

基于深度微分方程网络的复杂动态系统建模与控制

袁兆麟

北京科技大学



密 级: 公开

论文题目: **基于深度微分方程网络的复杂动态系统建模  
与控制**

学 号: B20170324  
作 者: 袁兆麟  
专业名称: 计算机科学与技术

2022 年 10 月 20 日



基于深度微分方程网络的复杂动态系统建模与控制

Modeling and Controlling for Complicated Dynamical  
System based on Deep Differential Equation Network

研究生姓名：袁兆麟

指导教师姓名：班晓娟

北京科技大学计算机与通信工程学院

北京 100083，中国

Doctor Degree Candidate: Zhaolin Yuan

Supervisor: Xiaojuan Ban

School of Computer and Communication Engineering

University of Science and Technology Beijing

30 Xueyuan Road, Haidian District

Beijing 100083, P.R.CHINA



分类号: TP312 密 级: 公开  
U D C:            单位代码: 10008

# 北京科技大学博士学位论文

**论文题目:** 基于深度微分方程网络的复杂动态系统建模与控制

**作者:** 袁兆麟

**指 导 教 师:** 班晓娟 教授   **单 位:** 北京科技大学  
**指导小组成员:** 李宁 教授   **单 位:** 北京信息科技大学  
**论 文 提 交 期 限:** 2022 年 10 月 20 日  
**学 位 授 予 单 位:** 北京科技大学



## 致 谢

在北京科技大学攻读博士的五年半对我来说意义深远。在这个过程中，我积累了知识、锻炼了心性、磨练了意志，知道了如何与人合作，明白了如何作为一个平凡的人参与伟大的事业。而我所获得的一切离不开老师、朋友、家人对我的帮助与支持。值此论文完成之际，对所有帮助过我的人表示感谢。

首先，最需要感谢的人是我的导师班晓娟教授。班老师除了在专业知识以及科研方法上提供了充足的指导外，为我创造了得天独厚的研究环境以及高自由度的发展空间，让我能够倚着自己的兴趣和好奇心探索科学的边界。另外班老师的那句“读博士，不光是写几篇论文，而是全方位的锻炼与培养”更是我读博期间始终铭记于心的教诲，我也因此受益良多。

同样要感谢的是香港浸会大学的戴弘宁教授、挪威科技大学的王浩教授、吴狄教授、以及 Miratlas SAS 的王也弯老师。他们待我如同自己的亲学生，我在学术道路上的成长离不开各位老师不遗余力的帮助与指导。

还要感谢人工智能与三维可视化实验室的各位老师和同学。感谢姚超老师在研究方案设计、论文写作、毕业论文选题等方面提供的全方位指导。感谢张雅澜、马博渊、刘斯诺等师兄师姐给我树立了优秀的标杆指引我前进，并在我毕业、择业之际分享了宝贵的经验。同时感谢曹宇宁、何润姿、胡金龙、刘婷、李佳、周佳城、李潇睿、韩方圆、张子轩等一起参与过采矿充填项目的各位同学、师弟师妹们，跟你们一起去非洲出差、一起讨论项目实施方案、一起通宵达旦赶项目进度，是我永生难忘的日子。感谢实验室所有的老师同学，与你们交流、讨论让我学到无数宝贵的知识，因为你们让我在实验室的 6 年时间变得丰富多彩。

同时要感谢卢东旭、苏日娜、秦运慧、郑远硕四位博士，一年多来与你们在一起交流、吃饭、娱乐，让我远离孤单，真切地感受到了友情带给人的温馨与快乐。

最后还要感谢我的父母给我无条件的支持与关爱，让我没有任何后顾之忧地追求自己的学业。你们在我获得成就时陪伴我一起喝彩，在我最困难的时期作为我最坚实的后盾，我爱你们。

感谢非洲矿业有限公司沈家华、肖金林、姚松、王京伟等同仁在数据采集、算法测试期间提供的帮助。

感谢各位评审老师的认真评阅以及所提出的宝贵意见。

本论文承蒙国家重点研发计划项目 (No.2019YFC0605300) 资助，特此致谢。



## 摘要

采矿、化工、能源等过程工业领域中存在诸多具有非线性、高随机、长时滞、不完全可观测特性的复杂动态系统。依赖于机理分析的传统动态系统辨识方法难以充分拟合系统的高复杂性。依靠大量训练数据驱动的参数化建模方法难以引入系统机理和先验知识，使得训练数据需求量大，在不同场景下的泛化能力差。系统的高复杂性与现有建模方法的局限性造成了系统机理难分析、动态方程难辨识、关键指标难控制等问题。本文围绕连续时间域下复杂动态系统的建模与优化这一问题展开讨论，重点研究了连续时间系统与深度学习的结合体——深度微分方程网络在复杂系统辨识与控制优化中的应用。本文的主要研究工作包括：

- 1) 针对具有非线性、长时延特性且本质上为连续时间过程的复杂工业系统，本文提出一种以 ODE-net 作为骨架结构的深度连续时间系统辨识模型，模型能够从连续时间域角度学习工业系统输出的自回归变化和输入对输出的非线性影响。通过真实工业数据进行验证，该模型在长时预测和短时预测场景下均获得较好效果。
- 2) 针对具有周期多阶段特性的复杂动态系统，本文提出了一种连续时间自跳变常微分方程网络以学习连续时间周期跳变过程。模型能够从非均匀采样数据中独立学习不同阶段下的系统动态特性。同时模型引入阶段转换预测器以实现开环预测时的阶段自转移。利用某工业制冷系统数据对模型进行验证，该模型能够准确地预测系统输出量以及阶段转移时间点，并辅助优化制冷系统的运行能耗。
- 3) 针对具有随机非确定性的复杂动态系统，本文提出了常微分方程循环状态空间模型。该模型在状态转移中引入随机路径以建模系统的随机非确定性，同时利用常微分方程网络建模相邻采样时间点之间的隐变量连续时间演化以支持从非均匀采样数据学习。另外，本文提出了一种时间导数的再参数化方法以解决批数据中时间间隔不均匀导致难以并行求解的问题。结果表明该模型在数据集存在随机非确定性以及稀疏非均采样的情况下，能获得相比基线模型更好的建模效果。
- 4) 针对复杂工业系统控制优化问题，本文提出了一种基于有模型强化学习的启发式评价网络值迭代算法。该方法采用微分方程网络构建系统的预测模型，并采用积分强化学习构建在线控制策略及自适应评价函数。经过尾矿浓缩机仿真模型验证，该方法在非均匀数据采样下能够获得比其他控制算法更优的控制性能，且该算法成功部署应用于某矿场的浓密机控制系统中，获得了较好的浓度控制效果。

关键词：动态系统建模，系统辨识，常微分方程网络，有模型强化学习



## Abstract

In the process industry, most complex equipments or systems have strong non-linearity, high stochasticity, and long time delay. These properties lead to the difficulties in analyzing system mechanisms, identifying dynamical equations, and controlling critical operational indices. Mechanistic-based dynamic system identification methods are weak to sufficiently fit the complexity of the system, which raises the required size of training data and reduces the generalization in different scenarios. From the continuous-time domain, this paper focuses on the application of deep differential equation networks, a combination of continuous time systems and deep learning, in the identification and control optimization of complex systems.

For modeling non-linear and long-delayed complex industrial systems which are essentially continuous-time processes, this paper proposes a continuous-time deep system identification model based on the ODE-net backbone. The model learns the auto-regressive changes of industrial system outputs and the non-linear effects from inputs on outputs in continuous-time domain. By evaluating the model with real industrial dataset, the model performs better results in both long-time prediction and short-time prediction.

To address the problem of modeling periodic multi-staged system, this paper proposes a continuous-time autonomous jump ordinary differential equation network to learn continuous-time period jump systems. The model is able to independently learn the sub-dynamics in different stages from non-uniformly sampled data. The model also introduces a stage transition predictor to realize the autonomous stage transformation in open-loop prediction. By evaluating the model with data from an industrial cooling system, we find the model is accurate enough to predict the system outputs and the duration time in each stage. The open-loop prediction of cooling system runtime also assists in optimizing the energy consumption of the cooling system.

For modeling the complex systems with stochasticity and uncertainty, we propose a novel continuous-time SSM named Ordinary Differential Equation Recurrent State Space Model. ODE-RSSM incorporates an ordinary differential equation (ODE) network (ODE-Net) to model the continuous-time evolution of latent states between adjacent time points. Inspired from the equivalent linear transformation on integration limits, This paper also propose an efficient reparameterization method for solving batched ODEs with non-uniform

time spans in parallel for efficiently training the ODE-RSSM with irregularly sampled sequences. Extensive experiments demonstrate that the ODE-RSSM achieves better performance than other baselines in predicting stochastic and uncertain system in open loop when the time spans of predicted points are uneven.

For controlling complex industrial systems without prior knowledge of mechanism, this paper proposes a novel model-based reinforcement learning online control algorithm, heuristic critic network value iteration. The method models the dynamical system with differential equation neural network and introduces the integral reinforcement learning to train online control policy and adaptive critic function. Experiments conducted on thickening simulation system verify that the proposed HCNVI outperforms the other control algorithms. In the meanwhile, the method is also deployed in a thickening control system of a real paste backfilling station. The algorithm is effective to control the underflow concentration in a stable range.

**Key Words:** **Dynamical system modeling, Model-based reinforcement learning, System identification, Ordinary Differential Equation Neural Network**

## 目 录

致谢 .....	I
摘要 .....	III
Abstract .....	V
表格清单 .....	IX
插图清单 .....	XI
1 引言 .....	1
1.1 研究背景与意义 .....	1
1.2 本文研究的关键问题 .....	3
1.2.1 具有长时延、强非线性的复杂连续时间系统建模问题 .....	3
1.2.2 具有周期多阶性的连续时间跳变系统建模问题 .....	4
1.2.3 具有随机性、非确定性的复杂连续时间系统建模问题 .....	4
1.2.4 复杂工业系统的控制策略构建与自适应优化问题 .....	5
1.3 本文的研究内容 .....	5
1.3.1 基于可微 ODE-Net 的高时延复杂工业系统预测 .....	6
1.3.2 基于自跳跃-常微分方程网络的连续时间周期性跳变系统建模 ..	7
1.3.3 基于深度常微分方程-马尔可夫模型的随机非确定性系统建模 ..	7
1.3.4 基于连续时间深度强化学习的复杂工业系统优化与控制 .....	8
1.4 论文的章节安排 .....	8
2 文献综述 .....	11
2.1 动态系统辨识及有模型控制的起源与发展 .....	11
2.2 连续时间系统辨识 .....	12
2.3 微分方程网络在复杂系统建模中的应用 .....	13
2.3.1 非马尔科夫系统及长时延系统建模 .....	15
2.3.2 跳变系统辨识 .....	16
2.3.3 动态系统随机非确定性的表示与推理 .....	17
2.4 强化学习及其在控制优化中的应用 .....	18
2.5 本章小结 .....	20

3 基于可微 ODE-Net 的高时延工业多输入输出系统预测 .....	21
3.1 问题形式化描述 .....	22
3.2 基于 ODE-Net 的多输入输出系统预测模型设计 .....	24
3.2.1 基于循环神经网络的历史序列编码 .....	24
3.2.2 基于可微常微分方程网络构建系统状态空间模型 .....	25
3.2.3 面向不同预测时长需求的常微分方程导数模块定义 .....	26
3.3 离散输入序列的可微并行插值方法及模型训练 .....	28
3.4 实验验证与分析 .....	28
3.4.1 膏体浓密机系统数据集 .....	29
3.4.2 实验设定 .....	30
3.4.3 不同模型预测结果对比 .....	31
3.4.4 不同 ODE 求解器对比 .....	32
3.4.5 稳定系统与非稳定系统对比 .....	33
3.4.6 探究序列插值阶数的对预测精度的影响 .....	36
3.5 本章小结 .....	37
4 基于自跳跃常微分方程网络的连续时间跳变系统建模 .....	39
4.1 问题形式化描述 .....	40
4.2 自跳跃常微分方程网络及编码器-解码器预测架构 .....	41
4.2.1 基于层次 ODE-Net 模型的复杂系统稳定及非稳定输出混合建模 .....	41
4.2.2 自跳跃常微分方程网络 .....	43
4.2.3 基于编码器解码器结构的微分方程网络初值估计与序列预测 ..	44
4.3 模型训练 .....	46
4.4 实验验证与分析 .....	48
4.4.1 制冷系统简介 .....	48
4.4.2 数据预处理：离散序列插及阶段标注 .....	49
4.4.3 模型训练 .....	51
4.4.4 应用研究 1：运行变量仿真 .....	51
4.4.5 应用研究 2：制冷系统进气口温度设定点优化 .....	54
4.5 本章小结 .....	57
5 基于深度常微分马尔可夫模型的随机非确定性系统建模 .....	59
5.1 问题形式化描述 .....	60
5.2 常微分方程网络-循环状态空间模型 .....	61
5.2.1 连续时间域下含随机隐变量的生成模型 .....	61

5.3 隐变量推理与训练 .....	63
5.4 基于高效隐空间超调的多步预测性能改善 .....	64
5.5 非均匀采样间隔下的批量常微分方程并行化求解方法 .....	67
5.6 实验验证与分析 .....	69
5.6.1 数据集介绍 .....	69
5.6.2 实验设定 .....	70
5.6.3 基线模型选择 .....	70
5.6.4 相比于离散时间模型, ODE-RSSM 建模非均匀采样系统的效 果探究 .....	71
5.6.5 相比于带有确定性状态演化的连续时间模型, ODE-RSSM 建 模随机系统的优势探究 .....	72
5.6.6 隐空间超调对于模型长期预测性能的影响性探究 .....	73
5.6.7 ODE-RSSM 模型在不同采样间隔分布下的泛化能力探究 .....	73
5.6.8 批常微分方程并行化求解算法的时间效率探究 .....	74
5.7 本章小结 .....	75
6 基于连续时间有模型强化学习的复杂工业系统优化与控制 .....	77
6.1 本章引言 .....	77
6.2 连续时间强化学习形式化描述 .....	78
6.3 启发式评价网络值迭代算法 .....	80
6.3.1 基于常微分方程网络的系统模型构建 .....	80
6.3.2 基于积分强化学习的策略评价 .....	81
6.3.3 基于随机梯度下降的动作生成 .....	83
6.3.4 短期经验回放 .....	84
6.4 本文方法在仿真工业浓密机系统底流浓度控制中的应用 .....	86
6.4.1 浓密机仿真模型 .....	87
6.4.2 恒定-阶跃型噪音输入下浓密机控制仿真实验 .....	89
6.4.3 高斯噪音波动输入下浓密机控制仿真实验 .....	92
6.4.4 真实工业场景下浓密机控制算法算法性能评估 .....	95
6.5 本章小结 .....	97
7 总结 .....	99
7.1 本文研究工作总结 .....	99
7.2 未来工作展望 .....	100
参考文献 .....	102



## 表格清单

3-1	膏体浓密机系统数据样例 . . . . .	29
3-2	底流浓度预测的相对方根误差 (RRSE)、平均平方误差 (MSE)、时间消耗 . . . . .	32
3-3	不同插值方法对预测精度的影响 . . . . .	36
3-4	不同初始隐状态 $h(t_0)$ 生成方法对于预测精度的影响 . . . . .	37
4-1	跳变系统在四种状态下循环，每种状态对应制冷系统的一个阶段 . . . . .	50
4-2	系统输入输出, 不同负载下的数据集和模型的操作 . . . . .	52
4-3	不同模型累积能耗预测精度和推理时间的对比 . . . . .	54
4-4	能耗优化效果 . . . . .	57
5-1	本章提出方法与现有系统建模方法的对比 . . . . .	60
5-2	数据集特性 . . . . .	70
5-3	连续搅拌釜式反应器 . . . . .	71
5-4	工业绕组数据集 . . . . .	71
5-5	浓密机数据集 . . . . .	72
5-6	测试集采样比例与训练集比例采样比例不同时, 模型预测性能对比 . . . . .	74
6-1	参量定义 . . . . .	87
6-2	仿真模型常量 . . . . .	88
6-3	部分变量计算方法 . . . . .	88
6-4	不同控制算法之间性能分析 . . . . .	94
6-4	Performances analysis of different algorithms . . . . .	94
6-5	真实工业场景下浓密机控制算法算法性能评估 . . . . .	95
6-6	Add caption . . . . .	96



## 插图清单

1-1	本文研究内容、研究目标及关键问题的对应关系	6
1-2	论文组织结构	9
3-1	基于 ODE-net 模型的输入输出系统预测模型整体结构	25
3-2	膏体浓密机系统运行过程及监测变量图示	29
3-3	训练集、验证集、测试集的构建过程图示	30
3-4	不同系统及不同 ODE 求解器在 $L = 60$ 短期预测任务中的性能比较	33
3-5	$L = 200$ 时，不同 ODE 求解器、系统动态的预测效果对比	34
3-6	不同预测长度下稳定系统和非稳定系统的预测精度变化	35
4-1	H-ODEnet 结构图示	42
4-2	AJ-ODEnet 模型结构	43
4-3	基于 AJ-ODEnet 的编码器-解码器预测框架	45
4-4	时间预测器的损失函数定义	47
4-5	工业制冷系统整体结构及热交换过程	49
4-6	在平均负载不同的四个数据集中，阶段启动和关闭阶段的持续时间分布箱线图	50
4-7	制冷系统运行机理简化图	51
4-8	不同模型预测进气口温度，制冷量，制冷机功率的效果对比	53
4-9	预测不同时间长度的功耗的 MAPE 的变化	55
4-10	在不同温度下限设定值下的进气口温度的仿真值	56
4-11	在不同热负载下改变温度设定下限值对功耗、COP、PUE 的影响	58
5-1	ODE-RSSM 中的生成过程和推理过程	61
5-2	三种导数模块定义方法图示	62
5-3	基于祖先采样的高效隐空间超调	66
5-4	Winding dataset.	72
5-5	时间复杂度改善效果评估	75
6-1	迭代梯度下降过程可视化	84
6-2	短期经验回放对评价网络的输出值的影响	86
6-3	噪音量变化曲线	89

6-4 HCNVI 与其他 ADP 算法在恒定噪音输入下的对比 . . . . .	90
6-4 短期经验回放对 HDP 与 HCNVI 的影响 . . . . .	91
6-5 实验一中 HDP 与 HCNVI 在时间消耗上的对比 . . . . .	92
6-6 噪音量变化曲线 . . . . .	93
6-7 HCNVI 与其他 ADP 算法在波动噪声输入下的对比 . . . . .	93
6-8 噪音持续变化下短期经验回放对 HCNVI 的影响 . . . . .	94
6-9 实验二中 HCNVI 算法与 HDP 算法在时间消耗上的对比 . . . . .	95
6-9 底流浓度控制差分对比图 . . . . .	96
6-10 底流浓度序列分解对比图 . . . . .	96
6-11 底流浓度序列分解对比图 . . . . .	97

# 1 引言

在工业场景下，动态系统建模在过程控制、状态估计、系统预测等众多领域都起到了举足轻重的支撑作用。由于现实世界大部分的动态系统具有非线性、高扰动、强耦合等复杂特性，从机理角度建模系统动态过程的传统分析方法难以满足实际应用要求。伴随着工业监测技术的不断完善以及生产自动化、信息化水平的不断提升，各种大型设备及生产过程均安装了用于实时监测数据的传感器。由于监测数据获取成本低廉，且现有辨识理论及方法存在一定局限性，使得基于数据驱动的复杂工业系统建模方法广受学者们的关注。然而，系统建模问题本身的高复杂性以及被辨识系统的多样性使得模型的辨识效果对于模型选择与设计极其敏感。如何根据目标系统的不同特性，设计合理的参数化模型以获得最佳的建模效果，是领域内亟待解决的关键问题。本文分别面向具有长时延非线性、周期多阶段性、随机非确定性的三种复杂工业系统，提出了以常微分方程网络为骨架的三种模型架构，有效实现了**以系统先验为指导、以离线数据为原料、以神经网络为骨架，端到端建模复杂动态系统的目标**。同时，并在识别模型的基础上，提出了**基于有模型强化学习的复杂工业系统在线控制优化方法**，并成功应用于工业实践，搭建了系统建模与控制决策之间的桥梁。

## 1.1 研究背景与意义

复杂系统及设备的分析与优化技术涉及了集工业制造、自动控制、计算机、人工智能等多学科知识，长久以来受到了国内外学者的广泛关注与深度研究。如何建模复杂系统的运行过程并认知系统的内在机理，是实现系统分析优化的基础。想要解决复杂系统的建模问题，其本质在于对系统关键变量的自相关趋势以及受其他协变量的影响进行建模。由于在工业环境下，复杂系统的运行过程存在非线性、高扰动、强耦合等复杂特性，从物理、化学、动力学角度建模系统动态过程的传统分析方法不再适用。伴随着工业监测技术的不断完善以及生产自动化、信息化水平的不断提升，各种大型设备上安装了用于实时监测生产数据的传感器。低廉的数据获取成本以及理论建模的局限性使得数据驱动方法成为建模复杂工业系统过程的主流方案。

最早追溯到 1950 年，为了进行控制系统设计，文献<sup>[1]</sup>首次提出了系统识别的概念用来建模动态系统。其核心目标是寻找一个与系统“相符”的模型，使得模型预测的输出尽量接近给定真实的系统输出。以数据为核心驱动力的传统系统识别方法已经发展为一个非常完善且成熟的研究领域。<sup>[2-6]</sup>

在《关于发布未来工业互联网基础理论与关键技术重大研究计划 2021 年度项目指南的通告》中指出，“实现动态扰动下系统分布式资源调控、数据驱动的系统建模、质量预测与控制以及全流程重构的多目标优化，结合航空航天……。”足以说明数据驱动系统建模技术在现代工业智能化发展路线中占据着举足轻重的地

位。

AI 界当代最著名巨擘之一、Meta 的 AI 实验室灵魂人物 Yann LeCun，长期致力于让机器对世界的运转理念有基础了解。在其设计的通用人工智能（Artificial General Intelligence, AGI）架构体系中，设计了配置器、短期记忆模块、感知器、决策器、世界模型、评价模型，其中世界模型放在了与感知器和决策器同等重要的位置上。理想的世界模型能够像拥有“常识”的人类一样，预见给定行为后将产生后果，并辅助智能体决策。本文探究的动态系统建模可以认为是世界模型在低维控制、低维观测限定下的特例。将单个输入输出控制系统的观测、感知、建模预测、决策定义为一个小的世界。与通用人工智能的研究在范式层面具有较强的相似性。

从机器学习的视角来看，系统建模本质上是一种有监督学习问题<sup>[7]</sup>。对于动态模型，假定其系统状态表示为  $s_t$ ，系统输入为  $a_t$ ，给定批量的状态转移数据，我们可以学习前向模型  $(s_t, a_t) \rightarrow s_{t+1}$ ，即给定当前状态和给定的动作，预测下一个状态。在基于模型的强化学习研究中，该模型也称为的前向模型。

由于现实生活中大部分的客观物理系统具有连续动作空间以及高维观测空间，依赖于表格存储形式的前向模型表示方法难以适用。函数近似也因此成为当前复杂动态系统建模问题的主流方法。如线性回归<sup>[8]</sup>，动态贝叶斯网络、随机森林、最近邻搜索、神经网络<sup>[9]</sup>。近十年来，人工智能领域发展迅猛，深度神经网络以其极强的特征提取与非线性表示能力成为解决高维、大数据、多模态机器学习问题的常用解决方案。基于深度网络模型的复杂系统建模方法逐渐收到大家的关注。相比于其他方法，神经网络能够很好地扩展至高维输入输出空间以及非线性系统。利用神经网路的前向传播过程模型可以模拟系统的动态过程<sup>[10,11]</sup>。特别地，循环神经网络（Recurrent neural network, RNN）因为存在隐状态，可以更好地处理长期预测问题并对系统进行建模<sup>[12,13]</sup>。本文的讨论内容也限定在基于深度学习的系统动态建模方法中，研究面向不同类型动态系统的深度网络设计以及学习方法研究。

依托于近年来机器学习、强化学习<sup>[14]</sup>、深度学习<sup>[15][16]</sup>、时间序列分析技术<sup>[17]</sup>的发展与普及，从离散时间（Discrete-time, DT）角度的建模研究方向涌现出了诸多成果，领域发展相对较为成熟，其原因与数字计算机将信息离散化的思想密不可分。

相较于离散时间系统，连续时间（Continuous-time, CT）系统作为最早的系统辨识研究范式，近年来其相关理论体系研究较为滞后。在当今数据驱动与深度学习盛行的时代背景下，尚未与主流技术形成深度融合。但是对比离散时间系统建模方法，从连续时间域角度进行系统建模的思想在**与物理属性的兼容性、引入先验知识的难易度、处理非均匀采样数据、采样间隔的自适应、计算效率与精度的可调节性、刚性系统建模**等方面具有天然的优势。此类特性在复杂工业系统中尤为常见。

发表自神经信息处理会议 2018 (Neural Information Processing, NIPS 2018) 的

一篇开创性文章<sup>[18]</sup>提出了一种常微分方程神经网络，其采用神经网络参数化微分方程的向量域<sup>[19]</sup>。在其基础上，神经受控微分方程网络<sup>[20]</sup>、神经随机微分方程网络<sup>[21]</sup>、神经偏微分方程网络<sup>[22]</sup>等其他神经微分方程（Neural Differential Equations, NDEs）被相继提出。许多流行的神经网络架构，如残差网络、循环神经网络均可视为神经微分方程的离散化形式。

神经微分方程网络搭建了连续时间微分方程与深度学习技术之间的桥梁，兼具现代机器学习以及传统数学建模方法的优势，包括复杂函数的强拟合能力、便于在模型空间中引入强先验假设、处理非均匀采样数据、空间利用率高等特点。因此神经微分方程开辟了解决连续时间系统建模问题的新途径。然而在工业领域中，大部分的动态系统机理复杂、特性迥异，基于单一前馈神经网络的神经微分方程无法适用于所有建模任务，需要结合被建模系统的不同特性、训练数据的统计规律、辨识模型的预测需求，设计合理的参数化模型结构，并选用合理的优化目标与训练方法以获得最佳的建模效果。因此，本文以膏体浓密机、工业制冷系统等复杂工业设备及生产过程为对象，面向长时延非线性、周期多阶段性、随机非确定性等三种复杂特性，提出基于常微分方程网络系统建模方法。该项研究能够有效拓宽连续时间域下深度学习技术及神经微分方程网络的应用边界，在数据驱动的工业系统建模与优化领域开创新的研究思路。

## 1.2 本文研究的关键问题

神经微分方程网络兼具了深度学习模型的强拟合能力与微分方程的连续时间演化特性，但是使用该模型拟合部分动态系统时，由于系统存在各类复杂特性，常规网络结构及建模方法会面临以下研究难点与挑战：

### 1.2.1 具有长时延、强非线性的复杂连续时间系统建模问题

随着大数据采集与处理技术的不断进步，很多企业会在复杂工业设备上安装大量传感器以监测工业系统的实时运行过程。构建时序预测模型并利用采集到的数据集进行训练，然后采用基于预测模型的系统优化及控制策略解决复杂系统的决策难题<sup>[23-25]</sup>。现有的辨识与预测方法在解决复杂工业系统建模问题时难以解决两方面问题，首先大多数工业系统都有极其复杂的高阶动力学方程，它们不是仿射系统或线性系统，经典的系统假设及参数估计方法无法适用。另外，系统当前状态的变化可能受很长一段时间之前系统外部输入的影响，直接建模输入输出之间的微分方程难以从数据中捕捉系统具备的长时延特性。其次，现有的大多数基于深度学习的系统建模及预测方法<sup>[24,26-29]</sup>都是基于离散时间域的，忽略了系统的连续时间特性。从模型结构与系统本体的一致性角度来看，忽视系统本身具备的物理先验特性会增加模型的自由度及拟合难度，进而限制了模型精度。最后，另外基于数据驱动的自回归系统辨识模型受制于预测过程累积误差的存在，难以同时处理系统短期预测和长期预测。

因此，本文围绕连续时间视角下的深度系统辨识模型进行研究，预期解决长时延、非线性复杂动态系统的长短期开环预测问题。

### 1.2.2 具有周期多阶性的连续时间跳变系统建模问题

为了从给定数据中学习连续时间跳变系统，以往的研究一般限定了系统的先验结构，采用 EM 算法<sup>[30]</sup>、序列蒙特卡罗 (SMC)<sup>[?]</sup> 和变分推理<sup>[31]</sup> 等方法估计给定“灰盒”系统的模型参数。建模跳变系统的另一个困难是数据集中涵盖了具有不同动态特性的多个子过程，不同子阶段之间的转换规则很难识别。以前的一些研究采用自适应性模型识别此类具有多个子过程的复杂时变系统。深度状态空间模型<sup>[32]</sup> 利用一个参数动态变化的线性状态空间模型建模系统输出，并引入循环神经网络 (RNN) 建模参数的演化过程。其他研究如 Embed to Control (E2C) citeWatter2015 和卡尔曼变分自编码器 (KVAE)<sup>[33]</sup> 采用多个时不变的线性状态空间模型建模系统在隐空间中的不同动态，并推断出一个随时间变化的权重  $\alpha(t)$  以分配每个线性状态空间模型的权值。上述用于建模时变动态系统的混合模型假定目标系统是一个黑盒，其多个子过程是完全混合的。而在某些情况下，对于一些自切换的跳变系统，系统在某一时刻的所处阶段是唯一的且阶段转移边界是明确的。目前还没有研究将系统阶段转移的先验知识引入到辨识模型的设计中，以预测跳变系统中的阶段自切换。另外，对于带有多输出项的工业系统，其输出过程可能由多个不同的物理特性共同影响。这会导致多输出项中同时存在稳定和非稳定过程<sup>[34]</sup>。目前，对于现有的未经过特定设计的预测模型难以解决此类带有混合时序特性的系统学习任务。

因此，本文将着重研究具有多输出量的连续时间跳变系统建模方法，针对多阶段之间周期性转换以及稳定、非稳定输出共存的跳变系统，提出基于常微分方程网络的系统建模解决方案，依赖已知的系统先验特性提高模型的辨识精度。

### 1.2.3 具有随机性、非确定性的复杂连续时间系统建模问题

现存的连续时间系统辨识及动态系统建模方法，如 Time-Aware RNN<sup>[35]</sup>、SNODE<sup>[36]</sup> 仅在确定性状态空间对模型进行表示。首先，确定性模型没有引入任何随机性成分，不便于实现蒙特卡洛采样，这使得某些基于随机采样预测的控制规划方法，如交叉熵 (Cross entropy Maximum, CEM)、蒙特卡洛树搜索 (Monte Carlo Tree Search, MCTS) 难以与此类系统建模方法进行配合使用。另外，确定性模型显然无法适用于建模带有随机性的模型，当被辨识系统的状态转移过程本身具备较强的随机性时，可以近似认为状态之间的转移过程服从某种复杂的分布，理想的辨识模型应该能够直接对该分布进行建模，如离散时间域下的循环状态空间模型 (Recurrent State Space Model, RSSM)。最后，确定性模型无法对系统当前状态的非确定性进行度量与表示。因为在现实世界中，尽管很多系统的转移过程本身是确定的，但由于其观测空间的不完备性，从可观测的输入输出数据中无法准确推理系统的内部

状态，可以近似认为系统是存在非确定性的。现有连续时间域的系统辨识方法仅能在隐空间中隐式地对系统的非确定性进行编码，而无法对其显示地量化与评估，制约了模型的可解释性与拟合能力。

因此本文着重研究存在随机性、非确定性的复杂系统连续时间域系统建模方法，使辨识模型能够对系统的随机特性进行拟合，并在给定观测数据下评估系统的非确定性。

#### 1.2.4 复杂工业系统的控制策略构建与自适应优化问题

大部分复杂工业生产过程往往伴随着较强的非线性、非确定性、高时滞性，因此难以建立准确的数学模型近似其运转机理，导致传统的控制优化方法无法适用于此类复杂工业设备。目前业界对基于强化学习理论的最优控制技术<sup>[37][38]</sup>寄予厚望，希望能够以免模型、数据驱动的方式实现复杂工业系统的自适应优化控制。Wei等<sup>[39]</sup>将煤炭气化过程的最优追踪控制转化为双人零和最优控制问题，并采用包含控制网络、模型网络、评价网络的迭代自适应动态规划方法求解最优控制律，同时给出了收敛稳定性的分析。Jiang等<sup>[40]</sup>利用穿插学习策略迭代(Interleaved Learning Policy Iteration, ILPI)并同样采用三网结构实现了对浮选过程操作指标优化的控制，获得了比传统值函数迭代(Value iteration, VI)、策略迭代(Policy iteration, PI)算法更佳的控制效果。然而，考虑到工业过程试错成本高，大部分强化学习算法随机设定策略模型的参数，在模型训练初始阶段，难以保证生产过程的安全。另外，在工业场景下进行设备在线控制对算法的实时性要求较高。为了保证控制模块的性能，需要采用实时生成的数据对网络进行训练，使得训练过程产生较大的时间开销，模型更新与推理的实时性难以保证。最后，受限于无模型强化学习存在高采样成本与低场景泛化能力等缺陷，无模型强化学习算法在真实的工业实践中难以部署应用。

因此本文研究基于模型的复杂工业系统优化控制策略，充分利用系统运行时的离线数据构建预测模型，并在辨识模型的基础上构建具有在线自学习能力的控制决策模型，该方法能够适应物料性质改变、设备老化等被控系统不断变化的情况。

### 1.3 本文的研究内容

针对第1.2节提出的复杂工业系统难以建模预测及优化控制的问题。本文以具有连续时间动态特性的复杂系统作为研究对象，针对系统存在的非线性、非确定性、多阶段混合、高时延、不同输出量统计特征不一致等特性，依托于连续时间域模型的灵活性与深度神经网络的强大表示能力，研究基于连续时间深度时序网络的系统建模方法。并在识别模型的基础上，研究基于有模型强化学习理论的在线优化控制方法，并应用于工业实践。图1-1展示了本文研究内容、研究目标以及相关科学问题的对应关系。接下来，将给出每一项研究内容的研究思路和简要介绍。

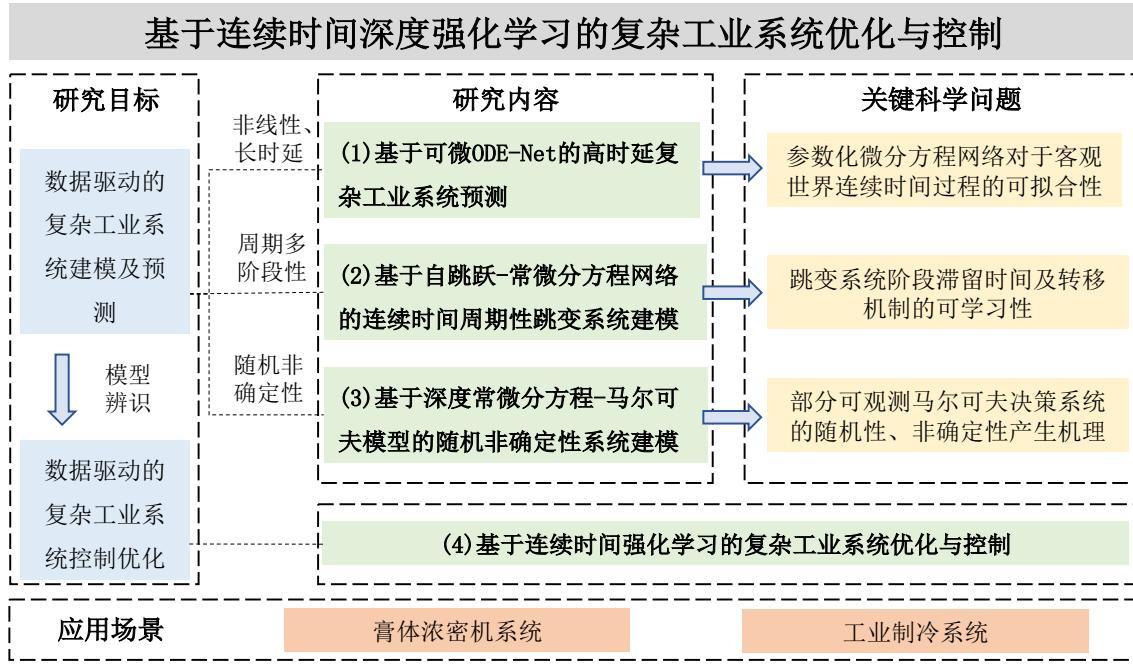


图 1-1 本文研究内容、研究目标及关键问题的对应关系

### 1.3.1 基于可微 ODE-Net 的高时延复杂工业系统预测

第一项研究内容基于可微 ODE-Net 的高时延复杂工业系统预测，重点讨论了连续时间视角下，非线性、高时延系统的系统建模问题。大部分复杂工业系统反馈延迟长，现有辨识方法难以拟合长距离的依赖关系，如果从状态空间模型的角度将系统状态和输入记忆进行压缩，会对待学习过程引入了极强的复杂性与非线性。另外基于数据驱动的自回归系统辨识模型受制于预测过程累积误差的存在，难以同时处理系统短期预测和长期预测问题。因此需要从探索**参数化微分方程网络对  
于客观世界连续时间过程的可拟合性**这一科学问题出发，研究深度学习网络与连续时间过程相结合的系统建模方法。

本文的第一个研究内容为了克服动态系统存在的非线性、非仿射的高复杂性。研究了基于深度学习的复杂工业系统建模方法，同时为了契合生产设备运转本质为连续时间过程的特性，提出以 ODE-net 作为模型骨架结构，从连续时间域角度拟合复杂工业系统的动态过程。针对系统输入输出之间长时延依赖关系难以建模的问题，研究了基于序列自编码器模型的输入输出长距离信息连接通道构建方法。针对深度序列模型受制于累积预测误差，难以有效进行长期预测的问题，基于时间序列稳定性理论，分析了普通常微分方程网络在长期预测中的非稳定特性，并提出两种分别适用于短期预测和长期预测任务的导数模块定义方法。同时，针对 ODENet 网络训练需要连续控制输入信号的问题，研究了面向训练过程的离散输入点并行插值方法。最后，提出了一套由序列编码器、状态解码器和导数模块组成深度连续时间 (Continuous Time, CT) 系统辨识网络模型，能够以端到端的方式学习工业系统输出的自回归变化过程和输入对输出的非线性影响。通过消融实验探

究了编码器输入序列长度、微分方程求解器选择对于预测精度的影响。最后，通过某真实铜矿场中膏体浓密机的运行数据验证了本文提出方法在解决高时延、非线性复杂系统长短期预测问题的有效性。

### 1.3.2 基于自跳跃-常微分方程网络的连续时间周期性跳变系统建模

第二项研究内容基于自跳跃-常微分方程网络的连续时间周期性跳变系统建模重点讨论了连续时间视角下周期跳变系统建模问题。工业场景中，系统在不同阶段下的动态特性彼此迥异，且各阶段出现的持续时间、状态转换的触发条件会同时受到内部、外部多种混杂因素共同影响，其影响机理复杂，可能超出领域知识的可解释范畴。因此，如果想要结合参数化深度模型对具有跳变特性的多阶段系统以端对端的方式进行数据驱动建模，需要从探索**跳变系统阶段滞留时间及转移机制的可学习性**这一科学问题出发，研究符合跳变系统先验特性的且具有阶段自识别、自转移能力的多阶段深度辨识方法。

具体地，该研究内容针对周期多阶段系统在不同阶段下动态特性彼此迥异，难以统一建模的问题，研究了传递式多 ODE-Net 集成结构，以独立建模系统在不同阶段下的动态特性，并支持开环预测阶段转换处隐空间的状态衔接。针对阶段持续时间难以预测、转换条件和位置难以识别的问题，提出了跳变系统辨识问题中的自跳跃 (Autonomous jump) 概念，并在多 ODE-Net 集成架构的基础上，研究了基于阶段持续时间预测网络的阶段自转移方法。除此之外，针对工业系统的多输出项可能同时存在稳定特性和非稳定特性的情况，研究了不同类别输出项的解耦建模方法，并提出了稳定 ODE 与非稳定 ODE 相结合的分层常微分方程网络单元。最后，通过某真实工业制冷系统的运行数据验证了本文提出方法在解决多输出周期跳变系统建模问题的有效性。

### 1.3.3 基于深度常微分方程-马尔可夫模型的随机非确定性系统建模

第三项研究内容“基于深度常微分方程-马尔可夫模型的随机非确定性系统建模”重点讨论了连续时间视角下，随机非确定性复杂系统建模的诸多问题。现实世界中的复杂系统往往具备典型的随机性 (Uncertainty) 和非确定性 (stochasticity)，确定性系统辨识模型只能以最小化期望误差的方式拟合系统随机演化函数在某分布下的期望，不仅不便于实现系统的随机采样预测，且无法对可观测数据呈现出的非确定性和随机性给出表示与度量。因此，想要在辨识模型中引入对于系统随机非确定性的感知与描述能力，并给出概率域下的识别及预测结果，需要从“**部分可观测马尔可夫决策系统的随机性、非确定性产生机理**”这一科学问题出发，研究贝叶斯视角下隐空间状态的时序生成过程及逆向推理方法。

具体地，针对随机非确定性连续时间系统难以表征、建模的问题，研究向确定性微分方程系统演化中添加随机转移路径的有效途径，进而提出深度常微分方程与马尔可夫模型相结合的随机非确定性系统建模方法，并结合时序变分推断理论

给出训练模型所需要优化的证据下界。同时，针对训练阶段下原始时序变分推理算法不便于实现（Backpropagation through time, bptt）难以保证多步预测精度的问题，研究了基于采样状态重用的高效隐空间超调技术，进而在不增加训练时间开销的情况下显著提升模型的开环预测精度。最后，针对训练阶段批数据中不同常微分方程积分区间不一致导致难以并行化训练的问题，研究了基于重参数变换的批常微分方程并行求解技术，成功实现了不同积分区间下的高效并行训练。最后，通过三个输入输出系统数据集验证了本文提出方法在解决随机非确定性系统建模及多步开环预测问题中的有效性。

### 1.3.4 基于连续时间深度强化学习的复杂工业系统优化与控制

第四项研究内容“基于连续时间深度强化学习的复杂工业系统优化与控制”重点讨论了连续时间视角下，基于数据驱动的复杂非线性工业系统的优化控制方法。大部分复杂工业系统的运行过程具有不完全观测、非线性、多变量、高时滞等特点，想要建立准确的数学模型描述其运转机理是极其困难的，因此基于模型的传统最优控制理论及方法难以适用。系统运行过程产生的历史数据为无模型数据驱动优化控制提供了可行的思路。想要充分利用离线数据构建近似的系统模型并衍生形成可靠的控制策略，需要以动态系统辨识为基础，研究基于辨识模型指导的复杂工业系统强化学习控制方法。

具体地，针对非线性、高时滞复杂工业系统控制优化难的问题，研究了基于数据驱动的有模型自适应动态规划控制方法，定义了系统的状态空间、动作空间、效用函数、状态转移函数等内容，并提出了离线系统建模与在线强化学习相结合的原始控制器构建及生产环境在线学习策略，通过利用离线采集数据以及在线监测数据有效解决了复杂工业系统控制优化难的问题。同时针对在线环境下策略网络增量训练开销大的，在线学习与控制难以满足实时性的问题，研究了基于自适应评价值迭代的控制动作求解算法，通过在模型架构中去掉策略网络有效减少了在线学习的计算开销。另外，针对评价模块参数收敛慢、难以准确给出策略优化方向的问题，研究了基于短期经验回放的评价网络训练技术，有效提高了模型对于局部评价值梯度变化的敏感性及预测准确性。最后，本文选用一种典型的复杂工业系统——尾矿浓缩机，利用其仿真模型验证了本文提出的自适应控制算法在控制精度、时间消耗方面的优势。同时将该控制算法应用于某真实矿山的深锥浓密机底流浓度控制场景中，相比于原始的规则控制算法，大幅度提高了出料浓度的追踪控制效果及稳定性。

## 1.4 论文的章节安排

本文针对连续时间域下的复杂动态系统建模及控制中的关键技术开展研究，全文共分为七章，各章主要研究内容及之间的关系如图1-2所示：

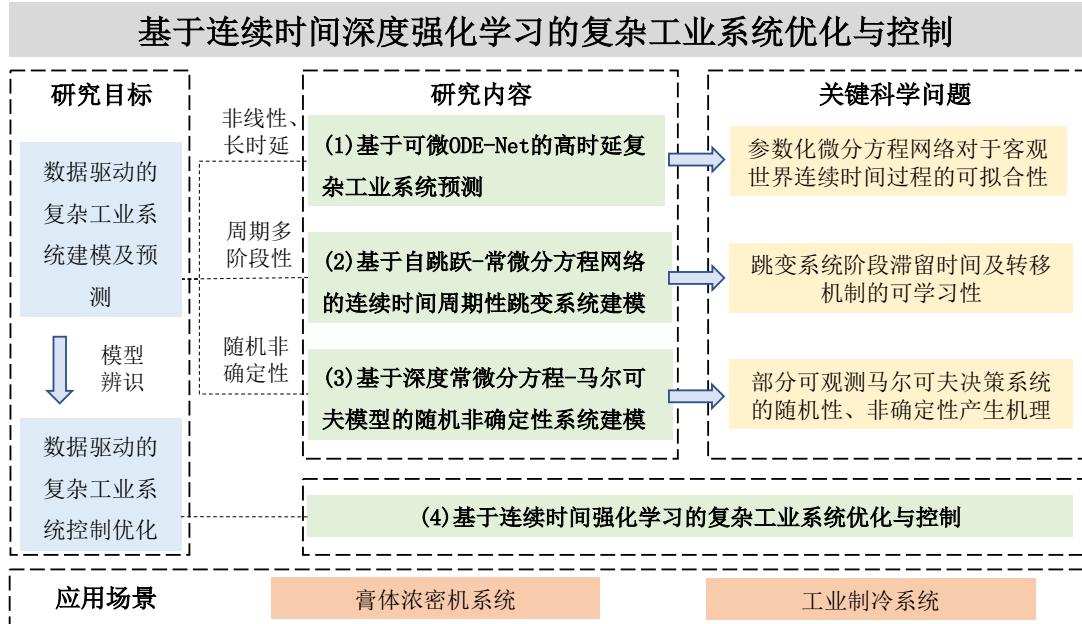


图 1-2 论文组织结构

第一章介绍本文工作的研究背景及意义，提出了基于数据驱动的复杂系统建模方法所面临的难点与挑战，并总结了本文的主要研究内容与创新点。第二章介绍了国内外关于数据驱动的动态系统建模的研究现状。第三章到第六章是本文的主体部分，详细介绍了本文的所有研究内容。

第三章研究了基于可微 ODE-Net 的高时延工业多输入输出系统预测，首先介绍了复杂输入输出系统预测问题的形式化表述并给出基于状态空间模型的表示方法。进一步地，介绍了本文提出的基于 ODE-Net 的连续时间输入输出系统预测模型，并分别介绍其序列编码器、导数模块以及状态解码器三大组成模块，同时给出了基于伴随状态的模型训练方法。另外，该章针对短期预测和长期预测两种预测场景，给出了两种导数模块定义方法。并提供了用于连续化系统输入序列的并行插值方法。最后，该章介绍了将上述模型应用于膏体浓密机系统预测问题的实验结果。并从微分方程求解器的选择、序列编码器输入的长度设定等多个角度进行了消融实验。

第四章研究了基于自跳跃常微分方程网络的连续时间跳变系统建模首先给出了连续时间跳变系统的变量符号定义以及系统预测问题的形式化描述。进一步地，介绍了本文提出的分层常微分方程网络的定义及结构，及如何将其用于学习同时带有稳定和非稳定时间序列输出的动态过程。然后，提出了用于建模连续时间周期跳变系统的 AJ-ODEnet 模型，并详细阐述了其状态转移方程、持续时间预测器等关键模块结构的定义。最后，给出了基于 AJ-ODEnet 模型的编码器解码器框架及其训练方法。实验环节介绍了如何将上述框架应用于模拟某实际工业制冷系统，预测给定热负载下的制冷功耗以及出气口温度变化，同时基于预测仿真结果，对制冷系统的起停温度设定值进行优化。

第五章研究了基于深度常微分方程-马尔可夫模型的随机非确定性系统建模，

首先给出了非均匀采样下随机系统的形式化描述及符号变量定义。技术部分首先介绍了本文提出的常微分方程-循环状态空间模型，包括其时序受控生成过程及隐变量的后验推理过程。并基于时序变分推断方法给出了用于训练模型的单步证据下界。进一步地，为了改善模型的多步预测性能，在单步下界的基础上提出了基于隐空间超调的多步变分下界。最后，提出了非均匀采样间隔下的批量常微分方程并行化求解方法以加速训练。实验环节采用两个共有数据集、一个私有数据集对常微分方程-循环状态空间模型以及多个基线模型在非均匀采样设定下的随机采样系统建模问题的有效性进行评估，并对多步预测性能实验环节采用两个共有数据集、一个私有数据集对常微分方程-循环状态空间模型以及多个基线模型进行对比评估，同时验证了隐空间超调技术对于多步预测的改善效果以及非均匀间隔批量常微分方程并行求解算法的有效性。

第六章研究了基于连续时间深度强化学习的复杂工业系统优化与控制，首先给出了基于连续时间强化学习的非线性系统控制过程的符号变量定义、形式化描述。技术部分主要介绍了本文提出的启发式评价网络值迭代控制算法，首先描述了如何利用常微分方程网络学习被控系统的近似参数化模型，然后基于积分强化学习理论给出折扣积分奖赏的定义及表示形式，并介绍如何利用评价网络对策略评价价值进行近似。进一步地，介绍了基于随机梯度下降的动作生成策略，最后描述了评价网络训练时，为了提升局部评价值梯度变化拟合结果的准确度，所采用的短期经验回放技术。实验环节评估了本文提出控制算法在浓密机仿真模型及某矿山真实深锥浓密机底流浓度控制问题中的有效性。

第七章对本文工作主要研究贡献和创新点进行了总结，最后面向对基于数据驱动的动态模型构建乃至世界模型构建问题，给出了几点关于模型设计的研究展望。

## 2 文献综述

### 2.1 动态系统辨识及有模型控制的起源与发展

从机器学习的视角来看，系统建模本质上是一种有监督学习问题<sup>[7]</sup>。本节主要从模型参数估计方法以及基于模型的优化控制两方面对动态系统辨识及有模型控制进行简要介绍。

对于动态模型，假定其系统状态表示为  $s_t$ ，系统输入为  $a_t$ ，给定批量的状态转移数据，我们可以学习前向模型  $(s_t, a_t) \rightarrow s_{t+1}$ ，即给定当前状态和选择的动作，预测下一个状态。对于所有的基于模型的强化学习，均需要关注于前向模型的构建。为了利用有监督学习算法学习上述前向模型，可以采用参数化方法和非参数化方法两种方式，进一步按照模型是否有随机性分类分为确定性模型和非确定性模型两种。

非参数化的系统建模方法直接存储以及利用历史数据进行动态模型的表示与预测。非参数化方法可以进一步分为精确建模和近似建模。对于精确建模，采用回放池<sup>[41]</sup>方法存储系统的历史轨迹，并将前向预测过程转化为检索问题。对于非参数的近似建模方法，比较常见的是高斯过程模型<sup>[42,42]</sup>，通过利用历史部分状态-动作点，可以外推预测某一新给定状态-动作点的高斯分布。该方法能够对系统演化的非确定性给出度量。

相比于非参数化方法，参数化方法的参数数量与观测数据集的大小是独立的。因此，参数化方法也是应用最广泛的模型近似方法。精确参数化方法也称为表格方法，对于每一种可能的转移维护独立的转移概率。如随机马尔可夫决策过程（Markov Decision Process, MDP），基于表格的最大似然模型能够统计每一种转移的概率：

$$T(s' | s, a) = \frac{n(s, a, s')}{\sum_{s'} n(s, a, s')} \quad (2-1)$$

$T$  代表真实系统  $\mathcal{T}$  的近似， $n(s, a, s')$  代表在状态  $s$  下执行动作  $a$  产生状态  $s'$  的频次。表格法作为最简单的系统建模方法，随着状态集  $\mathcal{S}$  的增大，表格的大小呈指数增加，因此难以扩展到高维问题。

另一种参数化建模的方式是对转移函数进行参数近似，在相似状态之间实现信息的泛化，以充分降低对于存储以及数据量的需求。函数近似也因此成为解决连续空间以及高维动态系统建模问题的主流方法。原则上，可以使用任意参数近似方法学习模型，如线性回归<sup>[8]</sup>，动态贝叶斯网络、随机森林、最近邻搜索、神经网络<sup>[9]</sup>。近十年来，深度神经网络已经成为解决通用函数近似问题的主流方法，也因此逐渐成为近似动态系统的主流方案。相比于其他方法，神经网络能够很好地扩展至高维输入输出空间以及非线性系统。利用神经网路的前向传播过程模型可以模拟系统的动态过程<sup>[10,11]</sup>。特别地，循环神经网络（Recurrent neural network, RNN）因为存在隐状态，可以更好地处理长期预测问题并对系统进行建模<sup>[12,13]</sup>。本

文的讨论内容也限定在基于深度学习的系统动态建模方法中，研究面向不同类型动态系统的深度网络设计以及学习方法研究。

## 2.2 连续时间系统辨识

在过去的几十年间，数字计算机技术发展迅猛。从离散时间（Discrete-time, DT）角度进行系统建模的研究方向涌现出了诸多成果，领域发展相对较为成熟，其原因与数字计算机将信息离散化的思想密不可分。

相比而言，连续时间（Continuous-time, CT）系统作为最早的系统辨识研究范式，近年来，其相关理论体系研究较为滞后，且在当今大数据、数据驱动的时代背景下，并未与主流技术形成深度融合。但是对比离散时间系统建模方法，从连续时间域角度进行系统建模的思想在某些情况下具有较大优势：

- 对于物理属性的兼容性：大部分的物理现象都服从连续时间设定，如化学、动力学、热力学等典型场景都可以利用微分方程进行表述，因此利用连续时间模型对物理系统进行建模可以更好地匹配客观世界物理规律，且增加模型的可解释性。
- 对于先验知识的表示：系统动态不同阶的相关度，如：位移、速度、加速度之间的相关性可以在连续时间系统下非常容易地表示出来，进而有效利用先验知识，降低模型求解的自由度。
- 潜在的数据滤波：一般的系统辨识过程需要执行特定的预过滤策略以消除数据中的噪音。而连续时间系统建模方法本质上包含了数据预过滤过程。
- 非均匀的数据采样：当数据采样间隔不均匀时，离散时间方法无法使用，但是连续时间辨识方法能够有效解决非均匀采样问题。
- 连续时间系统与离散时间系统的相互转换：对于连续时间系统可以通过变更采样率的方法获得任意时间间隔的离散时间系统。而对于离散时间辨识系统，当预测步长与下游应用所需的时间间隔不一致时，模型无法适用。
- 离散时间模型在高采样率下存在精确性问题：当数据采样率较高时，系统在相邻时刻间的状态变化较小，不适宜的 DT 模型难以从微小的状态变化中学习系统动态，导致模型的开环预测结果存在精度差、鲁棒性低的问题。连续时间模型本质上学习连续时间域下的微分方程，采样率越高，轨迹越完备，学习效果越好。
- 钢性系统建模：当系统同时存在快过程和慢过程时，离散采样的间隔选取不当很容易造成快过程或慢过程其中一类统计性质的丢失，对离散时间系统处理造成较大阻碍。而连续时间模型能够从频域角度对不同频率的演化过程进行独立建模。

上述七大特性在复杂工业系统中尤为常见。因此，最早的系统辨识方法研究也大多是围绕连续时间系统展开的。系统通常是由表征系统输入输出关系的数学模型描述的，这个模型有其特定的结构和参数。以代数方程表示的系统称为静态系统，考虑最简单的形式，系统的连续时间模型可以建模为常系数微分方程：

$$\frac{d^n y(t)}{dt^n} + a_1 \frac{d^{n-1} y(t)}{dt^{n-1}} + \cdots + a_n y(t) = b_0 \frac{d^m u(t)}{dt^m} + \cdots + b_m u(t) + v(t) \quad (2-2)$$

$\frac{d^i x(t)}{dt^i}$  代表信号  $x(t)$  对时间的  $i$  阶导数，该式表示了在任一时刻系统状态对时间的各阶导数与输入量对时间的各阶导数之间存在线性关系。

一般地，对于任意阶数的线性齐次微分方程可以转化为状态空间模型的形式：

$$\begin{cases} \dot{\mathbf{h}}(t) = f(\mathbf{h}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = C\mathbf{h}(t) \end{cases} \quad (2-3)$$

对于线性系统式 (2-2)， $f$  可以表示为线性函数， $C$  为投影矩阵， $\mathbf{h}(t)$  代表系统的隐空间状态。

然而，对于许多客观世界中的复杂动力学系统，基于理论驱动的模型在大部分时候无法准确描述系统的细节。动态  $f$  表现为某一未知函数且具有较强的非线性特性，通过将机理模型结构与深度学习等及其高性能函数逼近器<sup>[43]</sup> 相结合，可以缩小理论模型与观察数据之间的差距，准确地对函数  $f$  进行建模。

### 2.3 微分方程网络在复杂系统建模中的应用

发表自神经信息处理会议 2018 (Neural Information Processing, NIPS 2018) 的一篇开创性文章<sup>[18]</sup> 提出了一种常微分方程神经网络，其采用神经网络参数化微分方程的向量域<sup>[19]</sup>：

$$y(0) = y_0 \quad \frac{dy}{dt}(t) = f_\theta(t, y(t)) \quad (2-4)$$

其中  $\theta$  代表可学习的神经网络参数， $f_\theta : \mathbb{R} \times \mathbb{R}^{d_1 \times \cdots \times d_k} \rightarrow \mathbb{R}^{d_1 \times \cdots \times d_k}$  代表标准的神经网络结构。 $y : [0, T] \rightarrow \mathbb{R}^{d_1 \times \cdots \times d_k}$  代表微分方程的解。对于大部分应用， $f_\theta$  为简单的前馈神经网络。使用神经微分方程的核心是将微分方程求解器作为可学习微分计算图的一部分。ODE-Net 的主要应用包括 Res-Net 的替代、时间序列建模以及可逆正则化流<sup>[44]</sup>。本文主要围绕 ODE-Net 在时间序列建模及系统辨识中的应用展开研究。因为 ODE-Net 的连续时间特性，该方法能够将深度网络模型应用于连续时间域下的时间序列建模问题研究中，开辟了时间序列分析、连续时间系统建模研究的新思路。

以建模捕食者物种和被捕食物种之间相互作用的 Lotka-Volterra 模型为例，可

以采用带有未知参数的常微分方程表示：

$$\begin{aligned}\frac{dx}{dt}(t) &= \alpha x(t) - \beta x(t)y(t) \in \mathbb{R} \\ \frac{dy}{dt}(t) &= -\gamma x(t) + \delta x(t)y(t) \in \mathbb{R}.\end{aligned}\quad (2-5)$$

其中  $x(t)$  和  $y(t)$  分别表示被捕食者和捕食者的种群大小。在每个时间  $t$ , 右侧是理论经验公式, 表示物种之间的交互作用。但是在一般情况下, 尽管上述经验公式中的模型参数可以通过经典系统辨识方法进行估计, 但是模型结构本身过于理想化, 难以保证精确性。因此模型的理论预测值相对于实际观测数据会有明显差距。为了纠正该偏差, 可以在模型中引入神经网络  $f_\theta, g_\theta : \mathbb{R}^2 \rightarrow \mathbb{R}$ , 并构建如下模型:

$$\begin{aligned}\frac{dx}{dt}(t) &= \alpha x(t) - \beta x(t)y(t) + f_\theta(x(t), y(t)) \in \mathbb{R} \\ \frac{dy}{dt}(t) &= -\gamma x(t) + \delta x(t)y(t) + g_\theta(x(t), y(t)) \in \mathbb{R}\end{aligned}\quad (2-6)$$

理论模型能够在神经网络的帮助下被补充与增强。该方法也称为普适微分方程 (Universal Differential Equation, UDE)。

假设观察数据为  $x_i(t_j) \in \mathbb{R}, y_i(t_j) \in \mathbb{R}$ , 其中  $i = 1, \dots, N$  表示目标过程的独立观测值, 每个序列来自于不同的初始条件, 以及  $j = 1, \dots, M$  对应于不同的时间  $t_j \in [0, T]$  且  $t_1 = 0$ 。实际问题中, 往往  $N = 1$  且  $M$  较长。

对于式(2-5)或式(2-6), 假设  $x_{x_0, y_0}(t)$  表示给定初始条件  $x(0) = x_0$  和  $y(0) = y_0$  下  $x(t)$  和  $y(t)$  的解,  $y_{x_0, y_0}(t)$  同理。进而可以采用随机梯度下降法优化如下损失函数以拟合式(2-5)和式(2-6):

$$\frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M (x_{x_i(0), y_i(0)}(t_j) - x_i(t_j))^2 + (y_{x_i(0), y_i(0)}(t_j) - y_i(t_j))^2. \quad (2-7)$$

通过将式(2-5)中的经验模型替换为式(2-6), 充分利用了神经网络较强的函数近似能力, 以建模理论模型与观测数据之间的残差, 进而弥补了理论模型与实际模型之间的差距。用于动态系统建模的网络通常较小,<sup>[45]</sup>, 使用了宽度为 10, 层数为 10 的前向神经网络, Rac 采用宽度为 32 的单隐层神经网络。

对于难以利用第一性原理进行分析建模的复杂动态系统, 在拥有足够数据时, 使用微分方程网络进行建模是一种极其自然的方法。在这种情况下, 微分方程中的某一项缺乏精确的理论描述, 因此可以采用神经网络进行近似, 并从数据中学习。比如湍流建模问题, 在雷诺平均 Navier-Stokes 模型中, Ling 等<sup>[45]</sup> 使用神经网络近似系统的雷诺应力, 以满足某些物理不变性。同时, 在海洋气候模型领域, Ramadhan 等<sup>[46]</sup> 使用一个小型的 MLP 对湍流垂直热通量进行建模。在系统辨识领域, 在许多真实的工业场景中, 从被识别系统采样得到的序列数据经常是间隔非均匀的。利用微分方程网络模型能够有效建模系统输出或隐空间状态在外部输入

影响下的瞬时变化。Zhong 等<sup>[47]</sup>采用 ODE-Net 对符合哈密尔顿特性的动态系统进行建模学习，巧妙地将物理先验知识融入到学习模型的设计中。并有效地应用于符合哈密尔顿性质的刚体系统建模与控制问题中。Ayed 等<sup>[48]</sup>采用 ODE-Net 模型从系统状态的部分可观测信息学习复杂非线性时空过程。该方法有效应用于水流运动预测、Navier Stokes 方程、海洋温度分析。SNODE<sup>[36]</sup>使用谱方法用于快速和准确地训练神经常微分方程模型以辨识低维输入输出系统。ODE-RNN<sup>[49]</sup>在 RNN 网络相邻状态更新处插入 ODE-Net 模块，用于建模网络隐状态的连续时间演化。为了改善辨识模型在长期预测时的稳定性，如本文第三章所述的稳定性模型，现有研究<sup>[35,50]</sup>将递归神经网络的稳定性融入到微分方程的导数网络设计中以解决单位根问题，并提出了一种连续时间系统辨识模型——时间感知的 RNN 模型。

在常微分方程网络求解器效率的优化方面，Zhuang 等<sup>[51]</sup>提出了自适应检查点联合状态方法以改进原始联合敏感度法求解梯度的精度以及效率。正则化神经常微分方程（regularized neural ODE, RN-ODE）<sup>[52]</sup>基于理论保证的最优传输和稳定性正则化简化 ODE 系统，能够有效优化 ODE-Net 的数值积分求解效率。除此之外，对 ODE 方程的高阶导数正则化<sup>[53]</sup>对求解时间点添加随机扰动<sup>[54]</sup>等方案也能够起到正则化作用，加速 ODE-Net 的求解效率。

### 2.3.1 非马尔科夫系统及长时延系统建模

由于常微分方程的解仅由初始状态决定，如果直接用 NDE 对系统输出的连续时间演化进行建模，这要求目标系统或时间序列遵循马尔科夫特性。由于大部分客观物理过程往往是不完全可观测的，这样的假设对于难以成立，因此原始的神经微分方程结构难以处理长时关联性问题。一种较为直观的解决方案是将微分方程网络与循环神经网路相互结合，在状态循环更新的间隙，插入微分方程网络的求解过程以处理非均匀的采样序列。如基于 ODE-Net 衍生的 ODE-RNN<sup>[55]</sup>、GRU-ODE-Bayes<sup>[56]</sup>、ODE-LSTM<sup>[57]</sup>。

神经受控微分方程（Neural Controlled Differential Equation）<sup>[20]</sup>将受控信号的微分项融入在 ODE 网络的求解中。相比于在求解时间区间的间隔点处利用观测数据对隐状态进行更新的方法（如 ODE-RNN、ODE-LSTM 等），获得了更好的序列信息提取与表示能力。为了解决神经受控微分方程在长时间序列场景下难以训练的问题，基于对数签名变换（Log signature transform）的 Log-signature/NCDE 方法<sup>[58]</sup>通过签名变换对受控信号进行转换，能够增加模型训练速度、减少存储开销、改进模型的长期序列处理能力。针对常规的受控微分方程需要对离散序列进行样条插值，无法实现在线预测的情况，Morrill 等<sup>[59]</sup>描述了受控微分方程中连续控制信号应满足的性质，并给出了三次埃尔米特直线插值方法，使受控微分方程网络能够类似于 RNN 一样，实现在线的时间序列数据处理，而不需要预先对完整的序列数据进行插值。

另一种向 ODE-Net 中引入长时序特征的方式是将历史序列特征编码至常微分

方程的初始状态中。Latent ODE<sup>[55]</sup> 将 ODE-Net 的初始状态视为先验分布为标准高斯分布的因变量，利用编码器-解码器结构实现序列的重构、插值、预测，并使用变分贝叶斯方法对模型进行训练。文献<sup>[60]</sup> 基于贝叶斯理论，构建了 2 阶 ODE-Net 模型用于高维时间序列的建模，同时利用 ODE-Net 连续正则化流的特性估计待预测时间点隐变量的后验分布并引入分布正则化对隐状态的范围作出限定。为了建模来自于多智能体系统的非均匀采样数据，LG-ODE 模型<sup>[61]</sup> 将图神经网络、自注意力以及 ODE-Net 进行结合，利用时序自注意力模型构建微分方程求解的初态，采用图神经网络建模观测点的时序依赖关系和不同观测项之间的空间依赖关系，并以此为基础估计常微分方程中的隐状态导数。模型有效地应用于稀疏序列的插值预测与外推预测。

近年来，Transformer 模型<sup>[62]</sup> 因在序列数据处理问题上的优异表现受到了学者的广泛关注，基于 Transformer 的长序列预测模型，如 Informer<sup>[63]</sup>、Autoformer<sup>[64]</sup> 等均获得比传统时序预测算法更优的预测精度。对于 Transformer、AttentionSeq2Seq 等基于注意力的序列处理模型，利用 ODE-Net 的连续时间特性可以构建连续时间注意力模型<sup>[65]</sup>，并将注意力机制应用于非均匀采样的时间序列。得益于 ODE-Net 的连续时间特性，该模型能够与 Transformer 进行结合，处理非均匀采样数据或长序列预测及建模问题，如构建连续时间注意力<sup>[65]</sup>、辅助位置编码（position embedding）<sup>[66]</sup>。

### 2.3.2 跳变系统辨识

连续时间跳变系统包含多个连续时间子系统，且受到隶属有限状态集的内部阶段变量控制<sup>[67]</sup>。以最简单的连续时间跳变线性系统<sup>[68]</sup> 为例：

$$\dot{x}(t) = A(\sigma(t))x(t) + B(\sigma(t))u(t) \quad (2-8)$$

$A$  和  $B$  为模型参数。 $\sigma(t)$  为有限状态随机跳变过程，通常表示为时间自治的、状态有限的马尔科夫过程。

定义状态集  $\underline{N} = \{1, 2, \dots, N\}$ ，对于所有  $i, j \in \underline{N}$ ，定义状态转移概率矩阵：

$$\begin{aligned} p_{ii} &= 0 \\ q_i &= -q_{ii} = \sum_{l \neq i} q_{il} \\ p_{ij} &= \frac{q_{ij}}{q_i}, \quad (i \neq j). \end{aligned} \quad (2-9)$$

给定单步转移概率矩阵为  $(p_{ij})_{N \times N}$ ， $\{r_k; k = 1, 2, \dots\}$  表示离散时间马尔科夫链，也成为连续时间阶段信号  $\{\sigma(t)\}$  的嵌入马尔科夫链。相邻跳跃之间的滞留时间用  $\tau_k$  表示，初始状态表示为  $\sigma(0) = i$ 。过程以概率  $p_{ij}$  跳跃至状态  $j \neq i$ ，在阶段  $j$  下的滞留时间服从参数为  $q_i$  的指数分布。过程  $\{\sigma(t)\}$  访问过的状态表示为嵌

入马尔科夫链  $\{r_k; k = 1, 2, \dots\}$ , 相应的滞留时间  $\{\tau^{(r_k)}\}$  是服从独立指数分布的随机变量, 分布参数为  $q_{i_k}$ 。联合过程  $\{(r_k, \tau_k); k = 0, 1, \dots\}$  是时间同质的马尔科夫过程, 完全确定了跳变系统阶段变量  $\{\sigma(t)\}$ 。

为了从包含多阶段混合数据的离线序列中学习连续时间跳变系统参数  $\{A(i), B(i); i \in \underline{N}\}$  以及阶段切换参数  $\{q_i, p_{ij}; i, j \in \underline{N}\}$ , Ashley 等采用序列蒙特卡洛 (Sequential Monte Carlo, SMC)<sup>[69]</sup> 将辨识问题转换为一般的非线性状态空间估计问题。不过该方法计算量较大。为了更有效地利用有限的计算资源, 随机近似 (Stochastic approximation, SA)<sup>[70]</sup> 思想被引入 SMC 框架中。变分近似方法<sup>[31]</sup> 是其中的典型代表。

Balenzuela 等提出了一种基于 EM 算法的马尔科夫跳变系统辨识方法<sup>[30]</sup>。为了克服传统方法中 E 步计算复杂度随阶段数指数增长的问题, 该文章提出的 EM 方法在合并高斯成分的 E 步内使用近似算法, 以减少高斯分量的数量。原则上, 该方法不需要 SMC 和随机逼近方法中通常需要的渐近参数, 就可以得到精确解。

Oppen 等<sup>[31]</sup> 提出采用平均场理论, 在给定有噪音的观测数据下, 近似推理阶段变量以及估计系统参数。相比于马尔科夫链蒙特卡洛 (Markov chain Monte Carlo, MCMC) 方法关注于在阶段转换发生时从局部时间范围内进行采样, 该方法计算系统可能转移路径的概率分布的近似, 因而易于实现平均场的因子化。

### 2.3.3 动态系统随机非确定性的表示与推理

现有的大部分连续时间模型能够有效解决系统的不均匀采样问题, 但由于其模型内部状态的转换是确定性的, 因此对于存在随机性的系统, 其建模能力较弱。在建模随机性系统方面, 深度时序生成模型<sup>[71–73]</sup>, 一般也称谓深度状态空间模型 (Deep State Space Model, Dssm) 将变分自动编码器模型 (Variational Auto-encoder)<sup>[74]</sup> 扩展到了序列数据。

此类模型通过引入时序的随机隐变量学习序列的随机性。类似地, PR-SSM<sup>[75]</sup> 和 PILCO<sup>[42]</sup> 采用高斯过程来学习输入/输出系统的概率状态空间模型。利用上述方法构建的序列生成模型能够极为便捷地在给定条件序列输入的情况下, 预测输出分布并从中采样。这种性质也使得这些方法广泛应用于缺失数据填充<sup>[33]</sup>、开环预测<sup>[76]</sup>、基于模型的强化学习等任务<sup>[76]</sup>。

普通的常微分方程网络以及受控微分方程网络的状态演化过程是不包含随机性的。相比于常微分方程, 随机微分方程网络 (Stochastic Differential Neural Network, SDE) 在状态转移的向量域中添加了扩散项 (Diffusion process):

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t) dt + \sigma(\mathbf{x}_t) dW_t \quad (2-10)$$

其中  $\mathbf{f}(x) \in \mathbb{R}^D$  是系统的确定性状态演化, 也称为漂移项 (Drift process)。 $\sigma(x) \in \mathbb{R}^{D \times D}$  是一标量系数。 $W_t \in \mathbb{R}^D$  是多变量维纳过程, 其初始状态为  $W_0 = \mathbf{0}$ , 独

立的增量成分  $W_{t+s} - W_t \sim \mathcal{N}(\mathbf{0}, sI)$  服从标准差为  $\sqrt{s}$  的高斯分布。

随机微分方程网络即采用神经网络建模模式(2-10)中的  $\mathbf{f}(x)$  和  $\mathbf{f}(x)$ 。网络参数也可以采用联合敏感度法(Adjoint Sensitibity)进行训练<sup>[21]</sup>。由于随机微分方程网络的前向传播过程依赖于带有随机性的 Wiener 过程,为了避免存储完整的计算图,需要保证反向求解 SDE 时对 Wiener 过程的采样与前向传播保持一致。文献<sup>[21]</sup>利用基于虚拟布朗树(Virtual Brownian Tree)的伪随机数生成策略,仅需常数级存储  $\mathcal{O}(1)$  即可对的 SDE 网络前向传播的 Wiener 过程采样结果进行存储。节约存储空间的代价为对特定时间点下的 wiener 过程采样时间复杂度为  $\mathcal{O}(\log n)$ 。将微分方程网络作为 Res-Net 的替代品处理图像领域问题时,有研究表明将 ODE-Net 替代为 SDE-Net,在 ODE-Net 中增加随机扩散项,也能够起到随机正则化的作用,提升网络的泛化能力<sup>[77]</sup>。神经跳变随机微分方程(Neural Jump Stochastic Differential Equations, NJSDE)<sup>[78]</sup>将扩散项中对时间的微分替换为观测点次数的微分,能够在建模系统隐空间连续时间动态的同时对观测点出现事件本身以及时刻进行建模,该模型有效地应用于地震预测及药物预测。

## 2.4 强化学习及其在控制优化中的应用

在控制理论与应用领域中,强化学习<sup>[37]</sup>和自适应动态规划<sup>[79,80]</sup>是解决最优控制问题的常见解决方法。其中强化学习技术衍生于最优控制以及人工智能领域,用来解决马尔科夫决策问题。在连续时间系统系统中,引入强化学习解决最优控制问题也存在大量的研究先例。对于一般的连续时间系统:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \quad (2-11)$$

将强化学习用于解决连续时间系统决策问题一般采用积分强化学习方法,积分折扣累积奖赏定义为如下形式:

$$V^\mu(x(t)) = \int_t^\infty e^{-\gamma(\tau-t)} r(x(\tau), u(\tau)) d\tau \quad (2-12)$$

其中奖赏函数一般由人为给定且一般为二次形式:

$$r(x(\tau), u(\tau)) = X^T Q X + u^T R u \quad (2-13)$$

将积分过程在时间点 T 处分割,可以获得递归式:

$$V^\mu(x(t)) = \int_t^{t+T} \gamma^{\tau-t} r(x(\tau), u(\tau)) d\tau + \gamma^T V^\mu(x(t+T)) \quad (2-14)$$

其中,  $\mu(x(t))$  代表控制策略,  $V^\mu$  代表对策略  $\mu$  的评价, 是奖励函数,  $\gamma$  是折扣因子。上式以自举的形式给出了评价函数的定义,该方法可以很容易地将离散

系统强化学习方法中对于值函数的评估策略应用于连续系统中。对于控制量的求解可以通过对式的两侧求微分得：

$$\nabla V_x^T f(x(t), u(t)) - \gamma V + r(x(t), u(t)) = 0 \quad (2-15)$$

其中  $\nabla V_x$  代表评价函数对系统状态的导数，上式揭示了任一控制策略与其策略评价函数之间的等式约束。定义哈密尔顿函数：

$$H(X, V, u) = \nabla V_x^T f(x(t), u(t)) - \gamma V + r(x(t), u(t)) \quad (2-16)$$

根据稳态条件，可以推导出最优的追踪控制策略隐式表达：

$$\frac{\partial H}{\partial u}(X, V^*, u) = 0 \Rightarrow \frac{\partial r}{\partial u^*(t)} + \frac{\partial f}{\partial u^*(t)} \nabla V_x^*(X) = 0 \quad (2-17)$$

利用上式对最优控制策略  $u^*(t)$  求解的前提是已知  $\frac{\partial f}{\partial u^*(t)}$  和  $\frac{\partial r}{\partial u^*(t)}$ 。但是由于实际问题中连续时间系统函数  $f$  未知且往往带有一定的非线性，想要求得控制律的解析解是极其困难的。为了实现无模型非线性系统的强化学习控制，Abu-Khalaf 等<sup>[81]</sup> 采用策略迭代算法求解连续时间非线性系统的饱和控制器。为了避免解决非线性系统 CT 控制问题中对时间导数的计算，Vrabie and Lewis<sup>[82]</sup> 提出了积分强化学习技术，该方法可以在模型未知的情况下仅依靠系统运行数据实现对模型控制率的自学习。Vamvoudakis 等<sup>[83]</sup> 采用 actor-critic 算法结构构建同步策略迭代算法，同时调节利用神经网络构建的值函数与策略函数。最优控制领域的追踪控制问题旨在于最小化控制系统指标与目标轨迹之间的距离，基于强化学习或自适应动态规划方法解决追踪控制问题的工作包括<sup>[84,85]</sup>。然而此类方法要求已知目标轨迹的变化过程且对于系统形式具有较强限制，无法应用于复杂连续时间系统。得益于积分强化学习理论的发展，Modare 等<sup>[86]</sup> 利用基于积分强化学习的策略迭代算法实现了对连续时间系统的最优追踪控制。Lewis 等<sup>[87]</sup> 扩展了同策略迭代至连续时间非线性系统，并且保证了系统的稳定特性。上述算法均要求系统动态增益矩阵已知。为了打破限制，Jiang 等<sup>[88]</sup> 提出了一种免模型的，仅基于系统观测下迭代地学习最优追踪控制策略的方法，且该方法可以在执行策略与学习策略不同的情况下实现异策略学习。Modares 等<sup>[89]</sup> 考虑了在系统存在扰动下，采用  $H_\infty$  控制削减负效应，并利用离线迭代对策略进行更新。

上述强化学习研究大多使用简单的线性模型或者感知机模型建模策略函数和评价函数。深度强化学习作为深度神经网络与强化学习的结合体，是近年来人工智能领域的研究热点。DeepMind 发表在 Nature 上的两篇论文，AlphaGo<sup>[90]</sup> 和深度 Q 学习 (Deep Q-Learning Network, DQN)<sup>[91]</sup> 大幅度推动了 DRL 的发展。AlphaGo 在围棋领域横扫人类选手。DQN 模型成功玩通 Atari 上大部分游戏并且获得比人类玩家更优的成绩。学界以及工业界对于利用深度强化学习实现通用人工智能抱

有期待。

对于部分复杂控制系统而言，系统的高时滞特性与复杂非线性不满足传统控制理论的研究假设。而深度学习方法在处理复杂非线性映射、长序列编码方面具有极大优势，因此有部分学者提出了基于深度强化学习的过程控制研究思路。Spielberg 等<sup>[92]</sup>首先提出采用 DDPG 模型学习系统的过程追踪控制，将系统偏离目标值的 L1 距离作为惩罚函数，在仿真实验中测试了模型具有收敛能力。Yu 等<sup>[93]</sup>采用确定性策略梯度算法思想解决自动潜航器的最优追踪控制问题。仿真实验验证了该方法的性能由于 PID 算法。Kim 等<sup>[94]</sup>提出采用深度强化学习方法解决非线性系统有穷时间范围最优追踪控制问题，以 ADP 算法体系中的全局双启发式动态规划（Globalized Dual Heuristic Programming, GDHP）为模型总体结构，利用深度强化学习的模型训练方法来对协状态评价网络和动作网络进行训练，在一个通用的非线性系统控制仿真模型连续搅拌反应器上进行了实验测试，控制效果较好。Kim 等<sup>[95]</sup>后续又开展了有模型情况下，利用深度强化学习方法解决有穷时间范围非线性仿射系统的最优控制问题，模型结构同样是遵循 GDHP 的架构。由于 ODE-Net 模型能够利用非均匀采样数据拟合动态系统，因此可以构建可微分的系统状态演化估计器，配合连续时间强化学习辅助策略模块的学习，进而解决非均匀采样数据下的决策控制问题<sup>[96]</sup>。对于系统动态已知的连续时间系统，利用 ODE-Net 可以为策略网络构建连续时间梯度估计器<sup>[97]</sup>，使控制和仿真任务的学习更高效、更鲁棒。

## 2.5 本章小结

从上述文献可以发现，因微分方程网络所具备的连续时间特性，此类模型广泛应用于时间序列分析、时间点过程分析、连续时间动态系统识别等问题，并有效支撑了时序预测、稀疏序列插值、系统控制等应用任务。目前为止，基于微分方程网络的动态系统识别方法尚在研究初期，如何面向不同类型的复杂动态系统选择合适的网络结构以及模型训练方法是该领域尚待解决的关键问题。

### 3 基于可微 ODE-Net 的高时延工业多输入输出系统预测

在复杂过程工业系统控制研究领域中，基于传统控制理论的系统辨识及控制器设计方法受限于系统复杂性难以适用。而基于强化学习的无模型控制策略学习方法虽然不受系统复杂性的影响，但需要与环境之间不断进行交互反馈，因此同样不适用于试错代价极其高昂的工业场景。随着大数据采集与处理技术的不断进步，很多企业会在复杂工业设备上安装大量传感器以监测工业系统的实时运行过程。构建时序预测模型并利用采集到的数据集进行训练，然后采用基于预测模型的系统优化及控制策略解决复杂系统的决策难题<sup>[23-25]</sup>。为了实现面向复杂工业系统的建模与预测，需要对复杂工业系统的内在特性进行深入分析，我们发现此类系统通常具有以下典型特征：

- a) **非线性系统动力学**: 大多数工业系统都有极其复杂的高阶动力学方程，它们不是仿射系统或线性系统。
- b) **系统的部分可观测性**: 从传感器或其他可用方法提取的信息是不完整的。特别是，在这样的系统中存在许多未知的隐变量。
- c) **长延迟的影响**: 系统状态受外部输入或在很长一段时间之前发生的内部状态的影响。
- d) **连续时间演化**: 由于真实的工业系统遵循各种物理规律，它们的时间演化可以用 CT 微分方程表示。

对于上述复杂工业系统特点，许多工业界和学术界的科研人员通过构建不同预测模型来解决这些问题。其中，数据驱动方法为建模复杂工业过程最常用的技术<sup>[98]</sup>。特别地，在基于数据的连续时间系统预测及辨识领域中，研究人员主要利用采样于真实系统的有噪数据来拟合高阶微分方程。然而，这种方法不适用于建模部分可观测系统 (Partially Observed System) 和极端复杂的工业系统。伴随着深度神经网络技术的不断发展，得益于其强大的特征表示能力和可扩展的参数结构，深度学习在解决复杂工业系统的预测及分析问题时被广泛使用。如时间序列预测<sup>[24,26-29]</sup>、设备异常检测等<sup>[99]</sup>。然而，现有的大多数基于深度学习的系统建模及预测方法都是基于离散时间的，忽略了系统的连续时间特性。从模型结构与系统本体的一致性角度来看，忽视系统本身具备的物理先验特性会增加模型的自由度及拟合难度，进而限制了模型精度。

另一方面，利用本章节所介绍的复杂系统预测模型，可以从两方面辅助系统控制的研究：

首先，模型的预测结果能够为基于模型的控制算法提供短期预测功能，例如模型预测控制 (Model Prediction Control, MPC)<sup>[24]</sup>。在 MPC 方法中，预测模型提供了动力系统会如何受控制输入影响的先验知识，从而将系统最优控制近似为了序列优化问题。其次，预测模型可以模拟系统在长控制输入信号下的系统输出<sup>[100]</sup>，因此可以作为验证系统控制策略可行性的试错仿真场景。在复杂的过程工业系统

控制领域中，验证控制器的性能和安全性是有极其必要的。与短期预测相比，系统输出模拟需要预测模型有更高的稳健性和稳定性来提供长期的无反馈开环预测。然而，在现有围绕工业系统预测的方法研究中，很少有关于如何设计预测模型以适用于短期和长期预测，进而服务于 MPC 控制及系统仿真。

针对上述工业系统存在的复杂特性以及长短期预测需求，本章节以 ODE-net 作为模型骨架结构，从连续时间域角度拟合复杂工业系统的动态过程，由此提出了一种由序列编码器、状态解码器和导数模块组成的深度连续时间 (Continuous Time, CT) 网络，以端到端的方式学习工业系统输出的自回归变化过程和输入对输出的非线性影响。以实现给定历史系统运行轨迹及未来系统控制输入序列情况下，预测系统输出的未来变化。

具体地，由于工业系统往往存在长时间的系统延迟，因此本章节引入序列编码器以从历史系统运行轨迹中提取特征。在预测时，该方法构建了基于深度连续时间状态空间的导数模块。该模块相比于传统的离散序列模型，如 RNN 等。更适用于建模系统在连续时间域下的非线性演化，并且在模块中通过引入隐状态可以辅助推断系统的不可观测信息。最后，本方法将非平稳系统和平稳系统概念融入在模型的导数模块设计中，可以有效解决系统的短期和长期预测问题。本章研究在工业系统预测领域有三个方面贡献：

a) 提出了一种新的基于深度学习的工业系统连续时间预测模型。深度学习网络由三个部分组成：序列编码器、状态解码器和导数模块。

b) 基于平稳系统和非平稳系统概念，本章设计了两种导数模块，分别处理短期和长期预测任务。

c) 利用来源于某真实铜矿场中膏体浓密机的历史运行数据验证了本章提出方法的有效性。结果表明，该模型在预测具有非线性和高时滞特性的浓密机系统时具有较好性能。此外，本章节还进行了部分消融研究，以评估模型中每个模块的有效性。

本章节内容组织如下：3.1节主要介绍了复杂输入输出系统预测问题的形式化表述并给出基于状态空间模型的表示方法。3.2节主要介绍了基于 ODE-net 的输入输出系统预测模型，并分别介绍其序列编码器、导数模块以及状态解码器三大部分，同时给出了基于伴随状态的模型训练方法。另外，该节分别面向短期预测和长期预测两种情况，介绍了两种导数模块定义方法。最后提出了用于连续化系统输入序列的并行插值方法。3.4节为实验章节，介绍了将本章所述方法用于膏体浓密机系统预测问题的实验结果。3.5节对本章工作进行了总结，并讨论了未来的研究方向。

### 3.1 问题形式化描述

在复杂工业系统中，系统输出量定义为  $y(k) \in \mathbb{R}^n$ ，其中  $k$  为采样时间指标。将影响系统输出量的系统控制输入量定义为  $x(k) \in \mathbb{R}^m$ 。在诸多复杂工业场景中，系

统输入对系统输出的影响本质上是非线性、高时滞的。且由于监测技术的限制，系统的一些关键参数无法获得，系统的观测空间往往是不完备的，同时，观测数据会受到环境噪音以及系统本体中未知扰动的干扰。此类复杂特性使得传统系统辨识方法<sup>[98]</sup>或预测模型<sup>[101]</sup>难以对复杂工业系统进行有效的学习。因此本章采用基于深度学习的黑箱数据驱动方法建模复杂系统的状态空间模型，使模型能够利用序列数据推断系统的隐状态。进一步地，将系统预测问题建模为在已知未来系统输入和历史系统运行轨迹下估计未来系统输出。我们首先定义系统的历史输入序列为  $\mathbf{X}_p^{N_x} = [\mathbf{x}(-N_x), \mathbf{x}(-N_x+1), \dots, \mathbf{x}(-1)]$ ，历史系统输出为  $\mathbf{Y}_p^{N_y} = [\mathbf{y}(-N_y), \mathbf{y}(-N_y+1), \dots, \mathbf{y}(-1)]$ ，系统的未来输入为  $\mathbf{y}_f^M = [\mathbf{y}(0), \mathbf{y}(1), \dots, \mathbf{y}(M-1)]$ ，模型需要预测系统的未来输出  $\mathbf{Y}_f^M = [\mathbf{y}(0), \mathbf{y}(1), \dots, \mathbf{y}(M-1)]$ 。其中， $\mathbf{X}_f^M$  之所以已知，是因为在 MPC 控制及系统仿真问题中，系统输入是已知的。序列预测问题可用如下符号及公式表示：

$$\begin{cases} \mathbf{h}(0) = \mathcal{F}(\mathbf{X}_p^{N_x}, \mathbf{Y}_p^{N_y}), \\ \mathbf{H}^M = D(\mathbf{h}(0), \mathbf{X}_f^M), \\ \mathbf{y}(k) = g(\mathbf{h}(k)), \end{cases} \quad (3-1)$$

模块  $\mathcal{F}(\cdot)$  用于估计初始隐状态  $\mathbf{h}(0)$ ，该隐状态携带了历史序列  $\mathbf{X}_p^{N_x}$  和  $\mathbf{Y}_p^{N_y}$  的信息。构建合理的模型  $D(\cdot, \cdot)$ ，即可利用初始的隐状态以及未来的系统输入，估计完整的系统状态序列  $\mathbf{H}^M$ 。序列中的每个隐状态  $\mathbf{h}(k)$  携带了、或编码了系统在  $k$  时刻的系统信息。另外，通过非线性函数  $g(\cdot)$  解码  $\mathbf{h}(k)$  即可得到估计的系统输出。

公式(3-1)类似于自然语言处理领域的 Seq2Seq 模型<sup>[102]</sup>，将三个序列  $\mathbf{X}_p^{N_x}, \mathbf{Y}_p^{N_y}$ ，和  $\mathbf{X}_f^M$  映射为被预测的系统输出序列  $\hat{\mathbf{Y}}_f^M = [\hat{\mathbf{y}}(1), \hat{\mathbf{y}}(2), \dots, \hat{\mathbf{y}}(M)]$ 。然后，相比于机器翻译等序列转换任务，输入输出系统预测任务要求对于  $\mathbf{h}(k)$  的计算只能依赖于  $\{\mathbf{h}(i), i \leq k-1\}$  和  $\mathbf{X}_f^k = \{\mathbf{x}(i), i \leq k-1\}$ ，这样的限制促使本章在模型框架设计时采用自回归系统。

为了在预测模型中构造一个合理有效的模块  $D$ ，一种简单有效的解决方案是采用单步自回归的离散时间状态空间模型<sup>[103]</sup>：

$$\begin{aligned} \mathbf{h}(k) &= d(\mathbf{h}(k-1), \mathbf{x}(k-1)) \\ \mathbf{y}(k) &= g(\mathbf{h}(k)), \end{aligned} \quad (3-2)$$

其中隐状态  $\mathbf{h}(k)$  将系统的历史轨迹编码为紧密、定长的向量。系统控制输入的影响可以看作是对系统当前隐状态的单步非线性变换。当给定某一时刻输入  $\mathbf{x}(k-1)$  和上一时刻隐状态  $\mathbf{h}(k-1)$  时，模型可以立即预测新的隐状态  $\mathbf{h}(k)$ 。一些先前的研究<sup>[104][105]</sup>表明，大部分过程工业系统的运行本质为物理变化或化学变化，系统的动态过程更适合建模为连续域下的微分方程。因此，本章遵照工业系统的先验

知识，构建参数化连续时间微分方程模型，以拟合隐状态的一阶导数：

$$\dot{\mathbf{h}}(t) = d(\mathbf{h}(t), \mathbf{x}(t)). \quad (3-3)$$

在此处，离散下标  $k$  被替换为连续时间下标  $t$  以表示  $\mathbf{h}(t)$  的变化为连续时间过程。对于复杂系统的预测模型可以进一步表示为式(3-1)和式(3-3)。本章节的目标是利用收集得到的系统运行轨迹数据，对上述模型中的各个参数化模块进行学习，包括序列编码器  $\mathcal{F}(\cdot)$ 、连续时间导数模块  $d(\cdot, \cdot)$  和状态解码器  $g(\cdot)$ 。

## 3.2 基于 ODE-Net 的多输入输出系统预测模型设计

本章采用集成的深度神经网络以参数化拟合公式(3-1)和(3-3)。图 3-1展示了网络中包含的各个组件及其连接。对于给定条件范围内的历史序列轨迹  $\mathbf{X}_p^{N_x}$  和  $\mathbf{Y}_p^{N_y}$ ，以及预测范围内的系统输入序列  $\mathbf{X}_f^M$ ，模型输出  $\hat{\mathbf{Y}}_f^M$  以近似估计系统输出  $\mathbf{Y}_f^M$ 。

该模型的预测过程分成以下几个步骤。首先，采用基于循环神经网络的**序列编码器**，将历史轨迹序列  $\mathbf{X}_p^{N_x}$  和  $\mathbf{Y}_p^{N_y}$  编码为隐状态  $\mathbf{h}(t_0)$ 。然后，将  $\mathbf{h}(t_0)$  作为预测阶段待求解常微分方程的初始状态。常微分方程的一阶导数由模型中的**导数模块**定义，模型将隐状态  $\mathbf{h}(t)$  和连续时间域中任意时刻的外部控制信号  $\mathbf{x}(t)$  作为输入，以估计隐状态  $\dot{\mathbf{h}}(t)$  的瞬时导数。模型中引入的**并行样条插值**模块将离散的外部输入序列  $\mathbf{x}_f^M$  插值为连续时间形式  $\mathbf{x}(t)$  以作为导数模块的输入。接下来，利用 ODE 求解器求解基于初始状态  $\mathbf{h}(t_0)$  和导数模块定义的常微分方程，可以得到时间范围  $[0 \leq t \leq t]$  内的系统隐状态  $\mathbf{h}(t)$ 。最后，模型利用基于多层感知器 (MLP) 网络的**状态解码器**，将求解出的隐状态  $\mathbf{h}(t)$  映射为待预测的系统未来输出  $\hat{\mathbf{Y}}_f^M$ 。接下来，本章将详细介绍所述模型的每个模块的技术细节。

### 3.2.1 基于循环神经网络的历史序列编码

由于大部分工业过程具有较长的时间延迟，因此系统的历史运行轨迹数据对于模型预测极其重要。本模型引入了历史运行数据  $\mathbf{X}_p^{N_x}$  和  $\mathbf{Y}_p^{N_y}$  作为模型输入的一部分。具体地，利用一个基本的 RNN 模型，将  $\mathbf{Y}_p^{N_y}$  和  $\mathbf{X}_p^{N_x}$  两个历史序列编码为某一定长的隐状态，以此推断常微分方程的初始值  $\mathbf{h}(0)$ 。

$$\mathbf{h}(t_0) = \mathbf{h}(0) = f_{\text{RNN}}(\mathbf{Y}_p^{N_y}, \mathbf{X}_p^{N_x}, \theta_f), \quad (3-4)$$

对于一般的工业系统，可以根据生产经验估计其系统时延为  $T_d$ ，数据采样间隔为  $T_s$ ， $N_y$  和  $N_x$  可以近似估计为  $N_y = N_x = N = T_d/T_s$ 。在大部分工业系统中，当前系统状态与短期内的历史轨迹之间的互信息更大。也正是因为该性质，本文利用了 RNN 的遗忘特性，并将其用于编码系统的历史轨迹。利用序列编码器得到的隐状态  $\mathbf{h}(t_0)$  编码了历史系统轨迹中对于预测所需的信息，该状态将作为待解 ODE-net

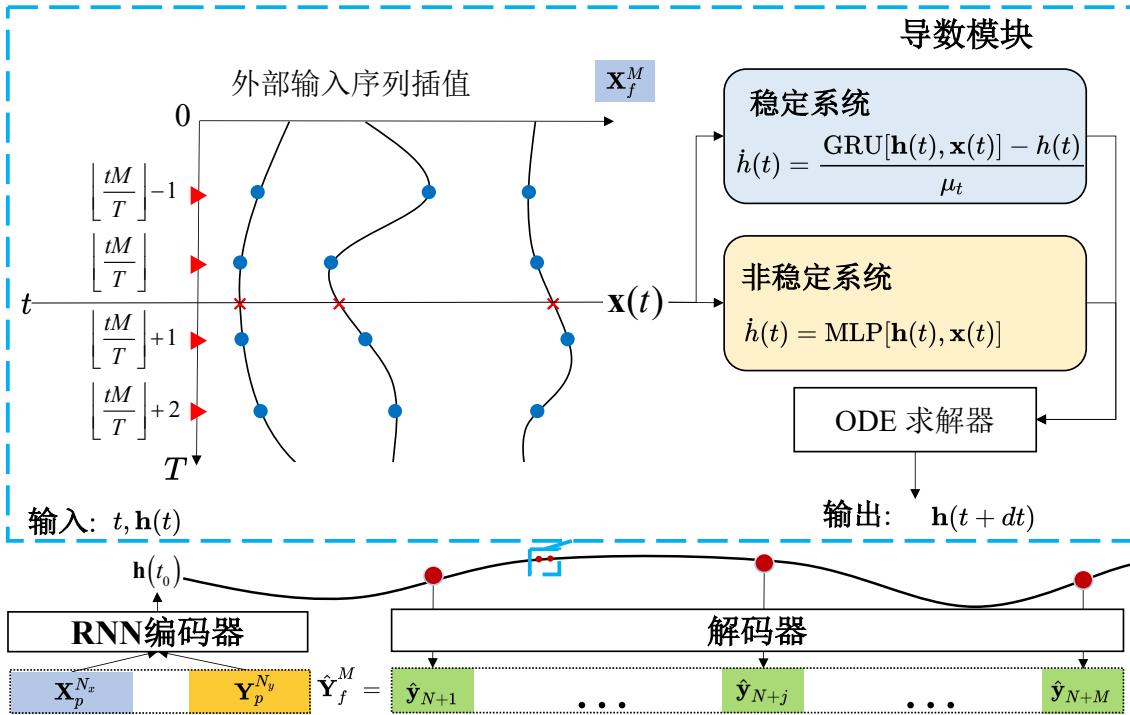


图 3-1 基于 ODE-net 模型的输入输出系统预测模型整体结构

的初始状态。

### 3.2.2 基于可微常微分方程网络构建系统状态空间模型

本小节中，我们采用参数化的连续时间状态空间模型表示系统输入、隐藏状态和输出之间的关系：

$$\dot{\mathbf{h}}(t) = d(\mathbf{h}(t), \mathbf{x}(t), \theta_d), \quad (3-5)$$

$$\mathbf{y}(t) = g(\mathbf{h}(t)). \quad (3-6)$$

状态空间模型能够将系统状态表示为定长的编码向量  $\mathbf{h}(t)$ ，隐藏状态  $\mathbf{h}(t)$  的使用对于建模长时延系统和不完全观测系统至关重要。

对于长度等于  $M$  的待预测序列，我们在整数离散索引  $[0, 1, \dots, M]$  和连续时间范围  $[t_0 \leq t_k \leq t_M]$  之间构造一个双射函数。给定初始状态为  $\mathbf{h}(t_0)$ ，每个  $\mathbf{h}(t_k)$  为 ODE 方程在  $t = t_k$  处的解。为了构造一个可学习的微分系统，本章使用可微 ODE-net<sup>[106]</sup> 学习上述状态空间模型。

对于由某一预测评价指标确定的标量损失函数  $L(\cdot)$ ，其输入  $\mathbf{h}(t_k)$  为 ODE 方

程在  $t = t_k$  时刻的解。将 ODE 求解器表示为 `ODESolve`, 我们有

$$\begin{aligned} L(\mathbf{h}(t_k)) &= L\left(\mathbf{h}(t_0) + \int_{t_0}^{t_k} d(\mathbf{h}(t), \mathbf{x}(t), \theta_d) dt\right) \\ &= L(\text{ODESolve}(\mathbf{h}(t_0), d, t_0, t_k, \theta_d)). \end{aligned} \quad (3-7)$$

本文待求解的 ODE 方程是参数化的, 为了训练参数  $\theta_d$  以最小化  $L(\cdot)$ , 我们需要根据上式计算损失函数对模型参数梯度  $\partial L / \partial \theta_d$ 。此处需要引入伴随状态 (adjoint), 即损失函数对隐状态的梯度  $\mathbf{a}(t) = \partial L / \partial \mathbf{h}(t)$ .

伴随状态的动态过程可由另一个 ODE 方程来描述, 根据链式法则推导如下:

$$\frac{d\mathbf{a}(t)}{dt} = -\mathbf{a}^\top(t) \frac{\partial d(\mathbf{h}(t), \mathbf{x}(t), \theta_d)}{\partial \mathbf{h}(t)}. \quad (3-8)$$

依赖于伴随状态, 损失函数  $L(\cdot)$  对参数  $\theta_d$  的梯度可以通过求解第三个常微分方程得到:

$$\frac{\partial L}{\partial \theta_d} = - \int_0^{t_M} \mathbf{a}^\top(t) \frac{\partial d(\mathbf{h}(t), \mathbf{x}(t), \theta_d)}{\partial \theta_d} dt. \quad (3-9)$$

详细的证明参见<sup>[106]</sup>. 在确定的网络结构  $d$  和参数  $\theta_d$  下, 任意的求解 ODE 方程的数值计算方法, 如欧拉法、中点法和龙格-库塔法, 均可同时求解三个微分方程, 进而获得  $\mathbf{h}(t)$ 、 $\mathbf{a}(t)$  和  $\partial L / \partial \theta$  在任意时刻的解。

一般情况下, 在使用数值方法求解常微分方程时, 具有较低误差容忍度的 ODE 求解器会增加调用微分函数  $d$  的频率。造成更多的时间消耗, 但其估计结果具有更高的准确性。当使用神经 ODE 网络建模时间序列数据集时, 这个准则同样是成立的。在实验环节, 本章也探讨了不同微分方程求解器对于预测精度和消耗时间的影响。时间成本和准确性的详细比较见 3.4 节。

对于在给定时刻, ODE-net 输出的状态  $\mathbf{h}(t)$ , 需要使用状态解码器将其转换为系统的预测输出。本文设计模型中采用的状态解码器本为全连接的网络:

$$\hat{\mathbf{y}}(t) = \mathbf{V}^\top \tanh(\mathbf{W}\mathbf{h}_t + \mathbf{b}_w) + \mathbf{b}_v. \quad (3-10)$$

一般的状态空间模型多使用普通的矩阵变换以构建编码空间到系统输出空间的映射。本文中, 由于公式 (3-11)中的累加形式使输入  $\mathbf{h}(t)$  的范围是非确定性的, 而实际系统的输出空间是有界的, 因此选择带有  $\tanh$  非线性归一化的解码器以限制解码器输出在合理的范围内。

### 3.2.3 面向不同预测时长需求的常微分方程导数模块定义

在3.2.2节中, 介绍了 ODE 方程的求解和参数梯度的求解方法。在本节中, 将具体探讨 ODE 方程的参数化结构定义。最基本的 ODE-net 模型采用普通的多层感

知机神经网络估计状态导数<sup>[18]</sup>，本文将此模型称为非平稳模型：

$$d(\mathbf{h}(t), \mathbf{x}(t), \theta_d) = \text{MLP}(\mathbf{h}(t), \mathbf{x}(t), \theta_d) \quad (3-11)$$

其中  $\text{MLP}(\cdot)$  表示多层感知器。使用数值 ODE 求解方法求解非平稳系统与普通的残差网络 (ResNet) 有很强的相似性。

在随机过程分析领域，非平稳系统指时间序列在各点处的均值和协方差随时间变化的随机过程<sup>[107]</sup>。差分操作<sup>[108]</sup>是一种消除序列趋势和季节性，使非平稳时间序列平稳的有效方法。一般来说，复杂工业系统的输出往往属于非稳定时间序列，带有较强的趋勢性和周期特性。采用差分操作可以有效提高拟合精度。在公式(3-11)中，导数模本质上学习了隐空间中隐状态的一阶差分。相比于 ARIMA 模型等直接对系统输出进行差分，面向隐状态的差分模型具有同等或更强的表示能力，以描述系统的多阶差分，进而建模更高阶的非平稳系统。

然而，非平稳系统(3-11)在处理长期预测任务时也面临着严重的问题。在求解长区间的 ODE-net 时，连续时间域内隐状态导数的不断积分会导致隐状态的取值范围显著扩增。因此，模型的估计误差会相应增大，导致解码器难以准确地估计系统输出。

因此，本文设计了一种平稳系统作为模型的导数模块，以处理模型的长期预测问题。具体地，其隐状态导数定义为：

$$d(\mathbf{h}(t), \mathbf{x}(t), \theta_d) = \frac{1}{\mu_t} (\text{GRU}(\mathbf{h}(t), \mathbf{x}(t), \theta_d) - \mathbf{h}(t)) \quad (3-12)$$

其中 GRU 表示门控循环单元。在平稳系统中， $\text{GRU}(\mathbf{h}(t), \mathbf{x}(t), \theta_d)$  根据当前外部输入  $\mathbf{x}(t)$  和隐状态  $\mathbf{h}(t)$  定义了隐状态的移动目标点。其中，因子  $\mu_t$  规范了到达该目标的速度。同时，由 GRU 网络的特性可知，其输出范围始终在  $(-1, 1)$  中。无论微分方程求解时间区间范围有，隐状态  $\mathbf{h}(t)$  将始终朝向 GRU 网络输出的目标值移动，因此无论微分方程求解时间区间有多大，求解 ODE 方程得到的任意时刻隐状态的范围都是稳定的，更便于解码器模块构建从隐空间到系统输出空间的稳定映射。在长期预测任务中，相比于非稳定系统，这一特性能够显著地提升模型预测的稳定性。

对于平稳系统，我们主要将其用于长期预测。由于 GRU 具有较强的携带长时间信息的能力，所以我们使用 GRU 网络来构造导数模块。对于非平稳系统中，主要将其用于短期预测。由于隐状态导数的连续累积，隐状态的变化类似于维纳过程，将会在无约束的范围内不断扩散。而 GRU、LSTM 等网络均假设模型输入应限定在有界范围内。因此，本文在非稳定系统中利用 MLP 模块学习给定  $\mathbf{h}(t)$  和  $\mathbf{x}(t)$  下  $\dot{\mathbf{h}}(t)$  的一阶导数。

### 3.3 离散输入序列的可微并行插值方法及模型训练

在等式(3-11)和(3-12)中,  $\dot{h}(t)$  的计算依赖于外部输入  $x(t)$ , 而训练数据中的外部输入序列  $X_f^M$  是离散的。ODE-net 的计算需要在连续时间范围内给定任意时刻的系统外部输入值  $x(t)$ 。因此, 在网络执行前向传播前, 需要将离散的外部输入序列转换到连续时间域中。

在深度神经网络的训练中, 通常需要利用图形处理单元 (GPU) 的并行计算能力, 将数据批量地送入模型并进行参数更新。因此, 本章基于标准的样条插值算法, 实现了一种并行样条插值机制, 能够将批量输入的离散序列, 并行地插值为连续时间信号。

不失一般性的, 为了便于表述, 我们假设系统输入的维度  $m$  为 1。给定以批 (batch) 的方式组织的输入序列为  $\mathbf{X} = [x^1, x^2, \dots, x^B]$ , 其中  $B$  为批大小。每个向量  $x^i = [x_1^i, x_2^i, \dots, x_M^i]$  为独立的输入序列, 由  $M$  个采样数据组成, 相邻数据点之间的采样间隔是一致的。假定与  $M$  个采样点相对应的求解常微分方程的时间区间为  $[0, T]$ 。对于任一给定的时间索引  $t, 0 \leq t \leq T$ , 我们希望并行化地估计  $\mathbf{X}(t) = [x^1(t), x^2(t), \dots, x^m(t)]$ 。首先构造离散下标  $k = \lfloor \frac{tM}{T} \rfloor$ , 给定  $n+1$  个数据点,  $\{(k, x_k), (k+1, x_{k+1}), \dots, (k+n, x_{k+n})\}$  可以计算  $n$  阶样条插值函数的系数矩阵  $A \in \mathbb{R}^{(n+1) \times m}$ :

$$A = \begin{bmatrix} k^0 & \cdots & k^n \\ (k+1)^0 & \cdots & (k+1)^n \\ \vdots & \ddots & \vdots \\ (k+n)^0 & \cdots & (k+n)^n \end{bmatrix}^{-1} \begin{bmatrix} x_k^1 & \cdots & x_k^m \\ x_{k+1}^1 & \cdots & x_{k+1}^m \\ \vdots & \ddots & \vdots \\ x_{k+n}^1 & \cdots & x_{k+n}^m \end{bmatrix}. \quad (3-13)$$

一批数据中, 所有离散序列在  $t$  时刻的插值结果如下:

$$[x^1(t), x^2(t), \dots, x^m(t)] = \left( \left[ 1, \frac{tM}{T}, \dots, \left( \frac{tM}{T} \right)^n \right] A \right). \quad (3-14)$$

在一般的深度学习框架中, 上述矩阵乘法和求逆矩操作可以高效地并行实现。由于上述并行插值过程以及模型中的编码、解码以及常微分方程求解过程都是可微的, 本章采用标准的反向传播算法训练完整模型, 损失函数定义为标准的均方误差损失函数:

$$\mathcal{O}(\hat{Y}^M, Y^M) = \frac{1}{M} \sum_{i=1}^M |y_i - \hat{y}_i|^2. \quad (3-15)$$

### 3.4 实验验证与分析

本节给出了所述方法在真实膏体浓密机系统数据集上的预测实验结果。实验只要探究三个问题:

**问题 1:** 相比于直接采用离散时序预测模型, 使用深度连续时间网络和高精度

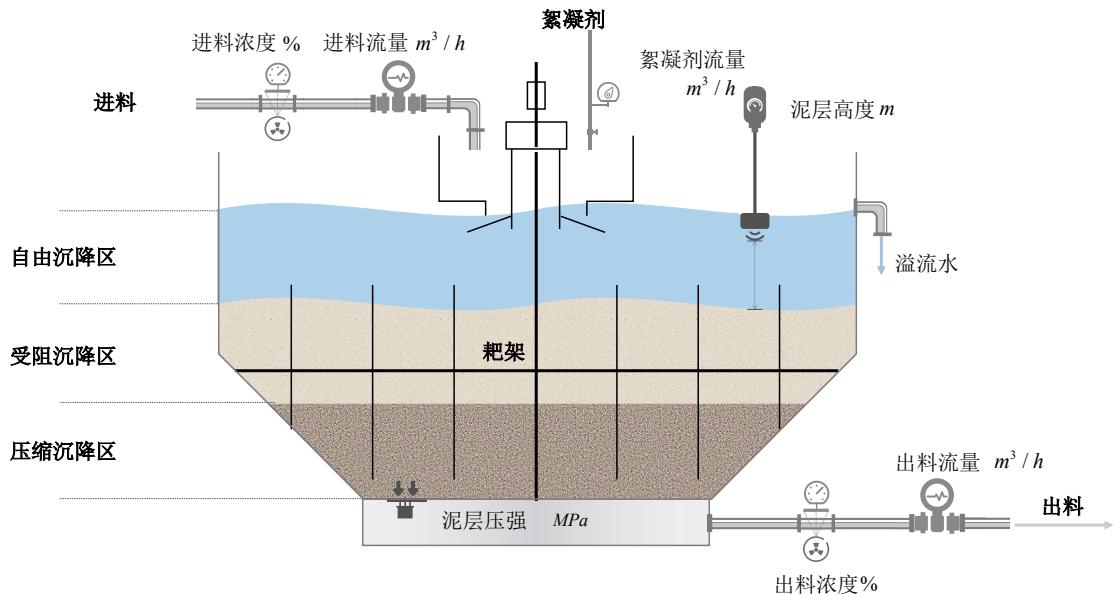


图 3-2 膏体浓密机系统运行过程及监测变量图示

ODE 求解器建模预测工业系统能否获得更优的效果?

**问题 2:** 在预测任务中使用平稳系统和非平稳系统的优缺点是什么?

**问题 3:** 不同的插值方法和序列编码器的编码长度会如何影响所提出的 CT 模型的准确性? 本节中, 我们将首先介绍数据集、模型超参数以及训练和测试参数的配置。然后给出详细的实验结果。

### 3.4.1 膏体浓密机系统数据集

在实验环节中, 本章所用的工业系统数据来自于赞比亚铜带省, NFCA 非洲矿业有限公司的膏体浓密机。该设备由 FLSmidth 公司生产。如图 3-2 所示。该设备用于在膏体充填站将铜尾矿浓缩成高浓度料浆以用于充填膏体的制备。两台设备都运行在闭环 PID 控制模式下。实验所用全部数据采集于 2018 年 5 月到 2019 年 2 月, 采集间隔为两分钟。表 5-2 展示了部分数据。收集的数据集来自于 7 个监测传

表 3-1 膏体浓密机系统数据样例

采集时间	进料 流量	进料 浓度	泥层 压力	耙架 转速	底流 流速	底流 浓度	絮凝剂 流速
2018/5/9 10:20	164.47	16.47	18.41	500.58	58.96	59.72	4.30
2018/5/9 10:22	169.21	15.51	17.99	500.16	61.56	58.88	4.06
2018/5/9 10:24	141.78	15.30	16.41	500.56	59.97	59.26	4.06
2018/5/9 10:26	305.67	25.31	16.11	500.99	59.46	58.77	4.07
2018/5/9 10:28	328.70	28.28	16.43	501.42	59.68	59.43	4.43
2018/5/9 10:30	323.96	25.90	17.11	501.56	61.40	60.09	4.40

感器, 系统输出变量  $y(k) \in \mathbb{R}^2$  中, 包括底流浓度  $y_1(k)$  和泥层压力  $y_2(k)$ 。 $y_1(k)$  和  $y_2(k)$  均受到控制输入  $x(k) \in \mathbb{R}^5$  影响, 包括进料流量  $u_1(k)$ 、进料浓度  $u_2(k)$ 、

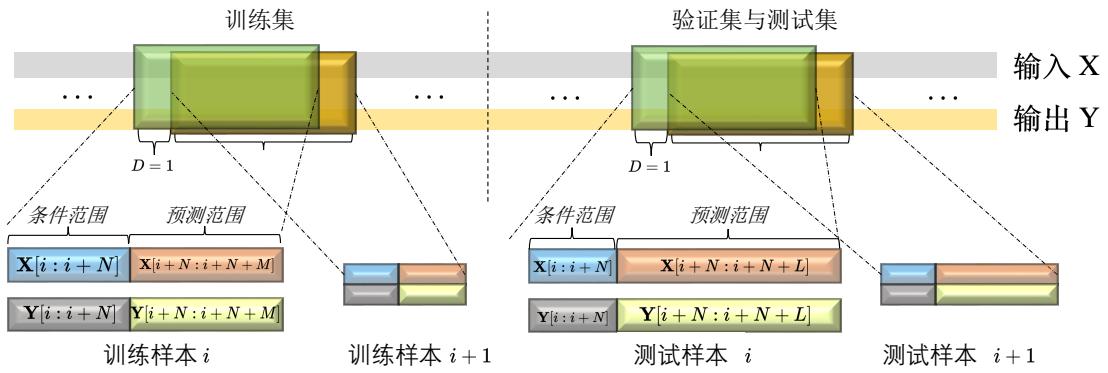


图 3-3 训练集、验证集、测试集的构建过程图示

耙架转速  $u_3(k)$ 、底流流量  $u_4(k)$  和絮凝剂流量  $u_5(k)$ 。删除系统停机时的异常监测数据，累计剩余 24,673 条数据。

本节采用滑动窗口法生成训练及测试样本对  $(\mathbf{X}_p^N, \mathbf{Y}_p^N, \mathbf{X}_f^M, \mathbf{Y}_f^M)$ 。具体地，原始数据集依照浓密机系统的启停时间，分为多个文件。每个文件包含了连续生产过程中 7 个传感器的监测数据序列，且各序列中数据点的采样间隔均为 2 分钟。对于每个文件中的多个序列，将大小为  $N + M$  的滑动窗口沿着序列的时间维度正向移动，移动步长为 1。当窗口到达  $i$  位置时，四个序列 ( $\mathbf{X}_p^N = \mathbf{X}[i:i+N]$ ,  $\mathbf{Y}_p^N = \mathbf{Y}[i:i+N]$ ,  $\mathbf{X}_f^M = \mathbf{X}[i+N:i+N+M]$ ,  $\mathbf{Y}_f^M = \mathbf{Y}[i+N:i+N+M]$ ) 被收集作为训练或测试样本。

在验证集和测试集数据中，滑动窗口的大小设置为  $N + L$ ，其中  $L$  表示待预测序列的长度（可能不等于训练集中待预测序列的长度  $M$ ）。模型仅通过训练集数据进行训练，然后使用不同的  $L$  在不同的验证集和测试集数据上进行验证和评估。构造数据集的详细过程如图 3-3 所示。

本节使用数据集的前 70% 用于模型训练。在剩下的 30% 中，前 15% 作为验证集，帮助确定最佳训练轮次，剩下的 15% 作为测试数据集用于评估模型准确性。训练时，历史序列的长度为  $N = 80$ ，预测序列长度为  $M = 60$ 。测试环节探究了  $L = 60, 200, 500$  三种情况下，模型的预测性能。根据图 3-3 所示的构建输入-输出序列对的方法，共计 17131 个样本对用于训练。对于不同预测长度  $L = 60, 200, 500$ ，分别有 3561、3421、3121 个样本对用于测试和验证。所有的数据在训练和测试之前均被归一化为标准正态分布。

### 3.4.2 实验设定

实验中，使用 mini-batch 随机梯度下降 (SGD) 和 Adam 优化器<sup>[109]</sup> 对模型进行训练。batch size 为 512，学习速率为 0.001，并且呈指数衰减。衰减率为 0.95，衰减周期为 10 个训练轮次。ODE-net 中隐藏状态  $\mathbf{h}(t)$  的大小为 32。RNN 序列编码器模块包含一个隐藏层，隐状态大小等于 32，这与隐藏状态  $\mathbf{h}(t)$  的大小一致。状态解码器中隐藏层的大小为 64。在设定自适应微分方程求解器参数时，由于降低近似误

差的容忍度，会导致求解常微分方程的时间剧烈增加。在所有的实验中，为了平衡时间消耗和预测精度，我们将所有 ODE 求解器容忍的最大相对误差设置为  $1e-4$ ，容忍的最大绝对误差设置为  $1e-5$ 。训练时，找到在验证集上预测精度最高的模型，将其用于测试数据集精度评估。训练和测试均在单个 Nvidia V100 GPU 上进行。代码基于 PyTorch 框架实现。对于给定的离散整数下标序列  $[0, 1, \dots, M]$ ，我们定义其连续时间区间为  $0 \leq t \leq M\delta_t$ 。相邻数据点的时间间隔  $\delta_t$  被设置为 0.1。因此，公式(3-12)中的归一化因子  $\mu_t$  也被设置为 0.1。当使用欧拉数值求解器求解稳定系统的 ODE 方程时，模型将等价于使用 GRU 单元在离散时间系统下进行预测：

$$\begin{aligned} \mathbf{h}(t + \delta_t) &= \mathbf{h}(t) + \delta_t \cdot \frac{\text{GRU}(\mathbf{h}(t), x(t), \theta_d) - \mathbf{h}(t)}{\mu_t} \\ &= \text{GRU}(\mathbf{h}(t), x(t), \theta_d). \end{aligned} \quad (3-16)$$

本节使用模型预测底流浓度的平均根相对平方误差 (RRSE) 和平均平方误差 (MSE) 评价不同模型的预测精度。对于预测长度  $L$ ，RRSE 定义如下：

$$\text{RRSE} = \sqrt{\sum_{j=1}^L \frac{e_j^2}{(\hat{y}_j - \bar{y})^2}}, \quad e_j = \hat{y}_j - y_j. \quad (3-17)$$

RRSE 可以解释为被归一化的均方根 (RMS) 误差。

### 3.4.3 不同模型预测结果对比

首先，我们研究了不同 ODE 求解器和导数模块对预测精度的影响。我们选择了四个 ODE 求解器：Euler, midpoint, 四阶 Runge—Kutta (RK4), Dormand—Prince (Dopri5)<sup>[106]</sup>，以及三阶 Bogacki—Shampine(Bosh)<sup>[110]</sup>。我们研究了这些微分方程求解器结合非稳定和稳定系统时的性能。此外，我们还对比了离散时间深度序列模型，包括状态空间 (DT-State-Space)、基于注意力机制的 Seq2Seq 模型 (Attention-Seq2Seq)<sup>[24]</sup> 和 Transformer 模型<sup>[111]</sup>。DT-State-Space<sup>[112]</sup> 模型采用循环神经网络 (RNN) 建模线性状态空间模型的参数，并基于预测得到的状态空间模型预测时间序列。其状态空间和 RNN 隐藏层的大小分别设置为 16 和 32。Transformer 和 Attention-Seq2Seq 的超参数设置与原文献保持一致。

我们进行了三组实验以探究不同预测长度  $L = 60, 200$  和  $500$  下，不同模型预测结果的 RRSE、MSE 和预测速度。

在表 3-2 中可以发现，离散时间域下的 Attention-Seq2Seq 模型、DT-State-Space 模型和 Transformer 的性能稍优于使用 Euler 求解器的模型效果，但差于使用高阶 ODE 求解器获得的预测效果，特别是在长期预测  $L = 200, 500$  时表现更为明显。结果表明，采用连续时间模型可以更好地反映浓密机系统的连续时间演化特征，从而相比于纯离散时间模型，呈现更高的预测精度。

表 3-2 底流浓度预测的相对方根误差 (RRSE)、平均平方误差 (MSE)、时间消耗

模型		L = 60			L = 200			L = 500		
		RRSE	MSE	Time (s)	RRSE	MSE	Time (s)	RRSE	MSE	Time (s)
非稳定 系统	Euler	3.18	9.07	1.71	5.09	80.25	3.81	3.95	152.21	4.65
	Mid-Point	3.10	8.95	3.23	5.24	80.29	7.36	4.16	172.43	9.15
	RK4	3.10	8.97	6.95	5.24	83.90	14.82	4.16	172.64	18.76
	Bosh	3.08	8.57	12.8	5.84	84.60	19.0	4.61	172.39	24.75
	Dopri5	<b>2.83</b>	<b>6.40</b>	9.63	5.31	84.60	13.8	4.19	175.39	25.75
稳定 系统	Euler	3.18	9.06	1.63	3.75	34.78	3.58	1.63	37.77	4.66
	Mid-Point	3.18	9.08	3.22	3.73	34.64	7.17	1.62	38.36	9.3
	RK4	3.18	8.96	6.80	<b>3.58</b>	<b>32.90</b>	15.17	<b>1.61</b>	<b>34.88</b>	18.66
	Bosh	N/A	N/A	>50	N/A	N/A	>200	N/A	N/A	>3000
	Dopri5	N/A	N/A	>50	N/A	N/A	>200	N/A	N/A	>3000
Attention-Seq2Seq <sup>[24]</sup>		3.13	8.97	0.41	4.02	33.90	0.41	1.82	40.53	0.42
DT-State-Space <sup>[112]</sup>		3.22	9.36	0.06	4.69	41.11	0.07	3.35	45.64	0.08
Transformer <sup>[111]</sup>		3.16	8.36	0.02	3.99	40.23	0.02	2.55	44.23	0.03

### 3.4.4 不同 ODE 求解器对比

进一步地，我们分别对采用了不同 ODE 求解器的稳定系统模型和非稳定系统模型进行分析比较。

当导数模块被定义为非稳定系统时，对于短期预测任务  $L = 60$ ，我们发现，虽然 Euler 求解器的时间消耗低于其他 ODE 求解器，但其预测结果的 RRSE 和 MSE 指标较高，预测精度差于使用其他四个 ODE 求解器。原因在于欧拉方法作为求解 ODE 方程最简单的方法，它在两个相邻时间点之间仅仅调用导数模块计算隐状态的一阶导数一次，然后计算状态差分。其运行本质等同于离散时间序列模型，算得 ODE 方程的解的精度较差，并没有充分利用模型为连续时间系统的性质。相应地，中值法和 4 阶 Runge-Kutta 法在两个相邻时间点之间，分别调用了导数模块计算隐状态导数 2 次和 4 次。因此，其对隐状态轨迹的求解精度高于欧拉法。该结果也证实了，基于 Euler 方法的离散时间序列模型忽略了系统的连续时间特性，因此不能精确地学习系统的动态过程。进一步地，作为隶属于龙格-库塔体系中的常微分方程自适应步长求解方法，Dopri5 和 Bosh 方法能够确保数值近似解与真实解之间的误差限定在指定误差范围内。随着给定误差限的缩小，求解 ODE 方程的时间消耗也随之增加。虽然两种方法求解 ODE 耗时较长，但精度较高。Dopri5 的预测性能稍好于 Bosh。

当导数模块定义为平稳系统时。两种自适应方法 Bosh 和 Dopri5 求解 ODE 方程的时间消耗将显著增加。在表3-2中，我们没有列出 Dopri5 和 Bosh 在稳定系统中的预测精度，因为其计算速度极慢，使得该方法不适用于实际工业应用。对比其他 ODE 求解器，我们发现 RK4 等高阶 ODE 求解器的求解误差小于低阶 ODE 求

解器，但在求解 ODE 方程时需要消耗更多的时间。

### 3.4.5 稳定系统与非稳定系统对比

为了更直观地比较使用稳定系统和非平稳系统定义导数模块的区别，我们进一步地可视化了不同 ODE 求解器对非稳定和稳定系统的预测效果。图3-4描述了对于  $L = 60$  的短期预测任务，使用不同 ODE 求解器求解非平稳系统和平稳系统得到的预测序列：

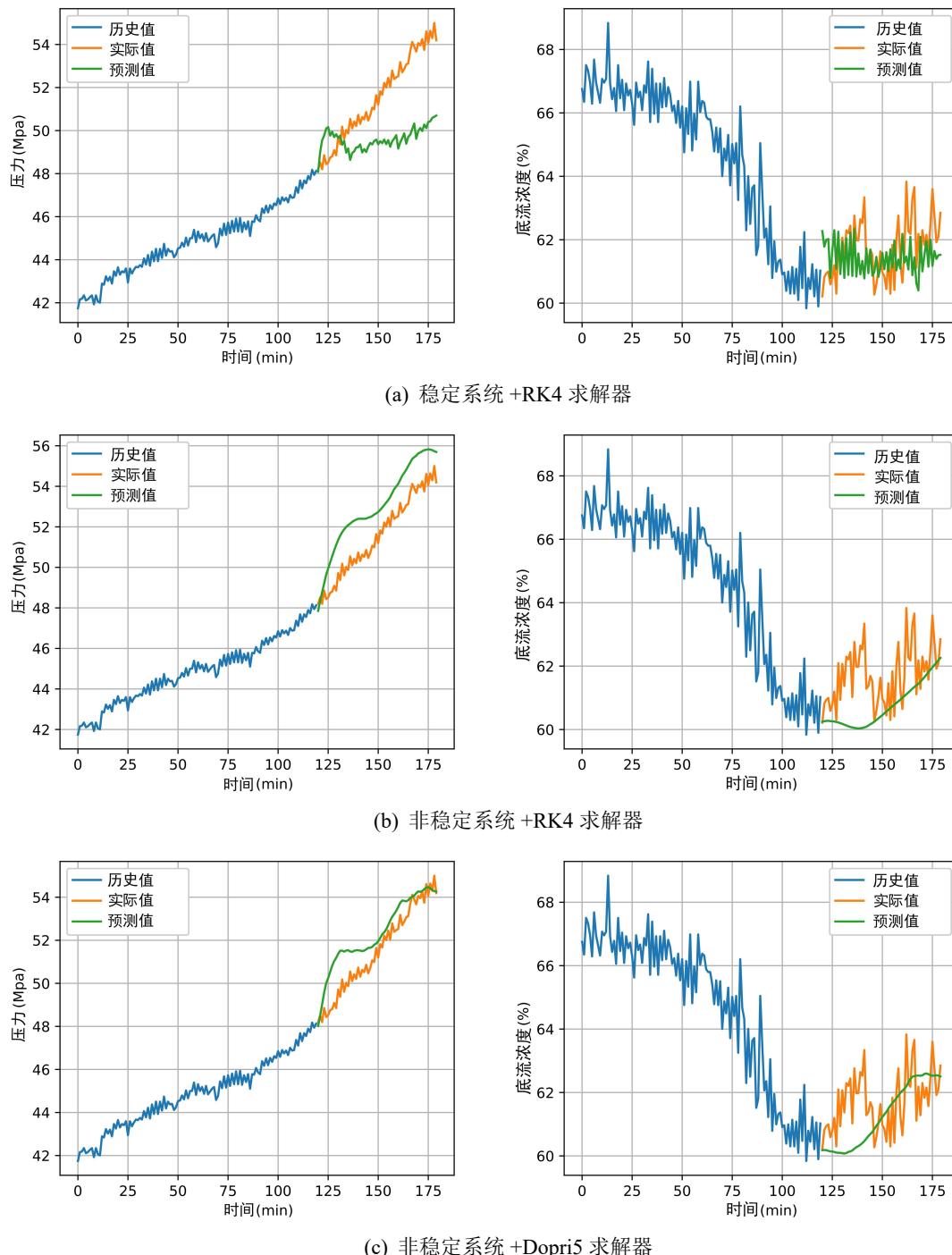


图 3-4 不同系统及不同 ODE 求解器在  $L = 60$  短期预测任务中的性能比较

结果表明，非平稳模型在短期预测任务中的表现优于平稳模型。非平稳模型的估计序列比平稳模型的估计序列更接近实际系统输出。非平稳系统的学习过程本质上相当于对隐状态进行差分，利用 MLP 网络学习平稳的系统一阶差分。此外，由于非稳定系统结构限制了隐状态只能以连续、缓慢的方式变化。因此模型可以平滑地预测系统的输出。该约束符合浓密机系统运行缓慢的特性，等价于缩小了模型参数的搜索空间，抑制了模型过拟合的情况。

图3-5展示了长期预测任务 ( $L = 200$ ) 中的模型预测结果。图中呈现的预测效

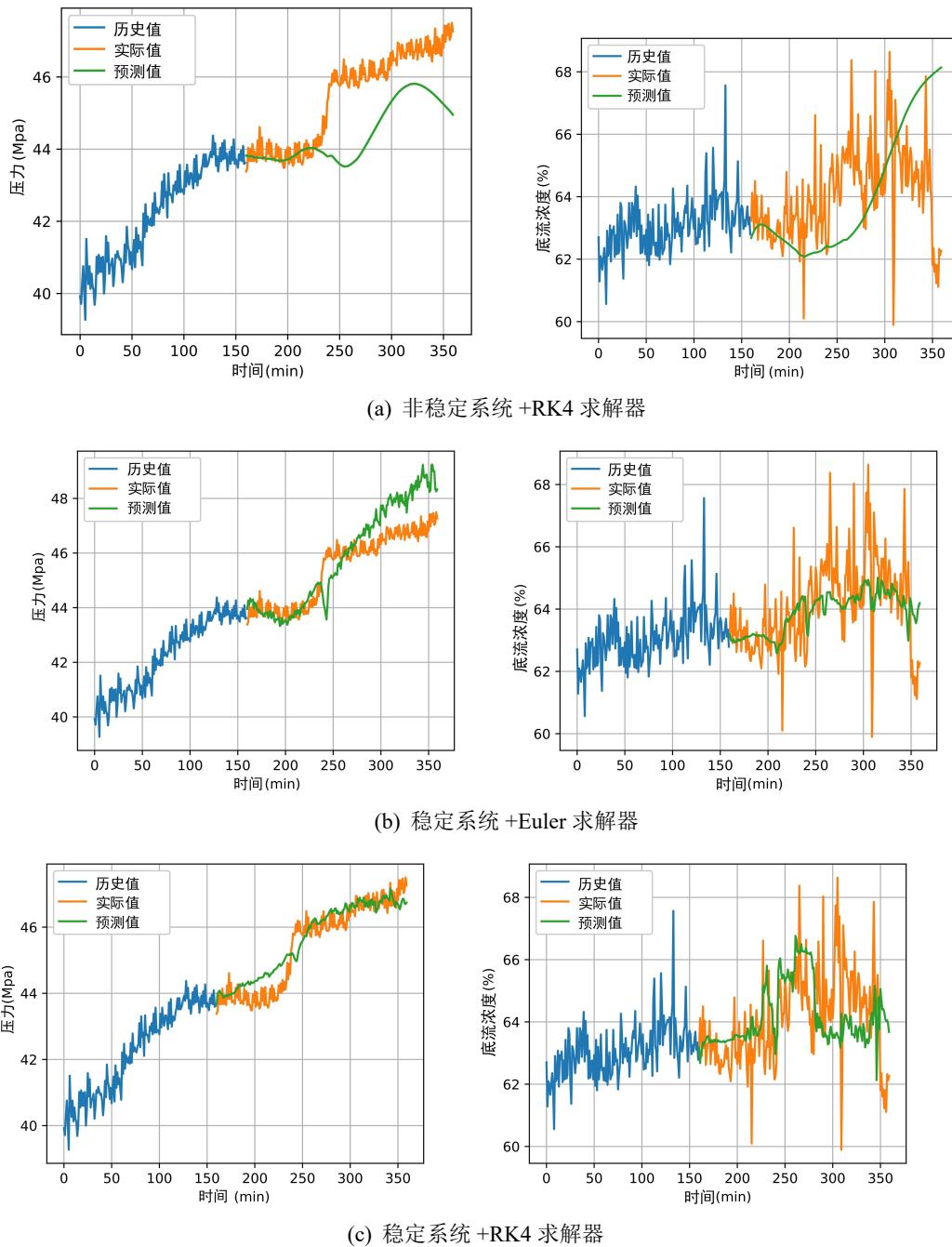


图 3-5  $L = 200$  时，不同 ODE 求解器、系统动态的预测效果对比

果与表3-2中的结果几乎一致。在长期预测任务中，非平稳模型的 RRSE 和 MSE

远远高于平稳模型，模型预测效果更差。从图3-4可以看出，非平稳模型在短期预测问题中获得较好的预测效果。而在图3.5(a)中，非稳定系统在长期预测场景下的预测精度显著衰减，且预测偏离程度伴随着预测序列长度的增加而增加。与之相对地，稳定系统的预测结果是稳定的，更接近于系统的真实输出，这证明了稳定系统模型在长期预测问题中具有更好的准确性。在由非稳定系统定义的导数模块中，其内部结构导致了隐状态在积分过程中是无约束的，状态的取值空间将逐渐扩张。虽然模型在解码器网络中嵌入  $\tanh$  函数，能够将预测的系统输出限制在合理的范围内，但解码器模块无法学习有效的映射函数，实现从极大的隐状态空间到系统输出空间的准确映射。

同样地，图 3-5也证明了在长期预测问题中，高阶 ODE 求解器（如 4 阶 Runge—Kutta）能够获得比低阶 ODE 求解器（如 Euler）更好的预测效果。

从实验结果可以看出，采用自适应求解器的连续时间 ODE-Net 的预测表现优于其他欧拉近似法和龙格库塔近似法。说明微分方程求解的精度会严重影响模型预测的精度。

虽然本文所提出的连续时间 ODE-Net 模型在预测精度上表现良好，但与其他模型相比，推理速度较慢。表3-2的最后一列表示预测序列长为  $L$  时，模型的平均消耗时间。因为在 ODE-net 的训练和推理中，其前向传播和反向传播过程都需要大量地计算隐状态导数。因此在预测系统动态变化时，其时间消耗远高于离散时间模型。一些前沿研究<sup>[52,53,113]</sup> 聚焦于优化 ODE-net 的训练或推理速度。这些方法为提高本文模型的预测效率提供了有借鉴意义的指导。

本节额外进行了数组实验，以评估在不同预测长度  $L$  下，稳定系统和非稳定系统预测的底流浓度误差 (MSE) 变化。图 3-6展示了五次重复实验中，底流浓度的预测误差波动情况 ( $\log_{10}\text{MSE} \pm 2\sigma$ )。虽然在长期预测任务中稳定系统优于非稳定系统，但非稳定系统在短期预测任务中的表现优于稳定系统 (如  $L < 100$ )。大约当  $L$  超过 120 时，非稳定系统的误差随预测长度的增加而显著增加。

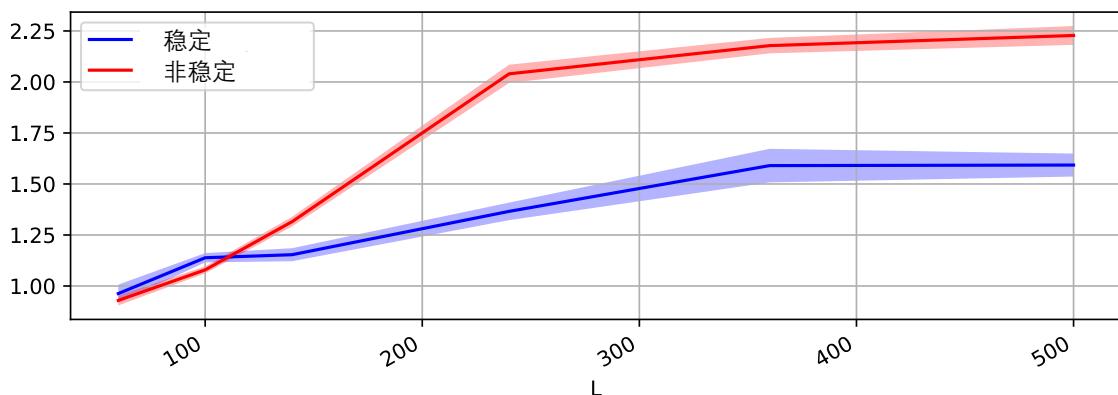


图 3-6 不同预测长度下稳定系统和非稳定系统的预测精度变化

### 3.4.6 探究序列插值阶数的对预测精度的影响

本小节将进行消融实验，以探究对离散输入序列进行插值的方法是否对预测精度产生影响。我们测试了四种不同阶数的样条插值方法，并比较了模型在测试集上的预测精度。表 3-3 的结果表明，模型中采用高阶插值的精度将稍优于低阶插值。这证明了本章所研究的膏体浓密机系统为复杂非线性受控系统，受控输入信

表 3-3 不同插值方法对预测精度的影响

模型	$L = 60$ (120 分钟)		$L = 200$ (400 分钟)		$L = 500$ (1000 分钟)	
	RRSE	MSE	RRSE	MSE	RRSE	MSE
三次样条插值	<b>3.083</b>	<b>8.565</b>	<b>3.581</b>	32.90	1.615	<b>34.88</b>
二次样条插值	3.097	8.993	3.593	<b>32.585</b>	<b>1.613</b>	36.741
线性插值	3.098	8.999	3.763	33.530	1.627	37.778
零阶样条插值	3.115	9.050	3.791	33.585	1.628	37.695

息对于预测系统的输出是必不可少的。高阶样条插值方法能够更充分地利用相邻位置离散输入信号的相关特征，相比于低阶插值方法，能够更精确地对空白区域进行插值填充。

### 序列编码器的有效性验证及系统时延探究

最后，本节探究了引入序列编码器对于解决含时延系统预测问题的有效性。具体地，我们研究了序列编码器的有无以及不同的编码长度  $N$  对模型预测精度的影响。特别地，当  $N$  设置为 1 时，将顺序编码器替换为具有单一隐藏层的神经网络，该神经网络将单步系统输出  $y(k-1)$  及输入  $x(k-1)$  编码为 ODE 系统的初始隐状态  $h(t_0)$ 。当  $N$  设置为 0 时，初始状态  $h(t_0)$  为可学习的初始隐状态<sup>[100]</sup> 或零向量，与历史系统轨迹不相关。

在不同的预测序列长度  $L = 60$ 、 $200$  和  $500$  的三个实验中，我们测试了不同的  $N$  对预测精度的影响。在  $L = 60$  的实验中，导数模块被设置为一个非平稳系统。当  $L = 200$  和  $500$  时，导数模块定义为带有 GRU 单元的稳定系统。所有实验中，求解 ODE 方程采用四阶龙格-库塔求解器。

实验结果如表 3-4 所示，相比于令  $N = 0$ ，完全忽略系统历史输出，引入序列编码器并从历史运行轨迹中提取特征能够获得更好的预测精度。

其直观解释为，待预测的系统输出序列与历史系统轨迹具有很强的统计相关性，从后者提取的特征对于序列预测有很重要的意义。根据实验发现，被编码序列的最优长度约为  $N = 80$ ，这与浓密机系统存在 2 至 3 小时时延的先验经验几乎一致。

表 3-4 不同初始隐状态  $h(t_0)$  生成方法对于预测精度的影响

$N$	$L = 60$ (120 分钟)		$L = 200$ (400 分钟)		$L = 500$ (1000 分钟)	
	RRSE	MSE	RRSE	MSE	RRSE	MSE
160	3.11	9.08	<b>3.56</b>	34.13	1.61	35.88
80	<b>3.10</b>	<b>8.97</b>	3.58	<b>32.92</b>	<b>1.61</b>	<b>34.88</b>
40	3.19	8.99	3.65	36.07	1.71	41.26
1	4.06	10.71	4.97	51.09	1.77	63.56
$N = 0$ , $h(t_0)$ 作为学习参数	5.26	20.68	4.84	58.68	1.77	63.91
$N = 0$ , 令 $h(t_0) = \mathbf{0}$	5.26	23.11	5.84	64.49	1.77	63.53

直观上看，短期预测任务比长期预测任务更能从历史系统轨迹中获益。随着被预测序列长度的增加，引入序列编码器带来的优势也随之降低。在  $L = 500$  的任务中，使用顺序编码器得到的预测精度约等于将  $h(t_0)$  作为学习参数的初始状态编码方案。

### 3.5 本章小结

本章针对高时延工业多输入输出系统预测问题，提出了一种基于 ODE-net 的连续时间网络模型，该模型由顺序编码器、状态解码器和导数模块三部分组成，模型的内部计算过程包括历史序列编码、离散输入序列插值以及常微分方程求解三部分，模型能够从连续时间域角度拟合复杂系统的动态过程。在膏体浓密机系统运行数据集上，我们进行了大量实验对所提出模型的各个部分进行研究，包括平稳系统和非平稳系统的选择以及不同 ODE 求解器的设定。结果表明，非平稳系统在短期预测任务中的表现优于平稳系统。但是，非平稳模型由于存在累加计算，会造成隐状态的波动范围过大，进而导致长期预测性能较差。该结果表明非平稳系统模型更适用于短期预测场景，如基于模型的反馈控制器（如 MPC 控制器）中作为预测模型辅助控制器输出。而平稳系统通过修改了导数模块结构，避免了隐状态漂移问题，因此在长期预测中表现较好。因此，当需要具有较强稳定性、鲁棒性的系统识别模型进行系统长期预测时（如系统仿真或控制器测试），具有平稳系统的模型是更好的选择。同时，通过消融实验表明模型中引入序列编码器和并行三次样条插值对于模型精度的提升起到了至关重要的作用。在工业数据分析领域，面向不均匀采样数据的分析与预测是极其常见的。虽然本文所用数据集的采样间隔是均匀的，由于本文所述模型是基于 ODE-net 进行构建的，具有连续时间域的系统学习及表示能力，因此该方法可以很自然地处理不均匀数据。该问题值得在今后的工作中被进一步的验证。另外将本文方法扩展为概率模型和时变模型，以辅助确定复杂系统中的未知噪音和不确定性，也是未来极具价值的研究方向。



## 4 基于自跳跃常微分方程网络的连续时间跳变系统建模

周期性跳变系统在现实生活中随处可见，其运行过程具有周期循环性。每个周期内，各阶段之间按特定顺序进行切换。系统在不同阶段会呈现明显不同的动态特性。例如，洗衣机启动后将按程序设定，周期性地在进水，洗涤，排水，甩干各个阶段之间循环运行，直至最后关机。冰箱和空调在工作期间会在运行（压缩机开启）和待机（压缩机关闭）两种状态之间不断进行切换。在一些自然界中其他的物理过程，也存在类似的系统，例如建筑物内的空调制冷系统、交通流量和季节变化等。例如，在对城市交通流量或某产品销量进行预测时，流量或销量通常伴随着特定的日期和季节变化而改变，并且其周期长度相对稳定，模型能够较容易地识别各预测时刻被预测对象所处的阶段。相反，对于一些复杂工业系统，其周期长度可能受到系统内部和外部多个变量的影响。例如系统的外部输入、当前内部状态、环境条件等。利用历史运行数据识别以及建模此类跳变系统对于理解和优化系统运行过程是极其重要的。然而，在面向跳变系统进行建模时面临两大技术难题。首先，跳变系统通常有多个阶段，每个阶段呈现不同的非线性特性。各个阶段的持续时间可能同时受到内部因素和外部因素的影响<sup>[114]</sup>。目前，面向跳变系统参数估计方法<sup>[30]</sup>通常依赖于对系统结构以及持续时间分布的先验认知，这需要大量的领域专家知识作为支撑。另外，对于带有多输出项的工业系统，其输出过程可能由多个不同的物理特性共同影响。这会导致多输出项中同时存在稳定和非稳定过程<sup>[34]</sup>。据我们所知，对于现有的未经过特定设计的现存模型，难以解决此类带有混合时序特性的系统学习任务。

为了实现上述目标，模型需要解决两个问题。首先，模型需要根据系统的外部输入以及系统状态，识别在当前时刻系统所处的阶段。另外，模型需要能够准确地识别系统在各阶段内的独立动态特性。针对上述挑战和问题，本章提出了一种新颖的自跳跃常微分方程网络（Autonomous jump ordinary differential equation net, AJ-ODEnet）以学习连续时间周期跳变系统。该模型是一种新颖的连续时间深度学习模型，能够从非均匀采样的系统历史轨迹进行学习。训练完成后，对于给定的系统外部输入模型能够实现实时在线预测。为了学习系统在不同阶段的动态特性，模型包含多个层次常微分方程网络（Hierarchical ODENets，H-ODENets）层次常微分网络是一种双层扩展的常微分方程网络，能够同时学习系统的稳定过程和非稳定过程。为了学习阶段转换，本章对原始训练数据进行了阶段类别标注，并在模型中引入阶段转换预测器学习每个阶段的持续时间。阶段转换预测器由多个持续时间预测器构成，在开环预测和仿真中，阶段转换预测器能够作为 H-ODENets 的调节器，在每个时间点指派合适的 H-ODENet 进行系统输出的预测，并预估当前所处阶段的持续时间，进而实现自发的阶段转移。

同时为了给定求解 AJ-ODEnet 的状态初值，本章遵循编码器解码器（Encoder-Decoder）范式，构建了两个 AJ-ODEnet 分别用于历史数据的编码以及在给定的系

统输入序列下以开环的方式预测系统的未来输出。除此之外，本章引入样条插值算法，以连续化离散的输入序列并作为 AJ-ODEnet 模型的输入。最后，本章将所属模型及框架应用于某一具有典型周期多阶段特性的工业制冷系统的预测与仿真问题中。通过给定系统的多变量运行数据，包括热负载以及环境温度，模型能够仿真系统的运行过程并准确地预测系统能耗（能耗预测误差小于 5%）。进一步地，通过模型的仿真功能能够优化系统的制冷温度设定参数，通过仿真实验表明利用优化之后的温度设定值能够显著改善能源利用效率并且节省超过 25% 的制冷能耗。

本章研究主要包含三方面贡献，概括如下：

- 为了辨识连续时间跳变系统，本章提出了一种新颖的深度学习模型——自跳跃常微分方程网络模型。该模型能够实现（1）在一个集成化的网络中学习系统的多阶段动态，（2）在长序列开环预测中实现阶段自转移。
- 为了学习同时包含稳定和非稳定项的多输出系统，本章结合每个输出项的统计特性，引入 H-ODENet 以改善两种输出项在长期预测中精度。
- 本章所述模型成功应用于某一工业制冷系统的预测与仿真问题中，该模型能够在给定热负载功率下，准确地预测制冷系统的功率消耗以及制冷气温度变化，依托于该仿真模型能够优化制冷系统的工作温度设定值，进而优化制冷系统的整体能耗约 25%。

本章的结构组织如下：本章4.1节给出了变量符号定义以及描述了周期性多阶段系统预测问题的形式化描述。本章4.2.1节介绍了分层常微分方程网络的定义及结构，以及如何将其用于学习同时带有稳定和非稳定时间序列输出的动态过程。本章4.2.2介绍了 AJ-ODEnet 框架，并详细阐述了连续时间域下的状态转移方程、关键模块结构和持续时间预测器的损失函数定义等内容。本章4.2.3给出了基于 AJ-ODEnet 模型的编码器解码器框架及其训练方法。本章4.4节介绍了将所述框架应用于模拟某实际工业制冷系统。模型能够根据制冷系统的热负载输入对其功耗以及出口气体温度变化进行模拟预测，同时能够实现不同制冷启动温度配置下的功耗仿真与优化。最后，6.5节对本章工作做出了总结，并对 AJ-ODENet 模型的潜在应用以及未来的研究方向做出了讨论。

## 4.1 问题形式化描述

在本文中，标量和向量在形式上分别用小写字母和粗体小写字母表示，例如  $y, \mathbf{y}$ 。假设输入输出数据由传感器观测得到，并带有时间戳  $t_i \in \mathbb{R}$ 。因此，从系统中采集的带有非均匀采样间隔的离散输入序列和输出序列可表示为： $\mathbf{X}_{1:L}, \mathbf{Y}_{1:L} = ((t_1, \mathbf{x}_1, \mathbf{y}_1), (t_2, \mathbf{x}_2, \mathbf{y}_2), \dots, (t_L, \mathbf{x}_L, \mathbf{y}_L))$ 。相应的，定义  $\mathbf{x}(t_i) = \mathbf{x}_i$  以及  $\mathbf{y}(t_i) = \mathbf{y}_i$ 。其连续时间过程可表示为  $\mathbf{x} : [t_1, t_L] \rightarrow \mathbb{R}^M$  和  $\mathbf{y} : [t_1, t_L] \rightarrow \mathbb{R}^N$ 。为了便于后文描述，本章将  $\mathbf{y} : [t_1, t_L]$  简化为  $\mathbf{Y}_{t_1:t_L}$ 。

连续时间跳变系统 (Continuous-time Jump System, CTJS) 由多个连续时间子系统组成, 其内部阶段转换由一个有限状态过程支配<sup>[67]</sup>。作为一种特殊的动态系统, CTJS 系统广泛应用于各类应用中, 如故障预测<sup>[115]</sup>、系统控制<sup>[116]</sup>, 以及状态估计<sup>[67]</sup>。典型的连续时间跳变系统定义形式如下:

$$\dot{\mathbf{h}}(t) = \mathcal{F}_\sigma(t)(\mathbf{h}(t), \mathbf{x}(t)), \quad (4-1)$$

其中  $\sigma(t) \in \{0, 1, \dots, N-1\}$  为一有限状态集下的跳变状态,  $\mathbf{x}(t)$  为满足利普希茨连续的系统输入。在一般的连续时间马尔科夫跳变系统中, 系统状态  $\sigma(t)$  的变化遵循马尔科夫过程。相比之下, 本章假设阶段变量  $\sigma(t)$  的转移是确定性地且循环的, 并将此系统定义为连续时间周期跳变非线性系统中 (Continuous-time periodic jump non-linear system), 分段恒定的状态  $\sigma(t)$  在循环  $\{0, 1, \dots, N-1, 0, \dots\}$  中不断变化。状态跳跃之间的停留时间取决于由系统先验属性决定的触发条件。为了明确区分跳变系统中的状态变量  $\sigma(t)$  以及循环神经网络中的隐状态, 本章将跳变系统状态  $\sigma(t)$  命名为阶段。

本章所关注的问题为根据系统的历历史数据信息和以及系统的未来序列输入, 预测系统的未来输出。具体地本章对于给定数据的时间范围  $[t_1 : t_{I+L}]$  分为两部分, 分别作为条件范围  $[t_1 : t_I]$  和预测范围  $[t_I : t_{I+L}]$ 。模型通过给定条件范围  $[t_1, t_I]$  下的系统输入和系统输出生成初始状态, 然后利用学得的跳变模型式 (4-1) 对预测范围  $[t_I, t_{I+L}]$  下的系统输出进行预测, 如式 4-2 所示:

$$\hat{\mathbf{Y}}_{t_I:t_{I+L}} = f(\mathbf{Y}_{t_1:t_I}, \mathbf{X}_{t_1:t_{I+L}}; \zeta) \quad (4-2)$$

公式中  $\hat{\mathbf{Y}}_{t_I:t_{I+L}}$  表示在预测范围内估计的输出。 $\mathbf{Y}_{t_1:t_I}$  表示在条件范围内系统的历历史输出。 $\mathbf{X}_{t_1:t_{I+L}}$  表示完整时间范围内的系统输入, 包括历史部分和预测序列部分。该系统输入在时间序列预测领域经常被称为协变量<sup>[111]</sup>。 $\zeta$  表示利用数据集训练的模型可学习参数。本章的目标为通过优化  $\zeta$  以最小化  $\|\hat{\mathbf{Y}}_{t_I:t_{I+L}} - \mathbf{Y}_{t_I:t_{I+L}}\|^2$ 。值得注意的是, 本章仍关注开环预测问题, 模型预测过程中无法收到来自系统输出的反馈。

## 4.2 自跳跃常微分方程网络及编码器-解码器预测架构

### 4.2.1 基于层次 ODE-Net 模型的复杂系统稳定及非稳定输出混合建模

大部分工业系统的输出是多维度的, 不同维度输出的时序统计特性存在一定差异, 某些输出量为均值方差恒定的稳定过程, 某些输出为均值方差不断变化的非稳定过程。对于同时含有稳定和非稳定输出的混合过程进行建模学习具有一定难度, 当前主流的基于深度学习的序列预测领域并未针对该类系统开展研究。针对该问题, 本章提出了分层常微分方程网络 (Hierarchical ODENet, H-ODEnet), 用

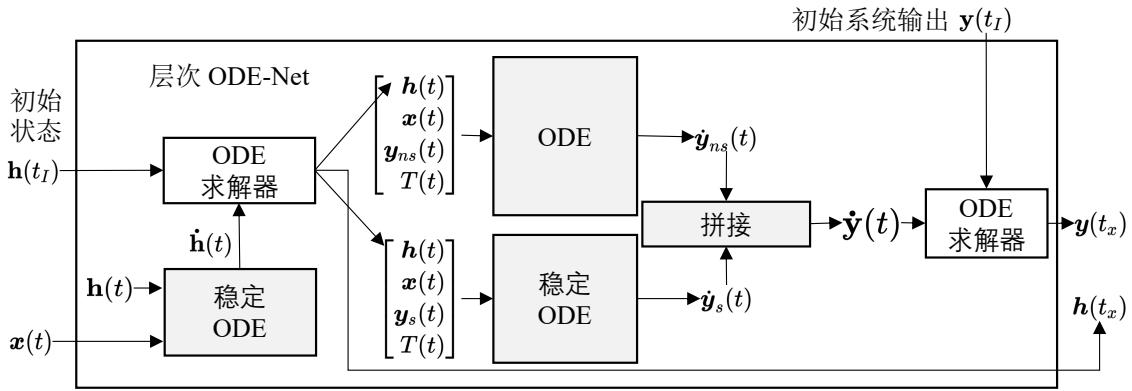


图 4-1 H-ODEnet 结构图示

于建模同时包含稳定过程和非稳定过程的系统。

具体地, H-ODEnet 将三个 ODEnet 集成为双层结构 (L-1 和 L-2)。图 4-1 展示了 H-ODE 中三个 ODE-net 之间的连接方式。在 L-1 层中, 包含一个 ODEnet, 用于根据外部输入对隐状态的导数进行建模。利用实时求解出的  $h(t)$ 、外部输入  $x(t)$  和当前阶段持续时间  $T(t)$ , 对由 L-1 中的 ODE-net 定义的常微分方程进行积分。解出的  $h(t)$  作为 L-2 中两个 ODE-net 的输入, 并分别用于求解稳定系统输出项和非稳定系统输出项的导数, 表示为  $y(t) = [y_s(t), y_{ns}(t)]$ 。最终, 可以解出任一时间点  $t_x$  下的预测结果。

其中, L-1 中的 ODEnet 为稳定结构, 由门控循环单元 (GRU) 网络实现<sup>[35]</sup>:

$$\frac{dh(t)}{\tau} = \frac{1}{\mu(t)} * \left[ \text{GRU}(x(t), h(t), \text{sigmoid}(T(t)), \theta_{\sigma(t)}^H) - h(t) \right] \quad (4-3)$$

其中  $\mu(t)$  表示原始数据集的平均采样间隔均值。稳定的 GRU 网络将预测的隐状态变量  $h(t)$  的变化表示为稳定过程, 并将其上下界限制在  $(-1, 1)$  范围内。

L-2 中的两个 ODEnet 被分别设计用于建模稳定和非稳定的系统输出。其中用于建模非稳定输出的网络采用非稳定增量形式预测输出  $y_{ns}(t)$ :

$$\frac{dy_{ns}(t)}{\tau} = \text{NN}(x(t), h(t), \text{sigmoid}(T(t)), \theta_{\sigma(t)}^{y_{ns}}) \quad (4-4)$$

另一个 ODEnet 类似于 L-1 中的 ODEnet, 由一个稳定的 GRU 模型构建, 用于预测系统的稳定输出  $y_s$ :

$$\frac{dy_s(t)}{\tau} = \frac{1}{\mu(t)} * \left[ \text{GRU}(x(t), h(t), \text{sigmoid}(T(t)), \theta_{\sigma(t)}^{y_s}) - y_{\sigma}(t) \right] \quad (4-5)$$

对于 L-1 和 L-2 中的 ODE-net, 均利用系统处于当前阶段的持续时间  $T(t)$  作为特征输入, 辅助估计  $y(t)$  的导数。 $\{\theta_i^{y_{ns}}, \theta_i^{y_s}, \theta_i^H\}_{i=1}^N$  是模型中的可学习参数。

一般情况下, 从传感器测量得到的数据通常是离散的、带有一定比例的不规则分布的缺失值。例如可能存在一个时刻  $i$ , 两区间  $t_i - t_{i-1} \neq t_{i+1} - t_i$ , or  $x(t_i)$ ,

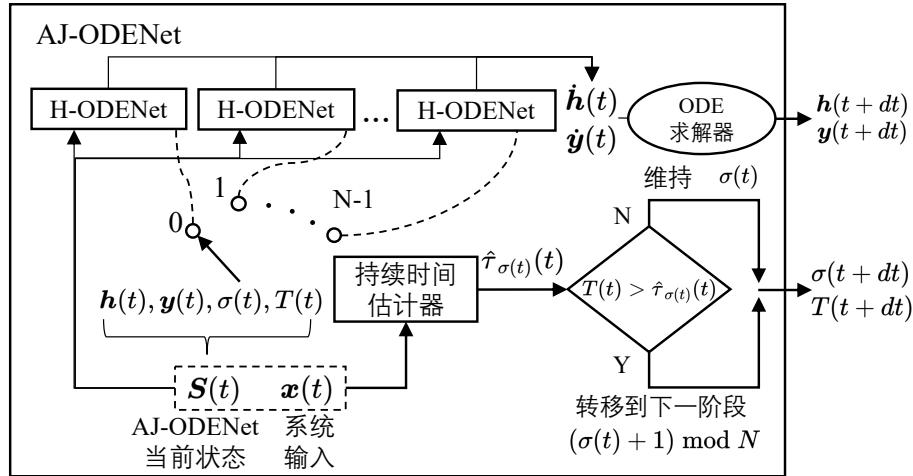


图 4-2 AJ-ODEnet 模型结构

或者存在  $x(t_i)$  存在，而  $y(t_i)$  缺失，并在数据集中表示为 *Null*。本章遵循<sup>[20,47]</sup>中的形式，将离散的控制输入序列连续化并作为 ODEnet 的输入。因此需要实现面向离散序列输入的连续化插值方法，将离散和采样不规则的数据格式转换为连续时间格式。为了实现该目标，有多种方法可以实现，如高斯过程<sup>[117]</sup>，核方法<sup>[118]</sup>或样条插值<sup>[20]</sup>。本章参考<sup>[20]</sup>采用三次样条插值方法。

#### 4.2.2 自跳跃常微分方程网络

部分工业系统存在多阶段、非确定和多物理过程混合等复杂特性。对这类系统进行高精度、高鲁棒的建模是一项具有挑战性的任务。在本章研究中，为了能够使模型准确地预测多阶段系统的阶段转换时刻以及预测系统输出，本章提出了 AJ-ODEnet 模型。具体地，我们首先根据系统先验知识构建系统的多阶段转换过程，对于每个阶段引入一个 H-ODEnet 以学习系统在该阶段内的动态特性，并对训练集中的序列数据进行阶段变量标注，引入阶段转换预测器学习每个阶段的持续时间。在测试阶段，阶段预测器可以作为不同 H-ODEnet 的调节器，实现不同阶段之间的自动转移。

图 4-2 简要介绍了 AJ-ODEnet 的主要组件及其结构。在图 4-2 中，AJ-ODEnet 模型嵌入了  $N$  个 H-ODEnet 模块，根据状态变量  $\sigma(t)$ ，在同一时刻只采用其中一个 H-ODEnet 模块用于系统预测。。相比于普通的 ODEnet 仅建模隐状态的导数。AJ-ODEnet 将隐状态扩展为四个参数，从而覆盖连续时间跳变系统中输出变量变化以及阶段变量的变化。其中，隐状态的计算和更新是连续时间域下的，表示为  $S(t) = [h(t), y(t), \sigma(t), T(t)]$ ，其中四项分别为**系统隐状态**、**预测的系统输出**、**系统所处的当前阶段**和**处于当前阶段已经持续的时间**。AJ-ODEnet 可以看作是 ODEnet 的扩展形式，计算更新过程可以视为给定时间点  $t$  和无穷小时间步长  $\tau$  计算微分方程的解，如式 (4-6) 所示：

$$S(t+dt) = \begin{bmatrix} \left. \begin{array}{l} \mathbf{h}(t) + \frac{d\mathbf{h}(t)}{\tau} * dt \\ \mathbf{y}(t) + \frac{d\mathbf{y}(t)}{\tau} * dt \end{array} \right\} & \text{见公式(4-3), (4-4), 和(4-5)} \\ s(t+dt) = \left\{ \begin{array}{ll} \sigma(t) + 1 \bmod N, & T(t) \geq \hat{\tau}_{\sigma(t)}(t) \\ \sigma(t), & \text{else} \end{array} \right. \\ T(t+dt) = \left\{ \begin{array}{ll} 0, & T(t) \geq \hat{\tau}_{\sigma(t)}(t) \\ T(t) + dt, & \text{else} \end{array} \right. \end{bmatrix} \quad (4-6)$$

对于复杂的周期性多阶段系统而言，很难用统一的模型拟合系统的所有阶段，在长期预测问题中该问题更加明显。对于周期稳定、各个阶段持续时间不变的系统，可以将系统的运行轨迹平均分解为若干区间，分别学习各阶段内的系统动态特性。但是，更普遍的情况是，系统中各阶段的持续时间长度会受到系统内部和外部多个变量的影响。为了使模型适用于系统中各阶段持续时间不稳定的情况，本章引入了“持续时间估计器”，通过学习的方式，在预测时预估每个阶段的持续时间，并辅助模型判断是否应切换到下一阶段。

具体地，我们结合周期性多阶段系统中的先验知识，可以很容易地设计一个状态转换过程描述系统在各个阶段之间的转换。从给定的数据集中学习相邻阶段之间的转换规则，使系统在运行时切换到不同的 H-ODEnet 进行预测，将系统输出预测和阶段识别进行解耦。

对于在  $t$  时刻的连续时间状态  $S(t) = [\mathbf{h}^T(t), \mathbf{y}^T(t), T(t), \sigma(t)]^T$ ,  $\sigma(t) \in \{0, \dots, N-1\}$  表示系统当前所处阶段， $T(t)$  表示系统处于阶段  $\sigma(t)$  的持续时间。

对于式 (4.2.2) 所述的状态演化过程，本章基于多层感知器实现了与当前阶段变量  $\sigma(t)$  绑定的持续时间估计器  $\hat{\tau}_{\sigma(t)}$  来预测当前阶段的持续时间。预测器的输入为当前隐状态  $\mathbf{h}(t)$  和外部输入  $\mathbf{x}(t)$ :

$$\hat{\tau}_{\sigma(t)}(t) = \text{NN}([\mathbf{h}(t), \mathbf{x}(t)], \phi_{\sigma(t)}) \quad (4-7)$$

其中  $\phi_{\sigma(t)}$  为可学习参数，当满足  $T(t) \geq \hat{\tau}_{\sigma(t)}(\mathbf{h}(t), \mathbf{x}(t))$  时，模型将切换到下一阶段  $(\sigma(t) + 1) \bmod N$  并将  $T(t)$  复位为 0。

### 4.2.3 基于编码器解码器结构的微分方程网络初值估计与序列预测

在周期跳变系统的开环预测问题中，确定系统当前所处的相位，即识别当前所处阶段以及当前阶段已经持续的时间，是极其重要的。与此同时，受限于测量技术及成本的限制，复杂工业系统的观测空间往往是不完备的。因此，为了实现

精确的预测，模型需要对系统的非确定性进行推理。本节将介绍如何根据给定条件范围下的序列数据推断系统的非确定性信息，并作为求解 AJ-ODEnet 所需的初始状态。

本章遵循<sup>[119,120]</sup> 中引入的编码器-解码器框架，使用两个 AJ-ODEnets 分别构建用于编码历史系统轨迹的编码器和预测系统输出的解码器，如图4-3。模型输入的时间序列数据包括条件范围  $\{\mathbf{Y}_{t_1:t_I}, \mathbf{X}_{t_1:t_I}\}$  和预测范围  $\{\mathbf{X}_{t_{I+1}:t_L}\}$ 。

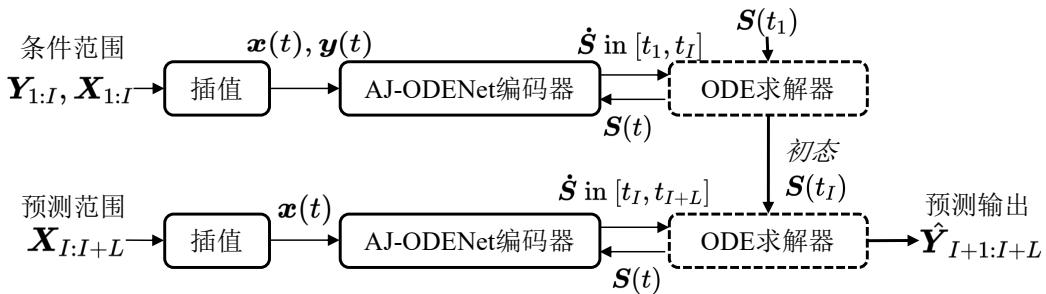


图 4-3 基于 AJ-ODEnet 的编码器-解码器预测框架

由于 ODEnet 中求解微分方程需要连续信号，需要对原始的离散序列数据处理转换为连续时间序列，利用编码器对条件范围内的数据进行编码，构建求解 AJ-ODEnet 解码器所需的初始状态  $S(t_1)$ 。

在条件范围内，根据初始状态  $S(t_1)$ ，求解 AJ-ODEnet 编码器对应的常微分方程，可以得到  $t_I$  时刻的状态  $S(t_I) = [\mathbf{h}(t_I), \mathbf{y}(t_I), s(t_I), T(t_I)]^T$ 。然后在预测阶段，基于 AJ-ODEnet 的解码器将会根据初始状态  $S_I$  和系统外部输入  $\{\mathbf{X}_{t_I:t_{I+L}}\}$  估计预测范围内隐状态的导数，进而预测系统输出。

具体来说，在条件范围  $[t_1, t_I]$  的编码阶段，我们将连续时间信号  $\mathbf{y}(t)$  和  $\mathbf{x}(t)$  合并起来作为 AJ-ODEnets 编码器的输入，生解码器所需的初始状态，如式 (4-8) 所示。

$$\tilde{S}(t_I) = \text{ODESolve}(\text{Encoder AJ-ODE-nets}, S(t_1), \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{y}(t) \end{bmatrix}, t_1, t_I) \quad (4-8)$$

其中  $t_1$  时刻的初始状态定义为:  $S(t_1) = [\mathbf{h}(t_1) = \mathbf{0}, \mathbf{y}(t_1), T(t_1) = 0, s(t_1)]^T$ 。根据求解得到的  $\tilde{S}(t_I) = [\tilde{\mathbf{h}}(t_I), \tilde{\mathbf{y}}(t_I), \tilde{s}(t_I), \tilde{T}(t_I)]^T$ 。可以获得状态  $S(t_I) = [\mathbf{h}(t_I), \mathbf{y}(t_I), \tilde{s}(t_I), \tilde{T}(t_I)]^T$   $\mathbf{h}(t_I)$  是从均值为  $\tilde{\mathbf{h}}(t_I)$ ，协方差矩阵为单位矩阵  $\mathbf{I}$  的对角多元高斯分布中采样得到的。由于在网络传播过程中存在采样操作。为了构建用于梯度传导的计算图以及减小训练时梯度估计的方差，上述采样过程使用了重参数化法<sup>[74]</sup>。

本节设计 AJ-ODEnets 编码器的最重要的目的是估计隐变量  $\mathbf{h}(t_I)$ 。上述从概率分布中采样的方式相当于将  $\mathbf{h}(t)$  视为条件序列生成模型中的隐变量，从过去的系统输出和输入中推测隐变量的近似状态后验<sup>[55,76]</sup>。

接下来，通过给定解码阶段的初始状态  $S(t_I)$  和连续时间系统输入  $\mathbf{x}(t)$ ，在  $[t_I, t_{I+L}]$  范围内求解 AJ-ODEnet 解码器对应的常微分方程，可以得到预测范围内的

隐变量状态  $S(t)$ , 其中包括系统在任意时刻  $t_i$  的系统输出  $\hat{y}(t_i)$ :

$$\hat{S}(t_{I+L}) = \text{ODESolve(AJ-ODEnet Decoder, } S(t_I), x(t), t_I, t_{I+L}) \quad (4-9)$$

AJ-ODEnets 编码器和 AJ-ODEnets 解码器之间有两个区别。首先, 系统输入  $x(t)$  和输出  $y(t)$  都均是编码阶段的输入, 而解码器的输入仅包含系统输入  $x(t)$ 。因此解码器和编码器中 H-ODENet 的输入层的大小是不一致的。其次, 对于条件范围下的编码过程来说, 由于系统的输入输出数据已知, 模型进行阶段转移时, 可直接根据系统先验属性得到阶段之间的转换规则, 不需要阶段转换预测器进行阶段的自转移。对于解码器, 由于系统输出  $y(t)$  未知, 因此需要利用阶段转换预测器进行状态自切换。这两个区别表明, 在求解初始状态的过程中, 编码器不引入阶段转换预测器, 而是根据实际的系统输出准确地识别系统当前所处的阶段, 并推导出  $S(t_I)$ ,  $T(t_I)$  和  $y(t_I)$ 。因此, 初始状态  $S(t)$  中的  $\sigma(t)$  和  $T(t)$  是准确可靠的。对于解码器来说, 相邻阶段之间的转移是依靠持续时间预测器进行判断, 无法保证绝对准确的。在长范围开环预测时会带来较大的累积相位误差。

### 4.3 模型训练

利用 AJ-ODENet 编码器根据条件范围下的序列输入数据获得初始状态后, 在预测范围内求解 AJ-ODEnet 解码器即可得到预测结果  $\hat{Y}_{t_{I+1}:t_{I+O}}$ 。利用数据集中真实的系统输出序列即可通过有监督学习的方式对网络进行端到端的训练。接下来本小节将对损失函数的定义进行介绍。

在所述的编码器解码器框架中, 待训练的参数包括三部分:  $\zeta = \{\Theta_d, \Theta_e, \Phi\}$

- $\Theta_e = \{\theta_i^{\mathcal{Y}^{ns}}, \theta_i^{\mathcal{Y}^{ns}}, \theta_i^{\mathcal{H}}\}_{i=0}^{N-1}$ : 在 AJ-ODEnet 编码器中的  $N$  个 H-ODEnet
- $\Theta_d = \{\theta_i^{\mathcal{Y}^{ns}}, \theta_i^{\mathcal{Y}^{ns}}, \theta_i^{\mathcal{H}}\}_{i=0}^{N-1}$ : 在 AJ-ODEnet 解码器中的  $N$  个 H-ODEnet
- $\Phi = \{\phi_i\}_{i=0}^{N-1}$ : AJ-ODEnet 解码器中的  $N$  个持续时间预测器。

为了能够以端到端的方式对上述参数进行训练。需要优化的模型损失函数包括两部分, 分别为模型预测系统输出的误差损失  $\mathcal{L}_P$  和持续时间预测器预测的各阶段持续时间的误差损失  $\mathcal{L}_{\ll}$ :

$$\mathcal{L}(\Theta_e, \Theta_d, \Phi) = \mathcal{L}_P(\Theta_e, \Theta_d) + \lambda \mathcal{L}_{\ll}(\Phi) \quad (4-10)$$

其中  $\lambda$  是平衡  $\mathcal{L}_P$  和  $\mathcal{L}_{\ll}$  的权重参数。

对于  $\mathcal{L}_P$ , 由于模型将  $h(t_I)$  作为隐变量进行后验推断, 因此可以采用变分贝叶斯优化的方法, 将最大化系统观测输出的证据下界 (Evidence lower bound, ELBO)

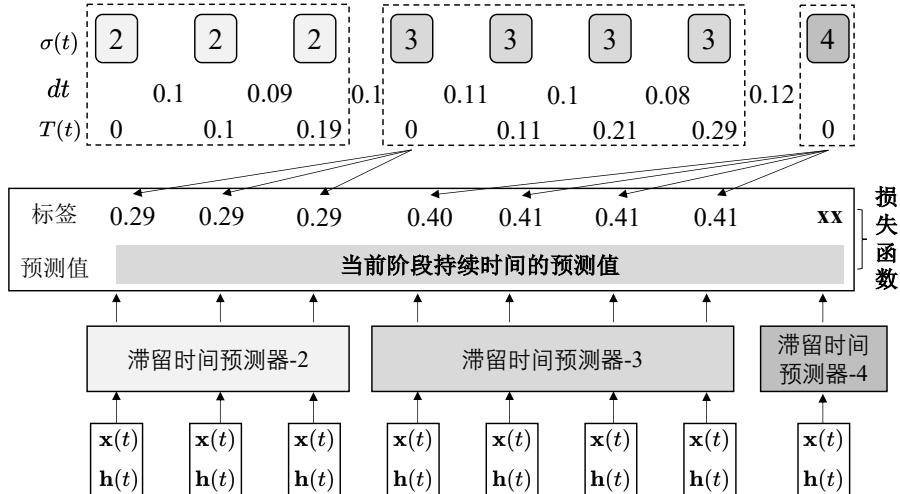


图 4-4 时间预测器的损失函数定义

作为模型训练的目标，以近似地最大化系统输出的边际似然<sup>[55]</sup>:

$$\text{ELBO}(\Theta_e, \Theta_d) = -\text{KL}[q_{\Theta_e}(\mathbf{h}(t_I) | \mathbf{Y}_{t_1:t_I}, \mathbf{X}_{t_1:t_I}) \| p(\mathbf{h}(t_I))] + \mathbb{E}_{\mathbf{h}(t_I) \sim q_{\Theta_e}(\mathbf{h}(t_I) | \mathbf{Y}_{t_1:t_I}, \mathbf{X}_{t_1:t_I})} \log p_{\Theta_d}(\{\mathbf{y}_i\}_{i=I+1}^{I+L} | \mathbf{h}_{t_I}, \mathbf{X}_{t_I:t_{I+L}}) \quad (4-11)$$

在本研究中，我们参考前人的工作<sup>[18,55,60]</sup>，对隐变量的先验分布和后验分布做出了如下假设。我们假设隐变量  $p(\mathbf{h}(t_I))$  的先验分布服从正态分布  $\text{Normal}(\mathbf{h}(t_I); \mathbf{0}, \mathbf{I})$ 。模型估计的后验分布  $q_{\Theta_e}(\mathbf{h}(t_I) | \mathbf{Y}_{t_1:t_I}, \mathbf{X}_{t_1:t_I})$  为对角多元高斯分布。由此，KL 散度项可以简化为对于  $\mathbf{h}(t_I)$  的正则项。对于解码器部分，我们定义生成模型  $p_{\Theta_d}(\mathbf{Y}_{I+1:I+L} | \mathbf{h}_{t_I}, \mathbf{X}_{t_I:t_{I+L}})$  为具有固定协方差矩阵的正态分布。分布的均值定义为 AJ-ODEnet 解码器预测出的系统输出。最大化重构似然的期望  $\mathbb{E}_{q_{\Theta_e}(\mathbf{h}_{t_I} | \dots)} [\log p_{\Theta_d}(\mathbf{Y}_{I+1:I+L} | \mathbf{h}_{t_I}, \mathbf{X}_{t_I:t_{I+L}})]$  可以简化为最小化系统输出预测值  $\hat{\mathbf{Y}}_{I+1:I+L}$  与实际系统输出  $\mathbf{Y}_{I+1:I+L}$  之间的  $L^2$  距离。综上所述，预测系统输出的损失和编码获得隐变量的损失可表示为：

$$\mathcal{L}_{\mathcal{P}}(\Theta_e, \Theta_d) = -\text{ELBO}(\Theta_e, \Theta_d) = \|\mathbf{h}(t_I)\|^2 + \sum_{i=I+1}^{I+L} \|\hat{\mathbf{y}}(t_i) - \mathbf{y}(t_i)\|^2 \quad (4-12)$$

阶段持续时间预测器的误差损失  $\mathcal{L}_{\tau}$  定义为训练时每个阶段下，预测器估计的持续时间与该阶段实际持续时间之间的平均平方误差。构建过程如图4-4所示：

$$\mathcal{L}_{\ll}(\Phi) = \sum_{i=I+1}^{I+L} \|\tilde{\tau}_{s(t_i)}(t_i) - \tilde{\tau}(t_i)\|^2 \quad (4-13)$$

其中  $\tilde{\tau}(t_i)$  代表阶段  $\sigma(t_i)$  的持续时间，为了计算  $\tilde{\tau}(t_i)$ ，需要找到  $t_i$  所处阶段的区间边界  $t_l$  和  $t_r$ 。 $t_r$  为  $[t_i, \infty)$  范围下，满足条件  $\sigma(t_r) = \sigma(t_i)$  且  $\sigma(t_r + dt) \neq \sigma(t_i)$  的最小值。 $t_l$  为  $(-\infty, t_i]$  范围下，满足条件  $\sigma(t_l) = \sigma(t_i)$  且  $\sigma(t_l - dt) \neq \sigma(t_i)$  的最大值。实际的阶段持续时间即为  $\tilde{\tau}(t_i) = t_r - t_l$ 。

完整的网络模型通过 Adam 优化器进行训练。完整的训练算法如算法1所示。

---

### 算法1 基于 AJ-ODEnets 的编码器-解码器训练过程

---

```

1: for 每一个训练轮次 do
2:   for  $k$  steps do
3:     在训练集  $S$  中随机抽样一批序列  $\{\mathbf{Y}_{1:I+L}, \mathbf{X}_{1:I+L}\}$ .
4:     //为了加快训练速度, 下面的步骤是并行执行的.
5:     给训练数据的各阶段打标签:  $s_{t_1:t_{I+L}}$ 
6:     使用样条插值来处理离散序列  $\mathbf{X}_{1:I+L}, \mathbf{Y}_{1:I+L}$ , 生成  $[t_1, t_{I+L}]$  的连续序列
       $\mathbf{X}(t)$  和  $[t_1, t_I]$  的连续序列  $\mathbf{Y}(t)$ 。
7:     编码阶段: AJ-ODEnet 编码器根据条件范围下的系统输入和输出序列估
      计初始状态  $\mathbf{S}(t_I)$ , (4-8)。
8:     预测阶段: AJ-ODEnet 解码器根据预测范围下的系统输入  $[t_I, t_{I+L}]$  预测
      离散时间点  $\{t_{I+1}, \dots, t_{I+L}\}$  下的系统输出, (4-9)。
9:     利用随机梯度  $\nabla_{\Theta_e, \Theta_d, \Phi} \mathcal{L}(\Theta_e, \Theta_d, \Phi)$  更新两个 AJ-ODEnet 中的参数
       $\Theta_e, \Theta_d, \Phi$ :
10:    end for
11: end for

```

---

## 4.4 实验验证与分析

接下来本章将使用 AJ-ODENet 模型及衍生出的编码器解码器预测框架解决某工业制冷系统的建模及预测问题, 并采用两个实验探究模型的预测效果并基于仿真模型给出制冷系统的运行优化策略。具体地, 在第一个实验中, 我们使用 AJ-ODENet 模型根据当前的负载功率和环境温度, 在线预测制冷系统的输出量。在第二个实验中, 依托于实验一中训练得到的预测仿真模型, 实现对不同制冷启动温度配置下的制冷功耗进行仿真, 并结合仿真结果给出制冷策略优化。接下来本章将首先介绍作为实验对象的制冷系统, 然后介绍模型的训练参数、评估结果和预测仿真结果。

### 4.4.1 制冷系统简介

实验中探究的制冷系统为施耐德电气公司<sup>[?]</sup>设计。制冷系统为某一大型熔炼设备提供压缩制冷。制冷系统的运行数据可通过调用 Seduce 平台<sup>[121]</sup>的 API 获取数据。Seduce 平台是用于电源和温度管理的物联网平台, 该系统以 1hz 的频率采集传感器监测数据, 包括能耗和温度数据。图 4-5展示了制冷系统提供热交换制冷的过程示意图。该制冷系统包括室内和室外两部分。室内部分的主要模块为液体-空气热交换器, 该模块吸收生产设备产生的热空气, 然后通过管道、冷凝器和风机将热量转移和疏散到室外, 通过压缩生成的冷空气输送至室内。

制冷系统作为常见的工业设施, 是一种典型的周期性多阶段系统, 其运行过程为部分可观测的 2 输入 3 输出系统。直观地, 热负载 (熔炼设备的运行总功率

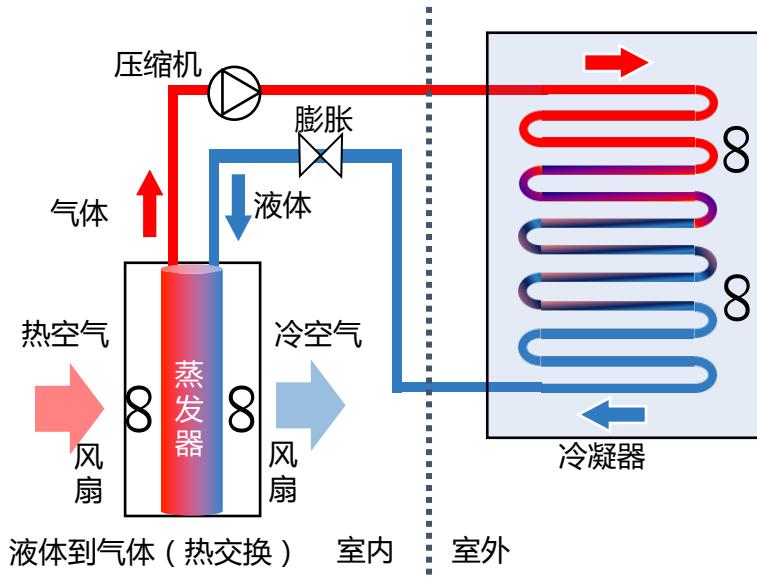


图 4-5 工业制冷系统整体结构及热交换过程

( $x_1(t)$ )) 的升高会导致制冷系统进气口温度 ( $y_3(t)$ ) 升高。另外，由于制冷系统和室外环境之间存在热交换，系统所处环境的温度 ( $x_2(t)$ ) 也会影响制冷系统的工作情况。为了保持生产过程安全稳定运行，制冷系统需要维持室内的进气空气温度限定某预先设定的区间上下限内。当进气口空气温度达到上限时，制冷系统会开启压缩机，制冷系统产生的制冷量 ( $y_2(t)$ ) 增加。当进气口温度温度低于设定值时，压缩机关闭，制冷系统变为待机状态。系统在压缩制冷时的运行功率 ( $y_1(t)$ ) 远大于关闭压缩机时的功耗。伴随着制冷与待机，制冷系统的运行过程具有典型的周期多阶段特性，其阶段转换如图 4-7所示。

当制冷系统待机时，进气口温度不断上升，其制冷量和制冷功率接近于 0。当制冷系统工作时，进气口温度不断下降，制冷量和制冷功率经过两个阶段的快速震荡后会趋近于某固定值。因此，对于该制冷系统，实时功率消耗  $y_1(t)$  和实时制冷量  $y_2(t)$  为系统的稳定输出  $\mathbf{y}_s(t) = [y_1(t), y_2(t)]$ 。进气口温度  $y_3(t)$  为系统的非稳定输出： $\mathbf{y}_{ns}(t) = [y_3(t)]$ 。将进气口温度定义为非稳定过程，这一性质与系统的先验特性是保持一致的。在第二个实验中，我们通过改变温度阈值而优化制冷系统能耗，需要模型对于进气口温度实现外推预测，在 H-ODENet 模型中将进气口温度定义为非稳定过程，对于实现可靠、合理的外推预测具有重要意义。

#### 4.4.2 数据预处理：离散序列插及阶段标注

为了使用微分方程网络对序列数据进行处理，需要使用插值方法将带有缺失值的不规则采样数据集转换为连续信号。在本章实验环节中，制冷系统数据集存在采样间隔不均匀、部分数据缺失的问题。本章参照前人工作<sup>[20]</sup>，使用三次样条插值方法对离散时间序列  $\mathbf{X}_{[1:I+L]}$  和  $\mathbf{Y}_{[1:I]}$  进行插值，构成连续时间信号  $\mathbf{Y} : [t_1, t_I]$  和  $\mathbf{X} : [t_1, t_{I+L}]$ 。在任意时刻  $t$ ,  $\mathbf{X}(t)$  和  $\mathbf{Y}(t)$  二阶可微，满足自适应步长求解器求

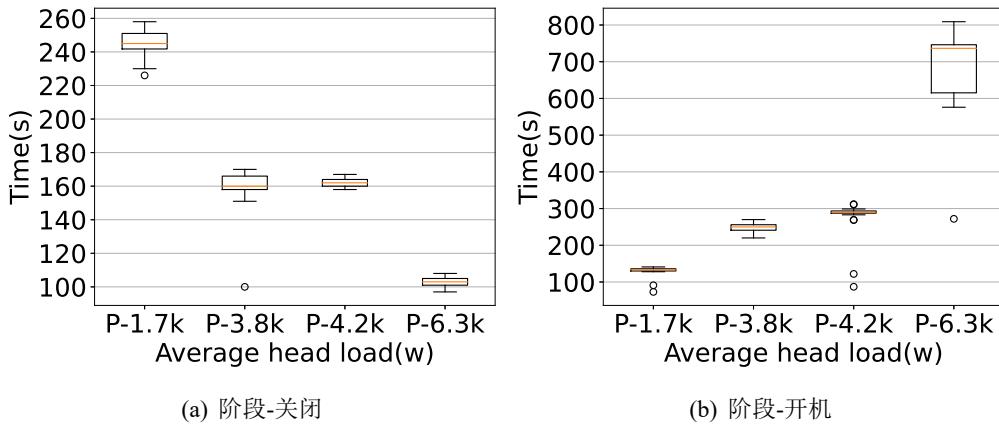


图 4-6 在平均负载不同的四个数据集中，阶段启动和关闭阶段的持续时间分布箱线图

表 4-1 跳变系统在四种状态下循环，每种状态对应制冷系统的一个阶段

状态	描述	下一阶段	转换触发条件
0-关闭	系统待机	1	$y_3(t) \geq Ti_{max}$
1-启动第一阶段	制冷系统启动的第一阶段	2	$y_1(t)$ 到达某一峰值
2-启动第二阶段	制冷系统启动的第二阶段	3	$y_2(t)$ 稳定
3-开机	系统运行	0	$y_3(t) \leq Ti_{min}$

解神经 ODE 系统时所需的必要条件<sup>[20]</sup>。当待处理的数据集包含的噪声比较少时，相比于其他插值算法，三次样条插值实现简单，且对原始序列数据的破坏较小，插值生成的连续信号中的噪音较小。

接下来，为了训练 AJ-ODEnet 模块中的阶段转换预测器，需要为序列数据标注各时间点所属的阶段。我们基于系统的先验知识构建了用于描述系统阶段转换的状态转换过程，并基于此自动地为序列数据标记所属阶段。

根据现场观察和对于制冷系统的了解，可知的系统先验知识如下：

- 根据当前进气口温度以及启停设定值，制冷系统的工作模式在压缩机启动和压缩机关闭两种状态之间来回切换。
- 系统包含有四个可观测的工作阶段，每个阶段下，系统输出呈现不同的动态特性。

根据上述先验知识，我们精确地定义了系统的各个阶段并给出了阶段之间的转换规则，如表 4-1 所示。制冷系统的运行阶段在  $N = 4$  个阶段之间依周期地循环切换。在图 4-6 中，我们统计了在不同的热负载下，每次 *On* 阶段和 *Off* 阶段的持续时间的分布情况。结果表明，阶段持续时间与服务器负载功率值有较强相关性。在特定的热负载下，阶段持续时间  $\tau_i$  的分布趋于稳定。这一结果说明利用系统的热负载输入以及环境温度预测各个阶段的持续时间是具有可行性的。

#### 4.4.3 模型训练

我们从 Seduce 平台采集了四组数据集用于模型的训练与评价，四组数据集均采集于生产运行稳定时段，且运行功率有较大差异，分别为 1.7kw, 3.8kw, 4.2kw, 6.3kw。运行功率  $x_1(t)$  越大，说明当前产热越多。表4-2中汇总了各个数据集描述和输入、输出介绍以及序列长度。对于四个数据集，每个数据集中的完整序列长度约为 8000-10000，采样时间点为连续非均匀的，平均采样频率约为 1 条/秒，序列对应的时间长度约为 8000s-10000s。每个数据集的前 50% 数据用于模型训练，剩下的数据中，25% 数据用于构建验证集，25% 数据用于模型测试。

对于训练集和验证集，本节使用大小为 1600s 的滑动窗口对原始序列进行遍历并生成训练样本。对于每一个训练样本，根据 4.1 定义，前 800s 为条件范围，用于模型求解预测阶段所需的初始状态，后 800s 为预测范围，模型预测该范围内的系统输出。在训练过程中，我们选择在验证集中表现最好的模型用于模型评估。由于输入的维数较低，且得益于伴随状态法（Adjoint state），训练 ODE-net 网络对于显存的消耗是极低的。我们选择了较大批量（batch size=4096）以加速训练。所有数据集的训练样本被随机排序并分批输入到模型训练。训练过程使用了单块型号为 NVIDIA TITAN RTX 的并行计算设备（GPU），其显存为 24G。隐状态变量  $h(t)$  的大小是 20，学习率设置为 0.005。

在测试阶段，没有像构造训练集一样对测试集中的完整序列进行分割。序列的前 800s 被送入编码器模型以生成初始状态，解码器模块在给定序列输入下，以开环的方式预测余下部分的系统输出。预测结果与测试集中的真实输出作对比以评估模型精度。

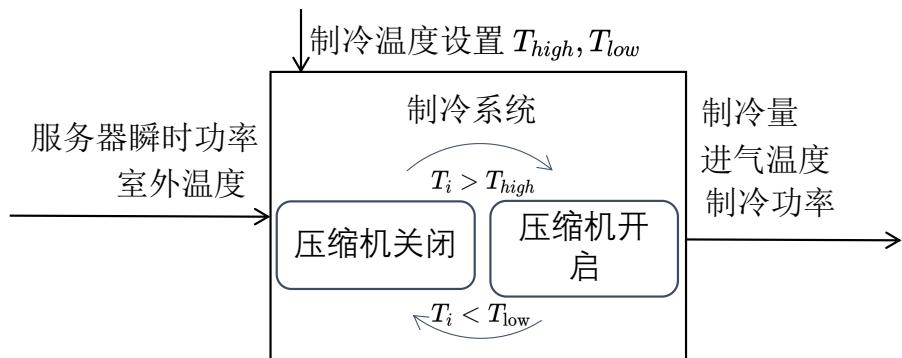


图 4-7 制冷系统运行机理简化图

#### 4.4.4 应用研究 1：运行变量仿真

本节使用基于 AJ-ODENet 结构的编码器-解码器框架仿真制冷系统的输出。模型的预测变量为三个：进气口温度，制冷量和制冷机功率。其中，进气口温度是受制冷系统影响的控制目标变量，表示制冷系统排出的空气温度。制冷量代表制冷系统单位时间产生的制冷量，其直接影响制冷系统中进气口温度的变化<sup>[122]</sup>，在压

表 4-2 系统输入输出、不同负载下的数据集和模型的操作

	变量定义	描述
输入	$x_1(t)$	服务器的瞬时功率, W
	$x_2(t)$	环境温度, °C
输出	$y_1(t)$	制冷系统功率, W
	$y_2(t)$	制冷量, W
数据集	$y_3(t)$	入气口温度, °C
	”1.7k”: 服务器稳定运行在 1.7kw 左右, 长度: 8853s	
	”3.8k”: 服务器稳定运行在 3.8kw 左右, 长度: 9771s	
	”4.2k”: 服务器稳定运行在 4.2kw 左右, 长度: 8472s	
”6.3k”: 服务器稳定运行在 6.3kw 左右, 长度: 8418s		

缩机不工作时, 制冷量为 0。制冷系统开始工作后, 制冷量会突然飙升至较高水平。制冷机功率, 表示整个制冷系统瞬时功耗。在制冷系统工作期间功耗较高, 待机期间功耗较低。本节同时引入普通的 ODE-Net、ODE-RNN<sup>[55]</sup>、CDE-Net<sup>[20]</sup> 作为本章 H-ODENet 的对比模型。对比结果如图4-8所示。其中图 4-8(a) 为真实的系统输出, 图 4-8(d) 为采用带有 H-ODENet 的 AJ-ODENet 模型的预测结果。在图 4-8(b) 中, 将 AJ-ODEs 模块替换为一个单一的稳定型 ODENet, 此时各阶段的转换不受持续时间预测器控制。此时模型难以对各个阶段间的系统切换边界处给出准确的拟合, 尤其是对于持续时间较短的阶段转换位置, 如阶段 1 以及阶段 2。在图 4-8(c) 中, 采用 ODE-RNN<sup>[55]</sup> 替换 AJ-ODENets 中的混合了稳定性和非稳定性输出的 H-ODENet 模块。可以发现该结构形成的预测结果难以拟合平滑的输出过程。相比于其他模型, 图 4-8(d) 中 AJ-ODENet 预测的系统输出十分精确, 且在阶段转换的边界处能够极好地识别系统输出的剧烈变化。上述实验表明, 使用多阶段模型能够将先验知识集成到模型中, 相比于单模型结构, 能够更好地处理多阶段系统的阶段转换边界预测。同时结合了非稳定输出和稳定输出的 H-ODENet 模型能够有效地对系统的多个输出项进行学习, 相比于忽视了系统输出时序特性的 ODE-RNN 模型, 能够更准确地预测系统的输出。

在预测精度量化方面, 由于本文关注于开环预测问题并且需要模型自发地进行阶段切换, 伴随着预测范围长度的增加, 对于相位的估计误差也将不断累积。当预测结果中各阶段开始、结束的位置与系统真实输出中阶段的起止位置无法对齐时, 此时点对点的误差计算并不适用于量化评估模型的预测误差。因此, 本节对模型预测功耗的长期累积结果进行评估。对于不同功率的数据集, 我们采用训练过的模型预测制冷系统未来在预测范围为  $[t_I : t_{I+L}]$  (共 120 分钟) 下的功耗。接下来, 依时间尺度对瞬时功耗进行积分可以得到某一时间长度 (T) 下的累积功耗。进一步地, 可以定义预测功耗的平均绝对百分比误差, MAPE(T):

$$\text{MAPE}(T) = \frac{100\%}{n} \sum_{i=1}^{\lfloor \frac{t_{I+L}-t_I}{T} \rfloor} \text{APE}(t_I + T * (i - 1), t_I + T * i) \quad (4-14)$$

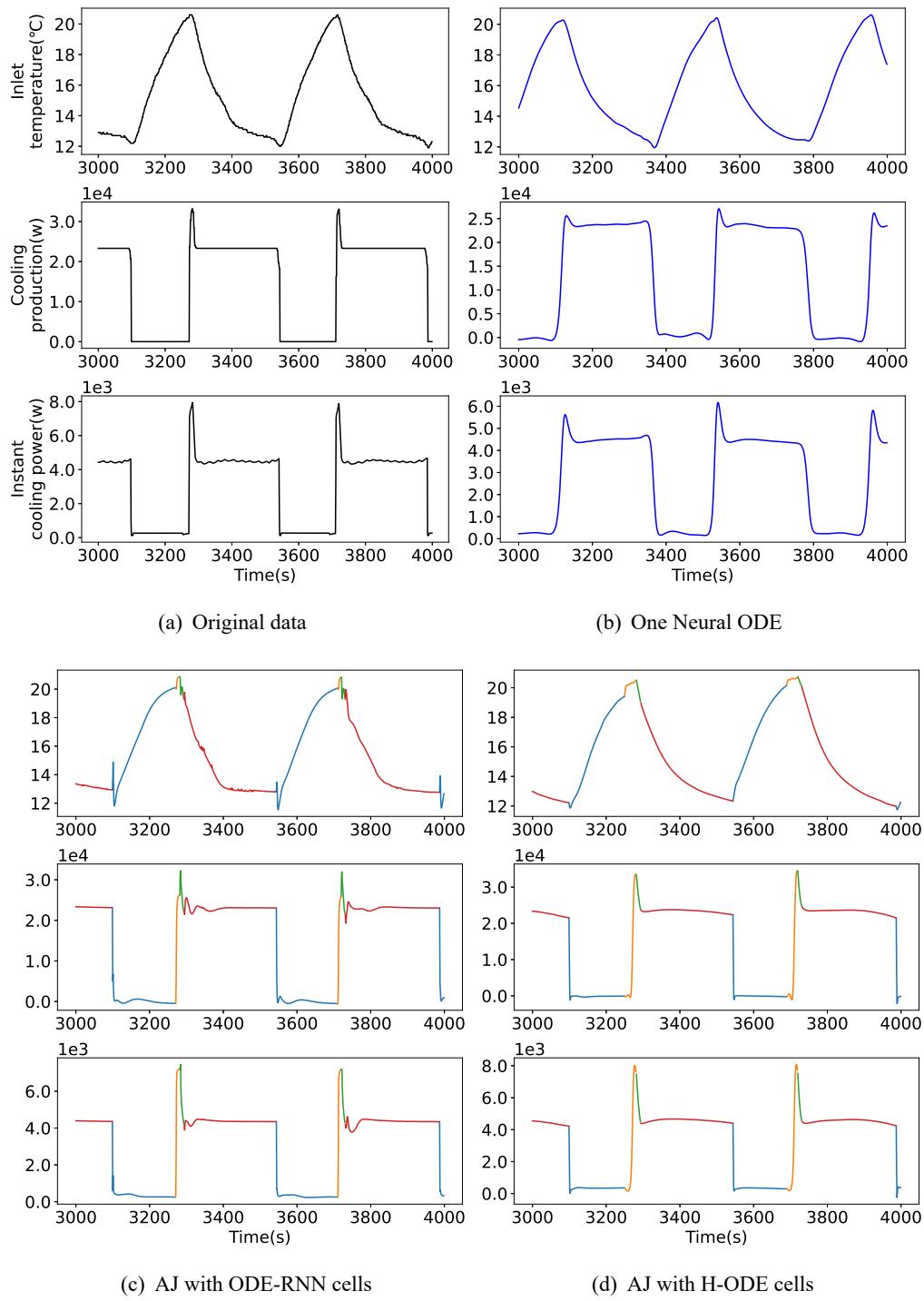


图 4-8 不同模型预测进气口温度、制冷量、制冷机功率的效果对比

表 4-3 不同模型累积能耗预测精度和推理时间的对比

	热负载				相对 时间
	1.7k	3.8k	4.2k	6.3k	
AJ-ODENet(本文模型)	<b>4.20</b>	<b>1.51</b>	<b>3.46</b>	<b>2.53</b>	3.2
One Neural ODE	6.18	19.99	6.24	3.39	1.0
Jump with CDE <sup>[20]</sup> cells	8.25	2.73	6.62	3.87	3.8
Jump with ODE-RNN cells	38.26	9.94	10.58	31.90	3.4

其中，其中绝对百分比误差 (APE) 定义如下：

$$APE(t_1, t_2) = \left| \frac{\int_{t_1}^{t_2} \hat{y}_1(t) dt - \int_{t_1}^{t_2} y_1(t) dt}{\int_{t_1}^{t_2} y_1(t) dt} \right| \quad (4-15)$$

其中，绝对百分比误差  $APE(t_1, t_2)$  衡量了时间范围  $[t_1, t_2]$  内估计功耗和真实功耗之间的相对误差。图 4-9 描述了评估窗口的大小  $T$  的选取如何影响评估结果  $MAPE(T)$ 。可以看出，随着  $T$  的增加， $MAPE(T)$  在开始时剧烈下降，然后缓慢下降。当评价窗口大小超过 35 分钟时，所有数据集的  $MAPE$  都趋于稳定，说明此时真实系统输出序列与预测序列之间的阶段起止位置相位差对于评估长期功耗的预测精度不再产生影响。因为本节对比了  $T = 30$  分钟下，不同模型对于累积功耗预测结果的  $MAPE$ ，结果如表 4-3 所示。结果表明对于所有数据集，AJ-ODENet 预测结果的  $MAPE$  稳定低于 5%，优于其他三个对比模型。充分证明了本文提出的 AJ-ODENet 框架能够以开环预测的方式在长时间尺度下精确地仿真制冷系统的累积功耗。

从计算复杂性的角度，本节对四个模型在测试集单批数据上执行(4-8)以及(4-9)两个过程的时间消耗进行了对比。Neural ODE 作为耗时最短的模型，将其时间消耗作为基准。其他模型的推理时间均处以该基准以得到相对尺度下的时间对比结果。Neural ODE 的相对时间消耗为 1.0。对比结果如表4-3的最后一列所示。相对时间尺度下的对比结果表明引入阶段转换预测器会增加计算的开销，该时间消耗主要用于更新阶段变量及阶段持续时间。

#### 4.4.5 应用研究 2：制冷系统进气口温度设定点优化

进气口温度设定点是影响制冷系统运行的关键设置参数之一。通常，系统要设定通入进气口的空气温度阈值（上限  $T_{i_{max}}$  和下限  $T_{i_{min}}$ ），以保证运行环境的安全。通常，对于大部分的制冷系统，初始的进气口温度设定点是固定的，配置值低于 ASHRAE TC9.9<sup>[123]</sup>（数据中心电源设备热指南和最佳实践）中定义的标准，没有充分考虑实际的制冷需求。以本章研究的制冷系统为例，其温度阈值为  $12^{\circ}\text{C}$ ,  $20^{\circ}\text{C}$ 。数据集中进气温度在两个阈值之间周期性变化。

出于提高能源效率和确保生产过程安全的目的，本节保持温度上限值不变，只调整下限值以减少能源消耗。然而，要确定最佳温度阈值是很棘手的。如果阈值

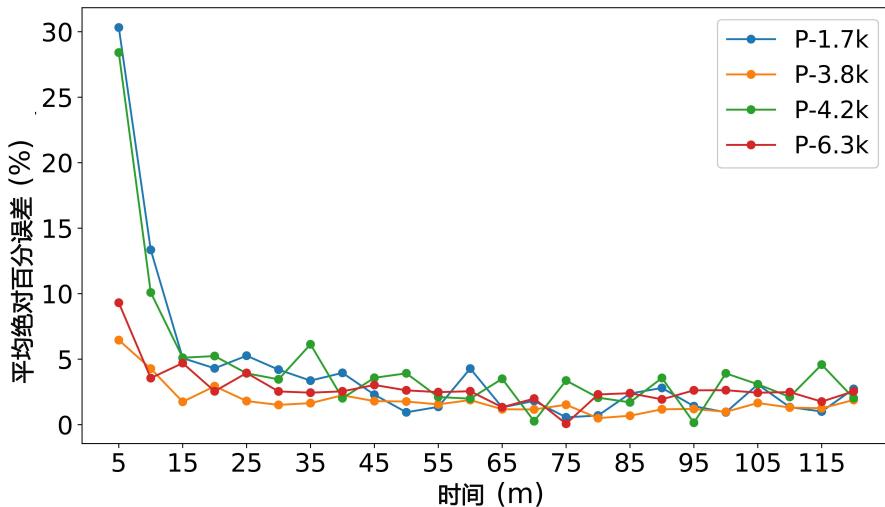


图 4-9 预测不同时间长度的功耗的 MAPE 的变化

设置得太低，由于过度冷却会浪费大量电能。如果阈值设置得太高，压缩机的制冷重启过程将更加频繁，这将在单位时间内形成更多的峰值功率。从图 ?? 中可以看出，制冷系统重新启动时的功耗显著高于提供稳定冷却时的功耗。因此，盲目地增加温度下限反而可能会导致的整体的能耗增加。

在本节实验中，我们试图寻找最佳的温度下限以达到更好的能源效率。形式化地，寻找最佳温度下限  $Ti_{min}^*$  可以表示为一个单目标优化问题，考虑到以下因素：热负载、环境温度、进气口温度的上下限：

$$Ti_{min}^* = \arg \min_{Ti_{min}} \int_0^T \hat{y}_1(t) dt \quad (4-16)$$

Where  $\hat{Y}_{0:T} = F(\mathbf{X}_{0:T}, Ti_{min}, Ti_{max}, \zeta)$

其中  $\hat{Y}_{0:T}$  和  $\hat{y}_1(t)$  分别表示模型预测的系统输出和模拟的瞬时功耗。在给定系统输入和恒定的上限值的条件下，其优化变量是温度下限的设定点，优化目标是最小化累计能耗。式(4-16)中的  $\mathbf{X}_{0:T}$  为测试数据集中的系统输入，其中包括所有时刻的热负载以及环境温度。式(4-16)中的  $F(\cdot)$  表示在给定输入和温度阈值  $Ti_{max}$  和  $Ti_{min}$  下，预测制冷系统的输出。

本节中，我们利用上一节 4.4.4 中训练的编码器-解码器 AJ-ODEnets 模型，在给定不同下边界温度设定点的情况下，仿真系统能耗以及温度变化，同时评估特定时间区间内的累积制冷功耗。实验采用第4.4.4节中所述的三个数据集，其的平均负载分别为 1.7k、3.8k 和 6.3k。温度下限值设置点从  $12^{\circ}\text{C}$  逐渐增加到  $18^{\circ}\text{C}$ ，调整间隔为  $0.5^{\circ}\text{C}$ 。为了模拟不同  $Ti_{min}$  下的冷却系统运行过程， $F(\cdot)$  中的预测过程与原始训练模型  $f(\cdot)$  略有不同。特别地，在阶段  $On$  下的持续时间预测器被特定的转换规则所替代。如表 4-1 中的规则，如果进气口温度经过冷却下降到  $Ti_{min}$ ，预测模型立即将阶段变量从 3 ( $On$ ) 过渡到 0 ( $Off$ )。

尽管本章所提出模型中的阶段转换预测器是基于持续时间预测器设计的,为了满足不同制冷系统运行参数的仿真需要,也可以手动设置状态转换规则。虽然在上述模拟过程中的温度下限与训练数据集对应的阈值参数是不同的,但 AJ-ODENet 仍然支持手动调整阶段过渡阈值以支持此类外推预测。相比于没有引入系统先验的稳态模型<sup>[?]</sup>, 基于先验知识的模型更具可解释性和可扩展性。同时,本章将进气口温度的变化建模成非平稳过程,使得在阶段 *Off* 期间,进气温度会稳定地上升,在阶段 *On* 期间,温度会稳定地下降,进而确保温度必然能够触达给定的阈值,这一模型特性与系统的先验知识一致,是保证进气口温度能够持续变化直至达到阈值的必要条件,有效避免了模型无限期停留某一阶段内难以跳出的情况。

图 4-10 显示了对于热负载约为 3.8k 的数据集,不同的进气口温度下限对制冷系统的影响。仿真时间为 1000 秒。随着温度下限设置点从 12°C 增加到 18°C, 在相同的持续时间内, 稳定制冷阶段的时间不断缩减。与此同时, 1000 秒内包含了更多的制冷系统启停周期。这导致系统的主要功耗来自于由 *Off* 阶段转变为 *on* 阶段的系统启动功耗,而非制冷功耗。

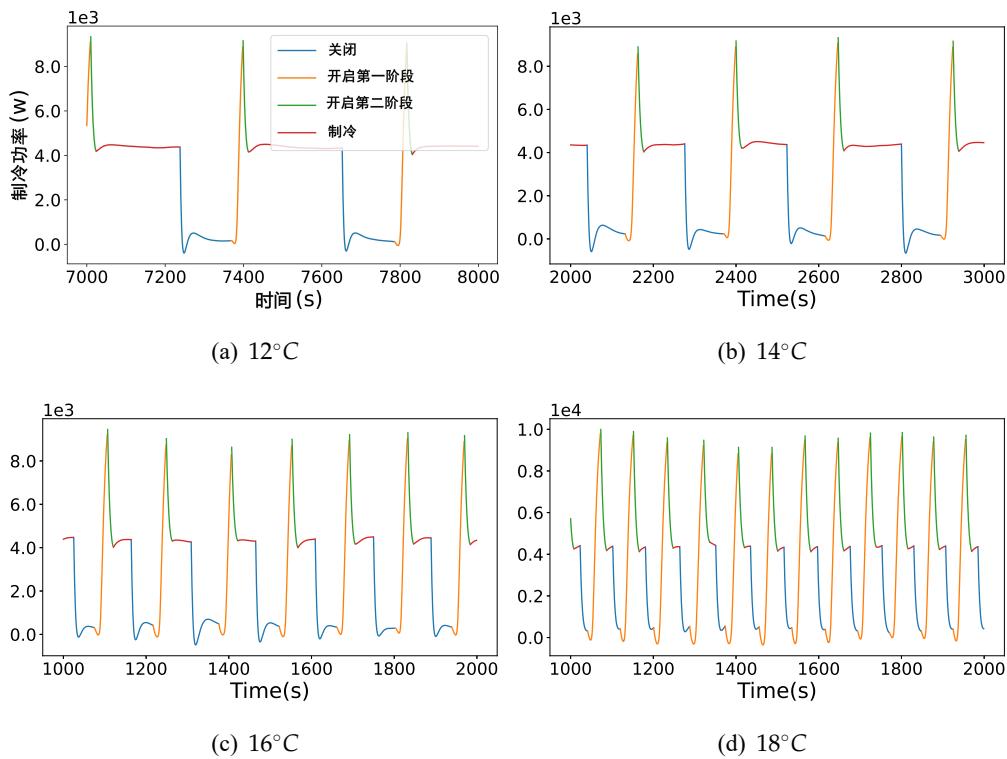


图 4-10 在不同温度下限设定值下的进气口温度的仿真值

在图 4-11 中, 我们评估了一个小时内, 不同  $T_{i_{\min}}$  下的预测累积能耗, PUE<sup>1</sup>, 以及性能系数(COP)<sup>2</sup>。PUE 代表整个总能耗(制冷能耗与设备能耗的和)与设备能耗的比。PUE 越接近 1, 说明整套系统越节能。COP 是实际产生的制冷量与所需能耗的比值。

<sup>1</sup>PUE = 总能耗 / 设备工作能耗<sup>[?]</sup>

<sup>2</sup>COP = 制冷量 / 制冷能耗<sup>[?]</sup>

表 4-4 能耗优化效果

不同负载的数据集	1.7k	3.8k	6.3k
最优温度设定值 (°C)	14	15	15.5
最大功耗优化比例 (%)	6.49	10.93	25.71

当温度下限值从  $12^{\circ}\text{C}$  增加到  $18^{\circ}\text{C}$  时，制冷系统总能耗先下降然后逐渐上升。由于设置较高的  $Ti_{\min}$  会缩小上下限之间的差距，使得压缩机更频繁地重启。虽然重启时间较短，但此时压缩机功耗是极高的。因此，从某一温度开始，由于压缩机重新启动导致的能量消耗占据了制冷总能耗的主要部分，在此之后总能量消耗就会上升。而这一拐点就是寻找的最佳  $Ti_{\min}$ 。对于三个不同热负荷的数据集，图 4.11(a)中标出了使总能耗最小的最佳的  $Ti_{\min}$ 。在图 4.11(b)中，最佳温度阈值与图 4.11(a)中的阈值一致。在图 4.11(c)中，COP 随着  $Ti_{\min}$  的增加不断降低。其原因为频繁的压缩机开闭过程会减少了稳定制冷的时间，进而导致单位功耗下产生的制冷量下降。COP 的下降趋势也解释了为什么提高温度下限值，所需的冷却量减少而制冷能耗却增加。

与此同时我们发现，最优温度设定点随负载的变化而变化。热负载越大，最优温度设定值越大。基于仿真实验推导出的最优温度设置点  $Ti_{\min}^*$ ，表 4-4 展示了优化温度设定点可以节省的功耗百分比。根据仿真结果，采用新的温度下限设定值可以节省约 6-25% 的能耗，特别对于高热负载情况下的能耗优化是极其显著的。因为实验 4.4.4 中评估了模型预测累积能耗的相对百分比误差小于 5%，可以认为本节对于不同温度下限设定值的仿真结果是有足够可信度。在未来的工作中，将进一步验证该温度设定策略在实际工业制冷系统中的能耗优化表现。

## 4.5 本章小结

针对连续时间周期跳变系统的建模预测问题，本文提出了一种基于 H-ODENet 的 AJ-ODEnets 模型，同时基于该模型构建了用于系统开环预测的编码器-解码器框架。该模型能够对给定的历史序列数据进行编码，在给定未来系统输入序列下仿真预测系统的未来输出。为了使 AJ-ODEnet 更好地学习不同阶段的动态特性，AJ-ODEnet 包含了多个层次常微分方程网络 (Hierarchical Neural ODE network)，独立地学习每个阶段下的系统动态特性，并分别建模系统的稳定输出和非稳定输出，同时构建阶段转换预测器以指定各个预测时刻所用的 H-ODENet。作为 ODENet 的扩展，H-ODENet 中的双层结构分别用于估计隐状态的导数和系统输出的导数，并且对于稳定输出和非稳定输出采用不同的导数模块进行建模。最后，本章利用所提出的基于 AJ-ODEnet 的编码器-解码器框架建模某一工业制冷系统。与其他方法相比，AJ-ODEnet 在对于制冷系统温度、功耗的开环预测问题中表现出了良好性能，并且能够准确地预估阶段转换点。此外，依托于训练得到的仿真模型，以最小化不同热负荷下的制冷能耗为目标，我们成功优化了制冷系统的温度下限设定点。仿真结果表明，通过采纳优化后温度阈值设置可以节省高达 6-25% 的制冷能耗。

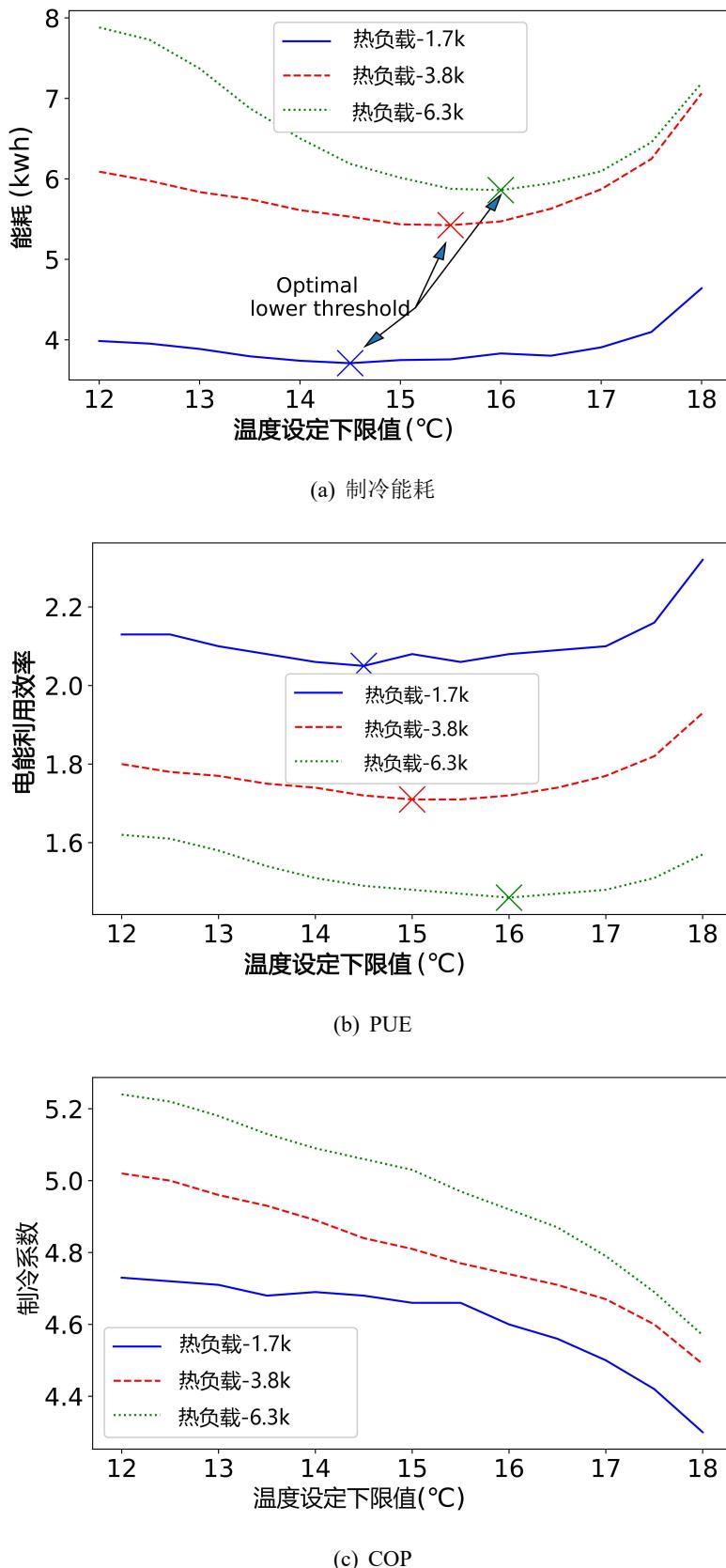


图 4-11 在不同热负载下改变温度设定下限值对功耗、COP、PUE 的影响

## 5 基于深度常微分马尔可夫模型的随机非确定性系统建模

基于深度学习的系统识别方法结合了深度神经网络在建模非线性复杂函数的优势，实现利用离线输入输出数据识别未知参数结构的黑盒系统动态。训练后的模型可以作为实际系统的近似替代，以支撑仿真预测、模型预测控制和基于模型的强化学习 (Model-based Reinforcement learning, MBRL) 等下游决策优化任务。然而，在建模真实工业系统时经常面临如下挑战：首先，在以往的研究中，许多基于神经网络的系统辨识模型属于确定性模型，无法对系统状态转移的随机性以及不完备观测造成的确定性进行建模。因此，此类模型仅适用于对确定性动态系统进行建模，不适用于带有随机性或观测空间存在非确定性的系统。

其次，在实际工业应用中，时间间隔不均匀的采样数据是普遍存在的<sup>[19]</sup>。依赖于均匀采样间隔假设的离散时间模型不便于从非均匀采样数据进行学习。尽管可以采用离线插值的方式强行调整训练集数据的采样间隔，不过模型仍要求推理数据和训练数据的采样间隔需保持一致，因此在实时工业应用中此类模型存在一定的局限性。

第三，在一些复杂的工业系统中，系统输入对系统输出的影响不是瞬时的。系统关键指标受外部输入的影响变化可能存在长时间的滞后性，当使用辨识的模型进行长期开环预测时，如何保证长期预测结果的鲁棒性和准确性对模型的设计及训练提出了更高的要求。最后，在某些应用场景下，训练完成的识别模型需要工作在线模式下，用于处理连续的流式数据，且其预测范围及输入序列长度可能是动态变化的<sup>[124]</sup>。这一要求进一步限制了某些端到端编码器-解码器模型的使用<sup>[49,60]</sup>，更倾向于选择具有循环状态更新的模型。

现有研究成果往往只能解决上述问题中的一个或几个。然而，目前尚没有某一模型能够同时解决上述所有问题。因此，本文提出常微分方程循环状态空间模型 (Ordinary Differential Equation Recurrent State Space Model, ODE-RSSM)，该模型一个定义在连续时间 (CT) 域中的随机过渡模型，该模型描述了系统状态在连续时间域下的随机演化，能够用于从不规则样本数据中学习具有随机性和长时滞特性的动态系统。具体地，相比于离散时间域下的循环状态空间模型，ODE-RSSM 通过在状态演化中加入 ODE-Net 以描述模型内部状态对时间的导数，以此建模任意长度时间间隔的状态演变。另外，在训练阶段，为了使该模型能够支持批量并行化推理与训练，需要解决同一数据批中求解时间点彼此不同、积分区间彼此不一致的问题。因此本章提出了一种批常微分方程并行求解方法，通过构造辅助常微分方程以对其各常微分方程的求解时间点，进而利用现有的深度学习框架高效地求解多个积分区间不同的常微分方程。此外，本章在训练阶段引入了一种计算复杂度更低的隐空间超调技术，在不显著增加时间复杂度的情况下有效改善模型的长期预测性能。在实验环节，我们在共 3 个系统辨识数据集 (2 个公共数据集和 1 个私有数据集) 上评估了所述模型在长期开环预测问题中的性能。实验表明在非均

表 5-1 本章提出方法与现有系统建模方法的对比

模型	随机性	非均匀采样	在线预测
RNN	✗	✗	✓
RSSM	✓	✗	✓
Time-Aware RNN	✗	✓	✓
ODE-RNN, SNODE	✗	✓	✓
Latent SDE	✓	✓	✗
<b>ODE-RSSM</b>	✓	✓	✓

匀采样下的随机系统预测任务上，ODE-RSSM 无论是在处理非均匀采样方面还是建模系统随机性方面均具有良好表现，预测精度显著优于离散时间状态空间模型以及带有确定性状态演变的连续时间模型。相比于其他基于神经网络的系统建模方法，本章提出模型的优势如表5-1所示。

本章所述工作的核心贡献总结如下：

1. 本章提出了 ODE-RSSM 模型，作为深度状态空间模型的扩展，该模型能够利用不均匀采样数据识别具有随机性和长时延特性的输入输出系统。
2. 为了改善模型的长期预测，本章在 ODE-RSSM 的训练阶段引入了采样状态重用的隐空间超调技术以提升模型的长期预测能力。
3. 本章使用三个输入/输出系统数据集对 ODE-RSSM 模型的预测性能和所述加速效果进行了评估。其中一个数据集导出于具有明显时滞和随机特性的工业浓密系统。该数据集为首次公开的私有数据集。

本章的结构组织如下：本章??节对本章相关工作进行简介。本章??节对基于变分编码机模型的系统建模方法进行间接。本章5.2节详细介绍了 ODE-RSSM 的模型细节及训练方法。本章6.4节详述了实验设计、实验结果、以及实验结果讨论。本章??节讨论本文模型的局限及其潜在应用。最后，6.5节对本文研究工作做出总结。

## 5.1 问题形式化描述

对于任一输入输出系统，对其非均匀采样数据进行定义，包括序列系统输入为  $\{\mathbf{u}_{t_i}\}_{i=1}^N$ ，以及序列系统输出为  $\{\mathbf{y}_{t_i}\}_{i=1}^N$ ，其中  $t_i$  为第  $i$  个位置数据点的采样时间， $N$  为完整序列长度。本文假设对于各维度输入和各维度输出的采样都是同步地。

本章将具有随机非确定性的系统表示为包含随机隐变量的受控状态空间模型。模型的隐状态由确定性部分  $\mathbf{h}_{t_i}$  和随机部分  $\mathbf{z}_{t_i}$  共同组成。为了描述简便，本章定义  $\mathbf{s}_{t_i} = [\mathbf{z}_{t_i}, \mathbf{h}_{t_i}]$  用于表示  $t_i$  时刻完整的隐状态。该序列隐变量描述了系统输出在受控输入下的条件生成过程。利用生成模型参数  $\theta$ ，定义序列隐变量和观测值的联

合概率如下:

$$\begin{aligned} p_{\theta}(\mathbf{y}_{1:N}, \mathbf{z}_{1:N}, \mathbf{h}_{1:N} | \mathbf{u}_{1:N}, \mathbf{h}_0) &= \prod_{i=1}^N p_{\theta}(\mathbf{y}_i | \mathbf{z}_i, \mathbf{h}_i) \times \\ &\quad \times p_{\theta}(\mathbf{z}_i | \mathbf{h}_i) \tilde{p}_{\theta}(\mathbf{h}_i | \mathbf{u}_{i-1}, \mathbf{z}_{i-1}, \mathbf{h}_{i-1}) \end{aligned} \quad (5-1)$$

本章提出的 ODE-RSSM 为时序变分自编码机模型，模型包括生成模块和推断模块两部分。推断模块能够在给定系统输入和输出的情况下推断序列隐状态  $\{\mathbf{s}_{t_i}\}_{i=1}^N$ ，其生成模型能够在给定系统输入  $\{\mathbf{u}_{t_i}\}_{i=1}^N$  下，识别  $\mathbf{z}_{1:N}$  和  $\mathbf{h}_{1:N}$  的变化，进而预测系统输出  $\{\mathbf{y}_{t_i}\}_{i=1}^N$  的条件分布。

## 5.2 常微分方程网络-循环状态空间模型

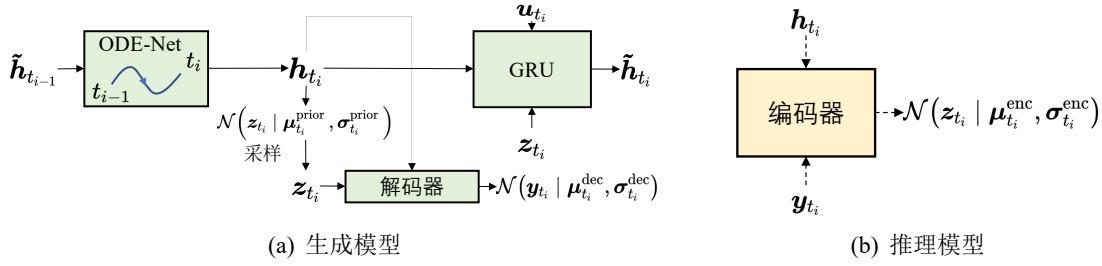


图 5-1 ODE-RSSM 中的生成过程和推理过程

本节首先介绍 ODE-RSSM 的生成模型和推断模型，如图 5-1 所示。在生成模型中，在给定系统输入的情况下，模型通过组合门控循环单元 (Gate Recurrent Unit, GRU) 和 ODE 网络建模隐状态的转移过程。为了训练生成模型参数  $\theta$ ，本章通过引入近似后验编码器  $q_{\phi}(\mathbf{z}_i | \mathbf{y}_i, \mathbf{h}_i)$  对隐变量的真实后验分布进行近似，以最大化证据下限 (Evidence Lower Bound, ELBO) 的方式协同训练参数  $\phi$  和  $\theta$ 。另一方面，我们在训练阶段进一步引入隐空间超调技术<sup>[76]</sup> 用于改善模型的多步预测。为了减少引入隐空间超调带来的额外时间开销，进而优化训练效率。本章通过重用临时采样状态的方式显著减少了求解多步 KL 散度的时间复杂度。

### 5.2.1 连续时间域下含随机隐变量的生成模型

ODE-RSSM 采用带有隐变量的条件概率模型表示序列生成过程，其描述了在给定系统输入条件下隐变量和系统输出的联合概率分布。给定初始时刻隐变量  $\mathbf{s}_{t_1}$ ，序列变量  $\mathbf{s}_{t_i}$  的条件先验分布由系统输入决定：

$$p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-1}}, \mathbf{u}_{t_{i-1}}) = p(\mathbf{s}_{t_i} | \mathbf{h}_{t_i}) * p(\mathbf{h}_{t_i} | \mathbf{s}_{t_{i-1}}, \mathbf{u}_{t_{i-1}}) \quad (5-2)$$

在式 (5-2) 中， $p(\mathbf{h}_{t_i} | \mathbf{h}_{t_{i-1}}, \mathbf{u}_{t_{i-1}})$  为德塔分布，并且在给定  $t_{i-1}$  时刻的的隐变量  $\mathbf{s}_{t_i}$  和系统输入  $\mathbf{u}_{t_i}$  下， $\mathbf{h}_{t_i}$  是确定性的。

由于在采样不规则的情况下，在不同时刻  $t_i$  下的采样间隔  $t_i - t_{i-1}$  不是恒等的，一般的离散时间序列模型不适用于建模此类可变时间范围上的隐状态演化。因此，本章受微分方程网络模型<sup>[49,125]</sup>的启发，引入常微分方程网络学习确定性隐状态  $\mathbf{h}_{t_i}$  在连续时间域上演化。

具体地，确定性状态  $\mathbf{h}_{t_i}$  的生成及转化过程分为两个阶段。在第一阶段，我们采用门控循环单元 (Gate Recurrent Unit, GRU) 建模系统输入  $\mathbf{u}_{t_i}$  以及  $\mathbf{z}_{t_i}$  对当前状态  $\mathbf{h}_{t_{i-1}}$  的瞬时影响，并构建中间变量  $\tilde{\mathbf{h}}_{t_{i-1}}$ ：

$$\tilde{\mathbf{h}}_{t_{i-1}} = \text{GRU}([\mathbf{u}_{t_{i-1}}, \mathbf{z}_{t_{i-1}}], \mathbf{h}_{t_{i-1}}) \quad (5-3)$$

下一步，引入参数化 ODE-Net  $f_\theta$  用于建模  $\mathbf{h}_t$  在时间范围  $t_{i-1} - t_i$  内对时间  $t$  的导数，通过求解 ODE 预测时间点  $t_i$  处的更新状态  $\mathbf{h}_{t_i}$ ：

$$\begin{aligned} \mathbf{h}_{t_i} &= \text{ODE Solve}(f_\theta, \tilde{\mathbf{h}}_{t_{i-1}}, t_{i-1}, t_i) \\ &= \tilde{\mathbf{h}}_{t_{i-1}} + \int_{t_{i-1}}^{t_i} f_\theta(\mathbf{h}_t, \mathbf{u}_{t_{i-1}}) dt \end{aligned} \quad (5-4)$$

此处，系统输入  $\mathbf{u}_{t_{i-1}}$  作为常微分方程的输入，被定义为分段恒定函数  $\mathbf{u}[t_{i-1}, t_i] = \mathbf{u}_{t_{i-1}}$ 。一般的常微分方程网络采用多层感知机建模导数函数。如本文第三章研究表明，连续地对导数进行积分会导致隐状态不断地增加，使得增量式的 ODE 方程不适用于长时间序列预测和系统辨识问题。本章提出一种正交的导数定义方法<sup>[126]</sup>：

$$\frac{d\mathbf{h}(t)}{dt} = \text{NN}_\theta(\mathbf{h}(t)) - \frac{\mathbf{h}(t)}{|\mathbf{h}(t)|} * \frac{\langle \text{NN}_\theta(\mathbf{h}(t)), \mathbf{h}(t) \rangle}{|\mathbf{h}(t)|}, \quad (5-5)$$

其中， $\text{NN}_\theta(\cdot)$  是标准的多层感知机， $\langle \cdot, \cdot \rangle$  是向量内积。普通的 ODE 单元、稳定的 ODE 单元以及本章提出的正交 ODE 单元之间的区别如5-2所示：正交形式保

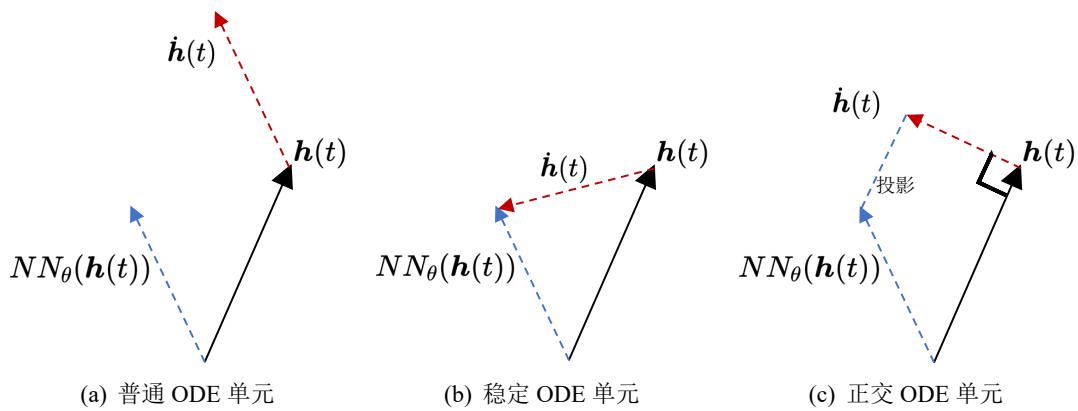


图 5-2 三种导数模块定义方法图示

证了导数预测值  $\frac{dh(t)}{dt}$  始终与当前状态  $\mathbf{h}(t)$  正交。因此，从  $t_i$  到  $t_{i+1}$  的积分过程中，只会改变  $\mathbf{h}(t)$  的方向，其二范数不变。 $|\mathbf{h}(t)|$  仅由 GRU 网络决定，且被严格限制

在  $[-1, 1]$ 。该导函数改进方法能够避免长采样间隔下由于增量式更新导致的隐状态范围过大的问题。

通过求解常微分方程得到  $\mathbf{h}_{t_i}$  后，进而可以预测  $\mathbf{z}_{t_i}$  的先验高斯分布：

$$p_{\theta}(\mathbf{z}_{t_i} | \mathbf{h}_{t_i}) = \mathcal{N}(\mathbf{z}_{t_i} | \boldsymbol{\mu}_{t_i}^{\text{prior}}, \boldsymbol{\sigma}_{t_i}^{\text{prior}}) \quad (5-6)$$

其中，采用多层感知机模型估计分布参数  $\boldsymbol{\mu}_{t_i}^{\text{prior}}$  和  $\boldsymbol{\sigma}_{t_i}^{\text{prior}}$ ，模型输入为确定性的隐状态  $\mathbf{h}_{t_i}$ 。在生成过程中， $\mathbf{h}_{t_i}$  可以表示在获得  $\mathbf{u}_{t_i}$  之前，在时刻  $t_i$  下预测的隐状态。而  $\tilde{\mathbf{h}}_{t_i}$  可表示在给定  $\mathbf{z}_{t_i}$  和  $\mathbf{u}_{t_i}$  的信息之后更新隐状态。

为了预测系统输出的分布，进一步引入解码器模块预测系统输出  $\mathbf{y}_{t_i}$  在给定隐状态  $s_t$  下的高斯分布：

$$p(\mathbf{y}_{t_i} | \mathbf{h}_{t_i}, \mathbf{z}_{t_i}) = \mathcal{N}(\mathbf{y}_{t_i} | \boldsymbol{\mu}_{t_i}^{\text{dec}}, \boldsymbol{\sigma}_{t_i}^{\text{dec}}) \quad (5-7)$$

从公式中可以看到，分布参数  $\boldsymbol{\mu}_{t_i}^{\text{dec}}, \boldsymbol{\sigma}_{t_i}^{\text{dec}}$  由确定性隐状态和随机隐状态共同决定。在给定序列系统输入的情况下，通过对序列隐状态分布进行反复预测与采样，我们可以通过开环的方式预测序列隐变量，进而预测系统序列输出。

### 5.3 隐变量推理与训练

为了训练生成模型中的参数  $\theta$ ，常规方法是最大化系统输出的对数似然  $\mathcal{L} = \sum_{i=1}^N \log p_{\theta}(\mathbf{y}_{t_i} | \mathbf{u}_{<t_i})$ 。因为直接计算隐变量  $\mathbf{s}_{t_1:t_N}$  的精确后验分布是极其困难的，我们无法通过对隐变量概率分布进行积分的方式估计精确的对数似然  $\mathcal{L}$ 。因此，我们引入变分分布  $q_{\phi}$  近似随机变量的后验分布，进而辅助生成模型的学习。

$$\begin{aligned} q_{\phi}(\mathbf{z}_{t_1:t_N} | \mathbf{y}_{t_1:t_N}, \mathbf{u}_{t_1:t_N}) &= \prod_{i=1}^N q_{\phi}(\mathbf{z}_{t_i} | \mathbf{h}_{t_i}, \mathbf{y}_{t_i}) \\ &= \prod_{i=1}^N \mathcal{N}(\mathbf{z}_{t_i} | \boldsymbol{\mu}_{t_i}^{\text{enc}}, \boldsymbol{\sigma}_{t_i}^{\text{enc}}) \end{aligned} \quad (5-8)$$

如图5.1(b)所示，变分分布  $q_{\phi}(\mathbf{z}_{t_i} | \cdot)$  为高斯分布，其分布参数  $\boldsymbol{\mu}_{t_i}^{\text{enc}}$  和  $\boldsymbol{\sigma}_{t_i}^{\text{enc}}$  由 MLP 网络构建的编码器模块进行估计。因为求解  $\mathbf{h}_{t_i}$  需要之前的随机隐状态  $\mathbf{z}_{<t_i}$ ，在推理序列后验  $q_{\phi}(\mathbf{z}_{t_1:t_N} | \mathbf{y}_{t_1:t_N}, \mathbf{u}_{t_1:t_N})$  时，生成过程和推理过程是被交互调用，不断采样出新的  $\mathbf{z}_{t_i}$  并预测  $\mathbf{h}_{t_i}$ 。

利用后验推理分布，可以通过最大化观测系统输出的证据下界 (ELBO) 训练 ODE-RSSM 模型：

$$\begin{aligned}
\ln p(\mathbf{y}_{t_1:t_N}) &= \ln \int \prod_{i=1}^N p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-1}}, \mathbf{u}_{t_{i-1}}) p(\mathbf{y}_{t_i} | \mathbf{s}_{t_i}) d\mathbf{s}_{t_1:t_N} \\
&\geq \sum_{i=1}^N \left( \mathbb{E}_{q(\mathbf{s}_{t_i})} [\ln p(\mathbf{y}_{t_i} | \mathbf{s}_{t_i})] \right. \\
&\quad \left. - \mathbb{E}_{q(\mathbf{s}_{t_i})} [\text{KL}[q(\mathbf{s}_{t_i}) \| p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-1}}, \mathbf{u}_{t_{i-1}})]] \right)
\end{aligned} \tag{5-9}$$

为了表述方便，此处将  $q_\phi(\mathbf{s}_{t_i} | \mathbf{y}_{\leq t_i}, \mathbf{u}_{< t_i})$  简化为  $q(\mathbf{s}_{t_i})$ 。

对于推理模型，除了可以用于近似隐变量的后验分布以估计证据下限，推理模型还可作为序列编码模型用于对系统状态进行编码。作为一个支持状态循环更新的编码器-解码器框架，ODE-RSSM 可以灵活地采用编码器模块进行推断并采用生成模型进行预测。在一些具有流式数据的在线任务中，两个模块可能需要被迭代调用。例如，在模型预测控制问题中，利用推理模型可以对给定的历史输入输出数据进行编码以得到隐状态的采样，利用生成模型可以对任意给定控制输入下的系统输出进行预测，进而辅助系统控制。

## 5.4 基于高效隐空间超调的多步预测性能改善

在基于模型的序贯决策问题中，模型需要根据给定的控制输入序列进行多步预测<sup>[76]</sup>。在这种情况下，需要重复地调用生成模型进行隐变量的预测，并将从前一步预测分布中采样得到的隐状态作为下一步预测的输入。然而，对于式(5-9)定义的损失函数，其 KL 散度项仅度量了近似后验分布和单步预测分布之间的差异。因此，在模型训练时，反向传播的梯度流只对生成模型的单步预测过程进行了优化。另外，在训练时，生成模型的隐变量输入始终为近似后验分布的采样。而多步预测时，其隐变量输入大多数情况下来源于上一步的预测分布。近似后验分布和预测先验分布之间的差异打破了生成模型输入在训练集和测试集上的同分布假设，这可能导致多步预测的准确性较差<sup>[?]</sup>。对于存在长时延或观测空间不完备的复杂系统，这一问题将更加明显。因此，本章在 ODE-RSSM 的训练过程中引入了隐空间超调技术<sup>[76]</sup> 用于改善生成过程中的多步预测性能。

具体地，对于长度为  $d$  的多步隐状态预测过程可以表示为：

$$\begin{aligned}
p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-d}}) &\triangleq \int \prod_{\tau=i-d+1}^i p(\mathbf{s}_{t_\tau} | \mathbf{s}_{t_{\tau-1}}, \mathbf{u}_{t_{\tau-1}}) d\mathbf{s}_{t_{i-d+1}:t_{i-1}} \\
&= \mathbb{E}_{p(\mathbf{s}_{t_{i-1}} | \mathbf{s}_{t_{i-d}})} [p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-1}})]
\end{aligned} \tag{5-10}$$

由于通过对所有中间隐变量的概率分布进行积分的方式求解多步预测分布是极其困难的。对于多步预测，我们通常采用多次祖先采样 (Ancestral sampling) 和重参数化技巧 (Reparameterization Trick) 不断地对单步预测分布中的中间状态进行采样。

基于式(5-10)定义的  $d$ -步预测先验分布，我们进而可以推导出定义在多步预测

先验分布上的 ELBO:

$$\begin{aligned}
 \ln p_d(\mathbf{y}_{t_1:t_N}) &= \ln \int \prod_{i=1}^N p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-d}}) p(\mathbf{y}_{t_i} | \mathbf{s}_{t_i}) d\mathbf{z}_{t_1:t_N} \\
 &\geq \sum_{i=1}^N \left( \mathbb{E}_{q(\mathbf{s}_{t_i})} [\ln p(\mathbf{y}_{t_i} | \mathbf{s}_{t_i})] \right. \\
 &\quad \left. - \mathbb{E} [\text{KL}[q(\mathbf{s}_{t_i}) \| p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-1}}, \mathbf{u}_{t_{i-1}})]]) \right. \\
 &\quad \left. p(\mathbf{s}_{t_{i-1}} | \mathbf{s}_{t_{i-d}}) q(\mathbf{s}_{t_{i-d}}) \right)
 \end{aligned} \tag{5-11}$$

根据文献<sup>[76]</sup>给出的证明,  $\ln p_d(\mathbf{y}_{t_1:t_N})$  是式(5-9)给出的单步预测 ELBO 的下限, 自然也是原始系统输出对数似然的下限。给定超参数  $D$ , 我们可以对多种可能的预测距离  $1 \leq d \leq D$  的变分下界求平均, 以改进不同距离下的多步预测精度:

$$\begin{aligned}
 \text{ELBO-D} &= \frac{1}{D} \sum_{d=1}^D \ln p_d(\mathbf{y}_{t_1:t_N}) \\
 &\geq \mathbb{E}_{q(\mathbf{s}_{t_1:t_N})} [\ln p(\mathbf{y}_{t_1:t_N} | \mathbf{s}_{t_1:t_N})] - \frac{1}{D} * \text{KL-D}
 \end{aligned} \tag{5-12}$$

其中

$$\text{KL-D} = \sum_{d=1}^D \sum_{i=1}^N \mathbb{E} [\text{KL}[q(\mathbf{s}_{t_i}) \| p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-1}})]]
 \frac{p(\mathbf{s}_{t_{i-1}} | \mathbf{s}_{t_{i-d}}) q(\mathbf{s}_{t_{i-d}})}{p(\mathbf{s}_{t_{i-1}} | \mathbf{s}_{t_{i-d}}) q(\mathbf{s}_{t_{i-d}})} \tag{5-13}$$

为简化表述, 此处省略了 KL 散度项中的  $\mathbf{u}$  和  $\mathbf{y}$ 。

利用多步 ELBO 训练时序 VAE 模型确实能够在一定程度上改善长期开环预测的准确度<sup>[76]</sup>。然而, 依照公式(5-13)直接估计 KL-D 的时间复杂度为  $\mathcal{O}(N * D^2)$ , 其中包括对于最大距离  $D$  的外循环  $1 \leq d \leq D$ , 以及式(5-10)所示的多步预测采样  $p(\mathbf{s}_{t_{i-1}} | \mathbf{s}_{t_{i-d}})(\mathcal{O}(d))$ 。在本章中, 我们提出了一种加速估计 KL-D 的优化方法, 该方法能够合并外循环和多步采样过程的计算复杂度, 使总的估计 KL-D 的时间复杂度变为  $\mathcal{O}(N * D)$ 。通过分析式(5-10)可以得知, 在  $i - d$  确定的情况下, 不同预测距离  $d$  下的多步采样操作之间是存在冗余的。具体地, 当  $i - d$  确定的情况下, 不管预测距离  $d$  如何变化, 对于  $i - d + 1 \leq \tau \leq i - 1$  下的单步预测过程  $p(\mathbf{s}_{t_\tau} | \mathbf{s}_{t_{\tau-1}})$  存在于所有的多步预测  $p(\mathbf{s}_{t_{i-1}} | \mathbf{s}_{t_{i-d}})$  中。因此, 对于不同的超调距离  $d$ , 我们可以共用中间的采样状态  $\mathbf{s}_{t_{i-d}:t_{i-1}}$ , 并得到 KL 散度的无偏估计。

具体地, 我们将 KL-D 的计算公式重写为:

$$\begin{aligned}
 \text{KL-D} &= \sum_{i=1}^N \sum_{d=1}^D \mathbb{E} [\text{KL}[q(\mathbf{s}_{t_{i+d}}) \| p(\mathbf{s}_{t_{i+d}} | \mathbf{s}_{t_{i+d-1}})]]
 \frac{p(\mathbf{s}_{t_{i+d-1}} | \mathbf{s}_{t_i}) q(\mathbf{s}_{t_i})}{p(\mathbf{s}_{t_{i+d-1}} | \mathbf{s}_{t_i}) q(\mathbf{s}_{t_i})} \\
 &= \sum_{i=1}^N \mathbb{E}_{\hat{\mathbf{s}}_{t_i} \sim q(\mathbf{s}_{t_i})} \mathbb{E}_{p(\hat{\mathbf{s}}_{t_{i+1}:t_{i+D-1}} | \hat{\mathbf{s}}_{t_i})} \sum_{d=1}^D \text{KL}(i + d)
 \end{aligned} \tag{5-14}$$

**算法 2** 非均匀采样下的多步 KL 散度求解算法

**输入:** 生成网络(包含 ODE-Net  $f_\theta$  和 GRU 模块), 后验编码模块, 序列输入  $\mathbf{u}_{t_1:t_N}$  以及序列输出  $\mathbf{y}_{t_1:t_N}$

**输出:** KL-D

- 1: 根据式(5-3)、(5-4)及(5-8)迭代采样  $\mathbf{z}_{t_1:t_N}, \mathbf{h}_{t_1:t_N}$  并推理近似后验分布  $q_\phi(\mathbf{z}_{t_1:t_N} | \mathbf{h}_{t_1:t_N}, \mathbf{y}_{t_1:t_N})$ 。
- 2: 初始化  $\hat{\mathbf{z}}_{t_1:t_N} = \mathbf{z}_{t_1:t_N}, \hat{\mathbf{h}}_{t_1:t_N} = \mathbf{h}_{t_1:t_N}$
- 3: **for**  $d := 1$  to  $D$  **do**
- 4:   适用 GRU 模块更新确定隐状态:  $\tilde{\mathbf{h}}_{t_d:t_N} = \text{GRU}([\mathbf{u}_{t_d:t_N}, \hat{\mathbf{z}}_{t_d:t_N}], \hat{\mathbf{h}}_{t_d:t_N})$
- 5:   根据式(5-18), 给定  $g_\theta = f_\theta(\mathbf{R}(t)) \circ (t_{d+1:N} - t_{d:N-1})$  及  $\mathbf{R}(0) = \tilde{\mathbf{h}}_{t_d:t_{N-1}}$ , 求解  $\hat{\mathbf{h}}_{t_{d+1}:t_N} = \mathbf{R}(1)$ 。
- 6:    $\text{KL}^d = \text{KL} [p(\mathbf{z}_{t_{d+1}:t_N} | \hat{\mathbf{h}}_{t_{d+1}:t_N}) || q(\mathbf{z}_{t_{d+1}:t_N})]$
- 7: **end for**
- 8: **返回**  $\text{KL-D} = \sum_{i=1}^D \text{KL}^d$

其中  $\text{KL}(i+d)$  为  $\text{KL}[q(\mathbf{s}_{t_{i+d}}) \| p(\mathbf{s}_{t_{i+d}} | \hat{\mathbf{s}}_{t_{i+d-1}})]$  的缩写。 $\hat{\mathbf{s}}_{t_i}$  是从后验推理分布中采样的初始状态,  $\hat{\mathbf{s}}_{t_{i+1}:t_{i+D-1}}$  为在基于祖先采样方法得到的序列隐状态, 即对单步预测分布进行迭代采样, 并采样状态作为下一步预测过程的输入。

根据式(5-14)可知, 改进后的采样方法能够获得 KL-D 的无偏估计, 且其时间复杂度为  $\mathcal{O}(N \times D)$ 。隐状态采样和计算 KL 散度的过程如图 5-3 所示。利用本章 5.5 所

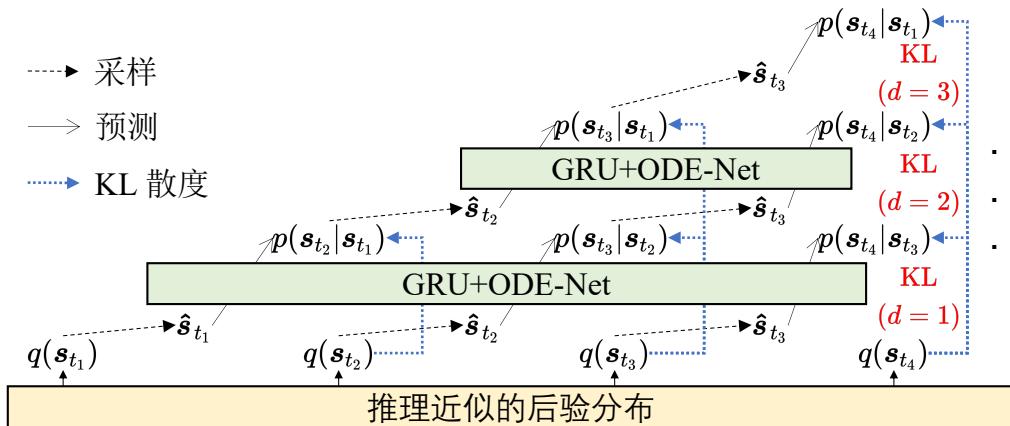


图 5-3 基于祖先采样的高效隐空间超调

述的批 ODE 求解算法能够并行化地求解式(5-14)中的循环  $1 \leq i \leq N$ , 当 GPU 的显存充足时, 估计 KL-D 的实际时间消耗将随着最大超调长度  $D$  线性增长。求解 KL-D 的完整过程如算法 2 所示。本章提出的高效隐空间超调技术能够在仅增加较小训练开销的代价下, 显著提升模型的多步预测能力, 且该方法对于其他深度状态空间模型的训练同样具有适用性。

## 5.5 非均匀采样间隔下的批量常微分方程并行化求解方法

网络通常被批量训练，以利用高效的数据并行 GPU 操作。利用 GPU 的并行计算能力对神经网络进行批量数据训练，可以显著减少神经网络的训练时间。这是当前训练深度模型的必要前提。因此，使 ODE-RSSM 模型支持并行推理和预测是极其重要的。在生成过程中，式(5-3)表示的 GRU 更新和式(5-6)表示的先验分布预测过程是很容易实现并行化的。而对于式(5-4)描述的求解 ODE 的过程，当序列采样点不均匀时，需要并行求解多个积分时间间隔彼此不同的数值积分现有的大多数神经微分方程求解工具假设批数据中的求解时间点是彼此相同的，并且大多数不支持对不同积分区间进行批处理<sup>[19]</sup>。为了让 ODE-net 模型适用批量的非均匀采样数据，现有的基于 ODE-net 的时间序列模型<sup>[49,127]</sup> 构建 batch 中所有时间点的有序并集，并对并集中的所有时间点求解所有常微分方程的解。实际应用中，在构造的时间点序列并集上求解 ODE 网络是非常耗时的。

在本章中，我们对原始的常微分方程网络进行了重新参数化，以加快训练阶段批量化求解多个 ODE 的速度。根据本章 5.6.8 节中的实验表明，本章提出的批常微分方程并行求解方法在时间效率上显著优于构建时间点并集的方法。

不失一般性地，我们假设求解批常微分方程的时间范围为从初始时间点  $\mathbf{T} = [t_1 \dots, t_B] \in R^B$  到终止时间点  $\mathbf{T}' = [T'_1 \dots, T'_B] \in R^B$ ，其中  $B$  是批大小。在初始时间点  $\mathbf{T}$  的批初始状态表示为  $\mathbf{H}_T = [\mathbf{h}_{t_1}, \dots, \mathbf{h}_{t_B}]$ 。我们希望求解  $B$  个初态不同、时间区间不同的常微分方程，并对结果进行拼接以得到批终态  $\mathbf{H}_{T'}$ 。

$$\mathbf{H}_{T'} = \begin{bmatrix} \text{ODE Solve}(f_\theta, \mathbf{h}_{t_1}, t_1, t'_1) = \mathbf{h}_{t_1} + \int_{t_1}^{t'_1} f_\theta(\mathbf{h}_t) dt \\ \dots \\ \text{ODE Solve}(f_\theta, \mathbf{h}_{t_B}, t_B, t'_B) = \mathbf{h}_{t_B} + \int_{t_B}^{t'_B} f_\theta(\mathbf{h}_t) dt \end{bmatrix} \quad (5-15)$$

为了并行化地求解  $B$  个常微分方程。我们首先给出一个定理以揭示积分区间的线性变换特性：

### 定理 1：任意定积分积分区间的线性变换等价性

对于任一标量函数  $f(\cdot)$ ，其在区间  $[a, b]$  上的定积分满足：

$$\int_a^b f(t) dt = \int_0^1 f(\tau(b-a) + a)(b-a) d\tau \quad (5-16)$$

通过替换  $\tau = \frac{t-a}{b-a}$ ，可以很容易地证明该理论成立。该理论启发我们可以通过对  $B$  个常微分方程进行再参数化，以保证其具有一致的积分区间。因此我们在原 ODE 方程的基础上构造了一个辅助常微分方程，其描述了构造状态  $\mathbf{R}(\tau)$  对于标

量时间变量  $\tau$  的导数。

$$\frac{d\mathbf{R}(\tau)}{d\tau} = g_\theta(\mathbf{R}(\tau), \tau) = f_\theta(\mathbf{R}(\tau)) \circ (T' - T) \quad (5-17)$$

其中  $\circ$  是支持广播的逐元素相乘。可以证明，对于给定的初态  $\mathbf{R}(0) = H_T$ ，辅助常微分方程在  $\tau = 1$  时刻的解等于原始  $\mathbf{B}$  个常微分方程的解的拼接：

$$H_{T'} = \mathbf{R}(1) = \text{ODE Solve}(g_\theta, \mathbf{R}(0) = H_T, 0, 1) \quad (5-18)$$

为了证明重参数化构造的初值问题 (Initial value problem, IVP) 的解等于原始 ODE 的解，可以定义两个常微分方程以及它们的解  $h(b)$  和  $v(1)$ 。

$$\begin{aligned} h(b) &= h(a) + \int_a^b f(h(t), t) dt, \\ v(1) &= v(0) + \int_0^1 g(v(\tau), \tau) d\tau. \end{aligned} \quad (5-19)$$

其中  $\alpha = b - a$ ，本章进一步假设  $g(x, \tau) = \alpha f(x, a + \tau\alpha)$  且  $v(0) = h(a)$ 。

给定任意正整数  $N$ ，可以对两个 ODE 的状态轨迹进行离散化，得到  $\{h_n | n \in 1, 2, \dots, N\}$  和  $\{v_n | n \in 1, 2, \dots, N\}$ 。

$$\begin{aligned} h_{n+1} &= h_n + f(h_n, a + \frac{n\alpha}{N}) \times \frac{\alpha}{N}, \text{ where } h_0 = h(a); \\ v_{n+1} &= v_n + g(v_n, \frac{n}{N}) \times \frac{1}{N}, \text{ where } v_0 = v(0). \end{aligned} \quad (5-20)$$

很明显可以得到  $h(b) = \lim_{N \rightarrow \infty} h_N$  和  $v(1) = \lim_{N \rightarrow \infty} v_N$ 。

进一步地，我们可以通过数学归纳法证明  $\forall i, 1 \leq i \leq N, h_i = v_i$ 。

**当  $n = 0$  时：**满足  $h_0 = v_0$ ；

**当  $n \geq 1$ ：**假设对于任意  $n \in \{1, 2, \dots, N\}$ ， $h_n = v_n$  成立，可以得到

$$\begin{aligned} v_{n+1} &= v_n + g(v_n, \frac{n}{N}) \times \frac{1}{N} \\ &= h_n + \alpha f(h_n, a + \frac{n\alpha}{N}) \frac{1}{N} \\ &= h_{n+1}, \end{aligned} \quad (5-21)$$

因此，可以证明当  $N \rightarrow \infty$  时，下式成立：

$$v(1) = v_N = h_N = h(b). \quad (5-22)$$

由此，式 (5-18) 得证。因为辅助常微分方程的积分区间为  $[0, 1]$ ，因此其求解过程非常简便。基于此，我们可以高效并行化地求解原始的多个常微分方程，且时间

消耗与批的大小无关。

## 5.6 实验验证与分析

在本节中，我们采用三个输入输出数据集对 ODE-RSSM 和其他基线模型进行评估，并根据实验结果回答如下几个问题：

- (问题 1): 相比于离散时间状态空间模型，ODE-RSSM 作为时间连续化的状态空间模型，在建模非均匀采样系统方面是否更具优势？
- (问题 2): 当被识别系统存在随机性时，ODE-RSSM 是否优于具有确定性状态转移的连续时间模型？
- (问题 3): 在训练阶段引入隐空间超调是否能改善模型的开环预测的准确度？
- (问题 4): ODE-RSSM 能否从非均匀采样数据中学习到有效的泛化，以支持采样率更为稀疏或更为稠密的预测场景。
- (问题 5): 本章提出的并行批 ODE 求解算法是否存在近似误差？求解效率如何。

### 5.6.1 数据集介绍

本节采用的实验数据集共有三个，具体包括：

- **连续搅拌釜式反应器<sup>[35]</sup>**: 该数据集来自于一个简单的 1 输入 2 输出连续搅拌釜式反应器放热模型，其共包含 7500 个连续均匀采样的观测数据(每分钟 10 个样本)。其单个输入量为冷却剂流量，在数据集中是分段恒定的，两个输出量分别是产物的浓度和温度。相比于其他两个数据集，CSTR 数据集的随机性较弱。一般确定性模型即可获得较好的系统辨识效果。
- **工业绕组数据集<sup>[35]</sup>**: 该数据采集于工业绕组过程，同样为公有数据集，共包含 2500 条均匀采样测量数据。系统共包括 5 个输入，其中包括 3 个卷轴的角速度，以及卷轴与放轴电机的设定值电流。系统输出的维度为 2，对应着不同卷轴之间连接网的拉力。
- **膏体浓密机泥层压力变化数据集**: 本章使用第三章所述的膏体浓密机数据集作为实验对象。具体地，定义系统输入为进料浓度、进料流量、出料浓度、出料流量，系统输出为浓密机内泥层压力。由于浓密机系统运行机理复杂，内部状态难以监测，因此其属于典型的不完全观测系统。同时，由于系统控制反馈的时滞极长，系统输入对系统输出的影响是长期的，因此该数据集对于模型的长时预测能力以及长依赖关系建模能力提出了更高的要求。

三个数据集的特性总结如下：

数据集	输入	输出	随机性	时延	M	L
连续搅拌釜式反应器	1	2	+	+	60	180
工业绕组数据集	5	2	+++	++	60	180
浓密机数据集	4	1	++	+++	30	60

表 5-2 数据集特性

### 5.6.2 实验设定

对于每个数据集，将其分成三个部分，分别用于训练(前 60%)、验证(中间 20%)和测试(最后的 20%)。对于训练集，引入一个固定大小的滑动窗口用于生成批量的训练数据。生成的每一批数据均为大小为  $B * N * K$  的张量，其中  $B$  是批大小， $N$  表示序列的长度， $K$  是系统输入和系统输出的维数之和。

在训练阶段，如式(5-11)所示，将长度为  $N$  的序列完全输入到模型中，并优化模型参数以最大化 ELBO。在模型验证和测试阶段，我们将每个长为  $N$  的序列分为两部分：编码序列(长度为  $M$ )和生成序列(长度为  $L$ )。不同数据集下  $M$  和  $L$  的设定如表 5-2 所示。对于编码序列，系统输入和输出序列数据被送入编码器，以推断序列第  $M$  个位置的隐状态  $s_{t_M}$ 。接下来，将推断出的状态  $s_{t_M}$  作为生成过程的初始隐状态输入到生成过程中并进行蒙特卡洛预测，通过序列采样可以得到  $n_{traj} = 128$  条长度为  $L$  的预测系统输出。根据真实的系统输出序列，我们求解  $n_{traj} = 128$  条生成序列的预测误差的均值，以评估模型的预测精度。本章采用 RMSE 和 RRSE<sup>[35]</sup> 作为误差评估指标。

对于所有数据集，相邻采样点之间的时间差  $t_{i+1} - t_i$  统一定义为 0.1。为了构造非均匀采样数据集，本章对每个原始序列随机抽取 25%，50% 的数据点以构造时间间隔非均匀的采样序列。同时，作为对照，我们又以均匀采样的方式，分别下采样 25%、50% 数据点以生成两个均匀采样数据集。

### 5.6.3 基线模型选择

由于 ODE-RSSM 是一个带有随机隐变量的连续时间状态空间模型，因此我们选择两种模型作为对照基线：

- 基于变分自编码机的离散时间状态空间模型：包括 VAE-RNN、SRNN、STORN 和 RSSM。
- 带有确定性隐变量转移的连续时间模型：ODE-RNN 和 Time-Aware RNN.

对于具有随机状态转换的 ODE-RSSM 模型和其他离散时间模型，本章从生成模型中并行采样  $n_{traj} = 128$  条轨迹，并估计  $\bar{y}_{t_i}$  的均值和  $\hat{\sigma}_{t_i}$  的标准差。并对所有轨迹计算 RRSE 和 RMSE 的均值。对于带有确定性隐状态转移的连续时间模型——ODE-RNN 和 Time-Aware RNN，将其输出投影层的输出替换为某一高斯分布均值和标准差，类似于式(5-7)的形式。为了评估离散时间模型在非均匀采样数据上的

识别效果，我们在离散时间模型的输入变量  $u_{t_i}$  中额外添加了时间差分  $t_{i+1} - t_i$ 。所有的模型均采用 adam 优化器训练的，学习率是 5e-4。当验证集损失在连续 100 个训练轮次内不下降时，训练停止。

### 5.6.4 相比于离散时间模型，ODE-RSSM 建模非均匀采样系统的效 果探究

表 5-3 连续搅拌釜式反应器

	25% (uneven)		25% (even)		50% (uneven)		50% (even)		100%	
	RRSE	RMSE	RRSE	RMSE	RRSE	RMSE	RRSE	RMSE	RRSE	RMSE
VAE-RNN	0.3118	0.2224	0.203	0.1407	0.1977	0.1372	0.1791	0.1238	0.1607	0.1099
STORN	0.3198	0.2235	0.1993	0.1379	0.2372	0.1664	0.1628	0.1115	0.155	0.1054
RSSM	0.3155	0.2158	0.1457	0.1026	0.2017	0.1447	0.0811	0.0594	0.0684	0.0499
RSSM-O	0.3114	0.2127	0.1507	0.1103	0.151	0.11	0.0872	0.0648	0.0797	0.0596
ODE-RNN	0.2627	0.1791	0.14	0.0996	0.1466	0.1047	0.0784	0.057	0.0668	0.0489
Time-Aware	0.3134	0.2138	0.1581	0.1131	0.1965	0.1411	0.136	0.0991	0.108	0.0786
ODE-RSSM	0.2979	0.1987	0.1376	0.0975	0.1486	0.1059	0.0798	0.0593	0.0739	0.0517
ODE-RSSM-O	0.2807	0.1913	0.1411	0.1003	0.1336	0.0956	0.0654	0.0477	0.0659	0.0474

提出的 ODE-RSSM 与离散时间模型在三个数据集上的比较如表所示。表 5-3, 5-4, 和 5-5 展示了 ODE-RSSM 与各个离散时间模型在三个数据集上的预测效果。结果表明，对于经过下采样的数据集，ODE-RSSM 的预测精度明显优于离散时间模型。特别是对于采样率为 0.25 和 0.5 的非均匀采样数据集 (unenven)，离散时间模型的预测误差显著增大。而 ODE-RSSM 模型预测精度的退化程度远小于离散时间模型。

在图 5-4 中，我们展示了 ODE-RSSM 模型和 RSSM 在 winding 数据集上的某一预测片段。我们发现，在 P3 处，ODE-RSSM 的预测效果显著优于离散时间的 RSSM，对于这样的结果，一种可能的解释是在 P3 处，相邻时间点之间的间隔相对较大。对于离散时间模型来说，模型主要依靠  $u_t$  中加入的  $t_{i+1} - t_i$  处理预测时间点非均匀的问题。而此处由于  $t_{i+1} - t_i$  过大，偏离于训练集中时间差的分布，进而导致预测效果可能退化。对于 ODE-RSSM 模型，尽管大间距  $t_{i+1} - t_i$  会导致较大的信息损失，但对于 ODE-RSSM 模型的负面影响要小于对 RSSM 的影响。

这一结果与前人研究的观点<sup>[36]</sup> 是吻合的，即基于 ODE-Net 的时间序列模型能够更好地从稀疏采样序列中提取特征。根据以上结果可以得出结论，相比于直接

表 5-4 工业绕组数据集

	25% (uneven)		25% (even)		50% (uneven)		50% (even)		100%	
	RRSE	RMSE	RRSE	RMSE	RRSE	RMSE	RRSE	RMSE	RRSE	RMSE
VAE-RNN	0.5242	0.5186	0.4276	0.4346	0.4708	0.4706	0.4521	0.4589	0.4192	0.4167
STORN	0.5282	0.5251	0.4151	0.4231	0.4799	0.4786	0.4058	0.4111	0.384	0.3807
RSSM	0.5667	0.5593	0.4155	0.4234	0.4932	0.4918	0.4013	0.4066	0.4011	0.3976
RSSM-O	0.5366	0.5313	0.4124	0.4202	0.5236	0.523	0.4098	0.415	0.3911	0.3876
ODE-RNN	0.5018	0.4953	0.402	0.4079	0.4247	0.4208	0.3852	0.3891	0.3808	0.3751
Time-Aware	0.6009	0.5961	0.4653	0.4748	0.4615	0.4613	0.431	0.4369	0.4017	0.398
ODE-RSSM	0.511	0.5042	0.4129	0.4238	0.4459	0.4442	0.3899	0.3946	0.3732	0.3688
ODE-RSSM-O	0.4934	0.4878	0.4063	0.4152	0.4209	0.4198	0.3776	0.3822	0.3448	0.3402

表 5-5 浓密机数据集

	25% (uneven)		25% (even)		50% (uneven)		50% (even)		100%	
	RRSE	RMSE	RRSE	RMSE	RRSE	RMSE	RRSE	RMSE	RRSE	RMSE
VAE-RNN	16.255	0.4696	14.471	0.4589	14.779	0.4361	13.915	0.4319	13.163	0.4065
STORN	11.336	0.3249	8.6047	0.2765	7.9986	0.2463	7.7936	0.2473	6.8105	0.2184
RSSM	1.619	0.0553	1.5165	0.0561	1.5808	0.0534	1.439	0.0532	1.5001	0.0548
RSSM-O	1.3725	0.0529	1.375	0.0517	1.4275	0.0519	1.4214	0.0518	1.4153	0.0522
ODE-RNN	1.4789	0.0507	1.2927	0.0494	1.6023	0.0566	1.3138	0.0489	1.3193	0.0487
Time-Aware	3.302	0.11	1.7698	0.0636	1.4732	0.0531	1.3882	0.0509	1.4108	0.0516
ODE-RSSM	1.5349	0.0558	1.4791	0.0549	1.4174	0.0522	1.4625	0.0544	1.5856	0.0577
ODE-RSSM-O	1.3678	0.0504	1.2947	0.0493	1.3064	0.0487	1.3082	0.0493	1.3302	0.05

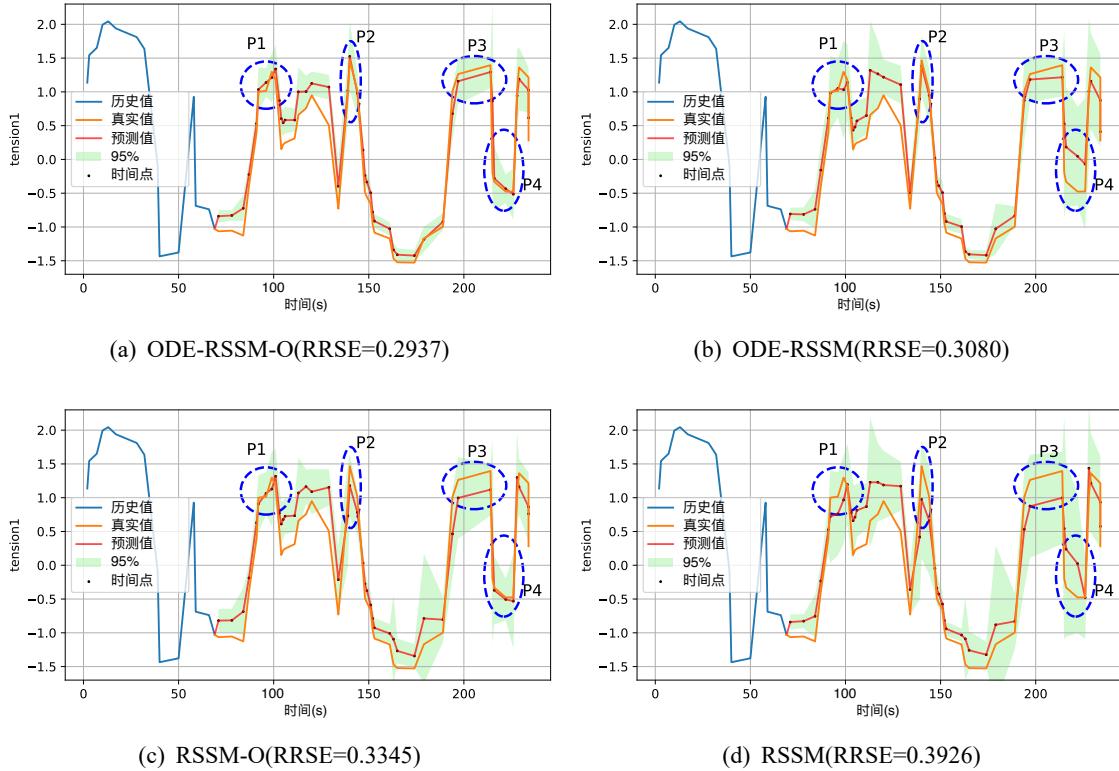


图 5-4 Winding dataset.

在输入量中添加时间的差分作为额外的输入变量，基于微分方程网络的系统辨识模型能够有效地学习系统状态在连续时域下的演化，进而有效地处理不规则时间步长。

### 5.6.5 相比于带有确定性状态演化的连续时间模型，ODE-RSSM 建模随机系统的优势探究

相比于 CSTR 数据集，Winding 数据集和 Thickening 数据集来自于随机性更强的输入输出系统，因此这两个数据集更适合于评价模型对于系统随机性的表示能力。如表 5-4 和 5-5 所示，带有随机隐变量的 ODE-RSSM 模型略优于两种连续时间确定性隐状态模型。

ODE-RSSM 本质上可以视为 ODE-RNN 模型的变种，它将瞬时状态更新过程

由 RNN 替换为 RSSM，从而使 ODE-RSSM 具有建模随机过程的能力。当被识别的系统接近于一个确定的系统，如 CSTR 数据集，表 5-3 数据说明此时采用 ODE-RSSM 的预测精度提升效果并不明显。

### 5.6.6 隐空间超调对于模型长期预测性能的影响性探究

在本节实验中，我们通过消融实验证在 RSSM 和 ODE-RSSM 中引入隐空间超调对于模型多步预测的改善效果。从表 5-3-5-5 可以看出，训练阶段添加隐空间超调能够有效改善模型的开环预测精度，且由于浓密机数据集存在更明显的系统时延于非确定性，改善程度更加明显。

在图 5-4 中，可以发现经过隐空间超调训练过的模型 (a 和 c) 在 P1 和 P4 位置的预测表现优于模型 (b 和 d)。一种可能的解释是这两个位置的采样点较为密集，利用隐空间超调技术训练过的模型能更有效地利用相邻区域的多步输入。

根据上述的预测指标结果和图形展示结果都可以看出，通过优化多步 k1 散度项训练的生成模型可以有效改善模型的开环预测能力，特别是对于具有长时延的部分可观测系统，优化效果更加明显。

### 5.6.7 ODE-RSSM 模型在不同采样间隔分布下的泛化能力探究

在本节中，我们通过实验探究使用非均匀采样数据集训练的 ODE-RSSM 模型是否对于任意采样间隔分布具有泛化能力，进而推广到与训练数据采样间隔分布不同的预测任务上。首先，我们采用绕线数据集(随机抽样比分别设置为 0.25 和 0.5，非均匀采样)分别训练 ODE-RSSM 和 RSSM 模型。为便于后文描述，将训练得到的四个模型命名为 ODE-RSSM(25%) ODE-RSSM(50%)，RSSM(25%) 和 RSSM(50%)。括号内的百分比表示训练数据集的下采样百分比。接下来，我们分别采用下采样比例为 0.25、0.5 和 1.0 三种情况的数据集对四个模型进行评估。所有的训练集和测试集均为非均匀采样。评估结果如表 5-6 所示。另外，相比于测试集与训练集下采样比率相等时的测试评估结果，我们还给出了各个下采样比率下预测误差的上升或下降情况。根据表 5-6 所示的结果可以发现，将测试集的采样率从 25% 提高到 50% 会使得预测误差明显下降。我们甚至发现当测试集的采样率为 50% 时，ODE-RSSM(25%) 的预测误差甚至低于 ODE-RSSM(50%)。对于这样的结果，一种可能的原因是：当提高测试集的下采样比例时，使用 ODE-RSSM(25%) 进行预测的本质是从稀疏的序列数据中学习，然后以更小的时间间隔预测输出，这一过程等价于预测和插值的组合。根据以往的研究<sup>[60]</sup> 表明，基于 ODE-Net 的序列模型在稀疏序列数据学习和插值方面具有明显优势。ODE-RSSM(25%) 比 ODE-RSSM(50%) 性能更好的另一个原因是低采样率减少了训练数据集中的噪声干扰，从而减少了模型的方差，有效避免过拟合。这一发现对于深度系统辨识的应用研究研究具有指导意义，合理地对训练数据进行下采样，可以更好地提升模型的泛化能力。

当测试数据集的下采样比例低于训练数据集的下采样比时，例如，用 25% 采样比的测试集评估 ODE-RSSM(50%) 和 RSSM(50%)，可以发现预测误差明显增加，ODE-RSSM(50%) 和 RSSM(50%) 误差的增加量是接近的。由于高下采样比会增加相邻采样点之间的整体间隔，因此测试时求解 ODE-Net 的时间积分范围会超过训练集中时间范围的总体分布。在这种外推预测下，ODE-RSSM 的性能难以得到保证，但其预测精度仍然优于离散时间模型 RSSM。

表 5-6 测试集采样比例与训练集比例采样比例不同时，模型预测性能对比

模型	25%(非均匀)		50%(非均匀)		100%	
	RRSE	RMSE	RRSE	RMSE	RRSE	RMSE
RSSM(25%)	0.5366 (0)	0.5313 (0)	0.4621 (-0.0745)	0.4617 (-0.0696)	0.4596 (-0.077)	0.4588 (-0.0725)
RSSM(50%)	0.6149 ( <b>0.149</b> )	0.6069 ( <b>0.1449</b> )	0.4659 (0)	0.462 (0)	0.4598 (-0.0061)	0.4583 (-0.0037)
ODE-RSSM(25%)	0.4934 (0)	0.4878 (0)	0.3976 ( <b>-0.0958</b> )	0.3962 ( <b>-0.0916</b> )	0.3861 ( <b>-0.1073</b> )	0.3837 ( <b>-0.1041</b> )
ODE-RSSM(50%)	0.5754 (0.1545)	0.5696 (0.1498)	0.4209 (0)	0.4198 (0)	0.3794 (-0.0415)	0.3771 (-0.0427)

### 5.6.8 批常微分方程并行化求解算法的时间效率探究

在本节中，我们将对批常微分方程求解算法的计算效率进行评估。图 5.5(b)展示了不同方法在求解批 ode 时的时间消耗。在进行模型训练时，假设训练序列的长度为  $N$ ，其决定了批常微分方程的批大小， $B = N \times \text{batch size}$ 。标记为“并行”的两条线代表使用式 (5-18)所示的积分区间变换方法并行求解多个常微分方程。标记为“并集”的曲线代表构建时间差分的有序并集  $dT = \{t_2 - t_1, \dots, t_N - t_{N-1}\}$ <sup>[49,127]</sup>，并按序求解有序集中所有时间点的解。

线标中的 Rk4 和 dopri5<sup>[50,125]</sup> 是求解 ODE-Net 的两个经典数值近似解算器。rk4 作为固定阶求解器时，任意两个相邻时间点之间对导数  $f_\theta$  的调用次数为固定值 4。与之相对地，dopri5 为自适应求解器，其访问导数网络的次数是自适应的，以确保解的误差在给定的误差限范围内。

我们首先发现标有“并集”的求解方法的时间消耗随长度  $N$  线性增加。特别地，“并集 +rk4”的求解方法比“并集 +dopri5”的求解方法消耗了更多的计算时间。其原因在于当有序并集合  $dT$  足够大时，相邻元素之间的距离特别小。当采用自适应的 dopri5 ODE 求解器时，相邻时间点之间导数的访问次数小于 4 次即可满足误差限要求，因此总的 ODE-Net 的平均访问次数比使用 rk4 求解器的访问次数要少。

相比于在时间点的并集上求解 ODE 方程，标记为“并行”的两种求解方法效率更高。

对于求解器组合‘并行 +dopri5’，模型必须保证满足给定的误差约束，因此随着长度  $N$  的增加，自适应算法评估 ODE-net 的次数与时间也随之增加。对于组合‘并行 +rk4’，相邻时间点之间的网络访问次数仅为 4，其耗时与序列长度  $N$  和批大小  $B$  无关，该方法提供了一种求解批常微分方程的常数时间复杂度的解决方案。在拥有充足的 GPU 显存且充分利用 CUDA 并行能力的情况下，估计多步证据下

限的实际时间消耗近似于  $\mathcal{O}(N + D)$ ，其中用于后验近似推理的复杂度为  $\mathcal{O}(N)$ ，估计 KL-D 的复杂度  $\mathcal{O}(D)$ 。

图 5.5(a)说明了当距离  $D$  线性增加时，解多步 KL 散度的时间消耗。

蓝线表示如 (5-14) 所示在估计 KL-D 中重用采样出的隐状态。如前文理论分析，通过重复使用中间采样潜伏状态，时间消耗随  $D$  近似线性增加。相比之下，式 (5-13) 所示的原始求解多步 KL 散度的方法所需时间是  $D$  的平方量级，如橙色线标记。在大多数情况下，较大的  $D$  通常会提高多步预测的效果。与原方法相比，该方法通过优化 KL-D 估计的时间复杂度，能够显著减少  $D$  较大时的训练时间。

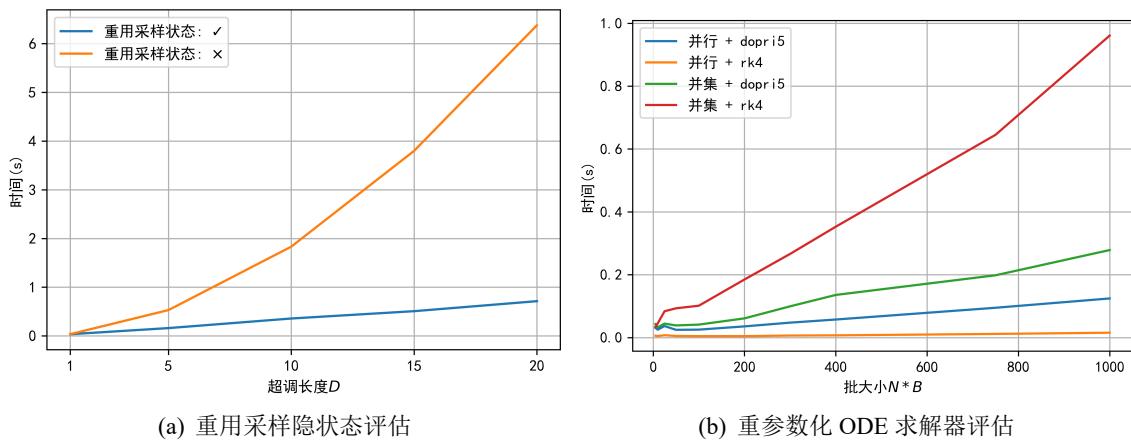


图 5-5 时间复杂度改善效果评估

## 5.7 本章小结

本章主要研究了如何利用非均匀间隔采样的序列数据识别具有随机性和长时滞特性的输入/输出系统。在标准的 RSSM 模型基础上进行扩展，本章提出了常微分方程-循环状态空间模型，该模型利用 ODE-net 建模相邻观测时间点之间确定性状态的连续时间演化。另外该模型作为状态空间模型，能够支持对流数据进行处理，进而有效解决工业传感器数据的在线推理和预测问题。为了使加入 ODE-Net 之后的模型能够支持批量并行推理与训练，本章提出了一种批量常微分方程并行求解方法，进而高效地求解多个积分区间不同的常微分方程。在训练过程中，为了改进 ODE-RSSM 中生成模型的多步预测能力，本章在模型训练阶段引入了一种高效的隐空间超调技术。相比于原始的多步 KL 散度计算方法，本章提出方法避免了重复的隐状态预测及采样，能够在更低的时间复杂度下获得多步 KL 散度的无偏估计。在实验环节，本章介绍了三个具有随机性的输入/输出系统数据集，并对原始数据集的观测时间点进行随机下采样以评估 ODE-RSSM 及其他基线模型在不规则采样数据下的预测效果。结果表明，针对具有随机性的输入输出系统，ODE-RSSM 模型在不规则数据采样下具有更好的预测准确性，在高缺失率的情况下优势更加明显。同时，基于多步隐空间超调训练的模型相比于采用原始单步变分下界训练

的模型有更好的预测精度。在讨论部分，本章分析了 ODE-RSSM 的应用领域及局限，并展望了未来可能继续深入的研究工作。

## 6 基于连续时间有模型强化学习的复杂工业系统优化与控制

### 6.1 本章引言

在现代复杂过程工业生产中，对控制性能指标进行优化是不同控制算法、控制系统的首要任务。在冶金、采矿领域等复杂过程工业场景下，浓密机是一种被广泛应用的大型沉降工具，它通过重力沉降作用可以将低浓度的固液混合物进行浓缩形成高浓度的混合物，起到减水、浓缩的作用。在对浓密机进行控制时，底流浓度是核心控制指标。该参量与其他过程监控变量如进料流量、进料浓度、出料流量、泥层高度有着复杂的耦合关系。在大部分的实际生产过程中，浓密机底流浓度的控制一般是操作员根据个人经验，通过对底流流量设定值、絮凝剂流量设定值进行调节，间接地使底流浓度追踪其工艺设定值。但是由于浓密机运行过程具有非线性、多变量、高时滞等特点，操作员难以维持底流浓度持续稳定，浓度存在偏差的底流会导致产品质量退化以及增加工业生产成本。

浓密机是一种典型的复杂过程工业设备，关于过程工业设备优化控制的研究一直是工业界、学术界研究的热点问题。对于机械结构明确、且能够精确建立动态模型的工业设备，可以采用基于模型的优化控制方法，如：实时优化控制（realtime optimization, RTO）<sup>[128]</sup>、模型预测控制（model predictive control, MPC）<sup>[129]</sup> 等。但由于浓密机系统机械结构复杂、部分变量难以观测，因此难以建立准确的数学模型近似其运转机理，导致基于模型的方法无法适用于此类复杂工业设备的控制。研究人员提出了基于数据驱动的控制方法来实现对此类无模型工业设备的控制。Dai 等<sup>[130]</sup> 提出了用于解决赤铁矿研磨系统控制问题的数据驱动优化（Date driven optimization, DDO）控制算法。Wang 等<sup>[131]</sup> 采用基于数据驱动的自适应评价方法解决连续时间未知非线性系统的无穷范围鲁棒最优控制问题。

近年来，基于强化学习<sup>[37][38]</sup> 理论的最优控制技术，也称为自适应动态规划（Adaptive dynamic programming, ADP）<sup>[132][133][134]</sup> 技术，是控制领域的研究热点话题。典型的自适应动态规划算法，如 HDP、双启发式动态规划（Dual heuristic programming, DHP）、动作依赖启发式动态规划（Action dependent heuristic dynamic programming, ADHDP）<sup>[133]</sup> 等均采用多个神经网络分别对被控系统动态模型、控制策略、策略评价模型进行建模。此类方法可以在模型未知的情况下以数据驱动的方式在线学习控制策略。Liu 等<sup>[135]</sup> 提出了一种在线自适应动态规划算法用来解决离散时间多输入多输出仿射系统控制问题，且该方法仅需要训练少量网络参数。Liu 等<sup>[136]</sup> 采用一种基于强化学习的自适应跟踪控制技术解决多输入多输出系统容错控制问题。Xu 等<sup>[137]</sup> 采用拉普拉斯特征映射算法提取被控系统全局特征，并将该全局特征用于 DHP 算法中以增强值函数网络的近似能力。

近年来，利用自适应动态规划方法解决过程工业控制问题也取得很大研究进展。Wei 等<sup>[39]</sup> 将煤炭气化过程的最优追踪控制转化为双人零和最优控制问题，并采用迭代自适应动态规划方法求解最优控制率，同时给出了收敛稳定性的分析。Jiang

等<sup>[40]</sup>利用穿插学习策略迭代(Interleaved Learning Policy Iteration, ILPL)实现了对浮选过程操作指标优化的控制,获得了比传统值函数迭代(Value iteration, VI)、策略迭代(Policy iteration, PI)算法更佳的控制效果。Jiang 等<sup>[138]</sup>将强化学习与举升方法结合(lifting technology),实现了对浮选过程设备层与操作层双速率系统的最优控制。

上述算法均使用被控系统实时生成的数据对神经网络进行训练,该训练方法忽略了系统在短期内产生的历史轨迹数据对模型学习的影响。同时,在工业场景下进行设备在线控制对算法实时性要求较高。上述方法对于控制量的计算均依托于表征控制策略的神经网络,而对于控制网络或动作网络的训练将产生较大的时间开销。为了解决上述问题,本文引入了短期经验回放技术<sup>[139][140]</sup>以对短期内的系统运行轨迹数据进行回放训练。实验证明该技术有效增强了算法收敛稳定性,且在其他ADP类在线控制算法中具有通用性。同时本文根据浓密机系统特性提出了一种迭代梯度优化算法,该算法可以在没有动作网络的情况下求解控制输入量。实验表明该方法能够在提升控制精度的同时,减少模型学习过程中产生的時間消耗。

本文主要贡献总结如下:

- 提出了一种基于ADP算法架构的启发式评价网络值迭代算法(Heuristic critic network value iteration, HCNVI)。该算法仅通过评价网络、模型网络和梯度优化算法即可求解系统最优控制输入。
- 提出了一种适用于评价网络训练的短期经验回放技术。训练评价网络时,将短期内系统运行轨迹数据共同用于模型训练,该方法可以有效增强评价网络收敛速度。
- 通过浓密机仿真实验验证了HCNVI算法的有效性。实验结果表明本文提出方法在时间消耗、控制精度上均优于其他对比方法。

本章内容组织如下:第6.2节对连续时间域下的复杂工业系统的控制过程进行形式化描述。第6.3.1节介绍了如何使用ODE-Net建模浓密机系统运行过程。第6.3.2节介绍了基于HCNVI算法以及如何利用该算法实现浓密机在线控制。第6.4节通过两组仿真实验验证本文提出控制模型的有效性。第6.5节对本章研究工作进行总结。

## 6.2 连续时间强化学习形式化描述

对于一般的复杂工业系统,其系统运行过程的控制效果往往根据某些监测指标的状态进行评价,假定其控制目标为 $y(t) \in \mathbb{R}^n$ 。该指标受控制输入、系统状态参量、及其他外部噪音扰动影响。本节将系统可控制输入量表示为 $\mathbf{u}(t) \in \mathbb{R}^m$ 。将非控制输出量表示为 $\mathbf{h}(t) \in \mathbb{R}^l$ ,该参量是表征当前系统状态的重要参量,它可被间接控制但不作为控制目标。由于在部分工业场景中,上游工序产生的物料性质是不可控的。为了使提出的控制模型具有通用性,因此本文引入噪音输入量 $\mathbf{c}(t) \in \mathbb{R}^d$

以定义那些会对生产系统产生影响的、可被监测，但无法被控制的监测量。因此，浓密机系统可表述为式 (6-26) 形式的非线性系统，其中  $f(\cdot)$  为未知非线性函数。

$$[\mathbf{y}(t_{k+1}), \mathbf{h}(t_{k+1})]^T = f(\mathbf{y}(t_k), \mathbf{u}(t_k), \mathbf{c}(t_k), \mathbf{h}(t_k), t_{k+1} - t_k) \quad (6-1)$$

定义系统状态量  $\mathbf{s}(t) = [\mathbf{y}(t), \mathbf{h}(t)]^T$ 。从连续时间域角度，动态系统可以进一步写成如下形式：

$$\dot{\mathbf{s}}(t) = \frac{d\mathbf{s}(t)}{dt} = \mathbf{f}(\mathbf{s}(t), \mathbf{u}(t)) \quad (6-2)$$

函数  $\mathbf{f}: \mathbb{R}^{n+m+l+d} \mapsto \mathbb{R}^{n+l+d}$  描述了系统当前状态与外部输入量到系统状态导数  $\dot{\mathbf{s}}(t)$  的映射。在任意实值时间  $t \in R_+$  的系统状态依赖于初始状态  $\mathbf{s}_0$  以及外部输入信号  $\mathbf{u}[0, t)$  的无穷小序列。求解过程为：

$$\mathbf{s}(t) = \mathbf{s}_0 + \int_0^t \mathbf{f}(\mathbf{s}(\tau), \mathbf{u}(\tau)) d\tau, \quad (6-3)$$

其中  $\tau$  为辅助时间变量。

对于系统运行过程式 (6-3)，本章提出的连续时间域下复杂工业系统控制算法可以根据任意时刻系统状态量  $\mathbf{s}(t)$ ，实时给出最优的控制输入量  $\mathbf{u}^*(t)$ ，使系统输出  $y(t)$  追踪其设定值  $y^*(t)$ 。另外，为了保证系统运行安全与仪器寿命，控制输入量必须满足一定的限制条件。因此，定义控制系统效用函数为：

$$U(\mathbf{s}(t), \mathbf{u}(t)) = (y(t) - y^*(t))^T Q (y(t) - y^*(t)) + (u(t) - u_{mid})^T R (u(t) - u_{mid}) \quad (6-4)$$

效用函数代表了在当前状态  $\mathbf{s}(t)$  下，执行控制输入  $\mathbf{u}(t)$  需要承受的代价。 $Q$ 、 $R$  是对称正定矩阵。 $\mathbf{u}_{mid} = \frac{\mathbf{u}_{max} + \mathbf{u}_{min}}{2}$ ， $u_{min}$ ， $u_{max}$  分别代表  $u(t)$  的上限和下限。

连续时间 (CT) 控制的目标为在满足控制约束下，最小化效用函数在无穷时间范围下的积分  $\int_t^\infty U(\mathbf{s}(\tau), \mathbf{a}(\tau)) d\tau$ 。综上本文将其转化为有约束的无穷范围折扣累积代价积分最优化问题：

$$\begin{aligned} \min_{\mathbf{u}[t, \infty)} V(\mathbf{s}(t)) &= \int_t^\infty e^{-\frac{\tau-t}{\eta}} U(\mathbf{s}(\tau), \mathbf{u}(\tau)) d\tau \\ \text{s.t. } \dot{\mathbf{s}}(t) &= \mathbf{f}(\mathbf{s}(t), \mathbf{u}(t)), \mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{imax} \end{aligned} \quad (6-5)$$

$V(\mathbf{s}(t))$  为折扣累计评价函数，用于在给定控制策略下，从长期角度评估控制策略的好坏。 $\eta$  是折扣因子。对于有约束的优化问题式 (6-5)，由于复杂系统  $\mathbf{f}$  未知，且具有非线性特性。难以利用动态规划或其他最优化方法求解最优的  $\mathbf{a}(t)$ 。因此本章提出连续时间强化学习方法 HCNVI，以解决上述最优控制问题。该方法充分利用浓密机系统运行缓慢的特性，能够在在线控制的情况下，根据系统闭环反馈数据学习最优控制策略，逐渐优化效用函数。

### 6.3 启发式评价网络值迭代算法

本章提出的启发式评价网络值迭代算法 (HCNVI) 为有模型强化学习算法。进行策略学习之前，首先需要对系统动态进行辨识。完成动态系统的学后，模型能够从在线控制过程中收集数据进行评价网络的训练。伴随着控制过程中的数据采集与参数更新，模型逐渐达到收敛，系统逐渐被控制到最优化。接下来，本节将从系统动态模型网络构建、评价网络构建、动作生成、经验回放四个方面介绍 HCNVI 算法的实现过程。

#### 6.3.1 基于常微分方程网络的系统模型构建

本节将介绍如何利用离线数据训练被控系统的预测模型。假设离线数据集共包含  $N$  组状态-控制轨迹： $\mathcal{D} = \left\{ \left( \mathbf{s}_{0:T}^{(n)}, \mathbf{a}_{0:T}^{(n)} \right) \right\}_{n=1}^N$ ，每条轨迹数据均采样于被控系统的实际生产过程。本节将利用该数据训练 ODE-Net 模型，以近似真实被控系统的运行过程。

对于观测时间点  $t_{0:T}$ ，轨迹中的第  $i$  个观测表示为  $\mathbf{s}_i := \mathbf{s}(t_i)$ 。与下一观测采样时间的间隔定义为  $\Delta t = t_{i+1} - t_i$ 。所述预测模型构建方法允许观测时间间隔是非均匀的。类似于本文的第二、三、四章，本章采用带有参数  $\theta$  的深度神经 ODE 模型  $\hat{\mathbf{f}}_\theta$  估计真实系统的状态微分，进而可以通过前向求解仿真 ODE-Net 轨迹的方式精确地重现真实系统 ODE 方程的轨迹。其学习目标为求解最优 ODE 参数  $\theta^*$  以最大化数据集的对数似然：

$$\mathcal{L}(\theta, \mathcal{D}) = \sum_{n=1}^N \sum_{i=0}^T \log \mathcal{N} \left( \mathbf{s}_i^{(n)} \mid \hat{\mathbf{s}}_i^{(n)}, \Sigma \right) \quad (6-6)$$

其中  $\Sigma$  为可训练的观测噪音协方差矩阵。通过前向仿真 ODE-Net 模型  $\hat{\mathbf{f}}_\theta(\mathbf{s}, \mathbf{u})$  获得轨迹采样  $\hat{\mathbf{s}}_{0:T}$ ：

$$\hat{\mathbf{s}}_i^{(n)} \mid \left( \mathbf{s}_0^{(n)}, \mathbf{u}_{0:t}^{(n)} \right) = \mathbf{s}_0^{(n)} + \int_0^{t_i} \hat{\mathbf{f}}_\theta \left( \hat{\mathbf{s}}_\tau^{(n)}, \mathbf{u}_\tau^{(n)} \right) d\tau \quad (6-7)$$

式 (6-7) 求解的状态轨迹完全依赖于积分时给定的初始状态  $\mathbf{s}_0$  以及连续时间过程的动作序列  $\mathbf{u}[0:t]$ 。由于数值积分过程可能在任意时间点上进行评估，因此需要保证动作信号在时间域上是连续的。本节采用带有二次平方指数核函数的核插值方法对数据集中离散的动作序列  $\mathcal{D}_a = \left\{ \mathbf{a}_{0:T}^{(n)} \right\}_{n=1}^N$  进行插值处理。

对于给定的非均匀采样控制信号  $\{(t_i, \mathbf{u}_i) | i = 1 \dots N\}$ ,  $\mathbf{t} = (t_1, t_2, \dots, t_N)^T, \mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N)^T$ , 可以估计高斯过程 (Gaussian Process, GP)。首先，基于平方指数 (Squared exponential, SE) 核函数定义控制信号的协方差矩阵  $\mathbf{K}(\mathbf{t}, \mathbf{t}) \in \mathbb{R}^{N \times N}$ , 其中：

$$K(t_p, t_q) = \exp \left( -\frac{1}{2} \frac{|t_p - t_q|^2}{\ell} \right) \quad (6-8)$$

。对于给定待插值时间点  $t_*$ , 其控制信号  $\mathbf{u}(t_*)$  的后验分布满足高斯分布:

$$\begin{aligned}\hat{\mathbf{u}}(t_*) &\sim \mathcal{N}(\mathbf{K}(t_*, \mathbf{t})^\top (K(\mathbf{t}, \mathbf{t})^{-1} + \sigma_n^2 I)^{-1} \mathbf{U}) \\ K(t_*, t_*) &= K(t_*, \mathbf{t}) K(\mathbf{t}, \mathbf{t})^{-1} K(\mathbf{t}, t_*)\end{aligned}\quad (6-9)$$

为了确保 ODE-Net 计算过程稳定, 需要保证估计的  $\hat{\mathbf{u}}(t_*)$  连续, 因此将高斯分布均值  $E[\hat{\mathbf{u}}(t_*)]$  作为式 (6-7) 中求解状态轨迹时输入的控制信号。

基于 ODE-Net 的预测模型训练过程如算法3所示。

---

### 算法 3 ODE 模型网络学习过程

---

**输入:** 给定共包含  $N$  组轨迹的原始数据集  $\mathcal{D} = \left\{ \left( \mathbf{s}_{0:T}^{(n)}, \mathbf{u}_{0:T}^{(n)} \right) \right\}_{n=1}^N$

1: **for**  $i = 1, \dots, Epochs$  **do**

2:     从  $\mathcal{D}$  中随机采样  $N_d$  个子序列, 每个时间长度为  $t_s$ :

$$\mathcal{D}' = \left\{ \left( \mathbf{s}_{t_n:t_n+t_s}^{(n)}, \mathbf{a}_{t_n:t_n+t_s}^{(n)} \right) \right\}_{n \in \Xi}, \Xi \subset \{1, \dots, N\}, |\Xi| = N_d, t_n \sim \mathcal{U}[0, T - t_s] \quad (6-10)$$

3:     对  $\mathbf{a}_{t_n:t_n+t_s}^{(n)}$  进行插值, 构建连续信号  $\mathbf{a}[t_n, t_n + t_s]^{(n)}$

4:     由式 (6-7) 求解  $\{\hat{\mathbf{s}}_{t_n:t_n+t_s}^{(n)}\}_{n \in \Xi}$

5:     由式 (6-6), 计算 ODE 网络梯度  $\nabla_\theta \mathcal{L}(\theta, \mathcal{D})$ , 采用梯度提升优化  $\theta$

6: **end for**

---

### 6.3.2 基于积分强化学习的策略评价

本节采用积分强化学习算法<sup>[141]</sup> 构建策略评价模型。对式 (6-5) 中定义的折扣积分奖赏, 可以表示为式 (6-11) 贝尔曼方程的形式:

$$\hat{V}^H(\mathbf{s}_0) = \left[ \int_0^H e^{-\frac{\tau}{\eta}} U(\hat{\mathbf{s}}(\tau), \hat{\mathbf{a}}(\tau)) d\tau + e^{-\frac{H}{\eta}} V(\hat{\mathbf{s}}_H) \right] \quad (6-11)$$

其中  $H$  为任一积分范围,  $\mathbf{s}_H$  为  $H$  时刻的终止状态。。根据贝尔曼最优原则,  $\mathbf{s}_0$  所在时刻的最优评价值函数  $V^*(s_0)$  满足连续哈密顿-雅可比-贝尔曼方程:

$$V^*(s_0) = \min_{\mathbf{u}[0:H]} \left\{ \int_0^H e^{-\frac{\tau}{\eta}} U(\hat{\mathbf{s}}(\tau), \hat{\mathbf{u}}(\tau)) d\tau + e^{-\frac{H}{\eta}} \hat{V}^*(\hat{\mathbf{s}}_H) \right\} \quad (6-12)$$

对于任意范围  $[0 : H]$ , 最优的控制输入  $\mathbf{a}^*[0 : H]$  可以表示为:

$$\mathbf{u}^*[0 : H](k) = \arg \min_{\mathbf{u}[0:H]} \left\{ \int_0^H e^{-\frac{\tau}{\eta}} U(\hat{\mathbf{s}}(\tau), \hat{\mathbf{u}}(\tau)) + e^{-\frac{H}{\eta}} \hat{V}^*(\hat{\mathbf{s}}_H) \right\} \quad (6-13)$$

通过对式 (6-13) 进行优化即可获得最优控制策略, 在式 (6-13) 等号右侧的第一项为积分奖赏项。可以通过(6-7)预测系统轨迹, 并对奖赏函数进行积分获得。第

二项为无穷积分项，无法显式求解，因此本章引入值函数近似的方法以评估未来的积分奖赏项。具体地，HCNVI 采用一个被称为评价网络的神经网络  $V_\xi(\cdot)$  来近似某一策略的评价函数。神经网络选择单隐层全连接神经网络：

具体定义如下：

$$\hat{V}_\xi(\mathbf{s}(t)) = W_{c2} \tanh(W_{c1}\mathbf{s}(t)) \quad (6-14)$$

$\tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}$  是网络的激活函数，网络输入层包含 4 个节点，隐层包含 14 个节点，输出层 1 个节点。网络参数  $\xi = [W_{c1}, W_{c2}]$  内参数均初始化为  $-1 \sim 1$  之间的随机数。该模型采用由浓密机控制过程中产生的在线数据进行网络训练。为了保证算法参数更新的实时性，本文采用单步时序差分误差（Temporal difference error, TD error）<sup>[37]</sup> 计算评价网络估计误差值。假设从浓密机系统中采集的相邻数据点为  $\{\mathbf{s}(t_k), \mathbf{u}(t_k)\}, \{\mathbf{s}(t_{k+1}), \mathbf{u}(t_{k+1})\}$ ，时序差分误差计算过程如式 (6-15) 所示：

$$e_c(\mathbf{s}(t_k)) = \hat{V}_\xi(t_k) - e^{-\frac{\Delta t}{\eta}} \hat{V}_\xi(t_{k+1}) + \int_{t_k}^{t_{k+1}} U(\mathbf{s}(\tau), \mathbf{u}(\tau)) d\tau \quad (6-15)$$

其中  $\Delta t = t_{k+1} - t_k$ ，网络损失函数为  $E_c(t_k) = e_c^2(\mathbf{s}(t_k))$ 。通过极小化该目标函数，可以使评价网络根据被控系统反馈的状态信号及效用值信号，实时地、增量式地逼近对于当前控制策略的评价函数。使用链式法则可以计算损失值  $E_c(t_k)$  对网络参数的梯度：

$$\begin{aligned} \frac{\partial e_c^2(\mathbf{s}(t_k))}{\partial W_{c2}} &= 2e_c(\mathbf{s}(t_k)) \tanh(W_{c1}\mathbf{s}(t_k))^T \\ \frac{\partial e_c^2(\mathbf{s}(t_k))}{\partial W_{c1}} &= 2e_c(\mathbf{s}(t_k)) [W_{c2}^T \odot (1 - \tanh^2(W_{c1}\mathbf{s}(t_k)))] \mathbf{s}(t_k)^T \end{aligned} \quad (6-16)$$

采用梯度下降算法对评价网络进行训练更新：

$$W_{ci}(t_k) = W_{ci}(t_k) - l_c \frac{\partial e_c^2(t_k)}{\partial W_{ci}(t_k)} \quad (6-17)$$

$l_c$  是学习率，由于浓密机所处环境的外界噪音是不断波动的，当外界噪音  $\mathbf{c}(t)$  改变时，网络需要根据训练数据快速收敛， $l_c$  需设定为固定值以保持学习能力。

由于不同物理量的取值差异很大，这会导致网络无法有效学习并且造成超参数设定困难。因此本文采用浓密机系统产生的离线数据中各参量的极值对所有训练数据利用式 (6-18) 进行归一化放缩。

$$\bar{z} = \frac{2(z - z_{\min})}{z_{\max} - z_{\min}} - 1 \quad (6-18)$$

### 6.3.3 基于随机梯度下降的动作生成

大部分的自适应动态规划算法，如 HDP、DHP<sup>[133]</sup> 等都通过建立动作网络计算控制输入，并利用评价网络输出值训练动作网络的参数。在本章提出的 HCNVI 方法中，以梯度下降的方式取代了动作网络，直接利用评价网络和 ODE-Net 模型网络计算控制动作。该方法可以在环境噪音改变时，使被控系统更快速地收敛，并且减少内存占用，同时省去了训练动作网络的时间开销。

给定系统状态  $\mathbf{s}(t_k)$  下，利用评价网络和模型网络计算控制动作  $\mathbf{u}(t_k)$  的过程如算法4所示。式 (6-19) 中在估计  $t_k + h$  时刻的折扣累计惩罚时，下一时刻浓密机系统所处外界噪音是未知的。不过由于真实工业环境下进料噪音都是连续变化的，很少出现突变，因此本模型采用当前时刻噪音  $\mathbf{c}(t_k)$  来充当未来一段时间的噪音  $\mathbf{c}[t_k : t_k + h]$ 。

---

#### 算法 4 利用迭代梯度下降算法计算控制动作

---

**输入:** 第  $t_k$  时刻系统状态  $\mathbf{s}(t_k) = [\mathbf{y}(t_k), \mathbf{h}(t_k), \mathbf{c}(t_k)]$

**输出:** 第  $k$  时刻的控制动作输出  $\mathbf{u}(t_k)$

- 1: 随机选取  $\mathbf{u}_0 = [v_1, v_2]^T$ ，其中  $v_1 \sim U(-1, 1), v_2 \sim U(-1, 1)$
- 2:  $c = 0$
- 3: **do**
- 4:   均匀采样  $M$  个时间点  $0 < h_1, \dots, h_M \leq H$ 。
- 5:   **for**  $\mathbf{h}_i \in \{h_1, \dots, h_M\}$  **do**
- 6:     给定  $\mathbf{u}[t_k : t_k + h_i] = \mathbf{u}_c$  下，评估  $t_k$  时刻的评价值  $\hat{V}(\mathbf{s}(t_k))_i$ 。

$$\hat{V}(\mathbf{s}(t_k))_i = \int_{t_k}^{t_k+h_i} U(\hat{\mathbf{s}}(\tau), \mathbf{u}(\tau)) + e^{-\frac{h_i}{\eta}} \hat{V}_{\xi}(\mathbf{s}(t_k + h_i)) \quad (6-19)$$

其中  $\hat{\mathbf{s}}(\tau)$  为 ODE 模型网络给出的预测轨迹， $\mathbf{u}(\tau) = \mathbf{u}_c$

- 7:   **end for**
- 8:   求取  $M$  轮采样下评价值的期望，并利用梯度下降算法对  $\mathbf{u}_c$  进行更新

$$\mathbf{u}_{c+1} = \mathbf{u}_c - \frac{l_u}{M} \sum_{i=1}^M \frac{\partial \hat{V}(\mathbf{s}(t_k))_i}{\partial \mathbf{u}_c} \quad (6-20)$$

- 9:   将  $\mathbf{u}_{c+1}$  限定在  $\Omega_{\mathbf{u}}$  的约束内

$$\mathbf{u}_{c+1} = \max([-1, -1]^T, \min([1, 1]^T, \mathbf{u}_{c+1})) \quad (6-21)$$

- 10:    $i = i + 1$
- 11:   **while**  $\|\mathbf{u}_{c+1} - \mathbf{u}_c\| > \epsilon_a$  and  $c < Na$
- 12:   反归一化，并求得  $\mathbf{u}(t_k)$

$$\mathbf{u}(t_k) = \frac{\mathbf{u}_{c+1} \odot (\mathbf{u}_{\max} - \mathbf{u}_{\min})}{2} + \mathbf{u}_{mid} \quad (6-22)$$

- 13: **return**  $\mathbf{u}(t_k)$
-

为了验证算法4的有效性，本文对式(6-19)算得的 $\mathbb{E}[\hat{V}(t_k)]$ 与 $\mathbf{u}(t_k)$ 的关系及迭代求解 $\mathbf{u}(t_k)$ 的过程进行了可视化探究。在实验一6.4.2介绍的仿真实验中挑选了三个时刻分析了 $\mathbb{E}[\hat{V}(t_k)]$ 与 $\mathbf{u}(k)$ 之间的函数关系。图6-1中的三个子图分别代表训练开始阶段、第一次系统达到稳态时、第二次系统达到稳态时的可视化结果。横纵坐标代表被归一化后的底流泵速和絮凝剂泵速，颜色深浅代表 $\mathbb{E}[\hat{V}(t_k)]$ 的大小。黄色箭头线代表利用算法4寻找最优控制输入 $\mathbf{u}(t_k)$ 的梯度下降轨迹。根据实验结果发现：在网络训练的三个阶段中，图中颜色最深的点，即 $\mathbb{E}[\hat{V}(t_k)]$ 的最小位置是唯一的，且不存在其他局部最优解。黄色箭头线能够准确地收敛至全局最优解。该结果说明由于浓密机运行过程缓慢，某一时刻的控制输入 $\mathbf{u}(t_k)$ 对下一时刻浓密机状态 $\mathbf{s}(t_k + h_i)$ 影响相对较小，且评价网络式(6-14)和效用函数式(6-4)具有连续、可微的性质，因此 $\mathbb{E}[\hat{V}(t_k)]$ 随 $\mathbf{u}(t_k)$ 变化的分布函数一般情况下为单峰函数。采用梯度下降算法可以有效地寻找到全局最优的 $\mathbf{u}^*(t_k)$ ，而不会收敛到局部最优解，进而满足式(6-13)的最小化条件，实现最优控制。

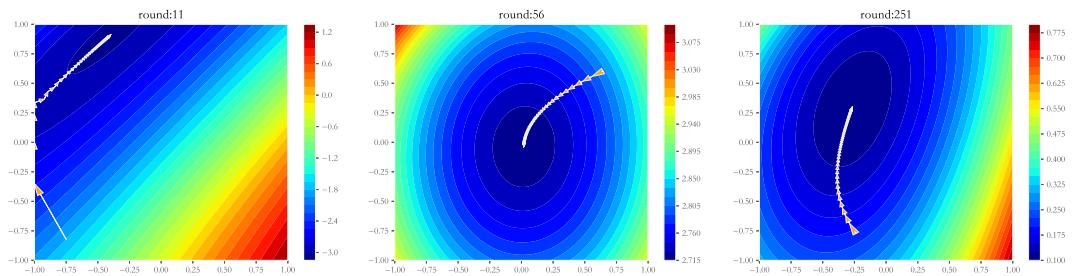


图 6-1 迭代梯度下降过程可视化

### 6.3.4 短期经验回放

为了增加评价网络训练的准确性和收敛速度，本文进一步提出短期经验回放方法优化网络训练损失函数，并计算优化梯度。短期经验回放方法将式(6-15)的误差值计算方法修改为：

$$e_c(t_k) = \frac{1}{L} \sum_{i=0}^{L-1} [\hat{V}_\xi(t_{k-i}) - e^{-\frac{\Delta t_i}{\eta}} \hat{V}_\xi(t_k) - \int_{t_{k-i-1}}^{t_{k-i}} U(\mathbf{s}(\tau), \mathbf{u}(\tau)) d\tau] \quad (6-23)$$

其中 $\Delta t_i = t_{k-i} - t_{k-i-1}$ 。在训练过程中，短期轨迹数据可以用来共同计算评价网络的损失值以及优化梯度方向。

诸如HDP、DHP以及本文提出的HCNVI算法都是面向状态值函数进行建模的在线控制算法，其策略模块的更新都是以模型网络作为媒介，计算评价网络输出值 $\hat{V}(t_k)$ 对于控制输入 $\mathbf{u}(t_k)$ 的梯度，并在此梯度基础上更新动作网络或者利用算法4优化 $\mathbf{u}(t_k)$ 。因此对于 $\mathbf{u}(k)$ 梯度估计的准确性极大地影响了策略模块的更新效果，进而影响整个控制系统的控制效果与收敛速度。 $\mathbf{u}(t_k)$ 的梯度表达式为式

(6-24)

$$\nabla \mathbf{u}(t_k) = \gamma \frac{\partial \mathbf{s}(t_{k+1})}{\partial \mathbf{u}(t_k)} \frac{\partial \hat{V}(t_{k+1})}{\partial \mathbf{s}(t_{k+1})} + \frac{\partial U(t_k)}{\partial \mathbf{u}(t_k)} \quad (6-24)$$

式中的  $\frac{\partial \hat{V}(t_{k+1})}{\partial \mathbf{s}(t_{k+1})}$  也称为  $(t_{k+1})$  时刻的协状态  $\lambda(t_{k+1})$ , 代表了评价网络输出值对于系统状态量的梯度。模型网络可以利用系统离线数据进行训练, 在训练数据量充足时可以达到极高的精度, 可以近似认为  $\frac{\partial \mathbf{s}(t_{k+1})}{\partial \mathbf{u}(t_k)}$  的估计是足够精确的。 $U(t_k)$  作为确定的效用函数,  $\frac{\partial U(t_k)}{\partial \mathbf{u}(t_k)}$  也是确定的。因此对于  $\nabla \mathbf{u}(t_k)$  的估计误差主要来源于对协状态  $\lambda(t_{k+1})$  的估计误差。

对于浓密机等大型过程工业设备来说, 系统的运行过程缓慢, 短时间内系统状态不会发生剧烈改变, 即  $\mathbf{s}(t_k) \approx \mathbf{s}(t_{k+1})$ , 且评价网络具有连续可微的性质。因此可以近似认为  $\lambda(t_k) \approx \lambda(t_{k+1})$ 。同样, 由于系统的运行过程缓慢会导致提供给控制模型学习的训练数据中系统状态参量分布非常集中, 可以近似认为式 (6-25) 成立。

$$\forall 1 \leq i < L, \|\mathbf{s}(t_{k-i}) - \mathbf{s}(t_i)\| < \delta \quad (6-25)$$

该式表明短期内系统状态点  $\mathbf{s}(t_{k-i})$  都在以  $\mathbf{s}(k)$  为圆心,  $\delta$  为半径的领域内。通过式 (6-23) 将短期  $L$  条数据共同用于评价网络训练, 可以使评价网络在  $\mathbf{s}(t_k)$  的邻域内学习地更佳充分, 进而更准确地估计  $\lambda(t_k)$ 。

为了更直观地展示增加短期经验回放对评价网络学习过程的影响, 本文对实验一6.4.2节中的评价网络进行了可视化, 实验结果如图6-2所示。该实验中采用等高线图对评价网络的输出值进行展示, 其中图6-2.(a) 代表不使用经验回放, 利用式 (6-15) 训练网络, 图6-2.(b) 代表使用短期经验回放, 回放数据点数  $L$  为 2, 利用式 (6-23) 训练网络。对于两种算法, 分别绘制了连续四次迭代中, 评价网络在更新后对不同泥层高度  $h(\cdot)$  和底流浓度  $y(\cdot)$  的评价值。图中横纵坐标分别代表被归一化的泥层高度和底流浓度。根据实验结果发现。在图6-2.(a) 中评价网络的输出值在不同输入下基本趋同。且在当前时刻系统状态点附近, 网络输出值的梯度很小。说明单数据点更新会造成评价网络很快地遗忘历史数据, 导致网络输出值整体漂移, 难以稳定地学习到正确的局部梯度。在图6-2.(b) 中, 当前系统状态  $(h(t_k), y(t_k))$  所处领域内, 网络输出值具有较大差异, 局部梯度值可以被较好地保持。准确的梯度  $\lambda(t_k)$  可以提高  $\nabla \mathbf{u}(t_k)$  估计的精确度, 因此对短期数据进行回放训练可以更好地指导控制策略输出更优控制动作, 促使评价网络和被控系统快速收敛。同时, 当经验回放数据量式 (6-23) 中  $L$  的过大, 会导致性能的退化。其原因在于本文提出的方法是同策略 (On-Policy) 强化学习方法, 而时间相差较远的历史数据点不能表征由当前控制策略产生的控制轨迹, 因此评价网络会学习到错误的评价值。另外,  $L$  过大将不再满足性质式 (6-25), 过多的历史数据回放将不再有助于评价网络学习  $\mathbf{x}(t_k)$  处的梯度值  $\lambda(t_k)$ , 进而不会提高对  $\nabla \mathbf{u}(t_k)$  估计的精确度。通过实验观察, 一般将  $L$  限定在 5 以内, 本文也将这种经验回放方法称为短期经验回放。

将 HCNVI 算法用于浓密机控制的具体流程如算法5所示。

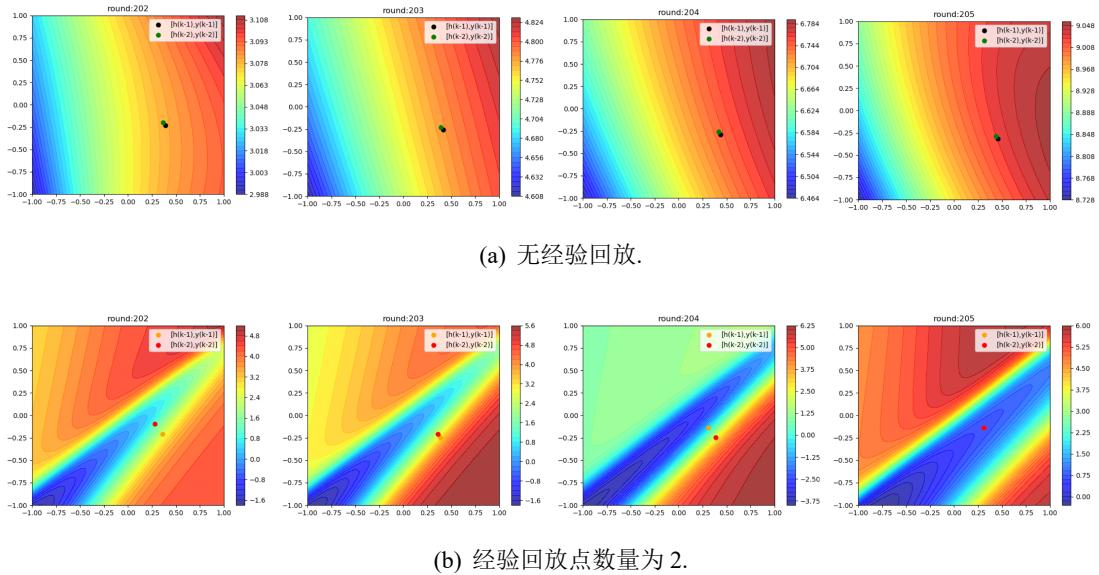


图 6-2 短期经验回放对评价网络的输出值的影响

### 算法 5 利用 HCNVI 算法实现浓密机在线控制

```

1: 使用浓密机运行离线数据, 用式 (3) 训练模型网络
2:  $k = 0$ 
3: while  $k < T$  do
4:   在新的采样时间点  $t_k$  获得浓密机系统观测值  $y(t_k), h(t_k), \mathbf{c}(t_k)$ 
5:   do
6:     令  $L = \min(L_c, k)$ , 用式 (6-23) 求解  $e_c(k)$ 
7:     利用式 (6-17) 训练评价网络
8:      $i = i + 1$ 
9:   while  $i < N_c$  and  $e_c(k)^2 > \epsilon_c$ 
10:    利用算法4求解  $\mathbf{u}(k)$ 
11:    将  $\mathbf{u}(t_k)$  作用于浓密机系统, 并等待下一次数据采样结果。
12:     $k = k + 1$ 
13: end while

```

## 6.4 本文方法在仿真工业浓密机系统底流浓度控制中的应用

当前, 工业场景下控制浓密机的方法主要依靠操作员手工控制。操作员根据生产经验给出絮凝剂添加量的设定值 ( $m^3/h$ ) 以及底流流量设定值 ( $m^3/h$ ), 浓密机内相配套的回路控制系统会根据设定值的大小自动调节絮凝剂泵速 (HZ) 与底流泵速 (HZ), 使絮凝剂的实时流量、底流实时流量追踪操作员给出的设定值。然而, 由于浓密机系统的复杂性, 操作员难以实时、完整地掌握系统运行参数, 因此无法及时、准确地设定目标点位。这导致在实际生产过程中, 浓密机常常处于非最优工作状态, 底流浓度大范围频繁波动, 偏离理想的底流浓度。

对于浓密过程式 (6-26), 控制系统的首要目标是使底流浓度  $y(t)$ , 追踪其设定值  $y^*(t)$ 。另外, 为了保证系统运行安全与仪器寿命, 控制输入必须满足一定的限制条件。

### 6.4.1 浓密机仿真模型

#### 浓密机仿真模型.

对于浓密沉降控制过程的性能进行评价，其核心控制指标为底流浓度  $y$ 。该因素受控制输入、系统状态参量、及其他外部噪音扰动影响。控制输入包括底流泵转速  $u_1(k)$  以及絮凝剂泵转速  $u_2(k)$ ，系统状态参量为泥层高度  $h(k)$ ，外部噪音输入为进料流量  $c_1(k)$ 、进料浓度  $c_2(k)$ 。由于在部分工业场景中，上游工序产生的物料浓度、物料流量是不可控的。为了使提出的浓密机控制模型具有通用性，因此本文将进料状态作为噪音输入量。浓密机进料颗粒大小，进料成分都会对浓密机底流浓度产生影响。不过由于此类变量无法观测且波动较小，为了简化问题，本文假定其保持恒定。根据上述定义，其中  $\mathbf{u}(t_k) = [u_1(t_k), u_2(t_k)] \in \mathbb{R}^2$  为可控制输入量， $\mathbf{c}(t_k) = [c_1(t_k), c_2(t_k)] \in \mathbb{R}^2$  为不可控但是可观测的噪音量， $h(t_k) \in \mathbf{R}$  为系统状态量，该参量是表征当前浓密机状态的重要参量，它可被间接控制但不作为控制目标。因此，浓密机系统可表述为式 (6-26) 形式的非线性系统，其中  $f(\cdot)$  为未知非线性函数。

$$[\dot{y}(t_k), \dot{h}(t_k)]^\top = f(y(k), \mathbf{u}(k), \mathbf{c}(k), h(k)) \quad (6-26)$$

由于在真实工业场景下进行浓密机控制实验成本较高，本节采用浓密机仿真模型验证本文提出控制算法的有效性，模型构建方法参考了<sup>[142][105][143][144][145][146]</sup>。该仿真模型建立在如下假设基础上：

- 进料都是球形颗粒。
- 絮凝剂在浓密机的静态混合器中作用完全。
- 流体的扩散以固液混合物形式进行。
- 忽略颗粒间相互作用、浓密机中耙机中轴的影响。

模型推导过程中出现的变量如表6-1，表6-2，表6-3所示

表 6-1 参量定义

变量	含义	量纲	初始值	补充说明
$f_i(t)$	进料泵频	Hz	40	扰动量
$f_u(t)$	底流泵频	Hz	85	控制量
$f_f(t)$	絮凝剂泵频	Hz	40	控制量
$c_i(t)$	进料浓度	$kg/m^3$	73	扰动量
$h(t)$	泥层高度	m	1.48	状态量
$c_u(t)$	底流浓度	$kg/m^3$	680	目标量

由文献<sup>[145]</sup>，可得泥层高度与泥层液固质量比之间的关系。

$$h(t) = \frac{W(t)\theta}{A\rho_s} + \frac{W(t)\theta}{A}r(t) \quad (6-27)$$

表 6-2 仿真模型常量

变量	含义	量纲	参考值
$\rho_s$	干砂密度	$kg/m^3$	4150
$\rho_e$	介质表观密度	$kg/m^3$	1803
$\mu_e$	悬浮体系的表观粘度	$Pa \cdot s$	1
$d_0$	进料颗粒直径	$m$	0.00008
$p$	平均浓度系数	无	0.5
$A$	浓密机横截面积	$m^2$	300.5
$k_s$	絮凝剂作用系数	$s/m^2$	0.157
$k_i$	压缩层浓度系数	$m^3/s$	0.0005*3600
$K_i$	进料流量与进料泵频的系数	$m^3/r$	50/3600
$K_u$	底流流量与底流泵频的系数	$m^3/r$	2/3600
$K_f$	絮凝剂流量与絮凝剂泵频的系数	$m^3/r$	0.75/3600
$\theta$	压缩时间	$s$	2300

表 6-3 部分变量计算方法

变量	含义	公式
$q_i(t)$	进料流量	$q_i(t) = K_i f_i(t)$
$q_u(t)$	底流流量	$q_u(t) = K_u f_u(t)$
$q_f(t)$	絮凝剂添加量	$q_f(t) = K_f f_f(t)$
$d(t)$	絮凝作用后的颗粒直径	$d(t) = k_s q_f(t) + d_0$
$u_t(t)$	颗粒的干涉沉降速度	$u_t(t) = \frac{d^2(t)(\rho_s - \rho_e)g}{18\mu_e}$
$u_r(t)$	底流导致的颗粒下沉速度	$u_r(t) = \frac{q_u(t)}{A}$
$c_l(t)$	泥层高度处单位体积含固量	$c_l(t) = k_i q_i(t) c_i(t)$
$c_a(t)$	泥层界面内单位体积含固量	$c_a(t) = p [c_l(t) + c_u(t)]$
$r(t)$	泥层内液固质量比	$r(t) = \rho_l \left( \frac{1}{c_a(t)} - \frac{1}{\rho_s} \right)$
$W(t)$	单位时间进入浓密机内的固体质量	$W(t) = c_i(t) q_i(t)$

根据固体守恒定律，泥层内固体质量变化量等于由进料导致泥层内固体量增加量与底流导致泥层内固体减少量的差。因此可以建立泥层内平均单位体积含固量与粒子沉降速度的关系。

$$\frac{d[c_a(t)Ah(t)]}{dt} = c_l(t)[u_t(t) + u_r(t)]A - c_u(t)u_r(t)A \quad (6-28)$$

对式 (6-28) 做变形可得式 (6-29)

$$c_a(t) \frac{dh(t)}{dt} + h(t)p \frac{dc_u(t)}{dt} = c_l(t)[u_t(t) + u_r(t)]A - c_u(t)u_r(t)A \quad (6-29)$$

联立式 (6-29), 式 (6-27), 可得泥层高度  $h(t)$  与底流浓度  $c_u(t)$  的一阶变化率

$$\frac{dh(t)}{dt} = -\frac{W(t)\theta}{Ac_a^2(t)} * \frac{c_l(t)[u_t(t) + u_r(t)] - c_u(t)u_r(t)}{h(t) - c_a(t)\frac{W(t)\theta}{Ac_a^2(t)}} \quad (6-30)$$

$$\frac{dc_u(t)}{dt} = \frac{c_l(t)[u_t(t) + u_r(t)] - c_u(t)u_r(t)}{p(h(t) - c_a(t)\frac{W(t)\theta}{Ac_a^2(t)})} \quad (6-31)$$

在该仿真模型中，絮凝剂泵速  $f_f$  和底流泵速  $f_u$  是控制输入  $\mathbf{u} = [f_u, f_f]^T$ ，进料泵速  $f_i$  和进料浓度  $c_i$  是外部干扰量  $\mathbf{c} = [f_i, c_i]^T$ ，底流浓度  $c_u$  为控制系统追踪变量  $y = c_u$ 。理想的控制系统能够在外界干扰量  $c$  不断波动下，通过在合理范围内调节  $u$ ，驱使  $y$  追踪其设定值  $y^*$ 。根据真实生产情况对部分变量做如下定义： $\mathbf{u}_{min} = [40, 30]$ ， $\mathbf{u}_{max} = [120, 50]$ ， $y_{min} = 280$ ， $y_{max} = 1200$ ， $\mathbf{c}_{min} = [40, 30]$ ， $\mathbf{c}_{max} = [120, 50]$ ， $y^* = 680$ 。接下来本章节将基于浓密机仿真模型式(6-30)、式(6-31)，分别进行两组实验验证在两种类型噪音量  $\mathbf{c}(k)$  输入下 HCNVI 模型的控制效果，并与其他算法进行比较。

#### 6.4.2 恒定-阶跃型噪音输入下浓密机控制仿真实验

第一组实验中设置干扰量输入  $\mathbf{c}$  为恒定值，并在某一时刻为其增加阶跃突变，噪音输入量如图6-3所示。该实验用来验证控制模型能否在浓密机外在环境发生大幅度变化下，快速寻找到  $\mathbf{u}^*$ ，使被控模型达到理想收敛稳态。

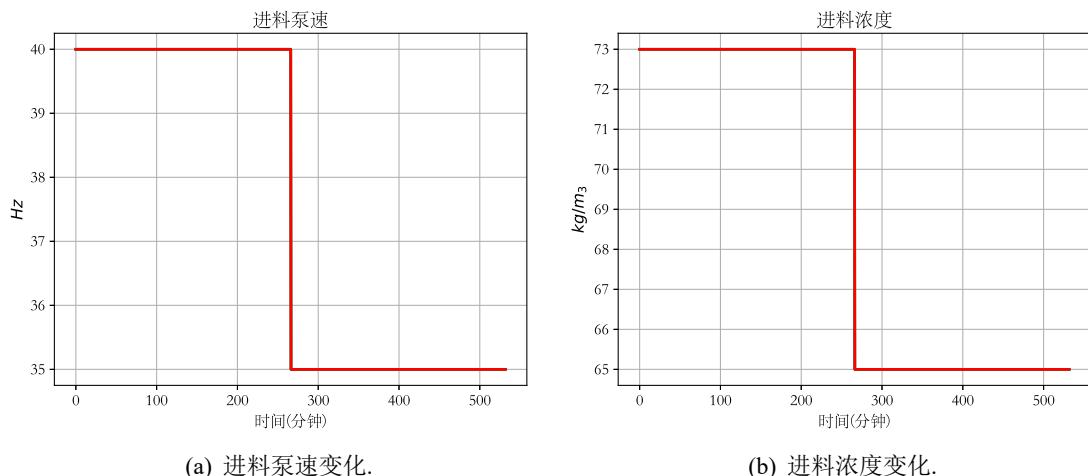


图 6-3 噪音量变化曲线

使用本文提出的 HCNVI 算法与 HDP、DHP、ILPL 算法进行对比实验。仿真实验参数如下：迭代轮次  $T = 270$ ，仿真步长  $T_d \sim \mathbb{U}(60s, 180s)$ ， $Q = 0.004$ ， $\gamma = 0.6$ ， $N_a = 4000$ ， $N_c = 500$ ， $\epsilon_c = 0.001$ ， $\epsilon_a = 0.0001$ ， $l_m = 0.01$ ， $l_c = 0.01$ ， $l_a = 0.009$ ， $l_u = 0.4$ ， $L_c = 2$ ， $\mathbf{L}_m = [0.01, 3]$ 。其中 HDP、DHP 算法也使用短期经验回放，回放点数  $L$  为 2。实验中 HDP、ILPL、HCNVI 的评价网络结构相同，且网络参数初始化为相同数值。实验结果如图6-4所示。

根据实验结果可以发现，对于不同控制算法，由于网络参数初始值均为随机设定值，训练初期底流浓度有较大幅度的波动，且在设定值两侧持续震荡。随着各个控制模型的学习，系统状态与网络参数不断趋于平稳，直到某一时刻底流浓

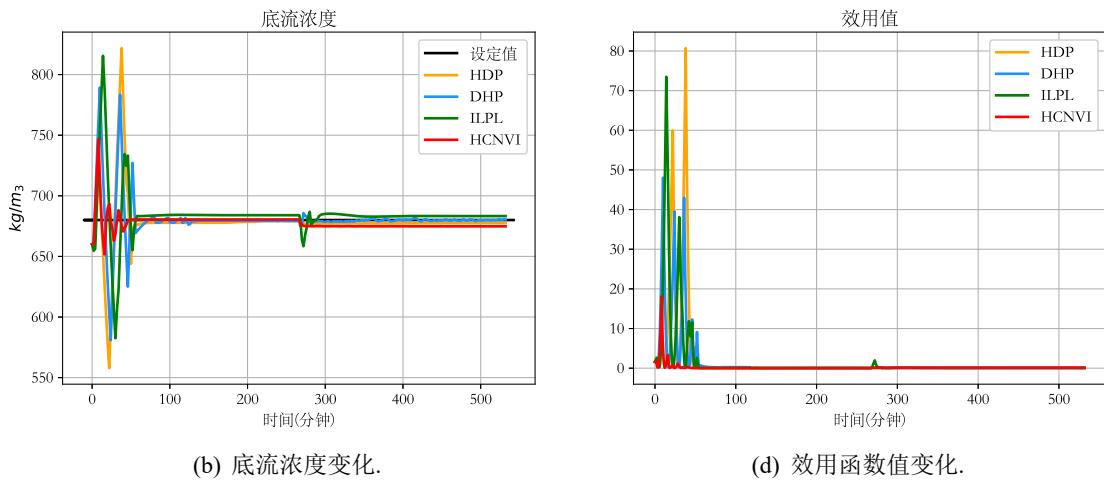


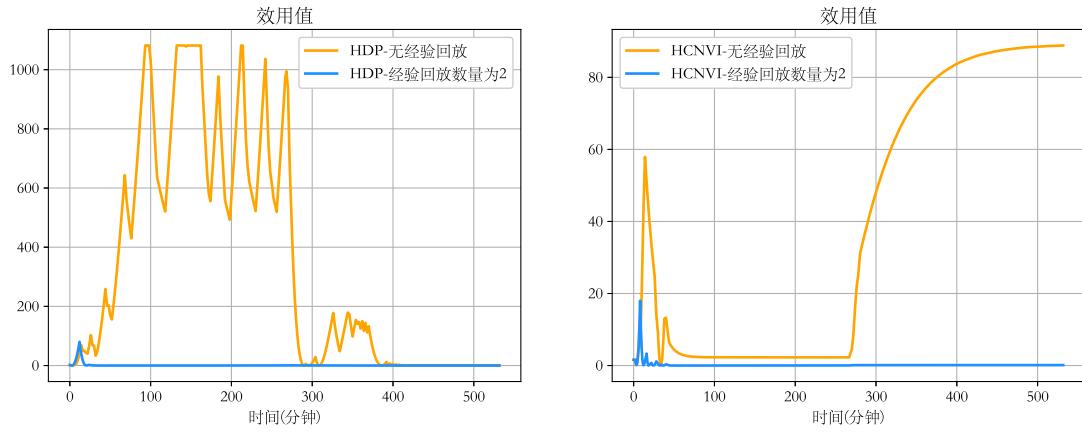
图 6-4 HCNVI 与其他 ADP 算法在恒定噪音输入下的对比

度开始稳定并与设定值重合且不再产生波动，此时控制模型参数也不再发生变化，被控系统和控制模型同时收敛到最优化。从效用值变化曲线也可以看出，早期由于底流浓度与其设定值偏差较大，效用值较高。但是随着模型与系统趋于稳态，效用值  $U(t)$  不断减小直到接近于 0 的位置。到达 270 分钟时，系统进料浓度、进料流量发生突变，底流浓度无法维持稳态，开始远离设定值。控制模型根据噪音量改变后的系统所产生的轨迹数据重新训练，将底流浓度拉回设定值位置。由于在第一阶段控制模型已经到达过一次稳态，在第二阶段仅需要少量迭代就可以使系统重归理想收敛稳态。通过观察不同控制算法产生的系统轨迹，可以发现不同控制算法到达最优化所需的时间有较大差别，且在收敛到最优化的过程中，底流浓度的波动也有较大差异。在实验第一阶段，为使系统达到稳态，HCNVI 算法所需要的迭代次数更少，训练过程中产生的底流浓度振幅也更小。并且在噪音量改变后，HCNVI 算法可以迅速地使模型重归最优化，且底流浓度几乎未发生大幅度波动。

HCNVI 的快速收敛能力主要来源于其采用迭代算法4得出的  $\mathbf{u}(k)$  严格满足式(6-13)的最小化条件，可以使评价网络更快地收敛到最优评价值函数。而其他 ADP 算法中引入了动作网络，这会使策略的更新存在一定的滞后性，进而拖慢评价网络的训练速度。

为了验证短期经验回放技术对控制算法性能的影响，本文分别对比了无经验回放、使用短期经验回放 ( $L = 2$ ) 情况下 HDP、HCNVI 的控制性能。对比结果如图6-4所示。在本实验中，仅比较了两种算法的效用值变化，效用值越快地收敛到 0 说明算法控制效果越佳。通过观察图6-4中无经验回放情况下的效用值变化曲线，可以发现曲线波动较大。相比于使用短期经验回放，无经验回放情况下控制模型需要更多的迭代轮次才能够使系统达到收敛。特别是在图6.4(a)的 HCNVI 的实验中，270 分钟时系统噪音输入量改变，效用值开始剧增，底流浓度开始偏离设定值，评价网络的学习结果如图??中的第四部分所示。评价网络对当前状态点  $\mathbf{x}(k)$  的局部梯度估计有较大偏差，使得利用算法4求解的  $\mathbf{u}(k)$  并没有驱使底流浓度向其设

定值移动，被控系统无法收敛。但在增加了短期经验数据回放后，无论是本文提出的 HCNVI 算法还是 HDP 算法，效用函数值可以快速收敛至最低点，有效实现对被控系统的控制。该实验结果表明短期经验回放技术对于控制模型的收敛速度改善效果明显，且对不同 ADP 算法具有通用型。



(a) 在 HDP 算法中引入经验回放对效用值的影响. (b) 在 HCNVI 算法中引入经验回放对效用值的影响.

图 6-4 短期经验回放对 HDP 与 HCNVI 的影响

另外本文进行了十组实验来对比 HCNVI 算法在时间上的优势。选取 HDP 算法作为参考对象， $T = 270$ ，结果如图6-5所示。由于每次实验中网络初始值不同，系统运行轨迹以及模型训练过程也不同，因此每组实验中模型学习以及控制所需的累积时间略有差异。但是从多次实验结果可以看出，由于 HCNVI 算法中去掉了动作网络，仅需要训练评价网络，所以模型整体训练时间大大缩减，尽管算法4中计算控制输入所需时间相比于 HDP 算法直接利用动作网络前向传播求解控制动作所需时间长，但是 HCNVI 算法总消耗时间明显少于 HDP 算法。

前人研究表明<sup>[147][148]</sup>，在启发式动态规划类算法中，去掉动作网络可以有效减少模型训练时间。但是在某些复杂系统控制问题中，去除动作网络会使模型难以拟合复杂策略函数，最终导致控制效果变差。在本文的实验中，由于浓密机系统运行缓慢且具有较高时滞性，当前时刻控制输入量  $\mathbf{u}(k)$  对  $\hat{\mathbf{x}}(k+1)$  的影响较小，即对  $\hat{f}(k)$  的影响较小。因此利用算法4求解的  $\mathbf{u}(k)$  满足式 (6-13) 的最小化条件。而在 HDP、DHP、ILPL 等方法中采用神经网络拟合出的控制策略，难以输出严格满足式 (6-13) 的  $\mathbf{u}(k)$ ，算法4的最优性代表 HCNVI 可以最大程度地利用评价网络给出的协状态信息优化当前控制策略，进而获得更高的控制效果。但 HCNVI 方法也具有一定的局限性，当被控系统状态变化速率较快， $\hat{f}(k)$  随  $\mathbf{u}(k)$  变化的分布函数不再是单峰函数，算法4求解出的  $\mathbf{u}(k)$  极容易陷入到局部最优解，算法控制效果及收敛速度必然变差。而此时在 HDP、DHP、ILPL 等方法中采用神经网络拟合的控制策略往往能够给出相对更优、鲁棒性更强的控制动作  $\mathbf{u}(k)$ ，其控制效果与收敛速率必然优于 HCNVI 算法。

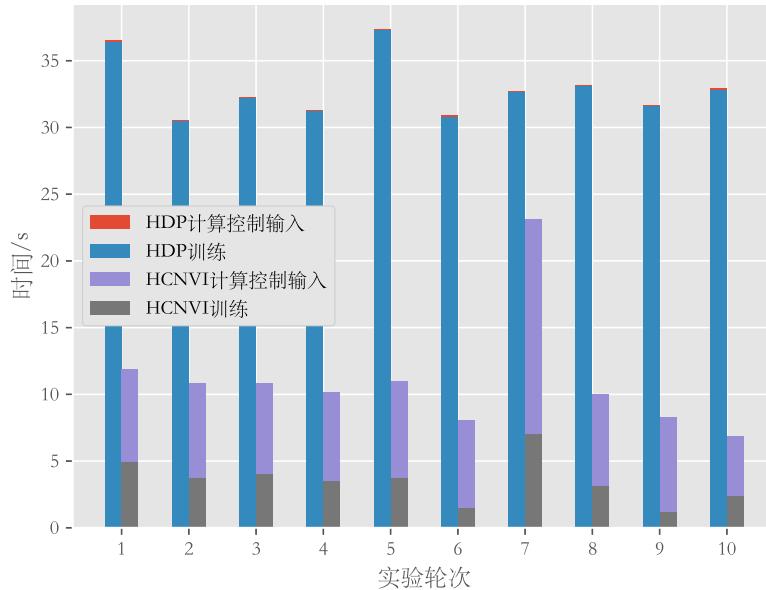


图 6-5 实验一中 HDP 与 HCNVI 在时间消耗上的对比

### 6.4.3 高斯噪音波动输入下浓密机控制仿真实验

实验一中仿真模型的进料状态是恒定的，只在某一时刻产生突变，其目的是为了更好地观察不同控制算法的收敛速度。而真实工业场景下，浓密机的进料浓度和进料流量是实时波动的。在本节实验中，进料流量和进料浓度两个噪音量持续波动，用来模仿真实工业场景下的浓密机系统环境。噪音输入的单步变化增量服从高斯分布，进料波动变化如图6-6所示。

$$\begin{aligned} \mathbf{c}(k+1) &= \mathbf{c}(k) + \Delta\mathbf{c} \\ \Delta\mathbf{c} &\sim N(\mu = 0, \Sigma = \text{diag}(0.6, 0.6)) \end{aligned} \quad (6-32)$$

本实验中 HCNVI 控制器参数与实验一6.4.2节中的算法参数相同，迭代轮次  $T = 270$ ，仿真步长  $T_d = 120s$ 。利用该仿真模型再次对比 HCNVI 与其他算法控制性能的差异，结果如图6-7所示。

通过观察实验结果发现在环境噪音连续变化条件下，浓密机底流浓度会发生持续震荡。随着对模型参数的不断训练，各个算法的控制性能趋于平稳，由于进料噪音导致的底流浓度波动稍有减弱。对比不同控制算法的控制性能，可以发现 HCNVI 相比于其他 ADP 算法能够更快地将底流浓度锁定在设定值临域范围内，且浓度振幅小于其他算法。从效用值变化曲线也可以看出，相比于其他算法，HCNVI 算法的效用值整体较小，且在训练后期几乎 0。

该实验结果与实验一6.4.2中进料噪音突变条件下的实验结果相吻合。HCNVI 算法在外界噪音频繁改变时，可以更快地响应外部变化，快速调节评价网络参数，

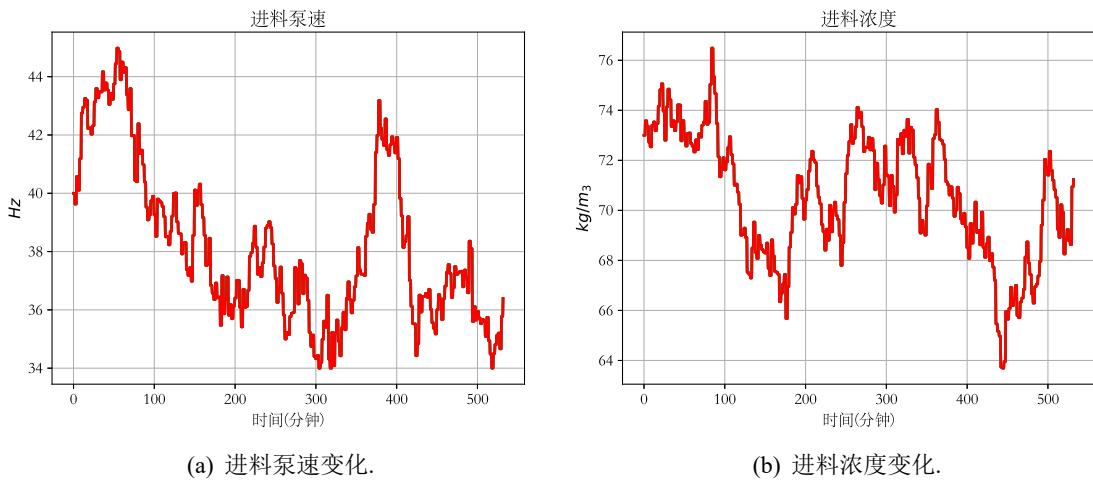


图 6-6 噪音量变化曲线

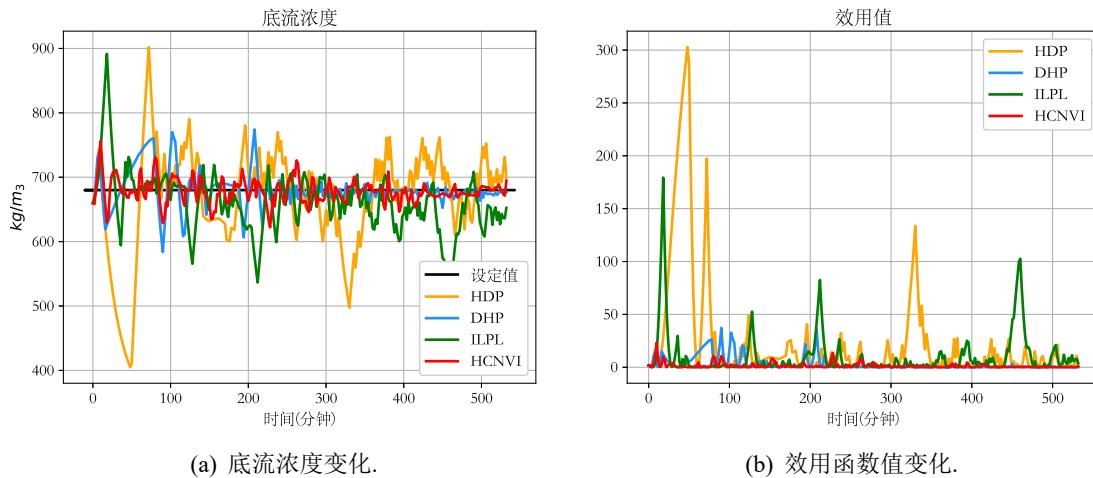


图 6-7 HCNVI 与其他 ADP 算法在波动噪声输入下的对比

将底流浓度稳定在目标值附近。其他算法由于增加了动作网络产生了训练滞后性，进而导致无法快速适应外部环境的变化，使其控制性能差于 HCNVI。

表6-4给出了不同算法在实验一6.4.2和实验二6.4.3节中底流浓度控制性能指标对比结果。相比其他算法，HCNVI 算法可以更好地控制底流浓度稳定在其设定值附近，其控制总体稳定性（由 MSE、IAE 体现）、控制鲁棒性（由 MAE 体现）更佳。在过程工业控制场景中，控制系统的 MAE 指标尤为重要，某一工序的物料性质发生剧烈波动会使下游物料加工工序出现连带波动，严重影响生产的稳定性和最终产品的质量。HCNVI 算法在 MAE 指标上的优势证实了其在过程工业控制问题中的适用性。

图6-8展示在环境噪音持续变化条件下，不使用经验回放和使用短期经验回放

<sup>1</sup>(Mean Square Error, MSE)= $\frac{1}{T} \sum_{k=1}^T |(y(k) - y^*(k))|^2$

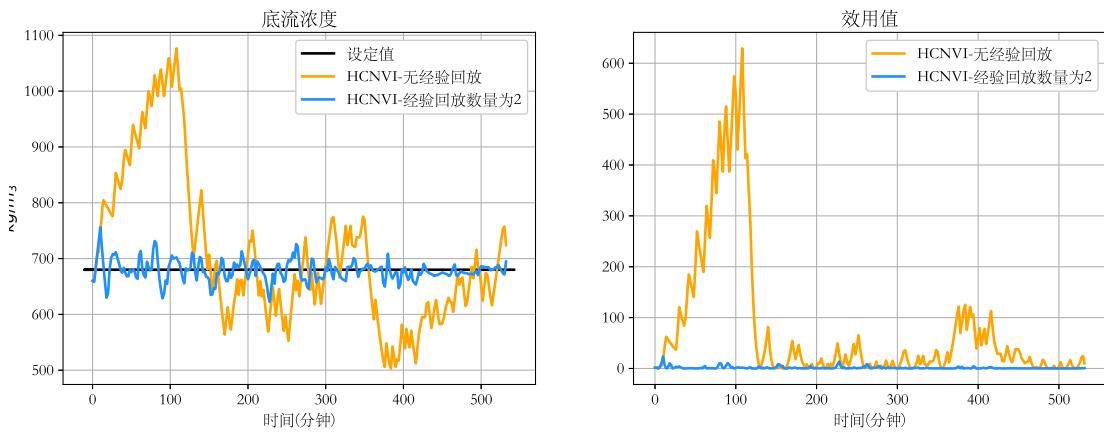
<sup>2</sup>(Max Absolute Error, MAE)= $\max_{1 \leq k \leq T} \{|y(k) - y^*(k)|\}$

<sup>3</sup>(Integral Absolute Error, IAE)= $\frac{1}{T} \sum_{k=1}^T |(y(k) - y^*(k))|$

**表 6-4 不同控制算法之间性能分析**  
**Table 6-4 Performances analysis of different algorithms**

实验组	实验一(恒定噪音)			实验二(随机噪音)		
	MSE <sup>1</sup>	MAE <sup>2</sup>	IAE <sup>3</sup>	MSE	MAE	IAE
HDP	414.182	141.854	7.246	6105.619	275.075	54.952
DHP	290.886	109.312	5.392	732.814	96.145	16.560
ILPL	364.397	135.474	8.289	2473.661	211.615	35.222
<b>HCNVI</b>	<b>44.445</b>	<b>66.604</b>	<b>3.867</b>	<b>307.618</b>	<b>76.176</b>	<b>12.998</b>

( $L = 2$ ) 两种情况下 HCNVI 算法控制性能。在无经验回放情况下，底流浓度稳定性明显较差，且效用值明显较高，使用短期经验回放 ( $L = 2$ ) 后模型控制效果较好。实验结果表明，短期经验回放技术在环境噪音持续变化下仍对模型控制效果与收敛速度有重要促进作用。



(a) 在 HCNVI 算法中引入经验回放对底流浓度的影响. (b) 在 HCNVI 算法中引入经验回放对效用值的影响.

图 6-8 噪音持续变化下短期经验回放对 HCNVI 的影响

为了展现在噪音持续变化条件下，HCNVI 算法在时间上的优势，再次重复了十次实验对比了 HCNVI 算法与 HDP 算法的时间消耗， $T = 270$ 。实验结果如图6-9所示。在噪音持续变化环境下，HCNVI 算法和 HDP 算法的总时间消耗相比于图6-5中的结果均有增加。这是由于当外部环境存在持续扰动时，被控系统和控制模型参数不再如实验一6.4.2节中达到稳定态，而是始终处于震荡状态，被控系统轨迹数据不断变化。每轮学习过程中，为了满足评价网络的精度  $e_c(k)^2 < \epsilon_c$  所需要的训练迭代次数增加，进而导致评价网络训练所需时间及模型总体训练时间增加。但通过横向对比 HCNVI 算法与 HDP 算法的总时间消耗，HCNVI 算法在训练和执行控制过程中所需的总时间消耗仍明显少于 HDP，说明利用算法4替代动作网络所产生的时间消耗削减在噪音连续波动条件仍十分明显。

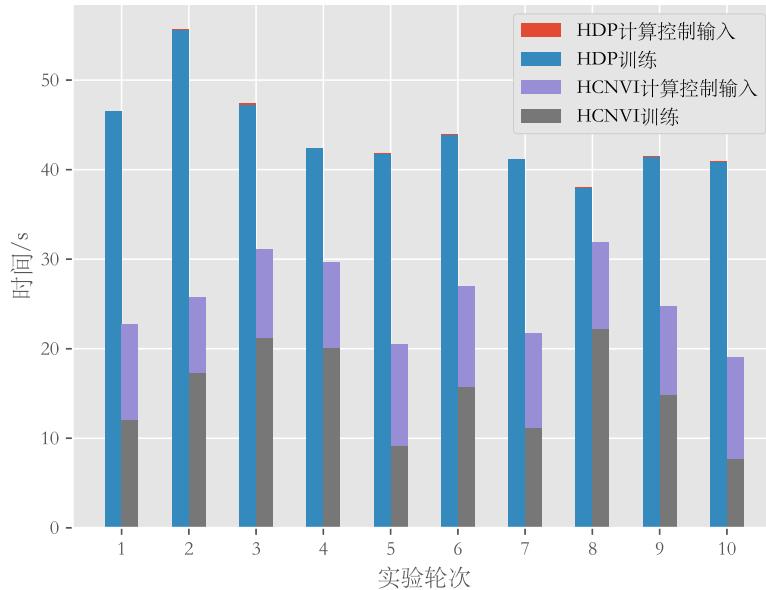


图 6-9 实验二中 HCNVI 算法与 HDP 算法在时间消耗上的对比

#### 6.4.4 真实工业场景下浓密机控制算法性能评估

本节测试了本章提出的控制算法 HCNVI 在真实膏体浓密机底流浓度控制场景下的控制精度，实验对象为第3.4.1节所述的膏体浓密机。具体地，我们分别采用 HCNVI 算法以及系统自带规则控制算法对浓密机底流浓度进行控制。每种算法下进行三组实验，每次实验过程持续 1 小时，对于每组结果，我们统计了五个指标量以评估不同算法的追踪效果以及浓度稳定效果，三组实验的实验结果以及平均值、最大值如表表6-5所示：其中  $\bar{c}_f - c_s$  为底流浓度实际值的平均值与底流浓度设定值

表 6-5 真实工业场景下浓密机控制算法性能评估

原始规则控制算法下的底流浓度控制指标					
组数	1	2	3	平均	最大
标准差	0.542	0.612	0.622	0.592	0.622
MAE	0.466	0.471	0.527	0.488	0.527
一阶差分标准差	0.378	0.445	0.439	0.421	0.445
$\bar{c}_f - c_s$	0.283	0.121	0.291	0.232	0.291
HCNVI 智能控制算法的底流浓度控制指标					
组数	1	2	3	平均	最大
标准差	0.401	0.497	0.586	0.494	0.586
MAE	0.309	0.463	0.507	0.426	0.507
一阶差分标准差	0.333	0.297	0.447	0.359	0.447
$\bar{c}_f - c_s$	0.155	0.29	0.158	0.201	0.29

的差。标准差和一阶差分标准差能够评估底流浓度的波动性，MAE 的大小能够体现底流浓度的追踪效果从结果来看，HCNVI 控制算法相比规则控制，无论在控制稳定性还是追踪效果均有明显的提升。

另外，我们采集了相同追踪设定值下 HCNVI 算法和规则控制算法下的过程监测数据，计算序列的各阶差分并绘制成折线图，结果如图6-9所示：从图中可以看

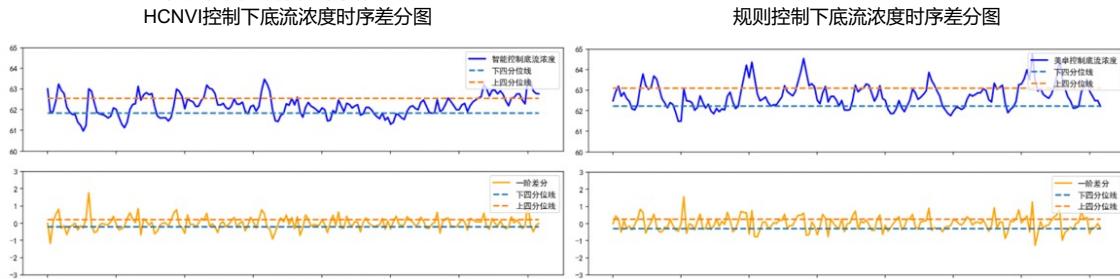


图 6-9 底流浓度控制差分对比图

到，HCNVI 控制下的底流浓度的上四分位点 (75%) 与下四分位点 (75%) 之间的间距明显小于规则控制，说明整体追踪控制精度更佳。对于底流浓度的一阶差分序列，HCNVI 控制下的四分位距为 0.4142，略小于规则控制的 0.5331，本章方法下底流浓度的控制稳定性更优。

与此同时，我们将图6-9所示的底流浓度数据进行时间序列分解，比较趋势项、季节项、噪音项的四分位距受控制算法的影响，对比结果如图6-10所示。从图中可以看到 HCNVI 控制算法下，季节项和残差项序列的波动较小，四分位数略小于规则控制结果，表表6-6对比了两种控制算法季节项和残差项的四分位距。结果说明

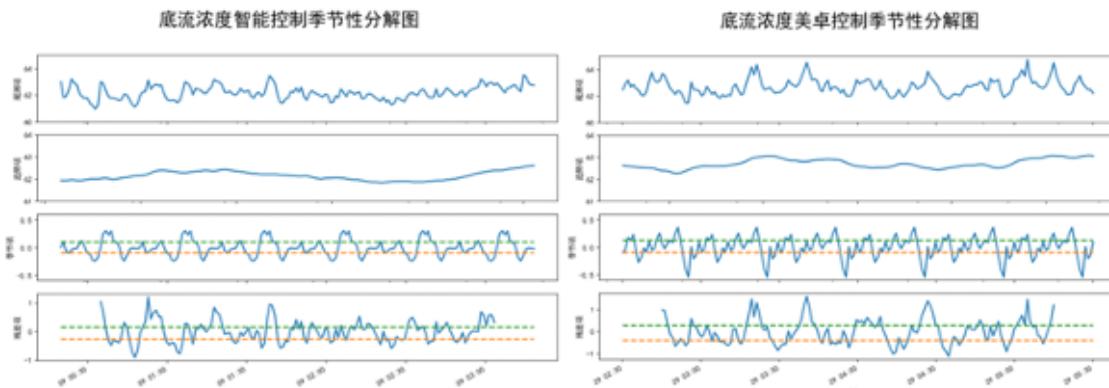


图 6-10 底流浓度序列分解对比图

系统在 HCNVI 算法的控制下的底流浓度能够整体保持稳定，且波动更小。

表 6-6 Add caption

自研智能控制算法	美卓控制算法
季节项四分位线距离 0.2007	季节项四分位距 0.2165
残差项四分位线距离 0.4267	残差项四分位距 0.6836

另外，为了进一步说明该控制算法对于工业生产的指导价值，我们测试了控制算法在使用前后的对比结果。如图6-11所示：

底流浓度智能控制效果图

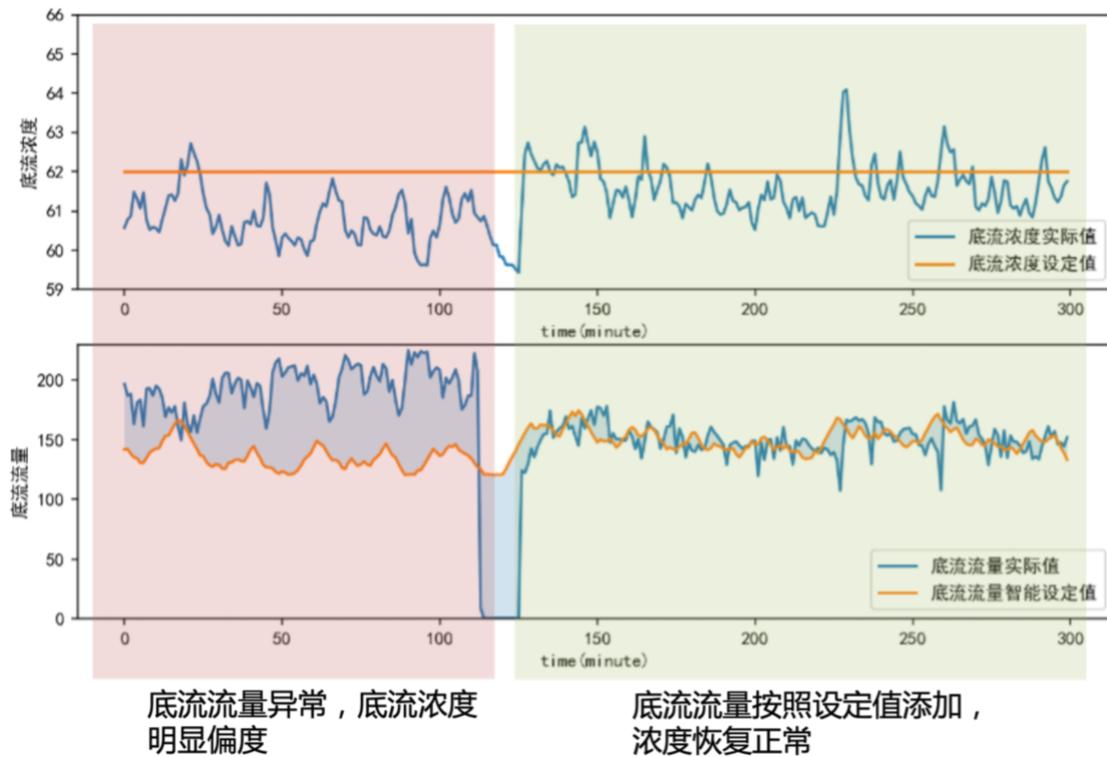


图 6-11 底流浓度序列分解对比图

在红色区域内，底流流量明显高于设定值，造成了浓密机屯料过少，底流浓度远低于预期值（62%）。在绿色区域内，底流流量图中的蓝色与橙色线较为接近，说明启用了控制算法给出的流量设定值。此时底流浓度不断升高，到达设定点附近震荡。该实验充分验证了本章控制算法对于底流浓度的追踪控制是切实有效。

## 6.5 本章小结

本文提出了基于强化学习的自适应控制算法 HCNVI，该算法通过构建用于识别系统动态方程的模型网络以及用于估计折扣累计代价的评价网络来解决浓密机控制问题。该方法可以在对浓密机系统未知的情况下，仅利用浓密机系统输出数据以及历史运行数据即可实现在线学习并获得较好的控制效果。另外本文提出的短期经验回放技术可以很好地增强评价网络训练的稳定性，在其他自适应动态规划算法中也具有较好通用性。根据仿真实验验证结果可以发现，相比其他在线 ADP 算法，由于 HCNVI 算法模型结构简单，且具有较高的学习敏捷性，因此在浓密机仿真系统控制问题中，HCNVI 算法消耗了更少的训练时间但获得了更优的控制效果。但是 HCNVI 算法也存在自身的局限性，其去掉动作网络的可行性是建立浓密机具有运行缓慢、稳定的特性基础之上的。但是当被控系统相对复杂且不再具有此特性时，如系统状态量变化过程并不连续或系统运行速度较快，HCNVI 依靠迭代算法求解的控制量难以保持最优性，控制性能极有可能产生退化。如何使 HCNVI

算法以及其他无动作网络类自适应动态规划类算法适用于此类复杂被控系统，在优化训练时间消耗的同时保证其控制性能与收敛速度，将是未来非常有意义的研究方向。

## 7 总结

### 7.1 本文研究工作总结

本文围绕基于微分方程网络的动态系统建模及预测技术开展研究，针对复杂动态系统存在的非线性、长时延、多阶段周期转换等特性，探索了采用微分方程网络建模及预测系统动态的基础理论及关键技术。

针对复杂动态系统存在的非线性、长时延特性，本文提出了基于可微 ODE-Net 的高时延工业多输入输出系统预测模型，该模型由顺序编码器、状态解码器、并行样条插值模块和导数模块组成，能够从连续时间域角度拟合复杂系统的动态过程。模型在长期预测和短期预测中均表现出了较好的性能。利用膏体浓密机系统运行数据集对本文所述模型进行评估，可以发现模型较好地克服了系统的长时延特性，在长期预测和短期预测场景中均表现出了较好的预测精度。

针对具有周期多阶段转换特性的复杂动态系统建模问题，本文提出了一种基于 ODE-Net 的自跳跃-常微分方程网络（AJ-ODEnets），同时采用该模型构建了用于长序列开环预测的编码器-解码器结构。实验环节使用具有周期多阶段特性的制冷系统运行数据对所提出的 AJ-ODEnets 及预测框架进行评估，验证了该模型对于阶段转换点以及系统输出量预测的有效性，功耗预测相对误差小于 5%。同时，该章基于预测框架能够实现对于系统制冷能耗的仿真，对于不同热负载给出了最优的制冷系启动温度配置，有效减少制冷能耗 6-25%。

针对具有随机非确定性的复杂动态系统建模问题，本文提出了常微分方程-循环状态空间模型，该模型能够利用非均匀间隔采样的序列数据识别具有随机非确定性和长时滞特性的输入输出系统。同时，本文提出了用于并行求解批常微分方程的参数化方法，使模型能够在时间点间隔不均匀的情况下并行、批量地进行推理与训练。实验环节采用三个具有随机非确定性的输入输出系统数据集评估该模型的建模效果，结果表明所提出模型在不规则数据采样下相比基线模型具有更好的预测准确性，在高缺失率以及测试集采样间隔分布改变的情况下优势更加明显。

针对复杂工业设备建模控制难的问题，本文提出了基于有模型强化学习的自适应控制算法——启发式评价网络值迭代算法。该方法可以在被控系统方程未知的情况下，仅利用系统实时输入输出数据实现控制策略的在线学习并获得较好的控制效果。另外本文提出了短期经验回放技术以增强评价网络训练的稳定性。实验环节采用尾矿浓密机仿真模型对控制算法性能进行验证，结果表明相比其他自适应动态规划算法，所提出方法的网络收敛速度更快，在线追踪控制能力更优。且该算法成功部署应用于某矿场的浓密机控制系统中，使得底流浓度控制误差更低、稳定性更优。

## 7.2 未来工作展望

基于微分方程网络能够从系统的非均匀采样数据中提取动态系统的运行模式及运行特征，基于该特征能够实现对于系统的预测与建模。目前本论文已完成的工作均面向系统预测问题。未来计划从动态系统控制决策角度，结合微分方程网络在系统建模方面的优势，实现面向非均匀采样数据的强化学习方案。

传统的基于理论科学及第一性原理的系统模型分析方法受限于基础理论发展的局限，难以适用于运行机理极为复杂的客观系统。利用数据驱动算法对机理模型进行补充完善，可以有效提升机理模型对于不同应用场景的适应能力，一定程度上也减少了机理分析、参数估计的工作量。从另一个角度看，向数据驱动模型注入已有机理模型的先验知识，能够提升数据驱动模型的可解释性，同时降低模型的自由度，减少训练数据需求量。另外，受机理先验的指导，模型在处理外推预测时的鲁棒性也会相应提升。

微分方程作为建模客观物理世界的一种基本工具，广泛存在于动力学、热力学、流体力学等复杂动态系统中。微分方程网络作为神经网路与微分方程的结合体，能够有效地搭建数据驱动算法与机理模型之间的桥梁，为解决数据科学问题带来新的解决思路。笔者在此处给出三种微分方程与机理模型可以相互结合的方案，并给出未来研究展望。

**利用微分方程网络预测理论模型的动态参数**将理论机理模型应用于工程实践时，尽管其模型方程可以做到准确、可解释，但是模型参数往往是难以获取的，且该参数很可能随时间发生演化。利用微分方程网络的数据拟合与动态预测能力可以对机理模型中的未知参数进行建模。根据参数估计结果，再利用机理模型对目标系统进行预测分析。该方法既发挥了神经网络的可学习能力，也充分利用了机理模型的先验知识。目前，该思想已被应用于可微物理建模领域中<sup>[149]</sup>。不过，将神经网络与机理模型进行前后串联，使得网络的训练依赖于损失函数从机理模型传来的梯度流。如何克服机理模型复杂导致的参数不可导、网络难训练的问题，以及如何建模混沌系统参数是未来极有意义的研究方向。

**利用微分方程网络拟合机理模型的残差**一般情况下，利用机理模型可以拟合出被辨识过程的大致趋势，而一些超越当前科学认知的复杂因素往往会使预测结果中带有一部分无法克服的残差项。利用微分方程网络等黑盒模型对这部分“没有先验指导”的残差项进行建模是可行的。这种模式可以认为是机理模型（白盒）与神经网络模型（黑盒）的浅层结合。这种结合方式在方案实现、网络设计、参数训练等方面十分简单。不过该方法对于先验知识的使用没有自适应性，过于依赖机理模型的可靠性，对于过程机理难以准确构建、模型参数无法精确识别的复杂系统，该方案很可能退化为纯黑盒网络模型，导致理论先验知识提供的价值无法充分利用。

**利用机理模型指导微分方程网络设计**传统的微分方程网络模型多采用全连接网络结构，这种黑盒模型的参数训练结果完全由训练数据集决定，因此对数据量有

极大的要求，且在处理外推预测时难以保证模型鲁棒性。利用机理模型中刻画的不同变量之间的相互关系或者约束关系，可以指导微分方程网络的设计，如 Hamilton 系统、拉格朗日系统等。不过现有的基于机理模型指导的网络设计方法都是定制化设计的，并没有形成通用的解决范式。对于机理公式及参数不完备、变量相关关系复杂的目标系统，如何设计更具有通用性的先验知识引入方法，是实现理论模型与微分方程网络深度结合的重要研究方向。



## 参考文献

- [1] ZADEH L. On the identification problem[J]. IRE Transactions on Circuit Theory, 1956, 3(4): 277–281.
- [2] LE-YI W, WEN-XIAO Z. System identification: new paradigms, challenges, and opportunities[J]. Acta automatica sinica, 2013, 39(7): 933–942.
- [3] GEVERS M. A personal view of the development of system identification: A 30-year journey through an exciting field[J]. IEEE Control systems magazine, 2006, 26(6): 93–105.
- [4] LJUNG L. Perspectives on system identification[J]. IFAC Proceedings Volumes, 2008, 41(2): 7172–7184.
- [5] LJUNG L, HJALMARSSON H, OHLSSON H. Four encounters with system identification[J]. European Journal of Control, 2011, 17(5-6): 449–471.
- [6] LJUNG L, ANDERSSON C, TIELS K, et al. Deep learning and system identification[J/OL]. IFAC-PapersOnLine, 2020, 53(2): 1175–1181.  
<http://dx.doi.org/10.1016/j.ifacol.2020.12.1329>.
- [7] JORDAN M I, RUMELHART D E. Forward models: Supervised learning with a distal teacher[J]. Cognitive science, 1992, 16(3): 307–354.
- [8] SILVER D, SUTTON R S, MÜLLER M. Sample-based learning and search with permanent and transient memories[C] // Proceedings of the 25th international conference on Machine learning. 2008 : 968–975.
- [9] WERBOS P J. Neural networks for control and system identification[C] // Proceedings of the 28th IEEE Conference on Decision and Control,. 1989 : 260–265.
- [10] TEMENG K O, SCHNELLE P D, MCAVOY T J. Model predictive control of an industrial packed bed reactor using neural networks[J]. Journal of Process Control, 1995, 5(1): 19–27.

- [11] TAN Y, VAN CAUWENBERGHE A. Nonlinear one-step-ahead control using neural networks: control strategy and stability design[J]. *Automatica*, 1996, 32(12) : 1701 – 1706.
- [12] DELGADO A, KAMBHAMPATI C, WARWICK K. Dynamic recurrent neural network for system identification and control[J]. *IEE Proceedings-Control Theory and Applications*, 1995, 142(4) : 307 – 314.
- [13] ZAMARREÑO J M, VEGA P. State space neural network. Properties and application[J]. *Neural networks*, 1998, 11(6) : 1099 – 1112.
- [14] SUTTON R S, BARTO A G. Reinforcement learning: An introduction[M]. [S.l.] : MIT press, 2018.
- [15] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. *nature*, 2015, 521(7553) : 436 – 444.
- [16] 段艳杰, 吕宜生, 张杰, et al. 深度学习在控制领域的研究现状与展望 [J]. 自动化学报, 2016, 42(5) : 643 – 654.
- [17] SHUMWAY R H, STOFFER D S, STOFFER D S. Time series analysis and its applications : Vol 3[M]. [S.l.] : Springer, 2000.
- [18] CHEN R T, RUBANOVA Y, BETTENCOURT J, et al. Neural ordinary differential equations[C] // Proceedings of the 32nd International Conference on Neural Information Processing Systems. 2018 : 6572 – 6583.
- [19] KIDGER P. On Neural Differential Equations[D]. [S.l.] : University of Oxford, 2021.
- [20] KIDGER P, MORRILL J, FOSTER J, et al. Neural controlled differential equations for irregular time series[J]. arXiv preprint arXiv:2005.08926, 2020.
- [21] LI X, WONG T-K L, CHEN R T, et al. Scalable gradients for stochastic differential equations[C] // International Conference on Artificial Intelligence and Statistics. 2020 : 3870 – 3882.
- [22] LI Z, KOVACHKI N, AZIZZADENESHELI K, et al. Fourier neural operator for parametric partial differential equations[J]. arXiv preprint arXiv:2010.08895, 2020.
- [23] YUAN Z, HU J, WU D, et al. A dual-attention recurrent neural network method for deep cone thickener underflow concentration prediction[J]. *Sensors (Switzerland)*, 2020, 20(5) : 1 – 18.

- [24] NÚÑEZ F, LANGARICA S, DÍAZ P, et al. Neural Network-Based Model Predictive Control of a Paste Thickener over an Industrial Internet Platform[J/OL]. IEEE Transactions on Industrial Informatics, 2020, 16(4) : 2859–2867.  
<http://dx.doi.org/10.1109/TII.2019.2953275>.
- [25] WU A, RUAN Z, BÜRGER R, et al. Optimization of flocculation and settling parameters of tailings slurry by response surface methodology[J]. Minerals Engineering, 2020, 156 : 106488.
- [26] ESSIEN A E, GIANNETTI C. A Deep Learning model for Smart Manufacturing using Convolutional LSTM Neural Network Autoencoders[J]. IEEE Transactions on Industrial Informatics, 2020, 16(9) : 6069–6078.
- [27] LIU J, WUN, QIAO Y, et al. Short-Term Traffic Flow Forecasting Using Ensemble Approach Based on Deep Belief Networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2020 : 1–14.
- [28] FEI J, LIU L. Real-Time Nonlinear Model Predictive Control of Active Power Filter Using Self-Feedback Recurrent Fuzzy Neural Network Estimator[J]. IEEE Transactions on Industrial Electronics, 2021 : 1–1.
- [29] NEU D A, LAHANN J, FETTKE P. A systematic literature review on state-of-the-art deep learning methods for process prediction[J]. Artificial Intelligence Review, 2021 : 1–27.
- [30] BALENZUELA M P, WILLS A G, RENTON C, et al. Parameter estimation for Jump Markov Linear Systems[J]. Automatica, 2022, 135 : 109949.
- [31] OPPER M, SANGUINETTI G. Variational inference for Markov jump processes[J]. Advances in neural information processing systems, 2007, 20.
- [32] Deep state space models for time series FORECASTINGRANGAPURAM S S, SEEGER M, GASTHAUS J, et al. Deep state space models for time series forecasting[J]. Advances in Neural Information Processing Systems, 2018, 2018-Decem(NeurIPS) : 7785–7794.
- [33] FRACCARO M, KAMRONN S, PAQUET U, et al. A disentangled recognition and nonlinear dynamics model for unsupervised learning[J]. Advances in Neural Information Processing Systems, 2017, 2017-Decem(section 5) : 3602–3611.
- [34] NASON G P. Stationary and non-stationary time series[J]. Statistics in volcanology, 2006, 60.

- [35] DEMEESTER T. System Identification with Time-Aware Neural Sequence Models[J], 2019.
- [36] QUAGLINO A, GALLIERI M, MASCI J, et al. SNODE: Spectral Discretization of Neural ODEs for System Identification[C/OL] // 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. [S.l.] : OpenReview.net, 2020.  
<https://openreview.net/forum?id=Sye0XkBKvS>.
- [37] SUTTON R S, BARTO A G. Reinforcement learning: An introduction[M]. [S.l.] : MIT press, 2018.
- [38] LEWIS F L, VRABIE D, SYRMOS V L. Optimal control[M]. [S.l.] : John Wiley & Sons, 2012.
- [39] WEI Q, LIU D. Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification[J]. IEEE Transactions on Automation Science and Engineering, 2013, 11(4) : 1020 – 1036.
- [40] JIANG Y, FAN J, CHAI T, et al. Data-driven flotation industrial process operational optimal control based on reinforcement learning[J]. IEEE Transactions on Industrial Informatics, 2017, 14(5) : 1974 – 1989.
- [41] LIN L-J, MITCHELL T M. Memory approaches to reinforcement learning in non-Markovian domains[M]. [S.l.] : Citeseer, 1992.
- [42] DEISENROTH M, RASMUSSEN C E. PILCO: A model-based and data-efficient approach to policy search[C] // Proceedings of the 28th International Conference on machine learning (ICML-11). 2011 : 465 – 472.
- [43] FUNAHASHI K-I, NAKAMURA Y. Approximation of dynamical systems by continuous time recurrent neural networks[J]. Neural networks, 1993, 6(6) : 801 – 806.
- [44] GRATHWOHL W, CHEN R T, BETTENCOURT J, et al. Ffjord: Free-form continuous dynamics for scalable reversible generative models[J]. 7th International Conference on Learning Representations, ICLR 2019, 2019 : 1 – 13.
- [45] LING J, KURZAWSKI A, TEMPLETON J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance[J]. Journal of Fluid Mechanics, 2016, 807 : 155 – 166.

- [46] RAMADHAN A, MARSHALL J, SOUZA A, et al. Capturing missing physics in climate model parameterizations using neural differential equations[J]. arXiv preprint arXiv:2010.12559, 2020.
- [47] ZHONG Y D, DEY B, CHAKRABORTY A. Symplectic ODE-Net: Learning Hamiltonian Dynamics with Control[C] // International Conference on Learning Representations. 2019.
- [48] AYED I, de BÉZENAC E, PAJOT A, et al. Learning dynamical systems from partial observations[J]. arXiv preprint arXiv:1902.11136, 2019.
- [49] RUBANOVA Y, CHEN R T, DUVENAUD D. Latent ODEs for irregularly-sampled time series[J]. Advances in Neural Information Processing Systems, 2019, 32(NeurIPS).
- [50] YUAN Z, LI X, WU D, et al. Continuous-Time Prediction of Industrial Paste Thickener System With Differential ODE-Net[J/OL]. IEEE/CAA Journal of Automatica Sinica, 2022, 9(4) : 686–698.  
<https://ieeexplore.ieee.org/document/9732304/>.
- [51] ZHUANG J, DVORNEK N, LI X, et al. Adaptive checkpoint adjoint method for gradient estimation in neural ODE[J]. 37th International Conference on Machine Learning, ICML 2020, 2020, PartF16814 : 11575 – 11585.
- [52] FINLAY C, JACOBSEN J-H, NURBEKYAN L, et al. How to train your neural ODE[C] // International Conference on Machine Learning. 2020 : 3154 – 3164.
- [53] KELLY J, BETTENCOURT J, JOHNSON M J, et al. Learning differential equations that are easy to solve[J]. arXiv, 2020, 3.
- [54] GHOSH A, TORR P H, BEHL H S, et al. STEER: Simple Temporal Regularization For Neural ODEs[J]. arXiv, 2020(NeurIPS).
- [55] RUBANOVA Y, CHEN R T Q, DUVENAUD D. Latent ODEs for Irregularly-Sampled Time Series[M] // Proceedings of the 33rd International Conference on Neural Information Processing Systems. Red Hook, NY, USA : Curran Associates Inc., 2019.
- [56] BROUWER E D, SIMM J, ARANY A, et al. GRU-ODE-Bayes: continuous modeling of sporadically-observed time series[C] // Proceedings of the 33rd International Conference on Neural Information Processing Systems. 2019 : 7379 – 7390.

- [57] LECHNER M, HASANI R. Learning long-term dependencies in irregularly-sampled time series[J]. arXiv preprint arXiv:2006.04418, 2020.
- [58] MORRILL J, SALVI C, KIDGER P, et al. Neural rough differential equations for long time series[C]// International Conference on Machine Learning. 2021 : 7829 – 7838.
- [59] MORRILL J, KIDGER P, YANG L, et al. Neural Controlled Differential Equations for Online Prediction Tasks[J]. arXiv preprint arXiv:2106.11028, 2021.
- [60] YILDIZ Ç, HEINONEN M, LÄHDESMÄKI H. ODE2VAE: Deep generative second order ODEs with Bayesian neural networks[J]. Advances in Neural Information Processing Systems, 2019, 32(NeurIPS).
- [61] HUANG Z, SUN Y, WANG W. Learning continuous system dynamics from irregularly-sampled partial observations[J]. arXiv, 2020(NeurIPS) : 1 – 14.
- [62] VASWANI A, SHAZER N, PARMAR N, et al. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017, 2017-Decem(Nips) : 5999 – 6009.
- [63] ZHOU H, ZHANG S, PENG J, et al. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting[J], 2020.
- [64] WU H, XU J, WANG J, et al. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting[J], 2021 : 1 – 11.
- [65] CHEN Y-H, CHIEN J-T. Continuous-time attention for sequential learning[C] // Proc. of AAAI Conference on Artificial Intelligence. 2021.
- [66] LIU X, YU H F, DHILLON I S, et al. Learning to encode position for transformer with continuous dynamical model[J]. 37th International Conference on Machine Learning, ICML 2020, 2020, PartF168147-9 : 6283 – 6291.
- [67] YANG H, YIN S. Actuator and Sensor Fault Estimation for Time-Delay Markov Jump Systems With Application to Wheeled Mobile Manipulators[J/OL]. IEEE Transactions on Industrial Informatics, 2020, 16(5) : 3222 – 3232.  
<http://dx.doi.org/10.1109/TII.2019.2915668>.
- [68] FANG Y, LOPARO K A. Stabilization of continuous-time jump linear systems[J]. IEEE Transactions on Automatic Control, 2002, 47(10) : 1590 – 1603.

- [69] ASHLEY T T, ANDERSSON S B. A sequential Monte Carlo framework for the system identification of jump Markov state space models[C] //2014 American Control Conference. 2014: 1144–1149.
- [70] SVENSSON A, SCHÖN T B, LINDSTEN F. Identification of jump Markov linear models using particle filters[C] //53rd IEEE conference on decision and control. 2014: 6504–6509.
- [71] FRACCARO M, SØNDERBY S K, PAQUET U, et al. Sequential neural models with stochastic layers[J]. Advances in Neural Information Processing Systems, 2016: 2207–2215.
- [72] CHUNG J, KASTNER K, DINH L, et al. A recurrent latent variable model for sequential data[J]. Advances in Neural Information Processing Systems, 2015, 2015-Janua : 2980–2988.
- [73] KARL M, SOELCH M, BAYER J, et al. Deep Variational Bayes Filters: Unsupervised Learning of State Space Models from Raw Data[C/OL] //5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. [S.l.] : OpenReview.net, 2017.  
<https://openreview.net/forum?id=HyTqHL5xg>.
- [74] KINGMA D P, WELLING M. Auto-Encoding Variational Bayes[C/OL] //BENGIO Y, LECUN Y. 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings. 2014.  
<http://arxiv.org/abs/1312.6114>.
- [75] DOERR A, DANIEL C, SCHIEGG M, et al. Probabilistic recurrent state-space models[C] //International Conference on Machine Learning. 2018 : 1280–1289.
- [76] HAFNER D, LILLICRAP T, FISCHER I, et al. Learning latent dynamics for planning from pixels[J]. 36th International Conference on Machine Learning, ICML 2019, 2019, 2019-June : 4528–4547.
- [77] OGANESYAN V, VOLOKHOVA A, VETROV D. Stochasticity in Neural ODEs: An Empirical Study[J]. arXiv, 2020 : 1–9.
- [78] JIA J, BENSON A R. Neural jump stochastic differential equations[J]. Advances in Neural Information Processing Systems, 2019, 32(2018).
- [79] POWELL W B. Approximate Dynamic Programming: Solving the curses of dimensionality : Vol 703[M]. [S.l.] : John Wiley & Sons, 2007.

- [80] ZHANG H, LIU D, LUO Y, et al. Adaptive dynamic programming for control: algorithms and stability[M]. [S.l.]: Springer Science & Business Media, 2012.
- [81] ABU-KHALAF M, LEWIS F L. Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach[J]. *Automatica*, 2005, 41(5): 779–791.
- [82] VRABIE D, LEWIS F. Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems[J]. *Neural Networks*, 2009, 22(3): 237–246.
- [83] VAMVOUDAKIS K G, VRABIE D, LEWIS F L. Online adaptive algorithm for optimal control with integral reinforcement learning[J]. *International Journal of Robust and Nonlinear Control*, 2014, 24(17): 2686–2710.
- [84] ZHANG H, WEI Q, LUO Y. A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm[J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2008, 38(4): 937–942.
- [85] KAMALAPURKAR R, DINH H, BHASIN S, et al. Approximate optimal trajectory tracking for continuous-time nonlinear systems[J]. *Automatica*, 2015, 51: 40–48.
- [86] MODARES H, LEWIS F L. Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning[J]. *IEEE Transactions on Automatic control*, 2014, 59(11): 3051–3056.
- [87] MODARES H, LEWIS F L. Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning[J]. *Automatica*, 2014, 50(7): 1780–1792.
- [88] JIANG Y, JIANG Z-P. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics[J]. *Automatica*, 2012, 48(10): 2699–2704.
- [89] MODARES H, LEWIS F L, JIANG Z-P. Tracking control of completely unknown continuous-time systems via off-policy reinforcement learning[J]. *IEEE transactions on neural networks and learning systems*, 2015, 26(10): 2550–2562.
- [90] SILVER D, SCHRITTWIESER J, SIMONYAN K, et al. Mastering the game of go without human knowledge[J]. *nature*, 2017, 550(7676): 354–359.

- [91] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. *nature*, 2015, 518(7540) : 529 – 533.
- [92] SPIELBERG S, GOPALUNI R, LOEWEN P. Deep reinforcement learning approaches for process control[C/OL] // 2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP). 2017 : 201 – 206.  
<http://dx.doi.org/10.1109/ADCONIP.2017.7983780>.
- [93] YU R, SHI Z, HUANG C, et al. Deep reinforcement learning based optimal trajectory tracking control of autonomous underwater vehicle[C] // 2017 36th Chinese control conference (CCC). 2017 : 4958 – 4965.
- [94] KIM J W, PARK B J, YOO H, et al. Deep reinforcement learning based finite-horizon optimal tracking control for nonlinear system[J]. *IFAC-PapersOnLine*, 2018, 51(25) : 257 – 262.
- [95] KIM J W, PARK B J, YOO H, et al. A model-based deep reinforcement learning method applied to finite-horizon optimal control of nonlinear control-affine system[J]. *Journal of Process Control*, 2020, 87 : 166 – 178.
- [96] YILDIZ C, HEINONEN M, LÄHDESMÄKI H. Continuous-time Model-based Reinforcement Learning[C] // International Conference on Machine Learning. [S.I.] : PMLR, 2021 : 12009 – 12018.
- [97] AINSWORTH S, LOWREY K, THICKSTUN J, et al. Faster Policy Learning with Continuous-Time Gradients[J], 2020, 144 : 1 – 14.
- [98] LARSSON E K, SÖDERSTRÖM T. Identification of continuous-time AR processes from unevenly sampled data[J]. *Automatica*, 2002, 38(4) : 709 – 718.
- [99] LI H, HU G, LI J, et al. Intelligent Fault Diagnosis for Large-Scale Rotating Machines Using Binarized Deep Neural Networks and Random Forests[J]. *IEEE Transactions on Automation Science and Engineering*, 2021 : 1 – 11.
- [100] DEMEESTER T. System identification with time-aware neural sequence models[C] // Proceedings of the AAAI Conference on Artificial Intelligence : Vol 34. 2020 : 3757 – 3764.
- [101] BOX G E, JENKINS G M, REINSEL G C, et al. Time series analysis: forecasting and control[M]. [S.I.] : John Wiley & Sons, 2015.
- [102] WEISS R J, CHOROWSKI J, JAITLEY N, et al. Sequence-to-Sequence Models Can Directly Translate Foreign Speech[J]. *Proc. Interspeech 2017*, 2017 : 2625 – 2629.

- [103] GEVERS M. A personal view of the development of system identification: A 30-year journey through an exciting field[J]. IEEE Control Systems Magazine, 2006, 26(6): 93 – 105.
- [104] CHAI T, JIA Y, LI H, et al. An intelligent switching control for a mixed separation thickener process[J]. Control Engineering Practice, 2016, 57: 61 – 71.
- [105] KIM B, KLIMA M S. Development and application of a dynamic model for hindered-settling column separations[J]. Minerals engineering, 2004, 17(3): 403 – 410.
- [106] CHEN R T Q, RUBANOVA Y, BETTENCOURT J, et al. Neural Ordinary Differential Equations[G] // Advances in Neural Information Processing Systems 31. [S.l.]: Curran Associates, Inc., 2018: 6571 – 6583.
- [107] GUIDOLIN M, PEDIO M. Chapter 4 - Unit Roots and Cointegration[G] // GUIDOLIN M, PEDIO M. Essentials of Time Series for Financial Applications. [S.l.]: Academic Press, 2018: 113 – 149.
- [108] CHRISTOFFERSEN P F. Forecasting Non-Stationary Economic Time Series[J]. Journal of the American Statistical Association, 2001, 96(453): 347 – 347.
- [109] KINGMA D P, BA J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
- [110] BOGACKI P, SHAMPINE L F. A 3 (2) pair of Runge-Kutta formulas[J]. Applied Mathematics Letters, 1989, 2(4): 321 – 325.
- [111] WU S, XIAO X, DING Q, et al. Adversarial Sparse Transformer for Time Series Forecasting[J]. Advances in Neural Information Processing Systems, 2020, 33(NeurIPS): 17105 – 17115.
- [112] RANGAPURAM S S, SEEGER M, GASTHAUS J, et al. Deep state space models for time series forecasting[J]. Advances in Neural Information Processing Systems, 2018, 2018-Decem(NeurIPS): 7785 – 7794.
- [113] POLI M, MASSAROLI S, YAMASHITA A, et al. Hypersolvers: Toward Fast Continuous-Depth Models[J], 2020(NeurIPS).
- [114] WANG Y, ZHANG Y, NÖRTERSHÄUSER D, et al. Model and data driven transient thermal system modelings for contained data centers[J/OL]. Energy and Buildings, 2022, 258: 111790.  
<http://dx.doi.org/https://doi.org/10.1016/j.enbuild.2021.111790>.

- [115] WANG Y, HE L, JIANG S, et al. Failure Prediction of Hard Disk Drives Based on Adaptive Rao–Blackwellized Particle Filter Error Tracking Method[J/OL]. IEEE Transactions on Industrial Informatics, 2021, 17(2) : 913–921.  
<http://dx.doi.org/10.1109/TII.2020.3016121>.
- [116] JANSCH-PORTO J P, HU B, DULLERUD G. Policy Learning of MDPs with Mixed Continuous/Discrete Variables: A Case Study on Model-Free Control of Markovian Jump Systems[C] // BAYEN A M, JADBABAIE A, PAPPAS G, et al. Proceedings of Machine Learning Research, Vol 120 : Proceedings of the 2nd Conference on Learning for Dynamics and Control. [S.l.] : PMLR, 2020 : 947–957.
- [117] LI S C-X, MARLIN B. A scalable end-to-end Gaussian process adapter for irregularly sampled time series classification[C] // Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016 : 1812–1820.
- [118] SHUKLA S N, MARLIN B. Interpolation-Prediction Networks for Irregularly Sampled Time Series[C] // International Conference on Learning Representations. 2018.
- [119] DU S, LI T, YANG Y, et al. Multivariate time series forecasting via attention-based encoder–decoder framework[J]. Neurocomputing, 2020, 388 : 269–279.
- [120] YUAN Z, HU J, WU D, et al. A dual-attention recurrent neural network method for deep cone thickener underflow concentration prediction[J]. Sensors, 2020, 20(5) : 1260.
- [121] Pastor J, Menaud J M. SeDuCe: a Testbed for Research on Thermal and Power Management in Datacenters[C] // Proceedings of the 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM2018). 2018 : 1–6.
- [122] ALONSO S, MORÁN A, PÉREZ D, et al. Estimating cooling production and monitoring efficiency in chillers using a soft sensor[J]. Neural Computing and Applications, 2020, 32(23) : 17291–17308.
- [123] COMMITTEE A T. 2011 Thermal Guidelines for Data Processing Environments –Expanded Data Center Classes and Usage Guidance[R]. [S.l.] : ASHRAE, 2011.
- [124] LIU Y, WANG X, XING Y, et al. Learning continuous-time dynamics by Stochastic Differential Networks[J]. arXiv, 2020 : 1–13.
- [125] CHEN R T Q, RUBANOVA Y, BETTENCOURT J, et al. Neural Ordinary Differential Equations[J]. Advances in Neural Information Processing Systems, 2018.

- [126] JIA J, BENSON A R. Neural jump stochastic differential equations[J]. Advances in Neural Information Processing Systems, 2019, 32.
- [127] YILDIZ C, HEINONEN M, LÄHDESMÄKI H. Continuous-time Model-based Reinforcement Learning[C] //International Conference on Machine Learning. 2021: 12009–12018.
- [128] YIN S, LUO H, DING S X. Real-time implementation of fault-tolerant control systems with performance optimization[J]. IEEE Transactions on Industrial Electronics, 2013, 61(5): 2402–2411.
- [129] KOURO S, CORTÉS P, VARGAS R, et al. Model predictive control—A simple and powerful method to control power converters[J]. IEEE Transactions on industrial electronics, 2008, 56(6): 1826–1838.
- [130] DAI W, CHAI T, YANG S X. Data-driven optimization control for safety operation of hematite grinding process[J]. IEEE Transactions on Industrial Electronics, 2014, 62(5): 2930–2941.
- [131] WANG D, LIU D, ZHANG Q, et al. Data-based adaptive critic designs for nonlinear robust optimal control with uncertain dynamics[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2015, 46(11): 1544–1555.
- [132] PROKHOROV D V, WUNSCH D C. Adaptive critic designs[J]. IEEE transactions on Neural Networks, 1997, 8(5): 997–1007.
- [133] WERBOS P J. Foreword: ADP-The Key Direction for Future Research in Intelligent Control and Understanding Brain Intelligence.[J]. IEEE Trans. Syst. Man Cybern. Part B, 2008, 38(4): 898–900.
- [134] DUAN Y, LV Y, ZHANG J, et al. Deep learning for control: the state of the art and prospects[J]. Acta Automatica Sinica, 2016, 42(5): 643–654.
- [135] LIU Y-J, TANG L, TONG S, et al. Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time MIMO systems[J]. IEEE Transactions on Neural Networks and Learning Systems, 2014, 26(1): 165–176.
- [136] LIU L, WANG Z, ZHANG H. Adaptive fault-tolerant tracking control for MIMO discrete-time systems via reinforcement learning algorithm with less learning parameters[J]. IEEE Transactions on Automation Science and Engineering, 2016, 14(1): 299–313.

- [137] XU X, YANG H, LIAN C, et al. Self-learning control using dual heuristic programming with global Laplacian eigenmaps[J]. IEEE Transactions on Industrial Electronics, 2017, 64(12): 9517–9526.
- [138] JIANG Y, FAN J, CHAI T, et al. Dual-rate operational optimal control for flotation industrial process with unknown operational model[J]. IEEE Transactions on Industrial Electronics, 2018, 66(6): 4587–4599.
- [139] MODARES H, LEWIS F L, NAGHIBI-SISTANI M-B. Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems[J]. Automatica, 2014, 50(1): 193–202.
- [140] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning[J]. arXiv preprint arXiv:1312.5602, 2013.
- [141] GUO X, YAN W, CUI R. Integral Reinforcement Learning-Based Adaptive NN Control for Continuous-Time Nonlinear MIMO Systems With Unknown Control Directions[J/OL]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, PP : 1–10.  
<http://dx.doi.org/10.1109/tsmc.2019.2897221>.
- [142] CHAI T, JIA Y, LI H, et al. An intelligent switching control for a mixed separation thickener process[J]. Control Engineering Practice, 2016, 57 : 61–71.
- [143] WANG L, JIA Y, CHAI T, et al. Dual-rate adaptive control for mixed separation thickening process using compensation signal based approach[J]. IEEE Transactions on Industrial Electronics, 2017, 65(4) : 3621–3632.
- [144] 王猛. 矿浆中和沉降分离过程模型软件的研发 [D]. [S.I.] : 东北大学, 2011.
- [145] 唐谟堂. 湿法冶金设备 [M]. [S.I.] : 湿法冶金设备, 2009.
- [146] 王琳岩, 李健, 贾瑶, et al. 混合选别浓密过程双速率智能切换控制 [J]. 自动化学报, 2018, 44(2) : 330–343.
- [147] LUO B, LIU D, HUANG T, et al. Model-free optimal tracking control via critic-only Q-learning[J]. IEEE transactions on neural networks and learning systems, 2016, 27(10) : 2134–2144.
- [148] PADHI R, UNNIKRISHNAN N, WANG X, et al. A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems[J]. Neural Networks, 2006, 19(10) : 1648–1660.

- [149] TAKAHASHI T, LIANG J, QIAO Y-L, et al. Differentiable fluids with solid coupling for learning and control[C] // Proceedings of the AAAI Conference on Artificial Intelligence : Vol 35. 2021 : 6138 – 6146.