

智能工业大数据系统

数据采集



◎ 智工数据

“我就是想让你知道，你不是唯一一个被爱的人。我也是爱你的，而且我愿意为了你做任何事情。但请相信我，我不会像你这样去爱别人。我只会爱你，直到永远。”

With the exception of the first stage of the study—where the researchers focused on the general population—the remaining stages were conducted among patients with a history of stroke.

...and the following day, I am back at the beach.

Digitized by srujanika@gmail.com

Digitized by srujanika@gmail.com

[View Details](#) [Edit](#) [Delete](#)

10. The following table shows the number of hours worked by 1000 employees in a company.

www.scholarone.com | www.scholarone.com | support@scholarone.com

Figure 1. The two sets of images.

[View Details](#)

[View Details](#) | [Edit](#) | [Delete](#)

Page 1

10.000-15.000 €

10 of 10

1

100

...
...
...

前言

报警系统对保障现代化工业系统的安全生产与高效运行发挥着至关重要的作用，然而由于干扰报警多且判断难、报警阈值设计缺乏与相关变量的关联、报警传播广且判断其根源难度大等原因，使得现有报警系统普遍存在“报警过多、处理困难”的突出问题，经常导致无法通过报警及时发现生产过程偏离正常工作范围，未能及时排除异常状况、消除报警，不仅严重影响生产效益，更带来大量生产安全问题甚至灾难性事故。

工业报警系统近年来得到越来越多的关注，国内外学术论文和相关科研项目的数量日益增多。作者近十年来一直从事工业报警系统的理论与应用研究工作，在报警器优化设计等方面取得了一些创新性成果，研发和实施了火电机组智能报警系统等工业应用软件。工业报警系统的研究课题直接来源于工业实践，然而目前只有屈指可数的工程手册类书籍，停留在陈述定性分析、基础数据统计等层面，缺乏一本对工业报警系统存在的关键科学技术问题进行系统科学研究的学术专著。因此，希望本书能对工业报警系统的理论研究和工业实践起到积极推动作用。

由于作者理论水平有限以及研究工作的局限性，特别是近年来工业报警系统的研究正处在快速发展过程中，书中难免存在一些不足和局限，恳请读者批评指正。本书的主要内容源自作者与多位合作者共同完成的学术论文，衷心感谢合作者们给予的无私支持和大力帮助，主要包括国内外专家学者：加拿大Alberta大学陈通文教授、黄彪教授、S.L. Shah教授、I. Izadi博士、胡文凯博士，清华大学周东华教授、杨帆教授，山东电力研究院赵岩院长、苏建军院长、毕贞福博士、高嵩高工、庞向坤高工、李军高工，山东科技大学白星振教授、张超教授等，以及作者指导的研究生们：陈矿博士、余彦博士、杨子江、王振、徐一洲、熊万启、许健伟、朱迪、欧阳子健，并感谢王振、徐一洲、周璇等研究生为本书编辑整理工作付出的辛勤劳动。本书涉及的研究工作得到了国家自然科学基金委员会、国网山东省电力公司、中国华电国际、中国华能集团等单位的大力支持，提供了科研项目经费和工业实践条件，在此表示诚挚的谢意！

王建东
2020年11月

目 录

第1章 绪论	1
1.1 引言	1
1.2 报警系统的重要作用	1
1.3 工业现场的突出问题及原因分析	4
1.4 国内外研究现状及发展趋势分析	7
1.5 需要解决的关键科学技术问题	9
1.6 本章小结	12
第2章 报警系统状态分析	23
2.1 引言	23
2.2 单变量报警状态分析	23
2.3 多变量报警状态分析	28
2.4 案例分析	30
2.5 本章小结	33
第3章 单变量报警器设计	35
3.1 引言	35
3.2 正常或异常运行状况推断	35
3.2.1 基于均值变化点检测的正常和异常数据点划分	36
3.2.2 基于线性分段表示的正常和异常数据段划分	38
3.3 报警延迟器设计	42
3.3.1 性能指标的定义	42
3.3.2 越限报警器的性能指标	42
3.3.3 报警延迟器的性能指标	44
3.3.4 报警延迟器的优化设计	50
3.3.5 案例分析	55
3.4 报警死区设计	58
3.4.1 报警死区适用性判断方法	58
3.4.2 报警死区的设计方法	66
3.4.3 案例分析	72

3.5 基于报警概率图的报警阈值设计	84
3.5.1 P_{ka} 和 T_{ka} 的计算表达式	85
3.5.2 P_{am} 和 T_{am} 的计算表达式	86
3.5.3 确定状态变量	88
3.5.4 计算报警阈值	89
3.5.5 案例分析	91
3.6 干扰报警的识别与消除	97
3.6.1 抖动报警	97
3.6.2 振荡报警	98
3.6.3 与模拟量相关的干扰报警	99
3.6.4 主要技术步骤	99
3.6.5 案例分析	100
3.7 本章小结	108
第4章 多变量报警器设计	111
4.1 引言	111
4.2 基于变化方向的多变量报警器设计	111
4.2.1 基于变量变化方向的正常和异常数据段识别方法	112
4.2.2 基于定性趋势组合的工业报警系统设计	128
4.2.3 多变量报警器设计步骤	131
4.2.4 案例分析	135
4.3 基于超椭球体的多变量报警器设计	145
4.3.1 问题描述	146
4.3.2 动态报警阈值的设计	146
4.3.3 可操纵变量的调整	150
4.3.4 案例分析	155
4.4 基于凸包的多变量报警器设计	162
4.4.1 问题描述	163
4.4.2 正常工作区的建立	164
4.4.3 动态报警阈值的计算	166
4.4.4 案例分析	169
4.5 基于状态变量的多变量报警器设计	173
4.5.1 导致长时间报警的主要原因	174
4.5.2 基于状态的动态报警器	177
4.5.3 报警生成机制	180
4.5.4 设计和实施步骤	181
4.5.5 案例分析	183

4.6 基于变化速率的多变量报警器设计	194
4.6.1 研究问题描述	194
4.6.2 多变量报警器设计主要思想	198
4.6.3 贝叶斯模型估计器	198
4.6.4 调谐参数设计	200
4.6.5 设计步骤	201
4.6.6 案例分析	203
4.7 本章小结	205
第5章 报警根源分析	210
5.1 引言	210
5.2 基于传递熵的报警根源分析	210
5.2.1 标准传递熵	210
5.2.2 标准直接传递熵	213
5.2.3 蒙特卡罗检验	214
5.2.4 详细步骤	218
5.2.5 案例分析	218
5.3 基于贝叶斯网络的报警根源分析	223
5.3.1 贝叶斯网络参数的更新概率	224
5.3.2 分析报警发生的根本原因	226
5.3.3 案例分析	228
5.4 基于定性趋势的报警根源分析	234
5.4.1 基于定性趋势的主要思想	234
5.4.2 求取定性趋势组合方法步骤	236
5.4.3 多特征主题模型	239
5.4.4 案例分析	244
5.5 本章小结	256
第6章 报警泛滥分析	260
6.1 引言	260
6.2 报警泛滥检测	260
6.2.1 两个检测报警泛滥的基本标准	260
6.2.2 报警泛滥新的检测标准	262
6.2.3 方法步骤	263
6.2.4 案例分析	264

6.3 报警泛滥相似性分析	272
6.3.1 基于BLAST的报警序列局部比对算法.....	275
6.3.2 基于报警事件集合的预匹配机制	275
6.3.3 种子搜索、延伸和回溯步骤	278
6.3.4 方法步骤	284
6.3.5 案例分析	284
6.4 报警泛滥预测	289
6.4.1 报警序列的相似性分数	289
6.4.2 预测报警事件的概率	290
6.4.3 方法步骤	292
6.4.4 案例分析	292
6.5 本章小结	294
附录 A 基于密度峰值的聚类算法	297
附录 B 凸包模型参数的计算.....	298

第1章 绪论

1.1 引言

报警系统是用来“生成和处理报警并将其呈现给工厂操作人员”的系统，可以实时对数以千计的生产过程变量进行监控，是现代工业中分散式控制系统（DCS）、数据采集与监控系统（SCADA）等必备的核心组成部分之一。报警是报警系统在监控到一个过程变量靠近并超过一个设定值时，向操作员发出的通知。最常见的监控方式是将过程变量的当前值与高/低报警阈值进行比较，当过程变量的当前值大于高报警阈值或小于低报警阈值时，相应的报警变量由0变为1（如图1.1所示）。通常报警是通过以下方法来提供给工厂操作人员的：每个报警都独立显示在报警窗口中，并有对应的测点名和编号。当报警激活时，报警警示灯会亮起，并将会有报警状态(是否确认)的指示以及报警优先级的指示。工厂操作人员可以通过指定“报警停闪”、“报警确认”等操作来管理这些报警。此时，报警信息出现在报警列表中，以高亮、闪烁等方式显示在生产运行人员的操作界面上，甚至触发声音警报，随后由生产运行人员对出现的报警进行确认，并采取相应的操作动作来消除报警。

报警系统对保障安全生产与高效运行发挥着至关重要的作用，当生产过程偏离高/低报警阈值所代表的正常区域时，报警可及时提醒运行人员采取相应操作动作使得生产过程回到正常区域。然而，由于报警变量设置、报警器及报警阈值设计缺乏科学性，又由于生产过程经常受到外部干扰的影响，导致出现大量不必要的干扰报警，正确报警反而被淹没，使得运行人员常因“狼来了”效应而忽视或不信任报警信息。由于判断引发报警的异常状况的难度大，运行人员常常无法及时采取适当的操作动作以消除报警，导致异常状况在大范围内传播，在短时间内引发大量报警而出现报警泛滥现象。在这种情形下，即使其中大部分报警是正确的，也会由于报警过多，超过运行人员处理报警的能力，使得报警系统无法发挥作用。上述情况不仅会严重影响生产效益，更会带来大量生产安全问题甚至灾难性事故！

1.2 报警系统的重要作用

首先，报警系统是及时发现与排除虚惊事件、避免安全事故的核心手段。在安全金字塔结构（图1.2）中，安全事件可依据后果的严重程度分成虚惊事件和事故两大类，其中绝大部分是可避免的虚惊事件，并且事故通常也不是突发的，而是逐步发展出现的，因此在事故出现前，往往伴随着多个虚惊事件的发生作为先兆。从图1.1中可以看出，当生产过程刚开始偏离正常区域时，报警可及时提醒生产运行人员了解生产过程的异常状况，以采取相应的操作动作，使得生产过程回到正常区域，从而及时发现与排除虚惊事件，避免触发紧急停车等安全事故。

其次，对大量安全事故的原因调查结果与改进建议也充分说明了报警系统的重要作用。2005年英国邦斯菲尔德（Buncefield）油库爆炸事故是欧洲迄今为止最大的一次工业爆炸事故，受伤43人，紧急转移2000余人，造成直接经济损失2.5亿英镑（约合35亿人民币）。



图 1.1 过程变量、报警变量与报警阈值之间的对应关系

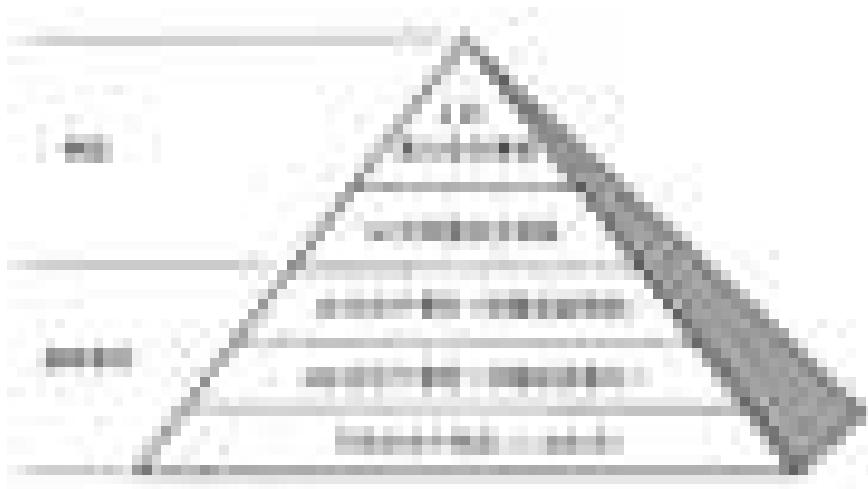


图 1.2 安全金字塔结构示意图^[1]

英国健康与安全执行委员会（Health and Safety Executive）出具的调查报告^[2]针对事故原因提出18项技术与管理建议，其中第8项指出需要配置独立的报警用于避免储罐的满溢。再以2013年11月22日发生的青岛“11·22”中石化东黄输油管道泄漏爆炸特别重大事故为例，输油管道泄漏原油进入市政排水暗渠，在形成密闭空间的暗渠内油气积聚遇火花发生爆炸，造成62人死亡、136人受伤，直接经济损失7.5亿元。国务院事故调查组给出的调查报告^[3]指出6项事故防范措施，其中包括“要开展油气管道长周期运行、泄漏检测报警、泄漏处置和应急技术研究，提高油气管道安全保障能力”；随后，国务院安委会在2013年12月6日下发《关于开展油气输送管线等安全专项排查整治的紧急通知》，其中明确规定：“查油气管线、危险化学品管道是否配备了压力、流量等信息的不间断采集和监测系统，设置了泄漏检测报警装置、紧急切断装置（第九条）；查管道经过的封闭空间是否设立了泄漏报警检测设施等（第十条）”。

最后，报警系统在保障生产过程高效运行方面的作用更不容忽视。由于生产过程偏离正常工作范围通常意味着产品质量下降、生产能耗加大等不利影响，因此报警系统的有效利用能够大幅度减少生产过程处于异常工作范围的可能性和停留时间。并且正常工作区内通常存在着某

个高效工作区，即在该区域内，产品质量与生产能耗都处于理想范围，因此与正常工作区类似，可针对高效工作区设立相应的高/低报警阈值，使得生产过程长期保持在高效工作区内。

报警系统的重要作用已经得到众多生产企业及其组成的行业组织的高度重视，这在多个行业组织制定的工业指南与标准中已得到了充分体现。美国核能管理委员会（NRC）在1994年和2000年两次发布先进报警系统的指南与技术基础NUREG/CR-61056684^[4]，欧洲工程设备和材料用户协会（EEMUA）分别于1999年、2007年和2013年三次发布报警系统指南EEMUA-191 v.1/2/3^[5]，国际化学工业测量和控制标准化协会（NAMUR）分别在2003年和2008年两次发布报警管理工业标准NAMURNA-102^[6]，美国电力研究所（EPRI）在2005年发布报警系统的要求与实施指南EPRI-1010076^[7]，异常状况管理联盟（ASM）在2009年发布高效报警管理实践指南^[8]，国际自动化学会（ISA）在2009年和2015年两次发布流程工业报警系统管理标准ANSI/ISA-18.2^[9]，美国石油学会（API）在2010年和2019年两次发布管线数据采集与监控系统（SCADA）报警管理工业标准API-1167^[10]，国际电工委员会（IEC）在2014年1月发布流程工业报警系统管理标准IEC-68682^[11]等。与此同时，艾默生（Emerson）、霍尼韦尔（Honeywell）、西门子（Siemens）、ABB、横河（Yokogawa）等DCS国际供应商都推出了报警管理系统软件，例如霍尼韦尔公司的Alarm Management Suite、横河公司的Advanced Alarm Administration Suite；而Invensys、Matrikon、TiPS等工业控制领域知名软件公司也推出了相应的第三方软件，例如Matrikon公司的Alarm Manager软件。然而，这些软件更多停留在报警次数统计、单一方式的报警过滤等较为简单的层面上，不能满足用户准确判断生产过程的运行状态、快速处理报警的需求。这就构成了一个亟待解决的矛盾，即工业标准对报警系统提出了很高的要求，但工业界缺乏满足这些工业标准的技术手段。

工业生产过程的监控与报警问题已经得到了我国政府的高度重视。在2006年2月国务院发布的《国家中长期科学和技术发展规划纲要（2006—2020年）》中，第58项优先主题为“重大生产事故预警与救援”：“重点研究开发矿井瓦斯、突水、动力性灾害预警与防控技术，开发燃烧、爆炸、毒物泄漏等重大工业事故防控与救援技术及相关设备。”2011年4月6日，工业和信息化部、科学技术部、财政部、商务部、国有资产监督管理委员会联合印发的《关于加快推进信息化与工业化深度融合的若干意见》（工信部联信2011-160号）中提出了到2015年的8项发展目标和主要任务，其中有两项充分体现了监控报警的重要地位：第2项“推动生产装备智能化和生产过程自动化，加快建立现代生产体系”指出，“在钢铁、石化、有色、建材、纺织、造纸、医药等行业加快普及先进过程控制和制造执行系统，实现生产过程的实时监测、故障诊断、质量控制和调度优化”；第4项“以信息化推动绿色发展，提高资源利用和安全生产水平”指出，“围绕危险作业场所的安全风险评估、多层防护、人机隔离、远程遥控、监测报警、灾害预警、应急响应和处置等方面，深化信息技术的集成应用，建立安全生产新模式”。2016年12月，国务院发布《关于推进安全生产领域改革发展的意见》，其中第27项“建立安全科技支撑体系”指出：“开展事故预防理论研究和关键技术装备研发，加快成果转化和推广应用”以及“运用大数据技术开展安全生产规律性、关联性特征分析，提高安全生产决策科学化水平”。2017年11月，国务院发布的《关于深化“互联网+先进制造业”发展工业互联网的指导意见》中明确指出：“依托工业互联网平台开展数据集成应用，形成基于数据分析与反馈的工艺优化、流程优化、设备维护与事故风险预警能力，实现企业生产与运营管理的智能决策和深度优化”。2018年6月，工业和信息化部、应急管理部、财政部、科技部印发的《关于加快安全产业发展的指导意见》中指出：在生产安全领域“重点发展交通运输、矿山开采、工程

表 1.1 EEMUA 标准与实际工业统计结果对比

	EEMUA标准	石油与天然气	石化	电力	其他
24小时平均报警数	144	1200	1500	2000	900
10分钟内最大报警数	10	220	180	350	180

施工、危险品生产储存、重大基础设施等方面监测预警产品和故障诊断系统”，在保证重点行业领域安全需求的前提下，以数字化、网络化、智能化安全技术与装备科研为重点方向，“支持符合条件的灾害防治、预测预警、监测监控、本质安全工艺和装备、安全服务等关键技术的研发”。2020年2月，国务院印发《关于全面加强危险化学品安全生产工作的意见》，就强化安全生产的基础支撑保障提出以下要求：“加强化工危险工艺本质安全、大型储罐安全保障、化工园区安全环保一体化风险防控等技术及装备研发。推进化工园区安全生产信息化智能化平台建设，实现对园区内企业、重点场所、重大危险源、基础设施实时风险监控预警”。

1.3 工业现场的突出问题及原因分析

报警系统对保障安全生产与高效运行发挥着至关重要的作用，然而在目前国内外已投入使用报警系统中，普遍存在着“报警过多、处理困难”的突出问题。表1.1给出了工程设备和材料用户协会（EEMUA）推荐的被广为接受的工业标准和对国外不同工业领域的调查结果对比^[12]，其中EEMUA 推荐的工业标准是每小时平均报警数不超过6 次（即24小时平均报警数不超过144次），该标准是依据运行人员对每个报警进行正确处理的平均时间至少是10 分钟来制定的。然而表1 中的数据显示，工业现状离这个标准差距甚远。虽然国内尚未见到相关的正式调查结果，但经过对电力、石化等多家生产企业的实地调查，发现国内工业现状与标准的差距相比表1的结果更为严重。以国内某300MW 燃煤发电机组2013 年1 月至2013年12月的报警数据为例，24 小时的平均报警数是7843次，1小时内最大报警次数是2299次，10分钟内最大报警数是483 次。在这些大量的报警中，一部分是由于噪声或生产波动产生的干扰报警，这些报警会自行出现或消除，因此运行人员不需要对干扰报警做出任何的反应动作；另一部分是由于多个过程变量之间存在关联关系，导致生产过程的某一种异常状态在短时间内（例如10分钟）引发了大量的报警，被称为报警泛滥现象（alarm flood），这些报警包含了很多正确报警，生产运行人员必须对正确报警做出相应的操作动作，例如调整阀门开度、启停某台设备等。

“报警过多、处理困难”的突出问题会引发严重后果，导致报警系统部分甚至完全丧失对安全生产与高效运行应有的保障作用。一方面由于存在大量的干扰报警，不仅不利于运行人员及时发现正确报警问题，更会造成运行人员长期忽视报警信息，不信任甚至关闭报警系统；另一方面在报警泛滥出现时，即使大部分报警都是正确的，由于报警数目过多，超过运行人员处理报警的能力，使得运行人员无法从报警系统中获得信息，这就等同于报警系统不存在！此类案例层出不穷：比如，1994年7月24日英国Texaco公司炼油厂的爆炸事故，雷雨天气使工厂受到较强的外部干扰，在生产过程波动5个小时后，催化裂化装置发生了爆炸事故，导致26人受伤和4800万英镑的巨大损失。由英国健康与安全执行委员会（Health and Safety Executive）出具的事故调查结果指出^[13]，在爆炸事故发生前共出现高级别报警1775次，特别是在爆炸事故发生前11分钟内共出现275次报警。报警优先级设计明显不合理、报警次数过多等问题使得运行人

员无法正确判断生产装置运行状态，这是没有及时消除生产过程的波动导致事故发生的主要原因之一；再如，2016年8月11日14时49分湖北省当阳市马店矸石发电有限责任公司热电联产项目在试生产过程中，2号锅炉高压主蒸汽管道上的“一体焊接式长径喷嘴”裂爆，导致发生一起重大高压蒸汽管道裂爆事故，造成22人死亡，4人重伤；又如，2019年7月19日17时43分河南省三门峡市河南省煤气（集团）有限责任公司义马气化厂C套空分装置发生重大爆炸事故，造成15人死亡、16人重伤，爆炸产生冲击波导致175人轻伤，直接经济损失高达8170万元。

目前的报警系统普遍存在着“报警过多、处理困难”的突出问题，从报警设计与消除的角度分析导致这些问题的主要原因有：

1) 报警变量配置缺乏科学性、不必要的报警变量多：在分散式控制系统（DCS）、数据采集与监控系统（SCADA）等现代化监控系统出现之前，每个报警是通过硬连接线的方式实现的，硬件成本很高（大约为1000美元），因此每个生产装置配置的报警数目大约是30~50个，而且每个报警变量的配置都经过慎重的论证。而在当前的DCS、SCADA等现代化监控系统中，添加一个报警变量只需要软件配置而没有其他成本，所以被认为是“免费”的。因此只要认为有必要监控一个过程变量或一个生产状况时，就可以轻易地设置相应的报警变量，而且一个过程变量可以通过不同配置生成多个报警变量，因此报警的数目急剧增加。例如，从原来的150个硬连接线报警增加到DCS中的14000个报警^[14]。再以国内某台300MW燃煤发电机组为例，在DCS中监测的过程变量总数是7328个，其中配置的报警变量总数是1836个。经过调查表明，只有少部分报警变量是在燃煤发电机组的规划设计阶段确定的，有相应的记录文档给出引发报警的异常状况、配置报警的理由等信息；而大部分报警变量是在调试运行过程中添加的，这些报警变量的配置缺乏科学设计，包括对配置报警的必要性、报警优先级别、报警阈值等重要设计因素都未进行科学规划与论证，这是导致现有报警系统中报警数目过多的根本原因。

2) 报警阈值设计缺乏与相关变量的关联：在DCS、SCADA等现代化监控系统中，大部分的报警是将过程变量的当前值与报警阈值进行比较而产生的（如图1.1所示），然而目前的报警阈值设计都是孤立的，仅基于自身的过程变量来设计，没有考虑与相关过程变量的关联关系。在现代工业的生产过程中，几乎每个生产装置都存在多个被监控的过程变量，它们彼此之间存在着密切的关联关系，共同表征了该生产装置的运行状况，因此这些过程变量是相互影响的，共同构成了一个高维度的几何空间，代表了生产装置的正常工作区域。以图1.3为例，由于两个过程变量存在相互关联关系，其正常工作区为图1.3中的区域A，因此两个过程变量的报警阈值是相互影响的，而且是随着两个过程变量的当前值变化而动态变化的；而在目前的报警系统中，这两个过程变量的报警阈值是单独设计、静态的，构成了一个矩形区域（如图3红色点划线矩形所示）。这导致某些正常工作点（例如图1.3中红色星号标记点）会引发误报警，严重影响报警系统的性能，更危险的情况是某些异常工作点（例如图3蓝色圆圈标记点）会被报警系统遗漏，产生漏报警，使得运行人员无法及时注意到生产装置的异常状况。特别值得注意的是，由于DCS、SCADA等监控系统需要对构成正常工作区域的过程变量进行单独监控，因此每个过程变量都必须有各自的报警变量和报警阈值^{[5] [9] [12]}；与之相对，基于多元统计分析的故障检测或统计过程控制方法是对过程变量进行组合或约减，形成T平方统计量等新变量，为新变量设计相应的报警阈值进行监控，这无法满足报警系统对构成正常工作区域的过程变量进行单独监控的要求。

3) 干扰报警多、判断难：由于工业大系统中外界环境、原料成分、公用工程等外部因素的变化，生产过程经常受到较强的外部干扰，导致过程变量存在较大的随机不确定性。这种随

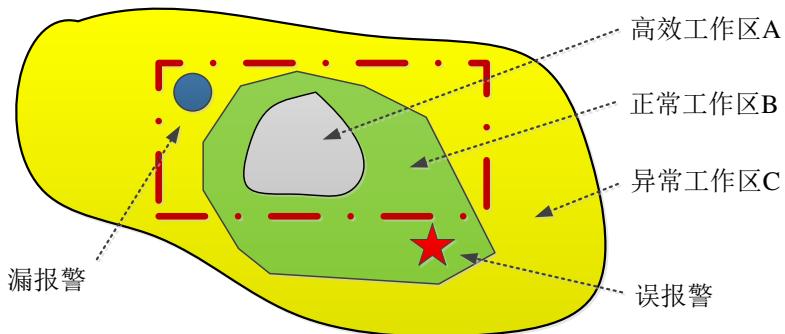


图 1.3 高效工作区A、正常工作区B、异常工作区C 的示意图

机性不确定性使得报警呈现了多个类别，包括在报警状态与正常状态之间快速切换的抖动报警（chattering/fleeting alarm）、以固定周期重复出现的周期报警（cycling/repeating alarm）、长期保持在报警状态的持续报警（standing/stale alarm）、生产运行状态与报警设计不匹配导致的状态报警（state-based alarm）等^{[12][15]}。在这些不同类别的报警中，很多报警属于干扰报警，会自行出现和消除，不含有用信息，不代表生产过程真正偏离了正常运行状况，因此运行人员不需要对干扰报警做出任何的反应动作。但是由于这些干扰报警可导致严重后果，因此非常有必要判断哪些报警是干扰报警，衡量干扰报警的严重程度，必要时对生成报警的报警器重新设计，以减少干扰报警的未来发生次数。然而干扰报警的类别多，而同一类干扰报警可能具有不同的特征，因此难以判断。例如，抖动报警的出现经常是由于过程变量在报警阈值附近受到噪声或外部干扰的影响，然而流量、压力等过程变量的变化速度较快，温度、液位等过程变量的变化速度较慢，因此不能采用相同的报警持续时间（例如5秒）来判断不同的过程变量所出现的报警是抖动报警或是正确报警。由于干扰报警难以判断，目前的报警系统对出现的报警不加以区分，直接将报警呈现在报警列表或操作界面上。

4) 报警传播广、判断其根源难度大：生产装置经常会遇到不同种类的异常状况，因此一个报警可由多种异常状况引发，而且这些异常状况可能单独出现，也可同时出现，这无疑加大了处理报警的难度。更为复杂的情况是，在现代化工业过程中，由于多个生产装置之间存在着物料、信息、能量的传递关系，因此一处异常可引发更大范围的报警，即当某个生产装置出现异常状况时，可能引发连锁反应，类似于多米诺骨牌效应，使得与之相关的其它生产装置出现异常状况，导致在短时间内大量的过程变量出现报警，即引发报警泛滥现象。例如发电机组的汽包水位报警，其原因就涉及主蒸汽流量、汽包压力、给水流量、传感器故障等多个影响因素，而且汽包水位的异常波动会直接影响汽轮机等下游装置，也会由于闭环控制的存在，直接影响汽动给水泵、电动给水泵、除氧器、高低压加热器、凝汽器等多个上游装置，可在短时间内引发多个报警。由于多个异常状况的存在以及上下游装置的互相影响，生产过程的运行人员通过人工方式单凭经验判断报警根源的难度大且不可靠，因此迫切需要科学技术手段辅助判断，以缩短判断时间，提高判断的准确性。

1.4 国内外研究现状及发展趋势分析

目前关于报警系统的研究文献主要包括工业标准与指南^[4-10]、专著^[12,16]、研究报告^[15]、综述性论文^[17-22]以及发表在国内外各期刊上的学术论文，本节内容将针对报警变量配置、报警阈值设计、干扰报警识别与处理、报警根源分析和报警预测分析五个方面，归纳与分析国内外的研究现状。

在报警变量的配置方面，关注的问题是哪些过程变量需要配置报警，且这些需要配置的报警变量如何划分优先等级。相关的主要研究结果主要分成两类：第一类研究结果主要是基于历史数据分析相关或相应的报警变量，以删除冗余报警变量或重新设计报警变量。其中文献^[23-27]采用模糊聚类方法、物元分析方法对过程变量进行分组，对同一组变量的报警优先等级进行排列。文献^[29-39]采用相关分析方法、报警泛滥近似度分析方法寻找具有关联关系的过程变量/报警变量。文献^[24,25]将报警变量分为不同的簇，并通过模糊聚类排序算法对每簇中的报警变量的优先级进行排序。文献^[28]基于ECA规则和实际工况，降低了报警泛滥中无关紧要的报警出现的次数；文献^[29]通过使用报警发生时刻和报警消除时刻之间的时间间隔，找到了相关的报警变量集。文献^[31]基于从二进制报警数据生成的伪连续时间序列的相关系数，对相关报警变量进行聚类。文献^[32]利用二值报警变量的Sorgenfrei 相似系数，以及相关延迟的分布来检测相关报警变量。文献^[40]基于Jaccard相似度为二进制的报警变量构建了报警相似度图。文献^[41]基于符号有向图实现了过程变量之间的因果关系可视化，对相关的报警变量进行分类。文献^[42,43]基于相关性延迟和格兰杰因果关系测度来检测相关报警。文献^[44]基于混合整数线性规划优化了报警系统的设计，延长了操作人员对故障做出反应的时间，同时将触发的报警数量保持在最低水平。第二类研究结果的主要特点是基于工艺知识来建立过程变量/报警变量与异常状况之间的关系，其中文献^[45]建立了异常状况的传播图，从传播路径中选取合适的过程变量为其配置报警，但仅给出了指导性的原则，缺乏必要的具体研究结果作为支撑；文献^[46,47]给出了基于风险的报警策略，据此确定报警优先级，但在风险定量计算方面很不完善，特别是缺少统计计量手段和模型方面的支撑。

在报警阈值设计方面，相关的主要研究结果有：文献^[48]给出改进报警系统的多项指导性建议，其中包括随着工作点变化的动态报警阈值设计，文献^[49]介绍一种基于几何过程控制方法的多变量报警系统，可实现动态的报警阈值设计，文献^[50]指出在生产过程的启停和工作点转换过程中对报警阈值进行动态设计的重要性，但这些文献都没有给出实现动态报警阈值的具体技术方法。文献^[51]依据生产过程工作点变化次序，采用贝叶斯估计方法设计随着工作点变化的动态报警阈值，但动态报警阈值是随着工作点变化的，没有与相关过程变量进行协同关联。文献^[52]根据多个过程变量的联合概率密度，通过最小化FAR 和MAR 来优化报警阈值。文献^[53]依据在过程监控中生成的残差数据，设计了在不同假设条件下的报警阈值。文献^[54]通过蚁群优化算法对FAR、MAR和AAD进行优化，得到多变量报警的阈值。文献^[55]结合FAP，MAP 和相关分析以优化工业过程中的多变量报警阈值。文献^[56]提出了基于相关一致性的多变量报警阈值优化方法。文献^[57]提出了用于多变量过程监控的最优误报警控制的支持向量数据描述。文献^[58]为光伏阵列设计了动态报警阈值调整方法。文献^[59]利用广义极值理论和数据质量评估技术对异常数据进行筛选，并根据筛选出的数据设置报警阈值。文献^[60]从历史数据中估计出报警概率图各个统计量的数值，再通过最优化决策函数的值来优化单变量工业报警系统的报警阈值。文献^[61]通过从历史数据中提取出多个过程变量的相关方向，实现了对多个过程变量的异

常数据检测。文献^[62,63]对工业过程的正常工作区域进行建模，并利用投影方法设计各个过程变量的动态报警阈值，实现了对多个过程变量原始正常工作范围的监控。

关于干扰报警，关注的问题是如何发现干扰报警，并对生成报警的报警器进行更新设计，以减少干扰报警在未来发生的次数。在识别干扰报警方面，现有的研究结果是判断历史数据中是否存在干扰报警，并对干扰报警出现的程度进行衡量，主要的研究结果有：文献^[64]通过观察报警事件与操作员动作之间的事件是否存在平衡关系来寻找干扰报警。文献^[40,65-67]提出抖动报警的量化指标，用于找出抖动报警过于频繁的过程变量。在对单个报警变量的报警器更新设计方面，现有的研究结果是在最基本的越限报警基础上（见图1.1，研究报警延迟器、死区环节、滑动平均滤波器等报警器的设计问题，主要的研究成果有：文献^[68-70]对过程变量及其报警变量进行特征分类，然后根据类别来设计报警阈值，用于抑制抖动报警的出现次数。文献^[71-73]研究误报率、漏报率和平均报警延迟这三个性能指标与报警延迟器等参数的理论关系式，给出基于这三个性能指标的报警延迟器设计方法。文献^[76-82]设计基于证据理论的报警器与滑动平均滤波器等，以获得更低的误报率和漏报率。文献^[83]设计了一种基于主成分分析法的异常工况检测方法，已获得更低的误报率和漏报率。文献^[70,84]基于统计过程控制技术，临时更改报警阈值或将报警变量设为搁置状态来减少抖动报警。文献^[12,16,65,71,85,86]设计了自适应报警死区，以减少抖动报警的数量。文献^[67,87,88]提出了一种通过调整报警阈值或使用报警延时器来消除抖动和重复报警的在线方法。文献^[89,90]利用可视化系统从历史报警泛滥中计算各个报警事件的权重，减少了报警泛滥中干扰报警的影响。文献^[91,92]通过基于关联规则挖掘建立报警推理机制，解析由可扩展标记语言表示的报警拓扑图，进而去除无效报警。文献^[93]提出了一种基于数据驱动的状态报警配置方法，通过修改报警属性或抑制状态来最小化主动报价的数量。文献^[94-96]设计了一种减少化工厂误报警数的分布式并行报警管理策略。文献^[97]提出了3种技术来检测和消除发电厂报警泛滥中的无效报警。文献^[98]使用中值滤波器消除抖动报警。文献^[99]通过数据挖掘对报警数据进行预处理，减少误报警并重建部分丢失报警。已有的研究成果大幅度减少了工业报警系统中抖动报警的数目。

在报警根源分析方面，国内外现有的方法主要包括“基于关联关系模型的根源分析”和“基于时间序列的根源分析”两类。基于关联关系模型的根源分析方法是利用现有的过程知识和专家经验，建立描述监控变量关联关系的数学模型，用于分析事故的根源。应用最为广泛的定性模型为符号有向图（Signed Directed Graph，简称SDG）^[100]。文献^[101]提出了基于SDG的分层建模方法，分析大规模工业系统中的异常状况传播路径。文献^[102]提出基于多级流模型的报警分析方法，找到引发工业异常状况的根本原因。文献^[103,104]利用计算机辅助工程交换模型（CAEX）分析器，解析由可扩展标记语言表示的厂级拓扑图，并联合扰动检测报告，判断引发厂级扰动的根本原因。文献^[105]提出基于空间解释结构模型的报警分析方法，识别引发报警的根本原因以及可视化故障的传播路径。这些定性模型通常不包含任何定量信息，无法识别各个因素的重要性。基于时间序列的根源分析方法是通过分析不同时间序列间的相关性、条件性、时间顺序等特征，挖掘监控变量之间的因果关系。常见因果分析方法有互相关分析、Granger因果分析、传递熵、贝叶斯网络等方法。文献^[106]利用时滞相关性分析，生成因果矩阵和系数矩阵，依据拓扑假设和一致性检验结果生成因果网络。文献^[107]建立了基于互相关函数的多尺度报警根源分析模型，用于分馏塔的冲塔风险分析。文献^[108]采用一种基于异常状况传播与因果关系的报警溯源方法，确定异常状况类型与位置；文献^[109-112]采用传递熵作为统计量，描述异常状况在过程变量之间的因果传递关系。文献^[113,114]利用贝叶斯网络建立概率图

模型，通过计算节点的条件概率以及后验概率来定位报警根源。文献^[115-117]基于定性趋势进行工业报警系统的设计，将先验知识与数据驱动建模相结合，辅助工厂操作人员理解系统运行规律并进行根源分析。文献^[118]将频域分析方法应用于化工过程，检测过程变量间的因果关系。基于连接关系模型和因果分析的混合方法近年来获得了较多关注，这类方法可充分利用已知信息，使得变量之间的因果关系更为可靠。

在报警预测分析方面，国内外现有的方法主要包括“基于模型的预测”和“数据驱动的预测”两类。基于模型的预测方法主要是建立生产事故与监控变量之间的关系模型。对于可以获得系统模型的工业过程，基于模型的预测方法利用可测信号估计系统状态或残差进行监控，包括参数估计方法、观测器方法、对偶关系方法等。文献^[119]基于物质和能量平衡建立动态过程模型，实现短时间窗口内的过程状态预测和预警。然而，复杂工业过程存在参数不确定、干扰变化和非线性等特性，导致监控或预测模型难以辨识。文献^[120]提出了一种新的报警泛滥检测标准，并基于新标准设计了一个报警泛滥检测方法，为报警泛滥的预测提供了数据基础。文献^[121]基于过程变量的历史报警数据，建立置信规则库模型，预测未来一段时间内报警状态的变化趋势；文献^[122]基于历史报警数据计算报警发生概率，实现对关键报警的动态预测；文献^[123, 124]采用基于序列相似性分析的预测方法，预测报警泛滥的出现。基于人工神经网络和深度学习等的预测方法也获得广泛关注。文献^[125]提出了一种基于词嵌入和递归神经网络的报警预测方法，利用报警日志信息进行报警预测。文献^[126]采用序列挖掘和深度学习算法，从历史事件数据中学习报警事件和响应时间的出现规律，从而对未来报警事件进行预测。数据驱动的预测从历史数据中建立预测性分析，不需要建立精确的预测模型，具有更好的通用性，但其预测效果依赖于数据样本的数量和质量，对过程机理的解释性较差。

1.5 需要解决的关键科学技术问题

本节主要围绕报警生命周期的三个阶段，即报警配置、报警设计和报警消除，结合1.3节中提到的导致“报警过多、处理困难”的四个原因，归纳总结目前工业报警系统需要解决的八个基础研究问题。

(1) 报警配置：报警配置主要涉及报警配置过程变量的选择，报警优先级的确定。这一阶段需要解决三个基础研究问题：

问题1：是否应该为一个给定的过程变量配置报警变量？对于此问题，我们主要关注的是确定异常事件与过程变量之间的关系，从而选择合适的过程变量来识别异常事件。如果建立了异常事件与过程变量之间的关系，则选择过程变量的变化反映出异常的存在，并配置报警。然而，建立这种关系的现有方法^[45, 127]都是基于过程知识或从历史操作数据中获得的，这些方法都有其局限性，例如，过程知识往往是不完整的和定性的，而基于数据的方法受到历史数据集中发生的现象的限制，并受到噪音/干扰的污染。因此，解决此问题的主要挑战是如何利用过程知识和历史数据建立关系和如何选择适当的过程变量配置报警。

问题2：报警变量应分配哪个优先级？对于此问题，目前在工业实践中的方法大多是定性的，基于对缺失报警的严重性的理解和对处理报警的安全时间的粗略估计。定量方法是作为定性方法的替代或补充而发展起来的。例如，可以从历史数据集获得更准确的安全时间估计。这里的安全时间是时间允许处理警报，以便关联负面后果不会发生。文献^[47]和^[46]中的基于风险的

方法是有前途的定量方法，用于为报警变量分配优先级。警报优先级可能不是静态的；相反，动态报警优先级对于不同的流程状态和操作场景可能更合理^[128]。

问题3：现有的报警系统中是否存在配置错误的报警变量？冗余(重复)报警变量是那些总是同时进入报警状态或在一个恒定的时间延迟的报警变量。冗余报警变量不包含有用的信息，不应该配置报警。同时，一些高度相关的报警也可能归结于不正确的报警配置。文献^[31, 32, 40, 42]中列出的相关报警分析方法能够有效检测出冗余、相关的报警变量。如果工厂操作人员的行为不能被明确定义为发生了报警事件，那么可能会导致相应的报警变量配置错误。文献^[30, 64]将工厂操作人员的动作与已发生的报警事件相关联；如果没有找到操作人员的操作，则判断发生的报警为滋扰报警。然而，这样做的问题是许多操作人员的动作或响应没有记录在历史数据库中，即使是记录下来的动作或响应，也很难自动地与已发生的报警事件相关联。

(2) 报警设计：报警设计阶段的第一个目标是设计报警产生机制，将配置有报警的过程变量转换为报警变量。这一部分需要解决的基础研究问题为：

问题4：如何设计一个性能良好的报警产生机制？设计报警产生机制的一个主要目的是减少由噪音所造成的干扰警报的数目；另一个主要目的是将相关变量纳入报警变量的生成中，此时报警产生机制会有许多不同的形式。想要系统地解决此问题，主要存在两个挑战：首先，要配置报警的过程变量对应的正常和/或异常工作区很难获得。正常和/或异常工作区是设计报警产生机制的先决条件，一些物理原理和规律可以用来建立正常或异常情况下过程变量的数学模型。然而，为了容忍正常操作的变化和噪声/扰动的影响，模型参数的不确定性必须从历史正常/异常数据集估计。正常/异常工作区也可以直接从正常和异常历史数据集中建立，但这些数据集通常都不是现成的，需要从历史数据样本中提取。一种方法是通过可视化和向工厂操作人员咨询的方式手动提取，这对小型数据集来说是可取的。文献^[72]提出了一种新的方法来检测不同数据集的样本均值，并通过假设检验将样本均值与报警触发点进行比较，从而分离单一过程变量的正常和异常数据集。但涉及多个过程变量时，静态报警的触发点不能真实反映正常操作区域，如图1.3所示；其次，报警产生机制与性能指标之间的关系难以建立。误报率(FAR)和漏报率(MAR)是最常用的性能指标，如何确定报警产生机制与误报率和漏报率之间的定量关系是设计报警产生机制时的一个关键点。对符合独立同分布的一维过程变量建立的单变量报警变量来说，这种定量关系已经建立在报警死区、延迟定时器和滤波器上^[71, 72, 79, 129]。但在更实际的假设下，比如过程变量不是独立同分布、报警变量是多变量等情况，这种定量关系就很难建立。

报警设计阶段的第二个目标是检测正在使用的工业报警系统是否存在干扰报警，以便重新设计报警以减少干扰报警的数量。这一部分需要解决的基础研究问题为：

问题5：是否有太多的干扰报警，因此需要重新设计报警生成机制？解决此问题的一种方法是基于干扰报警的一些特殊特性，如文献^[67, 87]中检测抖动报警和重复报警的基本原理是研究历史报警数据中的报警持续和报警间隔的统计特性。这样做的主要问题是某些干扰报警的特性很难描述，如因为错误的报警产生机制而产生的干扰报警。另一种方法是将工厂操作人员的动作和报警关联起来^[30, 64]。如果操作员的动作与报警事件的发生相关联，则触发的报警是正确的，否则判断其为干扰报警。然而，正如问题3中提到的情况，这种做法在实践中可能不可行。解决问题5的一个更可行的方法是将发生的报警与其后果联系起来。如果没有检测到有害的结果，则将其判定为干扰报警。报警变量与其后果之间的关系是在报警配置阶段建立的关系。由于噪音或干扰的存在，对结果的检测可能依赖于一些假设检验来进行统计分类。在发现干扰报警后，需要评估干扰报警的严重程度，以确定是否有必要重新设计报警系统，或确定哪

些报警变量需要优先处理^[65-67]。

报警设计阶段的第三个目标是对报警进行预测，以指示即将发生的严重异常。这一部分需要解决的基础研究问题为：

问题6：如何设计机制来对报警进行预测，以预测即将发生的关键异常事件？在工业报警系统中，应该尽可能避免报警泛滥的发生，因为“泛滥=事故=损失”^[50]。因此，必须预测即将发生的报警泛滥，并采取预防行动，以避免报警泛滥的发生。一种预测报警泛滥的方法是基于物理或混合模型，它预测过程变量的演化^[130, 131]。然而，开发物理模型在技术上具有挑战性和耗时。因此，这种方法只适用于设备数量有限的情况。由于报警泛滥通常由具有物理连接的报警变量组成，历史报警泛滥可能有一定的规律性可供利用。因此，处理报警泛滥的另一种方法是检测相似的历史报警泛滥，并提取这些相似报警泛滥的规则模板，最终通过将当前发生的报警泛滥与这些模式进行比较来预测即将发生的报警。文献^[33, 34, 132]中使用的方法仅限于从历史数据中检测相似的报警泛滥并提取模板，验证起来非常困难。如果可以通过报警变量的物理连接来补充这些方法，可以得到更令人信服的结论。因此，这一问题的一个主要挑战是如何获得与相似报警泛滥有关的过程知识并将其与历史数据的统计规律性相结合；另一个挑战是如何加快预测报警泛滥时的计算速度。

(3) 报警消除：报警清除阶段首先分析导致报警产生的根本原因，并建议运营商采取一些适当的措施避免后果的恶化，并推动过程变量回到其正常操作区，最终将发生的报警消除。这一阶段需要解决两个基础性研究问题：

问题7：报警产生的根本原因是什么？解决此问题的第一步是判断正在发生的报警是否对应有意义的根源，换句话说，需要判断它们是干扰报警还是正确报警。干扰报警不需要操作人员的行动或响应，移除这些报警时是报警设计阶段的主要目标之一。如果一个报警变量总是产生干扰报警，那么就要删除这个报警变量的报警配置。对干扰报警和正确报警进行在线分类是相当困难的，但使用基于统计推断的分类是可行的。对于正确的报警，问题7的目标是找出发生的异常作为报警的根本原因。如果能够建立异常事件与报警变量之间的关系，那么根本原因就可以轻松找到。在这个意义上，问题1和问题7有一个共同的目标，就是建立过程变量和异常状况之间的关系。因此，过程知识和历史操作数据的局限性也是解决问题7的主要挑战。

问题8：操作人员应该采取什么行动来处理发生的警报？文献^[49]提出了一种基于多元框架中最佳操作区域的几何过程控制方法，根据此方法对变量的纠正操作给出建议，这些建议对于帮助操作人员采取适当行动是有价值的。然而，该方法没有涉及平行坐标技术和射影几何理论的技术细节。这些操作建议是不可或缺的，特别是当涉及到许多过程和报警变量时，工厂操作人员必须根据其后果的严重程度按顺序处理报警，避免过程变量进一步偏离正常作业区域，导致事故的加剧。这与问题2密切相关，在问题2中，主要的挑战是制定定量方法以及完全基于过程知识的方法。在报警泛滥方面，最好的行动也许是根据问题6中的报警泛滥预测结果来采取预防措施，以避免报警泛滥的发生。

对于其中的一些问题，我们已经进行了相关的研究，研究结果将在本书剩余章节中进行介绍。还有若干问题尚未得到解决，我们将在后续的工作中着重研究。

1.6 本章小结

本书重点介绍了报警系统设计、报警根源分析和报警泛滥分析等方面的研究成果。

第一章绪论，介绍了工业报警系统的定义及其重要作用、目前工业报警系统面临的突出问题并分析了这些问题的产生原因、工业报警系统的国内外研究现状以及工业报警系统中需要解决的八个基础研究问题。

第二章从单个报警变量和多个报警变量（报警系统）两个角度介绍了一些常用的基础概念及性能指标，并给出了这些基础概念的定义和计算方法。

第三章介绍了单变量报警器的设计方法，分别通过设计报警延迟器、报警死区和报警阈值来消除工业报警系统中过多的干扰报警。

第四章介绍了多变量报警器的设计方法，考虑到多变量系统中的过程变量往往是相互关联的，采用基于变量变化方向和基于高维空间模型两种思路设计多变量报警器。

第五章介绍了报警根源分析的相关工作，报警根源分析对找到报警的根本原因至关重要，本章分别从基于传递熵、基于贝叶斯网络及基于定性趋势三个方面给出了报警根源分析方法的具体步骤。

第六章介绍了报警泛滥分析的相关工作，主要包括报警泛滥的检测、报警泛滥的相似性分析以及报警泛滥的预测三个方面的研究结果。

参考文献

- [1] PARIYANI A, SEIDER W D, OKTEM U G, et al. Incidents investigation and dynamic analysis of large alarm databases in chemical plants: A fluidized-catalytic-cracking unit case study[J]. Industrial & Engineering Chemistry Research, 2010, 49:8062-8079.
- [2] Buncefield Major Incident Investigation Board. The final report of the Major Incident Investigation Board[R/OL]. Buncefield Major Incident Investigation Board, 2008[2020-12-21]. <https://www.icHEME.org/media/13923/buncefield-miib-final-report-volume-2a.pdf>
- [3] 国务院山东省青岛市“11.22”中石化东黄输油管道泄漏爆炸特别重大事故调查组. 山东省青岛市“11.2”中石化东黄输油管道泄漏爆炸特别重大事故调查报告[R/OL]. 国家安全生产监督管理总局, 2013[2020-12-21]. http://www.mem.gov.cn/gk/sgcc/tbzdsdcbg/2013/201306/t20130626_245228.shtml
- [4] US Nuclear Regulatory Commission. NUREG/CR-61056684. Advanced alarm systems: Revision of guidance and its technical basis[S]. US Nuclear Regulatory Commission, 2000.
- [5] Engineering Equipment and Materials Users' Association. EEMUA-191. Alarm systems-A guide to design, management and procurement[S]. Engineering Equipment and Materials Users' Association, 2013.
- [6] Standardization Association for Measurement and Control in Chemical Industries. NAMURNA-102. Alarm management[S]. Standardization Association for Measurement and Control in Chemical Industries, 2008.
- [7] The Electric Power Research Institute. EPRI-1010076. Advanced control room alarm system: Requirements and implementation guidance[S]. The Electric Power Research Institute, 2005.
- [8] Abnormal Situation Management Consortium. ASM consortium guideline: Effective alarm management practice[S]. Abnormal Situation Management Consortium, 2005.
- [9] International Society of Automation. ANSI/ISA-18.2. Management of alarm systems for the process industries[S]. International Society of Automation, 2015.
- [10] American Petroleum Institute. API-1167. Pipeline SCADA alarm management[S]. American Petroleum Institute, 2019.
- [11] International Electrotechnical Commission. IEC-68682. Management of alarm systems for the process industries[S]. International Electrotechnical Commission, 2014.

- [12] ROTHENBERG D H. Alarm management for process control[M]. New York: Momentum Press, 2009.
- [13] Health and Safety Executive. The explosion and fires at the Texaco Refinery, Milford Haven, 24 July 1994: A report of the investigation by the Health and Safety Executive into the explosion and fires on the Pembroke Cracking Company Plant at the Texaco Refinery, Milford Haven on 24 July 1994[M]. London: Health and Safety Executive, 1997.
- [14] NIMMO I. Rescue your plant from alarm overload[J]. Chemical Processing, 2005, 28-33.
- [15] BRANSBY M L, JENKINSON J. The management of alarm systems[M]. Sheffield: Health and Safety Executive, 1998.
- [16] HOLLIFIELD B, HABIBI E. The alarm management handbook[M]. 2nd ed. Houston: PAS, 2010.
- [17] KIM I S. Computerized systems for online management of failures: A state-of-the-art discussion of alarm systems and diagnostic systems applied in the nuclear industry[J]. Reliability Engineering and System Safety, 1994, 44:279-295.
- [18] 杨帆, 萧德云. 智能报警管理若干研究问题[J]. 计算机与应用化学, 2011, 28:1485-1491.
- [19] WANG J D, YANG F, CHEN T W, et al. An overview of industrial alarm systems: Main causes for alarm overloading, research status, and open problems[J]. IEEE Transactions on Automation Science and Engineering, 2016, 13:1045-1061.
- [20] 朱群雄, 高慧慧, 徐圆. 工业过程报警管理研究进展[J]. 自动化报, 2017, 43:955-968.
- [21] KIRSCHEN D S, WOLLENBERG B F. Intelligent alarm processing in power systems[J]. Proceedings of the IEEE, 1992, 80:663-672.
- [22] GOEL P, DATTA A, MANNAN M S. Industrial alarm systems: Challenges and opportunities[J]. Journal of Loss Prevention in Process Industries, 2017, 50:23-36.
- [23] 顾详柏, 朱群雄, 耿志强. 现代化工流程报警系统分析及管理策略[J]. 化工进展, 2004, 23:1348-1352.
- [24] ZHU Q X, GENG Z Q. A new fuzzy clustering-ranking algorithm and its application in process alarm management[J]. Chinese Journal of Chemical Engineering, 2005, 13:477-483.
- [25] GENG Z Q, ZHU Q X, Gu X B. A fuzzy clustering-ranking algorithm and its application for alarm operating optimization in chemical processing[J]. Process Safety Progress, 2005, 24:66-75.
- [26] 耿志强, 朱群雄, 顾祥柏. 现代化工流程报警系统优化算法及应用[J]. 计算机与应用化学, 2005, 22:183-188.
- [27] 许圆, 朱群雄. 基于物元分析的过程工业报警优化[J]. 化工学报, 2008, 59:1609-1614.

- [28] JANG G, SUH S, KIM S, et al. A proactive alarm reduction method and its human factors validation test for a main control room for SMART[J]. Annals of Nuclear Energy, 2013, 51:125-134.
- [29] KORDIC S, LAM C P, XIAO J, et al. Using event-intervals to analyze alarm sequences in a chemical plant[J]. Journal of Frontiers of Computer Science and Technology , 2009, 3:267 - 281.
- [30] NODA M, HIGUCHI F, TAKAI T, et al. Event correlation analysis for alarm system rationalization[J]. Asia-Pacific Journal of Chemical Engineering, 2011, 6:497-50.
- [31] YANG F, SHAH S L, XIAO D, et al. Improved correlation analysis and visualization of industrial alarm data[J]. ISA Transactions, 2012, 51:499 - 506.
- [32] YANG Z J, WANG J D, CHEN T W. Detection of correlated alarms based on similarity coefficients of binary data[J]. IEEE Transactions on Automation Science and Engineering, 2013, 10:1014 - 1025.
- [33] AHMED K, IZADI I, CHEN T W, et al. Similarity analysis of industrial alarm flood data[J]. IEEE Transactions Automation Science and Engineering, 2013, 10:452-457.
- [34] CHENG Y, IZADI I, CHEN T W. Pattern matching of alarm flood sequences by a modified Smith-Waterman algorithm[J]. Chemical Engineering Research and Design, 2013, 91:1085-1094.
- [35] HU W K, WANG J D, CHEN T W. A local alignment approach to similarity analysis of industrial alarm flood sequences[J]. Control Engineering Practice, 2016, 55:13-25.
- [36] LAI S Q, CHEN T W. A method for pattern mining in multiple alarm flood sequences[J]. Chemical Engineering Research and Design, 2017, 117:831 - 839.
- [37] HAMAGUCHI T, SAKASHITA H, MORITANI H, et al. Method for designing alarm system using DAEs, CE matrices, and preference indices[J]. Journal of Chemical Engineering of Japan, 2017, 50:439-444.
- [38] HU W K, CHEN T W, SHAH S L. Detection of frequent alarm patterns in industrial alarm floods using itemset mining methods[J]. IEEE Transactions on Industrial Electronics, 2018, 65:7290-7300.
- [39] SHANG J, CHEN T W. Early classification of alarm floods via exponentially attenuated component analysis[J]. IEEE Transactions on Industrial Electronics, 2020, 67:8702-8712.
- [40] KONDAVEETI S R, IZADI I, SHAH S L, et al. Graphical tools for routine assessment of industrial alarm systems[J]. Computers and Chemical Engineering, 2012, 46:39 - 47.
- [41] PENG D, GU X, XU Y B, et al. Integrating probabilistic signed digraph and reliability analysis for alarm signal optimization in chemical plant[J]. Journal of Loss Prevention in the Process Industries, 2015, 33:279-288.

- [42] WANG J, LI H G, HUANG J W, et al. A data similarity based analysis to consequential alarms of industrial processes[J]. Journal of Loss Prevention in the Process Industries, 2015, 35:29 – 34.
- [43] Hu W K, Wang J D, Chen T W. A new method to detect and quantify correlated alarms with occurrence delays[J]. Computers and Chemical Engineering, 2015, 80:189-198.
- [44] VENKIDASALAPATHY J A, MANNAN M S, KRAVARIS C. A quantitative approach for optimal alarm identification[J]. Journal of Loss Prevention in the Process Industries, 2018, 55:213-222.
- [45] YAN L, LIU X, NODA M, et al. Systematic design approach for plant alarm systems[J]. Journal of Chemical Engineering of Japan, 2007, 40:765-772.
- [46] CHANG Y, KHANA F, AHMED S. A risk-based approach to design warning system for processing facilities[J]. Process Safety and Environmental Protection, 2011, 89:310-316.
- [47] ARIFIN B M S, CHOUDHURY M. An alternative approach of risk analysis for multivariable alarm system[J]. Journal of Chemical Engineering, 2011, 26:75-79.
- [48] BRISTOL E H. Improved process control alarm operation[J]. ISA Transactions, 2001, 40:191-205.
- [49] BROOKS R, THORPE R, WILSON J. A new method for defining and managing process alarms and for correcting process operation when an alarm occurs[J]. Journal of Hazardous Materials, 2004, 115:169-174.
- [50] BEEBE D, FERRER S, LOGEROT D. The connection of peak alarm rates to plant incidents and what you can do to minimize[J]. Process Safety Progress, 2013, 32:72-77.
- [51] ZHU J F, SHU Y D, ZHAO J S, et al. A dynamic alarm management strategy for chemical process transitions[J]. Journal of Loss Prevention in the Process Industries, 2014, 30:207-218.
- [52] ZANG H, LI H G. Optimization of process alarm thresholds: A multidimensional kernel density estimation approach[J]. Process Safety Progress, 2014, 33:292 – 298.
- [53] BURR T, HAMADA M S, HOWELL J, et al. Estimating alarm thresholds for process monitoring data under different assumptions about the data generating mechanism[J]. Science and Technology of Nuclear Installations, 2013, 36:72-90.
- [54] TIAN W D, ZHANG G X, LIANG H T. Alarm clustering analysis and ACO based multi-variable alarms thresholds optimization in chemical processes[J]. Process Safety and Environmental Protection, 2018, 113:132-140.
- [55] HAN L, GAO H H, XU Y, et al. Combining FAP, MAP and correlation analysis for multivariate alarm thresholds optimization in industrial process[J]. Journal of Loss Prevention in the Process Industries, 2016, 40:471 – 478.

- [56] GAO H H, LIU F F, ZHU Q X. A correlation consistency based multivariate alarm thresholds optimization approach[J]. ISA Transactions, 2016, 65:37 – 43.
- [57] KIM Y, KIM S B. Optimal false alarm controlled support vector data description for multivariate process monitoring[J]. Journal of Process Control, 2018, 65:1 – 14.
- [58] YU C, WANG H Z, YAO J X, et al. A dynamic alarm threshold setting method for photovoltaic array and its application[J]. Renewable Energy, 2020, 158:13 – 22.
- [59] TOSHKOVA D, ASHER M, HUTCHINSON P. Automatic alarm setup using extreme value theory[J]. Mechanical Systems and Signal Processing, 2020, 139:106417-106431.
- [60] YU Y, WANG J D, YANG Z J. Design of alarm tripoints for univariate analog process variables based on alarm probability plots[J]. IEEE Transactions on Industrial Electronics, 2017, 64:6496-6505.
- [61] YU Y, ZHU D, WANG J D, et al. Abnormal data detection for multivariate alarm systems based on correlation directions[J]. Journal of Loss Prevention in the Process Industries, 2017, 45:43-55.
- [62] YU Y, WANG J D, OUYANG Z J. Designing dynamic alarm limits and adjusting manipulated variables for multivariate systems[J]. IEEE Transactions on Industrial Electronics, 2020, 67:2314-2325.
- [63] YU Y, WANG J D. Alarm monitoring for multivariate processes based on a convex-hull normal operating zone[J]. IEEE Transactions on Control Systems Technology, 2019, 28:1-8.
- [64] YUKI Y. Alarm system optimization for increasing operations productivity[J]. ISA Transactions, 2002, 41:383-387.
- [65] NAGHOOSI E, IZADI I, CHEN T W. Estimation of alarm chattering[J]. Journal of Process Control, 2011, 21: 1243-1249.
- [66] KONDAVEETI S R, IZADI I, SHAH S L, et al. Quantification of alarm chatter based on run length distributions[J]. Chemical Engineering Research and Design, 2013, 91:2550-2558.
- [67] WANG J D, CHEN T W. An online method for detection and reduction of chattering alarms due to oscillation[J]. Computers and Chemical Engineering, 2013, 54:140-150.
- [68] YAMANAKA F, NISHIYA T. Application of the intelligent alarm system for the plant operation[J]. Computers and Chemical Engineering, 1997, 21:625-630.
- [69] AHNLUND J, BERGQUIST T, SPAANENBURG L. Rule-based reduction of alarm signals in industrial control[J]. Journal of Intelligent and Fuzzy Systems, 2003, 14:73-84.
- [70] SRINIVASAN R, LIU J, LIM K W, et al. Intelligent alarm management in a petroleum refinery[J]. Hydrocarbon Processing, 2004, 11:47-52.

- [71] ADNAN N A, IZADI I, CHEN T. On expected detection delays for alarm systems with deadbands and delay-timers[J]. Journal of Process Control, 2011, 21:1318-1331.
- [72] XU J W, WANG J D, IZADI I, et al. Performance assessment and design for univariate alarm systems based on FAR, MAR and AAD[J]. IEEE Transactions on Automation Science and Engineering, 2012, 9:296-307.
- [73] ADNAN N A, CHENG Y, IZADI I, et al. Study of generalized delay timers in alarm configuration[J]. Journal of Process Control, 2013, 23:382-395.
- [74] TULSYAN A, ALROWAIE F, GOPALUNI B. Design and assessment of delay timer alarm systems for nonlinear chemical processes[J]. AIChE Journal, 2018, 64:77-90.
- [75] TAHERI-KALANI J, LATIF-SHABGAHI G, ALVARI SHOOREDELI M. On the use of penalty approach for design and analysis of univariate alarm systems[J]. Journal of Process Control, 2018, 69:103-113.
- [76] 宋晓静, 徐晓滨, 文成林. 一种基于证据理论的工业报警器设计方法[J]. 杭州电子科技大学学报, 2011, 31:135-138.
- [77] 刘恒, 刘振娟, 李宏光. 基于数据驱动的化工过程参数报警阈值优化[J]. 化工学报, 2012, 63:2733-2738.
- [78] 赵劲松, 朱剑锋. 基于数据过滤的化工过程重复报警处理策略[J]. 清华大学学报, 2012, 52:77-281.
- [79] CHENG Y, IZADI I, CHEN T W. Optimal alarm signal processing: Filter design and performance analysis[J]. IEEE Transactions on Automation Science and Engineering, 2013, 10:446-451.
- [80] CAI Z S, ZHANG L B, HU J Q, et al. Comprehensive alarm information processing technology with application in petrochemical plant[J]. Journal of Loss Prevention in the Process Industries, 2015, 38:101-113.
- [81] XU X B, LI S B, SONG X J, et al. The optimal design of industrial alarm systems based on evidence theory[J]. Control Engineering Practice, 2016, 46:142-156.
- [82] TAN W, SUN Y K, AZAD I I, et al. Design of univariate alarm systems via rank order filters[J]. Control Engineering Practice, 2017, 59:55-63.
- [83] LI W, PENG M J, WANG Q Z. False alarm reducing in PCA method for sensor fault detection in a nuclear power plant[J]. Annals of Nuclear Energy, 2018, 118:131-139.
- [84] HENNINGSSEN A, KEMMERER J P. Intelligent alarm handling in cement plants[J]. IEEE Industry Applications Magazine, 1995, 1:1 - 15.

- [85] AFZAL M S, CHEN T W, BANDEHKHODA A, et al. Analysis and design of time-deadbands for univariate alarm systems[J]. Control Engineering Practice, 2018, 71:96 – 107.
- [86] WANG Z, BAI X Z, WANG J D, et al. Indexing and designing deadbands for industrial alarm signals[J]. IEEE Transactions on Industrial Electronics, 2019, 66:8093-8103.
- [87] WANG J D, CHEN T. An online method to remove chattering and repeating alarms based on alarm durations and intervals[J]. Computers and Chemical Engineering, 2014, 67(4):43 – 52.
- [88] SU J J, GUO C, ZANG H, et al. A multi-setpoint delay-timer alarming strategy for industrial alarm monitoring[J]. Journal of Loss Prevention in the Process Industries, 2018, 54:1-9.
- [89] CHARBONNIER S, BOUCHAIR N, GAYET P. A weighted dissimilarity index to isolate faults during alarm floods[J]. Control Engineering Practice, 2015, 45:110-122.
- [90] CHARBONNIER S, BOUCHAIR N, GAYET P. Fault template extraction to assist operators during industrial alarm floods[J]. Engineering Applications of Artificial Intelligence, 2016, 50:32-44.
- [91] WANG J, LI H G, HUANG J W, et al. Association rules mining based analysis of consequential alarm sequences in chemical processes[J]. Journal of Loss Prevention in the Process Industries, 2016, 41:178-185.
- [92] DOROGO G, ABONYI J. Sequence mining based alarm suppression[J]. IEEE Access, 2018, 6:15365-15379.
- [93] HU W K, CHEN T W, SHAH S L. Discovering association rules of mode-dependent alarms from alarm and event logs[J]. IEEE Transactions on Control Systems Technology, 2017, 99:1-13.
- [94] LI D W, HU J H, WANG H, et al. A distributed parallel alarm management strategy for alarm reduction in chemical plants[J]. Journal of Process Control, 2015, 34:117 – 125.
- [95] HU W K, WANG J D, CHEN T W. A local alignment approach to similarity analysis of industrial alarm flood sequences[J]. Control Engineering Practice, 2016, 55:13 – 25.
- [96] GUO C, HU W K, LAI S Q, et al. An accelerated alignment method for analyzing time sequences of industrial alarm floods[J]. Journal of Process Control, 2017, 57:102-115.
- [97] WANG J D, YANG Z J, CHEN K, et al. Practices of detecting and removing nuisance alarms for alarm overloading in thermal power plants[J]. Control Engineering Practice, 2017, 67:21-30.
- [98] SUN Y K, TAN W, Chen T W. A method to remove chattering alarms using median filters[J]. ISA Transactions, 2018, 73:201-207.
- [99] MANNANI Z, IZADI I, GHADIRI N. Preprocessing of alarm data for data mining[J]. Industrial and Engineering Chemistry Research, 2019, 58:11261-11274.

- [100] YANG F, DUAN P, SHAH S L, et al. Capturing connectivity and causality in complex industrial processes[J]. Springerbriefs in Applied Sciences and Technology, 2014.
- [101] YANG F, XIAO D Y, SHAH S L. Signed directed graph-based hierarchical modelling and fault propagation analysis for large-scale systems[J]. IET Control Theory and Applications, 2013, 7:537-550.
- [102] DAHLSTRAND F. Consequence analysis theory for alarm analysis[J]. Knowledge-Based Systems, 2002, 15:27-36.
- [103] YIM S, ANANTHAKUMAR H, BENABBAS L, et al. Using process topology in plant-wide control loop performance assessment[J]. Computers and Chemical Engineering, 2006, 31:86-99.
- [104] SCHLEBURG M, CHRISTIANSEN L, THORNHILL N F, et al. A combined analysis of plant connectivity and alarm logs to reduce the number of alerts in an automation system[J]. Journal of Process Control, 2013, 23:839-851.
- [105] GAO H H, XU Y, ZHU Q X. Spatial interpretive structural model identification and AHP-based multimodule fusion for alarm root-cause diagnosis in chemical processes[J]. Industrial and Engineering Chemistry Research, 2016, 55:3641-3658.
- [106] BAUER M, THORNHILL N F. A practical method for identifying the propagation path of plant-wide disturbances[J]. Journal of Process Control, 2008, 18:707-719.
- [107] 胡瑾秋, 蔡爽, 张来斌. 基于多尺度建模的炼油化工过程报警根源分析[J]. 石油学报(石油加工), 2018, 34:341-35.
- [108] YUAN T, QIN S J. Root cause diagnosis of plant-wide oscillations using Granger causality[J]. Journal of Process Control, 2014, 24:450-459.
- [109] BAUER M, COX J, CAVENESS M, et al. Finding the direction of disturbance propagation in a chemical process using transfer entropy[J]. IEEE Transactions on Control Systems Technology, 2007, 15:12-21.
- [110] DUAN P, YANG F, CHEN T W, et al. Direct causality detection via the transfer entropy approach[J]. IEEE Transactions on Control Systems Technology, 2013, 21:2052-2066.
- [111] DUAN P, YANG F, SHAH S L, et al. Transfer zero-entropy and its application for capturing cause and effect relationship between variables[J]. IEEE Transactions on Control Systems Technology, 2015, 23:855-867.
- [112] HU W K, WANG J D, CHEN T D, et al. Cause-effect analysis of industrial alarm variables using transfer entropies[J]. Control Engineering Practice, 2017, 64:205-214.
- [113] HU J Q, YI Y. A two-level intelligent alarm management framework for process safety[J]. Safety Science, 2016, 82:432-444.

- [114] WANG J D, YANG Z J, SU J J, et al. Root-cause analysis of occurring alarms in thermal power plants based on Bayesian networks[J]. International Journal of Electrical Power and Energy Systems, 2018, 103:67-74.
- [115] CHEN K, WANG J D. Normal and abnormal data segmentation based on variational directions and clustering algorithms[J]. Industrial and Engineering Chemistry Research, 2017, 56:7799-7813.
- [116] CHEN K, WANG J D. Design of multivariate alarm systems based on online calculation of variational directions[J]. Chemical Engineering Research and Design, 2017, 122:11-21.
- [117] CHEN K, WANG J D. Finding significant qualitative trend combinations for multivariate systems from historical data[J]. Industrial and Engineering Chemistry Research, 2019, 58:4518-4533.
- [118] 杨帆, 范牛军, 叶昊. PDC方法在化工过程变量间因果关系分析中的应用[J]. 清华大学学报(自然科学版), 2013, 53:210-214.
- [119] XU S C, ADHITYA A, SRINIVASAN R. Hybrid model-based framework for alarm anticipation[J]. Industrial and Engineering Chemistry Research, 2014, 53:5182-5193.
- [120] WANG J D, ZHAO Y, BI Z F. Criteria and algorithms for online and offline detections of industrial alarm floods[J]. IEEE Transactions on Control Systems Technology, 2017, 99:1-10.
- [121] 张泽生, 李宏光, 杨博, 等. 基于置信规则库推理的过程报警预测方法[J]. 控制工程, 2019, 26:765-772.
- [122] ZHU J F, WANG C L, LI C K, et al. Dynamic alarm prediction for critical alarms using a probabilistic model[J]. Chinese Journal of Chemical Engineering, 2016, 24:881-885.
- [123] LAI S Q, YANG F, CHEN T W. Online pattern matching and prediction of incoming alarm floods[J]. Journal of Process Control, 2017, 56:69-78.
- [124] XU Y Z, WANG J D, YU Y. Alarm event prediction from historical alarm flood sequences based on Bayesian estimators[J]. IEEE Transactions on Automation Science and Engineering, 2020, 17:1070-1075.
- [125] CAI S, PALAZOGLU A, ZHANG L B, et al. Process alarm prediction using deep learning and word embedding methods[J]. ISA Transactions, 2019, 85:274-283.
- [126] DORGO G, ABONYI J. Learning and predicting operation strategies by sequence mining and deep learning[J]. Computers and Chemical Engineering, 2019, 128:174-187.
- [127] YANG F, XIAO D Y, SHAH S L. Optimal sensor location design for reliable fault detection in presence of false alarms[J]. Sensor, 2009, 9:8579-8592.

- [128] BLAAUWGEERS E, DUBOIS L, RYCKAERT L. Real-time risk estimation for better situational awareness[J]. IFAC Proceedings Volumes, 2013, 46:232-239.
- [129] ADNAN N A, CHENG Y, IZADI I, et al. Study of generalized delay-timers in alarm configuration[J]. Journal of Process Control, 2013, 23:382-395.
- [130] VARGA T, SZEIFERT F, ABONYI J. Detection of safe operating regions: a novel dynamic process simulator based predictive alarm management approach[J]. Industrial and Engineering Chemistry Research, 2010, 49:658-668.
- [131] XU S, ADHITYA A, SRINIVASAN R. Hybrid mode-based framework for alarm anticipation[J]. Industrial and Engineering Chemistry Research, 2014, 53:5182-5193.
- [132] CHARBONNIER S, BOUCHAIR N, GAYET P. Fault isolation by comparing alarm lists using a symbolic sequence matching algorithm[J]. IFAC Proceedings Volumes, 2014, 47:7085-7090.

第2章 报警系统状态分析

2.1 引言

报警系统对于现代化工业的安全生产和高效运行具有至关重要的作用。当异常不存在、异常发生、异常持续和异常消失时，报警系统分别处于正常、进入异常、保持异常及离开异常的工作状态。但是随着工厂的现代化程度提高，运行设备日益复杂，其监测数据和报警数据随之增多，工厂的操作人员很难从这些大量的数据中获得关于报警系统工作状态的信息。因此亟需设计一种数据可视化工具，以清晰、简洁地展示报警系统工作状态信息。

本章通过状态分析图描述单个和多个报警变量的工作状态。状态分析图由一些新定义的状态指标组成，包括发生报警的报警变量个数、新进入报警状态的报警变量个数、长时间报警变量个数。这些状态图，从变量维度和时间维度两个方面进行表示。变量维度表示状态图是研究单个报警变量还是多个报警变量；时间维度是将较长时间段 T （例如24小时或1周）分成几个具有相等长度 Δt （例如10分钟或1小时）的时间窗，如果 T 是当前（过去）时刻，则可以描述在线（离线）状态。

2.2 单变量报警状态分析

单变量报警器的状态分析是分析单个报警变量是处于报警状态还是非报警状态以及报警变量在报警状态和非报警状态之间的变化情况。

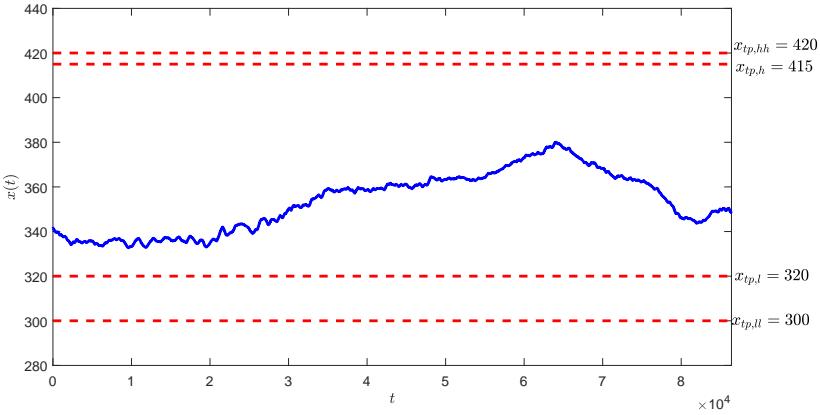
考虑一个过程变量 $x(t)$ ，其中 t 为时间 $t = 1, 2, 3, \dots, T$ 。令 $x_a(t)$ 为 $x(t)$ 对应的报警变量，当 $x(t)$ 处于正常工况时， $x_a(t)$ 处于非报警状态其值为‘0’；当 $x(t)$ 处于异常工况时， $x_a(t)$ 处于报警状态其值为‘1’。在实际工业中，最常用的是将报警变量 $x(t)$ 与一个固定的高（低）报警阈值 x_{tp} 比较的方法产生报警变量 $x_a(t)$ ：

$$x_a(t) = \begin{cases} 1, & x(t) \geq (\leq) x_{tp} \\ 0, & x(t) < (>) x_{tp} \end{cases}. \quad (2.1)$$

如图2.1通常报警系统可能会配置有四条报警阈值：高高报警阈值 $x_{tp,hh}$ ，高报警阈值 $x_{tp,h}$ ，低报警阈值 $x_{tp,l}$ 和低低报警阈值 $x_{tp,ll}$ 。定义一个包含四个元素的向量 X_{tp} 来表示这些报警阈值， $X_{tp} = [x_{tp,ll}, x_{tp,l}, x_{tp,h}, x_{tp,hh}]$ ，如果 X_{tp} 中某个元素不存在则用“NaN”代替。

除了报警阈值外， x_a 还有一些其他的属性 $x_{tag}, x_{desp}, x_{type}$ ：

- x_{tag} 代表了一个过程变量的标记名，这个标记名是工业分散式控制系统（DCS）中此过程变量的唯一标识符。工业中 x_{tag} 通常由数字、字母和符号组成，例如 $x_{tag} = X71NAO067$ 。
- x_{desp} 描述了一个过程变量的物理意义。例如，变量 $x_{tag} = X71NAO067$ 的描述为“催化剂上层和中层之间的压力差”。

图 2.1 过程变量 $x(t)$ 及其对应的报警阈值

- x_{type} 标记了一个变量是数字量还是模拟量，当一个变量是模拟量时， x_{type} 为‘1’，当这个变量是数字量时， x_{type} 为‘0’。例如， $x_{tag} = X71NAO067$ 是一个测量压力差的模拟量，因此 $x_{type}=1$ 。

为了更好的描述时间序列 $\{x_a(t), x(t)\}_{t=1}^T$ 的主要特征，下面定义了一些基础状态指标：

- 报警触发（清除）分别定义为 x_a 从‘0’变到‘1’（‘1’变到‘0’）。第 n 个报警触发和报警清除分别定义为

$$x_a(t_o(n)-1) = 0, \quad x_a(t_c(n)) = 0, \quad (2.2)$$

$$\sum_{t=t_o(n)}^{t_c(n)} x_a(t) = t_c(n) - t_o(n), \quad t_c(n) > t_o(n)$$

其中 $t_o(n)$ 和 $t_c(n)$ 为报警触发时刻和报警清除时刻。

- 报警持续时间为一个报警从报警触发时刻到报警清除时刻的持续时间，第 n 个报警持续时间的定义为

$$D_a(n) = (t_c(n) - t_o(n)) h, \quad (2.3)$$

其中 h 为采样周期。

- 报警间隔时间为从一个报警清除时刻到下一个报警触发时刻间的时间间隔，第 n 个报警间隔时间的定义为

$$I_a(n) = (t_o(n+1) - t_c(n)) h \quad (2.4)$$

其中 $t_o(n+1)$ 为第 $(n+1)$ 个报警触发时刻。

- 游程长度是从一次报警触发时刻到下一次报警触发时刻之间的时间长度，第 n 个游程长度定义为

$$R_a(n) = (t_o(n+1) - t_o(n)) h. \quad (2.5)$$

显然，报警持续时间、报警时间间隔和游程长度之间有关系 $R(n) = D(n) + I(n)$ 。

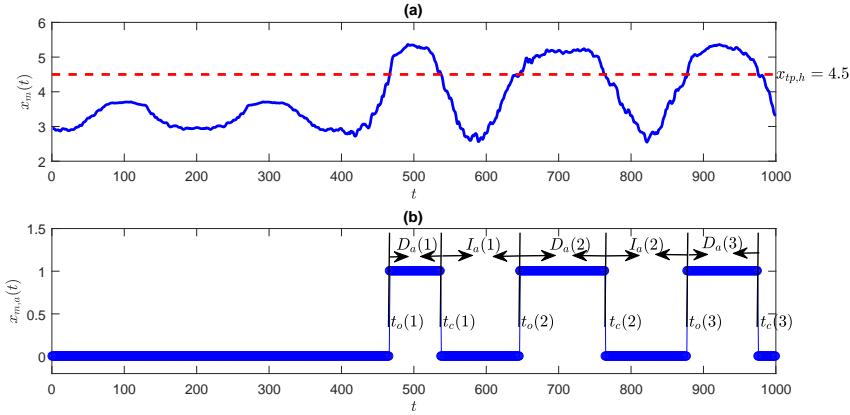


图 2.2 单报警变量状态指标说明示例

图2.2给出了报警触发时刻、报警清除时刻、报警持续时间和报警间隔时间的详细说明。

对于某一时间段 $[t_j, t_j + \Delta t]$ 的数据 ($j \in [1, J]$, Δt 为时间段长度), x_a 的报警触发数量为

$$N_{occ}(j) = \sum_{t=t_j}^{t_j+\Delta t} x_{a,o}(t), \quad (2.6)$$

其中 $x_{a,o}(t)$ 和 $x_a(t)$ 相关, 当且仅当 $x_a(t)$ 的值从‘0’变到‘1’时, $x_{a,o}(t)$ 的取值为‘1’, 其余时刻 $x_{a,o}(t)$ 的值均为‘0’, 即

$$x_{a,o}(t) = \begin{cases} 1, & x_a(t-1)=0 \text{ 且 } x_a(t)=1 \\ 0, & \text{其他} \end{cases}. \quad (2.7)$$

$x_a(t)$ 从‘0’变到‘1’的时刻称为报警发生时刻 $\{t_o(n)\}_{n=1}^{N_{occ}(j)}$, 即

$$x_{a,o}(t_o(n)) = 1, n \in [1, N_o(j)].$$

相应的, 时间段 $[t_j, t_j + \Delta t]$ 内的报警清除次数为

$$N_{cle}(j) = \sum_{t=t_j}^{t_j+\Delta t} x_{a,c}(t). \quad (2.8)$$

当且仅当 $x_a(t)$ 的值从‘1’变到‘0’时, $x_{a,c}(t)$ 的取值为‘1’, 其余时刻 $x_{a,c}(t)$ 的值均为‘0’, 即

$$x_{a,c}(t) = \begin{cases} 1, & x_a(t-1)=1 \text{ 且 } x_a(t)=0 \\ 0, & \text{其他} \end{cases}. \quad (2.9)$$

$x_a(t)$ 从‘1’变到‘0’的时刻称为报警清除时刻 $\{t_c(n)\}_{n=1}^{N_{cle}(j)}$, 即

$$x_{a,c}(t_c(n)) = 1, n \in [1, N_{cle}(j)].$$

第 j 个时间段内的报警持续时间、报警间隔时间和报警偏差为三个实数集合。但是由于报警触发次数和报警清除次数的不同, 他们的计算比式(2.3)(2.4)要更加复杂。

如果 $N_{occ}(j) + N_{cle}(j) \leq 1$, 则 x_a 有四种情况, 对应图2.3(a)、(b)、(c)和(d):

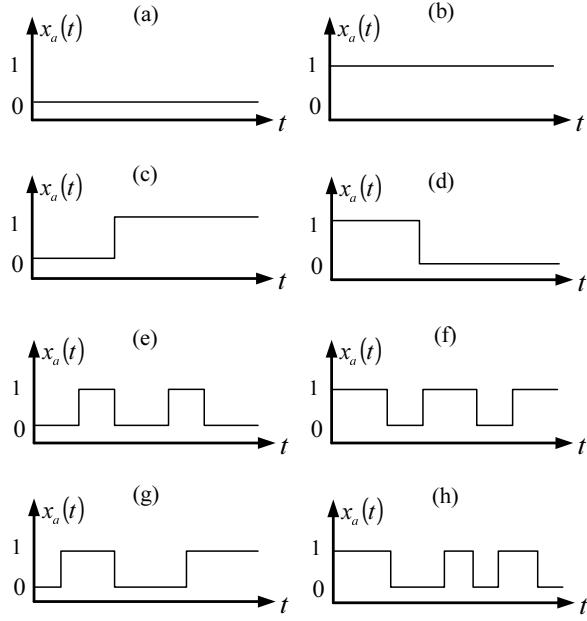


图 2.3 警报触发和报警清除的不同情况

- $x_a(t) = 0, \forall t \in [t_j, t_j + \Delta t], N_{occ}(j) = N_{cle}(j) = 0.$
- $x_a(t) = 1, \forall t \in [t_j, t_j + \Delta t], N_{occ}(j) = N_{cle}(j) = 0.$
- $x_a(t_j) = 1, x_a(t_j + \Delta t) = 0, N_{occ}(j) = 0, N_{cle}(j) = 1.$
- $x_a(t_j) = 0, x_a(t_j + \Delta t) = 1, N_{occ}(j) = 1, N_{cle}(j) = 0.$

此时，报警持续时间、报警间隔时间和报警偏差为

$$D_a(1) = h \cdot \sum_{t=t_j}^{t_j+\Delta t} x_a(t), \quad (2.10)$$

$$I_a(1) = h \cdot \sum_{t=t_j}^{t_j+\Delta t} (1 - x_a(t)). \quad (2.11)$$

如果 $N_{occ}(j) + N_{cle}(j) > 1$ ，则对应其他四种情况：

- $x_a(t_j) = 0, x_a(t_j + \Delta t) = 0$ (图2.3(e))。这种情况下，报警触发次数和报警清除次数相同 $N_{occ}(j) = N_{cle}(j)$ 。报警持续时间的集合 (定义为 $S_D(j)$ ，包含 $N_{occ}(j)$ 个元素) 为

$$S_D(j) := \{D(n) = t_c(n) - t_o(n)\}_{n=1}^{N_{occ}(j)}. \quad (2.12)$$

报警间隔时间的集合 (定义为 $S_I(j)$ ，包含 $N_{cle}(j) + 1$ 个元素) 为

$$S_I(j) := \left\{ \begin{array}{l} \{I(n) = t_o(n+1) - t_c(n)\}_{n=1}^{N_{cle}(j)-1}, \\ t_o(1) - 1, t_j + \Delta t - t_c(N_{cle}(j)) \end{array} \right\}. \quad (2.13)$$

- $x_a(t_j) = 1, x_a(t_j + \Delta t) = 1$ (图2.3(f))。在这种情况下，报警触发次数和报警清除次数相同，即 $N_{occ}(j) = N_{cle}(j)$ 。报警持续时间的集合（包含 $N_{occ}(j) + 1$ 个元素）为

$$S_D(j) := \left\{ \begin{array}{l} \{D(n) = t_c(n+1) - t_o(n)\}_{n=1}^{N_{occ}(j)-1}, \\ t_c(1)-1, t_j + \Delta t - t_o(N_{occ}(j)) \end{array} \right\}. \quad (2.14)$$

报警间隔时间的集合包含 $N_{cle}(j)$ 个从报警清除时刻到下一次报警触发时刻的时间间隔：

$$S_I(j) := \{I(n) = t_o(n) - t_c(n)\}_{n=1}^{N_{cle}(j)}. \quad (2.15)$$

- $x_a(t_j) = 0, x_a(t_j + \Delta t) = 1$ (图2.3(g))。在这种情况下，由于本段结尾的报警信号等于1，所以报警触发次数比报警清除次数多1 ($N_{occ}(j) = N_{cle}(j) + 1$)。报警持续时间的集合包含 $N_{cle}(j)$ 个成对的报警触发时刻和报警清除时刻之间的持续时间及末尾 x_a 等于1的部分，即

$$S_D(j) := \left\{ \begin{array}{l} \{D(n) = t_c(n) - t_o(n)\}_{n=1}^{N_{cle}(j)}, \\ t_j + \Delta t - t_o(N_{occ}(j)) \end{array} \right\}. \quad (2.16)$$

报警间隔时间的集合包括 $N_{cle}(j)$ 个成对的报警触发时刻和报警清除时刻之间的间隔时间及初始 x_a 等于0的部分，即

$$S_I(j) := \left\{ \begin{array}{l} \{I(n) = t_o(n+1) - t_c(n)\}_{n=1}^{N_{cle}(j)}, \\ t_o(1) - 1 \end{array} \right\}. \quad (2.17)$$

- $x_a(t_j) = 1, x_a(t_j + \Delta t) = 0$ (图2.3(h))。在这种情况下，由于本段初始部分 x_a 等于1，所以本段报警触发次数比报警清除次数少1，报警持续时间的集合为

$$S_D(j) := \left\{ \begin{array}{l} \{D(n) = t_c(n+1) - t_o(n)\}_{n=1}^{N_{occ}(j)}, \\ t_c(1) - 1 \end{array} \right\}. \quad (2.18)$$

报警间隔时间的集合为

$$S_I(j) := \left\{ \begin{array}{l} \{I(n) = t_o(n) - t_c(n)\}_{n=1}^{N_{occ}(j)}, \\ t_j + \Delta t - t_c(N_{cle}(j)) \end{array} \right\}. \quad (2.19)$$

基于上述单变量报警状态指标，表2.1给出了单变量报警典型的工作状态及不同的工作状态中状态指标的变化情况：

表 2.1 单变量报警典型工作状态及对应的状态指标

	工作状态	N_{occ}	D_a	I_a
1	处于非报警状态	0	0	大值
2	处于报警状态	0	大值	0
3	进入报警状态	增加	增加	减少
4	离开报警状态	减少	减少	增加
5	受干扰报警影响	大值	小值	小值

- 状态#1：处于非报警状态。如果报警变量处于非报警状态（如图2.3(a) 所示），

表 2.2 四个报警系统及其对应的属性。

X_a	X_{desp}	K	X_{pri}
$X_{a,1}$	MEH 系统	113	[12, 16, 85]
$X_{a,2}$	凝汽器除氧系统	171	[38, 32, 101]
$X_{a,3}$	制粉系统	186	[67, 31, 88]
$X_{a,4}$	发电机系统	96	[41, 13, 42]

则 $x_a(t)$ 一直为 0 且没有触发报警，因此 $N_{occ} = D_a = 0$ 而 I_a 为较大值。

- 状态#2：处于报警状态。如果报警变量处于报警状态（如图2.3(b)所示），则 $x_a(t)$ 一直为 1 且没有报警被触发，因此 $N_{occ} = I_a = 0$ 而 D_a 为较大值。
- 状态#3：进入报警状态。如果报警变量正在进入报警状态（如图2.3(c) 所示），这就意味着变量正在从状态#1 向#2变化。因此 N_{occ} 和 D_a 会增加而 I_a 会减少。
- 状态#4：离开报警状态。如果报警变量正在离开报警状态（如图2.3(d) 所示），意味着变量正在从状态#2向#1变化。因此 N_{occ} 和 D_a 会减少而 I_a 会增加。
- 状态#5：受干扰报警影响。干扰报警是由于随机噪声或干扰而产生的，干扰报警的特点是报警持续时间短和报警间隔时间短。如果报警变量受干扰报警影响，则 N_{occ} 为较大值而 D_a 和 I_a 则为较小值。

2.3 多变量报警状态分析

多变量报警器的状态分析是分析多个报警变量是处于正常状态还是异常状态以及多个报警变量在正常状态和异常状态之间的变化情况。

基于物理意义，将多个报警变量 $x_{a,1}, x_{a,2}, \dots, x_{a,K}$ 组合在一起的集合称为一个报警系统 $X_a = \{x_{a,k}\}_{k=1}^K$ 。报警系统 X_a 的基础属性包括：报警系统的描述 X_{desp} ； X_a 中报警变量的个数 K ；报警优先级向量 $X_{pri} := [N_H, N_M, N_L]$ ，其中 N_H, N_M, N_L 分别为 X_a 中高、中、低优先级报警变量的个数，表2.2展示了四个报警系统及其对应属性。对于第 j 个时间段 $[t_j, t_j + \Delta t]$ ， X_a 的几个状态指标如下：

- 报警变量总数 X_a 中发生过报警的报警变量个数：

$$N_{var}(j) = \sum_{k=1}^K \delta \left(\sum_{t=t_j}^{t_j+\Delta t} x_{a,k}(t) \right), \quad (2.20)$$

其中 $\delta(x)$ 为

$$\delta(x) = \begin{cases} 1, & x \geq 1 \\ 0, & x = 0 \end{cases}.$$

- 新进入报警状态的报警变量个数为 X_a 中当前时间段处于报警状态而前一时间段不在报警

表 2.3 多变量报警典型工作状态及对应的状态指标

	工作状态	N_{var}	N_{new}	N_{sta}	N_{occ}
A	处于正常状态	0	0	0	0
B	处于异常状态	常值	0	常值	0
C	进入异常状态	增加	增加	增加	增加
D	离开异常状态	减少	0	减少	0
E	受干扰报警影响	增加	0	0	大值

状态的报警变量个数:

$$N_{new}(j) = \sum_{k=1}^K I_k(j), \quad (2.21)$$

其中 $I_k(j)$ 用来指示第 k 个报警变量是否进入报警状态,

$$I_k(j) = \begin{cases} 1, & \delta_1 = 1 \text{ 且 } \delta_2 = 0 \\ 0, & \text{其他} \end{cases},$$

这里 $\delta_1 = \delta\left(\sum_{t=t_j}^{t_j+\Delta t} x_{a,k}(t)\right)$, $\delta_2 = \delta\left(\sum_{t=t_{j-1}}^{t_{j-1}+\Delta t} x_{a,k}(t)\right)$ 。

- 长时间报警变量个数为 X_a 中在第 j 个时间段内一直处于报警状态的报警变量个数:

$$N_{sta}(j) = \sum_{k=1}^K J_k(j), \quad (2.22)$$

其中 $J_k(j)$ 用来指示第 k 个报警变量是否一直处于报警状态,

$$J_k(j) = \begin{cases} 1, & \delta\left(\sum_{t=t_j}^{t_j+\Delta t} x_{a,k}(t)\right) = \Delta t \\ 0, & \text{其他} \end{cases}.$$

- 报警触发数量为 X_a 中所有报警变量的报警触发次数总和:

$$N_{occ}(j) = \sum_{k=1}^K \sum_{t=t_j}^{t_j+\Delta t} x_{a,o,k}(t). \quad (2.23)$$

第 k 个报警变量 $x_{a,k}$, $x_{a,o,k}(t)$ 被定义为式(2.7)所示。

基于上述多变量报警状态指标, 表2.3给出了多变量报警典型的工作状态及不同的工作状态中状态指标的变化情况:

- 状态#A: 处于正常状态。如果 X_a 处于正常状态, 则不会有报警出现。因此, N_{var} 、 N_{new} 、 N_{sta} 和 N_{occ} 的值都为0。
- 状态#B: 处于异常状态。如果 X_a 处于异常状态, 则会有一定的报警变量一直处于报警状态。因此, N_{var} 和 N_{sta} 为一常值且 $N_{new} = N_{occ} = 0$ 。
- 状态#C: 进入异常状态。如果 X_a 进入异常状态, 则会有一些报警变量从非报警状态转变为报警状态。因此 N_{var} 、 N_{new} 、 N_{sta} 和 N_{occ} 的值都会增加。

- 状态#D: 离开报警状态。如果 X_a 离开报警状态，则会有一些报警变量从报警状态转变为非报警状态。因此， N_{var} 和 N_{sta} 会减少，同时 N_{new} 和 N_{occ} 一直为0。
- 状态#E: 受干扰报警影响。如果 X_a 受干扰报警影响，则 N_{var} 和 N_{occ} 会增加，同时 N_{new} 和 N_{occ} 等于0或为很小的值。由于干扰警报在短时间内频繁出现和消失，所以此时 N_{occ} 的值很大。

2.4 案例分析

本部分介绍了上述状态分析图在某300mw火电机组的应用效果，并与文献中已有的解析图进行了比较。发电机组的主要工作原理是，将煤燃烧产生的热能通过发电动力装置（锅炉、汽轮机和发电机辅助装置等）转化成电能。发电机组的DCS中涉及24079个过程变量。模拟过程变量有8145个，数字过程变量有15934个，其中分别针对208个模拟过程变量和2018个数字过程变量配置了报警生成机制。

考虑一个包含26个报警变量的报警系统 X_a ，该系统涉及锅炉-汽轮机控制蒸汽产生过程。时间段 T 和时间长度 Δt 分别选择为24小时和1小时。图2.4展示了 X_a 的可视化状态分析

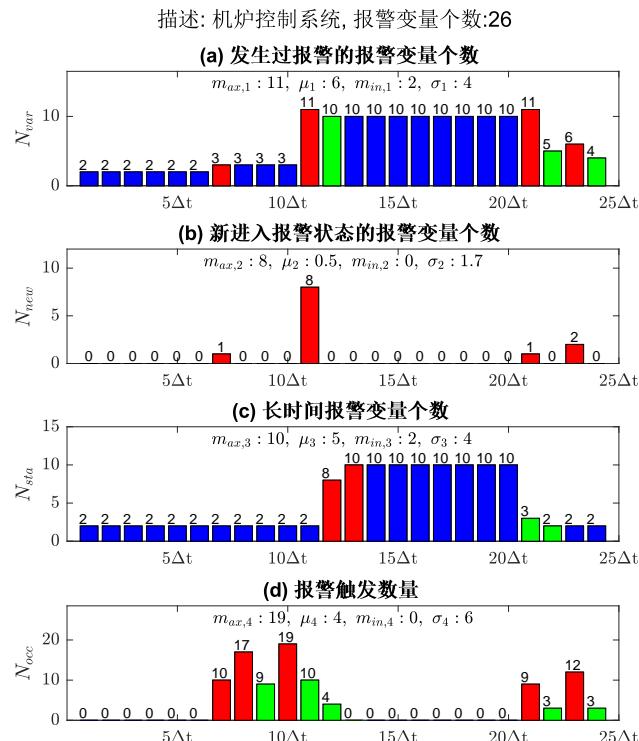


图 2.4 多变量可视化状态分析图工业案例

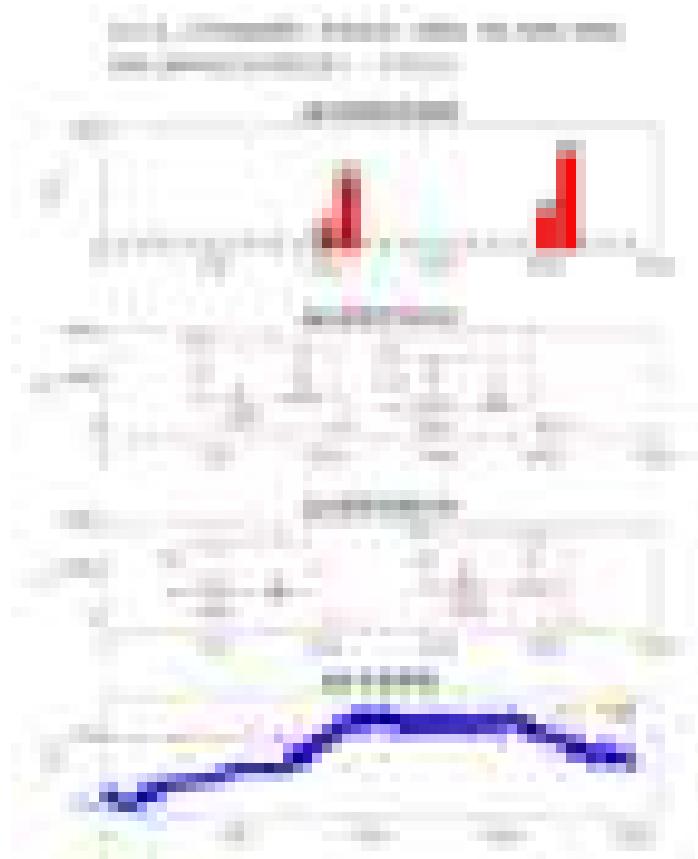


图 2.5 单变量可视化状态图工业案例

图。从图中可以看出，有两个报警变量在 $1\Delta t-10\Delta t$ 期间一直处于报警状态，此时 $N_{var}(j)=N_{sta}(j)=2$, $N_{new}(j)=N_{occ}(j)=0$ ($j=1, 2, \dots, 6$)。在 $7\Delta t$ 时有一个新的报警变量进入报警状态，此时 N_{var} 从2增加到3， N_{new} 从0增加到1， N_{sta} 一直为2， $N_{occ}(j)$ 在 $j=7, \dots, 10$ 时分别为[10, 17, 9, 19]。在 $11\Delta t$ 时 X_a 出现了异常，此时 $N_{var}(j)$ 从3增加到8， N_{new} 从0增加到8， N_{sta} 从2增加到8。并且这个异常在 $12-20\Delta t$ 期间一直持续，此期间 $N_{var}=N_{sta}=10$ 且 $N_{occ}(j)=N_{new}(j)=0$ 。也就是说，在 $11\Delta t$ 时有8个新的报警变量进入到报警状态并且这几个变量在 $13-20\Delta t$ 期间一直处于报警状态。最后根据图2.4可以看到在 $20\Delta t$ 之后异常逐渐消失，此时状态指标 $N_{var}(j)$ 、 $N_{new}(j)$ 、 $N_{sta}(j)$ 和 $N_{occ}(j)$ 的逐渐趋于它们的初始值。因此，图2.4中的状态分析图以简洁明了的方式揭示了表2.3 中的五种典型状态。

图2.5为本节的工业案例给出了一个单报警变量的可视化状态图，展示的是一个工业实际变量2015年某天（24小时）的时间序列及其对应的状态指标，图2.5顶部介绍了该变量的标记名、报警优先级、报警阈值和变量描述，子图(a)-(d)分别为报警触发次数、报警持续时间、报警间隔时间和报警偏差四个状态指标的箱型图，子图(e)为该变量的时间序列（蓝色实线）及报警阈值（红色虚线）。显然， x_a 在 $j \in [1, 9], j \in [22, 24]$ 期间处于非报警状态，因为此时 $N_{occ}(j)$ 和 $S_D(j)$ 为0而 $S_I(j) = 3600$ 。 x_a 在时间段 $j \in [12, 19]$ 处于报警状态，此时 $N_{occ}(j) = 0$, $S_D(j) = 3600$, $S_I(j) = 0$ 。因此， x_a 在 $11\Delta t$ 和 $12\Delta t$ 分别经历进入报警状态和

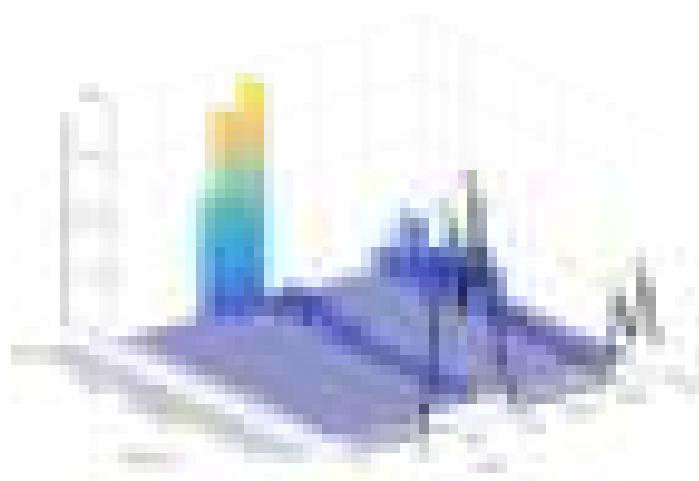


图 2.6 3D 高密度报警图

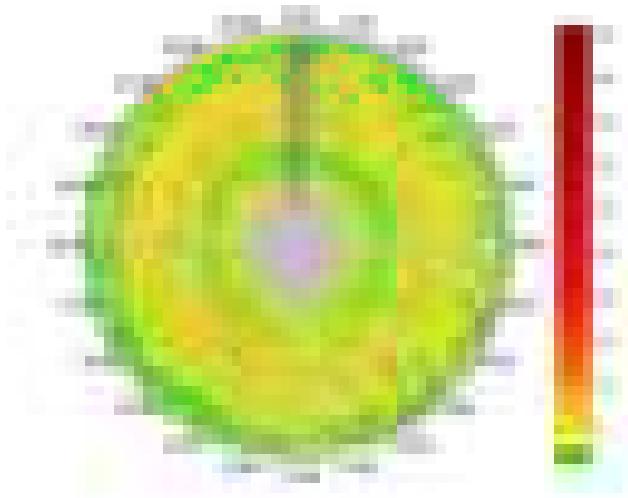


图 2.7 报警触发数量螺旋图

离开报警状态阶段。当 x_a 进入报警状态或离开报警状态时，出现了大量的干扰报警，图2.5的子图(b)和(c)放大区域可以看到在 $11-12\Delta t$ 和 $20-21\Delta t$ 期间大多数报警持续时间和报警间隔都小于10秒。因此，图2.5的状态分析图清晰地说明了表2.1中的五种典型状态。

本部分给出了文献中三种可视化分析图作为图2.4和图2.5的比较。

- 图2.6是3D高密度警报图^[4]。图中的每一行展示出了不同时间段内报警变量的报警次数。因此，它和图2.5(a)是类似的。最后一行的黑色实线与图2.4(d)中的 $N_{occ}(j)$ 相同。通过将报警变量与同一时间段内的报警事件对齐，可以很容易地找到相关的报警变量、报警洪水和滋扰报警。但是图2.6并没有涉及单变量的 D_a 和 I_a 指标以及多变量的 N_{var} 、 N_{new} 和 N_{sta} 指标。

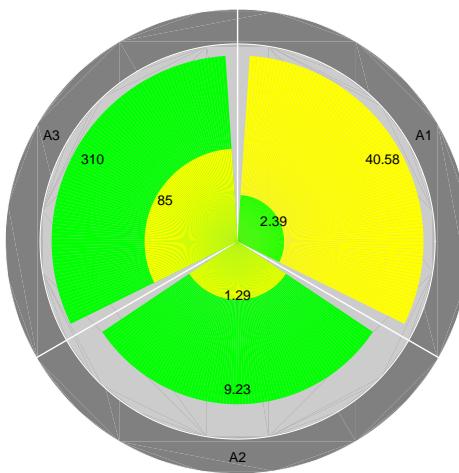


图 2.8 多层雷达图

- 图2.7是报警触发数量的螺旋图^[1]。图中的颜色与报警次数是成比例。因此，它类似于图2.4(d)中的 $N_{occ}(j)$ 。通过在长达30天的较长时间内对报警发生次数的可视化展示，可以从图中很容易地注意到报警的长期周期性模式。但是，它并没有考虑到单变量的 D_a 和 I_a 指标以及多变量的 N_{var} 、 N_{new} 和 N_{sta} 指标。
- 图2.8为多层次雷达图^[2]。图中比较了两个不同时间段的指标，分别用绿色和黄色表示。图中涉及指标分别为图2.5(d)中变量X71NAO067的报警间隔时间A1、报警持续时间A2和报警触发数量A3。该图能判断相邻时间段之间的状态变化，但是展示的时间范围和状态指标都相当有限。

2.5 本章小结

本章从单个报警变量和多个报警变量两个角度介绍了一些常用的基础概念及状态指标，包括连续时间段内单个报警变量的一些标准状态指标：报警触发数量、报警持续时间和报警间隔，以及连续时间段内多个报警变量的一些新定义状态指标：发生过报警的报警变量个数、新进入报警状态的报警变量个数和长时间报警变量。基于这些状态指标设计了可视化状态图，清晰、简洁地展示报警系统工作状态信息，同时总结了报警变量典型的工作状态及不同工作状态中状态指标的变化情况。

参考文献

- [1] HU W, AL-DABBAGH A W, CHEN T, et al. Design of visualization plots of industrial alarm and event data for enhanced alarm management[J]. Control Engineering Practice, 2018, 79:50-64.
- [2] AL-DABBAGH A W, HU W, LAI S, et al. Toward the advancement of decision support tools for industrial facilities: Addressing operation metrics, visualization plots, and alarm floods[J]. IEEE Transactions on Automation Science and Engineering, 2018, 15(4):1883-1896.

第3章 单变量报警器设计

3.1 引言

报警器的基本任务是在异常状况出现时发出正确的报警信息，避免噪声干扰导致的干扰报警。然而，工业调查结果表明，目前普遍存在的问题是报警信号的数量过多，有时会在短短几十分钟内会产生几百个报警，导致操作人员无法及时处理这些报警，以致于忽略报警信息，甚至关闭报警系统。这些问题的出现，会给生产过程带来极大的安全隐患，因此，工业报警系统的研究课题在工业领域和学术领域得到了越来越多的关注^[4-10]。

越限报警器、报警延迟器、报警死区等单变量报警器在工业现场被广泛使用，通过优化设计报警阈值、以及报警延迟器和报警死区的参数，可以有效地减少干扰报警的数量。本章首先给出了异常和正常运行状况的推断方法，用来判断工业数据是属于正常运行状况还是异常运行状况；然后，以处于正常和异常运行状况的历史数据为基础，给出了报警延迟器的延时参数和报警死区的死区宽度的优化设计，并设计了一种死区指数，用来判断针对某一个过程变量，报警延迟器和报警死区哪个更适用于消除干扰报警；之后，针对具有马尔科夫性的过程变量，提出了一种基于报警概率图的报警阈值设计方法；最后，综合使用了上述的方法，研制了火电机组智能报警系统，用于在工业实践中识别和消除干扰报警。

3.2 正常或异常运行状况推断

报警信息自身没有正确报警或干扰报警的标签，因此识别正确报警和干扰报警本质是，判断与报警信息对应的过程变量是处于正常或异常运行状况，即当过程变量处于正常运行状况时，出现的报警信息属于误报警；当过程变量处于异常运行状况时，缺失的报警信息属于漏报警。因此，自动推断处于正常或异常运行状况的数据段是非常重要的，是评价报警器性能和优化设计报警器的核心步骤。

拟解决的问题描述如下：假设报警变量 $x_a(t)$ 通过最为常见的越限报警器来生成，即当相应的过程变量 $x(t)$ 超越报警阈值 x_{tp} 时，报警器发出报警信息，

$$x_a(t) = \begin{cases} 1, & x(t) \geq (\leq)x_{tp} \\ 0, & x(t) < (>)x_{tp} \end{cases}. \quad (3.1)$$

在工业环境下，过程变量 $x(t)$ 通常包含有噪声，因此可以假设 $x(t)$ 包含无噪声部分 $x_o(t)$ 和噪声部分 $e(t)$ ，即 $x(t) = x_o(t) + e(t)$ 。由于噪声 $e(t)$ 的存在， $x_a(t)$ 混杂着正确报警和干扰报警。正确报警对应着真正的异常运行状况，需要得到现场运行操作员及时处理，避免异常运行状况恶化为生产事故；相反的，干扰报警并不对应真正的异常运行状况，是由噪声 $e(t)$ 引发的。给定历史数据样本 $\{x(t), x_a(t)\}_{t=1}^T$ 及对应的报警阈值 x_{tp} ，本节主要介绍如何划分处于正常和异常运行状况的数据段，以便进行单变量报警器的性能评价和优化设计。

3.2.1 基于均值变化点检测的正常和异常数据点划分

本节提出一种基于均值变化点检测的正常和异常数据段划分方法。该方法需要如下假设条件：在正常和异常运行状况下，过程变量 $x(t)$ 的概率密度函数具有不同的均值。

该假设意味着，可以依赖报警阈值 x_{tp} 和过程变量 $x(t)$ 在近期数据段的均值，判断 $x(t)$ 是处于正常或异常运行状况。例如，当 x_{tp} 是高报警阈值时，从统计角度出发，如果数据段的均值大于 x_{tp} ，则认为这段数据处于异常运作状况。在工业现场，可以依据过程变量对应的工艺知识判断该假设是否成立。关于均值的假设检验是易于进行的，因此这里的难点主要是识别 $x(t)$ 发生均值变化的时刻，即利用均值变化检测方法来确定 $\{x(t)\}_{t=1}^T$ 的均值变化点。^[1] Pettit 提出了单个均值变化点的非参数检测方法，这种非参数方法表现良好，但是不能直接用于检测多个均值变化点。基于这种方法，我们利用二分法的思想，提出了基于均值变化点检测的正常和异常数据段划分方法。具体步骤如下：

第一步：对于数据 $\{x(t)\}_{t=1}^T$ ，找出单个均值变化点^[1]：

1) 计算每个数据点所处的排名统计量

$$V_{t,T} = \sum_{j=1}^T \text{sgn}(x(t) - x(j)), \quad t = 1, 2, \dots, T$$

这里

$$\text{sgn}(x) = \begin{cases} 1, & x > 0; \\ 0, & x = 0; \\ -1, & x < 0. \end{cases}$$

计算检验统计量 $U_{1,T} = V_{1,T}$ 和 $U_{t,T} = U_{t-1,T} + V_{t,T}$ ，其中 $t = 2, 3, \dots, T$ 。

2) 找出使得 $|U_{t,T}|$ 最大的时刻 t_{max} ，然后利用下式计算相应的假设检验 P 值

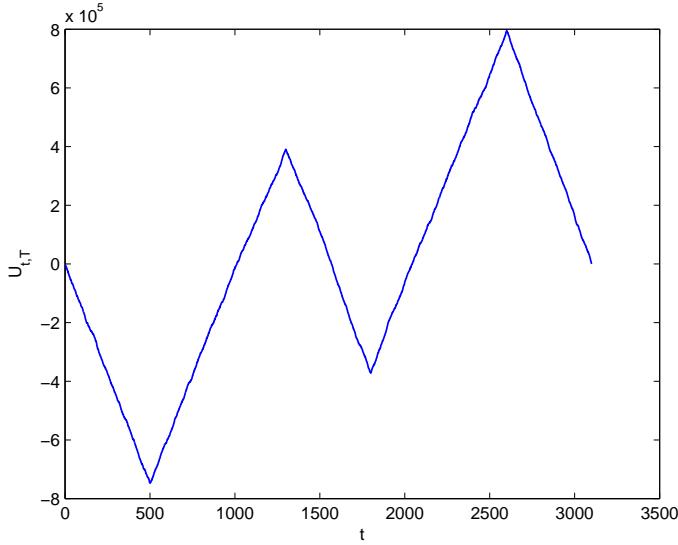
$$P = 2 \exp \left(\frac{-6 \max_{1 \leq t < T} |U_{t,T}|^2}{T^2 + T^3} \right).$$

3) 设定第一类错误的概率 α ，例如设定 $\alpha = 0.01$ ，定义原假设为： $x(t_{max})$ 不是均值变化点。如果 $P < \alpha$ ，则拒绝原假设， $x(t_{max})$ 是 $\{x(t)\}_{t=1}^T$ 的均值变化点。如果 $P > \alpha$ ，则接受原假设， $x(t_{max})$ 不是 $\{x(t)\}_{t=1}^T$ 的均值变化点，即不存在均值变化点。

第二步：根据第一步找出的 t_{max} ，把数据 $\{x(t)\}_{t=1}^T$ 分成 $\{x_1(t)\}_{t=1}^{t_{max}}$ 和 $\{x_2(t)\}_{t=t_{max}+1}^T$ 两段。回到第一步，找出每一个子段的均值变化点。然后重复第一步和第二步，直到找出所有的均值变化点。

第三步：找出所有的变化点后，基于这些变化点将数据 $\{x(t)\}_{t=1}^T$ 分成若干段。计算每个数据段的样本均值，并采用经典的均值假设检验方法^[1]与报警阈值 x_{tp} 比较。对于某一个数据段 $\{x(t)\}_{t=t_0}^{t_1}$ ，计算 t 分布检验统计量

$$t = \frac{\bar{x} - x_{tp}}{s/\sqrt{t_1 - t_0}},$$

图 3.1 数据段 $x(1:3100)$ 的检验统计量 $U_{t,T}$

其中 \bar{x} 和 s 是 x 的样本均值和标准差，即

$$\bar{x} = \frac{\sum_{t=t_0}^{t_1} x(t)}{t_1 - t_0 + 1}, \quad s = \sqrt{\frac{\sum_{t=t_0}^{t_1} (x(t) - \bar{x})^2}{t_1 - t_0}}.$$

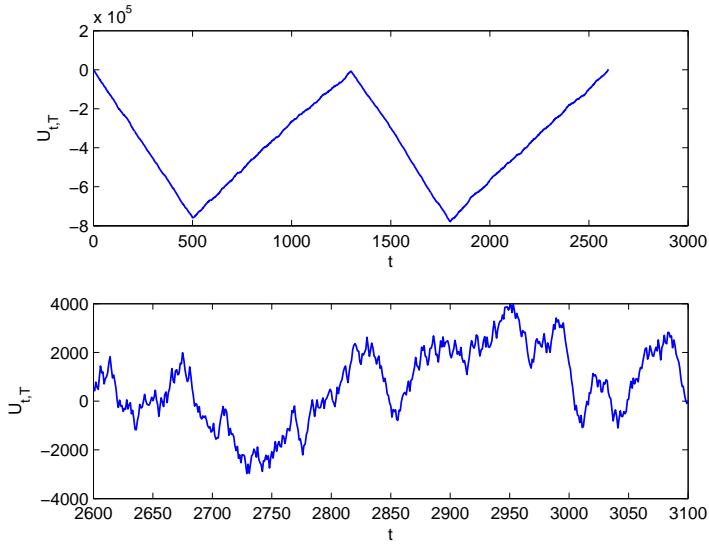
不失一般性，假设 x_{tp} 是高报警阈值。如果 t 大于临界值 t_{β,t_1-t_0} ，则样本均值 \bar{x} 在统计意义上大于报警阈值 x_{tp} ，数据段 $\{x(t)\}_{t=t_0}^{t_1}$ 处于异常运行状况；如果 t 小于临界值 $-t_{\beta,t_1-t_0}$ ，则样本均值 \bar{x} 在统计意义上小于报警阈值 x_{tp} ，数据段 $\{x(t)\}_{t=t_0}^{t_1}$ 处于正常运行状况。如果 t 在两个临界值之间，则认为样本均值 \bar{x} 在统计意义上等于报警阈值 x_{tp} ，无法判断这段数据是处于正常或异常运行状况。在均值假设检验中，将第一类错误的概率定义为 β ，例如 $\beta = 0.05$ 。最后，将所有检验出的正常数据段归为一组作为处于正常运行状况的历史数据集，将所有检测出的异常数据段归为另一组作为处于异常状况的历史数据集。

这里通过一个仿真例子来说明该方法的步骤。采用如下方式来产生过程变量 $x(t)$ 的数据

$$\left\{ \begin{array}{ll} x(t) & \sim N(0, 0.5^2), & t < 500 \\ x(t) & \sim \Gamma(3, 0.7), & 500 \leq t < 1300 \\ x(t) & \sim N(0, 0.5^2), & 1300 \leq t < 1800 \\ x(t) & \sim \Gamma(3, 0.7), & 1800 \leq t < 2600 \\ x(t) & \sim N(0, 0.5^2), & 2600 \leq t \leq 3100 \end{array} \right. \quad (3.2)$$

可以看出，过程变量 $x(t)$ 的均值变化点是 500, 1300, 1800, 2600。设定报警阈值 $x_{tp} = 1$ ，那么正常数据服从正态分布，异常数据服从 Gamma 分布。

首先，对于整个数据段 $x(1 : 3100)$ ，检验统计量 $U_{t,T}$ 的计算结果如图 3.1 所示。使得 $|U_{t,T}|$ 最大的时刻是 $t = 2600$ ，相应的 P 值为 $P = 1.03 \times 10^{-55}$ 。选择犯第一类错误的概率为 $\alpha = 0.01$ ，因为 $P < \alpha$ ，可知 $t = 2600$ 是被检测出的均值变化点。

图 3.2 数据段 $x(1:2599)$ (上) 和 $x(2600:3100)$ (下) 的检验统计量 $U_{t,T}$ 表 3.1 均值变化点检测结果及相应的 P 值

变化点位置	P 值
502	1.25×10^{-172}
1300	1.47×10^{-67}
1800	9.31×10^{-91}
2600	1.03×10^{-55}

其次，对于数据段 $x(1 : 2599)$ 和 $x(2600 : 3100)$ ，各自的检验统计量 $U_{t,T}$ 的计算结果如图3.2所示。相应的，对于数据段 $x(1 : 2599)$ ，使得 $|U_{t,T}|$ 最大的时刻是 $t = 502$ ， P 值为 $P = 1.25 \times 10^{-172}$ 。对于数据段 $x(2600 : 3100)$ ，使得 $|U_{t,T}|$ 最大的时刻是 $t = 2954$ ， P 值为 $P = 0.9457$ 。前者的 P 值小于 α ，后者则大于 α 。因此，可得结论： $t = 502$ 时刻的采样点是数据段 $x(1 : 2599)$ 的均值变化点， $t = 2954$ 时刻的采样点不是数据段 $x(2600 : 3100)$ 的均值变化点。

类似地，重复上述步骤，直到找出所有的均值变化点，其中，当实施第四次变化点检测的时候，已经找不到均值变化点。所有被检测出来的均值变化点及相应的 P 值见表3.1。

最后，按照上述实施步骤的第三步，在均值检验检验中，设定第一类错误的概率 $\beta = 0.01$ ，基于被检测出的均值变化点和报警阈值 x_{tp} ，可区分出正常数据段及异常数据段，如表3.2 所示，该结果与公式(3.2)中的真实情况是一致的。

3.2.2 基于线性分段表示的正常和异常数据段划分

本节提出了一种基于线性分段表示方法的正常和异常数据段划分方法。上一节方法假设

表 3.2 正常数据段及异常数据段的检测结果

正常数据	异常数据
x(1:501)	x(502:1299)
x(1300:1799)	x(1800:2599)
x(2600:3100)	

过程变量存在均值变化，换言之，同一均值的数据段是统计平稳的；与之相比，本节方法适用于统计平稳和非平稳的随机变量。该方法基于这样一个工艺知识：如果 t_a 时刻发生报警（即 $x_a(t_a) = 1$ ），我们需要根据 $x(t_a)$ 相邻的样本点来判断 $x(t_a)$ 是否属于异常运行状况，而不是仅仅只比较 $x(t_a)$ 和 x_{tp} 来判断 $x(t_a)$ 的运行状况。换言之，如果 $x(t_a)$ 相邻的数据点在统计意义上显著大于报警阈值 x_{tp} ，那么 $x(t_a)$ 属于异常运行状况。该工艺知识是普遍适用的，相比单个数据点的幅值，现场运行人员更关注多个相邻数据点的连续趋势，主要增加、平稳和减少这三种定性变化趋势。因此，本节采用线性分段表示这一时间序列挖掘技术，将长的时间序列划分为多个数据段，每个数据段用直线拟合来反映时间序列的定性变化趋势，进而通过假设检验判断数据段均值和报警阈值的大小关系，最终确定正常和异常数据段。

基于上述思想，该方法主要分为两个步骤。第一步是获取 $x(t)$ 的线性分段表示，假设将 $\{x(t)\}_{t=1}^T$ 划分为 K 个数据段 $\{x(t)\}_{t=t_1}^{t_2-1}, \{x(t)\}_{t=t_2}^{t_3-1}, \dots, \{x(t)\}_{t=t_K}^T$ ，其中 t_k 为第 k 个数据段初始值的时刻 ($k \in [1, K]$, $t_1 = 1$)。第 k 个数据段 $\{x(t)\}_{t=t_k}^{t_{k+1}-1}$ 可以用一个线性回归模型来拟合：

$$x(t) = a_k + b_k t + e(t), \quad (3.3)$$

其中 a_k 和 b_k 分别为线性回归模型中的截距和斜率参数。根据中心极限定理，可以假设误差 $e(t)$ 为零均值、方差 σ_e^2 的高斯白噪声。采用最小二乘法估计参数 a_k 和 b_k ：

$$\hat{a}_k = \frac{\sum_{t=t_k}^{t_{k+1}-1} x(t) \sum_{t=t_k}^{t_{k+1}-1} t^2 - \sum_{t=t_k}^{t_{k+1}-1} x(t) t \sum_{t=t_k}^{t_{k+1}-1} t}{N_k \sum_{t=t_k}^{t_{k+1}-1} t^2 - (\sum_{t=t_k}^{t_{k+1}-1} t)^2},$$

$$\hat{b}_k = \frac{N_k \sum_{t=t_k}^{t_{k+1}-1} x(t) t - \sum_{t=t_k}^{t_{k+1}-1} x(t) \sum_{t=t_k}^{t_{k+1}-1} t}{N_k \sum_{t=t_k}^{t_{k+1}-1} t^2 - (\sum_{t=t_k}^{t_{k+1}-1} t)^2},$$

其中

$$N_k = t_{k+1} - t_k.$$

拟合误差为

$$\eta_k = \sum_{t=1}^T (x(t) - \hat{x}(t))^2,$$

其中

$$\hat{x}(t) = \hat{a}_k + \hat{b}_k t.$$

当给定分段数目 K 时，可采用自底向上的线性分段表示方法确定每段的初始时刻 t_k ， $k =$

$1, 2, \dots, K^{[2]}$ 。自底向上的线性分段表示方法主要包括如下三个步骤：1) 对整体序列 $\{x(t)\}_{t=1}^N$ 而言，该方法的初始状态为每隔两个点为一条拟合直线，即 $\{x(1), x(2)\}, \{x(3), x(4)\}, \dots, \{x(N-1), x(N)\}$ ；2) 对每一个数据段而言，合并其相邻的数据段，计算相应的拟合误差，然后只保留合并后拟合误差最小的数据段，此时总体数据段的数量减少1个；3) 不断重复第2个步骤直到数据段的数量等于 K 。本节采用文献 [3] 的方法来确定合适的分段数目 K ，此处不再复述。

第二步是采用均值假设检验，将第 k 个数据段 $\{x(t)\}_{t=t_k}^{t_{k+1}-1}$ 划分为正常或异常数据段。第 k 个数据段 $\{x(t)\}_{t=t_k}^{t_{k+1}-1}$ 的均值为

$$\mu_k = \frac{\sum_{t=t_k}^{t_{k+1}-1} x(t)}{t_{k+1} - t_k}. \quad (3.4)$$

考虑到 μ_k 的不确定性，本节采用下述假设检验：

$$H_0 : \mu_k = x_{tp}, \quad H_1 : \mu_k \neq x_{tp}. \quad (3.5)$$

根据中心极限定理， μ_k 为高斯分布，所以 μ_k 的95%置信区间为

$$[\mu_{k,-}, \mu_{k,+}] = [\mu_k - 1.96\sigma_k, \mu_k + 1.96\sigma_k], \quad (3.6)$$

其中

$$\sigma_k = \sqrt{\frac{\sum_{t=t_k}^{t_{k+1}-1} (x(t) - \mu_k)^2}{t_{k+1} - t_k - 1}}. \quad (3.7)$$

如果高报警阈值 x_{tp} 大于（小于） $\mu_{k,+}$ （ $\mu_{k,-}$ ），则拒绝零假设，此时由于均值 μ_k 统计意义上小于（大于） x_{tp} ，该段数据被划分为正常（异常）数据段。在其它情况下，接受零假设，即 μ_k 在统计意义上等于 x_{tp} ，此时无法判断该数据段是正常还是异常数据段。定义一个指示变量 $I(k)$ ，采用‘1’，‘0’和‘-1’分别表示正常、未定义和异常数据段：

$$I(k) = \begin{cases} 1, & \mu_{k,+} < x_{tp} \\ 0, & \text{其它} \\ -1, & \mu_{k,-} > x_{tp} \end{cases}. \quad (3.8)$$

这里给出了一个数值仿真案例来说明该方法的具体步骤及有效性。过程变量 $x(t)$ 由如下方式产生

$$x(t) = \begin{cases} 0.3 + e_1(t), & 1 \leq t \leq 500 \\ 1.3 + e_1(t), & 500 < t \leq 1000 \\ 0.3 + e_1(t), & 1000 < t \leq 1500 \\ 0.002t - 3 + e_2(t), & 1500 < t \leq 2000 \\ -0.002t + 5 + e_2(t), & 2000 < t \leq 2500 \\ 0.3 + e_2(t), & 2500 < t \leq 3000 \end{cases},$$

其中 $e_1(t)$ 为零均值方差为 $\sigma_{e_1}^2 = 0.25$ 的高斯白噪声， $e_2(t)$ 为有色噪声 $e_2(t) = 0.4e_1(t) + 0.6e_1(t-1)$ 。如图3.3(a)所示， $x(t)$ 的高报警阈值为 $x_{tp,1} = 0.8$ ，显然 $x_1(1 : 500)$ 和 $x_1(1001 : 3000)$ 为正常数据段，而 $x_1(501 : 1000)$ 为异常数据段。

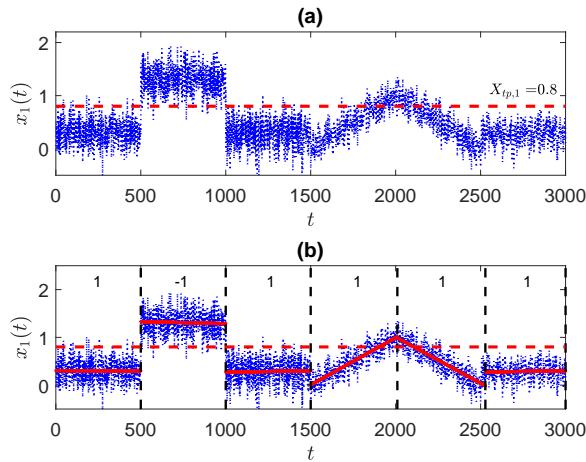


图 3.3 基于线性分段表示的正常和异常数据段划分结果

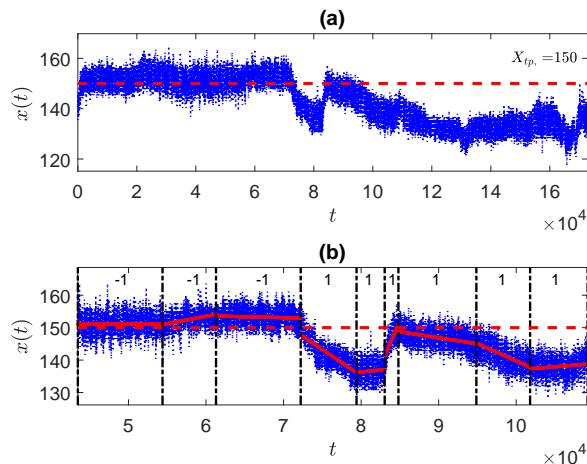


图 3.4 工业案例的正常和异常数据段划分结果

采用本节的方法，首先对 $x(t)$ 进行线性分段表示，图3.3(b)展示了线性分段表示，此处的六条拟合直线可以准确反映 $x(t)$ 在不同数据段的变化趋势。然后，采用假设检验式(3.5)来判断正常和异常数据段，图3.3(b)展示了指示序列 $I(k)$ 。上述结果表明该方法可以有效的划分正常和异常数据段。

最后，通过一个工业案例验证本节提出方法的效果。该案例来源于山东省某台300MW燃煤发电机组。过程变量 $x(t)$ 为反应区催化剂上层与中层前压力差，其变量名为X71NAO067，对应的单位为 Nm^3/h （标立方米每小时），配置有高报警阈值 $x_{tp} = 150 \text{ Nm}^3/\text{h}$ 。图3.4为变量 $x(t)$ 及其正常和异常数据段划分结果。图3.4(a)展示了48小时的时间序列，图3.4(b)展示了数据段划分结果和正常/异常数据段标记，显然划分结果和目测观察结果是一致的。

3.3 报警延迟器设计

本节首先介绍了单变量报警器的误报率、漏报率、平均报警延时等性能指标的定义，然后给出了越限报警器与延时报警器的误报率、漏报率与平均报警延时的计算方法，最后设计了达到最优性能指标的报警延迟参数和报警阈值。

3.3.1 性能指标的定义

考虑一个基于过程变量 x 的单变量报警器，如图3.5所示， $x(t)$ 的采样周期为 h ，报警阈值为 x_{tp} 。由于噪声干扰等因素，两类错误报警可能会出现，即误报警和漏报警。误报警是在过程变量 x 处于正常运行状况下发生的报警，而漏报警是在过程变量 x 处于异常运行状况下却没有发生的报警。通常情况下，过多的误报警会使得操作人员对报警系统产生不信任，而过多的漏报警则意味着报警系统的性能大打折扣，因此误报率和漏报率是评估报警系统性能的两个重要指标。除了误报率和漏报率两个指标，平均报警延时也是一个重要的性能指标。对于图3.5所示的过程变量 x ，假设它在 t_0 时刻从正常运行状况跳变到异常运行状况，而在 t_a 时刻， $x(t)$ 的样本值第一次越过报警阈值 x_{tp} 以触发报警。定义 t_0 与 t_a 的差为报警延时 T_d ，即

$$T_d = t_a - t_0.$$

显然， T_d 是一个离散随机变量，它的样本空间是 $\{0h, 1h, 2h, \dots\}$ 。为了使得报警延时的定义具有物理意义，通常不考虑 t_0 时刻之前的误报警，即 $t_a \geq t_0$ ，所以 T_d 为非负值。平均报警延时定义为 T_d 的均值，即

$$\bar{T}_d = E(T_d). \quad (3.9)$$

平均报警延时反映了报警系统的快速性：当 $x(t)$ 进入异常运行状况时，平均报警延时越小，报警产生的速度越快，从而给工业运行人员更多的时间去处理报警。

3.3.2 越限报警器的性能指标

本节给出越限报警器的误报率、漏报率、平均报警延时这三个性能指标的计算方法。

假设过程变量 $x(t)$ 配置有高报警阈值 x_{tp} ，采用越限报警器(3.1)来生成报警变量 $x_a(t)$ 。由

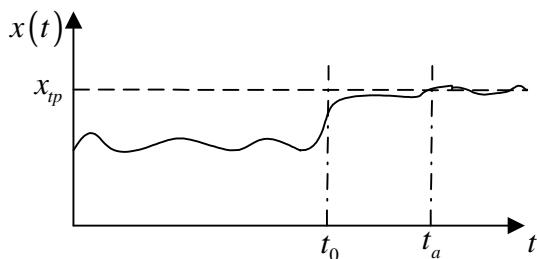
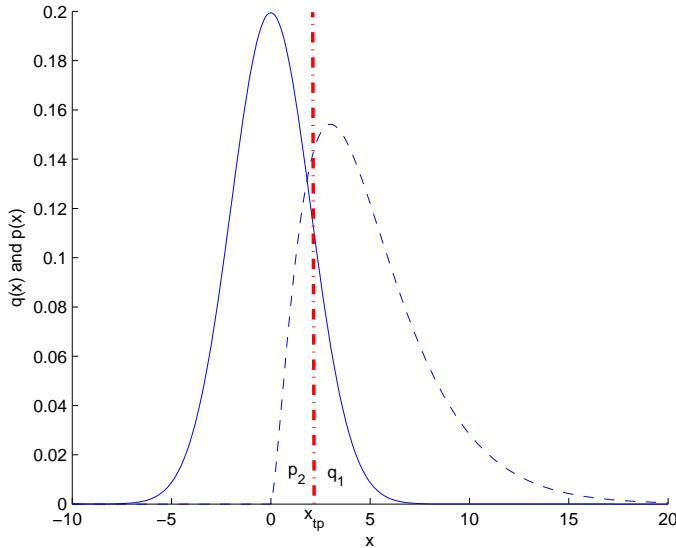


图 3.5 过程变量 x 与报警阈值 x_{tp}

图 3.6 正常和异常运行状况下 x 的概率密度函数

于噪声是引发误报警和漏报警的主要因素之一，因此当 $x(t)$ 处于稳态时，可假设 $x(t)$ 是独立同分布的（IID: independent and identical distributed）。给定处于正常和异常运行状况的概率密度函数（概率密度函数的估计方法将在第3.3.4 中详述），如图3.6所示。处于正常运行状况的概率密度函数记为 $q(x)$ （如图3.6中实线所示），则误报率为分布曲线 $q(x)$ 下 x 大于报警阈值 x_{tp} 的区域面积，即

$$\text{FAR} = \int_{x_{tp}}^{+\infty} q(x)dx. \quad (3.10)$$

类似地，漏报率为

$$\text{MAR} = \int_{-\infty}^{x_{tp}} p(x)dx, \quad (3.11)$$

其中 $p(x)$ 是过程变量 x 处于异常运行状况时的概率密度函数（如图3.6 中虚线所示）。在图3.6中，误报率和漏报率分别记为 q_1 和 p_2 。

一般地，平均报警延时 T_d 的计算需要知道 $x(t_0), x(t_0 + h), \dots, x(t_0 + ih)$ 的多维联合概率密度函数。当 $t \geq t_0$ 时（即过程变量 x 处于异常运行状况）， x 的概率密度函数是 $p(x)$ （图3.6中的虚线所示），如(3.11) 中定义， p_2 是概率密度函数 $p(x)$ 下 x 小于报警阈值 x_{tp} 的区域面积。由于 $x(t)$ 是独立同分布的， $T_d = ih$ 的概率分布函数为

$$P(T_d = ih) = P(x(t_0) < x_{tp}, \dots, x(t_0 + ih - h) < x_{tp}, x(t_0 + ih) > x_{tp}) = p_2^i p_1.$$

这里 $p_1 := 1 - p_2$ 。显然， T_d 服从几何分布，利用几何分布的均值表达式^[1]，平均报警延时为

$$\bar{T}_d = E(T_d) = \sum_{i=0}^{\infty} ih p_2^i p_1 = h \frac{p_2}{p_1}. \quad (3.12)$$

下面通过仿真算例来验证误报率的计算公式(3.10)、漏报率的计算公式(3.11)和平均报警延时的计算公式(3.12)的正确性。采用一个高斯白噪声随机过程来产生过程变量 x 的数据，其中在 t_0 时刻 x 从正常运行状况跳变到异常运行状况：

$$\begin{cases} x(t) \sim N(3, 1), & t < t_0 \\ x(t) \sim N(5, 1), & t \geq t_0 \end{cases}. \quad (3.13)$$

这里的跳变时间 $t_0 = 1000h$ ，采样周期 $h = 1$ 秒，数据长度为2000，报警阈值 $x_{tp} = 4$ 。根据式(3.10)、(3.11)、(3.12)可以求得误报率、漏报率和平均报警延时的理论值分别为

$$FAR = 0.1587, MAR = 0.1587, \bar{T}_d = 0.1886.$$

为了验证理论值的正确性，通过仿真产生500组独立的时间序列 $\{x(t)\}_{t=1}^{2000}$ ，针对每一组时间序列，将 $x(t)$ 在正常运行状况下的报警个数与数据点数之比作为这一组时间序列的误报率估计值。同理，针对每一组时间序列，利用 $x(t)$ 在异常运行状况下观察到的非报警个数与数据点数之比作为这一组时间序列的漏报率估计值。基于这500组独立的时间序列，可以得到误报率估计值和漏报率估计值的样本均值与样本标准差，

$$m(\hat{FAR}) = 0.1589, s(FAR) = 0.0115,$$

$$m(\hat{MAR}) = 0.1589, s(MAR) = 0.0118.$$

这些结果与误报率和漏报率的理论值是一致的。

对于每一次仿真实现，都可以得到跳变时间 t_0 及其之后的第一次报警产生时间 t_a ，所以可以计算得到相应的报警延时。通过仿真产生的500次独立实现，可以得到500次报警延时的实现，然后计算这500次报警延时的样本均值作为平均报警延时的估计值，记为 $\hat{\bar{T}}_d$ 。进行500组这样的500次独立实现，就可以得到500个估计值 $\hat{\bar{T}}_d$ ，它们的样本均值和样本标准差为

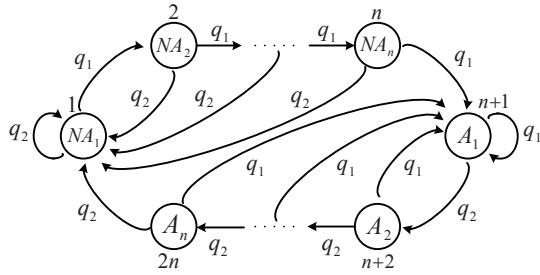
$$m(\hat{\bar{T}}_d) = 0.1875, s(\hat{\bar{T}}_d) = 0.0206.$$

这个结果与平均报警延时的理论值也是一致的。

3.3.3 报警延迟器的性能指标

报警延迟器是一种在工业实践中被广泛使用的单变量报警器^[12]。本节提出了报警延迟器的误报率、漏报率、平均报警延时这三个性能指标的计算方法^[5]。以配置有高报警阈值 x_{tp} 的过程变量 $x(t)$ 为例，报警延迟器是指，当且仅当 $x(t)$ 的连续 n 个样本值超过或低于 x_{tp} ，才产生报警或消除报警。这里 n 称为报警延迟参数。显然，越限报警器可以视为报警延迟器的特例，即报警延迟参数 $n = 1$ 。

报警延迟参数为 n 的报警延迟器包含 n 个不同的报警状态和 n 个不同的非报警状态，可以用马尔科夫链来描述^[13]。在正常运行状况下，过程变量 x 在这些状态下的跳变过程如图3.7所示。当过程变量 x 处于正常运行状况时，它的概率密度函数为 $q(x)$ （图3.6中实线所示），其定义见式(3.10)，且 q_1 是概率密度函数 $q(x)$ 下 x 大于报警阈值 x_{tp} 的概率，并简记 $q_2 := 1 - q_1$ 。如果报警延迟器下的某个采样值 $x(t_1)$ 处于第 i 个非报警状态 NA_i ，其中 $i = 1, 2, \dots, n$ ，并且下一个采样值 $x(t_1 + h)$ （ h 为采样周期）大于报警阈值 x_{tp} ，那么当 $i < n$ 时，就从当前状态 NA_i 跳变

图 3.7 $x(t)$ 处于正常状态下的马尔科夫链

到第 $i + 1$ 个非报警状态 NA_{i+1} ，当 $i = n$ 时，则跳变到报警状态 A_1 ；如果 $x(t_1 + h)$ 小于报警阈值 x_{tp} ，则从当前状态 NA_i 直接跳回第一个非报警状态 NA_1 。同理，如果报警延迟器下的某次样本值 $x(t_2)$ 处于第 i 个报警状态 A_i ，其中 $i = 1, 2, \dots, n$ ，并且下一个样本值 $x(t_2 + h)$ 小于报警阈值 x_{tp} ，那么当 $i < n$ 时，就从当前状态 A_i 跳变到第 $i + 1$ 个报警状态 A_{i+1} ，当 $i = n$ 时，则跳变到非报警状态 NA_1 ；如果 $x(t_2 + h)$ 大于报警阈值 x_{tp} ，则从当前状态 A_i 直接跳回第一个报警状态 A_1 。

命题 3.1 参数为 n 的报警延迟器对应的误报率为

$$\text{FAR} = \frac{q_1^n(1 + q_2 + \dots + q_2^{n-1})}{q_1^n(1 + q_2 + \dots + q_2^{n-1}) + q_2^n(1 + q_1 + \dots + q_1^{n-1})}. \quad (3.14)$$

证明 首先简记状态 $NA_1, NA_2, \dots, NA_n, A_1, \dots, A_n$ 为 $1, 2, \dots, n-1, n, \dots, 2n$ 。记 $T_{i,k}$ 为从状态 i 跳变到状态 k 所需要的步数， $T_{i,k} = l$ 的概率记为

$$P_{i,k}^{(l)} := P(T_{i,k} = l).$$

当 $l = 1$ 时， $P_{i,k}^{(1)}$ 为一步转移概率。对于图3.7所示的马尔科夫链，它的一步转移概率矩阵 $Q \in \mathbb{R}^{2n \times 2n}$ 为

$$Q = \begin{bmatrix} q_2 & q_1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ q_2 & 0 & q_1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ q_2 & 0 & \cdots & 0 & q_1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & q_1 & q_2 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & q_1 & 0 & q_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & q_1 & 0 & 0 & \cdots & q_2 \\ q_2 & 0 & 0 & \cdots & q_1 & 0 & 0 & \cdots & 0 \end{bmatrix},$$

其中，位于矩阵 Q 第 i 行第 j 列的元素是从状态 i 跳变到状态 j 的一步转移概率。根据平稳分布理论^[14]，对于一个不可约遍历(irreducible ergodic)马尔科夫链，其极限概率

$$\pi_k = \lim_{l \rightarrow \infty} P_{ik}^{(l)} > 0$$

存在，并且不依赖于初始状态，它满足

$$\sum_k \pi_k = 1. \quad (3.15)$$

因此此马尔科夫链只有有限个状态，所以极限概率满足如下方程

$$\Pi = \Pi Q, \quad (3.16)$$

其中

$$\Pi = [\pi_1 \ \pi_2 \ \cdots \ \pi_{2n}].$$

则方程(3.16)可以写成

$$\left\{ \begin{array}{rcl} q_2(\pi_1 + \cdots + \pi_n) + q_2\pi_{2n} & = & \pi_1 \\ q_1\pi_1 & = & \pi_2 \\ q_1\pi_2 & = & \pi_3 \\ \vdots & & \\ q_1\pi_{n-1} & = & \pi_n \\ q_1(\pi_n + \cdots + \pi_{2n}) & = & \pi_{n+1} \\ q_2\pi_{n+1} & = & \pi_{n+2} \\ \vdots & & \\ q_2\pi_{2n-1} & = & \pi_{2n} \end{array} \right. . \quad (3.17)$$

解方程(3.17)可得

$$\left\{ \begin{array}{rcl} \pi_2 & = & q_1\pi_1 \\ \vdots & & \\ \pi_n & = & q_1^{n-1}\pi_1 \end{array} \right. . \quad (3.18)$$

和

$$\left\{ \begin{array}{rcl} \pi_{n+2} & = & q_2\pi_{n+1} \\ \vdots & & \\ \pi_{2n} & = & q_2^{n-1}\pi_{n+1} \end{array} \right. . \quad (3.19)$$

利用方程(3.18)、(3.19)、(3.15)和(3.17)中的第一个等式可得

$$\left\{ \begin{array}{rcl} \pi_1(1 + q_1 + \cdots + q_1^{n-1}) + \pi_{n+1}(1 + q_2 + \cdots + q_2^{n-1}) & = & 1 \\ q_2\pi_1(1 + q_1 + \cdots + q_1^{n-1}) + q_2^n\pi_{n+1} & = & \pi_1 \end{array} \right. .$$

解这个方程可得

$$\pi_{n+1} = \frac{q_1^n}{q_1^n(1 + q_2 + \cdots + q_2^{n-1}) + q_2^n(1 + q_1 + \cdots + q_1^{n-1})}. \quad (3.20)$$

由图3.7可知，误报率是所有报警状态概率的总和

$$\text{FAR} = P(A_1) + P(A_2) + \cdots + P(A_n)$$

$$\begin{aligned}
&= \pi_{n+1} + \cdots + \pi_{2n} \\
&= \pi_{n+1}(1 + q_2 + \cdots + q_2^{n-1}).
\end{aligned}$$

其中最后一个等式由方程(3.19)可得。这个结果结合方程(3.20)可证得(3.14)。

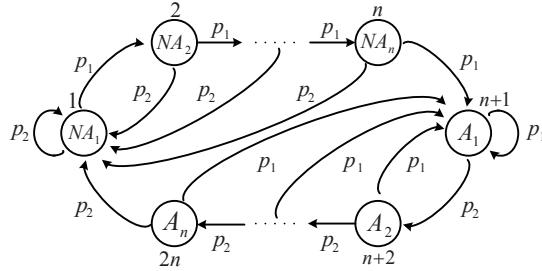


图 3.8 报警延迟参数为 n 的报警延迟器下 $x(t)$ 处于异常状态下的马尔科夫链

当过程变量 $x(t)$ 处于异常运行状况时，报警延迟器对应的状态转移过程可用图3.8 中的马尔科夫链描述，其中 p_2 是概率密度函数 $p(x)$ 下 x 小于报警阈值 x_{tp} 的概率（如图3.6所示），并简记 $p_1 := 1 - p_2$ 。漏报率是所有非报警状态概率的总和，即

$$\text{MAR} = P(NA_1) + P(NA_2) + \cdots + P(NA_n).$$

命题 3.2 参数为 n 的报警延迟器对应的漏报率为

$$\text{MAR} = \frac{p_2^n(1 + p_1 + \cdots + p_1^{n-1})}{p_2^n(1 + p_1 + \cdots + p_1^{n-1}) + p_1^n(1 + p_2 + \cdots + p_2^{n-1})}. \quad (3.21)$$

证明 对于图3.8所示的马尔科夫链，一步转移概率矩阵 $P \in \mathbb{R}^{2n \times 2n}$ 为

$$P = \begin{bmatrix} p_2 & p_1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ p_2 & 0 & p_1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_2 & 0 & \cdots & 0 & p_1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & p_1 & p_2 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & p_1 & 0 & p_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & p_1 & 0 & 0 & \cdots & p_2 \\ p_2 & 0 & 0 & \cdots & p_1 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

从而式(3.21)可由类似于命题3.1的证明过程来获得。

平均报警延时的定义如式(3.9)所示， t_a 是在报警延迟参数为 n 的报警延迟器下第一次发生报警的时刻。这里假设在 $t_0 - h$ 时刻没有触发报警是合理的，并且一旦发生报警，不考虑消除

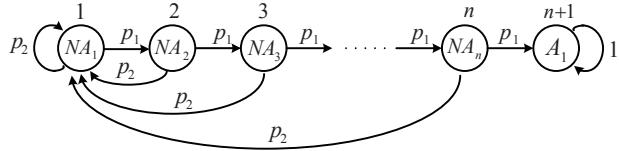


图 3.9 计算平均报警延时的马尔科夫链

报警的过程。因此，记 $t_0 - h$ 时刻为第一个非报警状态 NA_1 ，其跳变到报警状态 A_1 的过程与图3.8所示过程相同。因为只考虑第一个报警（当状态第一次到达报警状态 A_1 ），所以只需考虑图3.8 中上半部分的马尔科夫链。为了计算平均报警延时，规定一旦状态到达报警状态 A_1 ，就以概率1停留在这个状态，相应的状态转移机制如图3.9所示。值得注意的是， $t_0 - h$ 时刻的状态不一定是 NA_1 ，它可以是任何非报警状态 NA_i 之一，其中 $i = 1, \dots, n$ 。这里只考虑状态 NA_1 的原因是：

1) 从命题3.1的证明过程可得处于状态 NA_i 的概率为

$$P(NA_i) = \frac{q_1^{i-1} q_2^n}{q_1^n (1 + \dots + q_2^{n-1}) + q_2^n (1 + \dots + q_1^{n-1})},$$

其中 q_1 是 x 在正常状态下，概率密度函数 $q(x)$ 下 x 大于报警阈值 x_{tp} 的概率。因为通常情况下 q_1 很小，所以马尔科夫链从状态 NA_1 出发的概率比从其余状态出发的概率总和更大。

2) 在实践中， x 处于正常和异常运行状况时的概率密度函数是未知的，必须通过估计算法来获得。如果 $t_0 - h$ 时刻的状态时 NA_i , $i > 1$ 。即所有的采样值 $x(t_0 - ih), \dots, x(t_0 - h)$ 大于报警阈值 x_{tp} ，而采样值 $x(t_0 - ih - h)$ 小于报警阈值 x_{tp} ，当估计 x 处于正常和异常运行状况的概率密度函数时， $x(t_0 - ih), \dots, x(t_0 - h)$ 将被归类到异常数据段，因此检测出的从正常状态跳变到异常状态的变化点的前一个数据点的值将小于报警阈值 x_{tp} ，相应地，它的状态为 NA_1 。

由于上述两个原因，我们忽略 $t_0 - h$ 时刻处于其它非报警状态 NA_i , $i > 1$ 的情况。

命题 3.3 如果 $t_0 - h$ 时刻的状态为 NA_1 ，在报警延迟参数为 n 的报警延迟器下，平均报警延时为

$$\overline{T_d} = h \frac{(1 - p_1^n - p_2 p_1^n)}{p_2 p_1^n}. \quad (3.22)$$

证明 简记状态 $NA_1, NA_2, \dots, NA_n, A_1$ 分别为 $1, 2, \dots, n+1$ 。根据式(3.9)，在报警延迟参数为 n 的报警延迟器下，平均报警延时为

$$\overline{T_d} = E(T_d) = E(h T_{1,n+1}) - h, \quad (3.23)$$

其中 $T_{1,n+1}$ 是从状态 1 转移到报警状态 $n+1$ 所需要的步数。由于状态 1 发生在 $t_0 - h$ 时刻，所以在式(3.23)中减去一个采样周期 h 。 $E(T_{1,n+1})$ 可以通过如下步骤得到。首先，离散随机变量 $T_{i,n+1}$ ($i \in \{1, 2, \dots, n+1\}$) 的矩函数为

$$\Gamma_{i,n+1}(z) = \sum_{l=0}^{\infty} P_{i,n+1}^{(l)} z^l, \quad (3.24)$$

其中 $P_{i,n+1}^{(l)} := P(T_{i,n+1} = l)$, $z := e^t$ 。 $\Gamma_{i,n+1}(z)$ 对 z 求导得

$$\frac{d}{dz}\Gamma_{i,n+1}(z) = \sum_{l=0}^{\infty} lP_{i,n+1}^{(l)}z^{l-1},$$

从而从状态 i 到 $n+1$ 的平均转移步数为

$$E(T_{i,n+1}) = \sum_{l=0}^{\infty} lP_{i,n+1}^{(l)} = \left. \frac{d}{dz}\Gamma_{i,n+1}(z) \right|_{z=1}.$$

因此, 式(3.23)中 $E(T_{1,n+1})$ 的计算公式为

$$E(T_{1,n+1}) = \left. \frac{d}{dz}\Gamma_{1,n+1}(z) \right|_{z=1}. \quad (3.25)$$

利用 Chapman-Kolmogorov 方程^[14], 即

$$P_{i,k}^{(l)} = \sum_{j \in I} P_{i,j}^{(1)} P_{j,k}^{(l-1)},$$

及马尔科夫链理论的基本定义

$$\begin{cases} P_{i,k}^{(0)} &= 0, \quad i \neq k \\ P_{i,k}^{(0)} &= 1, \quad i = k \end{cases},$$

当 $i \neq n+1$ 时, 式(3.24)可写为

$$\begin{aligned} \Gamma_{i,n+1}(z) &= \sum_{l=0}^{\infty} P_{i,n+1}^{(l)} z^l \\ &= P_{i,n+1}^{(0)} z^0 + \sum_{l=1}^{\infty} \sum_{j \in I} P_{i,j}^{(1)} P_{j,n+1}^{(l-1)} z^l \\ &= \left(\sum_{j \in I} z P_{i,j}^{(1)} \right) \left(\sum_{l=1}^{\infty} P_{j,n+1}^{(l-1)} z^{l-1} \right) \\ &= \sum_{j \in I} z P_{i,j}^{(1)} \Gamma_{j,n+1}(z), \end{aligned} \quad (3.26)$$

其中 I 是整个状态空间 $\{1, 2, \dots, n+1\}$ 。因为状态 $n+1$ 是常返态, 根据马尔科夫链理论对于常返态 k 的两个基本定义

$$\begin{cases} P_{k,k}^{(l)} &= 0, \quad l > 0 \\ P_{k,k}^{(l)} &= 1, \quad l = 0 \end{cases}.$$

当 $i = n+1$ 时, 式(3.24)可写为

$$\Gamma_{n+1,n+1}(z) = \sum_{l=0}^{\infty} P_{n+1,n+1}^{(l)} z^l = 1.$$

结合方程(3.26)和图3.9所示的马尔科夫链可得

$$\left\{ \begin{array}{lcl} \Gamma_{1,n+1}(z) & = & zp_2\Gamma_{1,n+1}(z) + zp_1\Gamma_{2,n+1}(z) \\ \Gamma_{2,n+1}(z) & = & zp_2\Gamma_{1,n+1}(z) + zp_1\Gamma_{3,n+1}(z) \\ \vdots & & \\ \Gamma_{n,n+1}(z) & = & zp_2\Gamma_{1,n+1}(z) + zp_1\Gamma_{n+1,n+1}(z) \end{array} \right. . \quad (3.27)$$

因为 $\Gamma_{n+1,n+1}(z) = 1$, 解方程(3.27)可得 $\Gamma_{1,n+1}(z)$ 的表达式为

$$\Gamma_{1,n+1}(z) = \frac{z^n p_1^n}{1 - zp_2 - z^2 p_2 p_1 - \cdots - z^n p_2 p_1^{n-1}}.$$

因此, 式(3.25)可写为

$$E(T_{1,n+1}) = \frac{d}{dz}\Gamma_{1,n+1}(z)\Big|_{z=1} = \frac{(1-p_1^n)}{p_2 p_1^n},$$

进而式(3.23)的平均报警延时 (AAD) 为

$$\bar{T}_d = hE(T_{1,n+1}) - h = h \frac{(1-p_1^n - p_2 p_1^n)}{p_2 p_1^n}.$$

这里通过仿真算例来验证报警延迟器的误报率、漏报率和平均报警延时的计算公式(3.14)、(3.21)和(3.22)。假设报警延迟参数为 $n = 3$, 其它仿真条件均与3.3.2节中的例子相同。由式(3.14), (3.21)和(3.22), 可以得到误报率、漏报率和平均报警延时的理论值分别为

$$\text{FAR} = 0.0142, \text{ MAR} = 0.0142, \bar{T}_d = 3.2804.$$

为了验证理论值的正确性, 开展类似3.3.2节的蒙特卡洛仿真实验, 可得

$$m(\hat{\text{FAR}}) = 0.0144, s(\hat{\text{FAR}}) = 0.0082$$

$$m(\hat{\text{MAR}}) = 0.0143, s(\hat{\text{MAR}}) = 0.0085$$

$$m(\hat{\bar{T}}_d) = 3.2805, s(\hat{\bar{T}}_d) = 0.0945,$$

可见仿真结果与理论结果是一致的。

3.3.4 报警延迟器的优化设计

由于越限报警器等价于报警延迟参数 $n = 1$ 的报警延迟器, 本节只讨论报警延迟器的优化设计问题。问题描述如下: 假设已知 $x(t)$ 在正常和异常运行状况下的概率密度函数, 选择合适的报警阈值 x_{tp} 或报警延迟参数 n , 使得报警延迟器满足对误报率、漏报率和平均报警延时等性能指标的要求。以下将讨论三种不同情况的设计问题:

- 1) 当报警延迟参数 n 固定时, 设计报警阈值 x_{tp} ;
- 2) 当报警阈值 x_{tp} 固定时, 设计报警延迟参数 n ;
- 3) 同时设计报警延迟参数 n 和报警阈值 x_{tp} 。

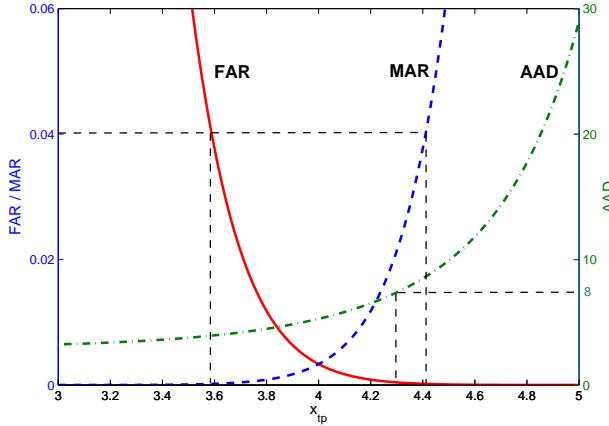


图 3.10 误报率、漏报率、平均报警延时与报警阈值 x_{tp} 的关系.

情况1：当报警延迟参数 n 固定时，设计报警阈值 x_{tp}

对于固定的报警延迟参数 n ，可以通过误报率和漏报率的权衡、误报率和平均报警延时的权衡来设计报警阈值 x_{tp} 。下面通过一个例子来说明设计原理。

过程变量 x 的产生方式和3.3.2节的仿真例子一样。这里的设计目标是：在报警延迟参数为 $n = 4$ 的报警延迟器下，设计报警阈值 x_{tp} 使得报警延迟器满足性能要求 $\text{FAR} \leq 4\%$, $\text{MAR} \leq 4\%$, $\text{AAD} \leq 8h$ ，其中 $h = 1$ 秒。

基于式(3.13)的概率密度函数，可得 $q_1 = 0.1587$, $q_2 = 1 - q_1$, $p_2 = 0.1587$, $p_1 = 1 - p_2$ 。然后，由式(3.14)、(3.21)和(3.22)可得，在报警延时参数 $n = 4$ 的报警延迟器下，误报率、漏报率、平均报警延时三个性能指标与报警阈值 x_{tp} 的关系，如图3.10所示。显然，图3.10存在误报率和漏报率的权衡、误报率和平均报警延时的权衡这两个权衡关系，即报警阈值 x_{tp} 越大，误报率越小，而漏报率和平均报警延时越大，反之亦然。因此，基于误报率、漏报率及平均报警延时三个性能指标要求，可以分别得到满足各自要求的报警阈值 x_{tp} 有效区间。这些有效区间的交集是同时满足三个性能指标要求的报警阈值 x_{tp} 的最终选择。如果交集为空，则说明通过单独调整报警阈值 x_{tp} 的方式无法同时满足三个性能指标要求。对于这个例子，基于图3.10所示的三条曲线，可得满足误报率、漏报率及平均报警延时三个性能指标要求的有效区间分别为 $x_{tp} \geq 3.59$, $x_{tp} \leq 4.11$ 及 $x_{tp} \leq 4.35$ ，故选择有效区间的交集 $x_{tp} \in [3.59, 4.35]$ 作为最终报警阈值的选择范围。

情况2：当报警阈值 x_{tp} 固定时，设计报警延迟参数 n

对于报警延迟器，报警延迟参数 n 是另外一个设计参数。对于固定的报警阈值 x_{tp} ，可以基于误报率和平均报警延时的权衡关系，以及漏报率和平均报警延时的权衡关系，来设计报警延迟参数 n 。我们通过下面的例子来说明设计原理。

过程变量 x 的产生方式和3.3.2节的仿真例子相同。这里的设计目标是：在报警阈值 $x_{tp} = 4.0$ 的报警延迟器下，设计报警延迟参数 n 使得报警延迟器满足性能要求 $\text{FAR} \leq 1\%$, $\text{MAR} \leq 1\%$, $\text{AAD} \leq 8h$ ，其中 $h = 1$ 秒。

与第一个例子类似，由式(3.14)、(3.21)和(3.22)可得，在报警阈值 $x_{tp} = 4.0$ 的报警延迟器

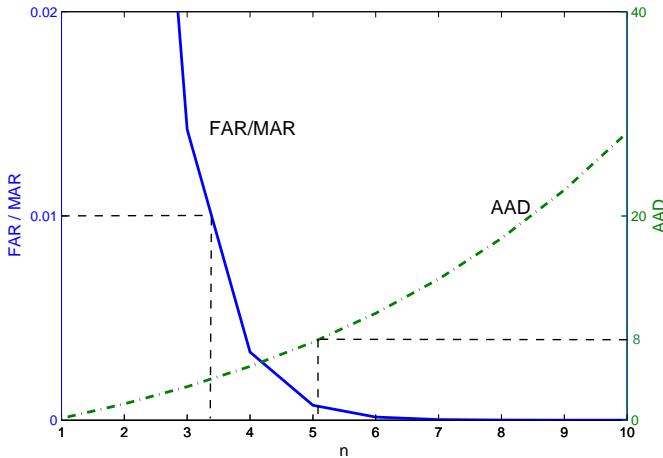


图 3.11 误报率、漏报率、平均报警延时与报警延迟参数 n 的关系.

下，误报率、漏报率、平均报警延时三个性能指标与报警延迟参数 n 的关系，如图3.11所示。显然，图3.11存在误报率和平均报警延时的权衡、漏报率和平均报警延时的权衡这两个权衡关系。对于固定的报警阈值 x_{tp} ，报警延迟参数 n 越大，误报率和漏报率越小，而平均报警延时越大。因此，基于误报率、漏报率及平均报警延时三个性能指标要求，通过上述两个权衡关系，分别可以得到满足性能指标要求的报警延时参数 n 有效区间。这些有效区间的交集将是同时满足三个性能指标要求的报警延时参数 n 的最终选择。如果交集为空，则说明通过单独调整报警延迟参数 n 的方式无法同时满足三个性能指标要求。对于这个例子，基于图3.11 所示的三条曲线，可得满足误报率、漏报率及平均报警延时三个性能指标要求的报警延迟参数的最终选择是 $n = 4$ 或者 $n = 5$ 。

情况3：同时设计报警延迟参数 n 和报警阈值 x_{tp}

如果要同时设计报警延迟参数 n 和报警阈值 x_{tp} ，设计步骤要比上面两种情况更加复杂一些。基本的设计原理是：先基于性能指标要求设计报警延迟参数 n ，然后针对每个 n 分别设计对应的报警阈值 x_{tp} 。我们将通过下面的例子来阐述设计步骤。

过程变量 x 的产生方式和3.3.2节的仿真例子相同。这里的设计目标是：在报警延迟器下，设计报警阈值 x_{tp} 和报警延迟参数 n 使得报警延迟器满足性能要求 $\text{FAR} \leq 1\%$ ， $\text{MAR} \leq 1\%$ ， $\text{AAD} \leq 10h$ ，其中 $h = 1$ 秒。

设计步骤如下：首先，对不同的报警延迟参数 n ，由式(3.14) 和式(3.21)可得报警阈值 x_{tp} 与误报率、漏报率的关系曲线，如图3.12 所示。例如，曲线AB段可以满足误报率和漏报率的要求。从图3.12可以得出，只有当报警延迟参数 $n \geq 4$ 时，才能满足误报率和漏报率的要求。其次，对不同的报警延迟参数 n ，通过式(3.22)可得平均报警延时和报警阈值 x_{tp} 的关系曲线，如图3.13 所示。根据平均报警延时性能指标的要求，有效报警延迟参数的范围是 $n \in [1, 9]$ 。最终，报警延迟参数 n 应选择 $n \in [4, 9]$ 。对于 $n \in [4, 9]$ 中的每个值，与之对应能够满足误报率和漏报率指标要求的报警阈值 x_{tp} 可从图3.12中得到，结果见表3.3第二列。为了满足平均报警延时要求，有效报警阈值 x_{tp} 可从图3.13中得到，结果见表3.3第三列。对于 $n \in [4, 9]$ 中的每个 n 值，可将表3.3第二列和第三列所示区间的交集作为相应的报警阈值选择范围。

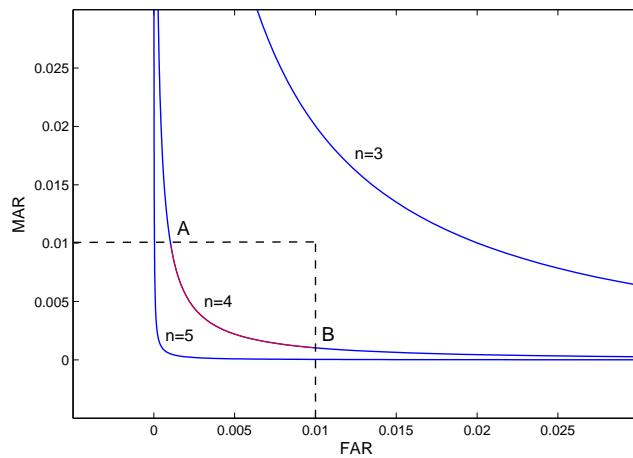
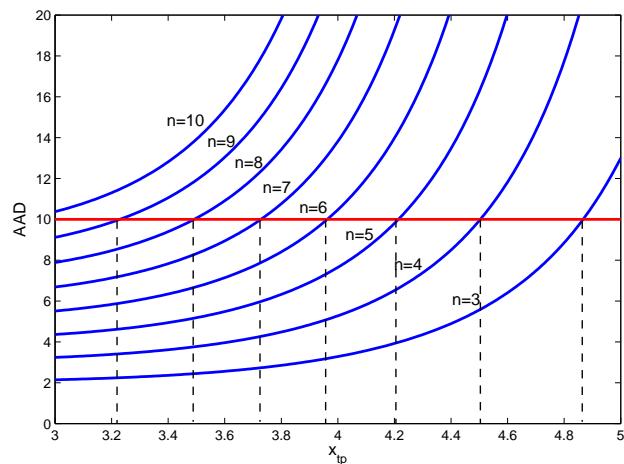


图 3.12 报警延迟器下误报率和漏报率的关系曲线.

图 3.13 报警延迟器下平均报警延时和报警阈值 x_{tp} 的关系曲线表 3.3 $n \in [4, 9]$ 对应的报警阈值 x_{tp} 的有效范围

n	$x_{tp}(\text{FAR}, \text{MAR})$	$x_{tp}(\text{AAD})$	x_{tp}
4	[3.83, 4.17]	[3, 4.50]	[3.83, 4.17]
5	[3.67, 4.33]	[3, 4.21]	[3.67, 4.21]
6	[3.56, 4.44]	[3, 3.96]	[3.56, 3.96]
7	[3.47, 4.53]	[3, 3.72]	[3.47, 3.72]
8	[3.41, 4.59]	[3, 3.49]	[3.41, 3.49]
9	[3.36, 4.64]	[3, 3.22]	\emptyset

表3.3给出了报警延迟参数 n 和报警阈值 x_{tp} 的有效范围，但最终只能选择一组 n 和 x_{tp} 的值，因此，需要基于某种优化准则来选出最优的一组值。定义加权和损失函数为

$$J(x_{tp}, n) = \omega_1 \frac{\text{FAR}}{\text{RFAR}} + \omega_2 \frac{\text{MAR}}{\text{RMAR}} + \omega_3 \frac{\text{AAD}}{\text{RAAD}}, \quad (3.29)$$

其中，RFAR、RMAR和RAAD分别表示FAR、MAR 和AAD三个性能指标的要求，即 $\text{FAR} \leq \text{RFAR}$ 、 $\text{MAR} \leq \text{RMAR}$ 和 $\text{AAD} \leq \text{RAAD}$ ， ω_1 、 ω_2 和 ω_3 分别是FAR， MAR和AAD的权重。最优的一组报警延迟参数 n 和报警阈值 x_{tp} 可以通过使得损失函数达到最小来得到，即

$$(x_{tp}, n) = \arg \min J(x_{tp}, n). \quad (3.30)$$

由式(3.14)、(3.21)和(3.22)可知，FAR、MAR 和AAD是关于 n 和 x_{tp} 的非线性函数，因此，通过二维网格搜索法求解式(3.30)可以得到最优的 n 和 x_{tp} 。由上述例子可知，RFAR = 1%， RMAR = 1%， RAAD = 10h。假设这三个性能指标具有同等的重要性，则 $\omega_1 = 1$ ， $\omega_2 = 1$ ， $\omega_3 = 1$ 。根据表3.3 中 n 和 x_{tp} 的有效范围，利用步长为0.01的网格搜索法，对于 n 的每个有效值，获得损失函数 $J(x_{tp}, n)$ 的最小值及其对应的报警阈值 x_{tp} 如表3.4所示，从中可知最优的报警延迟参数 n 和

表 3.4 报警延迟参数 n 、最优报警阈值 x_{tp} 和损失函数 $J(x_{tp}, n)$ 的最小值

n	x_{tp}	$J(x_{tp}, n)$ 的最小值
4	3.97	1.7063
5	3.88	1.5762
6	3.76	1.7764
7	3.65	2.0171
8	3.49	2.3708

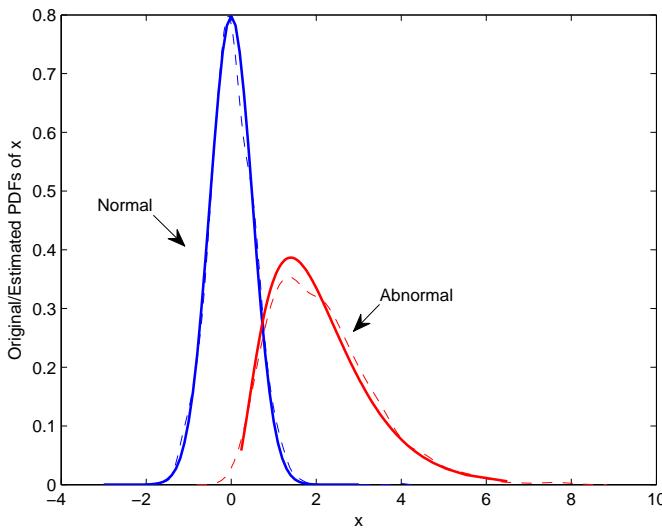


图 3.14 概率密度函数估计结果（虚线）和真实的概率密度函数（实线）

报警阈值 x_{tp} 是 $n = 5$, $x_{tp} = 3.88$ 。

在前两节中, 假设已知过程变量 x 在正常和异常运行状况下的概率密度函数。采用第3.2节中所提出的方法, 可得正常和异常历史数据集, 从而可估计 x 在正常及异常状态下的概率密度函数。例如, 使用高斯核函数进行估计^[18]

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2).$$

图3.14展示了3.3.2节中的 $x(t)$ 在正常和异常运行状况下的概率密度函数估计结果(虚线), 与真实的概率密度函数(实线)是一致的。

3.3.5 案例分析

本节给出一个工业案例, 用来说明报警延迟器的性能评估和优化设计步骤, 以满足对误报率(FAR)、漏报率(MAR)和平均报警延时(AAD)的要求。

过程变量 x 是来自某火电机组的主汽压力, 当前采用了越限报警器, 配置有低报警阈值 $x_{tp} = 23.6$, 即当 x 低于23.6时, 越限报警器将触发报警信息。图3.15展示了火电机组连续运行19个小时的历史数据, 其中采样周期为1秒, 累计 6.8×10^4 个样本数据点。由于生产运行状态的变化, 主汽压力的幅值经历了较大范围的变化, 多次触发报警信息。目标是对当前的越限报警器进行性能评估, 必要时重新设计报警延迟器以满足如下性能指标要求: $\text{FAR} \leq 5\%$, $\text{MAR} \leq 5\%$, $\text{AAD} \leq 5$ 秒。

第一步利用3.3.4节提出的方法估计过程变量 x 处于正常和异常运行状况下的概率密度函数。检测出的变化点及相应的 P 值如表3.5所示, 这里犯第一类错误的概率设定为 $\alpha = 0.05$ 。全部历史数据被分成正常数据段和异常数据段, 如表3.6所示, 这里犯第一类错误的概率设定为 $\beta = 0.05$ 。基于正常数据和异常数据估计出的概率密度函数如图3.16所示。

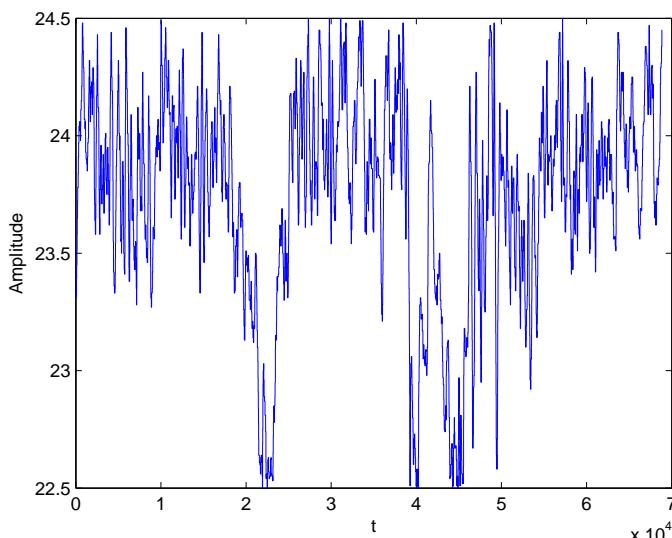


图 3.15 主汽压力的历史数据

表 3.5 均值变化点及相应的P值

变化点	P值
9395	1.10×10^{-144}
18354	0
25065	0
33886	6.80×10^{-72}
39063	0
46135	0
54410	0
63505	9.59×10^{-228}

表 3.6 正常数据段及异常数据段

正常数据	异常数据
$x(1:18353)$	$x(18354:25064)$
$x(25065:39062)$	$x(39063:46134)$
$x(46135:68873)$	

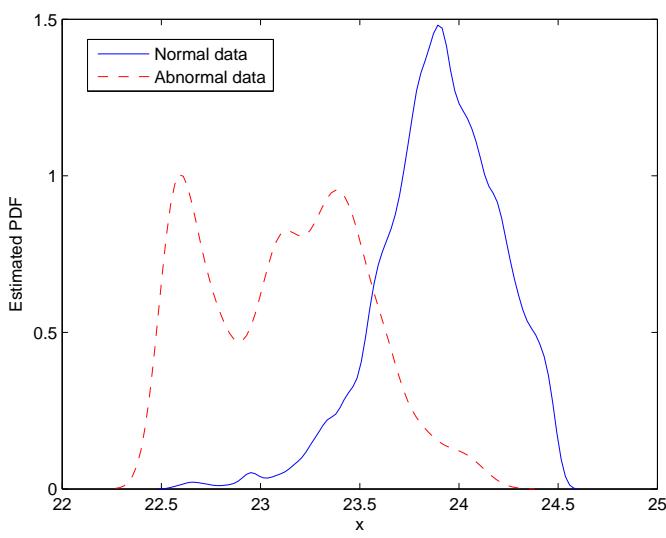


图 3.16 正常（蓝实线）和异常（红虚线）运行状况下的概率密度函数估计结果

第二步，利用图3.16所示的估计得到的概率密度函数，由式(3.10)、(3.11)和(3.12)，得当前报警性能指标FAR = 14.86%，MAR = 12.04%，AAD = 0.1369秒。可见误报率和漏报率比要求的性能指标大得多，需要重新设计报警器。

首先，考虑继续采用越限报警器的情况下设计报警阈值 x_{tp} 。根据误报率、漏报率、平均报警延时三者分别与报警阈值 x_{tp} 的关系，由 $FAR \leq 5\%$ 和 $MAR \leq 5\%$ 这两个要求，可以得到相应的报警阈值 x_{tp} 的范围分别为 $x_{tp} \in [23, 23.37]$ 及 $x_{tp} \in [23.79, 24]$ 。这两个范围没有交集，所以不可能通过设计报警阈值 x_{tp} 来同时满足三个性能指标要求。

接下来，针对当前的报警阈值 $x_{tp} = 23.6$ ，利用报警延迟器，设计报警延迟参数 n 。基于误报率、漏报率、平均报警延时三者分别与报警延迟参数 n 的关系（式(3.14)、(3.21)和(3.22)），由FAR, MAR和AAD三个性能指标要求，可以得到相应的报警延迟参数 n 的有效范围分别为： $n > 1$, $n > 1$ 及 $n < 5$ ，因此同时满足三个性能指标要求的报警延迟参数的有效范围为 $1 < n < 5$ ，相应的报警性能指标如表3.7所示。

最后，可以用与3.3.4节情况3一样的步骤去同时设计报警阈值 x_{tp} 和报警延迟参数 n 。根据式(3.14)、(3.21)和(3.22)，以及三个性能指标要求，报警阈值和报警延迟参数的有效范围如表3.8所示。

可以用与3.3.4节情况3例子相类似的步骤去选择最优的一组报警阈值和报警延迟参数。式(3.29)中的权重选择为 $\omega_1 = \omega_2 = \omega_3 = 1$ ，针对表3.8中的有效范围，利用二维网格搜索的方

表 3.7 固定 $x_{tp} = 23.6$ 时重新设计 n 后的性能指标

n	FAR	MAR	AAD
2	0.0468	0.0305	1.4294
3	0.0116	0.0060	2.8988
4	0.0025	0.0010	4.5694

表 3.8 不同报警延迟参数 n 值时报警阈值 x_{tp} 的有效范围

n	x_{tp} (FAR,MAR)	x_{tp} (AAD)	x_{tp}
2	[23.55,23.60]	[23.15,24]	[23.55,23.60]
3	[23.45,23.70]	[23.40,24]	[23.45,23.70]
4	[23.38,23.76]	[23.58,24]	[23.58,23.76]
5	[23.34,23.78]	[23.77,24]	[23.77,23.78]

表 3.9 不同 n 和 x_{tp} 组合损失函数的最小值

n	x_{tp}	损失函数的最小值
2	23.58	7.7036
3	23.58	2.2133
4	23.59	1.2548
5	23.77	3.8856

法找出与报警延迟参数 n 相对应的最优报警阈值 x_{tp} 及损失函数的最小值，如见表3.9所示，从中可以看出，满足性能指标要求的最优的一组参数为 $n = 4$ 、 $x_{tp} = 23.59$ 。

3.4 报警死区设计

报警死区是一种在工业现场广泛使用的单变量报警器，采用两条不同的报警阈值来触发和清除报警，可以有效地屏蔽掉因过程变量在报警阈值附近波动而触发的干扰报警。以配置高报警阈值 x_{tp} 的过程变量 $x(t)$ 为例，越限报警器在 $x(t)$ 超越 x_{tp} 时触发报警信息，

$$x_a(t) = \begin{cases} 1, & x(t) \geq x_{tp} \\ 0, & x(t) < x_{tp} \end{cases} \quad (3.31)$$

对于死区宽度为 d 的报警死区而言，只有当 $x(t)$ 高于报警阈值 $x_{tp} + d$ 时，才会触发报警，而且只有当 $x(t)$ 低于报警阈值 $x_{tp} - d$ 时，当前触发的报警才会清除，

$$x_a(t; d)(t) = \begin{cases} 1, & x(t) \geq x_{tp} + d \\ 0, & x(t) < x_{tp} - d \\ x_a(t-1; d), & \text{其它} \end{cases} \quad (3.32)$$

与之类似，对低报警阈值而言，报警死区给出的报警变量为

$$x_d(t; d)(t) = \begin{cases} 1, & x(t) \leq x_{tp} - d \\ 0, & x(t) > x_{tp} + d \\ x_d(t-1; d), & \text{其它} \end{cases} \quad (3.33)$$

图3.17(a)给出了过程变量 $x(t)$ 、高报警阈值 x_{tp} 和死区宽度 d ，图3.17(b)为越限报警器给出的报警变量 $x_a(t)$ ，图3.17(c)为报警死区对应的报警变量 $x_d(t; d)$ ，从图中可以看出报警死区可以有效地抑制因报警变量在报警阈值附近波动而触发的报警。

本节首先设计了一种死区指数，用来判断针对某个过程变量，报警延迟器和报警死区哪个更适用于消除干扰报警，然后以处于正常和异常运行状况的历史数据为基础，给出了报警死区宽度的优化设计，最后通过工业案例说明死区指数和报警死区设计方法的有效性。

3.4.1 报警死区适用性判断方法

现代流程工业中存在着大量配置报警的过程变量，变量的特征多种多样，显然报警死区不可能适用于所有的过程变量。如图3.18所示，展示了工业现场的两个过程变量的采样数据（黑色实线），红色虚线为高报警阈值。图中展示的均为处于正常运行状况的数据，因此触发的报警均为误报警。我们可以明显地看出两个过程变量的特征差距比较大：图3.18(a)的过程变量在报警阈值附近波动缓慢，越过报警阈值的幅值不大，但是持续时间比较长；与之相反，图3.18(b)的过程变量在报警阈值附近的波动就比较剧烈，报警持续时间比较短，但越过报警阈值的幅值较大。从报警死区的物理意义考虑，对图3.18 (a)的过程变量设置一个比较小的死区宽

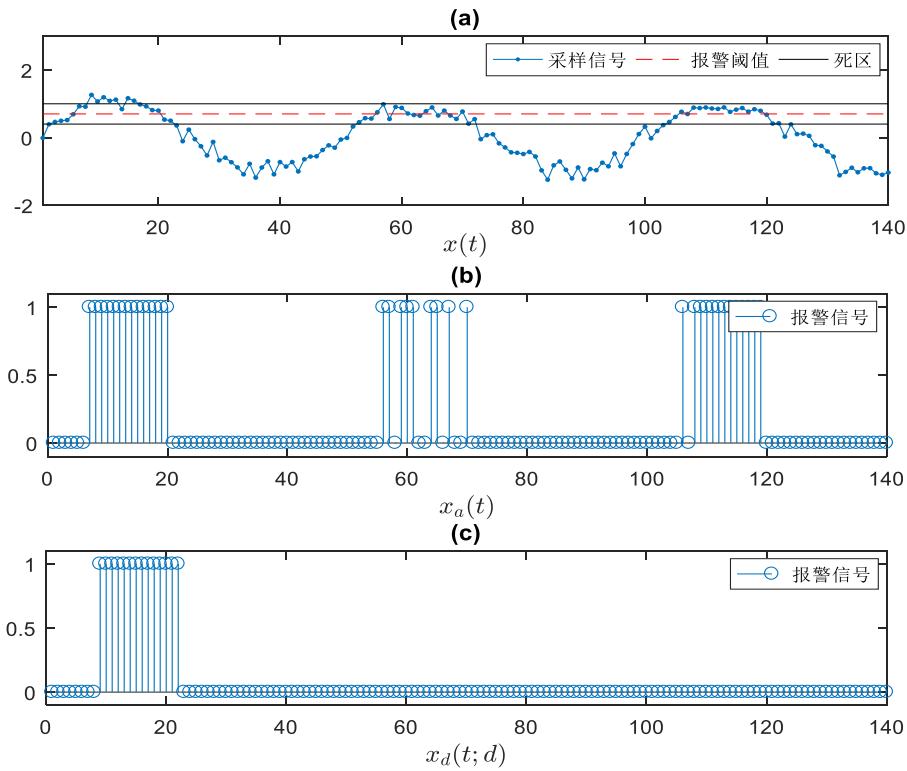


图 3.17 (a)过程变量 $x(t)$ 、高报警阈值 x_{tp} 和死区宽度 d , (b)越限报警器生成的报警变量 $x_a(t)$, (c)报警死区器生成的报警变量 $x_d(t; d)$

度即可避免干扰报警的出现，而对于图3.18(b)的过程变量则需要设置的死区宽度比较大，因此使用报警死区的代价更大，而使用报警延迟器的效果比使用报警死区的效果更好。

根据上述分析，在报警阈值附近波动缓慢、越过报警阈值幅值不大，但是持续时间比较长的过程变量明显适合采用报警死区；而在报警阈值附近波动比较剧烈、报警持续时间比较短、越过报警阈值幅值较大的过程变量明显不适合采用报警死区。因此，我们定义了过程变量及其报警变量的两个基础特征，以此为基础设计了一个死区指数，用来判断报警死区是否适用于消除干扰报警。第一个特征为报警持续时间，是从报警触发到此次报警被清除的持续时间。这里的报警触发指的是报警变量 $x_a(t)$ 从‘0’变到‘1’，而报警清除指的是报警变量 $x_a(t)$ 从‘1’变到‘0’。第*i*次报警的报警持续时间的数学定义为

$$t_{a,i} = (t_{1,i} - t_{0,i} + 1)h, \quad (3.34)$$

其中*h*是采样周期（例如*h*为0.5秒）， $t_{0,i}$ 和 $t_{1,i}$ 为第*i*次报警的报警触发时刻和报警清除时刻，

$$x_a(t_{1,i} - 1) = 0, \quad x_a(t_{0,i} + 1) = 0 \quad \sum_{t=t_{1,i}}^{t_{0,i}} x_a(t) = t_{0,i} - t_{1,i} + 1, \quad t_{1,i} > t_{0,i}.$$

第二个特征为报警偏差，指在一次报警中（即从报警触发到本次报警清除）过程变量 $x(t)$ 和报

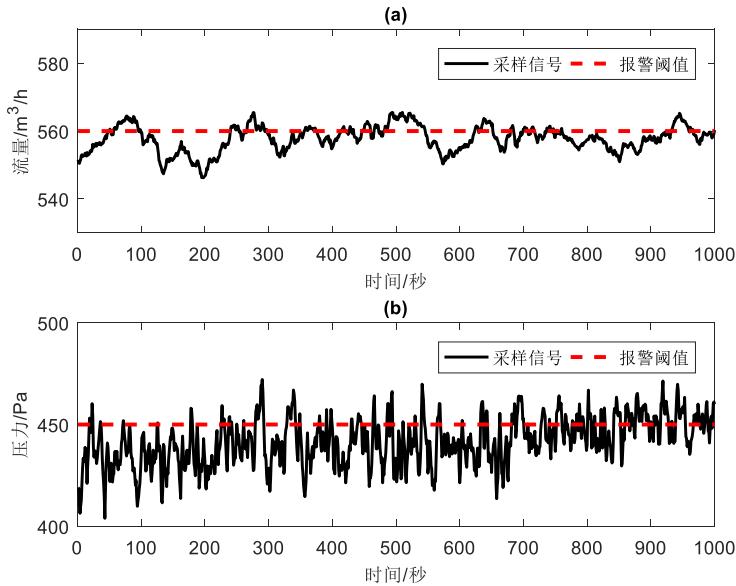


图 3.18 工业现场的两个过程变量（实线）及其高报警阈值（虚线）

警阈值 x_{tp} 差值的最大值。对于高报警阈值 x_{tp} 而言，第*i*次报警偏差的数学定义为

$$d_{a,i} = \max\{(x_{t0,i} - x_{tp}), ((x_{t0,i} + 1) - x_{tp}), \dots, (x_{t1,i} - x_{tp})\}. \quad (3.35)$$

对于这两个特征，报警持续时间反映了报警的时间特征，而报警偏差则反映了报警的幅值特征。根据报警死区的定义可知，报警持续时间长且报警偏差小的过程变量适合采用报警死区，而报警持续时间短且报警偏差比较大的过程变量显然并不适合采用报警死区，使用报警延迟器效果会更好。

对于报警偏差来说，不同的过程变量可能具有不同的量纲，而且过程变量的测量范围差距可能也非常大。例如，液位变量的量程可能是几十米的范围而频率变量的量程可能是几赫兹，所以对液位变量需要设置一个比较大的死区宽度，而对频率变量需要设置一个比较小的死区宽度。对于报警持续时间而言，有些变量变化速度比较慢，施加一个比较大的延迟时间是可以接受的，但是有些变量的变化速度比较快，显然不能接受一个比较大的延迟时间。在现代工业中，监控变量数以千/万计，为了能对大量的过程变量进行统一的比较，需要对报警持续时间和报警偏差进行标准化。标准化的报警持续时间为

$$T_{a,i} = \frac{t_{a,i}}{t_{a,max}}, \quad (3.36)$$

其中 $t_{a,max}$ 为可接受的最大报警持续时间，可将工业标准EEMUA中对报警延迟器的建议值作为 $t_{a,max}$ 的默认值，详见表3.10。标准化的报警偏差为

$$D_{a,i} = \frac{d_{a,i}}{d_{a,max}}, \quad (3.37)$$

其中 $d_{a,max}$ 为可接受的最大报警偏差。一种选择 $d_{a,max}$ 的方法是使得 $d_{a,max}$ 与 x_{tp} 成比例，即

$$d_{a,max} = p \cdot x_{tp}. \quad (3.38)$$

表 3.10 $t_{a,max}$ 和 p 的默认值

信号类型	$t_{a,max}$	p
流量	15秒	5%
液位	60秒	2%
压力	15秒	2%
温度	60秒	1%

可将工业标准EEMUA中对死区宽度的建议值作为p的默认值，表3.10 中给出了p的默认值。

采用一个线性方程描述标准化报警偏差 D_a 和标准化报警持续时间 T_a 之间的关系，

$$T_a = s \cdot D_a. \quad (3.39)$$

那么，以标准化报警持续时间和标准化报警偏差为基础，死区指数 $\theta \in [0^\circ, 90^\circ]$ 定义为标准化报警偏差 D_a 和标准化报警持续时间 T_a 构成直线的夹角，

$$\theta = \arctan(s). \quad (3.40)$$

根据标准化报警偏差和标准化报警持续时间的样本值 $\{D_{a,i}, T_{a,i}\}_{i=1}^n$ ，可以通过最小二乘法估计斜率系数 s ，

$$\hat{s} = \frac{\sum_{i=1}^n D_{a,i} \cdot T_{a,i}}{\sum_{i=1}^n D_{a,i}^2}. \quad (3.41)$$

所以相应的死区指数估计值为

$$\hat{\theta} = \arctan(\hat{s}). \quad (3.42)$$

标准化报警偏差 D_a 和标准化报警持续时间 T_a 的线性关系主要用来反映过程变量的幅值偏差特征、时间特征哪个更加显著。如果该直线更接近纵坐标时，死区指数较大，说明该变量的报警持续时间普遍较长，但报警偏差相对小很多。根据报警死区的定义，可以理解为报警死区是通过消除幅值上的特征来消除干扰报警，那么对于报警偏差小的变量，使用报警死区的代价会比较小，因此对死区指数较大的变量就比较适合采用报警死区。如果该直线更接近横坐标，死区指数较小，则说明该变量的报警持续时间普遍较短，而报警偏差都相对大很多；如果采用报警死区，那么死区宽度需要设置的非常大才能满足报警性能要求，使用的代价比较大，因此报警死区不适合这种变量。综上所述，可以根据死区指数的大小来判断过程变量是否适合使用报警死区，即死区指数大的变量适合使用报警死区，而死区指数较小的变量不适合使用报警死区。

通过式(3.36)、(3.37)对报警持续时间和报警偏差进行标准化，可以将 $D_{a,i}$ 和 $T_{a,i}$ 理解为对应最大报警偏差 $d_{a,max}$ 和最大报警持续时间 $t_{a,max}$ 的代价。显然，当两者相同时，对应的死区指数为判断是否适合使用报警死区的临界值，即死区指数的临界值为 45° 。换言之，当死区指数 θ 大于临界值 45° 时， $x(t)$ 适合采用报警死区，而当死区指数 θ 小于临界值 45° 时， $x(t)$ 适合采用报警延迟器。为了准确判断死区指数和临界值关系，需要计算死区指数的置信区间。根据中心极限定理，斜率系数估计值 \hat{s} 服从高斯分布，其在95%置信度下的置信区间为

$$[s_-, s_+] = [\hat{s} - 1.96\sigma_{\hat{s}}, \hat{s} + 1.96\sigma_{\hat{s}}], \quad (3.43)$$

上式中 $\sigma_{\hat{s}}$ 为

$$\sigma_{\hat{s}} = \sqrt{\frac{\sum_{i=1}^n (T_{a,i} - \hat{s}D_{a,i})^2}{(n-2) \sum_{i=1}^n (D_{a,i} - \bar{D}_a)^2}}, \quad (3.44)$$

其中

$$\bar{D}_a = \frac{1}{n} \sum_{i=1}^n D_{a,i}. \quad (3.45)$$

死区指数和斜率系数为反正切关系，而且反正切函数为单调函数，那么死区指数估计值 $\hat{\theta}$ 同样服从高斯分布，在95%置信度下的置信区间为

$$[\theta_-, \theta_+] = [\hat{\theta} - 1.96\sigma_{\hat{\theta}}, \hat{\theta} + 1.96\sigma_{\hat{\theta}}], \quad (3.46)$$

其中

$$\sigma_{\hat{\theta}} = \arctan(\hat{s}). \quad (3.47)$$

考虑到死区指数 θ 的估计误差，这里提出了一个假设检验：

$$H_0 : \theta = 45^\circ, H_1 : \theta \neq 45^\circ. \quad (3.48)$$

对于原假设 H_0 ，在显著性水平 $\alpha = 0.05$ 下的接受域为

$$[45 - 1.96\sigma_{\hat{\theta}}, 45 + 1.96\sigma_{\hat{\theta}}]. \quad (3.49)$$

当 $\hat{\theta} \geq (45 + 1.96\sigma_{\hat{\theta}})$ 时，拒绝原假设，判断该过程变量适合采用报警死区；当 $\hat{\theta} \leq (45 - 1.96\sigma_{\hat{\theta}})$ 时，无法拒绝原假设，判断该过程变量并不适合采用报警死区。

这里给出一个仿真案例来说明死区指数的有效性。两个过程变量 $x_1(t)$ 和 $x_2(t)$ 由公

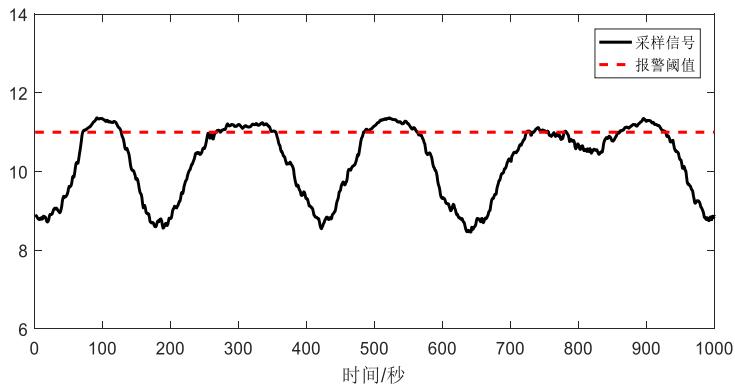
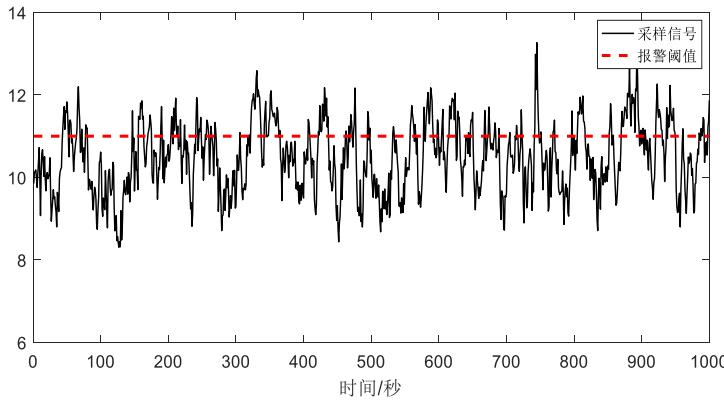


图 3.19 $x_1(t)$ 的部分样本数据

图 3.20 $x_2(t)$ 的部分样本数据

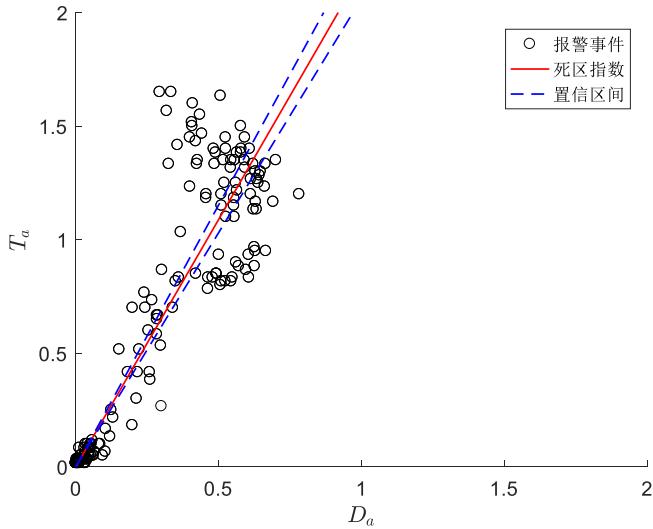
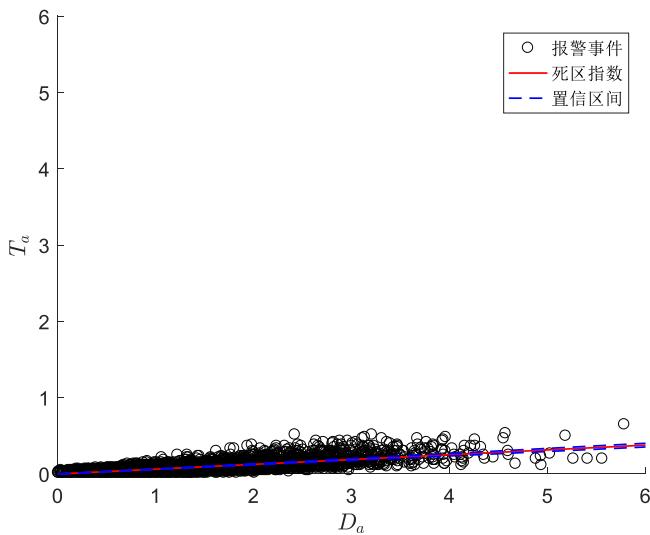
式(3.50)中描述的两个自回归模型产生。

$$\begin{aligned}x_1(t) &= -0.995 \cdot x_1(t-1) + e_1(t) + 10, \\x_2(t) &= -0.85 \cdot x_2(t-1) + e_2(t) + 10,\end{aligned}\quad (3.50)$$

其中 $e_1(t)$ 是均值为 0、方差为 0.01 的高斯白噪声， $e_2(t)$ 是均值为 0、方差为 0.25 的高斯白噪声，且 $e_1(t)$ 和 $e_2(t)$ 是相互独立的。假设 $x_1(t)$ 和 $x_2(t)$ 均处于正常运行状况下，数据长度均为 40000，采样周期为 1 秒，图 3.19 和图 3.20 分别给出了 $x_1(t)$ 和 $x_2(t)$ 的部分样本数据。两个过程变量配置的高报警阈值均为 $x_{tp} = 11$ （如图 3.19 和图 3.20 中虚线所示），由越限报警器 3.31 生成报警变量 $x_{a,1}(t)$ 和 $x_{a,2}(t)$ 。从图 3.19 和图 3.20 展现的部分样本数据可以看出， $x_1(t)$ 的特征是报警偏差较小而报警持续时间较长，而 $x_2(t)$ 的特征是报警偏差较大而报警持续时间较短，显然从直观上就可以判断 $x_1(t)$ 适合采用报警死区，而 $x_2(t)$ 不适合用报警死区。

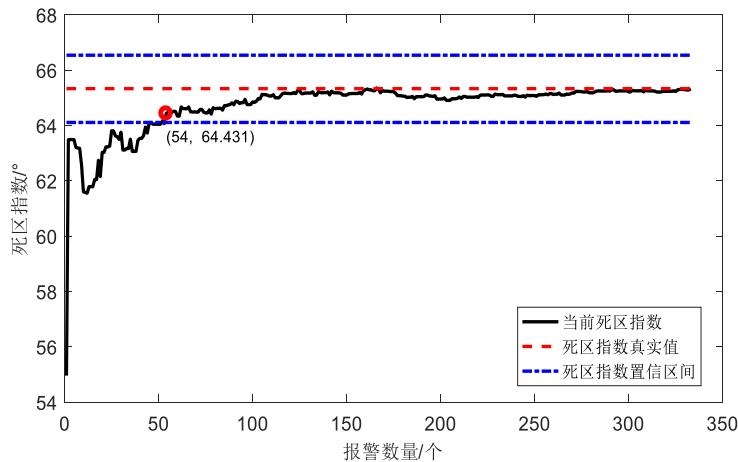
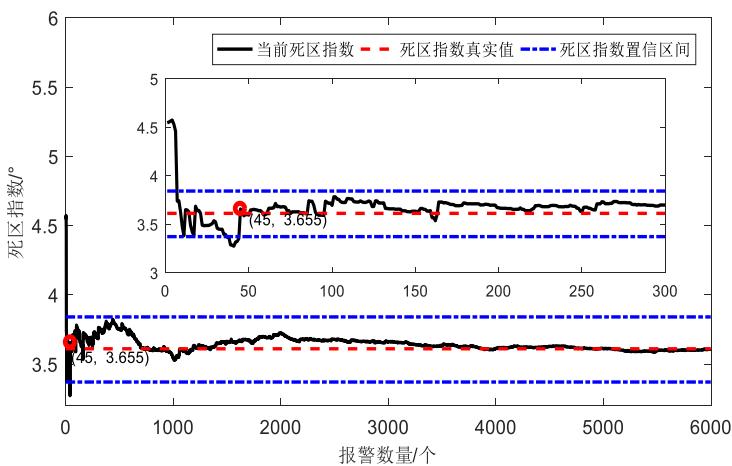
根据本节介绍的计算步骤，首先通过式(3.34)和(3.35)分别计算 $x_1(t)$ 和 $x_2(t)$ 的报警持续时间和报警偏差，再根据表 3.10 得到 $x_1(t)$ 和 $x_2(t)$ 的最大报警持续时间 $t_{a,max} = 60$ 秒、最大报警偏差 $d_{a,max} = 0.05 \cdot x_{tp} = 0.55$ ，并以此计算标准化警持续时间 T_a 和标准化报警偏差 D_a 。然后，根据标准化后的报警持续时间和报警偏差分别计算 $x_1(t)$ 和 $x_2(t)$ 的死区指数 $\hat{\theta}_1$ 、 $\hat{\theta}_2$ 及其置信区间。对于 $x_1(t)$ ，死区指数为 $\hat{\theta}_1 = 65.33^\circ$ ，置信区间为 $[43.78^\circ, 46.22^\circ]$ ，显然拒绝原假设 H_0 ，认为 $\hat{\theta}_1 > 45^\circ$ ，所以 $x_1(t)$ 适合使用报警死区。对于 $x_2(t)$ ，死区指数为 $\hat{\theta}_2 = 3.61^\circ$ ，置信区间为 $[3.37^\circ, 3.84^\circ]$ ，显然拒绝零假设 H_0 ，认为 $\hat{\theta}_2 > 45^\circ$ ，所以 $x_2(t)$ 不适合采用报警死区。如图 3.21 和图 3.22 所示，图中每个圆圈都对应一个报警事件，实线对应着死区指数，实线两侧虚线对应死区指数的置信区间。死区指数的判断结果和过程变量的特征保持一致。

在采用死区指数判断一个过程变量是否适合采用报警死区时，还有一个问题需要考虑，那就是报警数量的多少对死区指数估计结果的影响。显而易见，当使用较长时间的样本数据计算死区指数时，如果触发的报警数量很少，此时计算的死区指数几乎不具有可靠性，无法反映过程变量的特征。因此，在计算死区指数时，样本数据的长度对死区指数的应用效果影响不大，

图 3.21 $x_1(t)$ 的死区指数及置信区间图 3.22 $x_2(t)$ 的死区指数及置信区间

报警数量是关键的影响因素，即报警数量越多那么计算死区指数越可靠，越能够反映过程变量的特征。因此，具体的问题是：报警数量要达到多少，计算的死区指数才可靠？

为了判断报警数量对死区指数的影响情况，研究思路是：当报警数量逐渐增加时，计算对应的死区指数，死区指数计算结果必然会随着报警数量的增加收敛到稳定值，这时说明报警数量已经足够。根据上述研究思路，对上述的两个过程变量 $x_1(t)$ 和 $x_2(t)$ 进行测试。图3.23反映的是 $x_1(t)$ 的报警数量和死区指数的关系， $x_1(t)$ 的报警数量共有333个，使用全部样本数据获得的死区指数为 $\hat{\theta}_1 = 65.33^\circ$ （图中红色虚线所示），死区指数的置信区间为 $[64.11^\circ, 66.54^\circ]$ （图中蓝色点画线所示），当报警数量大于54个以后，死区指数已经进入到置信区间内部（图中红色

图 3.23 $x_1(t)$ 的报警数量和死区指数关系图图 3.24 $x_2(t)$ 的报警数量和死区指数关系图

圆圈所示），并逐渐接近死区指数的收敛值。图3.24反映的是 $x_2(t)$ 的报警数量和死区指数的关系， $x_2(t)$ 的报警数量共有5997个；为了清晰的展示，图3.24中小图展示了报警数量到300个时报警数量和死区指数的关系，使用全部样本数据得到的死区指数为 $\hat{\theta}_2 = 3.61^\circ$ （图中红色虚线所示），死区指数的置信区间为 $[3.37^\circ, 3.84^\circ]$ （图中蓝色点画线所示），当报警数量大于45个以后，死区指数已经进入到置信区间内部（图中红色圆圈所示），并逐渐接近死区指数的收敛值。因此，报警数量到达50个左右时，能够有效保证死区指数估计结果的可靠性。

3.4.2 报警死区的设计方法

本节首先研究了一种基于马尔科夫模型的报警死区设计方法，该方法假设正常和异常状态系统的工作状态是平稳地，其正常状态和异常状态对应的采样信号同为平稳信号，此时系统中得到的信号采样值主要受噪声干扰，认为这些采样信号均服从独立同分布。然后基于上述假设，计算报警系统的误报率、漏报率和平均报警延迟三个性能指标。最后基于三个性能指标之间的折衷设计最优死区宽度。当采用报警死区时，信号概率密度函数如图3.25所示，其中，

$$q_1 = \int_{x_{tp}+d}^{+\infty} q(x) dx, q_2 = \int_{-\infty}^{x_{tp}-d} q(x) dx, \quad (3.51)$$

$$p_1 = \int_{x_{tp}+d}^{+\infty} p(x) dx, p_2 = \int_{-\infty}^{x_{tp}-d} p(x) dx. \quad (3.52)$$

当死区宽度为 d ， q_1 为正常状态概率密度函数 $q(x)$ 大于报警死区高阈值 $x_{tp} + d$ 的概率， q_2 为正常状态概率密度函数 $q(x)$ 小于报警死区低阈值 $x_{tp} - d$ 的概率，且 $q_1 + q_2 \neq 1$ ； p_1 为异常状态概率密度函数 $p(x)$ 大于报警死区高阈值 $x_{tp} + d$ 的概率， p_2 为异常状态概率密度函数 $p(x)$ 小于报警死区低阈值 $x_{tp} - d$ 的概率，且 $p_1 + p_2 \neq 1$ 。

无论对于正常信号还是异常信号，都可以将信号当前所处的状态分为报警状态（A）和非报警状态（NA）两个状态。因为正常信号和异常信号均服从独立同分布，其状态之间的转移概率是固定的（例如对正常信号，从报警状态转移到非报警状态的概率固定为 q_1 ），而且从一个状态转移到下一个状态的概率，只与当前时刻所处状态有关，而与历史状态无关，所以状态的转移之间具有马尔科夫性。因此本方法可以采用马尔科夫模型计算误报率、漏报率和平均报警延迟三个性能指标，并基于这三个指标设计最优死区宽度。

当系统运行正常时，信号状态转移马尔科夫链如图3.26所示。根据图3.25、3.26及误报率

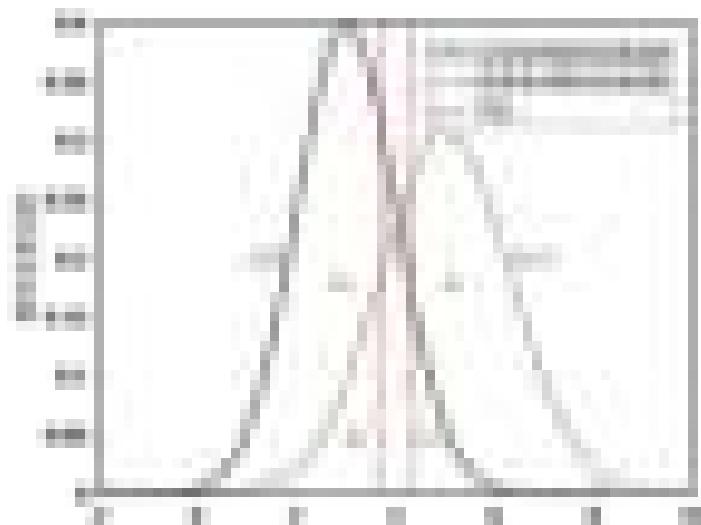


图 3.25 正常信号和异常信号的概率密度函数



图 3.26 正常信号状态转移马尔科夫链

的定义可知，报警死区的误报率为

$$FAR = \frac{q_1}{q_1 + q_2}. \quad (3.53)$$

此时，状态之间的转移概率可以用 P_n 表示，

$$P_n = \begin{bmatrix} 1 - q_1 & q_1 \\ q_2 & 1 - q_2 \end{bmatrix}.$$

类似的，系统运行出现异常时，信号状态转移马尔科夫链如图3.27所示。根据图3.25、3.27及漏报率的定义计算报警死区的误漏报率为

$$MAR = \frac{p_2}{p_1 + p_2}. \quad (3.54)$$

此时，状态之间的转移概率可以用 P_f 表示

$$P_f = \begin{bmatrix} 1 - p_1 & p_1 \\ p_2 & 1 - p_2 \end{bmatrix}.$$

当系统运行正常时，经过有限的过渡时间后，马尔科夫链将会达到稳定状态，其状态概率会收敛到一个不变向量，即马尔科夫链的平稳分布，对 P_n 计算得到平稳分布为

$$\pi_n = \left[\frac{q_2}{q_1 + q_2} \quad \frac{q_1}{q_1 + q_2} \right].$$

当故障发生时，马尔科夫过程从正常状态模型（图3.26由 P_n 表示）转移到异常状态模型（图3.27由 P_f 表示），此时正常状态下的平稳分布应该作为异常状态下的初始概率分布。本节讨论的马尔科夫链均为遍历不可约非周期的有限马尔科夫链，对于一个遍历不可约非周期的有限马尔科夫链，必然存在平稳分布，且极限是唯一的分布平稳分布，因此异常状态的马尔科夫链的初始条件唯一且必然存在。

令 $X(t), t \in T$ 为一组状态序列， $X(t)$ 表示 t 时刻的报警状态。如果系统在 t_0 时刻从正常状态



图 3.27 异常信号状态转移马尔科夫链

转移到异常状态，采用查普曼-柯尔莫哥洛夫方程，可以计算得到非报警状态和报警状态的概率为

$$\begin{aligned} [P(X(t_0) = NA) P(X(t_0) = A)] &= \pi_n P_f \\ &= [P(X(t_0 - 1) = NA) P(X(t_0 - 1) = A)] P_f \\ &= \left[\frac{q_1 p_2 + q_2(1 - p_1)}{q_1 + q_2} \frac{q_2 p_1 + q_1(1 - p_2)}{q_1 + q_2} \right]. \end{aligned} \quad (3.55)$$

报警延迟为0的概率（当从正常状态转移到异常状态时立即报警）为

$$P(T_d = 0) = P(X(t_0) = A) = \frac{q_1 p_2 + q_2(1 - p_1)}{q_1 + q_2}. \quad (3.56)$$

报警延迟为1的概率为

$$\begin{aligned} P(T_d = 1) &= P(X(t_0) = NA) \times P(X(t_0 + 1) = A | X(t_0 = NA)) \\ &= \frac{q_1 p_2 + q_2(1 - p_1)}{q_1 + q_2} \times p_1. \end{aligned} \quad (3.57)$$

类似的，其他的报警延迟可以计算为

$$\begin{aligned} P(T_d = 2) &= \frac{q_1 p_2 + q_2(1 - p_1)}{q_1 + q_2} \times p_1(1 - p_1) \\ &\vdots \\ P(T_d = n) &= \frac{q_1 p_2 + q_2(1 - p_1)}{q_1 + q_2} \times p_1(1 - p_1)^{n-1}. \end{aligned} \quad (3.58)$$

最后可以求得平均报警延迟（报警延迟的期望值）为

$$\begin{aligned} E(T_d) &= \sum_{n=0}^{\infty} n P(T_d = n) = \sum_{n=1}^{\infty} n p_1(1 - p_1)^{n-1} \frac{q_1 p_2 + q_2(1 - p_1)}{q_1 + q_2} \\ &= \frac{q_1 p_2 + q_2(1 - p_1)}{p_1(q_1 + q_2)}. \end{aligned} \quad (3.59)$$

根据式(3.50)、(3.51)和(3.59)可以看出当死区宽度不断增大时，误报率和漏报率会以相同的幅度减小，但是平均报警延迟会不断增大。误报率、漏报率可以反映报警系统的准确性，而平均报警延迟则反映了报警系统灵敏性，为了兼顾报警系统的准确性和灵敏性，需要根据三个性能指标的折衷设置最优死区宽度。当给出三个性能指标的要求时，即可计算得到一组满足性能指标的死区宽度，然后采用了如式(3.60)所示的评价函数来确定最优死区宽度，使J取最小的死区值即为最优死区宽度。

$$J(d) = \omega_1 \frac{FAR}{RFAR} + \omega_2 \frac{MAR}{RMAR} + \omega_3 \frac{AAD}{RAAD}, \quad (3.60)$$

其中，RFAR为误报率的性能要求，RMAR为漏报率的性能要求，RAAD为平均报警延迟的性能要求，例如当设计的报警系统性能要求为：误报率小于5%、误报率小于5%、平均报警延迟小于8h (h等于1秒，为信号采样周期) 时，RFAR = 5%、RMAR = 5%、RAAD = 8。 ω_1 、 ω_2 、 ω_3 反映三个性能指标的影响权重，这里并不考虑某个性能指标具有更高的影响权

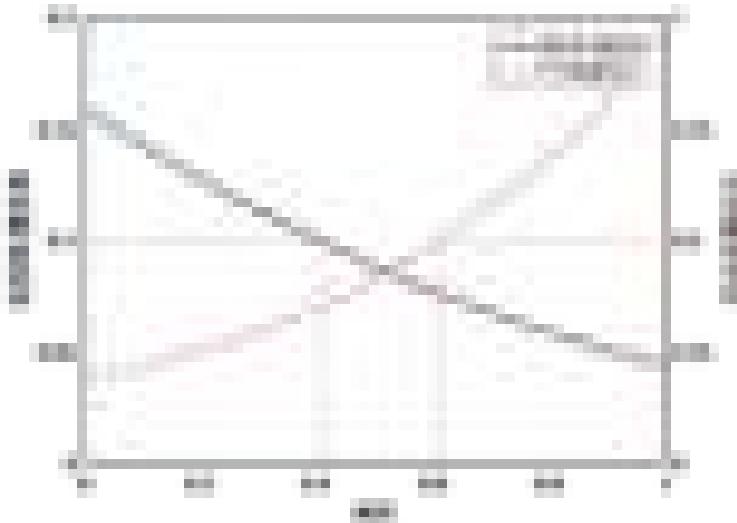


图 3.28 死区宽度和三个性能指标之间的关系

重，因此 ω_1 、 ω_2 、 ω_3 都取值为1。下面给出一个数值仿真详细说明设计方法。

数值仿真：根据式(3.61)产生信号 $x(t)$ ， $x(t)$ 的数据长度为2000，采样周期 h 为1s，前1000个为正常数据后1000个为异常数据，报警阈值 x_{tp} 为4。为该信号设计报警死区，使其满足误报率小于10%、漏报率小于10%、平均报警延迟小于0.4秒。

$$\begin{cases} x(t) \sim N(3, 1), & t < 1000 \\ x(t) \sim N(5, 1), & t \geq 1000 \end{cases} \quad (3.61)$$

图3.28反映了误报率、漏报率、平均报警延迟之间的关系，从图中可以看出，当满足误报率、漏报率时死区宽度的范围为[0.41,1]，当满足平均报警延迟时死区宽度的范围为[0,0.61]，两者之间存在交集，交集为[0.41,0.61]，交集即是满足所有性能指标要求的死区宽度的范围，此时根据式(3.62)计算当评价函数 $J(d)$ 取最小时，得到最优死区宽度为0.61。

本方法使用的前提是正常信号和异常信号均服从独立同分布，实际中很多工业满足这样的假设条件。但是在后续研究中发现，工业中大部分重要的信号并不服从这种假设，这些信号的变化比较随机，因此本节研究的报警死区设计方法对实际工业信号的适用性不足。

针对上述基于马尔科夫模型的报警死区设计方法适用性不足的缺陷，我们提出了一种新的报警死区设计方法，根据工业历史数据设计报警死区，不需要了解信号的统计分布。该方法主要基于报警系统性能指标：干扰报警持续时间比（Nuisance-Alarm Duration Ratio, NADR）来设计最优死区宽度，干扰报警持续时间比定义为干扰报警的总持续时间对报警数据 $\{x(t)\}_{t=1}^N$ 总时长的占比，报警数据总时长为 Nh （ h 为采样周期，单位为秒）。对根据式(2.1)产生的报警变量 $x_a(t)$ ，其干扰报警持续时间比为

$$F = \frac{\sum_{t=1}^N x_a(t)}{N}. \quad (3.62)$$

近似的，当采用报警死区，死区宽度为 d 时，对应的报警变量 $x_d(t; d)$ 如式(3.32)或(3.33)所示，

其干扰报警持续时间比为

$$\bar{F}(d) = \frac{\sum_{t=1}^N x_d(t; d)}{N}. \quad (3.63)$$

随着死区宽度 d 的增加，干扰报警持续时间比会不断地减小，但是检测到异常情况的延迟时间会不断的增加。表现为在工业中出现异常状况时，采样信号 $x(t)$ 也会进入异常状态，这个时候应该尽可能迅速的触发报警以提醒操作人员出现异常，使操作人员可以尽快处理，报警系统的设计初衷也是如此。实际情况是异常情况的发生与相应的报警发生之间会有一个小的时间差，这个时间差被称为报警延迟，报警延迟是一个随机的变量，报警延迟的平均值被称为平均报警延迟。对于一个工业报警系统来说，平均报警延迟越大意味着出现异常状况时触发相应的报警时间间隔越长，此时留给工厂操作人员处理异常状况的时间就越少。相反平均报警延迟小则说明报警系统灵敏性好，当被测监控过程变量异常时，报警系统动作迅速，使操作人员有更充足的时间来处理报警。

因此，本方法仍将平均报警延迟作为另一个与干扰报警持续时间比相平衡的性能指标。由于本节提出的方法和3.4.2节方法的假设不同，所以3.4.2节的计算平均报警延迟的方法在这里并不适用，而且实际工业信号从正常状态到异常状态的转换过程很难描述，所以平均报警延迟的解析表达式很难直接获得。考虑到根据实际工业信号难以直接计算报警器平均报警延迟的问题，我们提出了一种间接地计算方法，这个方法的主要思路是认为工业标准中报警死区和报警延迟器的建议值具有相同的代价，即当采用建议死区宽度时，报警器的平均报警延迟应该近似等于延迟的建议值，基于此可以得到平均报警延迟估计值 $\bar{m}(d)$ 为

$$\bar{m}(d) = \frac{d}{d_{a,\max}} \cdot t_{a,\max}, \quad (3.64)$$

其中 $t_{a,\max}$ 、 $d_{a,\max}$ 分别为最大报警持续时间和最大报警偏差，因为 $t_{a,\max}$ 和 $d_{a,\max}$ 均为标准化参数且具有可比性， $\bar{m}(d)$ 是具有实际物理意义的，所以报警延迟估计值可以用于比较报警死区的设计效果。最终满足式(3.65)、(3.66)约束条件的最小值即为最优死区宽度，

$$\bar{F}(d) \leq F_{\max}, \quad (3.65)$$

$$\bar{m}(d) \leq t_{a,\max}. \quad (3.66)$$

其中 F_{\max} 为干扰报警持续时间比的上界，例如 $F_{\max} = 0.05$ ，实质是对 d 进行网格搜索来求解最优死区宽度 d_{opt} 。

在死区指数判断过程中，当零假设 H_0 被拒绝且信号的死区指数满足 $\hat{\theta} \leq (45 - 1.96\sigma_{\theta})$ 时，信号具有报警持续时间短、报警偏差较大的特征，显然报警延迟器将更适合用来减少干扰报警。为了与本章的方法作对比，这里同时对每个信号设计了报警延迟器，当采用 m -报警延迟器时，其工作原理是当且仅当报警变量 $x_a(t)$ 中 m 个连续的信号都为‘1’时，报警延迟器触发报警，当报警变量 $x_a(t)$ 中 m 个连续的信号都为‘0’时报警延迟器才能清除报警，因此 m -报警延迟器的报警变量为

$$x_m(t; m) = \begin{cases} 1, & x_a(t - m + 1 : t) = 1 \text{ 且 } x_m(t - 1; m) = 0 \\ 0, & x_a(t - m + 1 : t) = 0 \text{ 且 } x_m(t - 1; m) = 1 \\ x_m(t - 1; m), & \text{其他} \end{cases}, \quad (3.67)$$



图 3.29 过程变量 $x_1(t)$ 死区宽度 d 和 $\bar{F}(d)/\bar{m}$ 关系图

其中 $x_a(t_1 : t_2)$ 代表了从样本时刻 t_1 到 t_2 的报警变量。根据以上描述可以看出 m -报警延迟器的延迟 m 具有明显和报警持续时间相关的物理意义，即报警变量 $x_a(t)$ 中所有触发的报警持续时间小于 mh 秒的报警都在 $x_m(t; m)$ 中消失了，由此可见报警延迟器可以有效减少报警持续时间较小的报警。下面给出一个数值仿真说明本部分提出的报警死区设计方法的有效性。

数值仿真：本仿真案例中的两个过程变量 $x_1(t)$ 和 $x_2(t)$ 由式(3.50)中描述的两个自回归模型产生，报警系统的性能要求是干扰报警持续时间比不超过5%。本节已通过死区指数判断得到过程变量 $x_1(t)$ 适合使用报警死区，而过程变量 $x_2(t)$ 并不适合使用报警死区。

根据式(3.63)，其中死区宽度 d 以0.01的步长变化计算得到一组，相应的根据式(3.64)计算出一组延迟估计值 $\bar{F}(d)$ 。如图3.29所示，其中实线为死区宽度 d 与 $\bar{F}(d)$ 对应关系，虚线为死区宽度 d 与 \bar{m} 对应关系。计算得到最优死区宽度为 $d_{opt} = 0.34$ ，满足 $\bar{F}(d_{opt}) = 0.043 < 0.05$ ，其对应的延迟估计值为 $\bar{m}(d) = 37.09$ 秒。

作为参照和对比，我们同时也采用报警延迟器对过程变量 $x_1(t)$ 设计报警系统。类似于式(3.63)，报警延迟器的干扰报警持续时间比为

$$F'(m) = \frac{\sum_{t=1}^N x_m(t; m)}{N}. \quad (3.68)$$

当延迟不断增大时，可以认为平均报警延迟等于延迟 m 。随着报警延迟器的延迟 m 的增加可以计算得到一组干扰报警持续时间比 $F'(m)$ 。如图3.30所示，其中实线为延迟 m 与 $\bar{F}(m)$ 的对应关系，虚线延迟 m 和平均报警延迟的对应关系。通过式(3.68)可以计算得到最优延迟应为 $m_{opt} = 88$ 秒，以满足性能指标的要求 $F'(m_{opt}) = 0.0469 < 0.05$ 。

当满足报警系统性能要求时，延迟估计值为 $\bar{m}(d) = 37.09$ 秒。而对报警延迟器来说，其最优延迟为 $m_{opt} = 88$ 秒，显而易见， 37.09 秒< 88 秒，与死区指数分析结果一致，过程变量 $x_1(t)$ 适合采用报警死区设计报警系统。作为对比，本节同时对过程变量 $x_2(t)$ 设计了报警死区和报警延迟器，使死区宽度 d 以0.01的步长变化计算出一组 $\bar{F}(d)$ 和对应的一组延迟估计值 $\bar{m}(d)$ 。如图3.31所示，根据报警系统性能指标即可计算得到最优死区宽度为 $d_{opt} = 2.22$ ，满

图 3.30 信号 $x_1(t)$ 延迟 m 和 $F'(m)/m$ 关系图图 3.31 过程变量 $x_2(t)$ 死区宽度 d 和 $\bar{F}(d)/\bar{m}$ 关系图

足 $\bar{F}(d_{opt}) = 0.049 < 0.05$ ，其对应的延迟估计值为 $\bar{m}(d) = 242.18$ 秒。

同时对于报警延迟器，随报警延迟器延迟 m 增加计算得到一组干扰报警持续时间比 $F'(m)$ ，如图3.32所示，当满足报警系统性能指标时最优延迟为 $m_{opt} = 22$ 秒。显然，对于过程变量 $x_2(t)$ 来讲，报警延迟器比报警死区更适合用来减少干扰报警。与基于马尔科夫模型的报警死区设计方法相比，本节提出的报警死区设计方法使用范围更广，无需对了解过程变量的统计分布即可计算最优死区宽度。

3.4.3 案例分析

本节提出的报警系统死区设计方法的主要流程包括两大部分：

图 3.32 过程变量 $x_2(t)$ 延迟 m 和 $F'(m)/m$ 关系图

第一部分介绍了报警死区适用性判断。根据过程变量和报警阈值计算报警持续时间和报警偏差，通过工业标准给出的建议值将报警持续时间和报警偏差进行标准化，并基于标准化报警持续时间和标准化报警偏差，计算死区指数和死区指数的置信区间，采用假设检验即可判断信号是否适合使用报警死区。

第二部分介绍了报警死区设计。以干扰报警持续时间比作为系统的性能指标，通过对 d 进行网格搜索来求解最优死区宽度。

根据上述介绍，报警死区设计具体的实现步骤如下：

步骤1：给出过程变量的历史数据 $\{x(t)\}_{t=1}^N$ 及报警阈值 x_{tp} ，根据历史数据 $x(t)$ 及报警阈值 x_{tp} ，计算每次报警的报警持续时间和报警偏差。

步骤2：为了使所有变量的报警持续时间和报警偏差具有可比性，通过式(3.36)和(3.37)对报警持续时间和报警偏差进行标准化。

步骤3：根据标准化后的报警持续时间 T_a 和报警偏差 D_a ，计算死区指数及死区指数的置信区间，如式(3.42)和(3.46)所示，并结合假设检验判断信号是否适合使用报警死区。

步骤4：计算死区宽度不同时系统的干扰报警持续时间比，以干扰报警持续时间比作为系统的性能指标，通过对 d 进行网格搜索来求解最优死区宽度。

下面通过工业案例分析验证本节提出方法的效果。本节案例分析的所有工业数据均来源于山东省某台300MW燃煤发电机组。本节所用数据为该机组2015年历史数据，所有数据都可认为是正常工作状况下的数据，且其采样周期均为1s，报警系统产生的报警信号均可认为是干扰报警，设计目标为减少这些干扰报警。基于历史数据的工业案例中，首先判断信号是否适合采用报警死区，然后设计满足性能要求的最优死区宽度。

表3.11中列出了部分工业信号，本节选取了其中的五个信号进行工业案例仿真，案例中涵盖了死区指数判断步骤中会出现所有的三种结果：适合使用报警死区、使用报警死区和其他技术差距不大、不合适使用报警死区，最后给出了适合使用报警死区的信号采用最优死区后报警数量减少的数量。

表 3.11 工业过程变量的死区指数及其置信区间

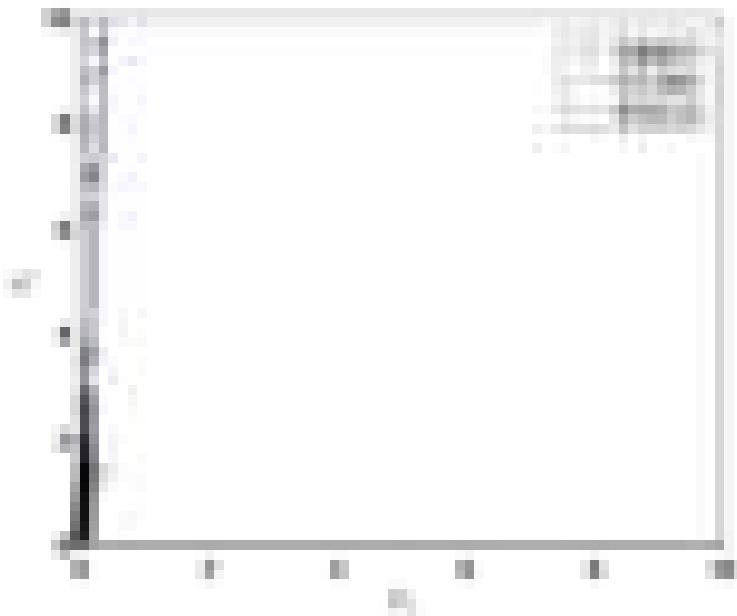
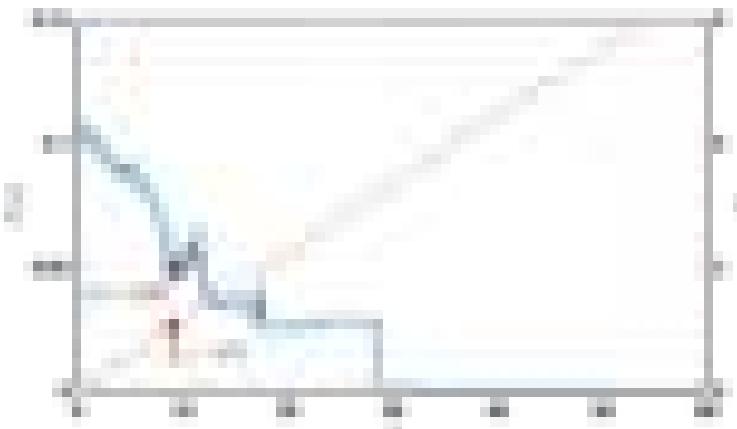
序号	信号	变置名	死区指数	置信区间
1	$x_1(t)$	X71NAO084	87.81°	[87.62°, 88°]
2	$x_2(t)$	4U42FT01	87.34°	[87.02°, 87.66°]
3	$x_3(t)$	4U29FT02B02	85.85°	[85.30°, 86.39°]
4	$x_4(t)$	4HX20CP11	44.72°	[43.23°, 46.21°]
5	$x_5(t)$	X71NAO095	13.82°	[13.13°, 14.52°]
6	$x_6(t)$	4U30LT58	10.32°	[8.62°, 12.01°]

案例1：案例1采用的信号为表3.11 中的 $x_1(t)$ ，描述为#4炉A氨/空气混合器入口稀释风流量，其变量名为X71NAO084，是发电机组脱硫脱硝系统中的变量，对应的单位为 Nm^3/h （标立方米每小时），配置的低报警阈值为 $x_{tp,1} = 1800 Nm^3/h$ ，该信号的报警数量为1112个。

火力发电厂在生产过程中会排放大量的氮氧化物，氮氧化物排放量过高是引发温室效应、酸雨和臭氧层破坏的主要原因，因此减少氮氧化物排放对环保意义重大。脱硝技术是目前火力发电厂中减少氮氧化物排放，改善氮氧化物污染环境问题的重要手段。脱硝是指将已经生成的氮氧化物还原为氮气，该技术主要通过火力发电厂中的脱硫脱硝系统实现。而且国家已经对电厂氮氧化物的排放量进行了严格的规定，所以保障火力发电厂中脱硫脱硝系统的正常运行对电厂来说有重要意义。

本案例中监测变量为氨/空气混合器入口稀释风流量。脱硝过程首先将氨与空气均匀混合，然后喷入发电厂产生的含氮氧化物的烟气中与烟气混合，再进入催化反应器反应，在催化剂的作用下，氨与氮氧化物发生催化氧化还原反应，将烟气中的氮氧化物还原成氮气和水，实现脱硝的功能。在这个过程中对氨的浓度有严格的限制，氨浓度过低则反应不完全，脱硝不彻底；氨浓度过高，则会造成氨逃逸率增加。氨逃逸率过高，会导致如下问题：吹灰过程中吹灰效果差或燃煤灰分高，系统飞灰量偏高，进而导致催化剂中毒，堵塞等，引发逃逸率继续增

图 3.33 过程变量 $x_1(t)$ 部分采样数据

图 3.34 过程变量 $x_1(t)$ 的死区指数及置信区间图 3.35 过程变量 $x_1(t)$ 死区宽度 d 和 $\bar{F}(d)/\bar{m}$ 关系图

高；过度浪费氨，机组运行不经济，也可能发生空预器堵塞。监测氨/空气混合器入口稀释风流量能有效的发现氨气与空气的混合情况，混合不正常时即产生报警。因此对氨/空气混合器入口稀释风流量的监测对脱硫脱硝的正常运行具有十分重要的意义。

根据表3.10中的信号类型“流量”，得到最大报警持续时间和最大报警偏差分别为 $t_{a,max} = 15$ 秒、 $d_{a,max} = 0.05 \cdot x_{tp,1} = 90\text{Nm}^3/\text{h}$ ，报警系统性能要求干扰报警持续时间比不超过5%。图3.33为过程变量 $x_1(t)$ 的部分采样数据。

根据采样信号和低报警阈值计算报警持续时间和报警偏差，并通过式(3.36)和(3.37)对报警持续时间和报警偏差进行标准化，如图3.34所示，计算得到该信号死区指数为 $\hat{\theta} = 87.81^\circ$ ，死区指数置信区间为 $[87.62^\circ, 88^\circ]$ 。通过假设检验判断，拒绝(3.48) 中的 H_0 ，且明显 $\hat{\theta} > 45^\circ$ ，所



图 3.36 过程变量 $x_1(t)$ 延迟 m 和 $F'(m)/m$ 关系图

以该信号适合采用报警死区。

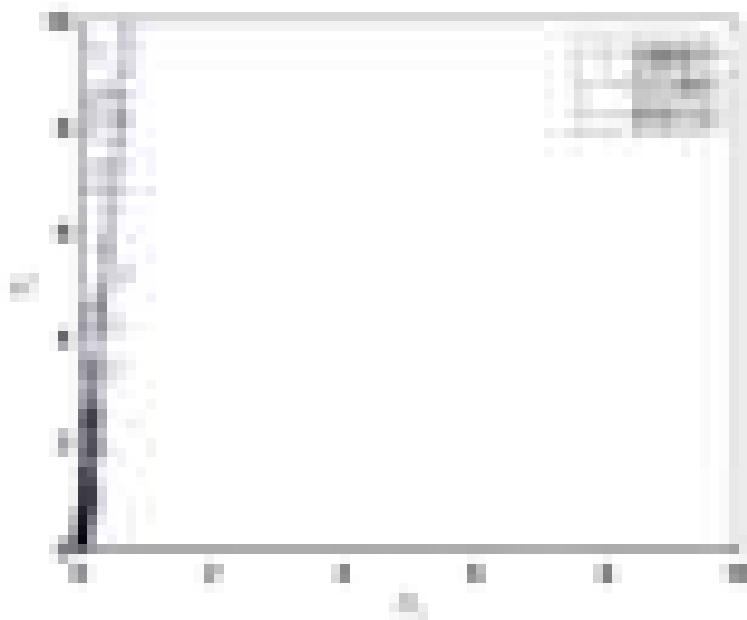
然后根据性能指标和约束条件即可计算死区最优宽度为 $d_{opt} = 9.1 \text{Nm}^3/\text{h}$ 以满足约束 $\bar{F}(d_{opt}) = 0.048 < 0.05$, 相应的延迟估计值为 $\bar{m}(d_{opt}) = 1.517$ 秒, 如图3.35 所示。同样采用报警延迟器作为对比, 计算最优的延迟为 $m_{opt} = 41$ 秒, 以满足性能指标的要求 $F'(m_{opt}) = 0.0487 < 0.05$, 如图3.36 所示。明显的 $\bar{m}(d_{opt}) = 1.517$ 秒远远小于 $m_{opt} = 41$ 秒, 信号 $x_1(t)$ 适合使用报警死区。

案例1根据设计步骤, 首先判断了该过程变量 $x_1(t)$ 适合采用报警死区来减少干扰报警数目, 然后根据报警系统性能指标要求和给出的约束条件, 得到死区宽度和报警延迟的估计值 \bar{m} 。为了增加对比, 我们也对报警延迟器进行了设计, 得到了最优延迟 m_{opt} , 明显 m_{opt} 比报警死区的报警延迟大的多, 说明当对过程变量 $x_1(t)$ 设计报警死区和报警延迟器时, 报警死区性能更好, 这与死区指数结论一致。

案例2: 案例2中的信号为表3.11中的 $x_3(t)$, 描述为B汽泵入口流量, 其变量名为4U29FT02 B02, 是给水泵中的变量, 对应的单位为 m^3/h , 配置的高报警阈值为 $x_{tp} = 560 \text{m}^3/\text{h}$, 该信号的报警数量为258个。该变量主要影响汽泵流量阀调节指令。汽泵全称汽动给水泵, 在锅炉机组稳定运行时控制给水流量, 保证机组正常运行。但是汽动给水泵自动调节性能差, 自动跟踪能力弱, 在机组负荷变化大, 锅炉燃烧调整时, 汽泵调节迟缓, 需要汽机人员及时调整。所以当汽动给水泵出现异常时, 应该及时处理。汽动给水泵的性能影响了电厂发电的稳定性, 而对汽泵入口流量监测可以有效的发现汽动给水泵是否出现异常, 提醒汽机人员及时调整。因此对汽泵入口流量的监测对电厂的正常运行具有十分重要的意义。

根据表3.10中的信号类型“流量”得到最大报警持续时间和最大报警偏差分别为 $t_{a,max} = 15$ 秒、 $d_{a,max} = 0.05 \cdot x_{tp,3} = 28 \text{m}^3/\text{h}$, 报警系统性能要求干扰报警持续时间比不超过5%。图3.37 过程变量 $x_3(t)$ 的部分采样数据。

根据采样信号和低报警阈值计算报警持续时间和报警偏差, 并通过式(3.36)和(3.37) 对报警持续时间和报警偏差进行标准化, 如图3.38所示, 计算得到该信号死区指数为 $\hat{\theta} = 85.85^\circ$, 死区指数置信区间为 $[85.30^\circ, 86.89^\circ]$ 。对于零假设 H_0 , 在显著性水平 $\alpha = 0.05$ 的接受域为 $[44.46^\circ, 45.54^\circ]$, 对于过程变量 $x_3(t)$, 通过假设检验判断, 拒绝(3.48)中的, 且明显 $\hat{\theta} >$

图 3.37 过程变量 $x_3(t)$ 部分采样数据图 3.38 过程变量 $x_3(t)$ 的死区指数及置信区间

45° , 所以该信号适合采用报警死区。

当对过程变量 $x_3(t)$ 设计报警死区时, 其最优死区宽度为 $d_{opt} = 17.24\text{m}^3/\text{h}$ 满足约束条件 $\bar{F}(d_{opt}) = 0.032 < 0.05$, 平均延迟估计值为 $\bar{m}(d_{opt}) = 9.236$ 秒, 小于最大报警延迟 $t_{a,\max} = 15$ 秒, 图3.39为死区宽度 d 与 $\bar{F}(d)/\bar{m}$ 之间的关系曲线。而对过程变量 $x_3(t)$ 设计报警延迟器时, 如图3.40所示, 最优延迟为 $\bar{m}(d_{opt}) = 126$ 秒, 远远大于最大报警延迟 $t_{a,\max} = 15$ 秒。显然该信号适合采用报警死区。

案例3: 案例3中的信号为表3.11中的 $x_4(t)$, 描述为炉声波吹灰器吹灰用气压力, 其变量名为4HX20CP11, 对应的单位为MPa, 配置的低报警阈值为 $x_{tp,4} = 0.55\text{MPa}$, 该信号的报警数量为263个。

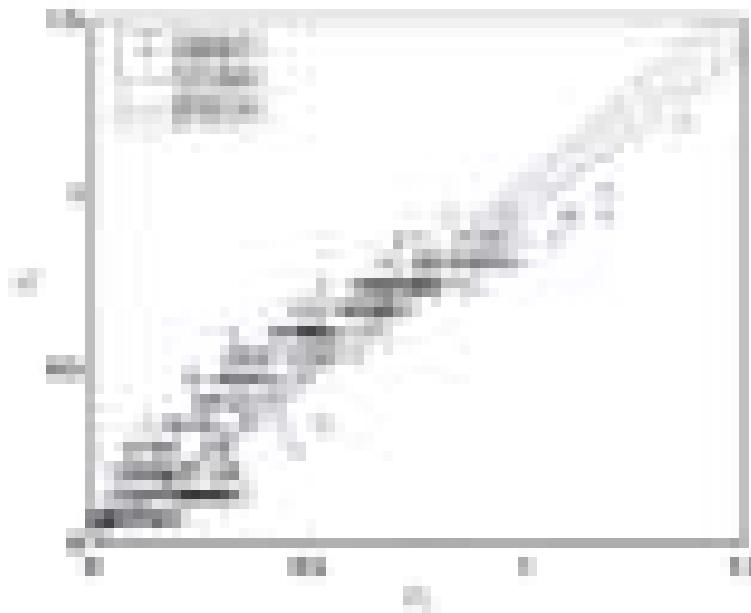
图 3.39 过程变量 $x_3(t)$ 死区宽度 d 和 $\bar{F}(d)/\bar{m}$ 关系图图 3.40 过程变量 $x_3(t)$ 延迟 m 和 $F'(m)/m$ 关系图

在火力发电厂脱硫脱硝系统中，催化剂对脱硝效果具有重要的影响，但是由于燃煤发电的烟气中含有大量的灰尘，在工业现场催化剂面临的主要问题就是堵塞积灰。在停机检查时，经常会发现催化剂上层积灰严重，有时角落积灰可能会达到二十公分以上，催化剂表面积灰清理十分困难，且工作量大。

在日常运行中，常用的避免积灰方法是使用吹灰器，且每层催化剂单侧都会布置有声波吹灰器，声波吹灰器的正常运行可以有效减少积灰。因此对炉声波吹灰器吹灰用气压力的监控对脱硫脱硝的运行具有十分重要的意义。

根据表3.10中的信号类型“压力”得到最大报警持续时间和最大报警偏差分别为 $t_{a,max} = 15$ 秒、 $d_{a,max} = 0.02 \cdot x_{tp,4} = 0.011\text{MPa}$ ，报警系统性能要求干扰报警持续时间比不超过1%。图3.41为过程变量 $x_4(t)$ 的部分采样数据。

基于信号采样数据和报警阈值即可计算过程变量 $x_4(t)$ 的报警持续时间和报警偏差，为了统一比较采用式(3.36) 和(3.37)对其进行标准化，根据标准化结果可计算得到死区指数

图 3.41 过程变量 $x_4(t)$ 部分采样数据图 3.42 过程变量 $x_4(t)$ 的死区指数及置信区间

为 $\hat{\theta} = 44.72^\circ$, 置信区间为 $[43.23^\circ, 46.21^\circ]$, 如图3.42所示。显然, 对于假设检验的零假设 H_0 , 在显著性水平 $\alpha = 0.05$ 下的接受域为 $[43.51^\circ, 46.49^\circ]$, 对于过程变量 $x_4(t)$, 使用报警死区的效果和其他技术差距不大。

我们对该信号同时设计报警死区和报警延迟器, 当满足性能指标时, 最优死区宽度为 $d_{opt} = 0.0104\text{MPa}$, $\bar{F}(d_{opt}) = 0.0088$ 平均报警延迟估计值 $\bar{m}(d_{opt}) = 14.18$ 秒; 作为对比, 报警延迟器的延迟最优值为 $m_{opt} = 12$ 秒、 $F'(m_{opt}) = 0.008$ 。因此, 最优死区宽度和最优延迟具有类似的性能, 与死区指数判断吻合。

案例4: 案例4的信号为表3.11中的 $x_5(t)$, 描述为#4炉B反应区催化剂上层与催化剂后压差, 其变量名为X71NAO095, 是发电机组脱硫脱硝系统中的变量, 对应的单位为Pa, 配置的

图 3.43 过程变量 $x_4(t)$ 死区宽度 d 和 $\bar{F}(d)/\bar{m}$ 关系图图 3.44 过程变量 $x_4(t)$ 延迟 m 和 $F'(m)/m$ 关系图

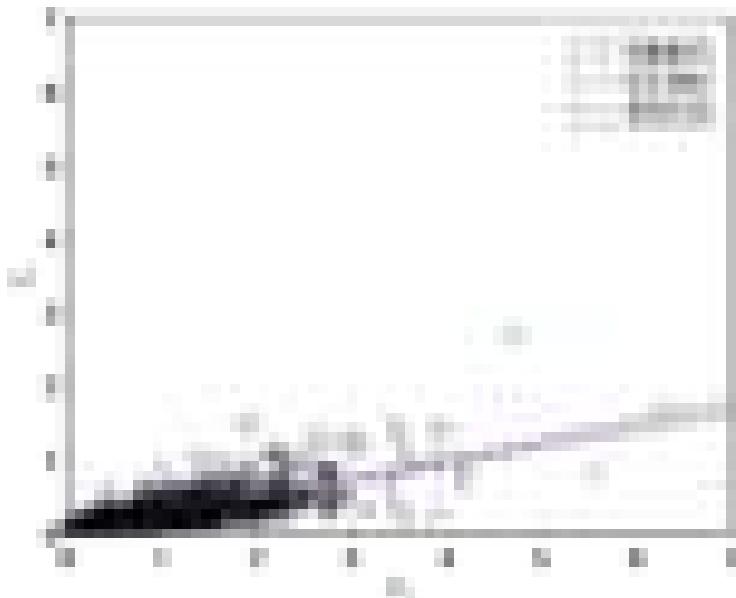
高报警阈值为 $x_{tp,5} = 450 \text{ Pa}$, 该信号的报警数量为 2095 个。

在火力发电厂的脱硫脱硝系统中, 氮氧化物的氧化还原反应离不开催化剂的作用, 催化剂是系统的重要组成部分, 但是催化剂的生产成本非常高。实际脱硫脱硝系统中, 催化剂的寿命一般是三年, 如果现场维护不好或者其他性能得不到满足会大大降低催化剂的寿命, 增加了生产成本, 这是电厂所不希望的。而且当催化剂的性能受到影响时, 会增加氨气消耗量, 降低氮氧化物的转换效率。所以在火力发电厂中, 要加强对脱硫脱硝系统中催化剂的监控和检修维护, 避免大量更换催化剂, 节约成本, 提高电厂的经济效益。

本案例中监测变量为反应区催化剂上层与催化剂后压差, 是监测催化剂性能的一个重要标准, 可以有效地反应催化剂的性能。因此对反应区催化剂上层与催化剂后压差的监测对脱硫脱硝的高效运行具有十分重要的意义。

根据表3.10中的信号类型“压力”得到最大报警持续时间和最大报警偏差分别为 $t_{a,max} = 15$ 秒、 $d_{a,max} = 0.02 \cdot x_{tp,5} = 9 \text{ Pa}$, 报警系统性能要求干扰报警持续时间比不超过5%。图3.45为过程变量 $x_5(t)$ 的部分采样数据。

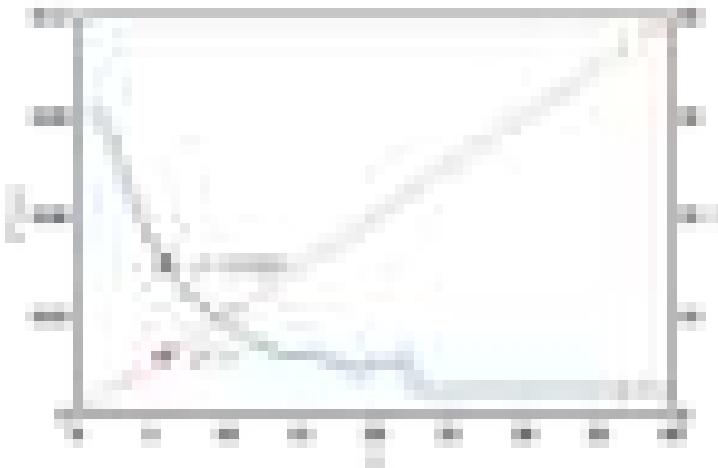
计算过程变量 $x_5(t)$ 的报警持续时间和报警偏差, 为了统一比较采用式(3.36)和(3.37)对其进行

图 3.45 过程变量 $x_5(t)$ 部分采样数据图 3.46 过程变量 $x_5(t)$ 的死区指数及置信区间

行标准化，根据标准化结果可计算得到死区指数为 $\hat{\theta} = 13.82^\circ$ ，置信区间为 $[13.13^\circ, 14.52^\circ]$ ，如图3.46所示。对于零假设 H_0 ，在显著性水平 $\alpha = 0.05$ 下的接受域为 $[44.31^\circ, 45.69^\circ]$ ，对于过程变量 $x_5(t)$ ，其并不适合采用报警死区。

当对过程变量 $x_5(t)$ 设计报警死区时，其最优死区宽度为 $d_{opt} = 0.0104\text{MPa}$ 满足约束条件 $\bar{F}(d_{opt}) = 0.0088$ ，然后平均延迟估计值为 $\bar{m}(d_{opt}) = 14.18$ 秒，大于最大报警延迟 $t_{a,max} = 15$ 秒，图3.47 为死区宽度 d 与 $\bar{F}(d)/\bar{m}$ 之间的关系曲线。而对过程变量 $x_5(t)$ 设计报警延迟器时，如图3.48所示，最优延迟为 $m_{opt} = 6$ 秒。显然过程变量 $x_5(t)$ 并不适合采用报警死区，报警延迟器效果会更好。

案例5：案例5中的信号为表3.11中的 $x_6(t)$ ，描述为#6低加水位，其变量名为4U30LT58，

图 3.47 过程变量 $x_5(t)$ 死区宽度 d 和 $\bar{F}(d)/\bar{m}$ 关系图图 3.48 过程变量 $x_5(t)$ 延迟 m 和 $F'(m)/m$ 关系图

对应的单位为 m , 配置的高报警阈值为 $x_{tp,6} = 38m$ 。该信号的报警数量为 132 个。根据表 3.10 中的信号类型“液位”得到最大报警持续时间和最大报警偏差分别为 $t_{a,max} = 60$ 秒、 $d_{a,max} = 0.05 \cdot x_{tp,6} = 1.9m$, 报警系统性能要求干扰报警持续时间比不超过 1%, 图 3.49 为过程变量 $x_6(t)$ 的部分采样数据。

计算过程变量 $x_6(t)$ 的报警持续时间和报警偏差, 为了比较, 采用式(3.36)和(3.37)对其进行标准化, 根据标准化结果计算得到死区指数 $\hat{\theta} = 10.32^\circ$, 置信区间为 $[8.62^\circ, 12.01^\circ]$, 如图 3.50 所示。通过假设检验判断, 拒绝式(3.48)中的 H_0 , 明显 $\hat{\theta} < 45^\circ$, 因此该信号不适合采用报警死区。

然后根据性能指标和约束条件即可计算死区最优宽度为 $d_{opt} = 5.12m$ 以满足约束 $\bar{F}(d_{opt}) = 0.01$, 相应的延迟估计值为 $\bar{m}(d_{opt}) = 161.68$ 秒, 如图 3.51 所示。同样采用报警延迟器作为对比, 计算最优的延迟为 $m_{opt} = 29$ 秒, 以满足性能指标的要求 $F'(m_{opt}) = 0.0098 < 0.01$, 如

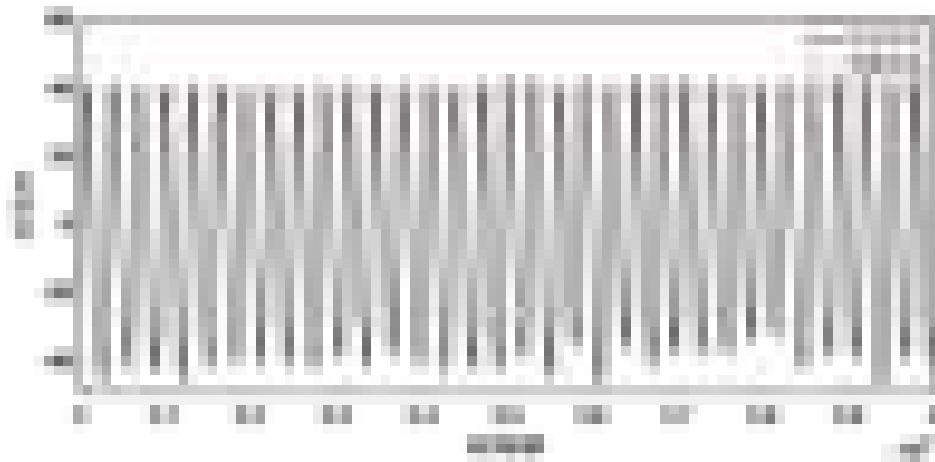
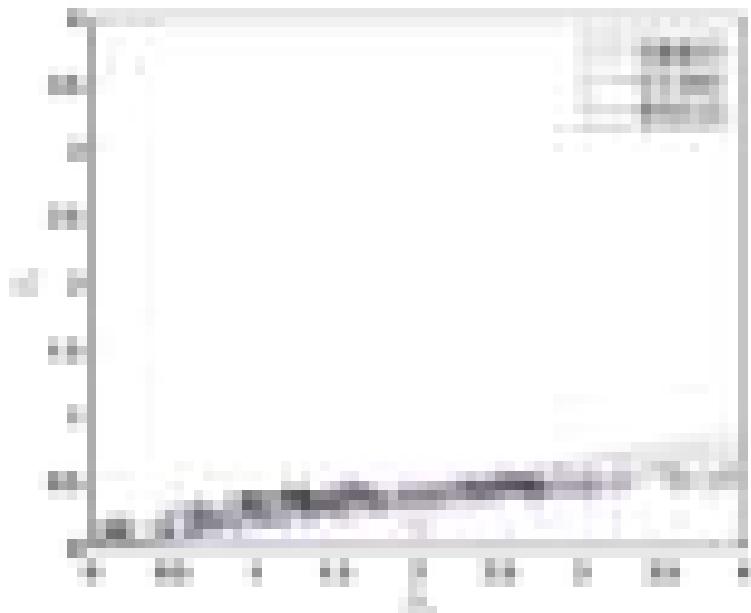
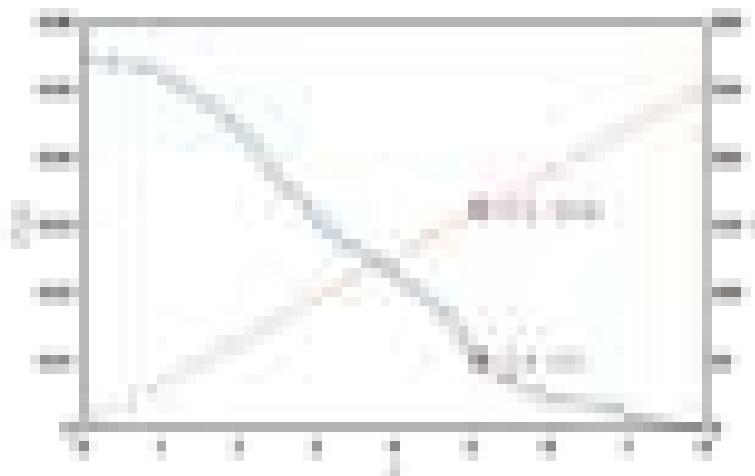
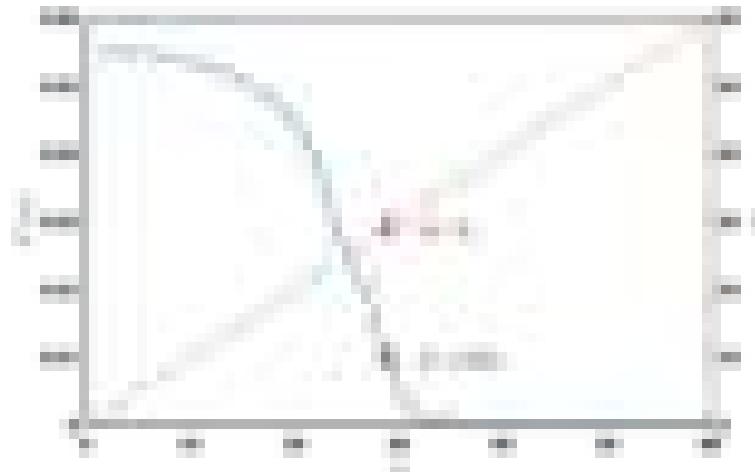
图 3.49 过程变量 $x_6(t)$ 部分采样数据图 3.50 过程变量 $x_6(t)$ 的死区指数及置信区间

表 3.12 使用报警死区后报警数量减少情况

信号	变量名	原报警数量	使用报警死区后报警数量	报警数量减少百分比
$x_1(t)$	X71NAO084	4836	80	98.34%
$x_3(t)$	4U29FT02B02	4866	59	98.79%

图3.52所示。而最大报警持续时间 $t_{a,max} = 60$ 秒，显然 $\bar{m}_{opt} = 161.68$ 秒 > 60 秒，而 $m_{opt} = 29$ 秒 < 60 秒，因此该信号不适合采用报警死区。

图 3.51 过程变量 $x_6(t)$ 死区宽度 d 和 $\bar{F}(d)/\bar{m}$ 关系图图 3.52 过程变量 $x_6(t)$ 延迟 m 和 $F'(m)/m$ 关系图

通过案例分析可知，过程变量 $x_1(t)$ 和 $x_3(t)$ 适合使用报警死区，然后根据案例1和案例3最优死区宽度的设计结果，采用2015年全年的数据对过程变量 $x_1(t)$ 和 $x_3(t)$ 使用报警死区后的报警数量进行了验证，如表3.12所示，结果表明使用报警死区以后，过程变量 $x_1(t)$ 和 $x_3(t)$ 报警数量减少了98%以上，大大减少了报警数量，验证了报警死区设计方法的有效性。

3.5 基于报警概率图的报警阈值设计

本节首先明确报警概率图中4个统计量的数学定义，然后利用马尔科夫链理论来推导它们的计算表达式。

假设过程变量 x 在 t 时刻的取值为 $x(t)$ ，它在 t 时刻对应的状态变量为 $x_s(t)$ 。状态变量的取值是离散的，如果一共有 n 个状态，那么它的取值范围就是 $\{1, 2, \dots, n\}$ 。假设状态 i 对应的量化

区间是 $[x_i, x_{i+1})$, 那么当 $x(t) \in [x_i, x_{i+1})$, $x_s(t) = i$ 。第一个报警状态被定义为 a , 也就是说, 当 $x(t) \in [x_a, x_{a+1})$, $x_s(t) = a$ 。此外, x_a 等于报警阈值 x_{tp} 。取值小于 a 的状态被定义为报警限下状态, 取值大于 a 的状态被定义为报警限上状态。基于状态变量, 报警概率图中4个统计量的数学定义分别为:

- 过程变量从某个报警限下状态 k 单调不减的转移到报警状态 a 的概率 P_{ka} ;
- 过程变量从某个报警限下状态 k 单调不减的转移到报警状态 a 的时间 T_{ka} ;
- 过程变量从报警状态首次转移到某个报警限上状态 m 的概率 P_{am} 。
- 过程变量从报警状态首次转移到某个报警限上状态 m 的时间 T_{am} 。

状态变量 $x_s(t)$ 被假设具有马尔科夫性, 这在工业上是一个很容易满足的假设, 可以通过假设检验来验证^[19]。状态变量 $x_s(t)$ 对应的马尔科夫链可以通过一步转移概率 $p_{ij}^{(1)}$ 来描述,

$$p_{ij}^{(1)} = \Pr[x_s(t+1) = j | x_s(t) = i], i, j \in V, \quad (3.69)$$

其中, $V = \{1, 2, \dots, v\}$ 是包含所有状态的集合。

3.5.1 P_{ka} 和 T_{ka} 的计算表达式

首先, 假设 $P_{i(i+1)}$ 表示 $x(t)$ 从状态 i 单调不减的转移到状态 $i+1$ 的概率。如果这个转移过程包含 l 步, 那么 l 步转移概率 $p_{i(i+1)}^{(l)}$ 的计算公式为

$$p_{i(i+1)}^{(l)} = \left(p_{ii}^{(1)}\right)^{l-1} p_{i(i+1)}^{(1)}.$$

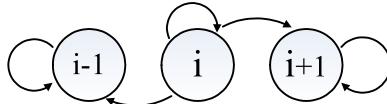
因为转移过程是单调不减的, 所以前 $l-1$ 步都停留在状态 i , 最后一步才从状态 i 转移到状态 $i+1$ 。显然, $P_{i(i+1)}$ 是所有 l 步转移概率之和, 即

$$\begin{aligned} P_{i(i+1)} &= \sum_{l=1}^{\infty} p_{i(i+1)}^{(l)} \\ &= \sum_{l=1}^{\infty} \left(p_{ii}^{(1)}\right)^{l-1} p_{i(i+1)}^{(1)} \\ &= \frac{p_{i(i+1)}^{(1)}}{1 - p_{ii}^{(1)}}. \end{aligned}$$

由于状态变量具有马尔科夫性, 所以 $P_{i(i+1)}$ 与 $P_{(i-1)i}$ 和 $P_{(i+1)(i+2)}$ 独立。因此, P_{ka} 是从状态 k 到状态 $a-1$ 的所有转移概率的乘积, 即

$$P_{ka} = \prod_{i=k}^{a-1} \frac{p_{i(i+1)}^{(1)}}{1 - p_{ii}^{(1)}}. \quad (3.70)$$

马尔科夫链中的状态可以被分为两类: 暂态和稳态。如果一个状态经过若干步的转移后依然会回到自身, 那么这个状态被定义为稳态。如果一个状态不是稳态, 那么它就是暂态。并

图 3.53 从状态 i 到状态 $i + 1$ 的状态转换图。

且, 如果一个稳态满足如下条件

$$\begin{cases} p_{ss}^{(1)} = 1 \\ p_{sj}^{(1)} = 0, j \neq s, \end{cases}$$

那么它被定义为吸收态。假设 T_{is} 表示从状态 i 转移到吸收态 s 的转移时间, 那么 T_{is} 是以下线性方程组的解^[20],

$$\begin{cases} T_{is} = 0, \text{ 如果 } i \text{ 为吸收态;} \\ T_{is} = 1 + \sum_{j \in V} p_{ij}^{(1)} T_{js}, \text{ 如果 } i \text{ 为暂态.} \end{cases} \quad (3.71)$$

假设 $T_{i(i+1)}$ 表示从状态 i 单调不减的转移到状态 $i + 1$ 时间, 并且从状态 i 到状态 $i + 1$ 的状态转换图如图3.53 所示。

基于式(3.71)和图3.53可以得到

$$\begin{bmatrix} 1 & 0 & 0 \\ p_{i(i-1)}^{(1)} & p_{ii}^{(1)} - 1 & p_{i(i+1)}^{(1)} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_{(i-1)(i+1)} \\ T_{i(i+1)} \\ T_{(i+1)(i+1)} \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}. \quad (3.72)$$

式(3.72)表示一个含有3个变量和3个方程的线性方程组, 利用高斯消元法进行求解, 可以得到 $T_{i(i+1)}$ 的计算公式为

$$T_{i(i+1)} = \frac{1}{1 - p_{ii}^{(1)}}.$$

由于状态变量的马尔科夫性, T_{ka} 是从状态 k 到状态 $a - 1$ 的所有转移时间之和, 因此

$$T_{ka} = \sum_{i=k}^{a-1} \frac{1}{1 - p_{ii}^{(1)}}. \quad (3.73)$$

3.5.2 P_{am} 和 T_{am} 的计算表达式

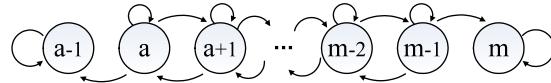
首先, 假设 P_{is} 表示过程变量从状态 i 出发, 最终转移到吸收态 s 的概率, 其计算公式为^[20]

$$\begin{cases} P_{ss} = 1, \\ P_{is} = 0, i \text{ 为吸收态, 并且 } i \neq s. \\ P_{is} = \sum_{j \in V} p_{ij}^{(1)} P_{js}, i \text{ 为暂态} \end{cases} \quad (3.74)$$

根据 P_{am} 的定义可知, 从报警状态 a 转移到报警限上状态 m 的过程不需要是单调不减的, 该转移过程的状态转换图, 如图3.54所示。

根据图3.54和式(3.74)可以得到

$$PA = C, \quad (3.75)$$

图 3.54 从报警状态 a 到报警限上状态 m 的状态转换图

其中，矩阵 P 的具体形式为

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ p_{a(a-1)}^{(1)} & p_{aa}^{(1)} - 1 & p_{a(a+1)}^{(1)} & 0 & \cdots & 0 & 0 & 0 \\ 0 & p_{(a+1)a}^{(1)} & p_{(a+1)(a+1)}^{(1)} - 1 & p_{(a+1)(a+2)}^{(1)} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & p_{(m-1)(m-2)}^{(1)} & p_{(m-1)(m-1)}^{(1)} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix},$$

向量 A 的具体形式为

$$A = [P_{(a-1)m}, P_{am}, P_{(a+1)m}, \dots, P_{(m-1)m}, P_{mm}]^T,$$

向量 C 的具体形式为

$$C = [0, 0, 0, \dots, 0, 1]^T.$$

根据式(3.75)，可以得到 P_{am} 的计算公式为

$$P_{am} = (P^{-1}C)_2, \quad (3.76)$$

其中，下标“2”表示向量 $P^{-1}C$ 的第2个元素。

接下来，结合图3.54和式(3.71)可以得到

$$PB = D, \quad (3.77)$$

其中， P 的具体形式与式(3.75)中的相同，而向量 B 的具体形式为

$$B = [T_{(a-1)m}, T_{am}, \dots, T_{(m-1)m}, T_{mm}]^T.$$

向量 D 的具体形式为

$$D = [0, -1, \dots, -1, 0]^T.$$

因此，从报警状态 a 转移到报警限上状态 m 的计算公式为

$$T_{am} = (P^{-1}D)_2, \quad (3.78)$$

其中，下标“2”同样是表示向量 $P^{-1}D$ 的第2个元素。

3.5.3 确定状态变量

为了估计报警概率图中4个统计量的值，需要将过程变量离散化为状态变量。假设取值小于报警阈值 x_{tp} 的样本被划分为 N_1 个状态，其他样本被划分为 N_2 个状态。我们通过古典概率来估计 N_1 和 N_2 的值。

首先，由式(3.70)、(3.73)、(3.76)和(3.78)可知，报警概率图中4个统计量的计算都需要用到一步转移概率 $p_{ij}^{(1)}$ 。在实际应用过程中， $p_{ij}^{(1)}$ 的值是从历史数据中估计得到，它的计算公式为

$$\hat{p}_{ij}^{(1)} = \frac{n_{ij}}{n_i}, j = i - 1, i, i + 1. \quad (3.79)$$

其中， n_i 表示状态*i*出现的次数， n_{ij} 表示状态*i*出现后状态*j*紧接着出现的次数。由式(3.79)可知，每个状态包含的样本数不能太小，否则就会导致 $p_{ij}^{(1)}$ 的估计值不可靠。与此同时，状态的个数应该足够的大，这样报警阈值就会落在一个取值范围比较小的区间内，从而方便我们找到最优的报警阈值。一个朴素的想法是这样的，每个状态都应该包含足够数目的样本，并且当过程变量的取值密集时，状态变量对应的区间范围应该狭窄。算法1详细阐述了这种思想，其中每个状态应该包含的最小样本数 z 是可调节变量。

Algorithm 1 确定状态变量的区间范围

```

 $x(t)$ , 过程变量
 $x_{tp}$ , 报警阈值
 $z$ , 每个状态包含的最小样本数
 $L_1=\text{sum}(x(t) < x_{tp})-1;$ 
 $L_2=\text{sum}(x(t) \geq x_{tp})+1;$ 
 $N_1=\text{floor}(L_1/z);$ 
 $N_2=\text{floor}(L_2/z)$ 
 $x_o(t)=\text{sort}(x(t));$ 
 $t = 1;$ 
for  $t = 1 \rightarrow L_1 + L_2$  do
    for  $k = 1 \rightarrow N_1 - 1$  do
        if  $x_o(1) \leq x(t) \leq x_o(L_1 - (N_1 - 1)z)$  then
             $x_s(t) = 1;$ 
        end if
        if  $x_o(L_1 - (N_1 - k)z + 1) \leq x(t) \leq x_o(L_1 - (N_1 - 1 - k)z)$  then
             $x_s(t) = k + 1;$ 
        end if
    end for
    for  $t = 1 \rightarrow L_1 + L_2$  do
        if  $x_o((k - 1)z + 1 + L_1) \leq x(t) \leq x_o(kz + L_1)$  then
             $x_s(t) = k + N_1;$ 
        end if
        if  $x_o((N_2 - 1)z + 1 + L_1) \leq x(t) \leq x_o(L_1 + L_2)$  then
             $x_s(t) = N_1 + N_2;$ 
        end if
    end for
end for
end for

```

数值仿真算法被用来确定每个状态包含的最小样本数 z 的值。过程变量的数据由一阶自回

归模型生成: $x(t) = 4.5 + 0.1x(t-1) + \eta(t)$, 其中 $\eta(t)$ 表示均值为0方差为4的高斯白噪声。这些数据被划分为10个状态, 当 $x(t) < 1$ 时, $x_s(t) = 1$; 当 $1 \leq x(t) < 2$ 时, $x_s(t) = 2$; \dots ; 当 $8 \leq x(t) < 9$ 时, $x_s(t) = 9$; 当 $x(t) \geq 9$ 时, $x_s(t) = 10$ 。不失一般性, 状态8被定义为报警状态。对于每个确定的 z 都进行100次蒙特卡洛仿真实验, 估计得到的 $P_{48}/E(P_{48})$ 和 $T_{48}/E(T_{48})$ 随 z 的变化趋势如图3.55所示。可以发现, 当 z 的值不小于600时, $P_{48}/E(P_{48})$ 和 $T_{48}/E(T_{48})$ 的估计值都在它们的均值附近变化, 并且变化幅度不超过均值的 $\pm 5\%$ 。因此, 只要 z 的值不小于600, 利用古典概率从历史数据中估计报警概率图的4个统计量的值就是可靠的。

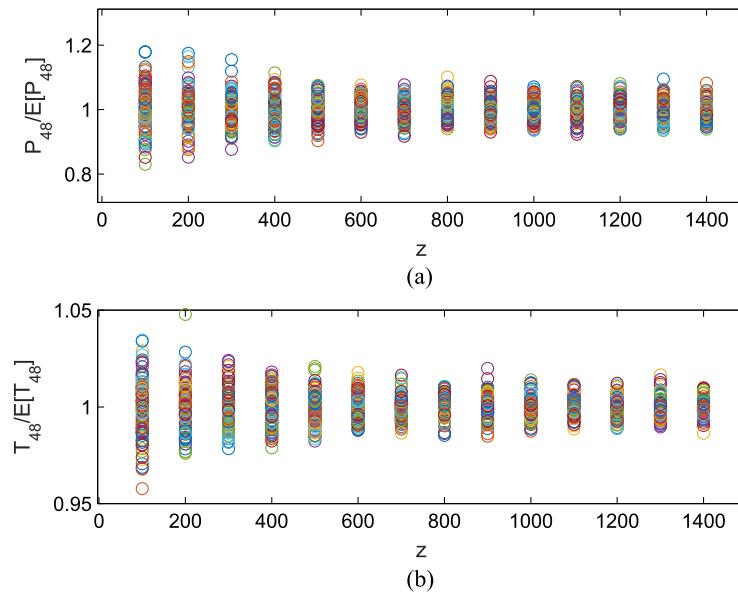


图 3.55 不同 z 值对应的转移概率和转移时间

其次, 需要注意的是, 如果状态 k 单调不减的转移到报警状态 a 的次数过少, 那么就会导致 P_{ka} 的值过小。因此, P_{ka} 的值被要求不能小于0.001。之所以选择0.001, 是因为如果一个事件发生的概率小于0.001, 这个事件就被认为是小概率事件。由式(3.70)可知, P_{ka} 是一个关于 k 的单调不减函数, 因此 P_{ka} 的最小值为 P_{1a} 。即 P_{1a} 的值不能小于0.001, 这可以通过调整报警限以下的状态数 N_1 的值来实现。将过程变量转化为状态变量的具体步骤如算法2所示。

方法2 将过程变量 $x(t)$ 转换成状态变量 $x_s(t)$

1. 通过将每个状态的样本数设为600来得到 N_1 的初始值 n_0 , 并且通过算法1 来得到每个状态对应的量化区间;
2. 计算 P_{1a} 的值, 如果小于0.001, 就令 $N_1 = N_1 - 1$;
3. 重复步骤 (1) 和 (2) 直到 $P_{1a} \geq 0.001$ 。

3.5.4 计算报警阈值

本小节详细介绍如何利用报警概率图来设计最优的报警阈值。对于一个给定的报警阈

值 x_{tp} , 可以根据上一节推导得到的公式计算报警概率图中各个状态的转移概率和转移时间, 然后构建决策函数

$$F(P_i, T_i) = \alpha P_i + (1 - \alpha) \frac{T_i}{\max(T_i)}. \quad (3.80)$$

其中, P_i 表示状态*i*对应的转移概率 P_{ia} 或者 P_{ai} , T_i 表示状态*i*对应的转移时间 T_{ia} 或者 T_{ai} , α 表示权重系数, 它的取值在0到1之间。一般情况下, α 的值被建议设为0.5。也就是说, P_i 和 T_i 的取值对决策函数的最终取值的影响同等重要。变量 P_i 的值越大, 就意味着过程变量越容易从报警限下状态*i*转移到报警状态, 或者越容易从报警状态转移到报警限上状态*i*。变量 T_i 的值越大, 就意味着操作人员有更多的时间对已经发生的报警或者将要发生的报警做出反应。因此, 最佳的报警阈值要使得决策函数式(3.80)的取值最大。也就是说,

$$x_{tp}^{(\text{opt})} = x_{i^{(\text{opt})}}. \quad (3.81)$$

其中, $x_{i^{(\text{opt})}}$ 是量化区间 $[x_{i^{(\text{opt})}}, x_{i^{(\text{opt})}+1})$ 的左边界, 并且区间*i*会使 $F(P_i, T_i)$ 的值最大化, 即

$$i^{(\text{opt})} = \operatorname{argmax} F(P_i, T_i).$$

由报警概率图中4个统计量的数学定义可知, 报警阈值不同, P_i 和 T_i 的取值也不同, 从而计算得到的报警概率图也不同。式(3.80)是利用一张报警概率图来设计最优的报警阈值, 也就是利用报警概率图来对已有的报警阈值进行优化。如果事先不知道报警阈值, 可以根据经验知识和过程知识来确定报警阈值的取值集合。该集合中的每一个元素都是报警阈值的一个可行解, 也都对应着一张报警概率图。对于有多张报警概率图的情况, 式(3.80)被更新为

$$F(P_{i,x_{tp}}, T_{i,x_{tp}}) = \alpha P_{i,x_{tp}} + (1 - \alpha) \frac{T_{i,x_{tp}}}{T_{\max}}. \quad (3.82)$$

其中, $P_{i,x_{tp}}$ 和 $T_{i,x_{tp}}$ 分别表示报警阈值 x_{tp} 对应的状态*i*的转移概率和转移时间, T_{\max} 表示 $T_{i,x_{tp}}$ 的最大值, 也就是所有报警概率图中各个状态的转移时间的最大值。在实际应用过程中, 不同的报警阈值对应的决策函数 $F(P_{i,x_{tp}}, T_{i,x_{tp}})$ 的最大值有时候会很接近。为了应对这种情况, 各个状态对应的决策函数的平均值被用来计算最优的报警阈值, 即

$$x_{tp}^{(\text{opt})} = \operatorname{argmax}_{x_{tp}} \overline{F(P_{i,x_{tp}}, T_{i,x_{tp}})}, \quad (3.83)$$

其中,

$$\overline{F(P_{i,x_{tp}}, T_{i,x_{tp}})} = \frac{\sum_i F(P_{i,x_{tp}}, T_{i,x_{tp}})}{N_{1,x_{tp}} + N_{2,x_{tp}}},$$

这里 $N_{1,x_{tp}}$ 表示报警阈值 x_{tp} 对应的报警限下的状态个数, $N_{2,x_{tp}}$ 表示报警阈值 x_{tp} 对应的报警限上的状态个数。

对于给定的历史数据 $\{x(t)\}_{t=1}^N$, 利用报警概率图设计最优报警阈值的详细步骤总结如下:

1. 通过经验知识和过程知识选择一个合适的 x_{tp} , 再根据算法1和算法2将过程变量 $x(t)$ 离散化为状态变量 $x_s(t)$;
2. 对状态变量进行平滑, 从而保证状态变量只在相邻状态之间转移。也就是说, 如果 $x_s(t) = i$ 并且 $x_s(t+1) \leq i - 1$, 那么就将 $x_s(t+1)$ 的值设为 $i - 1$; 如果 $x_s(t) = i$ 并且 $x_s(t+1) \geq i + 1$, 那么就将 $x_s(t+1)$ 的值设为 $i + 1$;

表 3.13 各个状态对应的 P_i 、 T_i 和 $F(P_i, T_i)$ 的值

状态 <i>i</i>	转移概率 P_i	转移时间 T_i	决策函数 $F(P_i, T_i)$
1	0.12	10.22	0.24
2	0.12	9.33	0.23
3	0.20	7.93	0.24
4	0.33	6.13	0.27
5	0.98	5.13	0.58
6	0.98	4.13	0.57
7	0.98	2.01	0.53
8	0.95	0.00	0.47
9	0.47	16.55	0.53
10	0.12	27.91	0.56

3. 从给定的历史数据中估计一步转移概率 $p_{ij}^{(1)}$ 的值;
4. 利用式(3.70)、(3.73)、(3.76)和(3.78)分别计算 P_{ka} 、 T_{ka} 、 P_{am} 和 T_{am} 的值;
5. 利用式(3.81)优化报警阈值, 或者重复步骤1到4, 再利用式(3.83)计算最优的报警阈值。

3.5.5 案例分析

本节通过一个仿真案例和一个工业案例具体阐述如何利用报警概率图设计单变量器的报警阈值, 以及验证该方法的有效性。

案例1: 本案例复现了技术报告^[21] 中展示的报警概率图, 并详细介绍如何利用一张报警概率图优化单变量器的报警阈值。考虑图3.56中的报警概率图, 状态8被定义为报警状态。表3.13列举了各个状态对应的转移概率 P_i 、转移时间 T_i 和决策函数 $F(P_i, T_i)$ 的值。

通过式(3.81)可知, 状态5对应着最优的报警阈值, 这与通过图3.56观察得到的结果是一致的。如果将报警阈值设在状态4的左端点处, 那么当过程变量达到报警状态4时, 它将会以0.67的概率返回到报警限以下的状态。由于不需要操作人员采取动作, 过程变量就会以较大的概率回到正常状态, 因此这种情况会产生较多的误报警。相比之下, 状态5拥有更大的转移概率 $P_5 = 0.98$ 。也就是说, 当过程变量达到状态5以后, 它会以非常大的概率继续增大到某个更大的值。因此, 如果将报警阈值设在状态5的左端点处, 报警系统产生误报警的概率就只有0.02。因为过程变量到达报警状态5以后, 它只会以0.02的概率自动回到报警限以下的状态。类似的, 如果将报警阈值设在状态6或者状态7的左端点处, 报警系统产生误报警的概率也只有0.02, 因为状态6和状态7的转移概率也都等于0.98。将报警阈值设在状态5的左端点处, 可以降低报警系统产生误报警的概率。除此之外, 状态5的转移时间要大于状态6和7的转移时间。也就是说将报警阈值设在状态5的左端点处, 而不是状态6和7的左端点处, 现场工作人员会有更多的时间对产生的报警做出反应。综上所述, 将报警阈值从状态8的左端点处更新到状态5的左端点处, 报警系统并不会产生更多的误报警, 而现场工作人员会有更多的时间来处理已经发

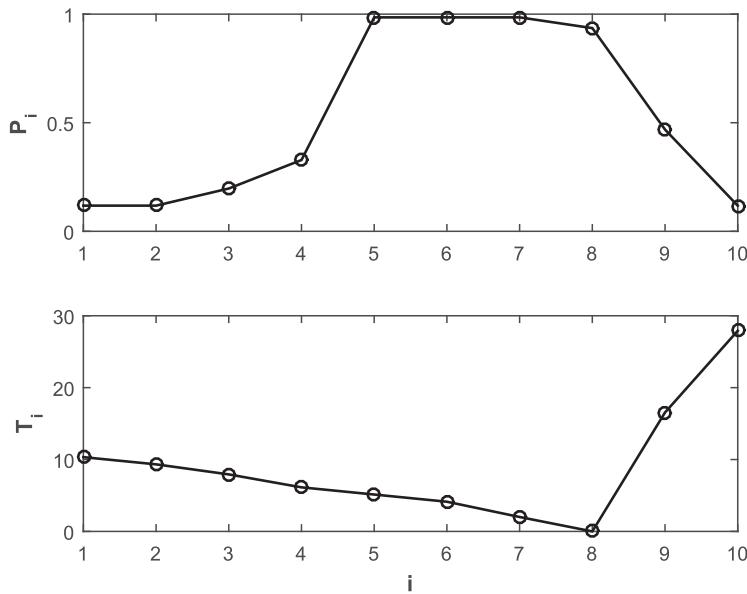


图 3.56 案例1中的报警概率图

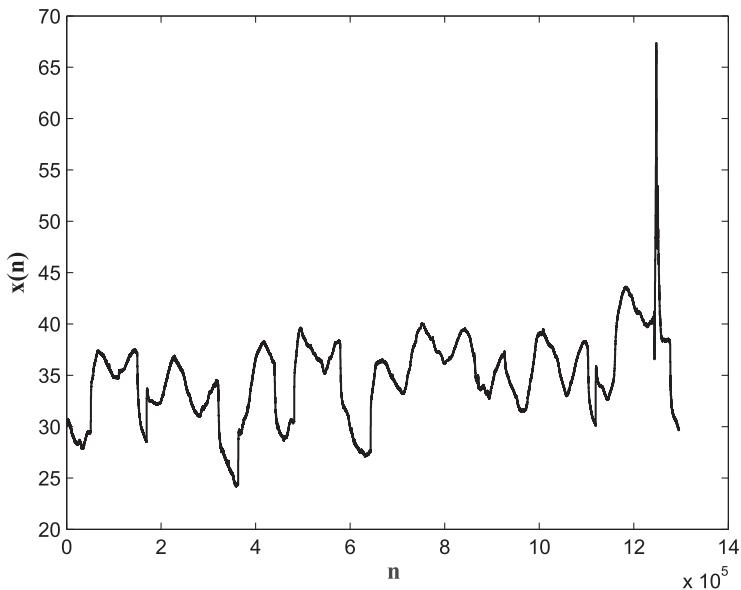
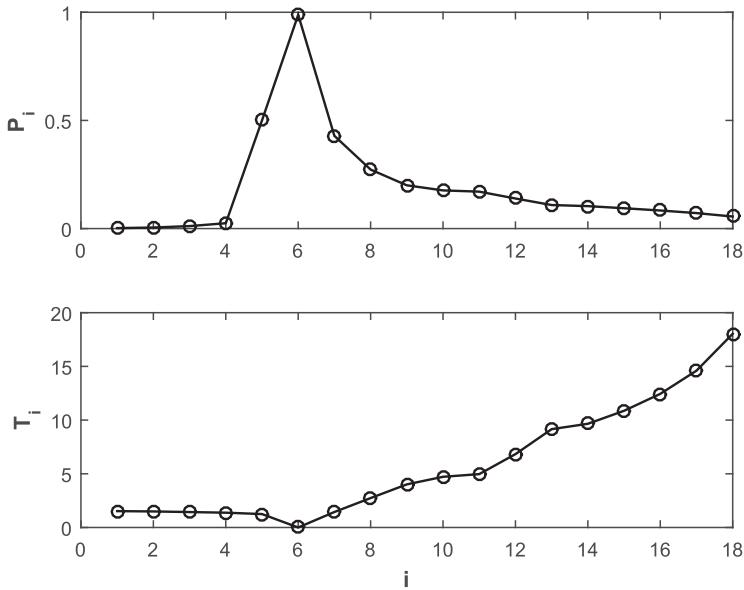
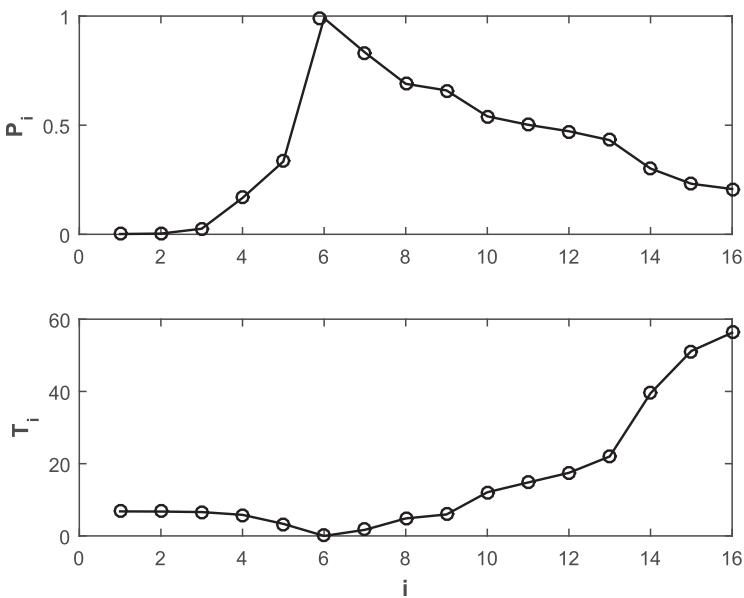


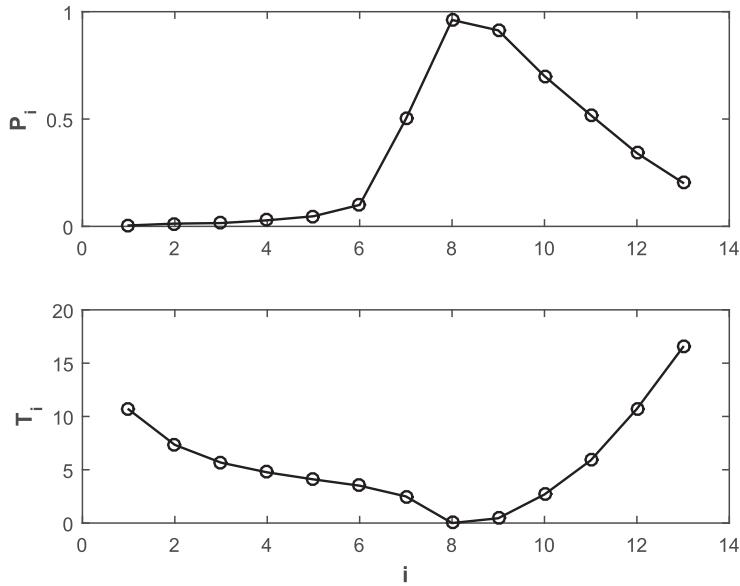
图 3.57 案例2中过程变量的时间序列图

生的报警。

案例2: 本案例利用中国山东某大型发电厂的数据来介绍如何在实际生产过程中应用报警概率图。发电厂的磨煤机会涉及到很多个可测量的过程变量，其中一个过程变量是非驱动端轴承的温度 $x(t)$ 。如果它的值大于 55°C ，磨煤机就会自动停止运行，以防止造成生产安全事故。因此，为 $x(t)$ 设计合理的报警阈值是非常必要的。图3.57展示了 $x(t)$ 在15天内的时间序列图，其

图 3.58 案例2中报警阈值 $x_{tp} = 45$ 对应的报警概率图图 3.59 案例2中报警阈值 $x_{tp} = 47$ 对应的报警概率图

中数据的采样周期为1s。显然，图3.57中的过程变量不是独立同分布的，所以无法应用那些需要过程变量满足独立同分布条件的传统方法来设计 $x(t)$ 的报警阈值 x_{tp} 。如果 x_{tp} 的取值过小，那么当报警发生后， $x(t)$ 会自动回到报警限以下，因此会产生大量的误报警；如果 x_{tp} 的取值过大，就会发生生产过程出现异常一段时间以后才会有报警发生，从而导致现场工作人员没有足够的时间对发生的报警做出反应。

图 3.60 案例2中报警阈值 $x_{tp} = 51$ 对应的报警概率图。

通过调整报警阈值的取值，可以得到不同的报警概率图。我们详细展示3个有代表性的报警阈值对应的报警概率图，分别是当报警阈值等于45，47和51时。这3个报警阈值对应的报警概率图分别如图3.58、图3.59和图3.60 所示，它们对应的各个状态的转移概率 $P_{i,x_{tp}}$ 、转移时间 $T_{i,x_{tp}}$ 和决策函数 $F(P_{i,x_{tp}}, T_{i,x_{tp}})$ 的值分别列举在表3.14、表3.15 和表3.16中。由这3个表的最后1列数据可知，3个报警阈值对应的决策函数的最大值都很接近。因此，各个状态的决策函数 $F(P_{i,x_{tp}}, T_{i,x_{tp}})$ 的平均值被用来挑选最优的报警阈值。当 x_{tp} 分别等于45、47和51时， $F(P_{i,x_{tp}}, T_{i,x_{tp}})$ 的平均值分别为0.1481、0.3414和0.2172。根据式(3.83)可以知道，最优的报警阈值 $x_{tp}^{(opt)} = 47$ 。

通过报警概率图验证最优报警阈值 $x_{tp} = 47$ 的有效性。由图3.58可知，当 $x(t)$ 的值达到报警阈值 $x_{tp} = 45$ 时，过程变量会以 $P_{i=7,x_{tp}=45} = 0.4286$ 概率继续增大。也就是说，过程变量大概率会回到报警限以下，而不是继续增大到某个比报警阈值更大的值。因此，将报警阈值设为45会导致大量的误报警。在图3.59和3.60中，无论报警阈值的取值是47还是51，当过程变量达到报警阈值后，它都会以很大的概率继续增大。例如，当 $x_{tp} = 47$ 时， $x(t)$ 继续增大的概率是 $P_{i=7,x_{tp}=47} = 0.8333$ ；当 $x_{tp} = 51$ 时， $x(t)$ 继续增大的概率是 $P_{i=9,x_{tp}=51} = 0.9130$ 。不过，如果将报警阈值的值设为47，而不是51，现场工作人员会有更多的时间对发生的报警做出反应。

表 3.14 当报警阈值 $x_{tp} = 45$ 时各个状态对应的 $P_{i,x_{tp}}$ 、 $T_{i,x_{tp}}/T_{\max}$ 和 $F(P_{i,x_{tp}}, T_{i,x_{tp}})$ 的值

状态 i	量化区间	$P_{i,x_{tp}}$	$T_{i,x_{tp}}/T_{\max}$	$F(P_{i,x_{tp}}, T_{i,x_{tp}})$
1	[43.51, 43.53)	0.0021	0.0075	0.0048
2	[43.53, 43.55)	0.0048	0.0073	0.0061
3	[43.55, 43.58)	0.0115	0.0071	0.0093
4	[43.58, 43.86)	0.0247	0.0068	0.0157
5	[43.86, 45.00)	0.5000	0.0061	0.2531
6	[45.00, 46.13)	0.9883	0.0000	0.4941
7	[46.13, 47.34)	0.4286	0.0071	0.2178
8	[47.34, 48.10)	0.2727	0.0135	0.1431
9	[48.10, 48.76)	0.2000	0.0197	0.1099
10	[48.76, 49.22)	0.1765	0.0232	0.0998
11	[49.22, 50.33)	0.1707	0.0244	0.0976
12	[50.33, 51.47)	0.1391	0.0336	0.0863
13	[51.47, 52.24)	0.1088	0.0450	0.0769
14	[52.24, 53.03)	0.1038	0.0475	0.0756
15	[53.03, 54.66)	0.0940	0.0534	0.0737
16	[54.66, 57.93)	0.0836	0.0612	0.0724
17	[57.93, 62.46)	0.0716	0.0718	0.0717
18	[62.46, 67.35)	0.0557	0.0886	0.0722

表 3.15 当报警阈值 $x_{tp} = 47$ 时各个状态对应的 $P_{i,x_{tp}}$ 、 $T_{i,x_{tp}}/T_{\max}$ 和 $F(P_{i,x_{tp}}, T_{i,x_{tp}})$ 的值

状态 i	量化区间	$P_{i,x_{tp}}$	$T_{i,x_{tp}}/T_{\max}$	$F(P_{i,x_{tp}}, T_{i,x_{tp}})$
1	[43.54, 43.57)	0.0014	0.0334	0.0174
2	[43.57, 43.61)	0.0034	0.0331	0.0183
3	[43.61, 44.71)	0.0256	0.0324	0.0290
4	[44.71, 45.89)	0.1667	0.0287	0.0977
5	[45.89, 47.00)	0.3333	0.0164	0.1748
6	[47.00, 47.91)	0.9900	0.0000	0.4950
7	[47.91, 48.57)	0.8333	0.0083	0.4208
8	[48.57, 49.15)	0.6897	0.0238	0.3567
9	[49.15, 49.98)	0.6593	0.0292	0.3442
10	[49.98, 51.10)	0.5405	0.0594	0.2999
11	[51.10, 52.16)	0.5018	0.0727	0.2873
12	[52.16, 52.77)	0.4722	0.0858	0.2790
13	[52.77, 53.96)	0.4314	0.1081	0.2698
14	[53.96, 57.03)	0.3014	0.1941	0.2478
15	[57.03, 60.98)	0.2316	0.2516	0.2416
16	[60.98, 67.35)	0.2076	0.2765	0.2420

表 3.16 当报警阈值 $x_{tp} = 51$ 时各个状态对应的 $P_{i,x_{tp}}$ 、 $T_{i,x_{tp}}/T_{\max}$ 和 $F(P_{i,x_{tp}}, T_{i,x_{tp}})$ 的值

状态 i	$P_{i,x_{tp}}$	$T_{i,x_{tp}}/T_{\max}$	$F(P_{i,x_{tp}}, T_{i,x_{tp}})$	
1	[44.40, 45.59)	0.0043	0.0524	0.0284
2	[45.59, 46.69)	0.0128	0.0360	0.0244
3	[46.69, 47.76)	0.0154	0.0278	0.0216
4	[47.76, 48.41)	0.0282	0.0234	0.0258
5	[48.41, 48.65)	0.0471	0.0201	0.0336
6	[48.65, 49.03)	0.1000	0.0172	0.0586
7	[49.03, 51.00)	0.5000	0.0123	0.2561
8	[51.00, 52.10)	0.9815	0.0000	0.4907
9	[52.10, 52.54)	0.9130	0.0022	0.4576
10	[52.54, 53.41)	0.7000	0.0132	0.3566
11	[53.41, 56.14)	0.5185	0.0289	0.2737
12	[56.14, 59.76)	0.3415	0.0526	0.1971
13	[59.76, 67.35)	0.2029	0.0812	0.1420
14	[53.96, 57.03)	0.3014	0.1941	0.2478
15	[57.03, 60.98)	0.2316	0.2516	0.2416
16	[60.98, 67.35)	0.2076	0.2765	0.2420

3.6 干扰报警的识别与消除

3.6.1 抖动报警

抖动报警指过程变量在报警状态和非警报状态之间快速转换的报警。下文涉及报警持续时间、报警间隔时间和游程长度这三个指标，其定义在第2章中已给出，分别用符号 T_1 、 T_0 和 r 表示。检测抖动报警的规则为^[22]：

规则1. 如果报警持续时间 T_1 或报警间隔时间 T_0 小于20s时，则发生抖动报警。

实际上，20s的时间对于工厂的操作员来说太短了，无法对报警的发生或清除做出任何响应。阈值20s是基于工业标准ISA-18.2的经验法则，即每分钟重复超过3次的报警被视为抖动报警，但它不是唯一的。文献[25]将抖动（重复）报警定义为在1分钟（15分钟）内触发并清除10次或更多次的报警，即抖动报警的阈值为6s，重复报警的阈值为90s。如果无法确定阈值，则可以使用抖动指数来测量抖动报警的严重程度，并赋予抖动报警严重的变量更高的优先级。

抖动指数在文献[23]定义为

$$\psi = \frac{2}{I} \sum_{i=1}^I \frac{1}{r(i)}, \quad (3.84)$$

其中， $r(i)$ 为第*i*个游程长度，*I*为一段特定的时间内报警发生的总次数。 ψ 的物理意义是游程长度倒数的平均值，因数2是为了使 ψ 在[0,1]内。指标 ψ 是基于游程长度，但是报警持续时间和报警间隔时间相比游程长度来说更合适。例如，如果报警持续时间较短且报警间隔较大，显然会出现抖动报警，反之亦然。但是，在任何一种情况下，游程长度都会很大，使得 ψ 取值较小，这将导致没有抖动报警的结果。因此，这里提出了一个新的抖动指数

$$\eta = \frac{1}{I} \sum_{i=1}^I \frac{1}{T(i)}, \quad (3.85)$$

其中， $T(i) = \min(T_1(i), T_0(i))$ 。指标 η 的范围为[0,1]，物理意义是报警持续时间或报警间隔时间的倒数。指标 η 越大，抖动报警越严重，因为报警持续时间或报警间隔时间越小。后文将给出工业案例来说明 η 相较于 ψ 的优势。

如果通过规则1和抖动指标检测出大量的抖动报警，就需要一个合适的报警生成机制。延迟器、报警死区、滤波器都经常被用来消除干扰报警。延迟器可以直接用于报警变量；相反，报警死区和滤波器都只适用于模拟信号而不适用于数字信号。因此我们用*m*-样本延迟器，即当且仅当*m*个连续的报警变量 $x_a(t)$ 为‘1’时报警发生^[5]。

参数*m*是在报警或非报警状态下要连续的样本数，并且与报警持续时间或报警间隔一致。因此，可以将*m*作为短报警持续时间或报警间隔的阈值，以检测抖动警报。基于不同类型过程变量的变化动态，工业标准推荐的*m*值在表3.17给出。例如， $m = 20s$ 作为规则1中的阈值。报警持续时间或报警间隔时间小于*m*的样本都将被*m*-样本延迟器消除。所以误报率和漏报率都会减少，但牺牲的是平均报警延迟（AAD）的增量，它是潜在异常和报警发生之间时间延迟的平均值。如果 $x_a(t)$ 是独立同分布信号，配置*m*-样本延迟器的平均报警延迟（AAD）为^[24]

$$AAD(m) = \frac{1 - p_1^m - p_2 p_1^m}{p_2 p_1^m} \quad (3.86)$$

表 3.17 工业标准ISA-18.2的延迟器因子 m 的推荐值

变量类型	$m(s)$
流量	15
水平	60
压力	15
温度	60

其中 p_2 为 $m = 1$ 时漏报率的大小，且 $p_1 = 1 - p_2$ 。式(3.86)表明AAD随着 m 的增加而增加。文献[22]的数值研究表明，如果 p_2 偏小（例如，小于0.05），则AAD(m)会比 m 稍大，但当 p_2 较大时， m 增大，AAD(m)会随 m 的增加而快速增加。因此， m 应该尽可能小。给定性能需求RFAR、RMAR和RAAD分别作为FAR、MAR和AAD的上限，最优值 m_{opt} 为满足下面两个不等式的最小值。

$$F_T(m_{opt}) \geq 1 - \min(\text{RFAR}, \text{RMAR}), \quad (3.87)$$

$$\text{AAD}(m_{opt}) \leq \text{RAAD}, \quad (3.88)$$

其中 F_T 为 $T(i)$ 的累积概率密度函数。

3.6.2 振荡报警

振荡报警，也称为循环报警，是在固定时间段内反复在报警状态和非报警状态之间转换的报警^[25]。它通常是由过程变量中的振荡引起，是过程工业中普遍存在的现象^[26]。例如，文献[5]讨论了由重复的开/关控制动作（循环时间为43分钟）引起振荡报警的一些工业示例。抖动和振荡报警之间的主要区别是报警持续时间的周期性或振荡报警的间隔。另外，抖动报警通常与较小的报警持续时间或间隔相关，而振荡报警的报警持续时间或间隔可能会很大。这里三个指标的表示方法同3.6.1节。

检测抖动报警的规则为^[22]：

规则2.如果报警持续时间 T_1 或报警间隔时间 T_0 保持为一个常数时，则发生振荡报警。

以报警持续时间为例，采用假设检验来说明 T_1 或 T_0 是一个常数。主要思想是如果报警持续时间的变化相对于报警持续时间的平均值较小，则报警持续时间被看作一个常数。原假设和替代假设是

$$H_0 : CV = 1, H_1 : CV > 1, \quad (3.89)$$

其中 CV 代表报警持续时间序列 $\{T_1(l)\}_{l=1}^L := \{T_1(l), \dots, T_1(L)\}$ 的变化系数，例如， $CV = \sigma_{T_1}/\mu_{T_1}$ 。 μ_{T_1} 、 σ_{T_1} 分别为 T_1 的均值和标准差。基于 CV 的置信区间，假设检验的实施如

下^[21], 如果满足不等式(3.90), 则原假设 H_0 被拒绝, 即 T_1 为非常数^[22]。

$$R_{T_1} = \frac{\sqrt{\chi^2_{L-1,\alpha/2}}}{\sqrt{L-1}S_{T_1}/\bar{T}_1} > 1, \quad (3.90)$$

这里 \bar{T}_1 和 S_{T_1} 分别为数据 $\{T_1(l)\}_{l=1}^L$ 和 $\mu_{T_1}\sigma_{T_1}$ 的估计值, 即

$$\bar{T}_1 = \frac{1}{L} \sum_{l=1}^L T_1(l), S_{T_1} = \sqrt{\frac{1}{L-1} \sum_{l=1}^L (T_1(l) - \bar{T}_1)^2}. \quad (3.91)$$

若要判断 T_0 是否为一个常数, 将上述检验过程中的 T_1 替换为 T_0 。

如果 T_0 或 T_1 通过了假设检验, 即出现了振荡报警, 这就需要设计 m -样本延迟器。如果 $x_a(t)$ 在大多数情况下处于非报警状态, 则其应该被移除来满足误报率的需求, 即

$$Pr(T_1(l) > m) \leq \text{RFAR}. \quad (3.92)$$

通过使用切比雪夫不等式, m_{opt} 的最优值为满足下面两个不等式的最小值, 即

$$m_{opt} \geq \bar{T}_1 + \frac{S_{T_1}}{\sqrt{2\text{RFAR}}}. \quad (3.93)$$

$$m_{opt} \leq \text{RAAD}. \quad (3.94)$$

若 $x_a(t)$ 在大多数情况处于报警状态, 那么式(3.93)、(3.94)中的 T_1 被 T_0 代替, S_{T_1} 被 S_{T_0} 代替, RFAR被RMAR代替。

3.6.3 与模拟量相关的干扰报警

抖动和振荡报警直接基于报警变量定义, 与基础过程变量的类型无关。如果式(2.1)中 x 是连续的模拟信号, 那么就可以利用其特征来区分干扰和真实警报。在干扰报警与真实报警混在一起且无法通过规则1和2检测到的情况下, 文献 [24] 提出了一种正常和异常数据分割方法, 实际上该方法以误报和漏报的形式检测干扰警报, 具体如3.2节所示。误报警和漏报警都是需要被消除的干扰报警。通常可以采用报警延迟器、报警死区或者设计报警阈值来消除干扰报警。下文中的工业案例都是以报警延迟器为例来消除干扰报警。

3.6.4 主要技术步骤

在前面的部分中, 针对单个警报变量介绍了三种主要技术。实际上, 工业报警系统中通常有成千上万个报警变量。对于如此大量的报警变量, 将在以下步骤中应用该技术:

步骤1: 收集所有报警变量的基本信息。基本信息包括报警变量和相关过程变量的名称(或标记名), 报警阈值和报警生成机制。例如, 配置高报警阈值为150, 变量名为‘X71NAO067’的模拟变量 x_i , 即

$$x_{ia}(t) = \begin{cases} 1, & x_i(t) \geq 150 \\ 0, & x_i(t) < 150 \end{cases}. \quad (3.95)$$

这里的 $x_{ia}(t)$ 是 x_i 所对应的报警变量。信息收集是一个耗时的步骤，因为这样的信息通常在工业工厂的档案中不可用，并且报警变量的数量通常很大。对于下一节中的发电单元，有2226个报警变量，并且每个报警变量的上述信息必须手动收集。

收集报警变量的高级信息，例如与报警变量关联的异常，忽略发生报警的后果以及对发生报警做出响应的允许时间。该做法需要深入的过程知识，具有一定的挑战性。但高级信息对于提高工业报警系统的性能非常有价值。在收集高级信息之后，通常会进行报警合理化以纠正报警变量的不正确配置，例如，删除重复的（不必要的）报警变量，重新调整报警阈值和调整报警变量的优先级。

步骤2：准备所有报警变量和相关过程变量的历史数据样本。通常将所有过程变量的历史数据样本存储在工厂数据库中。某些报警变量的相关过程变量的值相同。例如，配置高报警阈值为1，变量名为‘DBAK401’的数字变量 x_j ，即

$$x_{ja}(t) = \begin{cases} 1, & x_j(t) = 1 \\ 0, & x_j(t) = 0 \end{cases}. \quad (3.96)$$

因此，报警变量 x_{ja} 与 x_j 相同。但是，对于许多其他的报警变量，如式(3.95)中的 $x_{i,a}$ ，其必须通过将相关过程变量的数据样本通过相应的报警生成机制来获取其历史数据样本。

步骤3：寻找遭受干扰报警的报警变量。首先，抖动报警可以通过规则1来检测；另一个方法是计算每个报警变量出现的次数和式(3.85)中抖动指标 η 。应该注意具有大量报警发生和较大抖动指标值的报警变量。在筛选了抖动报警之后，可以利用规则2来检测振荡报警。最后，第3.6.3节中的方法可用于检测其余报警变量的误报和漏报，在这种情况下，干扰报警可能与真实报警混合在一起。

步骤4：对遭受大量干扰报警的报警变量设计报警器。基于步骤3检测到的报警变量，根据报警延迟器、报警死区和报警阈值的设计原理来消除干扰报警。延迟器是大多数工业报警系统中的集成模块，因此从技术意义上讲，设计延迟器的配置非常容易。但是，该配置必须由工厂运营商批准，且该批准基于报警变量的高级信息。后文中的工业案例也多采用报警延迟器来消除干扰报警。

步骤5：在固定的时间段（例如一天或一周）中，重复步骤2-4作为常规维护任务。首次执行步骤1-4后，预计工业报警系统的性能将大大提高。但是，当工厂经历某些运行状态或外部环境变化时，某些报警变量可能会遭受严重的干扰报警。通过定期重复上述步骤，工业报警系统可以长期保持良好的性能。

3.6.5 案例分析

本节介绍了将上述技术应用于某300MW火力发电厂的复杂工业设施的结果。

发电单元的主要操作是将通过燃烧煤炭而产生的燃烧热能转换成高压高温蒸汽的热能，并通过发电机利用蒸汽轮机提供的机械能发电。发电单元的分布式控制系统（DCS）涉及24079个过程变量。有8145个模拟过程变量，其中208个变量配置有报警变量。在15934个数字过程变量中，也有2018个已配置了报警变量。式(2.1)中最基本的形式用于在DCS的当前报警系统（CAS）中生成所有2226个警报变量。

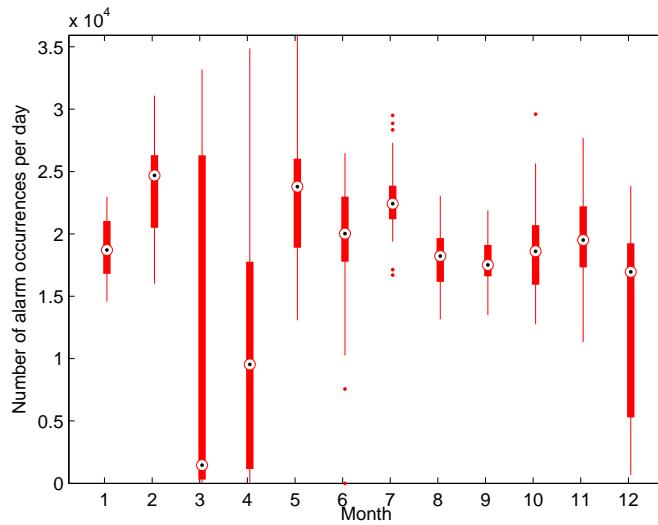


图 3.61 2015年12个月中CAS中报警发生次数的每日分布

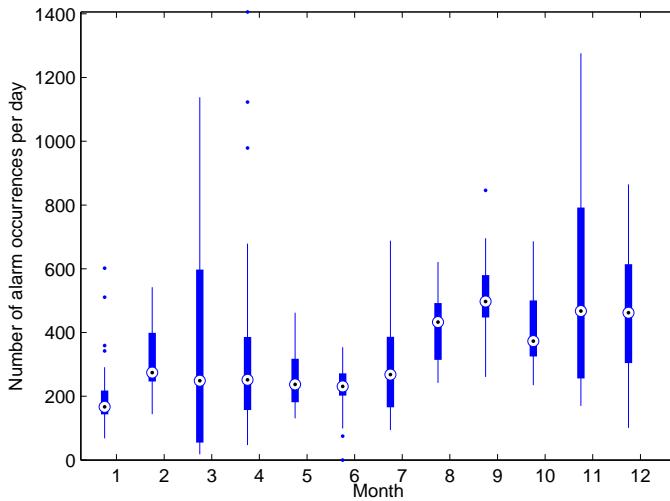


图 3.62 2015年12个月AAS中报警发生次数的每日分布

自2014年1月以来，我们一直在研究CAS的性能，并基于相同的2226个报警变量开发了一个与CAS并行运行的高级报警系统（AAS）。CAS和AAS均使用采样周期 $h = 1s$ 。第3.6.1节中的技术是基于2014年的历史数据集实施的。检测到严重滋扰报警的报警变量，以便在线设计和实施延迟器以消除干扰报警。在发电厂中，培训了在发电单元工作的25名操作员（每班5名操作员，共5名班次）同时使用CAS和AAS。由于AAS的性能优于CAS，操作人员对AAS的信任度逐渐提高，因此运营商最终倾向于对AAS做出回应。下一节将介绍CAS 和AAS 的性能比较。

表 3.18 工业报警的性能指标

	EEMUA标准	电力行业	CAS	AAS
平均报警/天	144	2000	18280	359
高峰报警/10分钟	10	350	405	43
平均报警/10分钟	1	8	127	2.5

这里将对CAS和AAS在2015年的表现进行详细比较。表3.18比较了EEMUA基准、电力行业调查^[25]、CAS和AAS中报警系统的三个指标。CAS的三个指标远比EEMUA基准差，而AAS 的三个指标非常接近EEMUA 基准。除此之外，对CAS和AAS在每个月的性能进行了调查，表3.19比较了2015年12个月中CAS和AAS中发生报警的次数。图3.61和3.62分别显示了每月CAS和AAS 中报警发生次数的每日分布（以箱形图的形式）。其中，CAS中每天平均发生报警的数量高达18280，而AAS中仅为359，说明AAS已经消除了CAS中约98.04%的报警发生。

移除的报警属于干扰报警，因为没有遗漏的真实报警被报告。换句话说，操作员能够通过AAS中发生的报警来检测异常的存在。原则上可以通过设计的延迟器来保证这一点，因为它们不会对异常持续时间长于延迟器参数 m 的真实报警产生任何影响（除了额外的检测延迟）。实际上，设计的延迟器中所有 m 值均已根据可接受的异常响应时间由工厂操作人员批准。

最后本节介绍了应用于发电单元的三个示例，以说明第3.6.1- 3.6.3节中的主要技术。

例1. 在2014年12月1日的24小时内，配置报警变量的57个过程变量发生了报警。计算公式(3.84) 和(3.85)的两个抖动指标 ψ 、 η ，如图3.63(a)和(b)所示。图3.63 (c)是57个变量发生报警的总次数。通过查看 ψ 和 η 不一致的变量，可以发现 ψ 的缺点。例如，第10个过程变量 x_{10} 的抖动指标 $\psi = 0.0191$ 、 $\eta = 0.9921$ 。这是一个数字信号，其变量名为‘DBAK401’，且配置高报

表 3.19 2015年12个月CAS和AAS报警数量

	CAS	AAS	报警减少率
一月	582615	6424	98.90%
二月	667302	8740	96.69%
三月	393220	10420	97.35%
四月	341794	10533	96.92%
五月	724686	7915	98.91%
六月	575079	6737	98.83%
七月	705110	8825	98.75%
八月	560846	12744	97.73%
九月	527927	15358	97.09%
十月	583834	12725	97.82%
十一月	588066	15884	97.30%
十二月	421565	14708	96.51%

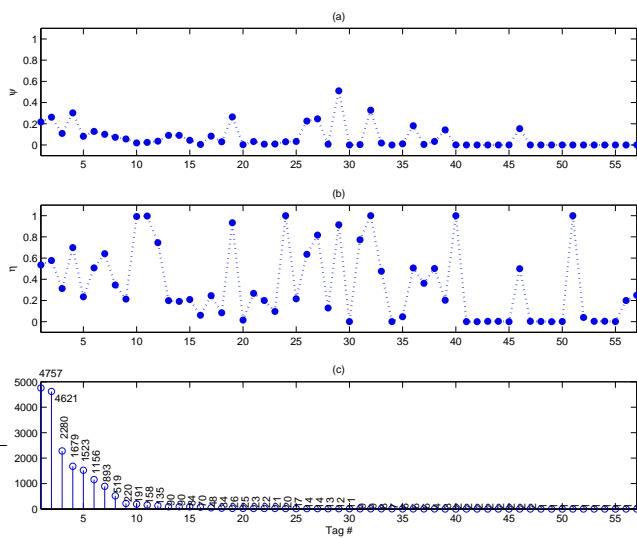


图 3.63 两个抖动指数及报警发生总数

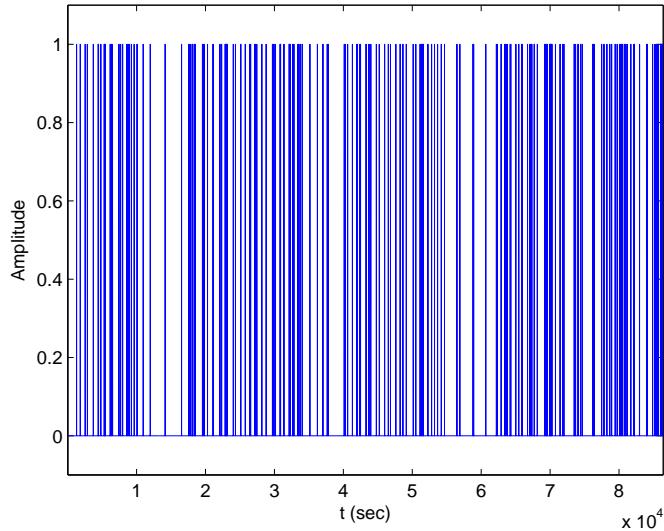


图 3.64 变量DBAK401的24小时时间趋势图

警阈值为1。图3.64中 x_{10} 的时间趋势说明出现了抖动报警。这里一共有191次报警出现，188个报警持续时间为1s，3个报警持续时间为2s。然而，报警行程长度很大，如图3.65 报警行程的柱状图所示。因此， ψ 错误地取了一个很小的值，与抖动报警的实际严重性不符。相反， η 是一个较大的值且接近最大值1，表明出现了严重的抖动报警。

x_{10} 是根据蒸汽驱动给水泵的转速传感器的三个独立测量值之间的最大差值生成的，该蒸汽驱动给水泵是为锅筒提供充足水的关键设备。一旦最大差值大于等于5rpm, x_{10} 取1, 否则

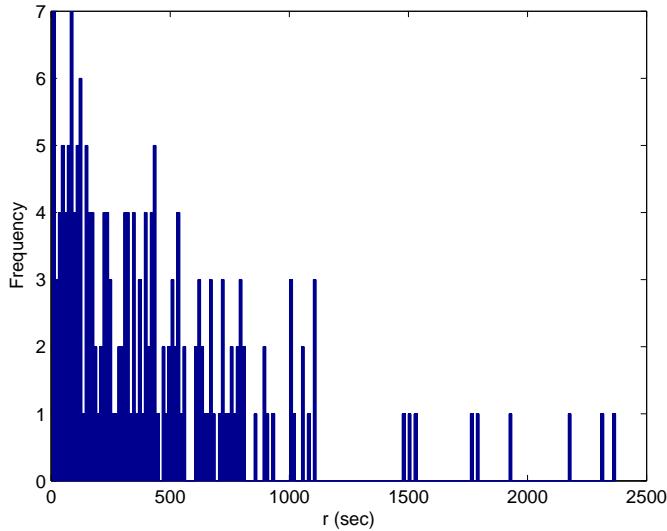


图 3.65 变量DBAK401的24小时报警运行长度的直方图

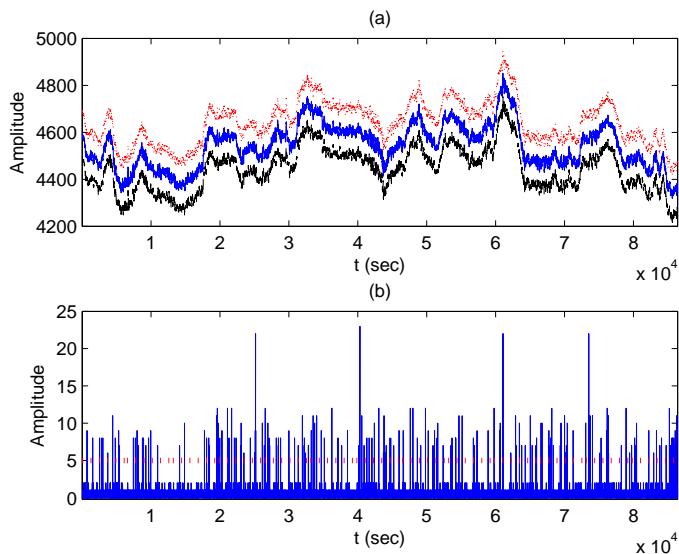


图 3.66 三个转速测量的时间趋势及它们的最大差

取0。因此，相应的报警变量 $x_{10,a}$ 起到监控三个转速测量值之间一致性的作用。如果一个传感器工作不正常，则 $x_{10,a}$ 预计会进入报警状态，并且应该及时采取传感器维护措施以进行可靠的测量。

在2014年12月1日至2014年12月31日的30天内， x_{10} 经历了4622次报警，其报警持续时间在表3.20中列举。图3.66给出了三个转速测量值的时间趋势及在2014年12月1日的24小时内其最大差值（配置阈值为5rpm）。这三个测量值之间的最大差值带有严重的噪声，并引发了报警持续时间不超过5秒的严重抖动报警。基于表3.20中的报警持续时间及 x_{10} 的物理意义，选

表 3.20 2014年12月31日30天内报警持续时间及其频率

持续时间 (s)	1	2	3	4	5
报警次数	4535	78	8	0	1

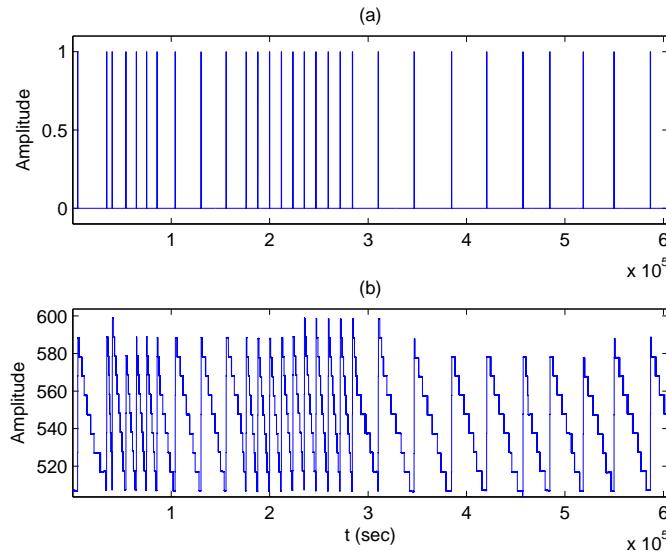


图 3.67 变量4U37LS95和变量4U37LT01的时间趋势图

择RFAR=RMAR=0.01, RAAD=20s是合理的。实际上，异常不能由过去过程变量观察到，所以MAR是不能得到的。因此式(3.86)中的AAD也是不能得到的。但是， p_2 应该非常小，这是由 x_{10} 的报警生成机制所决定的，即使用5rpm的小阈值来减小三个测量值之间的最大差异。因此，正如文献 [22]中对式(3.86)的数值研究所揭示的，AAD预计会比 m 稍大。因此，延迟器参数 m 选择 $m_{opt} = 4$ 作为满足两个不等式(3.87)和(3.88) 的最小值。上述延迟器设计已得到工厂设备操作员的认可，略大于4s的AAD可以使报警减少率达到99.9%。2015年中在抖动指标为 $\psi = 0.0148$ 、 $\eta = 0.9911$ 的CAS中 x_{10} 发生的报警次数为55629。相反，通过使用 $m = 4$ 的延迟器在抖动指标为 $\psi = 0.0185$ 、 $\eta = 0.0668$ 的AAS中 x_{10} 发生的报警次数为17。因此，CAS中的大量抖动报警被消除。

例2. 变量名为‘4U37LS95’的数字变量配置高报警阈值1。这是图3.63中第45个过程变量 x_{45} 。该变量来自一个开关传感器，以监控向发电机定子提供冷却水的储罐水位。如果水位低于某个确切高度约510毫米的水位，则 x_{45} 取1表示报警发生；否则， x_{45} 取0。水位由另一个传感器测量，该传感器由变量名为‘4U37LT01’的模拟变量表示，该变量配置有300mm的低报警阈值。因此，制定了两个报警变量来监督水位的不同异常情况。

图3.67显示了从2014年5月1日到2014年5月8日的7天内数字和模拟变量的时间趋势。显然，水位经历了振荡，导致 x_{45} 发生了29次报警。29个报警持续时间的直方图如图3.68所示。式(3.89)中的假设检验如下。式中的样本均值和方差为 $\bar{T}_1 = 308.9310$ 和 $S_{T_1} = 20.2748$ 。式中的统计量 $R_{T_1} = 0.0888$ ，其中置信水平 $\alpha = 0.05$, $L=29$ ，接受原假设。因此报警持续

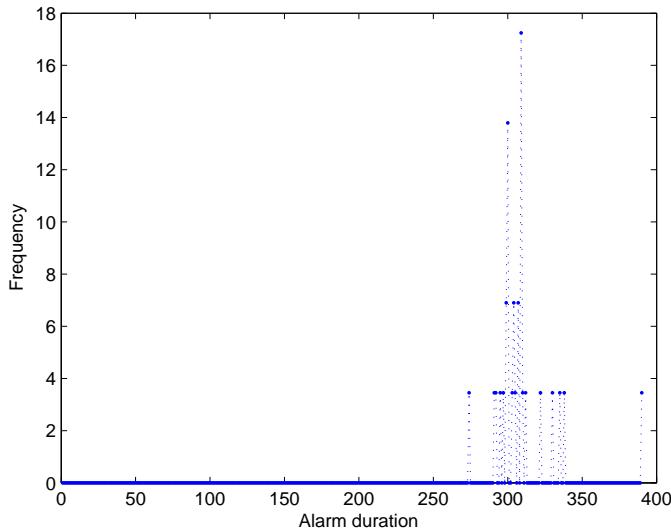


图 3.68 变量4U37LS95的报警持续时间直方图

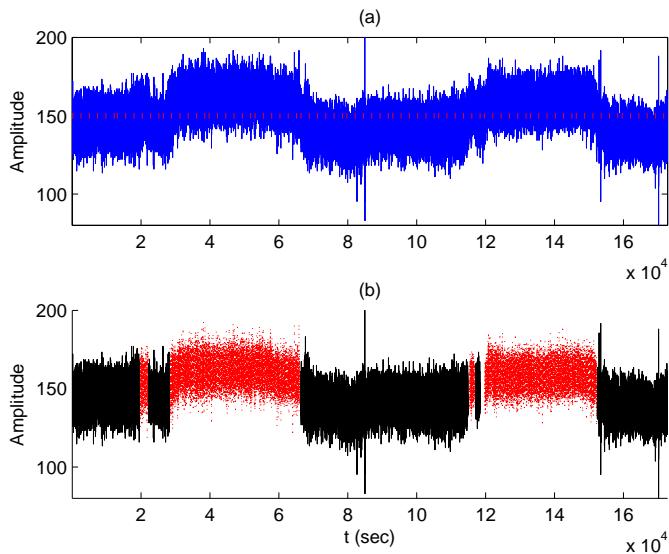


图 3.69 变量X71NAO067的48小时时间趋势及正常和异常数据段

时间是一个常数，并且存在振荡报警。选择RFAR=RMAR=0.01，RAAD=600s。选择最优参数 $m_{opt} = 452s$ 作为满足两个不等式(3.93)和(3.94)的最小值，工厂运营商已经批准通过了这个极大的 m 值。振荡是由发电机的正常运行引起的，并不表示需要采取纠正措施的实际异常情况。水位缓慢移动，如图3.67(b)所示，模拟变量的振荡周期较大，约为3小时。因此，即使发生低水位异常，也可以容许约6分钟的AAD，以对异常进行迅速的响应。在2015年全年，CAS中 x_{45} 报警发生总数为1344。相反，通过应用 $m = 452s$ 的延迟器，AAS 中 x_{45} 报警发生总数降低为20。

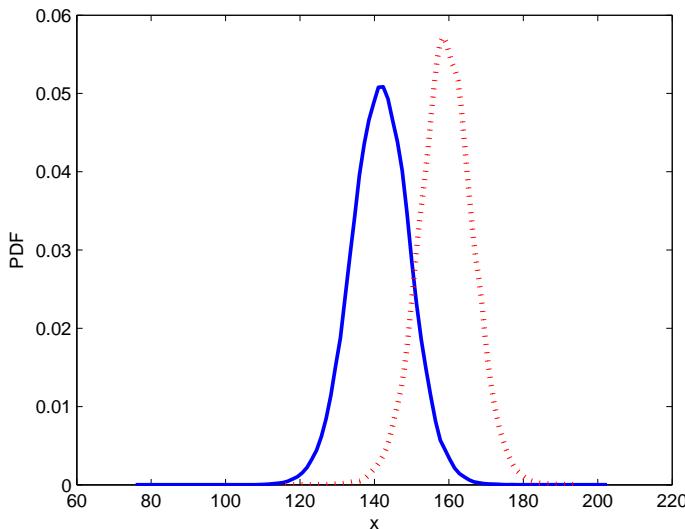
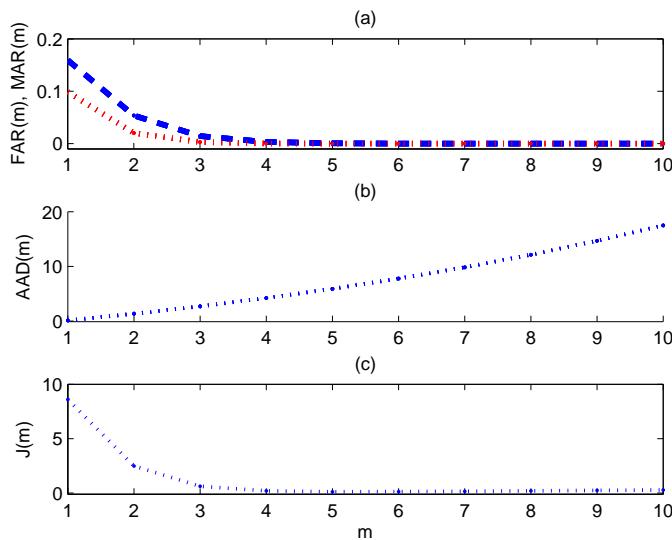


图 3.70 变量X71NAO067正常和异常数据概率密度函数估计值

图 3.71 变量X71NAO067性能指标及 $J(m)$ 变化图

例3.以图3.63中的第二个过程变量 x_2 为例，该模拟变量的变量名为‘X71NAO067’，且配置高报警阈值150pa。该变量测量了两个NOx还原反应器之一中两层之间的催化剂压力差。如果 x_2 大于150pa，则会出现异常，例如两层间的孔洞被阻塞，因此需要采取维护措施。

在过去，变量 x_2 发生了大量的报警。例如，在2014年12月1号的24个小时内发生了4621次报警，误报警和漏报警并存。图3.69(a)展示了 x_2 在2014年4月26日至2014年4月28日的48小时内的趋势。通过使用第3.2节的方法，可以得到正常和异常的数据部分，如图3.69(b)所示。 x_2 在正常和异常情况下的概率密度函数可以通过核密度估计得到，如图3.70所示，由此可以根据

第3.3节得到 $q_1 = 0.159$, $p_2 = 0.0986$ 。通过图3.71可以得到最优延迟因子 $m_{opt} = 5$ 。2015年中在抖动指标 $\psi = 0.2623$ 、 $\eta = 0.5774$ 的CAS中 x_2 发生的报警次数为1833495（209.3次每小时）。相反，通过配置 $m=5$ s的延迟器，在抖动指标为 $\psi = 0.008$ 、 $\eta = 0.0218$ 的情况下，AAS中 x_2 发生的报警次数降低到了19381次（2.2次每小时）。可以看出，98.94%的抖动报警已经被消除。

3.7 本章小结

本章主要内容是设计单变量报警器，针对各种工业流程中出现的干扰报警过多的问题，本章通过设计报警延迟器、报警死区和报警阈值来消除过多的干扰报警，同时展示了设计方法的长期实际应用效果。为了更准确的设计报警器，本章首先给出了正常和异常数据段划分方法。对于报警延迟器，给出了三个性能指标的表达式，并分别针对三种不同的条件来设计最优的报警延迟器参数。对于报警死区，讨论了死区的适用性判断问题，提出了一个死区指数用来判断信号是否适合使用死区，并对死区进行了最优设计。对于报警阈值，明确基于报警概率图中4个统计量的数学定义，并推导了它们的数学表达式，从而基于报警概率图设计了最优报警阈值，有效地减少了干扰报警。最后介绍了上述方法在实际工业中的长期应用效果，结果表明本章涉及的单变量报警器可以大大减少干扰报警的数量。

参考文献

- [1] PETTITT A. A non-parametric approach to the change-point problem[J]. *Applied Statistics*, 1979, 28(2):126-135.
- [2] KEOGH E, CHU S, HART D, et al. Segmenting time series: A survey and novel approach[J]. *Data Mining in Time Series Databases*, 2004, 57:1-22.
- [3] WANG J D, YU Y, CHEN K. Determining the number of segments for piece-wise linear representation of discrete-time signals[J]. *Computers and Chemical Engineering*, 2019, 120(5):46-53.
- [4] AHNLUND J, BERGQUIST T, SPAANENBURG L. Rule-based reduction of alarm signals in industrial control[J]. *Journal of Intelligent and Fuzzy Systems*, 2003, 14 (2):73-84.
- [5] BRANSBY M L, JENKINSON J. The management of alarm systems[M]. Sheffield: Health and Safety Executive, 1998.
- [6] BROOKS R, THORPE R, WILSON J. A new method for defining and managing process alarms and for correcting process operation when an alarm occurs[J]. *Journal of Hazardous Materials*, 2004, 115(1-3):169-174.
- [7] HOLLENDER M, BEUTHEL C. Intelligent alarming[J]. *ABB Review*, 2007, 1(1):20-23.
- [8] HUGO A J. Estimation of Alarm Deadbands[J]. *IFAC Proceedings Volumes*, 2009, 42(8):663-667.
- [9] YAMANAKA F, NISHIYA T. Application of the intelligent alarm system for the plant operation[J]. *Neural Networks*, 1997, 10(8):1391-1415.
- [10] YUKI Y. Alarm system optimization for increasing operations productivity[J]. *ISA Transactions*, 2002, 41(3):383-387.
- [11] KIESSLER P C. Probability and statistics for engineers and scientists[J]. *The American Statian*, 2006, 60(4):343-343.
- [12] IZADI I, SIRISH L S, DAVID S S, et al. An introduction to alarm analysis and design[C]// The 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes Barcelona, 2009.
- [13] IZADI I. Optimal Alarm Design[C]// Fault Detection, Supervision and Safety of Technical Processes. 2009.
- [14] PAPOULIS A, PILLAI S U. Probability, Random variables and stochastic processes[M]. New York: McGraw Hill, 2002.

- [15] MICHELE BASSEVILLE, NIKIFOROV I V. Detection of abrupt changes: Theory and application[M]. New Jersey: Prentice Hall, 1993.
- [16] CHEN J, ARJUN K G. Parametric statistical change point analysis: with applications to genetics, medicine, and finance[M]. 2nd ed. Switzerland: Birkhauser Boston, 2012.
- [17] PETTITT A N. A non-parametric approach to the change-point problem[J]. Journal of the Royal Statistical Society, 1979, 28(2):126-135.
- [18] SILVERMAN B W. Density estimation for statistics and data analysis[M]. New York: Chapman and Hall, 1986.
- [19] CHEN B, HONG Y. Testing for the Markov property in time series[J]. Econometric Theory, 2012, 28(1):130-178.
- [20] BERTSEKAS D P, TSITSIKLIS J N. Introduction to probability[M]. USA: Athena Scientific, 2002.
- [21] ABB. Alarm system engineering. <http://www.controlartsinc.com/Support/Articles/Alarm/%20System/%20Engineering.pdf>.
- [22] WANG J D, CHEN T W. An online method to remove chattering and repeating alarms based on alarm durations and intervals[J]. Computers and Chemical Engineering, 2014, 67(4):43-52.
- [23] KONDAVEETI S R, IZADI I, SHAH S L, et al. Quantification of alarm chatter based on run length distributions[J]. Chemical Engineering Research and Design, 2013, 91(12):2550-2558.
- [24] XU J W, WANG J D, IZADI I. Performance assessment and design for univariate alarm systems based on FAR, MAR, and AAD[J]. IEEE Transactions on Automation ence and Engineering, 2012, 9(2):296-307.
- [25] ROTHENBERG D H. Alarm management for process control[M]. New York: Momentum Press, 2009.
- [26] THORNHILL N F, HGGLUND T. Detection and diagnosis of oscillation in control loops[J]. Control Engineering Practice, 1997, 5(10):1343-1354.

第4章 多变量报警器设计

4.1 引言

多变量报警器在现代工业报警系统中的应用十分广泛。与单变量报警器不同，多变量报警器中的过程变量往往是相互关联的，一个过程变量的取值会受到其他过程变量取值的影响。传统的单变量报警器显然难以处理多个过程变量之间存在的复杂关联关系。因此，需要针对这种复杂关联关系设计可靠的多变量报警器，综合考虑各个过程变量的取值，合理确定多变量报警器的报警阈值，从而降低误报警和漏报警的数量，提高报警系统的可靠性。

对于多变量报警器的设计，本章首先提出了基于变化方向的多变量报警器设计方法，根据相关过程变量的定性趋势信息，结合自适应时间梯度方法来设计多变量报警器；其次建立了超椭球体和凸包模型，并利用该模型表示的系统正常工作区，定义和计算各个过程变量的动态报警阈值；然后研究了基于状态变量的多变量报警器设计，给出了量化长时间报警严重程度的指标；最后设计了基于过程变量变化速率的多变量报警器，通过贝叶斯模型估计器检测异常状态并产生报警。上述研究结果从不同角度给出了多变量报警器的设计方案，并通过案例分析展示了所提方案的有效性。

4.2 基于变化方向的多变量报警器设计

定性趋势分析是一种提取单变量的定性增加、减少或稳定特征的方法^[1]，可用于获得过程变量的变化方向。定性增加、减少或稳定特征可以使用各种基函数对原始信号进行表示来获得，例如，多项式函数^[3,23]、样条函数^[4]、小波变换^[5-7]、基元^[8,9]、神经网络^[10]和分段线性表示^[11,15]。然而，现有文献中的研究成果并没有很好地解决如下两个主要挑战：

- 在时间上连续的相似数据点可能会被合并到同一系统状态的数据段中，然而，很难将时间相隔很远的相似数据点合并到同一系统状态中；
- 幅值差变化较大的更容易检测到，然而，由随机噪声引起的大幅变化也容易导致算法产生错误检测。

基于多个过程变量的变化方向，本节提出了一种从历史数据集中自动查找正常和异常数据段的新方法。过程变量的变化被定义为在时间上相邻的两个不同系统状态之间的改变。考虑历史数据散点图中所显示的数据点密度来解决上述两个挑战，这里借助最小持续时间和最小幅值变化定义系统状态；然而，使用历史数据散点图会丧失数据点的时间连续性。在所提出的方法中使用了两次基于密度峰值的聚类算法。第一次应用聚类算法是包含时间维度的，确保每个聚类内的数据点保持时间连续性。之后去除时间维度进行第二次聚类，可以将相隔很远的相似数据点合并。对第一次聚类所得的聚类中心点应用聚类算法，查找所有满足基于最小持续时间和

最小幅值变化所定义的系统状态的变化。将系统状态的变化方向与来自先验知识的规则进行比较，从而将异常数据段与正常数据段分开。

基于相关过程变量的定性趋势信息，本节提出了一种构建多变量报警器的方法。以变频泵为例，当泵处于正常状态时，根据泵的运行规律能够明显观察到出口流量总是与泵转速保持同步地增加或减少。因此，通过检查出口流量和泵转速的变化方向是否一致则可以判断泵在当前时刻是否处于异常状态。

4.2.1 基于变量变化方向的正常和异常数据段识别方法

考虑包含多个过程变量的多变量系统 $X := [x_1, x_2, \dots, x_n]$ 。过程变量 $x_i, i = 1, 2, \dots, n$ 的历史时间序列数据为 $x_i(1:t) := [x_i(1), x_i(2), \dots, x_i(t)]^T$ ，其中 $t \in \mathbb{Z}$ (非负整数集合) 是 th 时刻的样本点的编号， h 是采样间隔时间；令 $h = 1$ 秒，并用 t 代替 th 。在 t 时刻，过程变量 x_i 的变化方向定义如下，

$$[x_i]_{t \rightarrow t+\Delta t} = \text{sign}(x_i(t + \Delta t) - x_i(t)),$$

其中 Δt 是一个正整数，并且当 $x > 0$, $x = 0$ 和 $x < 0$ 时， $\text{sign}(x)$ 取值分别为 $+1$, 0 和 -1 。因此，若 $[x_i]$ 等于 $+1$ 、 0 或 -1 ，则表明过程变量 x_i 正在增加、保持不变或是正在减少。

当多变量系统 X 处于正常状态时，过程变量 x_1, x_2, \dots, x_n 的变化方向应当有固定的定性关系。这种定性关系可以形式地表示为

$$f([x_1], [x_2], \dots, [x_n]) = 0. \quad (4.1)$$

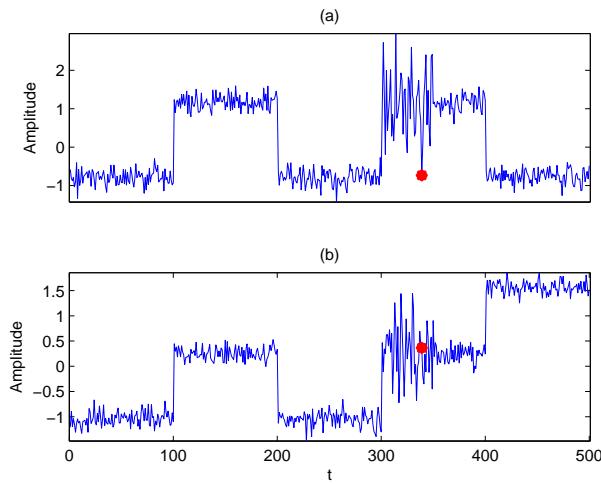
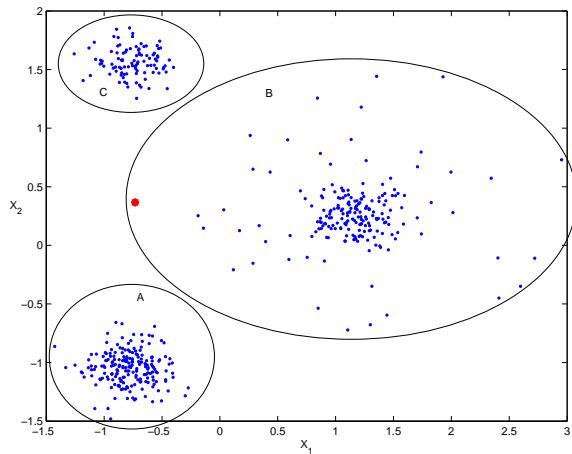
与此相对，当多变量系统 X 处于异常状态时，式(4.1)所示的定性关系则不再成立，即， $f([x_1], [x_2], \dots, [x_n]) \neq 0$ 。基于式(4.1)所示的定性关系，目标是自动地将多变量系统 X 的历史时间序列数据 $X(1:N)$ 分成若干正常和异常数据段。本节所述方法的主要思想说明如下：考虑由两个过程变量组成的双变量系统，即 $X := [x_1, x_2]$ ，假设当系统处于正常状态时 x_1 和 x_2 的变化方向相同，公式(4.1)中的关系可以表示为

$$[x_1] \cdot [x_2] = 1, [x_1] = [x_2] = 0. \quad (4.2)$$

图4.1和图4.2分别展示了多变量系统 X 的时间序列图和散点图。假设系统 X 在初始时刻 $t = 1$ 时处于正常状态。图4.2中展示了 $t \in [1, 400]$ 时间内系统 X 在状态 A 和 B 之间反复转换的过程。过程变量 x_1 和 x_2 满足式(4.2)所定义的关系。与此相对，过程变量 x_1 和 x_2 从聚类 B 转换到聚类 C 的过程不符合式(4.2)所定义的关系。因此，在 $t = 400$ 之前该系统处于正常状态，而在 $t = 400$ 之后系统进入异常状态。

图4.2中的散点图表明基于数据点密度的聚类技术可以解决前面所述的两个挑战。时间上相隔较远的相似数据点可以属于同一系统状态，例如，分别位于 $t \in [1, 100]$ 和 $t \in [201, 300]$ 的样本点 $x_1(t)$ 和 $x_2(t)$ 构成了散点图中的聚类 A 。另外，基于数据点密度的聚类技术拥有较强的抗噪声能力。相较于其他点，位于 $t \in [201, 350]$ 中的样本点 $x_1(t)$ 和 $x_2(t)$ 被更强的噪声所干扰，即便如此，聚类 B 仍然包括这些数据点。

在散点图中挖掘信息会导致时间维的信息丢失，造成不合理的数据分段。例如，考虑图4.1和图4.2中红色标记点的数据点，相较于到聚类 B 的距离，该点到聚类 A 或 C 的距离更近。

图 4.1 过程变量 x_1 和 x_2 的时间序列图图 4.2 过程变量 x_1 和 x_2 的散点图

但是，如果考虑时间信息，为了保持数据点的时间连续性，则该数据点应属于聚类B。为了减少这类影响，引入两步聚类方法。为了保持时间连续性，第一步在包含时间维度的数据空间中对数据点聚类。因此，第一步得到的聚类都是在时间上连续的若干数据点，且各个聚类之间无时间上的重叠。第二步，对第一步中的聚类中心再次聚类，这样即可不必考虑各个聚类内部会丧失时间连续性。

一般地，图4.1和图4.2中表示的各状态之间的转换过程，即 $A \rightarrow B \rightarrow A \rightarrow B \rightarrow C$ ，可以抽象为图4.3所示在聚类中心之间转换的质点运动过程。记 t 时刻质点的位置为 P_t ，且最新的正常系统状态为 P_k ，有 $k < t$ 。转换过程 $\overrightarrow{P_k P_t}$ 定义为从 P_k 到 P_t 的向量，基于式(4.1)所给出的定性

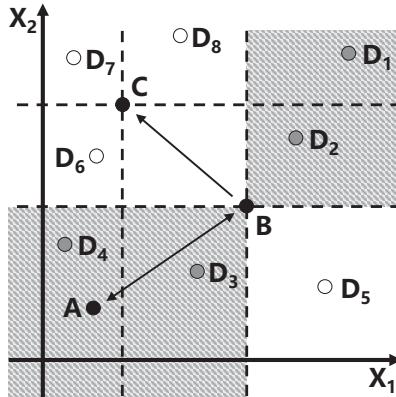


图 4.3 系统状态转换过程

关系, P_t 是否处于正常状态应当取决于向量 $\overrightarrow{P_k P_t}$, 即

$$P_t \text{ 是 } \begin{cases} \text{正常状态, 若 } f\left(\left[\overrightarrow{P_k P_t}\right]\right) = 0 \\ \text{异常状态, 若 } f\left(\left[\overrightarrow{P_k P_t}\right]\right) \neq 0 \end{cases}. \quad (4.3)$$

图4.3中, B 是最新的正常状态点, 由于 $f\left(\left[\overrightarrow{BC}\right]\right) \neq 0$, 则状态点 C 为异常点。因此, 下一个状态点 D 的可能位置中, 深色背景中的 $D_1 \sim D_4$ 为正常状态点, $D_5 \sim D_8$ 为异常状态点。

简言之, 所提方法的主要思想就是使用两次基于密度的聚类算法, 以便识别出系统的基本状态点, 并将当前状态点和最新正常状态点之间的变化方向与式(4.1)所定义的变化方向关系进行比较。如果变化方向一致(不一致), 则相应的数据段处于正常(异常)状态。

下面介绍基于变化方向的数据段划分详细步骤。引入两个参数以定义系统状态, 即最小持续时间 τ 和最小幅值变化 Δ 。最小持续时间 τ 根据常识进行定义, 即, 如果出现异常, 且异常至少持续一段时间才能认为确实发生了异常, 排除了由于噪声引起的短时随机变化。换句话说, 任何持续时间小于 τ 的变化都会被忽略, 只有持续时间大于 τ 的变化才会被视为系统状态变化。因此, τ 是时间维度上区分噪声和系统状态的分界值。 τ 的基本原理类似于报警延迟器^[6], 其作用是在多个连续数据点超过报警限制时发出报警。 τ 的默认值可以取 $\tau = 20$ 秒, 与报警延迟器^[6]的默认建议值相同。

除了时间维度, 幅值维度在定义系统状态中也同样重要, 且只有足够显著变化的幅值差才需要关注。最小幅值变化 Δ 定义为系统状态发生显著变化的下限, 即, 如果下述不等式成立, 则 $x(t_2)$ 与 $x(t_1)$ 处于不同的系统状态。

$$\|x(t_2) - x(t_1)\|_2 := \sqrt{\sum_{i=1}^n [x_i(t_2) - x_i(t_1)]^2} \geq \Delta, \quad (4.4)$$

其中, t_1 和 t_2 分别是系统状态变化的开始和结束时刻, Δ 是幅值维度上噪声和系统状态的分界

值。当系统 X 处于稳定状态时，默认值 Δ 可由时间序列 X 的标准差确定，即

$$\Delta = \sqrt{\sum_{i=1}^n (\text{std}(x_{i,ss}))^2}, \quad (4.5)$$

其中， $x_{i,ss}$ 表示过程变量 x_i 处于正常稳定状态时的时间序列数据。

现将系统基本状态的定义总结如下：使用聚类算法对到的数据点进行聚类，每个系统的基本状态是由 X 的若干历史数据点所属聚类的聚类中心点表示。这些聚类中心点需要满足两个要求：1) 与聚类中心关联的数据段应包含不少于 τ 个在时间上连续的数据点；2) 在时间上相邻的两个聚类中心具有不小于 Δ 的幅值变化，即，满足不等式(4.4)。

首先，所提出的方法需要满足如下基本假设：

- A1. 当系统 X 处于正常状态时，系统 $X := [x_1, x_2, \dots, x_n]$ 的变化方向可以通过式(4.1)进行定义，而当系统 X 进入异常状态时这种关系不成立；
- A2. 初始时刻 X 处于正常状态；
- A3. 操作人员可以借助先验知识给出最小持续时间 τ 和最小幅值变化 Δ 来定义系统状态。

满足这些假设需要操作人员对系统有一定的先验知识，需要了解相关过程变量的变化方向规律以及两个参数 τ 和 Δ 。因此，在应用所提出的方法之前应当验证这些基本假设是否满足并且可行。需要说明的是，在某些情况下，历史数据中的变化可能与这些基本假设不一致。例如，正常条件下的过程变量可能会遇到一些过渡状态，其中过程变量的变化方向与式(4.1)中的关系并不完全一致。如果过渡状态的持续时间和幅值变化分别大于 τ 和 Δ ，则假设A1无效并且在这种情况下不能应用所提出的方法。

所提出的方法包含如下步骤，如图4.4所示。

步骤1：消除时间延迟，对时间序列数据作标准化，并且将先验知识进行数量化。首先，估计过程变量组 x_1, x_2, \dots, x_n 中两两变量的时间延迟，并通过移动时间序列来消除时间延迟。已有文献 [14]对时间延迟的估计方法进行了很好的研究与说明。本节通过常用的互相关方法来估计两个时间序列 $x_i(t)$ 和 $x_j(t)$ 之间的时间延迟 d ，其中 $i \neq j$ ，

$$\hat{d}_{ij} = \arg \max_d \sum_t [x_i(t) x_j(t-d)].$$

通过将时间序列 $x_i(t)$ 的时间索引 t 向后移动 $|\hat{d}_{ij}|$ （如果 $\hat{d}_{ij} < 0$ 则向前移动）消除时间延迟。同时为了防止某些具有大幅值变化的过程变量主导整个计算流程，所有过程变量的时间序列都需要标准化为零均值和单位方差的形式。然后，操作人员根据先验知识选择最小持续时间 τ 和最小幅值变化 Δ ，或者采用默认值 $\tau = 20$ 秒和式(4.5)定义的 Δ 。接下来，将公式(4.1)的所有解构成矩阵 R ，其大小为 $r \times n$ ，

$$R := \begin{bmatrix} [x_1]_1 & [x_2]_1 & \cdots & [x_n]_1 \\ [x_1]_2 & [x_2]_2 & \cdots & [x_n]_2 \\ \vdots & \vdots & \ddots & \vdots \\ [x_1]_r & [x_2]_r & \cdots & [x_n]_r \end{bmatrix}, \quad (4.6)$$

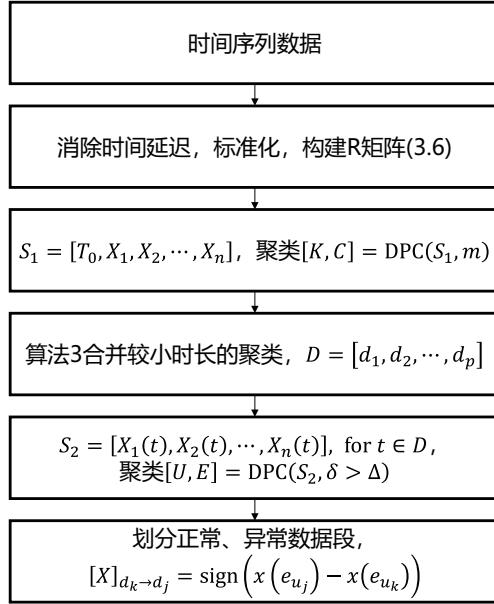


图 4.4 所提方法的流程图

其中, 矩阵 R 的每一行是式(4.1)的一个解。例如, 式(4.2)表示 x_1 和 x_2 同步变化, 即

$$R = \begin{bmatrix} +1 & +1 \\ 0 & 0 \\ -1 & -1 \end{bmatrix}. \quad (4.7)$$

步骤2: 在包含时间维度的数据空间中对数据点聚类。在 X 的历史数据中增加一列时间向量 $T_0 := t_0 \cdot [1, 2, \dots, N]^T$ 即

$$S_1 := [T_0, X] = [T_0, x_1, x_2, \dots, x_n],$$

其中, t_0 是时间维度加权因子, 默认值为 $t_0 = 1$ 。 t_0 的值可以进行调整, 以确保聚类中数据点的时间连续性 (稍后将在例1 中进行说明)。

使用一种基于密度峰值的聚类算法^[14] (density-peak clustering, DPC) 对 S_1 中数据点进行聚类, 本书附录A中给出了DPC 算法的简要介绍。为了确保聚类结果中保留了 X 中足够多的细节, 聚类中心的数量 (由符号 m 表示) 应该足够大。但是, 由于存在对最小持续时间 τ 的要求, 选择过大的 m 值是不可行的。因此, m 的默认值选为 $m = \lfloor N/\tau \rfloor$, 其中 N 是数据点的总数, $\lfloor \cdot \rfloor$ 表示取小于或等于操作数的最大整数。因此, 应用DPC 算法从 S_1 获得 m 个聚类的过程表示为

$$[K, C] = \text{DPC}(S_1, m),$$

其中向量 $C := [c_1, c_2, \dots, c_m]$ 包含了 m 个聚类中心的时间信息, 并且 $K := [k_1, k_2, \dots, k_N]$ 是指示 S_1 中每个点所属聚类编号的查找表, $k_t \in [1, m]$, $t = 1, 2, \dots, N$ 。

同一个聚类中每个数据点的时间索引必须严格按时间顺序排列, 此过程由算法2实现。指

Algorithm 2 叠加时间维并降噪

```

输入参数#1:  $X$ , 归一化后的数据序列;
输入参数#2:  $m$ , 集群中心数;
 $t_0 = 1$ ;
 $\eta = 1$ ;
while  $\eta == 1$  do
     $S_1 = [T_0, X]$ ;
     $[K, C] = \text{DPC}(S_1, m)$ ;
     $\eta = 0$ ;
    for  $i = 1$  to  $N - 1$  do
        if  $c_{k_i} > c_{k_{i+1}}$  then
             $\eta = 1$ ;
             $t_0 = t_0 + 1$ ;
            break the iteration;
        end if
    end for
end while

```

标 η 用于表示是否满足严格的时间顺序。如果 η 等于1, 即不满足时间顺序的要求, 则可增加时间维加权因子 t_0 并重复聚类步骤。

案例1: 本例用于说明 t_0 的作用以及保持同一聚类内数据点时间连续性的必要性。考虑图4.5中展示的 x_1 的时间序列。令 $t_0 = 1$ 添加时间维向量后, 将DPC算法应用于 $S_1 = [T_0, x_1]$, 图4.6(a)展示了DPC算法的决策图。图4.6(b)展示了两个聚类, 相比于聚类1, $t = 11$ 的数据点A在幅值上更接近聚类2。因此, 当 $t_0 = 1$ 时, 数据点A被分配给图4.6(b)中的聚类2。类似地, $t = 12$ 的数据点B被分配给图4.6(b)中的聚类1。因此, 聚类1中的数据点并非都严格地处于

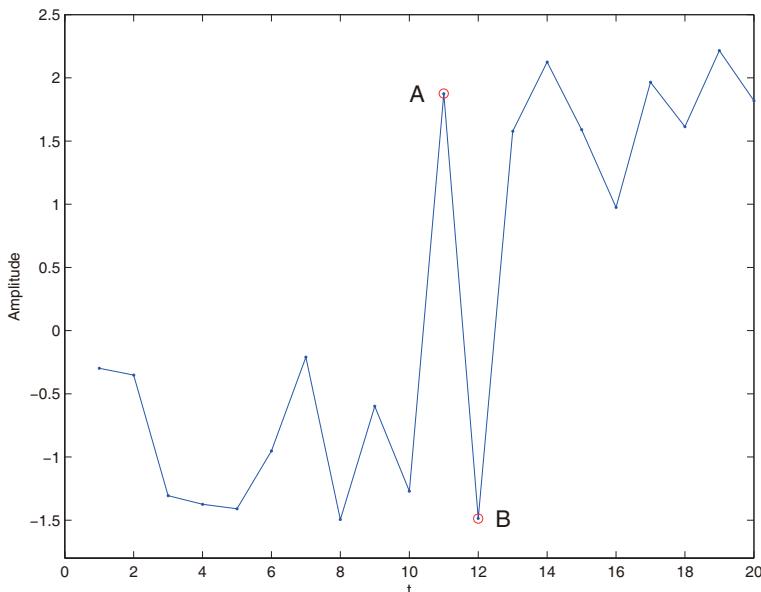
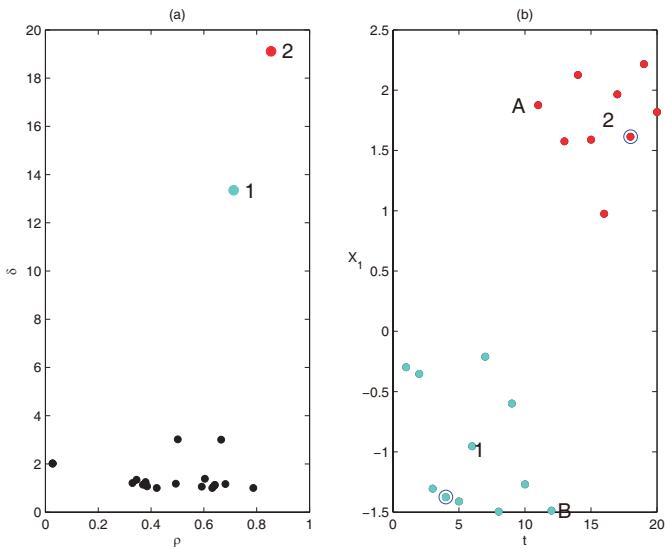
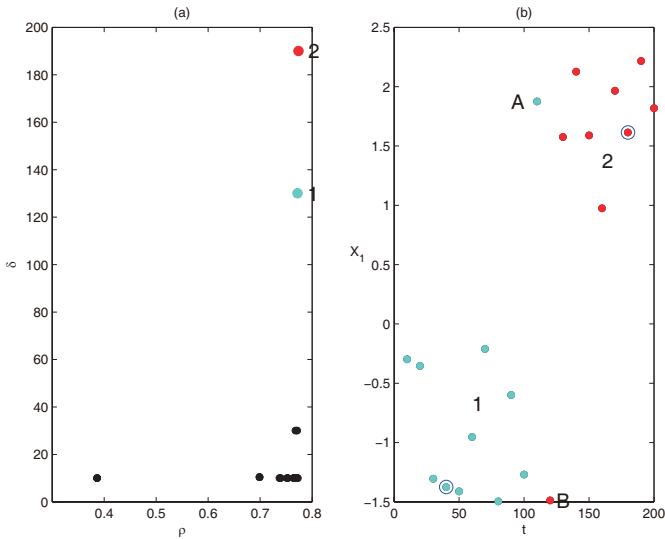


图 4.5 x_1 的时间序列图

图 4.6 决策图及 $t_0 = 1$ 时的聚类结果图 4.7 决策图及 $t_0 = 10$ 时的聚类结果

集群2中的数据点之前，即存在时间不连续的情况。当 t_0 增加时，时间维的权重相应提高，数据点在时间维上的差距增大。由于数据点的时间戳是单调增加的，易知增加 t_0 可以避免时间不连续性问题的发生，当 t_0 增加到10时，图4.7(b)中的时间戳相应增加。因此，数据点A和B可以分别正确分配给图4.7(b)中的聚类1和2。

步骤3：合并较小时长的聚类。根据最小持续时间 τ 的定义，持续时间少于 τ 个数据点的变化认为是随机噪声干扰的结果，不是系统状态变化。为避免生成持续时间短于 τ 的数据段，将步骤2中所有少于 τ 个数据点的聚类合并到另一个相邻时间聚类中。具体过程见算法3。算法3输出 p 个聚类，其中向量 D 包含 p 个聚类的聚类中心时间戳，即 $D := [d_1, d_2, \dots, d_p]$ 。

Algorithm 3 合并较小时长的聚类

```

输入参数#1:  $C$ , 步骤2中聚类中心的向量;
输入参数#2:  $m, C$ 的长度;
输入参数#3:  $\tau$ , 最短持续时间;
 $j = 1$ ;
 $d_1 = c_1$ ;
 $D = [d_1]$ ;
for  $i = 2$  to  $m$  do
    if  $c_i - d_j \geq \tau$  then
         $D = [D, c_i]$ ;
         $d_j = c_i$ ;
         $j = j + 1$ ;
    end if
end for

```

步骤4: 在幅值维度上筛选系统状态点。步骤3中得到的聚类中心点 D 的数据集合如下:

$$S_2 := [x_1(t), x_2(t), \dots, x_n(t)], t \in D.$$

根据最小幅值变化 Δ 的定义, 幅值维度中任何小于 Δ 的变化都可以被忽略。通过将DPC算法应用于 S_2 可以实现此过程。DPC算法所提供的决策图中纵坐标 δ 是聚类中心之间的欧式距离; 因此, 只需要选择 δ 高于 Δ 的数据点作为聚类中心即可保证任意两个聚类中心的欧式距离大于 Δ 。该步骤可以表示为

$$[U, E] = \text{DPC}(S_2, \delta > \Delta),$$

其中, 向量 $E := [e_1, e_2, \dots, e_q]$ 为聚类中心的时间戳; $U := [u_1, u_2, \dots, u_p]$ 是一个查找表, 用于将步骤3中获得的 p 个聚类中心分配到此步骤中的 q 个聚类中, $u_j \in [1, q], j = 1, 2, \dots, p$ 。

步骤5: 划分正常/异常数据段。根据查找表 U , D 的每个元素都被分配到相应 E 的聚类中心所表示的系统状态中。由于 D 的 p 个聚类中心按时间顺序排列, 则可以计算系统状态转换时的变化方向。对于 D 的聚类中心 d_j 和 d_k , 或 S_2 中的数据点 $x(d_j)$ 和 $x(d_k)$, 系统状态的变化方向是

$$[x]_{d_k \rightarrow d_j} = \text{sign}(x(e_{u_j}) - x(e_{u_k})). \quad (4.8)$$

算法4将系统状态变化方向与式(4.6)中矩阵 R 的每一行进行比较, 所得结果中分别用符号0和1标记正常和异常数据段。式(4.8)表明 $x(t)$ 可以由一系列水平线段近似, 由于线段的所有点共享相同的聚类中心 $x(d_j)$, 所以其幅值都是聚类中心点的幅值 $x(e_{u_j})$ 。 $x(t)$ 的近似值记为 $\hat{x}(t)$ 。

所提出方法的计算成本非常低, 大多时间用于两次DPC算法, 在本章附录中说明了DPC算法的时间复杂度。下面通过两个数值例子说明所提方法的具体步骤并验证其有效性。

案例2: 考虑双变量系统 $X = [x_1, x_2]$ 。图4.1和图4.2分别为该系统的时间序列图和散点图。图4.8展示了步骤2中应用DPC算法得到聚类的决策图, 其中DPC算法应用于 $S_1 = [T_0, x_1, x_2]$, $t_0 = 1$, $\tau = 20$ 且 $m = 150$ 。获得由不同的颜色区分的 m 个聚类, 图4.8(b)中蓝色圆圈标记的数据点为聚类中心。图4.9展示了 S_1 的聚类中心。在步骤3中合并具有较短持续时间的聚类之后得到如图4.10所示的数据点。在步骤4中将DPC算法应用于 S_2 以识别系统状

Algorithm 4 划分正常、异常数据段

```

输入参数#1:  $E$ , 步骤4中聚类中心的向量;
输入参数#2:  $U$ , 步骤4中的查询表;
输入参数#3:  $p$ ,  $U$ 的长度;
输入参数#4:  $R$ , 关系矩阵;
 $k = 1$ ; % 初始化
假设从1到 $d_1$ 的数据段处于正常状态;
for  $j = 2$  to  $p$  do
     $\gamma = 0$ ; % 正常标志
    for  $l = 1$  to  $r$  do
        if  $\text{sign}(x(e_{u_j}) - x(e_{u_k})) == R(l, :)$  then
            从 $d_{j-1}$ 到 $d_j$ 的数据段处于正常状态;
             $k = j$ ;
             $\gamma = 1$ ;
            break the iteration;
        end if
    end for
    if  $\gamma == 0$  then
        从 $d_{j-1}$ 到 $d_j$ 的数据段处于异常状态;
    end if
end for

```

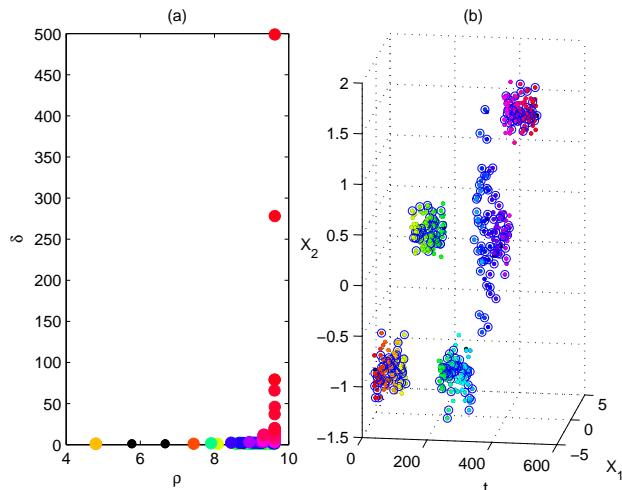


图 4.8 案例2步骤2的决策图及聚类结果

态。图4.11展示了步骤4中的决策图和所获得的聚类。图4.11(a)得到了三个聚类，其中水平红线表示最小幅值变化 $\delta = \Delta$ 。因此， S_2 中的数据点被分类为图4.11(b)中的三个聚类，分别标记为1、2、3。注意，图4.11中的决策图还说明了 Δ 有足够的选择空间，即从0.5到2都会得到相同的聚类结果。这说明所获得的聚类结果非常可靠。最后，如步骤5中所述，计算系统状态转换的变化方向，并与式(4.7)中矩阵 R 的每一行进行比较。图4.12给出了正常、异常数据段的最终划分结果，其中异常数据段从 $t = 398$ 开始。受 τ 样本的最小时长限制，正常和异常数据段划分结果与图4.1中的实际情况一致。

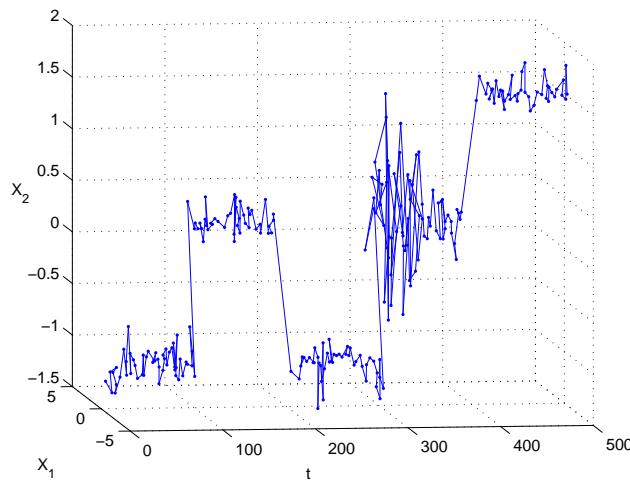


图 4.9 案例2步骤2的聚类中心的三维可视化

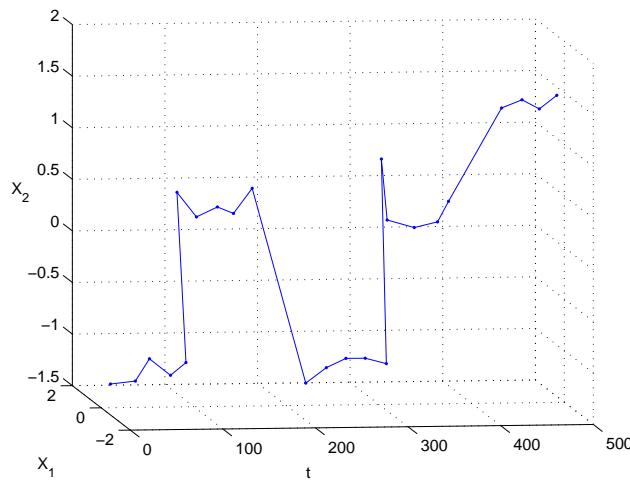


图 4.10 案例2步骤3所得的聚类中心的三维可视化

案例3：本例中使用单个过程变量 $x(t)$ 将所提出的方法与三种有代表性的线性分段方法进行比较。为简单起见，定义 $x(t)$ 增加的数据点为异常状态，剩余点视为正常，即

$$a(t) = \begin{cases} 1, & \text{若 } [x(t)] := x(t+1) - x(t) > 0 \\ 0, & \text{其它} \end{cases}, \quad (4.9)$$

其中 $a(t)$ 用于指示系统状态，正常状态时取值为0，异常状态时取值为1。本例中使用了两个测试变量来说明4.1节所述的两个主要挑战，图4.13的第一排子图展示了它们的时间序列图。

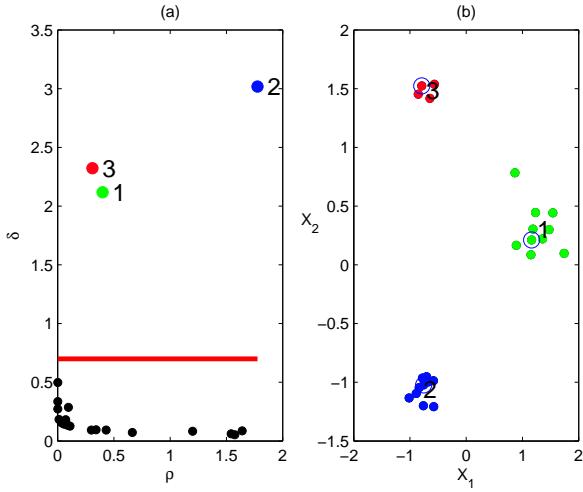


图 4.11 案例2步骤4的决策图及聚类结果

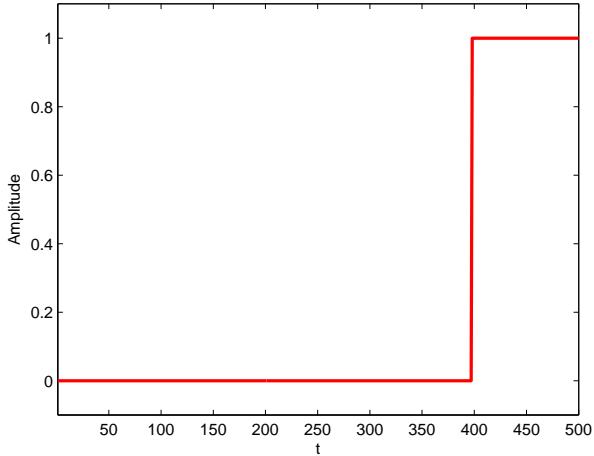


图 4.12 案例2中正常、异常数据段划分结果

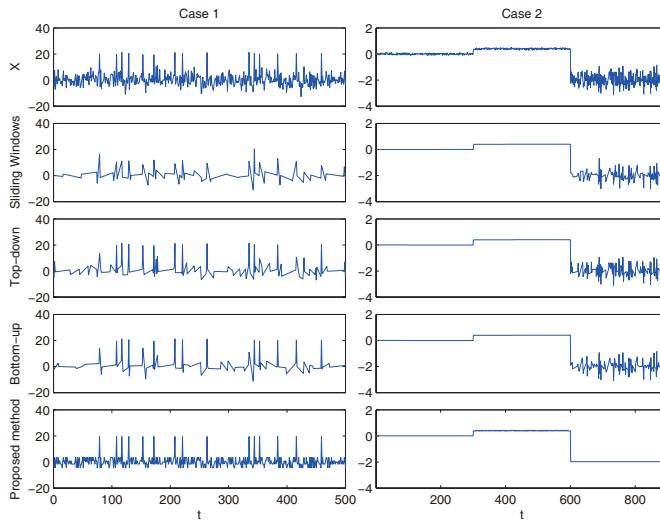
针对第一个挑战，测试变量 $x(t)$ 生成方式如下，

$$x_1(t) \sim \begin{cases} \mathcal{N}(\mu_1, \sigma_1), r(t) > r_0 \\ \mathcal{N}(\mu_2, \sigma_2), r(t) \leq r_0 \end{cases},$$

其中 $r(t) \sim \mathcal{N}(0, 1)$, r_0 为常数, $\mathcal{N}(\mu, \sigma)$ 表示均值为 μ 标准差为 σ 的高斯分布。令 $\mu_2 - \mu_1 > \max(\sigma_1, \sigma_2) > 0$, 根据式(4.9)中定义的规则得到正常、异常的指示变量为

$$a_1(t) = \begin{cases} 0, r(t) > r_0 \\ 1, r(t) \leq r_0 \end{cases}. \quad (4.10)$$

图4.13中展示了某次模拟中 $x_1(t)$ 的时间序列图（左上方子图）。由图可知， $x_1(t)$ 在大多数时间

图 4.13 案例3中 $x(t)$ 和四种方法对它的估计值的时间序列

段处于较低幅值的状态，偶尔会转换为另一个较高幅值的状态。

针对第二个挑战，测试变量 $x(t)$ 生成方式为

$$x_2(t) \sim \begin{cases} \mathcal{N}(\mu_1, \sigma_1), t \in [1, t_1] \\ \mathcal{N}(\mu_2, \sigma_1), t \in [t_1, t_2] \\ \mathcal{N}(\mu_3, \sigma_2), t \in [t_2, t_3] \end{cases}.$$

令 $\mu_2 > \mu_1 > \mu_3$ ，据式(4.9)可得

$$a_2(t) = \begin{cases} 0, t \neq t_1 \\ 1, t = t_1 \end{cases}, \quad (4.11)$$

其中 $\sigma_2 \geq |\mu_2 - \mu_1|$ 。图4.13展示了某次模拟中 $x_2(t)$ 的时间序列图（右上方子图）。由图可知，当 $t \in [t_2, t_3]$ 时， $x_2(t)$ 的噪声要强于其他位置处的噪声。

根据表4.1中不同均值和方差水平的噪声，对两种情况下的 $x(t)$ 执行蒙特卡罗模拟。将三种代表性的线性分段方法^[15]，即自上而下分段法、自下而上分段法和滑动窗分段法，与所提出的方法进行比较。这三种方法都是相同的超参数，即累积平方拟合误差的阈值 δ_x 。所提出方法的超参数是第二次聚类的数量，由 q 表示。表4.1列出了各个调整参数。为了比较指示变量的计算值与真实值，即式(4.10)中的 $a_1(t)$ 和式(4.11)中的 $a_2(t)$ ，调整所有方法的报警阈值确保检测到所有异常数据点，避免出现漏报警。因此，本例中无漏报警，仅需考虑误报警。图4.14比较了四种方法的误报数量。图4.14比较了四种方法的误报数量，清楚地表明所提出的方法在所有模拟中的误报率都是最低的，且几乎不随超参数的调整而改变，而其他三种方法的性能高度依赖于超参数设置，并且误报数量也很多。

当使用自上而下、自下而上和滑动窗分段三种方法时，序列 $x(t)$ 的估计值分别记为 $\hat{x}_{TD}(t)$ 、 $\hat{x}_{BU}(t)$ 和 $\hat{x}_{SW}(t)$ 。在步骤5中，所提方法得到 $x(t)$ 的估计值 $\hat{x}(t)$ 。图4.13展示了 $x(t)$ 的

表 4.1 两种测试情况下模拟参数和四种方法的超参数设置

测试信号		仿真参数	超参数设置	
			δ_x	q
1	$\mu_1 = 0$ $\mu_2 = 20$ $r_0 = -1.8$ $N = 500$	$\sigma_1 = 1$ $\sigma_2 = 0.15$	100	2
			150	3
			200	4
			250	5
			300	6
		$\sigma_1 = 2$ $\sigma_2 = 0.3$	100	2
			150	3
			200	4
			250	5
			300	6
		$\sigma_1 = 4$ $\sigma_2 = 0.6$	100	2
			150	3
			200	4
			250	5
			300	6
2	$\mu_1 = 0$ $\mu_2 = 0.1$ $\sigma_1 = 0.0125$ $\sigma_2 = 0.01$	$\mu_3 = -2$ $t_1 = 300$ $t_2 = 600$ $t_3 = 900$	1	3
			2.75	4
			4.5	5
			6.25	6
			8	7
		$\mu_2 = 0.2$ $\sigma_1 = 0.025$ $\sigma_2 = 0.2$	1	3
			2.75	4
			4.5	5
			6.25	6
			8	7
		$\mu_2 = 0.4$ $\sigma_1 = 0.05$ $\sigma_2 = 0.4$	1	3
			2.75	4
			4.5	5
			6.25	6
			8	7

时间序列图及某次仿真中估计值 $\hat{x}(t)$ 、 $\hat{x}_{TD}(t)$ 、 $\hat{x}_{BU}(t)$ 和 $\hat{x}_{SW}(t)$ 的时间序列图。

第一个测试情况中，导致自上而下、自下而上和滑动窗方法产生大量误报警的原因是这三种方法都只能在局部对 $x(t)$ 进行拟合。时间上相隔较远但相似的数据点无法被归入同一个系统状态中，会被错误地划分到与之相邻的不相似数据点所属的系统状态中。图4.13左侧中部三个子图所展示的 $\hat{x}_{TD}(t)$ 、 $\hat{x}_{BU}(t)$ 和 $\hat{x}_{SW}(t)$ 说明了这一点。与此相反，所提出的方法使用基于密度的聚类方法成功地解决了这个问题。步骤4中通过在不包含时间维度的 S_2 数据集合中聚类得到系统状态。因此，对于所提出的方法，时间上相隔较远但相似的数据点仍然可以被划分到同

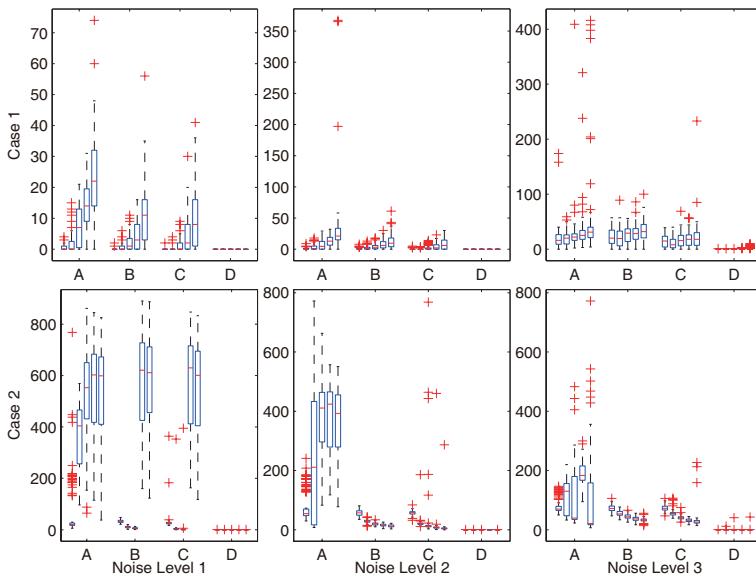


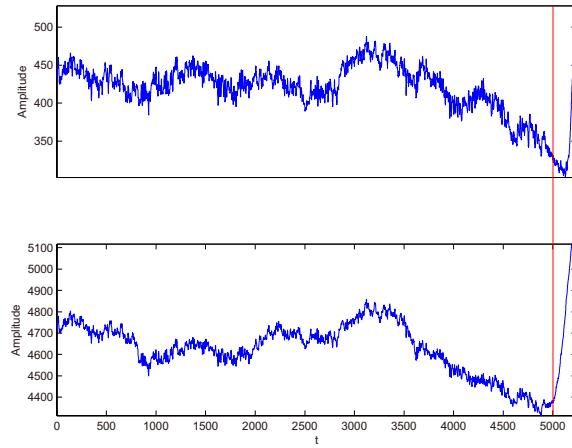
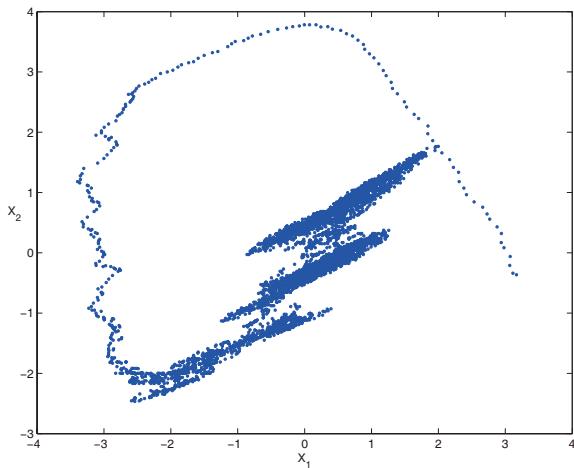
图 4.14 案例3四种方法误报警数量的箱线图

一个聚类中。

第二个测试情况中，位于 $t \in [t_2, t_3]$ 第三个分段时， $x_2(t)$ 的变化范围较大，有较大概率大于从位于 $t \in [1, t_1]$ 的第一个分段的平均值变化到位于 $t \in [t_1, t_2]$ 的第二个分段的平均值。自上而下、自下而上和滑动窗方法仅仅考虑了幅值变化。因此，为了确保位于 $t \in [1, t_2]$ 的数据段中无漏报警，第三个数据段中大量由噪声导致的变化会被错误地识别为系统状态变化，从而产生误报警。这个现象是由前述的第二个挑战导致的。与此相对，所提出的方法中DPC算法在步骤2和步骤4中同时考虑了聚类中心的密度和间距。对于 $x_2(t)$ ，从位于 $t \in [1, t_1]$ 的数据段到位于 $t \in [t_1, t_2]$ 的数据段的幅值变化是相对较小的（仍然大于最小幅值变化 Δ ）。然而，每段数据均包含了足够多的数据点，因而这两段数据的密度足够高，使得这两段数据成为两个聚类，其中 $x_2(t)$ 中位于 $t \in [t_2, t_3]$ 的第三个数据段距离前两段数据的距离较大。并且由于受噪声干扰较强，第三段中的数据点的密度要低于前两个数据段。因此，第三段中的数据点被单独划分为一个聚类。由此可见，本方法能有效应对第三个数据段中噪声的影响。如图4.13的右侧中部子图所示， $\hat{x}_{TD}(t)$ 、 $\hat{x}_{BU}(t)$ 和 $\hat{x}_{SW}(t)$ 仍然包含了较强的噪声，但所提出方法得到的 $\hat{x}(t)$ 达到了明显降噪的效果。

这两个测试案例充分说明了所提方法能够解决上述两个挑战。在所提出方法中，根据步骤2和步骤3进行第一次聚类并合并聚类，得到的每个聚类都至少包含 τ 个数据点。之后，步骤4中再去除时间维，这样才能将时间上相隔较远但仍然相似的数据点归类至同一聚类里。步骤4中使用距离约束 $\delta > \Delta$ 从上一步骤所得的 p 个聚类中心进一步聚类得到 q 个聚类。此外，步骤4得到聚类的过程中DPC 算法还需要考虑密度的影响。因此，所提出方法中获得的系统状态是受超参数 τ 和 Δ 影响的，并解决了上述两个挑战。

案例4：考虑一个变频泵中出口流量 x_1 和泵转速 x_2 构成的双变量系统。基于泵的物理规律，当系统在正常状态时 x_1 和 x_2 应同增或同减。因此描述过程变量之间的关系的矩阵 R 与式(4.7)所述的相同。图4.15展示了 x_1 和 x_2 在标准化之前的时间序列图。图4.16展示了二者标准

图 4.15 案例4中 x_1 和 x_2 时间序列图图 4.16 案例4中 x_1 和 x_2 标准化后散点图

化之后的散点图。

如图4.17所示，所提出的方法的步骤4中产生了13个聚类。图4.18展示了数据点的聚类随时间的分布，可以看到，聚类1和聚类2都包含了不连续但相似的两部分数据，因此所提出的方法可以识别出时间上相隔较远但相似的数据点。图4.19中展示了正常、异常数据的分段结果，可以看到，有两段异常数据。第一段从5010秒到5163秒，对应于图4.17中的聚类3到聚类8的过程，此后是一段正常数据，从聚类8到聚类10，再之后是第二段异常数据，从聚类10到聚类13。这个划分结果是与图4.15所展示的情况一致的。图4.15中可以观察到约5000秒时出现了第一次异常，即泵转速 x_2 持续增加但是出口流量 x_1 持续下降直到5163秒。

案例5：火力发电厂中煤粉燃烧过程的运行状态主要与6个过程变量密切相关，即，机组实发功率 x_1 、给煤量 x_2 、主蒸汽压力 x_3 、主蒸汽流量 x_4 、给水量 x_5 和总风量 x_6 。因此，多变量系统构建为 $X := [x_1, x_2, \dots, x_6]$ 。基于这六个过程变量的物理规律关系可知，当 X 处于正常状态

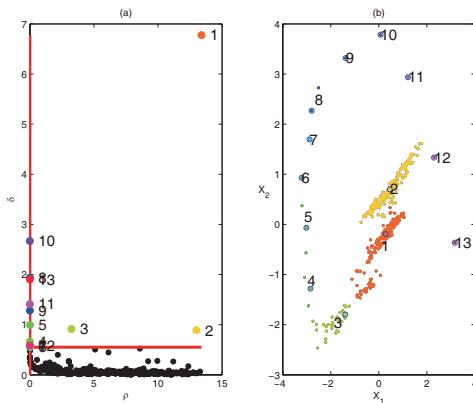
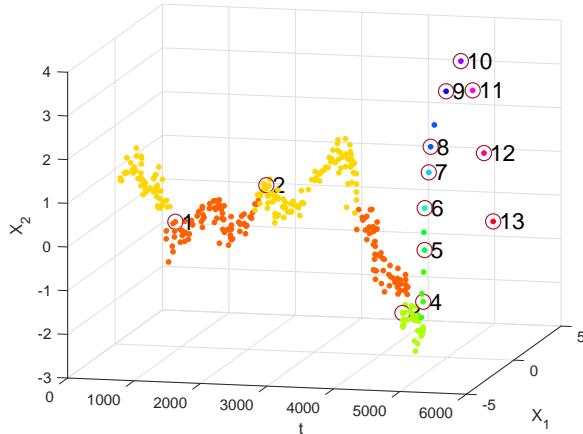


图 4.17 案例4步骤4的决策图及聚类结果

图 4.18 案例4过程变量 X_1 和 X_2 聚类结果

时，这六个过程变量保持同增同减的关系，即，矩阵 R 为

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}.$$

将所提出的方法应用于如图4.20所示的系统 X 的时间序列数据中。图4.21展示了两两过程变量的散点图以及各个过程变量的概率分布直方图。图4.20中展示了由竖线划分开的分别位于 $t \in [1, 6293]$ 和 $t \in [6294, 6800]$ 的正常、异常数据段。

由于该六维空间较难以可视化，这里对子集 $[x_3, x_6]$ 进行补充说明。图4.22展示了所提出方法中步骤4 的决策图及所得的聚类结果。图4.23展示了过程变量 x_3 和 x_6 随时间 t 变化的关系。图4.22和图4.23中聚类6、7和8之间的折返过程正好对应于异常数据段。

图4.20和图4.21可以对检出的异常数据段进行说明。图4.20中竖线右侧，主蒸汽压力 x_3 增加时其他过程变量减少。图4.21中 x_3 与其他过程变量的两两散点图上的折返过程也与矩阵 R 所定义的规则相冲突。检出的正常、异常数据段得到了现场操作人员的认同，该异常对应

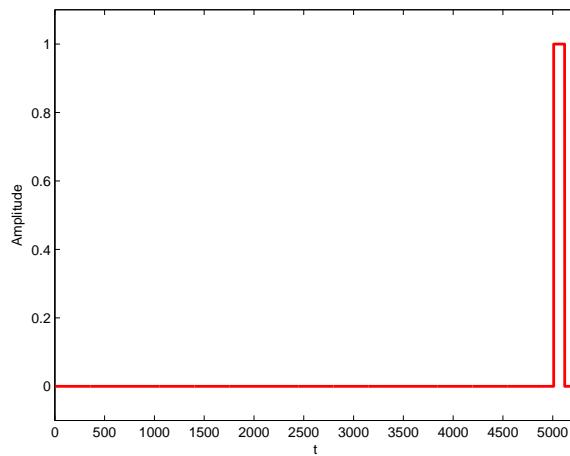


图 4.19 案例4的正常、异常数据段划分结果

图 4.20 案例5中 X 的时间序列图。

于2014年1月24日的一次紧急停机事故。在操作机组降低实发功率 x_1 的同时，操作人员错误地在短时间内提高了给煤量 x_2 ，导致主蒸汽压力 x_3 提高，即产生了 x_3 与其他过程变量变化方向相反的情况。由于主蒸汽流量 x_4 下降，大量内能无处释放，从而提高了主蒸汽压力，直到操作人员发现此异常并通过关闭磨煤机来显著降低给煤量 x_2 ，最终使系统停机。

4.2.2 基于定性趋势组合的工业报警系统设计

本节展示所提出方法进行报警器设计的主要思想。考虑一个离散时间采样的单变量 $x(t)$ ，其中 $t \in \mathbb{Z}$ 为采样时间戳， \mathbb{Z} 为非负整数集合。为了实时地计算时间梯度，记当前时刻的时间戳为 n 。 $t = n$ 时刻的时间梯度定义为在局部的时间窗口 $t \in [n - L + 1, n]$ 内线性回归模型的斜

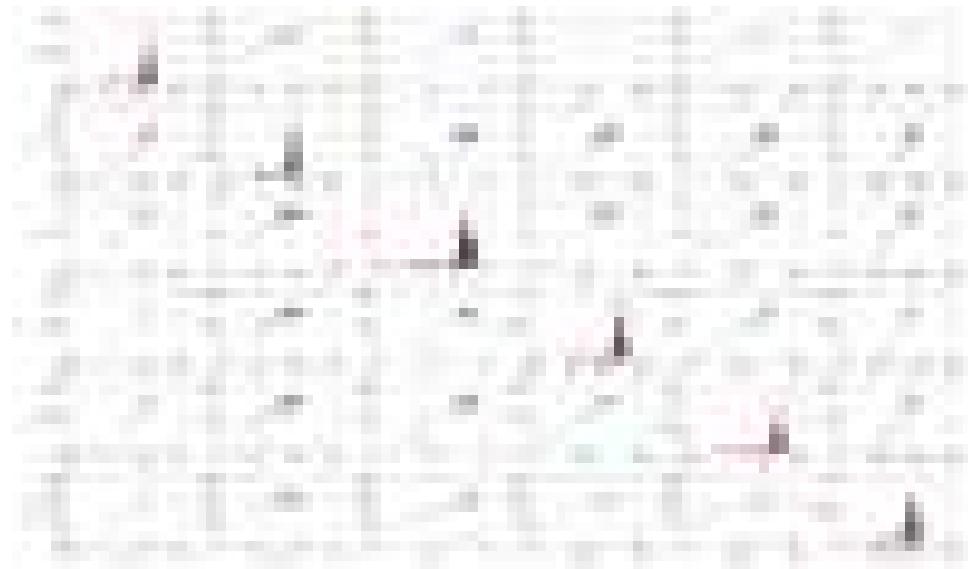


图 4.21 案例5中X的散点图。

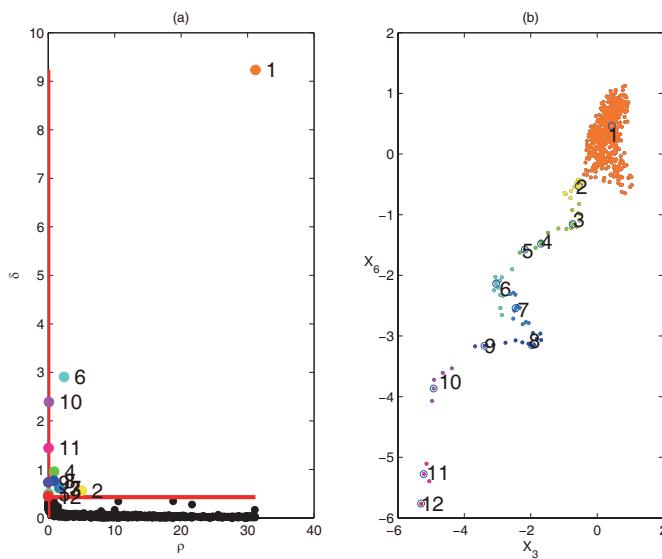


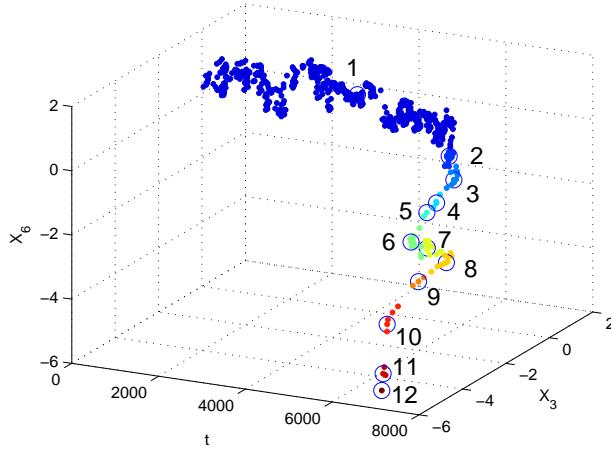
图 4.22 案例5步骤4的决策图及聚类结果

率 $k(n)$, 其中 L 为时间窗大小, 即

$$x(t) = k(n)t + b(n), t \in [n-L+1, n]. \quad (4.12)$$

该局部回归模型的损失函数为

$$G = \sum_{t=n-L+1}^n [x(t) - k(n)t - b(n)]^2 K_\lambda(n, t), \quad (4.13)$$

图 4.23 案例5过程变量 x_3 和 x_6 的聚类结果

其中 $K_\lambda(n, t)$ 为核函数,

$$K_\lambda(n, t) = \lambda^{n-t}, \quad (4.14)$$

其中 $\lambda \in (0, 1)$ 为遗忘因子。所以 λ^{n-t} 不超过1, 且随 t 增加而减少, 即离当前时刻 n 越远, 信息遗忘越严重。

定义时间尺度 C 为权重项 λ^{n-t} 从1衰减到一个非常小的值 η 所需的时间长度, 即

$$\lambda^C = \eta. \quad (4.15)$$

η 的默认值为1%。Bakshi和Stephanopoulos指出定性趋势的分辨率与时间尺度的概念密切相关^[30], 即, 若采用较高的分辨率分析信号的定性趋势, 则将会观察到更多更强的局部变化。因此, 有必要对时间尺度的合理范围进行限制。

令损失函数 G 对 $k(n)$ 和 $b(n)$ 的偏导数为零, 联立方程得到极小化 G 的解为

$$\begin{aligned} \hat{k}(n) &= \frac{\sum_t tx(t) \lambda^{n-i} \sum_t \lambda^{n-t} - \sum_t x(t) \lambda^{n-i} \sum_t t \lambda^{n-t}}{\sum_t t^2 \lambda^{n-t} \sum_t \lambda^{n-t} - \left(\sum_t t \lambda^{n-t}\right)^2}, \\ \hat{b}(n) &= \frac{\sum_t x(t) \lambda^{n-t} \sum_t t^2 \lambda^{n-t} - \sum_t tx(t) \lambda^{n-t} \sum_t t \lambda^{n-t}}{\sum_t t^2 \lambda^{n-t} \sum_t \lambda^{n-t} - \left(\sum_t t \lambda^{n-t}\right)^2}, \end{aligned} \quad (4.16)$$

其中时间截 t 的取值范围为从 $n-L+1$ 到 n 的整数。对于任意当前时刻 n , 都有如式(4.13)所定义的局部加权回归模型来极小化 G , 则当前时刻, $x(n)$ 的估计值为 $\hat{x}(n) = \hat{k}(n)t + \hat{b}(n)$ 。

显然时间梯度 $k(n)$ 与遗忘因子 λ 密切相关。当 λ 较大, 或者等效地 C 较大, 则说明 $k(n)$ 将包含更多的历史数据样本中的信息。因此, 对于当前时刻 n , 信号 $x(t)$ 的时间梯度 T_x 定义为某个合适的时间尺度 C 下的斜率 $k(n)$, 即

$$T_x(n) = k_C(n). \quad (4.17)$$

4.2.3 多变量报警器设计步骤

考虑包含多个过程变量的多变量报警器, $X := [x_1, x_2, \dots, x_m]$ 。每个过程变量 x_i 的时间序列在 t 时刻的值为 $x_i(t)$, $i = 1, 2, \dots, m$ 。 x_i 的定性趋势可以表示为时间梯度的函数, 即

$$\text{sign}(T_{x_i}) := \begin{cases} +1 & T_{x_i} \in (\mu, +\infty) \\ 0 & T_{x_i} \in [-\mu, \mu] \\ -1 & T_{x_i} \in (-\infty, -\mu) \end{cases}, \quad (4.18)$$

其中 T_{x_i} 是 x_i 的时间梯度, μ 为显著性水平。换言之, 当 $\text{sign}(T_{x_i})$ 分别取值为+1, 0或-1时, 表示 x_i 的定性趋势分别为增加、不变和减少。

若 X 处于正常状态, 则 x_1, x_2, \dots, x_m 的定性趋势的取值应当存在固定的关系, 可以通过下式表示,

$$f(\text{sign}(T_{x_1}), \text{sign}(T_{x_2}), \dots, \text{sign}(T_{x_m})) = 0. \quad (4.19)$$

与此相反, 若 X 处于异常状态, 则公式(4.19)所定义的正常关系将不再成立, 即

$$f(\text{sign}(T_{x_1}), \text{sign}(T_{x_2}), \dots, \text{sign}(T_{x_m})) \neq 0. \quad (4.20)$$

式(4.20)所定义的函数关系通常可以通过关于 X 的先验知识得到。

基于上述假设, 所提出的方法的目标为计算时间梯度并得到 X 的定性趋势, 进一步基于式(4.20)所定义的异常状态下函数关系产生异常报警信号。

本节首先展示了如何通过自适应时间梯度算法提取单变量的时间梯度, 然后展示了如何结合自适应时间梯度方法设计多变量报警器。首先需要做出两个基本假设如下,

1. 时间序列包含各种时间尺度上的变化。较小的时间尺度对应于快速的变化, 而较大的时间尺度对应于持续时间更长的变化。
2. 为了能够适应于时间序列中各种尺度上的变化, 则必须使用自适应的时间尺度。设当前的时间尺度 $C(n)$ 可以由时间序列的波动率 $V(n)$ 得到, 即

$$C(n) = g(V(n)). \quad (4.21)$$

基于公式(4.15)所定义的 λ 和 C 的函数关系, 可以得到式(4.21)的等效形式,

$$\lambda(n) = f(V(n)). \quad (4.22)$$

定义时间序列的波动率为当前时间窗口内时间梯度的极差, 即,

$$V(n) = \max_{t \in [n-C(n)+1, n]} k(t) - \min_{t \in [n-C(n)+1, n]} k(t). \quad (4.23)$$

时间尺度 C 的范围可由操作人员对系统的先验知识进行确定。最小时度 C_{\min} 的概念来自对操作人员确认异常的观察, 即如果异常发生, 操作人员仍然需要持续观察一段时间以确认当前的异常不是噪声导致的随机现象。因此, C_{\min} 是时间维上区分噪声和最高分辨率的趋势变化的边界。 C_{\min} 的合理性与报警延迟器^[6,12]相似, 只有当过程变量的幅值超过阈值持续多个样本时间才能产生一次报警。根据报警延迟器的最优取值20秒^[22], C_{\min} 的默认值可以取

为 $C_{\min} = 20$ 。最大时间尺度 C_{\max} 与大尺度上的趋势变化有关。最大时间尺度 C_{\max} 的概念来自于观察到某些事件对系统的影响具有一定的持续时间。通常 C_{\max} 可以取历史时间序列中保持某一趋势数据段的最大长度。

当 $V(n)$ 较小时， $\lambda(n)$ 接近于 $\eta^{\frac{1}{C_{\max}}}$ 并且 $C(n)$ 被限定不超过 C_{\max} 。相反，当 $V(n)$ 取值较大时， $\lambda(n)$ 接近于 $\eta^{\frac{1}{C_{\min}}}$ 并且 $C(n)$ 不少于 C_{\min} 。事实表明，较小(大)的时间尺度更适合于较大(小)的波动率。因此，公式(4.22)可以表示为

$$\lambda(n) = \max \left(-\beta_0 \cdot \max(0, V(n) - V_0) + \eta^{\frac{1}{C_{\max}}}, \eta^{\frac{1}{C_{\min}}} \right), \quad (4.24)$$

其中参数 β_0 和 V_0 将通过下一节的方法从历史数据中求得。

下面展示如何通过历史数据确定式(4.24)中的参数 β_0 、 V_0 和式(4.18)中的参数 μ 。首先选择一段历史数据 $\{s(t)\}_{t=1}^l := [s(1), s(2), \dots, s(l)]$ 作为训练数据， $l \geq L$ 。函数(4.24)描述了时间尺度和波动率之间的定量关系。信号的波动率由噪声和趋势变化共同影响。而操作人员需要选择时间尺度的范围 $[C_{\min}, C_{\max}]$ 仅由趋势变化决定。因此拟合式(4.24)的主要目的在于对噪声造成的波动率进行建模。因此选择历史时间序列 $s(t)$ 的要求是时间序列在小于 C_{\min} 的时间尺度上无明显趋势变化。基于此，需要忽略小于 C_{\min} 的时间尺度的趋势变化，并且要假设在历史数据中建模得到的噪声变化规律与在线过程中的噪声变化规律是相同的。

为了拟合函数式(4.24)，需要计算历史数据中的波动率。设时间尺度 $C_p = C_{\min} + p - 1$ ， $p = 1, 2, \dots, C_{\max} - C_{\min} + 1$ ，并有 $\lambda_p = \eta^{\frac{1}{C_p}}$ 。使用式(4.16)，对于每个样本点 $s(j)$ 都可以计算时间梯度 $k_{C_p}(j)$ 为

$$k_{C_p}(j) = \frac{\sum_t ts(t) \lambda_p^{j-t} \sum_t \lambda_p^{j-t} - \sum_t s(t) \lambda_p^{j-t} \sum_t t \lambda_p^{j-t}}{\sum_t t^2 \lambda_p^{j-t} \sum_t \lambda_p^{j-t} - \left(\sum_t t \lambda_p^{j-t} \right)^2}, \quad (4.25)$$

其中时间取值范围为 $j - L + 1$ 到 j 的整数。由于最大时间尺度限定为 C_{\max} ，则式(4.12)中的局部回归模型所使用的时间窗口不得超过 C_{\max} 。由于遗忘因子 λ_p 是自适应变化的，可令 $L = C_{\max}$ 。则 $s(j)$ 在时间尺度 C_p 上的波动率 $V_{C_p}(j)$ 为

$$V_{C_p}(j) = \max_{t \in [j-C_p+1, j]} k_{C_p}(t) - \min_{t \in [j-C_p+1, j]} k_{C_p}(t). \quad (4.26)$$

遍历 p 和 j 计算 $k_{C_p}(j)$ 和 $V_{C_p}(j)$ 可以得到各个数据点在各个时间尺度上的波动率和时间梯度。如此， $k_{C_p}(j)$ 和 $V_{C_p}(j)$ 均为二维数据矩阵，行标为 j ，列标为 p 。

画出 $V_{C_p}(j)$ 的每一行随 C_p 或 λ_p 的散点图，图4.24是案例3所使用的工业环境下泵速的示例。这些散点的包络线表示给定时间尺度 $C_p^{(Env)}$ 或相应的权重 $\lambda_p^{(Env)}$ 时的最大波动率 $V_p^{(Env)}$ 。给定时间尺度 $C_p^{(Env)}$ ，任何波动率小于 $V_p^{(Env)}$ 的数据点都对应更平缓的趋势变化，因而应当采用比 $C_p^{(Env)}$ 更大的时间尺度。对于任何时间尺度，只有拥有最大波动率的样本点才能加入对式(4.24)的拟合计算中。因此，包络线记为 $\{\lambda_p^{(Env)}, V_p^{(Env)}\}$ 或 $\{C_p^{(Env)}, V_p^{(Env)}\}$ ，通过下式进行计算，

$$\begin{cases} \lambda_p^{(Env)} := \eta^{\frac{1}{C_{\min} + p - 1}} \\ C_p^{(Env)} := C_{\min} + p - 1 \\ V_p^{(Env)} = \max_{j \in [C_{\max}, l]} (V_{C_p^{(Env)}}(j)) \end{cases}, \quad (4.27)$$

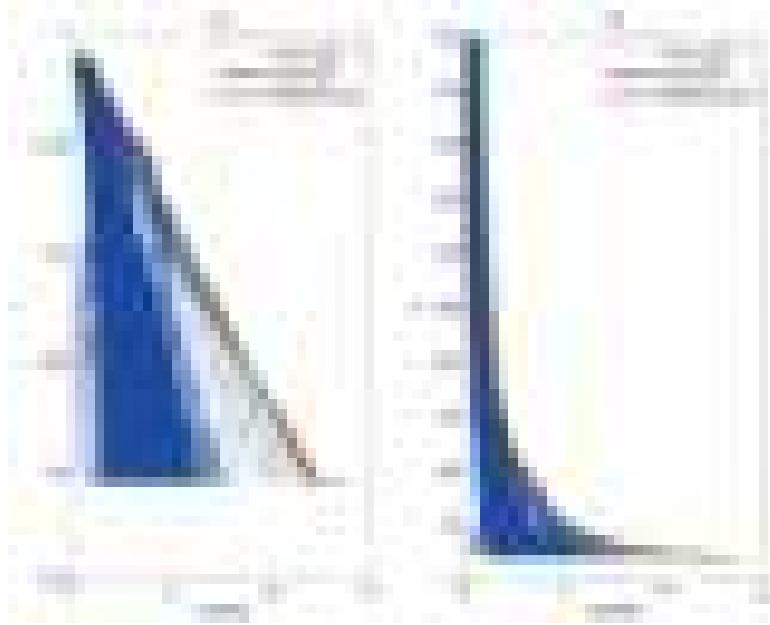


图 4.24 波动率 $V_{C_p}(j)$ 与遗忘因子 λ_p 和时间尺度 C_p 的散点图

其中 $p = 1, 2, \dots, C_{\max} - C_{\min} + 1$ 。使用最小二乘法将式(4.24)拟合到包络线 $\{\lambda_p^{(Env)}, V_p^{(Env)}\}$ 上, 得到参数的最优解为

$$\left\{ \begin{array}{l} V'_0 = V_{C_{\max} - C_{\min} + 1}^{(Env)} \\ \beta'_0 = \arg \min_{\beta_0} \sum_{p=1}^{C_{\max} - C_{\min} + 1} \left[f(V_p^{(Env)}, \beta_0, V'_0) - \lambda_p^{(Env)} \right]^2 \end{array} \right. . \quad (4.28)$$

拟合优度定义为

$$\rho = 1 - \frac{\sum_{p=1}^{C_{\max} - C_{\min} + 1} \left(\lambda_p^{(Env)} - \hat{\lambda}_p^{(Env)} \right)^2}{\sum_{p=1}^{C_{\max} - C_{\min} + 1} \left(\lambda_p^{(Env)} - \bar{\lambda}_p^{(Env)} \right)^2}, \quad (4.29)$$

其中 $\hat{\lambda}_p^{(Env)} = f(V_p^{(Env)}, \beta'_0, V'_0)$, $\bar{\lambda}_p^{(Env)} = \frac{1}{C_{\max} - C_{\min} + 1} \sum_{i=1}^{C_{\max} - C_{\min} + 1} \lambda_i^{(Env)}$ 。一般地, 当 $\rho > 0.8$ 时表明当前的拟合程度是可以接受的。

式(4.24)中的参数 μ , 是为避免将噪声带来的时间梯度的小幅变化识别为增加或者减少的定性趋势。若时间梯度取值位于 $[-\mu, \mu]$ 中则可认为过程变量保持不变的定性趋势, 不变的定性趋势表明波动率较低, 据式(4.24)可知最小的波动率对应于最大的时间尺度。因此式(4.18)中的显著性阈值 μ 取为

$$\mu' = 2\text{std}_j(k_{C_{\max}}(j)), \quad (4.30)$$

其中, $k_{C_{\max}}(j)$ 来自于式(4.25), $j \in [C_{\max}, l]$ 。式(4.30)来自于对零均值正态分布的显著性假设检验, 取显著性水平为 0.05。

所提出的ATG算法包含两个部分, 首先使用历史数据离线计算参数 $\{V_0, \beta_0, \mu\}$ 的取值, 然

后在线实时计算时间梯度 $T_x(n)$ 。

首先指定时间尺度范围 $[C_{\min}, C_{\max}]$, 并且选择一段历史数据 $s(t)$, 基于式(4.28)和(4.30)计算参数 $\{V_0, \beta_0, \mu\}$ 取值。若式(4.29)中的拟合优度 ρ 少于0.8, 则要更换一段历史数据, 并重新计算参数取值。

对于在线过程, 每当新数据点 $x(n)$ 可用时便自动执行如下步骤:

步骤1: 使用上一时刻的时间尺度 $C(n-1)$ 或等效的遗忘因子 $\lambda(n-1)$ 计算式(4.16)中的时间梯度, 如下,

$$\begin{aligned} T_x(n) &= k_{C(n-1)}(n) \\ &= \frac{\sum_t t x(t) \lambda(n-1)^{n-t} \sum_t \lambda(n-1)^{n-t} - \sum_t x(t) \lambda(n-1)^{n-t} \sum_t t \lambda(n-1)^{n-t}}{\sum_t t^2 \lambda(n-1)^{n-t} \sum_t \lambda(n-1)^{n-t} - \left(\sum_t t \lambda(n-1)^{n-t}\right)^2}, \end{aligned}$$

其中求和过程的时间 t 遍历范围是 $n - C_{\max} + 1$ 到 n 的整数。

步骤2: 更新当前时间尺度,

$$C(n) = \min \left\{ C_{\max}, \frac{\log(\eta)}{\log(\lambda(n-1))} \right\}.$$

步骤3: 基于更新后的时间尺度 $C(n)$ 计算当前数据点的波动率,

$$V(n) = \max_{t \in [n - C(n) + 1, n]} k(t) - \min_{t \in [n - C(n) + 1, n]} k(t).$$

步骤4: 更新遗忘因子,

$$\lambda(n) = \max \left(-\beta'_0 \cdot \max(0, V(n) - V'_0) + \eta^{\frac{1}{C_{\max}}}, \eta^{\frac{1}{C_{\min}}} \right).$$

使用所提出的ATG算法可以在线实时地获取多变量系统 $X = [x_1, x_2, \dots, x_m]$ 中各个过程变量 x_i 的时间梯度, $i = 1, 2, \dots, m$ 。过程变量 x_i 的定性趋势可以通过式(4.18)得到。基于此, 构建如下的多变量报警器。

为方便多个过程变量的定性趋势变化与式(4.19)所定义的关系进行对比, 定义大小为 $r \times m$ 的二维矩阵 R ,

$$R := \begin{bmatrix} \text{sign}(T_{1,x_1}) & \text{sign}(T_{1,x_2}) & \cdots & \text{sign}(T_{1,x_m}) \\ \text{sign}(T_{2,x_1}) & \text{sign}(T_{2,x_2}) & \cdots & \text{sign}(T_{2,x_m}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{sign}(T_{r,x_1}) & \text{sign}(T_{r,x_2}) & \cdots & \text{sign}(T_{r,x_m}) \end{bmatrix}. \quad (4.31)$$

R 的每一行为式(4.20)所定义的函数关系的一个解。例如, 假设仅考虑增加和减少两种定性趋势, 则对于正常状态下同增同减的双变量系统 $X := [x_1, x_2]$, 异常状态即为两个过程变量的变

Algorithm 5 判断系统当前时刻是否正常

```

输入参数#1: sign ( $T_X (n)$ ), 变向矢量;
输入参数#2:  $R$ ,  $X$ 的关系矩阵;
 $I(n) = zeros (r, 1); % r 是  $R$  中的行数;$ 
for  $i = 1$  to  $r$  do
     $I(n) = (\text{sign} (T_X (n)) == R(i,:));$ 
end for
if  $\text{sum} (I(n)) == 0$  then
     $X(n)$  处于正常状态,  $X_a(n) = 0$ ;
else
     $X(n)$  处于异常状态,  $X_a(n) = 1$ ;
end if

```

化方向相反, 因此矩阵 R 表示为

$$R = \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}. \quad (4.32)$$

基于各过程变量的时间梯度, 定义所有过程变量在当前时刻 n 的定性趋势组合为如下向量,

$$\text{sign} (T_x (n)) = [\text{sign} (T_{x_1} (n)), \text{sign} (T_{x_2} (n)), \dots, \text{sign} (T_{x_m} (n))]. \quad (4.33)$$

算法5将定性趋势组合向量 $\text{sign} (T_x (n))$ 与关系矩阵 R 对比以判断系统当前时刻 $x(n)$ 是否正常。具体地, 如果定性趋势组合向量与矩阵 R 的一行相一致, 则表明 $x(n)$ 处于异常状态, 需要产生报警变量 $x_a (n) = 1$ 。如果定性趋势组合与矩阵 R 的每一行都不符合, 则 $x(n)$ 应当处于正常状态, 报警变量取值为0, 即 $x_a (n) = 0$ 。

4.2.4 案例分析

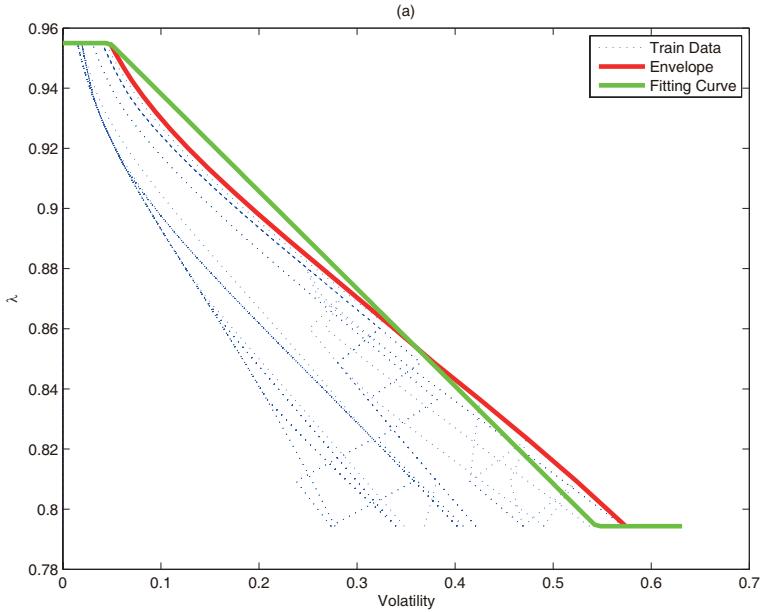
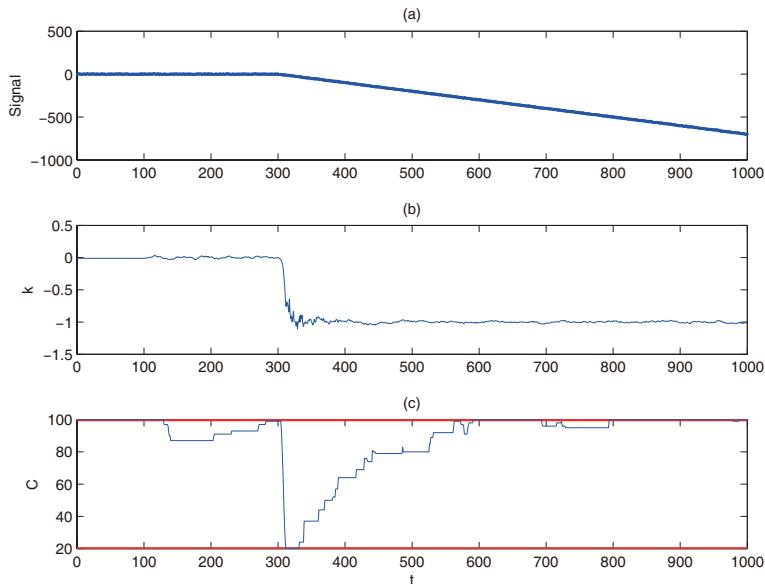
本节共展示三个案例。第一个数值仿真案例展示了ATG算法相对于文献中方法的有效性, 第二个案例展示了针对水箱仿真系统设计的多变量报警器, 第三个案例展示了工业环境下一个变频泵设计的报警器。

案例1: 考虑一个梯形信号, 定义如下:

$$x(t) = \begin{cases} e(t), & \text{for } t \in [1, 300] \\ -1 * (t - 300) + e(t), & \text{for } t \in [300, 1000] \end{cases},$$

其中 $e(t)$ 为零均值单位方差的高斯白噪声。图4.25展示了ATG算法的离线计算步骤的过程, $C_{\min} = 20$, $C_{\max} = 100$, $s(t) = x(1 : 200)$ 。公式(4.24)中参数 β_0 和 V_0 、公式(4.18)中的参数 μ 及公式(4.29)的拟合优度 ρ 计算结果如下:

$$\left\{ \begin{array}{l} V_0 = 0.0481 \\ \beta_0 = 0.3242 \\ \mu = 0.0073 \\ \rho = 0.9843 \end{array} \right.,$$

图 4.25 案例1中波动率 $V_C(j)$ 与 λ 的散点图图 4.26 原始信号 $x(t)$ 、时间梯度 $k_C(t)$ 与时间尺度 $C(t)$ 的时间序列图

其中 $\rho = 0.9843$ 表明拟合程度较高可以接受。图4.26展示了原始时间序列 $x(t)$ 、时间梯度 $k_C(t)$ 和时间尺度 $C(t)$ 的时间序列图，由图可知，ATG 算法能够跟踪 $t = 300$ 时刻的显著变化。因此，此时时间尺度 $C(t)$ 快速衰减到 C_{\min} 附近，而当 $t = 300$ 之后， $x(t)$ 保持稳定下降的定性趋势，时间尺度 $C(t)$ 逐渐增加到 C_{\max} 附近，并且时间梯度 $k_C(t)$ 的波动减小而变得平滑。

图4.26的结果可以与无噪声状态的时间梯度相对比，如图4.27所示。一种常用的Savitzky-

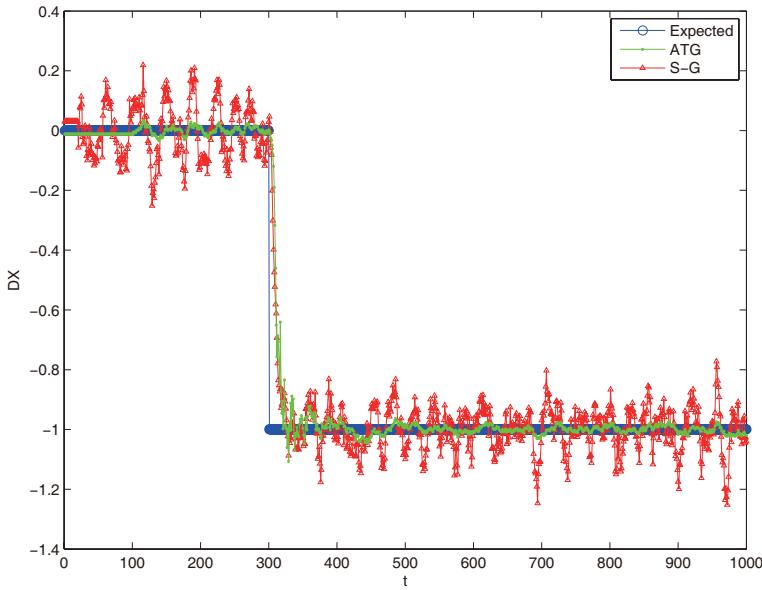


图 4.27 ATG 算法与 Savitzky-Golay 滤波器效果比较

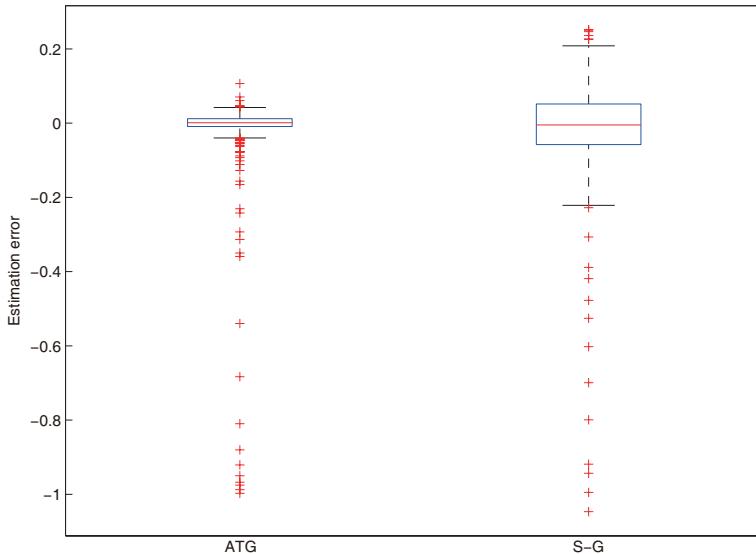


图 4.28 ATG 算法与 Savitzky-Golay 滤波器对时间梯度估计误差箱线图

Golay 滤波器^[23]的一阶导数计算结果也作为对比。Savitzky-Golay 滤波器在计算时间梯度时使用固定的时间窗口，选择时间尺度下界 C_{\min} 作为时间窗口大小。尽管 Savitzky-Golay 滤波器和 ATG 算法都能在 $t = 300$ 时刻快速响应 $x(t)$ 的变化，ATG 算法能够得到比 Savitzky-Golay 滤波器更平滑的时间梯度序列，从图 4.28 所示的估计误差的箱线图可以观察得到。ATG 算法能够取得比 Savitzky-Golay 滤波器更好的估计能力原因在于自适应的时间尺度。因此，当信号中出现较高（低）的波动率时，ATG 算法能够使用更小（大）的时间尺度来跟踪（平滑）信号。

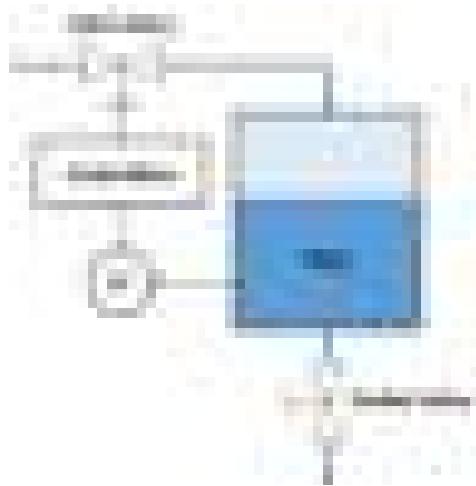
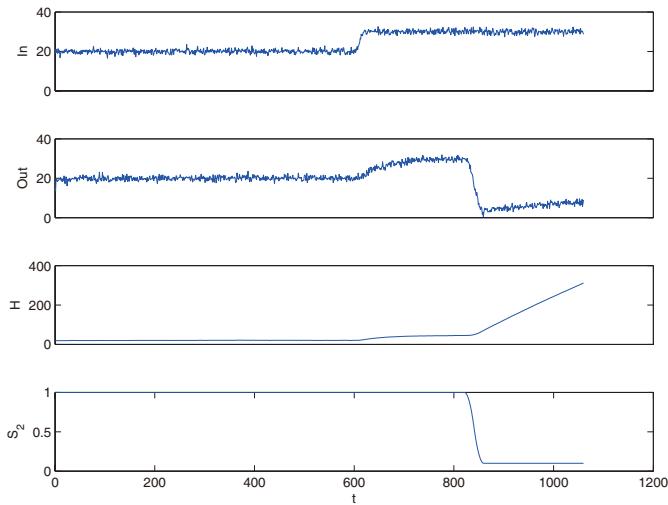


图 4.29 水箱仿真系统的示意图

图 4.30 入水流量 F_i 、出水流量 F_o 、液位高度 H 和出水阀门截面积 S_2 时间序列图

案例2：考虑如图4.29所示的一个水箱仿真系统。该系统可以通过如下动力学方程描述，

$$\begin{cases} S_1 \frac{dH}{dt} = F_i - F_o \\ F_o = S_2 \cdot \sqrt{2gH} + e \end{cases}, \quad (4.34)$$

其中 F_i 和 F_o 分别表示入水阀门的流量和出水阀门的流量， H 为液位高度， S_1 和 S_2 分别表示水箱和出水阀门的截面积， g 为重力常数， e 表示系统中存在的噪声。通过控制入水流量可以控制水箱中液位高度 H 到指定的位置，出水流量 F_o 与液位高度 H 相关。因此，式(4.34)表明当系统处于正常状态时 F_o 和 H 保持同增同减的关系。图4.30展示了 F_i ， F_o 和 S_2 的仿真运行数据。 $t = 600$ 时入水流量 F_i 增加，导致出水流量 F_o 和液位高度 H 增加， $t = 821$ 时出水阀门产生了堵塞的现象，则从 $t = 821$ 到 $t = 860$ 出水阀门面积 S_2 快速衰减到较低值。因此，通过检

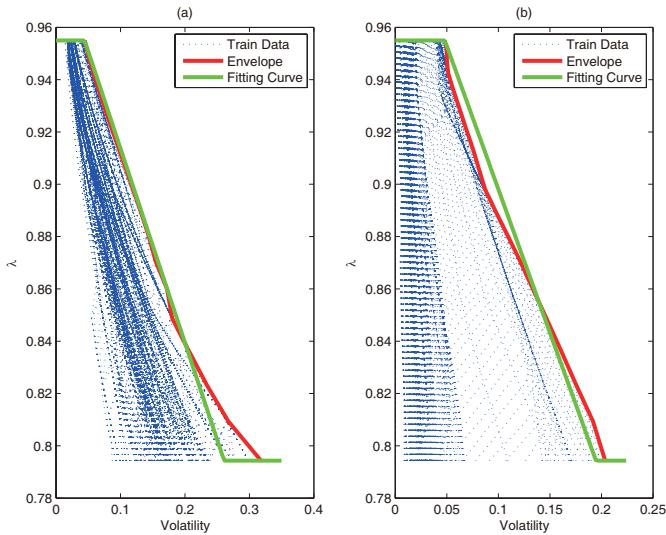
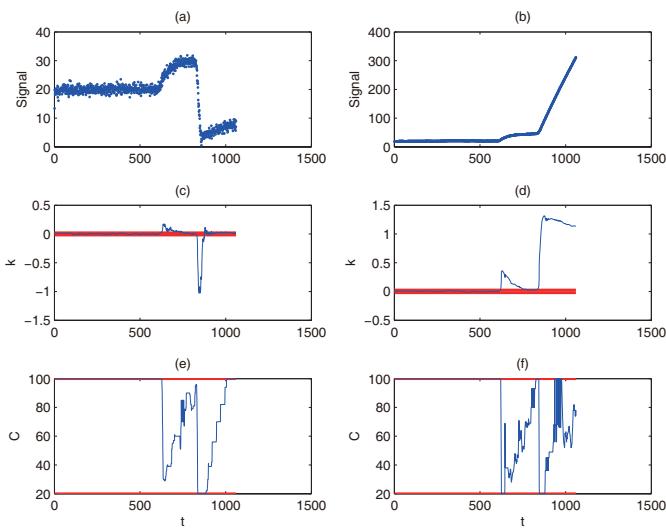
图 4.31 F_o 和 H 的历史数据波动率 $V_C(j)$ 与遗忘因子 λ 的散点图

图 4.32 出口流量、液位高度及对应的时间梯度和时间尺度时间序列图

查 F_o 和 H 的定性趋势，可以发现 $t \in [821, 860]$ 时产生了异常。

所提出的方法应用于系统 $X = [F_o, H]$ ，并取历史数据 $s(t) = X(1 : 400, :)$ ，最短时间尺度 $C_{\min} = 20$ ，最大时间尺度 $C_{\max} = 100$ ，并且根据先验知识可得到 R 与式(4.32)相同。图 4.31 展示了 F_o 和 H 的离线计算过程。对于 F_o 和 H ，式(4.24)中的参数 β_0 、 V_0 ，式(4.18)中的 μ 及

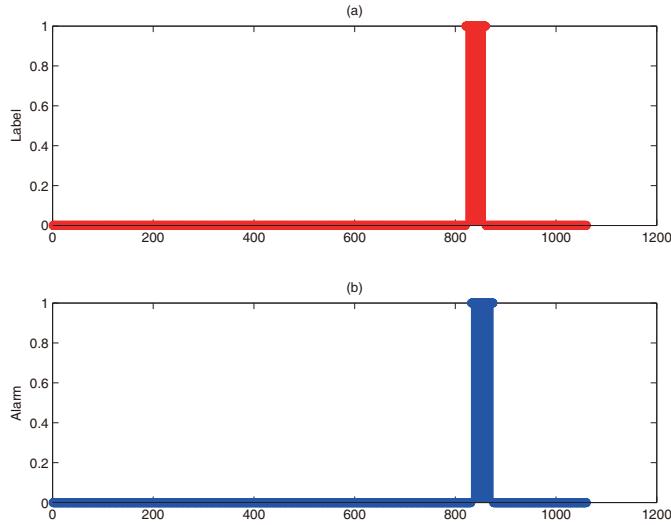
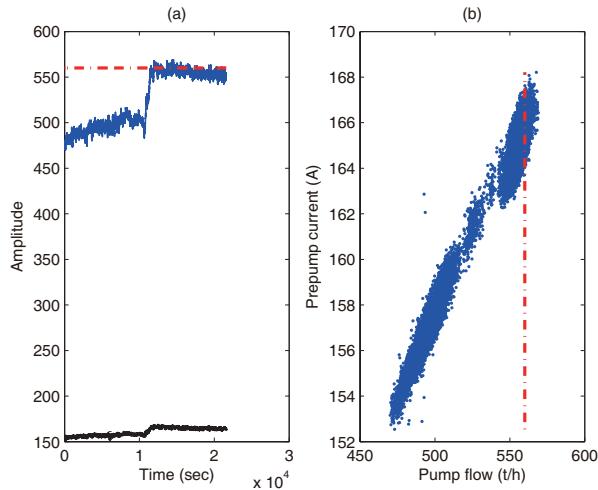
图 4.33 真实异常信号及多变量报警器产生的报警变量 $X_a(t)$ 

图 4.34 入水流量和前置泵电流的时间序列图及散点图

式(4.29)中的 ρ 分别如下：

$$\begin{cases} V_{0,F_o} = 0.0444 \\ \beta_{0,F_o} = 0.7425 \\ \mu_{F_o} = 0.0134 \\ \rho_{F_o} = 0.9919 \end{cases}, \begin{cases} V_{0,H} = 0.0481 \\ \beta_{0,H} = 1.0961 \\ \mu_H = 0.0161 \\ \rho_H = 0.9678 \end{cases}.$$

可以看到，两个时间序列的历史数据拟合优度 ρ_{F_o} 和 ρ_H 都非常接近于1，表明式(4.24)的拟合效果非常好。图4.32展示了时间梯度 $k_{F_o}(t)k_H(t)$ 以及时间尺度 $C_{F_o}(t)$ 和 $C_H(t)$ 。图4.32(c)和图4.32(d)中的水平线表示式(4.18)所定义的显著性阈值 $[-\mu, \mu]$ 。图4.32(e)和图4.32(f)中的水平

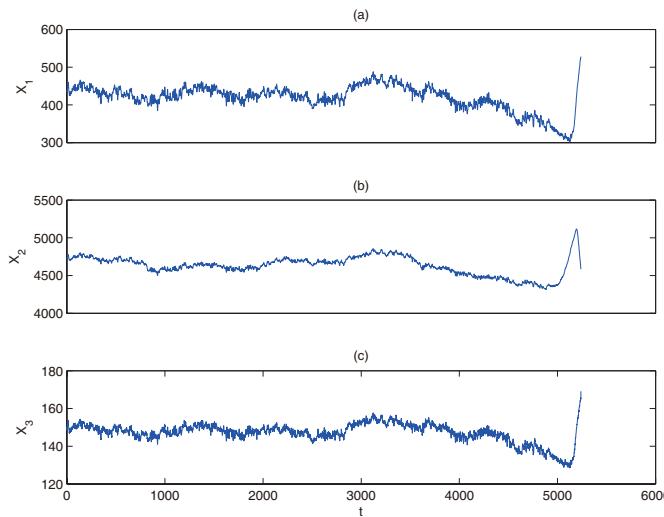


图 4.35 入水流量、泵转速和前置泵电流时间序列图

线表示时间尺度范围 $[C_{min}, C_{max}]$ 。由图可知，当 $F_o(t)$ 和 $H(t)$ 处于不变的定性趋势时，时间尺度 $C_{F_o}(t)$ 和 $C_H(t)$ 取得最大值 C_{max} ，而当 $F_o(t)$ 和 $H(t)$ 快速变化时，时间尺度也迅速减少。如图4.33所示，该系统在 $t \in [832, 875]$ 期间产生了报警，与发生出口阀门堵塞现象的时期 $t \in [821, 860]$ 非常接近。由于二者时差小于 $C_{min} = 20$ ，因此所设计的多变量报警器能够及时地检测出 F_o 和 H 所构成的双变量系统 X 中的异常。

案例3：本案例研究ATG算法在工业实际中的应用，数据来源于某大型火力发电厂的给水泵。目前针对该泵的一个报警变量是检测入水流量是否超过560吨/小时，若超过则产生报警。当入水流量需要超过560吨/小时给下游设备提供更多水时，这样的一个单变量报警器经常导致误报警。如图4.34所示，尽管入水流量超过了高报警线560吨/小时，入水流量和前置泵电流之间的关系仍然保持不变。因此该给水泵仍然处于正常状态，可以认为该单变量报警器产生的报警均为干扰报警。此外，该单变量报警器也有可能产生漏报警现象，将通过图4.35展示。

为了避免当前使用的单变量系统的误报警和漏报警现象，使用本节提出的方法对该泵构建了三变量的报警器 $X := [x_1, x_2, x_3]$ ，其中 x_1 ， x_2 和 x_3 分别是入水流量，泵转速和前置泵电流。基于泵的运行规律可知，当系统处于正常状态时这三个过程变量应当保持同增同减的形式，因而关系矩阵 R 如下：

$$R = \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & +1 & -1 & 0 & +1 & -1 & +1 & +1 & -1 & 0 & +1 \\ -1 & +1 & -1 & 0 & -1 & -1 & +1 & +1 & -1 & +1 & +1 & 0 \end{bmatrix}^T.$$

图4.35展示了2014年1月24日这三个过程变量的时间序列图，采样频率为每秒1次。由图可以看出，在 $t = 5035$ 之前系统 X 处于正常状态，而此后产生了两次异常。第一次于 $t \in [5035, 5160]$ ， x_2 增加而 x_1 和 x_3 减少。第二次于 $t \in [5195, 5239]$ ， x_2 减少而其他变量保持增加。这些结果可以通过离线的人工分析得到，将作为后面在线结果的基准。但这些异常识别结果无法使用当前的单变量报警器检测得到，因而会出现漏报警现象。

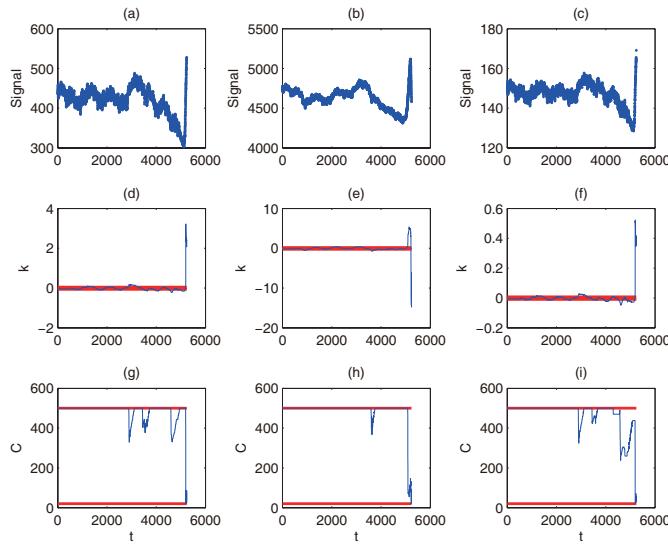
图 4.36 历史数据波动率 $V_C(j)$ 与遗忘因子 λ 的散点图

图 4.37 入水流量、泵转速和前置泵电流及其对应的时间梯度和时间尺度时间序列图

采用所提出的方法应用于该系统。参数 $C_{\min} = 20$, $C_{\max} = 500$ 和 $s(t) = X : (1, 2000)$ 。图4.36展示了离线步骤的计算结果。 x_1 , x_2 和 x_3 的拟合函数参数和拟合优度如下:

$$\begin{cases} V_{0,x_1} = 0.2096 \\ \beta_{0,x_1} = 0.0629 \\ \mu_{x_1} = 0.1093 \\ \rho_{x_1} = 0.9886 \end{cases}, \begin{cases} V_{0,x_2} = 0.8608 \\ \beta_{0,x_2} = 0.0178 \\ \mu_{x_2} = 0.4496 \\ \rho_{x_2} = 0.9962 \end{cases}, \begin{cases} V_{0,x_3} = 0.0319 \\ \beta_{0,x_3} = 0.3711 \\ \mu_{x_3} = 0.0173 \\ \rho_{x_3} = 0.9877 \end{cases}. \quad (4.35)$$

拟合优度 ρ_{x_1} , ρ_{x_2} 和 ρ_{x_3} 均接近于1, 表明式(4.24)的拟合效果很好。图4.37展示了 x_1 , x_2 和 x_3 的

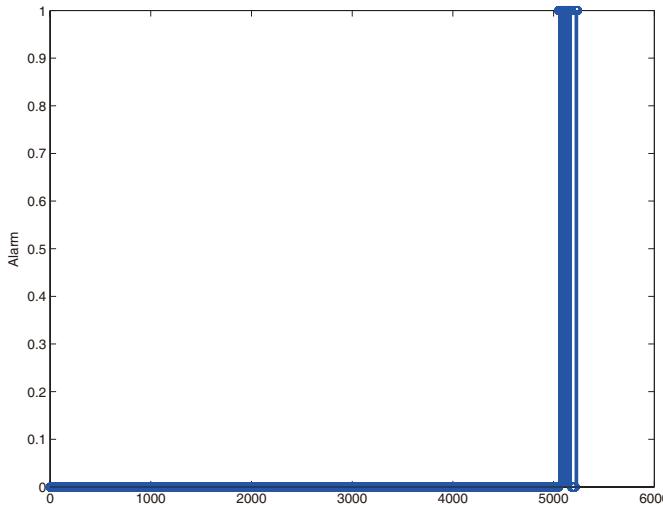
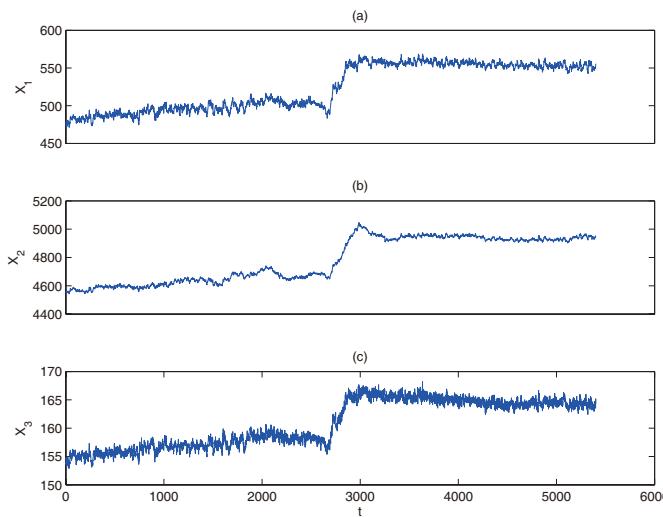
图 4.38 图4.37中所示的数据的报警变量 $x_a(t)$ 

图 4.39 入水流量、泵转速和前置泵电流时间序列图

时间梯度和时间尺度。图4.37(c)和图4.37(d)中的水平线表示式(4.18)所定义的显著性阈值 $[-\mu, \mu]$ 。图4.37(e)和图4.37(f)中的水平线表示时间尺度范围 $[C_{min}, C_{max}]$ 。在 $t = 5000$ 之前，该系统中各过程变量的波动率都较小因而使用较大时间尺度来计算时间梯度。而在 $t = 5080$ 附近出现了较大幅度的变化，导致时间尺度快速减少。根据所得的定性趋势产生报警变量如图4.38所示。由图可知， $t \in [5051, 5178]$ 和 $t \in [5215, 5239]$ 为报警时段。该结果十分接近于离线分析结果。

该多变量报警器能够避免图4.34所示的误报警。图4.39展示了相应的时间序列数据。公式(4.35)的离线计算结果如图4.40 所示， x_1 不断增加而在 $t = 2750$ 时超过报警阈值560吨/小时，与此同时， x_2 与 x_3 也有增加的定性趋势。因此，如图4.41所示，这段数据没有产生任何报警，因而避免了误报警现象。注意图4.40中的时间尺度能够自适应于 $t = 2750$ 附近的快速变化，并能够在其他时期增加使得时间梯度适应于平缓的情况。

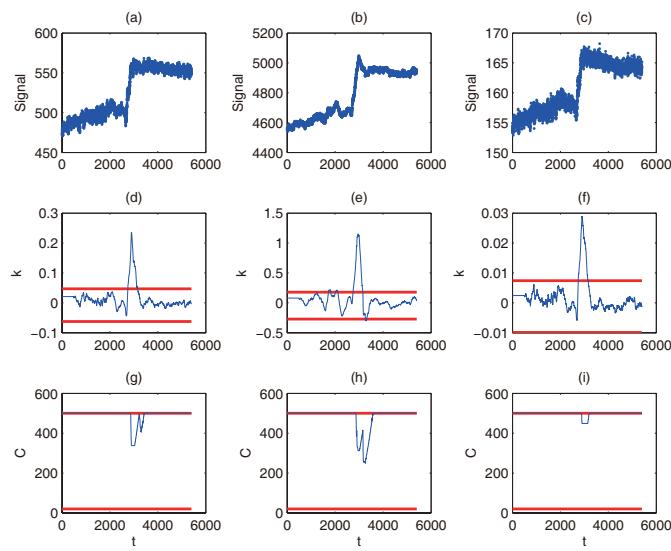


图 4.40 入水流量、泵转速和前置泵电流及其对应的时间梯度和时间尺度时间序列图

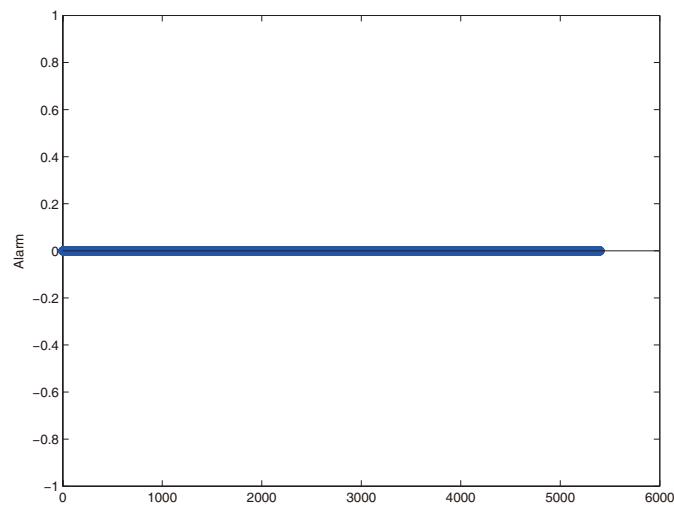


图 4.41 图4.40中所示的数据的报警变量 $x_a(t)$

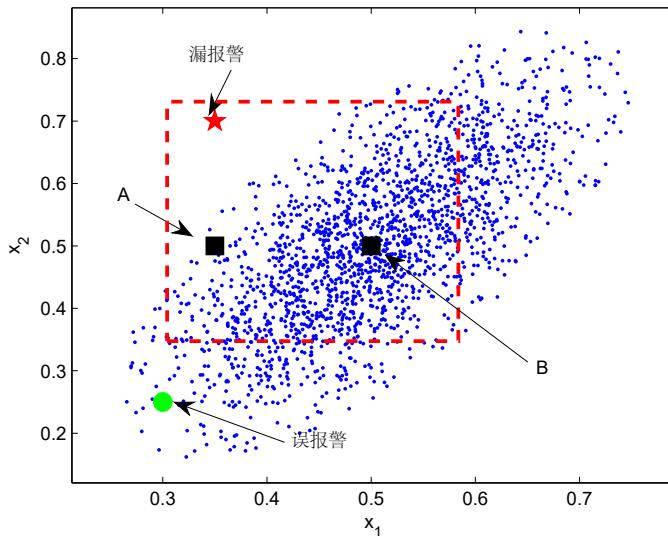


图 4.42 过程变量、报警阈值误报警和漏报警说明

4.3 基于超椭球体的多变量报警器设计

基于超椭球体的多变量报警器设计方法是从历史数据中估计超椭球体模型，并利用该模型来表示系统的正常工作区，然后基于建立的正常工作区，定义和计算各个过程变量的动态报警阈值。通过在平行坐标系中可视化降维缩减空间，可以确定哪些可操纵变量需要调整，并得到它们的调整幅度。

多变量系统中各个过程变量往往是相互关联的。图4.42展示了一个由两个过程变量组成的系统。如果两个过程变量的报警阈值都是固定值，那么由报警阈值定义的系统的正常工作区就是一个矩形，如图4.42中红色的虚线所示。蓝色的样本点占据的区域是系统真实的正常工作区。由于报警阈值定义的系统正常工作区与系统真实的正常工作区不匹配，因此会产生误报警和漏报警。目前，学术界主要基于多元统计分析的思想来设计多变量报警器。其中，应用最广泛的方法是主成分分析法。该方法的主要思想描述如下：假设过程变量服从高斯分布，主成分分析法通过正交变换将过程变量映射成一系列正交的主元，然后挑选特征值较大的若干个主元来构建 T^2 统计量和 Q 统计量。对于一个新的样本点，如果它对应的 T^2 统计量或者 Q 统计量的值小于给定的控制限，就认为这个样本点处于异常状态。

本节提出了一种多变量报警器动态报警阈值的设计方法。该方法的贡献主要有两个：通过利用超椭球体模型对系统的正常工作区进行建模，推导得到了各个过程变量的动态报警阈值的解析表达式；通过固定不可操纵变量的值，得到了降维的正常工作区，进而确定哪些变量需要调整及其调整幅度，并将需要调整的变量调整到它们对应的调整幅度之内以消除报警。

与传统的方法相比，该方法的创新之处有3个。首先，各个过程变量的动态报警阈值都是相互关联的。也就是说，一个过程变量的动态报警阈值会受到其他过程变量取值的影响。因此，样本点不同，对应的动态报警阈值也会发生变化，动态报警阈值是一个关于样本点在空间

中的位置的函数。以图4.42中的样本点A和样本点B为例，由于样本点A靠近系统正常工作区的边界，而样本点B靠近系统正常工作区的中心，因此样本点A和样本点B对应的过程变量 x_1 和过程变量 x_2 的报警阈值应该不同。已有的方法往往将报警阈值设为常数，所以会产生大量的误报警和漏报警。相比之下，该方法设计的动态报警阈值会随着样本点取值的变化而变化，从而使得报警系统的误报率和漏报率很小。其次，该方法是对原始的过程变量设计报警阈值，而多元统计分析方法是对隐变量设计报警阈值。隐变量是一个数学上的抽象概念，不具备明确的物理意义。因此，现场工作人员很难理解和接受隐变量。反之，该方法设计的动态报警阈值与过程变量的物理意义一致，所以现场工作人员很容易理解动态报警阈值的概念，也很容易将动态报警阈值应用到对应的过程变量当中。在实际应用过程中，现场工作人员都是通过经验或试错的方式来确定哪些变量需要调整以及对应的调整幅度。本节提出的方法是利用平行坐标系来确定调整变量和调整幅度，具有很强的理论依据和可操作性。

4.3.1 问题描述

考虑一个由 n 个过程变量 x_1, x_2, \dots, x_n 组成的多变量系统 X 。给定了系统位于正常状态下的历史数据，并将它们定义为

$$\{X(t) \mid t \in [1, N]\}, X(t) = [x_1(t), x_2(t), \dots, x_n(t)],$$

其中， t 表示采样时刻， N 表示样本点的个数。过程变量被认为是服从高斯分布的，并且过程变量处于稳定运行状态。也就是说，新的样本点的分布与历史数据的分布是一致的。在多元统计分析方法中，高斯分布和平稳运行是很常见的假设。本节提出的方法首先要基于各个过程变量之间的关系来为每个过程变量设计动态报警阈值。对于一个新的样本点 $x(t_0)$ ，如果它的各个过程变量的取值 $x_i(t_0)$ 都位于各个过程变量对应的动态报警阈值 $l_i(t_0)$ 和 $h_i(t_0)$ 之间，就认为样本点 $x(t_0)$ 是正常的，否则就认为该样本点是异常的。产生报警以后，接下来就应该采取正确的动作使系统回到正常状态。基于先验知识，将系统的前 m 个过程变量设为不可操纵变量，后 $n - m$ 个过程变量设为可操纵变量。该方法的第二个目标是确定哪些变量需要调整，以及它们的调整幅度。将需要调整的过程变量调整到它们对应的调整幅度之内，就可以消除已经产生的报警。

4.3.2 动态报警阈值的设计

本节首先介绍超椭球体模型的计算方法，并利用超椭球体模型来描述系统的正常工作区，然后介绍各个过程变量的动态报警阈值的计算方法。

正常的样本点位于高维空间中的某个区域之内，定义此区域为正常工作区。建立正常工作区需要考虑误报警和漏报警之间的平衡。如果建立的正常工作区太小，就会产生误报警；反之，如果建立的正常工作区太大，就会产生漏报警。样本点往往集中分布在样本中心附近。并且，离样本中心越远，样本的分布越稀疏。因此，靠近样本中心的样本点用来建立正常工作区，而不是利用所有的样本点来建立正常工作区。聚类算法用来挑选建立正常工作区的样本点，本节采用的是Maestri^[17]等提出的聚类算法，基于该算法来建立正常工作区，步骤如下：

Algorithm 6 正常工作区的建立算法

- 为了防止某些过程变量的取值较大，从而产生错误的计算结果，首先需要对各个过程变量进行预处理。也就是说，将各个过程变量的取值范围标准化为[0, 1]。对于一个原始的样本点 $\tilde{X}(t)$ ，它对应的标准化后的样本点 $X(t)$ 被定义为 $X(t) := [x_1(t), x_2(t), \dots, x_n(t)]$ ，并且

$$x_i(t) = \frac{\tilde{x}_i(t) - \min(\tilde{x}_i)}{\max(\tilde{x}_i) - \min(\tilde{x}_i)},$$

其中， $\min(\tilde{x}_i)$ 和 $\max(\tilde{x}_i)$ 分别表示过程变量 \tilde{x}_i 的最小值和最大值。

- 基于给定的误报率 $p \in (0, 1)$ 确定正常工作区应该包含的正常样本点的个数 r ，计算公式为 $r = \lfloor N - pN \rfloor$ 。其中，函数 $\lfloor \cdot \rfloor$ 为向下取整函数，即取不大于操作数的最大整数。
- 从正常样本点中随机挑选 r 个样本点构成集合 s ，计算集合 s 中的样本点的均值向量 μ_s 和协方差矩阵 W_s ，计算公式为

$$\mu_s = \frac{1}{r} \sum_{X(t) \in s} X(t), \quad (4.36)$$

以及

$$W_s = \sum_{X(t) \in s} [X(t) - \mu_s]^T [X(t) - \mu_s]. \quad (4.37)$$

- 计算样本点 $X(t)$ 到均值向量 μ_s 的马氏距离 $d^2(t)$ ，计算公式为

$$d^2(t) = [X(t) - \mu_s] W_s^{-1} [X(t) - \mu_s]^T,$$

其中， t 的取值为 $1, 2, \dots, N$ 。

- 将马氏距离 $d^2(t)$ 按照从小到大的顺序排列。
- 挑选马氏距离的排序在前 r 个位置的样本点构建一个新的集合 s' ，然后利用式(4.36)和(4.37)分别计算集合 s' 中的样本点的均值向量 $\mu_{s'}$ 和协方差矩阵 $W_{s'}$ 。
- 如果 $\det(W_{s'}) \neq \det(W_s)$ ，就将 W_s 的值更新为 $W_{s'}$ ，以及将 μ_s 的值更新为 $\mu_{s'}$ ，然后迭代执行该算法的步骤3到7。否则，算法终止迭代，并跳转到下一步。其中， $\det(\cdot)$ 表示操作数对应的行列式。
- 利用超椭球体模型对正常工作区进行建模，超椭球体模型的数学表达式为

$$[X(t) - C] P [X(t) - C]^T = 1, \quad (4.38)$$

其中，中心向量 C 和特征矩阵 P 分别等于均值向量 μ_s 和协方差矩阵 W_s^{-1} 。也就是说， $C = \mu_s$ ，以及 $P = W_s^{-1}$ 。

首先定义过程变量的动态报警阈值，然后推导动态报警阈值的计算表达式。产生报警的常见的方法是将过程变量 x_i 的值与两个固定常数 l_i 和 h_i 进行比较，如果 $x_i(t) < l_i$ 或者 $x_i(t) > h_i$ ，那么就产生报警，其中， l_i 和 h_i 分别表示低报警阈值和高报警阈值。在本节中，过程变量的动态报警阈值是基于正常工作区设计的。需要注意的是，“动态”意味着报警阈值的取值随样本点位置的变化而变化，而不是指多变量系统 X 是一个动态系统。对于一个正常的样本点 $X(t)$ ，

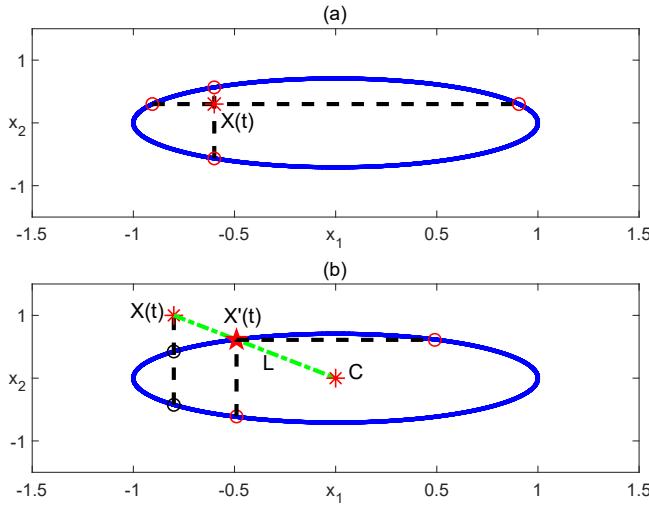


图 4.43 正常、异常样本点及其在椭圆上的投影

沿着各个维度方向在超椭球体模型表面上的投影被定义为该点的各个过程变量的动态报警阈值。通过这种方式，样本点\$X(t)\$的动态报警阈值反映了\$X(t)\$到正常工作区表面的距离，当某些过程变量的取值接近其动态报警阈值时，现场工作人员应该注意系统的运行状态，以防止生产过程出现异常。如果样本点\$X(t)\$是一个异常的样本点，它在超椭球体模型表面上的投影可能不存在。那么，样本点\$X(t)\$对应的正常的样本点\$X'(t)\$的动态报警阈值被定义为\$X(t)\$的动态报警阈值。其中，\$X'(t)\$是线段\$L\$与超椭球体模型表面的交点，线段\$L\$的两个端点分别是样本点\$X(t)\$和超椭球体模型的中心\$C\$。一个异常的样本点的动态报警阈值可以反映该样本点相对于正常工作区的偏离程度。

图4.43(a)展示了正常的样本点的动态报警阈值的计算。其中，星号表示正常的样本点\$X(t)\$，四个圆圈表示\$X(t)\$在椭圆上的投影。因此，过程变量\$x_1\$的动态低报警阈值为\$l_1(t) = -0.90\$，动态高报警阈值为\$h_1(t) = 0.90\$；过程变量\$x_2\$的动态低报警阈值\$l_2(t) = -0.56\$，动态高报警阈值为\$h_2(t) = 0.56\$。图4.43(b)展示了异常的样本点的动态报警阈值的计算。其中，五角星表示异常的样本点\$X(t)\$对应的正常的样本点\$X'(t)\$，两个圆圈表示\$X'(t)\$在椭圆上的投影。因此，过程变量\$x_1\$的动态低报警阈值为\$l_1(t) = -0.49\$，动态高报警阈值为\$h_1(t) = 0.49\$；过程变量\$x_2\$的动态低报警阈值\$l_2(t) = -0.61\$，动态高报警阈值为\$h_2(t) = 0.61\$。

接下来推导动态报警阈值的计算表达式。公式(4.38)中的\$n\$维超椭球体的解析形式可以被改写成

$$F(X(t)) = \begin{bmatrix} X(t) & 1 \end{bmatrix} \bar{P} \begin{bmatrix} X(t)^T \\ 1 \end{bmatrix} = 0, \quad (4.39)$$

其中，

$$\bar{P} = \begin{bmatrix} P & -PC^T \\ -CP & CPC^T - 1 \end{bmatrix}.$$

为了获得过程变量 x_i 的动态报警阈值, 定义

$$U(t) = [x_1(t), x_2(t), \dots, x_{i-1}(t)].$$

以及

$$V(t) = [x_{i+1}(t), x_{i+2}(t), \dots, x_n(t)].$$

于是式(4.39)变成

$$\begin{bmatrix} U(t) & x_i(t) & V(t) & 1 \end{bmatrix} \begin{bmatrix} \bar{P}_{11} & \bar{P}_{12} & \bar{P}_{13} & \bar{P}_{14} \\ \bar{P}_{12}^T & \bar{P}_{22} & \bar{P}_{23} & \bar{P}_{24} \\ \bar{P}_{13}^T & \bar{P}_{23}^T & \bar{P}_{33} & \bar{P}_{34} \\ \bar{P}_{14}^T & \bar{P}_{24}^T & \bar{P}_{34}^T & \bar{P}_{44} \end{bmatrix} \begin{bmatrix} U(t)^T \\ x_i(t) \\ V(t)^T \\ 1 \end{bmatrix} = 0, \quad (4.40)$$

其中,

$$\left\{ \begin{array}{l} \bar{P}_{11} = \bar{P}(1:i-1, 1:i-1), \bar{P}_{12} = \bar{P}(1:i-1, i), \\ \bar{P}_{13} = \bar{P}(1:i-1, i+1:n), \bar{P}_{14} = \bar{P}(1:i-1, n+1), \\ \bar{P}_{22} = \bar{P}(i, i), \bar{P}_{23} = \bar{P}(i, i+1:n), \\ \bar{P}_{24} = \bar{P}(i, n+1), \bar{P}_{33} = \bar{P}(i+1:n, i+1:n), \\ \bar{P}_{34} = \bar{P}(i+1:n, n+1), \bar{P}_{44} = \bar{P}(n+1, n+1). \end{array} \right.$$

并且符号 $\bar{P}(i_1 : i_2, j_1 : j_2)$ 表示矩阵 \bar{P} 的第 i_1 , 第 (i_1+1) , …, 第 i_2 行, 第 j_1 , 第 (j_1+1) , …, 第 j_2 列的元素构成的分块矩阵。式(4.40)是一个关于变量 $x_i(t)$ 的矩阵表达式, 其具体表现形式为

$$a_1 x_i^2(t) + 2a_2 x_i(t) + a_3 = 0, \quad (4.41)$$

其中,

$$\left\{ \begin{array}{l} a_1 = \bar{P}_{22}, \\ a_2 = U(t)\bar{P}_{12} + V(t)\bar{P}_{23}^T + \bar{P}_{24}^T, \\ a_3 = U(t)\bar{P}_{11}U(t)^T + V(t)\bar{P}_{33}V(t)^T + \bar{P}_{44} + 2V(t)\bar{P}_{13}^TU(t)^T + 2U(t)\bar{P}_{14} + 2V(t)\bar{P}_{34}. \end{array} \right.$$

式(4.41)的解为

$$\left\{ \begin{array}{l} l_i(t) = (-a_2 - \sqrt{a_2^2 - a_1 a_3})/a_1, \\ h_i(t) = (-a_2 + \sqrt{a_2^2 - a_1 a_3})/a_1. \end{array} \right. \quad (4.42)$$

因此, 通过将 i 的值分别设为 $1, 2, \dots, n$, 然后利用式(4.42), 就可以得到样本点 $X(t)$ 的动态报警阈值。如果式(4.41)不存在两个不同的解, 那么样本点 $X(t)$ 是一个异常的样本点。它的动态报警阈值等于它对应的正常的样本点 $X'(t)$ 的动态报警阈值。因为 $X'(t)$ 是以样本点 $X(t)$ 和超椭球体中心 C 为端点的线段上的一个点, 所以

$$X'(t) = \theta X(t) + (1 - \theta)C, \quad 0 \leq \theta \leq 1. \quad (4.43)$$

又因为 $X'(t)$ 位于超椭球体的表面上，因此

$$[X'(t) - C]P[X'(t) - C]^T = 1. \quad (4.44)$$

将式(4.43)代入式(4.44)中可以得到

$$[X(t) - C]P[X(t) - C]^T\theta^2 = 1. \quad (4.45)$$

对于一个确定的时间 t ，式(4.45)是一个关于变量 θ 的一元二次方程，解为

$$\begin{cases} \theta_1 = -\sqrt{1/[(X(t) - C)P(X(t) - C)^T]}, \\ \theta_2 = \sqrt{1/[(X(t) - C)P(X(t) - C)^T]}. \end{cases}$$

由于 $0 \leq \theta \leq 1$ ，所以 θ 的最终解为

$$\theta = \sqrt{1/[(X(t) - C)P(X(t) - C)^T]}. \quad (4.46)$$

将式(4.46)代入式(4.43)中，可以得到 $X'(t)$ 的值。然后将 $X'(t)$ 代入式(4.42)中，就可以得到样本点 $X(t)$ 的动态报警阈值。

获得各个过程变量 x_1, x_2, \dots, x_n 的动态报警阈值以后，接下来介绍如何判断样本点 $X(t)$ 是正常的还是异常的。也就是说，如果 $X(t)$ 的各个过程变量的取值都位于它们的动态报警阈值之间，那么就认为样本点 $X(t)$ 是正常的，否则就认为样本点 $X(t)$ 是异常的。除此之外，动态报警阈值还衡量了样本点 $X(t)$ 到正常工作区表面的距离。

4.3.3 可操纵变量的调整

本节介绍如何确定哪些需要调整的可操纵变量，以及它们对应的调整幅度，从而消除已经产生的报警。通过将一个异常的样本点 $X(t)$ 调整到它对应的正常样本点 $X'(t)$ ，可以使该异常样本点 $X(t)$ 回到正常工作区之内。如果某些过程变量不能被操纵，异常的样本点 $X(t)$ 就无法调整到它对应正常的样点 $X'(t)$ 。举例来说明这种现象，在图4.43(b)中，假设过程变量 x_2 可以被操纵，而过程变量 x_1 不能被操纵，不能通过只调整过程变量 x_2 的值将 $X(t)$ 调整到它对应的 $X'(t)$ 。然而，如果将过程变量 x_2 的值调整到范围 $[-0.56, 0.56]$ ，即使过程变量 x_1 的值保持不变，样本点 $X(t)$ 也会回到正常工作区之内。

假设前 m 个过程变量是不可操纵变量，后 $n - m$ 个过程变量是可操纵变量。定义

$$W(t) = [x_1(t), x_2(t), \dots, x_m(t)],$$

以及

$$\bar{X}(t) = [x_{m+1}(t), x_{m+2}(t), \dots, x_n(t)].$$

式(4.39)可以被改写为

$$\begin{bmatrix} W(t) & \bar{X}(t) & 1 \end{bmatrix} \begin{bmatrix} \bar{P}'_{11} & \bar{P}'_{12} & \bar{P}'_{13} \\ (\bar{P}'_{12})^T & \bar{P}'_{22} & \bar{P}'_{23} \\ (\bar{P}'_{13})^T & (\bar{P}'_{23})^T & \bar{P}'_{33} \end{bmatrix} \begin{bmatrix} W(t)^T \\ \bar{X}(t)^T \\ 1 \end{bmatrix} = 0, \quad (4.47)$$

其中,

$$\left\{ \begin{array}{l} \bar{P}'_{11} = \bar{P}(1:m, 1:m), \\ \bar{P}'_{12} = \bar{P}(1:m, m+1:n), \\ \bar{P}'_{13} = \bar{P}(1:m, n+1), \\ \bar{P}'_{22} = \bar{P}(m+1:n, m+1:n), \\ \bar{P}'_{23} = \bar{P}(m+1:n, n+1), \\ \bar{P}'_{33} = \bar{P}(n+1, n+1). \end{array} \right.$$

将 $W(t)$ 的值固定, 公式(4.47)可以被改写成

$$\begin{bmatrix} \bar{X}(t) & 1 \end{bmatrix} \bar{Q} \begin{bmatrix} \bar{X}(t)^T \\ 1 \end{bmatrix} = 0, \quad (4.48)$$

其中,

$$\bar{Q} = \begin{bmatrix} \bar{P}'_{22} & (\bar{P}'_{12})^T W(t)^T + \bar{P}'_{23} \\ W(t) \bar{P}'_{12} + (\bar{P}'_{23})^T & W(t) \bar{P}'_{11} W(t)^T + 2W(t) \bar{P}'_{13} + \bar{P}'_{33} \end{bmatrix}.$$

如果样本点 $X(t)$ 位于式(4.47)中的 n 维超椭球体的内部, 那么 $\bar{X}(t)$ 也应该位于式(4.48)中的 $(n-m)$ 维超椭球体的内部。因此, 将异常的样本点 $X(t)$ 调整到正常工作区之内就等价于将 $\bar{X}(t)$ 调整到式(4.48)中的超椭球体之内。

最后, 平行坐标系用来确定哪些过程变量需要被调整, 以及计算它们对应的调整幅度。式(4.48)中的 $(n-m)$ 维超椭球体被投影到 $(n-m-1)$ 个二维平面内, 分别是 $x_{m+1}x_{m+2}$ 平面, $x_{m+2}x_{m+3}$ 平面, …, $x_{n-1}x_n$ 平面, 从而得到 $(n-m-1)$ 个椭圆。这 $(n-m-1)$ 个椭圆被映射成平行坐标系中的 $(n-m-1)$ 对双曲线, 调整幅度可以通过双曲线的切线得到。具体的计算细节介绍如下。式(4.48)中的 $(n-m)$ 维超椭球体在笛卡尔坐标系中的 $x_i x_{i+1}$ 平面内的投影是一个椭圆^[16], 其解析表达式为

$$f([x_i(t), x_{i+1}(t)]) = \begin{bmatrix} x_i(t) & x_{i+1}(t) & 1 \end{bmatrix} A \begin{bmatrix} x_i(t) \\ x_{i+1}(t) \\ 1 \end{bmatrix} = 0, \quad (4.49)$$

其中,

$$A = \begin{bmatrix} \bar{Q}_{22} - \bar{Q}_{12}^T \bar{Q}_{11}^{-1} \bar{Q}_{12} & \bar{Q}_{23} - \bar{Q}_{12}^T \bar{Q}_{11}^{-1} \bar{Q}_{13} \\ (\bar{Q}_{23} - \bar{Q}_{12}^T \bar{Q}_{11}^{-1} \bar{Q}_{13})^T & \bar{Q}_{33} - \bar{Q}_{13}^T \bar{Q}_{11}^{-1} \bar{Q}_{13} \end{bmatrix},$$

以及

$$\begin{cases} \bar{Q}_{11} = \bar{Q}([1:i-1 \ i+2:n-m], [1:i-1 \ i+2:n-m]), \\ \bar{Q}_{12} = \bar{Q}([1:i-1 \ i+2:n-m], i:i+1), \\ \bar{Q}_{13} = \bar{Q}([1:i-1 \ i+2:n-m], n-m+1), \\ \bar{Q}_{22} = \bar{Q}(i:i+1, i:i+1), \\ \bar{Q}_{23} = \bar{Q}(i:i+1, n-m+1), \\ \bar{Q}_{33} = \bar{Q}(n-m+1, n-m+1). \end{cases}$$

笛卡尔坐标系中 $x_i x_{i+1}$ 平面内的椭圆被映射成平行坐标系中的双曲线 [16]。如果平行坐标系的相邻两个坐标轴之间的距离为 1，那么映射得到的平行坐标系中的双曲线在笛卡尔坐标系中的 xy 平面内的数学模型为

$$g([x(t), y(t)]) = \begin{bmatrix} x(t) & y(t) & 1 \end{bmatrix} \begin{bmatrix} d_1 & d_4 & d_5 \\ d_4 & d_2 & d_6 \\ d_5 & d_6 & d_3 \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \\ 1 \end{bmatrix} = 0, \quad (4.50)$$

$$\begin{cases} d_1 = a_3(a_1 + a_2 + 2a_4) - (a_5 + a_6)^2, \\ d_2 = a_1a_2 - a_4^2, \\ d_3 = a_2a_3 - a_6^2, \\ d_4 = a_6(a_1 + a_4) - a_5(a_2 + a_4), \\ d_5 = a_6^2 + a_5a_6 - a_3(a_2 + a_4), \\ d_6 = a_2a_5 - a_4a_6, \end{cases}$$

以及

$$\begin{cases} a_1 = A(1, 1), a_2 = A(2, 2), \\ a_3 = A(3, 3), a_4 = A(1, 2), \\ a_5 = A(1, 3), a_6 = A(2, 3). \end{cases}$$

将式(4.50)中的 $x(t)$ 的值设为 0，就可以得到过程变量 x_i 的变化范围，其计算表达式为

$$d_2y(t)^2 + 2d_6y(t) + d_3 = 0,$$

对应的计算结果为

$$\begin{cases} y_i^l(t) = (-d_6 - \sqrt{d_6^2 - d_2d_3})/d_2, \\ y_i^h(t) = (-d_6 + \sqrt{d_6^2 - d_2d_3})/d_2. \end{cases} \quad (4.51)$$

类似的，将式(4.50)中的 $x(t)$ 设为 1，就可以得到过程变量 x_{i+1} 的变化范围，其计算表达式为

$$d_2y(t)^2 + 2(d_4 + d_6)y(t) + (d_1 + d_3 + 2d_5) = 0,$$

对应的计算结果为

$$\begin{cases} y_{i+1}^l(t) = [-(d_4 + d_6) - \sqrt{(d_4 + d_6)^2 - d_2(d_1 + d_3 + 2d_5)}]/d_2, \\ y_{i+1}^h(t) = [-(d_4 + d_6) + \sqrt{(d_4 + d_6)^2 - d_2(d_1 + d_3 + 2d_5)}]/d_2. \end{cases}$$

需要调整的可操作变量可以通过算法7来确定。

Algorithm 7 确定需要调整的可操纵变量

- 如图4.44(b)和(c)所示, 如果 $x_i(t)$ 的值和它的变化范围满足

$$x_i(t) \notin [y_i^l(t), y_i^h(t)],$$

或者 $x_{i+1}(t)$ 的值和它的变化范围满足

$$x_{i+1}(t) \notin [y_{i+1}^l(t), y_{i+1}^h(t)],$$

那么可操纵变量 x_i 或者可操纵变量 x_{i+1} 需要调整。

- 如果 $x_i(t)$ 的值和它的变化范围满足

$$x_i(t) \in [y_i^l(t), y_i^h(t)],$$

并且 $x_{i+1}(t)$ 的值和它的变化范围满足

$$x_{i+1}(t) \in [y_{i+1}^l(t), y_{i+1}^h(t)].$$

可以分为两种情况, 一种情况如图4.44(d)所示, 点 $[x_i(t), x_{i+1}(t)]$ 对应的 xy 平面中的直线穿过式(4.50)中的双曲线, 因此可操纵变量 x_i 或者可操纵变量 x_{i+1} 需要调整。另一种情况如图4.44(a)所示, 点 $[x_i(t), x_{i+1}(t)]$ 对应的 xy 平面中的直线位于式(4.50)中的双曲线之间, 因此可操纵变量 x_i 和可操纵变量 x_{i+1} 都不需要调整。

假设可操纵变量 x_{i+1} 需要调整。给定 $x_i(t)$ 的值, x_{i+1} 的调整幅度计算如下。在点 $[0, x_i(t)]$ 处, 式(4.50)中的双曲线的切线的解析表达式被定义为

$$y(t) = kx(t) + x_i(t). \quad (4.52)$$

联式(4.50)和式(4.52)可以得到

$$\phi_1 x(t)^2 + 2\phi_2 x(t) + \phi_3 = 0,$$

其中,

$$\begin{cases} a_1 = \bar{P}_{22}, \\ a_2 = U(t)\bar{P}_{12} + V(t)\bar{P}_{23}^T + \bar{P}_{24}^T, \\ a_3 = U(t)\bar{P}_{11}U(t)^T + V(t)\bar{P}_{33}V(t)^T + \bar{P}_{44} + 2V(t)\bar{P}_{13}^TU(t)^T + 2U(t)\bar{P}_{14} + 2V(t)\bar{P}_{34}. \end{cases}$$

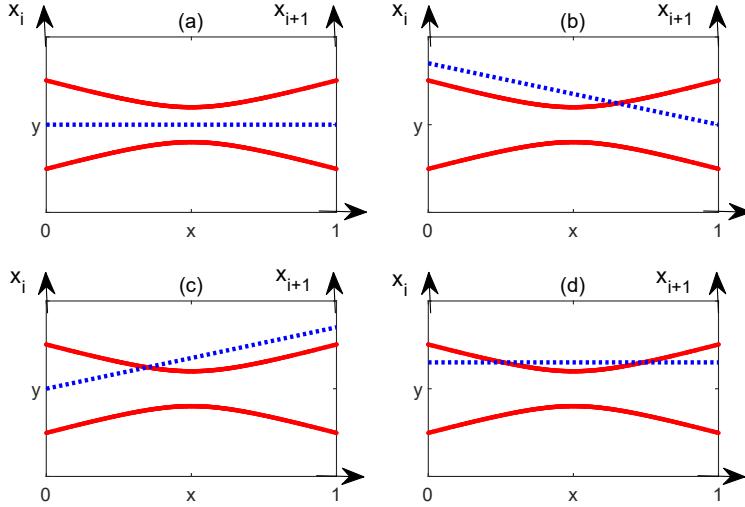


图 4.44 确定需要调整的可操纵变量

由于双曲线的切线与双曲线只有一个交点，因此

$$(2\phi_2)^2 - 4\phi_1\phi_3 = 0,$$

从而可以得到

$$\psi_1 k^2 + 2\psi_2 k + \psi_3 = 0, \quad (4.53)$$

其中，

$$\begin{cases} \psi_1 = d_6^2 - d_2 d_3, \\ \psi_2 = d_2 d_5 x_i(t) + d_5 d_6 - d_4 d_6 x_i(t) - d_3 d_4, \\ \psi_3 = d_4^2 x_i(t)^2 + 2d_4 d_5 x_i(t) + d_5^2 - d_1 d_2 x_i(t)^2 - 2d_1 d_6 x_i(t) - d_1 d_3. \end{cases}$$

式(4.53)的解为

$$\begin{cases} k_1 = (-\psi_2 - \sqrt{\psi_2^2 - \psi_1 \psi_3}) / \psi_1, \\ k_2 = (-\psi_2 + \sqrt{\psi_2^2 - \psi_1 \psi_3}) / \psi_1, \end{cases} \quad (4.54)$$

令 $x(t) = 1$ ，结合式(4.52)和式(4.54)可以得到 $x_{i+1}(t)$ 的调整幅度 $[r_{i+1}^l(t), r_{i+1}^h(t)]$ 。其中，

$$\begin{cases} r_{i+1}^l(t) = k_1 + x_i(t), \\ r_{i+1}^h(t) = k_2 + x_i(t). \end{cases} \quad (4.55)$$

调整幅度的计算原理如图4.45所示。

对于一个异常的样本点 $X(t) = [x_1(t), x_2(t), \dots, x_n(t)]$ ，调整可操纵变量来消除报警的方法总结如下。

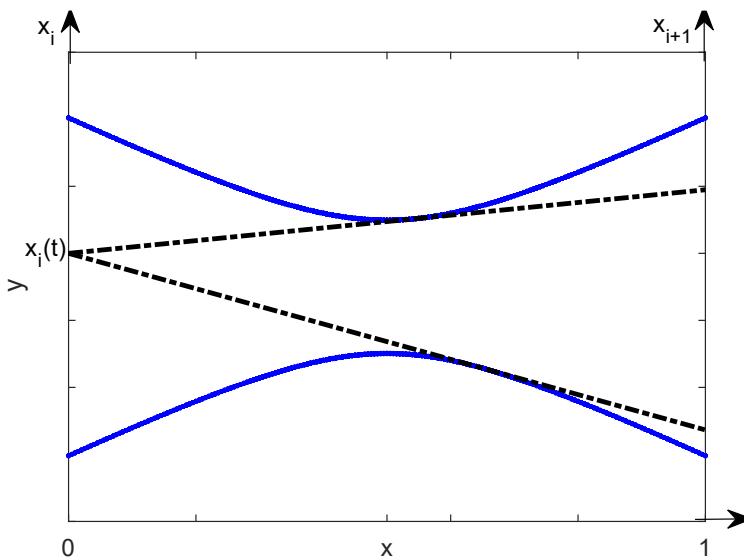


图 4.45 点 $[0, x_i(t)]$ 处的双曲线的切线在平行坐标系和笛卡尔坐标系中的表示

- 通过算法7确定需要调整的可操纵变量，并将得到的结果定义为

$$\{x_{m_1}, x_{m_2}, \dots, x_{m_s}\}.$$

- 通过式(4.55)为 $\{x_{m_1}, x_{m_2}, \dots, x_{m_s}\}$ 中的每一个元素计算它对应的调整幅度区间 $[r_{m_i}^l(t), r_{m_i}^h(t)]$ 。一般情况下，将需要调整的可操纵变量调整到它的调整幅度区间的中间位置，也就是令

$$x'_{m_i}(t) = \frac{r_{m_i}^l(t) + r_{m_i}^h(t)}{2}. \quad (4.56)$$

如果 x_{m+1} 是第一个可操纵变量，它的调整幅度区间定义为 $[y_{m+1}^l(t), y_{m+1}^h(t)]$ ，因此它对应的调整幅度为

$$x'_{m+1}(t) = \frac{y_{m+1}^l(t) + y_{m+1}^h(t)}{2}.$$

最后，讨论本节提出方法的计算复杂度。本节提出的方法分为离线阶段和在线阶段。离线阶段需要计算各个数据点到样本中心的距离，并且这些距离要按照从小到大的顺序重新排序。因此，离线阶段的计算复杂度为 $O(N \log(N))$ 。在线阶段需要计算各个过程变量的动态报警阈值，以及可操纵变量的调整幅。因为他们都可以通过推导得到的数学表达式计算得到，因此其计算复杂度为 $O(1)$ 。综上所述，本节提出算法的计算复杂度为 $O(N \log(N))$ ，这在实际应用过程中是可以接受的。

4.3.4 案例分析

本节通过数值案例和工业案例来阐述提出方法的具体步骤，并验证该方法在实际使用过程中的有效性。

案例1：考虑一个由3个过程变量组成的系统，超椭球体模型已经给定，它的数学表达式为

$$[X(t) - C_0]P_0[X(t) - C_0]^T \leq 1, \quad (4.57)$$

其中，

$$C_0 = \begin{bmatrix} 0.50 & 0.50 & 0.50 \end{bmatrix},$$

并且

$$P_0 = \begin{bmatrix} 13.39 & -6.01 & 9.88 \\ -6.01 & 9.93 & -5.99 \\ 9.88 & -5.99 & 13.12 \end{bmatrix}.$$

通过式(4.57)来生成1000个正常历史数据，并且将这些数据加上均值为0标准差为0.1的高斯白噪声。

首先，对于给定的误报率 $p = 0.05$ ，利用算法6可以得到系统的正常工作区。从正常历史数据中估计得到的中心向量为

$$C = \begin{bmatrix} 0.50 & 0.51 & 0.51 \end{bmatrix}.$$

估计得到的特征矩阵为

$$P = \begin{bmatrix} 13.48 & -6.16 & 9.94 \\ -6.16 & 10.01 & -6.05 \\ 9.94 & -6.05 & 13.22 \end{bmatrix}.$$

可以发现，中心向量和特征矩阵的估计值和它们的真实值非常接近。

然后，利用式(4.42)计算各个变量的动态报警阈值，并将计算结果与通过投影方式得到的结果进行对比。图4.47(a)展示了系统的正常工作区和一个新的数据点 $X(t_0) = [0.7, 0.7, 0.4]$ 。

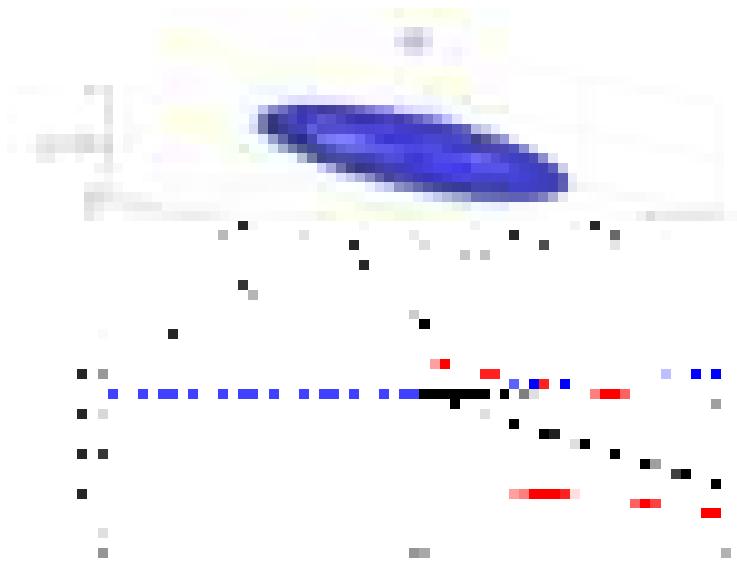


图 4.46 案例1中系统的正常工作区和它对应的降维缩减空间

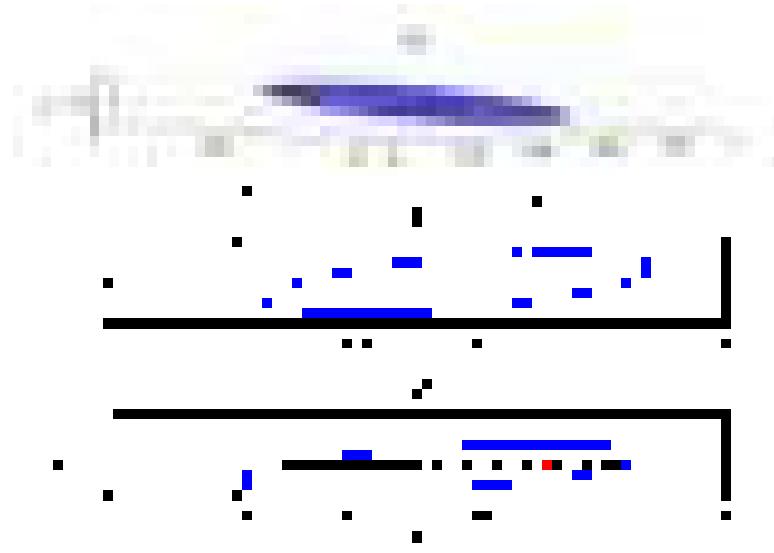


图 4.47 案例1中系统的正常工作区和它在二维平面内的投影

在图4.47(b)中，椭圆表示正常工作区在 $x_3(t) = 0.4$ 平面内的投影，红色的星号表示新的数据点在 x_1x_2 平面内的投影。当 $x_2 = 0.7$ 时， x_1 的变化范围是[0.46, 0.87]；当 $x_1 = 0.7$ 时， x_2 的变化范围是[0.28, 0.84]。在图4.47(c)中，椭圆表示正常工作区在 $x_1(t) = 0.7$ 平面内的投影，红色的星号表示新的数据点在 x_2x_3 平面内的投影。当 $x_3 = 0.4$ 时， x_2 的变化范围是[0.28, 0.84]；当 $x_2 = 0.7$ 时， x_3 的变化范围是[0.23, 0.65]。此外，通过式(4.42)计算得到的变量 x_1 ， x_2 和 x_3 的动态报警阈值分别为[0.46, 0.87]，[0.28, 0.84]和[0.23, 0.65]。因此，通过投影方法得到的动态报警阈值与通过本节提出的方法得到的动态报警阈值是一样的。将 $X(t_0)$ 中各个变量的取值与它们的动态报警阈值对比，可以发现 $X(t_0)$ 位于正常工作区之内。也就是说， $X(t_0)$ 是一个正常的样本点。

接下来，将变量 x_3 的值增大为0.8，得到一个新的数据点 $X(t_1) = [0.7, 0.7, 0.8]$ 。通过式(4.42)计算得到的变量 x_1 ， x_2 和 x_3 的动态报警阈值分别为[0.18, 0.64]，[0.64, 0.77]和[0.21, 0.71]。变量 x_1 和 x_3 的取值分别为0.7和0.8，它们的动态高报警阈值分别为0.64和0.71。这两个变量的取值都大于它们的动态高报警阈值，所以 $X(t_1)$ 是一个异常的样本点。如图4.46(a)所示，样本点 $X(t_1)$ 确实位于正常工作区的外部。

下面说明如何调整可操纵变量消除报警。假设变量 x_1 是不可操纵变量，变量 x_2 和 x_3 是可操纵变量。在图4.46(b)中，蓝色线段表示异常的样本点 $X(t_1)$ ，红色双曲线表示由变量 x_2 和 x_3 构成的降维缩减空间，黑色直线表示双曲线的切线。由于 $X_3(t_1) \notin [y_3^l(t_1), y_3^h(t_1)]$ ，所以变量 x_3 为需要调整的可操纵变量。通过式(4.55)能得到变量 x_3 的调整范围为[0.23, 0.65]，通过式(4.56)能得到变量 x_3 的调整幅度为 $x'_3(t_1) = 0.44$ 。样本点 $X'(t_1) = [0.7, 0.7, 0.44]$ 各个变量的动态报警阈值分别为[0.42, 0.85]，[0.31, 0.85]和[0.23, 0.65]。由于 $X'(t_1)$ 的各个变量的取值都位于它们的动态报警阈值之间，所以 $X'(t_1)$ 是一个正常的样本点。

最后，通过100次蒙特卡洛仿真实验来验证本章提出的方法的性能表现。在每次试验中，生成1000个样本作为正常历史数据。基于这100次蒙特卡洛仿真实验，计算得到的误报率均值

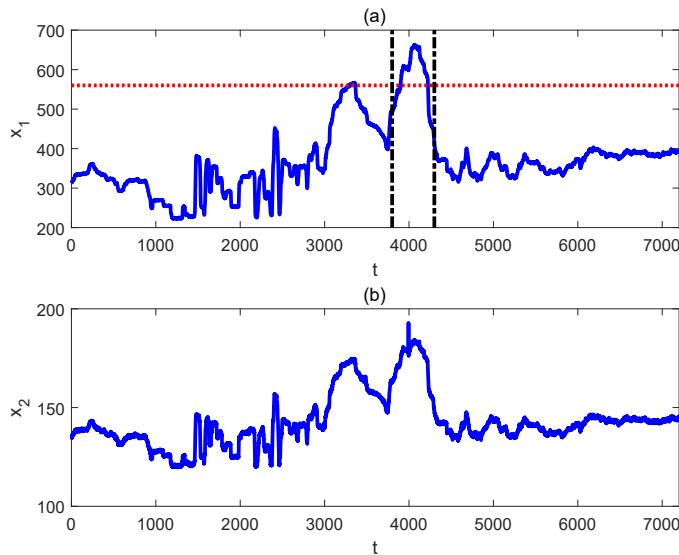


图 4.48 案例2中过程变量的时间序列图

和标准差分别为0.0329和0.0059。由于事先已经规定了报警系统的误报率不能超过0.05，因此通过本节提出的方法设计报警器是满足要求的。

案例2：本案例用于提出的方法与主成分分析法进行对比。在一个大型的发电机组中，给水泵的入口流量 x_1 配置了一个高报警阈值560t/h。图4.48展示了过程变量 x_1 和 x_2 在两个小时内的时间序列图，其中采用时间为1s。显然，当 $t \in [3301, 3363] \cup [3900, 4224]$ 时，过程变量 x_1 的取值大于它的高报警阈值，从而报警系统产生了388个报警。然而，在发电机组设备运行的历史文件中，并没有发现这段时间内机组出现了异常。过程变量 x_1 值的增大，是由于实际情况的需要造成的。因此，这388个报警都是误报警。根据现场工作人员提供的过程知识可以知道：如果前置泵电流 x_2 与给水泵入口流量 x_1 满足同增同减的关系，给水泵就处于正常工作状态。由图4.48可知，在报警发生期间，过程变量 x_1 和过程变量 x_2 之间的关系保持不变。由于设计过程变量 x_1 的报警阈值时没有考虑到过程变量 x_2 的取值对它的影响，所以产生了大量的误报警。

将本节提出的方法和主成分分析法应用到由过程变量 x_1 和过程变量 x_2 构成的系统中，计算结果分别如图4.49和4.50所示。为确保展示效果，这里只展示了 $t \in [3800, 4300]$ 的结果。这两种方法都用超椭球体模型对系统的正常工作区进行建模。为了公平比较，这两种方法的超椭球体模型的参数都被设为同一数值。因此，理论上这两种方法的性能表现相同。也就是说，这两种方法对应的误报率的结果一样。由图4.49和图4.50(b)可知，当 $t = 3994$ 和 $t = 4124$ 时，这两种方法都会产生报警。对于本案例中的系统，本节提出的方法和主成分分析法都只产生2个误报警，远远少于将过程变量 x_1 的报警阈值设为常数560时产生的误报警，后者产生388个误报警。然而，这两种方法监控样本点是否位于正常工作区之内的方式有很大区别。对于本节提出的方法，是将各个过程变量的取值与它们的动态报警阈值进行比较。在本案例中，是将 $x_1(t)$ 与 $l_1(t)$ 和 $h_1(t)$ 比较，以及将 $x_2(t)$ 与 $l_2(t)$ 和 $h_2(t)$ 比较。过程变量的动态报警阈值与过程变量的物理意义是一致的。对于主成分分析法，是将 T^2 统计量与它的控制限进行比较，其中控

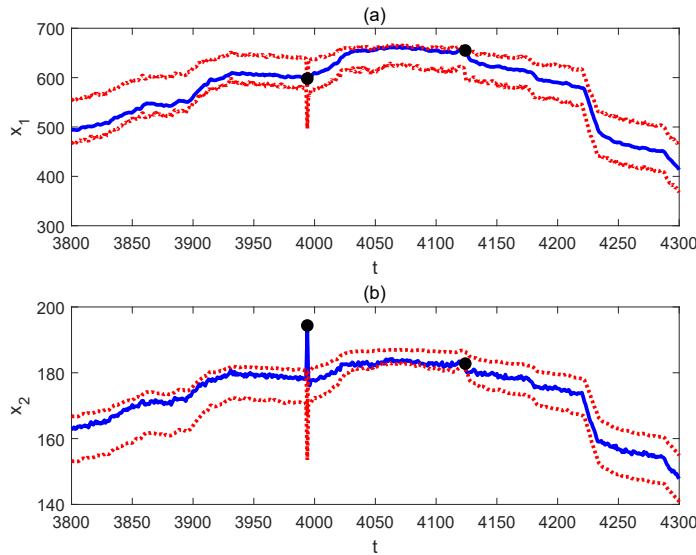


图 4.49 给水泵入口流量及前置泵电流的时间序列图和对应的动态报警阈值

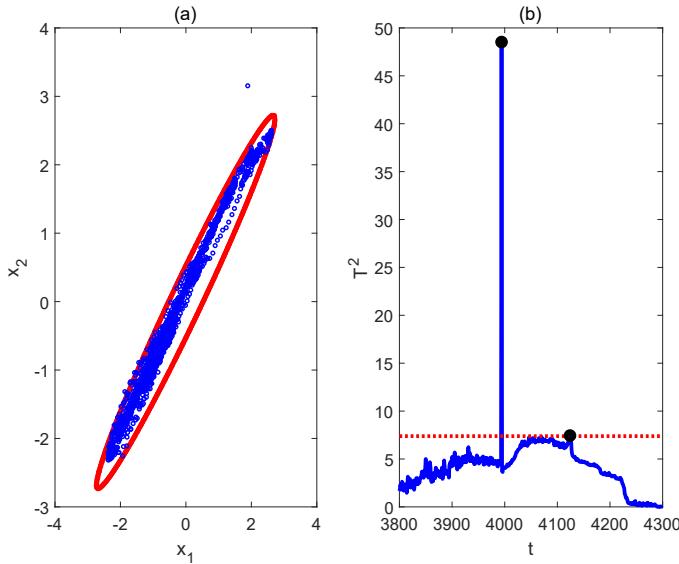


图 4.50 主成分分析法结果展示

制线的置信水平被设为0.99。在被案例中, T^2 统计量的数学表达式为

$$T^2 = 73.78x_1^2 - 146.56x_1x_2 + 73.78x_2^2.$$

显然, T^2 统计量不具有明确的物理意义。因此, 在实际应用过程中, 利用统计量来监控过程变量很难被现场工作人员理解和接受。

案例3: 本案例研究的是一个大型发电厂的发电机组。发电厂的离散控制系统实时监控着发电厂涉及到的24079个过程变量的变化。发电机组的整体运行状态可以通过6个过程变量来描

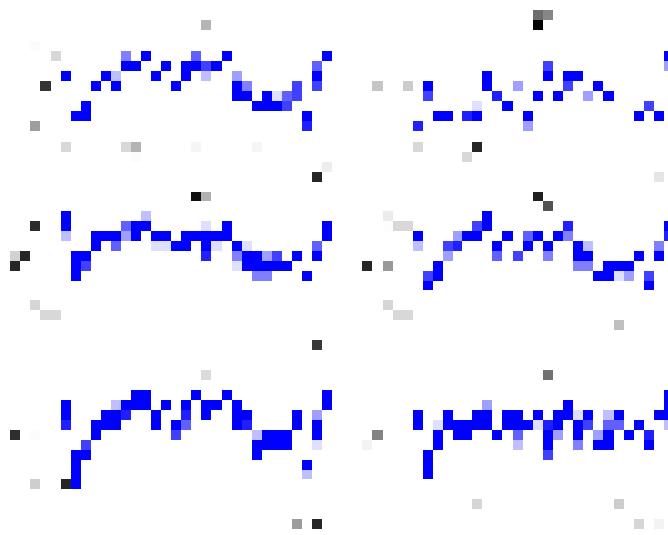


图 4.51 案例3中6个过程变量时间序列

述，分别是发电量 x_1 、主蒸汽压力 x_2 、主蒸汽流量 x_3 、给水量 x_4 、给空气量 x_5 和给煤量 x_6 。图4.51展示了这6个过程变量在5.5个小时内的时间序列图，其中采样时间为1s。

发电厂的运行原理以及这6个过程变量之间的物理联系简介如下。通过与空气混合的煤粉燃烧，发电机组将化石燃料中的化学能转换成高温高压水蒸汽中的热能；然后水蒸汽带动汽轮机的叶片旋转，将水蒸汽中的热能转换成机械能；最后汽轮机的叶片切割发电机定子的磁感线，从而将机械能转换成电能。当发电量发生变化的时候，给煤量、给水量、给空气量和主蒸汽的状态都要发生相应的变化。由于主蒸汽温度的变化很小，因此不将它加入到模型当中。基于相关的过程知识可知，发电量 x_1 和主蒸汽压力 x_2 是不可操纵变量，其它的4个过程变量是可操纵变量。可操纵变量的排列顺序确定如下：主蒸汽流量可以通过调节阀的开度来进行控制，因此它是最容易操纵的，并且它对发电量 x_1 和主蒸汽压力 x_2 有最直接的影响；为了保持给水量和主蒸汽流量之间的平衡，当主蒸汽流量 x_3 发生变化以后，给水量 x_4 也要紧接着进行变化；出于安全性的考虑，给空气量 x_5 往往要先于给煤量 x_6 发生变化。

本节提出的方法被应用到由发电量 x_1 、主蒸汽压力 x_2 、主蒸汽流量 x_3 、给水量 x_4 、给空气量 x_5 和给煤量 x_6 这6个过程变量构成的系统中。首先，将误报率设为 $p = 0.05$ ，利用算法6可以得到系统的正常工作区。本节提出方法的性能通过测试数据验证。图4.52展示了1000个测试数据的时间序列图和它们对应的动态报警阈值。所有的测试数据都是系统处于正常工作状态下采集得到的。在图4.52中，黑色的圆圈表示误报警。可以发现，一共有14个误报警，所以误报率为0.014。由于误报率的设定值为0.05，本节提出的方法的误报率小于0.05，因此本节提出的方法满足误报率的要求。

接下来，计算各个过程变量的动态报警阈值，并用它们来监控系统的运行状态。图4.53展示了300个测试数据的时间序列图，以及它们对应的动态报警阈值。在图4.53中，黑色的圆圈表示误报警。从时间 $t = 271$ 到时间 $t = 300$ ，在连续30s的时间内，过程变量 x_1 、 x_2 、 x_4 、 x_5 和 x_6 的取值都不位于它们的动态报警阈值之间，因此产生了报警信号。

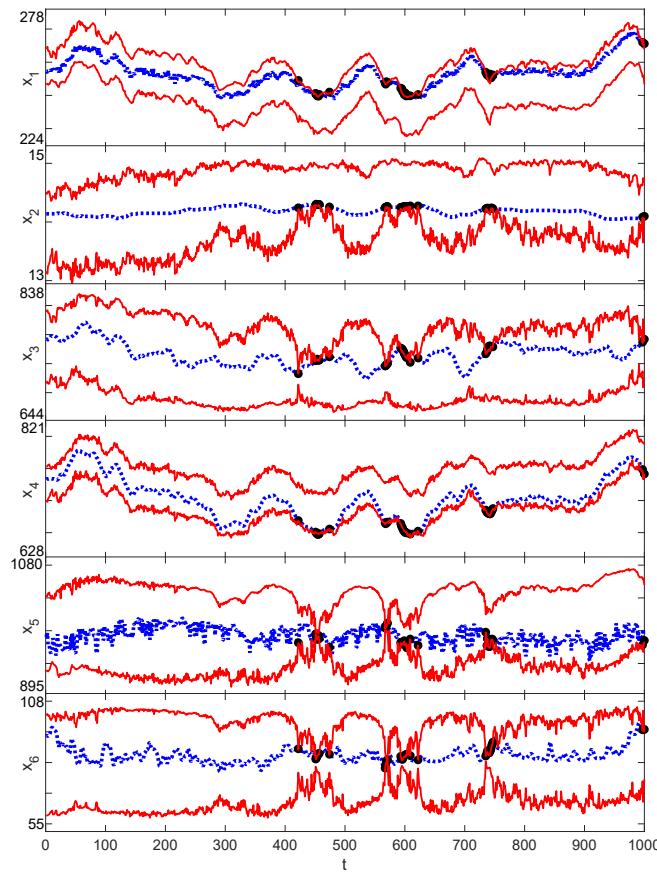


图 4.52 案例3中1000个测试数据的时间序列图及其对应的动态报警阈值

在发电机组设备运行的历史文件中找到了这段时间内发电机组出现的异常：由于给空气量 x_5 和给煤量 x_6 的值大幅度减小，而给水量 x_4 的值的变化没有跟上，导致这3个过程变量的取值出现了不平衡，从而化石燃料的燃烧过程没有释放足够的热量，最终使得主蒸汽压力 x_2 以及发电量 x_1 的值低于它们的正常值。本节提出的方法能够准确检测到该异常状况。

最后，确定需要调整的可操纵变量以及它们的调整幅度以消除报警。考虑一个导致报警产生的样本点 $X(t_0) = [256.47, 13.48, 770.44, 773.10, 959.89, 65.26]$ 。在图4.54中，红色双曲线表示将过程变量 x_1 的值固定为256.47，以及将过程变量 x_2 的值固定为13.48，得到正常工作区对应的降维缩减空间。通过算法7可知，过程变量 x_4 、 x_5 和 x_6 为需要调整的可操纵变量。将式(4.55)依次应用到过程变量 x_4 、 x_5 和 x_6 中，得到其对应的调整幅度范围分别为[734.29, 762.09]、[973.79, 1087.50]和[68.05, 106.50]，以及对应的调整幅度分别为 $x'_4(t_0) = 748.19$ 、 $x'_5(t_0) = 1030.65$ 和 $x'_6(t_0) = 87.28$ 。将过程变量 x_4 、 x_5 和 x_6 的值分别调整到748.19、1030.65和87.28，得到新样本点 $X'(t_0)=[256.47, 13.48, 770.44, 748.19, 1030.65, 87.28]$ ，并且该样本点位于正常工作区的内部。样本点 $X'(t_0)$ 的各个变量对应的动态报警阈值为[252.70, 260.11]、[12.89, 14.24]、[702.79, 828.00]、[734.72, 761.73]、[974.70, 1087.28]和[68.20, 106.30]。可以发现，样本点 $X'(t_0)$ 的各个变量的取值都位于它们的动态报警阈值之间，这也验证了样本点 $X'(t_0)$ 的

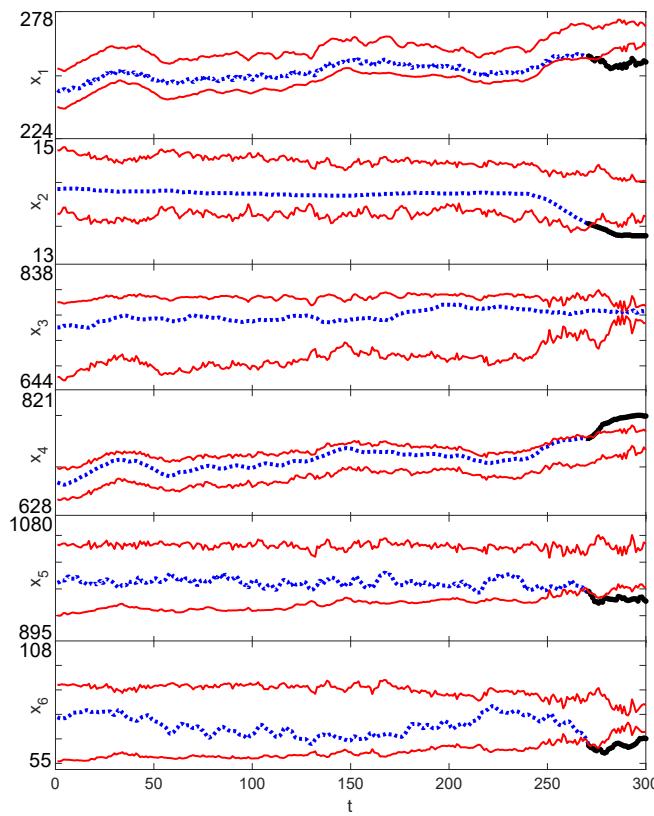


图 4.53 案例3中1000个测试数据的时间序列图及其对应的动态报警阈值

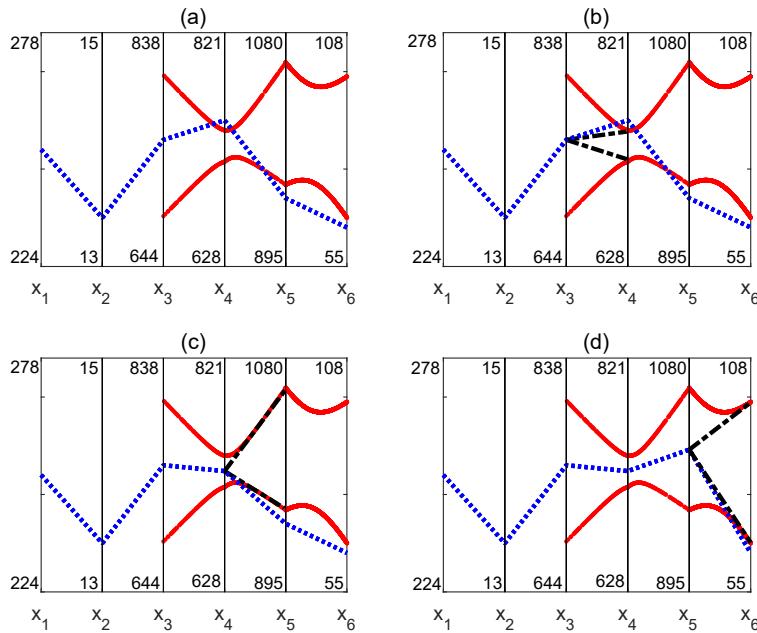
确位于正常工作区的内部。

4.4 基于凸包的多变量报警器设计

基于凸包的多变量报警器设计方法是基于定义的拟合度指标，从正常历史数据中估计凸包模型，并利用该模型来描述系统的正常工作区，然后根据系统的正常工作区，计算各个过程变量的动态报警阈值。如果所有过程变量的取值都位于它们的动态报警阈值之间，则系统正常，不触发报警；反之，系统异常，触发报警。

主成分分析法要求过程变量必须服从高斯分布。对于非高斯分布的情况，往往利用凸包来表示系统的正常工作区，并且判断一个样本点是否正常的方法与主成分分析法类似。也就是说，如果一个新的样本点位于凸包内部，就认为该样本点是正常的，否则就认为它是异常的。

本节提出了一种基于凸包的多变量报警器设计方法。系统的正常工作状态定义为系统的正常工作区，并利用凸包对系统的正常工作区建模。首先，定义一个拟合度指标，该指标衡量了凸包模型和系统的正常工作区之间的匹配程度。然后，利用拟合度挑选正常历史数据，并利用这些数据估计凸包模型的参数。最后，基于建立的凸包模型设计各个过程变量的动态报警阈值。对于一个新的样本点，如果它的所有过程变量的取值都位于各个过程变量对应的动态报警

图 4.54 样本点 $X(t_0)$ 和正常工作区对应的降维缩减空间

阈值之间，就认为这个样本点是正常的，否则就认为这个样本点是异常的。一个过程变量的动态报警阈值会受到其它过程变量取值的影响，并且过程变量的动态报警阈值与过程变量具有相同的物理意义。因此，本节提出的方法同时具有单变量报警器设计方法、多元统计分析方法以及传统的基于凸包方法的优点。

4.4.1 问题描述

考虑一个由 n 个过程变量 $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ 组成的多变量系统 \bar{X} 。给定系统位于正常状态下的历史数据，并将它们定义为

$$\{\bar{X}(t) \mid t \in [1, N]\}, \quad \bar{X}(t) := [\bar{x}_1(t), \bar{x}_2(t), \dots, \bar{x}_n(t)],$$

其中， t 表示采样时刻， N 表示样本点的个数。正常历史数据被认为构成了一个正常工作区，也就是说，如果样本点 $\bar{X}(t)$ 是正常的，它就应该位于正常工作区的内部。此外，对于新的样本点而言，它构成的正常工作区与正常历史数据构成的正常工作区是一致的。

本节的目标是基于给定的正常历史数据 $\{\bar{X}(t) \mid t \in [1, N]\}$ 为过程变量设计动态报警阈值，从而监控多变量系统的运行状态。对于一个新的样本点 $\bar{X}(t_0)$ ，它的每一个过程变量 \bar{x}_j 都有其自身对应的动态低报警阈值 $\bar{l}_j(t_0)$ 和动态高报警阈值 $\bar{h}_j(t_0)$ 。“动态”具有两层含义，一层含义是说 $\bar{l}_j(t_0)$ 和 $\bar{h}_j(t_0)$ 的值会随着 t_0 的变化而变化，另一层含义是说 $\bar{l}_j(t_0)$ 和 $\bar{h}_j(t_0)$ 的值会受到其它过程变量取值的影响。将 $\bar{x}_j(t_0)$ 的值与 $\bar{l}_j(t_0)$ 和 $\bar{h}_j(t_0)$ 的值进行比较，可以判断 $\bar{X}(t_0)$ 是否正常，同时可获得 $\bar{X}(t_0)$ 偏离正常工作状况的程度。

4.4.2 正常工作区的建立

首先，从正常历史数据 $\{\bar{X}(t)\}_{t=1}^N$ 中估计凸包模型，并用该模型描述系统的正常工作区。为了防止因某些过程变量的取值过大，而产生错误的结果，需要对数据进行标准化，也就是说，使得各个过程变量的均值为0，标准差为1。对于一个新的样本点 $\bar{X}(t)$ ，它对应的标准化后的样本点为

$$X(t) := \begin{bmatrix} x_1(t), & x_2(t), & \cdots, & x_n(t) \end{bmatrix},$$

其中，

$$x_j(t) = \frac{\bar{x}_j(t) - \mu_{\bar{x}_j}}{\sigma_{\bar{x}_j}}, \quad j = 1, 2, \dots, n. \quad (4.58)$$

在式(4.58)中， $\mu_{\bar{x}_j}$ 和 $\sigma_{\bar{x}_j}$ 分别表示 \bar{x}_j 的样本均值和标准差。对于一个含有 m 个面的凸包，它对应着 m 个超平面，并且它的第 i 个面 $f^{(i)}$ 是它对应的第 i 个超平面 $p^{(i)}$ 的子集。因此，凸包的数学模型可以表述为

$$AX_{ch}^T(t) - B \leq \mathbf{0}. \quad (4.59)$$

在式(4.59)中， $X_{ch}(t)$ 表示一个位于凸包内部或者凸包的一个面上的样本点。并且，

$$\begin{cases} A = [a^{(1)}, a^{(2)}, \dots, a^{(m)}]^T, \\ B = [b^{(1)}, b^{(2)}, \dots, b^{(m)}]^T, \\ \mathbf{0} = [0, 0, \dots, 0]. \end{cases}$$

其中， $a^{(i)} = [a_1^{(i)}, a_2^{(i)}, \dots, a_n^{(i)}]$ 表示 $p^{(i)}$ 的单位正交向量， $b^{(i)}$ 表示 $p^{(i)}$ 偏离原点的距离。超平面 $p^{(i)}$ 的数学表达式为

$$a^{(i)} X_p^T(t) - b^{(i)} = 0, \quad (4.60)$$

其中， $X_p(t)$ 表示 $p^{(i)}$ 上的一个数据点。式(4.60)中的 $a^{(i)}$ 和 $b^{(i)}$ 的值可以从 $\{\bar{X}(t)\}_{t=1}^N$ 中估计得到，详细的计算步骤请见附录B。接下来，定义拟合度指标来衡量凸包模型对数据点的拟合程度。假设标准化后的过程变量 x_j 的取值范围为 $[r_j^l, r_j^h]$ ，它的计算公式为

$$\begin{cases} r_j^l = \min(x_j), \\ r_j^h = \max(x_j), \end{cases}$$

其中， $\min(x_j)$ 和 $\max(x_j)$ 分别表示 x_j 的最小值和最大值。将凸包划分成若干个大小相等的网格，每个网格的数学表达式为

$$\{[r_j^l + (k_j - 1)\delta_j, r_j^l + k_j\delta_j]\}_{j=1}^n,$$

其中， $1 \leq k_j \leq \lfloor(r_j^h - r_j^l)/\delta_j\rfloor$ ， δ_j 表示 x_j 的最小变化幅度，并且它的值通过先验知识来确定。网格的中心点被定义为

$$C(k_1, k_2, \dots, k_n) = [c_1(k_1), c_2(k_2), \dots, c_n(k_n)],$$

其中， $c_j(k_j) = r_j^l + (k_j - 0.5)\delta_j$ 。如果一个中心点位于凸包内部，也就是满足不等式(4.59)，那么就认为它是一个内部点，内部点的个数被定义为 n_i 。对于一个内部点 $(k'_1, k'_2, \dots, k'_n)$ ，如

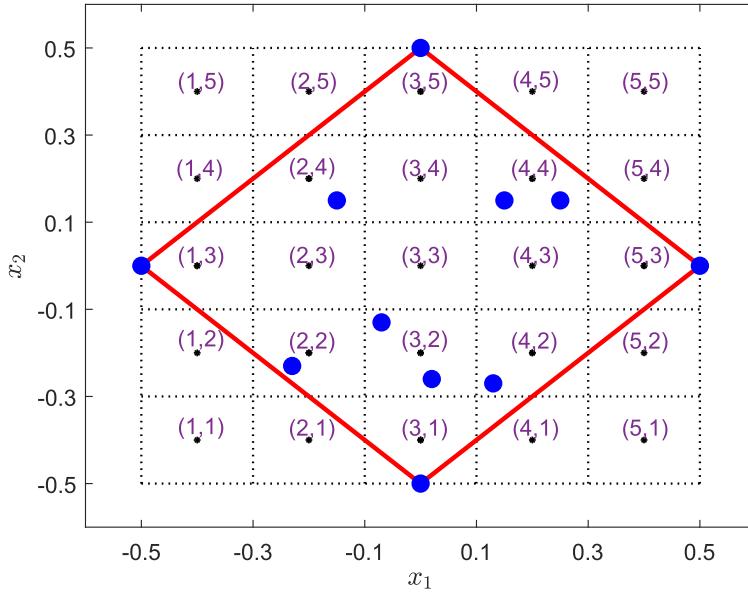


图 4.55 拟合度的示意图

果在它对应的网格内部存在样本点，也就是说

$$\exists t \in [1, N], |c_j(k'_j) - x_j(t)| \leq 0.5\delta_j, \forall j = 1, 2, \dots, n,$$

那么这个内部点被确定为计数点，计数点的个数被定义为 \$n_c\$。拟合度指标 \$\eta\$ 的数学表达式为

$$\eta = \frac{n_c}{n_i}. \quad (4.61)$$

如果 \$\eta\$ 的值较小，就意味着凸包内部有很多区域没有被正常历史数据占据，也就是存在大量的空白；如果 \$\eta\$ 的值较大，就意味着凸包模型可以很好的拟合正常历史数据。因此，\$\eta\$ 的值衡量了凸包模型与系统的正常工作区之间的匹配程度。

图4.55用一个简单的例子来阐述 \$\eta\$ 的计算过程。过程变量 \$x_1\$ 和 \$x_2\$ 的取值范围都是 \$[-0.5, 0.5]\$，并且它们的最小变化幅度都是0.2。在图4.55中，蓝色的点表示标准化后的正常历史数据点，黑色的点表示内部点，红色的实线表示凸包的边界。一共有25个中心点，其中中心点 \$(3,1), (2,2), (3,2), (4,2), (1,3), (2,3), (3,3), (4,3), (5,3), (2,4), (3,4), (4,4)\$ 和 \$(3,5)\$ 是内部点。除此之外，中心点 \$(3,1), (2,2), (3,2), (4,2), (1,3), (5,3), (2,4), (4,4)\$ 和 \$(3,5)\$ 是计数点。因此，图4.55中的凸包的拟合度 \$\eta = 9/13\$。

基于拟合度指标 \$\eta\$，从给定的正常历史数据 \$\{\bar{X}(t)\}_{t=1}^N\$ 中建立系统的正常工作区的算法实施步骤总结如下：

Algorithm 8 系统正常工作区的建立

1. 通过式(4.58)标准化 \$\{\bar{X}(t)\}_{t=1}^N\$，从而使得各个过程变量的均值为0，标准差为1。标准化后的数据被定义为 \$\{X(t)\}_{t=1}^N\$。
2. 将 \$\{X(t)\}_{t=1}^N\$ 划分成一系列的子集，例如 \$\{X(t)\}_{t=1}^{N_s}, \{X(t)\}_{t=1}^{2N_s}, \dots, \{X(t)\}_{t=1}^{dN_s}\$，其中，\$N_s\$ 表

示第一个子集中的元素的个数, $d = \lfloor N/N_s \rfloor$ 表示子集的个数。

3. 利用基于距离的聚类算法从子集 $\{X(t)\}_{t=1}^{kN_s}$ 中挑选 r 个数据来构建系统的正常工作区^[17], 其中, k 的初始值为 1, $r = \lfloor kN_s - \beta kN_s \rfloor$, $\beta \in (0, 1)$ 表示系统的误报率。
4. 基于步骤 3 中得到的 r 个数据点, 利用 quickhull 算法来寻找凸包的顶点^[18], 然后通过式(B.2) 和式(B.4) 来估计凸包模型的参数, 并且通过式(4.61) 来计算凸包的拟合度 η 。
5. 如果 η 的值趋于稳定, 并且它的稳定值大于给定的阈值 η_t , 算法终止, 并将步骤 4 中建立的凸包模型作为结果输出。否则, 将 k 的值更新为 $k + 1$, 然后重复迭代执行步骤 3 到步骤 5。

需要注意的是, 在步骤 3 中, 并不是所有的数据都被用来构建正常工作区。这是因为如果数据离样本中心的距离较远, 那么它们的分布就会很稀疏, 容易受到噪声和干扰的影响。因此, 要挑选那些靠近样本中心的数据来构建正常工作区, 而不是利用所有的数据。此外, 之所以选用文献 [17] 中的算法, 是因为该聚类算法本质上是一种 KNN 算法, 而 KNN 算法是一种在工业界广泛使用的聚类算法。

4.4.3 动态报警阈值的计算

首先, 定义各个过程变量的动态报警阈值。如果一个新的数据点 $\bar{X}(t_0)$ 满足不等式(4.59), 则它是一个正常的数据点, 沿着各个维度方向在凸表面上的投影被定义为 $\bar{X}(t_0)$ 各个过程变量的动态报警阈值。 $\bar{X}(t_0)$ 的动态报警阈值反映了 $\bar{X}(t_0)$ 与凸表面的距离。当某些过程变量的取值接近它们的动态报警阈值时, 现场工作人员应该注意系统的运行状态, 以防止系统出现异常。

对于一个异常数据点 $\bar{X}(t_0)$, 它对应的正常数据点 $\bar{X}'(t_0)$ 的动态报警阈值被定义为它的动态报警阈值, 其中, $\bar{X}'(t_0)$ 是凸表面上最靠近 $\bar{X}(t_0)$ 的一个数据点。一个异常数据点的动态报警阈值反映了它偏离系统正常工作区的程度。图 4.43(a) 展示了如何计算一个正常数据点的动态报警阈值。基于 $\bar{X}(t_0)$ 在凸表面上的投影, 可以得到 \bar{x}_1 的动态低报警阈值为 $\bar{l}_1(t_0) = -0.80$, 动态高报警阈值为 $\bar{h}_1(t_0) = 0.80$; \bar{x}_2 的动态低报警阈值为 $\bar{l}_2(t_0) = -0.70$, 动态高报警阈值为 $\bar{h}_2(t_0) = 0.70$ 。图 4.43(b) 展示了如何计算一个异常数据点的动态报警阈值。基于 $\bar{X}(t_0)$ 对应的正常数据点在凸表面上的投影, 可以得到 \bar{x}_1 的动态低报警阈值为 $\bar{l}_1(t_0) = -0.60$, 动态高报警阈值为 $\bar{h}_1(t_0) = 0.60$; \bar{x}_2 的动态低报警阈值为 $\bar{l}_2(t_0) = -0.40$, 动态高报警阈值为 $\bar{h}_2(t_0) = 0.40$ 。

接下来, 介绍如何推导动态报警阈值的解析表达式。对于一个正常数据点 $\bar{X}(t_0) = [\bar{x}_1(t_0), \bar{x}_2(t_0), \dots, \bar{x}_n(t_0)]$, 它对应标准化后的数据点为 $X(t_0) = [x_1(t_0), x_2(t_0), \dots, x_n(t_0)]$ 。将 $X(t_0)$ 分为三部分, 分别是 $x_j(t_0)$,

$$v_1 = [x_1(t_0), x_2(t_0), \dots, x_{j-1}(t_0)],$$

以及

$$v_2 = [x_{j+1}(t_0), x_{j+2}(t_0), \dots, x_n(t_0)].$$

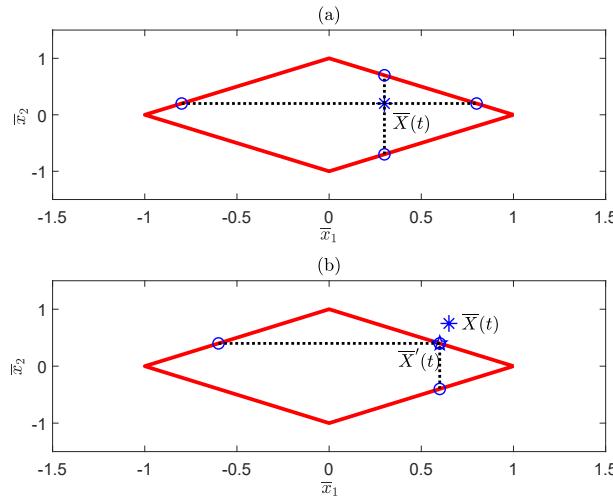


图 4.56 正常数据点和异常数据点在凸包表面上的投影

式(4.60)可以被改写成

$$\begin{bmatrix} u_1 & a_j^{(i)} & u_2 \end{bmatrix} \begin{bmatrix} v_1 & x_j(t_0) & v_2 \end{bmatrix}^T - b^{(i)} = 0, \quad (4.62)$$

其中,

$$\begin{cases} u_1 = [a_1^{(i)}, a_2^{(i)}, \dots, a_{j-1}^{(i)}], \\ u_2 = [a_{j+1}^{(i)}, a_{j+2}^{(i)}, \dots, a_n^{(i)}]. \end{cases}$$

式(4.62)是一个关于变量 $x_j(t_0)$ 的矩阵表达式,

$$a_j^{(i)}x_j(t_0) + u_1v_1 + u_2v_2 - b^{(i)} = 0, \quad (4.63)$$

它的解为

$$s_j^{(i)}(t_0) = -[u_1v_1 + u_2v_2 - b^{(i)}]/a_j^{(i)}. \quad (4.64)$$

因此, $X(t_0)$ 沿着第 j 个维度在超平面 $p^{(i)}$ 上的投影为

$$X_{p,j}^{(i)}(t_0) = [x_1(t_0), \dots, s_j^{(i)}(t_0), \dots, x_n(t_0)]. \quad (4.65)$$

如果 $X_{p,j}^{(i)}(t_0)$ 满足式(4.59), 那么它被定义为 $X(t_0)$ 沿着第 j 个维度在凸包的面 $f^{(i)}$ 上的投影。

在以下算法中, 详细介绍了如何基于凸包计算一个正常数据点 $\bar{X}(t_0)$ 的动态报警阈值。

Algorithm 9 正常数据点的动态报警阈值的计算

1. 通过式(4.65)计算 $X(t_0)$ 沿着第 j 个维度方向在超平面 $p^{(i)}$ 上的投影, 其中 $i = 1, 2, \dots, m$ 。计算结果被记录为 $X_{p,j}^{(i)}(t_0)$, 并且定义一个空集 z 。如果 $X_{p,j}^{(i)}(t_0)$ 满足式(4.59), 就将 $s_j^{(i)}(t_0)$ 加到集合 z 中。
2. 计算数据点 $X(t_0)$ 的过程变量 x_j 的动态报警阈值, 计算公式为

$$l_j(t_0) = \min(z), h_j(t_0) = \max(z), \quad (4.66)$$

其中, $\min(z)$ 和 $\max(z)$ 分别表示集合 z 中的元素最小值和最大值。利用式(4.58)和

式(4.66)可以得到数据点 $\bar{X}(t_0)$ 的过程变量 \bar{x}_j 的动态报警阈值

$$\bar{l}_j(t_0) = \sigma_{\bar{x}_j} l_j(t_0) + \mu_{\bar{x}_j}, \quad \bar{h}_j(t_0) = \sigma_{\bar{x}_j} h_j(t_0) + \mu_{\bar{x}_j}. \quad (4.67)$$

3. 将 j 分别设为 $1, 2, \dots, n$, 利用公式(4.67)就可以得到数据点 $\bar{X}(t_0)$ 的各个过程变量的动态报警阈值。

如果 $X(t_0)$ 是一个异常数据点, 那么它的动态报警阈值等于它对应的正常数据点 $X'(t_0)$ 的动态报警阈值, 其中 $X'(t_0)$ 是凸包面上离 $X(t_0)$ 最近的一个数据点。确定 $X'(t_0)$ 的值等价于满足

$$AX'^T(t_0) - B \leq \mathbf{0}, \quad (4.68)$$

求下面目标函数的最小值

$$\| X'(t_0) - X(t_0) \|^2, \quad (4.69)$$

其中, $\|\cdot\|$ 表示操作数的2范数。在式(4.69)中, 目标函数 $\| X'(t_0) - X(t_0) \|^2$ 是二次函数, 约束函数 $AX'^T(t_0) - B \leq \mathbf{0}$ 是凸函数。因此, 公式(4.69)是一个二次规划问题, 它可以在多项式时间内被求解^[19]。在MATLAB中, 有专门的函数“quadprog”可以用来有效计算式(4.69)的解。得到所有过程变量 $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ 的动态报警阈值后, 即可判断数据点 $\bar{X}(t_0)$ 是否正常。也就是说, 如果数据点 $\bar{X}(t_0)$ 的所有过程变量的取值都位于它们的动态报警阈值之间, 就认为数据点 $\bar{X}(t_0)$ 是正常的, 否则就认为它是异常的。

本节所提方法的计算流程图如图4.57所示, 该方法包括离线阶段和在线阶段。在离线阶段, 通过算法8来建立系统的正常工作区。算法8主要包含3个步骤, 分别是基于聚类算法挑选正常历史数据, 通过quickhull算法确定凸包的顶点以及利用高斯消元法估计凸包模型的参数。这3个步骤的计算复杂度分别是 $O(N \log(N))$ 、 $O(r \log(r))$ 和 $O(n^3)$, 其中, N 表示正常历史数据的个数, r 表示正常工作区包含的正常历史数据的个数, n 表示过程变量的个数。因此, 离线阶段的计算复杂度为 $O(N \log(N) + n^3)$, 这在实际应用过程中是很容易接受的。在

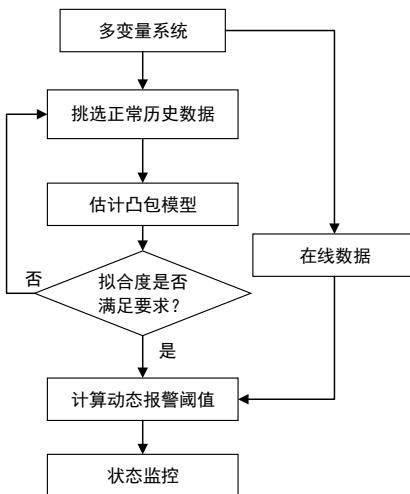


图 4.57 本章提出的方法的计算流程图

在线阶段，通过算法9来计算数据点的各个过程变量的动态报警阈值，算法9的计算复杂度为 $O(m)$ ， m 表示凸包的面的个数。因此，在线阶段的计算成本也很低廉。

4.4.4 案例分析

本节通过两个工业案例来详细阐述本节所提方法的应用步骤、实际情况下的性能表现以及传统方法和主成分分析法的对比分析。

案例1：考虑一个由给水泵入口流量 \bar{x}_1 和前置泵电流 \bar{x}_2 构成的系统。图4.59展示了该系统两个小时的正常历史数据，其中采样时间为1秒。在当前的报警系统中，过程变量 \bar{x}_1 配置了一个取值为560的高报警阈值，如图4.59(a)中黑色的点画线所示。根据先验知识可知，如果 \bar{x}_1 和 \bar{x}_2 满足同增同减的关系，系统就处于正常状态。

本节所提方法的应用过程如下：

首先，基于算法8建立系统的正常工作区。利用式(4.58)将正常历史数据 $\{\bar{X}(t)\}_{t=1}^{7200}$ 标准化，从而得到标准化后的数据 $\{X(t)\}_{t=1}^{7200}$ ，设定第一个子集中的元素个数 $N_s = 360$ 。因此 $\{X(t)\}_{t=1}^{7200}$ 被划分成20个子集，分别是 $\{X(t)\}_{t=1}^{360}$, $\{X(t)\}_{t=1}^{720}$, ..., $\{X(t)\}_{t=1}^{7200}$ 。对于每个子集，基于给定的误报警率 β ，其中 $\beta = 0.01$ ，利用文献 [17] 中的聚类算法挑选用于构建正常工作区的标准化数据；进而利用文献 [18] 中的quickhull 算法寻找凸包的顶点，并用凸包描述系统的正常工作区，凸包模型的参数可以通过式(B.2) 和式(B.4) 计算。基于相关过程知识设定 x_1 和 x_2 的最小变化幅度，分别为0.04和0.02，然后通过式(4.61) 计算每一个子集正常工作区的拟合度指标 η 的值，图4.58展示了 η 的值随子集索引 k 变化的示意图。当 $k \geq 15$ 时， η 的值趋于稳定，并且它的稳定值大于给定的阈值0.90，所以利用第15个子集 $\{X(t)\}_{t=1}^{5400}$ 来构建系统的正常工作区。图4.60(a)展示了过程变量 x_1 和 x_2 的散点图，其中正常工作区包含了子集 $\{X(t)\}_{t=1}^{5400}$ 中99% 的数据，其边界如图4.60(a)中红色的虚线所示。

然后，基于建立的正常工作区来获得各个过程变量的动态报警阈值。对于测试数据

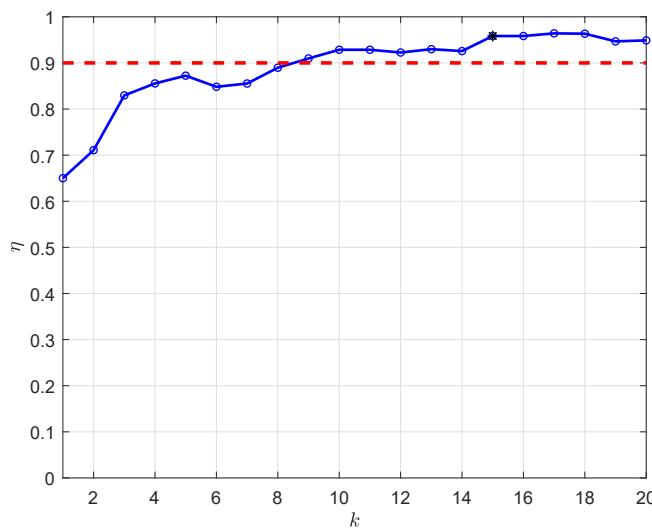


图 4.58 拟合度 η 随子集索引 k 变化示意图

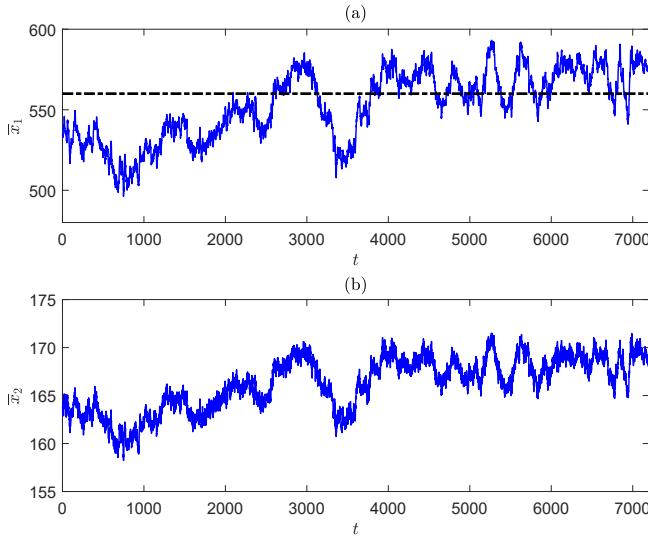


图 4.59 给水泵入口流量和前置泵电流时间序列图

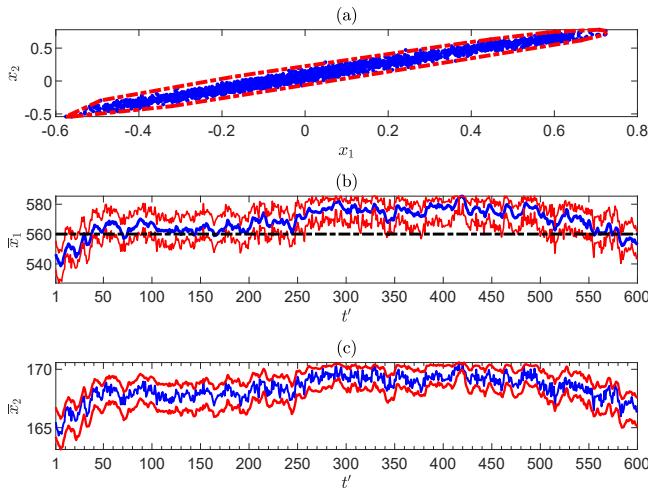


图 4.60 凸包模型及过程变量时间序列图和动态报警阈值展示

集 $\{\bar{X}(t')\}_{t'=1}^{600}$ 中的每一个样本点，通过式(4.67)计算它的各个过程变量的动态报警阈值，计算结果如图4.60(b)和(c)所示。当 $t' \in [1, 600]$ ，样本点 $\bar{X}(t')$ 的过程变量 \bar{x}_1 和 \bar{x}_2 的值都位于它们的动态报警阈值之间，因此这些样本点都是正常的样本点。

最后，基于测试数据集 $\{\bar{X}(t')\}_{t'=1}^{600}$ ，将本节提出的方法与传统方法及主成分分析法进行比较。首先介绍与传统方法的比较，如图4.60(b)所示，当 $t' \in [39, 580]$ 时， $\bar{x}_1(t')$ 的值大于静态报警阈值560，因此产生报警。但由如图4.60(a)中 $\bar{x}_1(t')$ 和 $\bar{x}_2(t')$ 的散点图可知，在报警期间 \bar{x}_1 和 \bar{x}_2 之间的关系并没有发生变化，所以产生的542个报警都是误报警。传统的方法将 \bar{x}_1 的报警阈值设为常数，由于没有考虑到相关过程变量 \bar{x}_2 的取值对 \bar{x}_1 的影响，因此产生了大量的误报警。接下来，比较本节提出的方法和主成分分析法。在图4.61中， $\{\bar{X}(t')\}_{t'=1}^{600}$ 中各个样本

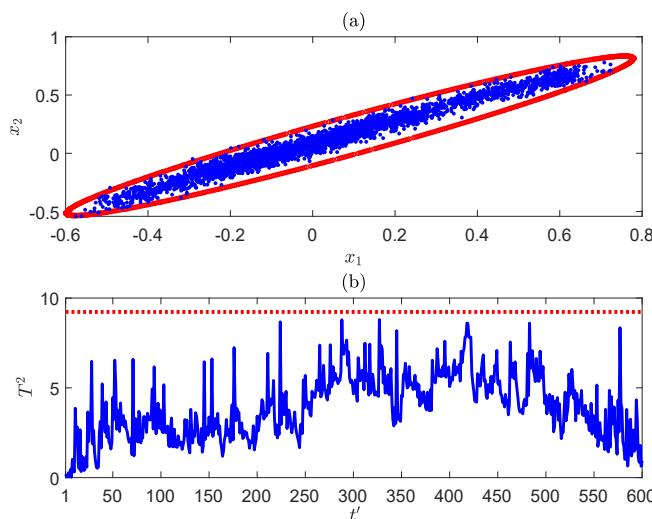


图 4.61 主成分分析法结果展示图

点的 T^2 统计量的值都小于它的控制限。因此，本节提出的方法和主成分分析法的性能表现相同，都不产生误报警。不同之处在于本节提出的方法用凸包来描述正常工作区，而主成分分析法用超椭球体来描述正常工作区，超椭球体可以视为面数无穷大的凸包，所以主成分分析法是本节提出方法的特例，即本节提出的方法具有更广泛的应用范围。更重要的是，这两种方法监控样本点是否正常的方式有很大不同，本节所提方法是将各个过程变量的取值与它们的动态报警阈值进行比较。另外，在实际应用过程中 T^2 统计量不具有明确的物理意义，因此很难被现场工作人员理解和接受。

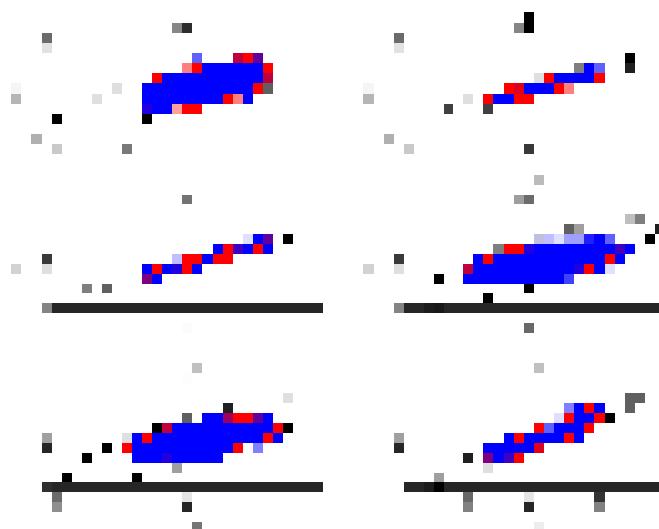


图 4.62 标准化变量的散点图

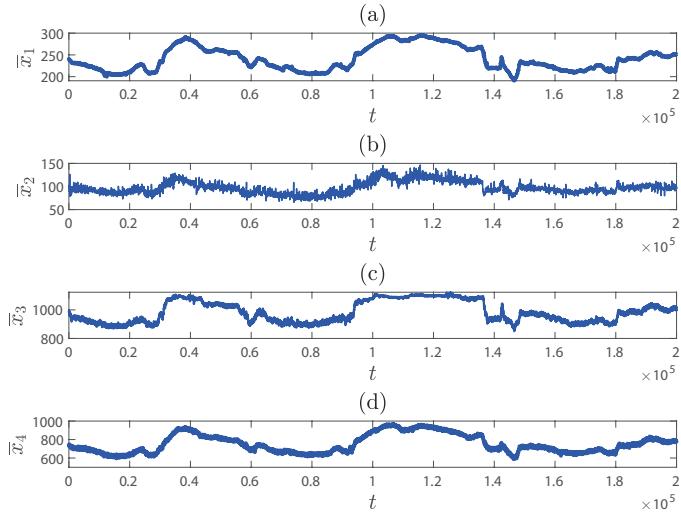


图 4.63 案例2中4个过程变量的时间序列图

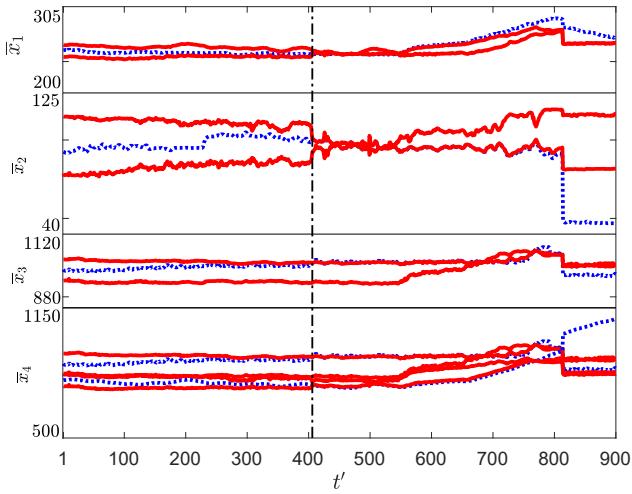
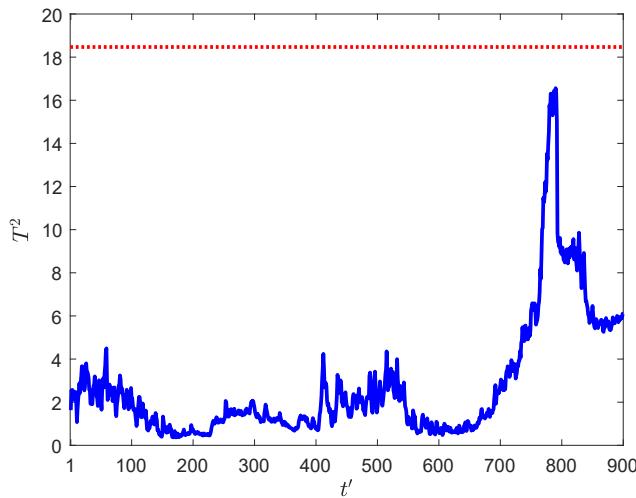


图 4.64 各个过程变量的时间序列图及其对应的动态报警阈值

案例2: 发电厂的燃烧系统主要涉及4个过程变量，分别是发电量 \bar{x}_1 、给煤量 \bar{x}_2 、给空气量 \bar{x}_3 和给水量 \bar{x}_4 。这4个过程变量在六小时内所采集正常历史数据 $\{\bar{X}(t)\}_{t=1}^{21600}$ 的时间序列图如图4.63所示，其中采样时间为1秒。

在离线阶段，基于 $\{\bar{X}(t)\}_{t=1}^{21600}$ 利用算法8来建立系统的正常工作区。由于4维的正常工作区无法在笛卡尔坐标系中可视化，因此子集 $\{X(t)\}_{t=1}^{10800}$ 被挑选来构建正常工作区，正常工作区包含 $\{X(t)\}_{t=1}^{10800}$ 中99%的数据，其中图4.62展示了两两标准化变量的散点图。在线阶段，通过式(4.67)计算测试数据集 $\{\bar{X}(t')\}_{t'=1}^{900}$ 中每个样本点的动态报警阈值，计算结果如图4.64所示。当 $t' < 406$ 时，样本点 $\bar{X}(t')$ 中4个过程变量的取值都位于它们自身的动态报警阈值之间，所以

图 4.65 T^2 统计量的时间序列图和它的控制线

这些时刻对应的样本点都是正常的样本点。然而，当 $t' \in [406, 900]$ 时，4个过程变量的取值要么大于它们的动态高报警阈值，要么小于它们的动态低报警阈值。因此，数据集 $\{\bar{X}(t')\}_{t'=406}^{900}$ 中的样本点都是异常的样本点，即系统在这段时间内出现故障。由现场工作人员的分析可知，故障的根源是给空气量和给水量之间的不平衡。由图4.64 可知，从 $t' = 200$ 到 $t' = 406$ 的时间间隔内， \bar{x}_3 的值很接近它的动态高报警阈值，并且 \bar{x}_4 很接近它的动态低报警阈，这意味着如果现场工作人员能够提前察觉该现象并采取正确的动作，该故障可能就不会发生。

在本节提出的方法与主成分分析法比较中，图4.65展示了主成分分析法的计算结果，显然 T^2 统计量的值一直位于其控制线之下，因此主成分分析法不能检测到发生的故障。由图4.62可证其主要原因是 $\{\bar{X}(t)\}_{t=1}^{10800}$ 中的样本点不满足高斯分布，从而导致超椭球体模型不能很好的描述系统的正常工作区。

4.5 基于状态变量的多变量报警器设计

本节介绍了长时间报警的定义，并提出了量化长时间报警严重程度的指标。长时间报警有几种紧密相关的定义。文献 [24] 将长时间报警视为连续24小时以上处于报警状态的报警。文献 [6] 具有与“变量处于长时间报警状态（例如24小时）”相似的定义。文献 [25] 将陈旧报警和长时间报警的定义分开：前者被认为报警但8-12小时未清除，后者被认为报警但24 小时或更长时间未被清除。文献 [12] 将长期存在的报警视为在整个工作班次或更长时间内处于报警状态的所有报警。上述定义的报警具有持续时间长的共同特征，但是报警持续时间的阈值不同。

给定一个过程变量 $x(t)$ ，其中，令 $x_a(t)$ 为 $x(t)$ 根据式(2.1)生成的报警变量，即

$$x_a(t) = \begin{cases} 1, & \text{if } x(t) \geq (\leq) x_{tp} \\ 0, & \text{if } x(t) < (>) x_{tp} \end{cases} \quad (4.70)$$

式(2.1)既可应用于模拟量也可用于数字量。如果 $x(t)$ 是模拟量，则这里的 x_{tp} 是一个真实

值；若 $x(t)$ 是数字量，则 x_{tp} 是0或1。报警持续时间的定义与3.6节一致。这里，报警持续时间用 T_{AD} 表示。此处定义1：如果报警持续时间大于阈值 T_0 ，即 $T_{AD} > T_0$ ，则出现长时间报警。在这种情况下，默认阈值为 $T_0 h = 8\text{h}$ ，这是操作员一般的常见时间间隔。因此，可利用上述定义1来检测长时间报警的存在。在实践中，大量报警变量可能会遭受长时间报警的困扰。因此，对不同报警变量根据其长期报警的严重程度进行量化是极其重要，以便优先处理严重程度较高的报警变量。对于报警序列 $X := \{x_a(t)\}_{t=1}^{T_{max}}$ ，我们提出了一个判断是否存在长期报警的指标 η

$$\eta = \frac{1}{T_{max}} \frac{1}{N} \sum_{i=1}^N T_{AD}(i). \quad (4.71)$$

这里的 $T_{AD}(i)$ 是第*i*个报警持续时间 T_{AD} ， N 是报警持续时间 $T_{AD}(i)$ 的总数量， T_{max} 是数据 X 的长度。显然，存在*i*使 T_{max} 为 $T_{AD}(i)$ 的最大值。为了使 η 有意义， T_{max} 应该不小于上述定义1中的 T_0 ，即， $T_{max} > T_0$ 。 η 的范围是[0,1]， η 越接近1，平均报警的持续时间越长，其长时间报警的严重程度就越高。值得指出的是，不应将所有 $T_{AD}(i)$ 的总和用作长期报警的索引，因为这样的索引可能会将具有许多小报警持续时间的抖动报警误分类为长期报警。

4.5.1 导致长时间报警的主要原因

本节确定长时间报警的主要原因，并提供工业示例作为支撑依据。工业实例来自某火力发电厂，以下称为A厂。所涉及变量的采样周期 $h=1\text{s}$ ，采样索引 t 也表示以秒为单位的时间变量。

长时间报警的原因多种多样。文献 [26] 和 [12] 给出了报警可能长期存在的四个原因：

- 该报警指示操作员应采取措施或应注意操作问题；
- 该报警指示其他人（通常是维护人员）应采取措施；
- 该报警表明存在无法在短期内解决的问题（例如，问题可能必须等到年度检修之后才能解决）；
- 报警是虚假的，因为它不需要采取任何操作并且不表示操作问题（即，非真实报警）。

区分干扰报警的公认标准是报警信息是否需要操作员处理^[25]，由此可得，因最后一个原因产生的长时间报警是一种干扰报警。相反，由于前三个原因而引起的长时间报警确实需要采取纠正措施，则不是干扰报警。因此，在这种情况下，由最后原因引起的长时间报警是要进行研究和消除的报警。我们通过查看更长时间报警（具体为干扰报警）的具体原因来完善最后一个原因，并按以下方式确定长时间报警的主要原因。

长期存在报警的第一个原因是不应该为报警变量配置报警。在现代计算机监控系统（例如分布式控制系统（DCS））中，通过单击鼠标并输入报警触发点值，可以很容易地从技术角度配置报警，这种快速的报警配置没有任何费用，被认为是“免费”的，并且此配置经常在没有仔细研究的情况下实施。因此，某些不应配置报警的报警变量可能会导致长期的报警。

例1. 工厂A的两个主风扇通过两个最终控制元件（即风门和变频器）进行控制。通常仅在发电装置的启动阶段使用风门。当满足某些条件时，变频器将承担控制主风扇的作用，以实

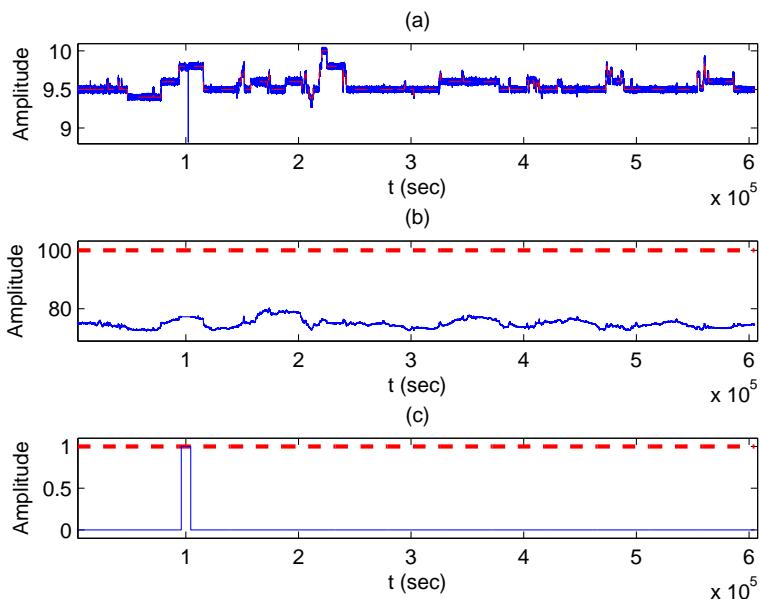


图 4.66 例1中变量时间序列图

现高能效。风门和变频器由两个控制器控制，这些控制器与同一控制目标并联运行，以获得所需的一次空气压力。两个控制器的手动模式变量已配置报警，也就是说，如果手动模式变量的值为“1”，则会发生报警，表明控制器处于手动模式。显然，同一主风扇的风门控制器和变频器控制器不能同时处于自动模式，而对于任一手动模式变量，长时间报警是不可避免的。如图4.66所示，子图(a)为控制器的主要空气压力(实线)和参考量(虚线)，子图(b)变频器控制器的输出(实线)，风门控制器输出(虚线)，子图(c)2014年3月1日至7日期间的变频器的手动模式变量(实线)和阻尼器(虚线)。在变频器的自动控制下，主风扇正常工作，同时，风门处于完全打开位置，风门控制器一直处于手动模式，从而产生长时间报警。

长期存在报警的第二个原因是报警变量的设计与离散值操作状态不一致。工业工厂的日常运营通常涉及多个状态，例如启动、半速率和全速率运行，每个状态对设备都有不同的需求。工业工厂从一个州转移到另一州后，有意停止了一些设备，这些报警变量通常仅针对设备的运行状态而设计。因此，关闭设备时会出现长时间报警。

例2. 工厂A的300MW发电机组配备了三个磨煤机，分别是磨煤机A、B和C。每个磨煤机都有两个由八个燃烧器组成的燃烧区。为了测量数字传感器，以监控燃烧器是否正常工作，配置了一个报警变量，也就是说，如果燃烧器没有着火，则报警变量处于报警状态。由于运行状态的变化，某些工厂可能必须采取相应的关闭措施。结果，当磨机关闭时，相关的燃烧器不会着火，并且燃烧器的报警变量长时间处于报警状态。但是，操作员无需采取任何措施来解决这些长期存在的报警，显然它们是干扰报警。图4.67显示了2014年3月1日至7日期间磨煤机C的整个装置的发电功率(a)，电动机的电流变量(b)和关闭状态变量以及磨煤机C的一个燃烧器的报警变量 $x_a(t)$ (c)。显然，发电机组一直处于正常运行状态，但磨床C关闭了几次。关闭磨床C时，报警变量长时间处于报警状态，如图4.67(c)，三个最大的报警持续时间分别

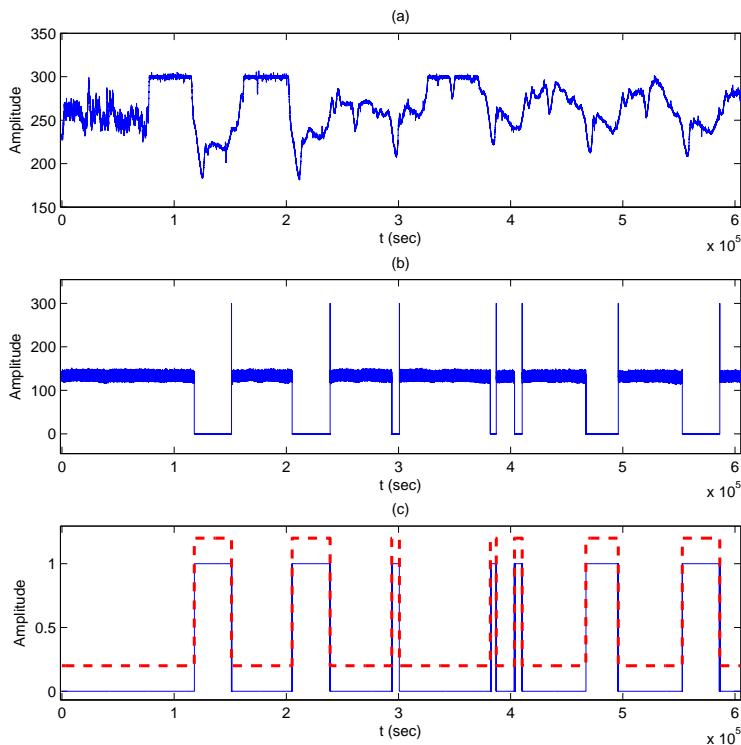


图 4.67 例2中变量时间序列图

为9.13、9.17和9.38h，根据定义1，它们是长时间报警。

长时间存在报警的第三个原因是报警跳变点与连续过程变量表示的运行状态不一致。将过程变量的测量值与高或低报警跳变点进行比较以生成报警变量。在现代报警系统中，报警触发点通常取恒定值，但是，过程变量可能会在某些连续过程变量所代表的不同操作状态下经历大规模变化。另外，恒定的报警跳变点通常仅适用于一种运行状态，并且几乎与其他运行状态不一致，因此可能会产生长时间报警。

例3. 再热器的出口蒸汽温度是与运行效率和安全性相关的关键过程变量之一。它分别配置有低报警跳变点520及高报警跳变点546，这些恒定的报警跳变点是为以300MW满负荷运行的发电设备设计的。图4.68分别展示了：(a)具有低报警跳变点520（虚线）的再热器（实线）的出口蒸汽温度；(b)报警变量；(c)在2014年3月13日至20日期间整个装置的发电功率。如图4.68所示，出口蒸汽温度通常低于低报警跳变点520。在时间间隔 $[1.11 \times 10^5, 1.54 \times 10^5]$ s中，最大报警持续时间等于12.01h，在此期间发电功率约为195MW。因此，这种长时间报警是由于低报警跳变点520和以MW为单位的发电功率所代表的运行状态之间的不一致引起的。

显然，由上述三种原因引起的长时间报警是干扰报警，应避免其出现。对于由第一个主要原因引起的长时间报警，通常选择重新配置相关报警变量。由于设计动态报警触发点的难度很大，因此，由第三种原因引起的长时间报警难以处理，留作以后的工作。下一节将设计一个基于状态的动态报警器，以处理由第二个主要原因引起的长时间报警。

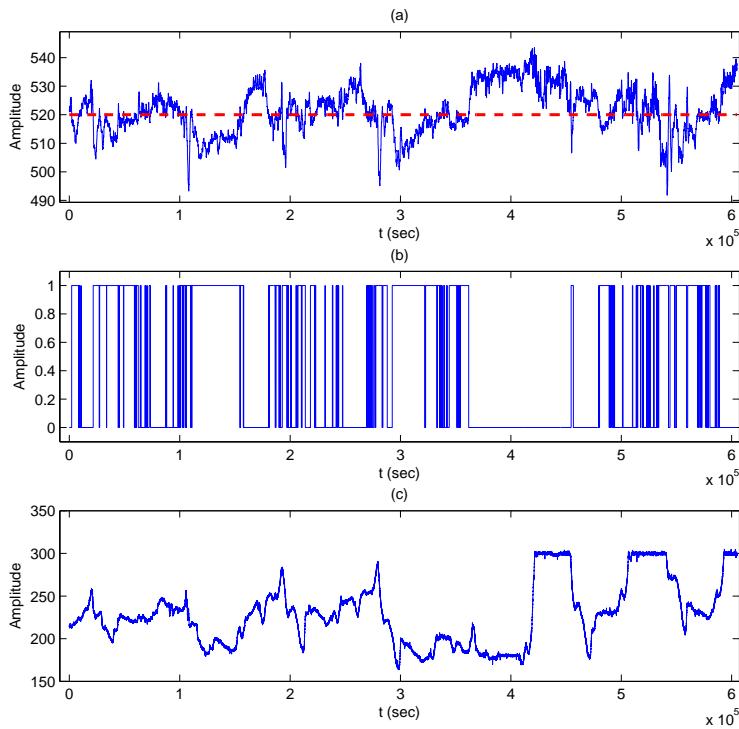


图 4.68 例3中变量时间序列图

4.5.2 基于状态的动态报警器

本节设计基于状态的动态报警器，以避免出现由报警设计和离散值操作状态之间不一致引起的长时间报警。上述讨论了基于状态的动态报警器的两个主要组件，即状态变量和报警产生机制，下面给出了该动态报警器的设计步骤和在线实现。

状态变量是指离散值的操作状态，定义为 $x_s(t)$ 。因此，它通常是一个数字变量，其二进制值为“0”和“1”。为了不失一般性， $x_s(t) = 0$ 代表不影响报警变量设计的状态。报警变量的选择分为两步：根据相关过程知识选择状态变量候选者；根据历史数据样本确定状态变量候选者之间的最终选择。对于第一步，一旦通过定义1和式(4.71)的 η 检测到报警变量具有长时间报警，根据与报警变量相关设备的物理原理，可以找到一些状态变量的候选者，有关说明请参见第4.5.5节的示例6。因此，此步骤特定于每个报警变量。对于第二步系统化的方法，首先，将状态变量设置以下限定规则：

规则1. 如果在 $t_0 - 1$ 时刻，报警变量 $x_a(t)$ 处于非报警状态，在 t_0 时刻状态变量 $x_s(t)$ 的值从0变为1，并伴随着报警的出现，即，在 t_0 时刻 $x_a(t)$ 转变为报警状态。规则1的基本原理基于以下有关报警和状态变量之间关系的观察结果：

1) 每当某些设备关闭时，如果先前的报警变量处于非报警状态，则相关的报警变量将一直在报警状态；

- 2) 当设备关闭时, 报警变量可能已经处于报警状态;
- 3) 当打开一台设备时, 报警变量不一定会切换为非报警状态;
- 4) 当设备处于运行或关闭状态时, 报警变量可以处于报警状态或非警报状态。

后三个观察结果是设备可能发生报警和清除报警的条件。图4.67(c)说明了以上的观察结果1)。图4.70的(a)和(b) (出现在4.5.3节的末尾) 导致了观察结果2), 图4.70(a)为报警变量 $x_a(t)$, (b)为状态 $x_s(t)$, (c)为式(4.81)中基于状态的报警变量 $x_{a,s}(t)$ (实线)和式(4.82)中的 $\bar{x}_{a,s}(t)$ (虚线, 向上平移0.2)。图4.69 ((a)整个装置的发电功率, (b)磨煤机C的电动机的电流变量, (c)2014年3月24日一个燃烧器的报警变量 (实线) 和磨煤机C 的电动机的状态变量 (虚线, 向上移0.2以更好地显示)) 与示例2中的图4.67对应的图, 在不同时间段内明确地支持了观察结果3) 和4)。

可以根据 $x_a(t)$ 和 $x_s(t)$ 的历史数据通过检查报警发生和 $x_s(t)$ 从0到1的转变的一致性来评估规则1。需要得到 $x_a(t)$ 和 $x_s(t)$ 的相关性来计算一致性。 $x_a(t)$ 的相对值, 指的是报警变量, 仅在 $x_a(t)$ 从非报警状态切换到报警状态的瞬间取1, 即

$$x'_a(t) = \begin{cases} 1, & \text{如果 } x_a(t-1) = 0 \text{ 且 } x_a(t) = 1 \\ 0, & \text{其他} \end{cases} \quad (4.72)$$

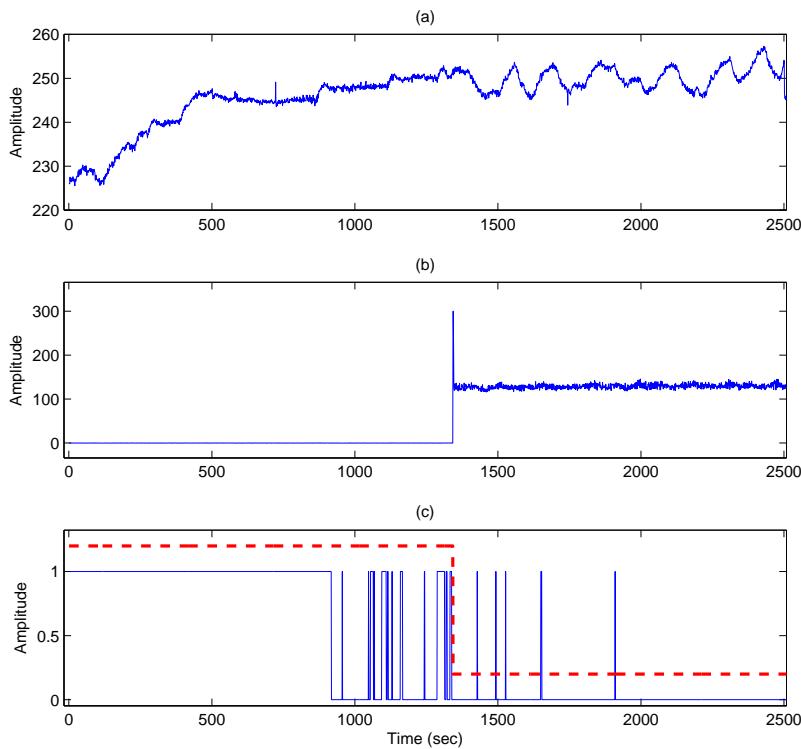


图 4.69 规则1观察结果 (3) 和 (4) 情况说明

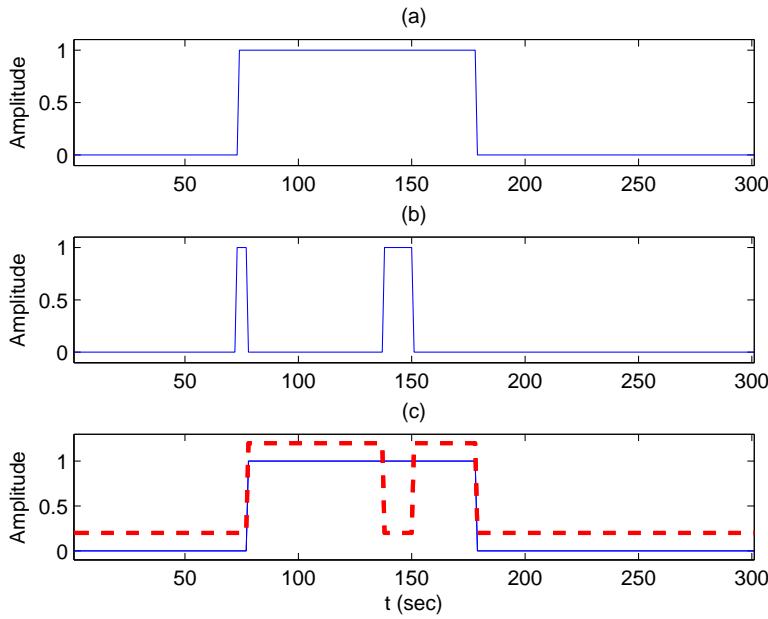


图 4.70 规则1观察结果 (2) 情况说明

$x_a(t)$ 的相对值，指的是状态变量，仅在 $x_s(t)$ 从0变为1的瞬间取1，规则1的限制条件是 $x_a(t-1)$ 是非报警状态，即

$$x'_s(t) = \begin{cases} 1, & \text{如果 } x_a(t-1) = 0, x_s(t-1) = 0 \text{ 且 } x_s(t) = 1 \\ 0, & \text{其他} \end{cases}. \quad (4.73)$$

数学上，基于两个序列 $\{x'_a(t), x'_s(t)\}_{t=1}^{T_{max}}$ ，可以评估规则1中的一致性，

$$\rho = \frac{\sum_{t=1}^{T_{max}} x'_s(t) \cdot x'_a(t)}{\sum_{t=1}^{T_{max}} x'_s(t)}.$$

一致性的比例 ρ 是在[0,1]之间的。为了满足规则1， ρ 必须取1，否则，规则1不成立， $x_s(t)$ 不是 $x_a(t)$ 的有效状态变量。

由于 $x_a(t)$ 和 $x_s(t)$ 之间的潜在动态以及随机噪声/干扰的存在， $x'_s(t)$ 中的“1”与 $x'_a(t)$ 中相应的“1”之间可能存在随机延迟 τ 。因此规则1被修订如下：

规则2. 在 t_0 时刻， $x'_s(t)$ 中的“1”都伴随着在 $(t_0 + \tau) \in [t_0, t_1]$ 时刻连续的报警 $x'_a(t) = 1$ ，这里的 t_1 指的是 $x_s(t)$ 从“1”变为“0”的时刻，即，在 $t \in [t_0, t_1]$ 和 $x_s(t_1) = 0$ 的条件下， $x_s(t) = 1$ 。

为了容忍这样的随机延迟 τ ，在 $x'_a(t)$ 的每个“1”前面都添加了额外的“1”，例如， $x'_a(t) = \cdots, 0, \underbrace{1, \cdots, 1}_L, 1, 0, \cdots$ 变为

$$x'_{a,L}(t) = \cdots, 0, \underbrace{1, \cdots, 1}_L, 1, 1, 0, \cdots. \quad (4.74)$$

这里的L是额外增加的1的数量，应该比 L_{max} 小， L_{max} 是 $x_s(t)$ 报警持续时间的最小值，即

$$L_{max} = \min_i T_{AD,x_s}(i), \quad (4.75)$$

这里的 $T_{AD,x_s}(i)$ 为 $x_s(t)$ 报警持续时间的序列。因此，一致性的比率 ρ 变为

$$\rho(L) = \frac{\sum_{t=1}^{T_{max}} x'_s(t) \cdot x'_{a,L}(t)}{\sum_{t=1}^{T_{max}} x'_s(t)}. \quad (4.76)$$

可能有多个状态变量 $x_{s,i}(t)$ ($i = 1, 2, \dots, M$) 用于满足 $\rho = 1$ 。在此情况下，软要求是 $x'_{s,i}(t)$ 中1的数量，即式(4.76)中的分母。大分母意味着可以使用基于状态的动态报警器来删除更多的长时间报警。因此，式(4.76)中具有最大分母的状态变量是最终的选择，即

$$x_s(t) = \arg \max_{x_{s,i}(t)} \left\{ \sum_{t=1}^{T_{max}} x'_{s,i}(t) \right\}_{i=1}^M. \quad (4.77)$$

4.5.3 报警生成机制

报警生成机制是基于状态变量产生报警变量 $x_{a,s}(t)$ ，基于报警变量 $x_a(t)$ 和状态变量 $x_s(t)$ ，可得如下形式：

$$x_{a,s}(t) = f(\tilde{x}_{a,s}(t)), \quad (4.78)$$

其中

$$\tilde{x}_{a,s}(t) = \begin{cases} 0, & \text{如果 } x_s(t) = 1 \text{ 且 } \tilde{x}_{a,s}(t-1) = 0 \\ x_a(t), & \text{其他} \end{cases}. \quad (4.79)$$

在实践中报警函数 $f(\cdot)$ 通常采用的是滤波器、死区和延迟器。延迟器被推荐为 $f(\cdot)$ 的选择，因为它直接作用于报警变量，而其他的选择则需要存在模拟过程变量。当且仅当 m 个连续的 $\tilde{x}_{a,s}(t)$ 的样本为1(0)， m -样本延迟器才会收集(清除)报警，即

$$x_{a,s}(t) = \begin{cases} 1, & \text{如果 } \tilde{x}_{a,s}(t-m+1:t) = 1 \text{ 且 } x_{a,s}(t-1) = 0 \\ 0, & \text{如果 } \tilde{x}_{a,s}(t-m+1:t) = 0 \text{ 且 } x_{a,s}(t-1) = 1 \\ x_{a,s}(t-1), & \text{其他} \end{cases}, \quad (4.80)$$

这里的 $\tilde{x}_{a,s}(t-m+1:t)$ 定义为 $\{\tilde{x}_{a,s}(t-m+1), \dots, \tilde{x}_{a,s}(t)\}$ 。可以根据报警器的性能指标，如误报率和漏报率，系统地设计因子 m 。文献[22]提出了一种 m 的设计方法。

在这里， $f(\cdot)$ 或 m 的设计并不是重点。为了不失一般性，令 $m = 1$ ，或者同样的， $x_{a,s}(t) = \tilde{x}_{a,s}(t)$ ，从而避免了 $f(\cdot)$ 或 m 设计的复杂性。因此，公式可以简化为

$$x_{a,s}(t) = \begin{cases} 0, & \text{如果 } x_s(t) = 1 \text{ 且 } x_{a,s}(t-1) = 0 \\ x_a(t), & \text{其他} \end{cases}. \quad (4.81)$$

式(4.81)的特点在于 $x_{a,s}(t)$ 依赖于先前的样本 $x_{a,s}(t-1)$ ，即，式(4.81)是一个动态系统。

这个特点使式(4.81)中的 $x_{a,s}(t-1)$ 与基于状态的报警器的直观设计有很大的不同

$$\bar{x}_{a,s}(t) = \begin{cases} 0, & \text{如果 } x_s(t) = 1 \\ x_a(t), & \text{如果 } x_s(t) = 0 \end{cases}. \quad (4.82)$$

式(4.82)基于状态的报警变量 $\bar{x}_{a,s}(t)$ 并不是一个合适的设计，因为它是一个静态的报警生成机制，而且并不考虑 $x_a(t)$ 之前的样本，规则1和规则2都说明了这是不可缺少的。相反，式(4.81)中 $x_{a,s}(t)$ 取决于 $x_a(t)$ 之前的样本。也就是说，只有在 $t \in [t_0, t_1]$ 时 $x_a(t)$ 不会传递给式(4.81)中的 $x_{a,s}(t)$ ，以至于 $x_a(t_0 - 1) = 0$ 且 $x_s(t_0 : t_1) = 1$ ，即

$$x_{a,s}(t) = 0, \quad t \in [t_0, t_1]. \quad (4.83)$$

这里 $x_a(t_0 - 1) = 0$ 且 $x_s(t_0 : t_1) = 1$ 。

等式(4.83)说明当且仅当导致 $x_a(t) = 1$ 的其他条件不成立时，条件 $x_s(t) = 1$ 才通过设置 $x_{a,s}(t) = 0$ 发挥作用。因此式(4.81)是合理的，而式(4.82)中的 $\bar{x}_{a,s}(t)$ 是不正确的。以下示例对此进行数字说明。

例4. 工业数据样本来自第4.5.5部分的示例6，其中报警变量与一个给煤机的速度相关联，状态变量是电动机的运行模式变量。图4.70比较了式(4.81)的 $x_{a,s}(t)$ 和式(4.82)的 $\bar{x}_{a,s}(t)$ 。状态变量 $x_s(t)$ 的值两次经历了从“0”到“1”的变化。在 $t \in [75, 80]$ 的第一次变化时，式(4.81)的 $x_{a,s}(t)$ 和式(4.82)的 $\bar{x}_{a,s}(t)$ 是一样的。而在 $t \in [135, 155]$ 的第一次变化时，报警变量 $x_a(t)$ 已经处于报警状态，可能是由于 $x_s(t) = 1$ 以外的某些条件迫使 $x_a(t)$ 处于报警状态。结果，等式(4.81)中的 $x_{a,s}(t)$ 不受 $x_s(t)$ 值变化的影响。相反，式(4.82)中的 $\bar{x}_{a,s}(t)$ 介绍了一个自动的报警事件，这完全是因为式(4.82)中基于状态的报警器的设计不当。

4.5.4 设计和实施步骤

设计基于状态的动态报警器需要采取以下步骤。这些步骤也显示在图4.71的流程图中。

第一步：收集历史数据样本来构造报警和状态变量的序列 $\{x_a(t)\}_{t=1}^{T_{max}}$ 和 $\{x_s(t)\}_{t=1}^{T_{max}}$ 以及式(6.2)中的 $\{x'_a(t)\}_{t=1}^{T_{max}}$ 和式(4.73)中的 $\{x'_s(t)\}_{t=1}^{T_{max}}$ ；

第二步：计算 $\{x_s(t)\}_{t=1}^{T_{max}}$ 的报警持续时间来得到式(4.75)中的 L_{max} ，并且增加额外的1的数量 L 附加到 $x'_a(t)$ 来得到式(4.74)中的 $x'_{a,L}$ ，直到式(4.76)中的一致性 ρ 等于1或 $L \geq L_{max}$ ；如果不满足 $\rho = 1$ 的硬性要求，那么 $x_s(t)$ 就不是一个有效的状态变量，并且这个设计过程将会终止。如果多个状态变量都满足 $\rho = 1$ ，式(4.77)将会产生最终状态变量的选择；

第三步：产生式(4.79)的中间报警变量 $\tilde{x}_{a,s}(t)$ ；

第四步：通过使用[22]中的方法设计基于 $\tilde{x}_{a,s}(t)$ 的报警延迟器因子 m 。

基于状态的动态报警器以在线方式实现，如图4.72的流程图所示。该系统由三部分组成。第一部分是一种在线机制，涉及图4.72中带有虚线边界的块，通过检查是否满足规则2来监视 $x_s(t)$ 的有效性。如果 $x'_s(t)$ 取1，然后开始搜索条件 $x'_a(t) = 1$ ；如果在 $x'_s(t)$ 再次取值“1”之前未找到这种情况，则 $x_s(t)$ 不是有效的状态变量，因此必须暂停基于状态的动态报警器。第二部分由图4.72中带有点划线边界的块组成，是等式(4.79)中基于状态的报警变量 $\tilde{x}_{a,s}(t)$ 的公式。

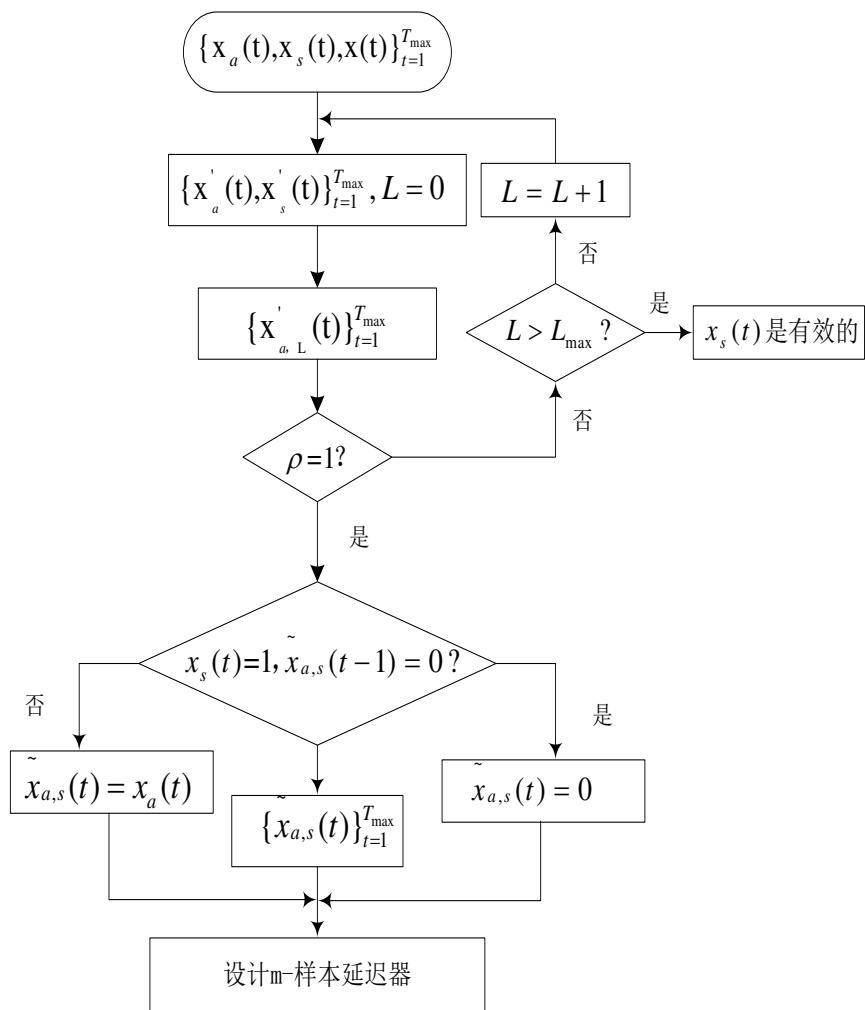


图 4.71 设计基于状态的动态报警器的步骤流程图

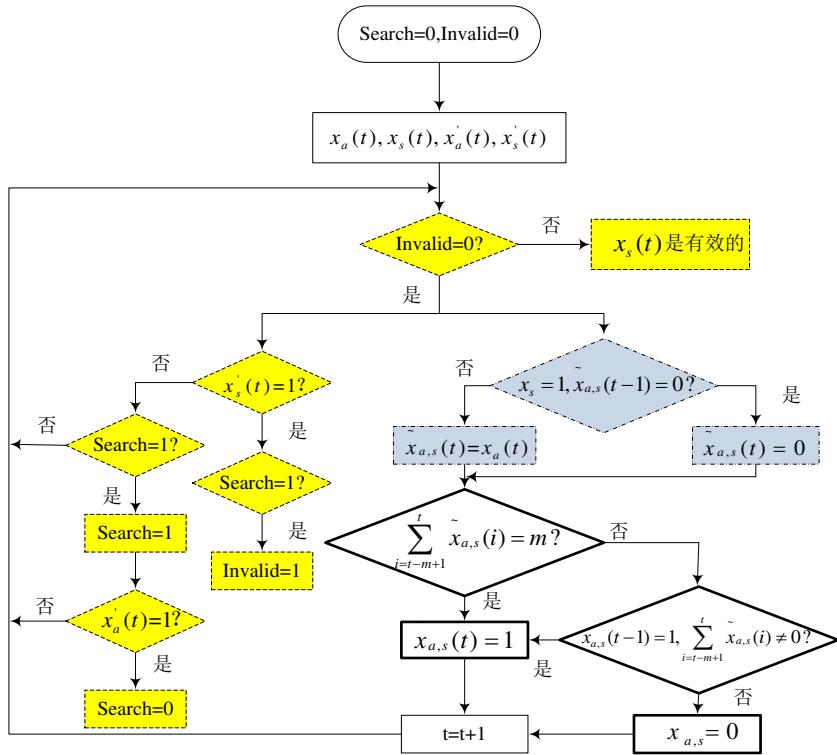


图 4.72 基于状态的动态报警器的在线实施流程图

第三部分，由带有粗实线的块组成，是式(4.80)中 m 采样延迟器的实现。这三个部分仅涉及简单的代数计算，可以在DCS 中轻松实现。

4.5.5 案例分析

本节介绍两个行业案例研究。第一个案例给出了工厂A长时间报警的总体状态，以说明实践中长期报警的严重后果，并展示了所设计动态报警器的有效性。第二个案例针对一个具有长时间报警的代表报警变量，设计和实现了基于状态的动态报警器。

例5. 本示例调查了A厂300MW发电机组工业报警系统中所有报警变量的长时间报警情况。选择定义1中报警持续时间的阈值 $T_0 = 8h$ 。对于在2014年3月1日至31日的31天内发生报警的523个报警变量，每8小时计算一次公式(4.71)中的 η 。图4.73展示了每个报警变量 η 的平均值和最大值、 η 的平均值和最大值的直方图。这里有137个报警变量 η 的平均值等于1，说明在31天内它们一直处于报警状态。还有116个报警变量 η 的平均值小于1但是最大值等于1，说明满足定义1，出现了长时间报警。总的来说，在31天内有253个报警变量存在长时间报警。图4.74展示了每8个小时出现长时间报警的报警变量的数量。也就是说，每8个小时至少有一个报警变量的报警持续时间等于 $T_0 = 8h$ 。图4.74中最大值，平均值和最小值分别为172、156、93。这些数字远远超过文献 [6] 和 [12] 中具有长时间报警的5-10个报警变量的基准。

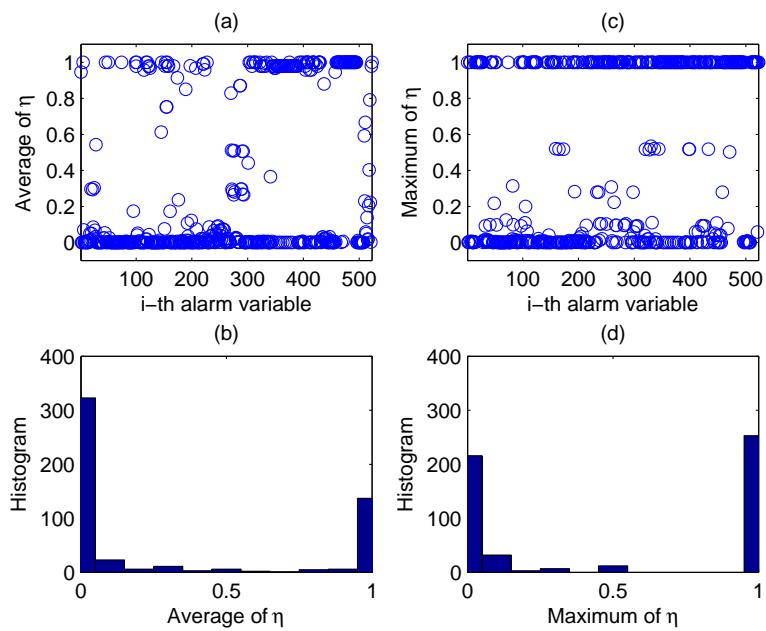
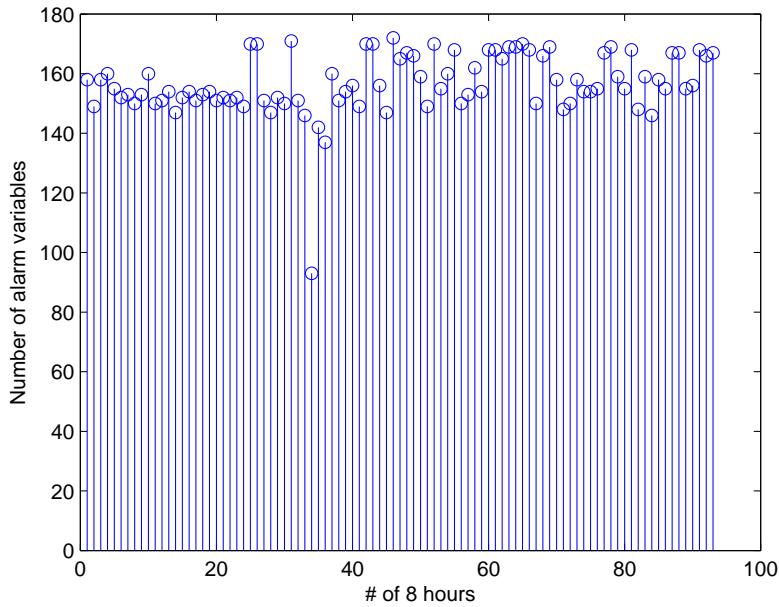
图 4.73 例5长时间报警指标 η 均值和最大值情况

图 4.74 例5每8个小时中有长时间报警的报警变量的数量

提出的基于状态的动态报警器已针对工厂A的大量报警变量在线实施。表4.2列出了2014年5月1日至7日具有长时间报警的锅炉燃烧系统中的35个报警变量及其状态变量。前32个报警变量 $x_a(t)$ 的长时间报警指数等于或接近1，而后3个报警变量具有较不严重的长时

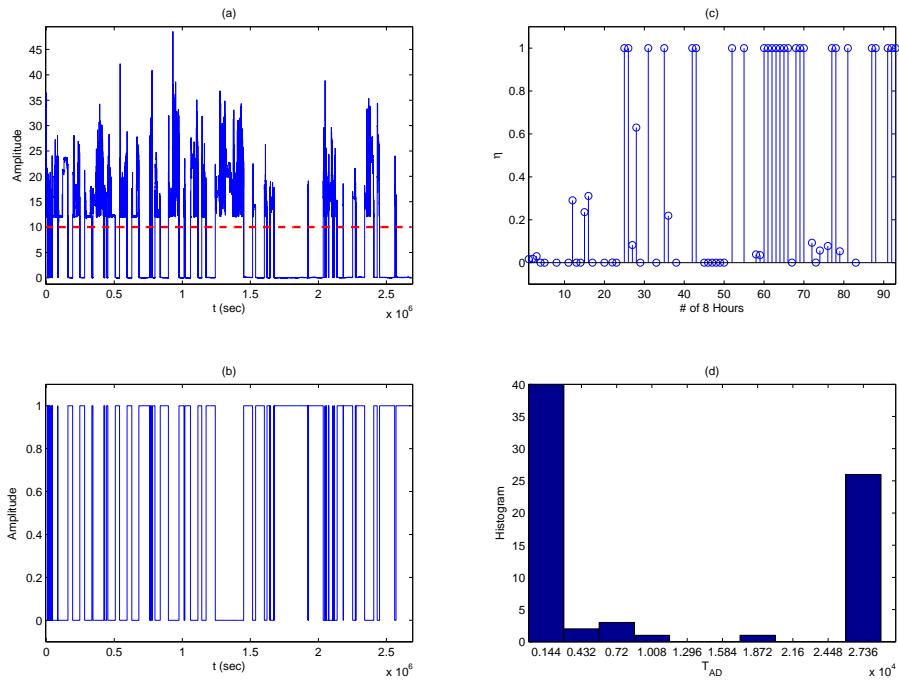


图 4.75 $x(t)$ 、 $x_a(t)$ 、 η 趋势图, 报警持续时间 T_{AD} 的直方图

间报警。公式(4.71)中的 η 对于基于状态的对应项 $x_{a,s}(t)$ 来说等于0, 这说明了设计的基于状态的动态报警器的有效性。

例6. 具有长时间报警的一个代表性报警变量 $x_a(t)$ 是与工厂A中磨煤机C的一个给煤机的速度 (表4.2中的第一个) 相关的变量。由于轧机C被设计为备用轧机, 因此经常会因操作变化而打开或关闭。C厂配备的两个给煤机也是如此。过程变量 $x(t)$ (一个给煤机的速度) 配置有报警变量 $x_a(t)$, 且具有低报警跳变点 $x_{tp} = 10$ 。图4.75展示了 $x(t)$ 和 $x_a(t)$ 的时间趋势, 每8个小时公式(4.71)中长时间报警的指标 η 的演变, 以及2014年3月1-31日31天的报警持续时间 T_{AD} 的直方图。许多报警持续时间与最大报警持续时间 $T_{max} = T_0 = 8h$ 相等, 并且指标 η 在一些时间达到1, 两者都指示 $x_a(t)$ 中存在长时间报警。

图4.76是给煤机的示意图, 具有三个主要部件, 即电动机、出口板和进口板, 它们可以直接影响给煤机速度 $x(t)$ 。基于此过程知识, 将获得三个状态变量候选值, 即电动机的关闭状态变量 $x_{s,1}(t)$, 出口板的位置状态变量 $x_{s,2}(t)$ 和入口板的位置状态变量 $x_{s,3}(t)$ 。三个候选值是二进制值, 值“1”代表关上/关闭状态。图4.77展示了 $x_a(t)$ 和 $x_{s,1}(t)$, 一些报警变量 $x'_a(t)$ 、状态变量 $x'_{s,1}(t)$ 的时间趋势, $x'_a(t)$ 和 $x'_{s,1}(t)$ 中“1”之间的随机延迟 τ 的直方图, 以及公式(4.76)中一致性比率 $\rho(L)$ 。为了获得更好的可视化效果, 图4.79(a)中的 $x_{s,1}(t)$ 向上移动0.2, 即, $x_{s,1}(t) + 0.2$ 。

图4.78和图4.79分别是图4.77中 $x_{s,2}(t)$ 和 $x_{s,3}(t)$ 的对应。图4.79(d)中的一致性比例没有达到1, 不满足规则2。因此, $x_{s,3}(t)$ 不是一个有效的状态变量。图4.80是 $t_0 = 92215$ 附近的相关变

表 4.2 锅炉燃烧系统报警变量及其长时间报警指标 η 的比较

#	报警变量	状态变量	$x_a(t)$ 的 η	$x_{a,s}(t)$ 的 η
1	磨煤机C-#1 给煤机转速	磨煤机C-#1 电机关断	1	0
2	磨煤机C-#2 给煤机转速	磨煤机C-#2 电机关断	1	0
3	磨煤机C-#1 气压	磨煤机C-#1 气阀关	1	0
4	磨煤机C-#2 气压	磨煤机C-#2 气阀关	1	0
5	磨煤机C-C1-1 密封阀	磨煤机C-C1-1 关	1	0
6	磨煤机C-C1-2 密封阀	磨煤机C-C1-2 关	1	0
7	磨煤机C-C1-3 密封阀	磨煤机C-C1-3 关	1	0
8	磨煤机C-C1-4 密封阀	磨煤机C-C1-4 关	1	0
9	磨煤机C-C2-1 密封阀	磨煤机C-C2-1 关	1	0
10	磨煤机C-C2-2 密封阀	磨煤机C-C2-2 关	1	0
11	磨煤机C-C2-3 密封阀	磨煤机C-C2-3 关	1	0
12	磨煤机C-C2-4 密封阀	磨煤机C-C2-4 关	1	0
13	油A1 有火	油A1 进气阀关	1	0
14	油A2 有火	油A2 进气阀关	1	0
15	油A3 有火	油A3 进气阀关	0.99	0
16	油A4 有火	油A4 进气阀关	0.96	0
17	油B1 有火	油B1 进气阀关	0.94	0
18	油B2 有火	油B2 进气阀关	1	0
19	油B3 有火	油B3 进气阀关	0.92	0
20	油B4 有火	油B4 进气阀关	0.94	0
21	油C1 有火	油C1 进气阀关	1	0
22	油C2 有火	油C2 进气阀关	1	0
23	油C3 有火	油C3 进气阀关	1	0
24	油C4 有火	油C4 进气阀关	1	0
25	煤C1-1 有火	煤C1-1 进气阀关	1	0
26	煤C1-2 有火	煤C1-2 进气阀关	1	0
27	煤C1-3 有火	煤C1-3 进气阀关	1	0
28	煤C1-4 有火	煤C1-4 进气阀关	1	0
29	煤C2-1 有火	煤C2-1 进气阀关	1	0
30	煤C2-2 有火	煤C2-2 进气阀关	1	0
31	煤C2-3 有火	煤C2-3 进气阀关	1	0
32	煤C2-4 有火	煤C2-4 进气阀关	1	0
33	煤A1-2 有火	煤A1-2 进气阀关	0.23	0
34	煤A1-4 有火	煤A1-4 进气阀关	0.31	0
35	煤B2-3 有火	煤B2-3 进气阀关	0.18	0

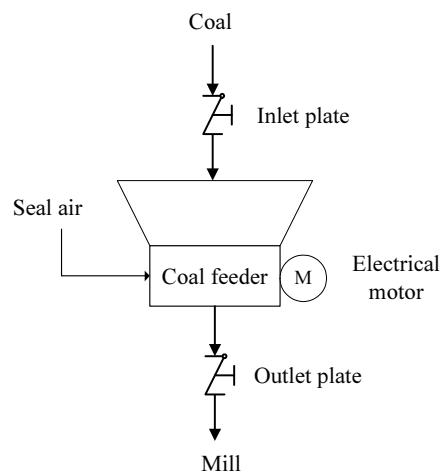
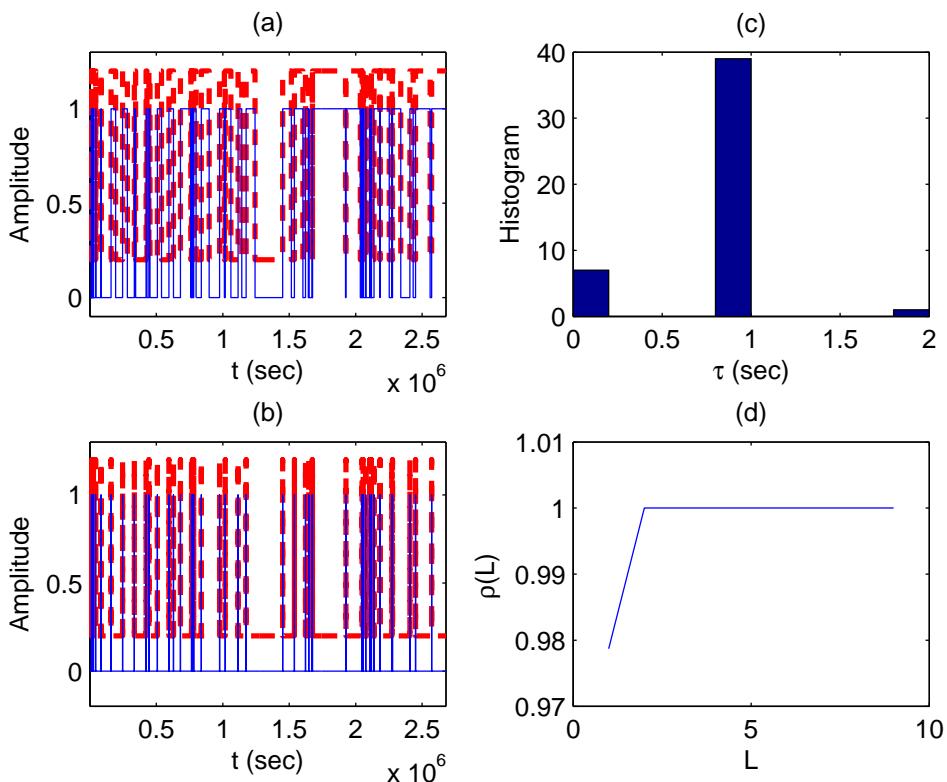
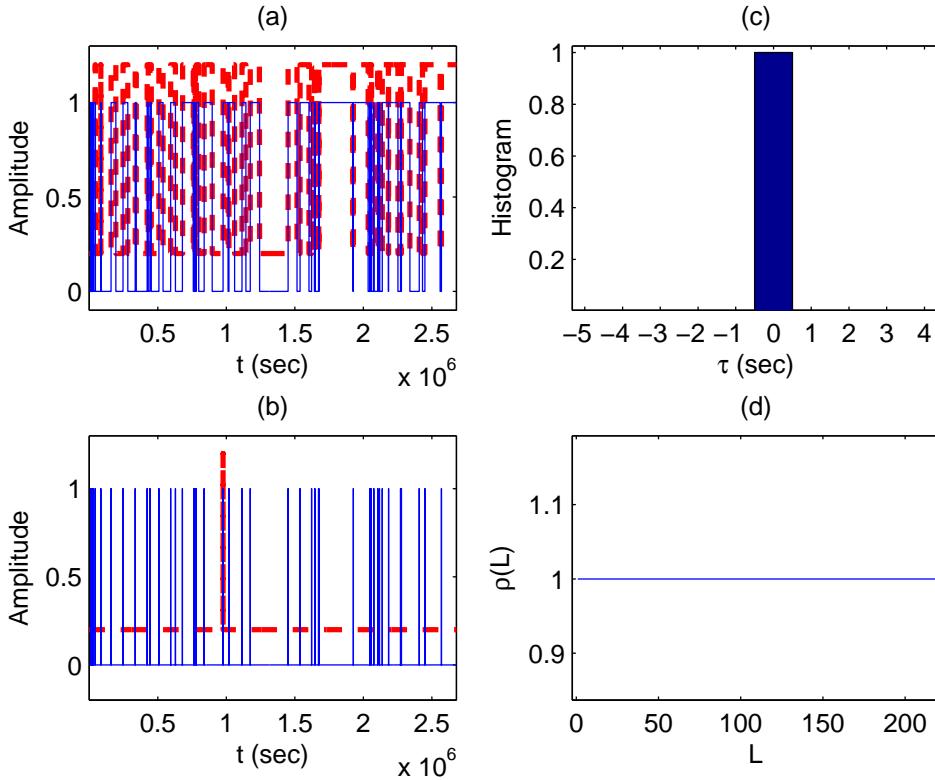


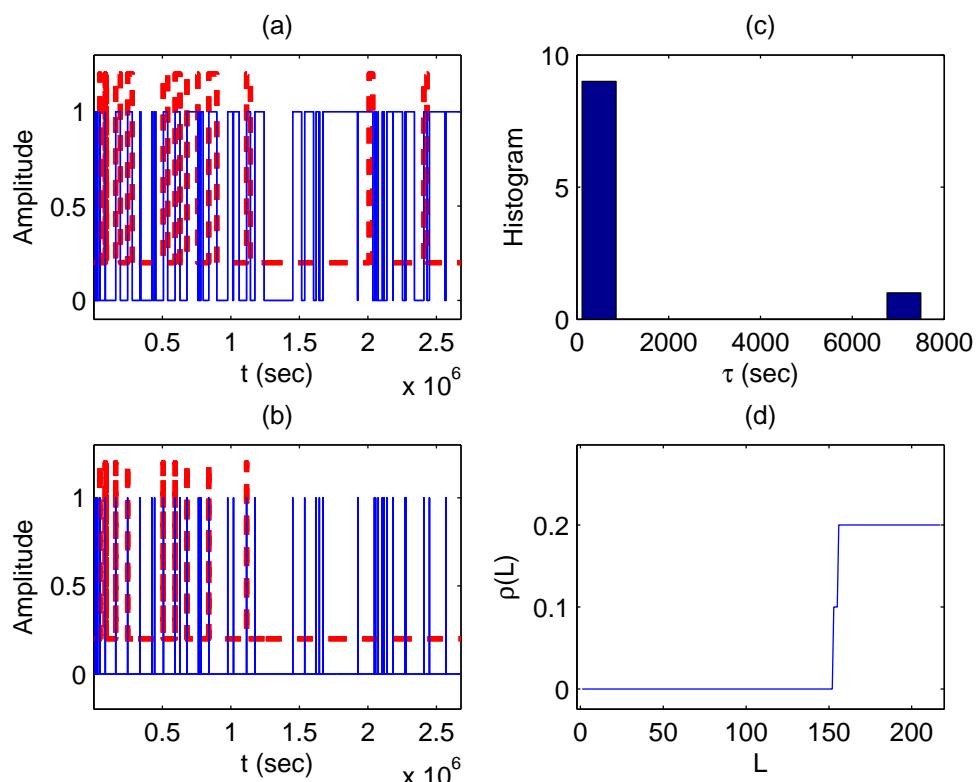
图 4.76 给煤机示意图。

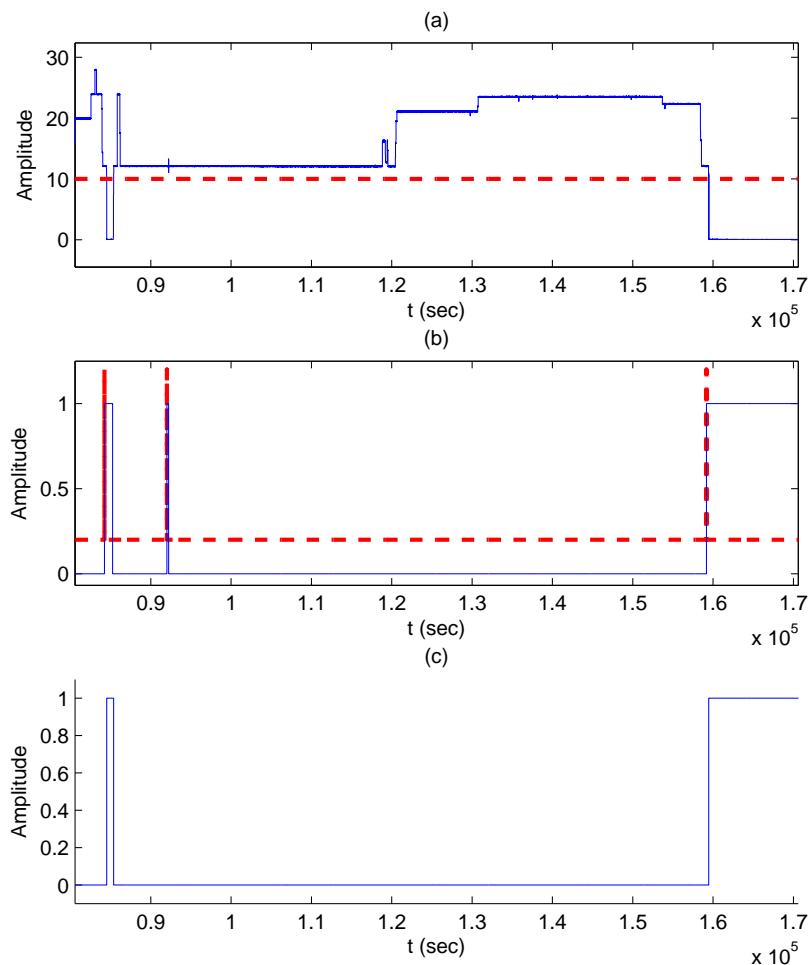
图 4.77 例6中 $x_a(t)$ 、 $x_{s,1}(t)$ 时间序列及对应的 τ 和一致性比率

图 4.78 例6中 $x_a(t)$ 、 $x_{s,2}(t)$ 时间序列及对应 τ 和一致性比率

量的放大视图，其中 $x'_{s,3}(t_0) = 1$ 违反了规则2。相反，图4.77(d)和图4.78(d)中的一致性比例达到了1，说明 $x_{s,1}(t)$ 和 $x_{s,2}(t)$ 都满足规则2中的要求。但是，如图4.78(b)所示， $x_{s,2}(t)$ 只包含一个1，然而，在图4.77(b)中 $x_{s,1}(t)$ 有47个时刻都取1。因此，式(4.77)将 $x_{s,1}(t)$ 作为最终状态变量的选择，因为基于 $x_{s,1}(t)$ 的状态动态报警器可以消除更多的长时间报警。

图4.72中基于状态的动态报警器的在线实现是在2014年4月1日至30日的接下来的31天内针对报警变量 $x_a(t)$ 执行的。为了进行比较，我们分别基于 $x_{s,1}(t)$ 和 $x_{s,2}(t)$ 获得两个基于状态的报警变量 $x_{a,s,1}(t)$ 和 $x_{a,s,2}(t)$ 。在图4.81、4.82、4.83中给出了 $x_a(t)$ ， $x_{a,s,1}(t)$ 和 $x_{a,s,2}(t)$ 的报警持续时间，分别展示了 $x_{a,s,1}(t)$ 和 $x_{a,s,2}(t)$ 以及过程变量 $x(t)$ ，报警变量 $x_a(t)$ 和状态变量 $x_{s,1}(t)$ 和 $x_{s,2}(t)$ 。显然， $x_{a,s,1}(t)$ 的报警持续时间比 $x_a(t)$ 小，而且每八个小时 $x_{a,s,1}(t)$ 的长期报警指标 η 都非常接近0。因此基于 $x_{s,1}(t)$ 的动态报警器是有效的。 $x_{a,s,1}(t)$ 中的这些报警是由于 $x'_{s,1}(t)$ 中的“1”与 $x'_{a,1}(t)$ 中的“1”之间存在随机延迟 τ 。因此，具有较小报警持续时间的报警可以视为抖动报警，并可通过m采样延迟器将其删除。相反，状态变量 $x'_{s,2}(t)$ 的值不为‘1’，则 $x_{a,s,2}(t)$ 实际上与 $x_a(t)$ 相同，表示 $x_{s,2}(t)$ 不是一个状态变量好的选择。

图 4.79 例6中 $x_a(t)$ 、 $x_{s,3}(t)$ 时间序列及对应的 τ 和一致性比率

图 4.80 例6中 $x(t)$ 、 $x_{s,3}(t)$ 及 $x_a(t)$ 部分放大图

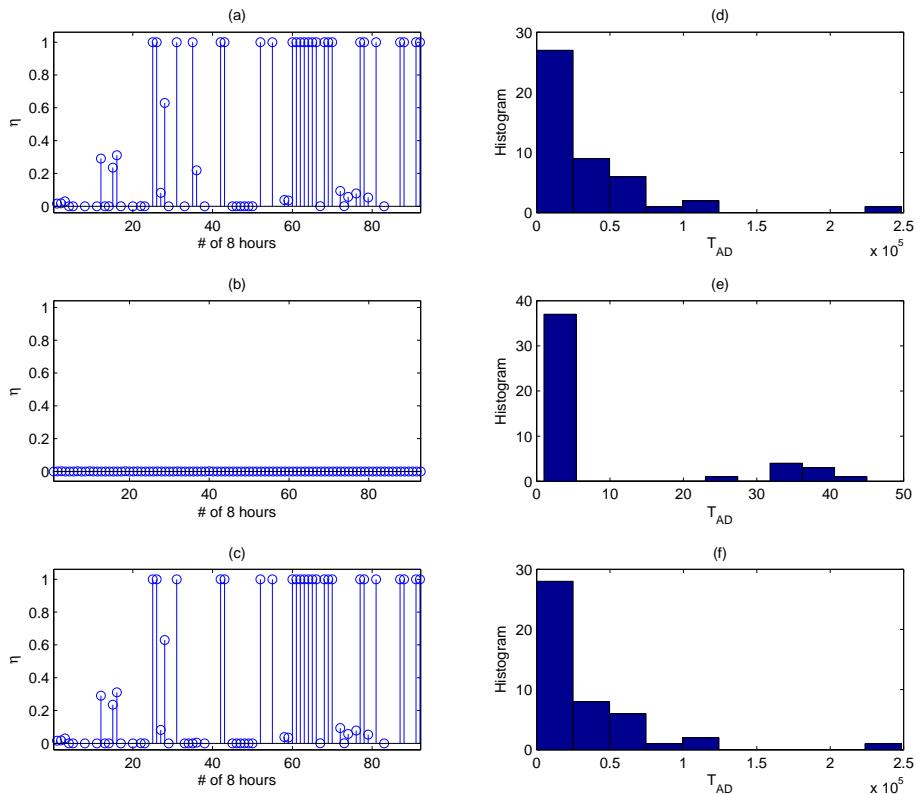


图 4.81 例6中 $x_a(t)$ 、 $x_{a,s,1}(t)$ 、 $x_{a,s,2}(t)$ 的报警持续时间的直方图及长期报警的指标

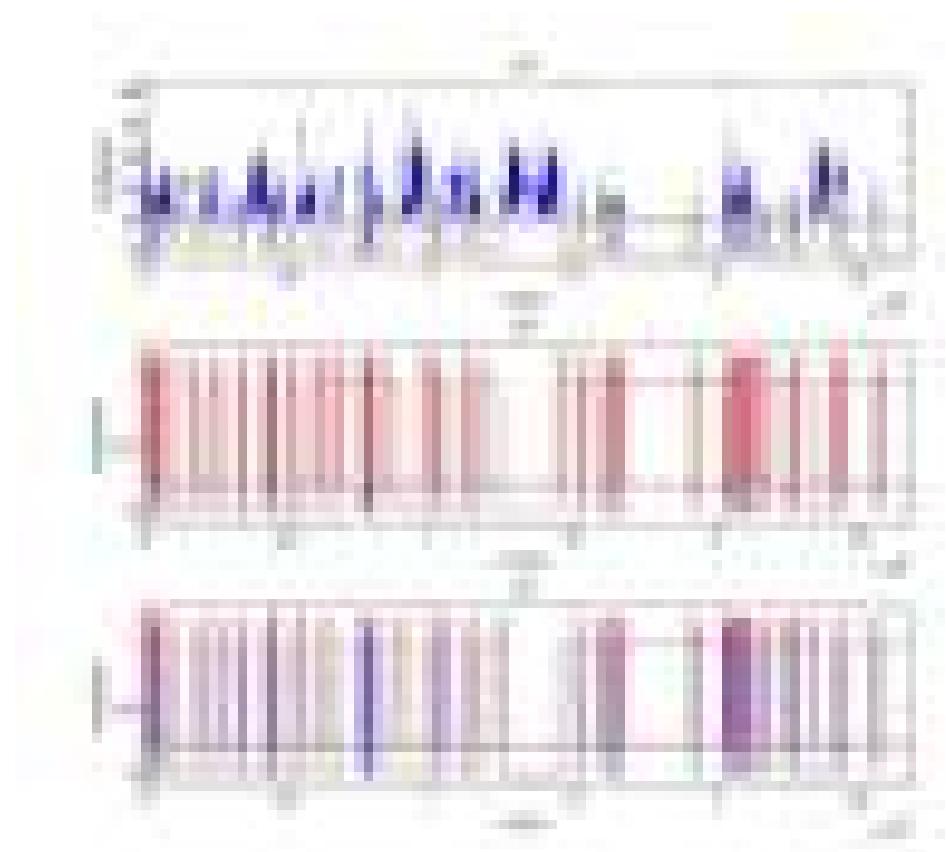


图 4.82 例6中 $x(t)$ 、 $x_a(t)$ 、 $x_{s,1}(t)$ 及 $x_{a,s,1}(t)$ 的时间序列

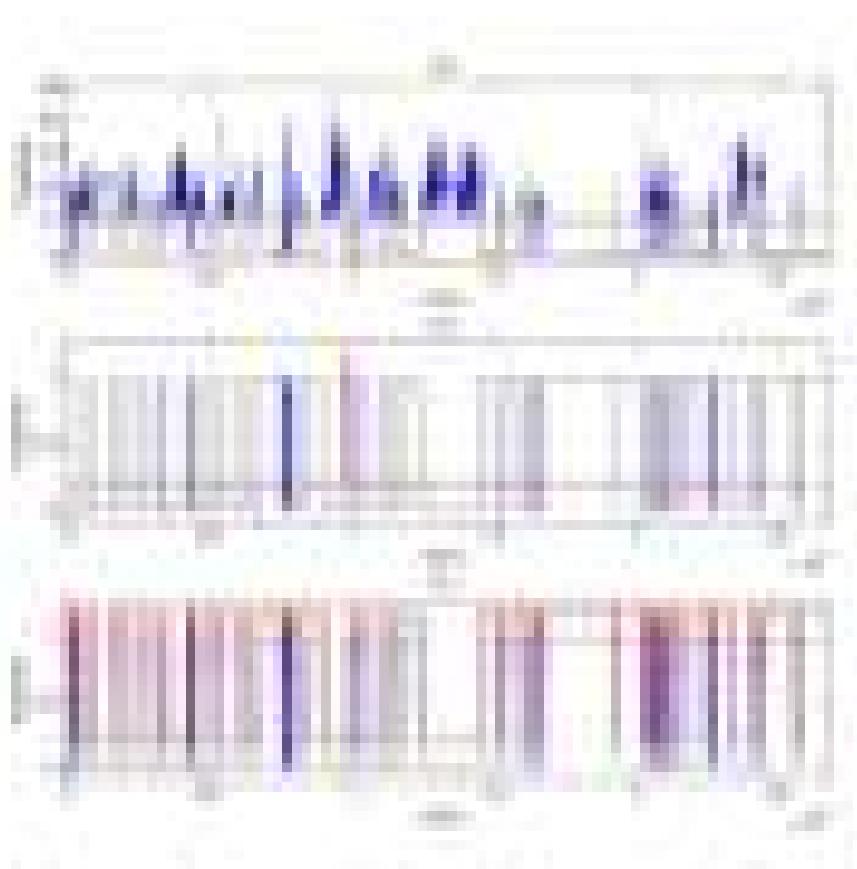


图 4.83 例6中 $x(t)$ 、 $x_a(t)$ 、 $x_{s,2}(t)$ 及 $x_{a,s,2}(t)$ 的时间序列

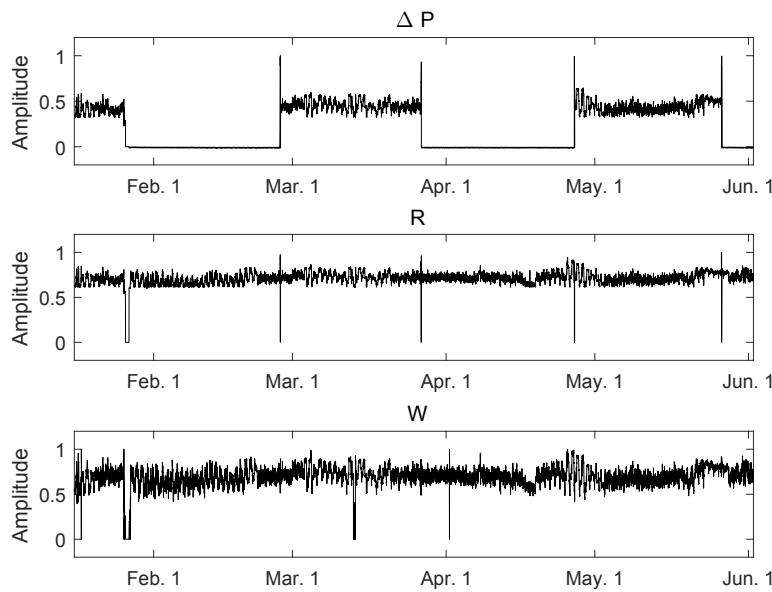


图 4.84 $\Delta P(t)$ 、 $R(t)$ 和 $W(t)$ 的时间序列

4.6 基于变化速率的多变量报警器设计

本节以现代工业中的供热通风与空气调节（Heating, Ventilation and Air Conditioning, HVAC）系统为研究对象，介绍了多变量时变过程的报警器设计方法。工业生产中，对变电站、恒温机房、仓库等的房间温度、湿度要求非常高。但一般情况下这类房间都处于无人值守状态，一旦HVAC系统发生故障，无法及时检测房间温度是否超出正常范围。因此，设计一套检测HVAC系统异常情况的报警器十分重要，及时排除空调系统的异常故障，从而减少经济损失。本部分以工业系统中的HVAC系统为对象，研究其电泵运行状态，提取正常和异常情况的模型特征来设计多变量时变过程报警器。

4.6.1 研究问题描述

本小节提出了一种区分电泵正常和异常状态的新特征，并描述了要解决的问题。考虑HVAC系统带变频驱动的冷凝水电泵。图4.84显示了该泵2015年1月15日至2015年6月1日三个过程变量的时间序列，即入口流体和出口流体之间的压差 $\Delta P(t)$ 、转速 $R(t)$ 和流量 $W(t)$ 。这里 $t \in Z$ 表示采样时间索引， Z 表示一组非负整数。为了消除不同振幅的影响，将三个变量的所有数据样本归一化到范围[0, 1]。2015年1月24日10:40、2015年3月25日23:35、2015年5月25日13:00发生三起水泵事故。在这些事故中，由于 $\Delta P(t)$ 从非零值突然变为零，泵在紧急情况下被迫关闭。一旦泵关闭，就自动启动备用泵，因为两个泵共享 $R(t)$ 和 $W(t)$ 的相同传感器，所以在事故中 $R(t)$ 和 $W(t)$ 的值不等于零。接下来在以下两种情况下进一步研究泵的三个过程变量。

在第一种情况下， $\Delta P(t)$ 、 $R(t)$ 和 $W(t)$ 从2015年3月15日的11:34:00到12:57:40的时间序列

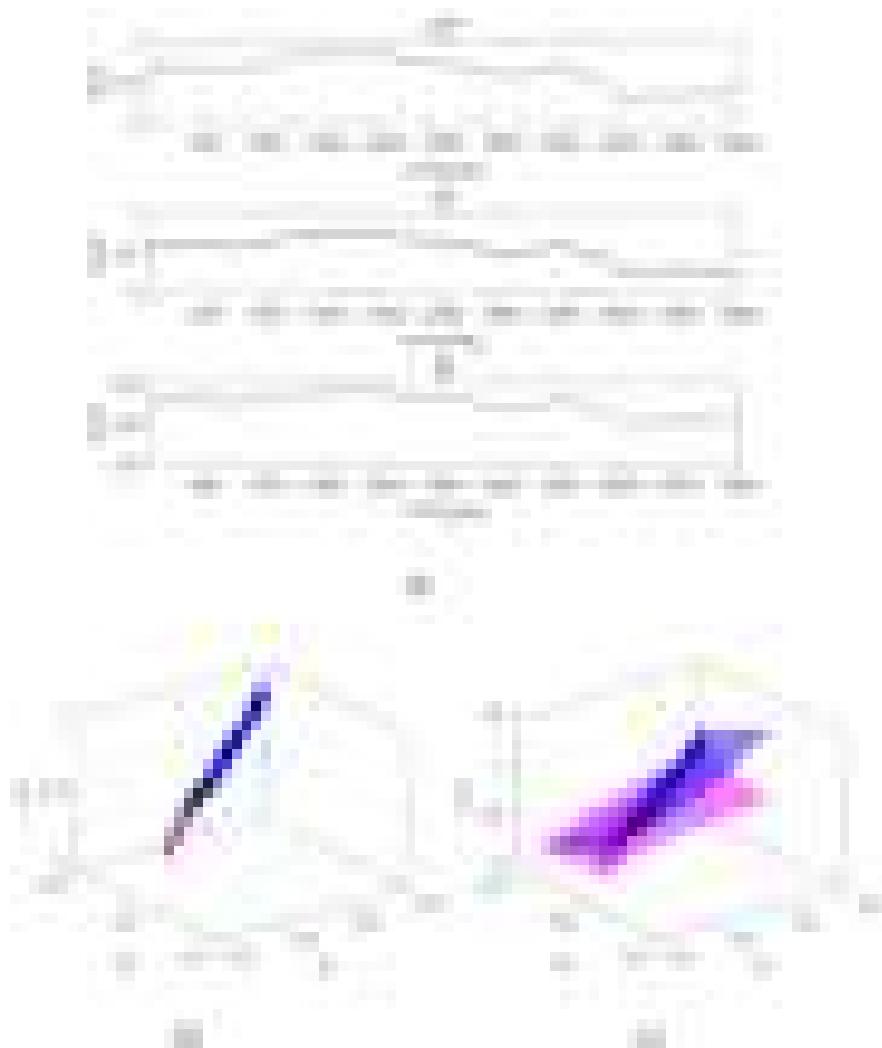


图 4.85 $\Delta P(t)$ 、 $R(t)$ 和 $W(t)$ 的时间序列及第一种情况下的三维散点图

显示在图4.85(a)中，其中已知泵在整个时间段内处于正常状态，采样周期为1s。三个变量从不同视角的三维散射图如图4.85(b)和(c)所示。从图可以看出形成了两组点簇，即前3750s的蓝星点和后1043s的紫色点，中间位置的其余207个黑三角点处于两组点簇之间的过渡阶段。这些簇代表了泵的两种不同的正常运行条件，它们可以使用如下数值方法来确认。基于物理原理，泵可以用具有时变参数的多元线性回归模型来描述：

$$y(t) = x^T(t)\beta(t), \quad (4.84)$$

其中

$$\begin{aligned} y(t) &:= \Delta P(t), \beta(t) := [\beta_0(t), \beta_1(t), \beta_2(t), \beta_3(t)]^T, \\ x(t) &:= [1, x_1(t), x_2(t), x_3(t)]^T = [1, R^2(t), W(t)R(t), W^2(t)]^T. \end{aligned}$$

由最小二乘法(OLS)估计参数向量 $\beta(t)$ 可得 $\hat{\beta}(t) = (X^T X)^{-1} X^T Y$ ，其中 $X := [x(1), \dots, x(t)]^T$

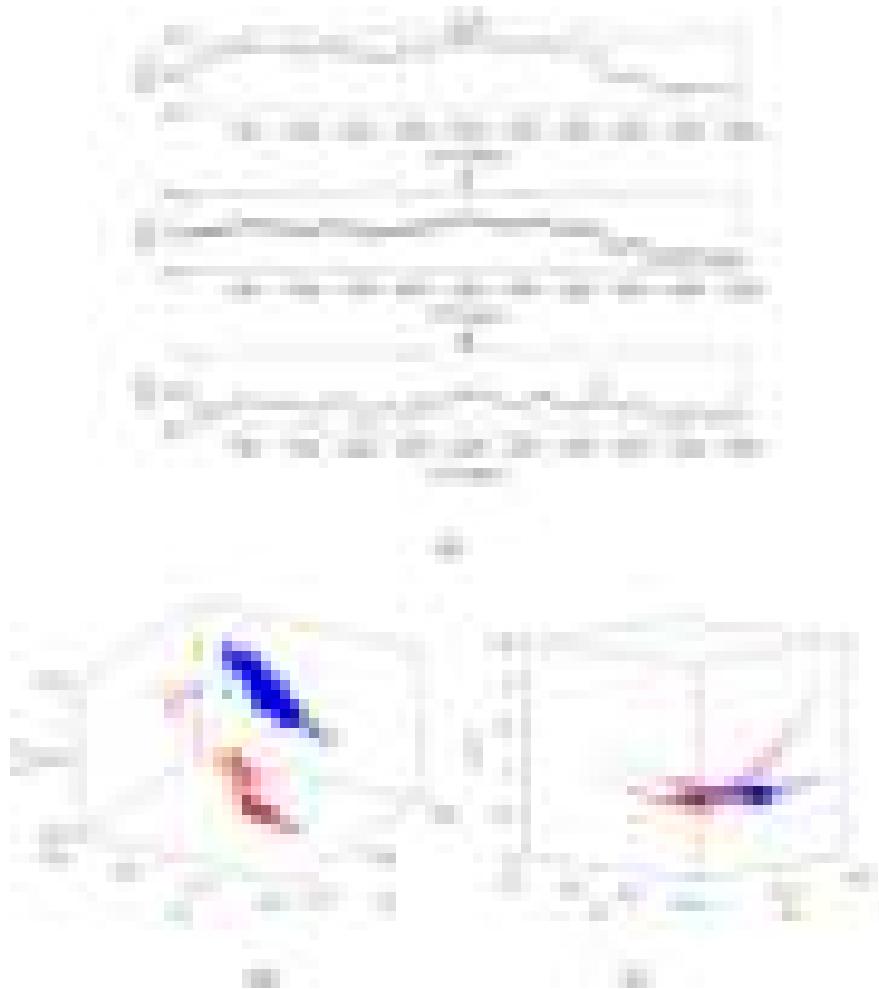


图 4.86 $\Delta P(t)$ 、 $R(t)$ 和 $W(t)$ 时间序列及第二种情况下的三维散点图

和 $Y := [y(1), \dots, y(t)]^T$ 。如果使用图4.85中的蓝色散点，可估计 $\beta(t)$ 的值为 $\hat{\beta}(t) = [-0.3649, 2.2064, -0.9629, 0.3658]^T$ ，以及相应地估计 $\hat{\Delta}p(t) = x^T(t)\hat{\beta}(t)$ ， $W(t)$ 和 $R(t)$ 如图4.85(b)中的浅蓝色表面所示。在温和条件下，估计参数向量 $\beta(t)$ 逐渐逼近高斯分布 $\mathcal{N}\left(\beta, \hat{\sigma}^2 (X^T X)^{-1}\right)$ ，其中 $\hat{\sigma}^2$ 是模型 $y(t) - x^T(t)\hat{\beta}(t)$ 样本的残差。由此，可以求得 $\beta(t)$ 的95%置信区间为 $[(-0.3708, 0.3590), (2.1876, 2.2252), (-0.9684, -0.9574), (0.3656, 0.3660)]^T$ 。类似地，如果使用图中紫点的散点，则 $\hat{\beta}(t) = [-0.9759, 2.4453, -0.4928, 0.1804]^T$ ，相应的曲面为图中紫色部分。而这两组 $\hat{\beta}(t)$ 是不一样的，因为紫色点 $\beta(t)$ 不在蓝色星点 $\beta(t)$ 的95%置信区间内；通过比较浅蓝色和紫色表面也可以清楚地看到它们的区别。因此，泵在前3750s与后1043s处于不同的正常运行状态。

第二种情况是2015年3月25日23:35:00泵发生的事故。图4.86(a)放大了事故发生前20:31:40至21:55:00的5000个数据点。图4.86(b)和图4.86(c)给出了具有不同视角的 $\Delta P(t)$ 、 $R(t)$ 和 $W(t)$ 的三维散点图。很明显有两组点簇，即前3670秒的蓝星点和前1327秒的红点。在两个集群之间，中间只有3个数据点（黑色三角形点）。与第一种情况类似，式(4.84)中的参数估计为 $\hat{\beta}(t) = [-0.7937, 2.7290, -1.1682, 0.4070]^T$ ，95%置信区间为 $[(-2.0732, 0.4858), (-1.0252, 6.4853), (-1.8579,$

图 4.87 $W(t)$ 的时间序列及报警变量

$-0.4785), (0.4069, 0.4071)]^T$, 将OLS方法应用于蓝星数据点, 则相应的估计 $\hat{\Delta}P(t)$ 、 $R(t)$ 和 $W(t)$, 可表示为图4.86(b)中的蓝色表面。如果使用红点, 则得到 $\hat{\beta}(t) = [-4.7213, -5.0859, 1.7412, 0.2953]^T$ 和红色表面。显然, 这两组点集代表了泵的两种不同的工作条件。从图4.86(a)可以观察到异常情况: 在最初的3670秒后, $\Delta P(t)$ 和 $R(t)$ 都下降了, 而 $W(t)$ 则以短而大的增量反向移动, 而此现象是异常的。因为当泵处于正常状态时, $\Delta P(t)$ 、 $R(t)$ 和 $W(t)$ 应在同一方向上发生变化。

测量的流体流量 $W_m(t)$ (在归一化之前)最初配置了具有高报警阈值560的单变量报警器。对于图4.84中的归一化变量 $W(t)$, 当 $W(t) \geq 0.77$ 时即发出报警, 即

$$W_a(t) = \begin{cases} 1, & W(t) \geq 0.77 \\ 0, & \text{其他} \end{cases}, \quad (4.85)$$

其中 $W_a(t)$ 表示与 $W(t)$ 关联的报警变量。下面我们来研究一下单变量报警器在上述两种情况下性能。图4.87(a) 和(b)显示了两种情况下具有报警阈值(红色破折号)的 $W(t)$ 的时间序列图; 图4.87(c)和(d)给出了式(4.85)中相应的报警变量 $W_a(t)$ 。对于第一种情况, 泵在整个时间段内都处于正常状态, 然而, 图4.87(c)中有一些错误报警。对于第二种情况, 泵出现异常, 然而图4.87(d)中却没有发生报警, 则 $W_a(t)$ 在缺少报警方面性能较差。因此, 单变量报警器会产生误报警和漏报警, 这就促使我们设计多变量报警器。注意到重新设计报警阈值或使用诸如延迟器和死区之类的报警策略不能解决单变量报警器中存在的漏报警和误报警的问题。增加报警阈值0.77可以消除图4.87(c)中的误报警, 但不能避免图4.87(d)中的漏报警。相比之下, 降低报警阈值0.77可以减轻图4.87(d)中的漏报警; 然而, 这样做会导致图4.87(c)中更多的误报警, 因为图4.87(a)中的 $W(t)$ 的最大值大于图4.87(b)中的对应值。另外, 使用延迟计时器可以消除图4.87(c)中的误报警, 但对图4.87(d)中的漏报警没有积极影响。

我们的目标是设计一个基于式(5.1)的通用模型进行描述的多变量时变过程报警器。

$$y(t) = x^T(t)\beta(t) + \varepsilon(t), \quad (4.86)$$

其中

$$x(t) := [1, x_1(t), x_2(t), \dots, x_M(t)]^T,$$

$$\beta(t) := [\beta_0(t), \beta_1(t), \beta_2(t), \dots, \beta_M(t)]^T.$$

此处 M 是正整数, $\varepsilon(t)$ 是建模误差。比较上述两种情况可知, 第二种情况中相邻操作条件之间的转换比第一种情况中的转换要短得多。可利用该特征区分正常和异常条件: 如果过程在正常(异常)条件下, 那么 $\beta(t)$ 在公式(4.86)中的变化率很小(大)。若利用模型估计器仅跟踪变化率小的 $\beta(t)$, 则当变化率大时, 绝对累积建模误差将很大, 这种异常将引起报警。

4.6.2 多变量报警器设计主要思想

定义 $\varphi(t)$ 为绝对的累计模型误差

$$\varphi(t) = \left| \sum_{l=t-N}^t \varepsilon(l) \right|, \quad (4.87)$$

其中 $\varepsilon(l)$ 是式(4.86)中的建模误差, 正整数 N 代表窗口长度。在贝叶斯滤波器的基础上, 综合当前数据点的先验估计和信息来设计一个模型估计器。通过考虑第二节中讨论的特性, 我们设计了这样一个过滤器, 当这些参数的变化率很小时, 跟踪时变参数, 而当变化率很大时, 不跟踪时变参数。在后一种情况下, 由于无法跟踪时变参数的转换以至于式(4.87)中的 $\varphi(t)$ 很大。因此, 正常情况下的 $\varphi(t)$ 值与异常情况下的对应值之间存在明显差异。利用这样一个明显的差异, 对较大的 $\varphi(t)$ 值发出报警, 即

$$a(t) = \begin{cases} 1, & \varphi(t) \geq \varphi_{th} \\ 0, & \text{其他} \end{cases}, \quad (4.88)$$

其中 φ_{th} 是后面待确定的报警阈值。

4.6.3 贝叶斯模型估计器

首先, 定义一个滑动数据窗口, 窗口长度为 N , 序列号为 j 。第 j 个窗口包括从第 $[N(j-1)+1]$ 个采样瞬间到第 Nj 个采样瞬间的 $x(t)$ 和 $y(t)$ 数据点。使用 y_j 、 x_j 、 β_j 和 ε_j 分别表示第 j 个窗口中的 y 、 x 、 β 和 ε 。因此, 输出变量可以估计为, 建模误差为

$$\varepsilon_j = y_j - \hat{y}_j. \quad (4.89)$$

其次, 根据贝叶斯理论, β_j 是通过最大化后验分布函数来估计获得的, 即

$$p(\beta_j | Y_j) = \frac{p(Y_j | \beta_j) \cdot p(\beta_j)}{p(Y_j)}$$

$$\propto p(Y_j | \beta_j) \cdot p\beta,$$

其中

$$Y_j = [y(N(j-1)+1) \ y(N(j-1)+2) \ \cdots \ y(Nj)]^T.$$

这里 p 代表概率密度函数。假设 ε_j 为高斯分布，平均值为零，方差为 σ_ε^2 ，则 β_j 的解析表达式可通过如下方式获得。由中心极限定理可知 β_j 和 β_{j-1} 是高斯分布的，即 $\beta_j \sim \mathcal{N}(\mu_{\beta_j}, \Sigma_{\beta_j})$ ， $\beta_{j-1} \sim \mathcal{N}(\mu_{\beta_{j-1}}, \Sigma_{\beta_{j-1}})$ 。其中， μ_{β_j} ($\mu_{\beta_{j-1}}$) 和 Σ_{β_j} ($\Sigma_{\beta_{j-1}}$) 分别是 β_j (β_{j-1}) 的均值向量和协方差矩阵， σ_ε^2 是 ε_j 的方差。在正常情况下， β_t 的变化率很小，因此以 β_{j-1} 作为 β_j 的先验估计是合理的。 β_j 的后验分布为

$$\begin{aligned} p(\beta_j | Y_j) &\propto \exp \left[-\frac{1}{2} (Y_j - X_j \beta_j)^T \sigma_\varepsilon^{-2} (Y_j - \beta_j) \right] \\ &\cdot \exp \left[-\frac{1}{2} (\beta_j - \mu_{\beta_{j-1}})^T \Sigma_{\beta_{j-1}}^{-1} (\beta_j - \mu_{\beta_{j-1}}) \right] \\ &\propto -\beta_j^T \left(\sigma_\varepsilon^{-2} X_j^T X_j + \Sigma_{\beta_{j-1}}^{-1} \right) \beta_j - 2 \left(\sigma_\varepsilon^{-1} Y_j^T X_j + \mu_{\beta_{j-1}} \Sigma_{\beta_{j-1}}^{-1} \right) \beta_j, \end{aligned} \quad (4.90)$$

其中

$$X_j = \begin{bmatrix} 1 & \cdots & 1 \\ x_1(Nj-N+1) & \cdots & x_1(Nj) \\ \vdots & \ddots & \vdots \\ x_M(Nj-N+1) & \cdots & x_M(Nj) \end{bmatrix}^T.$$

对于式(4.84)的模型， X_j 的分量是相互相关的。为了减少共线性对参数估计的负面影响，借鉴了回归的思想，将惩罚项引入后验分布，其中 k_r 是惩罚项的权重参数。此外，引入遗忘因子 $0 < k_p < 1$ 来控制先验信息的可信度。因此，公式(4.90)的右侧变成

$$\begin{aligned} &-\beta_j^T \left(\sigma_\varepsilon^{-2} X_j^T X_j + k_p \Sigma_{\beta_{j-1}}^{-1} \right) \beta_j - 2 \left(\sigma_\varepsilon^{-1} Y_j^T X_j + k_p \mu_{\beta_{j-1}} \Sigma_{\beta_{j-1}}^{-1} \right) \beta_j - k_r \beta_j^T \beta_j \sigma_\varepsilon^{-2} \\ &= -\beta_j^T \left(\sigma_\varepsilon^{-2} X_j^T X_j + k_r I \sigma_\varepsilon^{-2} + k_p \Sigma_{\beta_{j-1}}^{-1} \right) \beta_j - 2 \left(\sigma_\varepsilon^{-1} Y_j^T X_j + k_p \mu_{\beta_{j-1}} \Sigma_{\beta_{j-1}}^{-1} \right) \beta_j, \end{aligned}$$

其中 I 是单位矩阵。定义 $Z_1 := \sigma_\varepsilon^{-2} X_j^T X_j + k_r I \sigma_\varepsilon^{-2} + k_p \Sigma_{\beta_{j-1}}^{-1}$ 和 $Z_2 := \sigma_\varepsilon^{-2} Y_j^T X_j + k_p \mu_{\beta_{j-1}} \Sigma_{\beta_{j-1}}^{-1}$ 。这可以获得 β_j 的惩罚后验分布，

$$p(\beta_j | Y_j) \propto -(\beta_j - Z_1^{-1} Z_2)^T Z_1 (\beta_j - Z_1^{-1} Z_2). \quad (4.91)$$

通过最小化公式(4.91)中的 $p(\beta_j | Y_j)$ ，可以以均值矢量的形式求得 β_j ，即

$$\beta_j := \mu_{\beta_j} = Z_1^{-1} Z_2 = \left[\sigma_\varepsilon^{-2} (X_j^T X_j + k_r I) + k_p \Sigma_{\beta_{j-1}}^{-1} \right]^{-1} \cdot (\sigma_\varepsilon^{-2} X_j^T Y_j + k_p \mu_{\beta_{j-1}} \Sigma_{\beta_{j-1}}^{-1}). \quad (4.92)$$

以及协方差矩阵

$$\Sigma_j = Z_1^{-1} = \left[\sigma_\varepsilon^{-2} (X_j^T X_j + k_r I) + k_p \Sigma_{\beta_{j-1}}^{-1} \right]^{-1}. \quad (4.93)$$

这里 $\sigma_\varepsilon^{-2} (X_j^T X_j + k_r I)$ 和 $k_p \Sigma_{\beta_{j-1}}^{-1}$ 分别为第 j 个窗口和先验信息 β_{j-1} 信息的可靠性度量。由于遗忘因子 $k_p \in (0, 1)$ ， β_{j-1} 在第一步 $j = 1$ 的先验分布并不重要，因为随着 j 的增加，其对 β_j 的影响很快衰减。因此，在 $j = 1$ 时，通过忽略先验信息，从式(4.92)和式(4.93)中获得 β_j 的平均向

表 4.3 第一种情况下不同 k_p 值下的适应度值

	$k_p = 0$	$k_p = 0.3$	$k_p = 0.9$
F	96.29%	96.17%	94.42%

量和协方差矩阵,

$$\begin{cases} \mu_{\beta_j} = (X_j^T X_j + k_r I)^{-1} X_j^T Y_j, & j = 1. \\ \Sigma_{\beta_j} = \sigma_\varepsilon^2 (X_j^T X_j + k_r I)^{-1} \end{cases}$$

4.6.4 调谐参数设计

本节设计了模型估计器(4.92)中的三个参数, 即窗口长度 N 、遗忘因子 k_p 和惩罚因子 k_r 。小的 N 值对跟踪时变模型参数的变化有明显的帮助。然而, X_j 中的行数通常需要大于 X_j 中的列数, 以生成非奇异矩阵 $X_j^T X_j$ 。因此, 模型(4.86)使用 N 的最小可能值, 即

$$N = M + 2. \quad (4.94)$$

遗忘因子 k_p 是一个关键的调谐参数。需要两个历史数据集 (称为正常和异常数据集) 来确定 k_p 的适当值。正常数据集不包含异常情况, 而异常数据集包含异常情况。将 $\varphi_1(t)$ 和 $\varphi_2(t)$ 分别表示为式(4.87)中正常和异常数据集的 $\varphi(t)$ 值。计算 $y(t)$ 与其估计值 $\hat{y}(t)$ 之间的适应度来衡量模型质量,

$$F := \max \left(0, 1 - \frac{\|y(t) - \hat{y}(t)\|_2}{\|\overline{y(t)} - \hat{y}(t)\|_2} \right). \quad (4.95)$$

这里第 j 个数据窗口中的 $\hat{y}(t)$ 与式(4.89)中的 $\hat{y}(t)$ 相同, $\|\cdot\|$ 表示操作数的 2-范数, $\overline{y(t)}$ 是 $y(t)$ 的样本平均数。下面基于本节第一和第二个案例, 研究参数 k_p 的影响。图4.88(a)、(b)和(c)为三个不同的遗忘因子 $k_p = 0$ 、 $k_p = 0.3$ 和 $k_p = 0.9$, 以及第一种情况下的正常数据集获得的 $\varphi_1(t)$ 。表4.3展示了第一种情况下不同 k_p 值下的适应度值。正如预期的那样, 随着 k_p 的增加, 适应度降低, 这是因为先前信息的权重越大, 模型估计器的可追溯性就越低。图4.88(d)、(e)和(f)分别表示第二种情况下 $k_p = 0$ 、 $k_p = 0.3$ 和 $k_p = 0.9$ 的异常数据集 $\varphi_2(t)$ 。当异常情况发生时, 先验信息的权重越大, 建模误差越大。因此, 期望选择 k_p 的适当值可以使得在检测异常情况和跟踪正常条件下参数变化的能力之间达到平衡。这里, k_p 由最大化 $\varphi_1(t)$ 和 $\varphi_2(t)$ 之差确定, 同时满足适应度的要求

$$k_p^{(\text{opt})} = \arg \max_{k_p} (\max(\varphi_2(t)) - \max(\varphi_1(t))) \quad (4.96)$$

$$\text{s.t. } F \geq F_0,$$

其中, F_0 是用户定义的阈值, 作为适应度的下限, 例如可取 $F_0 = 95\%$ 。

由于共线性, k_r 值需要在式(4.89)中的建模误差 ε_j 与式(4.92)中的 β_j 的可变性之间折衷。测量估计回归系数变异性的因子 (VIF) 的方差用于选择合适的 k_r 值。在正常历史数据集的第 j 个

图 4.88 k_p 对 $\varphi(t)$ 的影响说明

窗口中，可以将 VIF_j 计算为

$$VIF_j = (X_j^T X_j + k_r I)^{-1} X_j^T X_j (X_j^T X_j + k_r I)^{-1}.$$

定义 VIF 为窗口 J 内所有 VIF_j 的平均值，

$$VIF = \frac{1}{J} \sum_{j=1}^J VIF_j.$$

k_r 选取为所有小于 0.1 的 VIF 值的最大值，即

$$k_r^{(opt)} = \max(k_r | VIF < 0.1). \quad (4.97)$$

4.6.5 设计步骤

如图 4.89 中的示意图所示，所提出的方法由以下步骤组成：

步骤1：设计调谐参数。基于历史正常和异常数据集确定模型估计器(4.92)中的调谐参数 N , k_p 和 k_r 。式(4.88)中的报警阈值 φ_{tp} 设定为

$$\varphi_{tp} = \frac{\max(\varphi_1(t)) + \max(\varphi_2(t))}{2}. \quad (4.98)$$

显然，步骤1是只需要执行一次的步骤。



图 4.89 多变量报警器设计流程图

步骤2：确定是否应该为一组新数据点发出报警。在当前采样时刻，采集模型(4.86)的数据点 $x_1(t), x_2(t), \dots, x_M(t)$ 和 $y(t)$ 。根据前一个数据窗口中的模型参数 β_{j-1} 计算得到式(4.89)中的绝对累积建模误差 $\varphi(t)$ ，进而根据报警阈值 φ_{tp} 生成报警变量(4.88)。

步骤3：进行假设试验，以确定是否应更新模型。在第 j 次采样时刻，引入假设检验，判断第 j 个窗口中的过程变量是否处于正常或异常状态，并确定模型是否需要更新。首先，第 j 个窗口中的报警数计为 A_j 。如果第 j 个窗口处于正常状态，则发生的报警通常是由随机噪声引起的，因此它们是独立的，分布相同。因此， A_j 服从二项分布，即：

$$A_j \sim \mathcal{B}(N, \text{FAR}). \quad (4.99)$$

这里FAR是所设计报警器的误报率。接下来，设定显着性水平 α ，例如 $\alpha=0.05$ ，假设 A_α 是满足如下等式的阈值，

$$\Pr(A_j \leq A_\alpha) = 1 - \alpha. \quad (4.100)$$

最后，模型的参数矢量由下式假设检验更新

$$\begin{cases} \beta_j = \beta_{j-1}, & \text{if } A_j > A_\alpha \\ \beta_j (4.92), & \text{if } A_j \leq A_\alpha \end{cases}. \quad (4.101)$$

如果 $A_j > A_\alpha$ ，则第 j 个窗口是异常的，因此，不应更新模型参数并保留 β_{j-1} 。如果 $A_j < A_\alpha$ ，则第 j 个窗口处于正常状态，并用式(4.92)更新模型参数。如果判断第 j 个窗口处于正常状态，

则第 j 个窗口中发生的报警是误报警;否则,在第 j 个窗口中取值为“0”的 $a(t)$ 的报警间隙是漏报警。通过计算过去数据窗口的错误和丢失报警的频率,可以获得设计的报警器的误报率和漏报率。

步骤4: 重复步骤2和3, 在每个采样时刻更新(4.88)中的报警变量 $a(t)$, 并且对于每 N 个样本更新(4.101)中的模型参数矢量。这两个在线步骤只涉及简单的代数计算, 可以非常快速地完成。因此, 所提方法也可实现实际中的在线运行。

4.6.6 案例分析

本节提供了两个HVAC系统冷凝水电泵的工业案例研究, 以说明本文所提方法的有效性。参数是通过将4.5.1节中的第一和第二种情况中的案例作为正常和异常的历史数据集来设计的, 如调谐参数 k_p 、 k_r 、报警阈值 φ_{tp} 和假设阈值 A_α 。

首先, 表4.4给出了第一种情况下不同 k_r 值对应的VIF, 则令(4.97)中的 $k_r^{(opt)}$ 值为0.009, 以使VIF的每个元素不大于0.1。其次, 针对第一种情况的不同 k_p 值, 表4.5列出了不同 k_p 值下的 $(\max(\varphi_2(t)) - \max(\varphi_1(t)))$ 的值和适应度 F 的值。通过求解 $\varphi_1(t)$ 和 $\varphi_2(t)$ 之差的最大值且满足适应性要求 $F \geq 95\%$, 求得 k_p 的最优值 $k_p^{opt} = 0.84$, 则报警阈值 $\varphi_{tp} = \frac{0.4245 + 0.15032}{2} = 0.2874$ 。最后, $\varphi_1(t)$ 的样本方差估计 $\hat{\sigma}_{\varphi_1}^2 = 0.1503$, 因此当 $\alpha=0.05$ 时, $A_\alpha=1$ 。

利用Kolmogorov-Smirnov检验来验证模型误差 $\varepsilon(t)$ 是否遵循高斯分布。图4.90比较了标准高斯累积分布函数(CDF)与第二节中第一种情况下 $(\varepsilon(t) - \mu_\varepsilon)/\sigma_\varepsilon$ 的CDF估计值, 其中 μ_ε 和 σ_ε 分别是 ε 的样本均值和标准差。从图4.90可看出两个CDF几乎重叠, 因而Kolmogorov-Smirnov检验是正确的。

下面进行多变量报警器设计, 设计的多变量报警器应用于两个数据集, 称为第三和第四种情况, 它们与第二节中的第一和第二种情况不同。

表 4.4 第一种情况下不同 k_r 值对应的VIF

	VIF
$k_r = 0.006$	$[0.1184, 0.0487, 0.1010, 0.1371]^T$
$k_r = 0.009$	$[0.0619, 0.0305, 0.0534, 0.0988]^T$
$k_r = 0.012$	$[0.0422, 0.0242, 0.0368, 0.0853]^T$
$k_r = 0.015$	$[0.0330, 0.0212, 0.0290, 0.0790]^T$

表 4.5 第一种情况下不同 k_p 值对应的适应度F

	$\max(\varphi_2(t)) - \max(\varphi_1(t))$	F
$k_p = 0.74$	$0.3216 - 0.1225 = 0.1902$	95.52%
$k_p = 0.79$	$0.3612 - 0.1331 = 0.2281$	95.32%
$k_p = 0.84$	$0.4245 - 0.1503 = 0.2742$	95.04%
$k_p = 0.89$	$0.5086 - 0.1758 = 0.3328$	94.56%



图 4.90 累积分布函数 (CDF) 估计效果

图 4.91 第三种情况下的 $\Delta P(t)$ 、 $R(t)$ 和 $W(t)$ 的时间序列

在第三种情况下，泵在整个时间段内处于正常状态：图4.91显示了从2015年5月17日9:32:10到10:23:20三个变量 $\Delta P(t)$ 、 $R(t)$ 和 $W(t)$ 的5000个数据点。 $\beta_j(t)$ 中的四个参数如图4.92所示，所有参数显然都不是恒定的，说明了泵处于一个时变过程。图4.93给出了绝对累积建模误差 $\varphi(t)$ 和设计的报警阈值 φ_{tp} ，由图可得模型估计器能够跟随时变参数的变化，使得 $\varphi(t) < \varphi_{tp}$ ，不发生报警。

在第四种情况下泵出现异常：图4.94显示了从2015年5月25日11:29:40到12:53:00的三个变量 $\Delta P(t)$ 、 $R(t)$ 和 $W(t)$ 的5000个数据点。在第3200个数据点附近明显发生了一些异常，几分钟后，泵处于紧急情况于13:00:00关闭。将设计的报警系统应用于图4.94中的5000个数据点。图4.95和图4.96分别给出 $\beta_j(t)$ 中的模型参数和绝对累积建模误差 $\varphi(t)$ 。如图4.96(b)所示，在



图 4.92 第三种情况的 $\beta_j(t)$ 值



图 4.93 第三种情况的绝对累积建模误差及报警阈值设计值

第3218个采样时刻发生报警，其中 $\varphi(t) > \varphi_{tp}$ ，并且 $\beta_j(t)$ 中的模型参数突然发生了变化。因此，报警变量 $a(t)$ 在其余采样时刻保持在报警状态，并且之后不再更新模型参数。

将图4.92中的模型参数与图4.95中的模型参数对比可得，图4.95中 $t = 3218$ 处 $\beta_j(t)$ 的转变比图4.92中的对应转变快得多。设计的多变量报警器能够遵循图4.92中第三种情况的模型参数的变化，并且不跟随图4.95中第四种情况的模型参数的变化而变化。因此，设计的多变量报警器性能良好，对于没有异常的第三种情况，如图4.93所示没有发生报警，而由图4.96(b)所示，报警器能成功地检测到第四种情况中的相关异常情况，并产生报警。

4.7 本章小结

本章从五个部分介绍了多变量报警器的设计方法：第一部分首先提出了一种自动划分多变量系统历史数据中正常、异常数据段的方法，然后提出了基于过程变量定性趋势组合的工业报



图 4.94 第四种情况下的 $\Delta P(t)$ 、 $R(t)$ 和 $W(t)$ 的时间序列

警器设计方法，最后提出了基于自适应梯度算法的过程变量定性趋势分析方法；第二部分建立了超椭球体模型来描述工业过程的正常工作区域，利用了投影方法设计各个过程变量的动态报警阈值来监控工业过程；第三部分建立了凸包模型来描述表面不光滑的正常工作区，重新设计了各个过程变量的动态报警阈值，弥补了超椭球体模型在多变量系统应用中的局限；第四部分介绍了长时间报警及其产生原因，并设计了基于状态的动态报警器以解决长时间报警的问题；第五部分给出了基于变化速率的多变量报警器设计方法，提出了一种区分正常和异常状态的新特征，实现了基于绝对建模累积误差的报警检测。同时展示了上述方法在实际工业中的长期应用效果，结果表明本章涉及的多变量报警器设计方法切实有效。

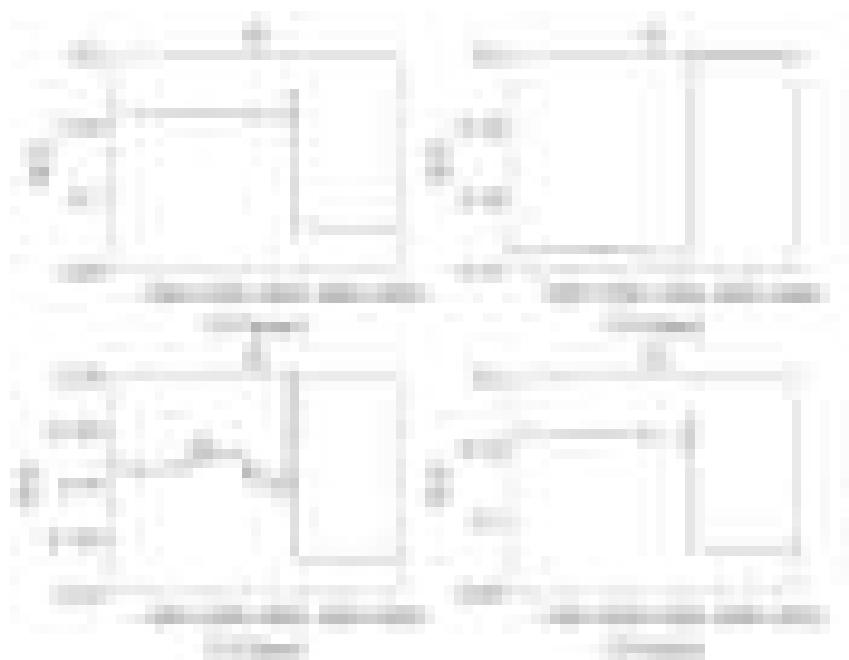


图 4.95 第四种情况的 $\beta_j(t)$ 值

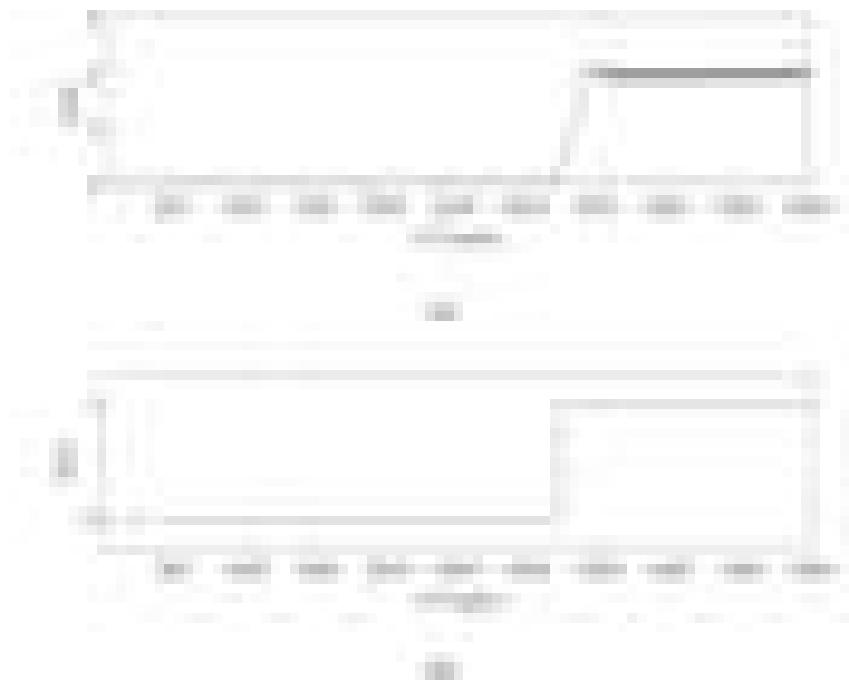


图 4.96 第四种情况的绝对累积建模误差及报警阈值设计值

参考文献

- [1] YIN K K. A review of process fault detection and diagnosis: Part III: Process history based methods[J]. Computers and Chemical Engineering, 2003, 27(3):327-346.
- [2] SAVITZKY A, GOLAY M J. Smoothing and differentiation of data by simplified least squares procedures[J]. Analytical Chemistry, 1964, 36(8):1627-1639.
- [3] ZHOU B, YE H, ZHANG H, et al. A new qualitative trend analysis algorithm based on global polynomial fit[J]. Aiche Journal, 2017, 63(8):3374-3383.
- [4] VEDAM H, VENKATASUBRAMANIAN V, BHALODIA M. A B-spline based method for data compression, process monitoring and diagnosis[J]. Computers and Chemical Engineering, 1998, 22(supp-S1):S827-S830.
- [5] FLEHMIG F, MARQUARDT W. Detection of multivariable trends in measured process quantities[J]. Journal of Process Control, 2006, 16(9):947-957.
- [6] FLEHMIG F, WATZDORF R V, MARQUARDT W. Identification of trends in process measurements using the wavelet transform[J]. Computers and Chemical Engineering, 1998, 22(supp-S1):S491-S496.
- [7] VILLEZ K, ROSEN C, ANCTIL F, et al. Qualitative Representation of Trends (QRT): Extended method for identification of consecutive inflection points[J]. Computers and Chemical Engineering, 2013, 48(10):187-199.
- [8] JANUSZ M E, VENKATASUBRAMANIAN V. Automatic generation of qualitative descriptions of process trends for fault detection and diagnosis[J]. Engineering Applications of Artificial Intelligence, 1991, 4(5):329-339.
- [9] VILLEZ K. Qualitative path estimation: A fast and reliable algorithm for qualitative trend analysis[J]. AIChE Journal, 2015, 61(5):1535-1546.
- [10] RENGASWAMY R, VENKATASUBRAMANIAN V. A syntactic pattern-recognition approach for process monitoring and fault diagnosis[J]. Engineering Applications of Artificial Intelligence, 1995, 8(1):35-51.
- [11] HUANG G, ZHOU X. A piecewise linear representation method of hydrological time series based on curve feature[C]// IEEE International Conference on Intelligent Human-machine Systems and Cybernetics, Institute of Electrical and Electronics Engineers, 2016:203-207.

- [12] International Society of Automation. ANSI/ISA-18.2. Management of alarm systems for the process industries[S]. International Society of Automation, 2009.
- [13] BJORKLUND S. A survey and comparison of time-delay estimation methods in linear systems[D]. Sweden: Linkoping University, 2003.
- [14] RODRIGUEZ A, LAIO A. Clustering by fast search and find of density peaks[J]. science, 2014, 344(6191):1492.
- [15] KEOGH E, CHU S, HART D, et al. An online algorithm for segmenting time series[C]// IEEE International Conference on Data Mining, Institute of Electrical and Electronics Engineers, 2001:289-296.
- [16] ALFRED INSELBERG. Parallel coordinates: Visual multidimensional geometry and its applications[M]. New York: Springer-Verlag New York, 2009.
- [17] MAURICIO M, ANDRES F, PABLO G, et al. A robust clustering method for detection of abnormal situations in a process with multiple steady-state operation modes[J]. Computers and Chemical Engineering, 2010, 34(2):223-231.
- [18] BARBER C B, DOBKIN D P, HUHDANPAA H. The quickhull algorithm for convex hulls[J]. ACM Trans.Mathematical Software, 1996, 22(4):469-483.
- [19] BOYD S, VANDENBERGHE L. Convex optimization[M]. Cambridge: Cambridge University Press, 2004.
- [20] BAKSHI B R, STEPHANOPOULOS G. Representation of process trends—III. Multiscale extraction of trends from process data[J]. Computers and Chemical Engineering, 1994, 18(4):267-302.
- [21] Engineering Equipment and Materials Users' Association. EEMUA-191. Alarm systems-A guide to design, management and procurement[S]. Engineering Equipment and Materials Users' Association, 2013.
- [22] WANG J D, CHEN T W. An online method to remove chattering and repeating alarms based on alarm durations and intervals[J]. Computers and Chemical Engineering, 2014, 67(4):43-52.
- [23] SAVITZKY A. Smoothing and differentiation of data by simplified least squares procedures[J]. Analytical Chemistry, 1964, 36(8):1627-1639.
- [24] HOLLIFIELD B, HABIBI E. The alarm management handbook[M]. 2nd ed. Houston: PAS, 2010.
- [25] ROTHEMBERG D H. Alarm management for process control[M]. New York: Momentum Press, 2009.
- [26] BRANSBY M L, JENKINSON J. The management of alarm systems[M]. Sheffield: Health and Safety Executive, 1998.

第5章 报警根源分析

5.1 引言

报警根源分析是报警管理的重要组成部分，在报警过载现象发生时，需要及时有效的识别出导致其他报警变量超过阈值的根源报警变量或关键报警变量，并指导操作人员作出调整，避免生产事故的发生。由于工业过程中报警数据驳杂，报警根源具有不确定性和相关性等特点，因此现有的报警根源分析仍然具有较大的研究空间。

对于报警根源分析问题，本章首先介绍了基于传递熵的报警变量根源分析方法，由于该方法需要一定的报警次数，不能完全有效的处理误报警、漏报警及根源不完整问题；然后在此基础上提出了一种基于贝叶斯网络的逻辑报警根源分析方法，能够在短时间内分析发生报警的根本原因，不需要过多的报警次数；最后，根据报警变量之间存在的潜在物理关系，提出了基于变量定性趋势的报警根源分析方法，提取过程变量的趋势信息，通过分析主要过程变量与一个或多个根本原因变量的定性趋势，获取系统的运行规律，并辅助操作人员进行根源分析。

5.2 基于传递熵的报警根源分析

根据式(2.7)配置离散时间报警变量 x_a ，以监控过程变量 x 的状态，假设报警变量具有以下两个特征：

- 1) 如果报警变量 x_a 与报警变量 y_a 之间存在关联关系，则 $y_a(t + \lambda) = 1$ 报警发生紧随报警事件 $x_a(t) = 1$ 的时间间隔 λ 是由于干扰而产生的随机变量。这种时间间隔 λ 称为随机时间延迟。
- 2) 报警变量 $x_a(t)$ 在每个时刻发生报警事件是相互独立的，即

$$Pr[x_a(t_1), x_a(t_2)] = Pr[x_a(t_1)] Pr[x_a(t_2)], \forall t_1 \neq t_2. \quad (5.1)$$

其中 $Pr(\cdot)$ 为报警事件发生的概率。

上述两个假设得到满足的原因如下：①干扰在工业过程中很常见，通常会导致随机延迟的存在^[1]；②由于在时间上存在间隔的异常通常相互独立，因此其相应的报警事件也相互独立。

5.2.1 标准传递熵

设两个离散模拟时间变量 x, y ，其时间序列表示为 $X = \{x_t \in \tilde{X}; t = 1, 2, \dots, L\}$, $Y = \{y_t \in \tilde{Y}; t = 1, 2, \dots, L\}$ ，其中 L 表示序列长度， $\tilde{X}、\tilde{Y}$ 分别表示 x_t 和 y_t 的变化范围。时间序列 X 与 Y 之间的传递熵（TE）为

$$T_{X \rightarrow Y} = H(y_{t+1} | y_t^{(k)}) - H(y_{t+1} | y_t^{(k)}, x_t^{(l)}), \quad (5.2)$$

其中

$$H(y_{t+1} | y_t^{(k)}, x_t^{(l)}) = \int_{y_{t+1} \in \tilde{Y}, y_t^{(k)} \in \tilde{Y}^{(k)}, x_t^{(l)} \in \tilde{X}^{(l)}} p(y_{t+1}, y_t^{(k)}, x_t^{(l)}) \log_2 \frac{1}{p(y_{t+1} | y_t^{(k)}, x_t^{(l)})}. \quad (5.3)$$

$$H(y_{t+1} | y_t^{(k)}) = \int_{y_{t+1} \in \tilde{Y}, y_t^{(k)} \in \tilde{Y}^{(k)}} p(y_{t+1}, y_t^{(k)}) \log_2 \frac{1}{p(y_{t+1} | y_t^{(k)})}. \quad (5.4)$$

式中 $p(\cdot)$ 为联合概率密度函数, $p(\cdot | \cdot)$ 为条件概率密度函数, \int 为所有可能的组合, $y_t^{(k)} = [y_t, y_{t-\tau}, \dots, y_{t-(k-1)\tau}]$ 为 Y 过去状态的嵌入向量, $x_t^{(l)} = [x_t, x_{t-\tau}, \dots, x_{t-(l-1)\tau}]$ 为 X 过去状态的嵌入向量, k, l 分别为 y 和 x 的嵌入维数, τ 为嵌入延迟。与模拟变量不同, 报警变量是取值为 0 或 1 的二值变量。TE 已被开发用于通信和神经科学领域的二进制数据, 然而很少有人将 TE 应用在报警数据中。而且由于存在第 4.10 节中的随机时间延迟, 式(5.2) 中传递熵的计算方法不能直接推广到报警变量。

因此, 给定两个离散时间报警变量 x_a 和 y_a , 其时间序列表示为 $X_a = \{x_a(t) \in H; t = 1, 2, \dots, L\}$ 和 $Y_a = \{y_a(t) \in H; t = 1, 2, \dots, L\}$, 其中 L 表示序列长度, H 表示 $x_a(t)$ 和 $y_a(t)$ 的变化范围, 取值为 $H = \{0, 1\}$ 。图 5.1 给出了 x_a 和 y_a 之间随机时间延迟的示例。在图 5.1 所示的虚线窗口中, X_a 中有四个报警出现。 X_a 中每一次报警事件的发生, Y_a 中都有一个相应的报警发生, 并且两者发生时间存在延迟 λ 。四个随机时间延迟的大小为 $\lambda(1) = 4, \lambda(2) = 3, \lambda(1) = 2, \lambda(1) = 5$ 。随机时间延迟导致我们只关心 X_a 中发生的报警, 基于此报警变量的传递熵为

$$T_{X_a \rightarrow Y_a} = H\left(y_a(t+1) | \overset{\circ}{y}_a^{(k_1)}(t)\right) - H\left(y_a(t+1) | \overset{\circ}{y}_a^{(k_1)}(t), \overset{\circ}{x}_a^{(l_1)}(t-d_1)\right), \quad (5.5)$$

其中

$$H\left(y_a(t+1) | \overset{\circ}{y}_a^{(k_1)}(t), \overset{\circ}{x}_a^{(l_1)}(t-d_1)\right) = \sum_{y_a(t+1), \overset{\circ}{y}_a^{(k_1)}(t), \overset{\circ}{x}_a^{(l_1)}(t-d_1) \in H^3} p(y_a(t+1), \overset{\circ}{y}_a^{(k_1)}(t), \overset{\circ}{x}_a^{(l_1)}(t-d_1))$$

$$\bullet \log_2 \frac{1}{p\left(y_a(t+1) | \overset{\circ}{y}_a^{(k_1)}(t), \overset{\circ}{x}_a^{(l_1)}(t-d_1)\right)}.$$



图 5.1 基于随机延迟影响下通过 X_a 的历史值预测 Y_a 。

$$H \left(y_a(t+1) \mid \overset{\circ}{y}_a^{(k_1)}(t) \right) = \sum_{y_a(t+1), \overset{\circ}{y}_a^{(k_1)}(t) \in H^2} p(y_a(t+1), \overset{\circ}{y}_a^{(k_1)}(t)) \\ \bullet \log_2 \frac{1}{p \left(y_a(t+1) \mid \overset{\circ}{y}_a^{(k_1)}(t) \right)}. \quad (5.7)$$

式中 $p(\cdot)$ 为联合概率密度函数, $p(\cdot \mid \cdot)$ 为条件概率密度函数, k_1 和 l_1 分别指 y_a 和 x_a 的时间范围, d_1 表示 x_a 和 y_a 之间的时间延迟, \sum 表示 $y_a(t+1), \overset{\circ}{y}_a^{(k_1)}(t), \overset{\circ}{x}_a^{(l_1)}(t-d_1)$ 所有可能的概率组合之和, 其中 $\overset{\circ}{y}_a^{(k_1)}(t), \overset{\circ}{x}_a^{(l_1)}(t-d_1)$ 可按照下式计算:

$$\overset{\circ}{x}_a^{(l_1)}(t-d_1) = \begin{cases} \cup_{u=t-d_1-l_1+1}^{t-d_1} x_a(u), & y_a(t+1) = 1 \\ x_a(t-d_1), & \text{其他} \end{cases} \quad (5.8)$$

$$\overset{\circ}{y}_a^{(k_1)}(t) = \begin{cases} \cup_{u=t-k_1+1}^t y_a(u), & y_a(t+1) = 1 \\ y_a(t), & \text{其他} \end{cases}. \quad (5.9)$$

其中, $\cup_{u=t_1}^{t_2} x_a(u) = x_a(t_1) \vee x_a(t_1+1) \vee \dots \vee x_a(t_2)$, $t_1 < t_2$, \vee 代表‘或’运算。总的来说, 只要 $x_a(t)$ 的历史窗口中任何时刻取值为 1, $\overset{\circ}{x}_a^{(l_1)}(t-d_1)$ 就等于 1, 由此可以消除随机时间延迟的影响。进一步可得, 标准化传递熵 (NTE) 为

$$NT_{X_a \rightarrow Y_a} = \frac{T_{X_a \rightarrow Y_a}}{H \left(y_a(t+1) \mid \overset{\circ}{y}_a^{(k_1)}(t) \right)} \\ = 1 - \frac{H \left(y_a(t+1) \mid \overset{\circ}{y}_a^{(k_1)}(t), \overset{\circ}{x}_a^{(l_1)}(t-d_1) \right)}{H \left(y_a(t+1) \mid \overset{\circ}{y}_a^{(k_1)}(t) \right)}. \quad (5.10)$$

其中, 条件熵 $H \left(y_a(t+1) \mid \overset{\circ}{y}_a^{(k_1)}(t), \overset{\circ}{x}_a^{(l_1)}(t-d_1) \right)$ 和 $H \left(y_a(t+1) \mid \overset{\circ}{y}_a^{(k_1)}(t) \right)$ 都是正值, 并且 $H \left(y_a(t+1) \mid \overset{\circ}{y}_a^{(k_1)}(t) \right) \geq H \left(y_a(t+1) \mid \overset{\circ}{y}_a^{(k_1)}(t), \overset{\circ}{x}_a^{(l_1)}(t-d_1) \right)$, 因此 $NT_{X_a \rightarrow Y_a}$ 的取值范围为 $[0, 1]$, 当 $NT_{X_a \rightarrow Y_a}$ 的取值靠近 0 (1) 时, 表示报警变量之间存在较弱 (强) 的关联关系。

式(5.5)对于传递熵的定义与传统二进制变量传递熵定义不同^[2], 修正后的传递熵为报警变量专用, 它考虑了随机时间延迟的存在, 而在以往的研究中并没有考虑到这一点。在计算报警变量的传递熵时, 三个非负整数参数 d_1, k_1, l_1 的选择应遵循以下原则: D 为用户定义的最大时间延迟值, 其中 $d_1 \leq D$ 。由于报警事件是相互独立的, 因此无需考虑数字量 Y_a 过去状态的影响, 变量 Y_a 的时间范围可以设置为 $k_1 = 0$ 。 l_1 表示 x_a 的历史状态, 并用于 Y_a 的预测。为了消除时间延迟对 $\overset{\circ}{x}_a^{(l_1)}(t-d_1)$ 的影响, l_1 选择标准表示为大于所有的随机时间延迟。设定 $X_a(t)$ 和 $Y_a(t)$ 中数量值出现取值为 1 的时间戳为 $S^{x_a} = [s_1^{x_a}, \dots, s_{M_{x_a}}^{x_a}]^T$, $S^{y_a} = [s_1^{y_a}, \dots, s_{M_{y_a}}^{y_a}]^T$, 则 Y_a 中报警发生与 X_a 中对应报警之间的时间间隔为

$$\Gamma = [\gamma]_{M_{y_a} \times M_{x_a}} = S^{y_a} \cdot (\mathbf{I}_{M_{x_a} \times 1})^T - \mathbf{I}_{M_{y_a} \times 1} \cdot (S^{x_a})^T, \quad (5.11)$$

其中 I 为全1向量, λ_i 为每行中的最小正元素

$$\lambda_i = \min \{\gamma_{i,j} : \gamma_{i,j} > 0, j = 1, 2, \dots, M_{x_a}\}. \quad (5.12)$$

大多数随机时间延迟可从 $1.5 \times IQR$ (四分位范围) 中找到, 而超出范围的值均是异常值。因此, 原因变量的时间范围可以设置为 $l_1 = 1.5 \times IQR_\lambda$, 其中 IQR_λ 是指 $\lambda_i, (i = 1, 2, \dots, M_{y_a})$ 第三个四分位数和第一个四分位数之间的差异。 l_1 的上限为20, 这是根据ISA标准得到无抖动警报的最大时间间隔。因此, 最终确定的 l_1 取值如下:

$$l_1 = \min \{1.5 \times IQR_\lambda, 20\}. \quad (5.13)$$

5.2.2 标准直接传递熵

为了检测两个连续过程变量之间是存在直接的关联关系还是间接的关联关系, Duan提出了直接传递熵(DTE)的概念^[3]。类似的, 报警变量之间的直接传递熵定义如下:

第三个离散时间报警变量 z_a , 其时间序列表示为 $Z_a = \{z_a(t) \in H; t = 1, 2, \dots, L\}$, 则 z_a 和 x_a 之间的传递熵为

$$NT_{X_a \rightarrow Z_a} = 1 - \frac{H(z_a(t+1) | \overset{\circ}{z}_a^{(m_1)}(t), \overset{\circ}{x}_a^{(l_2)}(t-d_2))}{H(z_a(t+1) | \overset{\circ}{z}_a^{(m_1)}(t))}. \quad (5.14)$$

其中 $\overset{\circ}{z}_a^{(m_1)}(t) = z_a(t) \vee z_a(t-1) \vee \dots \vee z_a(t-m_1+1)$, l_2, m_1 分别为 X_a 和 Z_a 的时间范围, d_2 表示 X_a 和 Z_a 之间的时间延迟。 Z_a 和 Y_a 之间的传递熵为

$$NT_{Z_a \rightarrow Y_a} = 1 - \frac{H(y_a(t+1) | \overset{\circ}{y}_a^{(k_2)}(t), \overset{\circ}{z}_a^{(m_2)}(t-d_2))}{H(y_a(t+1) | \overset{\circ}{y}_a^{(k_2)}(t))}. \quad (5.15)$$

其中 k_2, m_2 分别为 Y_a 和 Z_a 的时间范围, d_3 表示 Z_a 和 Y_a 之间的时间延迟。在此基础上, 可以得到数字量 X_a 与 Y_a 基于 Z_a 的直接传递熵

$$D_{X_a \rightarrow Y_a | Z_a} = H(y_a(t+1) | \overset{\circ}{y}_a^{(k')}(t), \overset{\circ}{z}_a^{(m_2)}(t-d_3)) - H(y_a(t+1) | \overset{\circ}{y}_a^{(k')}(t), \overset{\circ}{z}_a^{(m_2)}(t-d_3), \overset{\circ}{x}_a^{(l_1)}(t-d_1)), \quad (5.16)$$

其中 $k' = k_1 = k_2$,

$$H(y_a(t+1) | \overset{\circ}{y}_a^{(k')}(t), \overset{\circ}{z}_a^{(m_2)}(t-d_3)) = \sum_{y_a(t+1), \overset{\circ}{y}_a^{(k')}(t), \overset{\circ}{z}_a^{(m_2)}(t-d_3) \in H^3} p(y_a(t+1), \overset{\circ}{y}_a^{(k')}(t), \overset{\circ}{z}_a^{(m_2)}(t-d_3)) \cdot \log_2 \frac{1}{P(y_a(t+1) | \overset{\circ}{y}_a^{(k')}(t), \overset{\circ}{z}_a^{(m_2)}(t-d_3))}. \quad (5.17)$$

$$\begin{aligned}
H & \left(y_a(t+1) \mid \overset{\circ}{y}_a^{(k')}(t), \overset{\circ}{z}_a^{(m_2)}(t-d_3), \overset{\circ}{x}_a^{(l_1)}(t-d_1) \right) = \\
& \sum_{y_a(t+1), \overset{\circ}{y}_a^{(k')}(t), \overset{\circ}{z}_a^{(m_2)}(t-d_3), \overset{\circ}{x}_a^{(l_1)}(t-d_1) \in H^4} p(y_a(t+1), \overset{\circ}{y}_a^{(k')}(t), \overset{\circ}{z}_a^{(m_2)}(t-d_3), \overset{\circ}{x}_a^{(l_1)}(t-d_1)) \\
& \cdot \log_2 \frac{1}{P \left(y_a(t+1) \mid \overset{\circ}{y}_a^{(k')}(t), \overset{\circ}{z}_a^{(m_2)}(t-d_3), \overset{\circ}{x}_a^{(l_1)}(t-d_1) \right)}. \tag{5.18}
\end{aligned}$$

因此，基于式(5.16)得到 X_a , Y_a 基于 Z_a 的标准化直接传递熵计算公式为

$$\begin{aligned}
ND_{X_a \rightarrow Y_a | Z_a} &= \frac{D_{X_a \rightarrow Y_a | Z_a}}{H \left(y_a(t+1) \mid \overset{\circ}{y}_a^{(k')}(t), \overset{\circ}{z}_a^{(m_2)}(t-d_3) \right)} \\
&= 1 - \frac{H \left(y_a(t+1) \mid \overset{\circ}{y}_a^{(k')}(t), \overset{\circ}{z}_a^{(m_2)}(t-d_3), \overset{\circ}{x}_a^{(l_1)}(t-d_1) \right)}{H \left(y_a(t+1) \mid \overset{\circ}{y}_a^{(k')}(t), \overset{\circ}{z}_a^{(m_2)}(t-d_3) \right)}. \tag{5.19}
\end{aligned}$$

其中 $ND_{X_a \rightarrow Y_a | Z_a}$ 的取值范围为 $[0, 1]$ ，当 $NT_{X_a \rightarrow Y_a}$ 的取值靠近0(1)时，表示数字量之间存在较弱(较强)的直接关联关系。

5.2.3 蒙特卡罗检验

利用显著性检验检测 X_a 与 Y_a 的关联关系，建立原假设 $NT_{X_a \rightarrow Y_a}$ 不显著，即 X_a 与 Y_a 之间不存在关联关系。为了验证原假说，将实际得到的NTE与从无关联关系的代理数字量中计算得到的标准化传递熵显著性阈值进行比较。根据原始给定数字量序列 X_a 和 Y_a ，生成代理数字量序列 X^s 和 Y^s ， X^s 和 Y^s 长度的数字量取值为1的可能性分别与 X_a 和 Y_a 相同。数学上， X^s 服从伯努利分布， Y^s 是模拟生成的序列， X^s 和 Y^s 不存在关联关系，并具有以下特征：

- 1) X^s 的序列长度与 X_a 相同；
- 2) 对于每一个 $x_t^s \in X^s$, x_t^s 的取值为0或1；
- 3) $x_t^s = 1$ 的概率为 q , $q = \frac{M_x}{L}$, M_x 为在数字量序列 X_a 中取值为1的个数, L 为数字量序列 X_a 的长度。

从 X^s 到 Y^s 的标准传递熵记为 $NT_{X^s \rightarrow Y^s}$ ，代理数字量序列 X^s , Y^s 基于 Z^s 的标准直接传递熵记为 $ND_{X^s \rightarrow Y^s | Z^s}$ 。采用蒙特卡罗模拟检验方法^[4-6]，蒙特卡罗p值计算公式如下：

$$\varphi = \frac{b + 1}{N_s + 1}. \tag{5.20}$$

式中 N_s 为蒙特卡罗模拟检验的模拟次数， b 指代理数字量序列计算出的NTE大于原始给定数字量序列NTE的次数，若 $\varphi < \alpha$ ，则在 α 显著性水平上，原假设被拒绝。记一共进行了 N_s 次模拟检验，将第*i*次模拟中得到的 X_a 到 Y_a 的标准传递熵和标准直接传递熵分别记为 $NT_{X^s \rightarrow Y^s}^s(i)$, $ND_{X^s \rightarrow Y^s | Z^s}^s(i)$ 。在每次模拟中都能得到不同的标准传递熵和标准直接传递

熵，取 N_s 次模拟中标准传递熵和标准直接传递熵的最大值，记为标准传递熵和标准直接传递熵的显著性阈值 r_T 、 r_D 。为了使显著性检验更加稳固，对上述蒙特卡罗检验过程重复 K 次，对 K 次得到的显著性阈值取均值，得到

$$\gamma_T = \frac{1}{K} \sum_{k=1}^K \gamma_T(k), \quad (5.21)$$

$$\gamma_D = \frac{1}{K} \sum_{k=1}^K \gamma_D(k). \quad (5.22)$$

其中

$$\gamma_T(k) = \max_{i=1,2,\dots,N_s} NT_{X^s \rightarrow Y^s}^s(i), \quad (5.23)$$

$$\gamma_D(k) = \max_{i=1,2,\dots,N_s} ND_{X^s \rightarrow Y^s | Z^s}^s(i). \quad (5.24)$$

如果 $NT_{X \rightarrow Y} \geq \gamma_T(k)$ ，则原假设不成立，即 X_a 与 Y_a 之间存在关联关系，若有 $ND_{X \rightarrow Y | Z} \geq \gamma_D(k)$ ，则 X_a 与 Y_a 之间存在基于变量 Z_a 的关联关系。

蒙特卡罗检验重复次数 K 的取值对阈值大小存在影响，通过数值案例计算 K 的合适取值。两个独立的二进制报警序列 X_a 与 Y_a ，数据长度为 L 服从伯努利分布 $\mathcal{B}(1, q)$ ，其中 q 是报警发生的概率， X_a 与 Y_a 不存在关联关系。首先，以 $q = 0.015$ 和 $L = 10000$ 为例，说明蒙特卡罗检验次数 K 的大小对标准传递熵显著性阈值的影响。图5.2(a)展现了1000次蒙特卡罗模拟的标准传递熵显著性阈值，图5.2(b)给出了 γ_T 随 K 的变化趋势。可以发现，当 $K \geq 15$ 时， γ_T 的变化变得平滑，因此 $K = 15$ 是一个合理的选择。

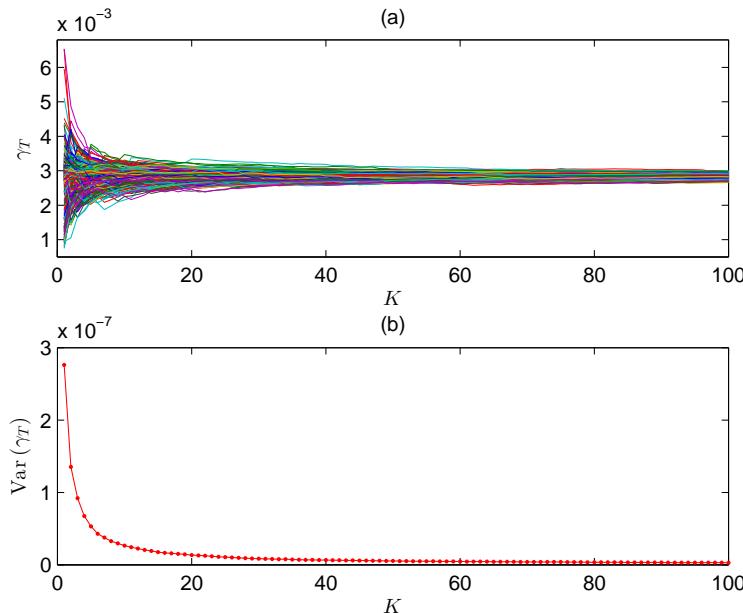


图 5.2 标准传递熵的显著性阈值 γ_T 随 K 的变化趋势。

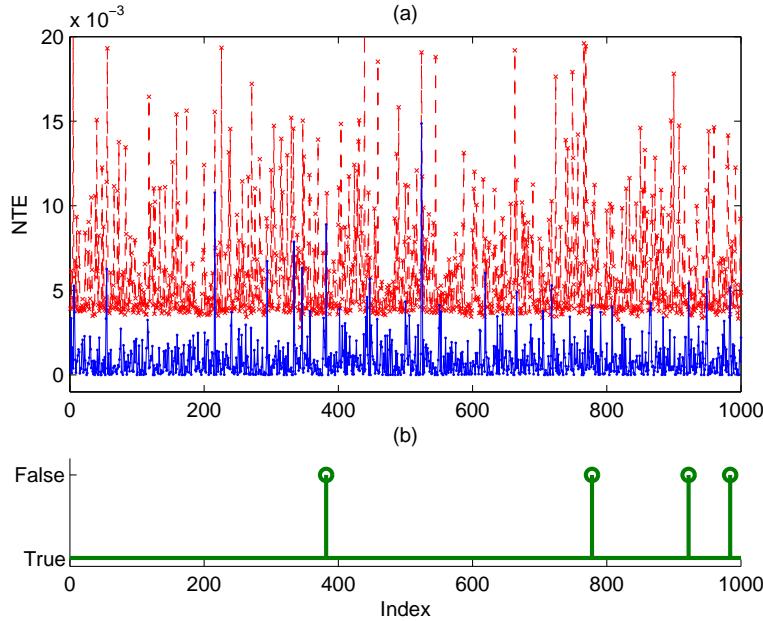


图 5.3 无因果关系情况下的1000次蒙特卡罗检验结果

接着，对时间序列长度 $L = 10000$ 和 q 服从 $q \sim \mathcal{U}(0.005, 0.05)$ 分布 (\mathcal{U} 是均匀分布) 的1000对报警序列进行蒙特卡罗检验得到 γ_T 和 $NT_{X \rightarrow Y}$ ，如图5.3(a)所示。此案例中只存在4组 $NT_{X \rightarrow Y} > \gamma_T$ ，如图5.3(b)所示，检测准确率为99.6%。对标准传递熵进行观察也有相似的结果，由此验证了式(5.21)和式(5.22)计算方法的有效性。

在利用传递熵进行报警根源分析时，对报警变量出现报警次数存在要求。通过一个数值案例，确定最低报警次数 N 要求。建立两个有关联关系的二进制报警序列 X_a 与 Y_a ，其中 X_a 数据长度为 L ，报警发生的概率为 q ，服从伯努利分布 $\mathcal{B}(1, q)$ 。记在 X_a 中第 i 次报警发生的时间戳为 $s_i^{x_a}$ ，对应 Y_a 中报警发生的时间戳为

$$s_i^{y_a} = s_i^{x_a} + \lambda_i. \quad (5.25)$$

其中 λ_i 为时间延迟，服从高斯分布 $\lambda_i \sim \mathcal{N}(v_m, v_s)$ 。图5.4(a)(c)展示了 X_a 与 Y_a 二进制报警序列，其中 $L = 20000$, $q = 0.005$, $v_m = 10$, $v_s = 1$ 。图5.4(b)(d)给出了时间延迟大小及其直方图。令 q 取不同值生成更多的报警变量序列，对于不同报警次数 $N \in \{N : N = 5 + 5j, j = 1, 2, \dots, 39\}$ ，NTE 的平均值计算为 $\overline{NT}_{X_S \rightarrow Y_S} = \sum_{i=1}^{M_w-N+1} NT_{X_S \rightarrow Y_S}^{(i)}$ 。当 $q \in \{q : q = 0.005j, j = 1, 2, \dots, 10\}$ ， $\overline{NT}_{X_S \rightarrow Y_S}$ 随报警次数 N 的变化趋势，如图5.5(a)所示。令 $\Delta \overline{NT}(N) = \overline{NT}_{X_S \rightarrow Y_S}(N) - \overline{NT}_{X_S \rightarrow Y_S}(N-5)$ ， $\Delta \overline{NT}(N)$ 随报警次数 $N = 15, 20, \dots, 200$ 的变化趋势，如图5.5(b)所示。从图中可以看出，当 $N > 50$ 时， $\overline{NT}_{X_S \rightarrow Y_S}$ 的变化趋势不再明显。因此，选择最小报警次数为 $N = 50$ ，这些报警事件可以分布在较长的时间段内，而不必在短时间内发生。

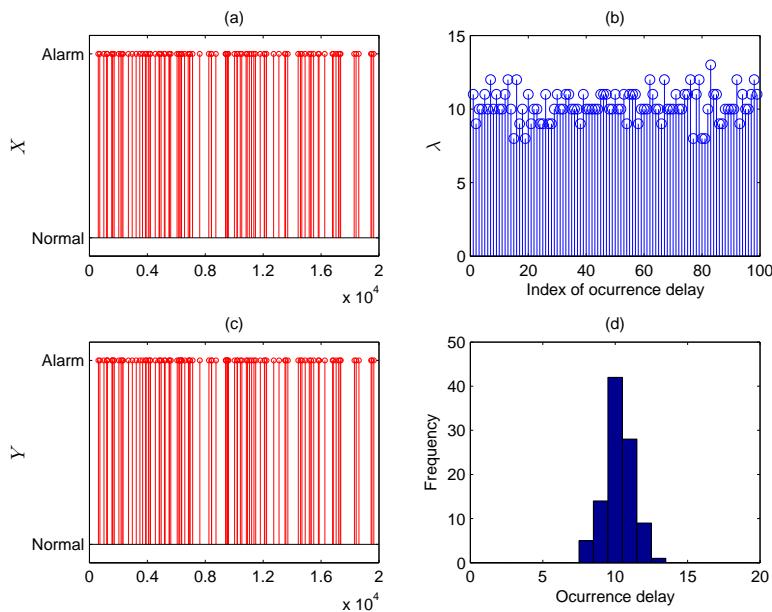


图 5.4 报警序列和随机时间延迟的示例

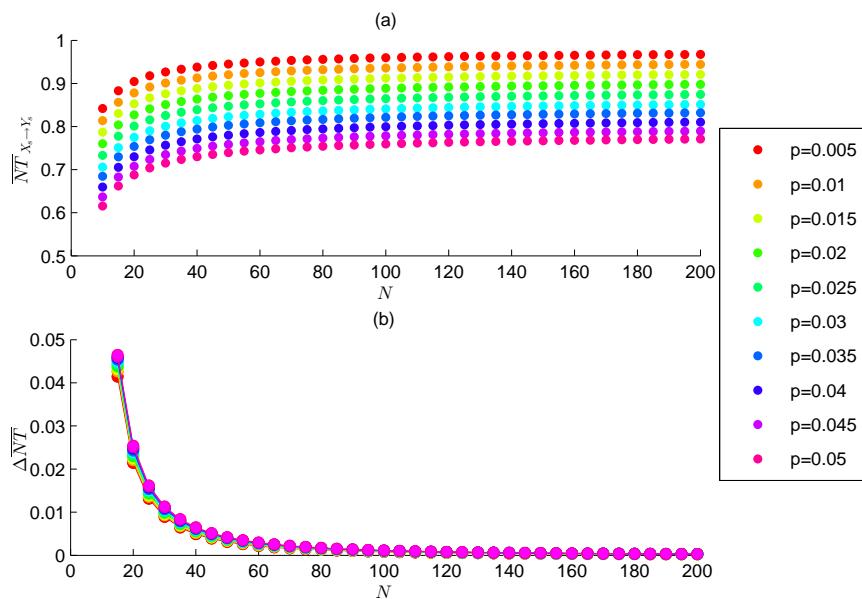


图 5.5 不同q下的平均标准传递熵和报警发生次数

5.2.4 详细步骤

在利用传递熵进行报警变量根源分析时，报警变量必须满足以下条件：①每个报警序列中至少应出现50次报警；②式(5.8)和(5.9)中的时间范围 l_1 和 k_1 是提前确定的，确定方法在5.2.1节有详细介绍。该方法可以分为两个部分，第一部分为标准传递熵的计算，其计算步骤如下：

- 1) 已知报警变量时间序列为 $X_K, K = 1, 2, \dots, M$ ；
- 2) 通过标准传递熵公式可知，不同时间延迟 d 下标准传递熵不同，根据式(5.10)计算不同时间延迟下报警变量 X_i 和 X_j , ($i \neq j$)的标准传递熵 $NT_{X_i \rightarrow X_j}(d)$ ，记 X_i 和 X_j 之间的标准传递熵为 $NT_{X_i \rightarrow X_j} = \max_d NT_{X_i \rightarrow X_j}(d)$ ，此时对应的时间延迟为 $d_{ij} = \arg \max_d NT_{X_i \rightarrow X_j}(d)$ ；
- 3) 用蒙特卡罗检验计算出标准传递熵的显著性阈值 γ_T ；
- 4) 如果 $NT_{X_i \rightarrow X_j} > \gamma_T$ ，则 $C_{ij} = 1$ ，否则 $C_{ij} = 0$ ；
- 5) 如果已计算出所有成对报警序列的标准传递熵，进入下一步；否则通过增加 i 或 j ，对另一对报警序列进行步骤2) 操作；
- 6) 根据 C_{ij} 得到表示信息流路径的关联图。如果 $C_{ij} = 1$ ，则画一条带箭头的线来表示 X_i, X_j 之间的关联关系；否则 X_i, X_j 之间没有关联关系。

该方法的第二部分是在标准直接传递熵的基础上去除关联图中多余的连接线。该部分由以下步骤组成，具体计算流程，如图5.6的流程图所示。

- 1) 从 $i = 1, j = 1$ 开始计算 X_i, X_j 基于 X_l 的标准直接传递熵；
- 2) 若之前计算的 $C_{ij} = 1$ ，计算进入下一步，否则进入步骤8)；
- 3) 从中间变量 $X_l, l = 1$ 开始计算；
- 4) 如果 $C_{il} = 1, C_{lj} = 1$ ，计算进入到下一步，否则转到步骤6)；
- 5) 根据式(5.19)计算出 $ND_{X_i \rightarrow X_j | X_l}$ ，公式中的时间间隔 d_{ij}, d_{lj} 分别取 $NT_{X_i \rightarrow X_j}(d), NT_{X_l \rightarrow X_j}(d)$ 最大值所对应的时间间隔；
- 6) 如果从 X_i 到 X_j 的标准直接传递熵已经计算了所有中间变量，则继续进行下一步；否则，通过将 i 增加1返回到步骤5)；
- 7) 通过取 X_i 到 X_j 不同中间变量的标准直接传递熵最大值，记为 ND_{max} ，将 ND_{max} 与通过式(5.22)计算出的 γ_D 相比较。若满足 $ND_{max} > \gamma_D$ ，则 X_i 与 X_j 之间存在直接的关联关系，否则 X_i 与 X_j 的关联关系是间接的， X_i 到 X_j 的直接连接线应被排除在关联图外；
- 8) 如果已计算出所有成对报警序列的标准直接传递熵，则停止计算；否则，通过将 i 或 j 增加1，返回到步骤2)。

5.2.5 案例分析

本节通过一个数值例子和一个工业实例具体阐述了如何基于传递熵进行报警变量根源分析，并验证了所提方法的有效性。

案例1：三个报警变量的因果关系如下所示： $x_1 \rightarrow x_2 \rightarrow x_3$ ，记在 x_1 中第*i*次报警发生的时

间戳为 s_i^1 , 对应 x_2, x_3 中报警发生的时间戳为:

$$\begin{cases} s_i^2 = s_i^1 + \lambda_i^{(1)} \\ s_i^3 = s_i^2 + \lambda_i^{(2)} \end{cases} \quad (5.26)$$

其中 $\lambda_i^{(1)}, \lambda_i^{(2)}$ 为时间延迟, 服从平均值20和标准差1的高斯分布。首先, 生成模拟报警数据, 其中序列长度为 $L = 20000$, 报警次数为100。图5.7展示了0到100之间不同时间延迟下的标准传递熵, 最终得到标准传递熵以及其相应的显著性阈值(括号内)见表5.1。图5.8(a)中的关联图描述了三个报警变量之间的关联关系。接着, 检验 x_1, x_3 之间是存在直接关联关系, 还是通过 x_2 存在关联关系。计算出标准直接传递熵 $ND_{x_1 \rightarrow x_3|x_2} = 0.0037$, 通过蒙特卡罗检验获得标准传递熵的阈值为 $\gamma_D = 0.025$ 。因为 $ND_{x_1 \rightarrow x_3|x_2} < \gamma_D$, 则 x_1, x_3 之间的关联关系是间接的, 排除从 x_1 到 x_3 的信息流路径, 绘制新的关联关系图, 如图5.8(b)所示, 这与实际关系一致。

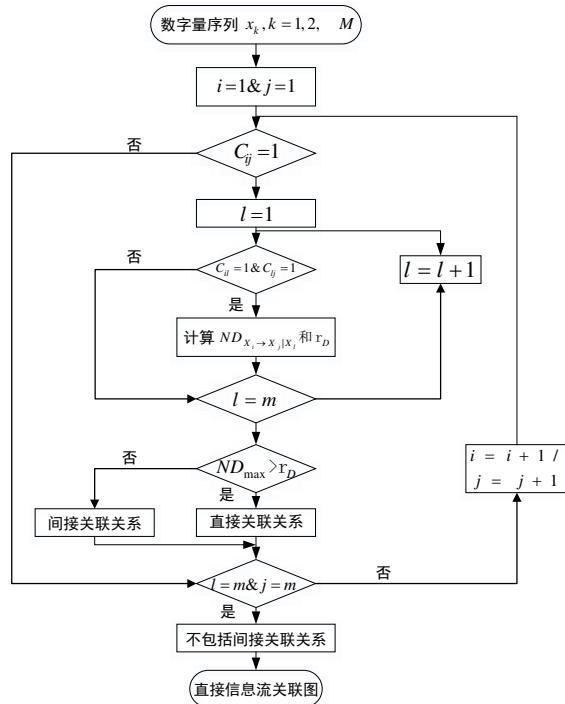


图 5.6 基于标准直接传递熵的直接关联关系检测流程图

表 5.1 标准传递熵以及显著性阈值

$T_{x_i \rightarrow x_j}$	x_1	x_2	x_3
x_1		0.73(0.022)	0.62(0.023)
x_2	0.109(0.112)		0.78(0.025)
x_3	0.110(0.111)	0.110(0.111)	

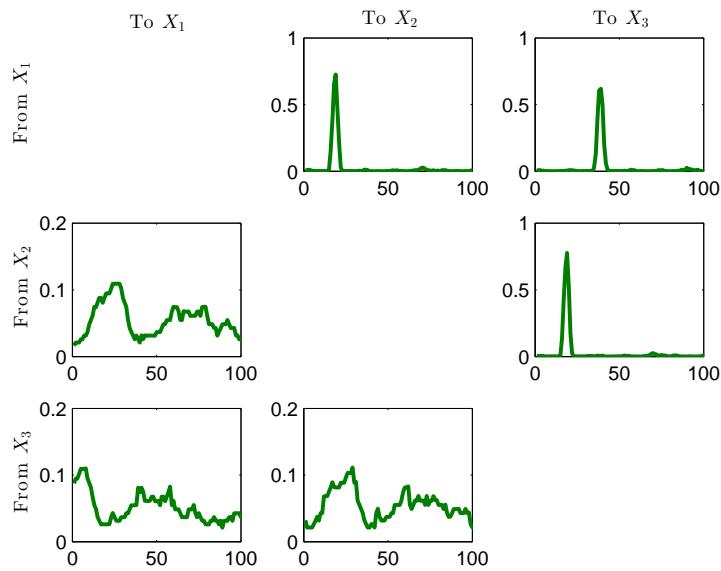


图 5.7 不同时间延迟下的标准传递熵

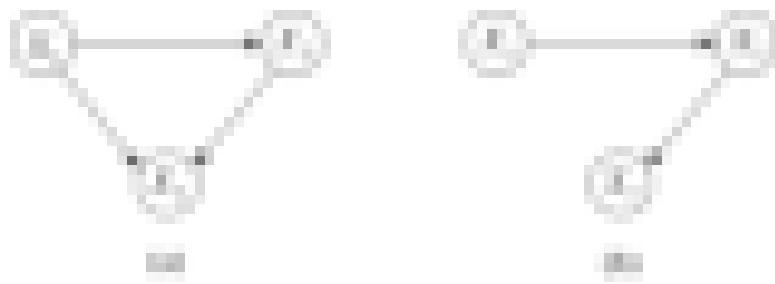


图 5.8 关联关系图

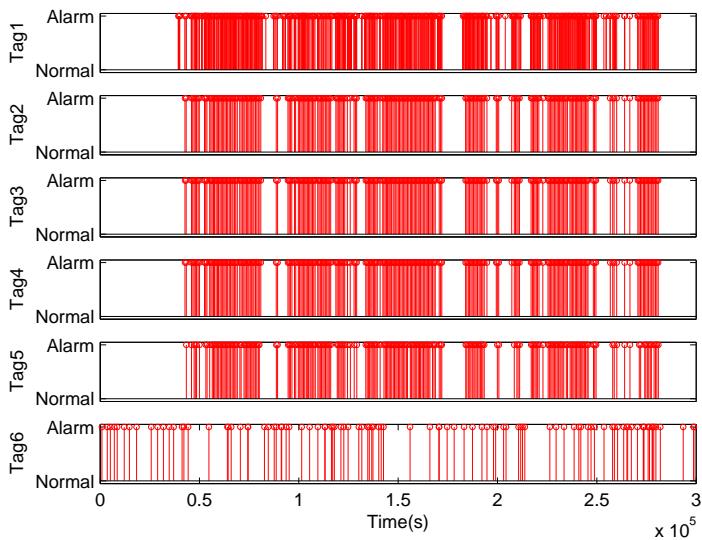


图 5.9 3.5天内报警变量的时序图

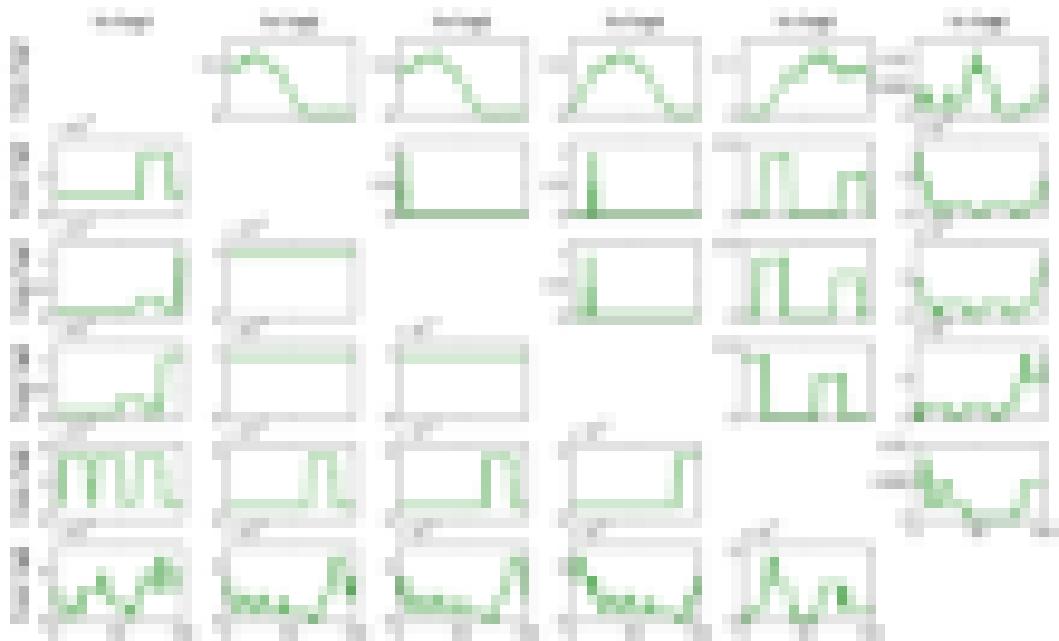


图 5.10 标准传递熵随时间延迟的变化

案例2: 加拿大的某家石油工业工厂内有五千多个报警变量，用于监控过程变量变化、操作和安全条件。在3.5天的时间内288个报警变量发生过报警，所提出的方法适用于这些报警变量。选取了6个报警变量，并将其记为 $Tag1 - 6$ ，用以证明所提方法的有效性。6个报警变量的历史数据样本，如图5.9 所示。在3.5天的时间段内，6个报警变量的报警次数分别

为653、242、242、242、219和106。通过计算抖振指数^[5]，发现报警变量Tag1中包含抖动报警。使用30秒关闭延迟定时器，大多数抖动警报被消除，报警发生的次数减少到282次。根据第5.2.3节可知，6个报警变量的报警次数足以达到传递熵的可靠估计。

首先，用该方法计算图5.10中报警变量之间的标准传递熵。图5.10给出了不同时间延迟下的传递熵，最终得到的标准传递熵及其相应的显著性阈值（括号内）见表5.2。例如，Tag1到Tag2的标准传递熵为0.2577，显著性阈值为0.0155，因此Tag1到Tag2之间存在关联关系。在图5.11中绘制描述信息流路径的关联图。

其次，根据标准传递熵大于其相应阈值的每对报警变量，计算其标准直接传递熵。以Tag1和Tag5为例，两者可能通过Tag2，Tag3，Tag4存在关联关系。因此，表5.3计算出Tag1到Tag5基于Tag2，Tag3，Tag4的标准直接传递熵。从表5.3可以看出报警变量之间是否存在直接关联关系。从图5.11的关联图中排除间接关联关系，最终得到所有直接信息流路径的关联图，如图5.12所示。

表 5.2 标准传递熵以及显著性阈值

	Tag1	Tag2	Tag3
Tag1		0.2577(0.0155)	0.2577(0.0153)
Tag2	0.0003(0.0127)		1(0.0011)
Tag3	0.0011(0.0134)	0.0001(0.0135)	
Tag4	0.0011(0.0130)	0.0001(0.0139)	0.0001(0.0140)
Tag5	0.0004(0.0122)	0.0004(0.0131)	0.0005(0.0131)
Tag6	0.0065(0.0090)	0.0058(0.0089)	0.0058(0.0087)
	Tag4	Tag5	Tag6
Tag1	0.2577(0.0155)	0.1293(0.0151)	0.0115(0.0215)
Tag2	1(0.0013)	0.4336(0.0142)	0.0082(0.0188)
Tag3	1(0.0011)	0.4336(0.0144)	0.0082(0.0183)
Tag4		0.4336(0.0135)	0.0082(0.0204)
Tag5	0.0005(0.0133)		0.0085(0.0167)
Tag6	0.0058(0.0089)	0.0045(0.0102)	

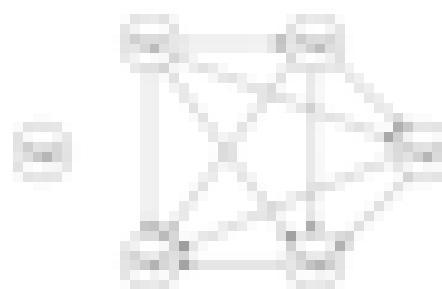


图 5.11 信息流路径关联图

表 5.3 基于标准直接传递熵的直接关联关系检测

原因变量	结果变量	中间变量	标准直接传递熵	显著性阈值
Tag1	Tag3	Tag2	0	0.0129
Tag1	Tag4	Tag2, Tag3	0	0.0122
Tag1	Tag5	Tag2, Tag3, Tag4	0.00014	0.0118
Tag2	Tag4	Tag3	0	0.0005
Tag2	Tag5	Tag3, Tag4	0	0.0149
Tag3	Tag5	Tag4	0	0.0152

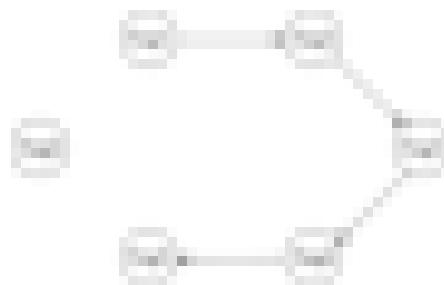


图 5.12 直接信息流路径的关联图

最后，图5.12所示的关联关系与6个报警变量的物理关联关系一致。第6个报警变量位于断开的单元中，前5个报警变量位于具有物理连接的单元中：*Tag2*代表燃烧器中的总气体流量，该气体流量来自进料室A和进料室B的两个供应商。*Tag1*表示来自进料室B的气体流量。*Tag3, 4, 5*与燃烧器的三种器件相关。因此，信息流路径为*Tag1* → *Tag2* → *Tag3* → *Tag4* → *Tag5*。表5.2计算得到的标准传递熵提供了有价值的信息。从*Tag2*到*Tag3*、从*Tag2*到*Tag4*、从*Tag3*到*Tag4*的标准传递熵一致，表示3个报警变量之间有很强的关联关系。相比之下，*Tag1*与其他报警变量的标准传递熵相对较小。因此，进料室B异常，使*Tag1*出现报警，导致随后报警变量的异常可能性为25%。从关联关系中，*Tag1*和*Tag2*被标识为主要报警变量，而*Tag3, Tag4*和*Tag5*是相应的次要报警变量。因此，当报警信号发生时，引起异常的根本原因，其搜索空间可以迅速被缩小到与输入气流相关的变量。

5.3 基于贝叶斯网络的报警根源分析

报警变量 x_a 可用二进制‘1’和‘0’表示，其分别对应报警状态与非报警状态。假设 $x_a = 1$ 可由 n 个相互独立的根本原因变量状态变化引起，用 x_1, x_2, \dots, x_n 表示 n 个根本原因变量。将根本原因变量用二进制表示， $x_i = 1$ （或者 $x_i = 0$ ）， $i \in [1, n]$ 代表第*i*个根本原因变量是有效（无效）的。如果 $x_i = 1$ ，将会导致报警变量 x_a 的发生。那么，给定观测数据集 $D := \{x(1), \dots, x(t-1), x(t)\}$ ，其中 $x(t) := [x_1(t), x_2(t), \dots, x_n(t), x_a(t)]^T$ ，目的是在上述假设下，分析 $x_a(t) = 1$ 发生报警的根本原因。

实现这一目标的一种常见做法是通过查看数据样本来手动分析根本原因，即如果 $x_a(t) = 1$ ，则导致报警的根本原因变量为 $x_i(t) = 1, i \in [1, n]$ 。然而，由于以下三个原因，在实践中往

往往会产生不准确的结果：第一，由于随机噪声的影响， x_a 或者 x_i 可能在短时间内取值为1，此时， x_a 或者 x_i 会产生假警报，导致根本原因分析失败。第二，可能同时存在多个根本原因，工业工程师手动分析根本原因时，倾向于关注最初出现的根本原因，而忽视其他共同存在的根本原因。第三，根本原因变量 x_1, x_2, \dots, x_n 可能不完整，根本原因变量不完整的一个常见原因是工业装置控制中心无法通过测量得到其他根本原因变量，而工业工程师必须在现场对这些根本原因进行验证。因此，这些根本原因变量不在集合中 $\{x_1, x_2, \dots, x_n\}$ ，它们不能由常规做法处理。针对这三个挑战，提出一种基于贝叶斯网络的报警根源分析方法，该方法的主要思想是通过确定贝叶斯网络参数的更新概率来分析报警的根本原因。

5.3.1 贝叶斯网络参数的更新概率

图5.13所示的贝叶斯网络用于描述报警变量 x_a 与根本原因变量 x_1, x_2, \dots, x_n 之间的关系。它是一个图形模型，通过有向无环图表示一组随机变量及其条件的依赖关系。随机变量由网络中的节点表示，并取一些离散值^[8]。对于图中的贝叶斯网络， x_1, x_2, \dots, x_n 是父节点， x_a 为子节点。在这种情况下，它们采用二进制值‘0’和‘1’， $\theta_{i,0}, \theta_{i,1}$ 分别表示 $x_i, i \in [1, n]$ 取值为‘0’和‘1’的概率，

$$\theta_{i,0} := P(x_i = 0), \theta_{i,1} := P(x_i = 1), i \in [1, n].$$

令 $R := \{x_1, x_2, \dots, x_n\}$ 是 x_a 的父节点集，且 $r_j, j \in [1, 2^n]$ 是 x_a 的父节点可能取值的第 j 个子集。例如， $n = 2$ ，那么 R 有四子集，分别是 $r_1 := \{0, 0\}$ ， $r_2 := \{1, 0\}$ ， $r_3 := \{0, 1\}$ ， $r_4 := \{1, 1\}$ 。参数 $\theta_{a,0|j}$ 和 $\theta_{a,1|j}$ 分别表示在 R 取 r_j 值的条件下， $x_a = 0$ 和 $x_a = 1$ 的先验条件概率，即：

$$\theta_{a,0|j} := P(x_a = 0|R = r_j), \theta_{a,1|j} := P(x_a = 1|R = r_j), j \in [1, 2^n].$$

并且这些参数都满足约束条件： $\theta_{i,0} + \theta_{i,1} = 1, \forall i \in [1, n], \theta_{a,0|j} + \theta_{a,1|j} = 1, \forall j \in [1, 2^n]$ 。

为了方便进行在线根源分析，设定贝叶斯网络中由观测样本和递归方式更新的概率参数集为 $\boldsymbol{\theta} := \{\theta_{i,0}, \theta_{i,1}, \theta_{a,0|j}, \theta_{a,1|j}\}$ ，其中 $i = 1, 2, \dots, n, j = 1, 2, \dots, 2^n$ 。更精确地说，概率参数集 $\boldsymbol{\theta}$ 需要根据先验概率参数集 $\bar{\boldsymbol{\theta}}$ 和观测数据样本 x_1, x_2, \dots, x_n 和 x_a 进行更新。Bauer等人提出了一种适用于历史数据集 $D := \{x(1), \dots, x(t-1), x(t)\}$ ， $x(t) := [x_1(t), x_2(t), \dots, x_n(t), x_a(t)]^T$ 的批量学习算法^[9]，其中，后验估计 $\hat{\boldsymbol{\theta}}$ 可由最大值化函数 $F(\boldsymbol{\theta}) = \lambda L_D(\boldsymbol{\theta}) - d(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}})$ 计算得到，即

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} (\lambda L_D(\boldsymbol{\theta}) - d(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}})), \quad (5.27)$$

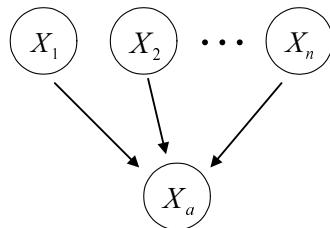


图 5.13 贝叶斯网络模型图

其中 $L_D(\boldsymbol{\theta})$ 是平均似然对数函数

$$L_D(\boldsymbol{\theta}) = \frac{1}{t} \sum_{l=1}^t \ln P_{\boldsymbol{\theta}}(x(l)).$$

这里, 设 $x(l_1)$ 和 $x(l_2)$ 对于所有的 $l_1 \neq l_2$ 相互独立, 惩罚函数 $d(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}})$ 是用来计算 $\boldsymbol{\theta}$ 与 $\bar{\boldsymbol{\theta}}$ 之间的 χ^2 距离, λ 是参数更新速率。Cohen 等人基于式(5.27)中的处理方法, 进一步提出了一种在线更新算法^[10]。最终历史数据集 D 由当前样本 $X(t)$ 替代, $\bar{\boldsymbol{\theta}}(t-1)$ 和 $\hat{\boldsymbol{\theta}}(t)$ 分别替代 $\bar{\boldsymbol{\theta}}$ 和 $\hat{\boldsymbol{\theta}}$ 。其中, $\bar{\boldsymbol{\theta}}$ 表示先验估计; $\hat{\boldsymbol{\theta}}$ 表示采用批量学习算法得到的后验估计; $\hat{\boldsymbol{\theta}}(t-1)$ 表示前一时刻的后验估计; $\hat{\boldsymbol{\theta}}(t)$ 表示当前时刻的后验估计。

将在线更新算法的主要思想用于更新贝叶斯网络中概率参数。先验估计 $\bar{\boldsymbol{\theta}}$ 可由当前数据样本 $x(t) := [x_1(t), x_2(t), \dots, x_n(t), x_a(t)]^T$ 和 $\hat{\boldsymbol{\theta}}(t-1)$ 得到, 假设 $x_a(t)$ 取值为 1, 并且 $R(t)$ (数据样本 $x(t)$ 的前 n 个) 取第 j 个父节点集 r_j , 即 $P(x_a(t) = 1, R(t) = r_j | x(t)) = 1$, $P(x_a(t) = 1 | x(t)) = 1$, 用当前样本 $x(t)$ 替代批处理中的历史数据集 D , 得到

$$\begin{aligned} \theta_{a,1|j}(t) &= \lambda \frac{P(x_a(t) = 1, R(t) = r_j | x(t))}{P(x_a(t) = 1 | x(t))} + (1 - \lambda) \theta_{a,1|j}(t-1) \\ &= \lambda + (1 - \lambda) \theta_{a,1|j}(t-1). \end{aligned} \quad (5.28)$$

同时, 为了满足等式 $\theta_{a,1|j}(t) + \theta_{a,0|j}(t) = 1$, 参数 $\theta_{a,0|j}(t)$ 更新为

$$\theta_{a,0|j}(t) = (1 - \lambda) \theta_{a,0|j}(t-1). \quad (5.29)$$

对于父节点集 r_k ($k \in [1, 2^n], k \neq j$), 其中相应的概率参数不变, 即

$$\theta_{a,1|k}(t) = \theta_{a,1|k}(t-1), \quad \theta_{a,0|k}(t) = \theta_{a,0|k}(t-1). \quad (5.30)$$

当 $x(t)$ 中 $x_i(t) = 1$, $i \in [1, n]$ 时, 即 $P(x_i(t) = 1 | x(t)) = 1$ 时相应的概率参数更新为

$$\begin{aligned} \theta_{i,1}(t) &= \lambda P(x_i(t) = 1 | x(t)) + (1 - \lambda) \theta_{i,1}(t-1) \\ &= \lambda + (1 - \lambda) \theta_{i,1}(t), \end{aligned} \quad (5.31)$$

和

$$\theta_{i,0}(t) = (1 - \lambda) \theta_{i,0}(t-1). \quad (5.32)$$

同样, 当 $x_a(t) = 0$, $R(t) = r_j$ 时, 式(5.28), (5.29)对应的概率参数可以更新为

$$\theta_{a,0|j}(t) = \lambda + (1 - \lambda) \theta_{a,0|j}(t-1), \quad (5.33)$$

$$\theta_{a,1|j}(t) = (1 - \lambda) \theta_{a,1|j}(t). \quad (5.34)$$

当 $x_i(t) = 0$ 时, 式(5.31), (5.32)相对应的概率参数可以更新为

$$\theta_{i,0}(t) = \lambda + (1 - \lambda) \theta_{i,0}(t-1), \quad (5.35)$$

$$\theta_{i,1}(t) = (1 - \lambda) \theta_{i,1}(t-1). \quad (5.36)$$

因此, 概率参数集在线更新为

$$\boldsymbol{\theta}(t) = \{\theta_{i,0}(t), \theta_{i,1}(t), \theta_{a,0|j}(t), \theta_{a,1|j}(t)\}, i = 1, 2, \dots, n, j = 1, 2, \dots, 2^n. \quad (5.37)$$

5.3.2 分析报警发生的根本原因

将 x_a 的值表示为 X_a , x_i 在第 j 个父节点集的值表示为 $X_{i,j}$ 。根据贝叶斯规则, 得到基于 X_a 的后验条件概率 r_j , 即

$$\begin{aligned} P(R = r_j | x_a = X_a) &= \frac{P(R = r_j, x_a = X_a)}{P(x_a = X_a)} \\ &= \frac{P(x_1 = X_{1,j}, \dots, x_n = X_{n,j}, x_a = X_a)}{\sum_j P(x_1 = X_{1,j}, \dots, x_n = X_{n,j}, x_a = X_a)}, \end{aligned} \quad (5.38)$$

根据链式规则以及 x_1, \dots, x_n 之间的相互独立性, 联合概率可以表示为

$$\begin{aligned} &P(x_1 = X_{1,j}, \dots, x_n = X_{n,j}, x_a = X_a) \\ &= P(x_a = X_a | x_1 = X_{1,j}, \dots, x_n = X_{n,j}) \\ &\quad \cdot P(x_1 = X_{1,j} | x_2 = X_{2,j}, \dots, x_n = X_{n,j}) \\ &\quad \cdots P(x_{n-1} = X_{n-1,j} | x_n = X_{n,j}) \cdot P(x_n = X_{n,j}) \\ &= \prod_{i=1}^n P(x_i = X_{i,j}) \cdot P(x_a = X_a | R = r_j). \end{aligned} \quad (5.39)$$

使用式(5.39)得到的联合概率以及从式(5.37)得到的更新概率参数 $\theta(t)$, 将式(5.38)的后验概率参数 $P(R = r_j | x_a = X_a)$ 记为 $\theta_{j|x_a}$, 并以在线的方式更新为

$$\theta_{j|x_a}(t) = \frac{\prod_{i=1}^n \theta_{i,X_{i,j}}(t) \cdot \theta_{a,X_a|j}(t)}{\sum_j \prod_{i=1}^n \theta_{i,X_{i,j}}(t) \cdot \theta_{a,X_a|j}(t)}.$$

此外, 为了不失一般性, 以 r_{2^n} 表示父节点集 $\{x_1(t) = 0, \dots, x_n(t) = 0\}$ 。当 x_a 处于非报警状态时, 并不需要分析报警的根本原因。将 $\theta_{j|x_a}(t)$ 的分子表示为 $\theta_{j|x_a}^{(N)}(t)$, 当 $x_a = 1$ 时将后验条件概率定义为一个向量:

$$\boldsymbol{\theta}_p := \left[\theta_{j=1|X_a=1}^{(N)}(t), \theta_{j=2|X_a=1}^{(N)}(t), \dots, \theta_{j=2^n|X_a=1}^{(N)}(t) \right].$$

将 $\boldsymbol{\theta}_p$ 向量以降序排列, 记为 $\boldsymbol{\theta}_{p,s}$:

$$(\boldsymbol{\theta}_{p,s}, \boldsymbol{\eta}) = \max(\boldsymbol{\theta}_p), \quad (5.40)$$

向量 $\boldsymbol{\eta}$ 包含已排列元素的下标, 则根本原因为

$$\hat{j}(t) = \boldsymbol{\eta}(1), \quad (5.41)$$

其中 $\boldsymbol{\eta}(1)$ 表示 $\boldsymbol{\eta}$ 的第一个元素。换句话说, 如果 $\theta_{j|1}(t), j \in [1, 2^n - 1]$ 是最大的后验条件概率, 那么 x_a 处于报警状态时, 根本原因为 r_j 父节点集中 x_1, \dots, x_n 取值为1的变量; 如果最后一个元素 $\theta_{2^n|1}(t)$ 为最大值, 那么可以得知, $x_a = 1$ 是由根源变量 x_1, x_2, \dots, x_n 之外的一些未知原因所导致。

记 $\theta_{a,0}$ 和 $\theta_{a,1}$ 分别为 x_a 取0和1的概率。如果 $x_a(t) = 1$, 那么 $\theta_{a,0}(t)$ 和 $\theta_{a,1}(t)$:

$$\theta_{a,1}(t) = \lambda + (1 - \lambda)\theta_{a,1}(t - 1),$$

$$\theta_{a,0}(t) = (1 - \lambda)\theta_{a,0}(t - 1).$$

当 $x_a(t) = 0$ 时, 对应式(5.35)和(5.36)有

$$\theta_{a,0}(t) = \lambda + (1 - \lambda)\theta_{a,0}(t - 1),$$

$$\theta_{a,1}(t) = (1 - \lambda)\theta_{a,1}(t - 1).$$

如果 $\theta_{a,0}(t) > \theta_{a,1}(t)$, 那么根源分析结果 $\hat{j}(t)$ 将被置零。因此导致报警的根本原因可以表示为

$$\hat{j}(t) = \begin{cases} \eta(1), & \theta_{a,0}(t) \leq \theta_{a,1}(t) \\ 0, & \theta_{a,0}(t) > \theta_{a,1}(t) \end{cases}. \quad (5.42)$$

判断通过上述方法得到根本原因分析结果的准确性。首先, 从Cohen等人的定理可得^[10], 如果 $0 < \lambda < 1$, 则式(5.37)得到的 $\theta(t)$ 的每个估计值都是无偏的, 记为 $\theta(t)$, 即

$$E(\theta(t)) = \theta^*,$$

$$Var(\theta(t)) = \frac{\lambda}{2 - \lambda}\theta^*(1 - \theta^*),$$

其中 θ^* 是 $\theta(t)$ 的真实值。其次, 将图5.13中贝叶斯网络的所有节点取二进制值‘0’和‘1’。由于随机噪声的影响, x_1, \dots, x_n 和 x_a 可能取值‘1’或‘0’, 与其实际处于相反的状态。最后, 错误值在前一种情况下称为误报警, 在后一种情况下称为漏报警。如果误报警和漏报警的概率不大于0.5, 则所提方法能分析出报警的真正根源。

定理1: 对于图5.13中的贝叶斯网络, 假设 $x_a = 1$ 的真正根本原因与 $R = r_{j^*}$ 相关。如果 x_1, \dots, x_n 和 x_a 中误报警和漏报警的概率小于0.5, 那么在足够的时间长度下, 式(5.42)给出了真正的根本原因变量 r_{j^*} 。

定理1的证明 由于存在误报警和漏报警, 可能存在 $\theta_{i,x_{i,j}}(t) \neq 0$, $\theta_{a,x_a|j}(t) \neq 0$, 其中 $i \in [1, n]$ 的情况。但是, 如果误报警和漏报警的概率小于0.5, 且 $x_a = 1$ 是由 $R = r_{j^*}$ 引起的, 则有 $\theta_{i,X_{i,j^*}}(t) \geq 0.5$, $\theta_{i,X_{i,j}}(t) < 0.5$, $\theta_{a,1}(t) \geq 0.5$, $\theta_{a,0}(t) < 0.5$, $\theta_{a,1|j^*}(t) \geq 0.5$ 和 $\theta_{a,1|j}(t) < 0.5$, 其中 $j \neq j^*$ 。当更新后的概率参数处于稳态状态时, 下面的不等式成立:

$$\prod_{i=1}^n \theta_{i,X_{i,j}}(t) \cdot \theta_{a,1|j}(t) < \prod_{i=1}^n \theta_{i,X_{i,j^*}}(t) \cdot \theta_{a,j^*|1}(t).$$

因为分母 $\theta_{1|1}(t), \theta_{2|1}(t), \dots, \theta_{2^n|1}(t)$ 相同, 则式(5.42)的报警根源分析给出了真正的根本原因 $\hat{j}(t) = j^*$ 。证明完成。

最终, 需要设置更新速率参数 λ 。为了达到 $\theta(t)$ 的无偏估计, 更新速率参数需满足 $\lambda \in [0, 1]$ 。当报警变量 x_a 由0转变到1, 其随后的 m 个样本 x_a 也取1, 那么 x_a 的真实状态就由0转变到1, 反之则认为本次报警为误报或者漏报。该方法在实践中被广泛用于消除干扰警报^[6]。工业标准ISA^[6]根据不同类型工艺变量的变化率, 给出了表5.4中 m 的推荐值。当 x_i 在0与1状态之

表 5.4 工业标准中变量相关样本的推荐值

变量类型	m (s)
流速	15
水平面	60
压力	15
温度	60

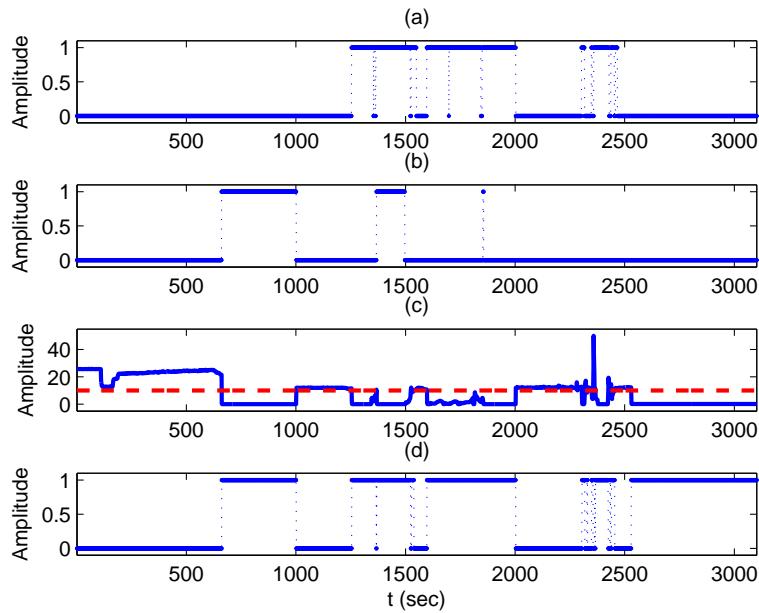


图 5.14 变量的时间序列图

间变化时，参数 $\theta_{i,j}$ 的更新速率需要满足表中的 m 推荐值。第5.3.1节中更新方程可以表示为

$$\theta(t) = \lambda + (1 - \lambda)\theta(t - 1),$$

这是一个一阶差分方程。如果 $\theta(0)$ 表示初始状态，则

$$\theta(t) = 1 - (1 - \lambda)^t(1 - \theta(0)), \quad (5.43)$$

由定理1可知， $\theta(t)$ 必须满足 $\theta(\infty) \geq 0.5$ ，以此来确保获得真正的根本原因变量。初始值 $\theta(0)$ 通常设置为0，并且从 $\theta(0)$ 到接近最终状态 $\theta(\infty)$ 至少需要 m 个样本。 λ 的值可由式(5.43)得出

$$\lambda = 1 - (0.5)^{\frac{1}{m}}. \quad (5.44)$$

5.3.3 案例分析

本节通过工业实例来说明所提出方法的有效性。该实例为中国山东省某台大型300MW火电机组，其发电机组分散式控制系统（DCS）提供了24079个过程变量的实时数据样本，其

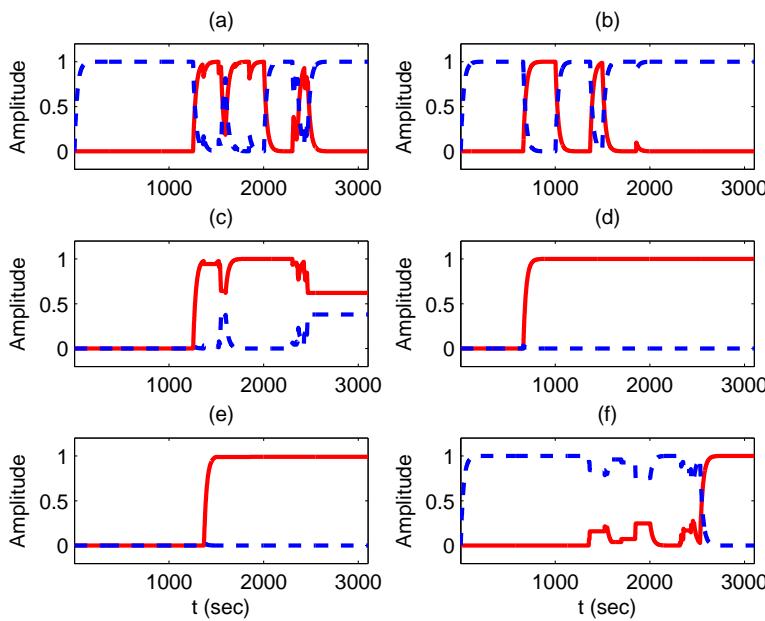


图 5.15 先验概率参数的时序图

中15934个数字过程变量中有2018个变量产生报警，8145个模拟过程变量中有208个变量产生报警。这里所考虑特殊类型的报警变量主要有两种形式：第一，报警变量和根本原因变量存在物理定律（如质量和能量平衡定律）关联，根本原因通常由过程知识可知。第二，通过DCS中与、或和非模块组成的逻辑机制来生成报警变量，可通过读取DCS中的配置图找到此类报警变量。到目前为止，我们已经将该方法应用到2226个报警变量中的211个报警变量。因此，本节提出的方法对这类报警变量发生报警的根本原因进行分析具有实际意义。以下述两个工业报警变量作为这两种特殊类型报警变量的代表进行详细说明。

案例1：给煤机向磨煤机提供原煤，是生产用于燃烧的煤粉的关键设备，给煤机必须保持足够的给煤流量。给煤流量 x_c 配置低报警阈值10t/h，根据工艺知识可知， x_a 的报警状态可能由多种原因引起。为了便于介绍，这里考虑了两个主要原因， x_1 进煤不足和 x_2 电机故障。当给煤机输入的煤量减少到一定程度时，数字传感器将向发电机组控制中心发送‘1’值，给煤机要么停机，要么切换到空转状态，以防止设备损坏。电动机为给煤机提供驱动力。如果电机出现故障，必须关闭给煤机直到故障排除。因此， x_1 和 x_2 本质上是数字变量。一旦 x_a 进入报警状态，就需要准确分析根本原因，采取相应的纠正措施，使给煤机恢复正常运行。否则，由于煤粉短缺，发电量很快就会减少，造成经济损失。

图5.14给出了关于 $x_1(t)$, $x_2(t)$, $x_c(t)$ 和 $x_a(t)$ 的数据样本，采样周期为1s。在某些情况下，例如，当 $x_a(t) = 1$, $t \in [662, 1001]$ ，通过目测可以很容易地看出 $x_2(t)$ 是报警的根本原因。然而，第5.3节讨论的三个问题都是在瞬间出现的，这阻碍了通过视觉进行报警根源分析。将上面提出的方法应用于 $x_1(t)$, $x_2(t)$ 和 $x_a(t)$ ，设定 $m = 20$ ，通过式(5.44)计算更新速率参数为 $\lambda = 0.0341$ 。更新的先验概率参数和后验参数，如图5.15和5.16所示，所有概率参数的初始值都设置为零，根本原因分析结果，如图5.17，其中 $x_1(t)$, $x_2(t)$ 和 $x_a(t)$ 分别

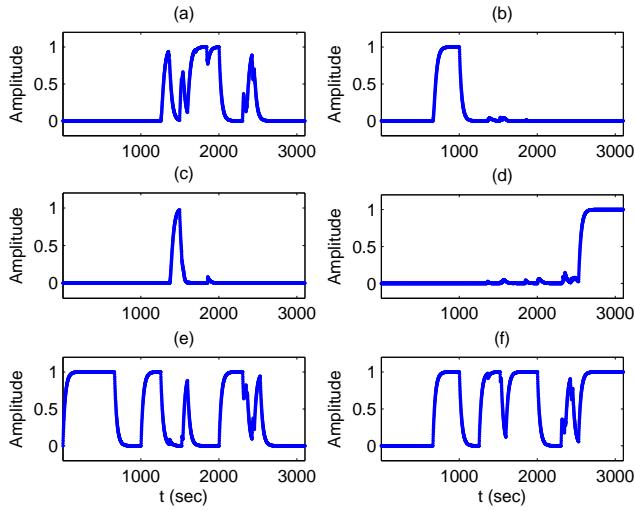
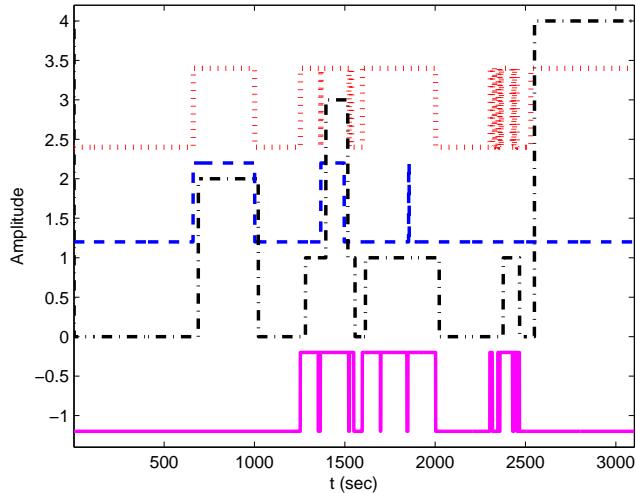
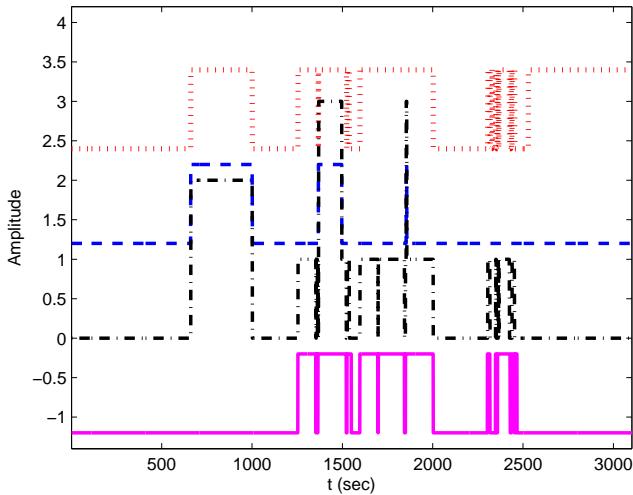


图 5.16 后验概率参数分子的时序图

图 5.17 $x_1(t)$ 、 $x_2(t)$ 、 $x_a(t)$ 及 $\hat{j}(t)$ 时序图

在振幅偏移上偏移-1.2、1.2、2.4能获得更好的视觉效果。参数 $j = 0, 1, 2, 3, 4$ 代表以下情况 $\{x_a = 0\}$, $\{x_1 = 1, x_2 = 0, x_a = 1\}$, $\{x_1 = 0, x_2 = 1, x_a = 1\}$, $\{x_1 = 1, x_2 = 1, x_a = 1\}$, $\{x_1 = 0, x_2 = 0, x_a = 1\}$ 。

该方法成功解决了第5.3节中讨论的三个挑战。第一，该方法消除了 x_1 , x_2 , x_a 中持续时间小于20个样本其误报警和漏报警的负面影响。 x_1 在 $t \in [2303, 2316]$, $t \in [2348, 2351]$ 内有误报警；在 $t \in [1354, 1355]$, $t \in [1362, 1364]$, $t \in [1522, 1525]$, $t = 1698$, $t \in [1844, 1849]$, $t \in [2429, 2436]$, $t \in [2453, 2457]$ 内有漏报警。 x_2 在 $t \in [1854, 1856]$ 内有误报警。 x_a 在 $t \in [2305, 2317]$, $t \in [2328, 2330]$, $t \in [2349, 2355]$, $t \in [2363, 2364]$ 内有误报警；在 $t \in [1367, 1368]$,

图 5.18 $x_1(t)$ 、 $x_2(t)$ 、 $x_a(t)$ 及 $\hat{j}(t)$ 时序图

$t \in [1524, 1527]$, $t \in [2425, 2432]$, $t \in [2436, 2439]$ 内有漏报警。通过对比, $\hat{j}(t)$ 不受这些误报警和漏报警的影响。第二, 提出的方法可以处理多个根本原因变量共存的问题。在时间间隔 $t \in [1250, 1600]$ 左右, x_1 是导致 x_a 报警状态的一个根本原因, x_2 在一段时间后也导致 x_a 处于报警状态。当两个根本原因共存时, 提出的方法得到了正确的结果, 即 $\hat{j}(t) = 3$ 。第三, 该方法可以解决已知根本原因的不完全性问题。在时间间隔 $t \in [2530, 3000]$ 左右, X_a 处于报警状态, 但 x_1 和 x_2 都取值 ‘0’。因此, 发生的报警是由 x_1 以外的根本原因引起的, 用该方法得到 $\hat{j} = 4$, 成功检测到了这种情况。

然而, 通常的做法是通过观察数据样本, 那么 $x_a(t) = 1$ 的根本原因是 $x_i(t) = 1$, $i \in [1, n]$ 。在这种情况下, 根本原因结果用 $\tilde{j}(t)$ 表示, 即:

$$\tilde{j}(t) = \begin{cases} 0, & x_a(t) = 0 \\ 1, & x_1(t) = 1, x_2(t) = 0, x_a(t) = 1 \\ 2, & x_1(t) = 0, x_2(t) = 1, x_a(t) = 1 \\ 3, & x_1(t) = 1, x_2(t) = 1, x_a(t) = 1 \end{cases}.$$

通过此方法得到的根本原因变量分析结果如图5.18所示, 其中 $\tilde{j}(t)$ 受到 $x_1(t)$ 和 $x_2(t)$ 中误报警的严重影响。因此, $\hat{j}(t)$ 比 $\tilde{j}(t)$ 具有更好的性能, 但在检测时延方面有所欠缺。

检测时间延迟是指 x_a 中报警发生与从 $\hat{j}(t)$ 中检测相应根本原因之间的时间差。如第5.3.2节所述, 概率参数 $\theta(t)$ 从 $\theta(0)$ 变为 $\theta(\infty)$ 需要 m 个样本。因为 $\hat{j}(t)$ 是从概率参数的乘积中得到的, 检测时间延迟将等于或大于 m , 这取决于更新概率参数的数量。例如, $x_a(t)$ 在 $t = 662$ 时将其值从 ‘0’ 更改为 ‘1’, 并且 $\hat{j}(t)$ 在 $t = 690$ 时将其值从 ‘0’ 更改为 ‘2’, 因此检测时间延迟为 28s, 图5.15显示了 $\theta_{2,1}(t)$ 和 $\theta_{a,1|j=2}(t)$ 更新过程。再如, $x_a(t)$ 在 $t = 2530$ 时从 ‘0’ 更改为 ‘1’, $\hat{j}(t)$ 在 $t = 2550$ 时从 ‘0’ 更改为 ‘4’, 因此检测时间延迟为 20s, 只有 $\theta_{a,1|j=4}(t)$ 和 $\theta_{a,0|j=4}(t)$ 正在更新, 如图5.15 所示。由此可知, 在贝叶斯网络中, 利用不同的概率更新参数可以减少检测时间延迟。

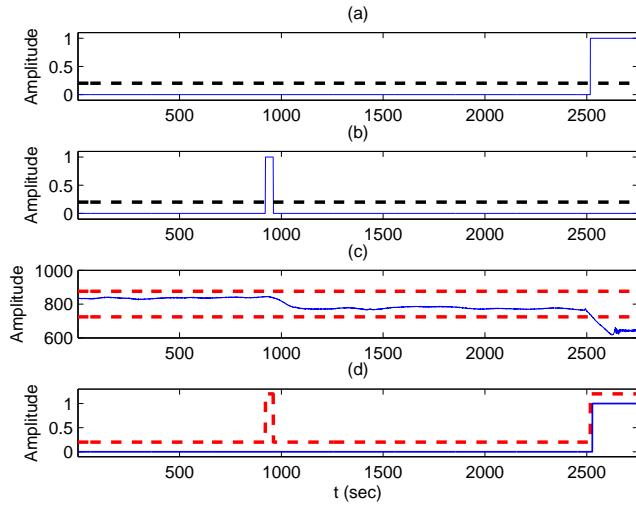
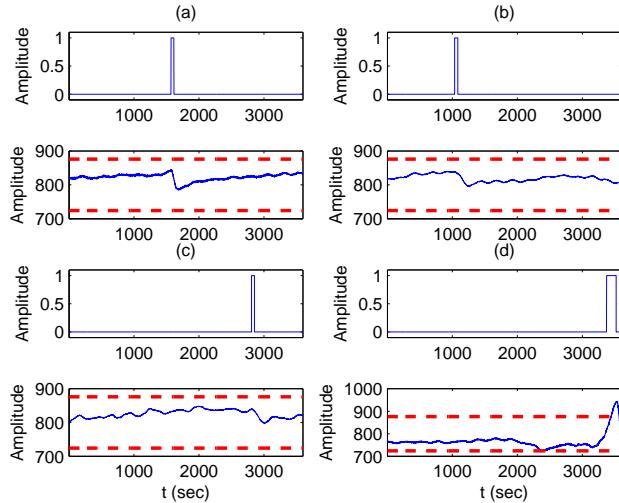


图 5.19 例2中收集的数据样本

图 5.20 x_3 , x_c 的历史数据样本

案例2: 凝汽器热井水位是一个重要的过程变量，水位过高会降低换热空间和冷凝器真空度，降低发电效率；如果水位过低，出口水泵会出现汽蚀现象，对除氧器水位产生不利影响。因此，配置报警变量 x_a 来指示凝汽器热井水位异常情况的发生。该变量通过逻辑机制产生并且有五个根本原因，其根本原因如下： x_1 , x_2 分别代表A、B出口凝汽器水泵故障； $x_3(x_4)$ 是表示凝汽器热井水位高于（低于）876mm(300mm)的硬性指标； x_5 是当冷凝器热井水位的模拟测量。 x_c 大于876mm或小于724mm时，单报警变量输出取值1。当 x_1 , x_2 , ..., x_5 中的任意一个取值为1时， x_a 处于报警状态。

首先，图5.19给出了关于 $x_a(t)$, $x_c(t)$, $x_1(t)$, $x_2(t)$, ..., $x_5(t)$ 的数据样本。图中 x_a 有

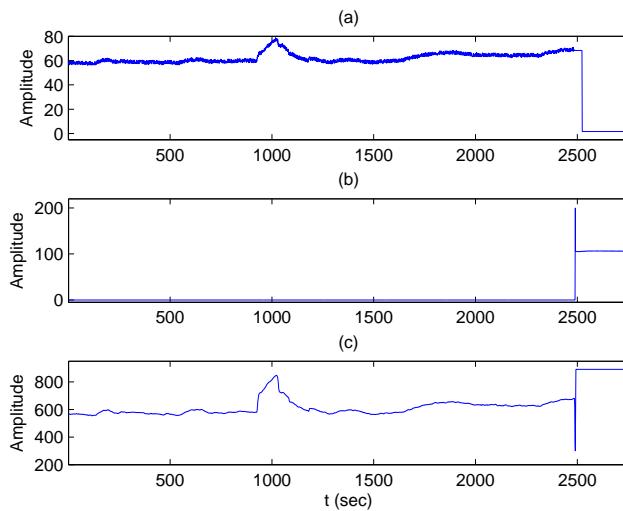


图 5.21 例2中相关变量的数据样本

两个时间间隔进入报警状态，通过可视化可以看出 $x_a(t)$ 和 $x_3(t)$ 取值为1时， $t \in [922, 960]$ ，然而 x_c 明显处于正常工作区间[724, 876]mm内。因此，可判定这些报警属于误报警。事实上 x_c 处在正常工作状态， x_3 经常出现误报警。图5.20给出了 x_3 和 x_c 的四组历史数据样本，其中a, b, c三组产生了误报警，d组产生真实报警。

其次， $x_a(t)$ ， $x_1(t)$ ， $x_5(t)$ 分别在时间2518，2518，2528处从切换到‘1’。图5.21(a)中，A出口水泵电流从非零值突变到零值，验证了 $x_a(t)$ 由于A出口凝汽器水泵故障进入报警状态。由于具有互锁机构，B出口冷凝器水泵自动打开，如图5.21(b)所示。图5.21(c)中的冷凝器出口水流量迅速增加到过大值。结果， x_c 减小到小于报警下限247mm，因此 x_5 将其值从‘0’切换到‘1’。

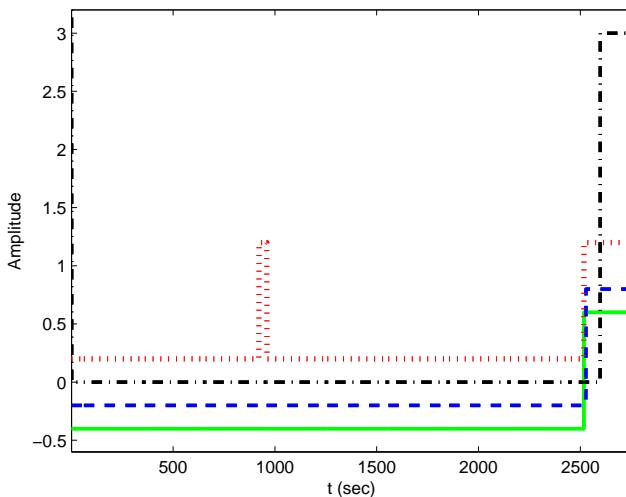


图 5.22 根本原因分析

最后，将所提出的方法应用到该报警变量的检测当中，其中 $\lambda = 0.0115$, $m = 60$ 。根源分析结果，如图5.22所示， $x_3(t)$, $x_5(t)$, $x_a(t)$ 的幅值变化幅度为-0.4、-0.2、0.2, $j = 0, 1, 2, 3, 4$ 分别代表如下情况： $x_a = 0$, $\{x_3 = 1, x_5 = 0, x_a = 1\}$, $\{x_3 = 0, x_5 = 1, x_a = 1\}$, $\{x_3 = 1, x_5 = 1, x_a = 1\}$, $\{x_3 = 0, x_5 = 0, x_a = 1\}$ 。分析结果 $\hat{j}(t)$ 不受 $x_3(t)$, $t \in [922, 960]$ 受误报警影响，在 $t = 2598$ 时刻成功检测到 x_3 , x_5 同时存在异常。

5.4 基于定性趋势的报警根源分析

包含 R 个过程变量的多变量系统可以记为 $x = [x_1, x_2, \dots, x_R]$ 。第 r 个过程变量 x_r 的时间序列记为 $X_r(1:L) = [x_r(1), x_r(2), \dots, x_r(L)]^T$, $r = 1, 2, \dots, R$ 。 x_r 的定性趋势记为 \vec{X}_r , 减少、不变和增加的定性趋势^[13]分别记为-1, 0, +1。 \vec{X}_r 的时间序列为 $\vec{x}_r(1:L) = [\vec{x}_r(1), \vec{x}_r(2), \dots, \vec{x}_r(L)]^T$ 。过程变量的定性趋势组合 z 定义如下：

$$z = [\vec{X}_1, \vec{X}_2, \dots, \vec{X}_R]. \quad (5.45)$$

系统 X 的历史数据段中所有的定性趋势组合构成了集合 $Z = \{z_1, z_2, \dots, z_K\}$, 其中第 k 个定性趋势组合 z_k 是(5.45)中的一种具体情况, $1 \leq k \leq K$ 。定性趋势组合中受大量高概率支持的数据段称为显著定性趋势组合。显著定性趋势组合构成集合 Z_0 , Z_0 是 Z 的子集, 即, $Z_0 \subset Z$ 。

基于时间序列 $X(1:L) = [x_1(1:L), x_2(1:L), \dots, x_R(1:L)]$, 目标是寻找 Z 和 Z_0 。除此之外, 还需找到服从各定性趋势组合 z_k 的相似数据段, $1 \leq k \leq K$ 。

5.4.1 基于定性趋势的主要思想

通过一个示例展示了所提方法的主要思想。考虑包含两个过程变量的双变量系统 $x = [x_1, x_2]$ 。图5.23(a)中展示了历史数据中10个数据段的时间序列图。5.4.2节将说明如何获取这些数据段。图5.23(b)展示了定性趋势的时间序列 \vec{X}_1 和 \vec{X}_2 。

各个数据段的数据长度可能不相同, 并且 x_1 和 x_2 的定性趋势变化并不总是严格同步。例如, 第五个数据段中 x_1 和 x_2 在 $t = 20$ 之前都是增加的, 但是 $t = 20$ 之后, x_2 开始减少而 x_1 继续保持增加。所以此数据段中同时包含了 $[+1, +1]$ 和 $[+1, -1]$ 的定性趋势组合。这种来自多变量的异步性是寻找 Z 和 Z_0 的主要挑战。

为了解决该异步性, 提出了多特征主题模型 (multi-feature topic model, MFTM), 通过对数据段的定性趋势直方图聚类来寻找定性趋势组合。如图5.23(c)所示, 对于每个数据段, 定性趋势直方图包含了各个过程变量 X_r 的减少、不变和增加趋势, 即 $f_{r,1} = -1$, $f_{r,2} = 0$, $f_{r,3} = +1$, $r = 1, 2, \dots, R$ 。尽管存在不同数据长度和定性趋势变化的异步性, 定性趋势直方图反映出前五个数据段的主要定性趋势组合为同增加, 即 $[+1, +1]$, 而后五个数据段的主要定性趋势组合为 $[-1, -1]$ 。因此, 所有可能的定性趋势组合总数为 $3 \times 3 = 9$, 其中只有两个组合是显著的, 即同时增加 $[+1, +1]$ 和同时减少 $[-1, -1]$ 。5.4.3节将说明MFTM的细节问题。

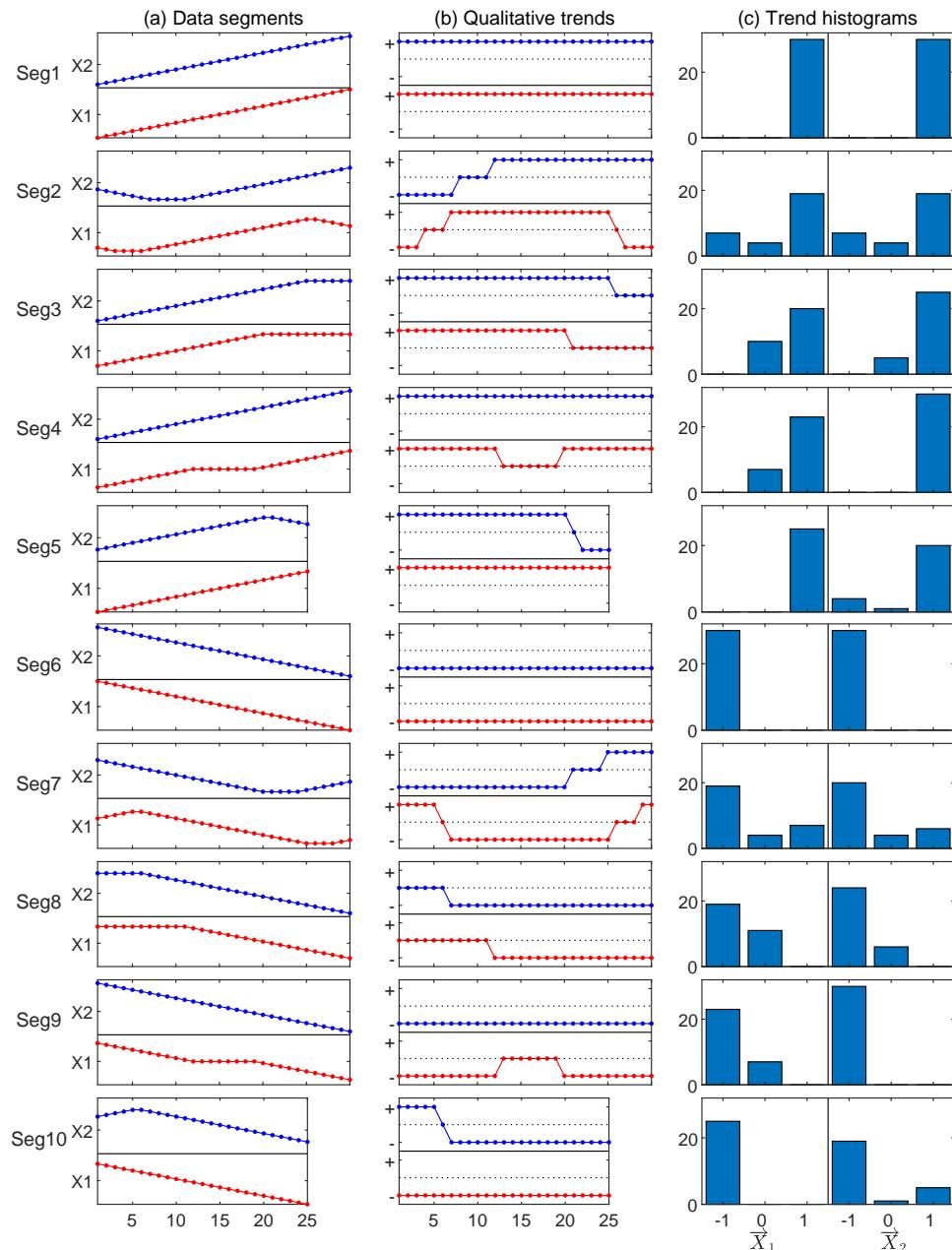


图 5.23 时间序列及其定性趋势直方图

5.4.2 求取定性趋势组合方法步骤

求取定性趋势组合首先要对数据段进行分段处理，在对数据段进行分段处理之前，对于多变量系统 $x = [x_1, x_2, \dots, x_R]$ ，首先应当消除历史数据当中的时间延迟。通过使用常见的互相关方法^[14]估计延迟时间，两个时间序列 $X_{r_1} (1 : L)$ 和 $X_{r_2} (1 : L)$ 之间的延迟可按下式估计，

$$\tau_{r_1, r_2} = \arg \max_{\tau} \sum_{t=1}^L [X_{r_1}(t) X_{r_2}(t - \tau)]. \quad (5.46)$$

其中 $1 \leq r_1, r_2 \leq R$ 。若 $\tau_{r_1, r_2} > 0$ ，则可通过把时间序列 $X_{r_1} (1 : L)$ 向后平移 τ_{r_1, r_2} 个时间单位消除时间延迟。反之，若 $\tau_{r_1, r_2} < 0$ ，则向前平移 τ_{r_1, r_2} 个时间单位即可。对于多变量系统，首先需要选取中心变量，即对系统 x 安全高效运行最关键的过程变量。之后其他过程变量的时间序列根据对中心变量的延迟时间进行平移。此外，通过应用定性趋势直方图，所提出的方法可以部分容忍时间延迟带来的异步性。例如，尽管5.4.1节中数据段的定性趋势并不严格同步，这种延迟并不会对定性趋势直方图所对应的定性趋势组合造成改变。

每个过程变量 x_r 的历史时间序列 $X_r (1 : L)$ 可以通过分段线性表示法 [15–18] (piecewise linear representation, PLR) 转换成 N_r 个线性分段，即

$$\left\{ \hat{X}_r(t_{i,r} : t_{i+1,r} - 1) \right\}_{i=1}^{N_r} = \text{PLR}(X_r(1 : L)), \quad (5.47)$$

其中 $t_{i,r}$ 为第*i*个数据段的开始时间戳， $i = 1, 2, \dots, N_r$ ， $t_{1,r} = 1$ ， $t_{N_r+1,r} = L + 1$ 。分段线性表示法中使用线性回归来获取各个线性分段，即

$$\hat{X}_r(t) = a_{i,r} + b_{i,r} \cdot t, \quad (5.48)$$

其中 $t_{i,r} \leq t \leq t_{i+1,r} - 1$ ， $i = 1, 2, \dots, N_r$ 。使用最小二乘法，截距 $a_{i,r}$ 和斜率 $b_{i,r}$ 计算如下：

$$\begin{cases} a_{i,r} = \frac{\sum t^2 \sum \hat{x}_r(t) - \sum t \sum t \hat{x}_r(t)}{(t_{i+1,r} - t_{i,r}) \sum t^2 - (\sum t)^2} \\ b_{i,r} = \frac{\sum t \hat{x}_r(t) - \sum t \sum \hat{x}_r(t)}{(t_{i+1,r} - t_{i,r}) \sum t^2 - (\sum t)^2} \end{cases}. \quad (5.49)$$

其中求和符号是 $\sum_{t=t_{i,r}}^{t_{i+1,r}-1}$ 的简写。

使用自下而上法来计算式(5.47)中的分段结果。之后，基于分段结果可得定性趋势。使用对斜率 $b_{i,r}$ 显著性的假设检验计算定性趋势^[20]

$$H_0 : b_{i,r} = 0, H_1 : b_{i,r} \neq 0. \quad (5.50)$$

当下式满足时，拒绝原假设 H_0

$$\frac{|b_{i,r}|}{\hat{\sigma}} \sqrt{S_{xx}} \geq t_{\frac{\alpha}{2}}(n - 2). \quad (5.51)$$

其中 $t_{\frac{\alpha}{2}}(n - 2)$ 是显著性水平为 α 时，自由度为 $n - 2$ 的t分布的分位数，例如 $\alpha = 0.05$ ，有

$$\begin{cases} \hat{\sigma} = \sqrt{\frac{1}{n-2} \sum_{t=t_{i,r}}^{t_{i+1,r}-1} [x_r(t) - a_{i,r} - b_{i,r} \cdot t]^2} \\ S_{xx} = \sum_{t=t_{i,r}}^{t_{i+1,r}-1} t^2 - \frac{1}{t_{i+1,r} - t_{i,r}} \left(\sum_{t=t_{i,r}}^{t_{i+1,r}-1} t \right)^2 \end{cases}.$$

因此, $t_{i,r} \leq t \leq t_{i+1,r} - 1$ 的定性趋势 $\vec{x}_r(t)$ 为

$$\vec{x}_r(t) = \begin{cases} \text{sign}(b_{i,r}), & (5.51) \text{ 成立} \\ 0, & \text{其他} \end{cases} \quad (5.52)$$

其中 $\text{sign}(\cdot)$ 为符号函数, $i = 1, 2, \dots, N_r$, $t_{1,r} = 1$, $t_{N_r+1,r} = L + 1$ 。

分段线性表示会对每个过程变量 x_r 都生成切分时间截的集合 $S_r = \{t_{i,r}, i = 1, 2, \dots, N_r\}$ 。为了避免生成太短的数据段, 基于最小数据长度 T 的约束将各个 S_r 聚合为统一的切分时间截集合 $S = \{t_i, i = 1, 2, \dots, N\}$ 。算法6展示了计算得到 S 的伪代码。此处长度 T 是构成定性趋势所需的最小长度。在实际中最小数据长度 T 的概念是十分常见的。工业标准 [6, 12] 建议在产生关于流量和压力变量的报警时, 需要先观察到连续至少15秒的数据超过报警线, 而对液位和温度变量则应采用60秒的时间。此外, 需要假设 x_1, x_2, \dots, x_R 之间的异步性局限于 T 内。在接下来的案例1中将讨论参数 T 的敏感性。使用切分时间截 $S = \{t_i, i = 1, 2, \dots, N\}$ 可以得到数据段集合 $D = \{d_1, d_2, \dots, d_N\}$, 即

$$d_i = [\vec{x}_1(t_i : t_{i+1} - 1), \vec{x}_2(t_i : t_{i+1} - 1), \dots, \vec{x}_R(t_i : t_{i+1} - 1)], \quad (5.53)$$

其中 $t_{i+1} - t_i \geq T$ 。例如, 图5.24(a)和图5.24(c)中两个时间序列可以被转换成图5.24(g)和5.24(i)中一系列的二维数据段, 记为 $1, 2, \dots, 9$, $T = 100$ 。切分时间截 S_1, S_2 和 S , 如图5.24(b)、图5.24(d) 和图5.24(e) 中的竖实线所示。图5.24(e) 中的虚线为未能满足最小数据长度约束的切分时间截。需要指出的是, 应用算法6后, 一个数据段中可能包含不止一种定性趋势, 如图5.24中数据段2是同时包含+1和-1的定性趋势。因此, 可采用定性趋势直方图来描述这种现象。

Algorithm 6 生成聚合切分时间截集合 S

```

输入参数#1:  $S_r$ , 每一个变量  $x_r$  切分时间截集合  $\{t_{i,r}, i = 1, 2, \dots, N_r\}$  ;
输入参数#2:  $T$ , PLR方法中定性趋势的最小数据长度;
 $S_0 = []$ ; % 空集
for  $r = 1$  to  $R$  do
     $S_0 = \text{sort}(\text{union}(S_0, S_r))$ ; % 对并集排序
end for
 $N = \text{length}(S_0)$ ;
 $S = [1]$ ;
for  $i = 2$  to  $N$  do
    if  $S_0(i) - S(end) \geq T$  then
         $S = [S, S_0(i)]$ ;
    end if
end for
Return  $S$ 

```

对于每个过程变量 x_r 的数据段, 定性趋势直方图包含减少、不变和增加三种定性趋势元素, 分别记为 $f_{r,1} = -1$, $f_{r,2} = 0$ 和 $f_{r,3} = +1$, $r = 1, 2, \dots, R$ 。基于从数据段 $D = \{d_1, d_2, \dots, d_N\}$ 得到的定性趋势直方图, MFTM模型可以获得 K 个聚类 $C = \{c_k\}_{k=1}^K$ 及其条件概率 $P(c_k|d_i)$ 和 $P(f_{r,j_r}|c_k)$, $k = 1, 2, \dots, K$, $i = 1, 2, \dots, N$, $r = 1, 2, \dots, R$, $j_r = 1, 2, 3$ 。对于每个过程变量 X_r , 选取具有最高概率 $P(f_{r,j_r}|c_k)$ 的定性趋势元素 f_{r,j_r} , 可以将每个聚类显

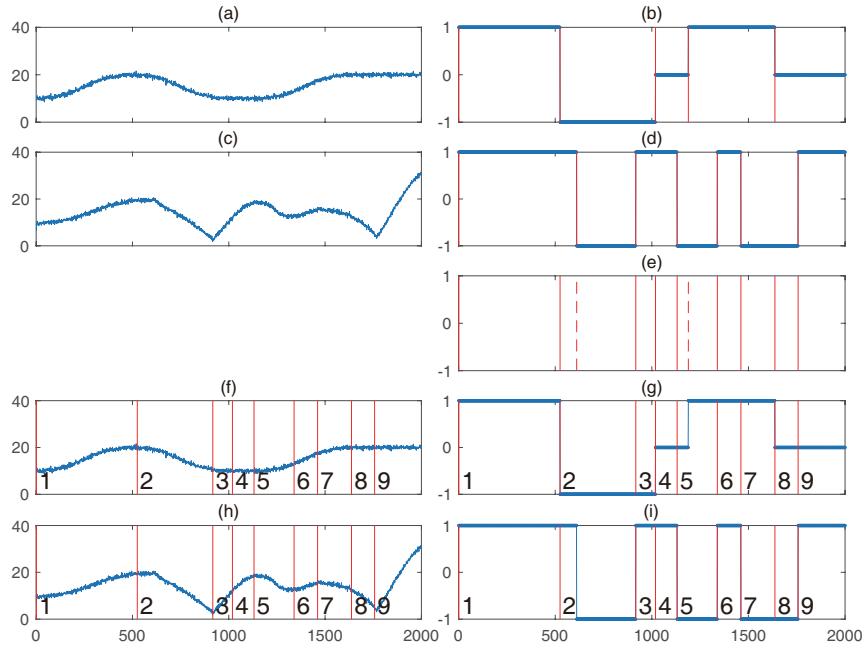


图 5.24 时间序列聚合切分展示

式地表达成对应的定性趋势组合 z_k , $k = 1, 2, \dots, K$, $r = 1, 2, \dots, R$, $j_r = 1, 2, 3$, 即

$$z_k = \left[\arg \max_{f_{1,j_1}} P(f_{1,j_1} | c_k), \arg \max_{f_{2,j_2}} P(f_{2,j_2} | c_k), \dots, \arg \max_{f_{R,j_R}} P(f_{R,j_R} | c_k) \right]. \quad (5.54)$$

数据段集合 D 中的每个数据段 d_i 都标记为其最有可能属于的定性趋势组合 \hat{z}_i , 即

$$\hat{z}_i = z \arg \max_k P(c_k | d_i), \quad (5.55)$$

同时, 相应的隶属概率为 p_i 表示为

$$p_i = P(\hat{z}_i | d_i). \quad (5.56)$$

正如5.4节所定义, 显著定性趋势组合 Z_0 是 $Z = \{z_1, z_2, \dots, z_K\}$ 的子集。而 Z_0 应被足够多的数据段以高概率支持。由于 $P(c_k | d_i)$ 的总和为 1, $k \in [1, K]$, 如果 D 中出现更多高概率 $P(c_k | d_i)$ 的数据段, 则 $c_k(z_k)$ 相对更显著。因此, 提出了一种寻找 Z_0 的方法, 如下所示:

1) 对于每个 c_k , 都计算 $[P(c_k | d_i)]^q$ 的均值, 即

$$\mu_k = \frac{1}{N} \sum_{i=1}^N [P(c_k | d_i)]^q. \quad (5.57)$$

其中, 为了提高较大值的权重, 要求指数阶次 q 是一个不小于 2 的整数。采用默认值 $q = 3$ 。

2) 计算 C 的全局均值, 即

$$\mu_0 = \frac{1}{NK} \sum_{i=1}^N \sum_{k=1}^K [P(c_k|d_i)]^q. \quad (5.58)$$

3) 选择满足约束 $\mu_k > \mu_0$ 的聚类 c_k 所对应的定性趋势组合 z_k 组成 Z_0 ,

$$Z_0 = \{z_k | \mu_k > \mu_0, z_k \in Z, 1 \leq k \leq K\}. \quad (5.59)$$

以上步骤可以通过算法7实现。此外, 图5.25的流程图展示了所提方法的具体步骤。

Algorithm 7 寻找 Z_0

```

输入参数#1:  $K$ , 聚类个数;
输入参数#2:  $P$ ,  $N * K$ 矩阵包含的条件概率 $P(c_k|d_i)$ ;
 $\mu = \text{zeros}(K, 1)$ ;
% 计算 $P(c_k|d_i)$ 均值
for  $k = 1$  to  $K$  do
     $\mu(k) = \text{std}(P(1:N, k))$ ;
end for
 $\mu = \text{sort}(\sigma)$ ;
% 计算全局均值
 $\mu_0 = \text{std}(P(:))$ ;
 $Z = []$ 
for  $k = 1$  to  $K$  do
    if  $\mu(k) \geq \mu_0$  then
         $Z = [Z, k]$ ;
    end if
end for
Return  $Z$ 

```

5.4.3 多特征主题模型

本节说明MFTM模型的详细内容。由于MFTM模型是PLSA模型的推广, 先给出PLSA模型的理论细节。

PLSA模型的目标在于提取文本集的主题^[21]。假设文本集合 $D = \{d_1, d_2, \dots, d_N\}$ 由词语集合 $W = \{w_1, w_2, \dots, w_M\}$ 组成, 并且使用隐变量 $C = \{c_1, c_2, \dots, c_K\}$ 表示主题。PLSA模型使用一元语言模型假设, 即文本中的每个单词都是以一定概率并独立同分布地产生于某个主题。因此, 一个观察数据对的联合概率为 $P(d_i, w_j)$, $i = 1, 2, \dots, N$, $j = 1, 2, \dots, M$,

$$\begin{aligned}
 P(d_i, w_j) &= P(d_i) P(w_j|d_i) \\
 &= P(d_i) \sum_{k=1}^K P(w_j|c_k) P(c_k|d_i).
 \end{aligned} \quad (5.60)$$

其中 $P(d_i)$, $P(w_j|c_k)$ 和 $P(c_k|d_i)$ 分别为第*i*个文本 d_i 的概率, 给定主题 c_k 产生第*j*个词语 w_j 的条件概率, 文本 d_i 中包含主题 c_k 的条件概率。

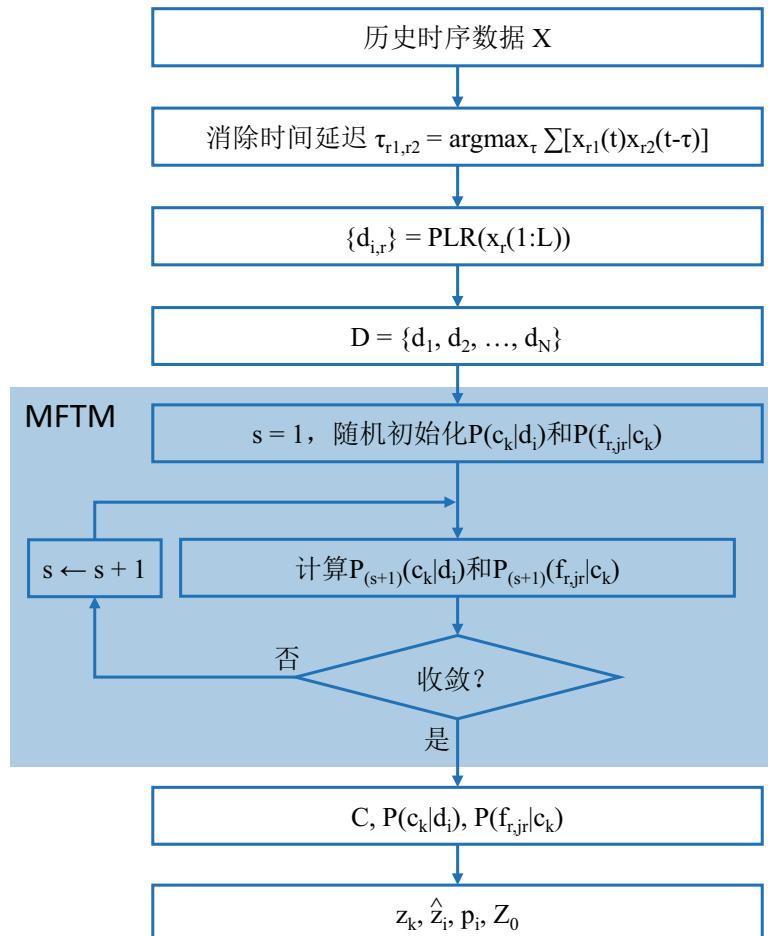


图 5.25 所提出方法的流程图

通过将词语替换为多种特征，式(5.60)所定义的PLSA模型可以扩展为MFTM模型，其中PLSA模型和MFTM模型都可以通过图5.26所示的贝叶斯网络描述。MFTM模型是PLSA模型在高维特征空间中的扩展形式，或者说，PLSA模型是MFTM模型的一种特例。

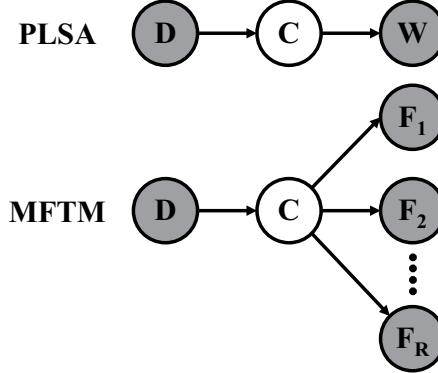


图 5.26 PLSA模型和MFTM模型的示意图

假设有 R 种特征需要进行分析，记特征集合为 $F = \{F_1, F_2, \dots, F_R\}$ 。每种特征 F_r 包含了 M_r 个元素，即， $F_r = \{f_{r,1}, f_{r,2}, \dots, f_{r,M_r}\}$ ， $r = 1, 2, \dots, R$ 。 F_r 定义为 X_r 的减少、不变和增加趋势，即， $f_{r,1} = -1$ ， $f_{r,2} = 0$ ， $f_{r,3} = +1$ ， $M_r = 3$ ， $r = 1, 2, \dots, R$ 。与PLSA模型的一元语言模型假设相一致，假设每种特征条件都依赖于聚类 c_k ，但彼此相互独立。因此，一个样本 $(d_i, f_{1,j_1}, f_{2,j_2}, \dots, f_{R,j_R})$ 可以拆分为多个相互独立的数据对 (d_i, f_{1,j_1}) ， (d_i, f_{2,j_2}) ， \dots ， (d_i, f_{R,j_R}) ，即

$$P(d_i, f_{1,j_1}, f_{2,j_2}, \dots, f_{R,j_R}) = P(d_i, f_{1,j_1}) P(d_i, f_{2,j_2}) \cdots P(d_i, f_{R,j_R}), \quad (5.61)$$

其中， $1 \leq j_r \leq M_r$ ， $r = 1, 2, \dots, R$ 。与式(5.60)类似， (d_i, f_{r,j_r}) 的联合概率为

$$\begin{aligned} P(d_i, f_{r,j_r}) &= P(d_i) P(f_{r,j_r} | d_i) \\ &= P(d_i) \sum_{k=1}^K P(f_{r,j_r} | c_k) P(c_k | d_i), \end{aligned} \quad (5.62)$$

其中， $r = 1, 2, \dots, R$ 。该式是MFTM模型的概率形式，其中 $P(d_i, f_{r,j_r})$ 可以通过对 d_i 中的 f_{r,j_r} 进行计数简单计算得到，稍后式(5.64)将给出计算方式，因而是已知的。而未知参数集合为

$$\theta = \{P(c_k | d_i), P(f_{r,j_r} | c_k)\},$$

其中， $k = 1, 2, \dots, K$ ， $i = 1, 2, \dots, N$ ， $r = 1, 2, \dots, R$ ， $j_r = 1, 2, \dots, M_r$ 。 θ 中一共有 $NK + \sum_{r=1}^R M_r K$ 个未知量。因此，求解MFTM模型是在已知 $P(d_i, f_{r,j_r})$ 的前提下求解未知参数集合 θ 的过程。极大似然估计方法（maximum likelihood estimation, MLE）常常被用于估计概率模型中的未知参数。然而，由于式(5.63)中的对数似然概率 $L_1(\theta)$ 在对数项中包含了求和过程，这种非

线性导致极大似然估计方法中的求偏导过程难以实施,

$$\begin{aligned} L_1(\theta) &= \sum_{i=1}^N \sum_{r=1}^R \sum_{j_r=1}^{M_r} [n(d_i, f_{r,j_r}) \log P(d_i, f_{r,j_r})] \\ &= \sum_{i=1}^N \sum_{r=1}^R \sum_{j_r=1}^{M_r} \left\{ n(d_i, f_{r,j_r}) \cdot \log \left[\sum_{k=1}^K P(f_{r,j_r}|c_k) P(c_k|d_i) \right] \right\}. \end{aligned} \quad (5.63)$$

其中第二个等式来自于式(5.62), 且对于某个具体的数据段 d_i 有 $P(d_i) = 1$ 。 $n(d_i, f_{r,j_r})$ 是 d_i 中出现 f_{r,j_r} 的次数, 即

$$n(d_i, f_{r,j_r}) = |\{t | \vec{x}_r(t) = f_{r,j_r}, t_i \leq t \leq t_{i+1} - 1\}|. \quad (5.64)$$

作为替代, 基于数据 $(d_i, f_{1,j_1}), (d_i, f_{2,j_2}), \dots, (d_i, f_{R,j_R})$, 采用期望最大化 (expectation maximization, EM) 算法^[22] 来估计未知参数 θ 。EM算法包含E步骤和M步骤, 如下所示:

记当前计算步为第 $(s+1)$ 次迭代, $s = 1, 2, \dots$ 。对于E步骤, θ 中的 $P(c_k|d_i)$ 和 $P(f_{r,j_r}|c_k)$ 取值为之前第 (s) 次迭代中M 步骤的计算结果。使用贝叶斯方法计算隐变量 z_k 的后验概率,

$$\begin{aligned} P_{(s+1)}(c_k|d_i, f_{r,j_r}) &= \frac{P_{(s)}(c_k, d_i, f_{r,j_r})}{\sum_{l=1}^K P_{(s)}(c_l, d_i, f_{r,j_r})} \\ &= \frac{P_{(s)}(f_{r,j_r}|c_k) P_{(s)}(c_k|d_i)}{\sum_{l=1}^K P_{(s)}(f_{r,j_r}|c_l) P_{(s)}(c_l|d_i)}. \end{aligned} \quad (5.65)$$

其中, 下标 (s) 和 $(s+1)$ 分别表明为第 (s) 次迭代和第 $(s+1)$ 次迭代。EM 算法的M步骤将式(5.63)中的 $L_1(\theta)$ 转换成其他形式, 即

$$\begin{aligned} L_1(\theta) &= \sum_{i=1}^N \sum_{r=1}^R \sum_{j_r=1}^{M_r} [n(d_i, f_{r,j_r}) \log P(d_i, f_{r,j_r})] \\ &= \sum_{i=1}^N \sum_{r=1}^R \sum_{j_r=1}^{M_r} \left[n(d_i, f_{r,j_r}) \log \sum_{k=1}^K P(c_k, d_i, f_{r,j_r}) \right] \\ &= \sum_{i=1}^N \sum_{r=1}^R \sum_{j_r=1}^{M_r} \left[n(d_i, f_{r,j_r}) \cdot \log \sum_{k=1}^K P(c_k|d_i, f_{r,j_r}) \frac{P(c_k, d_i, f_{r,j_r})}{P(c_k|d_i, f_{r,j_r})} \right]. \end{aligned} \quad (5.66)$$

使用琴生不等式^[23] (Jensen inequality), 即 $\log \sum_k \lambda_k y_k \geq \sum_k \lambda_k \log y_k$, 其中 $\sum_k \lambda_k = 1$, $\lambda_k \geq 0$, $y_k \geq 0$, 得到式(5.66)中 $L_1(\theta)$ 的下界 $B(\theta)$, 即

$$L_1(\theta) \geq \sum_{i=1}^N \sum_{r=1}^R \sum_{j_r=1}^{M_r} \left[n(d_i, f_{r,j_r}) \cdot \sum_{k=1}^K P(c_k|d_i, f_{r,j_r}) \log \frac{P(c_k, d_i, f_{r,j_r})}{P(c_k|d_i, f_{r,j_r})} \right] = B(\theta). \quad (5.67)$$

M步骤将最大化 $L_1(\theta)$ 的下界 $B(\theta)$, 进而 $L_1(\theta)$ 也可能增加。因此, 随着E步骤和M步骤的迭代过程, 似然概率 $L_1(\theta)$ 逐渐增加, 其解为

$$\hat{\theta} = \arg \max_{\theta} L_1(\theta) = \arg \max_{\theta} B(\theta). \quad (5.68)$$

由于当前迭代步的M步骤中认为 $P(c_k|d_i, f_{r,j_r})$ 是已知的，很容易证明式(5.67)中的 $B(\theta)$ 和式(5.69)中的似然概率函数 $L_2(\theta)$ 在同时取得最大化，

$$L_2(\theta) = \sum_{i=1}^N \sum_{r=1}^R \sum_{j_r=1}^{M_r} \left\{ n(d_i, f_{r,j_r}) \cdot \sum_{k=1}^K \{P(c_k|d_i, f_{r,j_r}) \log [P(f_{r,j_r}|c_k) P(c_k|d_i)]\} \right\}. \quad (5.69)$$

因此，其解为

$$\hat{\theta} = \arg \max_{\theta} B(\theta) = \arg \max_{\theta} L_2(\theta). \quad (5.70)$$

基于约束 $\sum_{j_r=1}^{M_r} P(f_{r,j_r}|c_k) = 1$ 和 $\sum_{k=1}^K P(c_k|d_i) = 1$ ，则拉格朗日函数表示为

$$H = L_2(\theta) + \sum_{r=1}^R \sum_{k=1}^K \tau_{r,k} \left(1 - \sum_{j_r=1}^{M_r} P(f_{r,j_r}|c_k) \right) + \sum_{i=1}^N \rho_i \left(1 - \sum_{k=1}^K P(c_k|d_i) \right),$$

其中 $\tau_{r,k}$ 和 ρ_i 拉格朗日乘数。令 H 对 $P(f_{r,j_r}|c_k)$ ， $P(c_k|d_i)$ ， $\tau_{r,k}$ 及 ρ_i 的偏导数为零，第 $(s+1)$ 次迭代步的解 $\hat{\theta} = \arg \max_{\theta} L_2(\theta)$ 表示为

$$\begin{cases} P_{(s+1)}(c_k|d_i) = \frac{\sum_{r=1}^R \sum_{j_r=1}^{M_r} n(d_i, f_{r,j_r}) P_{(s+1)}(c_k|d_i, f_{r,j_r})}{\sum_{r=1}^R n_r(d_i)} \\ P_{(s+1)}(f_{r,j_r}|c_k) = \frac{\sum_{i=1}^N n(d_i, f_{r,j_r}) P_{(s+1)}(c_k|d_i, f_{r,j_r})}{\sum_{m_r=1}^{M_r} \sum_{i=1}^N n(d_i, f_{r,m_r}) P_{(s+1)}(c_k|d_i, f_{r,m_r})} \end{cases}, \quad (5.72)$$

其中式(5.64)给出了 $n(d_i, f_{r,j_r})$ ，且 $n_r(d_i) = \sum_{j_r=1}^{M_r} n(d_i, f_{r,j_r})$ 为 d_i 中特征 F_r 各元素出现的次数之和。

总而言之，MFTM模型包含以下步骤：

1) 随机初始化 $P(c_k|d_i)$ 和 $P(f_{r,j_r}|c_k)$ 。

2) 迭代计算式(5.65)和式(5.72)。

3) 如果满足收敛条件 $\|\theta_{(s+1)} - \theta_{(s)}\| < \varepsilon$ ，则停止步骤2) 的迭代过程。其中 ε 为一个较小的正数，如 $\varepsilon = 0.001$ ， $\|\cdot\|$ 为计算欧氏范数的操作符。通过MFTM模型的困惑度 (perplexity) 可以确定聚类 c_1, c_2, \dots, c_K 的数量 K 。离散概率模型 $q(x)$ 和未知概率分布 $p(x)$ 的困惑度 [24] 定义为

$$\eta_q = \exp \left[- \sum_x \tilde{p}(x) \log q(x) \right], \quad (5.73)$$

其中 $\tilde{p}(x)$ 为 $p(x)$ 的样本分布，即， $\tilde{p}(x) = \frac{n}{N}$ 。若 x 在大小为 N 的样本集合中出现了 n 次。困惑度量了一个概率模型预测样本取值的不确定性程度。一般地，较低的困惑度表明该概率模型具有较好的预测能力。MFTM模型的困惑度 η 定义为各个特征的困惑度之积，即

$$\eta(K) = \prod_{r=1}^R \eta_r(K)$$

$$= \prod_{i=1}^N \prod_{r=1}^R \prod_{j_r=1}^{M_r} \exp \left[-\frac{n(d_i, f_{r,j_r})}{\sum_{i=1}^N n_r(d_i)} \cdot \log \left(\sum_{k=1}^K P(f_{r,j_r}|c_k) P(c_k|d_i) \right) \right]. \quad (5.74)$$

随着聚类数量 K 的增加，该模型将逐步拟合训练数据集并使得困惑度 $\eta(K)$ 减少。例如图5.29(a)展示了随聚类数量增加而递减的困惑度曲线。然而，该模型的泛化能力也同时衰减，导致模型在训练数据集上过拟合。因此，合适的聚类数量 K 应当使得 K 和 $\eta(K)$ 都不太大。一种经验方法是选择 K 从小到大直到 K_0 第一次满足下式，

$$\frac{\eta(K_0) - \eta(K_{\max})}{\eta(1) - \eta(K_{\max})} < \gamma. \quad (5.75)$$

其中 K_{\max} 是一个足够大的数使得 $\eta(K)$ 趋于平缓，比值 γ 是较小的数字。案例1将对参数 γ 进行敏感性分析。

5.4.4 案例分析

案例1：本例将说明所提定性趋势分析方法的准确性，并研究参数敏感性及噪声干扰的情况。图5.27所示的双水箱系统可以通过一个动力学模型表示，

$$\begin{cases} S_1 \frac{dH_1}{dt} = F_i - F_{o_1} - F_p \\ F_{o_1} = S_2 \cdot \sqrt{2gH_1} \\ S_1 \frac{dH_2}{dt} = F_p - F_{o_2} \\ F_{o_2} = S_2 \cdot \sqrt{2gH_2} \end{cases}. \quad (5.76)$$

其中 F_i 表示入水口流量， F_{o_1} 和 F_{o_2} 分别表示通过两个水箱各自出口阀门的流量。 H_1 和 H_2 表示两个水箱中的液位， S_1 和 S_2 分别表示水箱和阀门的截面积， g 表示重力常数，即 $g = 9.8$ 。入水流量 F_i 以固定的转换时间 Δt_s 在较高值 F_h 和较低值 F_l 之间来回转换。入水流量 F_i 在 F_h 和 F_l 上的保持时间 Δt 表示为 $\Delta t = \Delta t_0 + \Delta t_1$ ，其中 Δt_0 为常数， Δt_1 表示满足均值为 $\frac{1}{\Delta t_0}$ 的指数分布。两个水箱之间有一个电泵，当左侧水箱的水位 H_1 超过阈值 H_0 时将以固定流速 F_p 运行，持续时间 T_p 后关闭。其中 F_p ， H_0 和 T_p 均为常数。当电泵开机时，将水从左侧水箱抽到右侧水箱，因而 H_1 将下降且 H_2 增加。 F_{o_1} 和 F_{o_2} 分别受液位 H_1 和 H_2 控制。表5.5中列出了本案例的参数设置。图5.28展示了 F_i ， F_{o_1} ， F_{o_2} ， H_1 和 H_2 共5000秒的时间序列图，其每秒采样一次。

使用所提出的方法来分析多变量系统 $x = [F_i, F_{o_1}, F_{o_2}]$ 。由于 H_1 和 H_2 总是分别与 F_{o_1} 和 F_{o_2} 保持同增同减的关系，因此为简便起见而不加入多变量系统中进行分析。应用算法6后，图5.28被转换成满足最小数据长度约束 $T = 60$ 的 $N = 58$ 个数据段。基于图5.29(a)中所示的困惑度曲线，式(5.75)中取 $\gamma = 0.03$ ，选择定性趋势组合的数量为 $K = 7$ 。图5.29(b)展示了式(5.63)中对数似然概率 $L_1(\theta)$ 随EM算法的迭代而逐步极大化的过程。由图5.29可知，当迭代次数收敛于 $s = 99$ 时， $L_1(\theta)$ 达到最大值。

表 5.5 案例1的参数设置

参数	S_1	S_2	H_0	g	Δt_s	Δt_0	T_p	F_h	F_l	F_p
值	5	0.05	2	9.8	150	50	100	0.4	0.2	0.3

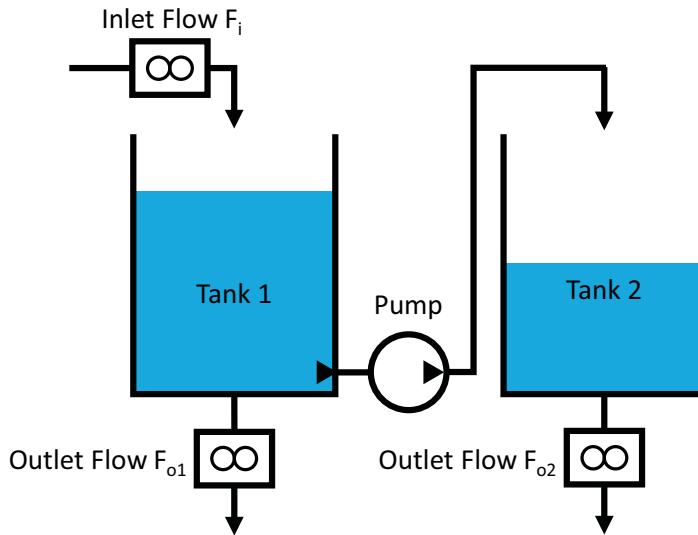


图 5.27 案例1中的双水箱的示意图

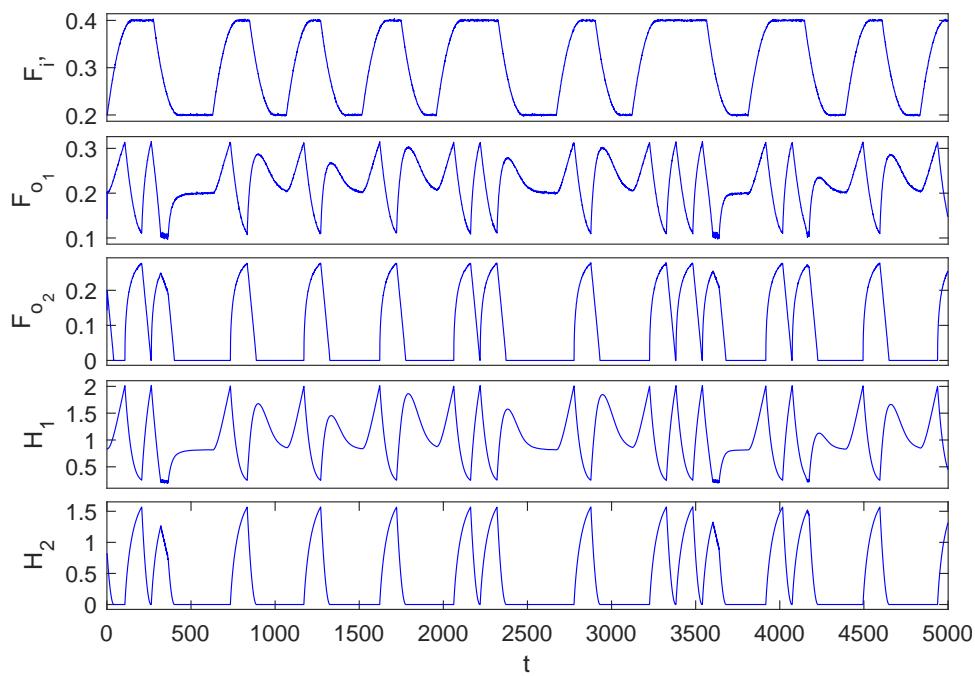
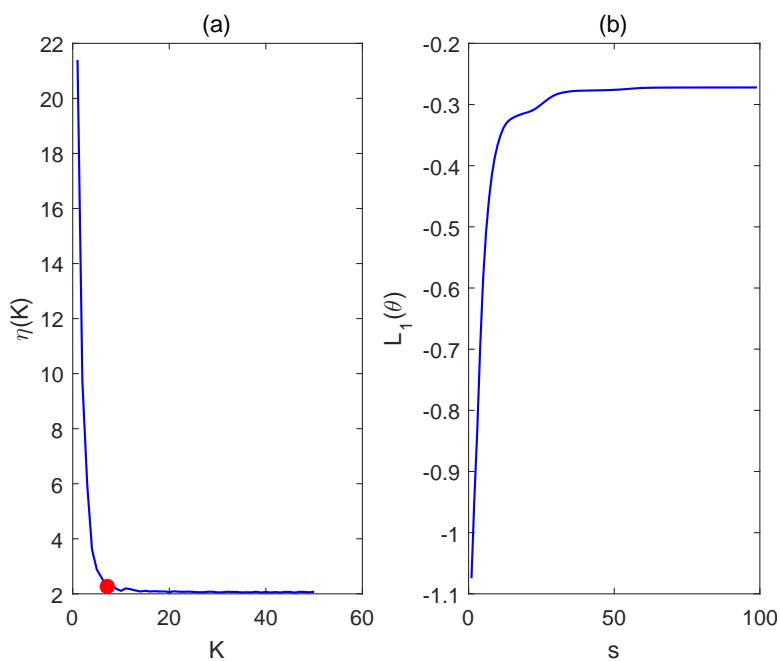
使用式(5.54)中的 $P(f_{r,j_r}|c_k)$ 从聚类 C 中推导出 Z , 如表5.6所示。图5.30给出了 $P(c_k|d_i)$ 的条形图, $1 \leq k \leq 7$ 。由图得到显著定性趋势组合 $Z_0 = \{z_1, z_6\}$ 。相较于其他定性趋势组合, z_1 和 z_6 收到更多数据段的高概率支持。例如, 对于 $c_6(z_6)$, 有13个数据段满足 $P(c_6|d_i) > 0.8$, 而对于 $c_3(z_3)$ 只有4个数据段满足 $P(c_3|d_i) > 0.8$, 因此 z_6 相比于 z_3 更加显著。据式(5.55)选择具有最高概率的 $P(c_k|d_i)$ 所对应的 c_k 得到 $\{\hat{z}_i\}_{i=1}^N$ 。图5.31展示了位于 $1 \leq t \leq 2000$ 的时间序列图。基于此, 很容易找到属于相同定性趋势组合的相似数据段。通过检查数据中的变化过程, 可以从中得到数据段 $D = \{d_i\}_{i=1}^N$ 定性趋势组合的真值 $\{z_i\}_{i=1}^N$ 。例如, 考虑图5.31中 $\{\hat{z}_i\}_{i=1}^N$ 位于 $1501 \leq t \leq 1622$ 的数据, 可知 F_i 从较低值开始增加, 并且电泵保持关闭状态, 因此左侧水箱中的液位上升并且没有水被泵入右侧水箱。因此, 此时的定性趋势组合应为 $z_1 = [+1, +1, 0]$ 。将计算值 $\{\hat{z}_i\}_{i=1}^N$ 与真值 $\{z_i\}_{i=1}^N$ 对比, 可以计算聚类准确率 α 如下,

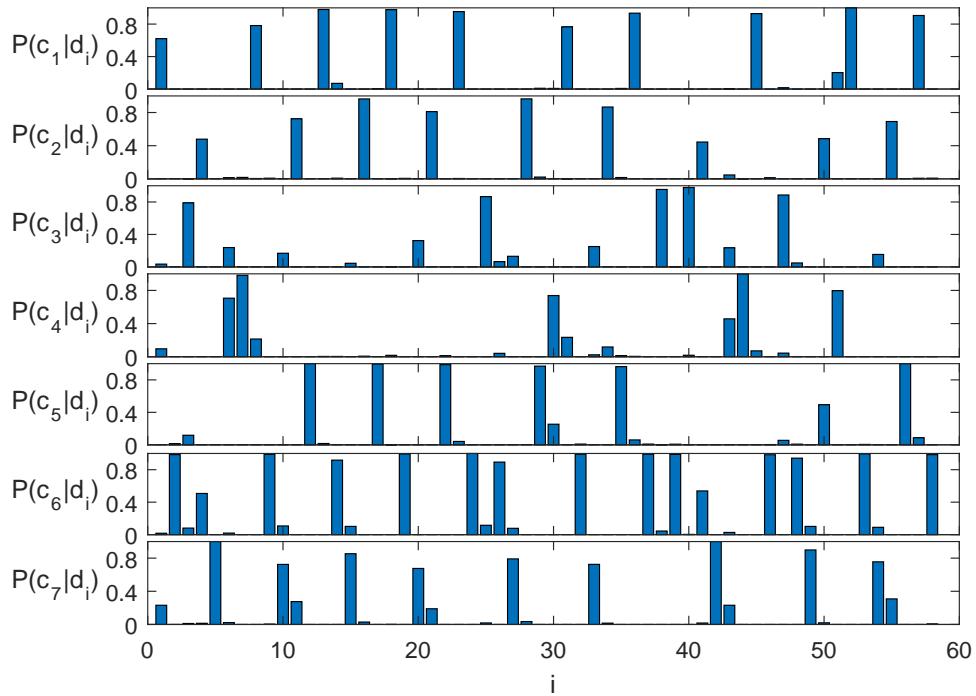
$$\alpha = \frac{|\{i|\hat{z}_i = z_i, 1 \leq i \leq N\}|}{N}. \quad (5.77)$$

本例中的准确率为94.8%。有5.2%的错误是源于离群值, 即少量具有 Z 中非常少见的定性趋势组合。例如, 图5.31位于 $264 \leq t \leq 326$ 中的数据被标记为 $z_6 = [0, -1, +1]$, 然而其定性趋势组

表 5.6 案例1中的定性趋势组合 Z , 对 Z_0 进行了加深标记

Z	\vec{X}_1	\vec{X}_2	\vec{X}_3
z_1	+1	+1	0
z_2	-1	-1	0
z_3	0	+1	-1
z_4	0	0	0
z_5	0	-1	0
z_6	0	-1	+1
z_7	-1	+1	-1

图 5.28 案例1中过程变量 F_i , F_{o1} , F_{o2} , H_1 和 H_2 的时间序列图图 5.29 困惑度曲线 $\eta(K)$ 及对数似然概率 $L_1(\theta)$

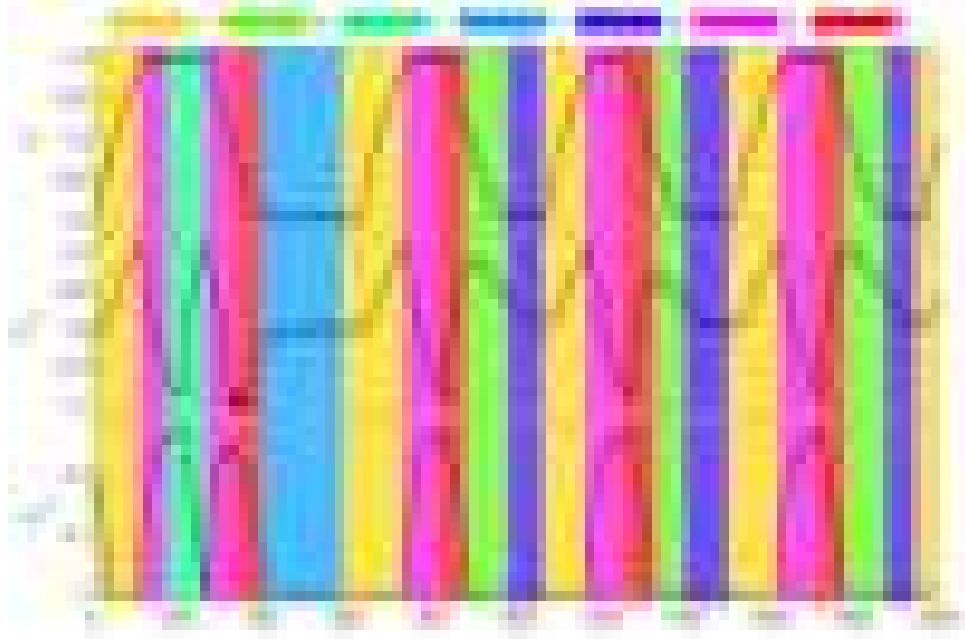
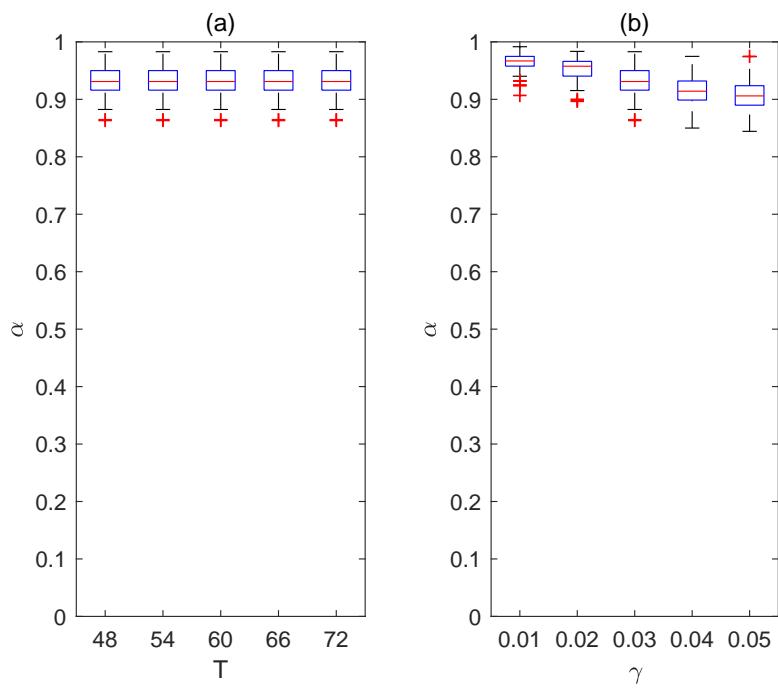
图 5.30 $P(c_k|d_i)$ 的条形图

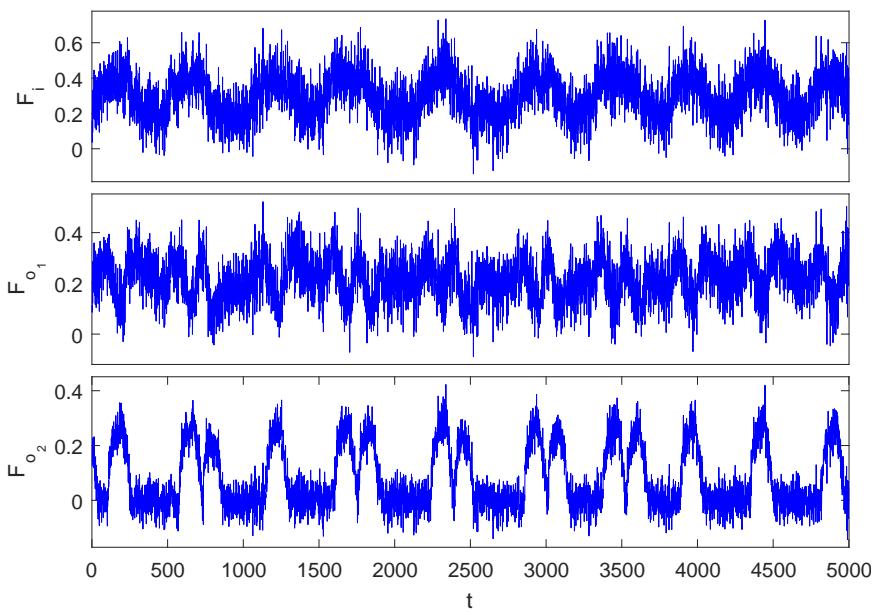
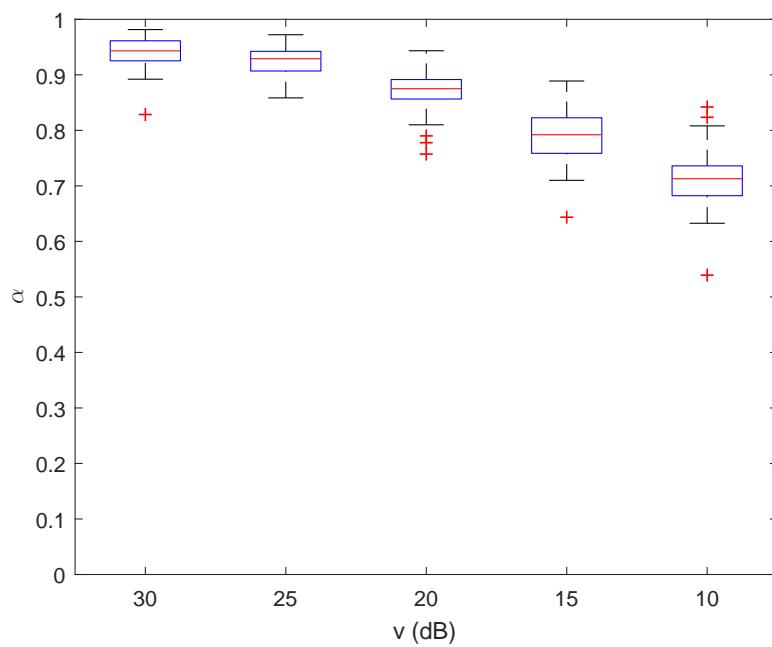
合的真值应当为 $[-1, -1, +1]$ 。因此，所提出的方法能够准确地得到多变量系统中的定性趋势组合。

定性趋势组合 z_1 , z_2 和 z_4 表明 F_{o_1} 随 F_i 的改变而改变，并且 F_{o_2} 处于保持不变，对于液位 H_1 低于阈值 H_0 且泵并未工作。定性趋势组合 z_3 , z_6 和 z_7 表明无论 F_i 取何值， F_{o_1} 和 F_{o_2} 都处于相反的走势，对应于电泵将水从左侧水箱输送到右侧水箱。定性趋势组合 z_5 对应于左侧水箱的入水流量转换到较低值后 F_{o_1} 下降的暂态过程。

基于蒙特卡罗仿真研究参数敏感性。在每次蒙特卡罗仿真中，通过对随机变量设置不同的随机种子进而得到时间序列数据。所提出的方法中需要操作人员指定的超参数，包括算法6中的最小数据长度 T 和式(5.75)中的比值 γ 。如式(5.52)定义，定性趋势的物理含义可被定义为过程变量的一阶变化符号，应当独立于数据段的时间长度。对于式(5.75)中的比值 γ ，若 γ 增加，则导致定性趋势组合的数量 K 减少。 K 值对结果的影响是：当 K 减少时，相似的定性趋势组合会被合并。在这种情况下，模型趋于欠拟合的状态，导致准确率下降。与上述基线模型相比较，图5.32展示了式(5.77)所定义的准确率随 T 从48到72的不同取值（ $\pm 20\%$ 相对于基线60）的箱线图，以及准确率随 γ 从0.01到0.05的不同取值（ $\pm 66.7\%$ 相对于基线0.03）的变化。与预期相符，准确率随 T 取不同值几乎保持不变，并且随着 γ 的增加而略微减少。因此 T 对所提方法的准确率并无显著影响，但 γ 是可以影响定性趋势组合数量的关键参数。

使用蒙特卡罗仿真进一步对噪声的影响进行研究。在每次蒙特卡罗仿真中，对系统 x 的每个过程变量 x_r 都叠加独立的高斯噪声 e_r ，即， $x_{r,e} = x_r + e_r$, $r = 1, 2, 3$ 。使用单位为分

图 5.31 不同定性趋势组合 $\{\hat{z}_i\}$ 的颜色区分情况图 5.32 不同 T 、 γ 下准确率 α 箱线图

图 5.33 案例1中信噪比为 $v = 10\text{dB}$ 的系统 x_e 的时间序列图图 5.34 准确率 α 随信噪比从 $v = 30\text{dB}$ 到 $v = 10\text{dB}$ 的不同取值的变化的箱线图

贝dB的信噪比计算方式度量各过程变量 x_r 的时间序列 $X_r(t)$ 噪声水平,

$$v_r = 10 \log_{10} \frac{\sum_{t=1}^L [X_r(t)]^2}{L \sigma_r^2},$$

其中 σ_r^2 为 e_r 的方差。简便起见, 三个过程变量的信噪比保持相同, 即 $v_1 = v_2 = v_3 = v$ 。图5.33展示了信噪比为 $v = 10$ dB时系统 $x_e = [x_{1,e}, x_{2,e}, x_{3,e}]$ 的时间序列图。将所提方法应用于不同信噪比 v 的 x_e 。较高的噪声水平将导致式(5.47)产生较多的错误分段结果, 进而影响式(5.52)中定性趋势的识别, 但是定性趋势直方图对错误的定性趋势也具有一定容忍能力。图5.34展示了信噪比从 $v = 30$ dB到 $v = 10$ dB不同取值下准确率 α 变化的箱线图。与预期相符, 当噪声处于较低水平时算法有极高的准确率, 而当噪声增加时准确率也处于可接受的范围内。

案例2: 本例将MFTM模型与其他相关的时间序列聚类方法进行对比。为方便对比, 此例中仅讨论单变量时间序列。一个时间序列 $x_0(1 : 300)$ 定义为

$$x_0 = \begin{cases} t, & 1 \leq t \leq 50 \\ 50, & 51 \leq t \leq 100 \\ 151 - t, & 101 \leq t \leq 150 \\ 1, & 151 \leq t \leq 200 \\ t - 200, & 201 \leq t \leq 250 \\ 50, & 251 \leq t \leq 300 \end{cases},$$

则它的定性趋势 $\vec{x}_0(1 : 300)$ 为

$$\vec{x}_0(t) = \begin{cases} +1, & t = 1 \\ \text{sign}(x_0(t) - x_0(t-1)), & 1 < t \leq 300 \end{cases},$$

其中 $\text{sign}(\cdot)$ 为符号函数。通过随机地从 $\vec{x}_0(1 : 300)$ 上进行切分, 得到等长度 T 的数据段 $D = \{d_1, d_2, \dots, d_N\}$,

$$d_i = \vec{x}_0(t_i : t_i + T - 1).$$

其中 t_i 满足 $[1, 200]$ 中的离散均匀分布, $i = 1, 2, \dots, N$ 。为了解决5.4.1节所述的异步性, 本例中的主要困难来自于数据段可能包含多种定性趋势。本例中, 数据长度约束 T 设置为75, 数据段的总数 N 为150。图5.35展示了一个示例数据段。

本例中所有数据均等长且无噪声, 且都是以相同的方式产生。因此, 根据每个数据段的幅

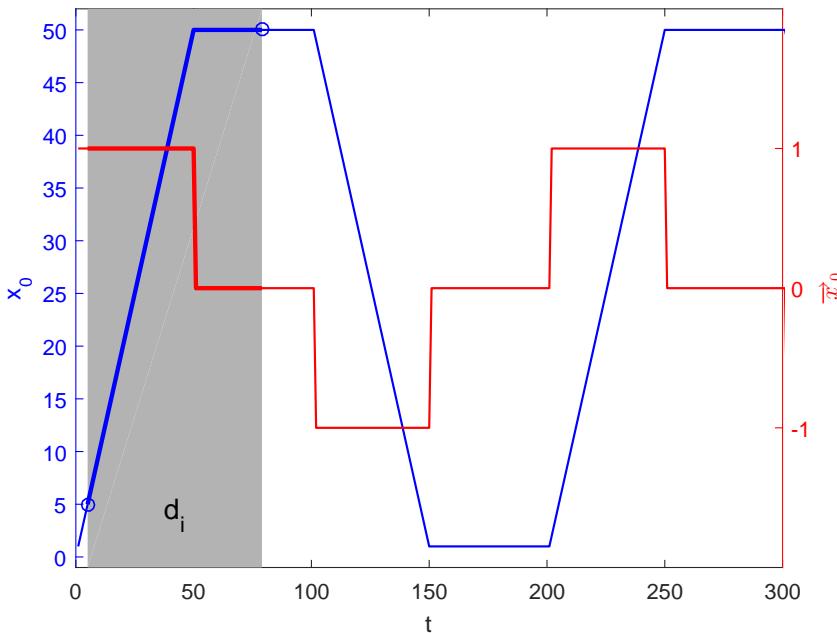


图 5.35 案例2示例数据段样本

值变化是否超过了 x_0 的一半可以得到每个数据段 d_i 的定性趋势 z_i , 即, $(1 + 50) / 2 = 24.5$,

$$z_i = \begin{cases} -1, & x_0(t_i + T - 1) - x_0(t_i) < -24.5 \\ 0, & |x_0(t_i + T - 1) - x_0(t_i)| \leq 24.5 \\ +1, & x_0(t_i + T - 1) - x_0(t_i) > 24.5 \end{cases} \quad (5.78)$$

本例中将通过对 D 中数据段 d_i 应用多种不同的时间序列聚类方法来得到标签 \hat{z}_i 。根据所使用的特征不同, 各个聚类方法归类如下:

- 基于原始数据段 $x_0(t_i : t_i + T - 1)$: (a)K均值聚类^[25], (b)基于欧式距离的层次聚类法^[26], (c)基于动态时间卷积距离^[27]的层次聚类法, (d)基于最长公共子序列距离^[28]的层次聚类法, (e)基于序列编辑距离^[29]的层次聚类法, (f)基于互相关距离^[30]的层次聚类法;
- 基于定性趋势数据段 $\vec{x}_0(t_i : t_i + T - 1)$: (g)K均值聚类^[25], (h)基于欧式距离的层次聚类法^[26], (i)基于动态时间卷积距离^[27]的层次聚类法, (j)基于最长公共子序列距离^[28]的层次聚类法, (k)基于序列编辑距离^[29]的层次聚类法, (l)基于互相关距离^[30]的层次聚类法;
- 基于定性趋势直方图 $[n(d_i, -1), n(d_i, 0), n(d_i, 1)]$: (m)K均值聚类^[25], (n)基于欧式距离的层次聚类法^[26], (o)模糊c均值聚类法^[31], (p)自组织网络聚类法^[32], (q)基于J散度的层次聚类法^[33], (r)5.4.3节所提出的MFTM模型。

由于本例中 $\{z_i\}_{i=1}^N$ 包含三种趋势, 即-1, 0和+1, 设置所有聚类方法的聚类数量均为3。对

于每一种聚类方法，使用式(5.77)计算所得的标签 $\{\hat{z}_i\}_{i=1}^N$ 对真实值 $\{z_i\}_{i=1}^N$ 的准确率 α 。图5.36展示了各聚类方法中准确率的箱线图。

结果来看，大部分应用于定性趋势数据段的方法获得了比相应应用于原始数据段更高的准确率，并且当应用于定性趋势直方图时达到了更高的准确率。该现象说明对于识别定性趋势来说，定性趋势直方图是相对较好的特征。最高的两个准确率均值分别为(q)基于J散度的层次聚类法的97.07%和(r)MFTM模型的97.61%。该原因是J散度和MFTM都将定性趋势直方图处理为离散概率分布，即存在约束 $n(d_i, -1) + n(d_i, 0) + n(d_i, +1) = n(d_i) = T$ ，而其他方法(m)、(n)、(o)和(p)都仅仅是将定性趋势直方图处理为三维空间中的向量。此外，MFTM模型以概率 $P(c_k|d_i)$ 赋予数据段标签，而不是采用类似基于J散度的层次聚类法中的硬划分方法。这有助于提高大量数据段其定性趋势不太明确时的准确率。因此，如图5.36所示，(r)MFTM模型的准确率 α 其下界显著地高于(q)基于J散度的层次聚类法中的准确率下界。所以对于获取相同定性趋势的聚类，MFTM模型能够取得比其他聚类方法更好的性能表现。

案例3：本例中展示了所提出的方法在实际流程工业环境中的应用。并利用显著定性趋势组合和相应的相似数据段可以进一步进行报警根源分析，给水泵对于某个大型火力发电厂是非常重要的工业设备。前置泵电流 x_1 、入水流量 x_2 、入水压力 x_3 、泵转速 x_4 和出水压力 x_5 等五个过程变量组成了多变量系统 $x = [x_1, x_2, x_3, x_4, x_5]$ 。当泵处于正常运行状态时，这五个过程变量应当保持同增同减的关系。图5.37展示了正常状态下这些过程变量的一个样本数据段的时间序列图。

基于系统X一个月的历史时间序列数据，使用算法6获取了 $N = 507$ 个数据段，得到 $K = 7$ 种定性趋势组合。基于图5.38的上侧子图，计算得出Z的显著定性趋势组合

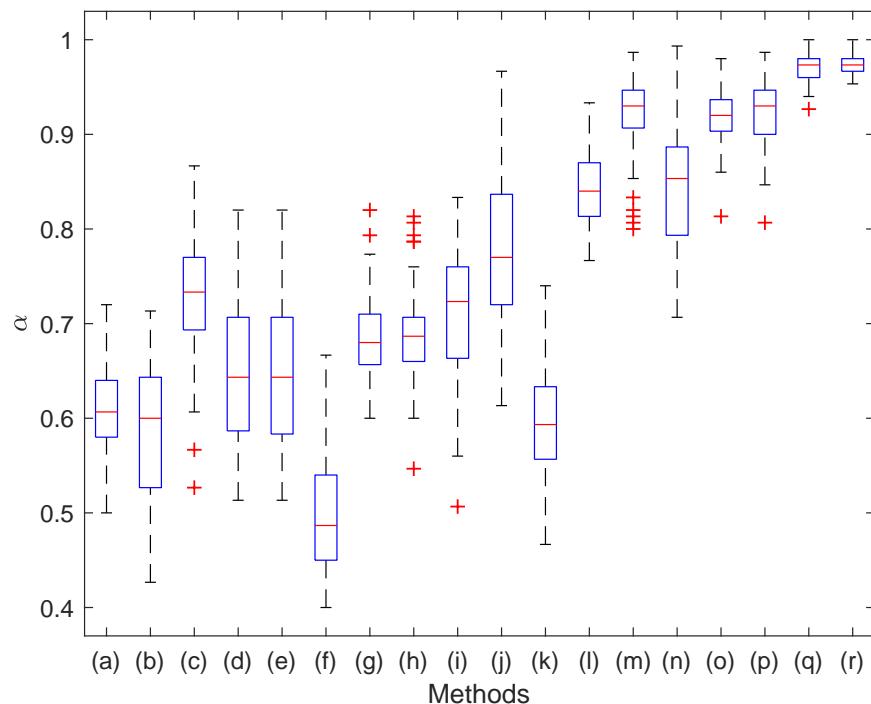


图 5.36 案例2中各方法聚类准确率 α 的箱线图

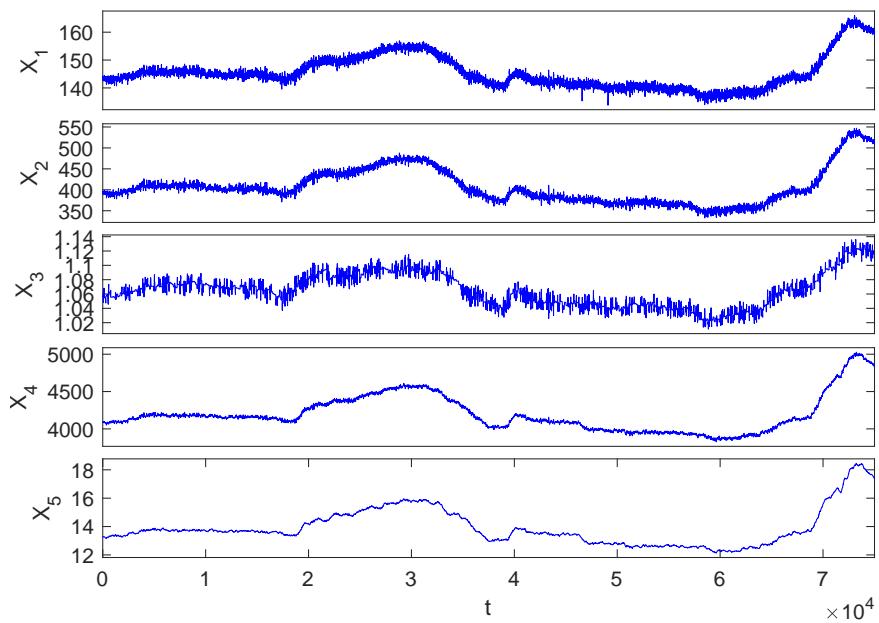


图 5.37 案例3过程变量正常状态时间序列图

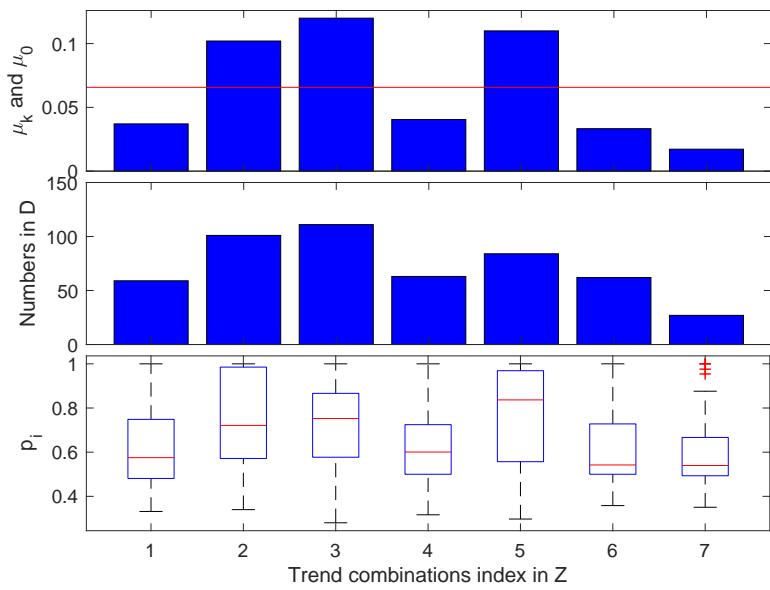
图 5.38 μ_k 、 \hat{z}_i 及 p_i 情况展示

表 5.7 案例3中的定性趋势组合 Z , 对 Z_0 进行了加深标记

Z	\vec{X}_1	\vec{X}_2	\vec{X}_3	\vec{X}_4	\vec{X}_5
z_1	-1	0	-1	-1	-1
z_2	0	0	0	0	0
z_3	+1	+1	-1	+1	+1
z_4	+1	0	0	0	0
z_5	-1	-1	+1	-1	-1
z_6	-1	0	+1	0	0
z_7	-1	-1	-1	0	0

图 5.39 定性趋势组合为 $z_2 = [0, 0, 0, 0, 0]$ 的三个数据段示例

为 $Z_0 = \{z_2, z_3, z_5\}$, 如表5.7中加深标记。这些显著定性趋势组合也可以通过如下信息进行验证。式(5.55)中标签 $\{\tilde{z}_i\}_{i=1}^N$ 的频数直方图和式(5.56)中隶属概率 p_i 的箱线图, 如图5.38的中部及下侧子图所示。通过检查图5.38中中部子图的条形图高度, 可以得知 z_2 , z_3 和 z_5 相较于其他定性趋势组合更为常见。此外, 基于图5.38的下侧子图可知, 标记满足 $\tilde{z}_i \in \{z_2, z_3, z_5\}$ 的数据段拥有更高的隶属概率 p_i 。正如5.4节所描述的, 显著定性趋势组合应当得到大量数据段的高概率支持。这个现象表明定性趋势组合 Z 中 z_2 , z_3 和 z_5 均为显著定性趋势组合。这与所得的结果 Z_0 相符。

图5.39、图5.40和图5.41分别展示了定性趋势组合 z_2 , z_3 和 z_5 的典型代表数据段。其中, $z_2 = [0, 0, 0, 0, 0]$ 可以解释为“所有过程变量同时保持不变”。这说明系统处于稳定状态。 $z_3 = [+1, +1, -1, +1, +1]$ 和 $z_5 = [-1, -1, +1, -1, -1]$ 表明过程变量 x_3 的变化方向与其他变量相反。

显著定性趋势组合和相应的相似数据段可以进一步用于报警生成及根源分析。对于报警生成, 当系统的定性趋势组合为 $z_3 = [+1, +1, -1, +1, +1]$ 或 $z_5 = [-1, -1, +1, -1, -1]$, 则可判定泵处于异常状态。因此, 实时计算 x_1, x_2, \dots, x_5 的定性趋势^[34], 当它们的组合为 z_3 或 z_5 时产生报警。注意由于多个过程变量所拥有的定性趋势组合数量呈指数增长, 在缺乏历史数据的支持而仅仅通过人工制定规则监控过程变量的定性趋势组合是低效的。对于报警根源分析, z_3



图 5.40 定性趋势组合为 $z_3 = [+1, +1, -1, +1, +1]$ 的三个数据段示例



图 5.41 定性趋势组合为 $z_5 = [-1, -1, +1, -1, -1]$ 的三个数据段示例

和 z_5 表明产生异常的根源变量来自于入水压力 x_3 的反向变化。若系统当前处于异常状态，现场操作人员可以将系统 x 当前的时间序列与历史数据中 z_3 和 z_5 的相似数据段相比较。如果检查发现有十分相似的数据段，则 x_3 的异常能迅速确认，操作人员可根据过去处理相似异常的经验来解决当前的异常情况。

5.5 本章小结

本章提出了三种多变量系统的报警根源分析方法：1) 基于传递熵进行报警根源分析，考虑到报警变量中随机时间延迟的存在以及报警事件之间的相互独立性，定义了报警变量的标准化传递熵和标准化直接传递熵，将其与通过蒙特卡罗检验计算的显著性阈值相比较得到报警变量关联关系；2) 基于一个子节点和多个父节点的贝叶斯网络进行报警根源分析，由观测数据样本递归更新贝叶斯网络的前验概率和后验概率，从具有最大后验条件概率的父节点集中确定发生警报的根本原因。3) 基于定性趋势进行报警根源分析，使用定性趋势直方图应对多变量系统中不等长数据段和定性趋势变化异步性的挑战，将现有的PLSA模型拓展为MFTM模型，从MFTM模型获取数据段，提出一种概率聚类方法自动查找多个过程变量历史序列的定性趋势组合，并且将历史数据段按照各自最有可能所属的定性趋势组合进行归类。本章展示了所提出的方法在数值案例和工业案例中的应用效果。

参考文献

- [1] HU W K, WANG J D, CHEN T W. A new method to detect and quantify correlated alarms with occurrence delays[J]. Computers and Chemical Engineering, 2015, 80:189-198.
- [2] YU W J, YANG F. Detection of causality between industrial alarm data based on transfer entropy[C]// Proceedings of the International Conference on Chemical Engineering, Dhaka: Bangladesh, 2014.
- [3] DUAN P, YANG F, CHEN T W. Direct causality detection via the transfer entropy approach[J]. IEEE Transactions on Control Systems Technology, 2013, 21 (6):2052-2066.
- [4] GANDY A. Sequential implementation of Monte Carlo tests with uniformly bounded resampling risk[J]. Journal of the American Statistical Association, 2009, 104(488):1504-1511.
- [5] Hope A C A. A simplified Monte Carlo significance test procedure[J]. Journal of the Royal Statistical Society Series B-Methodological, 1968, 30(3):582-598.
- [6] PHIPSON B, SMYTH G K. Permutation P-values should never be zero: Calculating exact P-values when permutations are randomly drawn[J]. Statal Applications in Genetics and Molecular Biology, 2010, 9(1).
- [7] KONDAVEETI S R, IZADI I, SHAH S L, et al. Quantification of alarm chatter based on run length distributions[J]. Chemical Engineering Research and Design, 2013, 91(12):2550-2558.
- [8] KOLLER D, FRIEDMAN N. Probabilistic graphical models: principles and techniques[M]. Massachusetts: MIT press, 2009.
- [9] BAUER E, KOLLER D, SINGER Y. Update rules for parameter estimation in Bayesian networks[J]. Eprint Arxiv, 2013:3-13.
- [10] COHEN I, BRONSTEIN A, COZMAN F G. Adaptive online learning of bayesian network parameters[J]. Hewlett Packard Laboratories Technical Report, 2001.
- [11] International Society of Automation. ANSI/ISA-18.2. Management of alarm systems for the process industries[S]. International Society of Automation, 2009.
- [12] Engineering Equipment and Materials Users' Association. EEMUA-191. Alarm systems-A guide to design, management and procurement[S]. Engineering Equipment and Materials Users' Association, 2013.

- [13] VENKATASUBRAMANIAN V, RENGASWAMY R, KAVURI S N. A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies[J]. *Computers and chemical engineering*, 2003, 27(3):13-326.
- [14] BJORKLUND S. A survey and comparison of time-delay estimation methods in linear systems[D]. Sweden: Linkoping University, 2003.
- [15] ZHOU B, YE H. A study of polynomial fit-based methods for qualitative trend analysis[J]. *Journal of Process Control*, 2016, 37:21-33.
- [16] CHARBONNIER S, PORTET F. A self-tuning adaptive trend extraction method for process monitoring and diagnosis[J]. *Journal of Process Control*, 2012, 22(6):1127-1138.
- [17] KEOGH E, CHU S, HART D, et al. Segmenting time series: A survey and novel approach[J]. *Data Mining in Time Series Databases*, 2003, 57:1-22.
- [18] FUCHS E, GRUBER T, NITSCHKE J, et al. Online segmentation of time series based on polynomial Least-Squares approximations[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010, 32(12):2232-2245.
- [19] MAURYA M R, RENGASWAMY R, VENKATASUBRAMANIAN V. Fault diagnosis using dynamic trend analysis: A review and recent developments[J]. *Engineering Applications of Artificial Intelligence*, 2007, 20(2):133-146.
- [20] HAMILTON J D. Time series analysis volume 2[M]. New Jersey: Princeton University Press, 2009.
- [21] HOFMANN T. Unsupervised learning by probabilistic latent semantic analysis[J]. *Machine Learning*, 2001, 42(1-2):177-196.
- [22] DEMPSTER A P, LAIRD N M, RUBIN D B. Maximum likelihood from incomplete data via the EM algorithm[J]. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1977, 39(1):1-22.
- [23] Bahrani M, Sameti H. A new bigram-PLSA language model for speech recognition[J]. *Eurasip Journal on Advances in Signal Processing*, 2010, 2010(1):308-437.
- [24] Blei D M, Ng A Y, Jordan M I, et al. Latent dirichlet allocation[J]. *Journal of Machine Learning Research*, 2003, 3(Jan):993-1022.
- [25] HASTIE T, TIBSHIRANI R, FRIEDMAN J. The elements of statistical learning: data mining, inference, and prediction[M]. Dordrecht: Springer Science and Business Media, 2009.
- [26] Kaufman L, Rousseeuw P J. Finding groups in data: An introduction to cluster analysis[M]. New York: John Wiley, 2009.

- [27] KEOGH E J, PAZZANI M J. Scaling up dynamic time warping to massive dataset[C]// European Conference on Principles of Data Mining and Knowledge Discovery. Springer, Berlin, Heidelberg, 1999.
- [28] BANERJEE A, GHOSH J. Clickstream clustering using weighted longest common subsequences[C]// Proceedings of the web mining workshop at the 1st SIAM conference on data mining, Chicago, 2001.
- [29] CHEN L, ZSU M T, ORIA V. Robust and fast similarity search for moving object trajectories[C]// Proceedings of the 2005 ACM SIGMOD international conference on Management of data, New York: Association for Computing Machinery, 2005.
- [30] PAPARRIZOS J, GRAVANO L. Fast and accurate time-series clustering[J]. ACM Transactions on Database Systems (TODS), 2017, 42(2):1-49.
- [31] BEZDEK J C. Pattern recognition with fuzzy objective function algorithms[M]. New York: Plenum press, 2013.
- [32] WANG X, SMITH K A, HYNDMAN R, et al. A scalable method for time series clustering[J]. Technical Report, 2004:1-20.
- [33] SHUMWAY R H. Time-frequency clustering and discriminant analysis[J]. Statistics and Probability Letters, 2003, 63(3):307-314.
- [34] CHEN K, WANG J D. Design of multivariate alarm systems based on online calculation of variational directions[J]. Chemical Engineering Research and Design, 2017, 122:11-21.

第6章 报警泛滥分析

6.1 引言

报警泛滥是指在短时间内发生大量的报警，超出了操作人员处理报警能力的现象。当报警泛滥出现时，工厂操作人员因为压力过大，可能会忽略许多正在发生的报警。这会导致与报警有关的异常情况被忽视，使得报警系统丧失了该有的预警功能。因此，报警泛滥的研究课题越来越受到学术界研究人员和工业工程师的重视。

对于报警泛滥问题，本章首先提出了一种报警泛滥检测算法，为报警泛滥相似性分析奠定基础。通过分析历史报警泛滥数据，提出一个根据新出现报警变量个数检测报警泛滥的新标准，基于新标准设计了一种报警泛滥检测算法，用于检测历史报警数据中的报警泛滥或判断当前是否发生了报警泛滥。然后，利用报警泛滥相似性分析算法，从历史数据报警泛滥数据库中寻找与当前发生的报警泛滥相似的历史报警泛滥序列，对于找到的相似历史报警泛滥序列，在当前出现的报警序列中选取近期发生的报警事件子序列，并根据具有相同子序列的历史报警泛滥序列来预测即将发生的报警事件。

6.2 报警泛滥检测

报警变量 x_1, x_2, \dots, x_N 的报警数据集在数学上可表示为

$$X(t - M + 1 : t) \triangleq [x_1(t - M + 1 : t), \dots, x_N(t - M + 1 : t)]. \quad (6.1)$$

这里 $x_i(t - M + 1 : t) \triangleq [x_i((t - M + 1)h), \dots, x_i(th)]^T, i \in [1, N]$ 是第*i*个报警变量 x_i 的数据集， t 为一个取值为正整数的采样间隔， N 为样本的数量，也是一个正整数。为了简化表示，假设 h 取值为1s，则 t 的物理解释为即使采样时刻单位为秒，如果报警变量 x_i 进入报警状态，则 x_i 取值变为1，反之则为0。报警泛滥检测的目的就是从式(6.1)中的报警数据集中检测报警变量 X_1, X_2, \dots, X_N 的一个子集在时间周期 $[t - M + 1, t]$ 内是否满足某一个报警泛滥的标准。

6.2.1 两个检测报警泛滥的基本标准

选择合适的度量是设计报警泛滥检测准则的第一步。洪水警报的定义是在10分钟内^[1,2]发生10个或更多的报警。然而，由于对报警一词目前有两种不同的理解，即报警发生和报警变量，前者理解是将报警发生的次数作为检测报警泛滥的度量，后者理解是将报警变量的个数作为检测报警泛滥的度量。

式(2.2)定义报警触发为 x_i 从‘0’变到‘1’。为了捕捉报警触发的时刻，我们提出了 x_i 的另一种

相关形式，即仅取报警发生时刻的值‘1’，取其他时刻的值‘0’，即

$$x'_i(t) = \begin{cases} 1, & \text{如果 } x_i(t-1) = 0 \text{ 且 } x_i(t) = 1 \\ 0, & \text{其它} \end{cases} \quad (6.2)$$

假设以报警触发的次数为检测度量标准，如果在10分钟这一时间段内出现10个或更多的报警，那么就会出现报警泛滥，即

标准A：如果满足条件： $\left(\sum_{i=1}^M \sum_{l=t-599}^t x'_i(l)\right) \geq 10$ ，则十分钟的数据集 $X(t-599:t)$ 中存在一个报警泛滥。

标准A在最近的报警泛滥研究中得到了广泛应用。然而实际情况中干扰报警的存在严重干扰了使用标准A检测报警泛滥的准确性。干扰报警是指即使被工厂操作人员忽略也没有任何影响的报警。相比之下，真正的报警要求操作人员注意或迅速采取行动；否则，与报警变量关联的异常情况将对工厂的安全或效率产生负面影响。抖动报警则是指那些在报警和非报警状态之间快速切换，不需要工厂操作人员注意或行动的报警，是工业报警系统中最常见的干扰报警。抖动报警的存在，意味着大量的报警发生，此时如果使用标准A来检测报警泛滥，极大地可能会出现错误检测的情况。

区分干扰报警和真实报警从本质上来说，是判断一个发生的警报是否与一个异常的情况相关联，而这通常是一个难题。但是，因为抖动报警的持续时间和间隔时间通常是很短或者周期性的，因此在某些情况下，消除抖动报警是可行的。式(2.3)定义报警的持续时间为 $D_a(n)$ ，式(2.4)定义报警的间隔时间为 $I_a(n)$ ，一个经验法则是，每分钟重复超过三次的报警被认为是抖动报警^[4]。因此，我们提出了一种检测抖动报警的规则：“如果一个报警的报警持续时间 $D_a(n)$ 或报警间隔时间 $I_a(n)$ 小于20s，则被视为抖动报警”^[4]。该规则将在之后的算法中用于消除抖振报警。

通常来说，工厂操作人员关心的是一个报警变量是否处于报警状态，而不是同一报警变量在10分钟内重复报警。因此，报警泛滥检测的另一个度量是检测报警状态下报警变量的数量。首先，定义报警变量 x_i 在十分钟的时间周期 $[t-599, t]$ 内总的报警持续时间 $T_{1,i}$ 为

$$T_{1,i} := \sum_{l=t-599}^t x_i(l).$$

其次，定义一个指示变量 \tilde{x}_i 指示 x_i 在10分钟内是否处于报警状态，即

$$\tilde{x}_i = \begin{cases} 1, & \text{如果 } T_{1,i} > 0 \\ 0, & \text{其它.} \end{cases}$$

假设以处于报警状态的报警变量个数作为检测度量标准，那么如果在10分钟的时间区间内有超过10个报警变量处于报警状态，则判断出现报警泛滥，即

标准B：如果 $\left(\sum_{i=1}^M \tilde{x}_i\right) \geq 10$ ，十分钟的数据集 $X(t-599:t)$ 中存在一个报警泛滥。

显然，标准B受到抖动报警的影响比标准A要小得多。然而，它的准确性会受到所谓的“常驻报警”的严重影响。常驻报警是指持续长时间处于报警状态的报警。常驻报警可以是真实报警也可以是干扰报警，对于前者来说，常驻报警指示了一种不正常的状况，并且无法在短期内解决；而对于后者来说，出现常驻报警的主要原因是报警变量配置不正确和报警变量设计与工作状态不一致。对于第二个原因，干扰常驻报警可以通过引入基于状态的报警生成机制来

消除^[4]。干扰常驻报警不能表明任何实际的异常情况，并可能导致错误的报警泛滥检测。而对于真实的常驻报警，其报警时间过长，以至于操作者已经了解了相应的异常情况。因此，也没有必要把这些常驻报警当作每10分钟就需要注意一次的报警。

6.2.2 报警泛滥新的检测标准

虽然在已有文献中有一些处理干扰报警的方法，但在实际情况中，抖动报警和常驻报警并不能完全被消除。因此，上节中提到的两个基本标准往往会导致错误检测报警泛滥的情况。本节根据报警状态中新出现的报警变量个数，提出了报警泛滥检测的新准则，该准则受抖动报警和常驻报警的负面影响较小。首先，定义 $I_1(t)$ 为报警变量 x_1, x_2, \dots, x_N 在时间区间 $[t - 599, t]$ 内的索引集

$$I_1(t) \triangleq \{i\} \text{ 如果 } \left(\sum_{l=t-599}^t x_i'(l) \right) > 0.$$

定义第二个集合 $I_2(t)$ 为报警变量在整个10分钟内都处于报警状态时的索引集，即

$$I_2(t) \triangleq \{i\} \text{ 如果 } \left(\sum_{l=t-599}^t x_i(l) \right) = 600.$$

定义第三个集合 $I_3(t)$ 为报警变量在最近30分钟内都处于报警状态时的索引集，即

$$I_3(t) \triangleq \{i\} \text{ 如果 } \left(\sum_{l=t-1799}^t x_i(l) \right) = 1800.$$

然后，定义报警状态下新出现报警变量的索引集 $I(t)$ 如下：

$$I(t) \triangleq I_1(t) \cup \left(I(t-600) \cap I_2(t) \cap \overline{I_3(t)} \right). \quad (6.3)$$

这里 $\overline{I_3(t)}$ 是 $I_3(t)$ 的补集， $I(t-600)$ 是 $I(t)$ 在前十分钟的对应集合。新的检测标准定义如下：

标准C：如果 $|I(t)| \geq 10$ ，十分钟的数据集 $X(t-599:t)$ 中存在一个报警泛滥。其中 $|I(t)|$ 是集合 $I(t)$ 的基数，代表 $I(t)$ 中元素的个数。

集合 $I(t)$ 由两组报警变量索引组成，第一组包括 $I_1(t)$ 中在最近10分钟时间周期 $[t - 599, t]$ 内进入报警状态的报警变量。这点的设计原理很显然，因为工厂操作员需要分析最近10分钟内发生的报警变量。抖动报警会影响 $I(t)$ ：如果一个报警变量发生了抖动报警，那么这个报警变量的索引会被记录在 $I_1(t)$ ，这可能会导致报警泛滥的错误检测。抖动报警对标准C的负面影响远小于标准A，因为标准A使用的度量是报警发生的数量，而集合 $I(t)$ 关心的是一个报警变量在时间周期内是否进入报警状态。不过，抖动报警对于标准C的影响会在引入延迟计时器之后被部分消除，我们会在下一节介绍这个内容。值得注意的是，第一组报警变量在当前时刻 t 可以处于无报警状态。报警状态设置为每次报警后60秒内自动清除，以便在之后在另一个导致该报警发生的异常状态出现后发出报警。因此，一旦报警变量最近处于报警状态，即使报警变量在时刻 t 处于非报警状态，操作人员也必须找到对应的根本原因。第二组 $[I(t-600) \cap I_2(t) \cap \overline{I_3(t)}]$ 的设计原理是为了从之前的索引集 $I(t-600)$ 中继承一些报警变量。在之前10分钟进入报警状态的报警变量，有可能会在过去10分钟内一直保持报警状态，而且操作人员可能会花费超过10分钟时间来分析这些报警。这些报警变量的索引

由 $I(t-600)$ 与 $I_2(t)$ 的交集中获取, $I_2(t)$ 包含报警变量最近10分钟内处于报警状态时的索引。然而, 索引集 $I_3(t)$ 中的报警变量处于报警状态的时间超过了30分钟, 那么这些报警都应该被操作人员详细了解, 而不需要进一步关注。去除 $I_3(t)$ 中的报警变量可以避免之前提到的常驻报警的负面影响。 $I_3(t)$ 中的30分钟时间窗口由用户选择, 并且应该选为操作人员完全分析一个刚发生的报警所需要的时间。如果时间窗选择较大, 则 $I(t)$ 更有可能受到常驻报警的负面影响, 并会导致更多的报警泛滥误检测。

6.2.3 方法步骤

本节介绍了一种基于标准C的报警泛滥检测算法。给定报警数据集 $X(t-M+1:t)$, 该算法的步骤如下:

1) 通过延迟定时器 (m -采样延迟定时器的数学定义见式(6.4)) 来消除报警持续时间和间隔时间小于20秒的抖动报警;

2) 在第 k 次迭代中的进行如下四个步骤:

a 准备报警数据集 $X(kT-599:kT)$, 其中更新窗口为 $1 \leq T \leq 600$, 窗口大小为用户定义的正整数;

b 根据第 $(k-1)$ 次迭代得到的索引集 $I[(k-1)T]$, 得到公式(6.3)中的索引集 $I(kT)$, 也就是,

$$I(kT) \triangleq I_1(kT) \cup \left((I(k-1)T) \cap I_2(kT) \cap \overline{I_3(kT)} \right).$$

c 计算 $|I(kT)|$ 作为 $I(kT)$ 中的元素数;

d 使用标准C检测是否存在报警泛滥, 当报警泛滥存在时, 检测变量 $x_{AF}(kT)$ 的值变为‘1’, 即

$$x_{AF}(kT) = \begin{cases} 1, & \text{如果 } |I(kT)| \geq 10 \\ 0, & \text{其它} \end{cases}$$

3) 通过人工分析 $I(kT)$ 中报警变量的时间趋势来验证检测结果。

该算法以报警数据集 $X(t-N+1:t)$ 为输入, 输出报警泛滥检测变量 $x_{AF}(kT)$ 和索引集 $I(kT)$ 。通过更新窗口大小 T , 算法可以将报警泛滥的在线检测和离线检测相结合。如果 $T = 1\text{s}$, 那么索引 kT 和采样索引 t 相同, 算法可以用于在线监测报警泛滥。但是对于报警泛滥的离线检测, $T = 1\text{s}$ 的计算时间对于大量的历史数据集来说是不可接受的。因此, 建议使用十分钟内尽可能大的 T 值来加快对报警泛滥的离线检测, 一旦检测到有报警泛滥位于某个时间区间, 再使用 $T = 1\text{s}$ 对这段时间相应的小型数据集进行进一步的分析。

为了减少抖动报警的负面影响, 在步骤2)d中引入 m 样本延迟定时器:

$$x_{AF}^{(m)}(kT) = \begin{cases} 1, & \text{如果 } x_{AF}((k-m+1)T:kT) = 1 \text{ 以及 } x_{AF}^{(m)}((k-1)T) = 0 \\ 0, & \text{如果 } x_{AF}((k-m+1)T:kT) = 0 \text{ 以及 } x_{AF}^{(m)}((k-1)T) = 1 \\ x_{AF}^{(m)}((k-1)T), & \text{其它} \end{cases} \quad (6.4)$$

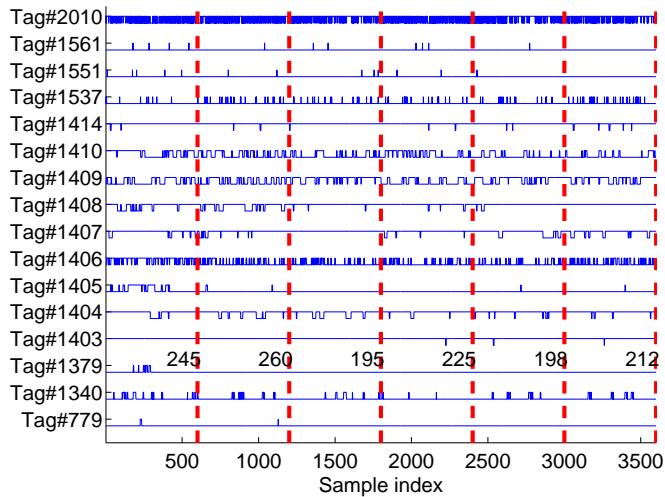


图 6.1 2015年6月1日上午7时至8时报警变量发生报警的时间趋势

因此，如果 $x_{AF}^{(m)}$ 的 m 个连续的采样取值为‘1’，则判断检测出了报警泛滥。抖动报警可能导致 $x_{AF}(kT)$ 在短时间内取值为‘1’，这个问题可以通过延迟计时器解决，因此 $x_{AF}^{(m)}$ 受到抖动报警的影响要小很多。作为牺牲，延迟计时器诱导了 m 秒的时间延迟来检测报警泛滥。这种牺牲对于离线检测报警泛滥来说不是问题。对于在线检测($T = 1s$)， m 的确定以不超过最大允许感应检测延迟为准。以发电机组为例，1分钟的延时通常是可以接受的。因此，之后的案例中我们使用 $m = 60$ 作为默认值。

6.2.4 案例分析

本节通过6个工业实例详细说明了所提报警泛滥检测方法的有效性。

案例1：该实例阐明了抖动报警会严重影响标准A的性能。以中国山东省某300MW发电机组的工业报警系统为例。发电机组的分布式控制系统每秒保存24079个过程变量值。有2226个过程变量被配置为报警变量。后面所有的案例都来自同一个工业报警系统。将标准A应用于2226个报警变量于2015年6月1日早上7点到8点的1小时数据集，我们发现有16个报警变量在一小时内触发了报警。报警变量的时间趋势，如图6.1所示。在6个不重叠的连续10分钟时间段内，警报发生的次数分别为245、260、195、225、198和212，以分隔10分钟时间段的垂直虚线表示。报警次数远远大于标准A中的检测阈值10，表示检测到了报警泛滥的存在。然而，从图6.1中这些报警变量的时间趋势可以看出，抖动报警明显存在，因为16个报警变量中大多数的报警持续时间 T_0 和报警间隔时间 T_1 都非常短。因此，抖动报警导致使用标准A时出现了报警泛滥的误检测。

在实际情况中，不可能消除所有的抖动报警。图6.2给出了2015年6月8日上午10点到11点1小时内9个报警变量发生报警的时间趋势，说明了这一困难。在6个不重叠的连续10分钟时间段内，报警次数分别为6、14、5、11、3和0。应用标准A检测出报警泛滥在两个时间段内存在，一个来自上午10点10分到10点20分，还有一次是上午10点半到10点20分晚10点。然而，从所涉

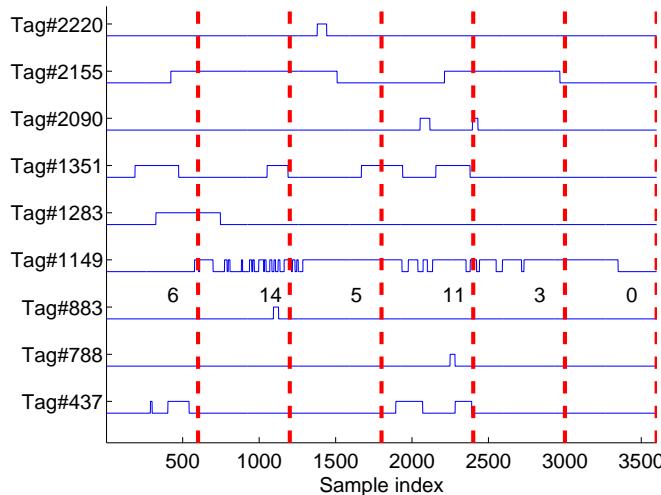


图 6.2 2015年6月8日上午10时至11时报警变量发生报警的时间趋势

表 6.1 案例3中各索引集中的元素

t	$I_1(t)$	$I(t - 600) \cap I_2(t) \cap \bar{I}_3(t)$	$I(t)$
600	{1340 2018}	{ }	{1340 2018}
1200	{ }	{ }	{ }
1800	{999 1349 1364 2018 2033}	{ }	{999 1349 1364 2018 2033}
2400	{2010 2018 2090}	{1349 1364}	{1349 1364 2010 2018 2090}
3000	{2018}	{1349 1364 2010}	{1349 1364 2010 2018}
3600	{1283 2018}	{2010}	{1283 2010 2018}

及报警变量的时间趋势来看，标签#1149的报警变量是导致报警泛滥误检测的主要原因。该报警变量触发了多次报警，且报警时间较短，这导致报警次数超过了报警泛滥的检测阈值10。因此，实际情况中并没有发生报警泛滥。

案例2：该案例阐明了常驻报警会严重影响标准B的性能。使用案例1中的第一个报警数据集，即2015年6月1日上午7时至8时，连续6个不重叠的10分钟时间段内处于报警状态的报警变量分别为227、225、223、223、224和222。如果使用标准B，则洪水警报在整个1小时内都存在。然而，这1小时中处于报警状态的报警变量有210个，包含了干扰报警和真实报警。以常驻干扰报警为例，电动给水泵的反馈控制回路的手动模式中，其一个报警变量一直处于报警状态，如图6.3(a)所示。然而，报警状态并没有指示任何异常情况。电厂常用的做法是在正常运行时使用蒸汽驱动给水泵，只有在启动阶段或紧急情况下才使用电动给水泵。因此，如图6.3(b)和(c)所示，电给水泵的电流和进口流量均已处于关机状态。作为对比，主给水流量处于正常状态，如图6.3(d)所示。由此可见，电动给水泵反馈控制回路处于手动状态是正常的，这种长期存在的干扰报警显然会导致报警泛滥的误检测。

图6.4(a)给出了一个真实常驻警报的例子。即，在NH3对NOX的选择性催化还原反应中，上下层之间的压力差超过140pa的报警变量在1小时内都处于报警状态。图6.4(b)-(d)分别展示了

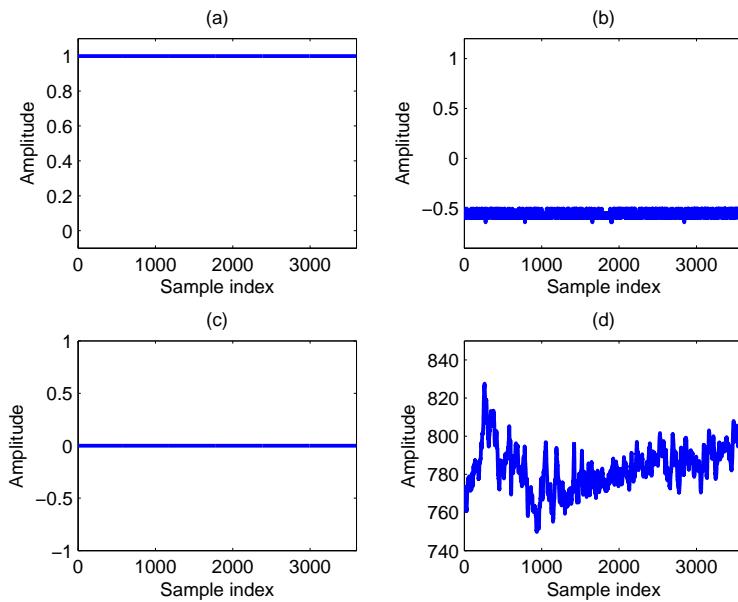


图 6.3 处于手动模式时四个报警变量的时间趋势

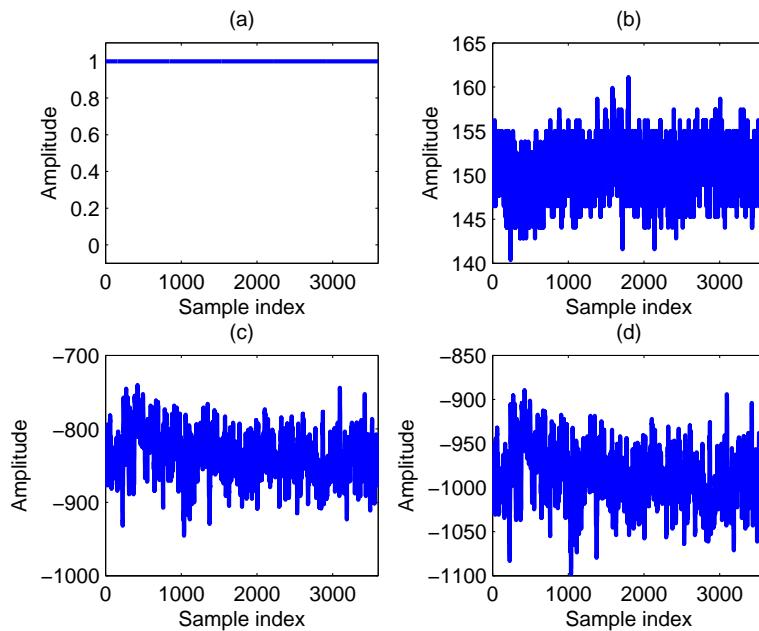
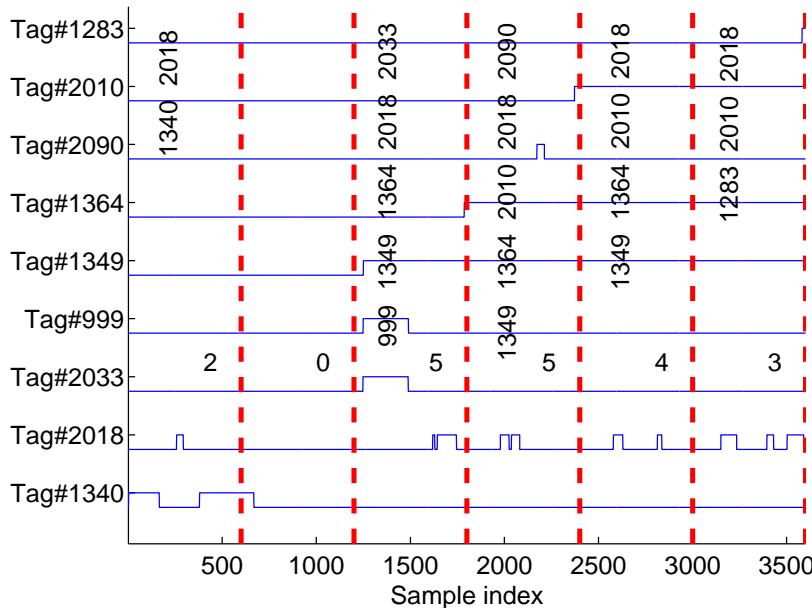


图 6.4 使用NH3对NOX的选择性催化还原反应中四个报警变量的时间趋势

计算的压力差和两个层的压力测量，高压力差通常表明有飞灰堵塞了两个层。然而，吹灰消除异常的操作只能在下一轮设备维护时实施，维护周期通常是8小时。预料之中，该报警变量长时间处于报警状态，如果每10分钟就把它作为操作人员需要关注的对象是无意义的。

图 6.5 索引集 $I(t)$ 中的报警变量的时间趋势

案例3：该案例阐明了计算索引集 $I(t)$ 和使用标准C的基本原理。图6.5为2015年7月1日12时至13时这一小时间指数集 $I(t)$ 中报警变量的时间变化趋势。连续六个不重叠的10分钟时间区间中 $I(t)$ 的基数分别为2、0、5、5、4和3。图6.5展示了每个时间区间中集合 $I(t)$ 的元素。表6.1也展示了连续六个不重叠的10分钟时间区间中集合 $I_1(t)$, $\left(I(t-600) \cap I_2(t) \cap \overline{I_3(t)}\right)$ 和 $I(t)$ 的元素。在第一个10分钟的时间段内，集合 $I(t-600)$ 和 $I_3(t)$ 被初始化为空集。因此，索引集 $I(t=600)$ 包含两个发生过报警的报警变量，即标记名为#1340和#2018的报警变量。在第二个10分钟的时间段内，#1340和#2018都回到非报警状态，其他报警变量均不处于报警状态，因此 $I(t=1200)$ 为空。在第三个10分钟的时间段内， $I(t=1800)$ 包含5个发生报警的报警变量，即#999、#1349、#1364、#2018 和#2033。在第四个10分钟时间段内，标记名为#2010、#2018 和#2090的三个报警变量发生报警，因此它们是集合 $I_1(t=2400)$ ，而标记名为#1349 和#1364的报警变量是从上一个10分钟的集合 $\left(I(t-600) \cap I_2(t) \cap \overline{I_3(t)}\right), t = 2400$ 中继承的，并且在整个10分钟内都处于警报状态。因此，索引集 $I(t=2400)$ 中有5个元素。在第5个10分钟的时间段内，集合 $I_1(t=3000)$ 有一个标记名为#2018的报警变量发生了报警，标记名为#1349、#1364、#2010 的报警变量处于报警状态，是从集合 $I(t=2400)$ 继承而来。因此，集合 $I(t=3000)$ 由这四个报警变量组成。在最后一个10分钟的时间段内，标记名为#1349和#1364的两个报警变量已经处于报警状态30分钟，因此它们不值得操作人员注意并被排除出集合 $I(t=3600)$ 。

案例4：基于标准C的报警泛滥检测算法应用于2226个报警变量在2015年6月25日上午10时至11时的1小时数据集。将更新窗口大小 T 设置为1s，这样索引 kT 可以替换采样索引 t 。(6.4)中延时计时器的参数 m 设为60。为了与标准A和B进行公平的比较，所有准则都应用于原始报警数据集，而没有通过延迟计时器去除抖动报警。图6.6(a)显示了基数 $|I(t)|$ 和检测阈值10

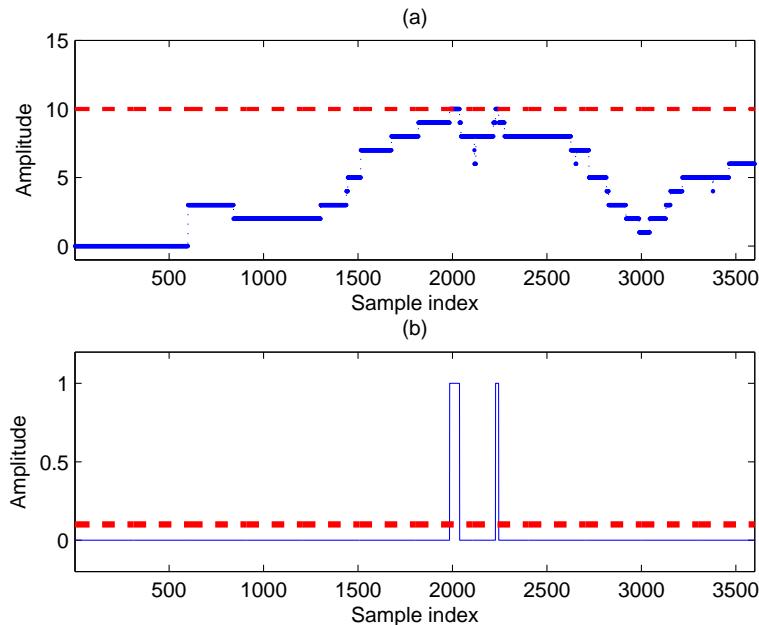


图 6.6 从2015年6月25日上午10点到上午11点之间的报警泛滥检测

（注意，前599个 $|I(t)|$ 样本被初始化为零）。图6.6(b)中的变量 $x_{AF}(t)$ （实线）在两个时间区间 $t \in [1985, 2037]$ 和 $t \in [2228, 2244]$ 中的值为‘1’。这两个时间区间都小于 $m = 60s$ ，因此，图6.6(b)中由延迟计时器产生的 $x_{AF}^{(m)}(t)$ （虚线）表示没有报警泛滥出现。

如图6.7所示的索引集 $I(t)$ ，其报警变量的时间趋势可视化验证了这一结果。其中红色虚线垂线表示10分钟时间段 $t \in [1368, 1985]$ 的起始和结束位置。索引集 $I(t = 1985)$ 中有10个报警变量，即标记名为#833、#844、#848、#1149、#1353、#1359、#1406、#1407、#1537 和#1790的报警变量。绿色虚线表示第二时间段 $t \in [1629, 2228]$ 的开始和结束位置，索引集 $I(t = 2228)$ 中的报警变量为#833、#844、#848、#1147、#1149、#1353、#1406、#1407、#1537 和#1790。然而，标记名为#1537的报警变量显然是抖动报警，因为它快速地在报警和非报警状态之间切换。标记名为#1406 和#1407的报警变量大部分时间处于报警状态，偶尔有小的报警间隔，所以它们是常驻报警。因此，拥有真实报警的报警变量数量不足10个，所以报警泛滥并不存在。这些干扰报警导致 $x_{AF}(t)$ 在短时间内等于‘1’，但 $x_{AF}^{(m)}(t)$ 并不受干扰报警的影响。

作为比较，标准A和标准B也应用于报警泛滥的检测。图6.8(a)和(b)分别表示了报警发生的次数和在报警状态下报警变量的数量，对应图6.6(a)中的 $|I(t)|$ 。注意，图6.8(a)和(b)中两个度量的前599个样本被初始化为零。标准A检测出了报警泛滥，如图6.7所示，这是一个由于抖动报警#1537 的存在导致的报警泛滥误检测。标准B也检测出了报警泛滥，这是因为在整个1小时内有10个报警变量处于报警状态。

案例5：图6.9(a)展示了2226个报警变量于2015年6月15日上午10点到11点的1小时数据集的报警洪水检测结果。更新窗口大小 T 设置为1s，延时计时器的参数 m 设为60。检测变量 $x_{AF}^{(m)}(t)$ 在时间区间 $t \in [2530, 3023]$ 中取值为‘1’。通过图6.10中索引集 $I(t = 2530)$ 中所有报警变量在所有时刻的时间趋势来验证报警泛滥的检测。这些报警变量的物理描述，如表6.2所

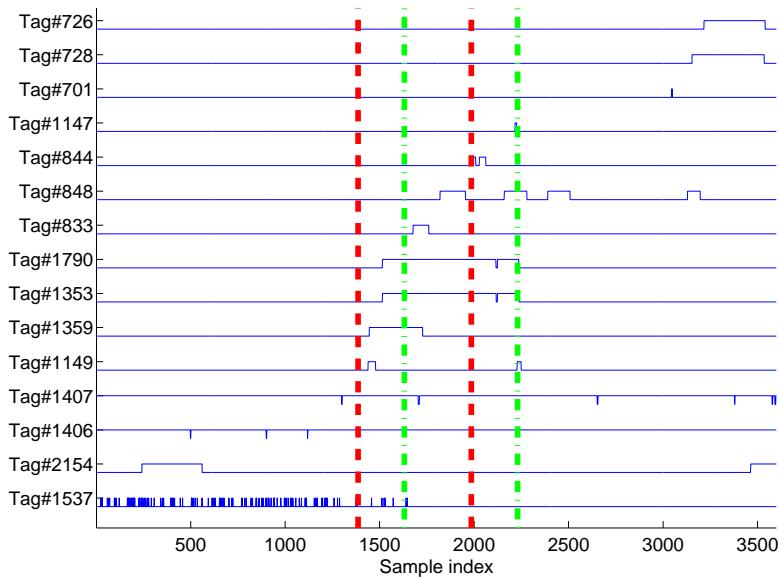
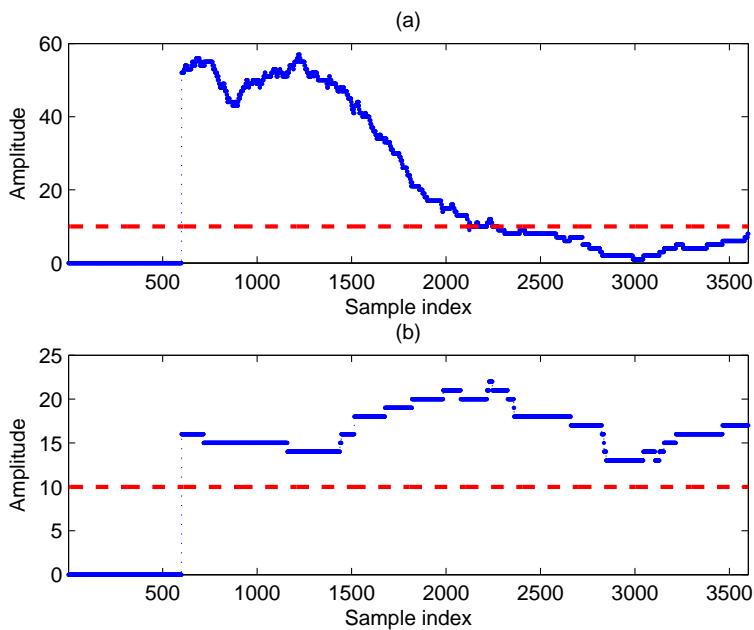
图 6.7 索引集 $I(t)$ 中的报警变量的时间趋势

图 6.8 从2015年6月25日上午10点到上午11点之间的报警泛滥检测

示。垂直虚线表示10分钟时间区间 $t \in [1931, 2530]$ 的起始和结束位置。在此时间段内，集合 $I(t = 2530)$ 中有13个报警变量，即有报警泛滥发生。

检测到的报警泛滥主要是由磨煤机C出现异常引起的，标记名为#517和#514的两个报警变

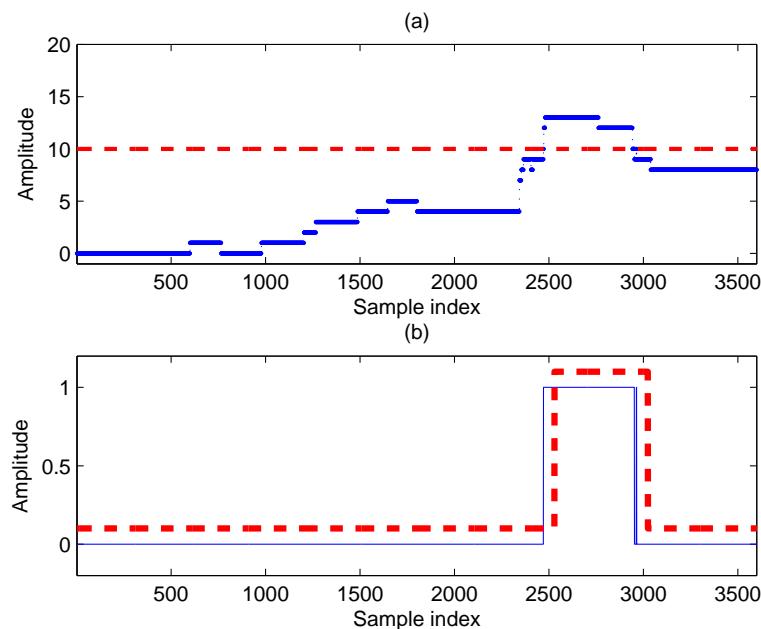
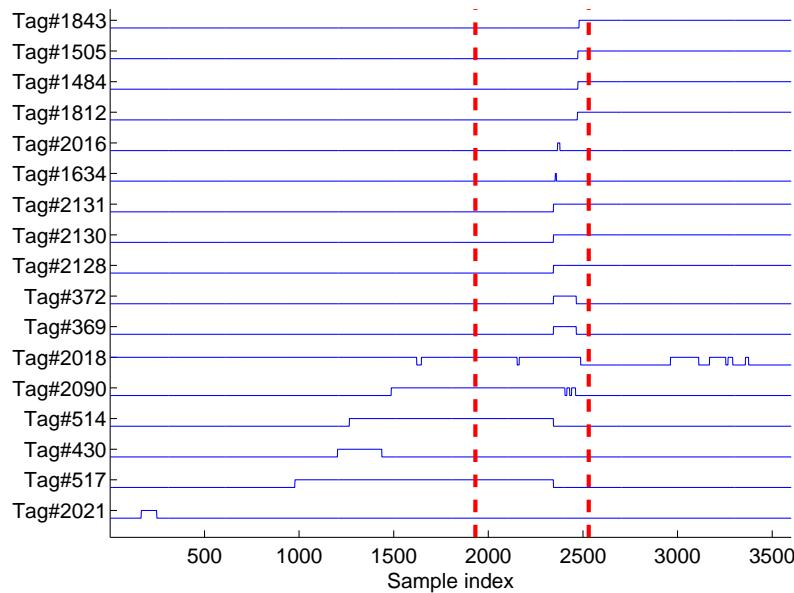


图 6.9 2015年6月15日上午10点到上午11点之间的报警泛滥检测

图 6.10 索引集 $I(t)$ 中报警变量的时间趋势

量表明了该异常是由磨煤机驱动端的轴承温度过高引起的。两个温度传感器的测量结果，如图6.11(a)所示。标记名为#517 和#514的两个报警变量在温度超过51°C时进入报警状态。如果其中一个温度超过55°C，另一个温度超过60°C，则另一个标记名为#2090的相关报警变量会进入

表 6.2 2015年6月15日上午10点到上午11点之间的报警泛滥所包含的报警变量

x_{tag}	x_{desp}
2021	反应器A出灰口NOx浓度高
517	磨煤机C驱动端轴承温度#2高
430	磨煤机A加煤能力低
514	磨煤机C驱动端轴承温度#1高
2090	输煤系统异常
2018	反应器A入灰口NOx浓度高
369	磨煤机C1主风扇手动模式
372	磨煤机C2主风扇手动模式
2128	磨煤机C紧急停机
2130	磨煤机C1紧急停机
2131	磨煤机C2紧急停机
1634	炉膛气压异常
2016	反应器B层压差高
1812	发电输出自动控制手动模式
1484	锅炉主控手动模式
1505	主风机控制手动模式
1843	氧气控制手动模式

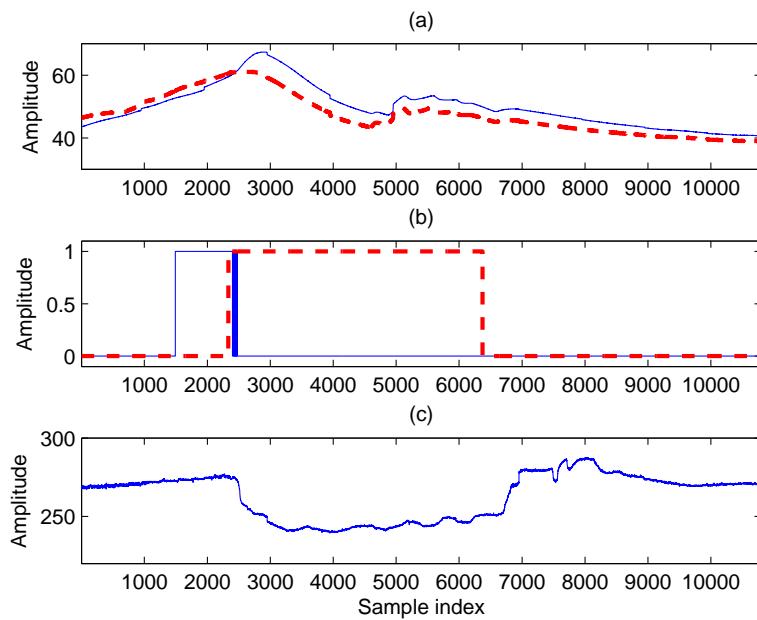


图 6.11 相关报警变量的时间趋势

报警状态。由于工厂操作人员忽略了这一异常，过了大概22分钟后，由于两个传感器检测的温度都超过了60°C，磨煤机C触发了安全保护被强制紧急停机（标记为#2128的报警变量），如

图中标记名为#2128、#2130和#2131的报警变量所示。并导致一系列跟磨煤机C相关的报警变量都进入了报警状态（#369、#372、#1484、#1505、#1634、#1812 和#1843）。如图6.11(c)所示，由于磨煤机C紧急停机，实际发电量由275 MW骤降至240 MW。这种异常工况给发电机组造成了重大的经济损失。

案例6：将检测算法应用于2015年全年共2226个报警变量的一年数据集。更新窗口大小 T 被设置为600秒。该算法先利用 $m = 20$ 的 m 采样延迟定时器来消除报警数据集中的抖动报警。将三个标准分别应用于经过预处理后的同一报警数据集。图6.12(a)表示的是标准C计算的基数 $|I(kT)|$ 。图6.12(b)和(c)分别给出了标准A的报警发生次数和标准b的进入报警状态时报警变量的数量，与 $|I(kT)|$ 相对应。为避免数值过大，图6.12中的所有值都是以10为底的对数，比如图6.12(a)中的 $\log_{10}(|I(kT)|)$ 。

图6.12中标准B的度量超过了检测阈值1（在进行对数操作之后）的次数是52047次，可以看出常驻报警的存在导致标准B完全失效。而标准A和C的度量超过检测阈值1的次数分别是1790次和396次。可以看出即使报警数据集已经通过 m 采样延迟定时器进行了预处理，也不可能去除所有的抖动报警。因此，剩余的抖动报警严重影响了以报警发生的次数作为度量的标准A，并导致许多报警泛滥的误检测。相比之下，标准C的度量索引集 $I_1(kT)$ 受到抖动报警的影响要小很多。因此，标准C的检测性能比标准A要好。

6.3 报警泛滥相似性分析

报警泛滥由一系列按时间顺序排序的报警事件组成，每个报警事件通常包括各种属性，如标记名、报警标识符、时间戳、报警优先级和过程描述^[5]。标记名是与报警事件相关联的报警变量（包括模拟变量和数字变量）的标签；报警标识符描述了报警事件的类型，例如PVHI表示模拟变量超过了高报警阈值。报警标记名和报警标识符共同构成独一无二的报警标签。例如，报警标识符PVHI适用于标签为1PT01的报警变量，则1PT01.PVHI是一个报警标签。时间戳记录了报警发生或报警清除的时刻。报警标记名 x_{tag} 和报警优先级 x_{pri} 已经在第2章中定义。

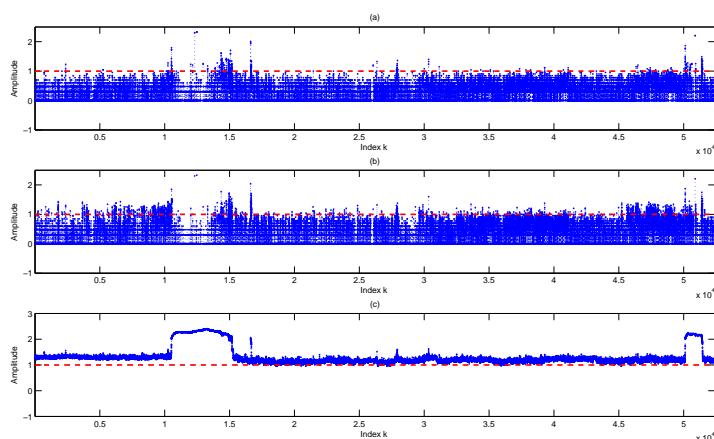


图 6.12 2015年全年的报警数据集使用标准A、B和C计算的度量以及相应的检测阈值

因此，报警泛滥 X_h 可以表示为

$$X_h = [x_{h_1}, x_{h_2}, \dots, x_{h_n}], \quad (6.5)$$

这里 n 为发生的报警事件的总数，其中 x_{h_i} 为 X_h 中第 i 个报警事件。我们用一个具有三个属性的元组来表示 x_{h_i} ，

$$x_{h_i} = (e_i, t_i, p_i), \quad (6.6)$$

其中 e_i 为 x_{h_i} 的报警标签， t_i 和 p_i 分别为对应的时间戳和报警优先级。为了便于计算，我们采用数字形式来表示报警标签 e_i ，所以我们将所有不同的报警标签映射为数字符号，并且构造了一个数字字母表，

$$\Sigma = \{1, 2, \dots, V\}. \quad (6.7)$$

其中 V 表示报警标签的总数。显然，字母表 Σ 和所有得报警标签的关系是一一映射的。

表6.3给出了一个报警泛滥的工业例子：第一列按时间顺序列出报警标签，第二列给出报警标签的数字符号，第三列表示时间戳，最后一列表示报警优先级。

如6.2节中所述，在报警泛滥分析中，抖动报警可能导致对报警泛滥的错误检测。因此，在报警泛滥的先决步骤是去除抖动报警。这里使用6.2节中提到的 m -采样延迟定时器来消除抖动报警（ m 设置为20）。

报警优先级表示报警标签的相对重要性，这个重要性是通过评估忽略该报警导致的后果严重性以及处理该报警相关异常工况的允许响应时间来确定的^[6]。如表6.4所示，优先级通常分为三到四个等级，并且有不同的名称。方法^[6]建议按从低到高三个优先级分配的报警标签比例为80%/15%/5%。较高的优先级通常配置给较少数量的报警标签，表明它们很少发生，但是一旦出现就表明存在严重的异常工况。相比之下，大多数与不太严重的异常情况相关的报警变量被分配了较低的优先级。因此，提出的报警泛滥序列相似性比对算法理应对优先级较高的报警事件更敏感。

表 6.3 一个报警泛滥的工业例子

报警标签	符号(e_i)	时间戳(t_i)	优先级(p_i)
T09.PVLO	9	10/5/2013 02:20:01	低
T05.PVHI	5	10/5/2013 02:20:55	低
T06.PVLO	6	10/5/2013 02:22:02	低
T02.PVHI	2	10/5/2013 02:22:42	高
T01.LOLO	1	10/5/2013 02:23:35	高
T03.PVHI	3	10/5/2013 02:24:22	低
T04.PVLO	4	10/5/2013 02:25:11	低
T03.PVHI	3	10/5/2013 02:27:02	低
T04.PVLO	4	10/5/2013 02:27:03	低
T07.PVLO	7	10/5/2013 02:27:03	紧急
T04.PVLO	4	10/5/2013 02:29:05	低
T05.LOLO	5	10/5/2013 02:30:05	低

表 6.4 报警系统中的报警优先级

报警优先级	列表1	列表1	列表1
优先级1 (p_1)	紧急	紧急	危险
优先级2 (p_2)	高	高	警告
优先级3 (p_3)	中	低	提醒
优先级4 (p_4)	低		

我们使用相似性分数 $s(x_{h_i}, y_{h_j})$ 来测量报警泛滥 X_h 和 Y_h 中两个报警 $x_{h_i} = (e_i^x, t_i^x, p_i^x)$ 和 $y_{h_j} = (e_j^y, t_j^y, p_j^y)$ 的相似性。根据 x_{h_i} 和 y_{h_j} 的报警标签是否相同, $s(x_{h_i}, y_{h_j})$ 的计算公式为

$$s(x_{h_i}, y_{h_j}) = \begin{cases} \phi(p_i^x) (\text{或 } \phi(p_j^y)) & \text{如果 } e_i^x = e_j^y \\ \mu & \text{如果 } e_i^x \neq e_j^y \end{cases}, \quad (6.8)$$

这里匹配分数 $\phi(\cdot)$ 是报警优先级的一个正值函数, 而不匹配分数 μ 是一个负值。对于一个按重要性降序排列的优先级列表 $\{p_1, p_2, \dots, p_L\}$, 匹配分数 $\phi(\cdot)$ 根据 l 单调递减,

$$\phi(p_l) = \alpha + \beta(L - l), \quad (6.9)$$

其中, α 和 β 是两个正值常数。值得注意的是, x_{h_i} 可以和间隙“-”匹配而不是与不同的报警标签匹配。在这种情况下, 基本相似度得分等于一个恒定的惩罚得分 $\delta < 0$,

$$s(x_i, -) = \delta. \quad (6.10)$$

为了更好的选择让报警标签与间隙匹配而不是与不同的报警标签匹配, 不匹配分数 μ 和惩罚得分 δ 应满足以下条件:

$$\mu < 2\delta < 0. \quad (6.11)$$

如表6.5所示, 我们使用式(6.9)中的线性匹配得分函数, 其中 $\alpha = 3$, $\beta = 1.5$, $\mu = -2.5$, $\delta = -1$ 。匹配分数、不匹配分数和惩罚得分的选择并不是唯一的。匹配分数 $\phi(p_l)$ 基于不同的报警优先级。例如, 考虑到“紧急”、“高”和“低”这三个优先级, 表6.5 中的匹配分数分别是6、4.5和3, 这里匹配分数 $\phi(p_l)$ 可以根据专家对不同报警优先级在报警管理中相对重要性的评估进行更改。通常只要满足 $\mu < 2\delta < 0$ 和 $\phi(p_l) > |\mu|, l = 1, 2, \dots, L$, $\phi(p_l)$, μ 和 σ 的选择对结果没有显著影响。

表 6.5 相似性分数

类别	符号	分数
与 p_l 匹配	$\phi(p_l)$	$3 + 1.5(L - l)$
不匹配	μ	-2.5
惩罚得分	δ	-1

6.3.1 基于BLAST的报警序列局部比对算法

BLAST算法可以查找到待查询序列和已知序列之间相似的序列段。BLAST算法主要由种子搜索和延伸这两个步骤组成。在种子搜索阶段，算法把需要比对的序列按照一定长度拆分为多个连续的段，并把序列中所有元素的所有组合构建成一个查找表。通过在查找表中搜索，可以快速定位待查询和已知序列之间的相似片段。相似度得分超过阈值的片段被保留为种子，其他的则被丢弃。接下来，在延伸步骤中，所有种子在两个方向上进行扩展，直到相似度得分低于一个截止阈值为止。相似度评分大于某一阈值的比对结果称为高分片段对。所有种子中得分最高的高分片段对被视为待查询序列和已知序列之间最相似的片段。这里展示了一个报警序列的案例以便更好地解释BLAST算法。

案例1：这里有四个不同的报警 $\Sigma = \{1, 2, 3, 4\}$ ，两个待匹配的报警序列分别为 $[3, 2, 1, 4, 3, 2, 2]$ 和 $[3, 4, 2, 1, 4, 2]$ 。首先，按照长度 $k = 3$ 把这四个报警的 $4^3 = 64$ 种组合构建成查找表。然后把第一个序列拆分为 $[3, 2, 1]$, $[2, 1, 4]$, $[1, 4, 3]$, $[4, 3, 2]$ 和 $[3, 2, 2]$ 。第二个序列则被拆分为 $[3, 4, 2]$, $[4, 2, 1]$, $[2, 1, 4]$ 和 $[1, 4, 2]$ 。最后通过对查找表进行索引，搜索两个序列中的哪些段具有相同的索引来确定匹配的对。这里两个序列中完全匹配的对，即 $[2, 1, 4]$ ，被视为种子。在延伸步骤中，该种子向两个方向延伸，并且使用了间隙比对方法。通过允许一个间隙比对，两个报警序列的最佳匹配如下：

$$\begin{array}{ccccccc} 3 & - & 2 & 1 & 4 & 3 & 2 & 2 \\ | & & | & | & | & & | \\ 3 & 4 & 2 & 1 & 4 & - & 2 & - \end{array} \quad (6.12)$$

其中‘-’表示插入间隙，‘|’表示一个匹配对。

为了使BLAST算法更适合应用于报警泛滥的相似性分析上，我们对BLAST进行了如下的重大修改，以获得较高的计算效率和更好的匹配结果。首先，考虑到报警标签种类繁多，我们预先设计了一套基于报警事件集合的匹配机制，排除不相关的报警泛滥和报警标签。其次，常规的BLAST算法需要一个预定义的查找表，其中包含所有可能固定大小的报警事件组合。但如果太多唯一的序列段，以此进行查找表的构造和查询就变得不可行。为了解决这一问题，我们对种子搜索阶段进行了改进，以便搜索所有长度不同的匹配对。基于此，我们采用了基于优先级的评分方案，使算法对较高优先级的报警更加敏感。最后，常规的BLAST算法没有考虑到时间信息，而报警事件包含了时间戳的信息。因此，在延伸阶段，我们提出了一种时间模糊容限策略，使算法对几乎同时发生的报警事件之间的顺序不那么敏感。

6.3.2 基于报警事件集合的预匹配机制

为了排除不相关的报警泛滥和报警标签，我们提出了一种基于报警事件集合的预匹配机制。首先，在两条报警泛滥序列 $X_h = [x_{h_1}, x_{h_2}, \dots, x_{h_M}]$ 和 $Y_h = [y_{h_1}, y_{h_2}, \dots, y_{h_N}]$ 之间找到常见的报警标记，两个二元索引向量被表示为

$$\left\{ \begin{array}{l} A_X = [a_1^x \ a_2^x \ \cdots \ a_M^x] \\ A_Y = [a_1^y \ a_2^y \ \cdots \ a_N^y] \end{array} \right., \quad (6.13)$$

其中

$$a_i^x = \begin{cases} 1 & \text{如果 } \exists j \in \{1, 2, \dots, N\} \text{ 使得 } e_j^y = e_i^x \\ 0 & \text{其它} \end{cases}, \quad (6.14)$$

$$a_i^y = \begin{cases} 1 & \text{如果 } \exists j \in \{1, 2, \dots, M\} \text{ 使得 } e_j^x = e_i^y \\ 0 & \text{其它} \end{cases}, \quad (6.15)$$

这里 e_i^x 和 e_i^y 分别是 X_h 中 x_{h_i} 的报警标签以及 Y_h 中 y_{h_j} 的报警标签。基于索引向量 A_X 和 A_Y , 定义基于集合的相似度指标为

$$S_{set}(X, Y) = \frac{\sum_{i=1}^M (a_i^x \times \phi(p_i^x)) \sum_{i=1}^N (a_i^y \times \phi(p_i^y))}{\sum_{i=1}^M (\phi(p_i^x)) \sum_{i=1}^N (\phi(p_i^y))}. \quad (6.16)$$

在不考虑时间顺序的情况下, 我们采用基于集合的相似度指标 $0 \leq S_{set}(X, Y) \leq 1$ 来衡量两个报警泛滥序列是否相似。也就是说, $S_{set}(X, Y) = 1$ 表示报警泛滥 X_h 和 Y_h 有相同的报警事件集合, $S_{set}(X, Y) = 0$ 表示 X_h 和 Y_h 没有相同的报警事件。如果两次报警泛滥之间有更多相同的报警标签, 则 $S_{set}(X, Y)$ 的值更接近于 1。

$S_{set}(X, Y)$ 的计算基于报警事件的集合而不是报警序列, 是选择合适的报警泛滥进行后续相似度分析的一个初始步骤。如果 $S_{set}(X, Y) \leq \gamma$, 则认为 X_h 与 Y_h 之间相似性指数太低, 其中 γ 是一个用户选择的参数。这种情况下不需要进行下一步的报警泛滥序列比对。最简单的选择是 $\gamma = 0$, 表示如果两个报警泛滥没有相同的报警事件, 序列比对将不进行。当报警泛滥数量较多时, 可以设置 $\gamma > 0$ 以避免在相同报警事件较少的报警泛滥序列比对中浪费时间。

其次, 即使两个报警泛滥 X_h 和 Y_h 之间的相似度指数 $S_{set}(X, Y)$ 大于 γ , 它们之间仍然可能有许多不同的报警标签。对随后的序列比对中计算这些不常见的报警标签是没有意义的。因此, 我们把原始报警泛滥序列 X_h 和 Y_h 中所有无关的报警去除, 只保留共同的报警。根据式(6.13)中的 A_x 和 A_y , 我们把 $a_i^x = 1$ 对应的 x_{h_i} 和 $a_j^y = 1$ 对应的 y_{h_j} 从 X_h 和 Y_h 中提取出来组成两个较短的序列 $\tilde{X} = [\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_{\tilde{M}}]$ 和 $\tilde{Y} = [\tilde{Y}_1, \tilde{Y}_2, \dots, \tilde{Y}_{\tilde{N}}]$ 。这里 $\tilde{M} = \sum_{i=1}^M a_i^x$ 和 $\tilde{N} = \sum_{i=1}^N a_i^y$ 分别表示 X 和 Y 中相同报警事件的数量。显然, $S_{set}(\tilde{X}, \tilde{Y}) = 1$ 。

案例2: 两个不同长度的报警序列 X_h 和 Y_h ($M = 12, N = 14$) , 如表6.6 所示。通过比较报警符号, 两个索引向量 A_X 和 A_Y 为

$$\begin{cases} A_X = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \\ A_Y = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0] \end{cases}. \quad (6.17)$$

可以发现 A_X 中的所有元素都为 1, 表示 X_h 中的所有报警标签都可以在 Y_h 中找到。而因为数字符号为 8 的报警标签在 X_h 中不存在, 所以 A_Y 中的第 10 个、第 12 个和第 14 个元素为 0。表 6.6 中的优先级有三级, 分别是“紧急 (p_1)”、“高 (p_2)”和“低 (p_3)”。根据表 6.5 中的基于优先级的相似性分数计算, 分配给每个优先级的分数分别为: $\phi(p_1) = 3 + 1.5 \times (3 - 1) = 6$ 、 $\phi(p_2) = 3 + 1.5 \times (3 - 2) = 4.5$ 和 $\phi(p_3) = 3 + 1.5 \times (3 - 3) = 3$ 。使用式(6.16)可以计算出相似度指标为 0.8125。因此, 基于报警事件集合的预匹配机制表明 X_h 和 Y_h 是非常相似的报警泛滥序列。通过把 Y_h 中的报警标签 8 去除, 我们得到对应的两条子序列 \tilde{X} 和 \tilde{Y} 分别为 $\tilde{X} = [9, 5, 6, 2, 1, 3, 4, 3, 4, 7, 4, 5]$ 和 $\tilde{Y} = [9, 5, 2, 1, 3, 4, 7, 3, 4, 9, 6]$ 。这两个子序列将在下面的小节中用于报警泛滥序列的相似性比对。

表 6.6 包含时间和优先级信息的报警泛滥序列 X_h 和 Y_h

序列 X_h			序列 Y_h		
符号	时间戳	优先级	符号	时间戳	优先级
9	00:00:01	低	9	11:00:01	低
5	00:00:55	低	5	11:01:11	低
6	00:02:02	低	2	11:01:59	高
2	00:02:42	高	1	11:02:18	高
1	00:03:35	高	3	11:03:11	低
3	00:04:22	低	4	11:04:01	低
4	00:05:11	低	7	11:06:32	紧急
3	00:07:02	低	3	11:06:33	低
4	00:07:03	低	4	11:06:33	低
7	00:07:03	紧急	8	11:07:16	低
4	00:09:05	低	9	11:08:45	低
5	00:10:05	低	8	11:09:22	低
			6	11:09:56	低
			8	11:09:57	低

因为物理连接的报警标签通常几乎同时出现，但在不同的报警泛滥中，它们可能有不同的时间顺序。因此，我们提出了一种包含报警时间信息的策略，以容忍这些几乎同时发生的报警在一定时间上的模糊性。在算法中采用的策略如下：为了重新定义式(6.28)中的基础相似性分数 $s(x_i, y_j)$ ，我们定义了两个向量，即时间距离向量 \mathbf{d}_i 和时间权重向量 \mathbf{w}_i 。 \tilde{X} 的第 i 个报警事件 x_i 的时间距离向量 \mathbf{d}_i 为

$$\mathbf{d}_i = [d_{i1} \ d_{i2} \ \cdots \ d_{iV}]^T, \quad (6.18)$$

其中

$$d_{iv} = \min_{1 \leq k \leq m} \{|t_i - t_k| : e_k^x = v\}. \quad (6.19)$$

这里 e_k^x 为 \tilde{X} 的第 k 个报警事件 x_k 的报警标签， v 是大小为 V 的字母表 Σ 中的数字符号。每个 d_{iv} 代表报警标签为 v 的第 i 个报警与最近的报警之间的时间间隔。时间权重向量 \mathbf{w}_i 为

$$\mathbf{w}_i = [w_{i1} \ w_{i2} \ \cdots \ w_{iV}]^T, \quad (6.20)$$

其中 $w_{iv} = f(d_{iv})$ 。 \tilde{X} 的权重函数 $f(\cdot)$ 可以被选择为标度高斯函数：

$$f(d_{iv}) = e^{-\frac{d_{iv}^2}{2\sigma^2}}, \quad (6.21)$$

\tilde{Y} 的对应部分为

$$f(d_{iv}) = \begin{cases} 1 & \text{if } d_{iv} = 0 \\ 0 & \text{if } d_{iv} \neq 0 \end{cases}. \quad (6.22)$$

式(6.21)中的标准偏差 σ 为用户选择参数。 σ 的值决定了多大的时间差可以被容忍。一般来说，同步发生的报警，它们之间的时间差在几秒钟的水平。如果使用较大的 σ 值，则认为两个时间

表 6.7 考虑时间戳信息的 $\tilde{Z}(1)$ 正向延伸得分矩阵 H

	7	3	4	9	6
	17	16	0	0	0
3	16	<u>19.3537</u>	19	0	0
4	15	18.3537	<u>22.3537</u>	22	0
7	14	21	23.3549	<u>27.3549</u>	26.3549
4	0	20	24	26.3549	25.3549
5	0	19	23	25.3549	24.3549
					23.3549

表 6.8 不同序列的基本组成部分

	基因组	蛋白质	文本字符串	报警泛滥
组成元素	核苷酸	氨基酸	字母表	报警标签
类别数	5	23	26	> 1000

差较大的报警顺序不重要。最后，将式(6.8)中的基本相似度评分 $s(x_i, y_j)$ 重新表示为

$$s(x_i, y_j) = \mu + (\phi(p_j^y) - \mu) \max_{1 \leq v \leq V} w_{iv}^x w_{jv}^y, \quad (6.23)$$

其中 w_{iv}^x 和 w_{jv}^y 分别是 \tilde{X} 的第 i 个报警事件 x_i 和 \tilde{Y} 的第 j 个报警事件 y_j 。

案例3：由表6.6可知， X_h 中的第8个、第9个和第10个报警标签的时间戳非常接近； Y_h 中的第7个、第8个和第9个报警标签的时间戳非常接近。这些报警的时间顺序在序列比对中并不重要。因此，我们采用时间模糊容忍策略，这里 σ 设置为2。利用式(6.23)中重新制定的相似度分数 $s(x_i, y_j)$ ，得到了 $\tilde{Z}(1) = [2, 1, 3, 4]$ 正向延伸的得分矩阵 H ，如表6.7所示。后向延伸中，最佳比对跟表6.11中相同。结合两个方向上的延伸，最终的最佳比对结果为

$$\begin{array}{cccccccccc} \tilde{X} : & 9 & 5 & 6 & 2 & 1 & 3 & 4 & \underline{3} & \underline{4} & 7 \\ & | & | & | & | & | & | & | & | & | & | \\ \tilde{Y} : & 9 & 5 & - & 2 & 1 & 3 & 4 & \underline{7} & 3 & 4 \end{array} \quad (6.24)$$

在这里，用下划线标记的报警标签并不完全匹配，但由于它们的时间戳很接近，因此被视为匹配的报警对。正向延伸和后向延伸的分数分别为 $H_b = 22$ 和 $H_f = 27.3549$ ，因此最佳得分为 $S(\tilde{Z}(1)) = 22 + 27.3549 - 15 - 2 \times 2 = 30.3549$ 。

6.3.3 种子搜索、延伸和回溯步骤

在常规的BLAST算法中，种子搜索是根据具有固定大小的所有可能的报警事件组合建立一个查找表。因为组成查找表的元素数量有限，所以对于生物序列和文本字符串是可行的。然而，对于工业报警系统，不同报警标签的数量相当大，通常大于1000，如表6.8所示。例如，某一个工厂的报警标签数量约为14000个¹⁴。因此，建立报警泛滥的查找表是不可行的。例如，如果为1000个报警标签构建一个大小为3的查找表，那么表的大小为 $1000^3 = 10^9$ ，所需的计算量太大了。因此，常规BLAST的种子搜索步骤不能直接应用于报警泛滥序列。

为了解决这个问题，需要修改种子搜索步骤。主要思想是找到所有匹配的对，而不是一个

表 6.9 报警匹配配对

匹配对	$z_{i,1}$	$z_{i,2}$	$z_{i,3}$	$h(Z(i))$	子序列
$Z(1)$	1	1	2	6	[9, 5]
$Z(2)$	1	10	1	3	[9]
$Z(3)$	3	11	1	3	[6]
$Z(4)$	4	3	4	15	[2, 1, 3, 4]
$Z(5)$	6	8	2	6	[3, 4]
$Z(6)$	8	5	3	12	[3, 4, 7]
$Z(7)$	8	8	2	6	[3, 4]
$Z(8)$	11	6	1	3	[4]
$Z(9)$	11	9	1	3	[4]
$Z(10)$	12	2	1	3	[5]

固定大小的报警事件组合。通过定位一个匹配对 $Z(i), i = 1, 2, \dots, K_b$ 在 \tilde{X} 和 \tilde{Y} 中的位置，我们可以得到以下向量：

$$\mathbf{z}_i = (z_{i,1}, z_{i,2}, z_{i,3}), \quad (6.25)$$

其中 $z_{i,1}$ 和 $z_{i,2}$ 分别为匹配对 $Z(i)$ 在 \tilde{X} 和 \tilde{Y} 中的起始位置， $z_{i,3}$ 表示 $Z(i)$ 的长度， K_b 表示匹配对的总数。匹配对 $Z(i)$ 满足如下两个性质：第一， \tilde{X} 和 \tilde{Y} 中的 $Z(i)$ 的所有报警标签应该是相同的， 即 $e_{z_{i,1}+k}^x = e_{z_{i,2}+k}^y, k = 0, 1, \dots, z_{i,3} - 1$ ， 其中 $e_{z_{i,1}+k}^x$ 和 $e_{z_{i,2}+k}^y$ 分别是 \tilde{X} 和 \tilde{Y} 中的第 $(z_{i,1} + k)$ 个和第 $(z_{i,2} + k)$ 个报警事件；第二，任何匹配对都不应该与其他匹配对重叠。给定一个匹配对 $Z(i)$ ，两个条件 $z_{i,1} \in [z_{j,1}, z_{j,1} + z_{j,3} - 1]$ 和 $z_{i,2} \in [z_{j,2}, z_{j,2} + z_{j,3} - 1]$ ($j \neq i$) 不能同时成立。

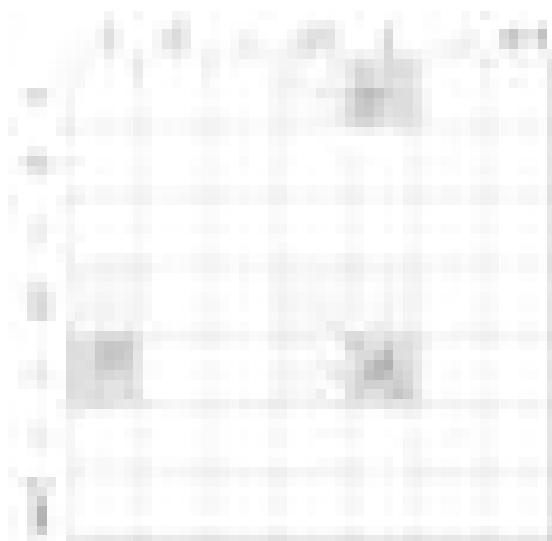


图 6.13 有间隙延伸中延伸分数矩阵的计算

通过计算每对匹配对 $Z(i)$ 的相似度得分，可以从匹配对中选择合适的种子，即

$$h(Z(i)) = \sum_{k=0}^{z_{i,3}-1} s(\tilde{x}_{z_{i,1}+k}, \tilde{y}_{z_{i,2}+k}), \quad (6.26)$$

其中 $s(\tilde{x}_{z_{i,1}+k}, \tilde{y}_{z_{i,2}+k})$ 是由式(6.8)所得。具有较高种子分数的匹配对被保留为种子，而其他匹配对被丢弃。种子和对应的种子分数分别被记为 $\tilde{Z}(k)$ 和 $h(\tilde{Z}(k))$ 。其中 $k = 1, \dots, K$, K 为前 K 高得分的匹配对数量，即种子的数量。以下案例说明了修改后的种子搜索步骤。

案例4：通过案例2中得到的两个短序列 $\tilde{X} = [9, 5, 6, 2, 1, 3, 4, 3, 4, 7, 4, 5]$ 和 $\tilde{Y} = [9, 5, 2, 1, 3, 4, 7, 3, 4, 9, 6]$ 计算出的匹配对，如表6.9所示。以 $z_4 = (4, 3, 4)$ 为例，匹配对为 $Z(4) = [2, 1, 3, 4]$ ，起始位置为 \tilde{X} 的第四个元素和 \tilde{Y} 的第三个元素。通过使用表6.5中基于优先级的相似性分数计算可以算出所有匹配对的相似性分数，如表6.9中的第五列所示。 $Z(4)$ 的种子分数为 $Z(4)$ is $h(Z(4)) = 4.5 + 4.5 + 3 + 3 = 15$ 。令 $K = 1$ ，种子分数最大的匹配对被选为种子，即 $\tilde{Z}(1) = [2, 1, 3, 4]$ ，对应的 $h(\tilde{Z}(1))$ 为15。

一旦得到了种子，常规BLAST算法就会向两个方向延伸种子，以找到高分片段对。在本小节中，使用基于优先级相似度评分来实施延伸步骤。有两种类型的延伸，即无间隙延伸和有间隙延伸。为了在比对报警序列时具有更大的灵活性，在本节中使用了有间隙延伸，将无关的报警事件与间隙对齐。

将 \tilde{X} 和 \tilde{Y} 的第 k 个种子 $\tilde{Z}(k)$ 在一个方向上的延伸分别记为长度为 m 的 X_s 和长度为 n 的 Y_s 。通过三条路径计算出的延伸分数矩阵 H ，如图6.13所示。最初， H 的第一个元素定义为

$$H_{1,1} = h(\tilde{Z}(k)) + U, \quad (6.27)$$

其中 $h(\tilde{Z}(k))$ 是 $\tilde{Z}(k)$ 的种子得分， U 是用来停止延伸的截止阈值。原则上，延伸会一直进行，直到扩展分数 $H_{i,j}$ 低于最大延伸分数减去阈值 U 为止，即 $H_{i,j} < H_{max} - U$ 。这里 H_{max} 表示延伸到 $H_{i,j}$ 之前获得的最高延伸分数。最初， $H_{max} = H_{1,1}$ 。因此，在 $H_{1,1}$ 中加入了 U ，避免了 $H_{1,1} < U$ 的退化情况。

对于所有的 $i = 2, 3, \dots, m + 1$ 和 $j = 2, 3, \dots, n + 1$ ，第 i 行和第 j 列的元素 $H_{i,j}$ 可以由图6.13中的第一条路径计算得出，

$$H_{i,j} = \max\{H_{i-1,j-1} + s(x_i, y_j), H_{i,j-1} + \delta, H_{i-1,j} + \delta, 0\}, \quad (6.28)$$

其中 $s(x_i, y_j)$ 是 x_i 和 y_j 之间的相似性得分。对于所有的 $i = 2, 3, \dots, m + 1$ 和 $j = 1$ ， $H_{i,1}$ 可以由图6.13中的第二条路径计算得出，

$$H_{i,1} = \max\{H_{i-1,1} + \delta, 0\}. \quad (6.29)$$

对于所有的 $i = 1$ 和 $j = 2, 3, \dots, n + 1$ ， $H_{1,j}$ 可以由图6.13中的第三条路径计算得出，

$$H_{1,j} = \max\{H_{1,j-1} + \delta, 0\}. \quad (6.30)$$

由图6.13可知，一共有三个延伸停止条件，分别对应公式(6.28)、(6.29)和(6.30)中的计算。首先，对于所有的 $i = 2, 3, \dots, m + 1$ 和 $j = 2, 3, \dots, n + 1$ ， $H_{i,j}$ 的延伸停止于

$$\max\{H_{i-1,j-1}, H_{i,j-1}, H_{i-1,j}\} < H_{max} - U, \quad (6.31)$$

其次，对于所有的 $i = 2, 3, \dots, m + 1$ 和 $j = 1, H_{i,1}$ 的延伸停止于

$$H_{i-1,1} < H_{\max} - U. \quad (6.32)$$

最后，对于所有的 $i = 1$ 和 $j = 2, 3, \dots, n + 1$, $H_{1,j}$ 的延伸停止于

$$H_{1,j-1} < H_{\max} - U. \quad (6.33)$$

Algorithm 8 有间隙延伸

```

 $H = \mathbf{0}_{m+1,n+1}; Q = \mathbf{0}_{m+1,n+1}; H_{1,1} = h(\tilde{Z}(k)) + U; H_{\max} = H_{1,1};$ 
for  $i = 2$  to  $m + 1$  do
    if  $H_{i-1,1} \geq H_{\max} - U$  then
         $H_{i,1} = \max\{H_{i-1,1} + \delta, 0\}; Q_{i,1} = 3;$ 
    end if
end for
for  $j = 2$  to  $n + 1$  do
    if  $H_{1,j-1} \geq H_{\max} - U$  then
         $H_{1,j} = \max\{H_{1,j-1} + \delta, 0\}; Q_{1,j} = 2;$ 
    end if
    for  $i = 2$  to  $m + 1$  do
        if  $\max\{H_{i-1,j-1}, H_{i,j-1}, H_{i-1,j}\} \geq H_{\max} - U$  then
             $H_{i,j} = \max\{H_{i-1,j-1} + s(x_i, y_j), H_{i,j-1} + \delta, H_{i-1,j} + \delta, 0\};$ 
            if  $H_{i,j} = H_{i-1,j-1} + s(x_i, y_j)$  then
                 $Q_{i,j} = 1;$ 
            else if  $H_{i,j} = H_{i,j-1} + \delta$  then
                 $Q_{i,j} = 2;$ 
            else if  $H_{i,j} = H_{i-1,j} + \delta$  then
                 $Q_{i,j} = 3;$ 
            end if
            if  $H_{i,j} > H_{\max}$  then
                 $H_{\max} = H_{i,j};$ 
            end if
        end if
    end for
    if  $\max_{1 \leq i \leq m+1} H_{i,j} < H_{\max} - U$  then
        break;
    end if
end for
```

单向有间隙延伸可以通过算法8中的伪代码来描述。算法8的输入为子序列 X_s 和 Y_s 以及种子分数 $h(\tilde{Z}(k))$ ，输出为延伸得分矩阵 H 。一旦停止延伸，则根据算法8中得到的指示矩阵 Q 进行回溯，得到最优的比对结果。它从得分最高的元素开始。如果 $Q_{i,j}$ 等于1、2或者3，则分别回溯到 $Q_{i-1,j-1}$ 、 $Q_{i,j-1}$ 或 $Q_{i-1,j}$ 。

将向后延伸和向前延伸的最大延伸分数分别表示为 H_b 和 H_f 。最终基于种子 $\tilde{Z}(k)$ 的最佳比对的最终相似性得分计算为

$$S(\tilde{Z}(k)) = H_b + H_f - h(\tilde{Z}(k)) - 2U. \quad (6.34)$$

当发现几个不同的种子时，这些种子的最佳排列可能是不同的。得分最高的子序列被视为最佳

比对，对应的分数被视为最终 \tilde{X} 和 \tilde{Y} 之间的最佳比对分数，记为 $S(\tilde{X}, \tilde{Y})$ 。

案例5：本案例使用种子 $\tilde{Z}(1) = < 2, 1, 3, 4 >$ 以及 $h(\tilde{Z}(1)) = 15$ 来说明有间隙延伸的细节。延伸阈值设置为 $U = 2|\delta| = 2$ ，即允许比对中有两个间隙存在。因此， $H_{1,1} = h(\tilde{Z}(1)) + U = 15 + 2 = 17$ 。种子 $\tilde{Z}(1) = < 2, 1, 3, 4 >$ 首先向前延伸到子序列 $X_f = [3, 4, 7, 4, 5]$ 和 $Y_f = [7, 3, 4, 9, 6]$ 。生成 X_f 和 Y_f 的扩展分数矩阵H，如表6.10所示。向前延伸中的回溯产生了最佳的对齐，如表6.10中有下划线标记分数的报警标签所示。然后种子 $\tilde{Z}(1) = < 2, 1, 3, 4 >$ 向后延伸到子序列 $X_b = < 6, 5, 9 >$ 和 $Y_b = < 5, 9 >$ 。生成的 X_b 和 Y_b 的扩展分数矩阵H，如表6.11所示。图6.14展示了种子 $\tilde{Z}(1) = < 2, 1, 3, 4 >$ 在两个方向上的延伸。绿色方块表示匹配的报警对，灰色方块表示延伸继续进行的条目。显然，在延伸的过程中，大部分由白色方块标记的搜索空间都被删除了，从而大大降低了计算量。

结合两个方向的延伸，最终实现最佳的比对为

$$\begin{array}{ccccccccccccc} \tilde{X}: & 9 & 5 & 6 & 2 & 1 & 3 & 4 & 3 & 4 & 7 & - & 4 \\ & | & | & & | & | & | & & & & | & & | \\ \tilde{Y}: & 9 & 5 & - & 2 & 1 & 3 & 4 & - & - & 7 & 3 & 4 \end{array} \quad (6.35)$$

值得一提的是，报警7有比报警3更高的优先级。因此，种子 $\tilde{Z}(1)$ 的正向延伸和后向延伸的最高分数分别是 $H_b = 22$ 和 $H_f = 23$ 。所以，最终得分是 $S(\tilde{Z}(1)) = 22 + 23 - 15 - 2 \times 2 = 26$ 。

种子搜索步骤的目的是找到所有匹配的报警序列段，其中分数前 K 大的报警序列段作为种子保留，其余的序列段丢弃。在延伸步骤中，所有种子在两个方向上进行延伸，直到相似度得分低于最大延伸分数减去截止阈值 U 为止。因此， K 和 U 是影响计算量和匹配精度的两个重要参数。下面的例子使用了蒙特卡罗仿真来测定 K 和 U 对算法的影响。

案例6：该案例实现了1000次蒙特卡罗仿真。在每次仿真中，一对符号序列 X_l 和 Y_l , $l = 1, 2, \dots, 1000$ 都是随机生成的。符号标记来自式(6.7)中的字母表 $\Sigma = \{1, 2, \dots, V\}$ ，其中三

表 6.10 种子 $\tilde{Z}(1) = [2, 1, 3, 4]$ 的正向延伸矩阵

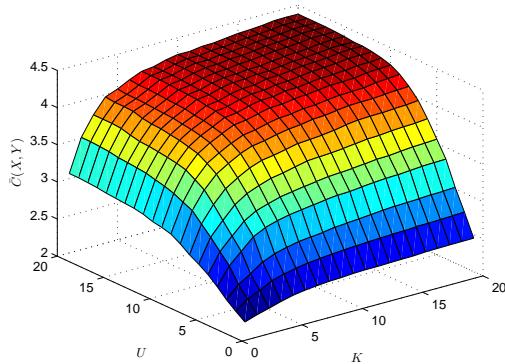
	7	3	4	9	6
17	16	0	0	0	0
3	<u>16</u>	15	0	0	0
4	<u>15</u>	14	0	0	0
7	14	<u>21</u>	<u>20</u>	19	0
4	0	20	19	<u>23</u>	22
5	0	19	18	22	21

表 6.11 种子 $\tilde{Z}(1) = [2, 1, 3, 4]$ 的后向延伸矩阵

	5	9
17	16	0
6	<u>16</u>	15
5	15	<u>19</u>
9	14	18



图 6.14 有间隙延伸的图形说明

图 6.15 不同的 K 和 U 对应的匹配对平均数量 $\bar{C}(X, Y)$

个优先级 $\{p_1, p_2, p_3\}$ 的比例分别为5%/15%/80%。这里使用的是表6.5中的相似性分数。序列的长度 L_x 和 L_y 都是 $[L_1, L_2]$ 间的随机整数。时间戳 $t_i^x, i = 1, 2, \dots, L_x$ 和 $t_j^y, j = 1, 2, \dots, L_y$ 为范围 $[1, t_{max}]$ 内的高斯随机变量，其中 t_{max} 是 $[t_{max,1}, t_{max,2}]$ 中的一个统一变量。参数设置为 $V = 100$, $L_1 = 50$, $L_2 = 500$, $t_{max,1} = 4320$ 秒(0.05天), $t_{max,2} = 86300$ 秒(1天)。算法在一台CPU为3.3GHz, 内存为4GB, 操作系统为64位的个人计算机上进行。通过计算 X 和 Y 之间的匹配对的数量 $C(X, Y)$ 来测量匹配精度，并记录该算法应用于 X 和 Y 匹配的计算时间 T 。图6.15为 K 和 U 不同取值下1000对 X_l 和 Y_l 的平均匹配对数，即 $\bar{C}(X_l, Y_l) := \frac{1}{1000} \sum_{l=1}^{1000} C(X_l, Y_l)$ 。图6.16是与图6.15对应的平均计算时间 $\bar{T} := \frac{1}{1000} \sum_{l=1}^{1000} T_l$ 。

由图6.15和图6.16可知，给定固定的种子数 K ，当 $U \leq 10$ 时， $\bar{C}(X, Y)$ 随着 U 的增加急剧增加；当 $U > 10$ 时， $\bar{C}(X, Y)$ 随着 U 的增加并没有明显变化。给定一个固定的阈值 U ，当 $K \leq 7$ 和 $K > 7$ 时，可以得到类似的观察结果。很明显，两个参数的增大会增加计算时间。为了在短时间内获得准确的比对结果， U 和 K 的推荐值分别为10和7。

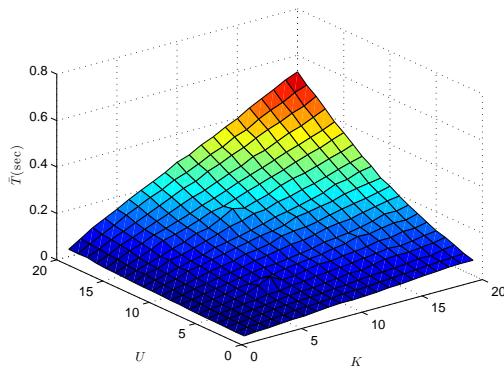


图 6.16 不同的 K 和 U 对应的平均计算时间 \bar{T}

6.3.4 方法步骤

本节提出的局部比对算法由以下步骤组成，流程图如图6.17所示。

1. 去除两个报警泛滥序列 X_h 和 Y_h 中的抖动报警。
2. 对 X_h 和 Y_h 使用了用于基于报警事件集合的预匹配机制，即计算式(6.16)中的 $S_{set}(X, Y)$ 。如果 $S_{set}(X, Y) > \gamma$ ，则可以通过保留 X_h 和 Y_h 的共同报警事件来得到子序列 \tilde{X} 和 \tilde{Y} ；否则，计算将被终止。
3. 在种子搜索步骤中， \tilde{X} 和 \tilde{Y} 的匹配段确定为 $Z = (z_1, z_2, z_3)$ 。计算它们的种子分数 $h(Z)$ 。拥有较大种子分数的序列段被选为种子 $\tilde{Z}(k)$, $k = 1, \dots, K$ ，其它的则被丢弃。每个种子的得分记为 $h(\tilde{Z}(k))$ 。
4. 使用算法8将间隙延伸应用到每个种子 $\tilde{Z}(k)$, $k = 1, \dots, K$ 上。在每次迭代中，基于式(6.18)-(6.23)计算 $s(x_i, y_j)$ ，基于式(6.28)-(6.30)计算 $H_{i,j}$ ，并记录 $Q_{i,j}$ 。如果式(6.31)-(6.33)中的停止条件被满足，则终止延伸。通过式(6.34)计算出 $\tilde{Z}(k)$ 中最佳比对的延伸分数。
5. 找到得分最高的对齐子序列作为 X_h 和 Y_h 的最佳序列对齐。为了测量 X_h 和 Y_h 比对的计算精度和效率，定义两个量为： X_h 和 Y_h 的匹配对数量 $C(X, Y)$ 用于测量比对精度；记录算法应用于 X_h 和 Y_h 的计算时间 T 。

6.3.5 案例分析

本小节提供工业案例研究，以验证所提方法的有效性。在案例7中，对历史数据库的多次查询，研究了多条报警泛滥。在案例8中，对一条报警泛滥序列进行了详细的分析。

案例7：本案例研究的工业过程是一个石油转化装置，这在石化行业中是相当普遍的。它把原油转变成有价值的石油产品，如挥发油、汽油和柴油燃料。如图6.18所示，石油转换装置

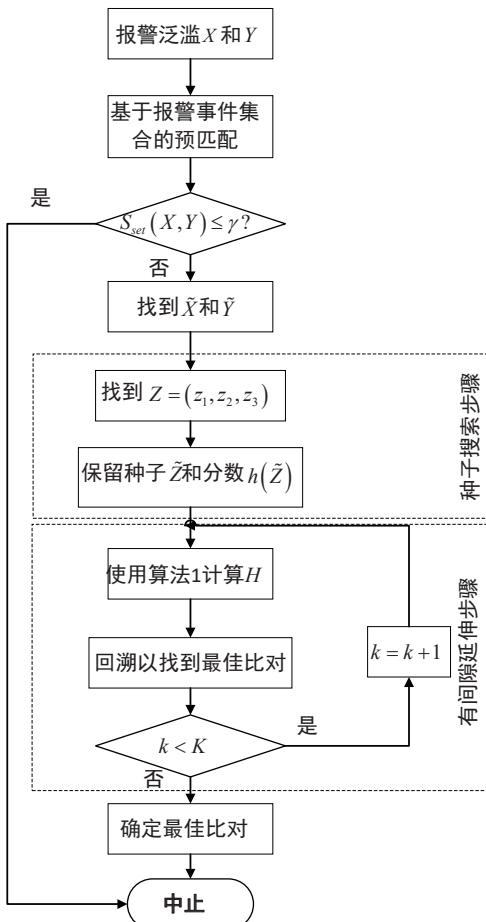


图 6.17 使用的局部比对算法的流程图

由许多相互连接的管道、加热器、反应器、分离器和容器组成。原油和氢气在原料加热器和氢气加热器中被加热到一定的温度，并与两个反应器中的催化剂床接触。气体和液体流出物从反应器溢出到一个公共出口管道，该管道通向分离器，在那里蒸汽、液态碳氢化合物和水相被分离。液态烃流被带到常压塔，并被分馏成所需的产品，这些产品被汽油剥离器剥离。精炼过程由一个分布式控制系统操作。为了保证油转换装置的安全运行，工厂在不同的区域配置了大量的报警变量。

从2013年5月12日至2014年11月12日的石油转换装置，即在包含1547个独特的报警和3个报警优先级的历史报警数据中，检测出了389条去除了抖动报警的报警泛滥序列，并保存为报警泛滥数据库 $\mathbb{B} = \{Y_l : l = 1, 2, \dots, 389\}$ 。其中最长的报警泛滥包含了1840个报警事件，而最短的报警泛滥只有10个报警事件。这些报警泛滥的平均长度为69.2。

为了说明所提出的局部对准算法的有效性，我们研究了数据库B中在不同时间发生的十条报警泛滥序列 X_{h_i} , $i = 1, \dots, 10$ 。首先，使用基于报警事件集合的预匹配机制， γ 被设置为0，以排除数据库中无关紧要的报警泛滥序列，而剩下的序列则保留下进行序列比对。表6.12的第三列列出了每条报警泛滥 X_i 的相关报警泛滥数量。然后，使用本节提出的算法对序列进

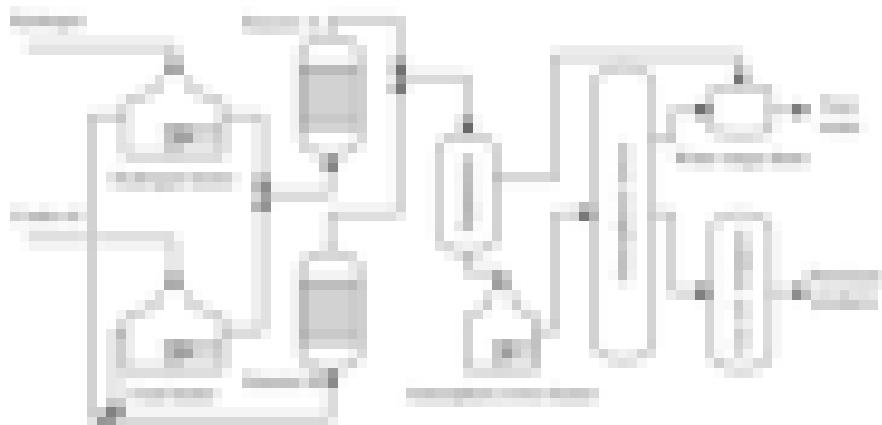


图 6.18 石油转换装置简图

表 6.12 由匹配对的平均数目测量的对准精度 $\bar{C}(X, Y)$ 和计算时间 T

索引	$L(X_i)$	$N(Y)$	使用的算法	
			$\bar{C}(X, Y)$	$T(\text{sec})$
1	1010	130	6.65	8.32
2	848	341	7.07	11.52
3	661	382	8.21	10.02
4	420	291	6.91	5.53
5	347	327	7.65	7.63
6	290	305	5.91	4.83
7	249	320	5.71	5.66
8	196	223	6.75	3.53
9	186	276	6.19	4.42
10	114	125	5.49	1.86

行比对，参数的设置与案例6中相同。匹配对的平均数量 $\bar{C}(X, Y) := \frac{1}{389} \sum_{l=1}^{389} C(X_i, Y_l)$ 和总计算时间 T 由表6.12中的最后两列所示。

案例8：对案例7中的第十条报警泛滥序列进行了详细的分析。如表6.12所示，这条报警泛滥序列 X_h 包含114个报警事件。数据库B中有125条报警泛滥序列拥有和 X_h 相同的报警事件。图6.19展示了所用算法中匹配对的数量和计算时间。通过所用算法的检测，如图6.19所示的第117条报警泛滥 Y_h 与 X_h 最为相似。通过算法8中的回溯步骤，我们可以得到 X_h 和 Y_h 的最佳比对方式。图6.21展示了两条序列的比对结果。两次报警泛滥 X_h 和 Y_h 的报警事件按时间排序，而 X_h 和 Y_h 的发生日期不同。在40次比对中，30对报警事件完全按照相同的时间顺序匹配，而另外10对报警事件则没有按照时间顺序匹配。观察这些无序匹配的时间戳，可以发现这些报警几乎同时发生。因此，它们的排列顺序不重要。

由于图6.18中两种氢气供应中的一种的净化再循环气体供应不足，导致报警液泛滥。为便于说明，简化的氢加热器原理图，如图6.20所示。供给量不足导致出口对流壁过热，为了保持出口温度在设计范围内，工厂激活了一系列的自动调节和保护策略。通过分析 Y_h 发生的原因，

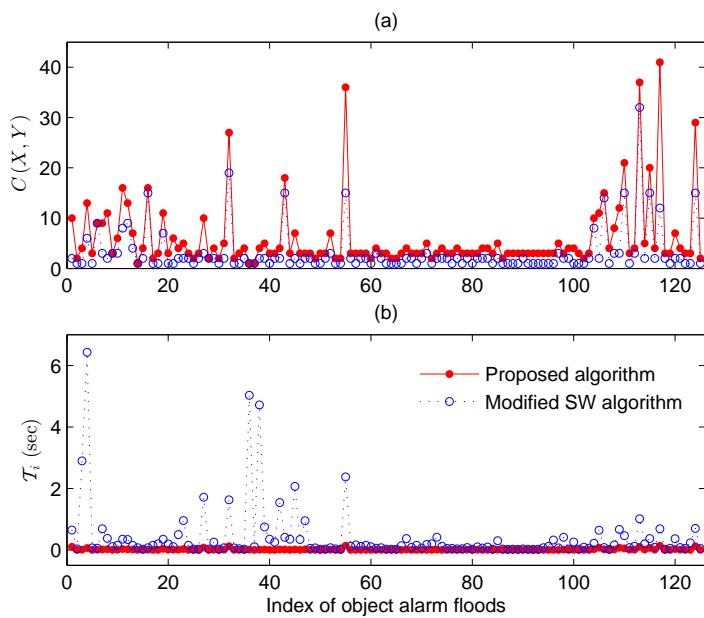


图 6.19 使用的算法的性能统计

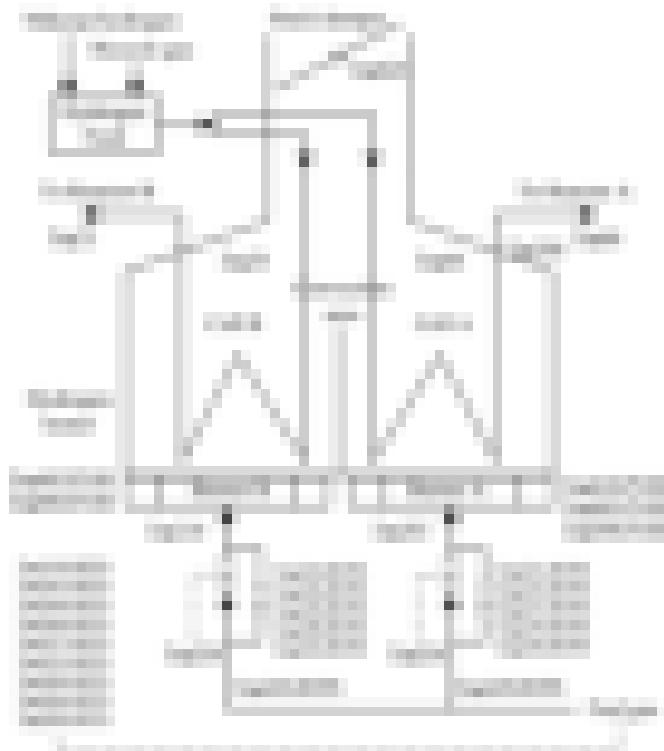


图 6.20 氢加热器简化示意图

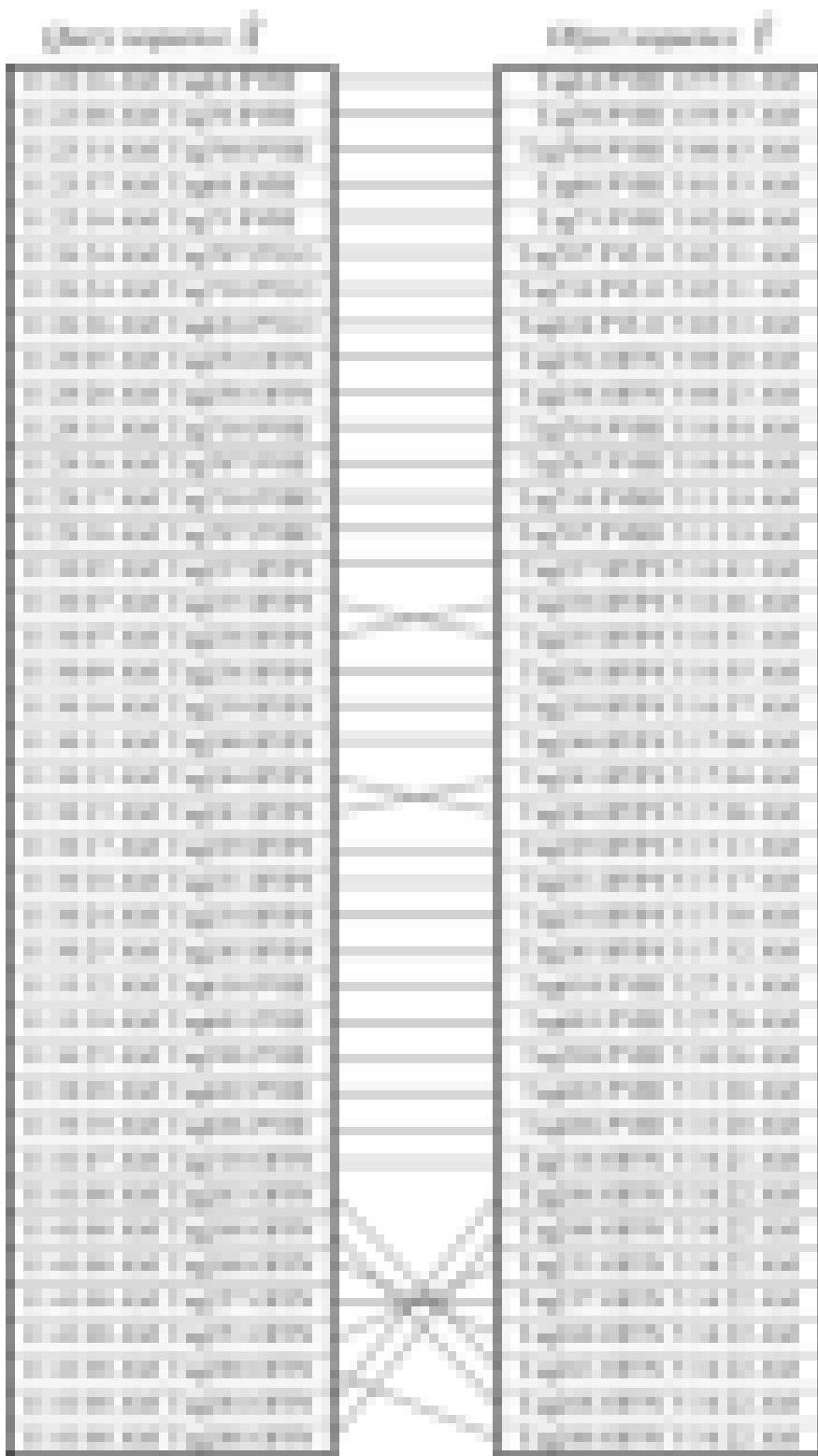


图 6.21 报警泛滥序列 X_h 和数据库B中的第117条报警泛滥数据的最佳比对结果

可以预测报警泛滥 X_h 是由氢饲料供应不足引起的。通过查阅历史记录，发现补给氢（两种补给氢中的一种）的供应减少，这支持了 X_h 和 Y_h 之间的相似性。

6.4 报警泛滥预测

报警事件是指当发生异常时，报警变量 x_i 进入报警状态的事件。假设历史报警泛滥数据集为 $\{X_h(n)\}_{n=1}^N$ ，其中 n 为报警序列的索引， N 为历史报警泛滥的次数。第 n 个历史报警泛滥序列 $X_h(n) := [x_{h_1}, x_{h_2}, \dots, x_{h_n}]$ 由按时间顺序排列的报警事件组成，其中 x_{h_i} 为 $X_h(n)$ 中第 i 个报警事件。历史报警泛滥数据集可以通过6.2节中的报警泛滥检测算法获得。

假设当前正在出现的报警序列为 $X_c = [x_{c,1}, x_{c,2}, \dots, x_{c,i-1}]$ ，其中出现的报警事件同样是以时间顺序排列。由于导致 X_c 出现的异常可能同样导致了 $\{X_h(n)\}_{n=1}^N$ 中一些历史报警泛滥序列的产生，所以可以从这些历史报警泛滥序列中推断即将发生的报警事件。因此，本部分的目的是利用 $\{X_h(n)\}_{n=1}^N$ 中的信息来预测 X_c 中下一个将出现的报警事件 $x_{c,i}$ 。

6.4.1 报警序列的相似性分数

如果 $\{X_h(n)\}_{n=1}^N$ 中的一些历史报警泛滥序列与当前出现的报警序列 X_c 差异较大，我们就认为这些历史报警泛滥序列与 X_c 无关，所以它们不应该被用来预测 X_c 中即将发生的报警事件。否则，预测的精确性将会大打折扣。因此，需要计算历史报警泛滥 $X_h(n)$ 与 X_c 之间的相似度评分，并将相似度评分较低的 $X_h(n)$ 从预测将要使用的集合中剔除。

首先，介绍的是6.3节中提到的间隙比对，由于两个序列可能有不同的长度，它们的长度通过在一个或两个序列中插入间隙（由符号“-”表示）来相等。这样一个序列中的每个元素在另一个序列中的相同位置上都有对应的元素。其次，计算相似度分数是为了在某种意义上选择最优的比对，因为任意两个序列都有很多不同的比对，这是因为间隙可以任意添加。对于两个对齐序列中的元素对 (u,v) ，相似度评分函数为

$$s(u, v) = \begin{cases} \lambda, & \text{如果 } u = v \\ \mu, & \text{如果 } u \neq v \\ \delta, & \text{如果 } u \text{ 或者 } v \text{ 是一个间隙} \end{cases}.$$

其中 λ 为正常数，而 μ 和 δ 为两个负常数。相对于不匹配元素的比对方式，间隙比对更可取，因此 μ 和 δ 需要满足不等式 $\mu < 2\delta < 0$ 。这里使用的参数为 $\lambda = 1$ 、 $\mu = -1$ 和 $\delta = -0.4$ 。因为只考虑相似度评分的顺序，而不考虑相似度评分的具体值，所以本方法对 λ 、 μ 和 δ 的值不敏感。

最后，使用SW算法来寻找 $X_h(n)$ 和 X_c 的最高相似性分数。SW算法生成一个相似性得分矩阵 H ，其中矩阵的第 $(i+1)$ 行和第 $(j+1)$ 列的元素 $H_{i+1,j+1}$ 分别表示序列 $X_{h,1:i} := [x_{h_1}, x_{h_2}, \dots, x_{h_i}]$ 和 $X_{c,1:j} := [x_{c,1}, x_{c,2}, \dots, x_{c,j}]$ 的相似性分数。矩阵的第一行和第一列的元素都被设置为0。 $H_{i+1,j+1}$ 的值可以递归计算为

$$\begin{aligned} H_{i+1,j+1} &= \max \{H_{i,j} + s(x_{h_i}, x_{c_j}), H_{i,j+1} + \delta, \\ &\quad H_{i+1,j} + \delta, 0\}. \end{aligned}$$

H 的最后一行和最后一列的元素记为 $S(X_h(n), X_c)$, 表示 $X_h(n)$ 和 X_c 中最佳比对的最高相似度分数。我们提出的方法只需要最高相似度分数 $S(X_h(n), X_c)$, 而不需要对应的 $X_h(n)$ 和 X_c 的具体比对情况。

6.4.2 预测报警事件的概率

预测即将发生的报警事件本质是, 如果某些历史报警泛滥序列与当前报警泛滥序列相似, 那么即将发生的报警事件很可能与这些历史报警泛滥序列对应的异常工况相似。则预测步骤如下:

首先, 使用一种名为 m -gram的概率语言模型, 基于 X_c 中最近出现的 $(m - 1)$ 个报警事件 $x_{c,i-m+1}, x_{c,i-m+2}, \dots, x_{c,i-1}$ 来预测即将到来的报警事件 $x_{c,i}$ 。每条历史报警泛滥序列 $X_h(n)$ 被分为长度为 $(m - 1)$ 的序列段, 相邻段的不同报警事件的值都记为1, 即

$$\{X_h(n; m - 1)\} := \{x_{h_1:h_{m-1}}, x_{h_2:h_m}, \dots, x_{h_n-m+2:h_n}\},$$

其中 $x_{h_{j-m+2}:h_j} := [x_{h_{j-m+2}}, \dots, x_{h_j}]$ 。例如, 对于3-gram模型, 历史报警泛滥序列 $X_h(n) := [x_1, x_3, x_4, x_5, x_6]$ 被分为4段: $[x_1, x_3]$ 、 $[x_3, x_4]$ 、 $[x_4, x_5]$ 和 $[x_5, x_6]$ 。与 $x_{c,i-m+1:i-1}$ 相同的序列段的数量被记为 $k_1(m)$, 即

$$k_1(m) = \sum_{n=1}^N |x_{c,i-m+1:i-1} \equiv \{X_h(n; m - 1)\}|, \quad (6.36)$$

其中, 如果比对序列相同, 则符号‘ \equiv ’为真值‘1’, 运算符 $|\cdot|$ 计算了真值的数量。 $X_h(n)$ 中长度为 m 的序列段的数量记为 $k_2(m; x_l)$, 即

$$k_2(m; x_l) = \sum_{n=1}^N |[x_{c,i-m+1:i-1}, x_l] \equiv \{X_h(n; m)\}|, \quad (6.37)$$

其中

$$\{X_h(n; m)\} := \{x_{h_1:h_m}, x_{h_2:h_{m+1}}, \dots, x_{h_n-m+1:h_n}\}.$$

也就是说, $x_{h_{j-m+1}:h_j}$ 的前 $(m - 1)$ 个元素与 $x_{c,i-m+1:i-1}$ 相同, 并且最后一个元素是报警事件 x_l 。因此, X_c 中即将到来的报警事件为 x_l 的概率可以通过 m -gram模型得到,

$$p(x_l; m) = \begin{cases} \frac{k_2(m; x_l)}{k_1(m)}, & \text{如果 } k_1(m) \neq 0 \\ 0, & \text{其它} \end{cases}. \quad (6.38)$$

概率 $p(x_l; m)$ 为 $\{X_h(n)\}_{n=1}^N$ 中 $x_{c,i-m+1:i-1}$ 的数量与 $\{X_h(n)\}_{n=1}^N$ 中 $[x_{c,i-m+1:i-1}, x_l]$ 的数量的比例。它有一个清晰的物理解释, 然而, $p(m; x_l)$ 不能说明预测的可靠性。

然后, 我们利用贝叶斯估计器对预测的报警事件 x_l 的概率 $p(x_l)$ 进行估计, 并将其置信区间作为预测可靠性的定量度量。为了简化符号, 真实(未知)概率 $p(x_l)$ 被记为 $\theta := p(x_l)$, 所以在贝叶斯估计器中 θ 被视为一个随机变量, $p(\theta)$ 是 θ 的先验概率。基于当前发生的序

列 $x_{c,i-m+1:i-1}$, θ 的后验概率为

$$p(\theta|x_{c,i-m+1:i-1}) = \frac{p(x_{c,i-m+1:i-1}|\theta)p(\theta)}{p(x_{c,i-m+1:i-1})}. \quad (6.39)$$

式(6.36)中的 $k_1(m)$ 和式(6.37)中 $k_2(m; x_l)$ 分别为序列 $x_{c,i-m+1:i-1}$ 的数量和历史报警泛滥数据中 $[x_{c,i-m+1:i-1}, x_l]$ 的数量。 $[x_{c,i-m+1:i-1}, x_l]$ 中最后一个元素在实验中可视为两个互斥结果的离散随机事件: x_l 或者非 x_l 。给出 θ 作为预测结果 x_l 的概率, x_l 的数量等于 $k_1(m)$ 次试验中 $k_2(m; x_l)$ 的个数。因此, 条件概率 $p(x_{c,i-m+1:i-1}|\theta)$ 符合二项分布, 即

$$p(x_{c,i-m+1:i-1}|\theta) = C_{k_1(m)}^{k_2(m;x_l)} \theta^{k_2(m;x_l)} (1-\theta)^{k_1(m)-k_2(m;x_l)}, \quad (6.40)$$

其中

$$C_{k_1(m)}^{k_2(m;x_l)} = \frac{k_1(m)!}{k_2(m;x_l)! (k_1(m) - k_2(m;x_l))!}.$$

因为一些密切相关 (完全独立) 的警报事件可能总是 (从不) 同时出现, θ 可以取 $[0, 1]$ 间的任何实值。这等价于没有关于 θ 的先验知识情况。在这种情况下, 一种常见的做法是 $p(\theta)$ 可以采用连续均匀分布。式(6.39)中的分母 $p(x_{c,i-m+1:i-1})$ 可以通过全概率定理来计算, 即

$$p(x_{c,i-m+1:i-1}) = \int_0^1 p(x_{c,i-m+1:i-1}|\theta)p(\theta)d\theta. \quad (6.41)$$

因此, 式(6.39)中 $p(\theta|x_{c,i-m+1:i-1})$ 可以通过使用式(6.40)、(6.41)来计算, 并被记为 $p(\theta|m, N) := p(\theta|x_{c,i-m+1:i-1})$ 。 $p(\theta|m, N)$ 的可靠性可以通过置信水平为 $1 - \alpha$ 的置信区间来测定,

$$\min(p_2(\theta; m, N) - p_1(\theta; m, N)) \quad (6.42)$$

$$\text{s.t. } \int_{p_1(\theta; m, N)}^{p_2(\theta; m, N)} p(\theta|x_{c,i-m+1:i-1})d\theta = 1 - \alpha.$$

其中 α 是一个正的小数值, 如 $\alpha = 0.05$ 。式(6.42)中的优化问题可以通过网格搜索方法来解决。

m 的值越大表示预测的可靠性越高, 因为预测中涉及的报警事件越多。这里引入了一个权重因子 $c(m)$

$$c(m) = 1 - \frac{1}{m!},$$

其中 $1/m!$ 测量两个报警子序列具有相同的 m 个连续报警事件的概率。计算加权后验概率 $p'(\theta|m, N)$

$$p'(\theta|m, N) = c(m) \cdot p(\theta|m, N), \quad (6.43)$$

以及加权下限和上限

$$\begin{aligned} p'_1(\theta; m, N) &= c(m) \cdot p_1(\theta; m, N), \\ p'_2(\theta; m, N) &= c(m) \cdot p_2(\theta; m, N). \end{aligned} \quad (6.44)$$

因为, 我们更倾向于选择预测的报警事件中可靠性较高的结果, 这里使用式(6.44)中的置信区间下限 $p'_1(\theta; m, N)$ 来确定 m 值的选取。因此, 预测报警事件为 x_l , 概率为 $p'_1(\theta; m, N)$ 。

6.4.3 方法步骤

本节将介绍所提方法的详细步骤。简单地说，该方法由两个迭代组成。第一次迭代是从历史报警泛滥序列集合中删除与当前报警序列相似度得分最小的序列；第二次迭代是增加相邻报警事件的数量 m ，用于预测即将发生的报警事件。这两个迭代会在找到式(6.44)中 $p'_1(\theta; m, N)$ 的最大值时停止。

给出当前报警序列 $X_c = [x_{c,1}, x_{c,2}, \dots, x_{c,i-1}]$ 和历史报警泛滥序列集 $\{X_h(n)\}_{n=1}^N$ ，算法由以下步骤组成：

1. 计算相似度评分 $S(X_h(n), X_c)$, $n = 1, 2, \dots, N$, 并将历史报警泛滥数据集以相似性分数的降序重新排列为 $\{X_{h,sort}(n)\}_{n=1}^N$;
2. 得到式(6.39)中的先验概率 $p(\theta|m, N)$ 、式(6.42)中对应的置信区间 $[p_1(\theta; m, N), p_2(\theta; m, N)]$ 、式(6.43)中它们的加权部分 $p'(\theta|m, N)$ 以及式(6.44)中的 $[p'_1(\theta; m, N), p'_2(\theta; m, N)]$, 对于 $m = 2$, 去掉最后一个序列 $X_{h,sort}(N)$, 将 N 更新为 $N - 1$;
3. 对于 $m = 3, 4, \dots$, 重复步骤2直到没有更大的 $p'_1(\theta; m, N)$ 值可以获得;
4. 确定有最大加权下界 $\max p'_1(\theta; m, N)$ 的报警事件 x_l 为最优预测报警事件。

6.4.4 案例分析

本节给出一个数值例子具体阐明了如何利用历史报警泛滥序列预测报警时间，并证明了所提方法的有效性。

案例1：田纳西伊士曼（TE）过程是一种成熟的基准过程，在过程控制和故障诊断领域得到了广泛的应用。有 A 、 B 、 C 、 D 和 E 五种原料气体， F 、 G 和 H 是三种产品。TE进程包含21个预设故障。它们被分为阶跃故障和随机故障，分别与某些过程变量的阶跃变化和增量随机变化有关。当阶跃故障发生时，许多过程变量会发生重大变化，从而导致报警泛滥的出现。相比之下，当出现随机故障时，几个过程变量只受到很小的变化的影响。因此，我们采用阶跃故障引起的报警泛滥的方式，来得到历史报警泛滥序列的集合和模拟当前正在出现的报警序列，而随机故障引起的报警则被视为阶跃故障引起的报警事件的“噪声”。

这里考虑了五个故障，即A/C比值的阶跃故障1，B成分的阶跃故障2，A、B、C成分的随机故障7，D温度的随机故障8和C温度的随机故障9。应用100次蒙特卡罗仿真，通过在激活故障7、8、9之后再激活阶跃故障1或者2，我们得到了表6.13所示的历史报警泛滥数据集。其

表 6.13 历史报警泛滥序列

序号	报警泛滥序列	数量
1	$x_1 x_4 x_5 x_6 x_7 x_8 x_9 x_{11} x_{12} x_{13} x_{30} x_{32} x_{33}$	20
2	$x_{14} x_{15} x_{16} x_{17} x_{18} x_6 x_{20} x_{19} x_{21} x_{22} x_{26} x_{27} x_{28}$	20
3	$x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{30} x_{32} x_{33}$	30
4	$x_{14} x_{15} x_{16} x_{18} x_6 x_{20} x_{19} x_{21} x_{22} x_{23} x_{27} x_{28} x_{31}$	30

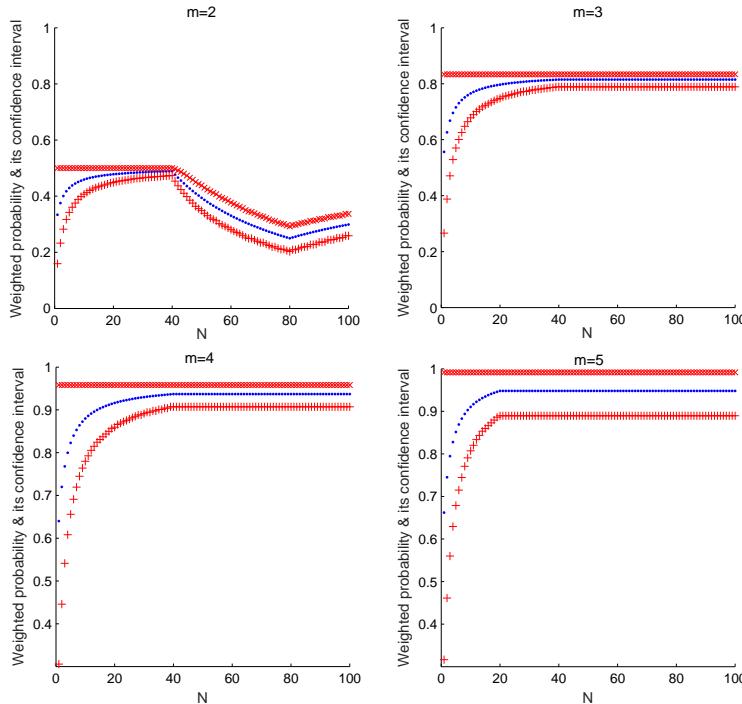


图 6.22 预测概率 $P'(\theta; m, N)$, $\theta = p(x_7)$ 和对应的置信区间 $[p'_1(\theta; m, N), p'_2(\theta; m, N)]$

表 6.14 重新排序的历史报警泛滥序列

序号	报警泛滥序列	数量
3	$x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}x_{30}x_{32}x_{33}$	30
1	$x_1x_4x_5x_6x_7x_8x_9x_{11}x_{12}x_{13}x_{30}x_{32}x_{33}$	20
2	$x_{14}x_{15}x_{16}x_{17}x_{18}x_6x_{20}x_{19}x_{21}x_{22}x_{26}x_{27}x_{28}$	20
4	$x_{14}x_{15}x_{16}x_{18}x_6x_{20}x_{19}x_{21}x_{22}x_{23}x_{27}x_{28}x_{31}$	30

表 6.15 $X_c = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ 的预测结果

m	预测的报警事件	概率	置信区间
2	x_7	0.4889	[0.4734, 0.5]
3	x_7	0.8148	[0.7890, 0.8333]
4	x_7	0.9370	[0.9074, 0.9583]
5	x_7	0.9482	[0.8896, 0.9917]

中序号1和3的报警泛滥序列是由故障1产生的，序号2和4的报警泛滥序列是由故障2产生的。表6.16展示了报警事件 x_1, x_2, \dots, x_{23} 以及它们关联的过程变量、报警限和单位。

该方法可用于预测当前报警事件序列 X_c 中即将出现的报警事件。若 $X_c = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ 为当前报警序列，首先计算相似度评分 $S(X_h(n), X_c)$ 并且重新排列历史报警泛滥序列为 $\{X_{h,sort}(n)\}_{n=1}^{N=100}$ ，如表6.14所示。不同的 m 值下算法其他步骤的预测结果，如表6.15所示。然后对于不同的 m 值，图6.22展示了对应的加权概率 $p'(\theta; m, N)$ 和它的置信区间，这里

置信水平取为0.95。如果 $m = 2$, 预测报警事件为 x_7 的加权下限 $p'_1(\theta; m, N)$ 是很小的, 这是因为历史报警泛滥序列中有两种可能: ‘ x_6x_7 ’和‘ x_6x_{20} ’。并且 $p'_1(\theta; m, N)$ 的值随着不相关报警泛滥2和4的去除不断变大。最后, 因最大的 $p'_1(\theta; m, N)$ 受权重因子 $c(m = 2) = 0.5$ 约束。如果 $m = 3, m = 4$ 或 $m = 5$, 只有 x_7 一种可能, $p'_1(\theta; m, N)$ 的值会随着相似报警泛滥序列的去除单调递减。总体而言, 该方法给出了概率加权下界为0.9074的最优预测报警事件 x_7 。这是一个合理的结果, 因为表6.14中最后两行的报警泛滥序列跟 X_c 的相关性分数很低, 是由故障2而非故障1产生的。

6.5 本章小结

本章主要内容是报警泛滥的检测、相似性分析和预测。针对工业过程中产生的报警数据, 本章首先通过设计报警泛滥检测算法从历史数据中提取历史报警泛滥序列。接着, 利用报警泛滥相似性分析找到与当前出现的报警序列相似的历史报警泛滥数据。最后, 用与当前报警序列拥有相同子序列的历史报警泛滥序列来预测报警事件的发生。

表 6.16 报警事件的描述

过程变量	报警限 (类型)	符号	单位
产品分离器温度.	89.5 (低)	x_1	°C
分离器冷却水温度	88 (低)	x_2	°C
回流流量	29.5 (低)	x_3	kscm/h
压缩机功率	250 (低)	x_4	kw
汽提塔液位	47 (低)	x_5	%
汽提塔温度	65 (低)	x_6	°C
反应器入口速率	49 (高)	x_7	kscm/h
反应器压力	2820 (高)	x_8	kPa
反应器液位	67 (高)	x_9	%
产品分离器液位	42 (低)	x_{10}	%
汽提塔压力	3360 (高)	x_{11}	kPa
汽提塔底部流量	24 (高)	x_{12}	m3/h
反应器冷却水温度	102 (低)	x_{13}	°C
产品分离器温度	95 (高)	x_{14}	°C
分离器冷却水温度	96 (高)	x_{15}	°C
回流速率	34 (高)	x_{16}	kscm/h
压缩机功率	290 (高)	x_{17}	kw
汽提塔液位	60 (高)	x_{18}	%
搅拌速率	0.7 (高)	x_{19}	kscm/h
汽提塔温度	70 (高)	x_{20}	°C
反应器入口速率	45 (低)	x_{21}	kscm/h
反应器液位	60 (低)	x_{22}	%
产品分离器液位	60 (高)	x_{23}	%
搅拌速率	0.4 (低)	x_{24}	kscm/h
回流速率	34.5 (高)	x_{25}	kscm/h
反应器压力	2650 (低)	x_{26}	kPa
汽提塔压力	3020 (低)	x_{27}	kPa
分离器底部流量	22.3 (低)	x_{28}	m3/h
汽提塔底部流量	20.5 (低)	x_{29}	m3/h
搅拌速率	0.2 (低)	x_{30}	kscm/h
反应器冷却水温度	130.5 (高)	x_{31}	°C
压缩机功率	376 (高)	x_{32}	kW
分离器底部流量	27 (高)	x_{33}	m3/h

参考文献

- [1] HOLLIFIELD B, HABIBI E. The alarm management handbook[M]. 2nd ed. Houston: PAS, 2010.
- [2] International Electrotechnical Commission. Management of alarm systems for the process industries[M]. International Electrotechnical Commission, 2014.
- [3] WANG J D, CHEN T W. An online method to remove chattering and repeating alarms based on alarm durations and intervals[J]. Computers and Chemical Engineering, 2014, 67 (4):43-52.
- [4] WANG J D, CHEN T W. Main causes of long-standing alarms and their removal by dynamic state-based alarm systems[J]. Journal of Loss Prevention in the Process Industries, 2016, 43:106-119.
- [5] KONDAVEETI S R, IZADII I, SHAH S L, et al. Quantification of alarm chatter based on run length distributions[J]. Chemical Engineering Research and Design, 2013, 91 (12):2550-2558.
- [6] International Society of Automation. ANSI/ISA-18.2. Management of alarm systems for the process industries[S]. International Society of Automation, 2015.
- [7] NIMMO I. Rescue your plant from alarm overload[J]. Chemical Processing, 2005, 4:28-33.

附录 A 基于密度峰值的聚类算法

本节对基于密度峰值的聚类算法^[14]进行简介。DPC算法将比其邻居拥有更高密度且距离更高密度点较远的数据点作为聚类中心点。设总共有 N 个数据点，记高维空间中第 i 个数据点与第 j 个数据点的欧式距离为 d_{ij} , $1 \leq i, j \leq N$ 。第 i 个数据点的局部密度 ρ_i 定义为

$$\rho_i = \sum_{j=1}^N \exp\left(-\left(d_{ij}/d_c\right)^2\right).$$

截断距离 d_c 定义为数据集 $\{d_{ij}\}$ 的第2百分位数。DPC算法与 ρ_i 的相对大小有关，因而 d_c 的数值大小并无显著影响。第 i 个数据点的距离 δ_i 定义为其与所有密度更高的点的距离中的最小值，即，

$$\delta_i = \min_{j: \rho_j > \rho_i} (d_{ij}).$$

密度最高点的距离定义为

$$\delta_k = \arg \max_i (\rho_i) = \max_j (d_{kj}).$$

$[\rho_i, \delta_i]$ 的散点图是DPC算法的决策图。根据决策图可以指定聚类中心点。有两种方法。一种方法是首先选定聚类数量为 m ，再计算欧式距离 D_i 如下，

$$D_i = \sqrt{\left(\frac{\rho_i}{\max_j (\rho_j)}\right)^2 + \left(\frac{\delta_i}{\max_j (\delta_j)}\right)^2}, \quad (\text{A.1})$$

并选择前 m 个最大 D_i 的数据点作为聚类中心。另一种方法是选择 ρ 和 δ 的阈值，则数据点拥有大于密度阈值 ρ 的 ρ_i 和距离阈值 δ 的 δ_i 都成为聚类中心。本书所提出的方法中采用了这两种方法，其中第一次聚类使用了第一种方法，第二次聚类使用了第二种方法。当聚类中心确定后，其余数据点都被分配至离它最近的高密度聚类中心所属聚类中。

由于数据点数量为 N ，可得到所有欧式距离 d_{ij} 的时间复杂度为 $O(N^2)$ 。计算所有密度 ρ_i 的时间复杂度为 $O(N)$ 。最差情况下，计算所有 δ_i 的时间复杂度为 $O(N \log N)$ 。最终分配聚类的过程的时间复杂度为 $O(N)$ 。因此DPC算法的时间复杂度为 $O(N^2)$ 。该复杂度在离线计算的情况下是可以接受的。

附录 B 凸包模型参数的计算

本节介绍如何从历史数据中估计凸包模型的参数。因为超平面 $p^{(i)}$ 通过凸包面 $f^{(i)}$ 的 n 个顶点，分别是 $X(I_{i,1})$, $X(I_{i,2})$, ..., 和 $X(I_{i,n})$ 。这些顶点可以通过文献 [18] 中的 quickhull 算法计算得到，该算法返回一个 $m \times n$ 阶的矩阵 I 。其中， m 表示凸包的面数， n 表示过程变量的个数。矩阵 I 的第 i 行第 j 列元素 $I_{i,j}$ 表示凸包的第 i 个面的第 j 个顶点的索引号。将这 n 个顶点代入公式(4.60) 中可以得到

$$a^{(i)} X^T(I_{i,k}) - b^{(i)} = 0, \quad k = 1, 2, \dots, n. \quad (\text{B.1})$$

因为 $X(I_{i,k}) = [x_1(I_{i,k}), x_2(I_{i,k}), \dots, x_n(I_{i,k})]$ ，所以公式(B.1) 可以被改写成

$$x_1(I_{i,k})a_1^{(i)} + x_2(I_{i,k})a_2^{(i)} + \dots + x_n(I_{i,k})a_n^{(i)} - b^{(i)} = 0. \quad (\text{B.2})$$

其中， $k = 1, 2, \dots, n$ 。公式(B.2) 表示一个含有 n 个方程和 $n+1$ 个变量的线性方程组。不失一般性，将 $b^{(i)}$ 设为自由变量。利用高斯消元法，可以得到公式(B.2) 的解，它们的表述形式被定义为

$$a_k^{(i)} = g_k(b^{(i)}), \quad k = 1, 2, \dots, n. \quad (\text{B.3})$$

需要注意的是， $a_k^{(i)}$ 和 $b^{(i)}$ 之间的数学关系可以通过公式(B.2) 推导得到。然而，它们的解析形式过于复杂，因此我们用符号 $g_k(\cdot)$ 来代替。此外， $a^{(i)}$ 是一个单位向量。也就是说，

$$a_1^{(i)} \cdot a_1^{(i)} + a_2^{(i)} \cdot a_2^{(i)} + \dots + a_n^{(i)} \cdot a_n^{(i)} = 1. \quad (\text{B.4})$$

联立公式(B.3) 和(B.4) 可以得到 $b^{(i)}$ 的解。然后，将 $b^{(i)}$ 的解代入到公式(B.3) 中，可以得到 $a_1^{(i)}$, $a_2^{(i)}$, ..., $a_n^{(i)}$ 。将 i 的值分别设为 $1, 2, \dots, m$ ，可以得到凸包的所有超平面的数学表达式，从而可以得到公式(4.59) 中的凸包模型的参数。