

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 Complex 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、北京信息科技大学李宁教授、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，使得训练模型所需的数据量大，且模型在不同场景下泛化能力差。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，所提出的模型及算法有效应用于膏体充填场景下的设备数据分析与管控过程中。具体地，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长时滞、非线性影响。经过膏体浓密机系统运行数据的验证，该模型在长时预测和短时预测场景下均获得较好效果。

2) 针对带有随机性的动态系统在非均匀采样设定下的系统辨识问题，本文提出了常微分方程循环状态空间模型。该模型在基础循环神经网络的状态转移中引入随机路径以建模系统状态转移的随机性。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非均匀采样设定下的模型训练。另外，本文提出了一种求解批常微分方程的再参数化方法以解决时间间隔不均匀时微分方程难以并行求解、模型难以并行训练的问题。利用膏体浓密机系统数据集和两个公有系统辨识数据集对模型进行验证，结果表明该模型在数据集存在随机性以及稀疏非均匀采样间隔时，能获得相比基线模型更好的建模效果。

3) 针对复杂工业系统动态机理未知、难以控制优化问题，本文提出了基于连续时间有模型强化学习的启发式评价网络值迭代算法。该算法采用常微分方程网路构建动态系统的预测模型，并基于积分强化学习构建控制策略的自适应评价函数。同时，依托于评价模块与模型网络的评价、预测能力，采用随机梯度下降法生成控制指令，使得整套控制模型支持高效的在线控制与参数更新。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。同时，该算法成功部署应用于真实矿场的膏体浓密机控制系统中，相比

原始的基于固定规则的控制策略获得了更好的浓度控制效果。

4) 针对具有周期多阶段特性的动态系统辨识问题, 本文提出了连续时间自跳变常微分方程网络。在给定序列输入输出数据的所属阶段标注下, 该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学习连续时间跳变系统的阶段转移过程, 进而在模型开环预测时实现内部阶段变量的自转移。实验环节利用膏体制备过程中的水泥添加系统数据对模型进行验证, 结果表明模型能够在给定系统受控输入下, 准确地预测阶段间的转移时间点以及各阶段内的系统输出, 基于仿真结果成功优化膏体制备过程的水泥成本及设备开关机损耗成本 8.2%。

**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, 浓密机, 膏体充填

## Abstract

In the processing industry, most complex devices or systems have strong non-linearity, high stochasticity, and long time delay. These properties lead to the difficulties in analyzing system mechanisms, identifying dynamic equations, and controlling critical operational indices. Mechanistic-based dynamic system identification methods are not ideal to sufficiently fit the complexity of the system. Deep learning methods ignore the limitations from system mechanisms and prior knowledge, which demands a larger size of training data and lacks generalization in different scenarios. From the continuous-time domain, this dissertation 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.

(1) For the modeling of non-linear and long-delayed complex industrial systems which are essentially continuous-time processes, this dissertation 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 and time-delayed effects from inputs on outputs in continuous-time domain. The evaluation conducted on real industrial datasets of paste thickeners indicates that the model achieves comparable performances on both long-time and short-time predictions.

(2) For the modeling of the complex systems with stochasticity, this dissertation proposes the Ordinary Differential Equation Recurrent State Space Model(ODE-RSSM). On the basis of Recurrent Neural Network, an additional stochastic path defined by latent variable is embedded in the state transition for modeling stochasticity. Furthermore, the ODE-RSSM incorporates an ordinary differential equation network to model the continuous-time evolution of latent states between adjacent time points. Inspired from the equivalent linear transformation on integration limits, this dissertation 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 on three datasets, including the paste thickening dataset and two public system identification datasets, demonstrate that the ODE-RSSM outperforms other baselines in predicting stochastic system in open loop when the time spans of predicted points are uneven.

(3) For the controlling of complex industrial systems without prior knowledge of mechanism, this dissertation proposes a novel continuous-time model-based reinforcement learning control algorithm, Heuristic Critic Network Value Iteration (HCNVI). The method models the dynamic system with differential equation neural network and introduces the integral reinforcement learning to approximate the adaptive critic function under the control law. Depending on the evaluation of critic network and the prediction of model network, the controlling action is determined by stochastic gradient descent algorithms. 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. In comparison to the original rules-based controlling strategy, the algorithm is effective to control the underflow concentration in a stable range.

(4) To address the problem of modeling periodic multi-staged system, this dissertation proposes continuous-time Autonomous Jump Ordinary Differential Equation (AJ-ODE-Net) to learn continuous-time period jump systems. Given sequential inputs and outputs belonged to labeled stages, the model is able to independently learn the sub-dynamics in different stages from non-uniform 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 dataset from an cement addition system in paste production, the results indicates that the AJ-ODE-Net can accurately predict the system outputs and the duration time in each stage. The open-loop prediction of paste system runtime also assists in optimizing 8.2% cost of backfilling.

**Key Words:** **Dynamic system modeling, Model-based reinforcement learning, System identification, Ordinary Differential Equation Neural Network, Thickener, Paste backfilling**

## 目 录

致谢 .....	I
摘要 .....	III
Abstract .....	V
插图清单 .....	IX
表格清单 .....	XII
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 .....	4
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 .....	4
1.2.2 具有随机性的连续时间系统建模问题 .....	5
1.2.3 具有周期多阶段性的连续时间跳变系统建模问题 .....	5
1.2.4 复杂工业系统的无模型控制优化问题 .....	6
1.3 本文的研究内容 .....	6
1.3.1 高时延复杂工业系统建模与预测 .....	7
1.3.2 随机性系统建模与预测 .....	8
1.3.3 连续时间域下的有模型强化学习控制 .....	8
1.3.4 周期跳变系统建模及优化 .....	9
1.4 论文的章节安排 .....	9
2 文献综述 .....	13
2.1 动态系统辨识及有模型控制 .....	13
2.1.1 动态系统建模与辨识 .....	13
2.1.2 基于模型的决策与控制 .....	14
2.2 连续时间系统辨识 .....	15
2.3 深度微分方程网络介绍 .....	16
2.4 微分方程网络在复杂系统建模中的应用 .....	18
2.4.1 非马尔科夫系统及长时延系统建模 .....	18
2.4.2 动态系统随机转移的表示与推理 .....	20
2.4.3 连续时间跳变系统辨识 .....	21

2.5 强化学习及其在控制优化中的应用 .....	22
2.6 本章小结 .....	24
3 高时延复杂工业系统建模与预测 .....	25
3.1 问题形式化描述 .....	27
3.2 基于 ODE-Net 的多输入输出系统预测模型设计 .....	28
3.2.1 利用历史序列编码器构建常微分方程初始状态 .....	29
3.2.2 基于序列编码器的常微分方程初态构建 .....	29
3.2.3 利用可微常微分方程网络构建系统状态空间模型 .....	29
3.2.4 面向不同预测时长需求的常微分方程导数模块定义 .....	30
3.3 离散输入序列的可微并行插值方法及模型训练 .....	31
3.4 实验验证与分析 .....	32
3.4.1 膏体浓密机系统数据集 .....	33
3.4.2 实验参数及评价指标 .....	34
3.4.3 不同模型预测结果对比 .....	35
3.4.4 不同 ODE 求解器下稳定系统以及非稳定系统对比 .....	36
3.4.5 序列编码器的有效性验证及系统时延探究 .....	37
3.4.6 探究序列插值阶数的对预测精度的影响 .....	40
3.5 本章小结 .....	41
4 随机性系统建模与预测 .....	43
4.1 问题形式化描述 .....	44
4.2 常微分方程网络-循环状态空间模型 .....	45
4.2.1 带有随机隐变量的连续时间的生成模型 .....	45
4.2.2 隐变量推理与训练 .....	47
4.3 基于高效隐空间超调的多步预测性能改善 .....	48
4.4 非均匀采样间隔下的批常微分方程并行化求解方法 .....	50
4.5 实验验证与分析 .....	52
4.5.1 数据集介绍 .....	53
4.5.2 实验设定与基线模型选择 .....	54
4.5.3 非均匀采样下不同模型的辨识效果对比 .....	55
4.5.4 隐空间超调对于模型长期预测性能的影响性探究 .....	56
4.5.5 ODE-RSSM 模型在不同采样间隔分布下的泛化能力探究 .....	57
4.5.6 批常微分方程并行化求解算法的时间效率探究 .....	58
4.5.7 探究隐空间超调时重用采样隐状态对于求解 KL-D 效率的影 响 .....	59

4.6 本章小结 .....	59
5 连续时间域下的有模型强化学习控制 .....	61
5.1 连续时间的无模型系统控制形式化描述 .....	63
5.2 启发式评价网络值迭代算法 .....	64
5.2.1 基于常微分方程网络的被控系统模型网络构建 .....	64
5.2.2 基于积分强化学习的策略评价 .....	66
5.2.3 基于随机梯度下降的动作生成 .....	67
5.2.4 短期经验回放 .....	68
5.3 仿真实验验证 .....	71
5.3.1 浓密机仿真模型 .....	72
5.3.2 恒定-阶跃型噪音输入下浓密机控制仿真实验 .....	73
5.3.3 高斯噪音波动输入下浓密机控制仿真实验 .....	77
5.4 真实工业场景下浓密机控制算法性能评估 .....	79
5.5 本章小结 .....	81
6 周期跳变系统建模及优化 .....	83
6.1 问题形式化描述 .....	84
6.2 自跳跃常微分方程网络 .....	85
6.2.1 基于分层 ODE-Net 的稳定及非稳定输出混合建模 .....	85
6.2.2 自跳跃微分方程状态空间定义及阶段自转移 .....	86
6.3 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 .....	88
6.4 模型训练 .....	89
6.5 实验验证与分析 .....	91
6.5.1 数据集介绍 .....	92
6.5.2 数据采集、预处理及模型训练细节 .....	93
6.5.3 跳变系统运行过程辨识与仿真 .....	95
6.5.4 基于预测仿真模型的水泥添加系统启停策略优化 .....	99
6.6 本章小结 .....	101
7 总结 .....	103
7.1 本文研究工作总结 .....	103
7.2 未来工作展望 .....	104
参考文献 .....	107



## 插图清单

图 1-1 人工智能技术在复杂工业设备感知、建模、评价、决策领域中的应用 . . . . .	2
图 1-2 本文研究内容、研究目标、应用场景的对应关系 . . . . .	7
图 3-1 基于 ODE-Net 模型的输入输出系统预测模型整体结构 . . . . .	28
图 3-2 膏体浓密机系统运行过程及监测变量图示 . . . . .	33
图 3-3 训练集、验证集、测试集的构建过程图示 . . . . .	34
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	38
图 3-5 $L = 200$ 时不同 ODE 求解器、不同导数模块下的预测效果对比 . . . . .	39
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 . . . . .	40
图 4-1 ODE-RSSM 中的生成过程和推理过程 . . . . .	45
图 4-2 三种导数模块定义方法图示 . . . . .	47
图 4-3 基于祖先采样的高效隐空间超调 . . . . .	50
图 4-4 引入微分方程网络及多步隐空间超调的效果分析 . . . . .	57
图 4-5 时间复杂度改善效果评估 . . . . .	59
图 5-1 迭代梯度下降过程可视化 . . . . .	69
图 5-2 短期经验回放对评价网络的输出值的影响 . . . . .	70
图 5-3 噪音量变化曲线 . . . . .	74
图 5-4 HCNVI 与其他 ADP 算法在恒定噪音输入下的对比 . . . . .	74
图 5-4 短期经验回放对 HDP 与 HCNVI 的影响 . . . . .	76
图 5-5 HDP 与 HCNVI 在时间消耗上的对比 . . . . .	76
图 5-6 噪音量变化曲线 . . . . .	77
图 5-7 HCNVI 与其他 ADP 算法在波动噪声输入下的对比 . . . . .	78
图 5-8 噪音持续变化下短期经验回放对 HCNVI 的影响 . . . . .	79
图 5-9 底流浓度控制差分对比图 . . . . .	80
图 5-10 底流浓度序列残差分解对比图 . . . . .	80
图 5-11 开启智能控制前后浓密机底流浓度变化对比 . . . . .	81
图 6-1 H-ODE-Net 结构图示 . . . . .	86
图 6-2 AJ-ODE-Net 模型结构 . . . . .	87
图 6-3 基于 AJ-ODE-Net 的编码器-解码器预测框架 . . . . .	88
图 6-4 时间预测器的损失函数定义 . . . . .	90

图 6-5 膏体制备系统工作流程示意图 . . . . .	93
图 6-6 水泥添加控制过程机理简化图 . . . . .	94
图 6-7 不同模型预测水泥添加量、膏体流量、膏体浓度的效果对比。 . .	96
图 6-8 不同模型预测进气口温度、制冷量、制冷机功率的效果对比。 . .	98
图 6-9 不同时间长度下水泥消耗量预测值的 MAPE 变化 . . . . .	99
图 6-10 在不同浓度上限设定下的水泥添加量的仿真预测值 . . . . .	101
图 6-11 不同浓度上阈值对充填成本的影响 . . . . .	102

## 表格清单

表 3-1 膏体浓密机系统数据样例 . . . . .	33
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 . . . . .	35
表 3-3 不同初始隐状态 $h(t_0)$ 生成方法对于预测精度的影响 . . . . .	40
表 3-4 不同序列插值方法对预测精度的影响 . . . . .	41
表 4-1 本章提出方法与现有系统建模方法的对比 . . . . .	44
表 4-2 数据集特性 . . . . .	54
表 4-3 连续搅拌釜式反应器 . . . . .	55
表 4-4 工业绕组数据集 . . . . .	55
表 4-5 浓密机数据集 . . . . .	56
表 4-6 模型在测试集采样比例与训练集采样比例不同时的迁移效果评估 . . . . .	58
表 5-1 参量定义 . . . . .	72
表 5-2 仿真模型常量 . . . . .	72
表 5-3 部分变量计算方法 . . . . .	73
表 5-4 不同控制算法之间性能分析 . . . . .	78
表 5-5 真实工业场景下浓密机控制算法性能评估 . . . . .	79
表 6-1 膏体制备系统阶段转换判别规则 . . . . .	95
表 6-2 制冷系统阶段转换判别规则 . . . . .	95
表 6-3 使用不同模型预测水泥消耗量时的精度和推理时间的对比 . . . . .	99
表 6-4 不同模型累积能耗预测精度和推理时间的对比 . . . . .	100



# 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，且近年来大数据挖掘、机器学习等基础理论及技术日趋完善，使得基于深度学习的复杂工业系统建模方法广受学者们的关注。然而，实际应用问题中的被辨识系统特性迥异，且系统建模问题本身具有较高的复杂性，不同的模型结构及训练方法会极大地影响系统辨识精度以及训练效率。如何根据目标系统的不同特性，设计合理的参数化模型并有效地适配系统的先验属性，以获得最佳的建模效果，是复杂工业系统建模、智能优化控制等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，本文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**，并有效应用于工业膏体充填过程，成功实现了深度学习技术在真实工业系统建模与决策场景下的落地。

## 1.1 研究背景与意义

复杂工业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。随着物联网、云计算、大数据、人工智能等新一代信息技术（Information Technology）、操作技术（Operation Technology）的发展和广泛应用，数字孪生平台、赛博系统等数字化、智能化生产理念正在逐步替代传统的自控模式。同时，以数据驱动为核心的设备全生命周期一体化感知、诊断、评价及优化成为学界、工业界关注的热点问题。

通过梳理现有工业智能领域的先进研究方向以及科研成果，同时参考 LeCun 提出的自治机器智能的系统架构<sup>[1]</sup>，图1-1展示了面向复杂工业设备的感知、建模、评价、决策一体化智能框架实现思路。其中**配置层**代表智能系统与操作员或用户之间的人机交互接口，用于对系统的设定参数以及运行模式选择进行调节。感知层通过传感器、摄像头等信息采集设备实时获得工业系统运行状态及环境参数，并对采集数据进行分析以感知复杂系统内外部环境、原料及产品的高层次状态描述。**决策层**旨在于根据配置层指定的系统运行目标，同时结合感知层提供的检测结果，对系统可控变量进行干预控制，进而达到产品优化、稳定生产的目的。在某些复杂的工业生产场景中，人工设计控制策略是极其困难的，引入**预测及仿真模型能**

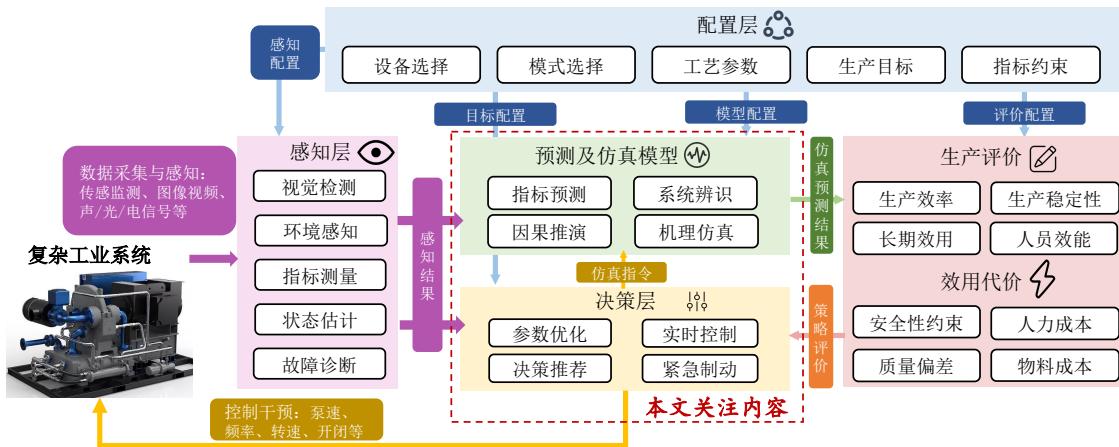


图 1-1 人工智能技术在复杂工业设备感知、建模、评价、决策领域中的应用

够对决策层提供的仿真指令进行模拟，从因果推理角度对系统的未来演化进行仿真与预测。进一步地，结合**生产评价**和**效用代价**模块，从生产效能、成本、质量等多方面因素对模拟仿真结果给出评估，基于此指明决策层模型需要改进和优化的方向。

在上述体系架构中，感知层作为人工智能领域近十年来的热门研究方向，相较于其他板块发展较为成熟，如状态分类、参量回归预测、目标检测、语义分割等技术，只要训练数据充足，很多场景下可以作为整体人工智能方案中可依赖的模块。生产评价模块的构建往往依赖于人的经验以及外部限制，比如误差代价、安全约束、运行成本等，需要根据生产目标、系统特性以及外部环境进行针对性设计。相较于其他模块而言，**现有的预测建模以及决策优化方法在理论体系以及技术方案方面尚不完善，制约了工业人工智能技术的发展。**

想要实现复杂工业动态系统建模，并实现系统的决策优化，其本质在于对系统中目标变量的演化机理以及受其他协变量的影响因果关系进行建模，并基于此实现可控变量的逆向优化。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，系统运行数据的获取成本逐渐降低，这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，有效突破了理论建模方法存在的局限性。

最早追溯到 1950 年，为了进行控制系统设计，Zadeh 等<sup>[2]</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系统的真实输出。以数据为核心驱动力的传统系统识别方法已经逐渐发展为非常完善且成熟的研究领域<sup>[3-7]</sup>。近年来，伴随着人工智能以及机器学习技术领域取得重大突破，传统控制系统建模与优化技术也迎来了新的发展浪潮。动态系统建模作为一项被前人深入研究数十载的“经典问题”，逐渐朝着观测更高维、动态更复杂的方向演化。

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中占据着举足轻重的地位。

人工智能界当代最著名巨擘之一、Meta AI 实验室主任 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，预见给定行为将产生的后果，并辅助智能体决策。本文探究的动态系统建模问题可以认为是世界模型在低维控制、低维观测限定下的特例。图1-1中定义的单个复杂控制系统的感知、建模、决策闭环可以视为一个封闭的“小世界”，本质上与 LeCun 提出的通用人工智能研究范式极其相似。

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>[8]</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  $(s_t, a_t) \rightarrow s_{t+1}$ ，即给定当前状态和动作，预测下一时刻状态。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>[9]</sup>、动态贝叶斯网络、随机森林、最近邻搜索、神经网络<sup>[10]</sup>。近十年来，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受到大家的关注，该方法利用神经网路的前向传播过程模拟系统的动态方程<sup>[11,12]</sup>，再用系统离线数据训练网络参数。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，且其连续可微的特性能够有效适配基于梯度优化的控制决策算法。因此，本论文的讨论内容也限定在基于深度学习的系统建模方法中，具体研究面向不同类型动态系统建模问题的网络设计以及学习方法。

依托于数字计算机技术的应用普及，以及近年来机器学习、强化学习<sup>[13]</sup>、深度学习<sup>[14][15]</sup>、时间序列分析等技术<sup>[16]</sup>的快速发展，离散时间（Discrete-time, DT）域下的动态系统建模研究涌现出了诸多成果，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，连续时间模型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、可变采样间隔、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，而上述特性或问题在复杂工业系统中尤为常见。因此，如何让深度学习

等先进的人工智能技术充分赋能传统的连续时间系统辨识及控制决策理论，是推进传统控制领域发展，加速人工智能技术实现产业应用的有效路径。

荣获神经信息处理会议 2018 (Neural Information Processing, NIPS 2018) 最佳论文奖的《Neural Ordinary Differential Equations》<sup>[17]</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>[18]</sup>，成功搭建了连续时间微分方程与深度学习技术之间的桥梁。在其基础上，神经受控微分方程<sup>[19]</sup>、神经随机微分方程<sup>[20]</sup>、神经偏微分方程<sup>[21]</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、训练显存占用低等优点。该范式一经提出就受到了国内外学者的广泛关注，并应用于时间序列分析、时间点过程分析、动态系统建模、可逆生成模型等领域。

对于大部分工业系统，其底层运行机理所遵循的化学、动力学、热力学定律均可用连续时间微分方程描述，这与微分方程网络的基本特性极其契合。因此，NDE 成功开辟了解决连续时间域下复杂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。同时，对于如何利用训练得到的神经微分方程模型指导目标系统的优化控制，存在一定的研究空白。**为了解决上述建模难的问题，需要结合被建模系统的固有特性、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，本文以膏体充填过程中的深锥浓密机、膏体制备系统为对象，面向长时延非线性、周期多阶段性、随机性三种复杂特性，提出了三种基于常微分方程网络的系统辨识模型，并在辨识模型的基础上，提出了连续时间域下的复杂工业系统在线优化控制方法。本文给出的算法与技术成功应用于工业实践，有效推动了深度学习技术在真实工业系统建模与决策场景下的落地。同时，也为数据驱动的工业系统建模与优化领域开创了新的研究思路。

## 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复杂动态系统时，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

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，而非仿射系统或线性系统，经典的系统假设及参数估计方法难以适用。另外，对于具有长时延的复杂系统，当

前系统输出的变化可能受很长一段时间内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的长时延特性。最后，原始的常微分方程网络对导函数不断进行积分，其模型本质类似于在离散时间序列预测问题中学习序列的差分，在短期预测时模型精度较好。但在长序列开环预测时，对导数持续积分会造成极大的累积误差。

因此，本文围绕连续时间域下的深度系统辨识模型展开研究，以解决长时延、非线性复杂动态系统的长短期开环预测问题。

### 1.2.2 具有随机性的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>[22]</sup>、SNODE<sup>[23]</sup>仅在确定状态空间构建系统的动态模型。由于确定性模型没有引入任何随机成分，不便于实现蒙特卡洛随机采样，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配合使用。另外，确定性模型无法适用于建模带有随机性的系统。当被辨识系统的状态转移过程本身具备较强的随机性时，可以近似认为可状态之间的转移过程服从包含隐变量的随机分布，理想的辨识模型应该能够直接对隐变量的转移分布进行建模，最后，确定性模型无法对系统当前状态的转移分布进行表示。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完整状态，从结果来看这种不完全可观测性也是系统存在随机性的主要原因。

现有连续时间域的系统辨识方法仅能在隐空间中通过最小化误差期望的方式学习状态转移的均值，而无法实现分布表示与状态采样，不仅制约了模型的拟合精度，也难以应用在随机优化问题中。因此本文着重研究存在随机性的复杂连续时间系统建模方法，使辨识模型能够对系统动态的随机转移进行拟合，并在给定观测数据下评估系统的随机性与不确定性。

### 1.2.3 具有周期多阶段性的连续时间跳变系统建模问题

跳变系统是一种在多个子阶段之间随机切换的非线性系统，每阶段下的系统动态方程彼此不同，在每个阶段内的滞留时间会同时受到内部和外部因素的影响<sup>[24]</sup>。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（Expectation-Maximization Algorithm）<sup>[25]</sup>、序列蒙特卡罗（SMC）<sup>[26]</sup> 和变分推理<sup>[27]</sup> 等方法估计系统参数。上述方法通常依赖于对系统结构和阶段滞留时间分布的先验认识，而对于部分复杂的“黑盒”跳变系统，这些先验知识是难以获得的。以往的一些研究采用基于深度学习的自适应性模型识别此类具有多个子过程的复杂时变“黑盒”系统。如 Embed to Control (E2C)<sup>[28]</sup> 和卡尔曼变分自编码器 (KVAE)<sup>[29]</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重。

此类建模时变动态系统的混合模型假定目标系统由多个“黑盒”子系统按权重混合而成。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。目前尚无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，其输出空间同时存在稳定和非稳定过程<sup>[30]</sup>。目前，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多输出系统学习任务。

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辨识模型，依赖已知的系统先验特性提高模型的辨识精度。

#### 1.2.4 复杂工业系统的无模型控制优化问题

大部分复杂工业生产过程往往伴随着较强的非线性、随机性、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>[31,32]</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。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，难以保证生产控制过程的安全。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。为了保证控制模型的控制效果，控制系统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，难以保证模型更新与推理的实时性。同时，在线采样数据时常是间隔非均匀的，大部分离散时间域下的强化学习算法难以适用。最后，受限于无模型强化学习存在采样数据需求量大与不同场景下泛化能力差等缺陷，无模型强化学习算法在真实的工业实践中难以部署应用。

因此本文研究连续时间域下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非均匀采样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。

### 1.3 本文的研究内容

针对第1.2节列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为研究对象，针对系统存在的非线性、随机性、多阶段混合、高时延、不同输出量统计特征不一致等问题，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微分方程网络的系统建模及辨识方法。并在辨识模型的基础上，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1-2展示了本文研究内容、研究目标以及不同应用场景的对应关系。接下来，将给出每一项研究内容的研究思路和简要介绍。

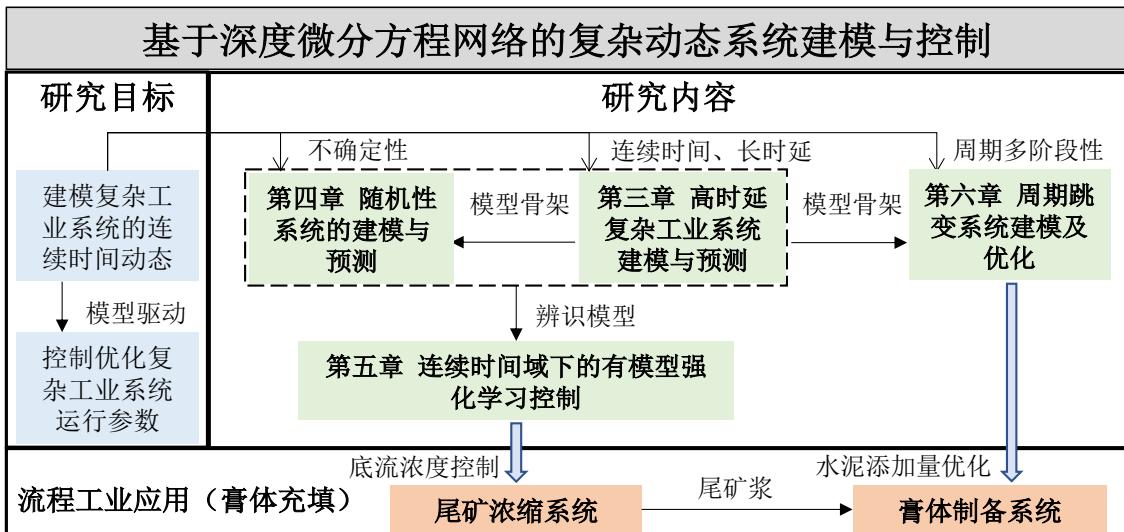


图 1-2 本文研究内容、研究目标、应用场景的对应关系

### 1.3.1 高时延复杂工业系统建模与预测

第一项研究内容“高时延复杂工业系统建模与预测”，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。由于大部分工业系统运行过程遵循连续时间特性，使用连续时间辨识模型与系统的机理先验更佳契合。另外，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，直接利用状态空间模型对系统状态和输入记忆进行压缩是极其困难的。想要将深度学习模型的长距离依赖提取能力引入到连续时间辨识模型中，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的连续时间特性，提出以 ODE-Net 作为模型骨架结构辨识模型，从连续时间域角度拟合目标系统的受控动态过程。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。针对深度自回归序列模型受制于累积预测误差存在，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 ODE-Net 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的深度连续时间（Continuous Time, CT）系统辨识网络，该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设定、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对于解决长时延、非线性工业系统长短期预测问题的有效性。

### 1.3.2 随机性系统建模与预测

第二项研究内容“随机性系统建模与预测”重点讨论了连续时间视角下，对于随机性系统以及非均匀采样场景下系统建模时会遇到的诸多问题。第二项研究内容“随机性系统建模与预测”重点讨论了非均匀采样场景下，对于随机性系统进行拟合建模时会遇到的诸多问题。现实世界中的复杂系统往往具备典型的随机性（*uncertainty*），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转移的建模能力，并给出概率域下的状态转移分布的显示描述，需要从“部分可观测马尔可夫决策系统的随机性表示”这一科学问题出发，研究贝叶斯视角下系统隐空间状态的时序生成过程及逆向推理方法。

具体地，该研究内容针对随机性系统难以建模的问题，研究如何向确定性微分方程系统演化中添加随机转移路径，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无法实现沿时间反向传播（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，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对于解决随机性系统多步开环预测问题的有效性。

### 1.3.3 连续时间域下的有模型强化学习控制

第三项研究内容“连续时间域下的有模型强化学习控制”重点讨论了连续时间域下，基于数据驱动的复杂工业系统优化控制方法。由于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。另一方面，系统运行过程产生的历史数据为实现数据驱动的优化控制提供了可行性。

该研究内容针对非线性、高时滞复杂工业系统控制优化难的问题，研究了连续时间域下的自适应动态规划控制方法，定义了被控系统的状态空间、动作空间、效用函数、状态转移函数，同时提出了离线系统建模与在线强化学习相结合的控制器构建及生产环境在线更新策略，另外，针对在线环境下策略网络增量训练开销大、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，结合评价网络与模型网络的长期效用评估能力，采用梯度优

化算法直接求解控制指令，避免控制网络在线学习的计算开销。与此同时，针对传统自适应动态规划算法中评价模块参数收敛慢、难以准确给出策略优化方向的问题，本文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，本文选用尾矿浓密机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精度及稳定性。

### 1.3.4 周期跳变系统建模及优化

第四项研究内容“周期跳变系统建模及优化”重点讨论了连续时间视角下周期跳变系统建模问题。某些工业系统的运行过程呈现周期多阶段特性，系统在不同阶段下的动态方程彼此迥异，且各阶段的持续时间、状态跳变的触发条件会同时受到内部、外部多种混杂因素共同影响。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差异大，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，并在开环预测时，支持阶段转换处隐空间状态的衔接。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，本文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，研究了面向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，利用膏体制备过程的水泥添加系统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。

## 1.4 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，各种内容安排如下：

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。

第四章研究了基于深度常微分方程-马尔可夫模型的随机性系统建模，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和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常微分方程并行求解算法的有效性。

第五章研究了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，首先，该章基于连续时间强化学习理论给出系统优化控制的符号变量定义以及形式化描述。然后，描述了如何利用神经常微分方程构建模型网络以近似被控系统的动态方程。进一步地，结合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值进行近似。同时，该章提出了启发式评价网络值迭代控制算法，该算法利用评价网络与模型网络的预测、评价能力，结合随机梯度下降法生成控制指令。最后，该章提出了短期经验回放技术以提升预测局部评价值梯度的准确度，实验环节评估了该章提出的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。

第六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。其次，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。再次，提出了用于建模连续时间周期跳变系统的自跳跃常微分方程模型，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。进一步地，给出了基于该模型的编码器解码器框架及其训练方法。最后，实验环节介绍了如何将上述框架应用于建模膏体制备系统中的水泥添加过程，预测给定浓密机出料浓度及流量下的水泥消耗以及膏体浓度变化，同时基于预测仿真结果，对控制水泥添加系统启停的浓度设定值进行优化。

第七章对本文各章的主要研究内容和创新点进行了回顾，并总结了本文工作

在基于深度学习的连续时间动态系统辨识领域所作出的学术贡献。同时，该章围绕机理先验与数据驱动算法结合的混合动态模型构建这一领域，分析了向模型中引入系统先验知识的实现思路，并分别从机理模型参数建模、残差拟合、基于先验指导的网络设计三个方面，讨论了现有方法的优势与局限，最后，对该领域未来的未来研究方向做出了展望。



## 2 文献综述

动态系统辨识及优化控制作为控制领域中具有悠久历史的研究问题，一直受到智能控制、计算机控制系统、人工智能等领域学者的持续关注。每年都有大量研究成果发表于控制以及人工智能领域的顶级会议与期刊上。本章将首先介绍动态系统辨识及有模型控制的分类及发展现状。然后围绕连续时间域下的系统辨识，介绍其基本概念、现有方法，并对比连续时间模型相比于离散时间模型的优势。同时，具体针对非马尔可夫系统、跳变系统、随机性系统等与本文研究内容相关的复杂系统，重点介绍现有方法在此类系统中的应用以及方法存在的瓶颈。最后对本章综述内容进行总结。

### 2.1 动态系统辨识及有模型控制

对于机理模型未知的动态系统，为了实现无模型情况下的系统控制优化，一种可行的方案是利用数据驱动方法，从系统离线数据中学习系统的近似动态方程，并以拟合模型为基础，构建系统的控制策略，进而实现闭环或开环的优化控制。接下来，本节将分别介绍动态系统建模与辨识、基于模型的决策控制两部分。

#### 2.1.1 动态系统建模与辨识

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>[8]</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  $(s_t, a_t) \rightarrow s_{t+1}$ ，即给定当前时刻系统状态和选择的动作，预测下一时刻状态。利用学习得到的前向模型可以辅助实现系统的控制优化，比如有模型强化学习、模型预测控制等<sup>[33]</sup>。基于有监督学习的前向模型辨识方法可以按照模型结构分为参数化方法和非参数化，以及按照是否具有外推泛化能力分为精确模型和近似模型。

非参数化的精确建模方法采用回放池<sup>[34]</sup>方法存储系统的历历史轨迹，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>[35,35]</sup>，通过利用数据集中的状态-动作点，可以外推预测某一新给定状态-动作对下，对应衍生状态的高斯分布  $p(f^* | x^*, f, X) = \mathcal{N}(f^* | \mu, \sigma^2)$ ，其中  $X$  代表完整数据集， $f$  代表  $X$  上的观测。 $x^* = [s_t, a_t]$  代表给定当前时刻状态和控制输入下，系统的预测结果。该方法预测出的高斯分布能够对系统的随机状态转移进行表示，预测出的标准差  $\sigma$  隐式地包含模型预测的不确定性和系统本身的随机性。

相比于非参数化方法需要存储所有历史数据集的模式，参数化方法的存储空间占用以及计算量与观测数据集的大小是独立的。因此，参数化方法的应用更为广泛。参数化的精确建模方法也称为表格方法，对于离散控制空间和状态空间下的每一种可能的状态转移维护转移概率矩阵。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:

$$P(s' | s, a) = \frac{n(s, a, s')}{\sum_{s'} n(s, a, s')} \quad (2-1)$$

其中  $P$  代表真实系统  $\mathcal{P}$  中状态转移概率矩阵的近似,  $n(s, a, s')$  代表在数据集中状态  $s$  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或者连续状态/动作空间。

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。由于该方法适用面广、空间占用少、计算速度快, 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。理论上, 可以使用任意参数近似方法学习模型, 如线性回归<sup>[9]</sup>, 动态贝叶斯网络、随机森林、最近邻搜索、神经网络<sup>[10]</sup>等。

近十年来, 深度学习逐渐成为解决复杂函数近似问题的主流方法。因为复杂动态系统往往具备典型的高维输入输出空间以及非线性特性, 基于深度神经网络的动态系统辨识模型受到广泛关注。利用神经网络模型的前向传播过程可以模拟系统的动态过程<sup>[11,12]</sup>。循环神经网络 (Recurrent neural network, RNN) 作为一种拥有马尔科夫特性的网络模型, 与动态系统极为类似, 且因为隐状态的存在, RNN 及衍生模型 LSTM、GRU 等能够很好地处理长时依赖问题以及长期预测问题, 因此在系统建模领域被广泛应用<sup>[36,37]</sup>。本文的研究重点也限定在参数化的、函数近似的、基于深度学习的系统动态建模方法中, 探究面向不同类型动态系统的深度网络设计以及学习方法。

### 2.1.2 基于模型的决策与控制

在基于模型的系统优化控制方面, 可以按照辨识模型的使用方式分为离散规划 (Discrete planning) 和可微规划 (Differential planning) 两类。

离散规划是过去人工智能和强化学习领域的核心研究问题。一般的方法是将控制序列与辨识模型给出的预测结果存储在树或者表结构中并寻找最优的控制序列, 或者将上述轨迹数据作为训练目标以优化评价值函数和策略函数。原则上任何的辨识模型都可应用于离散规划, 一些典型模型包括: 概率限制搜索<sup>[38]</sup>、宽度及深度限制搜索<sup>[39]</sup>、蒙特卡洛搜索<sup>[9]</sup>、蒙特卡洛树搜索<sup>[40]</sup>、极小-极大法或者简单的单步搜索<sup>[41]</sup>等。这些方法不需要辨识模型可微, 但是执行决策时的计算量较大。

可微规划一般依赖于可微的辨识模型, 通过计算累积奖赏目标对控制序列或者策略模型参数的梯度, 实现基于梯度的决策优化。尽管现实世界中的环境或者仿真器时常是不可微的, 但是训练得到的近似模型 (如神经网络) 通常是可微的, 策略模型可以充分利用辨识模型的可微特性。基于可微特性给出的一阶或二阶导数可以帮助策略快速找到局部最优的控制序列或模型参数。可微模型也可以通过

物理规则得到，比如可微分的物理引擎<sup>[42,43]</sup>。一个典型的可微规划的例子是迭代线性二次规划，该方法利用线性模型构建搜索策略中的规划器<sup>[44]</sup>。在强化学习领域中，基于梯度的规划方法也常被称为值梯度法。典型的例子包括使用高斯过程动态模型的 PILCO 算法<sup>[35]</sup> 以及使用隐空间神经网络动态模型的 Dreamer<sup>[45]</sup> 和时序分割模型（Temporal Segment Models）<sup>[46]</sup>。

当系统的转换方程以及奖赏函数相对平滑的情况下，基于梯度的规划算法十分有效，因此在机器人以及控制领域被广泛应用，但是对于离散问题以及奖励函数稀疏的情况难以适用。

## 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）角度进行系统建模成为系统辨识理论与应用研究领域的主流方向，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，近年来其相关技术体系发展相较于离散时间体系较为滞后。但是对比离散时间模型，从连续时间域角度进行系统建模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真实系统进行建模可以更好地匹配客观世界中的物理规律，且增加模型的可解释性。
- **对于先验知识的表示：**以动力学系统为例，刚体的位移、动量、加速度之间的相关关系在连续时间系统内可以被非常容易地表示出来，而对于离散时间系统表示此类先验特性会更加困难。因此连续时间模型能够更容易地嵌入系统先验知识，降低模型的自由度。
- **潜在的数据滤波：**一般的离散时间系统辨识方法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，避免了预处理对于原始数据特性的破坏。
- **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差异中学习系统动态，导致模型的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是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。

- **刚体系统建模:** 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下的演化过程进行协同建模。

对于复杂工业系统, 上述七大特性或需求是时常出现的。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。所要辨识的模型通常由表征系统输入输出关系的数学模型描述, 该模型有其特定的结构和参数。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:

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

$\frac{d^i y(t)}{dt^i}$  代表信号  $y(t)$  对时间的  $i$  阶导数,  $v(t)$  表示常数项。上式表示了在任一时刻系统输出  $y(t)$  对时间的各阶导数与输入量  $u(t)$  对时间的各阶导数之间存在的线性关系。

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:

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

$f$  为系统动态,  $C$  为投影矩阵,  $\mathbf{h}(t)$  代表系统的隐空间状态。对于线性系统式 (2-2),  $f$  可以表示为线性函数。然而, 对于许多客观世界存在的复杂动力学系统, 动态  $f$  时常为具有较强非线性特性的某一未知函数, 想要基于第一性原理近似推导出系统的方程结构是极其困难,  $f$  结构未知也导致传统的系统参数辨识方法难以适用此类复杂系统, 制约了现有控制优化理论在复杂工业系统中的应用。因此, 该领域亟需一种先进的系统建模或辨识方案, 能够在模型参数结构均未知的情况下, 从连续时间域角度对黑盒系统进行高精度辨识。

## 2.3 深度微分方程网络介绍

荣获神经信息处理会议 2018 (Neural Information Processing, NIPS 2018) 最佳论文奖的《Neural Ordinary Differential Equations》<sup>[17]</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>[18]</sup>,

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

其中  $f_\theta : \mathbb{R} \times \mathbb{R}^{d_1 \times \cdots \times d_k} \rightarrow \mathbb{R}^{d_1 \times \cdots \times d_k}$  代表标准的神经网络结构,  $\theta$  代表可学习的神经网络参数。 $y : [0, T] \rightarrow \mathbb{R}^{d_1 \times \cdots \times d_k}$  代表微分方程的解。神经微分方程的核心是将微分方程求解过程作为可微算子嵌入在计算图, 利用联合敏感度定理<sup>[17]</sup> 计算损失函数对连续时间隐状态  $y(t)$  和  $\theta$  的梯度, 进而训练微分方程网络中的参数。

ODE-Net 的主要应用包括作为 Res-Net 的替代、时序生成过程建模以及可逆正则化流<sup>[47]</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。ODE-Net 将 2.1 节所述的连续时间模型的优势引入到基于深度学习的时间序列模型中，因此开辟了解决时间序列分析、动态系统建模问题的新途径。

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建模种群大小变化：

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

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  $t$ ，等号右侧是理论经验公式，表示物种之间的交互作用。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有效估计，但是模型方程往往是经过一定程度的问题简化得到的，模型的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：

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

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)<sup>[48]</sup>。

假设观察数据为  $x_i(t_j) \in \mathbb{R}$ ,  $y_i(t_j) \in \mathbb{R}$ , 其中  $i = 1, \dots, N$  表示各个独立的观测序列，每个序列来自于不同的初始条件，以及  $j = 1, \dots, M$  对应于不同的观测时间点  $t_j \in [0, T]$  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  $x(t)$  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训练  $\alpha, \beta, \gamma, \theta$ :

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

将式 (2-5) 中的经验模型替换为式 (2-6) 能够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。

一般情况下，用于动态系统建模的网络通常较小，Ling 等<sup>[49]</sup> 在湍流过程建模中使用了宽度为 10，层数为 10 的前向神经网络。Rackauckas 等<sup>[48]</sup> 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式。Zhong 等<sup>[50]</sup>

采用 ODE-Net 对符合哈密尔顿特性的刚体运动系统进行建模学习，巧妙地将物理先验知识融入到学习模型的设计中。并有效地应用于刚体运动过程的优化控制问题中。Ayed 等<sup>[51]</sup> 采用 ODE-Net 模型从系统状态的部分可观测信息学习复杂系统的非线性时空变换过程。类似于该方法，Ling 等<sup>[49]</sup> 使用神经网络近似雷诺平均 Navier-Stokes 模型中的雷诺应力，以满足某些物理不变性，有效实现了对于湍流现象的建模。同时，在海洋气候建模领域中，Ramadhan 等<sup>[52]</sup> 使用一个小型的多层感知机对湍流垂直热通量进行建模。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 Demeester 等<sup>[22]</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。Yuan 等<sup>[53]</sup> 在时间感知 RNN 模型中引入了历史序列编码器以估计常微分方程的初始状态，并成功应用于深锥浓密机底流浓度预测问题。SNODE<sup>[23]</sup> 使用谱方法以快速、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>[54]</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。

尽管微分方程网络在系统先验引入、非均匀采样序列处理等方面具备明显优势，但在计算效率上相比同类模型存在一定劣势。首先，作为状态循环更新的串行计算模型，ODE-Net 无法实现类似于一维卷积神经网络的并行计算。另外，受限于数值求解器的计算效率制约，想要得到高精度的解，必然需要设定更小的迭代步长，这会显著增加网络训练和推理的时间开销。近年来关于微分方程网络求解器效率优化的研究正在逐步攻克这一难题，Zhuang 等<sup>[55]</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>[56]</sup> 提出具有理论保证的最优传输和稳定性正则化的 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 方程的高阶导数进行正则化<sup>[57]</sup>、对求解时间点添加随机扰动<sup>[58]</sup> 等方案也能够起到正则化作用，加速 ODE-Net 的求解效率。可以相信在不久的将来，微分方程网络在求解效率以及问题适用性等方面都能拥有新的突破。

## 2.4 微分方程网络在复杂系统建模中的应用

### 2.4.1 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，想要使用神经微分方程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。但是，大部分客观物理过程往往是不完全观测且具有长时延的，马尔科夫特性假设往往难以成立。对于存在长时关联或观测空间不完全的非马尔科夫系统，单一的神经微分方程结构难以对其进行有效建模。一种较为简单有效的解决方案是将微分方程网络与循环神经网络相互结合，利用循环网络的长期记忆能力弥补微分方程模型难以记忆长期历史信息的缺陷。另外，在循环神经网络状态更新的间隙，利用

神经微分方程建模隐状态在相邻时间点间的连续时间演化。该方式能够使循环神经网络具备处理非均匀采样序列的能力。如基于 ODE-Net 衍生的 ODE-RNN<sup>[59]</sup>、GRU-ODE-Bayes<sup>[60]</sup>、ODE-LSTM<sup>[61]</sup>。

神经受控微分方程（Neural Controlled Differential Equation）<sup>[19]</sup> 将控制信号的微分项融入在 ODE 网络的求解中。相比于在输入序列采样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），神经受控微分方程不仅显著减少计算图的大小，节省显存开销，同时能够获得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 神经受控微分方程<sup>[62]</sup> 通过签名变换对受控信号进行转换，该方法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，不便于实现在线预测的情况，Morrill 等<sup>[63]</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>[59]</sup> 将 ODE-Net 的初始状态视为先验分布为标准高斯分布的隐变量，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。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>[64]</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求解微分方程所需的初态，并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。模型有效应用于多智能体场景下稀疏时间序列的插值预测与外推预测。

近年来，Transformer 模型<sup>[65]</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>[66]</sup>、Autoformer<sup>[67]</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，得益于 ODE-Net 的连续时间特性，模型能够与 Transformer 进行结合，处理非均匀采样数据下的长序列预测及建模问题，如构建连续时间注意力<sup>[68]</sup>、连续空间位置编码（position embedding）<sup>[69]</sup> 等。

对于具有长时滞的工业输入输出系统，简单地将依赖于马尔科夫假设的微分网络模型直接应用于建模系统输出变化是不可行的，需要结合长序列编码技术提取系统历史输入、输出数据中对于未来预测有帮助的信息，并辅助模型在隐空间中进行动态系统预测，而现有技术难以实现这一目标。另外，复杂系统预测模型一般需要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。如 MPC 中的预测模型模拟了被控系统在受控输入下的系统输出变化。其次，预测模型可以作为系统在长控制输入信号下的仿真环境，验证控制器的性能和安全性。两种场景分别对应系统的短期预测和长期预测。

然而，在现有连续时间系统预测的方法研究中，很少有工作关注于如何设计模型结构以同时适用于短期和长期预测。

### 2.4.2 动态系统随机转移的表示与推理

对于现有的大部分连续时间模型，由于模型内部的状态转换是确定性的，因此对于存在随机性的系统，其建模能力受限。在建模系统的随机性方面，深度时序生成模型<sup>[70-72]</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（Variational Auto-encoder）框架<sup>[73]</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。无参数化的 PR-SSM<sup>[74]</sup> 和 PILCO<sup>[35]</sup> 采用高斯过程学习输入/输出系统的概率状态空间模型。ODE<sup>2</sup>VAE<sup>[75]</sup> 基于贝叶斯理论，构建二阶 ODE-Net 模型用于高维时间序列的建模，同时利用 ODE-Net 能够描述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约束。利用上述方法构建的序列生成模型能够在给定条件序列输入的情况下，通过推理隐变量的后验分布，对系统的随机性给出量化描述。这些方法广泛应用于缺失数据填充<sup>[29]</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）：

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t) dt + \sigma(\mathbf{x}_t) dW_t \quad (2-8)$$

随机微分方程网络采用神经网络建模式(2-8)中的  $\mathbf{f}(\mathbf{x})$  和  $\sigma(\mathbf{x})$ 。其中  $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^D$  是系统的确定性状态演化，也称为漂移项（Drift process）。 $\sigma(\mathbf{x}) \in \mathbb{R}^D$  是用神经网络表示的扩散项系数。 $W_t \in \mathbb{R}^D$  是多变量维纳过程，其初始状态为  $W_0 = \mathbf{0}$ ，经过时间  $s$  之后的状态增量  $W_{t+s} - W_t \sim \mathcal{N}(\mathbf{0}, sI)$  服从标准差为  $\sqrt{s}$  的高斯分布。

SDE 网络参数也可以采用联合敏感度法（Adjoint Sensitivity）进行训练<sup>[20]</sup>。由于神经随机微分方程的前向传播过程依赖于带有随机性的 Wiener 过程，为了保证反向求解 SDE 时对 Wiener 过程的采样与前向传播保持一致，作者提出基于虚拟布朗树（Virtual Brownian Tree）的伪随机数生成策略实现反向传播阶段的随机数重塑，该方法仅需常数级  $\mathcal{O}(1)$  存储空间，即可对 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是特定时间点下，wiener 过程采样时间复杂度为  $\mathcal{O}(\log n)$ 。神经跳变随机微分方程（Neural Jump Stochastic Differential Equations, NJSDE）<sup>[77]</sup> 将扩散项中对时间的微分替换为观测点次数的微分，该方法能够在建模系统隐空间动态的同时对观测点出现事件本身以及出现时刻进行建模，该模型能够有效地应用于地震预测及药物预测。

在某些应用场景下，训练完成的识别模型需要在线地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>[78]</sup>。这限制了上述介绍的 Laten-

tODE、Latent SDE、ODE2VAE 等离线模型的使用，而需要选择具有循环状态更新能力的模型。另外，带有随机隐变量的时序生成模型一般采用最大化证据下限的方式进行训练，难以实现一般循环神经网络的沿时间反向传播，因此在长序列开环预测时难以保证较高精度。

### 2.4.3 连续时间跳变系统辨识

连续时间跳变系统（Continuous Time Jump System, CTJS）包含多个连续时间子系统，系统演化过程受到隶属于有限状态集的内部阶段变量控制<sup>[79]</sup>。作为一种特殊的连续时间系统，CTJS 广泛应用于异常检测<sup>[80]</sup>，优化控制<sup>[81]</sup>，状态估计<sup>[79]</sup>等领域。

以最简单的连续时间跳变线性系统<sup>[82]</sup>为例：

$$\dot{x}(t) = A(\sigma(t))x(t) + B(\sigma(t))u(t) \quad (2-9)$$

其中  $A$  和  $B$  为模型参数。 $\sigma(t)$  为随机跳变的阶段变量，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：

$$\begin{aligned} p_{ii} &= 0 \\ q_i &= -q_{ii} = \sum_{l \neq i} q_{il} \\ p_{ij} &= \frac{q_{ij}}{q_i}, \quad (i \neq j). \end{aligned} \quad (2-10)$$

给定单步转移概率矩阵为  $(p_{ij})_{N \times N}$ ，定义  $\{r_k; k = 1, 2, \dots\}$  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。假设在时刻  $t$ ，系统所处阶段为  $r_k = i$ ，滞留时间  $\tau_k$  服从参数为  $q_i$  的指数分布，即  $\tau_k \sim \text{Exp}(q_i)$ 。过程以概率  $p_{ij}$  跳跃至状态  $j \neq i$ 。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  $\{\sigma(t)\}$  的变化过程。

为了从包含多阶段混合数据的离线序列中学习跳变系统参数  $\{A(i), B(i); i \in \underline{N}\}$  以及阶段跳变过程参数  $\{q_i, p_{ij}; i, j \in \underline{N}\}$ ，Ashley 等采用序列蒙特卡洛（Sequential Monte Carlo, SMC）<sup>[83]</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，Balenzuela 等将随机近似（Stochastic approximation, SA）<sup>[27,84]</sup> 思想引入 SMC 框架中，提出了一种基于 EM 算法的马尔科夫跳变系统辨识方法<sup>[25]</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 方法在合并多个高斯成分的 E 步内使用近似算法，以减少高斯分量的数量。Opper 等<sup>[27]</sup> 采用平均场理论，在给定有噪音的观测数据下，近似推理阶段变量并估计系统参数。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，该方法计算系统潜在转移路径概率分布的近似，因而易于实现平

均场的因子化。

现有复杂跳变系统建模方法会遇到两大技术难题。首先，跳变系统各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。大多阶段跳变参数估计方法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，部分复杂工业场景下先验知识难以获得。另外，对于带有多输出项的工业系统，其输出项中可能同时存在稳定和非稳定过程<sup>[30]</sup>。目前为止，现有的未经过特定设计的辨识模型难以解决此类带有混合时序特性的系统学习任务。

## 2.5 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>[31]</sup> 和自适应动态规划<sup>[85,86]</sup> 是解决最优控制问题的常见解决方案。其中强化学习技术衍生于智能控制以及人工智能领域，主要用来解决序贯马尔科夫决策问题。积分强化学习是在连续时间系统中学习评价函数的有效方式。对于一般的连续时间系统：

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

定义积分折扣累积奖赏为如下形式：

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

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 处做分割，可以得到递归式：

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

其中， $\mu$  代表控制策略， $u(\tau) = \mu(x(\tau))$ 。 $V^\mu$  代表对策略  $\mu$  的长期评价， $r$  是效用函数。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。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>[87]</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>[88]</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。

作为最优控制领域的典型问题，追踪控制旨在于最小化控制系统指标与目标轨迹之间的距离。得益于积分强化学习理论的发展，Modare 等<sup>[89]</sup> 利用基于积分

强化学习的策略迭代算法实现了对非线性连续时间系统的最优追踪控制。Lewis 等<sup>[90]</sup>扩展了离散时间域下的同策略迭代到连续时间非线性系统中，保证了系统的稳定特性。进一步地，为了打破要求系统动态增益矩阵已知的限制，Jiang 等<sup>[91]</sup>提出了一种仅利用系统观测迭代学习最优追踪控制策略的免模型方法，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>[92]</sup>考虑了在系统存在扰动时，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（Deep reinforcement learning, DRL）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>[40]</sup> 和深度 Q 学习（Deep Q-Learning Network, DQN）<sup>[93]</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极大的期待。

对于部分复杂系统而言，系统的高时滞特性与复杂非线性不满足传统控制技术要求的前提假设。而深度学习方法在处理复杂非线性映射、长时滞关联方面具有极大优势，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>[94]</sup>首先提出采用 DDPG 模型实现系统的过程追踪控制，将系统输出偏离目标值的 L1 距离作为惩罚函数，在仿真实验中测试了模型的控制追踪能力。Yu 等<sup>[95]</sup>采用确定性策略梯度算法解决了自动潜航器的最优追踪控制问题。仿真实验表明该方法的性能优于 PID 算法。Kim 等<sup>[96]</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）为主体结构，利用深度强化学习算法对协状态评价网络和动作网络进行训练。经过连续搅拌反应器模型的验证，该方法控制效果较好。Kim 等<sup>[97]</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下非线性仿射系统的最优控制问题。

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>[98]</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可微的梯度估计器<sup>[99]</sup>，使控制和仿真任务的学习更高效、更鲁棒。连续时间域下的深度强化算法具有处理非等间隔观测、实时控制等优势，通过结合基于微分方程网络的系统辨识模型，能够实现有模型的连续时间强化学习，在决策智能体中引入系统的先验信息，优化模型的训练效率与控制效果。

现有在线控制算法的学习过程中大多依赖于增量的单点数据，未充分利用系统短期内产生的历史轨迹数据，在数据利用率方面存在改进空间。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。上述方法均依赖决策网络进行控制量计算，而对于控制网络的在线训练将产生较大的时间开销，使得现有方案

难以满足工业场景下的实时性要求。

## 2.6 本章小结

本章首先对动态系统辨识及有模型控制的起源、分类以及研究进展进行了综述，然后重点介绍了连续时间系统辨识方法的基本概念，以及相对于离散时间方法的优势。由于传统的连续时间系统辨识方法应用于复杂黑盒系统时存在一定局限性，引出了神经微分方程网络这一新兴连续时间系统辨识方法在复杂系统建模中的应用，同时，具体针对非马尔可夫系统、跳变系统、随机性系统等与本文研究内容相关的复杂系统，重点介绍了现有方法的研究进展及其在实际系统应用中所面临的局限性。

### 3 高时延复杂工业系统建模与预测

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。模型预测控制方法首先构建被控系统的预测仿真模型，然后依赖于预测模型的推演结果实现系统优化及控制策略，有效解决了复杂系统的控制决策难题<sup>[100-102]</sup>。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，并利用采集到的数据集以数据驱动方法训练时序预测模型<sup>[103]</sup>。为了获得鲁棒、高精度的预测模型，需要对复杂工业系统的内在特性进行深入分析，大部分复杂工业系统通常具有以下典型特征：

- **非线性系统:** 大多数工业系统的运行过程具有极强的非线性，而非简单的仿射系统或线性系统。
- **系统的部分观测性:** 一般情况下，从传感器或利用其他感知方法提取的系统信息是不完整的，导致系统中存在许多未知的隐变量。
- **长时延的影响:** 系统当前状态的变化受很长一段时间之前外部输入及内部状态的影响。
- **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连续时间的微分方程表示。

对于上述复杂工业系统特点，许多工业界和学术界的销售人员通过构建不同的预测模型来解决这些问题。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>[101,104-107]</sup>、设备异常检测等<sup>[108]</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连续时间先验特性会增加模型的自由度及拟合难度，最终限制模型精度。

另一方面，复杂系统预测模型一般需要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。如 MPC 方法中，预测模型能够模拟被控系统会如何受控制输入影响的，从而将系统最优控制近似为序列优化问题。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>[109]</sup>，因此可以作为验证系统控制策略可行性的试错仿真环境。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，系统模拟需要预测模型有更高的稳健性和鲁棒性以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，很少有工作关注于如何设计预测模型以同时适用于短期和长期预测。

针对上述工业系统存在的复杂特性以及长短期预测需求，本章以 ODE-Net 作为模型骨架结构，提出一种由序列编码器、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。最终实现在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，本章提出方法构建了基于深度网络的连续时间状态空间模型。该模块相比于传统的离散序列模型，如循环神经网络（Recurrent Neural Network, 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：

- 提出了一种基于深度学习的工业系统连续时间预测模型。网络由三个部分组成：序列编码器、状态解码器和导数模块。
- 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，分别用于处理短期和长期预测任务。
- 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，本章节还进行了消融实验，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连续时间域下的状态空间模型表示方法。3.2节主要介绍了基于 ODE-Net 的多输入输出工业系统连续时间预测模型，并分别介绍了序列编码器、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。3.3介绍了用于训练辨识模型的损失函数以及针对离散输入序列的可微并行插值方法。3.4节为实验章节，首先介绍了膏体浓密机系统预测问题以及数据集形式，然后对比了本章所述方法相比于其他系统建模预测模型的优势。最后通过消融实验探究了导数模块形式、序列编码长度、插值阶数对于预测效果的影响。3.5节对本章工作进行了总结，并对未来的研方向做出展望。

### 3.1 问题形式化描述

对于一般的动态系统，系统输出量定义为  $\mathbf{y}(k) \in \mathbb{R}^n$ ，系统控制输入量定义为  $\mathbf{x}(k) \in \mathbb{R}^m$ ，其中  $k$  为采样时间点。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。因此需要从时序角度引入目标系统的历史信息以消除不完全观测带来的非确定性。具体地，本章关注的系统预测问题为在已知未来系统输入和历史系统运行轨迹下预测系统未来输出。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  $\mathbf{X}_f^M = [\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(M-1)]$ ，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，是因为在模型预测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  $\mathbf{h}(0)$ ，该隐状态携带有历史序列  $\mathbf{X}_p^{N_x}$  和  $\mathbf{Y}_p^{N_y}$  的信息。通过构建合理的模型  $D(\cdot, \cdot)$ ，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  $\mathbf{h}(k)$  即可估计系统输出。

为了在预测模型中构造合理有效的模块  $D$ ，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>[110]</sup>：

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

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。控制输入对系统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>[111, 112]</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。对于复杂的预测模型可以进一步表示为式(3-1)和式(3-3)。本章的目标是利用

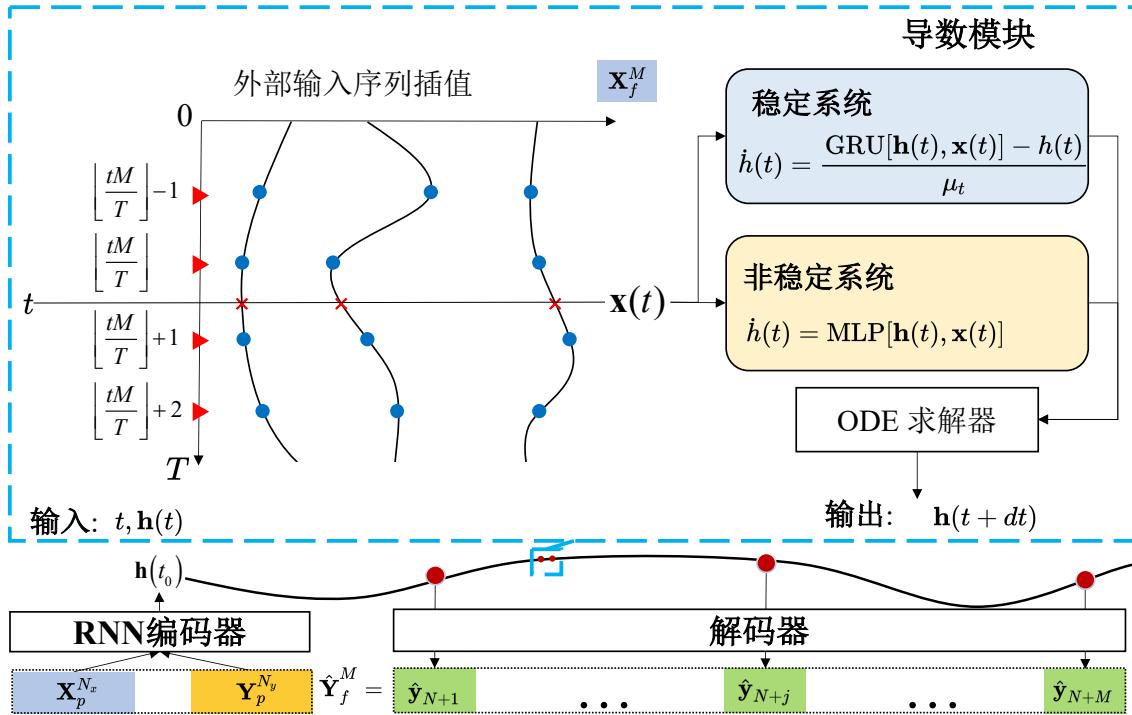


图 3-1 基于 ODE-Net 模型的输入输出系统预测模型整体结构

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  $d(\cdot, \cdot)$  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  $\hat{\mathbf{Y}}_f^M$  以近似系统的真实输出  $\mathbf{Y}_f^M$ 。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  $\mathbf{h}(t)$  的瞬时导数  $\dot{\mathbf{h}}(t)$ 。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  $\mathbf{x}(t)$  作为导数模块的输入。接下来，基于初始状态  $\mathbf{h}(t_0)$  和导数模块定义的常微分方程，利用 ODE 求解器对微分方程进行求解得到时间范围  $[t_0 \leq t \leq t_M]$  内的系统隐状态  $\mathbf{h}(t)$ 。

最后，模型利用多层感知器 (MLP) 网络构建**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输出  $\hat{\mathbf{Y}}_f^M$ 。

下面，本章将详细介绍所述模型的每个模块的技术细节。

### 3.2.1 利用历史序列编码器构建常微分方程初始状态

### 3.2.2 基于序列编码器的常微分方程初态构建

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。本章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，本章利用了 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有帮助的信息，该状态将作为待解 ODE-Net 的初始状态。

### 3.2.3 利用可微常微分方程网络构建系统状态空间模型

本小节将采用参数化的连续时间状态空间模型表示系统输入、隐状态和输出之间的关系：

$$\begin{aligned} \dot{\mathbf{h}}(t) &= d(\mathbf{h}(t), \mathbf{x}(t), \theta_d), \\ \mathbf{y}(t) &= g(\mathbf{h}(t)). \end{aligned} \quad (3-5)$$

状态空间模型中将系统状态表示为定长的编码向量  $\mathbf{h}(t)$ ，该状态表征了系统的长时延特征以及非确定性。将原始的非马尔科夫系统转换为以  $\mathbf{h}(t)$  所在状态空间为核心的马尔科夫系统。

对于长度为  $M$  的待预测序列，本章在整数离散索引  $[0, 1, \dots, M]$  和连续时间范围  $[t_0 \leq t_k \leq t_M]$  之间构造双向映射。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。为了构造可学习的微分系统，本章使用可微 ODE-Net<sup>[17]</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

$$\begin{aligned} L(\mathbf{h}(t_k)) &= L\left(\mathbf{h}(t_0) + \int_{t_0}^{t_k} d(\mathbf{h}(t), \mathbf{x}(t), \theta_d) dt\right) \\ &= L(\text{ODESolve}(\mathbf{h}(t_0), d, t_0, t_k, \theta_d)). \end{aligned} \quad (3-6)$$

本章待求解的常微分方程是参数化的，为了训练参数  $\theta_d$  以最小化  $L(\cdot)$ ，需要根

据上式计算损失函数对模型参数的梯度  $\partial L / \partial \theta_d$ 。此处需要引入伴随状态（Adjoint state），即损失函数对隐状态的梯度  $\alpha(t) = \partial L / \partial h(t)$ 。

利用链式法则可以证明任一时刻的伴随状态可由另一 ODE 描述<sup>[17]</sup>:

$$\frac{d\alpha(t)}{dt} = -\alpha^\top(t) \frac{\partial d(h(t), x(t), \theta_d)}{\partial h(t)}. \quad (3-7)$$

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：

$$\frac{\partial L}{\partial \theta_d} = - \int_{t_0}^{t_M} \alpha^\top(t) \frac{\partial d(h(t), x(t), \theta_d)}{\partial \theta_d} dt. \quad (3-8)$$

在确定的网络结构  $d$  和参数  $\theta_d$  下，任意的 ODE 数值求解方法，如欧拉法（Euler）、中点法（Mid-point）和龙格-库塔法（Runge Kutta），均可同时求解三个微分方程，进而获得  $h(t)$ 、 $\alpha(t)$  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，但方程求解结果的准确性更高。当使用神经 ODE 网络建模时间序列数据集时，该准则同样是成立的。在实验环节，本章3.4节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，

对于给定时刻 ODE-Net 输出的状态  $h(t)$ ，需要使用状态解码器将其转换为系统的预测输出。本章采用的状态解码器为普通的多层感知机网络：

$$\hat{y}(t) = V^\top \tanh(W h_t + b_w) + b_v. \quad (3-9)$$

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。本章中，由于公式 (3-10)中的累加形式使输入  $h(t)$  的范围是无界的，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保证状态解码器的输出在合理的范围内。

### 3.2.4 面向不同预测时长需求的常微分方程导数模块定义

3.2.3节介绍了求解 ODE-Net 和网络参数梯度的方法。在本节中，将更进一步探讨 ODE 网络结构的设计方法。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>[17]</sup>，本章将此模型称为非平稳模型：

$$d(h(t), x(t), \theta_d) = \text{MLP}(h(t), x(t), \theta_d) \quad (3-10)$$

该结构与普通的残差网络（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各个时刻的统计量（如均值和方差）随时间变化的随机过程<sup>[113]</sup>。一般来说，复杂工业系统的输出往往属于非稳定时间序列，带有较强的趋势性。差分操作<sup>[114]</sup> 是一种消除序列趋势和季节性的有效方法。该操作计算原始序列相邻

位置之间的一阶差分或多阶差分，形成的差分序列往往是平稳的，更易于模型学习。在预测阶段，将模型的预测结果进行积分即可复原为原始的非稳定序列。在公式(3-10)中，导数模块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，进而能够建模更高阶的非平稳系统。

然而，非平稳形式(3-10)在处理长期预测任务时面临着严重的问题。在长时区间内求解 ODE-Net 时，对连续时间域内隐状态导数的不断积分会导致隐状态的值域范围显著扩增。假设 ODE-Net 的预测误差服从白噪音分布  $\mathcal{N}(0, \delta^2)$ ，经过时间  $t_M - t_0$  之后的累积误差服从分布  $\mathcal{N}(0, (t_M - t_0)^2 \delta^2)$ 。由此可见，在预测长度足够长时，模型预测的误差将被持续累积，解码器难以准确地估计系统输出。

为了解决上述非稳定结构带来的误差累积问题，本章设计了一种基于平稳系统的导数模块形式，以处理模型的长期预测问题。具体地，其隐状态导数定义为：

$$d(\mathbf{h}(t), \mathbf{x}(t), \theta_d) = \frac{1}{\mu_t} (\text{GRU}(\mathbf{h}(t), \mathbf{x}(t), \theta_d) - \mathbf{h}(t)) \quad (3-11)$$

其中 GRU 表示门控循环单元。对于无穷小时间步  $dt$ ，可得：

$$\lim_{dt \rightarrow 0} \mathbf{h}(t + dt) = (1 - \frac{dt}{\mu_t}) \mathbf{h}(t) + \frac{dt}{\mu_t} \text{GRU}(\mathbf{h}(t), \mathbf{x}(t), \theta_d) \quad (3-12)$$

平稳系统能够根据当前外部输入  $\mathbf{x}(t)$  和隐状态  $\mathbf{h}(t)$  确定隐状态的移动目标点。隐状态  $\mathbf{h}(t)$  将始终朝向 GRU 网络输出的目标值移动，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。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解也在  $(-1, 1)$  中，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，这一特性能够显著提升模型预测的准确性与稳定性。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 网络构造导数模块。

考虑到非稳定系统和稳定系统在不同预测长度下各具优势，本章将非平稳系统应用于短期预测，将平稳系统应用于长期预测。

### 3.3 离散输入序列的可微并行插值方法及模型训练

在求解 ODE-Net 时，需要在连续时间域内给定任意时刻的系统外部输入值  $\mathbf{x}(t)$ ，而训练数据中的外部输入序列  $\mathbf{X}_f^M$  是离散采样的。因此，在网络前向传播之前，需要将离散的外部输入序列转换到连续时间域中。

训练深度神经网络时，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。这要求模型中的所有运算操作能够并行执行。因此，本章基于标准的样条插值算法，实现了一种可微且并行的样条插值

算法，能够将批量输入的离散序列并行地插值为连续时间信号。

不失一般性地，本节假设系统输入的维度  $m$  为 1。给定输入序列的批 (batch) 为： $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^B]$ ，其中  $B$  为批大小。每个向量  $\mathbf{x}^i = [x_1^i, x_2^i, \dots, x_M^i]$  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  $M$  个采样点相对应的常微分方程求解时间区间为  $[0, T]$ 。对于任一给定的时间索引  $t$ ， $0 \leq t \leq T$ ，我们希望并行地估计  $\mathbf{X}(t) = [\mathbf{x}^1(t), \mathbf{x}^2(t), \dots, \mathbf{x}^B(t)]$ 。为了实现这一目标，首先构造从连续时间点与输入序列离散下标之间的转换。定义  $k = \lfloor \frac{tM}{T} \rfloor$ ，给定  $n+1$  个数据点， $\{(k, \mathbf{x}_k), (k+1, \mathbf{x}_{k+1}), \dots, (k+n, \mathbf{x}_{k+n})\}$  可以计算时间  $t$  局部的  $n$  阶样条插值函数的系数矩阵  $\mathbf{A} \in \mathbb{R}^{(n+1) \times B}$ ：

$$\mathbf{A} = \begin{bmatrix} k^0 & \cdots & k^n \\ (k+1)^0 & \cdots & (k+1)^n \\ \vdots & \ddots & \vdots \\ (k+n)^0 & \cdots & (k+n)^n \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{x}_k^1 & \cdots & \mathbf{x}_k^B \\ \mathbf{x}_{k+1}^1 & \cdots & \mathbf{x}_{k+1}^B \\ \vdots & \ddots & \vdots \\ \mathbf{x}_{k+n}^1 & \cdots & \mathbf{x}_{k+n}^B \end{bmatrix}. \quad (3-13)$$

批数据中，原始离散序列在  $t$  时刻的插值为：

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

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：

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

### 3.4 实验验证与分析

本节将采用真实工业系统数据对所提出的基于常微分方程网络的系统建模方法进行验证。实验主要探究三个问题：

- **问题 1：**对于工业系统建模预测问题，相比于直接采用离散时序预测模型，使用连续时间的微分方程网络和高精度 ODE 求解器能否获得更优的效果？
- **问题 2：**在不同预测长度下，使用平稳系统和非平稳系统会对预测效果产生怎样的影响？
- **问题 3：**不同的离散序列插值阶数和不同的序列编码长度会如何影响预测模型的准确性？

本节中，我们将首先介绍数据集、模型超参数以及模型训练和测试时的参数配置，然后围绕上述三个问题进行实验探究。

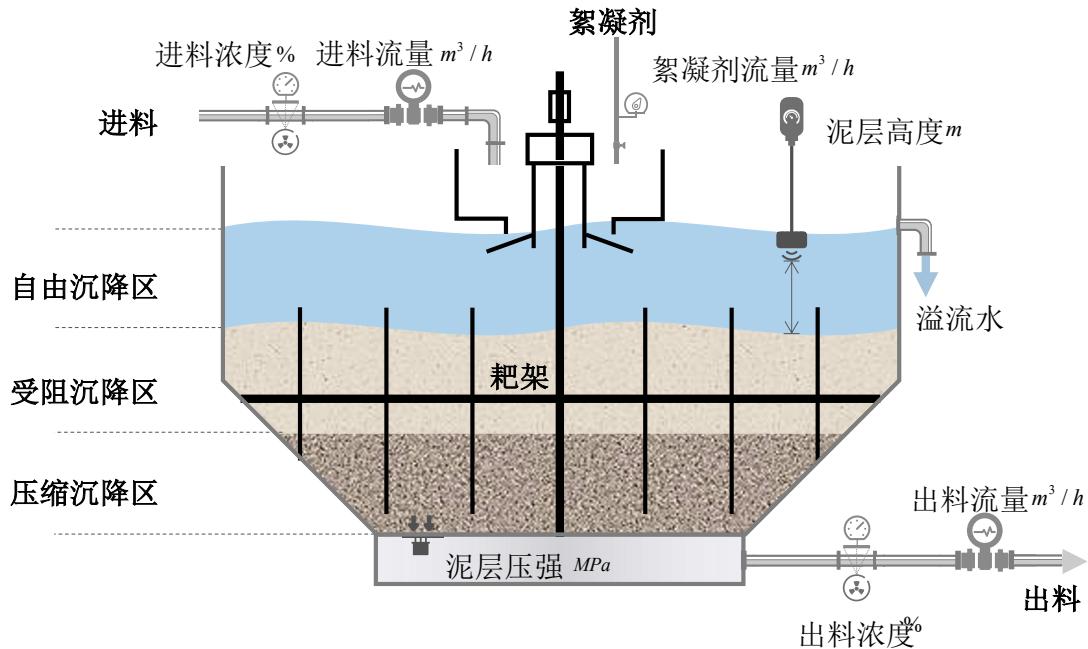


图 3-2 膏体浓密机系统运行过程及监测变量图示

### 3.4.1 膏体浓密机系统数据集

本章所选用的工业系统输入输出数据来自于赞比亚铜带省内某矿业公司的一台膏体浓密机。如图 3-2 所示，该设备由 FLSmidth 公司生产，主要用于将铜尾矿浓缩成高浓度料浆以制备充填膏体。实验所用全部数据采集于 2018 年 5 月-2019 年 2 月之间，共包含 7 个监测项，采集间隔为两分钟。表 3-1 展示了部分数据。

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

其中，系统输出变量  $y(t) \in \mathbb{R}^2$  包括底流浓度  $y_1(t)$  和泥层压力  $y_2(t)$ 。 $y_1(t)$  和  $y_2(t)$  均受到控制输入  $x(t) \in \mathbb{R}^5$  影响，包括进料流量  $u_1(t)$ 、进料浓度  $u_2(t)$ 、耙架转速  $u_3(t)$ 、底流流量  $u_4(t)$  和絮凝剂流量  $u_5(t)$ 。删除系统停机等异常监测数据，累计剩余 24,673 条。

本节采用滑动窗口法生成训练及测试样本对  $(X_p^N, Y_p^N, X_f^M, Y_f^M)$ 。具体地，原始数据集依照浓密机系统的启停时间，被划分为多个文件。本节使用前 70% 的文件用于模型训练。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。每个文件包含了 7 个传感器在一段

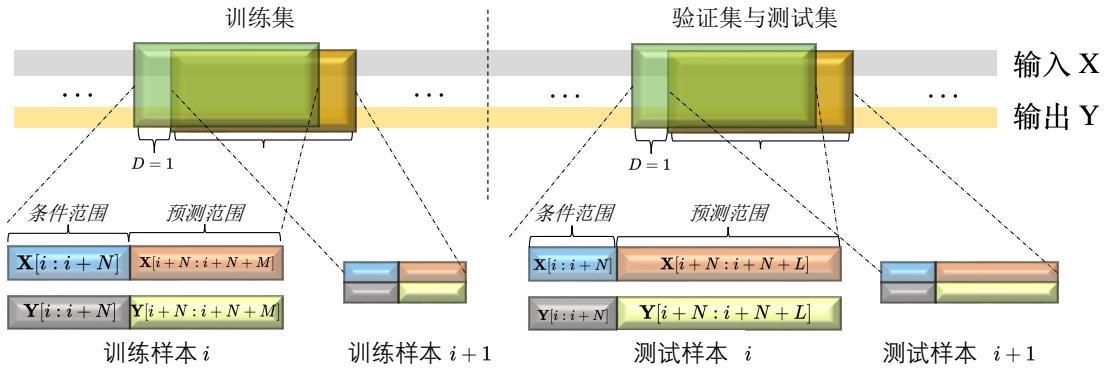


图 3-3 训练集、验证集、测试集的构建过程图示

连续时间内的监测序列，且各序列中数据点的采样间隔均为 2 分钟。在生成训练集和验证集时，对每个文件中的多个序列，构建大小为  $N + M$  的滑动窗口沿时间维度正向移动，移动步长为 1。当窗口到达位置  $i$  时，采集四个序列  $\mathbf{X}_p^N = \mathbf{X}[i:i+N]$ ,  $\mathbf{Y}_p^N = \mathbf{Y}[i:i+N]$ ,  $\mathbf{X}_f^M = \mathbf{X}[i+N:i+N+M]$ ,  $\mathbf{Y}_f^M = \mathbf{Y}[i+N:i+N+M]$  作为训练和验证样本。

在生成测试集数据时，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预测序列的长度  $M$ ）。相应地，在位置  $i$  采集的四个序列为  $\mathbf{X}_p^N = \mathbf{X}[i:i+N]$ ,  $\mathbf{Y}_p^N = \mathbf{Y}[i:i+N]$ ,  $\mathbf{X}_f^L = \mathbf{X}[i+N:i+N+L]$ ,  $\mathbf{Y}_f^L = \mathbf{Y}[i+N:i+N+L]$  作为测试样本。使用不同的  $L$  将生成不同的测试集。

构造数据集的详细过程如图 3-3 所示。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。模型仅利用训练集进行训练，验证集上表现最好的模型将被保留，并评估其在三个测试集上的表现。根据图 3-3 所示的构建输入-输出序列对的方法，共计 17,131 个样本用于训练。对于不同预测长度  $L = 60, 200, 500$ ，分别有 3561、3421、3121 个样本用于测试。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 实验参数及评价指标

实验中，使用 mini-batch 随机梯度下降（SGD）和 Adam 优化器<sup>[115]</sup> 对模型进行训练。批大小为 512，初始学习率为 0.001 且按衰减率 0.95 指数衰减。衰减周期为 10 个训练轮次。ODE-Net 中隐藏状态  $h(t)$  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（与  $h(t)$  大小一致）。状态解码器中隐藏层的大小为 64。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  $[0, 1, \dots, M]$ ，本章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，式(3-11)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

型将等价于直接使用 GRU 网络在离散时间系统下进行预测:

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

本节使用底流浓度预测值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。RRSE 是被归一化的均方根 (RMS) 误差。对于预测长度  $L$ , RRSE 定义如下:

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

### 3.4.3 不同模型预测结果对比

首先, 本节研究了不同 ODE 求解器和导数模块对预测精度的影响。对比的 ODE 求解器包括五种: 欧拉法 (Euler)、中值法 (Midpoint)、四阶龙格库塔 (Runge—Kutta, RK4)、Dormand—Prince (Dopri5)<sup>[17]</sup>、以及三阶 Bogacki—Shampine (Bosh)<sup>[116]</sup>。为了平衡时间消耗和预测精度, 实验中所有自适应求解器的相对误差限设置为  $1e - 4$ , 绝对误差限设置为  $1e - 5$ 。本节研究了上述微分方程求解器在非稳定和稳定系统两种情况下的求解性能。此外, 我们还评估了部分离散时间深度序列模型的预测效果, 包括深度状态空间模型 (DT-State-Space)、基于注意力机制的 Seq2Seq 模型 (Attention-Sq2Seq)<sup>[101]</sup> 和 Transformer 模型<sup>[117]</sup>。DT-State-Space<sup>[118]</sup> 模型采用循环神经网络 (RNN) 建模线性状态空间模型的动态参数, 并利用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-Sq2Seq 的超参数设置与原文献保持一致。

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	时间 (s)	RRSE	MSE	时间 (s)	RRSE	MSE	时间 (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-Sq2Seq <sup>[101]</sup>		3.13	8.97	0.41	4.02	33.90	0.41	1.82	40.53	0.42
DT-State-Space <sup>[118]</sup>		3.22	9.36	0.06	4.69	41.11	0.07	3.35	45.64	0.08
Transformer <sup>[117]</sup>		3.16	8.36	0.02	3.99	40.23	0.02	2.55	44.23	0.03

本节共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 和预测时间。从表 3-2 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 求解器时的预测效果，但差于使用高阶 ODE 求解器的预测效果，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，从而相比于离散时间模型具有更高的预测精度。另外，Transformer、Seq2Seq 模型侧重于提取长时关联信息以进行序列预测。而本文所关注的是系统辨识问题，模型需要及时提取来自系统输入的信息以估计系统的当前输出。纯时间序列预测模型只能有效利用  $\mathbf{X}_p^N$  和  $\mathbf{X}_p^N$  中的信息而难以充分  $\mathbf{X}_f^M$  的实时信息。因此，在长序列预测时，伴随着历史序列对于未来系统长期输出的影响逐渐变小，Transformer、Seq2Seq 模型精度明显变差。

### 3.4.4 不同 ODE 求解器下稳定系统以及非稳定系统对比

表 3-2 分别评估了稳定系统模型和非稳定系统模型在不同 ODE 求解器下的预测误差。导数模块为非稳定系统时，对于短期预测任务 ( $L = 60$ )，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 较高，预测精度差于其他四个 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调用一次导数模块以计算隐状态的单步差分。其运行本质等同于离散时间序列模型，因为并没有充分利用模型为连续时间系统的性质，所得 ODE 方程的解的精度较差。相应地，中值法和 4 阶龙格库塔法在两个相邻时间点之间分别调用了导数模块两次和四次。因此，这两种方法对隐状态轨迹的求解精度高于欧拉法，预测精度优于基于 Euler 方法的离散时间模型。

另外，Dopri5 和 Bosh 方法作为自适应步长的求解器，能够确保数值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 的耗时较长，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。表 3-2 中没有列出 Dopri5 和 Bosh 在导数模块定义为稳定系统时的预测精度，因为其计算速度极慢，无法在限定时间内给出 ODE 的解。求解和训练 ODE 所需的时间会大幅增加，其原因是相比于非稳定系统，稳定系统会使得隐状态的轨迹波动更加剧烈。本质上，稳定系统中的  $\text{GRU}(\mathbf{h}(t), \mathbf{x}(t), \theta_d)$  根据  $\mathbf{x}(t)$  定义了目标状态  $\mathbf{h}^\infty(t)$ ，满足  $\text{GRU}(\mathbf{h}^\infty(t), \mathbf{x}(t), \theta_d) = \mathbf{h}^\infty(t)$ 。隐状态沿着方向  $\overrightarrow{\mathbf{h}^\infty - \mathbf{h}(t)}$  向目标状态移动。由于数据集中外部输入  $\mathbf{x}(t)$  是持续变化且不稳定的，因此会导致  $\mathbf{h}^\infty(t)$  剧烈变化。当采用 Euler、rk4 等导数访问次数固定的求解器时，使用固定步长  $\Delta t$  无法在  $\mathbf{x}(t)$  持续变化的情况下精确地求解系统轨迹。自适应步长求解器 Dopri5、Bosh 为了满足给定的误差限要求，不得不急剧减小  $\Delta t$  以求解精确的隐状态轨迹，因此使求解 ODE 所需的时间剧烈增加。

总体来看，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。虽然本章所提出的连续时间 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>[56,57,119]</sup> 聚焦于优化 ODE-Net 的训练或推理速度。这些方法为提高本章模型的预测效率提供了有借鉴意义的指导。

为了更直观地比较稳定系统和非平稳系统在长短期预测时的区别，本节对预测结果进行了可视化分析。图3-4描述了在  $L = 60$  的短期预测任务中，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：

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，预测得到的序列相比平稳模型更接近实际系统输出。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。该约束符合浓密机系统运行缓慢的特性，等价于减小模型参数的搜索空间，抑制模型过拟合的情况。图3-5展示了在长期预测任务 ( $L = 200$ ) 中的模型预测结果。图中呈现的预测效果说明稳定型的导数模块在长序列预测时的表现更优，与表3-2中的结果几乎一致。而非平稳模型在长期预测任务中的 RRSE 和 MSE 远远高于平稳模型，模型预测效果较差。从图3-4可以看出，非平稳模型在短期预测问题中展示了更好的预测效果。而在图3-5(a)中，非稳定系统在长期预测时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较为稳定的，更接近于系统的真实输出，证明稳定系统在长期预测中准确度更高。在由非稳定系统定义的导数模块中，其内部结构导致隐状态在积分过程中是无约束的，状态的取值范围将逐渐扩张。虽然本章在状态解码器网络中嵌入  $\tanh$  函数，能够将预测的系统输出限制在合理的范围内，但解码器模块难以学习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，说明微分方程求解的精度会影响模型预测的精度，结果与表3-2一致。

本节额外进行了数组实验，以评估在其他预测长度  $L$  下，稳定系统和非稳定系统预测的底流浓度误差 ( $\log_{10} \text{MSE}$ )。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 ( $\mu \pm 2\sigma$ )。虽然在短期预测任务中非稳定系统精度优于稳定系统，但当  $L$  超过 120 时，稳定系统表现更优。

### 3.4.5 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长时延系统预测问题的有效性。具体地，该节对比了没有序列编码器以及引入序列编码器并将编码长度  $N$  设置为不同值时，模型预测精度的变化情况。特别地，当  $N$  设置为 1 时，将序列编码器替换为具有单一隐藏层的神经网络，该网络将单步系统输出  $\mathbf{y}(k-1)$  及输入  $\mathbf{x}(k-1)$  编码为 ODE 系统的初始隐状态  $\mathbf{h}(t_0)$ 。当  $N$  为 0 时，初始状态  $\mathbf{h}(t_0)$  被设定为可

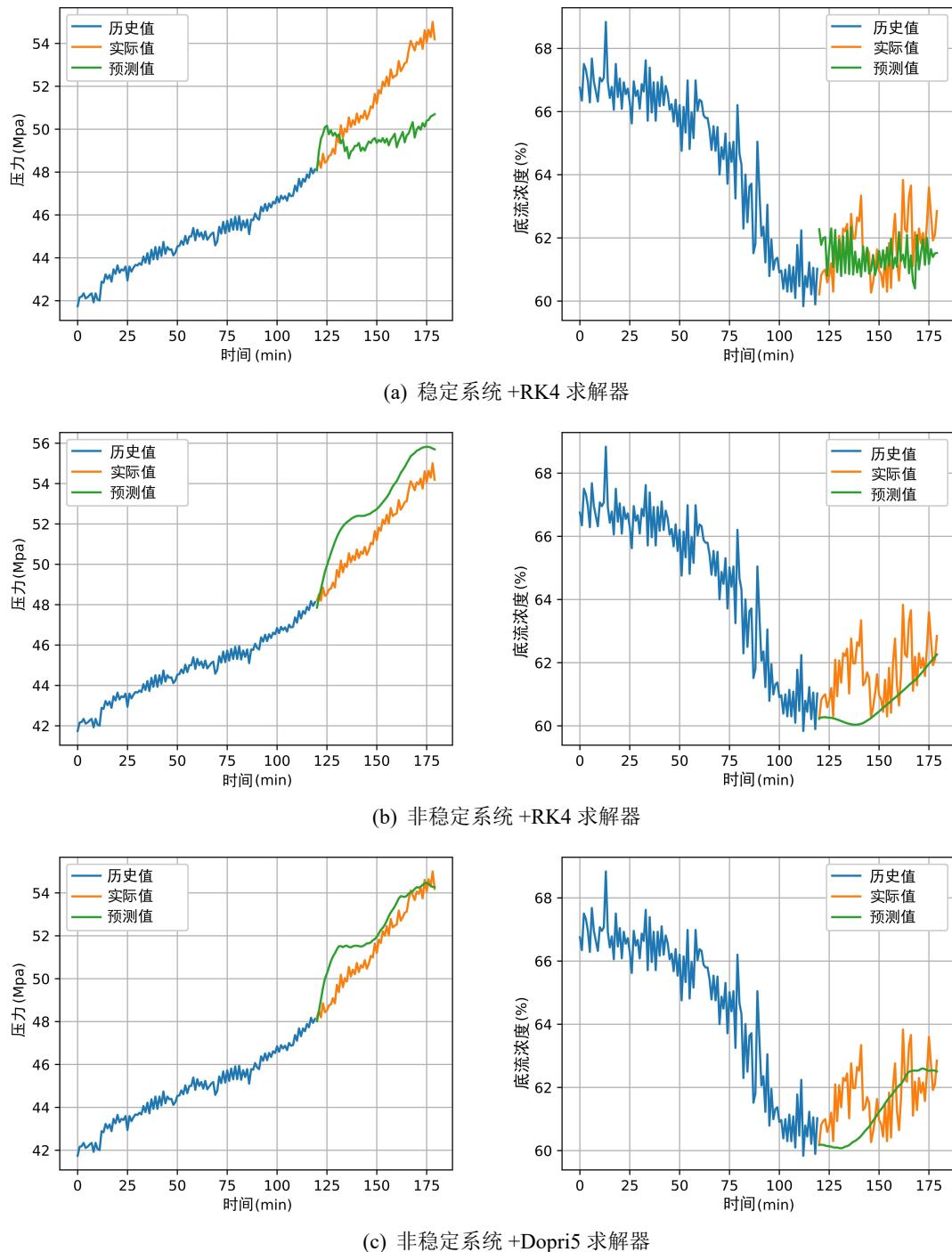
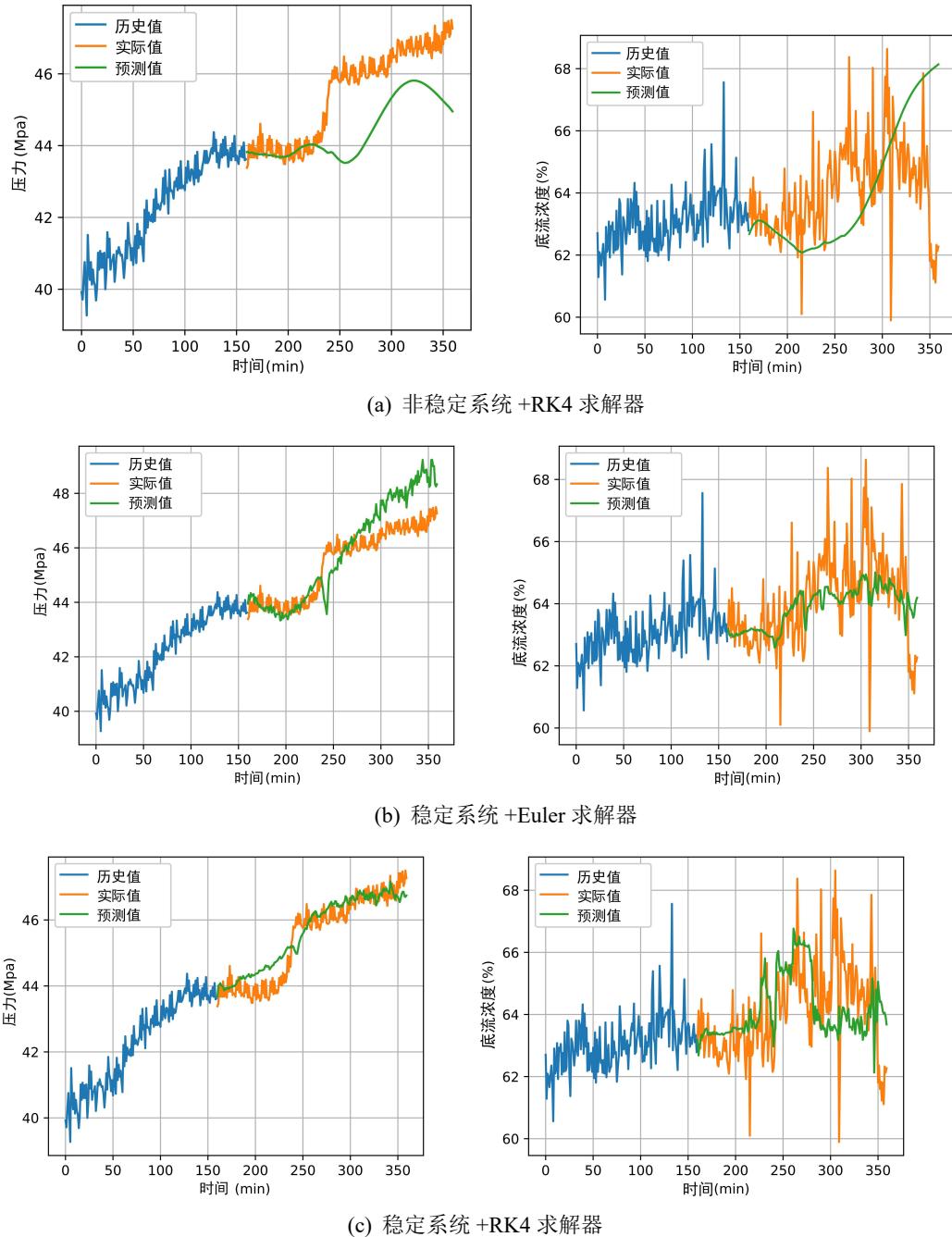


图 3-4 不同系统及不同 ODE 求解器在  $L = 60$  短期预测任务中的性能比较

图 3-5  $L = 200$  时不同 ODE 求解器、不同导数模块下的预测效果对比

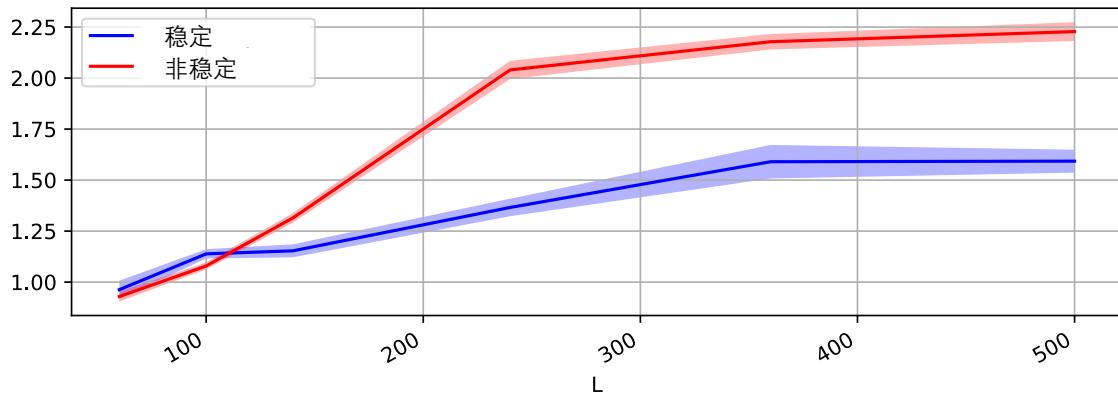


图 3-6 不同预测长度下稳定系统和非稳定系统的预测精度变化

表 3-3 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

学习的初始隐状态<sup>[109]</sup>或零向量，与历史系统轨迹无关。

该实验仍然在  $L = 60$ 、200 和 500 三种情况下比较不同初态估计方法对预测精度的影响。在  $L = 60$  的实验中，导数模块被定义为非平稳系统。当  $L = 200$  和 500 时，导数模块定义为带有 GRU 单元的稳定系统。六组实验中，均采用四阶龙格-库塔求解器求解 ODE 方程。

从表3-3中的结果可以看出，相比于完全忽略历史系统输出 ( $N = 0$ )，引入序列编码器并利用历史运行轨迹构建常微分方程的初态能够获得更好的预测精度。直觉上的解释为，待预测的系统未来输出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具有重要意义。利用序列编码器能够将此部分相关性嵌入在 ODE 初态中。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。

另外，引入序列编码模块的收益在短期预测任务更加明显。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变弱。

### 3.4.6 探究序列插值阶数的对预测精度的影响

本小节进行消融实验以探究不同离散输入序列插值方法对预测精度的影响。测试对象包括四种不同阶数的样条插值方法，最终比较模型在不同预测长度下的预测精度。预测长度设置、导数形式、以及 ODE 求解器选择均与实验3.4.5一致。

表 3-4 所示结果表明，采用高阶样条插值时的精度将稍优于低阶插值。该结果说明

表 3-4 不同序列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

本章所研究的膏体浓密机系统为复杂非线性受控系统，受控输入信息对于预测系统输出是十分重要的。高阶样条插值方法能够更充分地利用相邻位置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。

### 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，该模型由序列编码器、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。本章采用膏体浓密机系统运行数据集进行实验，以探究导数模块的不同定义方式、ODE 求解器选择、离散序列插值阶数、序列编码长度对于预测精度的影响。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。在实际应用中，当构建基于模型的反馈控制器时，可使用非平稳系统作为短期预测模型辅助控制序列优化。但是，非平稳模型在长期预测时隐状态的波动范围过大且存在较大的累积误差，而平稳系统通过修改导数模块结构，避免了隐状态漂移问题，因此在长期预测中表现较好。当需要具有稳定鲁棒的系统识别模型进行长期预测时，如系统仿真或控制器测试，平稳系统模型是更好的选择。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能够有效改善模型精度。



## 4 随机性系统建模与预测

基于深度学习的系统识别方法充分利用了深度神经网络在建模非线性复杂函数方面的优势，实现利用系统离线输入输出数据识别参数结构未知的黑盒系统动态。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难题：

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，仅适用于对确定性动态系统进行建模，无法对不完全观测造成的系统随机性进行建模。

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>[18]</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，进而满足训练需求。但是在模型推理阶段，仍要求推理场景下的数据采样间隔和训练数据的采样间隔保持一致，因此在实时工业应用中模型应具备非均匀采样数据处理能力。

第三，在某些应用场景下，训练完成的识别模型需要在线地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>[78]</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>[54,75]</sup>，而需要选择具有循环状态更新能力的模型。

现有系统建模方法往往只能解决上述问题中的一个或几个。目前尚没有某一模型能够同时解决上述所有问题。因此，本章提出常微分方程循环状态空间模型（Ordinary Differential Equation Recurrent State Space Model, ODE-RSSM），该模型为定义在连续时间（Continuous time）域中的随机马尔科夫模型。ODE-RSSM 建模了系统状态在连续时间域下的随机演化，进而能够从采样间隔不均匀的样本数据中学习具有随机性和长时滞特性的动态系统。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，以建模任意采样间隔间的状态演化。另外，在训练阶段，由于非均匀采样间隔下同一数据批中求解时间点无法对齐、积分区间彼此不一致，现有的常微分方程求解器难以实现批量并行化推理与训练。因此本章提出了一种批常微分方程并行求解方法，该方法利用定积分区间的线性变换等价性对齐各常微分方程的求解积分区间，进而构造辅助的初值问题（Initial Value Problem, IVP）并利用现有的深度微分方程求解器高效地求解。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，能够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。相比于其他基于神经网络的动态系统建模方法，本章提出模型的优势如表4-1所示。

在实验环节，本章使用三个系统辨识数据集，包括两个公共数据集和第三章使用的膏体浓密机系统数据集，评估了所提出模型在解决长期开环预测问题时的

表 4-1 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 <sup>[22]</sup>	✗	✓	✓
ODE-RNN <sup>[54]</sup> 、SNODE <sup>[23]</sup>	✗	✓	✓
Latent ODE <sup>[20]</sup>	✓	✓	✗
<b>ODE-RSSM</b>	✓	✓	✓

性能。实验结果表明，对于非均匀采样下的随机系统建模任务，ODE-RSSM 无论在处理非均匀间隔还是建模系统随机性方面均具有良好表现，其预测精度显著优于其他离散时间状态空间模型以及带有确定性隐空间状态的连续时间辨识模型。

本章工作的核心贡献总结如下：

- 本章提出深度状态空间模型的扩展 ODE-RSSM，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。
- 为了改善模型对于长时延相关性的提取能力以及多步预测效果，本章在 ODE-RSSM 的训练中引入了采样状态重用的隐空间超调技术，在时序变分自编码机训练中实现了低时耗的沿时间反向传播。
- 为了优化模型在非均匀采样间隔下的训练速度，本章提出了一种常微分方程的重参数化方法，解决了批中不同方程之间积分区间间隔不一致时，难以并行求解 ODE 的问题，有效提升了模型训练效率。
- 本章使用深锥浓密机系统数据集以及两个公有输入输出系统辨识数据集验证了 ODE-RSSM 模型的预测效果以及上述训练改进及重参数化方法对于改进模型多步预测精度、优化训练效率的有效性。

本章的结构组织如下：4.1节对非均匀采样下随机性系统的建模问题给出形式化描述。4.2节详细介绍了 ODE-RSSM 中隐变量的生成过程、推理过程以及训练方法。4.3节详细介绍了基于隐空间超调的模型多步预测性能改进方法。4.4节详细介绍了在非均匀采样间隔下，用于并行求解批常微分方程的重参数化方法。4.5节首先使用三个数据集评估了 ODE-RSSM 与其他对比模型在不同采样间隔分布下的辨识效果，并通过消融实验分析了隐空间超调技术对于模型长期预测的改善效果，最后验证了并行求解批常微分方程方法相比于现有求解方案的时间优越性。最后，4.6节对本章研究工作做出总结，并对未来研究方向做出展望。

## 4.1 问题形式化描述

本章将具有随机性的系统表示为包含随机隐变量的受控状态空间模型。定义被辨识系统的非均匀采样输入序列为  $\{\mathbf{u}_{t_i}\}_{i=1}^N$ ，系统序列输出为  $\{\mathbf{y}_{t_i}\}_{i=1}^N$ ，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。本章假设对于各维输入和各维

输出的采样都是同步的，仅采样间隔不均匀。对于系统的随机性状态转移引入随机隐变量  $s_{t_i}$  进行描述，该隐变量在相邻采样时刻之间的转移过程是非确定的。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。在给定系统的受控输入下，系统输出来自于含有时序隐变量的条件生成过程。给定生成模型参数  $\theta$ ，定义序列隐变量和观测值的联合概率如下：

$$p_\theta(\mathbf{y}_{t_1:t_N}, \mathbf{s}_{t_1:t_N}, | \mathbf{u}_{t_1:t_N}, \mathbf{s}_{t_0}) = \prod_{i=1}^N p_\theta(\mathbf{y}_{t_i} | \mathbf{s}_{t_i}) p_\theta(\mathbf{s}_{t_i} | \mathbf{u}_{t_{i-1}}, \mathbf{s}_{t_{i-1}}) \quad (4-1)$$

## 4.2 常微分方程网络-循环状态空间模型

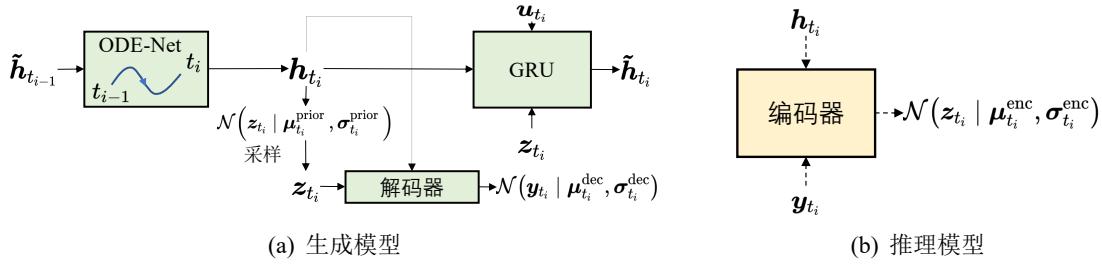


图 4-1 ODE-RSSM 中的生成过程和推理过程

本节首先介绍 ODE-RSSM 中隐变量的生成模型和推理模型，模型假设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，即  $\mathbf{s}_{t_i} = [\mathbf{z}_{t_i}, \mathbf{h}_{t_i}]$ 。如图 4-1 所示。在生成模型中，模型通过组合门控循环单元（Gate Recurrent Unit, GRU）和 ODE-Net 建模隐状态在非均匀间隔控制输入下的转移过程。为了训练生成模型参数  $\theta$ ，本章引入用于近似后验推理的编码器  $q_\phi(\mathbf{z}_{t_i} | \mathbf{y}_{t_i}, \mathbf{h}_{t_i})$  对隐变量的真实后验分布进行近似，并采用最大化证据下限（Evidence Lower Bound, ELBO）的方式协同训练参数  $\phi$  和  $\theta$ 。

### 4.2.1 带有随机隐变量的连续时间的生成模型

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序列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  $\mathbf{s}_{t_i}$  的条件先验分布定义如下：

$$p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-1}}, \mathbf{u}_{t_{i-1}}) = p(\mathbf{z}_{t_i} | \mathbf{h}_{t_i}) * p(\mathbf{h}_{t_i} | \mathbf{s}_{t_{i-1}}, \mathbf{u}_{t_{i-1}}) \quad (4-2)$$

在式 (4-2) 中， $p(\mathbf{h}_{t_i} | \mathbf{s}_{t_{i-1}}, \mathbf{u}_{t_{i-1}})$  为德尓塔  $\delta$  分布，在给定  $t_{i-1}$  时刻的隐变量  $\mathbf{s}_{t_{i-1}}$  和系统输入  $\mathbf{u}_{t_{i-1}}$  下， $\mathbf{h}_{t_i}$  是确定的。

本章中不同时刻  $t_i$  与上一时刻之间的采样间隔  $t_i - t_{i-1}$  不是恒等的，一般的离散时间序列模型不适用于建模可变时间间隔下的隐状态演化。因此，本章受常

微分方程网络模型<sup>[17,54]</sup>的启发，引入 ODE-Net 学习确定性隐状态  $\mathbf{h}_{t_i}$  在相邻采样时间点之间的演化。

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

$$\tilde{\mathbf{h}}_{t_{i-1}} = \text{GRU}([\mathbf{u}_{t_{i-1}}, \mathbf{z}_{t_{i-1}}], \mathbf{h}_{t_{i-1}}) \quad (4-3)$$

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，求解 ODE-Net 以预测时间点  $t_i$  处的更新状态  $\mathbf{h}_{t_i}$ ：

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

积分时间范围内的系统输入被定义为分段恒定函数  $\mathbf{u}[t_{i-1}, t_i] = \mathbf{u}_{t_{i-1}}$ ，将其作为 ODE 的输入以控制隐状态对时间的导数，进而实现输入输出系统辨识的目的。在线预测场景下，上述定义方式能够很好地适配控制信号  $\mathbf{u}_{t_i}$  是间断采集的情况。在上述生成过程中， $\mathbf{h}_{t_{i-1}}$  表示在获得  $\mathbf{u}_{t_{i-1}}$  之前， $t_{i-1}$  时刻的确定性隐状态的预测结果。而  $\tilde{\mathbf{h}}_{t_{i-1}}$  表示在给定  $\mathbf{z}_{t_{i-1}}$  和  $\mathbf{u}_{t_{i-1}}$  的信息之后更新的隐状态。 $\mathbf{h}_{t_i}$  表示模型不知道  $t_i$  时刻的外部输入下，模型自回归预测的确定性隐变量在  $t_i$  时刻的取值。

一般的常微分方程网络采用多层感知机建模导函数。本文第三章研究表明，对导数进行积分会导致隐状态范围不断扩张且会增加累积误差。因此增量式的 ODE 网络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网络定义方法<sup>[120]</sup>：

$$\frac{d\mathbf{h}(t)}{dt} = NN_\theta(\mathbf{h}(t)) - \frac{\mathbf{h}(t)}{|\mathbf{h}(t)|} * \frac{\langle NN_\theta(\mathbf{h}(t)), \mathbf{h}(t) \rangle}{|\mathbf{h}(t)|}, \quad (4-5)$$

其中， $NN_\theta(\cdot)$  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 单元之间的区别如图4-2所示。正交形式保证了导数预测值  $\frac{d\mathbf{h}(t)}{dt}$  始终与当前状态  $\mathbf{h}(t)$  正交。因此，从  $t_{i-1}$  到  $t_i$  的积分过程中，只会改变  $\mathbf{h}(t)$  的方向，其范数不变。 $|\mathbf{h}(t)|$  仅由每次 GRU 网络更新过程决定，且被严格限制在  $[-1, 1]$ 。这种导函数网络结构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。

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

$$p_\theta(\mathbf{z}_{t_i} | \mathbf{h}_{t_i}) = \mathcal{N}(\mathbf{z}_{t_i} | \boldsymbol{\mu}_{t_i}^{\text{prior}}, \boldsymbol{\sigma}_{t_i}^{\text{prior}}) \quad (4-6)$$

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

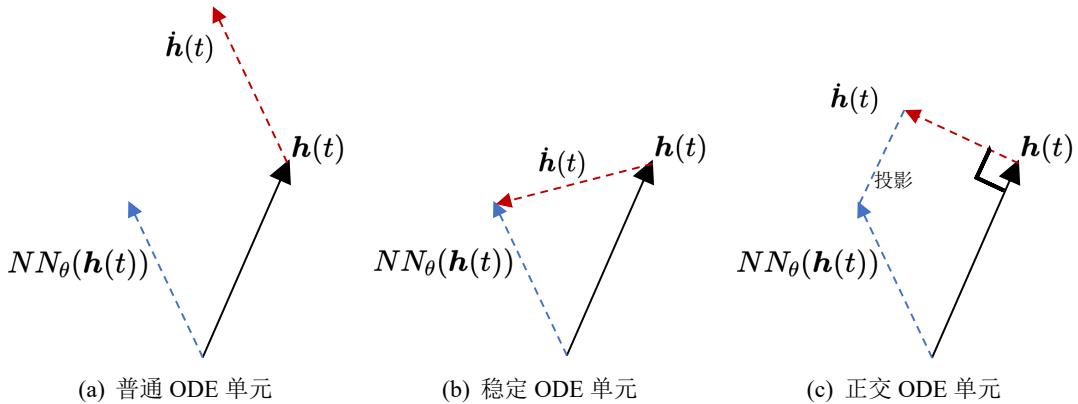


图 4-2 三种导数模块定义方法图示

状态  $\mathbf{h}_{t_i}$ 。

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

$$p(\mathbf{y}_{t_i} | \mathbf{h}_{t_i}, \mathbf{z}_{t_i}) = \mathcal{N}(\mathbf{y}_{t_i} | \boldsymbol{\mu}_{t_i}^{\text{dec}}, \boldsymbol{\sigma}_{t_i}^{\text{dec}}) \quad (4-7)$$

分布参数  $\boldsymbol{\mu}_{t_i}^{\text{dec}}, \boldsymbol{\sigma}_{t_i}^{\text{dec}} = \text{NN}_{\theta_d}(\mathbf{h}_{t_i}, \mathbf{z}_{t_i})$  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下，通过对序列隐状态分布进行反复预测与采样，模型可以开环地预测序列隐变量并预测系统输出。

#### 4.2.2 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，因此无法采用对隐变量概率分布进行积分的方式估计精确的对数似然  $\mathcal{L}$ 。本章引入变分分布  $q_\phi$  近似随机隐变量的后验分布，进而辅助生成模型的学习。

$$\begin{aligned} q_\phi(\mathbf{z}_{t_1:t_N} | \mathbf{y}_{t_1:t_N}, \mathbf{u}_{t_1:t_N}) &= \prod_{i=1}^N q_\phi(\mathbf{z}_{t_i} | \mathbf{h}_{t_i}, \mathbf{y}_{t_i}) \\ &= \prod_{i=1}^N \mathcal{N}(\mathbf{z}_{t_i} | \boldsymbol{\mu}_{t_i}^{\text{enc}}, \boldsymbol{\sigma}_{t_i}^{\text{enc}}) \end{aligned} \quad (4-8)$$

如图4-1(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  $\boldsymbol{\sigma}_{t_i}^{\text{enc}}$  由基于 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，可以通过最大化系统观测证据下界的方式训练 ODE-RSSM 模型：

$$\begin{aligned}
\ln p(\mathbf{y}_{t_1:t_N}) &= \ln \int \prod_{i=1}^N p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-1}}, \mathbf{u}_{t_{i-1}}) p(\mathbf{y}_{t_i} | \mathbf{s}_{t_i}) d\mathbf{s}_{t_1:t_N} \\
&\geq \sum_{i=1}^N \left( \mathbb{E}_{q(\mathbf{s}_{t_i})} [\ln p(\mathbf{y}_{t_i} | \mathbf{s}_{t_i})] \right. \\
&\quad \left. - \mathbb{E}_{q(\mathbf{s}_{t_i})} [\text{KL}[q(\mathbf{s}_{t_i}) \| p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-1}}, \mathbf{u}_{t_{i-1}})]] \right)
\end{aligned} \tag{4-9}$$

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

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外，推理模型还可作为序列编码模型用于在给定输入输出下对系统状态进行编码。作为支持状态循环更新的编码器-解码器框架，ODE-RSSM 可以灵活地采用推理模型进行隐变量推断并采用生成模型进行在线预测。在一些具有流式数据的在线任务中，两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后验分布并采样，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。

### 4.3 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>。对于 ODE-RSSM，需要重复地调用生成模型进行隐变量的单步预测，并从预测分布中采样得到隐状态，将其作为下一步预测的输入。然而，对于式(4-9)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，在训练过程中，生成模型预测时的隐变量输入始终来自于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采样于生成模型给出的预测分布。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的隐变量输入在训练集和测试集上的同分布假设，这会导致多步预测的准确性较差<sup>[121]</sup>。对于观测空间不完备的随机性复杂系统，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>以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：

$$\begin{aligned}
p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-d}}) &\triangleq \int \prod_{\tau=i-d+1}^i p(\mathbf{s}_{t_\tau} | \mathbf{s}_{t_{\tau-1}}, \mathbf{u}_{t_{\tau-1}}) d\mathbf{s}_{t_{i-d+1}:t_{i-1}} \\
&= \mathbb{E}_{p(\mathbf{s}_{t_{i-1}} | \mathbf{s}_{t_{i-d}})} [p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-1}})]
\end{aligned} \tag{4-10}$$

想要精确地求解  $\mathbf{s}_{t_{i-d}}$  下  $\mathbf{s}_{t_i}$  的多步预测分布，常规方法是对所有中间隐变量的概率分布进行积分，但在实际应用中这是极为困难的。对于多步预测，可以采用多次祖先采样（Ancestral sampling）和重参数化技巧（Reparameterization Trick）不断地对单步预测分布给出的中间状态进行采样。最终  $\mathbf{s}_{t_i}$  的采样分布将服从多步预测的

边缘分布。

基于式(4-10)定义的  $d$ -步预测先验分布, 进而可以推导出定义在多步预测先验分布上的 ELBO:

$$\begin{aligned} \ln p_d(\mathbf{y}_{t_1:t_N}) &= \ln \int \prod_{i=1}^N p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-d}}) p(\mathbf{y}_{t_i} | \mathbf{s}_{t_i}) d\mathbf{z}_{t_1:t_N} \\ &\geq \sum_{i=1}^N \left( \mathbb{E}_{q(\mathbf{s}_{t_i})} [\ln p(\mathbf{y}_{t_i} | \mathbf{s}_{t_i})] \right. \\ &\quad \left. - \mathbb{E} [\text{KL}[q(\mathbf{s}_{t_i}) \| p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-1}}, \mathbf{u}_{t_{i-1}})]]) \right. \\ &\quad \left. p(\mathbf{s}_{t_{i-1}} | \mathbf{s}_{t_{i-d}}) q(\mathbf{s}_{t_{i-d}}) \right) \end{aligned} \quad (4-11)$$

由前人研究<sup>[76]</sup>给出的证明,  $\ln p_d(\mathbf{y}_{t_1:t_N})$  是式(4-9)中单步预测 ELBO 的下界, 自然也是原始系统输出对数似然的下界。最大化  $\ln p_d(\mathbf{y}_{t_1:t_N})$  能够起到增大输出对数似然的目的。给定超参数  $D$ , 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:

$$\begin{aligned} \text{ELBO-D} &= \frac{1}{D} \sum_{d=1}^D \ln p_d(\mathbf{y}_{t_1:t_N}) \\ &\geq \mathbb{E}_{q(\mathbf{s}_{t_1:t_N})} [\ln p(\mathbf{y}_{t_1:t_N} | \mathbf{s}_{t_1:t_N})] - \frac{1}{D} * \text{KL-D} \end{aligned} \quad (4-12)$$

其中

$$\text{KL-D} = \sum_{d=1}^D \sum_{i=1}^N \mathbb{E} [\text{KL}[q(\mathbf{s}_{t_i}) \| p(\mathbf{s}_{t_i} | \mathbf{s}_{t_{i-1}})]]) p(\mathbf{s}_{t_{i-1}} | \mathbf{s}_{t_{i-d}}) q(\mathbf{s}_{t_{i-d}}) \quad (4-13)$$

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

利用多步证据下界训练时序变分自编码机模型能够在一定程度上改善模型的长期开环预测精度<sup>[76]</sup>。然而, 依照公式(4-13)直接估计 KL-D 的时间复杂度为  $\mathcal{O}(N * D^2)$ , 除了对序列不同位置的计算循环  $1 \leq d \leq D$  可以并行化处理, 还包括对于最大距离  $D$  的外循环  $1 \leq d \leq D$ , 以及式(4-10)所示的多步预测采样  $p(\mathbf{s}_{t_{i-1}} | \mathbf{s}_{t_{i-d}})$ , 其时间复杂度为  $\mathcal{O}(d)$ 。

本章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, 使估计 KL-D 的时间复杂度变为  $\mathcal{O}(N * D)$ 。通过分析式(4-10)可以得知, 对于不同的  $i, d$ , 在  $i - d$  相同的情况下, 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, 对于超调距离  $d$  不同但后验推理位置  $i - d$  相同的多步预测, 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，将 KL-D 的计算公式重写为：

$$\begin{aligned} \text{KL-D} &= \sum_{i=1}^N \sum_{d=1}^D \mathbb{E} [\text{KL}[q(s_{t_{i+d}}) \| p(s_{t_{i+d}} | s_{t_{i+d-1}})]] \\ &\quad p(s_{t_{i+d-1}} | s_{t_i}) q(s_{t_i}) \\ &= \sum_{i=1}^N \mathbb{E}_{\hat{s}_{t_i} \sim q(s_{t_i})} \mathbb{E}_{p(\hat{s}_{t_{i+1:t_{i+D-1}} | \hat{s}_{t_i}})} \sum_{d=1}^D \text{KL}(i+d) \end{aligned} \quad (4-14)$$

其中  $\text{KL}(i+d)$  为  $\text{KL}[q(s_{t_{i+d}}) \| p(s_{t_{i+d}} | \hat{s}_{t_{i+d-1}})]$  的缩写。 $\hat{s}_{t_i}$  是从后验推理分布中采样的初始状态， $\hat{s}_{t_{i+1:t_{i+D-1}}}$  为采用祖先采样法逐步预测采样得到的序列隐状态。改进后的采样方法能够获得 KL-D 的无偏估计，且时间复杂度仅为  $\mathcal{O}(N \times D)$ 。隐状态采样和计算 KL 散度的过程如图4-3所示。

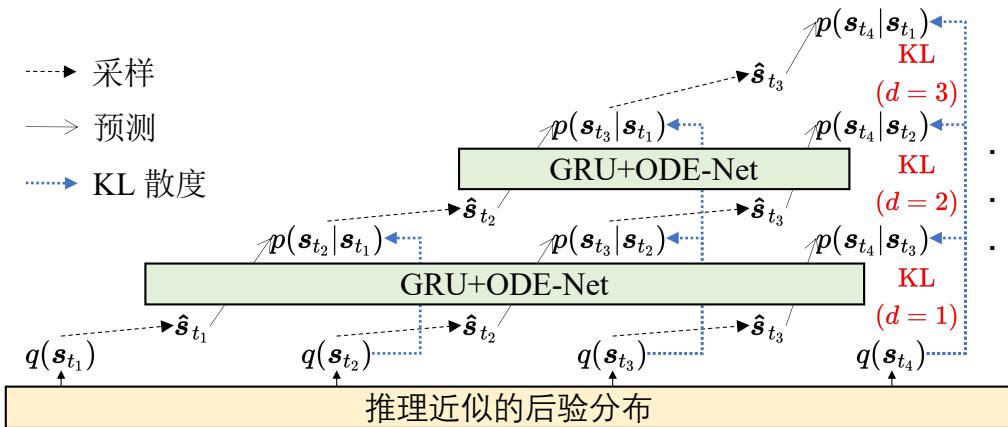


图 4-3 基于祖先采样的高效隐空间超调

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。

#### 4.4 非均匀采样间隔下的批常微分方程并行化求解方法

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训练时间，这也是当前训练深度网络模型的必要前提。因此，使 ODE-RSSM 模型支持并行推理与训练是极其重要的。在生成模型中，式(4-3)表示的 GRU 更新和式(4-6)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(4-4)描述的微分方程求解过程，当序列采样点不均匀时，需要并行求解多个积分区间彼此不同的常微分方程。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，并且不支持对不同积分区间进行批处理<sup>[18]</sup>。为了让 ODE-Net 模型适用于非均匀采样数据下的批量计算，现有的基于 ODE-Net 的时间序列模型<sup>[54,122]</sup>一般构建批中所有时间点的有序并集，并求解批中所有常微分方程在所有时间点上的解。实际应用中，对点集进行排序以及求解所有时间点处的解是非常耗时的。

本章提出一种针对批常微分方程网络的并行求解方法，以在积分区间彼此不一致时，加快训练阶段求解各个位置对应 ODE 的速度。根据本章 4.5.6 节中的实验表明，所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，假设求解批常微分方程的时间范围为从初始时间点  $\mathbf{T} = [t_1, \dots, t_I] \in \mathbb{R}^I$  到终止时间点  $\mathbf{T}' = [t'_1, \dots, t'_I] \in \mathbb{R}^I$ ，其中  $I$  是批大小。在初始时间点  $\mathbf{T}$  的批初始状态表示为  $\mathbf{H}_T = [\mathbf{h}_{t_1}^1, \dots, \mathbf{h}_{t_I}^I]$ 。模型需要求解  $I$  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  $\mathbf{H}_{T'}$ 。

$$\mathbf{H}_{T'} = \begin{bmatrix} \text{ODE Solve}(f_\theta, \mathbf{h}_{t_1}, t_1, t'_1) = \mathbf{h}_{t_1} + \int_{t_1}^{t'_1} f_\theta(\mathbf{h}_t^1) dt \\ \dots \\ \text{ODE Solve}(f_\theta, \mathbf{h}_{t_I}, t_I, t'_I) = \mathbf{h}_{t_I} + \int_{t_I}^{t'_I} f_\theta(\mathbf{h}_t^I) dt \end{bmatrix} \quad (4-15)$$

为了并行化地求解  $I$  个常微分方程。本节首先给出定理 1 以揭示定积分中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：

$$\int_a^b f(t) dt = \int_0^1 f(\tau(b-a) + a)(b-a) d\tau \quad (4-16)$$

通过替换  $\tau = \frac{t-a}{b-a}$ ，可以很容易地证明该定理成立。经过该定理启发，本节对  $I$  个常微分方程进行重参数化，以保证其具有一致的积分区间。具体地，利用原常微分方程构造辅助常微分方程，其描述了构造状态  $\mathbf{R}(\tau)$  对于标量时间变量  $\tau$  的导数。

$$\frac{d\mathbf{R}(\tau)}{d\tau} = g_\theta(\mathbf{R}(\tau), \tau) = f_\theta(\mathbf{R}(\tau)) \circ (\mathbf{T}' - \mathbf{T}) \quad (4-17)$$

其中  $\circ$  是按位的逐元素相乘。可以证明，对于给定的初态  $\mathbf{R}(0) = \mathbf{H}_T$ ，辅助常微分方程在  $\tau = 1$  时刻的解等于原始  $I$  个常微分方程的解：

$$\mathbf{H}_{T'} = \mathbf{R}(1) = \text{ODE Solve}(g_\theta, \mathbf{R}(0) = \mathbf{H}_T, 0, 1) \quad (4-18)$$

接下来，将证明重参数化方法构造的初值问题（Initial value problem, IVP）的解等于原始 ODE 的解。首先，定义两个常微分方程以及它们的解  $h(b)$  和  $v(1)$ 。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  $v(1)$ 。

$$\begin{aligned} h(b) &= h(a) + \int_a^b f(h(t), t) dt, \\ v(1) &= v(0) + \int_0^1 g(v(\tau), \tau) d\tau. \end{aligned} \quad (4-19)$$

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  $\{v_n | n \in 1, 2, \dots, N\}$ 。

$$\begin{aligned} h_{n+1} &= h_n + f(h_n, a + \frac{n\alpha}{N}) \times \frac{\alpha}{N}, \quad h_0 = h(a); \\ v_{n+1} &= v_n + g(v_n, \frac{n}{N}) \times \frac{1}{N}, \quad v_0 = v(0). \end{aligned} \quad (4-20)$$

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，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  $h_n = v_n$ ；

**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

$$\begin{aligned} v_{n+1} &= v_n + g(v_n, \frac{n}{N}) \times \frac{1}{N} \\ &= h_n + \alpha f(h_n, a + \frac{n\alpha}{N}) \frac{1}{N} \\ &= h_{n+1}, \end{aligned} \quad (4-21)$$

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

$$v(1) = v_N = h_N = h(b). \quad (4-22)$$

由此，式 (4-18) 得证。

因为辅助常微分方程的积分区间为  $[0, 1]$ ，利用标准的常微分方程求解器即可求解。对于训练阶段 (4-14) 中的循环  $1 \leq i \leq N$ ，利用上述方法可以高效并行化地求解多个常微分方程。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，且与  $N$  的大小无关。求解 KL-D 的完整过程如算法 1 所示。

## 4.5 实验验证与分析

本章采用三个输入输出系统数据集对 ODE-RSSM 和其他基线模型进行评估，以探究如下几个问题：

- **问题 1：**当被识别系统具有较强随机性时，ODE-RSSM 是否优于具有确定性

**算法 1** 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 (4-3)、(4-4) 及 (4-8) 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(4-18)，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:      $\text{KL}^d = \text{KL} \left[ p(\mathbf{z}_{t_{d+1}:t_N} | \hat{\mathbf{h}}_{t_{d+1}:t_N}) || q(\mathbf{z}_{t_{d+1}:t_N}) \right]$
- 7: **end for**
- 8: **返回**  $\text{KL-D} = \sum_{i=1}^D \text{KL}^d$

状态转移的连续时间模型？ODE-RSSM 作为带有随机隐变量且时间连续化的状态空间模型，其在建模非均匀采样系统时是否具有优势？

- **问题 2:** 在训练阶段引入隐空间超调是否能改善模型开环预测的准确度？
- **问题 3:** 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 4:** 本章提出的批常微分方程并行求解算法以及采样隐状态重用技巧是否能够优化模型的训练效率？

#### 4.5.1 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** (Continuous stirred tank reactor, CSTR)<sup>[22]</sup>: 该公有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 7,500 条连续均匀采样的观测数据（每分钟 10 个样本）。其一维输入量为冷却剂流量，在数据集中是分段恒定的，两维输出量分别为产物的浓度和温度。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** (Winding)<sup>[22]</sup>: 该数据采集于工业绕组过程，同样为公有数据集，共包含 2,500 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** (Thickening): 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，因此该系统属于典型的带有较强随机性的不

表 4-2 数据集特性

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

完全可观测系统。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及对比如表4-2所示。

### 4.5.2 实验设定与基线模型选择

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。生成的每一组数据序列均为大小为  $B * N * K$  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度之和。在训练阶段，如式(4-11)所示，将长度为  $N = M + L$  的序列完全输入到模型中，并最大化 ELBO 以训练模型参数。在模型验证和测试阶段，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 4-2 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  $s_{t_M}$ 。然后，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模型中并进行蒙特卡洛预测，通过并行的序列采样可以得到  $n_{traj} = 128$  条长度为  $L$  的系统输出预测序列。根据真实的系统输出序列，求解  $n_{traj}$  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 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，本章对每组原始序列随机抽取 25%、50% 的数据点以构造时间间隔非均匀的采样序列。同时，作为对照组，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。

由于 ODE-RSSM 为带有随机隐变量的连续时间状态空间模型，因此本章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。
- 带有随机隐变量的时间序列生成模型 Latent ODE 和 Latent SDE。

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，并估计各个位置的预测均值  $\bar{y}_{t_i}$ ，然后计算预

测均值与真值之间的 RRSE 和 RMSE。

对于带有确定性隐变量转移的连续时间模型——ODE-RNN 和 Time-Aware RNN，将其输出层修改为预测某一高斯分布的均值和标准差，类似于式(4-7)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，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  $t_{i+1} - t_i$ 。所有模型均采用 Adam 优化器训练，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。

### 4.5.3 非均匀采样下不同模型的辨识效果对比

表 4-3 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
Latent-ODE	0.2595	0.1797	0.1982	0.1407	0.1672	0.1194	0.1591	0.1149	0.1427	0.1016
Latent-SDE	0.3019	0.2046	0.1587	0.1127	0.1633	0.1171	0.0971	0.071	0.0827	0.0604
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

ODE-RSSM 与其他基线模型在三个数据集上的预测对比结果如表 4-3、4-4、和4-5所示。结果表明，对于被下采样过的数据集，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 25% 和 50% 的非均匀采样数据，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。

图 4-4展示了 ODE-RSSM 模型和 RSSM 在工业绕组数据集上的某一预测片段。可以发现，在 P3 处，ODE-RSSM 的预测效果显著优于离散时间的 RSSM。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  $u_{t_i}$  中加入的时间差分特征  $t_i - t_{i-1}$  处理预测时间点非均匀的情况。而此处由于  $t_i - t_{i-1}$  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

表 4-4 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
Latent-ODE	0.5314	0.5224	0.4716	0.4796	0.4779	0.4725	0.4558	0.4629	0.4515	0.4511
Latent-SDE	0.6610	0.6561	0.4674	0.4771	0.4577	0.4562	0.3690	0.3723	0.3287	0.3248
ODE-RSSM	0.5389	0.5327	0.4648	0.4738	0.4444	0.4429	0.3837	0.3880	0.3683	0.3633
ODE-RSSM-O	0.4768	0.4709	0.4169	0.4244	0.4045	0.4029	0.3689	0.3726	0.3324	0.3283

表 4-5 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
Latent-ODE	6.8852	0.225	6.7402	0.228	4.7758	0.1479	4.7416	0.1546	3.3587	0.1157
Latent-SDE	4.3114	0.1441	3.2254	0.1116	2.4136	0.0823	2.1839	0.0763	2.1006	0.0734
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

退化。对于 ODE-RSSM 模型，尽管大间距  $t_i - t_{i-1}$  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。

这一结果与现有研究认为基于 ODE-Net 的时间序列模型更善于处理稀疏采样序列是吻合的<sup>[23]</sup>。相比于直接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学习系统状态在连续时域下的演化，进而有效地处理非均匀采样数据。

在建模系统的随机性方面，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 4-4 和 4-5 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。说明在模型中引入的随机变量  $z_{t_i}$  有效表征了系统转移的随机成分。

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 具有建模随机性系统的能力。但是当被识别系统的随机性较弱时，如 CSTR 数据集，表 4-3 说明此时采用 ODE-RSSM 的预测精度提升效果并不明显。

#### 4.5.4 隐空间超调对于模型长期预测性能的影响性探究

本通过消融实验验证了在 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 4-3 至 4-5 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与随机性，改善程度更加明显。

对比图 4-4(a) 和图 4-4(c) 可以发现经过隐空间超调训练过的模型在 P1 和 P4 位置的预测表现优于图 4-4(b) 和图 4-4(d) 中优化单步证据下限的模型。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。

从上述的预测指标和图形展示结果可以看出，通过优化多步 KL 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。

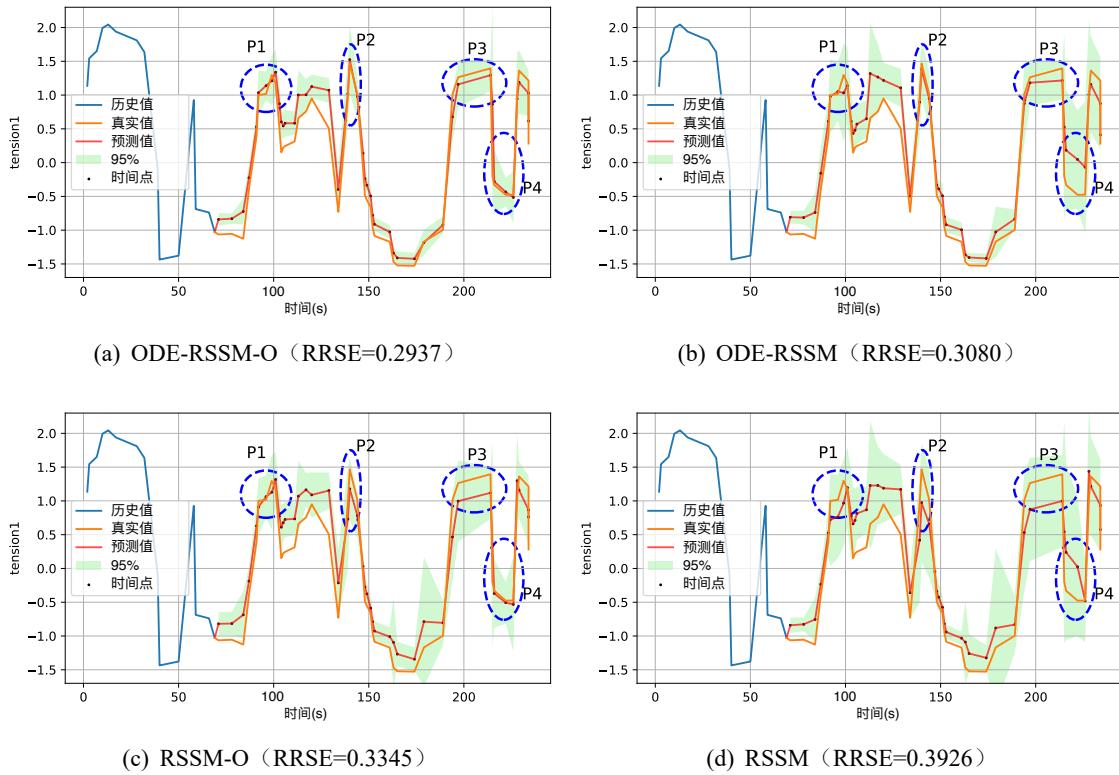


图 4-4 引入微分方程网络及多步隐空间超调的效果分析

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

本节将探究使用非均匀采样数据集训练的 ODE-RSSM 模型是否对于不同采样间隔分布具有泛化能力。首先，本节采用工业绕组数据集（随机抽样比分别设置为 25% 和 50%，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，分别采用下采样比例为 25%、50% 和 100% 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 4-6 所示。表中不仅给出了不同下采样比率下的评估结果，在误差指标之后的括号中还标注了相比于训练集下采样比率，预测误差的上升或下降情况。如 ODE-RSSM (50%) 模型在 25% 采样比率下的 RRSE 记录为 0.575 (0.155)，其中 0.155 代表此时的 RRSE 比 50% 采样比率下的 RRSE (0.421) 高出 0.155。

表 4-6 结果表明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。甚至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>[75]</sup> 表明，基于 ODE-Net 的序列模型在稀疏序列建模和序列内插方面具有明显优势。ODE-RSSM (25%) 预测误差低于 ODE-RSSM (50%) 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%)，此时预测误差明显增加，且 ODE-RSSM (50%) 和 RSSM (50%) 的误差增量是接近的。由于低采样比会增加相邻采样时刻之间的整体长度，因此测试时求解 ODE-Net 的积分范围会超过训练集中时间间隔的整体分布。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 4-6 模型在测试集采样比例与训练集采样比例不同时的迁移效果评估

模型	25%(非均匀)		50%(非均匀)		100%	
	RRSE	RMSE	RRSE	RMSE	RRSE	RMSE
RSSM(25%)	0.537 (0)	0.531 (0)	0.462 (-0.075)	0.462 (-0.070)	0.460 (-0.077)	0.459 (-0.073)
RSSM(50%)	0.615 ( <b>0.149</b> )	0.607 ( <b>0.145</b> )	0.466 (0)	0.462 (0)	0.460 (-0.006)	0.458 (-0.004)
ODE-RSSM(25%)	0.493 (0)	0.488 (0)	0.398 ( <b>-0.096</b> )	0.396 ( <b>-0.092</b> )	0.386 ( <b>-0.107</b> )	0.384 ( <b>-0.104</b> )
ODE-RSSM(50%)	0.575 (0.155)	0.570 (0.150)	0.421 (0)	0.420 (0)	0.380 (-0.042)	0.377 (-0.043)

#### 4.5.6 批常微分方程并行化求解算法的时间效率探究

本节将对提出的批常微分方程并行化求解算法的计算效率进行评估。实验设备为 Nvida DGX station，计算卡为单块 Nvidia V100。图 4-5(a)展示了不同方法在模型训练时求解批 ODE 的时间消耗。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， $I = N \times B$ 。标记为“并行”的两条线代表使用式 (4-18)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<sup>[54,122]</sup>，并按序求解有序集中所有时间点下 ODE 的解。线标中的 rk4 和 dopri5 是求解 ODE-Net 的两个经典数值近似解算器<sup>[17,53]</sup>。在本文第三章已经给出了详细介绍。

从图 4-5(a)可以看出，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较大时，相邻元素之间的距离特别小。当采用自适应的 dopri5 常微分方程求解器时，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 访问次数少于使用 rk4 求解器的访问次数。

相比于在时间点并集上求解 ODE 方程的解，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’，由于求解过程必须满足给定的误差约束，因此随着  $I$  的增加，自适应求解器评估 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  $I$  无关，该方法提供了一种常数时间复杂度下求解批常微分方程的高效方法。

#### 4.5.7 探究隐空间超调时重用采样隐状态对于求解 KL-D 效率的影响

图 4-5(b)说明了当距离  $D$  线性增加时, 求解多步 KL 散度的时间消耗。蓝线表示如式 (4-14)所示在估计 KL-D 时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隐状态, 时间消耗将随  $D$  近似线性增加。相比之下, 式 (4-13)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, 该方法通过优化估计 KL-D 的时间复杂度, 在  $D$  较大时能够显著减少模型训练时间。

通过结合批常微方程的并行求解方法以及重用采样隐状态技巧, 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  $\mathcal{O}(N + D)$ , 其中后验近似推理环节需要递归地计算 GRU 网络的输出, 需要串行计算, 其时间复杂度为  $\mathcal{O}(N)$ , 估计 KL-D 的复杂度  $\mathcal{O}(D)$ 。相比于直接估计 KL-D 进行训练时的计算复杂度  $\mathcal{O}(N + N * D^2)$ , 本章提出的计算效率改进方法是十分可观的。

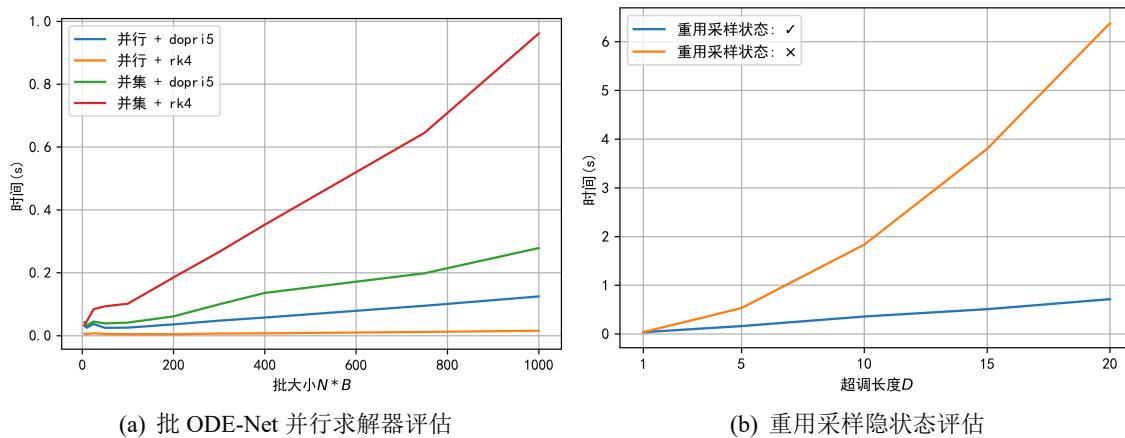


图 4-5 时间复杂度改善效果评估

## 4.6 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的输入输出系统。通过在标准的 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, 该模型引入时序随机隐变量生成过程以建模系统的非确定性, 并利用 ODE-Net 建模非均匀间隔观测时间点之间隐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。为了使模型能够在非均匀采样间隔下支持批量并行推理与训练, 本章提出了一种批常微分方程并行求解算法, 进而高效地求解多个积分区间不同的常微分方程。在训练时, 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改进的高效隐空间超调技术。相比于原始的多步 KL 散度计算方法, 该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，本章对三个输入输出系统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均匀数据采样下的预测效果。结果表明，ODE-RSSM 模型在不规则数据采样下，特别是高缺失率的情况下具有更好的预测准确性，对于随机系统的建模效果优于确定性状态转移的连续时间系统辨识模型。同时，基于多步隐空间超调训练的模型相比于采用单步变分下界训练的模型有更好的开环预测精度。最后，通过对比不同序列长度以及批大小下的损失函数计算求解时间，证实批常微分方程并行求解算法以及改进的高效隐空间超调技术能够将估计多步变分下界的计算复杂度从  $\mathcal{O}(N + N * D^2)$  优化至  $\mathcal{O}(N + D)$ ，显著增加模型训练效率。

## 5 连续时间域下的有模型强化学习控制

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。在大部分的实际生产过程中，复杂工业系统的控制一般是结合局部环路反馈控制器与操作员的个人经验，利用 PID 等控制算法对底层执行设备的频率、泵速、开度进行调节，间接地使关键指标追踪人为给定的工艺设定值。但是由于大部分复杂工业系统运行过程具有非线性、多变量、高时滞等特点，操作员难以实时地给出合理的工艺设定值，底层变量之间难以满足线性或单调关系，简单的反馈控制算法难以适应。当关键指标存在控制偏差时会导致产品质量退化以及增加工业生产成本。

关于复杂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>[123]</sup>、模型预测控制（model predictive control, MPC）<sup>[124]</sup> 等。但对于机械结构复杂、部分变量难以观测的、动力学方程未知的复杂系统，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>[125]</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（Data-driven optimization, DDO）控制算法。Wang 等<sup>[126]</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鲁棒最优控制问题。

近年来，基于强化学习<sup>[31][32]</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>[127][128][129]</sup> 技术，是控制领域研究的热点话题。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>[128]</sup> 等均采用多个神经网络分别对被控系统的动态模型、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>[130]</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。Xu 等<sup>[131]</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。Li 等<sup>[132]</sup> 为了消除 ADP 算法的跟踪误差提出了一种新的值函数表达形式，忽略了效用函数中控制输入的二次项，使得值函数的有界性和收敛性与折扣因子无关。新的值函数形式可以在不给定控制输入设定点的情况下推导出最优控制策略。

近年来，自适应动态规划方法也常被用于解决过程工业控制问题。Wei 等<sup>[133]</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律，同时给出了收敛稳定性的分析。Jiang 等<sup>[134]</sup> 利用穿插学习策略迭代（Interleaved Learning Policy Iteration, ILPI）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>[135]</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，但其控制器与评价模块的设计均建立在系统数据采样是均匀的这一假设之上，而无法适用于采样间隔非均匀的情况。另外，上述训练方法没有在模型学习过程中充分利用系统短期内产生的历史轨迹数据，在数据利用率方面存在改进空间。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，而对于控制网络的训练将产生较大的时间开销。

为了解决上述问题，本章提出了基于连续时间有模型强化学习的复杂工业系统优化控制方法。该方法首先使用系统离线运行数据建模被控系统的连续时间动态模型，在线控制场景下，采集系统实时数据并执行有模型强化学习过程以训练在线控制策略并调优动态模型。同时，针对复杂工业系统存在运行缓慢的特性，提出基于梯度迭代优化的控制求解算法，该算法可以在不引入控制网络的情况下高效求解控制输入量，在提升控制精度的同时，避免了训练控制网络的时间消耗。另外，本章引入了短期经验回放技术<sup>[136][137]</sup> 对短期内的系统运行轨迹数据进行回放训练，以提升局部状态评价及梯度值拟合的准确性。最后，实验环节分别将 HCNVI 控制算法应用于浓密机仿真模型及某矿山深锥浓密机系统，证实了该控制算法在解决底流浓度追踪控制问题时的有效性以及在训练效率、控制精度方面的优越性。

本章主要贡献总结如下：

- 提出了一种基于连续时间有模型强化学习的复杂工业系统控制算法——启发式评价网络值迭代算法（Heuristic critic network value iteration, HCNVI）。该算法能够利用非均匀系统采样数据在线训练连续时间域的评价网络、模型网络，并采用梯度优化算法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。
- 通过工业系统仿真实验验证了 HCNVI 算法在训练时间消耗、控制精度上均优于其他对比方法。
- 在真实工业场景下，成功将 HCNVI 算法应用于浓密机控制，有效稳定设备的底流浓度。

本章内容组织如下：5.1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。5.2节介绍了 HCNVI 算法的模型训练及在线控制过程，具体包括基于神经常微分方程的模型网络构建、基于积分强化学习的策略评价、基于随机梯度下降的动作生成、以及用于评价网络训练的短期经验回放。5.3节分别通过浓密机仿真实验以及真实工业实验验证本章提出控制模型的有效性。5.5节对本章研究工作进行总结。

## 5.1 连续时间的无模型系统控制形式化描述

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  $\mathbf{y}(t) \in \mathbb{R}^n$ 。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  $\mathbf{u}(t) \in \mathbb{R}^m$ ，将不作为控制目标的系统输出量表示为  $\mathbf{h}(t) \in \mathbb{R}^l$ ，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过程中的评价项。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，因此本章引入噪音输入量  $\mathbf{c}(t) \in \mathbb{R}^d$  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，被控系统可表述为式 (5-1) 形式的非线性系统，其中  $f(\cdot)$  为未知非线性函数。

$$[\mathbf{y}(t_{k+1}), \mathbf{h}(t_{k+1})]^T = f(\mathbf{y}(t_k), \mathbf{u}(t_k), \mathbf{c}(t_k), \mathbf{h}(t_k), t_{k+1} - t_k) \quad (5-1)$$

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

$$\dot{\mathbf{s}}(t) = \frac{d\mathbf{s}(t)}{dt} = \mathbf{f}(\mathbf{s}(t), \mathbf{u}(t)) \quad (5-2)$$

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：

$$\mathbf{s}(t) = \mathbf{s}_0 + \int_0^t \mathbf{f}(\mathbf{s}(\tau), \mathbf{u}(\tau)) d\tau, \quad (5-3)$$

其中  $\tau$  为辅助时间变量。

对于系统运行过程式 (5-3)，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：

$$\begin{aligned} U(t) &= U(\mathbf{s}(t), \mathbf{u}(t)) \\ &= (\mathbf{y}(t) - \mathbf{y}^*(t))^T Q (\mathbf{y}(t) - \mathbf{y}^*(t)) + (\mathbf{u}(t) - \mathbf{u}_{\text{mid}})^T R (\mathbf{u}(t) - \mathbf{u}_{\text{mid}}) \end{aligned} \quad (5-4)$$

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。 $\mathbf{u}_{\text{mid}} = \frac{\mathbf{u}_{\text{max}} + \mathbf{u}_{\text{min}}}{2}$ ， $\mathbf{u}_{\text{min}}$ 、 $\mathbf{u}_{\text{max}}$  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  $\int_t^\infty U(\mathbf{s}(\tau), \mathbf{u}(\tau)) d\tau$ 。综上本章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:

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

$V(\mathbf{s}(t))$  为折扣累计评价值函数, 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 (5-5), 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  $\mathbf{u}(t)$ 。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。该方法充分利用一般工业系统具有的运行缓慢的特性, 能够在线控制场景下, 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。

## 5.2 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, 首先需要利用系统历史运行数据对动态模型进行辨识并获得模型网络, 然后从在线控制过程中增量式地收集数据进行评价网络的训练以及模型网络的优化。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及评价网络构建、动作生成、经验回放四个方面介绍 HCNVI 算法的实现过程。

### 5.2.1 基于常微分方程网络的被控系统模型网络构建

本节将介绍如何利用离线数据, 基于神经常微分方程构建被控系统的模型网络。假设离线数据集共包含  $N$  组状态-控制轨迹:  $\mathcal{D} = \left\{ \left( \mathbf{s}_{0:T}^{(n)}, \mathbf{u}_{0:T}^{(n)} \right) \right\}_{n=1}^N$ , 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动态模型。

对于观测时间点序列  $t_{0:T}$ , 定义  $t_0 = 0$ 。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  $\Delta t = t_{i+1} - t_i$ 。预测模型构建方法允许观测时间间隔是非均匀的。类似于本文第二、三、四章, 本章采用带有参数  $\theta$  的神经常微分模型  $\hat{\mathbf{f}}_\theta$  估计系统输出的微分, 进而通过前向求解 ODE-Net 预测真实系统的运行轨迹。模型的学习目标为拟合最优 ODE-Net 参数  $\theta^*$  以最大化数据集的对数似然:

$$\mathcal{L}(\theta, \mathcal{D}) = \sum_{n=1}^N \sum_{i=0}^T \log \mathcal{N} \left( \mathbf{s}_i^{(n)} \mid \hat{\mathbf{s}}_i^{(n)}, \Sigma \right) \quad (5-6)$$

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  $\hat{\mathbf{s}}_{0:T}$ :

$$\hat{\mathbf{s}}_i^{(n)} \mid \left( \mathbf{s}_0^{(n)}, \mathbf{u}_{0:t}^{(n)} \right) = \mathbf{s}_0^{(n)} + \int_{t_0}^{t_i} \hat{\mathbf{f}}_\theta \left( \hat{\mathbf{s}}^{(n)}(\tau), \mathbf{u}^{(n)}(\tau) \right) d\tau \quad (5-7)$$

式 (5-7) 求解的状态轨迹完全依赖于给定的初始状态  $\mathbf{s}_0$  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对  $\hat{\mathbf{f}}_\theta$  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高斯过程 (Gaussian Process, GP) 模型对数据集中的离散动作序列  $\mathcal{D}_u = \left\{ \mathbf{u}_{0:T}^{(n)} \right\}_{n=1}^N$  进行插值处理。

对于给定的非均匀采样控制信号  $\{(t_i, \mathbf{u}_i) | i = 1 \cdots M\}$ , 定义  $\mathbf{t} = (t_1, t_2, \dots, t_M)^T$ ,  $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M)^T$ , 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  $\mathbf{K}(\mathbf{t}, \mathbf{t}) \in \mathbb{R}^{M \times M}$ , 其中:

$$K(t_p, t_q) = \exp \left( -\frac{1}{2} \frac{|t_p - t_q|^2}{\ell} \right) \quad (5-8)$$

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

$$\hat{\mathbf{u}}(t_*) \sim \mathcal{N}(\mathbf{K}(t_*, \mathbf{t})^\top \mathbf{K}^{-1}(\mathbf{t}, \mathbf{t}) \mathbf{U}, K(t_*, t_*) - \mathbf{K}(t_*, \mathbf{t})^\top \mathbf{K}(\mathbf{t}, \mathbf{t})^{-1} \mathbf{K}(\mathbf{t}, t_*)) \quad (5-9)$$

为了确保 ODE-Net 计算过程稳定, 需要保证输入的  $\hat{\mathbf{u}}(t_*)$  连续, 因此将高斯分布均值  $E[\hat{\mathbf{u}}(t_*)]$  作为式 (5-7) 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2所示。

---

## 算法 2 基于 ODE-Net 的模型网络训练过程

---

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

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

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

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

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 (5-7) 求解  $\{\hat{\mathbf{s}}_{t_n:t_n+t_s}^{(n)}\}_{n \in \Xi}$

5:     由式 (5-6), 计算数据集的对数似然并计算 ODE-Net 的梯度  $\nabla_\theta \mathcal{L}(\theta, \mathcal{D})$ , 采用梯度提升法优化  $\theta$

6: **end for**

---

### 5.2.2 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>[138]</sup>训练策略评价模型。对式(5-5)中定义的折扣积分奖赏，可以表示为贝尔曼方程的形式：

$$\hat{V}^H(\mathbf{s}_0) = \int_0^H e^{-\frac{\tau}{\eta}} U(\hat{\mathbf{s}}(\tau), \hat{\mathbf{a}}(\tau)) d\tau + e^{-\frac{H}{\eta}} V(\hat{\mathbf{s}}_H) \quad (5-10)$$

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：

$$V^*(s_0) = \min_{\mathbf{u}[0:H]} \left\{ \int_0^H e^{-\frac{\tau}{\eta}} U(\hat{\mathbf{s}}(\tau), \hat{\mathbf{u}}(\tau)) d\tau + e^{-\frac{H}{\eta}} \hat{V}^*(\hat{\mathbf{s}}_H) \right\} \quad (5-11)$$

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

$$\mathbf{u}^*[0 : H] = \arg \min_{\mathbf{u}[0:H]} \left\{ \int_0^H e^{-\frac{\tau}{\eta}} U(\hat{\mathbf{s}}(\tau), \hat{\mathbf{u}}(\tau)) d\tau + e^{-\frac{H}{\eta}} \hat{V}^*(\hat{\mathbf{s}}_H) \right\} \quad (5-12)$$

原则上，利用式(5-12)即可获得最优控制策略。式(5-12)等号右侧的第一项为积分奖赏项，可以通过(5-7)预测系统轨迹，并在求解轨迹上同时对效用函数进行积分以近似  $[0 : H]$  之间的控制代价。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 构建评价网络  $V_\xi(\cdot)$  近似给定策略的评价函数：

$$\hat{V}_\xi(\mathbf{s}(t)) = W_{c2} \tanh(W_{c1} \mathbf{s}(t)) \quad (5-13)$$

$\tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}$  是网络的激活函数，网络输入层包含  $n + l + d$  个节点，输出层 1 个节点。网络参数为了保证算法参数更新的实时性，采用系统在线控制过程中产生的增量数据训练网络。具体地，采用单步时序差分误差(Temporal difference error, TD error)<sup>[31]</sup>计算评价网络的估计误差值。假设从被控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(5-14)所示：

$$e_c(\mathbf{s}(t_k)) = \hat{V}_\xi(t_k) - e^{-\frac{\Delta t}{\eta}} \hat{V}_\xi(t_{k+1}) + \int_{t_k}^{t_{k+1}} e^{-\frac{\tau-t_k}{\eta}} U(\mathbf{s}(\tau), \mathbf{u}(\tau)) d\tau \quad (5-14)$$

其中  $\Delta t = t_{k+1} - t_k$ ， $\hat{V}_\xi(t_k) = \hat{V}_\xi(\mathbf{s}(t_k))$ ，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:

$$\begin{aligned}\frac{\partial e_c^2(\mathbf{s}(t_k))}{\partial W_{c2}} &= 2e_c(\mathbf{s}(t_k)) \tanh(W_{c1}\mathbf{s}(t_k))^T \\ \frac{\partial e_c^2(\mathbf{s}(t_k))}{\partial W_{c1}} &= 2e_c(\mathbf{s}(t_k))[W_{c2}^T \odot (1 - \tanh^2(W_{c1}\mathbf{s}(t_k)))]\mathbf{s}(t_k)^T\end{aligned}\quad (5-15)$$

采用梯度下降算法对评价网络进行训练更新:

$$W_{ci}(t_k) = W_{ci}(t_k) - l_c \frac{\partial e_c^2(t_k)}{\partial W_{ci}(t_k)}, i = 1, 2 \quad (5-16)$$

其中  $l_c$  是学习率。由于一般工业系统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, 网络需要利用实时采集的数据快速收敛到新的参数, 因此  $l_c$  需设定为固定值以保持在线学习能力。

由于不同物理量的取值差异很大, 会导致网络无法有效学习并且造成超参数设定困难。因此本章采用系统产生的离线数据中各参量的极值对所有训练数据进行归一化放缩。

$$\bar{z} = \frac{2(z - z_{\min})}{z_{\max} - z_{\min}} - 1 \quad (5-17)$$

### 5.2.3 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>[128]</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, 以随机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 模型计算控制动作。该方法可以在环境噪音改变时, 及时更新控制策略, 使被控系统更快速地收敛到理想状态。同时能够减少内存占用并省去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3所示。式 (5-18) 中在估计  $t_k + h$  时刻的折扣累计惩罚时, 下一时刻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  $\mathbf{c}(t_k)$  充当未来一段时间的外部噪音  $\mathbf{c}[t_k : t_k + h]$ 。

为了验证算法3的有效性, 本章对式 (5-18) 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。在实验一5.3.2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5-1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二维控制量, 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3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,

**算法 3** 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  $t_k$  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: 随机采样  $M$  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: 评估积分区间为  $[t_k, t_k + h_i]$  下,  $t_k$  时刻的评价值  $\hat{V}(\mathbf{s}(t_k))_i$ 。

$$\hat{V}(\mathbf{s}(t_k))_i = \int_{t_k}^{t_k+h_i} e^{-\frac{\tau-t_k}{\eta}} U(\hat{\mathbf{s}}(\tau), \mathbf{u}(\tau)) d\tau + e^{-\frac{h_i}{\eta}} \hat{V}_{\xi}(\mathbf{s}(t_k + h_i)) \quad (5-18)$$

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

7: **end for**

8: 求取  $M$  轮采样下评价值的均值, 并利用梯度下降算法对  $\mathbf{u}_c$  进行更新

$$\mathbf{u}_{c+1} = \mathbf{u}_c - \frac{l_u}{M} \sum_{i=1}^M \frac{\partial \hat{V}(\mathbf{s}(t_k))_i}{\partial \mathbf{u}_c} \quad (5-19)$$

9: 将  $\mathbf{u}_{c+1}$  限定在控制约束内

$$\mathbf{u}_{c+1} = \max([-1, -1]^T, \min([1, 1]^T, \mathbf{u}_{c+1})) \quad (5-20)$$

10:  $c = c + 1$

11: **while**  $\|\mathbf{u}_c - \mathbf{u}_{c-1}\| > \epsilon_a$  and  $c < Na$

12: 反归一化, 并求得  $\mathbf{u}(t_k)$

$$\mathbf{u}(t_k) = \frac{\mathbf{u}_c \odot (\mathbf{u}_{\max} - \mathbf{u}_{\min})}{2} + \mathbf{u}_{mid} \quad (5-21)$$

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

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。该结果说明由于最大积分长度  $H$  较小, 同时工业系统运行过程满足利普希茨连续特性且, 某一时刻的控制输入  $\mathbf{u}(t_k)$  对下一时刻系统状态  $\mathbf{s}(t_k + h_i)$  影响相对较小, 且评价网络式 (5-13) 和效用函数式 (5-4) 具有连续、可微的性质, 因此  $\mathbb{E}[\hat{V}(t_k)]$  随  $\mathbf{u}(t_k)$  变化的分布函数在给定的控制约束下一般为单峰函数。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 (5-12) 的最小化条件, 实现最优控制。

## 5.2.4 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, 本章进一步提出短期经验回放方法优化网络训练的损失函数, 以达到优化局部评价值梯度准确性的目的。短期经验回放方法创建经验回访池存储  $L$  步内的状态-动作轨迹  $\{(\mathbf{s}(t_k), \mathbf{u}(t_k)), \dots, (\mathbf{s}(t_{k+L}), \mathbf{u}(t_{k+L}))\}$ ,

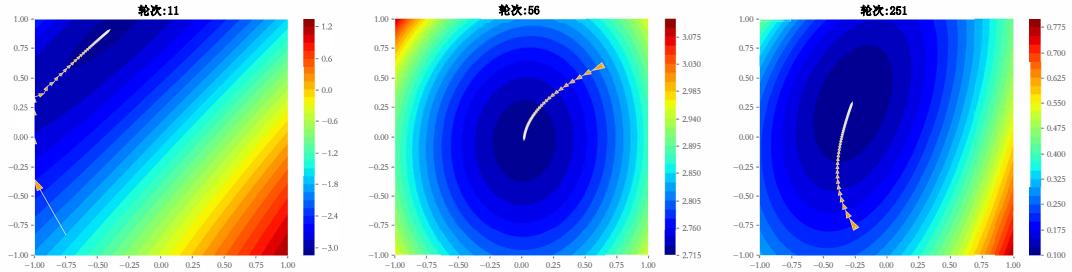


图 5-1 迭代梯度下降过程可视化

在训练过程中，短期轨迹数据可以用于共同计算评价网络的损失值以及优化梯度方向。在当前时刻  $t_{k+L}$ ，式 (5-14) 的误差值计算方法修改为：

$$e_c(t_{k+L}) = \frac{1}{L} \sum_{i=0}^{L-1} [\hat{V}_\xi(t_{k+i}) - e^{-\frac{\Delta t_i}{\eta}} \hat{V}_\xi(t_{k+i+1}) - \int_{t_{k+i}}^{t_{k+i+1}} e^{-\frac{\tau-t_{k+i}}{\eta}} U(\mathbf{s}(\tau), \mathbf{u}(\tau)) d\tau] \quad (5-22)$$

其中  $\Delta t_i = t_{k+i+1} - t_{k+i}$ 。诸如 HDP、DHP 以及本章提出的 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 3 优化  $\mathbf{u}(t_k)$ 。因此对于  $\mathbf{u}(t_k)$  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 (5-23)

$$\nabla \mathbf{u}(t_k) = \gamma \frac{\partial \mathbf{s}(t_{k+1})}{\partial \mathbf{u}(t_k)} \frac{\partial \hat{V}(t_{k+1})}{\partial \mathbf{s}(t_{k+1})} + \frac{\partial U(t_k)}{\partial \mathbf{u}(t_k)} \quad (5-23)$$

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使得  $\frac{\partial U(t_k)}{\partial \mathbf{u}(t_k)}$  是准确的。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。

对于大部分的大型过程工业设备，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 (5-24) 成立。

$$\forall 1 \leq i < L, \|\mathbf{s}(t_{k-i}) - \mathbf{s}(t_i)\| < \delta \quad (5-24)$$

该式表明短期内系统状态点  $\mathbf{s}(t_{k-i})$  都在以  $\mathbf{s}(k)$  为中心， $\delta$  为半径的邻域内。通过式 (5-22) 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  $\lambda(t_k)$ 。

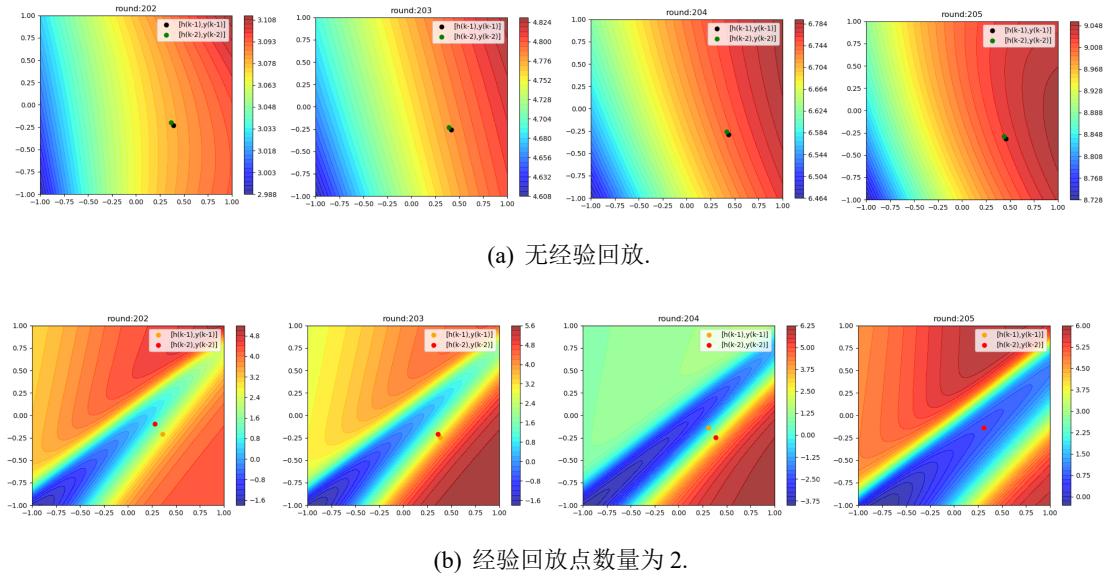


图 5-2 短期经验回放对评价网络的输出值的影响

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，本章对实验一5.3.2节中的评价网络进行了可视化，实验结果如图5-2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5-2(a)代表不使用经验回放，利用式(5-14)训练网络，图5-2(b)代表使用短期经验回放，回放数据点数  $L$  为 2，利用式(5-22)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后的泥层高度和底流浓度。根据实验结果发现。在图5-2(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5-2(b)中，当前系统状态  $(h(t_k), y(t_k))$  所处邻域内，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(5-22)中  $L$  的过大，会导致性能的退化。其原因在于本章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(5-24)，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，本章也将这种经验回放方法称为短期经验回放。

将 HCNVI 算法用于复杂工业系统在线控制的具体流程如算法4所示。

**算法 4 利用 HCNVI 算法实现复杂工业系统无模型在线控制**


---

```

1: 使用系统运行离线数据, 用式(2)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(5-22)求解  $e_c(k)$ 
7:     利用式(5-16)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3求解  $\mathbf{u}(k)$ 
11:    将  $\mathbf{u}(t_k)$  作用于被控系统, 并等待下一次数据采样结果。
12:     $k = k + 1$ 
13: end while

```

---

### 5.3 仿真实验验证

接下来, 本章将以工业浓密机系统为实验对象, 探究 HCNVI 算法对于复杂工业系统的控制效果。浓密机系统的工作原理及相关参数介绍参见本文第三章实验部分。

底流浓度是浓密沉降控制过程的核心评价指标。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, 因此本章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, 本章假定其保持恒定。浓密机控制的首要目标是通过调节絮凝剂流量和底流流量使底流浓度  $y(t)$  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。

当前大部分工业场景下, 浓密机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, 由于浓密机系统具有较高的复杂性, 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, 最终导致浓密机在实际生产过程中时常处于非最优工作状态, 底流浓度偏离理想设定值且频繁剧烈波动。因此利用数据驱动算法实现浓密机系统的智能优化控制, 能够有效降低操作员的作业难度及工作量, 同时提升出料浓度的稳定性与控制精度。

### 5.3.1 浓密机仿真模型

由于在真实工业场景下进行浓密机控制实验成本较高，不便于对不同控制算法的精度进行对比分析。因此本节首先采用浓密机仿真模型验证本章提出控制算法的有效性，下一节将介绍控制算法在真实工业浓密机系统上的控制效果。仿真模型构建方法参考了<sup>[112,139–143]</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、浓密机中把机中轴的影响。

模型推导过程中出现的变量如表5-1，表5-2，表5-3所示

表 5-1 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

表 5-2 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

由文献<sup>[142]</sup>，可得泥层高度与泥层液固质量比之间的关系。

$$h(t) = \frac{W(t)\theta}{A\rho_s} + \frac{W(t)\theta}{A}r(t) \quad (5-25)$$

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

表 5-3 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	$W(t) = c_i(t) q_i(t)$

粒子沉降速度的关系。

$$\frac{d[c_a(t)Ah(t)]}{dt} = c_l(t)[u_t(t) + u_r(t)]A - c_u(t)u_r(t)A \quad (5-26)$$

对式 (5-26) 做变形可得式 (5-27)

$$c_a(t) \frac{dh(t)}{dt} + h(t)p \frac{dc_u(t)}{dt} = c_l(t)[u_t(t) + u_r(t)]A - c_u(t)u_r(t)A \quad (5-27)$$

联立式 (5-27), 式 (5-25), 可得泥层高度  $h(t)$  与底流浓度  $c_u(t)$  的一阶变化率

$$\frac{dh(t)}{dt} = -\frac{W(t)\theta}{Ac_a^2(t)} * \frac{c_l(t)[u_t(t) + u_r(t)] - c_u(t)u_r(t)}{h(t) - c_a(t)\frac{W(t)\theta}{Ac_a^2(t)}} \quad (5-28)$$

$$\frac{dc_u(t)}{dt} = \frac{c_l(t)[u_t(t) + u_r(t)] - c_u(t)u_r(t)}{p(h(t) - c_a(t)\frac{W(t)\theta}{Ac_a^2(t)})} \quad (5-29)$$

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 (5-28)、式 (5-29), 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。

### 5.3.2 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5-3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。

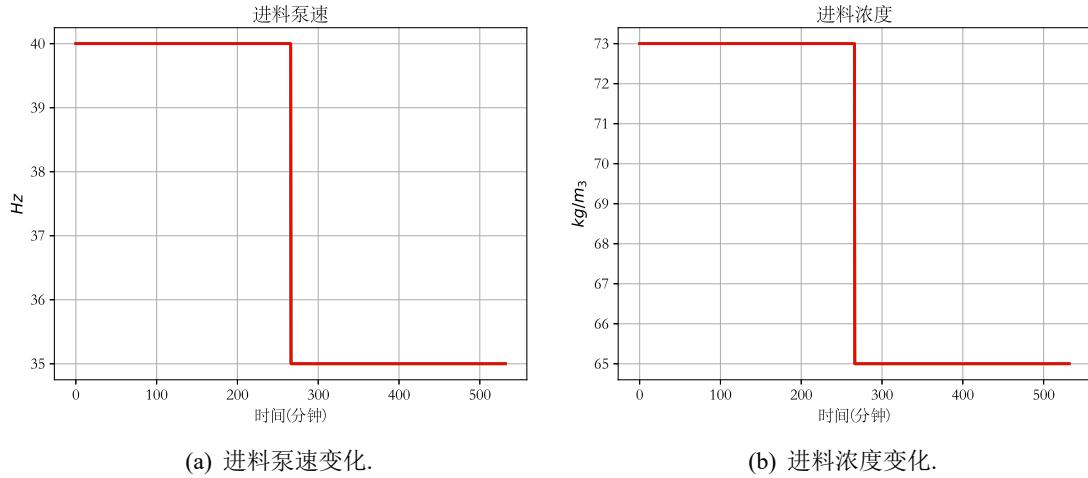


图 5-3 噪音量变化曲线

使用本章提出的 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5-4所示。

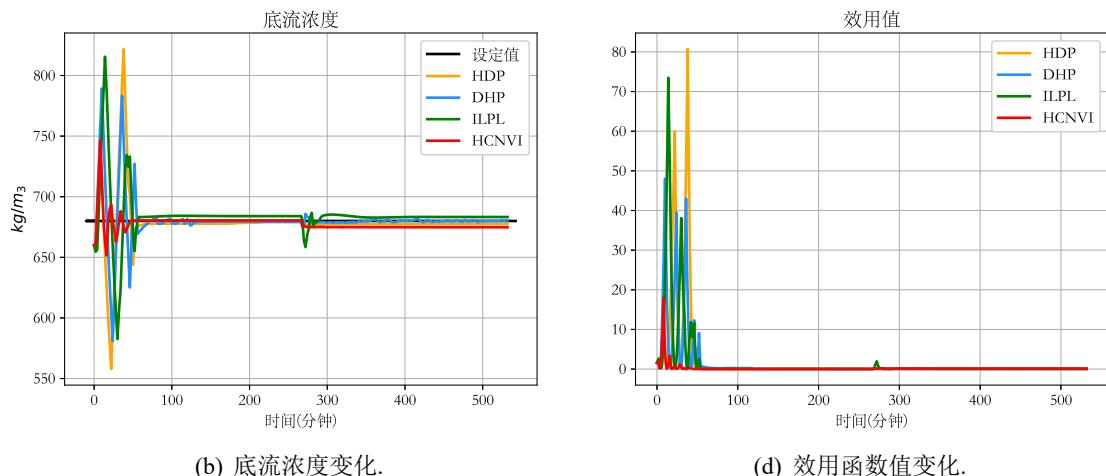


图 5-4 HCNVI 与其他 ADP 算法在恒定噪音输入下的对比

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

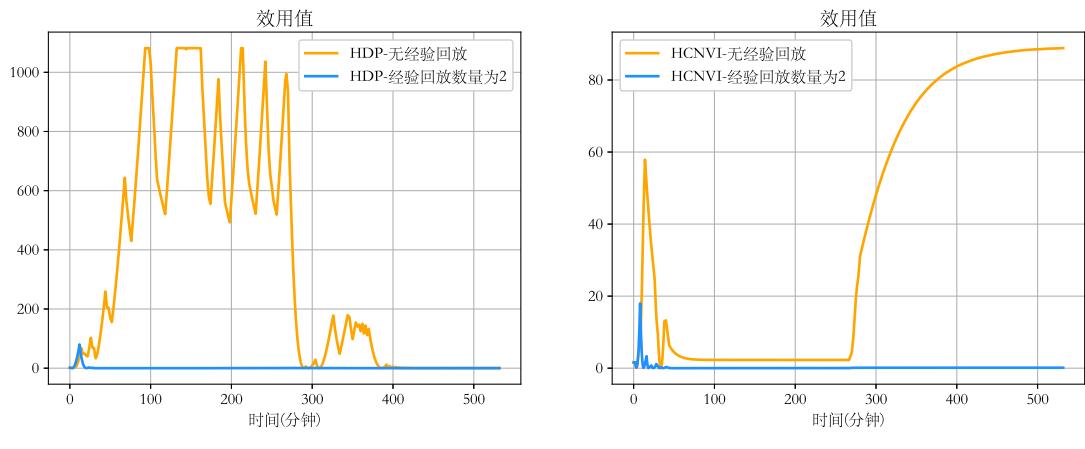
于底流浓度与其设定值偏差较大，效用值较高。但是随着模型与系统趋于稳态，效用值  $U(t)$  不断减小直到接近于 0 的位置。到达 270 分钟时，系统进料浓度、进料流量发生突变，底流浓度无法维持稳态，开始远离设定值。控制模型根据噪音量改变后的系统所产生的轨迹数据重新训练，将底流浓度拉回设定值位置。由于在第一阶段控制模型已经到达过一次稳态，在第二阶段仅需要少量迭代就可以使系统重归理想收敛稳态。通过观察不同控制算法产生的系统轨迹，可以发现不同控制算法到达最优化所需的时间有较大差别，且在收敛到最优化的过程中，底流浓度的波动也有较大差异。在实验第一阶段，为使系统达到稳态，HCNVI 算法所需要的迭代次数更少，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3得出的  $\mathbf{u}(k)$  严格满足式(5-12)的最小化条件，可以使评价网络更快地收敛到最优评价函数。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，本章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5-4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5-4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5-7(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5-2(a)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3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，无论是本章提出的 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。

另外本章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。实验用机的 CPU 型号为 2.6GHz、六核 Intel Core i7，网络均采用 CPU 进行训练。仿真时间为  $T = 270$ ，结果如图5-5(a)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3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，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<sup>[144,145]</sup>。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



(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 5-4 短期经验回放对 HDP 与 HCNVI 的影响

拟合复杂策略函数，最终导致控制效果变差。在本章的实验环节中，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  $\hat{j}(k)$  的影响较小。因此利用算法3求解的  $\mathbf{u}(k)$  满足式 (5-12) 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 (5-12) 的  $\mathbf{u}(k)$ ，算法3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， $\hat{j}(k)$  随  $\mathbf{u}(k)$  变化的分布函数不再是单峰函数，算法3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。

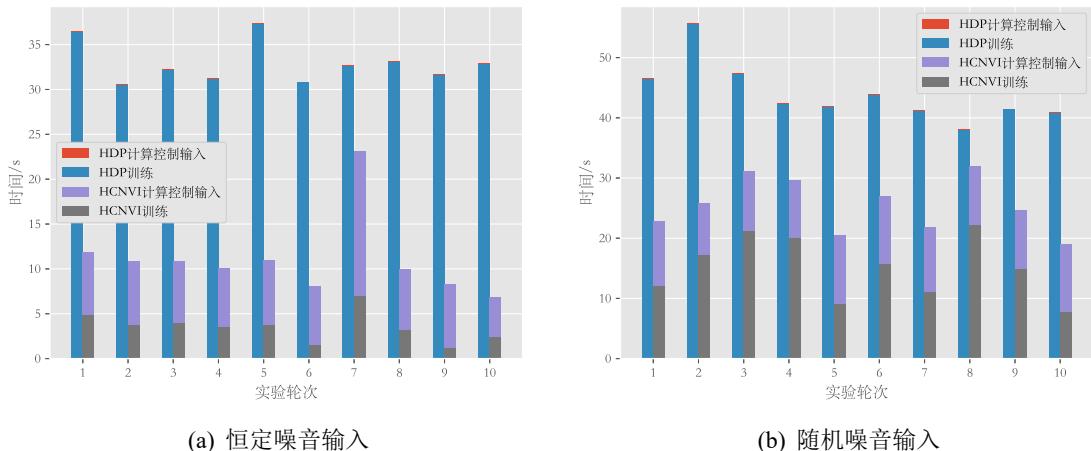


图 5-5 HDP 与 HCNVI 在时间消耗上的对比

### 5.3.3 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5-6所示。

$$\begin{aligned} \mathbf{c}(k+1) &= \mathbf{c}(k) + \Delta\mathbf{c} \\ \Delta\mathbf{c} &\sim N(\mu = 0, \Sigma = \text{diag}(0.6, 0.6)) \end{aligned} \quad (5-30)$$

本实验中 HCNVI 控制器参数与实验一5.3.2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

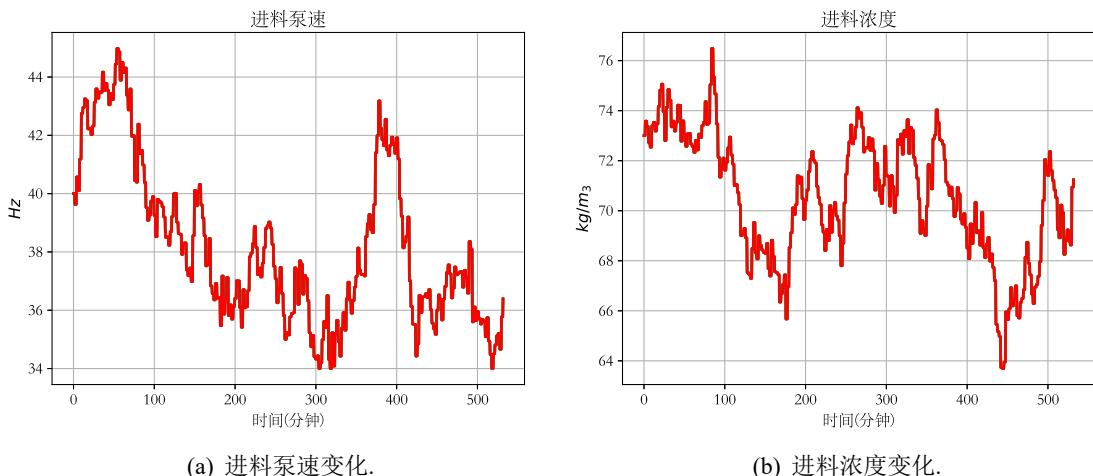


图 5-6 噪音量变化曲线

性能的差异，结果如图5-7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邻域范围内，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。

该实验结果与实验一5.3.2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，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5-4给出了不同算法在实验一5.3.2和实验二5.3.3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

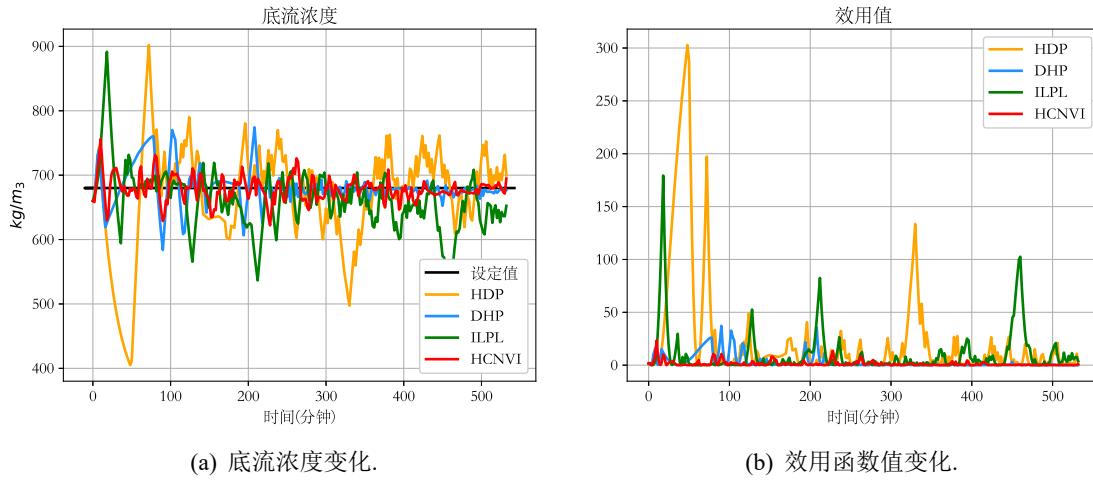


图 5-7 HCNVI 与其他 ADP 算法在波动噪声输入下的对比

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。

表 5-4 不同控制算法之间性能分析

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>

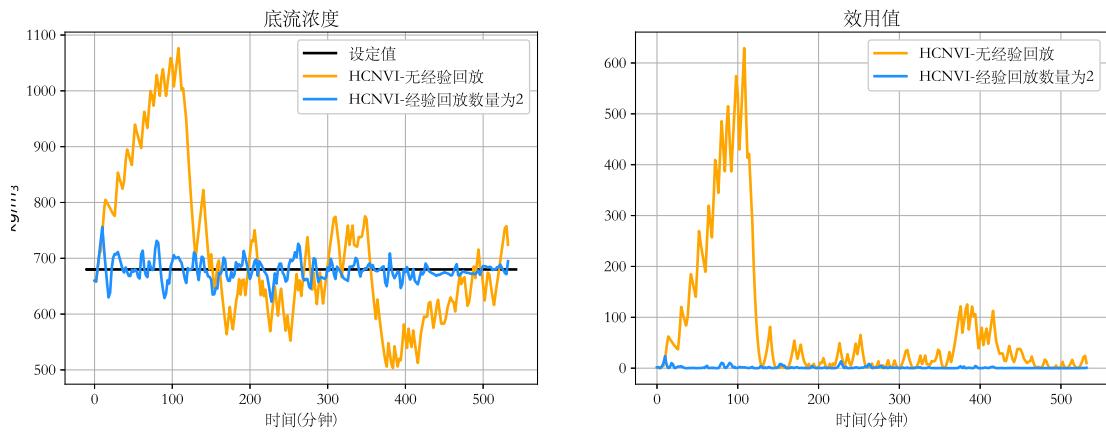
图5-8展示在环境噪音持续变化条件下，不使用经验回放和使用短期经验回放( $L = 2$ )两种情况下 HCNVI 算法控制性能。在无经验回放情况下，底流浓度稳定性明显较差，且效用值明显较高，使用短期经验回放( $L = 2$ )后模型控制效果较好。实验结果表明，短期经验回放技术在环境噪音持续变化下仍对模型控制效果与收敛速度有重要促进作用。

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5-5(b)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(a)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一5.3.2节中达到稳定态，而是始终处于震荡状态，被控系统

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

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

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



(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 5-8 噪音持续变化下短期经验回放对 HCNVI 的影响

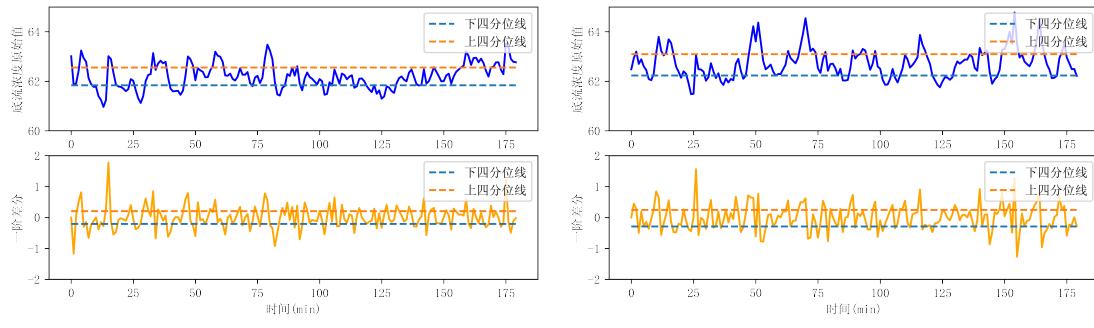
轨迹数据不断变化。每轮学习过程中，为了满足评价网络的精度  $e_c(k)^2 < \epsilon_c$  所需要的训练迭代次数增加，进而导致评价网络训练所需时间及模型总体训练时间增加。但通过横向对比 HCNVI 算法与 HDP 算法的总时间消耗，HCNVI 算法在训练和执行控制过程中所需的总时间消耗仍明显少于 HDP，说明利用算法3替代动作网络所减少的时间消耗在噪音连续波动条件仍十分明显。

## 5.4 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3.4.1节所述的膏体浓密机。具体地，本章分别采用 HCNVI 算法以及浓密机出厂控制系统中的原始规则控制算法对浓密机底流浓度进行控制。对比算法的控制周期为 5 分钟，在浓度低于目标值 1% 时将出料流量的设定值降低  $5m^3/h$ ，高于目标值 1% 时出料流量的设定值增加  $5m^3/h$ 。由于真实工业场景下进行实验无法像仿真实验一样重现系统的外部环境噪音，因此对每种算

表 5-5 真实工业场景下浓密机控制算法性能评估

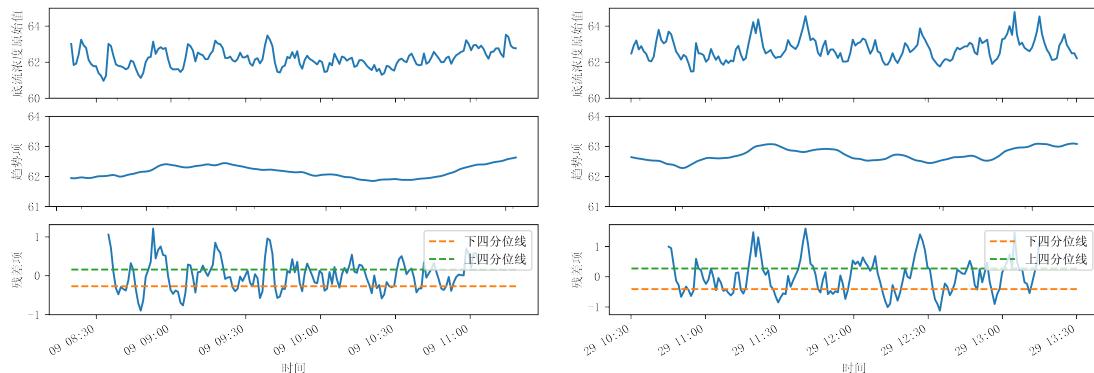
原始规则控制算法下的底流浓度控制指标					
组数	1	2	3	平均	最大
标准差	0.542	0.612	0.622	0.592	0.622
MAE	0.466	0.471	0.527	0.488	0.527
一阶差分标准差	0.378	0.445	0.439	0.421	0.445
$\bar{c}_f - c_s$	0.283	0.121	0.291	0.232	0.291
HCNVI 智能控制算法的底流浓度控制指标					
组数	1	2	3	平均	最大
标准差	0.401	0.497	0.586	0.494	0.586
MAE	0.309	0.463	0.507	0.426	0.507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



(a) HCNVI 控制时底流浓度时序差分图

(b) 规则控制时底流浓度时序差分图

图 5-9 底流浓度控制差分对比图



(a) HCNVI 控制时底流浓度序列残差分解图

(b) 规则控制时底流浓度序列残差分解图

图 5-10 底流浓度序列残差分解对比图

法进行三组实验，每次实验过程持续 1 小时，对于每组结果，统计了五个指标量以评估两种算法的追踪控制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 5-5 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的差的绝对值。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，本节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 5-9 所示：

从图中可以看到，HCNVI 控制下的底流浓度的上四分位点与下四分位点之间的间距明显小于规则控制的间距，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 控制下的四分位线间距为 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 5-9 所示的底流浓度数据进行时间序列分解，比较残差项的四分位距以评估浓度波动大小，对比结果如图 5-10 所示。从图中可以看到 HCNVI 控制算法下，残差项序列的波动较小且更接近 0，四分位距（0.4267）小于规则控制四分位距（0.6836），

另外，为了进一步说明该控制算法对于浓密机运行过程的控制能力，本节测试

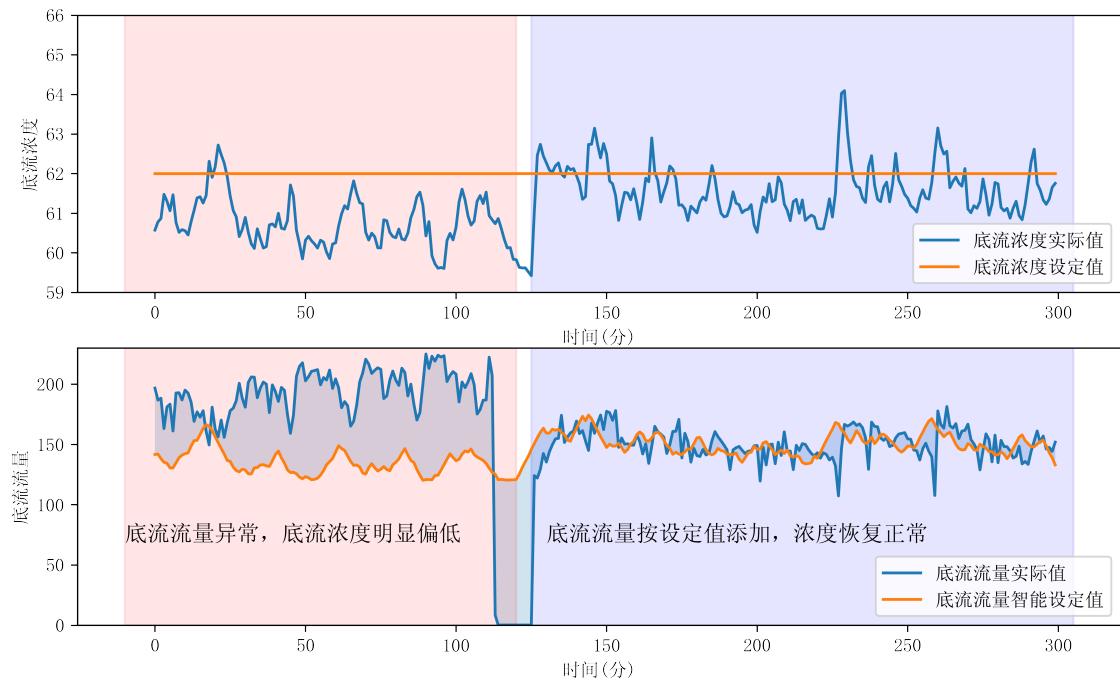


图 5-11 开启智能控制前后浓密机底流浓度变化对比

并观察了控制算法在开闭前后的浓度变化情况。如图5-11所示。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真实工业浓密机系统是有效的。

## 5.5 本章小结

本章提出了基于连续时间有模型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积分代价的评价网络来解决复杂系统优化控制问题。该方法可以在对被控系统未知的情况下，仅利用被控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。另外本章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，因此消耗了更少的训练时间但获得了更优的控制效果。最后，本章基于 HCNVI 算法开发了深锥浓密机底流浓度智能控制系统并应用于某真实矿山生产场景中，现场实验表明，智能控制系统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精确度及控制稳定性。



## 6 周期跳变系统建模及优化

周期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。连续时间周期跳变系统指各阶段依周期转换，且在不同阶段内会呈现不同动态特性的连续时间跳变系统。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，甩干等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）两种状态之间不断切换。

利用历史数据建模其阶段跳变过程与各阶段内的动态模型对于优化此类系统的运行参数与运转模式是极其重要的。然而，在对复杂跳变系统进行建模时会面临两大技术难题。首先，跳变系统通常有多个阶段，每个阶段内系统将呈现完全不同的非线性特性，且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>[24]</sup>。现有的跳变系统参数估计方法<sup>[25]</sup>通常依赖于对系统结构以及滞留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，其输出空间中可能同时存在稳定和非稳定过程<sup>[30]</sup>。目前为止，现有的未经过特定设计的辨识模型难以解决此类带有混合时序特性的系统学习任务。

针对上述技术难题，本章提出了自跳跃常微分方程网络（Autonomous jump ordinary differential equation net, AJ-ODE-Net）以学习同时带有稳定和非稳定输出的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，模型能够开环地给出仿真预测结果。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 ODE-Nets，H-ODE-Net）。该网络是一种双层扩展的常微分方程网络，能够建模同时带有稳定输出和非稳定输出的动态系统。为了实现开环预测时的阶段自转移，本章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，每个预测器能够预估当前所处阶段的持续时间。在开环预测和仿真中，阶段转换预测器能够作为 H-ODE-Net 的调节器，在每个时间点指派合适的 H-ODE-Net 进行系统输出的预测，并自发地转移至下一阶段。同时为了给定 AJ-ODE-Net 的状态初值，本章提出了基于编码器解码器（Encoder-Decoder）框架的预测模型。该模型包含两个 AJ-ODE-Net，编码 AJ-ODE-Net 用于编码历史数据以给定解码 AJ-ODE-Net 的初态，解码 AJ-ODE-Net 能够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。

最后，本章将所述模型及编解码预测框架成功应用于两个具有典型周期多阶段特性的系统中，有效实现了系统运行模式及输出的预测与仿真。特别地，对于膏体制备水泥添加过程，模型实现了在给定系统的多变量输入数据下（浓密机底流浓度以及底流流量），能够仿真系统的运行过程并准确地预测膏体浓度及水泥消耗，其中水泥量消耗量的预测误差小于 2%。进一步地，利用模型的仿真预测功能能够优化膏体制备过程的浓度设定参数，仿真实验表明利用优化之后的浓度设定

值能够在保证膏体浓度满足要求的前提下，优化螺旋给料器的开机频率及水泥消耗，有效降低充填成本 8.2% 左右。

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-ODE-Net 以改善长期开环预测中两种输出项的预测精度。
- 本章提出的 AJ-ODE-Net 模型成功应用于两种周期多阶段系统的预测仿真问题中。特别是对于膏体制备系统，模型能够在给定浓密机出料浓度与流量下，准确地预测系统的水泥消耗以及膏体浓度变化，利用该仿真模型能够优化控制螺旋给料器启停的浓度设定值，有效减少单位体积充填成本 8.2%。

本章内容的结构组织如下：6.1节给出了周期性多阶段系统预测问题中的变量符号定义以及形式化描述。6.2节介绍了 AJ-ODE-Net 模型的基本结构，并详细阐述了如何采用分层常微分方程网络学习同时带有稳定和非稳定时间序列输出的动态过程，以及如何构建连续时间域下的跳变系统状态转移方程。6.3节介绍了基于双 AJ-ODE-Net 模型的编码器解码器框架，并详细介绍了 AJ-ODE-Net 的初态求解方法。6.4节给出了用于训练持续时间预测器和预测模型的损失函数。6.5节将介绍利用所述框架建模两种周期跳变系统。同时，对于膏体制备系统的水泥添加过程，介绍了如何利用辨识模型优化制备系统的浓度设定值以降低设备的重启频率并降低水泥消耗。最后，6.6节对本章工作进行总结，并对 AJ-ODE-Net 模型的未来研究方向以及其他领域的应用做出了展望。

## 6.1 问题形式化描述

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： $\mathbf{X}_{1:I}, \mathbf{Y}_{1:I} = ((t_1, \mathbf{x}_1, \mathbf{y}_1), (t_2, \mathbf{x}_2, \mathbf{y}_2), \dots, (t_I, \mathbf{x}_I, \mathbf{y}_I))$ 。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  $\mathbf{x}: [t_1, t_I] \rightarrow \mathbb{R}^M$  和  $\mathbf{y}: [t_1, t_I] \rightarrow \mathbb{R}^N$ 。为了便于后文描述，本章将  $\mathbf{y}: [t_1, t_I]$  简化为  $\mathbf{Y}_{t_1:t_I}$ 。

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，其内部阶段转换受有限状态过程控制<sup>[79]</sup>。典型的连续时间跳变系统定义如下：

$$\dot{\mathbf{h}}(t) = \mathcal{F}_{\sigma(t)}(\mathbf{h}(t), \mathbf{x}(t)), \quad (6-1)$$

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。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**。在一般的连续时间马尔科夫跳变系

统中, 系统阶段  $\sigma(t)$  的变化遵循马尔科夫过程。而本章假设阶段变量  $\sigma(t)$  的转移过程是依周期的且循环过程是确定的, 分段恒定的  $\sigma(t)$  在循环  $\{0, 1, \dots, N-1, 0, \dots\}$  中不断变化。阶段每次跳跃之间的停留时间取决于由系统本体属性决定的触发条件。因此本文所关注的系统也称为连续时间周期跳变系统 (Continuous-time periodic jump system)。

同本文第三章, 本章要解决的问题仍为根据系统的历史数据信息以及系统的未来序列输入, 开环地预测系统的未来输出。具体地, 本章将给定数据的时间范围  $[t_1 : t_{I+L}]$  分为两部分: 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, 然后利用训练得到的跳变模型式 (6-1) 对预测范围  $[t_I, t_{I+L}]$  下的系统输出进行预测, 如式 6-2 所示:

$$\hat{\mathbf{Y}}_{t_I:t_{I+L}} = f(\mathbf{Y}_{t_1:t_I}, \mathbf{X}_{t_1:t_{I+L}}; \zeta) \quad (6-2)$$

公式中  $\hat{\mathbf{Y}}_{t_I:t_{I+L}}$  表示预测范围内的模型输出。 $\mathbf{Y}_{t_1:t_I}$  表示在条件范围内系统的历史输出。 $\mathbf{X}_{t_1:t_{I+L}}$  表示完整时间范围内的系统输入, 包括历史输入和未来输入部分, 其未来输入也被称为协变量<sup>[117]</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, 上述过程仍为开环预测, 模型预测过程中无法收到来自系统输出的反馈。

## 6.2 自跳跃常微分方程网络

### 6.2.1 基于分层 ODE-Net 的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, 某些输出项为均值、方差恒定的稳定过程, 某些输出项为均值、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, 本章提出分层常微分方程网络 (Hierarchical ODE-Net, H-ODE-Net) 以建模同时包含稳定过程和非稳定过程的复杂系统。

具体地, H-ODE-Net 将三个 ODE-Net 集成为双层结构 (L-1 和 L-2)。图 6-1 展示了 H-ODE-Net 中三个 ODE-Net 之间的连接方式。每个阶段  $\sigma(t)$  下, 构建独立的 H-ODE-Net。其中, 在 L-1 层中的一个 ODE-Net 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  $\mathbf{h}(t)$ 、外部输入  $\mathbf{x}(t)$  和当前阶段持续时间  $T(t)$  对隐状态导数进行预测。对 L-1 中定义的常微分方程进行积分解出的  $\mathbf{h}(t)$  将作为 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, 表示为  $\mathbf{y}(t) = [\mathbf{y}_s(t), \mathbf{y}_{ns}(t)]$ 。

其中, L-1 中的 ODE-Net 为第三章所述的稳定结构, 由门控循环单元 (GRU)

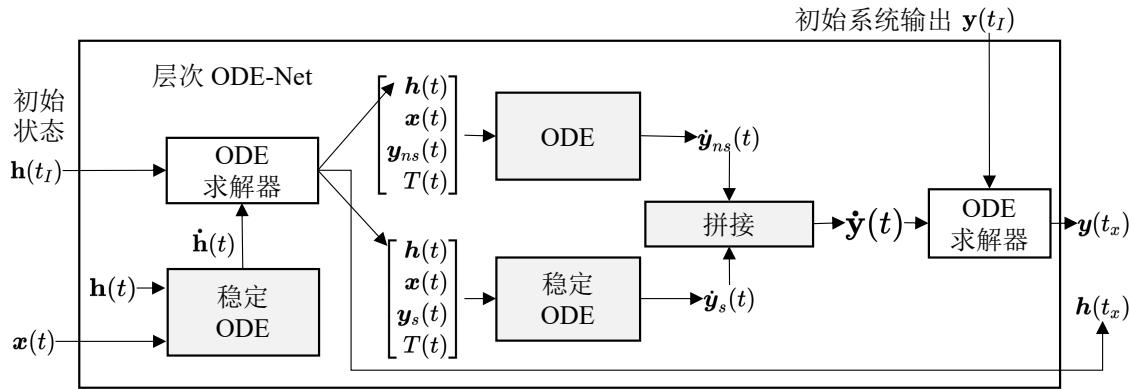


图 6-1 H-ODE-Net 结构图示

网络实现:

$$\frac{dh(t)}{dt} = \frac{1}{\mu} * \left[ \text{GRU}(x(t), h(t), \text{sigmoid}(T(t)), \theta_{\sigma(t)}^{\mathcal{H}}) - h(t) \right] \quad (6-3)$$

其中  $\mu$  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 ODE-Net 被分别用于建模稳定和非稳定的系统输出。其中被用于建模非稳定输出的导数网络采用非稳定增量形式预测输出  $y_{ns}(t)$ :

$$\frac{dy_{ns}(t)}{dt} = \text{NN}(x(t), y_{ns}(t), h(t), \text{sigmoid}(T(t)), \theta_{\sigma(t)}^{y_{ns}}) \quad (6-4)$$

另一个 ODE-Net 类似于 L-1 中的 ODE-Net，由稳定的 GRU 模型构建，用于预测系统的稳定输出  $y_s(t)$ :

$$\frac{dy_s(t)}{dt} = \frac{1}{\mu} * \left[ \text{GRU}(x(t), y_s(t), h(t), \text{sigmoid}(T(t)), \theta_{\sigma(t)}^{y_s}) - y_s(t) \right] \quad (6-5)$$

对于 L-2 中的两个 ODE-Net，均将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。 $\{\theta_i^{y_{ns}}, \theta_i^{y_s}, \theta_i^{\mathcal{H}}\}_{i=1}^N$  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且数据点之间的采样间隔不均匀。例如可能存在某一时刻  $i$ ，两区间  $t_i - t_{i-1} \neq t_{i+1} - t_i$  或者存在  $x(t_i)$  存在，而  $y(t_i)$  缺失的情况。本章采用第五章 5.2.1 中介绍的并行可微插值方法，将离散的控制输入序列插值为连续时间信号并作为 ODE-Net 的输入。

## 6.2.2 自跳跃微分方程状态空间定义及阶段自转移

为了使辨识模型准确地预测周期跳变系统中的阶段转换并建模各个阶段下的系统输出，本节提出了自跳跃常微分方程网络模型。具体地，本章首先利用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-ODE-Net 以学习系统在该阶段内的动态特性，通过对训练集中的序列数据进行阶段变量标注，可以

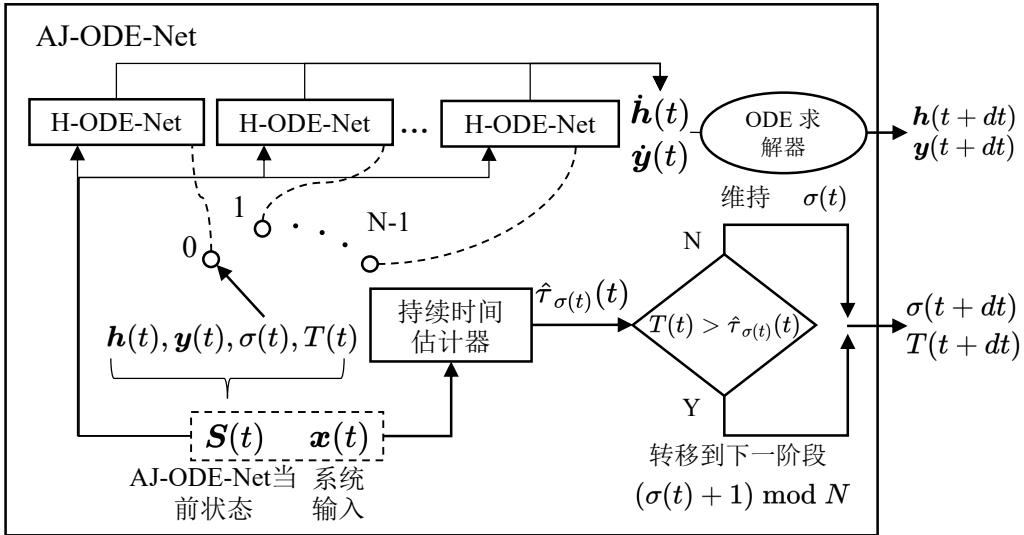


图 6-2 AJ-ODE-Net 模型结构

定向地训练各个 H-ODE-Net 的参数。同时，在预测框架中引入阶段转换预测器学习预测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-ODE-Net 的调节器，实现不同阶段之间的自动转移。图 6-2 简要介绍了 AJ-ODE-Net 的主要组件及其结构。

在图 6-2 中，AJ-ODE-Net 模型嵌入了  $N$  个 H-ODE-Net 模块，根据状态变量  $\sigma(t) \in \{0, \dots, N-1\}$ ，在同一时刻只使用一个 H-ODE-Net 用于系统预测。相比于普通的 ODE-Net 仅建模系统隐状态的导数，AJ-ODE-Net 将隐状态扩展为四个参数，即  $S(t) = [h(t), y(t), \sigma(t), T(t)]$ ，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**。这种定义形式能够覆盖连续时间跳变系统中建模系统输出变化以及阶段变量变化所需的全部信息。AJ-ODE-Net 可以看作是 ODE-Net 的扩展形式，其状态的计算与更新均在连续时间域下进行。给定时间点  $t$  和无穷小时间步长  $\tau$ ，状态更新方程如式 (6-6) 所示：

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

对于周期稳定、各个阶段持续时间不变的系统，可以将系统的运行过程分解为若干区间，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。

对于式(6-6)描述的状态演化过程，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 $\hat{\tau}_{\sigma(t)}$ 来预测当前阶段的最终持续时间。预测器的输入为当前隐状态 $\mathbf{h}(t)$ 和外部输入 $\mathbf{x}(t)$ ：

$$\hat{\tau}_{\sigma(t)}(t) = \text{NN}([\mathbf{h}(t), \mathbf{x}(t)], \phi_{\sigma(t)}) \quad (6-7)$$

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。

### 6.3 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，包括识别当前系统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-ODE-Net 所需的初始状态。

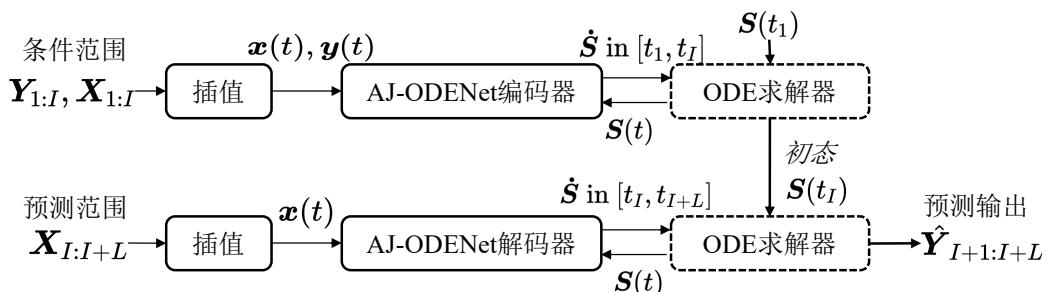


图 6-3 基于 AJ-ODE-Net 的编码器-解码器预测框架

本章遵循编码器-解码器框架<sup>[146,147]</sup>，使用两个 AJ-ODE-Net 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6-3，首先给定条件范围内的系统输入序列和系统输出序列 $\{\mathbf{Y}_{t_1:t_I}, \mathbf{X}_{t_1:t_I}\}$ ，根据初始状态 $\mathbf{S}(t_1)$ 求解 AJ-ODE-Net 编码器的常微分方程，得到 $t_I$ 时刻的系统状态 $\mathbf{S}(t_I) = [\mathbf{h}(t_I), \mathbf{y}(t_I), s(t_I), T(t_I)]^T$ 。进一步地，将 $\mathbf{S}(t_I)$ 作为预测阶段 AJ-ODE-Net 解码器中常微分方程网络的初始状态，在给定预测范围系统外部输入 $\{\mathbf{X}_{t_I:t_{I+L}}\}$ 下，求解该范围内隐状态序列，进而预测系统输出。具体地，在条件范围 $[t_1, t_I]$ 的编码阶段，本节将连续时间信号 $\mathbf{y}(t)$ 和 $\mathbf{x}(t)$ 合并作为 AJ-ODE-Net 编码器的输入，如式(6-8)所示。

$$\tilde{\mathbf{S}}(t_I) = \text{ODESolve}(\text{AJ-ODE-Net 编码器}, \mathbf{S}(t_1), \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{y}(t) \end{bmatrix}, t_1, t_I) \quad (6-8)$$

其中  $t_1$  时刻的初始状态定义为:  $\mathbf{S}(t_1) = [\mathbf{h}(t_1) = \mathbf{0}, \mathbf{y}(t_1), T(t_1) = 0, s(t_1)]^T$ 。根据求解得到的  $\tilde{\mathbf{S}}(t_I) = [\tilde{\mathbf{h}}(t_I), \tilde{\mathbf{y}}(t_I), \tilde{s}(t_I), \tilde{T}(t_I)]^T$ 。可以获得状态  $\mathbf{S}(t_I) = [\mathbf{h}(t_I), \mathbf{y}(t_I), \tilde{s}(t_I), \tilde{T}(t_I)]^T$ 。其中  $\mathbf{h}(t_I)$  是从均值为  $\tilde{\mathbf{h}}(t_I)$ , 协方差阵  $\mathbf{I}$  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>[73]</sup>。对于式6-8中的求解过程, 其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分布, 并从分布中进行采样以重构未来的系统输出<sup>[59,76]</sup>。

接下来, 通过给定解码阶段的初始状态  $\mathbf{S}(t_I)$  和预测范围内连续时间系统输入  $\mathbf{x}(t)$ , 在  $[t_I, t_{I+L}]$  范围内求解 AJ-ODE-Net 解码器对应的常微分方程, 可以得到该范围内的隐变量状态  $\mathbf{S}(t)$ , 其中包括系统在任意时刻  $t_i$  的系统输出  $\hat{\mathbf{y}}(t_i)$ , 进而达到系统预测的目的:

$$\hat{\mathbf{S}}(t_{I+L}) = \text{ODESolve}(\text{AJ-ODE-Net 解码器}, \mathbf{S}(t_I), \mathbf{x}(t), t_I, t_{I+L}) \quad (6-9)$$

AJ-ODE-Net 编码器和 AJ-ODE-Net 解码器之间有两个区别。首先, 系统输入  $\mathbf{x}(t)$  和输出  $\mathbf{y}(t)$  都均是编码阶段的输入, 而解码器的输入仅包含系统输入  $\mathbf{x}(t)$ 。因此解码器和编码器中 H-ODE-Net 的输入层的大小是不同的。另外, 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得到不同阶段之间的转换位置, 不需要阶段转换预测器进行阶段的自转移。推导出的初始状态  $S(t_I)$  中的  $\sigma(t_I)$  和  $T(t_I)$  是准确可靠的。对于解码器, 由于系统输出  $\mathbf{y}(t)$  未知, 因此需要利用式6-6所示的阶段转换预测器进行状态自切换。由于相邻阶段之间的转移是依靠持续时间预测器进行判断的, 无法保证绝对准确, 在长范围开环预测时会带来较大的累积相位误差。

## 6.4 模型训练

利用 AJ-ODE-Net 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-ODE-Net 解码器即可得到预测结果  $\hat{\mathbf{Y}}_{t_{I+1}:t_{I+L}}$ 。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, 接下来本小节将详细介绍损失函数的定义。

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-ODE-Net 编码器中的  $N$  个 H-ODE-Net
- $\Theta_d = \{\theta_i^{\mathcal{Y}^{ns}}, \theta_i^{\mathcal{Y}^{ns}}, \theta_i^{\mathcal{H}}\}_{i=0}^{N-1}$ : 在 AJ-ODE-Net 解码器中的  $N$  个 H-ODE-Net
- $\Phi = \{\phi_i\}_{i=0}^{N-1}$ : AJ-ODE-Net 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各阶段持

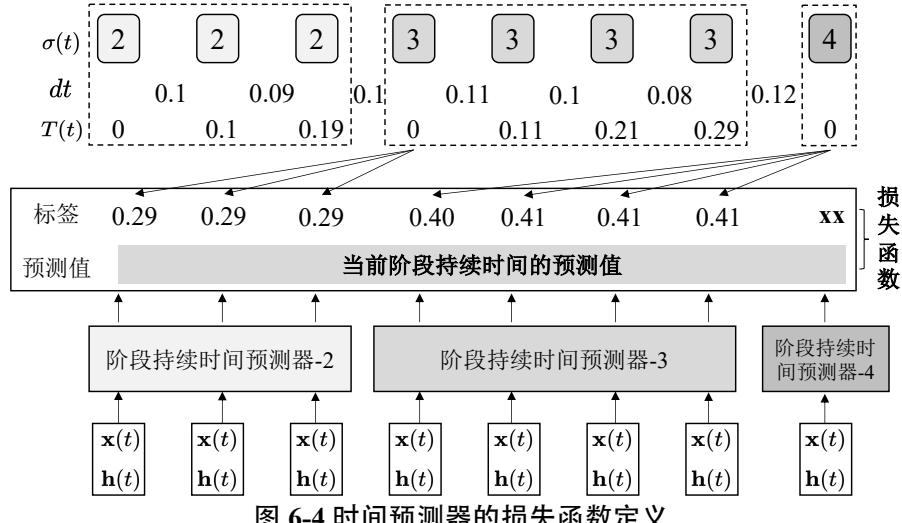


图 6-4 时间预测器的损失函数定义

续时间的误差损失  $\mathcal{L}_\tau$ :

$$\mathcal{L}(\Theta_e, \Theta_d, \Phi) = \mathcal{L}_{\mathcal{P}}(\Theta_e, \Theta_d) + \lambda \mathcal{L}_\tau(\Phi) \quad (6-10)$$

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

对于  $\mathcal{L}_{\mathcal{P}}$ , 由于模型将  $\mathbf{h}(t_I)$  作为隐变量进行后验推断, 因此可以采用变分贝叶斯优化方法, 将最大化系统观测输出的证据下界 (Evidence lower bound, ELBO) 作为模型训练的目标, 以最大化系统输出边际似然的下界<sup>[59]</sup>:

$$\begin{aligned} \text{ELBO}(\Theta_e, \Theta_d) &= -\text{KL}[q_{\Theta_e}(\mathbf{h}(t_I) | \mathbf{Y}_{t_1:t_I}, \mathbf{X}_{t_1:t_I}) \| p(\mathbf{h}(t_I))] + \\ &\mathbb{E}_{\mathbf{h}(t_I) \sim q_{\Theta_e}(\mathbf{h}(t_I) | \mathbf{Y}_{t_1:t_I}, \mathbf{X}_{t_1:t_I})} \log p_{\Theta_d}(\{\mathbf{y}_{t_i}\}_{i=I+1}^{I+L} | \mathbf{h}_{t_I}, \mathbf{X}_{t_I:t_{I+L}}) \end{aligned} \quad (6-11)$$

本章参考前人工作<sup>[17,59,75]</sup>, 对隐变量的先验分布和后验分布做出了如下假设。首先假设隐变量  $p(\mathbf{h}(t_I))$  的先验分布服从正态分布  $\mathcal{N}(\mathbf{h}(t_I); \mathbf{0}, \mathbf{I})$ 。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, 本节假设生成模型的输出概率分布  $p_{\Theta_d}(\mathbf{Y}_{I+1:I+L} | \mathbf{h}_{t_I}, \mathbf{X}_{t_I:t_{I+L}})$  为具有固定协方差矩阵的正态分布。分布的均值定义为 AJ-ODE-Net 解码器预测出的系统输出。最大化重构似然的期望  $\mathbb{E}_{q_{\Theta_e}(\mathbf{h}_{t_I})} [\log p_{\Theta_d}(\mathbf{Y}_{I+1:I+L} | \mathbf{h}_{t_I}, \mathbf{X}_{t_I:t_{I+L}})]$  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, 系统预测系统输出的损失和隐变量推理的损失可表示为:

$$\mathcal{L}_{\mathcal{P}}(\Theta_e, \Theta_d) = -\text{ELBO}(\Theta_e, \Theta_d) = \|\mathbf{h}(t_I)\|^2 + \sum_{i=I+1}^{I+L} \|\hat{\mathbf{y}}(t_i) - \mathbf{y}(t_i)\|^2 \quad (6-12)$$

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。

$$\mathcal{L}_\tau(\Phi) = \sum_{i=I+1}^{I+L} ||\hat{\tau}_{s(t_i)}(t_i) - \tilde{\tau}(t_i)||^2 \quad (6-13)$$

其中  $\tilde{\tau}(t_i)$  代表阶段  $\sigma(t_i)$  的实际持续时间。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  $\tilde{\tau}(t_i) = t_r - t_l$ 。图6-4给出了损失值计算过程的样例。

网络的所有模块通过 Adam 优化器进行训练。完整的训练过程如算法5所示。

---

#### 算法 5 基于 AJ-ODE-Net 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:     利用先验知识对序列的各个位置指定阶段标签  $s_{t_1:t_{I+L}}$ 。
5:     //为了加快训练速度, 以下步骤并行执行。
6:     使用并行样条插值算法处理离散序列  $\mathbf{X}_{1:I+L}, \mathbf{Y}_{1:I+L}$ , 生成连续时间信号
       $\mathbf{X}(t)$  和  $\mathbf{Y}(t)$ 。
7:     编码阶段: AJ-ODE-Net 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  $\mathbf{S}(t_I)$ , (6-8)。
8:     预测阶段: AJ-ODE-Net 解码器根据预测范围下的系统输入预测离散时
      间点  $\{t_{I+1}, \dots, t_{I+L}\}$  下的系统输出, (6-9)。
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-ODE-Net 中的参数
       $\Theta_e, \Theta_d, \Phi$ 。
10:    end for
11:  end for

```

---

## 6.5 实验验证与分析

接下来本章将使用两个数据集验证 AJ-ODE-Net 模型及衍生出的编码器解码器框架对于周期跳变系统的建模预测效果。其中一个数据集为公有的数据中心制冷系统运行数据集, 该数据集描述了制冷系统能耗、进气温度、服务器热负载之间的动态关系。制冷系统的阶段性工作模式使得该数据集具有典型的周期多阶段特性。本节将使用基于 AJ-ODE-Net 的编码器-解码器框架建模该系统的跳变运行过程, 以开环仿真系统能耗以及进气温度变化, 实验中使用的另一个数据集为膏体制备场景下的水添加系统运行数据集。水泥添加系统的间断性工作模式使得该数据集存在明显的周期多阶段特性。本章提出的自跳变模型能够根据给定的浓密机底流浓度和底流流量在线预测制备系统的水泥消耗及膏体浓度变化。下一小节中, 将利用训练得到的预测模型, 仿真不同膏体浓度阈值对系统水泥消耗及给料装置启停变化的影响, 同时利用仿真结果优化给料系统的启停浓度设定值, 进而优化膏体制备成本。接下来本节将首先介绍两个数据集的来源及输入输出项, 然

后介绍训练集与测试集的构建细节，以及评估模型预测精度的评价指标。紧接着，对比本章方法与其他同类方法的预测效果。最后介绍如何利用预测模型辅助膏体制备系统的运行参数。

### 6.5.1 数据集介绍

#### 1) 数据中心制冷系统简介

制冷系统公有数据集采集于施耐德电气公司设计的行间制冷系统<sup>[24]</sup>。制冷系统的运行数据可通过调用 Seduce 平台<sup>[148]</sup> 的 API 获取。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，包括制冷能耗和气体温度等数据。该制冷系统包括室内和室外两部分。室内的液体-空气热交换器吸收服务器产生的热空气，通过管道和风机将热量转移和疏散到室外。室外的冷凝器将通过压缩生成的冷空气输送至室内。

制冷系统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  $x_1(t)$ ）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温度限定在某预先设定的区间上下限内。当进气口空气温度达到上限  $T_{i_{\max}}$  时，制冷系统会开启压缩机，制冷系统产生的制冷量  $y_2(t)$  大幅增加，室内温度逐渐下降，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。当进气口温度低于下阈值  $T_{i_{\min}}$  时，压缩机关闭，制冷系统变为待机状态。此阶段内进气口温度不断上升，其制冷量和制冷功率接近于 0。因此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。由于制冷与待机两阶段交替出现，制冷系统的运行过程具有典型的周期多阶段特性。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： $\mathbf{y}_{ns}(t) = [y_3(t)]$ 。

#### 2) 膏体制备系统简介

实验中探究的膏体制备系统为飞翼公司设计生产。该系统的运行数据可通过 OPC-UA 协议访问集散控制系统实时获取。数据采集系统以 10 秒为周期采集传感器监测数据，包括浓密机出料的浓度、出料流量、水泥添加量、制备的膏体浓度、产出膏体流量等数据。图 6-5展示了膏体制备系统的运行过程，以及该系统与本文第三、四、五章所述的膏体浓密机系统之间的关系。该制备系统以双轴卧式搅拌机为核心，将深锥浓密机排出的高浓度料浆为输入，添加一定质量的水泥粉并在搅拌槽内混合均匀，最终在下料口处排出高浓度的膏体用于采空区充填。该设备是膏体制备的最后一道工序，是维持膏体浓度及强度稳定、满足充填标准的关键环节。

由于深锥浓密机出料浓度、流量不稳定，理论上需要动态调节水泥添加量以确保产出的膏体浓度满足充填要求。然而，由于工业场景下用于水泥添加控制的

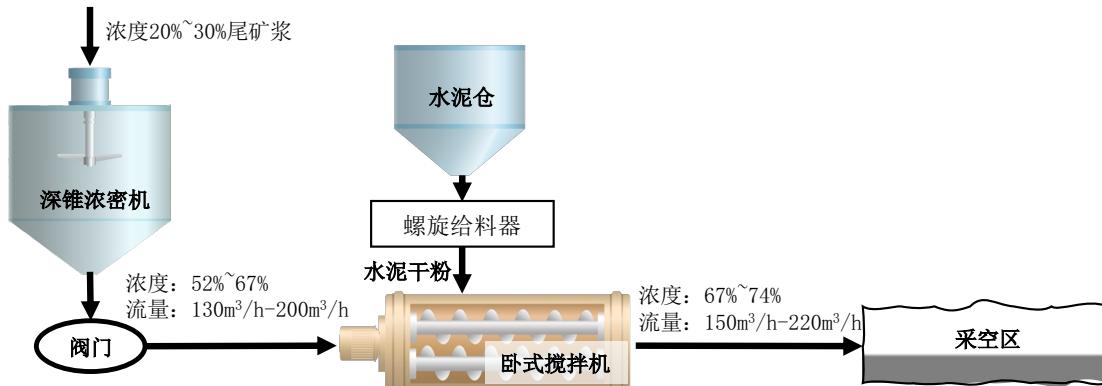


图 6-5 膏体制备系统工作流程示意图

螺旋微分秤难以调控，想要动态地改变固体出料量设定值并满足实时精准控制是极其困难的。考虑到卧式搅拌槽能够作为暂存尾矿浆和水泥的缓冲池，工业场景下一般将水泥添加量设为固定值，按需间隔添加。为了保证膏体浓度满足充填最低要求且尽量减少水泥消耗，制备系统需要维持膏体浓度限定在某预先设定的区间上下限内。当膏体浓度低于最低下限时，系统自动开启螺旋给料器并持续加入水泥粉，水泥添加量  $y_1(t)$  经过快速攀升后会趋近于某一固定值，此时充填浓度逐渐上升。当膏体浓度升高至某一阈值时，给料器关闭，水泥添加量快速降至 0。膏体浓度不断下降。由于螺旋给料器开机与待机两阶段交替出现，加料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 6-6 所示。

对于膏体制备过程，当浓密机出料浓度  $x_1(t)$  较高时，水泥的需求量较少，此时开启螺旋给料器后，膏体浓度  $y_3(t)$  升高较快。当浓密机出料流量  $x_2(t)$  较高时，按比例地，也需要添加更多的水泥以满足浓度要求。此时搅拌机排出的膏体流量  $y_2(t)$  也会增加。因此，建模浓密机出料浓度、流量对于给料器起停过程的影响，进而预测水泥消耗以及膏体浓度变化，对于优化膏体制备成本具有重要意义。

对于膏体制备系统，水泥添加量  $y_1(t)$  和膏体流量  $y_2(t)$  为系统的稳定输出  $\mathbf{y}_s(t) = [y_1(t), y_2(t)]$ 。膏体浓度  $y_3(t)$  为系统的非稳定输出： $\mathbf{y}_{ns}(t) = [y_3(t)]$ 。将膏体浓度变化定义为非稳定过程，这一性质与系统的先验特性是保持一致的。在后续介绍的膏体制备成本优化实验中，本节预期通过改变浓度阈值优化物料及设备损耗成本，这要求模型能够对不同膏体浓度阈值实现外推预测，在 H-ODE-Net 模型中将膏体浓度定义为非稳定过程，能够保证该阶段内浓度的预测值持续上升或下降，这对于实现可靠、合理的外推预测具有重要意义。

### 6.5.2 数据采集、预处理及模型训练细节

对于制冷系统数据集，本节从 Seduce 平台采集了四组数据集用于模型的训练与评价，且四组数据集的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，说明当前热负载越高。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

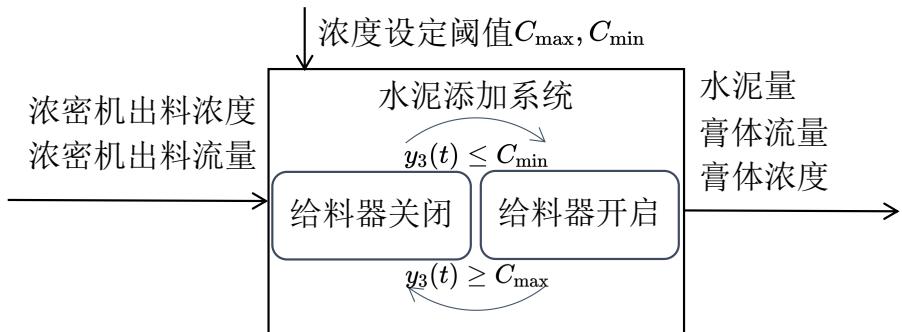


图 6-6 水泥添加控制过程机理简化图

条/秒，序列对应的时间长度约为 8000s-10000s。

对于膏体制备系统数据集，本节从 DCS 系统中采集约 11 万条数据用于模型的训练与评价，所有数据均来源于生产稳定运行的时段，且不同时段浓密机出料流量和浓度有较大差异，流量波动范围为  $130m^3/h - 200m^3/h$ ，浓度波动范围为 52% – 67%。

每个数据集的前 50% 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 6.1 节的定义，前 800s 为条件范围，用于编码器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。在测试阶段，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 800s 之后的序列输入下，以开环的方式估计预测范围内的系统输出。预测结果与测试集中的真实输出作对比以评估模型精度。

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序列转换为连续信号。在本章实验中，膏体制备系统数据集存在采样间隔不均匀、部分数据缺失的问题。本章采用第五章 5.2.1 中的带有二次平方指核函数的高斯过程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解神经 ODE 系统时所需的必要条件<sup>[19]</sup>。由于数据集中水泥添加量监测值  $y_1(t)$  具有较大噪音，因此不适合采用第三章介绍的样条插值方法。

接下来，为了训练 AJ-ODE-Net 模块中的阶段转换预测器，需要在训练前标注序列数据各时间点的所属阶段。本节基于系统先验知识构建了识别系统阶段转换的规则，并基于此自动地为序列数据赋予阶段标签。膏体制备系统和制冷系统的阶段转换关系分别如表 6-1 和表 6-2 所示。其中，制冷系统在四个阶段之间依周期地循环切换，膏体制备系统在两个阶段之间循环切换。

在训练过程中，选择验证集中表现最好的模型用于最终的模型评估。由于系统输入输出的维数较低，且得益于伴随状态法，不需要存储求解 ODE 时的完整计算图，训练 AJ-ODE-Net 网络对于显存的消耗是极低的。因此本节选择了较大的批

表 6-1 膏体制备系统阶段转换判别规则

状态	描述	下一阶段	转换触发条件
0-关闭	螺旋给料器关闭, 浓度持续降低	1	$y_3(t) \geq C_{\max}$
1-开机	螺旋给料器开启, 浓度持续增加	0	$y_3(t) \leq C_{\min}$

表 6-2 制冷系统阶段转换判别规则

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	$y_3(t) \leq Ti_{\min}$

大小 (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  $h(t)$  的维度是 20，学习率设置为 0.005。

### 6.5.3 跳变系统运行过程辨识与仿真

本节首先使用基于 AJ-ODE-Net 的编码器-解码器框架建模预测膏体制备过程，在给定浓密机出料浓度和出料流量输入下，预测膏体制备系统的输出。模型的预测变量为三个：膏体浓度、膏体流量和水泥瞬时添加量。

本节采用单 ODE-Net、ODE-RNN<sup>[59]</sup>、CDE-Net<sup>[19]</sup> 作为本章 AJ-ODE-Net 的对比模型。除单 ODE-Net 外，其他对比模型也可以与本章的阶段转换预测器组合，进而具备跳变系统辨识能力。AJ-ODE-Net 与其他模型的对比结果如图6-7所示。其中橙色代表水泥添加系统的开机阶段，蓝色代表水泥添加系统的关机阶段。其中图 6-7(a) 为真实的系统输出，图 6-7(d) 为采用带有 H-ODE-Net 的 AJ-ODE-Net 模型的预测结果。在图 6-7(b) 中，将 AJ-ODE-Net 替换为一个单一的稳定型 ODE-Net，此时各阶段的转换不受持续时间预测器控制。可以发现，使用单 ODE 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，特别在系统突变点处预测结果与真实系统输出偏离较大。在图 6-7(c) 中，采用 ODE-RNN<sup>[59]</sup> 替换了 AJ-ODE-Net 中用于建模稳定和非稳定混合输出的 H-ODE-Net 模块。可以发现经过替换后，网络输出在阶段转移边界处存在较大震荡。相比于其他模型，图 6-7(d) 中 AJ-ODE-Net 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，相比单模型结构，能够更好地预测多阶段系统的阶段转换边界。同时，结合了非稳定输出和稳定输出的 H-ODE-Net 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。

类似地，对于制冷系统数据集，本节仍然采用单 ODE-Net、AJ+ODE-RNN、AJ+CDE-Net 作为本章 H-ODE-Net 的对比模型，对比结果如图6-8所示。其中橙色代表制冷系统启动第一阶段，绿色代表制冷系统启动第二阶段，蓝色代表制冷

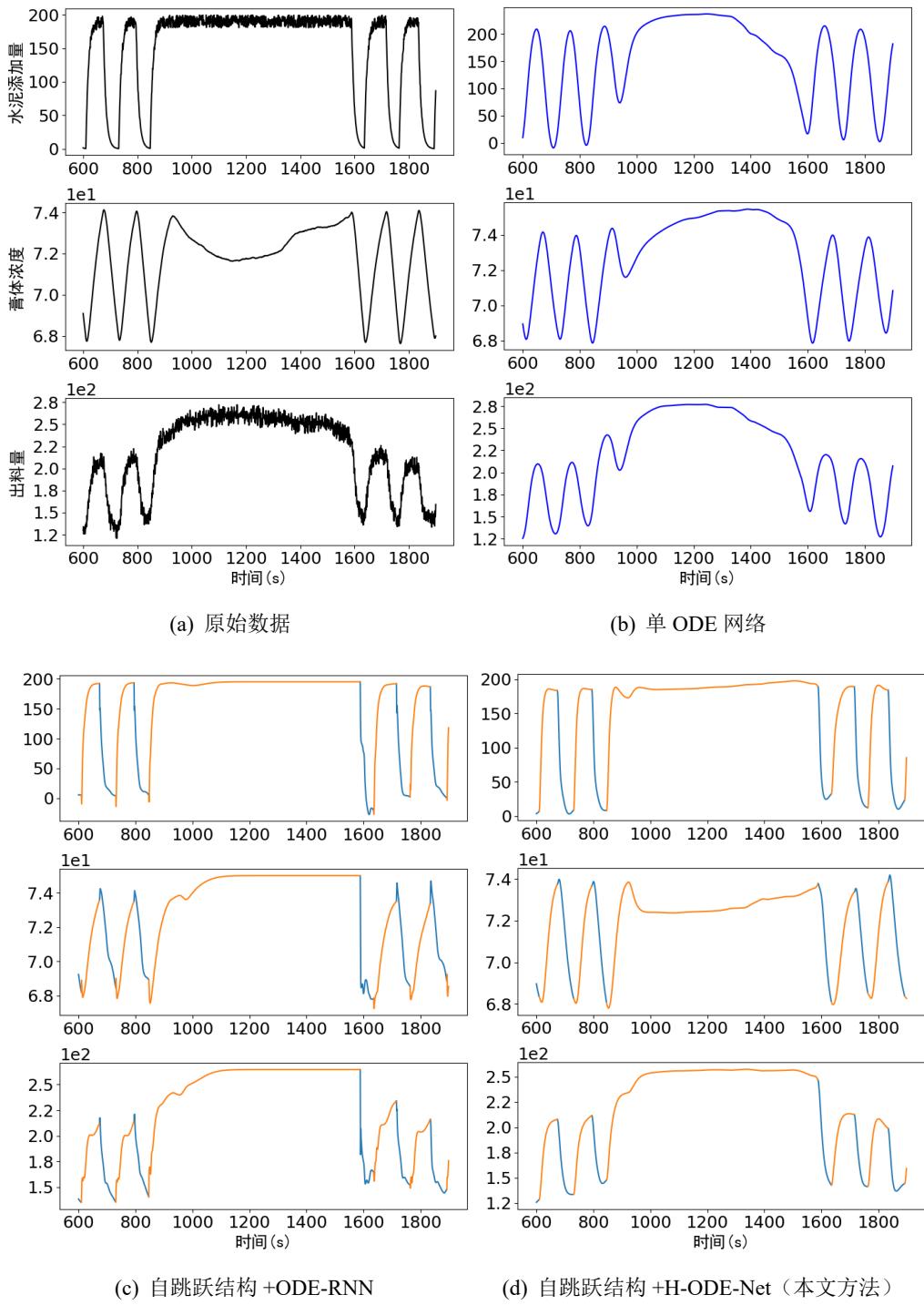


图 6-7 不同模型预测水泥添加量、膏体流量、膏体浓度的效果对比。

系统关闭，**红色**代表稳定制冷阶段。实验结果与膏体制备系统的预测结果类似。单一 ODE 模型时难以对各个阶段间的系统切换边界处给出准确的拟合，对于持续时间较短的阶段转换位置，如阶段 1 与阶段 2，预测效果明显较差。在图 6-8(c) 中，采用 ODE-RNN 替换 AJ-ODE-Net 中的 H-ODE-Net 模块后难以平滑地预测系统输出。图 6-8(d) 中 AJ-ODE-Net 预测的系统输出十分精确，且在阶段转换的边界处有效地识别制冷系统输出的剧烈变化。

接下来，本小节将定量地评估不同模型的预测精度。由于本文关注于开环预测问题，模型需要在预测过程中自发地进行阶段切换。伴随着预测范围长度的增加，模型对于当前阶段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，此时点对点的误差评估不适合作为比较模型预测精度的评价指标。因此，本节分别选取制冷系统的累积功耗预测值的积分以及水泥添加量预测值的积分作为指标，评估模型在长期预测范围内仿真水泥消耗量和系统功耗的精确度。

具体地，对于膏体制备系统数据集，采用经过训练后的模型预测膏体制备系统在预测范围  $[t_I : t_{I+L}]$ （共 120 分钟）下的水泥消耗量。在序列预测结果基础上，对给定预测时间范围内的瞬时水泥量进行分段积分以得到某一时间长度 ( $T$ ) 下水泥累积消耗的预测值。进一步地，可以定义预测值的平均绝对百分比误差 MAPE( $T$ )：

$$\text{MAPE}(T) = \frac{100\%}{n} \sum_{i=1}^{\lfloor \frac{t_{I+L}-t_I}{T} \rfloor} \text{APE}(t_I + T * (i - 1), t_I + T * i) \quad (6-14)$$

其中，绝对百分比误差 (APE) 定义如下：

$$\text{APE}(t_1, t_2) = \left| \frac{\int_{t_1}^{t_2} \hat{y}_1(t) dt - \int_{t_1}^{t_2} y_1(t) dt}{\int_{t_1}^{t_2} y_1(t) dt} \right| \quad (6-15)$$

绝对百分比误差 APE( $t_1, t_2$ ) 衡量了时间范围  $[t_1, t_2]$  内水泥消耗预测值和真实消耗值之间的相对误差。图6-9描述了评估窗口大小  $T$  对于评估结果 MAPE( $T$ ) 的影响。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。当评估窗口大小超过 30 分钟时，MAPE 趋于稳定，说明此时真实输出序列与预测序列之间的阶段起止位置差对于评估长期水泥消耗的影响较小。因次本节对比了  $T = 30$  分钟下，不同模型对于水泥累积消耗的预测结果的 MAPE，结果如表 6-3 所示。通过对比不同模型的预测效果，AJ-ODE-Net 预测结果的 MAPE 稳定低于 2%，显著优于其他三个对比模型。实验充分证明了本章提出的 AJ-ODE-Net 模型及编码器解码器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备系统的水泥累积消耗。

从计算复杂性角度，本节对四个模型在测试集单批数据上执行式(6-8)以及式(6-9)两个过程的时间消耗进行了对比。普通 ODE-Net 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除以该基准以得到相对尺度下的耗

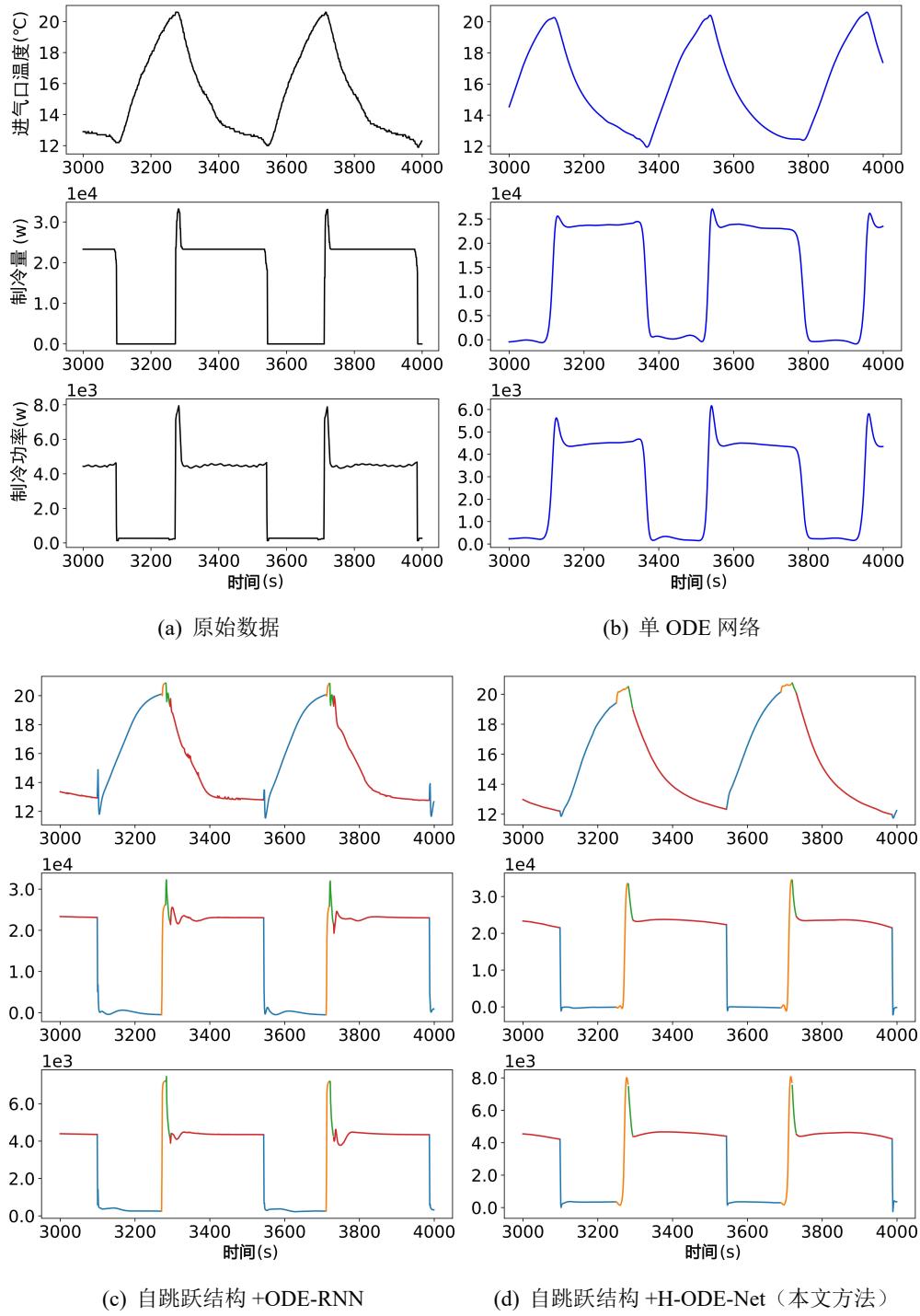


图 6-8 不同模型预测进气口温度、制冷量、制冷机功率的效果对比。

表 6-3 使用不同模型预测水泥消耗量时的精度和推理时间的对比

模型	MAPE	相对时间消耗
AJ-ODE-Net	1.32	3.2
单个 ODE-Net	2.21	1.0
自跳跃结构 +Neural CDE	13.23	3.8
自跳跃结构 +ODE-RNN	10.31	3.4

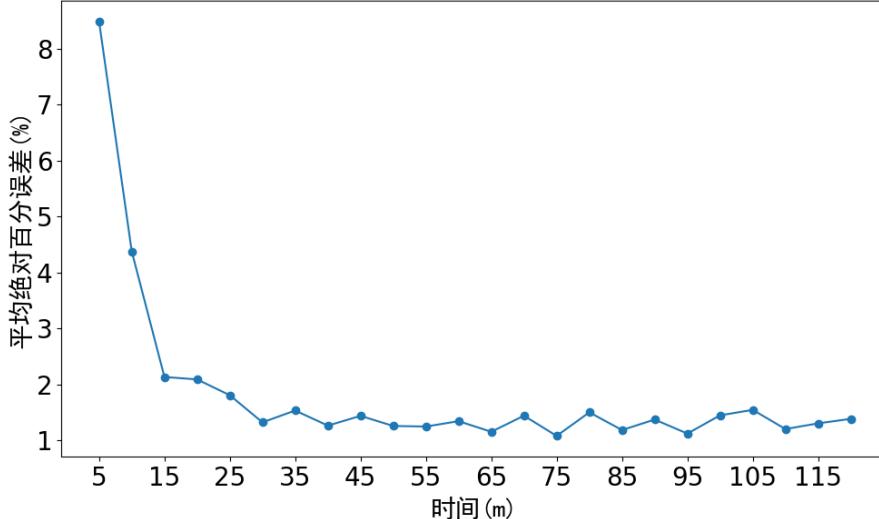


图 6-9 不同时间长度下水泥消耗量预测值的 MAPE 变化

时评价。相应地，普通 ODE-Net 的相对时间消耗为 1.0。对比结果如表6-3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开销，增加的时间主要被用于更新阶段变量及阶段持续时间。

对于制冷系统数据集，本节采用制冷系统累积功耗的预测值作为评价对象，计算对应的 MAPE 指标。对比结果如表6-4所示。对于所有数据集，AJ-ODE-Net 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模型能够精确地仿真不同服务器热负载下制冷系统的累积功耗。

#### 6.5.4 基于预测仿真模型的水泥添加系统启停策略优化

对于水泥添加系统，膏体浓度上下限设定值（上限  $C_{\max}$  和下限  $C_{\min}$ ）是影响添加系统运行以及水泥消耗的关键设置参数。对于本章研究的制备系统，其默认膏体浓度上下限阈值分别为 67%、74%。从数据集可以观察到膏体浓度在上下阈值之间周期变化。由于默认浓度设置未充分考虑实际输入尾矿浆的浓度、流量、水泥价格、设备开关机损耗等因素，因此水泥添加环节的生产成本具有极大的优化空间。在保证膏体浓度满足充填要求的前提下，出于降低水泥消耗以及减少设备磨损的目的，本节保持浓度下阈值不变，通过优化上阈值以削减生产成本。然而，想要精确地找到最佳浓度阈值是困难的。如果上阈值设置过高，每次螺旋给料器打开后需要长时间添加水泥才能使膏体浓度到达阈值，造成物料成本浪费。然而，

表 6-4 不同模型累积能耗预测精度和推理时间的对比

	热负载			
	1.7k	3.8k	4.2k	6.3k
AJ-ODE-Net(本文模型)	<b>4.20</b>	<b>1.51</b>	<b>3.46</b>	<b>2.53</b>
单个 ODE-Net	6.18	19.99	6.24	3.39
自跳跃结构 + 受控微分方程网络 <sup>[19]</sup>	8.25	2.73	6.62	3.87
自跳跃结构 +ODE-RNN	38.26	9.94	10.58	31.90

而如果阈值设置太低，给料器的重启过程将极其频繁，每次设备重启都会带来一定的设备磨损，缩短设备使用寿命，进而增加设备损耗成本。因此盲目地降低浓度上阈值也可能导致生产成本升高。

本节试图寻找最优的浓度上限以最小化充填成本（设备损耗成本 + 物料消耗成本）。形式化地，寻找最优浓度上限  $C_{\max}^*$  可以表示为一单目标优化问题。

$$\begin{aligned} C_{\max}^* = \arg \max_{C_{\max}} & \int_0^T Q * \hat{y}_1(t) dt + \frac{P * |S|}{2} \\ \text{s.t. } & \hat{Y}_{0:T} = F(X_{0:T}, C_{\min}, C_{\max}, \zeta), C_{\min} \leq \hat{y}_3(t) \leq C_{\max} \end{aligned} \quad (6-16)$$

其中  $S = \{t_i | \sigma(t_i) \neq \sigma(t_i^+)\}$  代表阶段发生转换的时间点。 $P$  代表每次给料器开机的设备损耗成本， $Q$  表示水泥价格。 $\hat{Y}_{0:T}$  和  $\hat{y}_1(t)$  分别表示模型预测的系统输出以及预测出的瞬时水泥量。在给定系统输入和恒定下阈值的条件下，被优化变量为控制给料器启动的膏体浓度上阈值，优化目标为最小化累计生产成本。式(6-16)中的  $X_{0:T}$  为测试数据集中的系统输入，其中包括所有时刻的浓密机出料浓度和流量。 $F(\cdot)$  表示在给定系统序列输入和阈值  $C_{\max}$  和  $C_{\min}$  下，预测膏体制备系统的输出。

本节利用上一节 6.5.3 中训练的编码器-解码器 AJ-ODE-Net 模型，在给定不同下阈值设定点的情况下，仿真 30 分钟内的系统水泥添加量、膏体浓度变化、以及 720 秒内的累积充填流量。用于仿真的浓度上阈值设置点从 74% 逐渐降低到 68%，调整间隔为 0.5% $C$ 。

为了模拟不同  $C_{\max}$  下的系统运行过程， $F(\cdot)$  中的预测过程与式6-2中原始训练模型  $f(\cdot)$  略有不同。在阶段 1-开机下的持续时间预测器被特定的转换规则所替代。如表 6-2所示的阶段转换规则，当持续添加水泥使得膏体浓度不断上升直到  $C_{\max}$  时，预测模型立即将阶段变量从 1-开机转换到 0-关闭。尽管 AJ-ODE-Net 中的阶段转换预测器是基于持续时间预测器设计的，为了满足不同运行参数仿真的需要，模型支持将持续时间预测器替换为其他的状态转换规则。虽然在上述模拟过程中的浓度上限与训练数据集对应的阈值参数是不同的，但 AJ-ODE-Net 允许手动调整阶段过渡阈值以支持外推预测。相比于没有引入系统先验的稳态模型，基于先验知识设计的模型具有更好的可扩展性，更便于实现灵活的系统仿真。同时，本章将膏体浓度的变化过程建模成非平稳过程，使得系统在开机阶段内，膏体浓度将会持续地上升，在关机阶段内，膏体浓度会持续地下降。这一特性与系统的先验

知识一致，是保证膏体浓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在某一阶段内难以跳出的情况。

图 6-10 展示了设置不同浓度上限时，膏体制备系统在某一仿真片段（720s）内的水泥添加量变化情况。随着浓度设定点从 68% 逐渐增加到 74%，在相同的时间段内，系统处于水泥添加阶段的时间不断增加，系统运行过程中包含了更少的给料器启停周期。

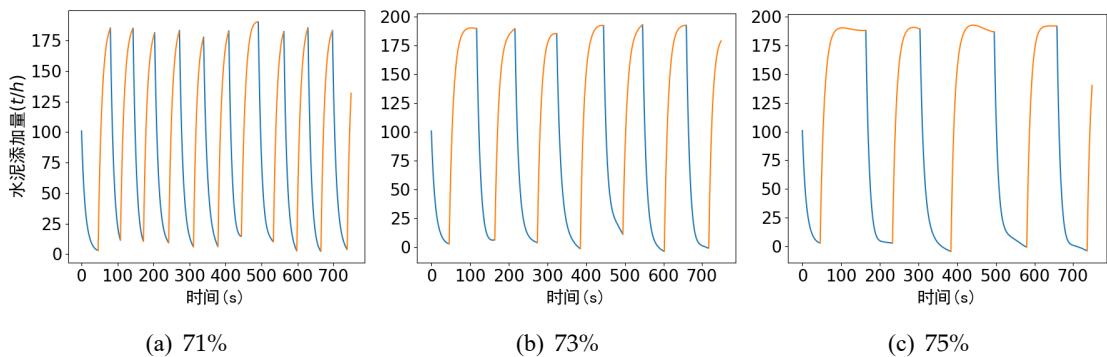


图 6-10 在不同浓度上限设定下的水泥添加量的仿真预测值

图 6-11 展示了在不同  $C_{\max}$  下，制备系统在两小时内的成本预测值，其中包括水泥成本和设备损耗成本。同时，结合实际充填膏体体积计算了单位体积充填成本<sup>1</sup>。

当浓度上阈值从 74% 降低到 68% 时，膏体制备总成本先下降然后逐渐上升。适当降低  $C_{\max}$  时，可以有效地避免水泥的过度添加，从而减少物料成本。但是过度降低  $C_{\max}$  会缩小浓度上下限之间的间距，使得给料器更频繁地重启，显著增加设备损耗成本。当  $C_{\max}$  降低到某一值后，由于给料器频繁重启增加的设备损耗成本与减少水泥过度添加节省的物料成本持平后，此后继续降低  $C_{\max}$  会使总成本增加。这一拐点就是本项工作希望寻找的最佳  $C_{\max}$ 。图 6-11 中标出了使总成本最小的  $C_{\max}$ 。同时，考虑到水泥粉本身具有一定体积，会影响实际充填流量，图中也画出了单位空区体积的充填成本。

利用仿真实验推导出的最优浓度设置点为  $C_{\max}^* = 71\%$ ，采用该上限设定值可以节省约 8.2% 的充填成本。因为实验 6.5.3 中评估了模型预测水泥累积消耗量的精度，其绝对百分比误差小于 2%。可以认为本节对于不同浓度上限设定值的仿真结果以及得到的最优浓度设定值具有足够的可信度。在未来的工作中，将进一步验证该浓度设定策略在实际膏体制备系统中的成本优化效果。

## 6.6 本章小结

本章针对连续时间周期跳变系统的建模预测问题，提出了一种自跳跃常微分方程网络模型，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。

<sup>1</sup> 单位体积充填成本 = (物料成本 + 设备损耗) / 充填膏体体积

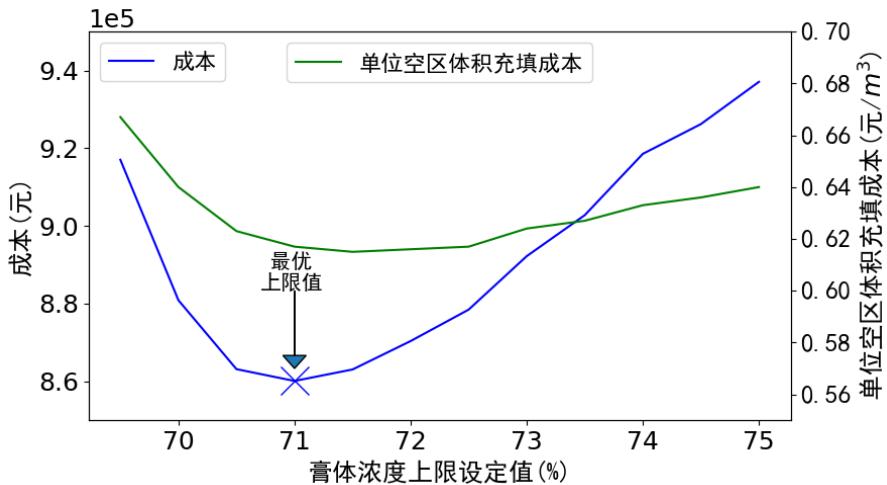


图 6-11 不同浓度上阈值对充填成本的影响

该框架能够对给定的历史序列数据进行编码，通过给定系统的未来序列输入，模型能够预测周期跳变系统的未来阶段变化以及系统输出。为了使模型更好地学习跳变系统在不同阶段内的动态特性，AJ-ODE-Net 包含了多个层次常微分方程网络，每个 H-ODE-Net 能够独立地学习各阶段下的系统动态，并分别建模系统的稳定输出和非稳定输出。同时，AJ-ODE-Net 构建阶段转换预测器以指定各个预测时刻所用的 H-ODE-Net。最后，本章利用所提出的基于 AJ-ODE-Net 的编码器-解码器框架成功建模了两个真实的复杂周期跳变系统。对于其中的膏体制备系统，AJ-ODE-Net 能够很好地预测制备系统的膏体浓度变化、水泥消耗量以及膏体流量变化，并且能够准确地预估系统的阶段转换点。依托于训练得到的水泥量消耗预测模型，成功优化了控制螺旋给料器启停的浓度上限设定点，在满足充填浓度要求的同时，有效减少了设备损耗以及物料消耗，降低了单位体积充填生产成本。仿真结果表明，采纳优化后的浓度阈值可以节省高达 8.2% 的充填成本。

在未来的研究工作中，可以尝试将 AJ-ODE-Net 推广至更通用的马尔科夫连续时间跳变系统，将固定的周期转换过程推广至随机的马尔科夫链。另外，本文采用自定义状态转移规则的方式对原始系统输出序列进行阶段标注。未来将考虑引入无监督统计学习方法，实现更加智能的自动标注<sup>[149]</sup>。

## 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、随机性、多阶段周期转换等特性，探索了采用微分方程网络建模系统动态并实现优化控制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，第三章开创性地提出了基于可微 ODE-Net 的高时延工业多输入输出系统预测模型，该模型由序列编码器、状态解码器、并行样条插值模块和导数模块组成，能够以极低的存储空间消耗从连续时间域角度拟合复杂系统的动态过程。实验环节利用膏体浓密机系统运行数据集对所述模型进行评估，可以发现该模型较好地克服了系统的非线性及长时延特性，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。

针对具有随机性的复杂动态系统建模问题，第四章提出了常微分方程-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辨识具有随机性和长时滞特性的输入输出系统。同时，该章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、批量地推理与训练。同时该章提出的高效隐空间超调技术能够以极低的额外训练开销大幅度改善模型的多步预测性能。实验环节采用三个具有随机性的输入输出系统数据集评估该模型的系统辨识效果，结果表明所提出模型在非均匀数据采样下获得比其他基线模型更好的预测效果，在高随机性、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，第五章提出了基于连续时间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，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。另外该章提出的短期经验回放技术能够有效提升评价网络参数的训练效率。实验环节采用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，使得底流浓度控制精度及稳定性优于原始规则控制策略。

针对具有周期多阶段转换特性的复杂动态系统建模问题，第六章提出了一种基于 ODE-Net 的自跳跃-常微分方程网络 (AJ-ODE-Nets)，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。实验环节使用两个具有周期多阶段特性的系统运行数据集对所提出的 AJ-ODE-Nets 及预测框架进行评估。对于其中的膏体制备系统数据集，模型能够有效地预测螺旋给料器的启停点以及系统输出量。其中，水泥累积消耗量的预测误差小于 2%。同时，该章基于预测仿真模型给出了最优的膏体浓度上限值以优化螺旋给料器的启停策略，有效减少单位体积充填成本 8.2%。

## 7.2 未来工作展望

本文研究了面向复杂流程工业系统的建模方法，并提出了基于辨识模型的优化控制技术。尽管在各章实验环节主要介绍了上述方法在膏体充填场景的应用，但是本文从实际工业系统中抽象出的数学模型在众多领域是有通用性的。因此，本节将对基于微分方程网络的建模及控制技术在其他领域的潜在应用进行展望。

**(1). 电力、能源、化工、矿业等流程工业场景下的系统建模与优化：**包括本文着重探究的膏体充填过程在内，对于电力、能源、化工、矿业等流程工业场景，原料加工成为成品材料的物质转化过程是包含物理化学反应的气液固多相共存的连续化复杂过程，这一连续转化特性与本文关注的连续时间特性是一致的。另外，流程工业系统中广泛存在不同时间尺度、强非线性、多变量强耦合、生产条件变化频繁、随机性、长时延等特性，对于本文提出的基于微分方程与深度神经网络相结合的辨识模型，其模型结构与核心思想在不同工业系统中具有适用性。如何将本文提出的理论、技术在更多流程工业场景进行落地应用，如本文探究的充填系统一样，实现关键指标的预测与建模，并进一步从产品质量、物料成本角度实现优化，是笔者未来研究的重点方向。

**(2). 可微物理仿真技术**物理仿真的经典范式是给定初值或边界条件，根据反映物理规律的微分方程进行推演，以模拟各种物理现象的演变过程。近年来可微物理仿真技术成为了计算机图形学和机器人等领域的研究热点，并取得了广泛应用。作为经典物理仿真的延伸和拓展，可微物理仿真的核心思想是结合自动微分技术，将优化目标梯度信息沿时间反向回传，用于对物理系统中的材料参数、结构设计和控制信号等进行优化提升，以实现特定的设计或控制目标。在计算设计、材料参数估计等领域有着广泛的应用前景。相比于可微物理仿真技术，本文提出的基于微分方程网络的系统建模等价于利用神经元架构描述隐空间中的系统状态变化，二者在微分方程求解、梯度传播、模型训练、建模目标等方面具有很强的相似性，在理论及应用方面的研究成果能够相互借鉴，推动彼此的发展。

**(3). 机械动力学系统建模与优化：**以机器人作为典型代表的动力学系统具有典型的连续时间特性，且相比于流程工业系统，机械动力学系统具备观测空间更完备、外部噪音小等特点，该领域也是研究连续时间系统辨识的主要场景。传统机器人学中主要依靠机理分析的方式对关节点的加速度、阻尼、位移的动态关系进行建模。本文提出的连续时间域建模方法为解决机械动力学系统建模与优化提供了新的研究思路。同时，将动力学系统中已有的、可靠的机理先验与微分方程网络模型进行结合相对容易，部分学者已经围绕这一话题展开研究<sup>[50,50]</sup>，本节在接下来也将对如何实现二者结合这一问题进行讨论与展望。

另外，本文仅围绕常微分方程网络（ODE-Net）进行研究。由于ODE模型仅能描述系统对于时间的确定性导数，目前学界已经开始了其他微分方程系统与神经网络的结合研究，如本文第二章所介绍的随机微分方程网络、受控微分方程网络、以及偏微分方程网络（Partial differential equation network，PDE-Net）。如何利

用上述模型的特异性以针对性地建模客观复杂系统，是未来值得深入研究的科研方向。

在数据驱动建模方法广泛应用之前，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系统机理分析、模型参数估计的工作量。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，成功搭建了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了新的解决思路。笔者在此处给出三种微分方程与机理模型相互结合的方案，并给出未来研究展望。

**(1. 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变。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。结合估计出的参数，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>[150]</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从机理模型传入来自于损失函数的梯度流。如何克服由于机理模型复杂导致的参数不可导、网络难训练的问题是未来极有意义的研究方向。

**(2. 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、模型参数无法精识别、机理先验不够完备的复杂系统，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被充分利用。

**(3. 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] LECUN Y. A Path Towards Autonomous Machine Intelligence Version 0.9. 2, 2022-06-27[EB/OL]. 2022.  
[https://www.infinityexplorers.com/wp-content/uploads/2022/10/a\\_path\\_towards\\_autonomous\\_mach.pdf](https://www.infinityexplorers.com/wp-content/uploads/2022/10/a_path_towards_autonomous_mach.pdf).
- [2] ZADEH L. On the identification problem[J]. IRE Transactions on Circuit Theory, 1956, 3(4): 277–281.
- [3] LE-YI W, WEN-XIAO Z. System identification: new paradigms, challenges, and opportunities[J]. Acta automatica sinica, 2013, 39(7): 933–942.
- [4] 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.
- [5] LJUNG L. Perspectives on system identification[J]. Annu. Rev. Control., 2010, 34(1): 1–12.
- [6] LJUNG L, HJALMARSSON H, OHLSSON H. Four encounters with system identification[J]. European Journal of Control, 2011, 17(5-6): 449–471.
- [7] LJUNG L, ANDERSSON C, TIELS K, et al. Deep learning and system identification[J]. IFAC-PapersOnLine, 2020, 53(2): 1175–1181.
- [8] JORDAN M I, RUMELHART D E. Forward models: Supervised learning with a distal teacher[J]. Cognitive science, 1992, 16(3): 307–354.
- [9] 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.
- [10] WERBOS P J. Neural networks for control and system identification[C] // Proceedings of the 28th IEEE Conference on Decision and Control,. 1989: 260–265.
- [11] 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.
- [12] 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.
- [13] SUTTON R S, BARTO A G. Reinforcement learning: An introduction[M]. [S.l.]: MIT press, 2018.
- [14] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. nature, 2015, 521(7553): 436–444.
- [15] 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.
- [16] SHUMWAY R H, STOFFER D S, STOFFER D S. Time series analysis and its applications : Vol 3[M]. [S.l.] : Springer, 2000.
- [17] 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.
- [18] KIDGER P. On Neural Differential Equations[D]. [S.l.] : University of Oxford, 2021.
- [19] KIDGER P, MORRILL J, FOSTER J, et al. Neural Controlled Differential Equations for Irregular Time Series[C] // NeurIPS 2020, December 6-12, 2020, virtual. 2020.
- [20] 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.
- [21] LI Z, KOVACHKI N B, AZIZZADENESHELI K, et al. Fourier Neural Operator for Parametric Partial Differential Equations[C] // International Conference on Learning Representations. 2020.
- [22] DEMEESTER T. System identification with time-aware neural sequence models[C] // Proceedings of the AAAI Conference on Artificial Intelligence : Vol 34. 2020 : 3757–3764.
- [23] QUAGLINO A, GALLIERI M, MASCI J, et al. SNODE: Spectral Discretization of Neural ODEs for System Identification[C] // 8th International Conference on Learning Representations, ICLR 2020. 2020.
- [24] WANG Y, ZHANG Y, NÖRTERSHÄUSER D, et al. Model and data driven transient thermal system modelings for contained data centers[J]. Energy and Buildings, 2022, 258 : 111790.
- [25] BALENZUELA M P, WILLS A G, RENTON C, et al. Parameter estimation for Jump Markov Linear Systems[J]. Automatica, 2022, 135 : 109949.
- [26] ASHLEY T T, ANDERSSON S B. A Sequential Monte Carlo framework for the system identification of jump Markov state space models[C/OL] // 2014 American Control Conference. 2014 : 1144–1149.

- [http://dx.doi.org/10.1109/ACC.2014.6859280.](http://dx.doi.org/10.1109/ACC.2014.6859280)
- [27] OPPER M, SANGUINETTI G. Variational inference for Markov jump processes[C] // NIPS. [S.l.] : Curran Associates, Inc., 2007 : 1105 – 1112.
- [28] WATTER M, SPRINGENBERG J T, BOEDECKER J, et al. Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images[C] // NIPS. 2015 : 2746 – 2754.
- [29] FRACCARO M, KAMRONN S, PAQUET U, et al. A Disentangled Recognition and Nonlinear Dynamics Model for Unsupervised Learning[C] // GUYON I, LUXBURG U V, BENGIO S, et al. Advances in Neural Information Processing Systems : Vol 30. [S.l.] : Curran Associates, Inc., 2017.
- [30] NASON G P. Stationary and non-stationary time series : Vol 60[M]. [S.l.] : Geological Society of London, London, 2006.
- [31] SUTTON R S, BARTO A G. Reinforcement learning: An introduction[M]. [S.l.] : MIT press, 2018.
- [32] LEWIS F L, VRABIE D, SYRMOS V L. Optimal control[M]. [S.l.] : John Wiley & Sons, 2012.
- [33] MOERLAND T M, BROEKENS J, JONKER C M. Model-based reinforcement learning: A survey[EB/OL]. <https://arxiv.org/abs/2006.16712> : arXiv preprint arXiv:2006.16712, 2020.
- [34] LIN L-J, MITCHELL T M. Memory approaches to reinforcement learning in non-Markovian domains[M]. [S.l.] : Citeseer, 1992.
- [35] 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.
- [36] 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.
- [37] ZAMARREÑO J M, VEGA P. State space neural network. Properties and application[J]. Neural networks, 1998, 11(6) : 1099 – 1112.
- [38] LAI M. Giraffe: Using deep reinforcement learning to play chess[EB/OL]. 2015.
- [39] FRANÇOIS-LAVET V, BENGIO Y, PRECUP D, et al. Combined reinforcement learning via abstract representations[C] // Proceedings of the AAAI Conference on Artificial Intelligence : Vol 33. 2019 : 3582 – 3589.
- [40] SILVER D, SCHRIJTTWIESER J, SIMONYAN K, et al. Mastering the game of go without human knowledge[J]. nature, 2017, 550(7676) : 354 – 359.
- [41] SUTTON R S. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming[G] // Machine learning proceed-

- ings 1990. [S.l.] : Elsevier, 1990 : 216 – 224.
- [42] DEGRAVE J, HERMANS M, DAMBRE J, et al. A differentiable physics engine for deep learning in robotics[J]. *Frontiers in neurorobotics*, 2019 : 6.
- [43] de AVILA BELBUTE-PERES F, SMITH K A, ALLEN K R, et al. End-to-End Differentiable Physics for Learning and Control[C] // NeurIPS. 2018 : 7178 – 7189.
- [44] LEVINE S, ABBEEL P. Learning Neural Network Policies with Guided Policy Search under Unknown Dynamics[C] // NIPS. 2014 : 1071 – 1079.
- [45] HAFNER D, LILLICRAP T P, BA J, et al. Dream to Control: Learning Behaviors by Latent Imagination[C] // ICLR. [S.l.] : OpenReview.net, 2020.
- [46] MISHRA N, ABBEEL P, MORDATCH I. Prediction and control with temporal segment models[C] // International Conference on Machine Learning. 2017 : 2459 – 2468.
- [47] GRATHWOHL W, CHEN R T Q, BETTENCOURT J, et al. FFJORD: Free-Form Continuous Dynamics for Scalable Reversible Generative Models[C] // 7th International Conference on Learning Representations, ICLR 2019. [S.l.] : OpenReview.net, 2019.
- [48] RACKAUCKAS C, MA Y, MARTENSEN J, et al. Universal differential equations for scientific machine learning[EB/OL]. <https://arxiv.org/abs/2001.04385> : arXiv preprint arXiv:2001.04385, 2020.
- [49] 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.
- [50] ZHONG Y D, DEY B, CHAKRABORTY A. Symplectic ODE-Net: Learning Hamiltonian Dynamics with Control[C] // International Conference on Learning Representations. 2019.
- [51] AYED I, de BÉZENAC E, PAJOT A, et al. Learning dynamical systems from partial observations[EB/OL]. <https://arxiv.org/abs/1902.11136> : arXiv preprint arXiv:1902.11136, 2019.
- [52] RAMADHAN A, MARSHALL J, SOUZA A, et al. Capturing missing physics in climate model parameterizations using neural differential equations[EB/OL]. 2020.
- [53] YUAN Z, LI X, WU D, et al. Continuous-Time Prediction of Industrial Paste Thickener System With Differential ODE-Net[J]. *IEEE/CAA Journal of Automatrica Sinica*, 2022, 9(4) : 686 – 698.
- [54] RUBANOVA Y, CHEN R T, DUVENAUD D. Latent ODEs for irregularly-sampled time series[J], 2019, 32.
- [55] ZHUANG J, DVORNEK N, LI X, et al. Adaptive checkpoint adjoint method for gradient estimation in neural ode[C] // International Conference on Machine Learn-

- ing. 2020 : 11639–11649.
- [56] FINLAY C, JACOBSEN J-H, NURBEKYAN L, et al. How to train your neural ODE[C] // International Conference on Machine Learning. 2020 : 3154–3164.
- [57] KELLY J, BETTENCOURT J, JOHNSON M J, et al. Learning Differential Equations that are Easy to Solve[C] // NeurIPS. 2020.
- [58] GHOSH A, BEHL H S, DUPONT E, et al. STEER : Simple Temporal Regularization For Neural ODE[C] // NeurIPS. 2020.
- [59] 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.
- [60] 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.
- [61] LECHNER M, HASANI R. Learning long-term dependencies in irregularly-sampled time series[EB/OL]. <https://arxiv.org/abs/2006.04418> : arXiv preprint arXiv:2006.04418, 2020.
- [62] 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.
- [63] MORRILL J, KIDGER P, YANG L, et al. Neural Controlled Differential Equations for Online Prediction Tasks[EB/OL]. <https://arxiv.org/abs/2106.11028> : arXiv preprint arXiv:2106.11028, 2021.
- [64] HUANG Z, SUN Y, WANG W. Learning Continuous System Dynamics from Irregularly-Sampled Partial Observations[C] // NeurIPS. 2020.
- [65] VASWANI A, SHAZEE N, PARMAR N, et al. Attention is All you Need[C] // NIPS. 2017 : 5998–6008.
- [66] ZHOU H, ZHANG S, PENG J, et al. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting[C] // AAAI. 2021 : 11106–11115.
- [67] WU H, XU J, WANG J, et al. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting[C] // NeurIPS. 2021 : 22419–22430.
- [68] CHEN Y-H, CHIEN J-T. Continuous-time attention for sequential learning[C] // Proc. of AAAI Conference on Artificial Intelligence. 2021.
- [69] LIU X, YU H-F, DHILLON I, et al. Learning to encode position for transformer with continuous dynamical model[C] // International conference on machine learning. 2020 : 6327–6335.

- [70] FRACCARO M, SØNDERBY S K, PAQUET U, et al. Sequential Neural Models with Stochastic Layers[C] // NIPS. 2016 : 2199–2207.
- [71] CHUNG J, KASTNER K, DINH L, et al. A Recurrent Latent Variable Model for Sequential Data[C] // NIPS. 2015 : 2980–2988.
- [72] KARL M, SOELCH M, BAYER J, et al. Deep Variational Bayes Filters: Unsupervised Learning of State Space Models from Raw Data[C] // ICLR (Poster). 2017.
- [73] KINGMA D P, WELLING M. Auto-Encoding Variational Bayes[C] // BENGIO Y, LECUN Y. 2nd International Conference on Learning Representations, ICLR 2014, Conference Track Proceedings. 2014.
- [74] DOERR A, DANIEL C, SCHIEGG M, et al. Probabilistic recurrent state-space models[C] // International Conference on Machine Learning. 2018 : 1280–1289.
- [75] YILDIZ C, HEINONEN M, LAHDESMAKI H. ODE2VAE: Deep generative second order ODEs with Bayesian neural networks[C] // WALLACH H, LAROCHELLE H, BEYGELZIMER A, et al. Advances in Neural Information Processing Systems : Vol 32. [S.l.] : Curran Associates, Inc., 2019.
- [76] HAFNER D, LILLICRAP T, FISCHER I, et al. Learning latent dynamics for planning from pixels[C] // International conference on machine learning. 2019 : 2555–2565.
- [77] JIA J, BENSON A R. Neural Jump Stochastic Differential Equations[C] // NeurIPS. 2019 : 9843–9854.
- [78] LIU Y, XING Y, YANG X, et al. Continuous-Time Stochastic Differential Networks for Irregular Time Series Modeling[C] // International Conference on Neural Information Processing. 2021 : 343–351.
- [79] YANG H, YIN S. Actuator and Sensor Fault Estimation for Time-Delay Markov Jump Systems With Application to Wheeled Mobile Manipulators[J]. IEEE Transactions on Industrial Informatics, 2020, 16(5) : 3222–3232.
- [80] 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]. IEEE Transactions on Industrial Informatics, 2021, 17(2) : 913–921.
- [81] 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[J]. PMLR, 2020, 120 : 947–957.
- [82] FANG Y, LOPARO K A. Stabilization of continuous-time jump linear systems[J]. IEEE Transactions on Automatic Control, 2002, 47(10) : 1590–1603.
- [83] 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.

- [84] 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.
- [85] POWELL W B. Approximate Dynamic Programming: Solving the curses of dimensionality : Vol 703[M]. [S.l.] : John Wiley & Sons, 2007.
- [86] ZHANG H, LIU D, LUO Y, et al. Adaptive dynamic programming for control: algorithms and stability[M]. [S.l.] : Springer Science & Business Media, 2012.
- [87] 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.
- [88] 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.
- [89] 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.
- [90] 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.
- [91] 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.
- [92] MODARES H, LEWIS F L, JIANG Z-P.  $H_\infty$  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.
- [93] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. nature, 2015, 518(7540): 529–533.
- [94] SPIELBERG S, GOPALUNI R, LOEWEN P. Deep reinforcement learning approaches for process control[C] // 2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP). 2017: 201–206.
- [95] 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.
- [96] KIM J W, PARK B J, YOO H, et al. Deep reinforcement learning based finite-horizon optimal tracking control for nonlinear system[C] // IFAC : Vol 51. [S.l.] : Elsevier, 2018: 257–262.
- [97] 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.
- [98] YILDIZ C, HEINONEN M, LÄHDESMÄKI H. Continuous-time Model-based Reinforcement Learning[C] // International Conference on Machine Learning. [S.l.] : PMLR, 2021 : 12009–12018.
- [99] AINSWORTH S K, LOWREY K, THICKSTUN J, et al. Faster Policy Learning with Continuous-Time Gradients[C] // Proceedings of Machine Learning Research, Vol 144 : L4DC. [S.l.] : PMLR, 2021 : 1054–1067.
- [100] 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.
- [101] NÚÑEZ F, LANGARICA S, DÍAZ P, et al. Neural Network-Based Model Predictive Control of a Paste Thickener over an Industrial Internet Platform[J]. IEEE Transactions on Industrial Informatics, 2020, 16(4) : 2859–2867.
- [102] 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.
- [103] LARSSON E K, SÖDERSTRÖM T. Identification of continuous-time AR processes from unevenly sampled data[J]. Automatica, 2002, 38(4) : 709–718.
- [104] 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.
- [105] LIU J, WU N, 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.
- [106] 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.
- [107] NEUD 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.
- [108] 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.
- [109] DEMEESTER T. System identification with time-aware neural sequence models[C] // Proceedings of the AAAI Conference on Artificial Intelligence : Vol 34. 2020 : 3757–3764.

- [110] 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.
- [111] 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.
- [112] 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.
- [113] 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.
- [114] CHRISTOFFERSEN P F. Forecasting Non-Stationary Economic Time Series[J]. Journal of the American Statistical Association, 2001, 96(453) : 347 – 347.
- [115] KINGMA D P, BA J. Adam: A Method for Stochastic Optimization[C] // ICLR (Poster). 2015.
- [116] BOGACKI P, SHAMPINE L F. A 3 (2) pair of Runge-Kutta formulas[J]. Applied Mathematics Letters, 1989, 2(4) : 321 – 325.
- [117] WU S, XIAO X, DING Q, et al. Adversarial Sparse Transformer for Time Series Forecasting[C] //NeurIPS. 2020.
- [118] 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.
- [119] POLI M, MASSAROLI S, YAMASHITA A, et al. Hypersolvers: Toward Fast Continuous-Depth Models[C] // NeurIPS. 2020.
- [120] JIA J, BENSON A R. Neural jump stochastic differential equations[J]. Advances in Neural Information Processing Systems, 2019, 32.
- [121] VENKATRAMAN A, HEBERT M, BAGNELL J A. Improving multi-step prediction of learned time series models[C] // Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015.
- [122] YILDIZ C, HEINONEN M, LÄHDESMÄKI H. Continuous-time Model-based Reinforcement Learning[C] // International Conference on Machine Learning. 2021 : 12009 – 12018.
- [123] 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.
- [124] KOURO S, CORTÉS P, VARGAS R, et al. Model predictive control—A simple and powerful method to control power converters[J]. IEEE Transactions on indus-

- trial electronics, 2008, 56(6) : 1826 – 1838.
- [125] 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.
- [126] 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.
- [127] PROKHOROV D V, WUNSCH D C. Adaptive critic designs[J]. IEEE transactions on Neural Networks, 1997, 8(5) : 997 – 1007.
- [128] 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.
- [129] 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.
- [130] 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.
- [131] 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.
- [132] LI C, DING J, LEWIS F L, et al. A novel adaptive dynamic programming based on tracking error for nonlinear discrete-time systems[J]. Automatica, 2021, 129 : 109687.
- [133] 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.
- [134] 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.
- [135] 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.
- [136] 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.

- [137] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing Atari with Deep Reinforcement Learning[EB/OL]. 2013.  
<http://arxiv.org/abs/1312.5602>.
- [138] GUO X, YAN W, CUI R. Integral Reinforcement Learning-Based Adaptive NN Control for Continuous-Time Nonlinear MIMO Systems With Unknown Control Directions[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, PP : 1 – 10.
- [139] 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.
- [140] 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.
- [141] 王猛. 矿浆中和沉降分离过程模型软件的研发 [D]. [S.l.]: 东北大学, 2011.
- [142] 唐谟堂. 湿法冶金设备 [M]. [S.l.]: 湿法冶金设备, 2009.
- [143] 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.
- [144] 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.
- [145] 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.
- [146] DU S, LI T, YANG Y, et al. Multivariate time series forecasting via attention-based encoder–decoder framework[J]. Neurocomputing, 2020, 388 : 269 – 279.
- [147] 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.
- [148] 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.
- [149] HALLAC D, VARE S, BOYD S P, et al. Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data[C] // KDD. [S.l.] : ACM, 2017 : 215 – 223.
- [150] 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.



## 附录

各章研究内容相对应的开源代码如下：

第三章高时延复杂工业系统建模与预测

- <https://github.com/y18810919727/UnderflowConcentrationseq2seq>

第四章随机性系统建模与预测

- <https://github.com/y18810919727/system-modelling>

第五章连续时间域下的有模型强化学习控制

- [https://github.com/y18810919727/Control\\_Exp1001/tree/master/Control\\_Exp1001/demo](https://github.com/y18810919727/Control_Exp1001/tree/master/Control_Exp1001/demo)

第六章周期跳变系统建模及优化

- <https://github.com/y18810919727/cooling>



## 作者简历及在学研究成果

### 一、作者入学前简历

起止年月	学习或工作单位	备注
2013 年 9 月至 2017 年 7 月	在北京科技大学计算机科学与技术 专业攻读学士学位	

### 二、在学期间从事的科研工作

1. 国家重点研发计划重点专项：基于大数据的金属矿开采装备智能管控技术研发与示范(编号：2019YFC0605300)，主要参与人员，2019.12-2022.12。

### 三、在学期间所获的科研奖励

1. 2020 年中国黄金协会科学技术一等奖
2. 2022 年北京科技大学太湖人才王道科技奖奖学金
3. 2017 年国际大学生程序设计竞赛 ACM-ICPC 亚洲区域赛（西安站）金奖
4. 2017 年国际大学生程序设计竞赛 ACM-ICPC 亚洲区域赛（青岛站）银奖

### 四、在学期间发表的论文

- [1] **Yuan Z** , et al. ODE-RSSM: Learning Stochastic Recurrent State Space Model from Irregularly Sampled Data. In Proceedings of the 37th AAAI Conference on Artificial Intelligence, 2023. (CCF-A 类推荐英文会议)
- [2] **Yuan Z**, Wang Y, Ban X, et al. Autonomous-Jump-ODENet: Identifying Continuous-Time Jump Systems for Cooling-System Prediction[J]. IEEE Transactions on Industrial Informatics, 2022, doi: 10.1109/TII.2022.3207835.(已录用. SCI 检索. 中科院分区小类 1 区. IF=11.648)
- [3] **Yuan Z**, Li X, Wu D, et al. Continuous-time prediction of industrial paste thickener system with differential ODE-net[J]. IEEE/CAA Journal of Automatica Sinica, 2022, 9(4): 686-698.(已检索. SCI 检索. 中科院分区小类 1 区. IF=7.847. 检索号：WOS:000766623800011)
- [4] **Yuan Z**, Yuan Z, Ban X, Han F, et al. Integrated three-dimensional visualization and soft-sensing system for underground paste backfilling[J]. Tunnelling and Underground Space Technology, 2022, 127: 104578.(已检索. SCI 检索. 中科院分区小类 1 区. IF=6.407. 检索号：WOS: 000811527500003)

- [5] 袁兆麟, 何润姿, 姚超, 等. 基于强化学习的浓密机底流浓度在线控制算法 [J]. 自动化学报, 2021, 47(7): 1558-1571.(已检索. EI 刊源. CCF-A 类中文期刊)
- [6] 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. (已检索. SCI 检索. 中科院分区小类 3 区. IF=3.847. 检索号: WOS: 000525271500022)
- [7] Yuan Z, Ban X, Hu J. Improving Word Representation Quality Trained by word2vec via a More Efficient Hierarchical Clustering Method[C]//International Conference on Cooperative Design, Visualization and Engineering. Springer, Cham, 2018: 299-303. (EI 会议)
- [8] Li J, Yuan Z, Ban X. An Improved Reinforcement Learning Based Heuristic Dynamic Programming Algorithm for Model-Free Optimal Control[C]//International Conference on Artificial Neural Networks. Springer, Cham, 2020: 282-294. (CCF-C 类推荐会议)
- [9] Wang H, Yuan Z, Chen Y, et al. An industrial missing values processing method based on generating model[J]. Computer Networks, 2019, 158: 61-68.(已检索. SCI 检索. 中科院分区小类 3 区. IF=5.493. 检索号: WOS:000472243200005)
- [10] 李潇睿, 班晓娟, 袁兆麟, 乔浩然. 工业场景下基于深度学习的时序预测方法及应用 [J]. 工程科学学报, 2022, 44(4): 757-766. doi: 10.13374/j.issn2095-9389.2021.12.02.004

## 五、在学期间申请及授权的发明专利

- [1] 班晓娟; 韩方圆; 曹宇宁; 袁兆麟; 胡国斌; 肖金林; 张金忠. 一种膏体充填进度实时测量与可视化方法及系统: 中国, 202010997613.2 [P]. 2020.09.21
- [2] 班晓娟; 何润姿; 袁兆麟; 刘婷; 王贻明. 一种充填场景下的深锥浓密机智能控制方法: 中国, 201910373119.6 [P]. 2019.08.06.
- [3] 班晓娟; 袁兆麟; 刘婷; 李佳; 何润姿. 一种基于强化学习的浓密机在线控制方法: 中国, 201910636652.7 [P]. 2019.11.01.
- [4] 班晓娟、袁兆麟、李佳、姚松、李潇睿、沈家华、刘璞. 一种基于连续时间神经网络的浓密机预测控制方法及系统: 中国, 202011493186.0 [P]. 2020.12.16.
- [5] 吴爱祥; 周佳城; 马博渊; 袁兆麟; 王贻明; 阮竹恩. 一种膏体浓度非接触式自动检测方法: 中国, 202011026025.0 [P]. 2020.09.25.
- [6] 班晓娟、刘婷、袁兆麟、王贻明、王青海、赵占斌一种针对工业监测数据缺失的补全方法及补全装置: 中国, 201910055378.4 [P]. 2020.09.25.
- [7] 黄海友; 袁兆麟; 马博渊; 胡金龙; 魏晓燕; 刘婷一种文本信息自动提取方法 201810975598.4 [P]. 2018.08.24
- [8] 班晓娟; 刘婷; 袁兆麟; 王笑琨; 王贻明; 阮竹恩. 一种充填管道裂纹检测方法: 中国, 202010544835.9 [P]. 2020.06.15.

## 独创性说明

本人郑重声明：所呈交的论文是我个人在导师指导下进行的研究工作及取得研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写的研究成果，也不包含为获得北京科技大学或其他教育机构的学位或证书所使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

签名: \_\_\_\_\_ 日期: \_\_\_\_\_

## 关于论文使用授权的说明

本人完全了解北京科技大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

**(保密的论文在解密后应遵循此规定)**

签名: \_\_\_\_\_ 导师签名: \_\_\_\_\_ 日期: \_\_\_\_\_



## 学位论文数据集

<b>关键词*</b>	<b>密级*</b>	<b>中图分类号*</b>	<b>UDC</b>	<b>论文资助</b>
微分方程网络	公开	TP391.8		
<b>学位授予单位名称*</b>		<b>学位授予单位代码*</b>	<b>学位类别*</b>	<b>学位级别*</b>
北京科技大学		10008	工学	博士
<b>论文题名*</b>		<b>并列题名</b>		<b>论文语种*</b>
基于深度微分方程网络的复杂 动态系统建模与控制				汉语
<b>作者姓名*</b>	袁兆麟		<b>学号*</b>	B20170324
<b>培养单位名称*</b>		<b>培养单位代码*</b>	<b>培养单位地址</b>	<b>邮编</b>
北京科技大学		10008	北京市海淀区 学院路 30 号	100083
<b>学科专业*</b>		<b>研究方向*</b>	<b>学制*</b>	<b>学位授予年*</b>
计算机科学与技术		智能控制	5	2023
<b>论文提交日期*</b>	2020 年 12 月 12 日			
<b>导师姓名*</b>	班晓娟		<b>职称*</b>	教授
<b>评阅人</b>	<b>答辩委员会主席*</b>		<b>答辩委员会成员</b>	
	陈益强		田丰、贺威、殷绪成、罗熊、曾广平	
<b>电子版论文提交格式</b> 文本 <input checked="" type="checkbox"/> 图像 <input type="checkbox"/> 视频 <input type="checkbox"/> 音频 <input type="checkbox"/> 多媒体 <input type="checkbox"/> 其他 <input type="checkbox"/>				
<b>推荐格式:</b> application/msword; application/pdf				
<b>电子版论文出版(发布)者</b>		<b>电子版论文出版(发布)地</b>		<b>权限声明</b>
<b>论文总页数*</b>	122			
共 33 项，其中带*为必填数据，为 22 项。				