

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 03 月 20 日

基于深度微分方程网络的复杂动态系统建模与控制

Modeling and Controlling for complicated dynamical
system based on deep differential equation

研究生姓名：袁兆麟

指导教师姓名：班晓娟

北京科技大学土木与环境工程学院

北京 100083，中国

Doctor Degree Candidate: Zhaolin Yuan

Supervisor: Xiaojuan Ban

School of Civil and Environmental 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

UDC:

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

论文提交日期: 2022 年 03 月 20 日

学位授予单位: 北 京 科 技 大 学

致 谢

在此我要感谢

摘 要

本文基于 \LaTeX 使用手册，依据北京科技大学博士学位论文 Word 模板，

关键词： \LaTeX ，博士论文，北京科技大学

Abstract

Based on the \LaTeX manual and referred to USTB word model for Doctor of Philosophy, I

Key Words: \LaTeX , Doctor of Philosophy Thesis, USTB

序 言

由于北京科技大学的论文模板在 \LaTeX 方面存在空白，因此此项目具有重大的意义。

目 录

致谢	I
摘要	III
Abstract	V
序言	VII
表格清单	IX
插图清单	XI
1 基于自跳跃常微分方程网络的连续时间跳变系统建模	XV
1.1 前置工作及问题形式化描述	XVI
1.2 基于层次 ODE-Net 模型的复杂系统稳定及非稳定输出混合预测	XVII
1.3 自跳跃常微分方程网络	XIX
1.3.1 状态定义	XIX
1.3.2 基于时间预测器的阶段自转移	XX
1.4 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	XXI
1.5 模型训练	XXII
1.6 周期性制冷系统预测仿真及系统优化	XXIV
1.6.1 制冷系统简介	XXV
1.6.2 数据预处理：离散序列插及阶段标注	XXVI
1.6.3 模型训练	XXVII
1.6.4 应用研究 1: 运行变量仿真	XXVIII
1.6.5 应用研究 2: 制冷系统进气口温度设定点优化	XXXI
1.7 本章小结	XXXIII
参考文献	XXXVI

表格清单

1-1	跳变系统在四种状态下循环, 每种状态对应制冷系统的一个阶段 . .	XXVII
1-2	系统输入输出, 不同负载下的数据集和模型的操作	XXVIII
1-3	The comparison of energy consumption prediction and the relative com- putational efficiency	XXX
1-4	optimized power	XXXIII

插图清单

1-1	H-ODEnet 结构图示	XVIII
1-2	AJ-ODEnet 模型结构	XIX
1-3	基于 AJ-ODEnet 的编码器-解码器预测框架	XXI
1-4	时间预测器的损失函数定义	XXIV
1-5	工业制冷系统整体结构及热交换过程	XXV
1-6	在平均负载不同的四个数据集中, 阶段启动和关闭阶段的持续时间 分布箱线图	XXVI
1-7	制冷系统运行原理概述	XXVII
1-8	不同模型预测进气口温度, 制冷量, 制冷机功率的效果对比	XXIX
1-9	预测不同时间长度的功耗的 MAPE 的变化	XXXI
1-10	在不同温度下限设定值下的进气口温度的仿真值	XXXIII
1-11	在不同热负载下改变温度设定下限值对功耗、COP、PUE 的影响	XXXV

1 基于自跳跃常微分方程网络的连续时间跳变系统建模

周期性跳变系统在现实生活中随处可见，其运行过程具有周期循环性。每个周期内，各阶段之间按特定顺序进行切换。系统在不同阶段会呈现明显不同的动态特性。例如，洗衣机启动后将按程序设定，周期性地在进水，洗涤，排水，甩干各个阶段之间循环运行，直至最后关机。冰箱和空调在工作期间会在运行（压缩机开启）和待机（压缩机关闭）两种状态之间不断进行切换。在一些自然界中其他的物理过程，也存在类似的系统，例如建筑物内的空调制冷系统、交通流量和季节变化等。例如，在对城市交通流量或某产品销量进行预测时，流量或销量通常伴随着特定的日期和季节变化而改变，并且其周期长度相对稳定，模型能够较容易地识别各预测时刻被预测对象所处的阶段。相反，对于一些复杂工业系统，其周期长度可能受到系统内部和外部多个变量的影响。例如系统的外部输入、当前内部状态、环境条件等。利用历史运行数据识别以及建模此类跳变系统对于理解和优化系统运行过程是极其重要的。然而，在面向跳变系统进行建模时面临两大技术难题。首先，跳变系统通常有多个阶段，每个阶段呈现不同的非线性特性。各个阶段的持续时间可能同时受到内部因素和外部因素的影响^[2]。目前，面向跳变系统参数估计方法^[2]通常依赖于对系统结构以及持续时间分布的先验认知，这需要大量的领域专家知识作为支撑。另外，对于带有多输出项的工业系统，其输出过程可能由多个不同的物理特性共同影响。这会导致多输出项中同时存在稳定和非稳定过程^[1]。据我们所知，对于现有的未经过特定设计的现存模型，难以解决此类带有混合时序特性的系统学习任务。

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

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

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

本章研究主要包含三方面贡献，概括如下：

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

本章的结构组织如下：本章1.1节给出了变量符号定义以及描述了周期性多阶段系统预测问题的形式化描述。本章1.2节介绍了分层常微分方程网络的定义及结构，以及如何将其用于学习同时带有稳定和非稳定时间序列输出的动态过程。本章1.3介绍了 AJ-ODENet 框架，并详细阐述了连续时间域下的状态转移方程、关键模块结构和持续时间预测器的损失函数定义等内容。本章1.4给出了基于 AJ-ODENet 模型的编码器解码器框架及其训练方法。本章1.6节介绍了将所述框架应用于模拟某实际工业制冷系统。模型能够根据制冷系统的热负载输入对其功耗以及出气口气体温度变化进行模拟预测，同时能够实现不同制冷启动温度配置下的功耗仿真与优化。最后，1.7节对本章工作做出了总结，并对 AJ-ODENet 模型的潜在应用以及未来的研究方向做出了讨论。

1.1 前置工作及问题形式化描述

连续时间跳变系统（Continuous-time Jump System, CTJS）由多个连续时间子系统组成，其内部阶段转换由一个有限状态过程支配^[2]。作为一种特殊的动态系统，CTJS 系统广泛应用于各类应用中，如故障预测^[3]、系统控制^[4]，以及状态估计^[2]。典型的连续时间跳变系统定义形式如下：

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

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

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

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

$$\hat{\mathbf{Y}}_{t_I:t_{I+L}} = f(\mathbf{Y}_{t_1:t_I}, \mathbf{X}_{t_1:t_{I+L}}; \boldsymbol{\zeta}) \quad (1-2)$$

公式中 $\hat{\mathbf{Y}}_{t_I:t_{I+L}}$ 表示在预测范围内估计的输出。 $\mathbf{Y}_{t_1:t_I}$ 表示在条件范围内系统的历史输出。 $\mathbf{X}_{t_1:t_{I+L}}$ 表示完整时间范围内的系统输入, 包括历史部分和预测序列部分。该系统输入在时间序列预测领域经常被称为协变量^[5]。 $\boldsymbol{\zeta}$ 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 $\boldsymbol{\zeta}$ 以最小化 $\|\hat{\mathbf{Y}}_{t_I:t_{I+L}} - \mathbf{Y}_{t_I:t_{I+L}}\|^2$ 。

以制冷系统为例, 可以将任意长度的热负载序列数据作为模型输入, 模型能够仿真预测区间下的温度变化。值得注意的是, 本章仍关注开环预测问题, 模型预测过程中无法收到来自系统输出的反馈。对于存在随机性及非确定性的系统, 长期开环仿真是难以实现精准预测的。因此, 在训练阶段, 预测范围长度与条件范围长度是一致的, 为 800s。在实验一中, 预测范围的长度为 30 分钟, 本章考虑到由于误差累积导致的相位偏差, 因此没有采用点对点式的评估指标, 而是采用预测累积能耗的估计误差对模型可靠性进行评估。

1.2 基于层次 ODE-Net 模型的复杂系统稳定及非稳定输出混合预测

大部分工业系统的输出是多维度的, 不同维度输出的时序统计特性存在一定差异, 某些输出量为均值方差恒定的稳定过程, 某些输出为均值方差不断变化的

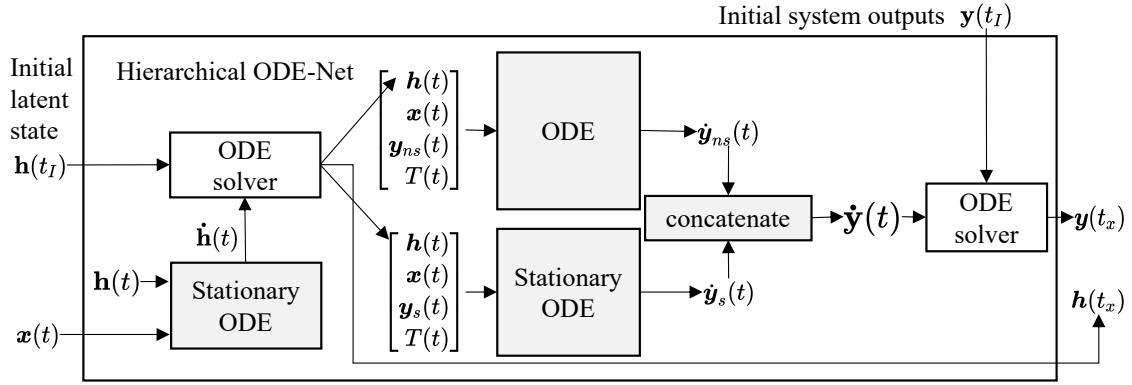


图 1-1 H-ODEnet 结构图示

非稳定过程。对于同时含有稳定和非稳定输出的混合过程进行建模学习具有一定难度，当前主流的基于深度学习的序列预测领域并未针对该类系统开展研究。针对该问题，本章提出了分层常微分方程网络（Hierarchical ODENet, H-ODEnet），用于建模同时包含稳定过程和非稳定过程的系统。

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

其中，L-1 中的 ODEnet 为稳定结构，由门控循环单元 (GRU) 网络实现^[6]:

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

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

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

$$\frac{dy_{ns}(t)}{\tau} = \text{NN}(x(t), h(t), \text{sigmoid}(T(t)), \theta_{\sigma(t)}^{y_{ns}}) \quad (1-4)$$

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

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

对于 L-1 和 L-2 中的 ODE-net，均利用系统处于当前阶段的持续时间 $T(t)$ 作

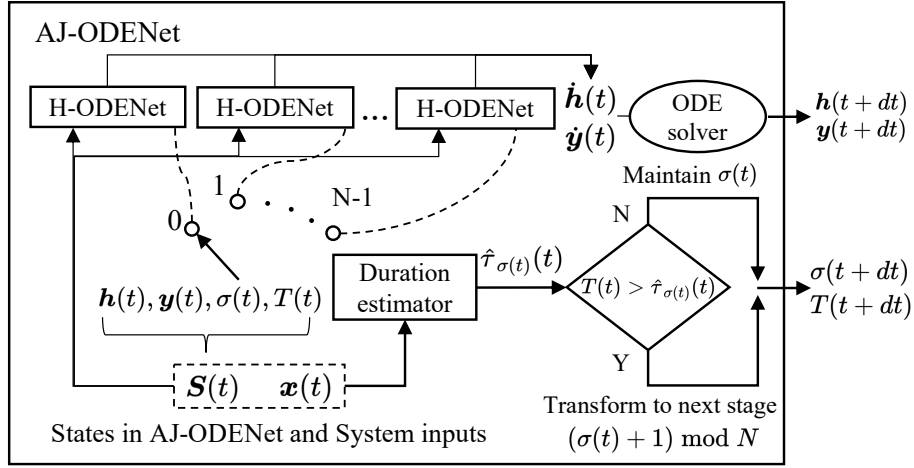


图 1-2 AJ-ODENet 模型结构

为特征输入，辅助估计 $\mathbf{y}(t)$ 的导数。 $\{\theta_i^{y^{ns}}, \theta_i^{y^{ns}}, \theta_i^{\mathcal{H}}\}_{i=1}^N$ 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、带有一定比例的不规则分布的缺失值。例如可能存在一个时刻 i ，两区间 $t_i - t_{i-1} \neq t_{i+1} - t_i$ ，或 $\mathbf{x}(t_i)$ ，或者存在 $\mathbf{x}(t_i)$ 存在，而 $\mathbf{y}(t_i)$ 缺失，并在数据集中表示为 *Null*。本章遵循^[7,8]中的形式，将离散的控制输入序列连续化并作为 ODEnet 的输入。因此需要实现面向离散序列输入的连续化插值方法，将离散和采样不规则的数据格式转换为连续时间格式。为了实现该目标，有多种方法可以实现，如高斯过程^[9]，核方法^[10]或样条插值^[8]。本章参考^[8]采用三次样条插值方法。

1.3 自跳跃常微分方程网络

1.3.1 状态定义

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

图 1-2 简要介绍了 AJ-ODENet 的主要组件及其结构。在图 1-2 中，AJ-ODENet 模型嵌入了 N 个 H-ODENet 模块，根据状态变量 $\sigma(t)$ ，在同一时刻只采用其中一个 H-ODENet 模块用于系统预测。。相比于普通的 ODEnet 仅建模隐状态的导数。AJ-ODENet 将隐状态扩展为四个参数，从而覆盖连续时间跳变系统中输出变量变化以及阶段变量的变化。其中，隐状态的计算和更新是连续时间域下的，表示为 $\mathbf{S}(t) = [\mathbf{h}(t), \mathbf{y}(t), \sigma(t), T(t)]$ ，其中四项分别为系统隐状态、预测的系统输出、系统

所处的当前阶段和处于当前阶段已经持续的时间。AJ-ODEnet 可以看作是 ODEnet 的扩展形式，计算更新过程可以视为给定时间点 t 和无穷小时间步长 τ 计算微分方程的解，如式 (1-6) 所示：

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

1.3.2 基于时间预测器的阶段自转移

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

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

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

对于式 (1.3.2) 所述的状态演化过程，本章基于多层感知器实现了与当前阶段变量 $\sigma(t)$ 绑定的持续时间估计器 $\hat{\tau}_{\sigma(t)}$ 来预测当前阶段的持续时间。预测器的输入为当前隐状态 $\mathbf{h}(t)$ 和外部输入 $\mathbf{x}(t)$ ：

$$\hat{\tau}_{\sigma(t)}(t) = \text{NN}([\mathbf{h}(t), \mathbf{x}(t)], \phi_{\sigma(t)}) \quad (1-7)$$

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。

1.4 基于编码器解码器结构的微分方程网络初值估计与序列预测

在周期跳变系统的开环预测问题中，确定系统当前所处的相位，即识别当前所处阶段以及当前阶段已经持续的时间，是极其重要的。与此同时，受限于测量技术及成本的限制，复杂工业系统的观测空间往往是不完备的。因此，为了实现精确的预测，模型需要对系统的非确定性进行推理。本节将介绍如何根据给定条件范围下的序列数据推断系统的非确定性信息，并作为求解 AJ-ODENet 所需的初始状态。

本章遵循^[11,12]中引入的编码器-解码器框架，使用两个 AJ-ODENets 分别构建用于编码历史系统轨迹的编码器和预测系统输出的解码器，如图1-3。模型输入的时间序列数据包括条件范围 $\{Y_{1:t_I}, X_{1:t_I}\}$ 和预测范围 $\{X_{t_I+1:t_L}\}$ 。

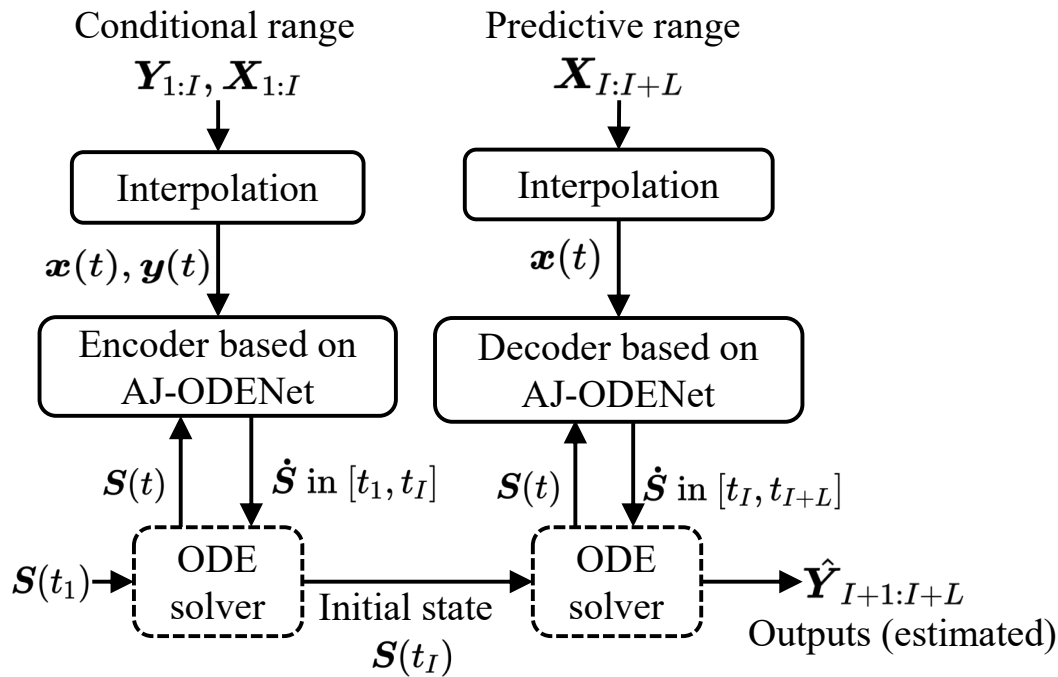


图 1-3 基于 AJ-ODENet 的编码器-解码器预测框架

由于 ODENet 中求解微分方程需要连续的数据，所有范围的数据都需要先经过数据预处理转换为连续时间序列，利用编码器对条件范围内的数据进行编码，构建求解 AJ-ODENet 解码器所需的初始状态 $S(t_I)$ 。

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

具体来说，在条件范围 $[t_1, t_I]$ 的编码阶段，我们将连续时间信号 $y(t)$ 和 $x(t)$ 合并起来作为 AJ-ODENets 编码器的输入，生了解码器所需的初始状态，如式 (1-8) 所示。

$$\tilde{\mathbf{S}}(t_1) = \text{ODESolve}(\text{Encoder AJ-ODE-nets}, \mathbf{S}(t_1), \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{y}(t) \end{bmatrix}, t_1, t_1) \quad (1-8)$$

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 $\tilde{\mathbf{S}}(t_1) = [\tilde{\mathbf{h}}(t_1), \tilde{\mathbf{y}}(t_1), \tilde{s}(t_1), \tilde{T}(t_1)]^T$ 。可以获得状态 $\mathbf{S}(t_1) = [\mathbf{h}(t_1), \mathbf{y}(t_1), \tilde{s}(t_1), \tilde{T}(t_1)]^T$ $\mathbf{h}(t_1)$ 是从均值为 $\tilde{\mathbf{h}}(t_1)$, 协方差矩阵为单位矩阵 \mathbf{I} 的对角多元高斯分布中采样得到的。由于在网络传播过程中存在采样操作。为了构建用于梯度传导的计算图以及减小训练时梯度估计的方差, 上述采样过程使用了重参数化法^[13]。

本节设计 AJ-ODEnets 编码器的最重要的目的是估计隐变量 $\mathbf{h}(t_1)$ 。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^[14,15]。

接下来, 通过给定解码阶段的初始状态 $\mathbf{S}(t_1)$ 和连续时间系统输入 $\mathbf{x}(t)$, 在 $[t_1, t_{1+L}]$ 范围内求解 AJ-ODEnet 解码器对应的常微分方程, 可以得到预测范围内的隐变量状态 $\mathbf{S}(t)$, 其中包括系统在任意时刻 t_i 的系统输出 $\hat{\mathbf{y}}(t_i)$:

$$\hat{\mathbf{S}}(t_{1+L}) = \text{ODESolve}(\text{AJ-ODEnet Decoder}, \mathbf{S}(t_1), \mathbf{x}(t), t_1, t_{1+L}) \quad (1-9)$$

AJ-ODEnets 编码器和 AJ-ODEnets 解码器之间有两个区别。首先, 系统输入 $\mathbf{x}(t)$ 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-ODENet 的输入层的大小是不一致的。其次, 对于条件范围下的编码过程来说, 由于系统的输入输出数据已知, 模型进行阶段转移时, 可直接根据系统先验属性得到阶段之间的转换规则, 不需要阶段转换预测器进行阶段的自转移。对于解码器, 由于系统输出 $\mathbf{y}(t)$ 未知, 因此需要利用阶段转换预测器进行状态自切换。这两个区别表明, 在求解初始状态的过程中, 编码器不引入阶段转换预测器, 而是根据实际的系统输出准确地识别系统当前所处的阶段, 并推导出 $\mathbf{S}(t_1)$, $T(t_1)$ 和 $\mathbf{y}(t_1)$ 。因此, 初始状态 $\mathbf{S}(t)$ 中的 $\sigma(t)$ 和 $T(t)$ 是准确可靠的。对于解码器来说, 相邻阶段之间的转移是依靠持续时间预测器进行判断, 无法保证绝对准确的。在长范围开环预测时会带来较大的累积相位误差。

1.5 模型训练

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

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

- $\Theta_e = \{\theta_i^{y^{ms}}, \theta_i^{y^{ms}}, \theta_i^H\}_{i=0}^{N-1}$: 在 AJ-ODEnet 编码器中的 N 个 H-ODENet

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

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

$$\mathcal{L}(\Theta_e, \Theta_d, \Phi) = \mathcal{L}_p(\Theta_e, \Theta_d) + \lambda \mathcal{L}_{\ll}(\Phi) \quad (1-10)$$

其中 λ 是平衡 \mathcal{L}_p 和 \mathcal{L}_{\ll} 的权重参数。

对于 \mathcal{L}_p ，由于模型将 $\mathbf{h}(t_I)$ 作为隐变量进行后验推断，因此可以采用变分贝叶斯优化的方法，将最大化系统观测输出的证据下界 *Evidence lower bound* (ELBO) (ELBO) 作为模型训练的目标，以近似地最大化系统输出的边际似然^[14]：

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

在本研究中，我们参考前人的工作^[14,16,17]，对隐变量的先验分布和后验分布做出了如下假设。我们假设隐变量 $p(\mathbf{h}(t_I))$ 的先验分布服从正态分布 $\text{Normal}(\mathbf{h}(t_I); \mathbf{0}, \mathbf{I})$ 。模型估计的后验分布 $q_{\Theta_e}(\mathbf{h}(t_I) | \mathbf{Y}_{t_1:t_I}, \mathbf{X}_{t_1:t_I})$ 为对角多元高斯分布。由此，KL 散度项可以简化为对于 $\mathbf{h}(t_I)$ 的正则项。对于解码器部分，我们定义生成模型 $p_{\Theta_d}(\mathbf{Y}_{I+1:I+L} | \mathbf{h}_{t_I}, \mathbf{X}_{t_I:t_{I+L}})$ 为具有固定协方差矩阵的正态分布。分布的均值定义为 AJ-ODEnet 解码器预测出的系统输出。最大化重构似然的期望 $\mathbb{E}_{q_{\Theta_e}(\mathbf{h}_{t_I} | \dots)} [\log p_{\Theta_d}(\mathbf{Y}_{I+1:I+L} | \mathbf{h}_{t_I}, \mathbf{X}_{t_I:t_{I+L}})]$ 可以简化为最小化系统输出预测值 $\hat{\mathbf{Y}}_{I+1:I+L}$ 与实际系统输出 $\mathbf{Y}_{I+1:I+L}$ 之间的 L^2 距离。综上所述，预测系统输出的损失和编码获得隐变量的损失可表示为：

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

阶段持续时间预测器的误差损失 \mathcal{L}_τ 定义为训练时每个阶段下，预测器估计的持续时间与该阶段实际持续时间之间的平均平方误差。构建过程如图1-4所示：

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

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

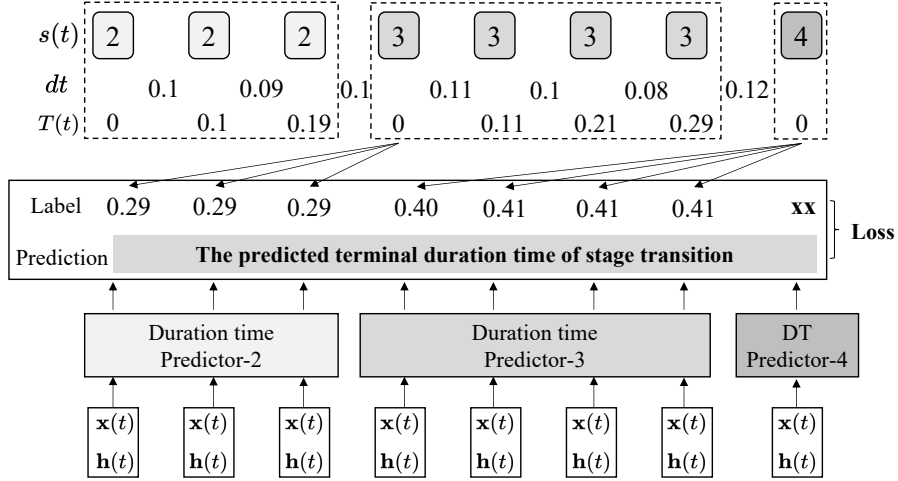


图 1-4 时间预测器的损失函数定义

完整的网络模型通过 Adam 优化器进行训练。完整的训练算法如算法1所示。

算法 1 基于 AJ-ODEnets 的编码器-解码器训练过程

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

1.6 周期性制冷系统预测仿真及系统优化

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

1.6.1 制冷系统简介

实验中探究的制冷系统为施耐德电气公司^[9]设计。制冷系统为某一大型熔炼设备提供压缩制冷。制冷系统的运行数据可通过调用 Seduce 平台^[18]的 API 获取数据。Seduce 平台是用于电源和温度管理的物联网平台，该系统以 1hz 的频率采集传感器监测数据，包括能耗和温度数据。图 1-5 展示了制冷系统提供热交换制冷的过程示意图。该制冷系统包括室内和室外两部分。室内部分的主要模块为液体-空气热交换器，该模块吸收生产设备产生的热空气，然后通过管道、冷凝器和风机将热量转移和疏散到室外，通过压缩生成的冷空气输送至室内。

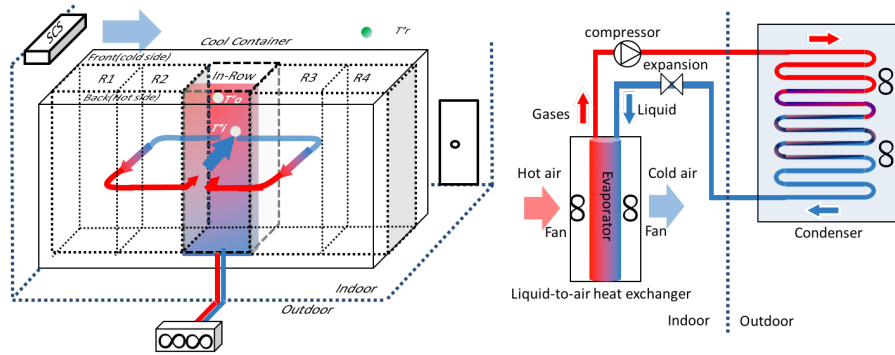


图 1-5 工业制冷系统整体结构及热交换过程

制冷系统作为常见的工业设施，是一种典型的周期性多阶段系统，其运行过程为部分可观测的 2 输入 3 输出（Multi-Input Multi-Output, MIMO）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空气温度限定某预先设定的区间上下限内。当进气口空气温度达到上限时，制冷系统会开启压缩机，制冷系统产生的制冷量 $y_2(t)$ 增加。当进气口温度低于设定值时，压缩机关闭，制冷系统变为待机状态。系统在压缩制冷时的运行功率 $y_1(t)$ 远大于关闭压缩机时的功耗。伴随着制冷与待机，制冷系统的运行过程具有典型的周期多阶段特性，其阶段转换如图 1-7 所示。

当制冷系统待机时，进气口温度不断上升，其制冷量和制冷功率接近于 0。当制冷系统工作时，进气口温度不断下降，制冷量和制冷功率经过两个阶段的快速震荡后会趋近于某固定值。因此，对于该制冷系统，实时功率消耗 $y_1(t)$ 和实时制冷量 $y_2(t)$ 为系统的稳定输出 $y_s(t) = [y_1(t), y_2(t)]$ 。进气口温度 $y_3(t)$ 为系统的非稳定输出： $y_{ns}(t) = [y_3(t)]$ 。将进气口温度定义为非稳定过程，这一性质与系统的先验特性是一致的。在第二个实验中，我们通过改变温度阈值而优化制冷系统能耗，需要模型对于进气口温度实现外推预测，在 H-ODENet 模型中将进气口温度定义为非稳定过程，对于实现可靠、合理的外推预测具有重要意义。

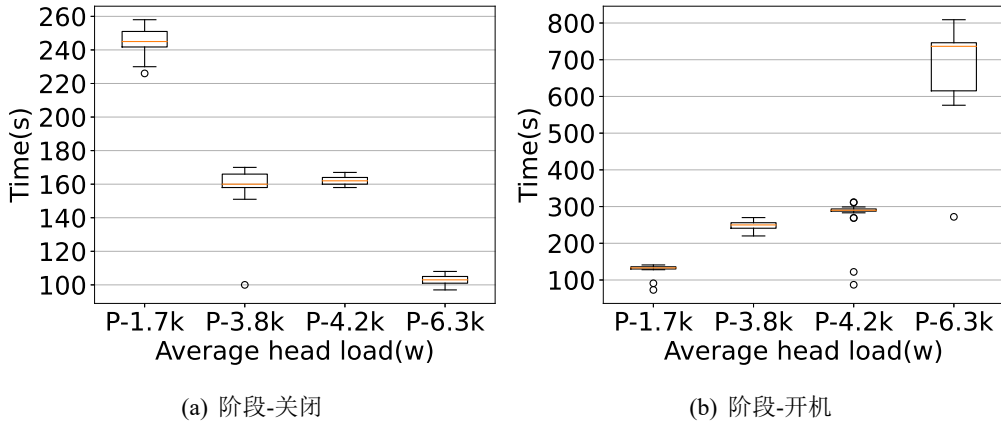


图 1-6 在平均负载不同的四个数据集中，阶段启动和关闭阶段的持续时间分布箱线图

1.6.2 数据预处理：离散序列插及阶段标注

为了使用微分方程网络对序列数据进行处理，需要使用插值方法将带有缺失值的不规则采样数据集转换为连续信号。在本章实验环节中，制冷系统数据集存在采样间隔不均匀、部分数据缺失的问题。本章参照前人工作^[8]，使用三次样条插值方法对离散时间序列 $\mathbf{X}_{[1:I+L]}$ 和 $\mathbf{Y}_{[1:I]}$ 进行插值，构成连续时间信号 $\mathbf{Y}: [t_1, t_I]$ 和 $\mathbf{X}: [t_1, t_{I+L}]$ 。在任意时刻 t ， $\mathbf{X}(t)$ 和 $\mathbf{Y}(t)$ 二阶可微，满足自适应步长求解器求解神经 ODE 系统时所需的必要条件^[8]。当待处理的数据集包含的噪声比较少时，相比于其他插值算法，三次样条插值实现简单，且对原始序列数据的破坏较小，插值生成的连续信号中的噪音较小。

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

根据现场观察和对于制冷系统的了解，可知的系统先验知识如下：

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

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

表 1-1 跳变系统在四种状态下循环，每种状态对应制冷系统的一个阶段

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

1.6.3 模型训练

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

对于训练集和验证集，本节使用大小为 1600s 的滑动窗口对原始序列进行遍历并生成训练样本。对于每一个训练样本，根据 1.1 定义，前 800s 为条件范围，用于模型求解预测阶段所需的初始状态，后 800s 为预测范围，模型预测该范围内的系统输出。在训练过程中，我们选择在验证集中表现最好的模型用于模型评估。由于输入的维数较低，且得益于伴随状态法 (Adjoint state)，训练 ODE-net 网络对于显存的消耗是极低的。我们选择了较大批大小 (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 $h(t)$ 的大小是 20，学习率设置为 0.005。

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

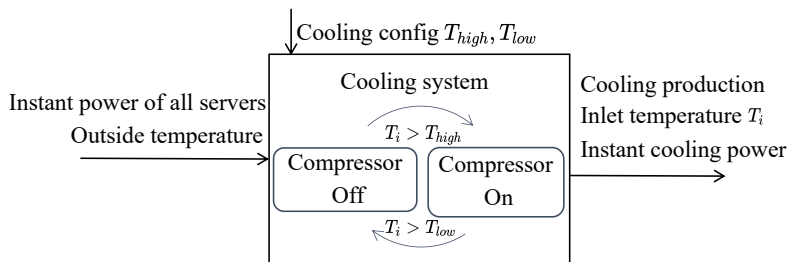


图 1-7 制冷系统运行原理概述

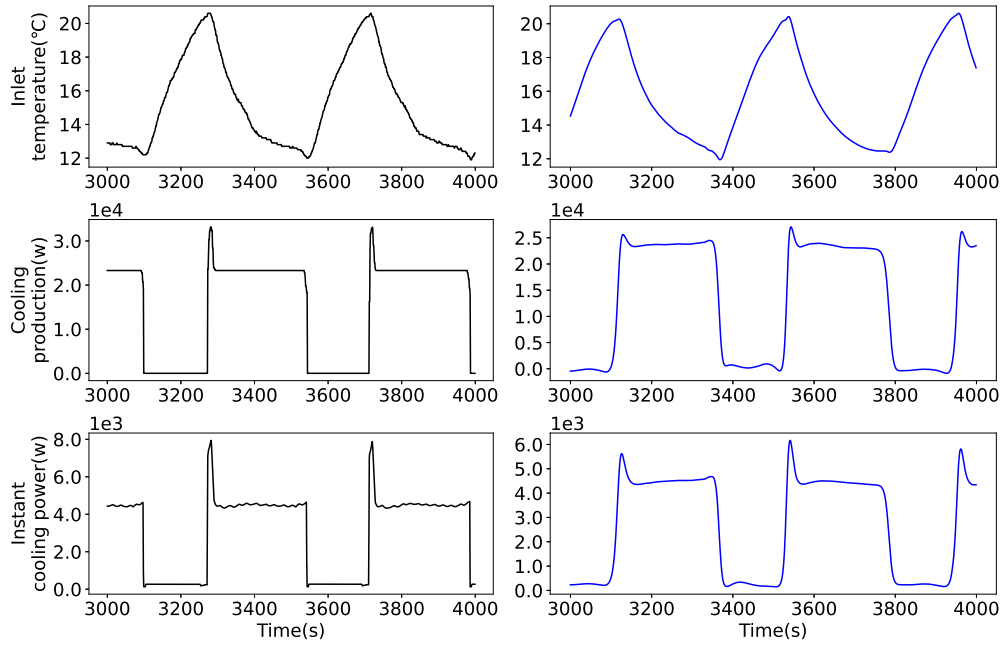
表 1-2 系统输入输出, 不同负载下的数据集和模型的操作

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

1.6.4 应用研究 1: 运行变量仿真

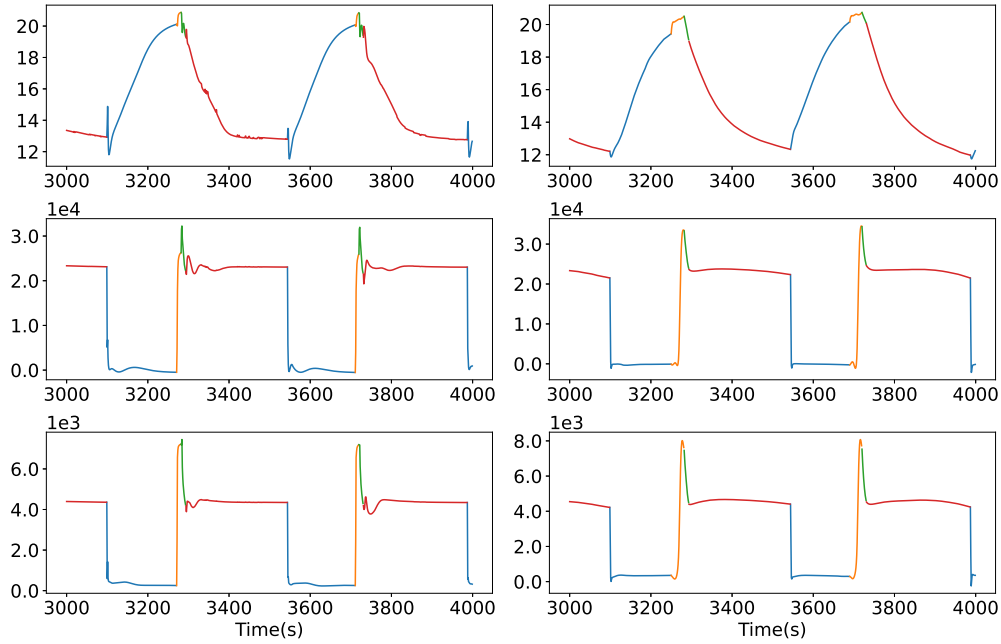
本节使用基于 AJ-ODENet 结构的编码器-解码器框架仿真制冷系统的输出。模型的预测变量为三个: 进气口温度, 制冷量和制冷机功率。其中, 进气口温度是受制冷系统影响的控制目标变量, 表示制冷系统排出的空气温度。制冷量代表制冷系统单位时间产生的制冷量, 其直接影响制冷系统中进气口温度的变化^[19], 在压缩机不工作时, 制冷量为 0。制冷系统开始工作后, 制冷量会突然飙升至较高水平。制冷机功率, 表示整个制冷系统瞬时功耗。在制冷系统工作期间功耗较高, 待机期间功耗较低。本节同时引入普通的 ODE-Net、ODE-RNN^[14]、CDE-Net^[8] 作为本章 H-ODENet 的对比模型。对比结果如图 1-8 所示。其中图 1-8(a) 为真实的系统输出, 图 1-8(d) 为采用带有 H-ODENet 的 AJ-ODENet 模型的预测结果。在图 1-8(b) 中, 将 AJ-ODEs 模块替换为一个单一的稳定型 ODENet, 此时各阶段的转换不受持续时间预测器控制。此时模型难以对各个阶段间的系统切换边界处给出准确的拟合, 尤其是对于持续时间较短的阶段转换位置, 如阶段 1 以及阶段 2。在图 1-8(c) 中, 采用 ODE-RNN^[14] 替换 AJ-ODENets 中的混合了稳定性和非稳定性输出的 H-ODENet 模块。可以发现该结构形成的预测结果难以拟合平滑的输出过程。相比于其他模型, 图 1-8(d) 中 AJ-ODENet 预测的系统输出十分精确, 且在阶段转换的边界处能够极好地识别系统输出的剧烈变化。上述实验表明, 使用多阶段模型能够将先验知识集成到模型中, 相比于单模型结构, 能够更好地处理多阶段系统的阶段转换边界预测。同时结合了非稳定输出和稳定输出的 H-ODENet 模型能够有效地对系统的多个输出项进行学习, 相比于忽视了系统输出时序特性的 ODE-RNN 模型, 能够更准确地预测系统的输出。

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



(a) Original data

(b) One Neural ODE



(c) AJ with ODE-RNN cells

(d) AJ with H-ODE cells

图 1-8 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 1-3 The comparison of energy consumption prediction and the relative computational efficiency

	Average heat load of dataset				Relative Time
	1.7k	3.8k	4.2k	6.3k	
AJ-ODENet(ours)	4.20	1.51	3.46	2.53	3.2
One Neural ODE	6.18	19.99	6.24	3.39	1.0
Jump with CDE ^[8] cells	8.25	2.73	6.62	3.87	3.8
Jump with ODE-RNN cells	38.26	9.94	10.58	31.90	3.4

来，依时间尺度对瞬时功耗进行积分可以得到某一时间长度（T）下的累积功耗。进一步地，可以定义预测功耗的平均绝对百分比误差，MAPE(T)：

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

其中，其中绝对百分比误差 (APE) 定义如下：

$$\text{APE}(t_1, t_2) = \left| \frac{\int_{t_1}^{t_2} \hat{y}_1(t) dt - \int_{t_1}^{t_2} y_1(t) dt}{\int_{t_1}^{t_2} y_1(t) dt} \right| \quad (1-15)$$

其中，绝对百分比误差 $\text{APE}(t_1, t_2)$ 衡量了时间范围 $[t_1, t_2]$ 内估计功耗和真实功耗之间的相对误差。图 1-9描述了评估窗口的大小 T 的选取如何影响评估结果 $\text{MAPE}(T)$ 。可以看出，随着 T 的增加， $\text{MAPE}(T)$ 在开始时剧烈下降，然后缓慢下降。当评价窗口大小超过 35 分钟时，所有数据集的 MAPE 都趋于稳定，说明此时真实系统输出序列与预测序列之间的阶段起止位置相位差对于评估长期功耗的预测精度不再产生影响。因为本节对比了 $T = 30$ 分钟下，不同模型对于累积功耗预测结果的 MAPE，结果如表 1-3所示。结果表明对于所有数据集，AJ-ODENet 预测结果的 MAPE 稳定低于 5%，优于其他三个对比模型。充分证明了本文提出的 AJ-ODENet 框架能够以开环预测的方式在长时间尺度下精确地仿真制冷系统的累积功耗。

从计算复杂性的角度，本节对四个模型在测试集单批数据上执行(1-8)以及(1-9)两个过程的时间消耗进行了对比。Neural ODE 作为耗时最短的模型，将其时间消耗作为基准。其他模型的推理时间均处以该基准以得到相对尺度下的时间对比结果。Neural ODE 的相对时间消耗为 1.0。对比结果如表1-3的最后一列所示。相对时间尺度下的对比结果表明引入阶段转换预测器会增加计算的开销，该时间消耗主要用于更新阶段变量及阶段持续时间。

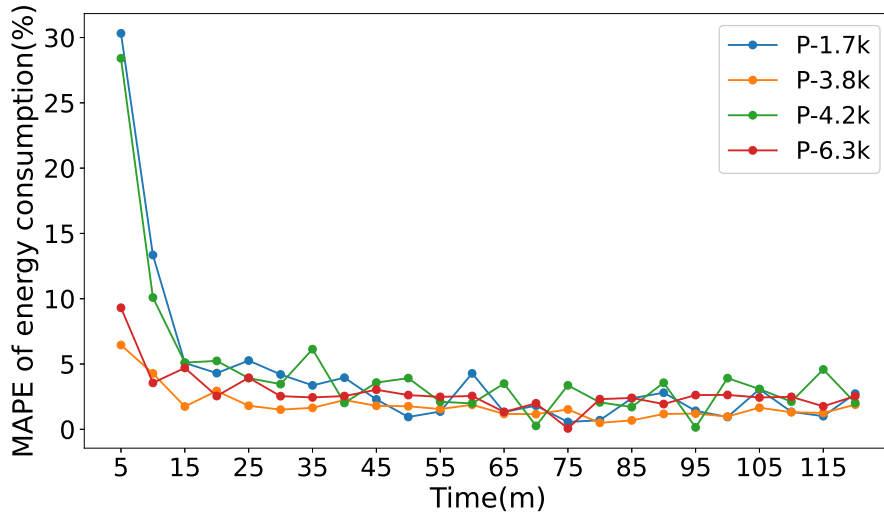


图 1-9 预测不同时间长度的功耗的 MAPE 的变化

1.6.5 应用研究 2: 制冷系统进气口温度设定点优化

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

出于提高能源效率和确保生产过程安全的目的，本节保持温度上限值不变，只调整下限值以减少能源消耗。然而，要确定最佳温度阈值是很棘手的。如果阈值设置得太低，由于过度冷却会浪费大量电能。如果阈值设置得太高，压缩机的制冷重启过程将更加频繁，这将在单位时间内形成更多的峰值功率。从图 ?? 中可以看出，制冷系统重新启动时的功耗显著高于提供稳定冷却时的功耗。因此，盲目地增加温度下阈值反而可能会导致的整体的能耗增加。

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

$$Ti_{min}^* = \arg \min_{Ti_{min}} \int_0^T \hat{y}_1(t) dt \quad (1-16)$$

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

其中 $\hat{\mathbf{Y}}_{0:T}$ 和 $\hat{y}_1(t)$ 分别表示模型预测的系统输出和模拟的瞬时功耗。在给定系统输入和恒定的上限值的条件下，其优化变量是温度下限的设定点，优化目标是最小化累计能耗。式(1-16)中的 $\mathbf{X}_{0:T}$ 为测试数据集中的系统输入，其中包括所有时刻的热负载以及环境温度。式(1-16)中的 $F(\cdot)$ 表示在给定输入和温度阈值 Ti_{max} 和 Ti_{min}

下，预测制冷系统的输出。

本节中，我们利用上一节 1.6.4 中训练的编码器-解码器 AJ-ODEnets 模型，在给定不同下边界温度设定点的情况下，仿真系统能耗以及温度变化，同时评估特定时间区间内的累积制冷功耗。实验采用第 1.6.4 节中所述的三个数据集，其的平均负载分别为 1.7k、3.8k 和 6.3k。温度下阈值设置点从 12°C 逐渐增加到 18°C，调整间隔为 0.5°C。为了模拟不同 $T_{i_{min}}$ 下的冷却系统运行过程， $F(\cdot)$ 中的预测过程与原始训练模型 $f(\cdot)$ 略有不同。特别地，在阶段 *On* 下的持续时间预测器被特定的转换规则所替代。如表 1-1 中的规则，如果进气口温度经过冷却下降到 $T_{i_{min}}$ ，预测模型立即将阶段变量从 3 (*On*) 过渡到 0 (*Off*)。

尽管本章所提出模型中的阶段转换预测器是基于持续时间预测器设计的，为了满足不同制冷系统运行参数的仿真需要，也可以手动设置状态转换规则。虽然在上述模拟过程中的温度下限与训练数据集对应的阈值参数是不同的，但 AJ-ODENet 仍然支持手动调整阶段过渡阈值以支持此类外推预测。相比于没有引入系统先验的稳态模型^[2]，基于先验知识的模型更具可解释性和可扩展性。同时，本章将进气口温度的变化建模成非平稳过程，使得在阶段 *Off* 期间，进气温度会稳定地上升，在阶段 *On* 期间，温度会稳定地下降，进而确保温度必然能够触达给定的阈值，这一模型特性与系统的先验知识一致，是保证进气口温度能够持续变化直至达到阈值的必要条件，有效避免了模型无限期停留某一阶段内难以跳出的情况。

图 1-10 显示了对于热负载约为 3.8k 的数据集，不同的进气口温度下限对制冷系统的影响。仿真时间为 1000 秒。随着温度下限设置点从 12°C 增加到 18°C，在相同的持续时间内，稳定制冷阶段的时间不断缩减。与此同时，1000 秒内包含了更多的制冷系统启停周期。这导致系统的主要功耗来自于由 *Off* 阶段转变为 *on* 阶段的系统启动功耗，而非制冷功耗。

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

当温度下阈值从 12°C 增加到 18°C 时，制冷系统总能耗先下降然后逐渐上升。由于设置较高的 $T_{i_{min}}$ 会缩小上下限之间的差距，使得压缩机更频繁地重启。虽然重启时间较短，但此时压缩机功耗是极高的。因此，从某一温度开始，由于压缩机重新启动导致的能量消耗占据了制冷总能耗的主要部分，在此之后总能量消耗就会上升。而这一拐点就是寻找的最佳 $T_{i_{min}}$ 。对于三个不同热负荷的数据集，图 1.11(a) 中标出了使总能耗最小的最佳的 $T_{i_{min}}$ 。在图 1.11(b) 中，最佳温度阈值与图 1.11(a) 中的阈值一致。在图 1.11(c) 中， COP 随着 $T_{i_{min}}$ 的增加不断降低。其原因因为频繁的压缩机开闭过程会减少了稳定制冷的时间，进而导致单位功耗下产生的

¹PUE = 总能耗 / 设备工作能耗^[2]

²COP = 制冷量 / 制冷能耗^[2]

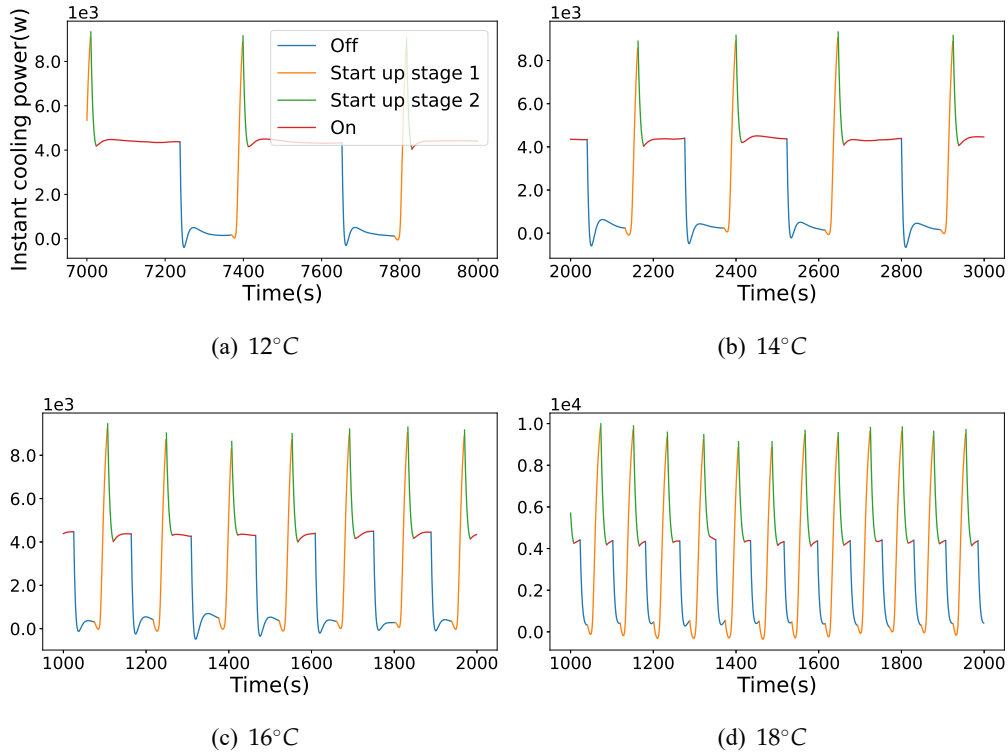


图 1-10 在不同温度下限设定值下的进气口温度的仿真值

表 1-4 optimized power

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

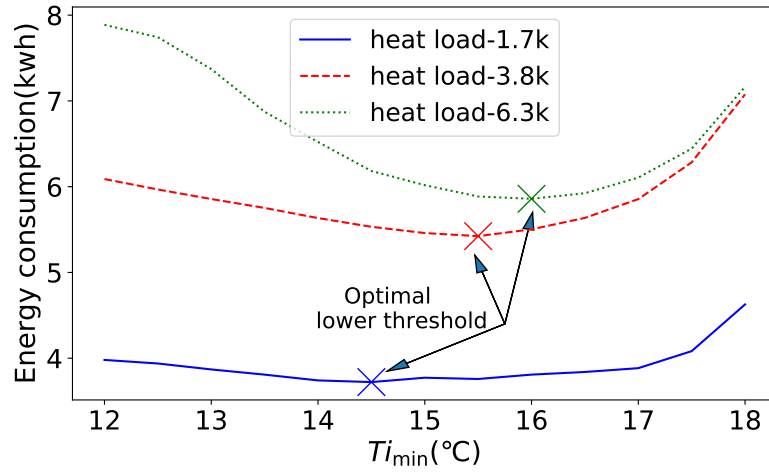
制冷量下降。COP 的下降趋势也解释了为什么提高温度下限值，所需的冷却量减少而制冷能耗却增加。

与此同时我们发现，最优温度设定点随负载的变化而变化。热负载越大，最优温度设定值越大。基于仿真实验推导出的最优温度设置点 Ti_{min}^* ，表 1-4展示了优化温度设定点可以节省的功耗百分比。根据仿真结果，采用新的温度下限设定值可以节省约 6-25% 的能耗，特别对于高热负载情况下的能耗优化是极其显著的。因为实验1.6.4中评估了模型预测累积能耗的相对百分比误差小于 5%，可以认为本节对于不同温度下限设定值的仿真结果是有足够可信度。在未来的工作中，将进一步验证该温度设定策略在实际工业制冷系统中的能耗优化表现。

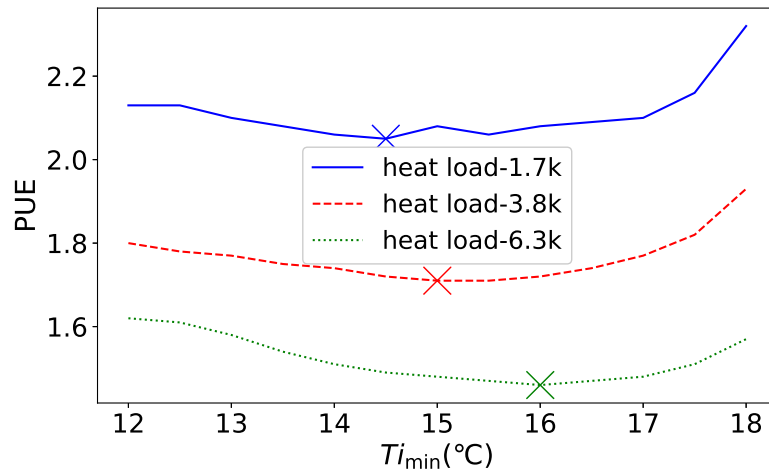
1.7 本章小结

针对连续时间周期跳变系统的建模预测问题，本文提出了一种基于 H-ODENet 的 AJ-ODENets 模型，同时基于该模型构建了用于系统开环预测的编码器-解码器框架。该模型能够对给定的历史序列数据进行编码，在给定未来系统输入序列下仿真预测系统的未来输出。为了使 AJ-ODENet 更好地学习不同阶段的动态特性，AJ-ODENet 包含了多个层次常微分方程网络 (Hierarchical Neural ODE network), 独

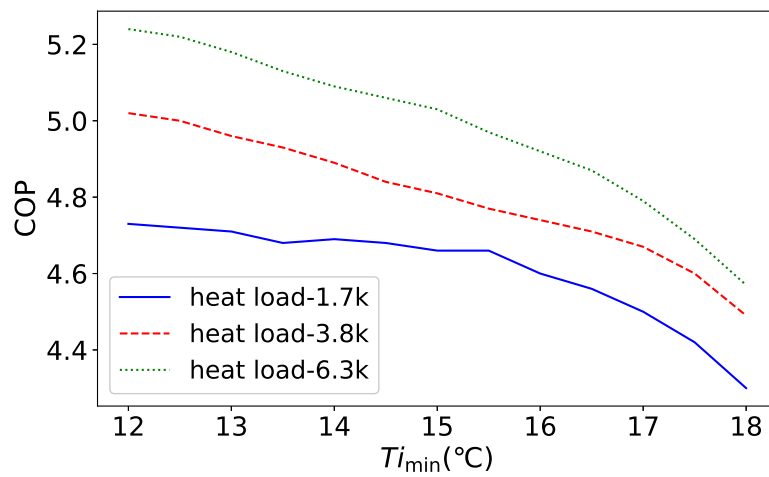
立地学习每个阶段下的系统动态特性，并分别建模系统的稳定输出和非稳定输出，同时构建阶段转换预测器以指定各个预测时刻所用的 H-ODENet。作为 ODENet 的扩展，H-ODENet 中的双层结构分别用于估计隐状态的导数和系统输出的导数，并且对于稳定输出和非稳定输出采用不同的导数模块进行建模。最后，本章利用所提出的基于 AJ-ODENet 的编码器-解码器框架建模某一工业制冷系统。与其他方法相比，AJ-ODENet 在对于制冷系统温度、功耗的开环预测问题中表现出了良好性能，并且能够准确地预估阶段转换点。此外，依托于训练得到的仿真模型，以最小化不同热负荷下的制冷能耗为目标，我们成功优化了制冷系统的温度下限设定点。仿真结果表明，通过采纳优化后温度阈值设置可以节省高达 25% 的能耗。



(a) 制冷能耗



(b) PUE



(c) COP

图 1-11 在不同热负载下改变温度设定下限值对功耗、COP、PUE 的影响

参考文献

- [1] NASON G P. Stationary and non-stationary time series[J]. Statistics in volcanology, 2006, 60.
- [2] YANG H, YIN S. Actuator and Sensor Fault Estimation for Time-Delay Markov Jump Systems With Application to Wheeled Mobile Manipulators[J/OL]. IEEE Transactions on Industrial Informatics, 2020, 16(5): 3222–3232.
<http://dx.doi.org/10.1109/TII.2019.2915668>.
- [3] WANG Y, HE L, JIANG S, et al. Failure Prediction of Hard Disk Drives Based on Adaptive Rao – Blackwellized Particle Filter Error Tracking Method[J/OL]. IEEE Transactions on Industrial Informatics, 2021, 17(2): 913–921.
<http://dx.doi.org/10.1109/TII.2020.3016121>.
- [4] JANSCH-PORTO J P, HU B, DULLERUD G. Policy Learning of MDPs with Mixed Continuous/Discrete Variables: A Case Study on Model-Free Control of Markovian Jump Systems[C] // BAYEN A M, JADBABAIE A, PAPPAS G, et al. Proceedings of Machine Learning Research, Vol 120: Proceedings of the 2nd Conference on Learning for Dynamics and Control. [S.l.]: PMLR, 2020: 947–957.
- [5] WU S, XIAO X, DING Q, et al. Adversarial Sparse Transformer for Time Series Forecasting[J]. Advances in Neural Information Processing Systems, 2020, 33(NeurIPS): 17105–17115.
- [6] DEMEESTER T. System Identification with Time-Aware Neural Sequence Models[J], 2019.
- [7] ZHONG Y D, DEY B, CHAKRABORTY A. Symplectic ODE-Net: Learning Hamiltonian Dynamics with Control[C] // International Conference on Learning Representations. 2019.
- [8] KIDGER P, MORRILL J, FOSTER J, et al. Neural controlled differential equations for irregular time series[J]. arXiv preprint arXiv:2005.08926, 2020.

- [9] LI S C-X, MARLIN B. A scalable end-to-end Gaussian process adapter for irregularly sampled time series classification[C] // Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016 : 1812 – 1820.
- [10] SHUKLA S N, MARLIN B. Interpolation-Prediction Networks for Irregularly Sampled Time Series[C] // International Conference on Learning Representations. 2018.
- [11] DU S, LI T, YANG Y, et al. Multivariate time series forecasting via attention-based encoder–decoder framework[J]. *Neurocomputing*, 2020, 388 : 269 – 279.
- [12] 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.
- [13] KINGMA D P, WELING M. Auto-Encoding Variational Bayes[C/OL] // BENGIO Y, LECUN Y. 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings. 2014. <http://arxiv.org/abs/1312.6114>.
- [14] 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.
- [15] HAFNER D, LILLICRAP T, FISCHER I, et al. Learning latent dynamics for planning from pixels[J]. 36th International Conference on Machine Learning, ICML 2019, 2019, 2019-June : 4528 – 4547.
- [16] 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.
- [17] YILDIZ Ç, HEINONEN M, LÄHDESMÄKI H. ODE2VAE: Deep generative second order ODEs with Bayesian neural networks[J]. *Advances in Neural Information Processing Systems*, 2019, 32(NeurIPS).
- [18] 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.

- [19] ALONSO S, MORÁN A, PÉREZ D, et al. Estimating cooling production and monitoring efficiency in chillers using a soft sensor[J]. *Neural Computing and Applications*, 2020, 32(23): 17291 – 17308.
- [20] COMMITTEE A T. 2011 Thermal Guidelines for Data Processing Environments – Expanded Data Center Classes and Usage Guidance[R]. [S.l.]: ASHRAE, 2011.