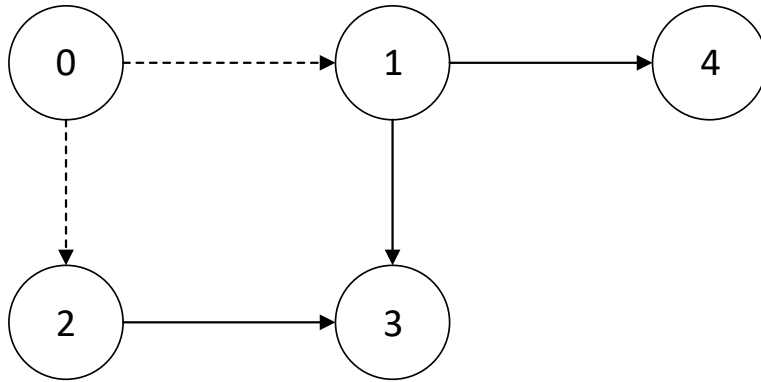


图 3-3 无人机通信拓扑结构

由图 3-3 中所示的无人机通信拓扑结构可以看出, 只有编号为 1 和 2 的无人机可以直接获取到领导者智能体 0 的信息, 即对角矩阵 $P = \text{diag}\{1, 1, 0, 0\}$ 。同时也可以得到无人机之间通信拓扑结构图 $\mathcal{G}$ 的拉普拉斯矩阵, 表示如下:

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} \quad (3.40)$$

将所提分布式点对点 ILC 算法(算法 3.1)应用于上述的控制任务中,分别设置权重增益 $\gamma_1 = 1.4 \times 10^{-6}$, $\gamma_2 = 1.6 \times 10^{-6}$, $\gamma_3 = 7.2 \times 10^{-7}$, $\gamma_4 = 1.8 \times 10^{-6}$, 总共进行 80 次试验,即 $k = 80$ 。

在进行仿真过程中,可以得到 $\left\| \left(I_{IMS} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{IM}) \right)^{-1} \right\| = 6.6772 > 1$,

满足充要条件(3.23)。可以得到 $\left\| I_{IMS} - G^p (\Upsilon \otimes I_{mN}) (G^p)^T ((L+P) \otimes I_{IM}) \right\| = 2.2194 > 1$, 并不满足必要性条件(3.22)。因此,有必要按照注释 3.4 的补充,从充分性的条件出发,进一步推导出其充分必要的收敛条件。

图 3-4 到图 3-7 给出了系统中 4 个无人机分别随迭代批次变化的输出情况,可以看出四个无人机迭代至第 80 批次时,基本实现收敛。根据通信拓扑结构 3-3,第 1 个无人机和第 2 个无人机并不受其他无人机的影响,它们的信息来源只有编号为 0 的领导者智能体,所以对于第 1 个无人机和第 2 个无人机的控制是等价的,它们控制效果的不同是因为权重控制增益 γ_1, γ_2 的不同而导致的。

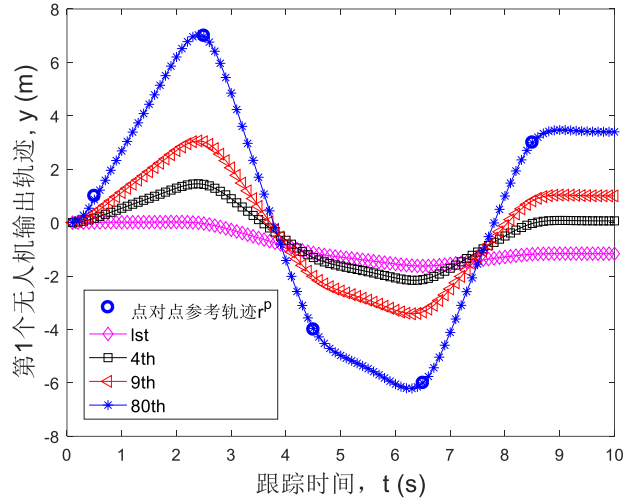


图 3-4 第 1 个无人机输出轨迹

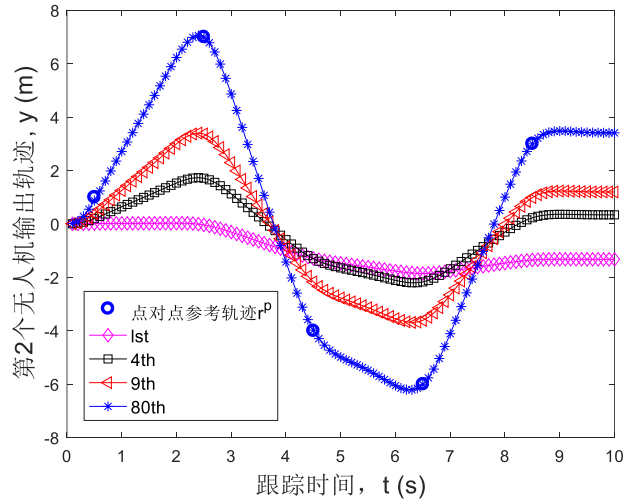


图 3-5 第 2 个无人机输出轨迹

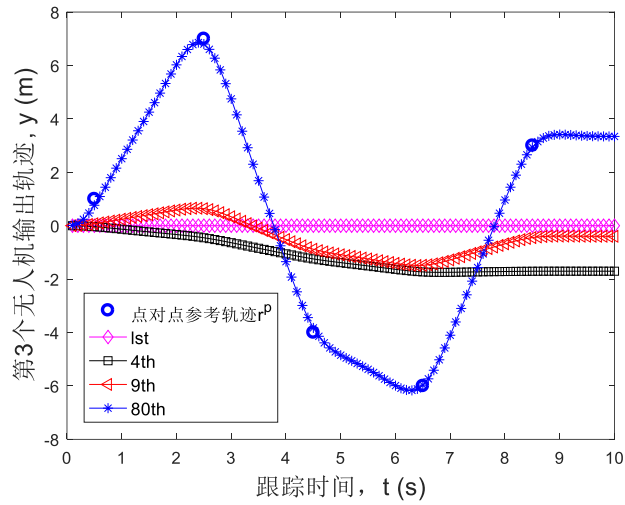


图 3-6 第 3 个无人机输出轨迹

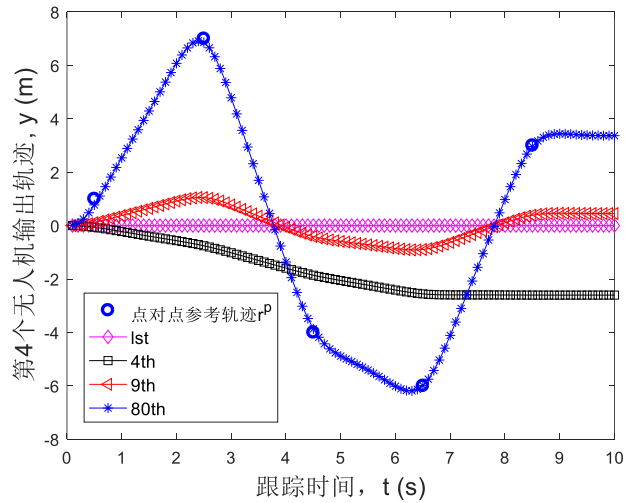


图 3-7 第 4 个无人机输出轨迹

为了直观地描述系统收敛的变化过程，图 3-8 给出了系统中 4 个无人机随迭代增加的跟踪误差范数的变化曲线。由于第 1 个无人机和第 2 无人机可以直接获取参考轨迹的

信息，这也就导致了第 1 个无人机和第 2 个无人机的收敛速度明显快于第 3 个无人机和第 4 个无人机。

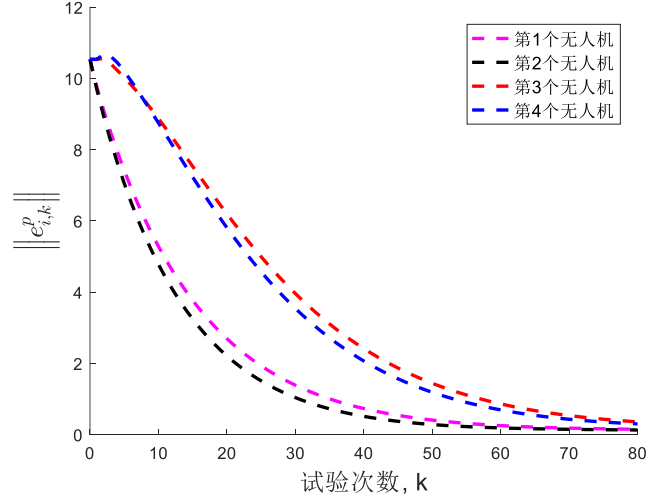


图 3-8 沿迭代批次轴每个无人机跟踪误差的变化

以上是所提算法 3.1 应用到标称系统下的结果，在实际情况中可能会存在模型不确定性，假设 $\|\Delta_1\| = 0.5571$ ， $\|\Delta_2\| = 0.3847$ ， $\|\Delta_3\| = 0.0481$ ， $\|\Delta_4\| = 0.2404$ ，并且满足鲁棒收敛条件(3.29)，分别在图 3-9 到图 3-12 给出了系统中 4 个无人机随迭代批次变化的输出情况。可以看到，即使系统中存在模型不确定性，但是在满足鲁棒收敛条件下，算法 3.1 仍然能够保证系统在有限批次内跟踪上所需要跟踪的关键点位置，从而体现了所提算法具有一定的鲁棒性。

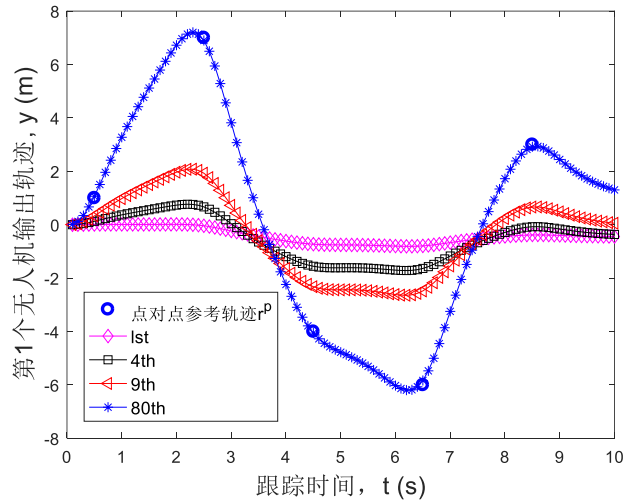


图 3-9 模型不确定性下第 1 个无人机输出轨迹

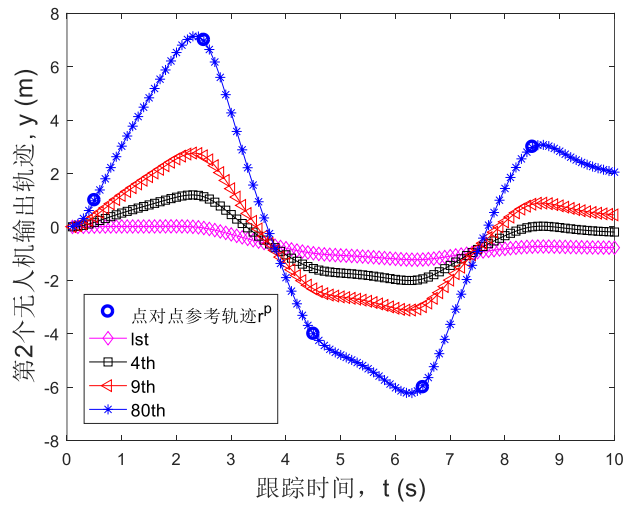


图 3-10 模型不确定性下第 2 个无人机输出轨迹

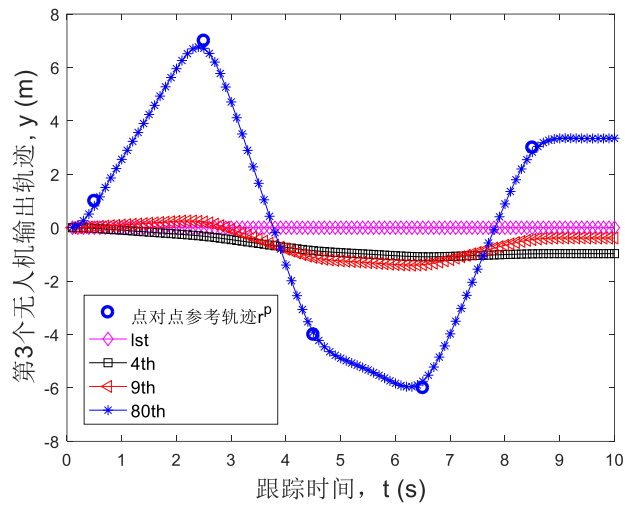


图 3-11 模型不确定性下第 3 个无人机输出轨迹

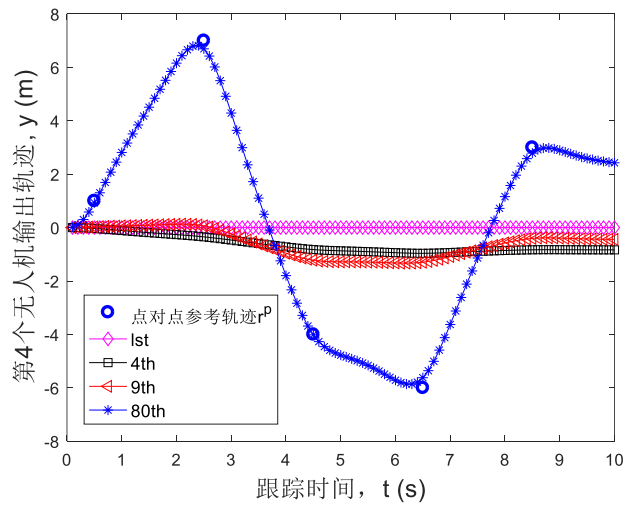


图 3-12 模型不确定性下第 4 个无人机输出轨迹

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/547115035130006045>