University of Alberta

Data-driven Techniques on Alarm System Analysis and Improvement

by

Yue Cheng

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Control Systems

Department of Electrical & Computer Engineering

Fall 2013
Edmonton, Alberta

Dedicated to my wife and my parents for their support and love throughout my life.

Abstract

To meet the demands of safety, quality and efficiency, process monitoring is of great importance. However, a serious problem exists in the industry: too many alarms are raised for operators to handle. Consequently, techniques need to be developed in order to reduce nuisance and false alarms to an acceptable level. Motivated by this, my thesis focuses on alarm systems improvement, specifically the development of data-driven techniques for the analysis and design of alarm systems.

Developed methods are based on either process data or alarm data, the two types of data mainly used in alarm systems. Three problems are considered. First, a univariate alarm signal filtering technique is discussed. The design of an optimal alarm filter for the best alarm accuracy, namely, minimizing a weighted sum of false and missed alarm rates (probabilities), is presented. Moreover, a sufficient condition for moving average filters being optimal linear alarm filters is also provided. Second, alarm flood pattern analysis based on multivariate alarm data is addressed. A modified Smith-Waterman algorithm considering time stamp information is proposed for alarm flood pattern matching. Third, the application of a new multivariate statistical analysis technique, the principal component pursuit (PCP) method, to process monitoring is thoroughly discussed. An optimal scaling method is proposed as the preprocessing step. A coordinate descent algorithm is provided to search for the optimal scaling vector, whose global convergence is proved. After multi-

variate process modeling, a PCP-based fault detection and diagnosis approach is introduced.

An industrial project on a major extraction process in Alberta to improve its alarm system is described. Based on the historical data, a predicted reduction on the alarm rate by applying a variety of alarm system rationalization techniques is estimated. A significant improvement on the alarm system is expected.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Dr. Tongwen Chen. He allowed me the freedom to conduct the research that is interesting to me, while providing help to keep me on the right track. Working under his supervision was one of the most positive experiences of my life. I would also like to thank Dr. Sirish L. Shah. I also thank my PhD committee members, Dr. Daniel Miller, Dr. Stevan Dubljevic, and Dr. Alan Lynch, for their valuable suggestions to improve the contents and presentation of this thesis. Thanks are also due to my family; my parents, my wife for bearing with me for all this time. Finally, I would like to thank all members of our research group for active discussions and sharing their research.

Contents

List of Tables

List of Figures

List of Symbols

$\|\mathbf{X}\|_*$	nuclear norm (sum of singular values)
$\|\mathbf{X}\|_0$	0 norm of a matrix (number of non-zero entries)
$\|\mathbf{X}\|_1$	ℓ_1 norm of a matrix (sum of magnitudes of all entries)
$\|\mathbf{X}\|_\infty$	ℓ_∞ norm of a matrix (maximum magnitude of all entries)
$\text{diag}(\mathbf{X})$	column vector whose entries are the diagonal entries of the square matrix $\mathbf{X}$
$\mathbf{D}(\mathbf{x})$	diagonal and square matrix with the elements of $\mathbf{x}$ on its diagonal
$\max(\mathbf{x})$	the maximum value in a vector or a set $\mathbf{x}$
$\min(\mathbf{x})$	the minimum value in a vector or a set $\mathbf{x}$
$\lceil x \rceil$	smallest integer that is no less than x
$\mathbf{I}$	identity matrix with compatible size
$\mathbf{0}$	zero vector or matrix with compatible size
$\mathbf{1}$	vector whose entries are all 1
$\mathbf{e}_i$	the i-th standard basis vector with compatible length
$\mathbf{y} > \mathbf{0}$	all entries of a vector $\mathbf{y}$ are greater than 0
$\emptyset$	empty set

List of Abbreviations

AADA	Automatic Alarm Data Analyzer
ADD	Average Detection Delay
ANN	Artificial Neural Network
ARL	Average Run Length
BPCS	Basic Process Control System
CDF	Cumulative Distribution Function
CUSUM	CUmulative SUM
DCS	Distributed Control System
DE	Differential Evolution
EEMUA	Engineering Equipment & Materials Users' Association
EWMA	Exponentially Weighted Moving Average
EWMS	Exponentially Weighted Mean Square
EWMV	Exponentially Weighted Mean Variance
FAR	False Alarm Rate
FIR	Finite Impulse Response
GLR	Generalized Likelihood Ratio
GSD	Generalized Sequential Patterns
HMI	Human Machine Interface
IHR	Increasing Hazard Rate

IIR	Infinite Impulse Response
ISA	International Society of Automation
LLR	Log-Likelihood Ratio
MAR	Missed Alarm Rate
PCA	Principal Component Analysis
PCP	Principal Component Pursuit
PCS	Principal Component Subspace
ROC	Receiver Operating Characteristic
RS	Residual Subspace
SDG	Signed DiGraphs
SIS	Safety Instrumented System
SPC	Statistical Process Control
SVD	Singular Value Decomposition

Chapter 1

Introduction

1.1 Motivation and Background

In large industrial plants, faults may occur within any of their thousands of components and therefore result in unsatisfactory performance, equipment failures, and even hazardous accidents. According to [1], petrochemical plants on average suffer a major accident every three years that leads to devastating consequences. Moreover, it costs industries billions of dollars annually due to unplanned shutdown, equipment damage, performance degradation and operation failure.

To meet the demands of safety, quality and efficiency, process monitoring is of great importance. Alarm systems play an important role in industry. An online survey conducted by the International Society of Automation (ISA) in 2008 indicated that "what automation industry observers and practitioners felt that near-term trends were going to be showed that alarm management and security scored at 14%, one of the top 5 technologies that the facility would rely on" [109]. As defined in [15],"an alarm is some signal designed to alert, inform, guide or confirm, and an alarm system is a system for generating and processing alarms and presenting them to users". A detailed description of an alarm system is provided in [58]. An alarm system may include an alarm generating part in the basic process control system (BPCS) and safety instrumented system (SIS), the alarm log, and the communicating part to operator via a human machine interface (HMI). There are also some external systems that are of importance to the alarm system including an alarm historian. The schematic of an alarm system is shown in Fig. 1.1.

According to the alarm management lifecycle described in [58], control engineers are mainly involved in the "monitoring and management of the

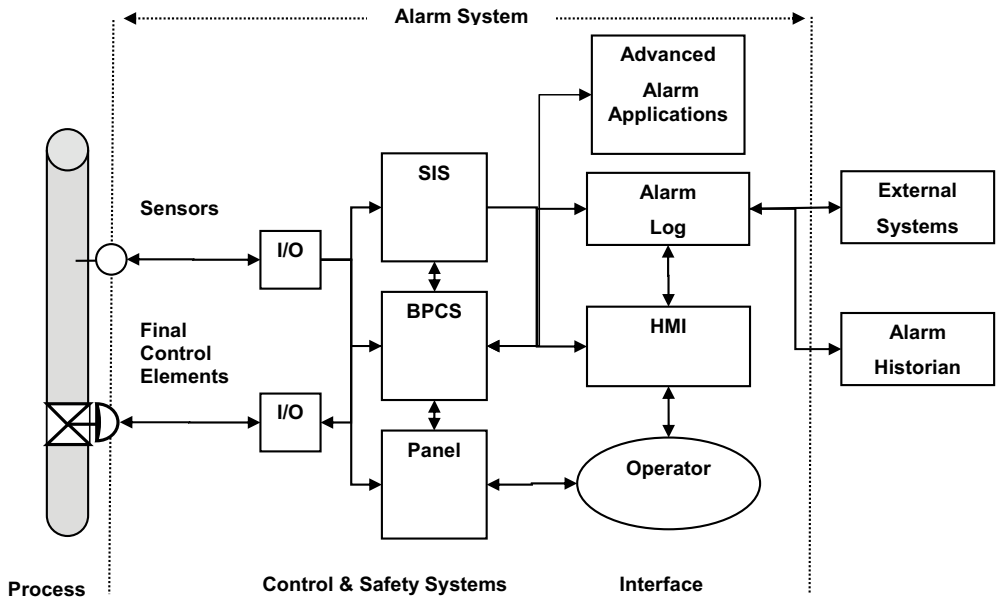

Figure 1.1: Alarm system dataflow [58].

change loop". This loop is triggered when the alarm system is working as designed while the monitoring and assessment requirements are not achieved; so new techniques should be adopted to improve the performance of the alarm system.

Nowadays, hardware and software advances, in particular the modern distributed control system (DCS) technique, make it easy to access almost every process variable that can be measured; this provides the alarm system with more process data. However, the modern DCS also makes it easy to add alarms without significant effort, cost, or justification, which makes alarm management more challenging. It is shown in [93] that an average operator takes about 10 minutes to process and respond to an alarm, so EEUMA Guide and ISA Standard [58, 93] suggest that an operator should not receive more than six alarms per hour. To find more detailed alarm performance metrics and their target values, one can refer to the "alarm performance metrics summary" section in [58]. There is a large gap between these performance requirements and the reality observed in industry. In practice the number of alarms an operator receives per hour is in the magnitude of tens, hundreds or even thousands. A majority of these alarms are false or nuisances. As described in [87], "end users are swamped with alarms, only some of which

require any real action to be taken". Too many false and nuisance alarms only distract the operator from operating the plant, which can cause critical alarms to be ignored, and lead to distrust of the alarms by operators. Some incident cases reported in [15] also support the statement that serious faults which may cause incidents are probably disguised by false and nuisance alarms in alarm floods. As a result, there exist strong industrial needs and major potential economic benefits for better interpreting and managing the alarms, as well as redesigning the alarm systems to reduce false and nuisance alarms and to avoid alarm floods.

As this thesis mainly focuses on data-driven techniques, it is necessary to clarify the data related to alarm management. There are two types of data that are informative and accessible resources for alarm management: process data and alarm data [62].

Process data are process variable measurements. They are usually stored in a database. An example of a segment of process data extracted from a process database is shown in Fig. 1.2.

	300LC1066	300FI2031	300LC2066	300FI2117
Timestamp	300LC1066 - SnapShot	300FI2031 - SnapShot	300LC2066 - SnapShot	300FI2117 - SnapShot
2012/8/23	56.48187256	3070.739502	54.74227142	111.2259598
2012/8/23 0:01	57.61133957	3076.950684	54.74488449	111.7646942
2012/8/23 0:02	59.06160736	3037.957031	54.74749756	111.465332
2012/8/23 0:03	59.04538727	3061.270996	54.75011063	112.242424

Figure 1.2: A segment of process data extracted from the database.

The historical process data is a valuable resource for model identification, controller design, and performance analysis. In industry, most of control related alarms are raised based on process data. One process variable or a function of several process variables provides the alarm system with an alarm signal. Alarms are usually raised when the corresponding alarm signal exceeds a predetermined limit.

Alarm data is a set of text messages generated by the DCS and stored in an alarm log. When an alarm is raised, a message is generated. Usually an alarm message contains several fields of information: time stamp, namely, the time instant when the message is generated, tag name, tag identifier, e.g., 'PVHI', 'PVLO', 'OFFNORM', and other information such as the priority, the value of the process variable, the trip point and so on [62]. The tag name plus tag identifier reflects what type of alarm occurs, and the time stamp reflects when it occurs. An example of a segment of alarm data extracted from an alarm

log is shown in Fig. 1.3.

EventID	Timestamp	Alarm Identifier	Message Type	Tag	Tag Descr	Plant	Area	Priority	Limit	Value
52870289	3/14/2009 19:02:58.000	PVLO	Alarm	39TI317	STG3 PEW	Base Plar	Primary Extraction	HIGH	170	169.261
52870290	3/14/2009 19:03:07.000	PVLO	Alarm	82PC866	HPW FROM	Base Plar	Primary Extraction	HIGH	55	54.809
52870398	3/14/2009 19:04:49.000	PVLO	ReturnToNormal	82PC866	HPW FROM	Base Plar	Primary Extraction	HIGH	55	56.041
52870399	3/14/2009 19:04:58.000	PVLO	ReturnToNormal	39TI317	STG3 PEW	Base Plar	Primary Extraction	HIGH	170	176.636
52870654	3/14/2009 19:10:58.000	PVLO	Alarm	39TI317	STG3 PEW	Base Plar	Primary Extraction	HIGH	170	167.356

Figure 1.3: A segment of alarm data extracted from the database.

To clarify the relationship of process data and alarm data, a simple example is provided. Assume that there are only two process variables, x_1 and x_2, in an alarm system. For each process variable, two thresholds (an upper limit and a lower limit) are set. Under normal situations, the process variables should operate between their upper and lower limits. In other words, when their values are larger (smaller) than their upper (lower) limits, PVHI (PVLO) alarms will be raised, while alarms will be cleared when the process variables return to their normal regions between their upper and lower limits. Hence, there are a total of 4 types of alarms: x_1.PVHI, x_1.PVLO, x_2.PVHI, and x_2.PVLO. Fig. 1.4 shows the generation of these 4 types of alarms according to the process variables x_1 and x_2. At the instant that a process variable exceeds a threshold, a red bar is plotted on the corresponding alarm plot. For example, the process variable x_1 crosses the upper limit at 30s, hence a red bar is shown at 30s on the alarm x_1.PVHI plot. Table 1.1 provides the corresponding alarms in the log.

1.2 Literature Survey

1.2.1 Guides and the Standard on Alarm Management

The topic of alarm management draws much attention from related industrial societies. In 1998, a contract research report for the Health and Safety Executive was prepared by Bransby Automation Ltd [15]. This report surveyed alarm systems in chemical and power industries, justified the value of improvement on alarm systems, and summarized the best practice at that time. From this report, the Engineering Equipment & Materials Users' Association (EEMUA) developed a guide on alarm systems. The revised version from 2007 became a document widely accepted in industrial alarm practice and design [41]. This guide covered topics on alarm system philosophy, design principles of alarm systems, implementation issues, measuring performance, managing

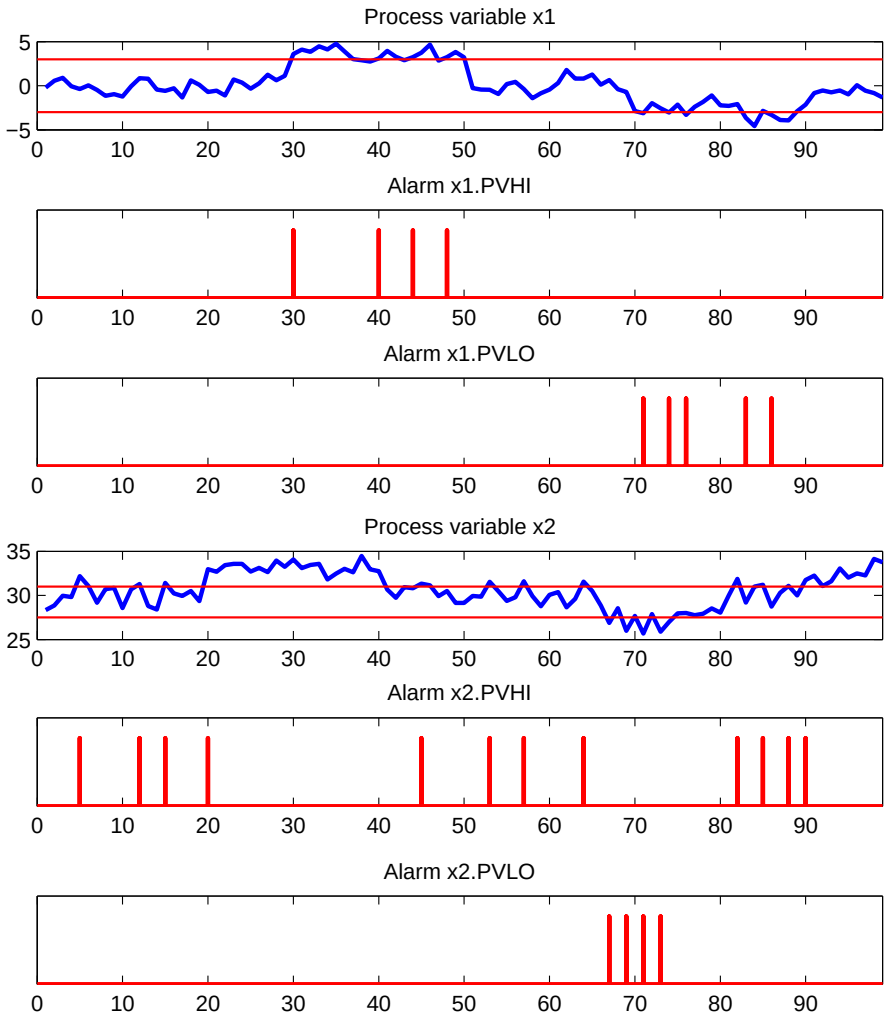

Figure 1.4: Example of alarm generation.

Table 1.1: Alarm message log.

Tag Name	Tag Identifier	Time Stamp
x_2	PVHI	5s
x_2	PVHI	12s
x_2	PVHI	15s
x_2	PVHI	20s
x_1	PVHI	30s
x_1	PVHI	40s
x_1	PVHI	44s
x_2	PVHI	45s
x_1	PVHI	48s
x_2	PVHI	53s
x_2	PVHI	57s
x_2	PVHI	64s
x_2	PVLO	67s
x_2	PVLO	69s
x_1	PVLO	71s
x_2	PVLO	71s
x_2	PVLO	73s
x_1	PVLO	74s
x_1	PVLO	76s
x_2	PVHI	82s
x_1	PVLO	83s
x_2	PVHI	85s
x_1	PVLO	86s
x_2	PVHI	88s
x_2	PVHI	90s

improvement programs, and purchasing of new alarm systems.

The standard on Management of Alarm Systems for the Process Industries developed by ISA [58] is seen as a milestone in the field of alarm management, since it is the first standard in this field. The importance of the standard has been emphasized in many articles such as [50, 87, 101]. The standard dealt with the development, design, installation, and management of alarm systems in process industries to improve safety, quality and productivity. This standard is now being popularized in North America.

The documents mentioned above outlined alarm strategy development, as well as provided alarm performance metrics and the recommended target values. However, only the direction was made clear and the targets provided; technical details in order to reach these goals were not discussed.

1.2.2 Technical Work

Due to the great demand from industry, some research work has been done in the field of alarm management. The goal of these studies is to make alarm systems more accurate, swifter, and more informative, since the end users are operators who expect accuracy when alarm systems alert them to real abnormalities without significant delay, and instruct them to act appropriately via the HMI in a way that is easy to understand.

Most of the studies can be classified into three major topics: univariate alarm analysis and processing, multivariate alarm analysis and processing, and alarm system visualization and operator overload analysis. Regardless of the correlations between tags, univariate signal analysis and processing focus on only one tag at a time. Techniques are proposed to make the alarm operation on this single tag as accurate and swift as possible. In contrast, multivariate studies emphasize the relationship between tags. Problems such as plant connectivity analysis, alarm correlation and propagation analysis are considered. These two topics are from the alarm system point of view, while the third topic is from the end users' perspective. Such problems as how to estimate the workload of an operator, as well as how to present the detection result of the alarm system to operators and engineers are studied.

Univariate Alarm Analysis and Processing

In [60] and [61], a general framework for univariate alarm systems was proposed. In this framework, the three most important metrics are false alarm

rate (FAR), missed alarm rate (MAR), and average detection delay (ADD). Receiver operating characteristic (ROC) curves are recommended to analyze the tradeoff on FAR and MAR with respect to the threshold. As discussed in [61], three main techniques to improve the alarm performance are deadbands, delay timers, and filters. The guideline to set a proper deadband was discussed in [55]. Reference [82] focused on the relationship between deadband and chattering alarms. The papers [3, 115] systematically discussed the delay timer design based on these three metrics. The design of alarm limit based on historical process data was studied in [51].

The filter design problem in univariate alarm system is closely related to the abrupt change detection in the statistical process control community, whose goal is also to adopt different signal processing techniques, especially filtering techniques, to detect abrupt changes of the signal. Instead of FAR and ADD, ARL0 and ARL1 are used to measure the performance, where ARL0 indicates the average run length when a process is in control, and ARL1 indicates the average run length when a process is out of control. Actually, there is a straightforward relationship between ARL0 and FAR; moreover, ADD and ARL1 are the same.

The tools most utilized for univariate statistical process monitoring are different kinds of control charts. The basic control chart, Shewhart chart, was firstly proposed by Walter A. Shewhart in 1920s. As the most classical univariate statistical monitoring strategy, the Shewhart chart is often introduced at the very beginning in textbooks [14, 32]. Note that 2 sigma or 3 sigma, sigma being the standard deviation, is often set as the control limit for measurements, and false alarm rate is approximately calculated under the Gaussian distribution assumption. Some additional action rules for the Shewhart chart were devised by Western Electric [43] which can be considered as simple signal processing methods on the process signals. For example, the third rule can be considered as filtering the maximum/minimun value of the latest 8 samples and its threshold is the operating point.

Some popular advanced control charts include cumulative sum (CUSUM) chart and the exponentially weighted moving average (EWMA) chart. These advanced control charts filter a process signal by certain linear or nonlinear filters, and then compare the filtered signal to a threshold. ARL0 and ARL1 are the metrics most researchers focus on. The CUSUM chart was first proposed

by Page in 1954 [88]; it is in the form of a nonlinear filter

$$g_{n+1} = \max\left(g_n + \log\frac{p_{\Theta_1}(X_n)}{p_{\Theta_0}(X_n)}, 0\right),$$

where $\log\dfrac{p_{\Theta_1}(X_n)}{p_{\Theta_0}(X_n)}$ is the log-likelihood ratio (LLR) of abnormal and normal cases. This form is convenient for theoretical analysis, while the following simplified form is more practical:

$$g_{n+1} = \max\left(g_n + X_{n+1} - K, 0\right).$$

EWMA, also called geometric moving average, is in an infinite impulse response (IIR) filter form

$$g_{n+1} = (1-\lambda)g_n + \lambda X_{n+1}.$$

Considering variance instead of mean change, EWMS (exponentially weighted mean square) or EWMV (exponentially weighted mean variance) are studied.

Research on the advanced control charts mainly includes evaluation of metrics, setting the parameters to improve performance, removal of assumptions such as i.i.d. and modification of existing methods, as well as combining these control charts with certain intelligent methods. An approach to approximate the probability distribution of ARL1 by discretization was provided in [16]. A recent work [118] extensively studied the distributions of ARL0 and ARL1 by considering the CUSUM signal as a Wiener process. A similar work was done for EWMA in [94] with the assumption of normal distribution by a numerical method. A comprehensive analysis of ARL0 considering both independent and autocorrelated observations for EWMS and EWMV was provided in [80]. But the result was approximated and was under the assumption of normal distribution. In [100], the optimal weights for EWMA were designed in the sense of the minimum ARL1 with a fixed control limit under the i.i.d. and normal distribution assumptions. In [16], a new objective function was provided while considering the quality impacts on run length. The shift size was random but following a known normal distribution, and the optimal parameters for EWMA and CUSUM for different distributions of shift size were listed in a table. In [106], an adaptive threshold for CUSUM was designed. The threshold was calculated by Monte Carlo experiments at each step. Certain combinations of artificial neural network (ANN) and CUSUM or EWMA were attempted in [31, 108]. The ANN was used as a residual generator and the

generated residue was filtered by CUSUM or EWMA. A comprehensive summary and analysis on CUSUM-type and generalized likelihood ratio (GLR) algorithms were provided in [8]. We can see that most of the results were under Gaussian distribution or exponential family distribution assumption.

Multivariate Alarm Analysis and Design

It was shown in [51] that a large number of unimportant alarms fall within three categories: standing alarms, repetitive alarms, and consequence alarms. Univariate alarm design techniques mentioned above are effective on standing alarms and repetitive alarms, especially for bad alarm tags.

Under steady operating conditions, the alarm rate can be effectively reduced to a manageable level thanks to these techniques. However, in abnormal situations, it is still very difficult to eliminate alarm floods though these techniques can suppress some of the alarms. The remaining alarms are mainly consequence alarms. Recently, the connection between alarm flood and plant incidents is thoroughly discussed from industrial perspective in [10]. The authors claimed that alarm floods usually occur when the plant state changes. Univariate approaches such as bad tag management and static rationalization can hardly have any effect on mitigation of alarm floods.

Compared with standing and repetitive alarms, consequence alarms are more complicated and more difficult to manage. Multivariate alarm analysis and design concern the relationship between alarms, which is appropriate for analysis and reduction of consequence alarms.

Multivariate analysis and design can be classified into two categories: quantitative methods, and qualitative methods.

Multivariate statistical process control (SPC) methods are popular and extensively studied quantitative methods. The methods often use the principal component analysis (PCA) technique as a modeling tool, and the T^2 and Q statistics of the PCA model are often used for alarm generation – details can be found in [32] and the references therein. In [66], it was shown by simulation results on Tennessee Eastman process that the PCA method could improve the accuracy of the performance of the alarm system. PCA-based abnormality diagnosis was proposed and analyzed in [63, 96, 97]. In these references, the angle and distance between the principal subspaces for historical and current multivariate process data were similarity factors by which similar historical segments were screened to help experts make their decisions. The methods

were validated on a simulated continuous stirred-tank reactor model and the Tennessee Eastman process. Automation on abnormality diagnosis discussed in these references can greatly reduce the operators' workload when alarm floods occur and thus is a rational resolution for alarm flood problems. Other techniques were proposed to quantitatively determine the abnormal operating region for alarm generation including the parallel coordinate envelope technique [57], and a dynamic process simulator-based method [105]. The former technique is intuitive and easy to understand, but the operation region can only be determined by rule of thumb; the latter technique requires an exact dynamic model of the plant, and only considers a noise-free situation, which is not practical.

Intelligent alarm processing was introduced as a traditional quantitative method in [64, 111] and attempted to present a clear picture of the global situation. A rule-based framework was provided. When alarms are raised, the system will decide the fault pattern following these rules. This alarm processing method is commonly used in transmission and distribution systems [21, 103]. A significant problem for such kind of alarm processing methods is how to create knowledge bases for different systems [112]. In one implementation mentioned in [103], there were more than 15,000 rules (called chronicles in the paper), which were mainly based on human expert experience on the specific system. Some connectivity modeling tools such as signed digraphs (SDG) are helpful for abnormality propagation path searching and hence for setting up the rule database [104]. However, such tools are still knowledge-based methods, which usually require involvement of expert, and are very time consuming to implement.

To reduce analysis time, data mining techniques are a good choice. Such data-driven techniques as Granger causality [48] and transfer entropy [9, 39] capture process connectivity based on process data. However, the results are usually not very reliable and used only for verification of the knowledge-based connectivity modeling.

Other researchers attempted to directly detect the connection among alarm tags based on alarm data. Event correlation analysis approaches were discussed in [85, 86, 117]. If the consequential relationship of two alarms is fixed, correlation analysis approaches work well. For example, in [85] the correlation analysis method can easily find an alarm sequence caused by a daily routine operation. However, if the relationship varies in different abnormal situations,

analysis on the whole alarm event sequences instead of pairs of alarm tags is more suitable. Some pioneering work has been done on the application of sequential pattern mining to consequence alarm suppression [33, 45, 46, 54]. In [33, 54], the authors modified the generalized sequential patterns (GSP) algorithm [99] to search for frequent alarm sequences (patterns) in the historical alarm records. They emphasized that aside from the chronological order, triggering time was also very important, so that time stamps of alarms were considered in order to add time constraints in their algorithm. They also pointed out that a potential use of the pattern analysis result was dynamic alarm suppression. Since one alarm pattern is usually related to a specific underlying abnormality, all of the alarms in a certain pattern can be temporarily suppressed when the corresponding abnormality is detected. In [45, 46], the authors set up Automatic Alarm Data Analyzer (AADA) automatons and alarm-sequence automatons to represent the frequent alarm sequences in historical alarm records. However, it was also mentioned that one of the deficiencies of the algorithm was the lack of robustness to disturbances, e.g., similar alarm sequences with only one different alarm were recognized as different alarm sequences.

Alarm System Visualization and Operator Workload Analysis

Visualization is a practical issue in alarm systems. An ineffectively visualized HMI hinders the transfer of information from the alarm system to operators. Even if the alarm system can make decisions accurately and swiftly, namely, the end users, the operators, can hardly benefit from the alarm system. Visualization of historical data is also significant for the alarm rationalization group to understand the performance of the alarm system.

The parallel coordinate technique [57] is a good practice on process data visualization, which aligns many process variables in a single coordinate. The additional benefit of 3D visualization was justified in [53]. The study measured the reaction time and processing time to handle problem situations in 3D visualizations. Compared with 2D, the results showed an advantage for 3D representation. The authors of [45] further discussed how to design 3D process control objects.

A series of visualization tools to represent historical alarm data were proposed in [65]. High density alarm plots, alarm similarity color maps, and run length distribution plots were developed.

An operator behavior model was established in [78], and then a simulator based on the model was used to evaluate the performance of an alarm system. This work provided researchers with a new perspective on alarm system evaluation.

1.2.3 Implementation of Alarm Management Techniques

An alarm management tool box was introduced in [11, 70]. Nuisance alarm reduction via existing DCS functions such as filtering, time delay, and dead band was discussed. However, in these papers, the only method for choosing the algorithm was practicing several algorithms simultaneously and selecting the one with the best performance. In [77], a software was introduced to resolve nuisance alarms. Some simple rules were used to determine whether an alarm was a nuisance, and subsequently the control limit would be dramatically changed to keep the alarm activated in a nuisance alarm case.

1.2.4 Summary

After going through a survey of the existing literature, we find that there is overwhelming evidence for the univariate alarm signal processing problem; however these results mainly concentrate on several fixed forms of control charts. Although some papers study the optimal setting of the control chart parameters, the results are often derived under the normal distribution assumption.

In the multivariate case, rule-based alarm pattern analysis has been developed and applied to some power systems, but the research on data-driven techniques is still quite limited.

1.3 Thesis Contributions

The major contributions in this thesis that distinguish it from other work are listed below:

1. Provided a performance index to measure alarm accuracy, and developed a procedure for optimal linear alarm filter design.

2. Proved that under certain conditions, moving average filters are optimal linear filters in terms of high alarm accuracy.

3. Proposed a novel similarity index and a corresponding modified Smith-Waterman algorithm for event sequences with time stamps to measure the similarity of two alarm flood sequences.

4. Pointed out the importance of scaling in the PCP technique, and proposed a novel and convergent algorithm to optimize the scaling parameters in the sense of low coherence to improve the quality of the data matrix.

5. Provided an easy and practical method to detect oscillation tags based on alarm data.

1.4 Thesis Outline

This thesis has been prepared according to the guidelines from the Faculty of Graduate Studies and Research (FGSR) at the University of Alberta. The rest of the thesis is organized as follows.

Chapter 2 is concerned with univariate alarm filter design. We present the design of an optimal alarm filter for the best alarm accuracy, minimizing a weighted sum of false and missed alarm rates (probabilities). It turns out that the general form of such optimal alarm filters is the so-called LLR filters, which can be highly nonlinear and difficult to implement in practice. With fixed filter structures (FIR), design of optimal linear alarm filters is studied, and a numerical optimization procedure is proposed. Some key elements in the optimal design include the use of characteristic functions from probability theory to facilitate computation of the objective function, and a differential evolution (DE) algorithm for optimization (the optimization problem is non-convex and with small gradients). A sufficient condition for moving average filters being optimal linear alarm filters is also provided.

Chapter 3 is concerned with alarm flood pattern analysis which is helpful for root cause analysis of historical floods and for incoming flood prediction. In this chapter, a data driven method for alarm flood pattern matching is proposed. An alarm flood is represented by a time-stamped alarm event sequence. A modified Smith-Waterman algorithm considering time stamp information is proposed to calculate a similarity index of two alarm floods. The effectiveness of the algorithm is validated by a case study on real industrial data from a major refinery process.

Chapter 4 discusses the application of a new statistical analysis technique, the principal component pursuit, to process monitoring. A new scaling preprocessing step is proposed to improve quality of data matrices for low coherence. An algorithm is proposed for optimal scaling vector search. The convergence of the algorithm to a global optimal point is also proved. A residual generator and a post-filter suitable for PCP generated process models are also provided. The post-filtered residual represents the fault signal, which makes the fault detection, isolation, and reconstruction procedure simple and straightforward. A numerical example is also provided to illustrate the PCP-based process modeling and monitoring procedure.

Chapter 5 describes an alarm system rationalization project on an industrial oil-sand extraction plant. The project is still in progress, and the first stage, univariate alarm rationalization, has finished. Delay timers and filters have been designed for the alarm tags that occur frequently (bad tags). An alarm data based oscillation detection method has also been provided to detect oscillation tags in the historical data set. Because of its simplicity, the method is also very easy to be implemented online.

Finally, Chapter 6 draws concluding remarks for this dissertation and presents some future work.

Chapter 2

Optimal Univariate Alarm Signal Processing: Filter Design and Performance Analysis*

2.1 Overview

In this chapter, filtering techniques on univariate alarm signals are investigated. In light of the tradeoff between accuracy and interpretability, we mainly focus on linear filters. An algorithm is provided for optimal linear filter design on alarm signals to minimize the weighted sum of false and missed alarm rates. The relationship between moving average filters, a kind of popular filters commonly applied, and optimal linear alarm filters is scrutinized.

2.2 Problem Formulation

Usually an alarm system follows some fault detection module that generates a residual signal. From the alarm processing perspective, we view the residual signal as the raw alarm signal, denoted by $x[k]$. It is assumed that $x[k]$ has two modes, normal and abnormal; and it may experience a variation from the normal mode to the abnormal mode when abnormality or fault occurs. An alarm filter is a system that processes this alarm signal for a better alarm activation.

Since swiftness is important for an alarm system, i.e., the detection delay should be reasonably small, we focus on the class of finite memory causal

*A version of this chapter has been published in [29], and a short version has been published in [28].

filters of length N:

$$y[k] = f\left(x[k], x[k-1], \cdots, x[k-N+1]\right). \tag{2.1}$$

Then we can determine whether there exists abnormality by comparing the filtered signal $y[k]$ with a threshold (usually named a trip point in industry). The goal of alarm filter design is to find a fixed length alarm filter as well as its threshold that improves the accuracy of the alarm system–reducing the false alarm rate (FAR) in the normal case and the missed alarm rate (MAR) in the abnormal case. We do not consider the transient phase since it lasts a short amount of time compared to the stationary phase. Although very important in the analysis of detection delay, the transient phase can be neglected in FAR and MAR calculations. As a result the objective function to be optimized is as follows:

$$\begin{aligned} J(f) = {} & c_1 P\left(f(x_{k:k-N+1}) > y_t \mid \text{normal mode}\right) \\ & + c_2 P\left(f(x_{k:k-N+1}) < y_t \mid \text{abnormal mode}\right), \end{aligned}$$

where y_t is the threshold of the filtered data, and $P(\cdot)$ denotes the probability. The objective function is a weighted sum of FAR and MAR when the length of the filter is fixed at N. Usually, both weights c_1 and c_2 are positive. Moreover, not the absolute values but the ratio of them affects the solution of the optimization problem. As a result, without loss of generality, we fix the sum of c_1 and c_2 to be 1.

Generally, the computation of the objective function requires the joint probability density functions (PDFs) of the discrete time series $x[k]$. For simplicity, here we assume that $x[k]$ is independent, and it is identically distributed on the normal mode and abnormal mode, respectively; thus we take the PDFs of $x[k]$ in the normal condition and abnormal condition as $p_n(x)$ and $p_a(x)$, respectively. Correspondingly, the filtered signal $y[k]$ has PDFs $p_{Yn}(y)$ and $p_{Ya}(y)$ in the two conditions. Notice that the filtered signal is still identically distributed, but it is no longer independent because the computation of $y[k]$ uses historical data of the alarm signal $x[k]$. The expression of objective function can be written as:

$$J(f) = c_1 \int_{y_t}^{+\infty} p_{Y_n}(y)\mathrm{d}y + c_2 \int_{-\infty}^{y_t} p_{Y_a}(y)\mathrm{d}y. \tag{2.2}$$

The optimization problem described by (2.2) without any constraint is a classic classification problem. The optimal classification is obtained via the log-likelihood ratio. A concise proof is given in [72].

The structure of the filter may be complicated if complicated PDFs are considered. While the industry prefers a filter with simple structure that is

easy to understand and implement. As a result, the optimal filters in the sense of fixed simple structures are also worth studying. In this paper, we will focus on linear filters. Moreover, because of the swiftness consideration we mentioned above, the finite impulse response (FIR) filters will be discussed. As a result, we constrain the structure of filter (2.1) as follows:

$$y[k] = \sum_{i=0}^{N-1} \theta_i x[k-i], \tag{2.3}$$

where θ_i are filter parameters. These filter parameters and the threshold y_t are what we will design.

Obviously, if we exchange the values of θ_i and θ_j, the distribution of filtered data $y[k]$ does not change. Therefore the value of the cost function will not be affected (but there will be influence on the detection delay). Moreover, if we simultaneously divide the θ_i's and the threshold y_t by a positive number, the value of the cost function will not be affected. As a result, we can fix θ_0 at 1, and constrain θ_1 to θ_{N-1} in the [-1, 1] hypercube without loss of generality.

However since we fix the first filter parameter to be positive, we should determine whether higher or lower than the threshold means abnormality. As a result, a slight modification is needed on the objective function:

$$J_m(\theta) = \min\left(J(\theta), J_1(\theta)\right), \tag{2.4}$$

where

$$\begin{aligned}
&\theta = [\theta_1, \theta_2, \cdots, \theta_{N-1}, y_t]^T; \\
&J(\theta) = c_1 \textstyle\int_{y_t}^{+\infty} p_{Y_n}(y|\theta)\mathrm{d}y + c_2 \int_{-\infty}^{y_t} p_{Y_a}(y|\theta)\mathrm{d}y; \\
&J_1(\theta) = c_1 \textstyle\int_{-\infty}^{y_t} p_{Y_n}(y|\theta)\mathrm{d}y + c_2 \int_{y_t}^{\infty} p_{Y_a}(y|\theta)\mathrm{d}y.
\end{aligned}$$

Linear filters are easy to implement on industrial systems. Moreover, if, by scaling, we take the sum of all filter coefficients to be 1, the filtered signal has a similar physical meaning to the original one, since in this case the steady-state gain of the filter is 1.

2.3 Optimal FIR Filter Design

Linear filters are most commonly used filters in industry. One of the most popular filters, the moving average filter in which all data are weighted the same, is included in this set. It can be proven that moving average filter is the linear least square filter to a stationary signal with i.i.d. zero-mean noise, which means it can minimize the energy of noise of the filtered signal.

Intuitively a smaller noise means a better classification. However a ques-

tion is whether the moving average filter is always an optimal linear filter in the sense of minimum false and missed alarm rates, namely objective function (2.4). If both normal and abnormal data follow Gaussian distributions, Gaussianities are kept by the filtered data, so that the objective function (2.4) only depend on the variance of the filtered data. Thus moving average filter is the optimal linear filter that provides the minimum objective function (2.4) over all linear filters. In general case, however, the answer is no. A very simple counterexample is given as follows.

Example 2.1. The PDFs of normal and abnormal data are:

$$p_n(x) = \begin{cases} 0.01 & x \in (-11,-1) \cup (1,11) \\ 0.4 & x \in [-1,1] \\ 0 & \text{elsewhere} \end{cases};$$

$$p_a(x) = \begin{cases} 0.01 & x \in (-9,1) \cup (3,13) \\ 0.4 & x \in [1,3] \\ 0 & \text{elsewhere} \end{cases}.$$

The distributions of the normal and abnormal data are shown in Fig. 2.1.

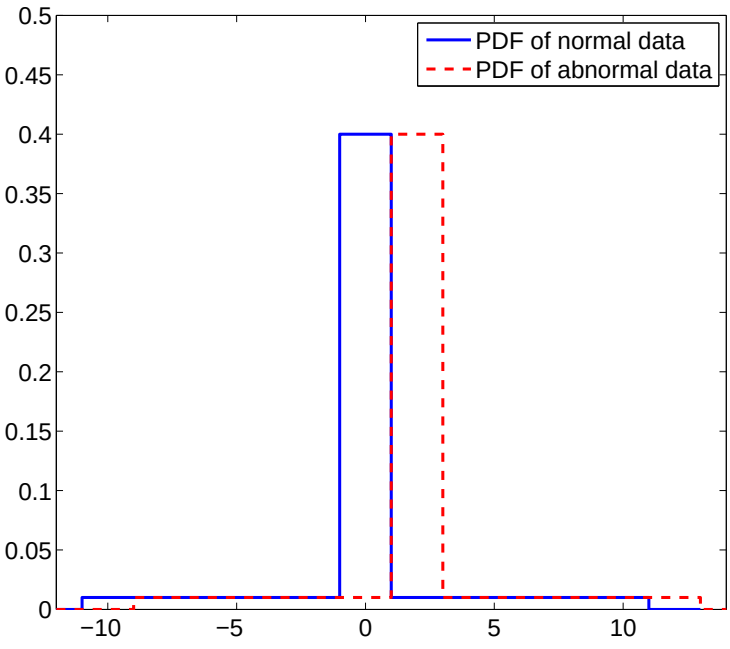

Figure 2.1: Distributions of normal and abnormal data of the counterexample.

Consider the case that length N is 2, and weight ratio is 1, which means we

emphasize MAR and FAR equally. We find that the moving average filter performs worse than some other linear filters and both the MAR and FAR are about 16%, which are 6% higher than the MAR and FAR just using raw data.

As a result, the optimal linear alarm filter design is not so trivial. Two questions we ask are: first, how to design an optimal FIR alarm filter; and second, in what cases is the moving average filter necessarily optimal? In this section, we focus on the first problem. The main difficulty is to calculate the PDF of $y[k]$ in objective function (2.2) from the PDF of $x[k]$, since for any optimization algorithm evaluating objective function is a necessary step. The calculation of PDF of linear combination of independent random variables requires multiple integral operations. In probability theory, characteristic functions are often used to simplify the computation in this case. The characteristic function is actually a kind of Fourier transform of the PDF, and because time domain convolution is equivalent to frequency domain multiplication, the number of times of integration will be greatly reduced when N is large, e.g., 10 or 15.

Applying the characteristic function idea on objective function (2.4), we obtain that

$$\begin{aligned} J(\theta) &= \tfrac{c_1}{2\pi}\int_{y_t}^{+\infty}\int_{-\infty}^{+\infty}\varphi_n(t)\prod_{i=1}^{N-1}\varphi_n(\theta_i t)e^{-jxt}\mathrm{d}t\mathrm{d}x \\ &+ \frac{c_2}{2\pi}\int_{-\infty}^{y_t}\int_{-\infty}^{+\infty}\varphi_a(t)\prod_{i=1}^{N-1}\varphi_a(\theta_i t)e^{-jxt}\mathrm{d}t\mathrm{d}x. \\ J_1(\theta) &= \frac{c_1}{2\pi}\int_{-\infty}^{y_t}\int_{-\infty}^{+\infty}\varphi_n(t)\prod_{i=1}^{N-1}\varphi_n(\theta_i t)e^{-jxt}\mathrm{d}t\mathrm{d}x \\ &+ \frac{c_2}{2\pi}\int_{y_t}^{+\infty}\int_{-\infty}^{+\infty}\varphi_a(t)\prod_{i=1}^{N-1}\varphi_a(\theta_i t)e^{-jxt}\mathrm{d}t\mathrm{d}x \end{aligned}$$

where

$$\begin{aligned} \varphi_n(t) &= \int_{-\infty}^{+\infty} p_n(x)e^{jxt}\mathrm{d}x, \\ \varphi_a(t) &= \int_{-\infty}^{+\infty} p_a(x)e^{jxt}\mathrm{d}x. \end{aligned}$$

To optimize objective function (2.4), many optimization algorithms, conventional and heuristic, are available. Since the optimization problem is not convex or even unimodal, a heuristic method is more suitable.

A swarm algorithm called the differential evolution (DE) algorithm is applied to search in the filter parameter space to find one set of optimal θ_is. The

algorithm is a popular heuristic optimization algorithm which has received intensive attention in the last two decades. The algorithm is able to handle nonlinear and multimodal objective functions. Details about the DE algorithm can be found in [102] and the references therein. As to the optimization of the threshold, we use a line search. The reason is twofold. On one hand, the value of the objective function changes quite slow when the threshold is far from the central regions of the two distributions. A region where the variation of the objective function is very small is difficult for optimization. On the other hand, the line search on threshold can reduce the computational burden. The procedure is shown below.

Algorithm 1 Optimal linear filter search.

1: Obtain the expressions of characteristic functions of PDFs for normal and abnormal cases.
2: Initialize $(N-1) \times p$ (usually p is in the range of [5,10]) agents of $N-1$ dimension vectors (filter parameters) in the [-1, 1] hypercube.
3: Line search the optimal threshold for each agent and calculate objective function (2.4).
4: **while** A prescribed bound of iteration numbers doesn't reach **do**
5: Compute the agents' potentially new positions by DE mutation and crossover.
6: Line search the optimal threshold for each potentially new positions and calculate objective function (2.4).
7: If the objective function for a potentially new position is lower than its original position, update the agent's position.
8: **end while**
9: **outputs:** filter coefficients θ_is and threshold y_t.

To obtain the PDFs of normal and abnormal data, the segmentation method provided in [115], which is based on a non-parametric statistic, is firstly used to separate the signal into normal and abnormal parts. Then the PDFs can be calculated based on the data. If there is prior knowledge on the distributions, parametric estimation can be done. Otherwise, the Gaussian kernel technique is applied, since the characteristic functions of PDFs in the Gaussian kernel form are very easy to calculate. If the PDFs computed from different abnormal segments of one process variable are inconsistent, the optimization procedure will be performed based on the worst case, namely, the value of the objective function equals the largest among those using all abnormal PDFs, to obtain a conservative optimal filter. Since the method is a data-driven one, if there is no sufficient abnormal data, the method can only

guarantee the performance when an abnormality exists in the historical data set.

The determination on filter length N is dependent on the swiftness requirement of the system, such as operators' reaction time, equipment safety demands, and so on. After the PDFs and filter length are determined, we can run the procedure to obtain the optimal filter coefficients.

At the end of this section, we provide two examples to illustrate our method.

Example 2.2. The normal and abnormal data respectively follow logistic distributions:

$$p_n(x) = \frac{e^{-x}}{(1+e^{-x})^2};$$
$$p_{ab}(x) = \frac{e^{-(x-2)/2}}{2(1+e^{-(x-2)/2})^2}.$$

The general optimal filter (LLR filter) is

$$y[k] = \sum_{i=k-N+1}^{k} \left(0.5x[i] + 2\ln\frac{1+e^{-x[i]}}{1+e^{-(x[i]-2)/2}} + \ln 2 + 1 \right),$$

whose structure is complicated. Then we set the filter length N to be 5 and apply the procedure mentioned above to search for an optimal FIR filter. The number of agents is 28. We find that optimal filters with all different c_1 to c_2 ratios are very close to the moving average filter, but with different thresholds. This illustrates the good performance of moving average filters. Performance curves of raw data, the optimal FIR filter, and the general optimal filter are shown in Fig. 2.2. The performance of the optimal linear filter we obtained is better than using raw data, but not as good as the general optimal filter. This is expected since an optimal linear filter is just suboptimal in the general sense.

Example 2.3. Here we continue to use the PDFs in Example 2.1. The filter length N is 5. The number of agents is 28, and we initialize the first agent to the moving average filter to guarantee that the performance of obtained optimal filters is no worse than the moving average filter. The optimal performance curve we obtained is shown in Fig. 2.3. We can find that the performance using raw data is better than the moving average filter in some cases, but worse in other cases. The optimal linear filter performs better than both of them according to the optimal performance curve. The performance

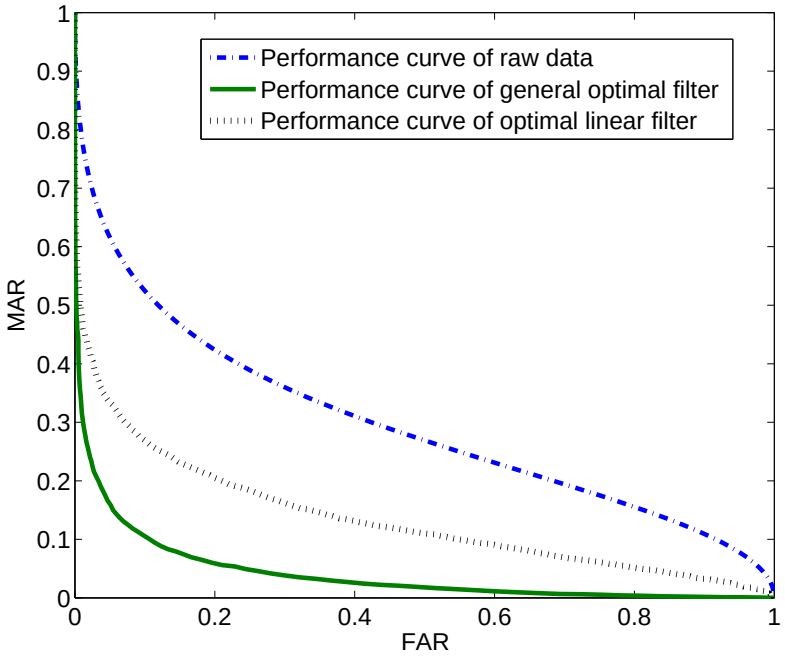

Figure 2.2: Performance curves of raw data and optimally filtered data for Example 2.2.

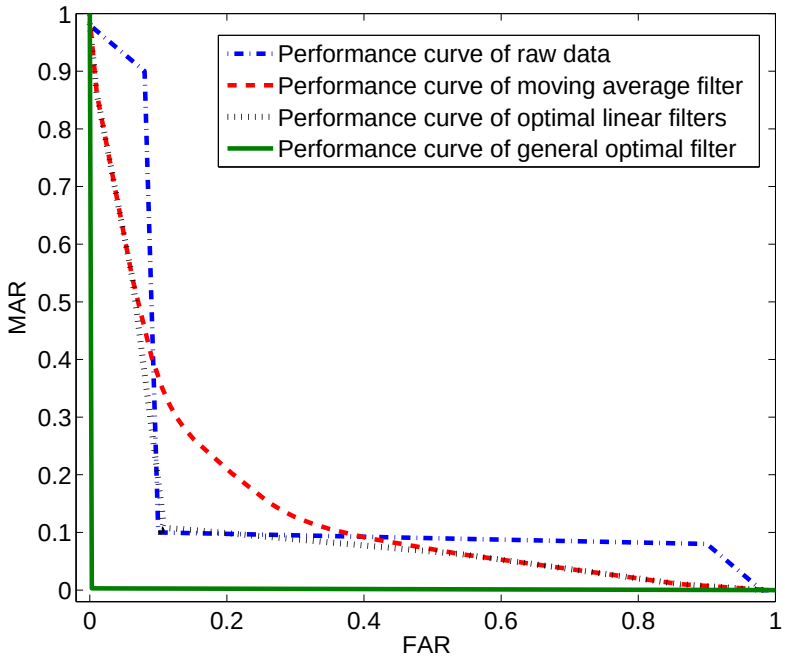

Figure 2.3: Performance curves of raw data, moving average filtered data, and optimal filtered data for Example 2.3.

curve via the general optimal filter can almost reach the ideal point, namely, the origin, while the optimal linear filter cannot get such a good performance. Moreover, when the ratio of c_1 to c_2 is far from 1, the optimal linear filter is inclined to the moving average filter. For example, when the ratio is 19, the four filter parameters are in the range of 0.98 to 1. But in the case that the ratio is close to 1, optimal linear filter are inclined to using the raw data only. When the ratio is 1, the four optimal linear filter parameters we obtain are all no larger than 0.01.

2.4 Moving Average Filter v.s. Optimal Linear Filter

It is shown in Section 2.3 that the moving average filter is not necessarily the optimal linear filter. However the moving average filter is popular in industry. As a result, it is important to find out the relationship between the moving average filter and the optimal linear filter, particularly to find out some sufficient conditions under which the moving average filter is optimal among all FIR filters. To discuss this problem, a constraint is added: the threshold is between the mean values of normal and abnormal distributions.

Firstly, an lemma is introduced:

Lemma 2.1. *The moving average filter is optimal among all FIR filters with fixed length N, if $p_n(x)$ and $p_a(x)$ are symmetric and unimodal, and for any parameter set $\{\theta_0, \theta_1, \cdots, \theta_{N-1}\}$, the cumulative distribution functions (CDFs) of the normal and abnormal distributions satisfy:*

$$\begin{array}{ll} F_n^N(x|\theta_0,\theta_1,\cdots,\theta_{N-1}) \le F_n^N\left(x|\frac{1}{N},\frac{1}{N},\cdots,\frac{1}{N}\right) & x > \mu_n \\ F_n^N(x|\theta_0,\theta_1,\cdots,\theta_{N-1}) \ge F_n^N\left(x|\frac{1}{N},\frac{1}{N},\cdots,\frac{1}{N}\right) & x \le \mu_n \\ F_a^N(x|\theta_0,\theta_1,\cdots,\theta_{N-1}) \le F_a^N\left(x|\frac{1}{N},\frac{1}{N},\cdots,\frac{1}{N}\right) & x > \mu_a \\ F_a^N(x|\theta_0,\theta_1,\cdots,\theta_{N-1}) \ge F_a^N\left(x|\frac{1}{N},\frac{1}{N},\cdots,\frac{1}{N}\right) & x \le \mu_a \end{array} \tag{2.5}$$

The definition of unimodal is given in [7]: a PDF $f(x)$ is said to be unimodal if there exists a mode $m \in \Omega$ such that $f(x) \le f(y)$ for all $x \le y \le m$ or for all $m \ge y \ge x$.

This lemma is intuitively obvious, since a better alarm filter should concentrate probability to the mean. The mathematical proof is as follows:

In the case that $\mu_n < \mu_a$, the objective function is:

$$\begin{aligned} J(\theta) = \; & c_1(1 - F_n^N(\widehat{y}_t|\theta_0,\theta_1,\cdots,\theta_{N-1})) \\ & + c_2 F_a^N(\widehat{y}_t|\theta_0,\theta_1,\cdots,\theta_{N-1}). \end{aligned}$$

$\widehat{y}_t$ is the optimal threshold for the parameters $(\theta_0, \theta_1, \cdots, \theta_{N-1})$. Noticing that $\widehat{y}_t$ is between μ_n and μ_a, we have

$$\begin{aligned} J(\theta) \geq \; & c_1(1 - F_n^N\left(\widehat{y}_t|\tfrac{1}{N}, \cdots, \tfrac{1}{N}\right)) \\ & + c_2 F_a^N\left(\widehat{y}_t|\tfrac{1}{N}, \cdots, \tfrac{1}{N}\right) \geq J\left(\{\tfrac{1}{N}, \cdots, \tfrac{1}{N}\}\right). \end{aligned}$$

In the case that $\mu_n > \mu_a$, it can be proved in the same way.

Thus, the moving average filter is optimal among all FIR filters with fixed length N. □

Then based on Lemma 2.1, we can get a sufficient condition under which the moving average filter is optimal. In the condition, the concept Log-concave is used. Log-likelihood concavity is a very important concept in non-parametric statistics. Log-concave probability distributions constitute a broad class, including uniform distribution, normal distribution, logistic distribution, Gamma distribution with the shape parameter no less than 1, and so on. Details about log-likelihood concavity can be found in [7] and the references therein. Theorem 2.1 is as follows:

Theorem 2.1. *The moving average filter is optimal among all FIR filters in the sense of minimum weighted sum of FAR and MAR, if the PDFs of normal and abnormal cases $p_n(x)$ and $p_a(x)$ are symmetric and log-concave.*

Proof: Consider an FIR filter with fixed length N:

$$y[k] = \sum_{i=0}^{N-1} \theta_i x[k-i].$$

Without loss of generality, the sum of the filter parameters θ_i's is fixed to be 1. Thus the mean value is unchanged after filtering.

$F_n(x)$ and $F_a(x)$ denote the CDFs of the raw data in the normal and abnormal cases respectively. $F_n^N(x|\theta_0, \theta_1, \cdots, \theta_{N-1})$ and $F_a^N(x|\theta_0, \theta_1, \cdots, \theta_{N-1})$ denote the CDFs of filtered data via an length N FIR filter with parameters $\{\theta_0, \theta_1, \cdots, \theta_{N-1}\}$. The CDFs for filtered data via the moving average filter are $F_n^N\left(x|\tfrac{1}{N}, \tfrac{1}{N}, \cdots, \tfrac{1}{N}\right)$ and $F_a^N\left(x|\tfrac{1}{N}, \tfrac{1}{N}, \cdots, \tfrac{1}{N}\right)$. μ_n and μ_a denote the mean values of normal and abnormal data. Since we fixed the steady gain at 1, the raw data and filtered data share the same mean value.

Since a log-likelihood concave distribution is always unimodal [56], to prove Theorem 2.1, we only need to prove that inequalities (2.5) always hold if $p_n(x)$ and $p_a(x)$ are symmetric log-concave. It can be proved by mathematical induction.

Basis: In the case that $N = 2$, (2.5) is satisfied.

We rewrite the filter as follows:

$$f_2(z) = \frac{1}{1+a} + \frac{a}{1+a} z^{-1}.$$

So that

$$F_n^2\left(x|\frac{1}{1+a}, \frac{\alpha}{1+a}\right) = \int_{-\infty}^{+\infty} F_n\left(\frac{t}{a}\right) p_n((1+a)x - t)dt,$$

and

$$\begin{array}{rcl} \frac{\partial F_n^2}{\partial a} & = & \int_{-\infty}^{+\infty} -\frac{t}{a^2} p_n\left(\frac{t}{a}\right) p_n((1+a)x-t)dt \\ & & + \int_{-\infty}^{+\infty} x F_n\left(\frac{t}{a}\right) p_n'((1+a)x-t)dt \\ & = & \int_{-\infty}^{+\infty} -\frac{t}{a^2} p_n\left(\frac{t}{a}\right) p_n((1+a)x-t)dt \\ & & + \int_{-\infty}^{+\infty} \frac{x}{a} p_n\left(\frac{t}{a}\right) p_n((1+a)x-t)dt \\ & = & \int_{-\infty}^{+\infty} t p_n(x-t) p_n(x+at)dt \\ & = & \int_0^{+\infty} t(p_n(2\mu_n + t - x) p_n(x+at) \\ & & - p_n(x+t) p_n(2\mu_n + at - x))dt \\ & = & \int_0^{+\infty} t \det(A(x,t))dt, \end{array}$$

where

$$A(x,t) = \left[\begin{array}{ll} p_n(2\mu_n - x + t) & p_n((2\mu_n - x + t) - (1-a)t) \\ p_n(x+t) & p_n((x+t) - (1-a)t) \end{array} \right].$$

Consider the case that $x > \mu_n$. Obviously, $2\mu_n - x + t < x + t$, and $\det(A) = 0$ if $a = 1$. For any $a < 1$ and $t \geq 0$, $(1-a)t \geq 0$. According to [7], as long as $p_n(x)$ is log-concave, $\det(A) \geq 0$. So the partial derivative of F_n^2 w.r.t. a is no less than 0 when $a < 1$. For the same reason, the partial derivative of F_n^2 w.r.t. a is no larger than 0 when $a > 1$. As a result,

$$F_n^2(x|\theta_0, \theta_1) \leq F_n^2(x|0.5, 0.5) \qquad x > \mu_n.$$

The following three inequalities in (2.5) can be similarly proved.

Inductive step: If in the case of $N = k$, (2.5) holds, then also in the case that $N = k + 1$, (2.5) holds.

The filter with length $k+1$ can be written in the form of the combination of raw data and filtered data via length k filter:

$$f_{k+1}(z) = \theta_j z^{-j} + (1 - \theta_j) \sum_{i=0, i\neq j}^{k} \theta_i z^{-i}.$$

So

$$\begin{array}{l} F_n^{k+1}(x|\theta_0, \cdots, \theta_k) \\ = \int_{-\infty}^{+\infty} \frac{p_n(\frac{t}{\theta_j})}{\theta_j} F_n^k\left(\frac{x-t}{1-\theta_j} | \frac{\theta_0}{1-\theta_j}, \cdots, \frac{\theta_{j-1}}{1-\theta_j}, \frac{\theta_{j+1}}{1-\theta_j}, \cdots, \frac{\theta_k}{1-\theta_j}\right) dt. \end{array}$$

For convenience, let

$$\begin{array}{l} F_n^k\left(x|\frac{1}{k},\frac{1}{k},\cdots,\frac{1}{k}\right)=\widehat{F}_n^k(x),\\ F_n^{k+1}(x|\frac{1-\theta_j}{k},\cdots,\frac{1-\theta_j}{k},\theta_j,\frac{1-\theta_j}{k},\cdots,\frac{1-\theta_j}{k})=\widetilde{F}_n^{k+1}(x|\theta_j),\\ F_n^k(x|\theta_0,\theta_1,\cdots,\theta_{k-1})=F_n^k(x),\\ F_n^{k+1}(x|\theta_0,\theta_1,\cdots,\theta_k)=F_n^{k+1}(x). \end{array}$$

Thus

$$\begin{array}{l} \widetilde{F}_n^{k+1}(x|\theta_j)-F_n^{k+1}(x)\\ =\int_{-\infty}^{+\infty}\frac{1}{\theta_j}p\left(\frac{1}{\theta_j}\right)\left(\widehat{F}_n^k\left(\frac{x-t}{1-\theta_j}\right)-F_n^k\left(\frac{x-t}{1-\theta_j}\right)\right)dt\\ =\int_{(1-\theta_j)\mu_n}^{+\infty}\frac{1}{\theta_j}\left(p\left(\frac{x-t}{\theta_j}\right)-p\left(\frac{x-2(1-\theta_j)\mu_n+t}{\theta_j}\right)\right)\\ \cdot\left(\widehat{F}_n^k\left(\frac{t}{1-\theta_j}\right)-F_n^k\left(\frac{t}{1-\theta_j}\right)\right)dt. \end{array}$$

According to the assumption in the case of $N=k$,

$$\left(\widehat{F}_n^k\left(\frac{t}{1-\theta_j}\right)-F_n^k\left(\frac{t}{1-\theta_j}\right)\right)\geq 0$$

for any $t\geq(1-\theta_j)\mu_n$. Moreover, a symmetric unimodal distribution satisfies

$$\begin{array}{ll} p\left(\frac{x-t}{\theta_j}\right)-p\left(\frac{x-2(1-\theta_j)\mu_n+t}{\theta_j}\right)\geq 0 & \forall t\geq(1-\theta_j)\mu_n, x\geq\mu_n;\\ p\left(\frac{x-t}{\theta_j}\right)-p\left(\frac{x-2(1-\theta_j)\mu_n+t}{\theta_j}\right)\leq 0 & \forall t\geq(1-\theta_j)\mu_n, x<\mu_n. \end{array}$$

Thus

$$\begin{array}{ll} \widetilde{F}_n^{k+1}(x|\theta_j)\geq F_n^{k+1}(x) & x\geq\mu_n;\\ \widetilde{F}_n^{k+1}(x|\theta_j)\leq F_n^{k+1}(x) & x<\mu_n. \end{array}$$

As a result, for $x\geq\mu_n$,

$$\begin{array}{c} F_n^{k+1}(x)\leq\widetilde{F}_n^{k+1}(x|\alpha_0)\\ \widetilde{F}_n^{k+1}(x|\alpha_0)\leq\widetilde{F}_n^{k+1}(x|\alpha_1)\\ \widetilde{F}_n^{k+1}(x|\alpha_1)\leq\widetilde{F}_n^{k+1}(x|\alpha_2)\\ \vdots \end{array}$$

where the sequence $\{\alpha_0,\alpha_1,\alpha_2,\cdots\}$ is $\left\{\theta_j,\frac{1-\alpha_0}{k},\frac{1-\alpha_1}{k},\cdots\right\}$.

Obviously, the sequence $\left\{\alpha_i-\frac{1}{k+1}\right\}$ is a geometric sequence with common ratio $-\frac{1}{k}$. So the sequence $\{\alpha_i\}$ will converge to $\frac{1}{k+1}$, and then

$$\lim_{j\to\infty}\widetilde{F}_n^{k+1}(x|\alpha_j)=\widehat{F}_n^{k+1}(x).$$

So $\widehat{F}_n^{k+1}(x)\geq F_n^{k+1}(x)$ for any $x\geq\mu_n$. For the same reason, $\widehat{F}_n^{k+1}(x)\leq F_n^{k+1}(x)$ for any $x\leq\mu_n$. The proofs of the third and forth inequalities in (2.5) are obtained in the same way.

To sum up, if the moving average filter with length k is the optimal FIR filter with length k, then the moving average filter with length $k+1$ is optimal among all FIR filters with length $k+1$.

Since both the basis and the inductive step have been proved, it has now been proved by mathematical induction that Theorem 1 holds for all filter lengths. □

Remark 1. In the proof we add a constraint: the threshold is between the mean values of normal and abnormal distributions. The constraint is reasonable because a linear filter is mainly applied in the case of mean changes, and it is not acceptable if the high (low) threshold is higher (lower) than the mean value of abnormal case or lower (higher) than the mean value of normal case. Moreover, the segmentation method provided in [115] also confirms this, since only the segments with the mean larger than the threshold are treated as abnormal data.

Remark 2. It is well known empirically that the moving average filter is not good at removing outliers. Many outliers manifest themselves as a heavy tail on the PDF. According to [7], an important property of a log-likelihood concavity distribution is that it at most has an exponential tail. As a result, the sufficient condition in our theorem seems to be consistent with the experience.

Remark 3. In the proof, the log-likelihood concavity condition is only used in the basis step of the mathematical induction, and in the rest of the proof only symmetry and unimodality are required. As a result, for normal and abnormal distributions that are symmetric unimodal but not log-likelihood concave, as long as we can prove that in the case of length 2 the moving average filter is optimal, we can say that the moving average filter is optimal for all filter lengths.

Remark 4. The threshold and the filter coefficients of the optimal linear filter may be coupled to each other. However, according to the proof of Theorem 1, as long as the sufficient condition is satisfied, for any fixed threshold, the moving average filter is always optimal. This property is important, since changing filter coefficients is often preferable than tuning the threshold in practice.

A consequent issue is how to test the sufficient condition based on data set. There are some well developed tests for unimodality and symmetry, such as Wilcoxon signed rank test and Kolmogorov-Smirnow test for symmetry

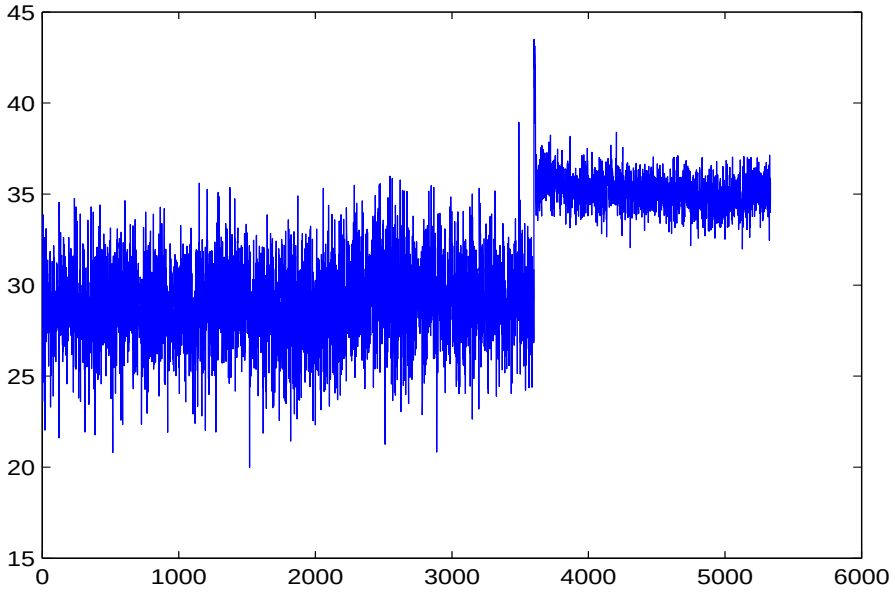

Figure 2.4: Flow rate signal.

[44, 95] and dip test for unimodality [49]. However the test for PDF log concavity is very hard. As discussed in [7], since a log-concave PDF implies a log-concave CDF and an increasing hazard rate (IHR), the CDF log concavity test and IHR test are often used for PDF log concavity analysis. According to [35], up to now there is no method to determine whether or not the PDF is log-concave from data.

2.5 Case Study

To illustrate the procedure, the proposed method is applied to a flow rate process variable from a petrochemical process.

Firstly, we adopt the segmentation method provided in [115] to separate the signal into normal and abnormal parts. In our case, the signal is separated to two segments: a normal part followed by an abnormal part. Since the effects of the transient state can be neglected in the FAR and MAR analysis, we further pick out the stationary part of each segment to meet the i.i.d. condition. Fig. 2.4 and 2.5 show the original signal and the kernel estimated PDFs of the normal and abnormal parts.

The second step is to test whether the distributions of both segments are

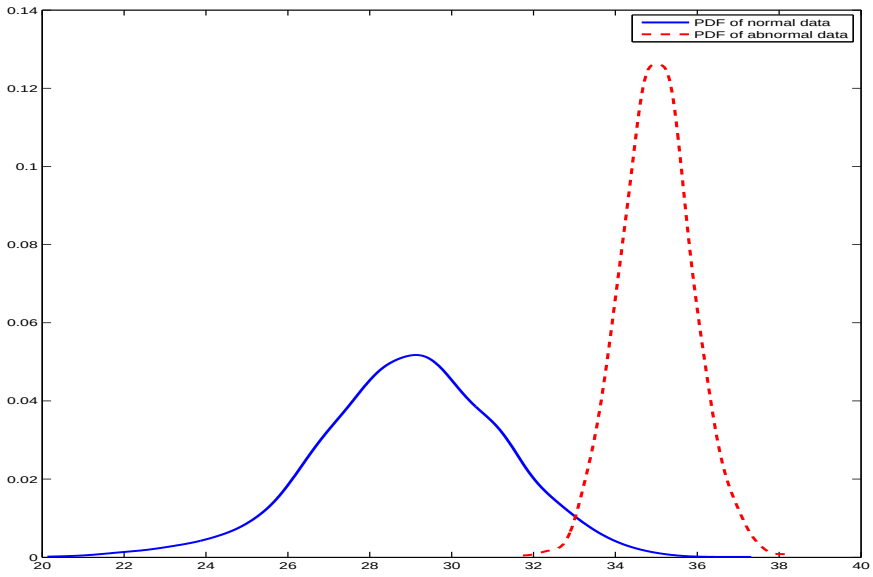

Figure 2.5: Kernel estimated PDFs of the normal and abnormal parts.

Table 2.1: Gaussianity tests results.

	JB Test		Lilliefors Test	
	Normal	Abnormal	Normal	Abnormal
Gaussian	False	True	False	True
p-value	0	0.1832	0	0.3555

Gaussian, since there is extensive discussion on alarm filter design methods in the Gaussian case. The Jarque-Bera test and Lilliefors test are conducted to make the decision. Unfortunately the data cannot pass both the tests. Table 2.1 shows the results using Matlab functions jbtest and lillietest. We can find that the normal part can never pass the test; thus it is non-Gaussian. As to the abnormal part, although the hypothesis of Gaussian distribution cannot be rejected at the 5% significance, the p-values are not high for both tests.

Since the Gaussianity test cannot be passed, we need to move to the third step: test whether the condition provided in Theorem 1 is satisfied. If this is the case, moving average filters are the optimal FIR filters. Otherwise, we have to use the procedure provided in Section 2.3 to design an optimal FIR filter.

To test the symmetry of the data, we apply the Kolmogorov-Smirnow test. The Kolmogorov-Smirnow test is designed to compare the distributions

of two data sets. Both the abnormal and normal data can pass the test, and the optimal medians and p-values are shown in Table 2.2. The IHR test is also carried out. The test that was proposed in [89] is based on the normalized spacings. In the ideal case, the normalized spacings should be a decreasing series for IHR distributions. Thus the probability that the previous terms in the normalized spacing series is greater than the latter ones is used to determine whether the distribution has an increasing hazard rate. Although IHR cannot guarantee that the PDF is log-concave, it is a necessary condition. Both parts pass the IHR test. The logarithm functions of the kernel estimated PDFs are shown in Fig. 2.6. The IHR test result, as well as the log PDF curves in Fig. 2.6, is in favor of PDF log concavity. So we approximately reach the conclusion.

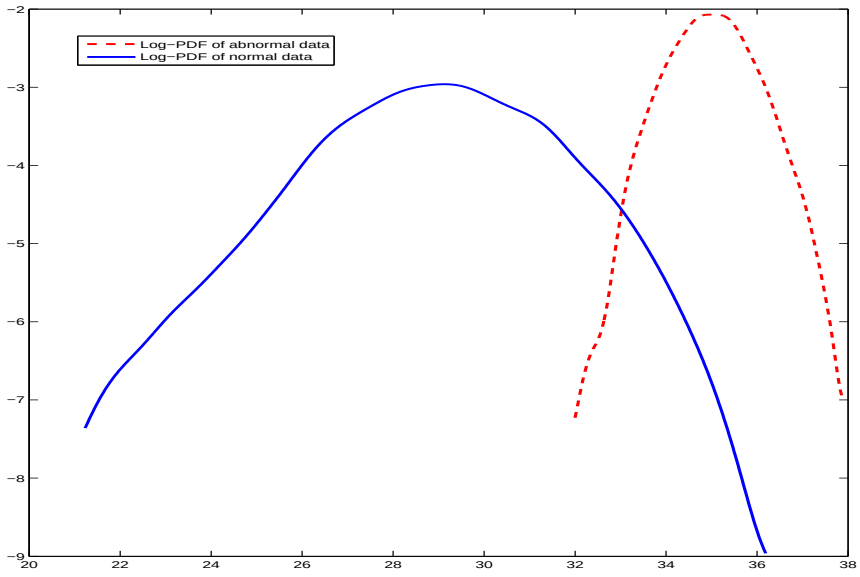

Figure 2.6: Logarithm of kernel estimated PDFs of the normal and abnormal parts.

Since the conditions in Theorem 1 are satisfied, a moving average filter is an optimal FIR filter. We fix the length of the filter at 3. The FAR and MAR of the raw signal, the moving averaged signal, and the filtered signals via two other randomly selected FIR filters are shown in Fig. 2.7. For any threshold between the means of normal and abnormal data, both the FAR and MAR of the moving averaged data is lower than the other three cases. When

Table 2.2: Symmetry test results.

	Normal	Abnormal
Symmetric	True	True
Optimal median	29.04	35.00
p-value	0.8617	1

the moving average filter is used, we have tried different thresholds and find that the false alarms are less than the case of using raw data or other linear filters. For example, when the threshold is set to be 33, the moving average filtered signal gives 5 alarms, with the first 4 false alarms; three of them last 2 samples and the other one lasts only 1 sample. The last alarm represents the real abnormality and it is standing for the whole abnormal period. However, when using the raw signal and threshold 33, 107 false alarms take place in the first 3000 samples that are under normal situation, and the alarm is raised 17 times during the abnormal period because the alarm is interrupted when the signal falls below the threshold.

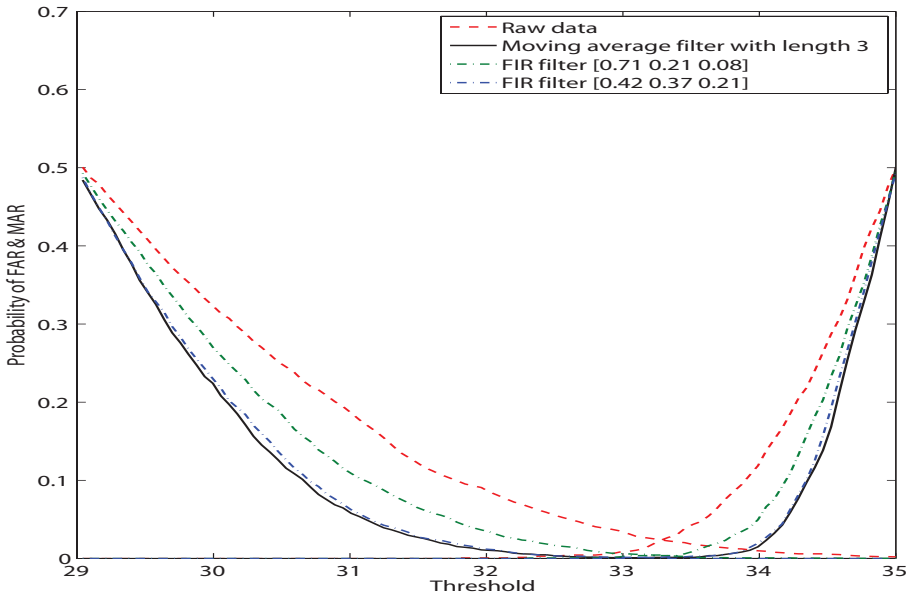

Figure 2.7: Performances of different alarm filters.

Chapter 3

Pattern Matching of Alarm Flood Sequences by a Modified Smith-Waterman Algorithm*

3.1 Overview

In this chapter, we address the alarm flood pattern matching problem based on alarm data. All alarms raised in an alarm flood form an alarm flood sequence, which is a typical event sequence with time information. A modified Smith-Waterman algorithm suitable for the alignment of alarm flood sequences for the purpose of evaluating their similarities is proposed. An industrial case study illustrates the validity of this proposed technique.

3.2 Problem Description

As mentioned in Section 1.2.2, among three main categories of unimportant alarms, consequence alarms are the most complicated ones. We focus on the problem of consequence alarms in this chapter, but our problem formulation is somewhat different from [33, 45, 46, 54]. We consider the alarms during one alarm flood as one alarm flood sequence, and compare the alarm flood sequences of different floods. Similar alarm flood sequences usually relate to the same kind of underlying abnormality; so by clustering similar alarm flood sequences we can find out flood pattern candidates. Then experts can analyze these pattern candidates to determine real flood patterns and their root causes which will be shown to the operators when a new flood with high similarity

*A version of this chapter has been published in [30].

to a certain flood pattern is forthcoming. Since the purpose is clustering similar alarm sequences instead of identifying exactly the same sequences, the deficiency of the method provided in [45, 46] can be avoided.

The goal is very similar to [63, 96, 97] that made use of process data for abnormality diagnosis. However, in this chapter, we use alarm data rather than process data to measure the similarity of different abnormalities.

Approximate sequence matching techniques can be applied to calculate similarities of alarm flood sequences. However, current approximate matching techniques rarely consider time stamp information of each element in the sequence, since they are mainly designed for text pattern and biological sequences matching. Our main contribution is modifying the formulation of the Smith-Waterman algorithm by adding time stamp information, and proposing a modified Smith-Waterman algorithm suitable for alarm flood pattern matching.

3.3 Preliminary on Sequence Data Mining

Brief Survey on Sequence Data Mining

Sequence data is studied in many application domains such as intrusion detection [52, 69], customer purchases pattern mining [23], text pattern matching/detection [6, 34], biological sequences matching/detection [2, 98], and so on. Since purposes and natures of input data in different applications are not the same, there are a variety of problems formulations, e.g., exact or approximate matching of sequences [2, 6, 34, 98], frequently repeated sequential pattern-mining [79, 99], abnormal sequences detection[52, 69], and so on. Here we would only offer some discussion on the research topics related to the application of alarm sequences analysis.

Frequently repeated sequential pattern-mining tries to discover short subsequences that are frequently appear in a large sequential database. An exhaustive survey was provided in [79]. In the survey, algorithms were classified into three categories: apriori-based, pattern-growth, and early-pruning algorithms. Because of the different techniques used, the algorithms have different memory and time consumptions. The pioneering work on consequence alarms suppression [33, 45, 46, 54] apply frequently repeated sequential pattern-mining techniques.

Anomaly detection is another extensively discussed topic. The objective

of anomaly detection is to identify abnormal sequences or subsequences from normal ones. The survey paper [24] provided a structured overview of the research and categorized the techniques to three distinct frameworks: sequence-based anomaly detection, subsequence-based anomaly detection, and pattern frequency-based anomaly detection. The problem of anomaly detection is different from ours, since in our situation, all the events appear in the sequences are alarm events that indicate abnormalities. However, the problem formulation of sequence-based anomaly detection is very similar to the one of this chapter. The only difference is that in anomaly detection, the training data set is a set of normal sequences, and they mainly focus on the test sequences that are not similar to any cluster of normal sequences. While in our problem, the training data set is a set of abnormal sequences, namely, alarm floods, and we mainly focus on the test sequences that may belong to one or several clusters (patterns) in the training set.

In the framework of similarity-based techniques for sequence-based anomaly detection, a key point is similarity measurement. Many similarity measurement techniques are based on approximate matching. The topic of approximate matching is extensively discussed in [83] and the reference therein. As defined in [83], "approximate matching is modeled using a distance function that tells how similar two strings are". A first such distance function, edit distance, was proposed in [73], which is the minimum number of insertions, deletions, and substitutions to make two sequences equal. To increase flexibility of setting different penalty weights on insertion, deletion and substitutions, variations of the edit distance are proposed in [98] and [84], together with corresponding dynamic programming algorithms. From then on, such fast algorithms as BLAST and FASTA [4, 75] were designed to decrease the time consumption, but the objective function, namely, the distance function, was not changed. A similar idea applied to real valued time series is dynamic time warping [75]. It is proposed to measure the distance between two real valued time series. Although some researchers also tried to apply it to symbolic sequence matching [2, 92], they had to firstly convert a symbolic sequence to a meaningful real valued time series. Compared with such statistical model-based techniques as Markov model and hidden Markov model [47, 116], the approximate matching technique is more suitable for sequences that are not very long, since it does not require large amount of data to model statistical properties of the sequences.

Brief Introduction of the Smith-Waterman Algorithm

The Smith-Waterman algorithm was first proposed in [98]. Its objective is "to find a pair of segments, one from each of two long sequences, such that there is no other pair of segments with greater similarity (homology)" [98].

The Smith-Waterman algorithm is a local sequence alignment method. Before discussing the algorithm, we first introduce the concept of local alignment. Given a pair of symbolic segments, one from each of two symbolic sequences, we can equalize the length of the two segments by inserting gaps (symbol '-') in one or both of them (if the two segments have the same length, we can also choose to insert no gap). Then each symbol in one segment has a corresponding symbol in the other segment at the same position. This is called alignment. Since the two symbolic segments are two contiguous subsequences of the two symbolic sequences, respectively, the alignment on the pair of segments is called local alignment of the two sequences.

For example, consider two symbolic segments:

$$X = [4\ 5\ 4\ 6\ 7]$$
$$Y = [6\ 6\ 7\ 4].$$

An alignment of this pair of segments is:

$$\begin{array}{rcccccccc} X' & = & [& 4 & 5 & 4 & 6 & 7 & - &] \\ Y' & = & [& 6 & - & - & 6 & 7 & 4 &]. \end{array}$$

The two aligned segments have the same length 6. Each symbol in aligned segment X' has a corresponding symbol in the aligned segment Y', and the two symbols compose a symbolic pair. For instance, the first symbolic pair of the aligned segments X' and Y' is $(4, 6)$ and the second pair is $(5, -)$.

Obviously, for one pair of segments there are a huge amount of different alignments since gaps can be added arbitrarily. Consider the example mentioned above, we can also align the pair of symbolic segments as:

$$\begin{array}{rcccccccccc} X'_1 & = & [& 4 & 5 & 4 & 6 & 7 & - & - &] \\ Y'_1 & = & [& 6 & - & 6 & - & - & 7 & 4 &], \end{array}$$

or:

$$\begin{array}{rcccccccc} X'_2 & = & [& 4 & 5 & 4 & 6 & 7 &] \\ Y'_2 & = & [& 6 & - & 6 & 7 & 4 &]. \end{array}$$

As a result, a scoring system is required to differentiate good alignments from bad ones, and the similarity of the two symbolic segments should depend on the optimal alignment.

For a symbolic pair (a, b) in a pair of aligned symbolic segments, where a and b are two symbols, e.g., two discrete events, two letters, two nucleic acid bases, if neither a nor b is the gap symbol '-', a similarity score function $s(a, b) : \Sigma \times \Sigma \rightarrow \mathbb{R}$, where Σ is the alphabet of the concerned sequences, is provided. The value of the similarity score function $s(a, b)$ is positive for a match ($a = b$), and is non-positive for a mismatch ($a \neq b$). Usually, a uniform score for all matched pairs (kept at 1) and a uniform negative score μ for all mismatched pairs are chosen if there is no additional prior information on the symbols.

For a symbolic pair (a, b) including a gap symbol '-', the similarity score is always negative as a penalty of inserting a gap. The value is related to l, the number of contiguous gap symbols prior to the current gap. In [88], if l is 0, it is called opening a gap, which suffers a heavier penalty. In the case that $l > 0$, it is called extending a gap, and a lighter penalty is used. In the case that opening a gap is not of greater significance, a uniform penalty value δ for all l is preferred, since it can simplify the algorithm. Therefore, the uniform penalty value δ is used in this chapter.

The similarity score of an alignment is the summation of all the similarity scores of its symbolic pairs. The similarity score calculation of the aligned segments X' and Y' is shown below.

Rewrite the alignment:

$$\begin{array}{rcl} X' & = & [\ 4\ \ 5\ \ 4\ \ 6\ \ 7\ \ -\ \ ] \\ Y' & = & [\ 6\ \ -\ \ -\ \ 6\ \ 7\ \ 4\ \ ]. \end{array}$$

By using the uniform gap penalty value $\delta = -0.4$ and similarity score function

$$s(a, b) = \begin{cases} 1, & \text{if } a = b \\ -0.6, & \text{if } a \neq b \end{cases}$$

we can calculate the similarity score of this alignment as:

$$\begin{array}{rl} & s(4, 6) + \delta + \delta + s(6, 6) + s(7, 7) + \delta \\ = & -0.6 + (-0.4) + (-0.4) + 1 + 1 + (-0.4) \\ = & 0.2. \end{array}$$

The higher the similarity score is, the better the alignment is. The alignment of a segment pair that have the highest similarity score is the optimal alignment of this pair, and the corresponding similarity score is defined as the similarity index of this segment pair.

Given concepts of alignment and the similarity index of a segment pair, it is easy to formulate the local alignment problem.

Consider two symbolic sequences with arbitrary lengths:

$$A = [a_1\ a_2\ \cdots\ a_M]$$
$$B = [b_1\ b_2\ \cdots\ b_N],$$

where $a_m, b_n \in \Sigma,\ m = 1, 2, \cdots, M,\ n = 1, 2, \cdots, N$. $A_{i:p}$ and $B_{j:q}$ denote segments, or called contiguous subsequences, of A and B, respectively:

$$A_{i:p} = [a_i\ a_{i+1}\ \cdots\ a_p];$$
$$B_{j:q} = [b_j\ b_{j+1}\ \cdots\ b_q].$$

$I(A_{i:p}, B_{j:q})$ denotes the similarity index of the segment pair $(A_{i:p}, B_{j:q})$. The goal of the local alignment problem is searching for the optimal segment pair whose similarity index is the highest among all segment pairs. This highest similarity index is then defined as the similarity index of the two sequences A and B, which is denoted by $S(A, B)$. The only exception is that the two sequences are totally different, and all the segment pairs have a negative similarity index. In this case, $S(A, B)$ is set to be 0, which means no similarity at all. In a nut shell, the optimal local alignment of sequences A and B is the optimal alignment of the optimal segment pair of two sequences, and the similarity index $S(A, B)$ of these two sequences can be expressed by the following equation:

$$S(A, B) = \max_{1 \le i \le p \le M, 1 \le j \le q \le N} (I(A_{i:p}, B_{j:q}), 0).$$

The Smith-Waterman algorithm provides a procedure to find out the similarity index $S(A, B)$ and the corresponding local alignment.

The Smith-Waterman algorithm generates an index matrix H whose element $H_{p+1,q+1}$ denotes the maximum positive similarity index of segment pairs ending in a_p and b_q, respectively. If there is no positive similarity index, $H_{p+1,q+1} = 0$. In other words,

$$H_{p+1,q+1} = \max_{1 \le i \le m, 1 \le j \le n} (I(A_{i:p}, B_{j:q}), 0).$$

When one or both of the segments are empty, the segments pair similarity index is 0. As a result, $H_{1,q} = 0$, and $H_{p,1} = 0$ for any p and q.

$H_{p+1,q+1}$ can be recursively calculated by the following equation with a uniform gap penalty δ:

$$H_{p+1,q+1} = \max\left\{\ H_{p,q} + s(a_p, b_q),\quad H_{p,q+1} + \delta,\quad H_{p+1,q} + \delta,\quad 0\ \right\}, \qquad (3.1)$$

Equation (3.1) is somewhat different from the one provided in [88], since a uniform gap penalty is used here. In the review article [66], the authors described the algorithm by the same equation as equation (3.1) except the sign of δ, since in that paper δ was set a positive value and a minus sign was then used before the penalty term.

Based on equation (3.1), a dynamic programming algorithm can be easily developed to solve this optimization problem. The algorithm is described by the following steps:

1. Build a matrix $H \in R^{(M+1)\times(N+1)}$, and initialize the first row and column of H to be 0.

2. Calculate the other entries in matrix H from upper left to lower right by equation (3.1).

3. Find the highest value in the matrix H. This value is the similarity index of the two sequences ($S(A, B)$).

4. Go backward from this highest value until meet an entry with value 0. The path shows the optimal local alignment (detailed in the following example).

As an example consider two sequences $A = [1\ 2\ 1\ 4\ 1]$ and $B = [1\ 3\ 2\ 1\ 1\ 2\ 3]$. The penalty of a gap is $\delta = -0.4$; the match score is 1; and the penalty for mismatching is $\mu = -0.6$. The matrix H and the optimal local alignment are given by Fig. 3.1. The calculation of H is straightforward. For instance, the calculation of $H_{6,6}$ is based on the values of $H_{5,5}$, $H_{5,6}$ and $H_{6,5}$. Since the fifth symbols in both sequences are '1', we should calculate the values of $H_{5,5} + 1$, $H_{5,6} + \delta$ and $H_{6,5} + \delta$. They are 3.2, 1.6, and 1.4, respectively. So the value of $H_{6,6}$ should be the largest positive value among them, namely, 3.2.

The backward path search is based on the H matrix calculation. Since $H_{6,6}$ has the largest value in the matrix, the path search should begin at this entry. Since the value of $H_{6,6}$ is $H_{5,5} + 1$, we should go back from $H_{6,6}$ to $H_{5,5}$ instead of $H_{5,6}$ or $H_{6,5}$. Then we find that the value of $H_{5,5}$ is calculated from $H_{4,5}$, so we should go back to $H_{4,5}$. Continuously do this until we meet the first 0 entry at $H_{1,1}$. Now the whole path is obtained. The entries of the path are underscored in Fig. 3.1.

Then we go forward the path to complete the optimal local alignment. For each diagonal move, add a corresponding symbol in both aligned segments.

For instant, the path start at $H_{1,1}$ and moves to $H_{2,2}$; so the first symbol in the aligned segments should be a_1 and b_1, namely, '1' and '1', respectively. For each horizontal move, add a corresponding symbol in the aligned segment of B, and add a gap in the aligned segment of A. For instant, the second move in the path is a horizontal one, so the second symbol in the aligned segment of B is b_2, namely, '1', while the second symbol in the aligned segment of A is a gap. For each vertical move, add a corresponding symbol in the aligned segment of A and a gap in the aligned segment of B. Following these rules, we finally reach the optimal local alignment: $[1 - 2\ 1\ 4\ 1]$ and $[1\ 3\ 2\ 1 - 1]$.

	Δ	1	3	2	1	1	2	3
Δ	0	0	0	0	0	0	0	0
1	0	1	0.6	0.2	1	1	0.6	0.2
2	0	0.6	0.4	1.6	1.2	0.8	2	1.6
1	0	1	0.6	1.2	2.6	2.2	1.8	1.4
4	0	0.6	0.4	0.8	2.2	2	1.6	1.2
1	0	1	0.6	0.4	1.8	3.2	2.8	2.4

(a) Similarity table H

$A_{1:5}$:	1		–		2		1		4		1		
Score:	1	+	–0.4	+	1	+	1	+	–0.4	+	1	=	3.2
$B_{1:5}$:	1		3		2		1		–		1		

(b) Optimal local alignment

Figure 3.1: Local alignment result.

3.4 Modified Smith-Waterman Algorithm for Alarm Flood Pattern Matching

The alarm flood pattern matching problem is to calculate a certain similarity index between alarm floods, and determine which floods should be classified in one group (pattern) based on the similarity index. The common segment of alarm floods in a pattern may be used as a symptom of this kind of alarm floods to determine whether a incoming flood belongs to that pattern.

If we use a unique symbol to denote each alarm type, finding the similarity index and common segment of alarm flood sequences is very similar to the goal of approximate matching. Particularly, a good sequence matching algorithm for alarm sequences should have the following properties:

1. Tolerant to some irrelevant alarms occurring in one or both of the alarm sequences.

2. Somewhat tolerant to ambiguity of order.

The reason why the first property is required is obvious. Standard approximate matching algorithms considering gap and mismatch can achieve this requirement very well. As for the second property, it is possible that several strongly connected alarms are raised almost simultaneously, but the order of them in alarm sequences varies from time to time. Moreover, because of random detection delay, one alarm may be rasied after its cascaded alarms. As a result, when the time stamps of two alarms are close, the order of these two alarms is not so important. In other words, we should make the order of these alarms vague.

In the approximate matching area, a similar topic is swap. A few papers focus on this topic such as [36, 76]. However they discuss the sequences without time stamps. In [33, 54], the authors considered time constraints. However the pattern-growth methods are for frequent sequence mining, but not for approximate pattern matching. Moreover, hard time constraints instead of soft ones are concerned in those methods. Thus those methods are not suitable for our problem. As a result, it is necessary to propose a modified approximate matching algorithm that is suitable for alarm flood pattern matching.

We assume that the alphabet of alarm types is

$$\Sigma = 1, 2, \cdots, K.$$

The size of the alphabet is K. This assumption is only for convenience (without loss of generality) since we can map any finite alphabet to this one. A time-stamped alarm sequence is defined as follows:

$$\begin{aligned} &A = a_1\ a_2\ \cdots\ a_M; \\ &a_m = (e_m, t_m), m = 1, 2, \cdots, M, \end{aligned}$$

where e_m is the alarm type, namely an integer from 1 to K, and t_m is the time stamp of the alarm a_m.

The essential idea of the modification on the Smith-Waterman algorithm for the time-stamped alarm sequence is to redefine the similarity score of a symbolic pair $s(a, b)$ to a similarity score of a time-stamped alarm pair. In order to explain this new similarity score clearly, two new concepts are introduced: 'time distance vector' and 'time weight vector'.

A time distance vector is defined for each alarm in an alarm sequence. The time distance vector for the m-th alarm a_m is as follows:

$$\begin{aligned} &\mathbf{d}_m = [d_m^1\ d_m^2\ \cdots\ d_m^K]^T, \\ &d_m^k = \begin{cases} \min\limits_{1\le i\le M} \{|t_m - t_i| : e_i = k\}, & \text{if the set is not empty;} \\ \infty, & \text{otherwise,} \end{cases} \\ &\qquad\qquad\qquad\qquad \text{for } k = 1, 2, \cdots, K. \end{aligned} \tag{3.2}$$

An entry d_m^k in the time distance vector $\mathbf{d}_m$ carries the information of the time gap between the m-th alarm and the nearest alarm on the time axis with alarm type k. If there is no type k alarm in the alarm sequence, namely $e_i \neq k$ for all $i \in [1, M]$, the time gap is ∞. Obviously, since the m-th alarm has the alarm type e_m, $d_m^{e_m}$ is zero.

Then we define a time weight vector for each alarm in an alarm sequence. The time weight vector for the m-th alarm is as follows:

$$\begin{aligned} \mathbf{w}_m &= [w_m^1\ w_m^2\ \cdots\ w_m^K]^T \\ &= [f(d_m^1)\ f(d_m^2)\ \cdots\ f(d_m^K)]^T, \end{aligned} \tag{3.3}$$

where $f(\centerdot) : \mathbb{R} \mapsto \mathbb{R}$ is a time weighting function with respect to the time distance d_m^k. A function that satisfies the following conditions can be used as $f(\centerdot)$:

1. Monotonically decreasing on the positive axis.

2. $f(0) = 1,\ f(\infty) = 0.$

As a result, the e_m-th entry of the time weight vector $\mathbf{w}_m$ is 1, since $d_m^{e_m}$ is zero. If one type of alarm, say the j-th type, is raised very close to the m-th alarm on the time axis, the j-th entry in $\mathbf{w}_m$ has a large value almost 1. On the other hand, if one type of alarm does not exist in a neighborhood of the m-th alarm on the time axis, it will get a low weight in vector $\mathbf{w}_m$, and the weight will go to zero if the time gap approaches ∞. After applying the time weight vector, one alarm event does not belong to a single alarm type, but is attributed to all alarm types with different weight. We take the simple alarm

sequence shown in Fig. 3.2 as an example. In this alarm sequence, we have four alarm events belong to alarm types 1, 3, 2, and 4, respectively. By using kernels instead of impulses to represent alarms on the time axis, we can vague the order of the second and third alarms. In this case, the second alarm event is considered as an alarm attributed to type 2 and 3 with high degrees but attributed to type 1 and 4 with low degrees.

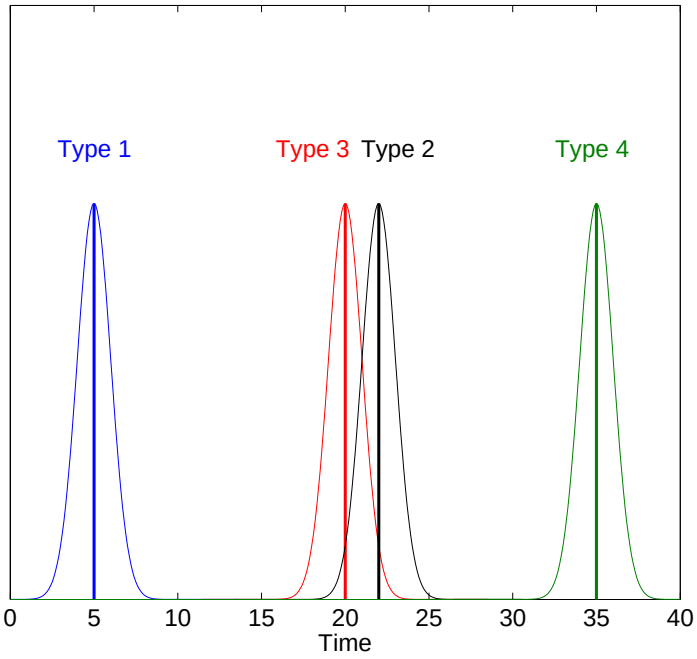

Figure 3.2: Alarm sequence represented by time weight vector.

In the following example and case study provided in this chapter, when we do the sequences matching on a pair of time-stamped alarm sequences, we choose the scaled Gaussian function

$$f(x) = e^{-\frac{x^2}{2\sigma^2}} \tag{3.4}$$

as the time weighting function for the first sequence, and for the $f(\centerdot)$ of the second sequence,

$$f(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{if } x \neq 0 \end{cases}. \tag{3.5}$$

The reasons why we choose two different time weighting functions for the two sequences are twofold. Firstly, the time difference is a relative but not absolute concept; so only blurring the order of one sequence actually can affect both

sequences. Secondly, if we use the scaled Gaussian function for both sequences, one matching pair may be counted more than once when several alarms are closely raised, which leads to a false high similarity index.

Now we redefine the similarity score $s(a, b)$ for a time-stamped alarm pair $((e_a, t_a), (e_b, t_b))$. In the classical Smith-Waterman algorithm, the similarity score function is a two-value function. The only possible values are the match score 1 and the mismatch penalty μ. In the modified algorithm, we define the similarity score function value as a linear combination of the match score and mismatch penalty, and the linear combination ratio is according to the time weight vectors $\mathbf{w}_a$ and $\mathbf{w}_b$ of the two alarms:

$$s((e_a, t_a), (e_b, t_b)) = \max_{1 \leq k \leq K} [w_a^k \times w_b^k](1 - \mu) + \mu, \tag{3.6}$$

Substituting the redefined similarity score for the original one in equation (3.1), the Smith-Waterman algorithm is modified to a time-stamped sequence approximate matching algorithm that is suitable for alarm flood pattern matching.

The modified algorithm has the following properties:

1. The largest similarity index of any two sequences with lengths M and N, respectively, is $\min(M, N)$.

2. If two sequences with the same length M have identical alarm sequences, the similarity index is M, regardless of time stamps.

3. If the scaled Gaussian function is used as the time weighting function, a larger σ leads to a larger or equal similarity index. If $\sigma = 0$, the modified algorithm reduces to standard Smith-Waterman algorithm.

4. The similarity calculation is not commutative, i.e., $S(A, B) \neq S(B, A)$.

According to the first property, we can normalize the similarity index to be between 0 and 1. The second property states that if the alarm sequences are the same, the normalized similarity index is 1. Property 3 gives some hints on how to choose the time weighting function. The fourth property means that the calculated similarity index of sequence A to sequence B may not equal to that of sequence B to sequence A. To fix this problem, we choose the greater one as the similarity index of the pair of sequences. The reason why we choose the greater one is that we want to discover as many similar alarm flood candidates as possible.

At the end of this section, we provide a simple example to illustrate the modified algorithm. Consider two time-stamped alarm sequences A and B:

$$A = (4,0),(1,3),(3,5),(1,7.5),(2,8),(3,13),(1,20);$$
$$B = (1,0),(3,1),(2,4),(1,4.2),(4,7),(3,9),(2,12.5).$$

The alphabet is $\Sigma = \{1,2,3,4\}$ with the size $K = 4$; and the lengths of both sequences are 7, i.e., $M = 7$. For the alarm sequence A we need to calculate 7 time distance vectors since there are 7 alarms, and each vector should consist of 4 entries since the size of the alphabet is 4. Since the alarm type of the first alarm is $e_1 = 4$, the fourth entry of d_1 is 0. The nearest alarm with alarm type 1 occurs at time instant 3, so $d_1^1 = 3 - 0 = 3$. Similarly, $d_1^2 = 8 - 0 = 0$ and $d_1^3 = 5 - 0 = 0$. We can complete all the 7 time distance vectors in the same way to reach the following result:

$$[\ \mathbf{d}_1 \quad \mathbf{d}_2 \quad \cdots \quad \mathbf{d}_7\ ] = \begin{bmatrix} 3 & 0 & 2 & 0 & 0.5 & 5.5 & 0 \\ 8 & 5 & 3 & 0.5 & 0 & 5 & 12 \\ 5 & 2 & 0 & 2.5 & 3 & 0 & 7 \\ 0 & 3 & 5 & 7.5 & 8 & 13 & 20 \end{bmatrix}.$$

Element-wisely calculate the time weighting function (3.4) of the distance vectors, the time weight vectors are obtained:

$$[\ \mathbf{w}_1 \quad \mathbf{w}_2 \quad \cdots \quad \mathbf{w}_7\ ]$$
$$= \begin{bmatrix} 0.325 & 1.000 & 0.607 & 1.000 & 0.969 & 0.023 & 1.000 \\ 0.000 & 0.044 & 0.325 & 0.969 & 1.000 & 0.044 & 0.000 \\ 0.044 & 0.607 & 1.000 & 0.458 & 0.325 & 1.000 & 0.002 \\ 1.000 & 0.325 & 0.044 & 0.001 & 0.000 & 0.000 & 0.000 \end{bmatrix}.$$

Setting the algorithm parameters $\delta = -0.4$, $\mu = -0.6$, the matrix H for similarity index of alarm sequences A to B and the optimal local alignment is shown in Fig. 3.3. The calculation is very similar to the classical Smith-Waterman algorithm that we introduced in Section 3.3. The only difference is the calculation of similarity score. For instance, when we calculate the value of $H_{5,4}$, we need to get the similarity score of the 4th alarm in sequence A and the 3rd alarm in sequence B. We know that the time weight vector of the 4th alarm in sequence A is $\mathbf{w}_4 = [1\ 0.969\ 0.458\ 0.001]^T$. For the second sequence B, the time weight vector of the 3rd alarm is $[0\ 1\ 0\ 0]^T$ using the time weighting function (3.5). Do the element-wise multiplication of these two vectors and pick up the largest element, which is 0.969, we can calculate the similarity score as: $0.969 \times (1 + 0.6) - 0.6 = 0.951$. Then the value of $H_{5,4}$ can

be calculated by the following equation:

$$H_{5,4} = \max\left\{ \begin{array}{cccc} H_{4,3} + 0.951, & H_{4,4} - 0.4, & H_{5,3} - 0.4, & 0 \end{array} \right\} = 2.951.$$

Since the time difference of the 4-th and 5-th alarms in sequence A is small, our algorithm can swap them to match the 3-rd and 4-th alarms in sequence B as shown in Fig. 3.3(b). While the algorithm will not do the same for the 5-th and 6-th alarms in sequence A to match the last two alarms in sequence B, because the time gap of these two alarms is large. Then we repeat this procedure to calculate the matrix H for the similarity index of B to A which is shown in Fig. 3.4.

	Δ	(1,0)	(3,1)	(2,4)	(1,4.2)	(4,7)	(3,9)	(2,12.5)
Δ	0	0	0	0	0	0	0	0
(4,0)	0	0	0	0	0	1	0.6	0.2
(1,3)	0	1	0.6	0.2	1	0.6	1.370	0.970
(3,5)	0	0.6	2	1.6	1.2	0.8	1.6	1.290
(1,7.5)	0	1	1.6	2.951	2.6	2.2	1.8	2.551
(2,8)	0	0.951	1.2	2.6	3.902	3.502	3.102	2.8
(3,13)	0	0.551	1.951	2.2	3.502	3.302	4.502	4.102
(1,20)	0	1	1.551	1.8	3.2	2.902	4.102	3.902

(a) Similarity table H

$A_{2:6}$:	(1,3)		(3,5)		(1,7.5)		(2,8)		−		(3,13)	
Score:	1	+	1	+	0.951	+	0.951	+	−0.4	+	1	= 4.502
$B_{1:6}$:	(1,0)		(3,1)		(2,4)		(1,4.2)		(4,7)		(3,9)	

(b) Optimal local alignment

Figure 3.3: Local alignment result by modified Smith-Waterman algorithm.

It can be found that in this example, the similarity indices of A to B and B to A are close to each other; and the optimal local alignments are the same. Usually this is the case. Finally, we obtain the normalized similarity index of the two sequences:

$$S(A, B) = \frac{\max(4.502, 4.584)}{\min(7, 7)} = 0.655.$$

3.5 Case Study

The proposed method is applied to data from a hydrocracking unit of a typical refinery process consisting of equipment such as furnaces, pumps, compressors,

	Δ	(4,0)	(1,3)	(3,5)	(1,7.5)	(2,8)	(3,13)	(1,20)
Δ	0	0	0	0	0	0	0	0
(1,0)	0	0	1	0.812	1	0.6	0.812	1
(3,1)	0	0	0.812	2	1.624	1.224	1.6	1.624
(2,4)	0	0	0.992	1.6	2.992	2.624	2.224	2.592
(1,4.2)	0	0.001	1	1.2	2.6	3.984	3.584	3.224
(4,7)	0	1	0.6	1.370	2.2	3.584	4.355	3.955
(3,9)	0	0.6	0.490	1.6	1.8	3.184	4.584	4.184
(2,12.5)	0	0.2	0.090	1.2	1.4	2.8	4.184	3.984

Figure 3.4: B to A similarity table H.

separation drums, etc. In the alarm data over a period of 9 months, there are totally 506 alarm tags and each alarm tag may have several different alarm identifiers such as 'PVLO', 'PVHI', 'PVLL' or 'PVHH'. As a result, the alarm data includes more than 1,000 types of alarms. Thanks to the good basic alarm rationalization performance that process control engineers have done, the process has a good normal alarm statistics, namely under 6 alarms/h. The main concern is alarm floods when some abnormalities or events take place. Our pattern matching technique is suitable for this kind of alarm event data, since it is easy for us to identify the alarm flood periods simply by counting the alarm frequency in the event data. According to the definitions of alarm flood in [41, 58], we identified 39 alarm floods over this period. The lengths of the 39 alarm flood sequences vary from 10 to 297 alarms.

We first preprocess the 39 alarm flood sequences by merging the alarms with the same alarm type and occurring within 5 seconds (5 seconds off-delay timer) to eliminate chattering alarms. The rationality and effect of the preprocessing is thoroughly discussed in [54]. Then the proposed algorithm is applied on each pair of preprocessed alarm flood sequences to calculate the normalized similarity index. Finally we obtain a 39×39 similarity matrix which is show in Fig. 3.5 in a color map format. The results are obtained when we set $\delta = -0.4$, $\mu = -0.6$, and $\sigma^2 = 4$. In the color map, the darker the color the more similar the two sequences are. All the diagonal entries in the color map are black, i.e. have the value 1. It is expected, since properties 1 and 2 of the proposed algorithm guarantee that the normalized similarity

index of two identical alarm sequences is always 1. We can also find some pairs with high similarity (normalized similarity index is greater than 0.6) such as (1, 14), (26,36), (24,30) and so on. It is confirmed by the site engineers that the found similar pairs are reasonable.

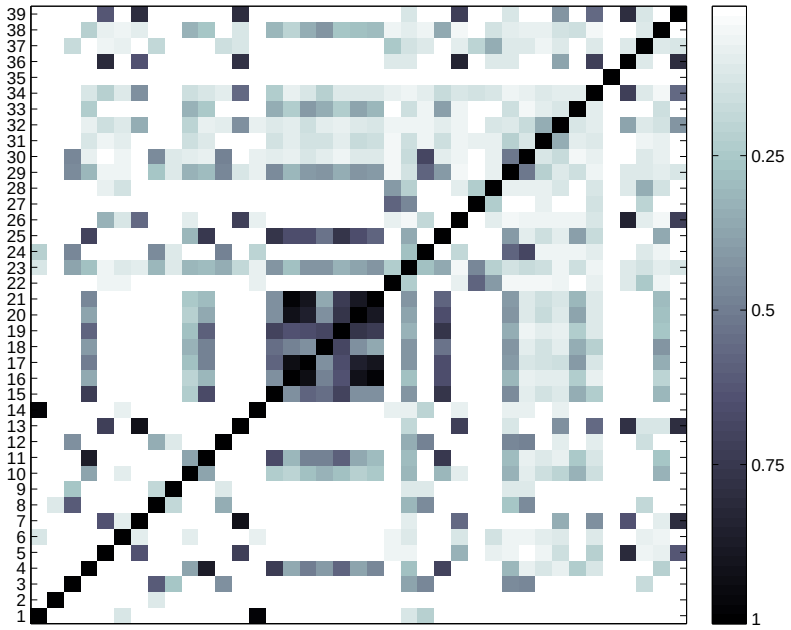

Figure 3.5: Similarity color map.

Fig. 3.6-3.8 show the alignment of several pairs of alarm sequences, which can give us a clearer picture of the algorithm. For the first pair (floods 1 and 14), no swap is needed to get the optimal alignment. In this case, the modified algorithm works similar as the standard algorithm. In the second pair (floods 26 and 36), the modified algorithm finds that if we shift the alarm '001.PVHI' by 2 seconds, a better local alignment can be obtained, which can increase the similarity of the two alarm flood sequences. For the third pair (floods 24 and 30), a lot of alarms are raised almost simultaneously. In this case, the order of the sequence is of little significance, therefore the original Smith-Waterman algorithm cannot handle it. But the modified algorithm can discover the similarity between the two sequences since it is insensitive to the sequence, to some extent.

Finally, we cluster the floods using single-linkage method [108]. This clus-

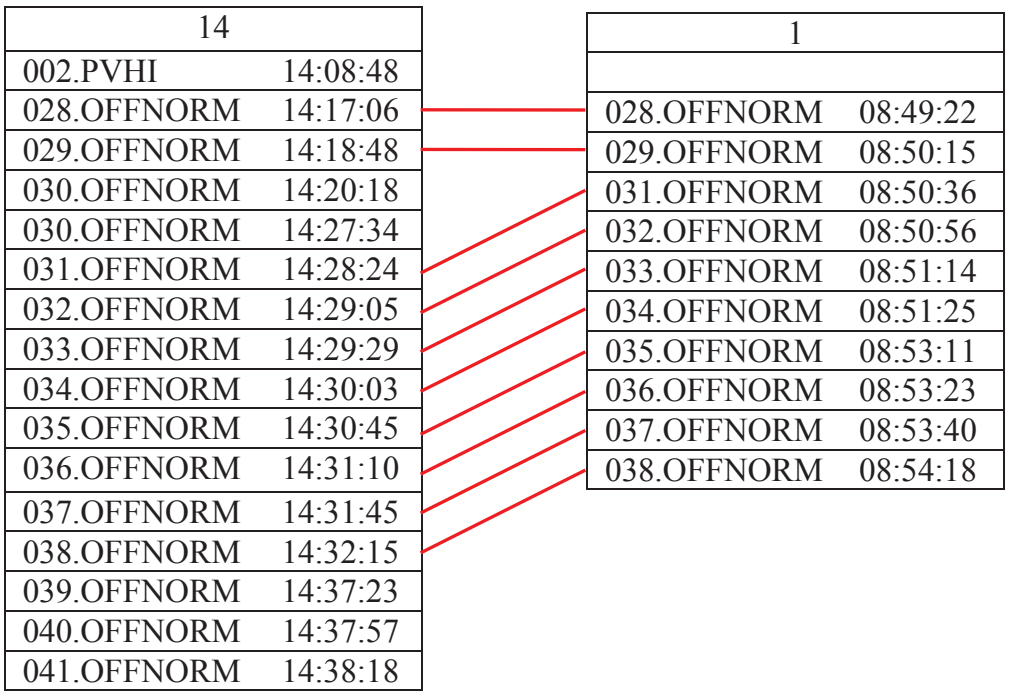

Figure 3.6: Alarm flood sequences alignment (flood 1 and 14).

tering method is also applied in alarm data correlation analysis to cluster related types of alarms [8, 109]. Fig. 3.9 shows the clustered similarity color map. Setting the threshold at 0.6, the groups we obtain from the 39 floods are: (16, 21, 17, 20), (15, 19, 25), (4, 11), (5, 7, 13, 39), (26, 36), (24, 30), (1, 14). If we decrease the threshold, the first three groups and the 4th and 5th groups merge. After clustering, we can obtain the characteristic subsequence(s) of a flood group according to the optimal alignment of alarm flood sequence pairs in the group. For example, for the first group (16, 21, 17, 20), we can find the following characteristic subsequence: [‘046.PVHI’, ‘046.BADPV’, ‘047.BADPV’, (‘048.OFFNORM’ and ‘047.PVHI’), (‘046.PVHI’ and ‘046.BADPV’)].

We have also tried the classical Smith-Waterman algorithm on the same alarm data set to make a comparison. It is easy to prove that the similarity indices obtained by the modified algorithm are no smaller than the ones calculated by the original algorithm, since the optimal alignment of the original algorithm always falls in the feasible region (but not necessary the optimal one) of the modified one.

In the case study, there is no obvious discrepancy on the similarity indices between the modified and the original Smith-Waterman algorithms for the

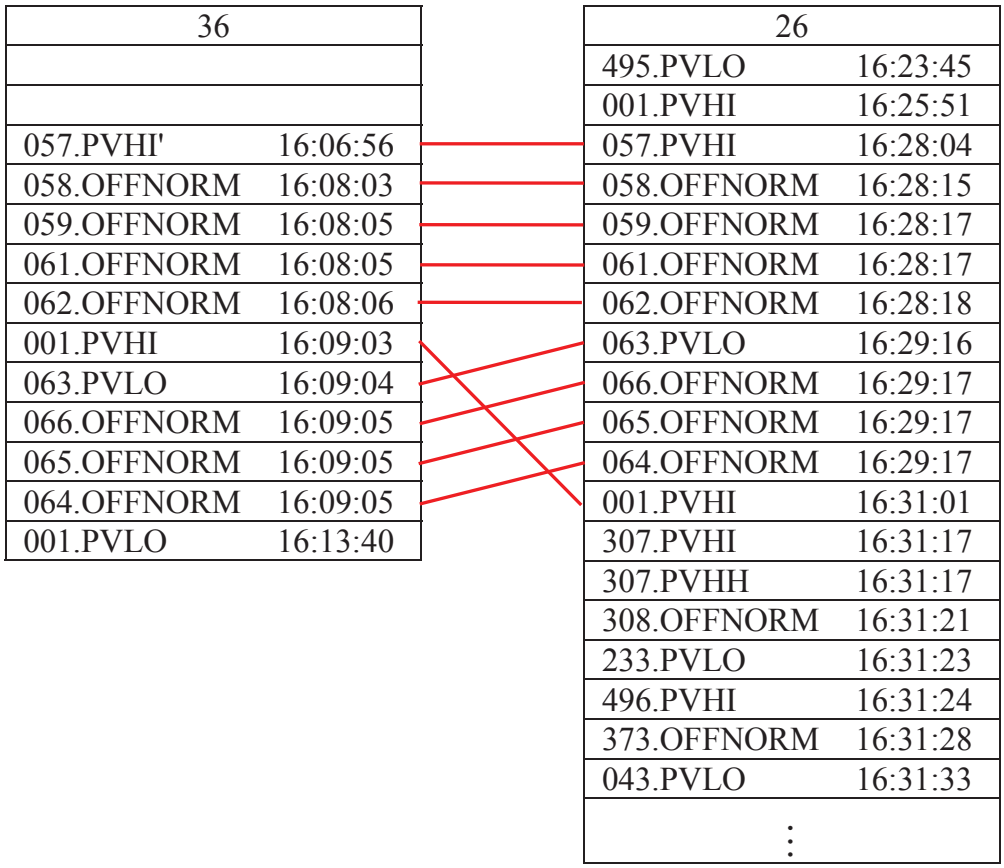

Figure 3.7: Alarm flood sequences alignment (flood 26 and 36).

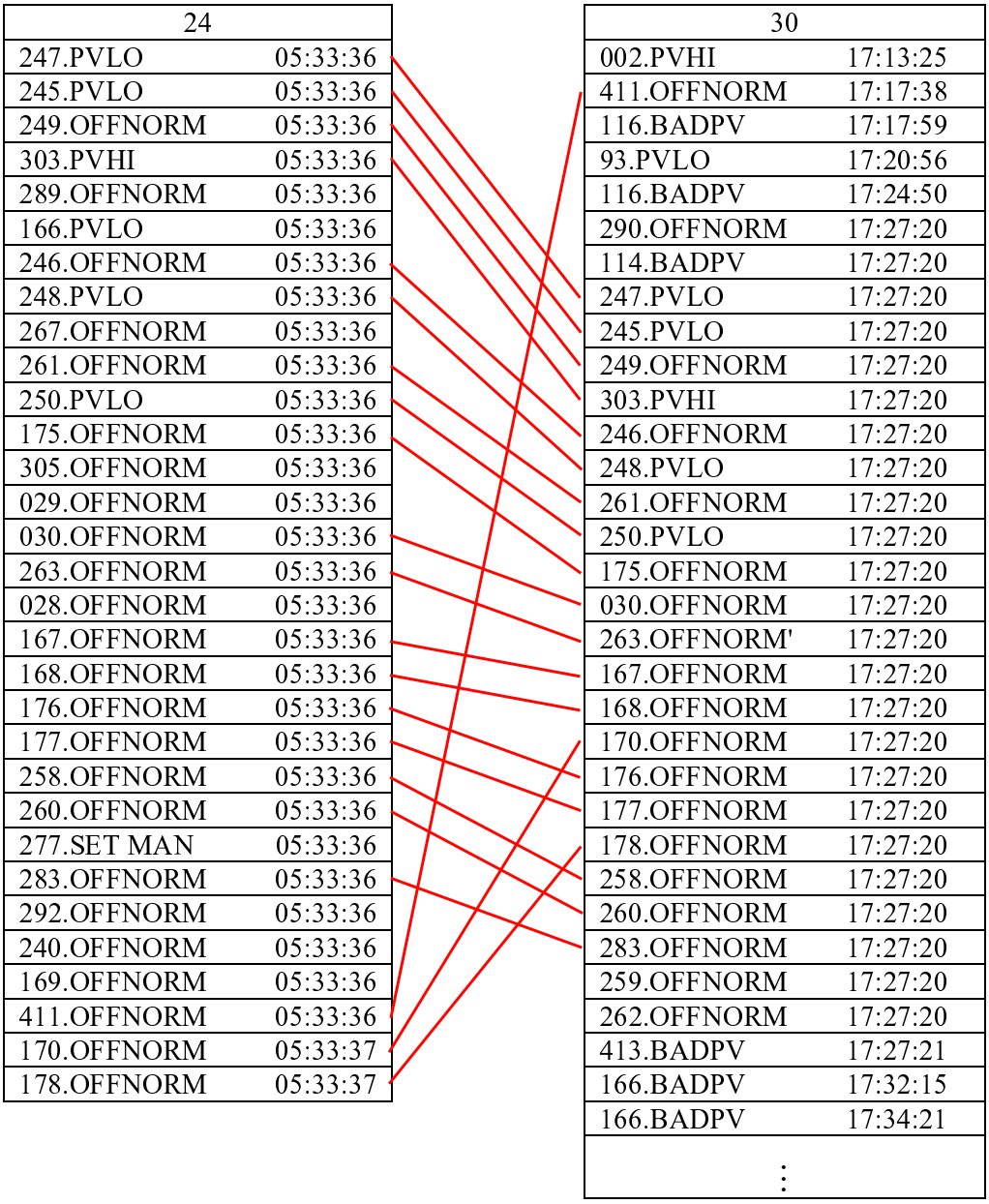

Figure 3.8: Alarm flood sequences alignment (flood 24 and 30).

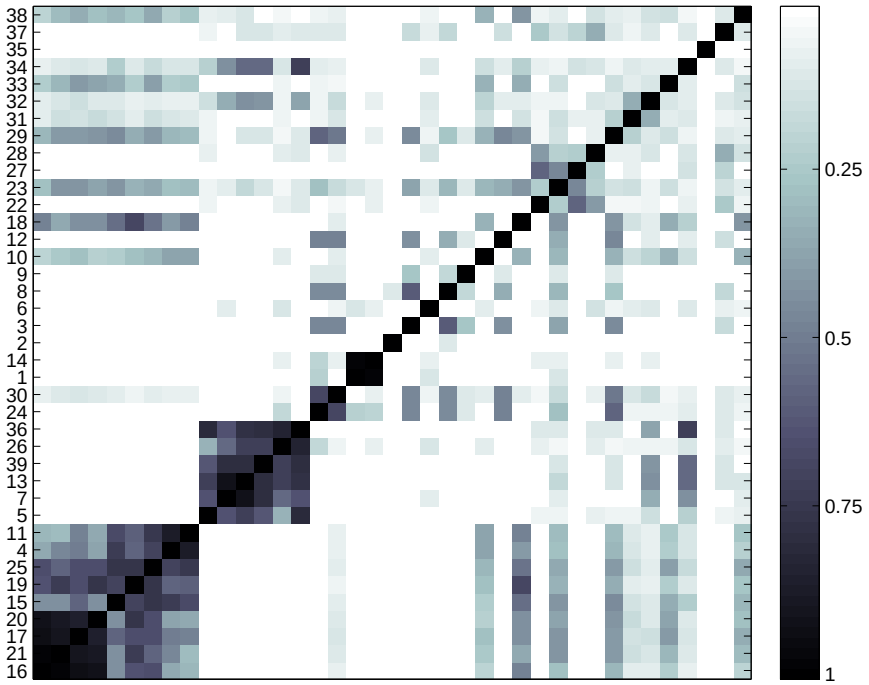

Figure 3.9: Clustered similarity color map.

alarm sequences whose chronological orders are very clear, e.g., flood 1 and flood 14 shown in Fig. 3.6. More than 90% alarm sequence pairs in our case study are in this case. However, the similarity indices of the remaining ones using the modified algorithm increase obviously. For example, the similarity index of floods 24 and 30 shown in Fig. 3.8 decreases from 0.63 to 0.46 if the modified algorithm is replaced by the original one. Another typical example is floods 7 and 13 as shown in Fig. 3.10. The similarity index generated by the modified algorithm is pretty high (0.9), while reduced to 0.68 by the original algorithm. The reason of this reduction is the segments marked by red boxes in Fig. 3.10. The time intervals of these three segments are very short, so the orders of the alarms affect the original algorithm much more than the modified algorithm. In this group, a low-flow alarm leads to a low pressure. Then the low pressure affects three sequential pressure variables to be abnormal almost simultaneously. In total we can find 9 more pairs of alarm sequences whose similarity indices are greater than 0.6 by the modified algorithm.

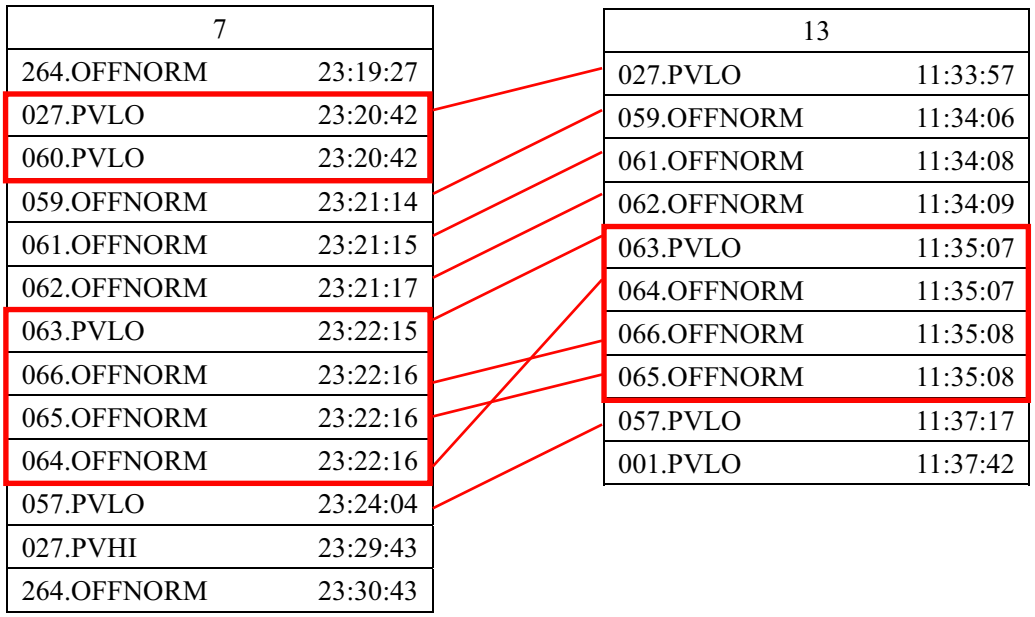

Figure 3.10: Alarm flood sequences alignment (flood 7 and 13).

The modified algorithm can find more similar floods than the classical one, since the modified similarity indices are always no less than the original one. This is the main advantage of the modified algorithm: decreasing the probability of false negative. It can get more similar flood pairs and discover similarities that the original algorithm cannot find, such as floods 1 and 14,

and floods 7 and 13. Meanwhile it may also be a disadvantage, since it may introduce more false positive similarities. Since the goal of our work is to construct a pool of similar flood group candidates to aid the engineers or experts, emphasizing more on low false negative rate is reasonable.

Moreover, the modified algorithm introduces a new control parameter: the variance of the scaled Gaussian weighting function. The modified algorithm converges to the classical one when the variance approaches 0. When the variance is very large, the order information is totally discarded, and the algorithm only counts the common alarms in the two floods. As a result, this control parameter affects the matching result. Choose other weighting functions instead of the Gaussian function may further impact the result. Although it makes the algorithm more flexible, it also increase the difficulty to handle the algorithm. Weighting function selection and its relationship to the distribution of detection delay are still open questions of the modified algorithm.

Another disadvantage of the modified algorithm is its heavier computational burden. The calculation of classical Smith-Waterman similarity indices of all the flood pairs in this case study finished in 0.3 second using Matlab, but it takes about half a minute to finish all the calculation for the modified algorithm. The main reason is the calculation of similarity score. In the original algorithm, only one Boolean calculation is required, but in the modified algorithm, K multiplication operations, where K is the size of the alphabet, and one maximum operation are necessary. As a result, the computational complexity for one flood pair is $O(MNK)$ (the classical Smith-Waterman algorithm's complexity is $O(MN)$), where M and N are the length of the two floods, respectively. When there are Q floods in the historical data, the similarity indices of $Q(Q-1)/2$ flood pairs are required, so the total computational burden is $O(MNKQ^2)$. However, the processing time is acceptable since this analysis is off-line.

Chapter 4

Application of the Principal Component Pursuit Technique to Process Fault Detection and Diagnosis*

4.1 Overview

In the previous chapter, we studied consequence alarms based on alarm data, which is one way to discover the connections between different alarm tags. In this chapter, we focus on multivariate process data. As mentioned in Section 1.1, aside from a single process variable, a function of several process variables may also be used as an alarm signal. For example, the deviation of the set point and the flow rate of a pump can be used as an alarm signal to monitor the performance of the pump. A classical technique for designing such linearly combined alarm signals based on process data is the principal component analysis (PCA). Recently, a new statistical analysis technique, principal component pursuit (PCP), has been proposed. It is also named robust PCA due to its robustness to outliers. In this chapter, the application of PCP to abnormality detection and diagnosis is examined from training data preprocessing to residual signal post-filtering.

*A version of this chapter has been submitted to Automatica, and a short version has been published in [27]

4.2 Preliminary on Principal Component Pursuit

Background

PCP has recently been introduced and discussed in several references [17, 25, 26, 114, 119]. PCP is not sensitive to outliers since it only concerns on number of outliers instead of their magnitudes as classical PCA. Therefore, an advantage of PCP over classical PCA lies in its enhanced robustness against outliers. Moreover, the new technique outperforms other robust PCA methods with regards to computational complexity and technical constraints imposed on original problems.

The PCP technique stems from compressed sensing [20, 37], which reveals a surprising message: the minimum number of data needed to reconstruct a signal may overcome the limitation imposed by the Nyquist-Shannon criterion if the signal is sparse in a certain sense. Inspired by this idea, a matrix completion method was proposed to recover a data matrix from only a few entries [18]. Furthermore, the pioneers focused on a more challenging problem: recovering a low rank data matrix contaminated by gross errors in several of its entries. A novel approach, the PCP technique, was then developed in [17, 25]. The essential idea of the PCP technique is to replace the original non-convex optimization problem of the matrix rank and the count of non-zero entries by a convex optimization problem of the nuclear and ℓ_1 norms. In [25, 26], deterministic conditions for the equivalence of the two optimization problems have been provided. Statistical counterparts are available in [17]. While these conditions are relatively mild, they greatly depend on the coherence of the uncontaminated data, which was first introduced in compressed sensing [19] and then adopted to the problems of matrix completion and PCP [17, 18]. Generally speaking, the smaller the coherence indices, the lower the requirement on the completeness of the signal/matrix.

The PCP technique has found successful applications in the area of image processing, e.g., video surveillance [17, 91] and face recognition [113]. There are also some attempts to apply PCP to latent sematic indexing [81]. Apart from these areas, PCP has the potential to be applied to a broader range of problems including that of process monitoring, which is studied in this chapter. In [59], a preliminary comparison between process monitoring approaches based on PCA and PCP has been given. It can be concluded that the PCP

technique is promising in process monitoring because the PCP-based method can overcome most of the shortcomings of PCA-based methods. Despite this, in-depth research on the application of PCP to process monitoring still remains open.

Problem Formulation and Main Result of PCP

Suppose $\mathbf{L}_0 \in \mathbb{R}^{n_1 \times n_2}$ is a low rank data matrix. $\mathbf{S}_0 \in \mathbb{R}^{n_1 \times n_2}$ is a sparse perturbation matrix. $\mathbf{M}$ is the sum of $\mathbf{L}_0$ and $\mathbf{S}_0$, namely, the observed data matrix. The goal of PCP is to recover $\mathbf{L}_0$ from $\mathbf{M}$, which is formulated as the following optimization problem:

$$\begin{aligned} &\text{minimize} \quad \operatorname{rank}(\mathbf{L}) + \lambda \|\mathbf{S}\|_0 \\ &\text{subject to} \quad \mathbf{L} + \mathbf{S} = \mathbf{M}. \end{aligned} \tag{4.1}$$

Here, $\mathbf{L}$ and $\mathbf{S}$ are the decision variables to be solved. $\|\mathbf{S}\|_0$ is the 0 norm of $\mathbf{S}$, namely, the number of non-zero entries in $\mathbf{S}$. The optimal solution of $\mathbf{L}$ and $\mathbf{S}$ are good estimations of the low rank data matrix $\mathbf{L}_0$ and the sparse perturbation matrix $\mathbf{S}_0$, respectively.

In the context of process monitoring, $\mathbf{M}$ represents the measurement data matrix of a multivariate process with n_2 measurement signals. Each row represents an observation vector, and n_1 samples are collected in total. The measurement data matrix $\mathbf{M}$ is contaminated by outliers, namely the outlier matrix $\mathbf{S}_0$, and the data matrix $\mathbf{L}_0$ is the outlier-free data matrix, namely $\mathbf{M} - \mathbf{S}_0$. Solving the optimization problem in (4.1) is the procedure of recovering the outlier-free data matrix $\mathbf{L}_0$ from the contaminated measurement data matrix $\mathbf{M}$.

The optimization problem in (4.1) is very difficult to solve. However, the authors of [17, 25] pointed out that the ℓ_1 norm is a good replacement of the 0 norm and the nuclear norm is a good substitute for the matrix rank. As a result, the original optimization problem in (4.1) can be modified as follows:

$$\begin{aligned} &\text{minimize} \quad \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 \\ &\text{subject to} \quad \mathbf{L} + \mathbf{S} = \mathbf{M}. \end{aligned} \tag{4.2}$$

This is a convex optimization problem, which can be solved efficiently and reliably by a variety of algorithms such as the alternating direction method [17]. Nevertheless, is the solution to problem (4.2) a good approximation

of problem (4.1)? In other words, is the solution to problem (4.2) a good approximation of the low rank matrix $\mathbf{L}_0$ and sparse matrix $\mathbf{S}_0$? A positive answer is important, since it makes no sense to use the solution to problem (4.2) for the purpose of data recovery if this is not the case.

Theorem 1.1 in [17] answers this question. In the theorem, the concept of coherence is used. The coherence of a subspace is defined in [18]:

Definition 4.1. Let $\mathbf{\Omega}$ be a subspace of $\mathbb{R}^n$ of dimension r and $\mathbf{P}_{\mathbf{\Omega}}$ be the orthogonal projection onto $\mathbf{\Omega}$. Then the coherence of $\mathbf{\Omega}$ is defined to be

$$\mu(\mathbf{\Omega}) = \frac{n}{r} \max_{1 \leq i \leq n} \|\mathbf{P}_{\Omega} \mathbf{e}_i\|^2, \tag{4.3}$$

where $\mathbf{e}_i$ is the i-th standard basis vector.

Suppose the singular value decomposition (SVD) of the rank-r matrix $\mathbf{L}_0$ is given by $\mathbf{L}_0 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{n_1 \times r}$ and $\mathbf{V} \in \mathbb{R}^{n_2 \times r}$. The SVD used here is different from the conventional one. Since $\mathbf{L}_0$ is a low rank matrix, there are some zero singular values of the matrix. So, $\mathbf{\Sigma}$ used here is a diagonal matrix only including these non-zero singular values, and matrices $\mathbf{U}$ and $\mathbf{V}$ only include their columns that correspond to non-zero singular values. This compact expression keeps all of the information in the conventional SVD. The coherences of the column and row spaces of $\mathbf{L}_0$, namely, $\mathbf{U}$ and $\mathbf{V}$, are:

$$\begin{aligned} \mu(\mathbf{U}) &= \frac{n_1}{r} \max \|\mathbf{P}_{\mathbf{U}} \mathbf{e}_i\|^2; \\ \mu(\mathbf{V}) &= \frac{n_2}{r} \max \|\mathbf{P}_{\mathbf{V}} \mathbf{e}_i\|^2. \end{aligned} \tag{4.4}$$

The mutual-coherence of $\mathbf{U}$ and $\mathbf{V}$ can also be defined as:

$$\mu_1 = \sqrt{\frac{n_1 n_2}{r}} \|\mathbf{U}\mathbf{V}^T\|_{\infty}. \tag{4.5}$$

Since the matrices $\mathbf{U}$ and $\mathbf{V}$ are the orthonormal basis of the corresponding subspaces, we have that

$$\begin{aligned} &\mathbf{P}_{\mathbf{U}} = \mathbf{U}\mathbf{U}^T; \\ &\mathbf{P}_{\mathbf{V}} = \mathbf{V}\mathbf{V}^T; \\ &\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}. \end{aligned}$$

Then, equation (4.4) can also be written as

$$\begin{aligned} \mu(\mathbf{U}) &= \frac{n_1}{r} \max \|\mathbf{U}\mathbf{U}^T \mathbf{e}_i\|^2 = \frac{n_1}{r} \max \|\mathbf{U}^T \mathbf{e}_i\|^2; \\ \mu(\mathbf{V}) &= \frac{n_2}{r} \max \|\mathbf{V}\mathbf{V}^T \mathbf{e}_i\|^2 = \frac{n_2}{r} \max \|\mathbf{V}^T \mathbf{e}_i\|^2. \end{aligned} \tag{4.6}$$

Denote the maximum value of $\mu(\mathbf{U})$, $\mu(\mathbf{V})$ and μ_1 by μ, Theorem 1.1 in [17] can be expressed as follows:

Theorem 4.1. *Suppose* $\mathbf{L}_0$ *has coherence index* μ*, and that the support set, namely, the set of non-zero positions in the matrix, of* $\mathbf{S}_0$ *is uniformly distributed among all sets of cardinality* m*. Then, there is a numerical constant* c *such that with a probability of at least* $1 - c\max(n_1, n_2)^{-10}$*, PCP with* $\lambda = 1/\sqrt{\max(n_1, n_2)}$ *is* exact*, provided that*

$$\begin{aligned} &\operatorname{rank}(\mathbf{L}_0) \leq \rho_r \min(n_1, n_2)\mu^{-1}\log^{-2}(\max(n_1, n_2)), \\ &\text{and } m \leq \rho_s n_1 n_2. \end{aligned} \tag{4.7}$$

In other words, the matrix $\mathbf{L}_0$ can be exactly recovered with a probability of nearly one. The probability converges to one when the size of the matrix goes to infinity, whose convergence rate is related to numerical constants ρ_s and ρ_r.

Remark 5. According to Theorem 4.1, the condition for the equivalence of optimization problems in (4.1) and (4.2) is related to the coherence μ. The lower the μ is, the milder the condition is.

4.3 Process Model Building Based on PCP

To build a process model, there are two steps. The first step is preprocessing the data matrix to make it more suitable for PCP analysis; and the second step is applying PCP to the preprocessed data and finding out the process model, namely the row subspace of the outlier-free data matrix $\mathbf{L}_0$. Since the second step has been extensively discussed in the PCP literature, we mainly focus on the first step: data preprocessing.

4.3.1 Scaling in Data Preprocessing

Preprocessing is an important but commonly overlooked step in PCA-based methods. A main issue in preprocessing is scaling. Variables in different units, e.g., meters or centimeters, should be scaled. Variables in the same unit or in different types of units are also usually scaled if their variances are very different. Currently, the simplest yet most widely used scaling method is normalization, namely, dividing each variable by its standard deviation.

However, there is debate over the effect of scaling on data quality. Specifically, how can one guarantee that the distortion induced by scaling is always

acceptable? Although some papers try to answer this question [12, 110], their statements are solely based on case studies; thus we believe that a definite and general conclusion has not yet been illustrated.

Intuitively, the PCP technique should not be influenced by variable scaling, since neither the rank nor the sparsity of a data matrix changes after scaling. However, this is not true. The PCP technique can recover the low rank and sparse matrices when the conditions in (4.7) are satisfied. However, ill scaling may greatly increase the coherence index and violate the conditions. Consequently, PCP may fail to recover the data matrices even if the rank and sparsity do not change. An example is shown below.

Example 4.1. A numerical example is studied. The data set has 35 variables and 1000 samples. So, $\mathbf{L}_0$ and $\mathbf{S}_0$ are 1000×35 matrices. The underlying rank is 5, i.e., $\text{rank}(\mathbf{L}_0) = 5$, and 5% of data are outliers. Time trends of the original signals (only the first seven and last seven variables) are shown in Fig. 4.1. There are some scaling issues, particularly with the third and last four variables. The variances of these five variables are extremely large, leading to large values of $\mu(\mathbf{V})$ and μ_1. The PCP result is far from the real $\mathbf{L}_0$ and S_0 we used to set up the example. By using the normalized data, $\mu(\mathbf{V})$ decreases from almost 7 to 2.4. μ_1 also decreases from greater than 10 to less than 6. PCP can reach a correct result for the normalized data. By using other scaling parameters we can further decrease $\mu(\mathbf{V})$ to about 1.2 and μ_1 to less than 5.

As a result, we may decrease the coherence of the data matrix and improve PCP performance by optimizing the scaling parameters. In recent years, a similar topic has been extensively discussed in the area of compressed sensing [40, 42]. The fundamental idea is to minimize the coherence between the dictionary matrix and the projection matrix by selecting appropriate projection matrices. However, to the best of our knowledge, no work addresses this idea in the context of PCP-based process monitoring. The reasons may be twofold: firstly, the projection matrix in compressed sensing is an arbitrarily designed matrix while the data matrix in PCP is not designable; secondly, current applications of PCP are mainly in the field of image processing, where scaling is not a serious problem.

Given the observed data matrix $\mathbf{M} = \mathbf{L}_0 + \mathbf{S}_0 \in \mathbb{R}^{n_1 \times n_2}$, each column denotes a process variable, and each row represents a sample. Since we usually have sufficient historical data, the observed matrix has more rows than

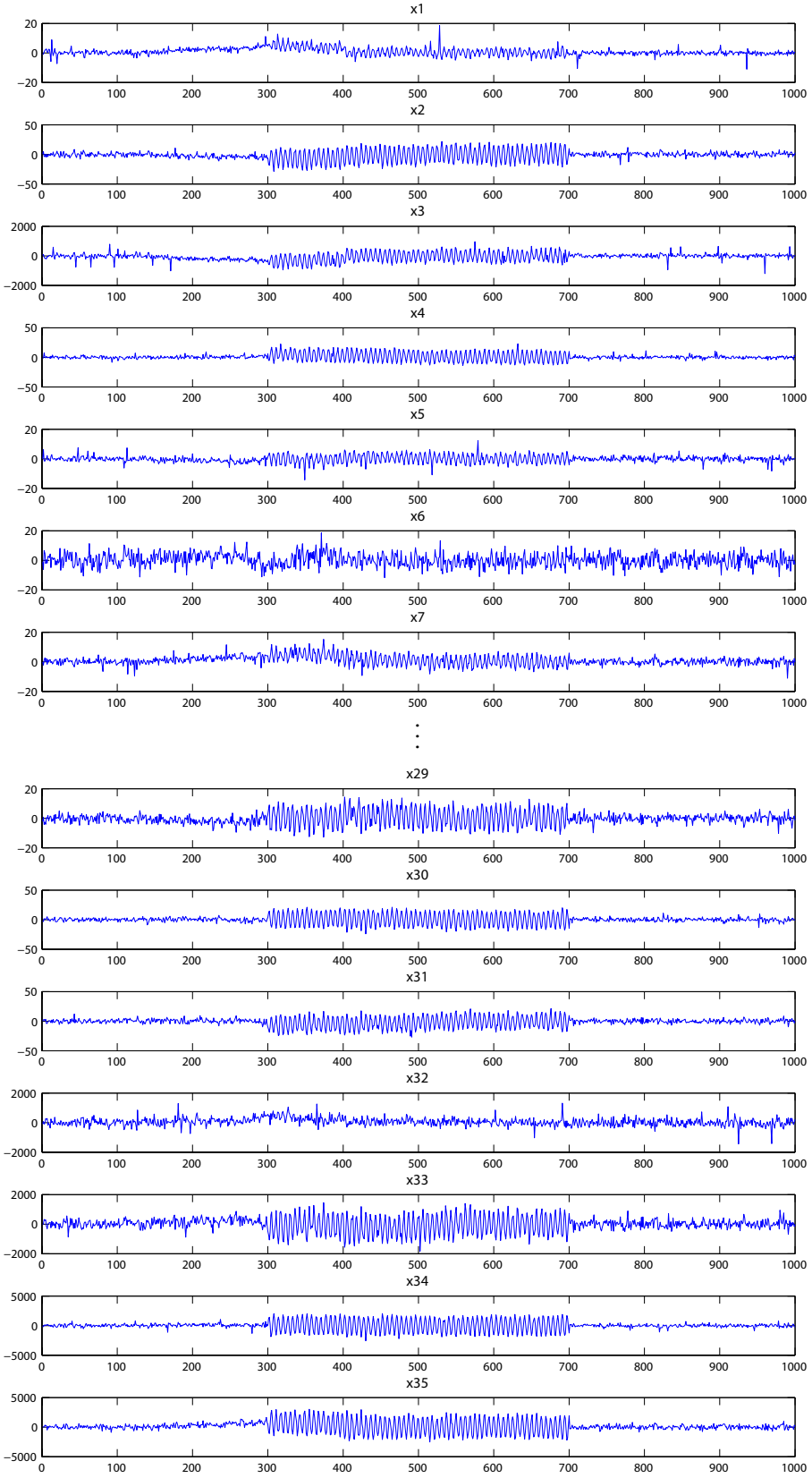

Figure 4.1: Time trend plots of the original signals.

columns, i.e., $n_1 > n_2$. Denote the scaling vector by $\mathbf{x} = [\ x_1 \quad x_2 \quad \cdots \quad x_{n_2}\ ]^T$, $x_i \neq 0$ for $i = 1, \cdots, n_2$, and we have the SVD of the outlier-free matrix: $\mathbf{L}_0 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, thus the scaled outlier-free matrix is:

$$\mathbf{L}_s = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{D}(\mathbf{x}).$$

Analogously, we have the SVD of the scaled matrix: $\mathbf{L}_s = \mathbf{U}_s\mathbf{\Sigma}_s\mathbf{V}_s^T$, where $\mathbf{U}_s \in \mathbb{R}^{n_1 \times r}$, $\mathbf{V}_s \in \mathbb{R}^{n_2 \times r}$. The left null space of $\mathbf{L}_0$ is the same as that of $\mathbf{L}_s$. As a result, $\mu(\mathbf{U}) = \mu(\mathbf{U}_s)$, since

$$\begin{aligned}\frac{r}{n_1}\mu(\mathbf{U}) &= \max \|\mathbf{U}\mathbf{U}^T\mathbf{e}_i\|^2 = \max(\mathrm{diag}(\mathbf{U}\mathbf{U}^T)) \\ &= 1 - \min(\mathrm{diag}(\mathbf{I} - \mathbf{U}\mathbf{U}^T)) \\ &= \max(\mathrm{diag}(\mathbf{U}_s\mathbf{U}_s^T)) \\ &= \frac{r}{n_1}\mu(\mathbf{U}_s).\end{aligned}$$

However, $\mu(\mathbf{V}) \neq \mu(\mathbf{V}_s)$.

Since $\mu(\mathbf{U})$ cannot be altered by scaling, and the upper bound of μ_1 is $\sqrt{r\mu(\mathbf{U})\mu(\mathbf{V}_s)}$ [18], a rational way to decrease coherence is to minimize $\mu(\mathbf{V}_s)$ by optimizing the scaling vector $\mathbf{x}$.

4.3.2 A Coordinate Descent Algorithm for Data Scaling

To solve the optimization problem mentioned above, a coordinate descent algorithm is proposed. Due to some convenient properties of the objective function, the algorithm can ultimately converge to a global optimal point, as shown later. According to the previous discussion, the objective function to be minimized can be chosen as follows:

$$J(\mathbf{x}) = \frac{r}{n_2}\mu(\mathbf{V}_s) = \max(\mathrm{diag}(\mathbf{V}_s\mathbf{V}_s^T)) \tag{4.8}$$

In order to simplify the problem, we need to rewrite the objective function as an explicit expression with respect to $\mathbf{x}$. The key is to find out an explicit expression of $\mathbf{V}_s$ with respect to $\mathbf{x}$. $\mathbf{L}_s$ can be expressed as follows:

$$\begin{aligned}\mathbf{L}_s &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{D}(\mathbf{x}) \\ &= \mathbf{U}\mathbf{\Sigma}\left(\mathbf{V}^T\mathbf{D}(\mathbf{x})^2\mathbf{V}\right)^{1/2}\left(\mathbf{V}^T\mathbf{D}(\mathbf{x})^2\mathbf{V}\right)^{-1/2}\mathbf{V}^T\mathbf{D}(\mathbf{x}).\end{aligned}$$

Note that $\mathbf{U}\mathbf{\Sigma}\left(\mathbf{V}^T\mathbf{D}(\mathbf{x})^2\mathbf{V}\right)^{1/2} \in \mathbb{R}^{n_1 \times r}$ is a full column rank matrix. So its SVD is:

$$\mathbf{U}\mathbf{\Sigma}\left(\mathbf{V}^T\mathbf{D}(\mathbf{x})^2\mathbf{V}\right)^{1/2} = \mathbf{U}_{s1}\mathbf{\Sigma}_{s1}\mathbf{V}_{s1}^T,$$

where $\mathbf{U}_{s1} \in \mathbb{R}^{n_1 \times r}$, $\mathbf{V}_{s1} \in \mathbb{R}^{r \times r}$. thus:

$$\mathbf{L}_s = \mathbf{U}_{s1}\mathbf{\Sigma}_{s1}\left(\mathbf{V}_{s1}^T\left(\mathbf{V}^T\mathbf{D}(\mathbf{x})^2\mathbf{V}\right)^{-1/2}\mathbf{V}^T\mathbf{D}(\mathbf{x})\right).$$

We know that the SVD of $\mathbf{L}_s$ is $\mathbf{U}_s\mathbf{\Sigma}_s\mathbf{V}_s^T$, and notice that

$$\left(\mathbf{V}_{s1}^T\left(\mathbf{V}^T\mathbf{D}(\mathbf{x})^2\mathbf{V}\right)^{-1/2}\mathbf{V}^T\mathbf{D}(\mathbf{x})\right)\left(\mathbf{V}_{s1}^T\left(\mathbf{V}^T\mathbf{D}(\mathbf{x})^2\mathbf{V}\right)^{-1/2}\mathbf{V}^T\mathbf{D}(\mathbf{x})\right)^T = \mathbf{I},$$

so

$$\begin{aligned}&\mathbf{U}_s = \mathbf{U}_{s1};\ \mathbf{\Sigma}_s = \mathbf{\Sigma}_{s1};\\ &\mathbf{V}_s^T = \mathbf{V}_{s1}^T\left(\mathbf{V}^T\mathbf{D}(\mathbf{x})^2\mathbf{V}\right)^{-1/2}\mathbf{V}^T\mathbf{D}(\mathbf{x}).\end{aligned} \tag{4.9}$$

As a result, the objective function can be rewritten as:

$$\begin{aligned}J(\mathbf{x}) &= \max\left(\operatorname{diag}\left(\mathbf{D}(\mathbf{x})\mathbf{V}\left(\mathbf{V}^T\mathbf{D}(\mathbf{x})^2\mathbf{V}\right)^{-1}\mathbf{V}^T\mathbf{D}(\mathbf{x})\right)\right)\\ &= \max\left(\operatorname{diag}\left(\mathbf{V}\left(\mathbf{V}^T\mathbf{D}(\mathbf{x})^2\mathbf{V}\right)^{-1}\mathbf{V}^T\mathbf{D}(\mathbf{x})^2\right)\right).\end{aligned}$$

Let $\mathbf{y}$ denote $[\ x_1^2 \quad x_2^2 \quad \cdots \quad x_{n_2}^2\ ]^T$. The optimization problem can be expressed as:

$$\begin{aligned}&\text{minimize} \quad J(\mathbf{y})\\ &\text{subject to} \quad \mathbf{y} > 0,\end{aligned} \tag{4.10}$$

where

$$J(\mathbf{y}) = \max\left(\operatorname{diag}\left(\mathbf{V}\left(\mathbf{V}^T\mathbf{D}(\mathbf{y})\mathbf{V}\right)^{-1}\mathbf{V}^T\mathbf{D}(\mathbf{y})\right)\right). \tag{4.11}$$

Before introducing the algorithm, one assumption is added to the matrix $\mathbf{V}$:

Assumption 1. Matrix $\mathbf{V}\mathbf{V}^T$ cannot be transformed to a block diagonal matrix by row and column switchings. In other words, let $\mathbf{v}_k$ denote the k-th column of $\mathbf{V}^T$, thus for any partition that divides the set of indices $\{1, 2, \cdots, n_2\}$ into two subsets $\{i_1, i_2, \cdots, i_f\}$ and $\{j_1, j_2, \cdots, j_g\}$, the following inequality always holds:

$$[\ \mathbf{v}_{i_1} \quad \mathbf{v}_{i_2} \quad \cdots \quad \mathbf{v}_{i_f}\ ]^T[\ \mathbf{v}_{j_1} \quad \mathbf{v}_{j_2} \quad \cdots \quad \mathbf{v}_{j_g}\ ] \neq \mathbf{0}.$$

If such a partition exists, it is easy to prove that the i_1-th, i_2-th, $\cdots$, i_{ni}-th diagonal elements of the matrix $\mathbf{V}\left(\mathbf{V}^T\mathbf{D}(\mathbf{y})\mathbf{V}\right)^{-1}\mathbf{V}^T\mathbf{D}(\mathbf{y})$ only depend on y_{i_1}, y_{i_2}, $\cdots$, y_{i_f} and the other diagonal elements only depend on y_{j_1}, y_{j_2}, $\cdots$, y_{j_g}. So, the original problem can be decoupled into two smaller optimization problems with decision variable sets $\{y_{i_1}, y_{i_2}, \cdots, y_{i_f}\}$ and $\{y_{j_1}, y_{j_2}, \cdots, y_{j_g}\}$, respectively. As a result, when the assumption cannot be satisfied, we can separate the original optimization problem to several decoupled subproblems,

each of which satisfies the assumption, and then solve these subproblems one by one.

The coordinate descent algorithm applied in this chapter involves an inner iteration loop and an outer iteration loop. The outer loop is described in Algorithm 2, and Algorithm 3 provides details of the inner loop. Several new symbols are introduced for the description of Algorithm 3:

$$\begin{aligned}
&\mathbf{y}_{\setminus i} = [\ y_1 \ \cdots \ y_{i-1} \ \ y_{i+1} \cdots \ y_{n_2}\ ]^T;\\
&\mathbf{V}_{\setminus i} = [\ \mathbf{v}_1 \ \cdots \ \mathbf{v}_{i-1} \ \ \mathbf{v}_{i+1} \cdots \ \mathbf{v}_{n_2}\ ]^T;\\
&\mathbf{Q}_i(\mathbf{y}) = \mathbf{V}_{\setminus i}^T \mathbf{D}(\mathbf{y}_{\setminus i}) \mathbf{V}_{\setminus i};\\
&q_i(\mathbf{y}) = \mathbf{v}_i^T \mathbf{Q}_i(\mathbf{y})^{-1} \mathbf{v}_i;\\
&\mathbf{A}_i(\mathbf{y}) = \mathbf{V}_{\setminus i} \mathbf{Q}_i(\mathbf{y})^{-1} \mathbf{V}_{\setminus i}^T \mathbf{D}(\mathbf{y}_{\setminus i});\\
&\mathbf{B}_i(\mathbf{y}) = \mathbf{V}_{\setminus i} \mathbf{Q}_i(\mathbf{y})^{-1} \mathbf{v}_i \mathbf{v}_i^T \mathbf{Q}_i(\mathbf{y})^{-1} \mathbf{V}_{\setminus i}^T \mathbf{D}(\mathbf{y}_{\setminus i}),
\end{aligned}$$

for $i = 1, 2, \cdots, n_2$.

Algorithm 2 Optimal Scaling Search

1: **initialize:** $\mathbf{y}_0 = \mathbf{1}$.
2: calculate $J(\mathbf{y}_0)$.
3: **while** $J(\mathbf{y})$ does not converge ($|J(\mathbf{y}_k) - J(\mathbf{y}_{k-1})| > e$) and a prescribed bound of number of iterations doesn't reach **do**
4: calculate $\mathbf{y}_{k+1}$ from $\mathbf{y}_k$ by Algorithm 3;
5: **end while**
6: **outputs:** $\mathbf{y}$ and $J(\mathbf{y})$.

The algorithm can converge to a global optimal solution under some mild conditions, as shown in Section 4.4.

To apply the proposed algorithm, we need to know the matrix $\mathbf{V}$. However, what we have is the contaminated data matrix $\mathbf{M}$ rather than the outlier-free matrix $\mathbf{L}_0$. In order to obtain $\mathbf{V}$, we may use $\mathbf{M}$ to estimate $\mathbf{V}$, and use the approximation for optimization.

Example 4.2. We continue with Example 4.1. An approximate space basis $\mathbf{V}$ is obtained using the observed data. Then, the coordinate descent algorithm proposed above is performed. The algorithm stops when the value of objective function of two consecutive outer iterations is smaller than 10^{-10}. The algorithm converges to the global optimal point 1/7 in less than 30 outer iterations. After this optimization, $\mu(V_s)$ decreases from almost 7 to 1. μ_1 also decreases dramatically. We then apply Algorithm 1 in [17] to the well scaled data matrix and standard normalized data matrix. The number of iterations

Algorithm 3 Inner loop of Algorithm 2

1: **initialize:** $\widehat{\mathbf{y}} = \mathbf{y}_k$.
2: **for** j=1 to n_2 **do**
3: calculate $q = q_j(\widehat{\mathbf{y}})$, $\mathbf{A} = \mathbf{A}_j(\widehat{\mathbf{y}})$, and $\mathbf{B} = \mathbf{B}_j(\widehat{\mathbf{y}})$;
4: calculate $\Gamma = \max_{1 \le t \le n_2 - 1} (\gamma_t)$, where

$$\gamma_t = \frac{\mathbf{e}_t^T \mathbf{A} \mathbf{e}_t}{\mathbf{e}_t^T \mathbf{B} \mathbf{e}_t + q - \mathbf{e}_t^T \mathbf{A} \mathbf{e}_t q},$$

for $t = 1, 2, \cdots, n_2 - 1$;
5: update the j-th element of $\widehat{\mathbf{y}}$ by:

$$\left(\max\left\{\Gamma, \mathbf{e}_j^T \widehat{\mathbf{y}}\right\} + \mathbf{e}_j^T \widehat{\mathbf{y}}\right)/2;$$

6: **end for**
7: **outputs:** $\widetilde{\mathbf{y}} = \widehat{\mathbf{y}}$, and $\mathbf{y}_{k+1} = \dfrac{\widetilde{\mathbf{y}}}{\max\limits_{1 \le j \le n_2} (\widetilde{\mathbf{y}} \mathbf{e}_j)}$.

of the algorithm to converge using the well scaled data is smaller than using the standard normalized data.

4.4 Algorithm Convergence

The main convergence result will be proven by starting a few lemmas.

Lemma 4.1. *There exists a partition to divide the set of indices* $\{1, 2, \cdots, n_2\}$ *into two subsets* $\{i_1, i_2, \cdots, i_f\}$ *and* $\{j_1, j_2, \cdots, j_g\}$*, such that*

$$\mathbf{E}_i^T \mathbf{V} \mathbf{V}^T \mathbf{E}_j = \mathbf{0},$$

where

$$\mathbf{E}_i = [\ \mathbf{e}_{i_1} \quad \mathbf{e}_{i_2} \quad \cdots \quad \mathbf{e}_{i_f}\ ],$$
$$\mathbf{E}_j = [\ \mathbf{e}_{j_1} \quad \mathbf{e}_{j_2} \quad \cdots \quad \mathbf{e}_{j_g}\ ],$$

if and only if this partition satisfies the following equation with respect to $\mathbf{V}_s$*:*

$$\mathbf{E}_i^T \mathbf{V}_s \mathbf{V}_s^T \mathbf{E}_j = \mathbf{0}. \tag{4.12}$$

This lemma shows us that original variables can be divided into two groups that are independent if and only if the same partition can also divide the scaled variables into two independent groups. Because of this lemma, we can convert Assumption 1 with respect to $\mathbf{V}$ to the one with respect to $\mathbf{V}_s$, and

then simplify the proofs of the following lemmas and the main theorem by excluding many cases that Assumption 1 does not hold. A proof of Lemma 4.1 is provided in Appendix A.

Lemma 4.2. *The objective function satisfies*

$$\frac{r}{n_2} \leq J(\mathbf{y}) < 1.$$

This lemma provide a lower bound and an upper bound of the objective function. It will be shown later in Theorem 4.2 that the lower bound is tight in most cases. A proof of Lemma 4.2 is provided in Appendix A.

The third lemma provides four useful properties of Algorithm 3 that will mainly be used in the proof of Lemma 4.4.

Lemma 4.3. *In each inner iteration, namely, steps 2 to 6 in Algorithm 2, only the j-th element of $\widehat{\mathbf{y}}$ may change. Moreover, let $\widehat{y}_{0j}$ and $\widehat{y}_{1j}$ denote the value of the j-th element of $\widehat{\mathbf{y}}$ before and after updating, respectively. Let $\widehat{\mathbf{y}}_0$ and $\widehat{\mathbf{y}}_1$ denote the value of $\widehat{\mathbf{y}}$ before and after updating, respectively. Let*

$$d_i(\mathbf{y}) = \mathbf{e}_i^T \mathbf{V}(\mathbf{V}^T \mathbf{D}(\mathbf{y})\mathbf{V})^{-1}\mathbf{V}^T \mathbf{D}(\mathbf{y})\mathbf{e}_i,$$

for $i = 1, 2, \cdots, n_2$. The inner iteration has the following four properties:

1. *$\widehat{y}_{1j} \geq \widehat{y}_{0j}$. The equality holds if and only if $J(\widehat{\mathbf{y}}_0) = d_j(\widehat{\mathbf{y}}_0)$.*
2. *$d_i(\widehat{\mathbf{y}}_0) \geq d_i(\widehat{\mathbf{y}}_1)$ for all is other than j.*
3. *$J(\widehat{\mathbf{y}}_0) \geq d_j(\widehat{\mathbf{y}}_1)$. The equality holds if and only if $J(\widehat{\mathbf{y}}_0) = d_j(\widehat{\mathbf{y}}_0)$.*
4. *$J(\widehat{\mathbf{y}}_0) \geq J(\widehat{\mathbf{y}}_1)$. The equality holds if and only if there exists such an index $o \in \{1, \cdots, n_2\}$ that $J(\widehat{\mathbf{y}}_0) = d_o(\widehat{\mathbf{y}}_0) = d_o(\widehat{\mathbf{y}}_1)$*

A proof of Lemma 4.3 is shown in Appendix A.

Based on Lemma 4.3, the fourth lemma points out that the objective function does not increase or stay at a value before the lower bound is reached.

Lemma 4.4. *For each outer loop, namely, steps 3 to 5 in Algorithm 2, $J(\mathbf{y}_k) \geq J(\mathbf{y}_{k+1})$. For each n_2 loops, the decreasing is strict, i.e., $J(\mathbf{y}_k) > J(\mathbf{y}_{k+n_2-1})$ unless $J(\mathbf{y}_k) = \frac{r}{n_2}$, which is the lower bound of the objective function.*

A proof of Lemma 4.4 is shown in Appendix A.

Finally, the main result of convergence can be introduced.

Theorem 4.2. *The coordinate descent algorithm proposed in this chapter, namely, Algorithms 2 and 3, converges to a global optimal point whose objective function equals* $\frac{r}{n_2}$ *provided that: for any* i *rows of the matrix* $\mathbf{V}$ *whose spanned space has a rank, denoted by* $\check{n}_i$*, smaller than* i *and* n_2*, 1) there are more than* $\frac{(r-\check{n}_i)n_2}{r}$ *rows that do not lie in the row space of these* i *rows; 2) the rank of the spanned space of these* i *rows and any other* $r-\check{n}_i$ *rows that do not lie in the row space of these* i *rows is* r*.*

In order to prove this Theorem, two cases are considered: the algorithm does not converge to a boundary point; and the algorithm converges to a boundary point. In the former case, the proof of convergence is similar to the proof of Lyapunov stability theory. The algorithm can be considered as a discrete time dynamic system. The objective function (4.11) minus its lower bound provided in Lemma 4.2 can be considered as an energy function. Moreover, Lemma 4.4 shows that the energy function decreases unless the lower bound is reached. In the latter case, the problem is more complicated, since the reduction in each iteration may be larger than but converge to 0. That is why the two conditions are provided in Theorem 4.2. It can be proven that the latter case never occurs if the two conditions hold. A detailed proof of Theorem 4.2 is provided in Appendix A.

4.5 Fault Detection and Diagnosis

In Section 4.3, the modeling procedure is discussed. A main purpose of building process models is process monitoring, namely, fault detection and diagnosis (FDD). Applications of PCA to FDD are extensively discussed, e.g. [22, 32, 90]. A PCA generated process model determines two subspaces: the principal component subspace (PCS) and the residual subspace (RS). A measurement is decomposed by projecting it to the two subspaces. The ℓ_2 norm of the projection in RS (the residual vector) is called the squared prediction error. Another statistic, T^2 statistic, describes the variation of a process in the PCS. Control limits of these two indices are provided for fault detection [22].

The PCP generated model also provides PCS and RS. $\mathbf{V}\mathbf{V}^T$ denotes the PCS and $\mathbf{V}_\perp\mathbf{V}_\perp^T$ is the RS. The projections of a measurement in the two

subspaces are usually determined by an optimization problem:

$$\begin{aligned} \text{minimize} \quad & ||\tilde{\mathbf{x}}|| \\ \text{subject to} \quad & \hat{\mathbf{x}} + \tilde{\mathbf{x}} = \mathbf{x} \\ & \mathbf{V}_{\perp}\hat{\mathbf{x}} = 0, \end{aligned} \tag{4.13}$$

where $\mathbf{x}$ is the measurement, $\hat{\mathbf{x}}$ and $\tilde{\mathbf{x}}$ are its projections in PCS and RS, respectively. When the ℓ_2 norm is used in the objective function, the optimal solution is the orthogonal projection, which is the projection used in PCA methods. However, in the case that sparse outliers exist, ℓ_1 norm, which is a heuristic of 0 norm, is a better choice. Moreover, since the fault cardinality is usually small, the sparsity of the residual can still hold in the abnormal case. One question that must be answered is whether the solution obtained by minimizing the ℓ_1 norm can well estimate the sparse residual vector. According to [38], a key point to guarantee the equivalence of the sparse solution and the minimal ℓ_1 norm solution is that sparse vectors cannot lie in the complement of the RS, namely, the PCS. This is also the requirement of PCP decomposition, and the reason why we search for optimal scaling parameters. As a result, the RS of the well-scaled data rather than the original data should be used for residual vector calculation.

In the PCP-based method, we will only use the residual vectors $\tilde{\mathbf{x}}$ in fault detection. The main reason why we avoid using the outlier-free vector $\hat{\mathbf{x}}$ is the stationary requirement of that vector. Only when the outlier-free signal is statistically stationary do the variation tests, such as the T^2 test, make sense. However, the stationary requirement usually cannot hold in practice, especially when the process works at several different operating points.

As stated in [59], the main advantage of applying the PCP-based fault detection, isolation, and reconstruction approach is its simplicity. All three purposes can be reached simultaneously. A fault is detected and isolated on a certain variable when the corresponding residual is non-zero, and the null hypothesis that the non-zero samples are outliers is rejected. $\hat{\mathbf{x}}$ is the reconstructed fault-free and outlier-free vector. In order to distinguish between outliers and faults, certain univariate post-filter should be applied to each residual variable. Because of the existence of outliers, also called impulse noise in the signal processing literature, distributions of residual variables in the fault-free case have heavy tails. As a result, a generalized median filter is a good choice [13, 71]. The design of this post-filter is based on an

assumption that abnormality or fault continues for several consecutive samples in a univariate signal, while outliers appear sparsely in one sample.

Example 4.3. We continue with Examples 4.1 and 4.2. 200 more samples are generated in the original PCS with 5% outliers. Three faults are added:

1. from the 161st sample, the 3rd variable cannot be updated, so the magnitude stays at the value of the 160th sample;

2. the 101st to 120th samples of the 5th variable have an offset of -4;

3. from the 31st sample, an offset of 800 is added to the 35th variable.

Using the optimized scaling vector and the corresponding PCS and RS we obtained in Example 4.2, we solve the optimization problem in (4.13) for each sample. Finally the residual signal is obtained. Since the residual is the summation of faults and outliers, it is hard to distinguish the faults from it. But the filtered signal via a generalized median filter provides a clear view of the faults. The time trends of the faulty signals (only the first seven and last seven variables, blue dash curves) and the corresponding residual signals (red solid curves) are shown in Fig. 4.2. Because of the interconnections among process variables, many of them diverge from their operating point in faulty case thought indeed the fault is on only one of them. For example, in the interval from the 101st sample to the 120th sample, we can observe changes on many of the signals, but no residual signal except the one of x_5 diverges from 0. So that we can detect this fault, and locate it on process variable x_5 simultaneously by PCP-based fault detection and diagnosis. The residual signals of PCA are also provided in Fig. 4.3. We cannot directly isolate and identify the faults according to these residual signals. If we apply the PCA-based isolation and reconstruction method proposed in [22], we need to try $35 + \frac{35!}{2! \times 33!} = 630$ different combinations, since two independent faults may occur simultaneously. If the cardinality of faults further increases, the number of combinations increases dramatically, which leads to a large computational burden.

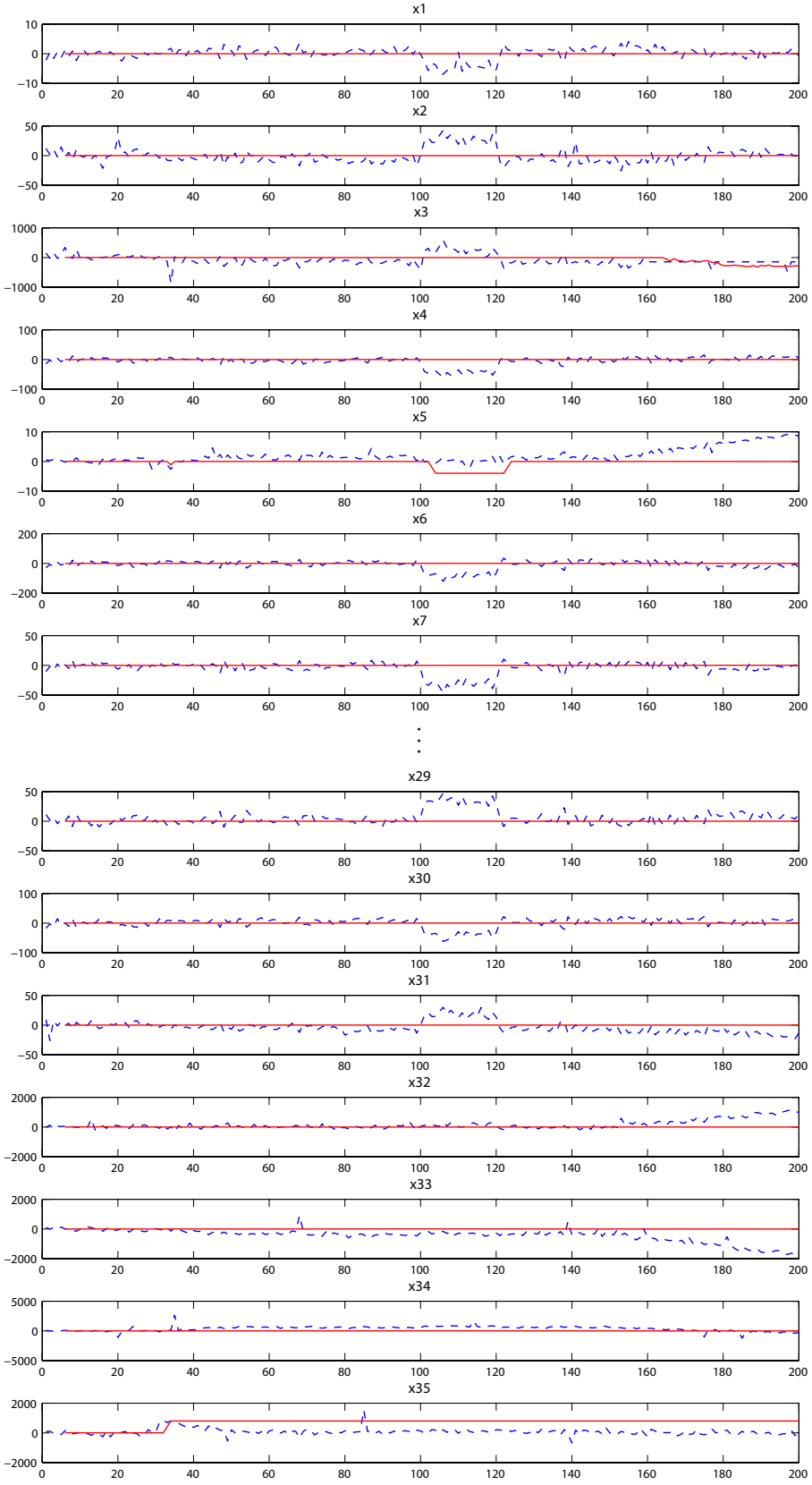

Figure 4.2: Time trend plots of the faulty signals and filtered residual signals.

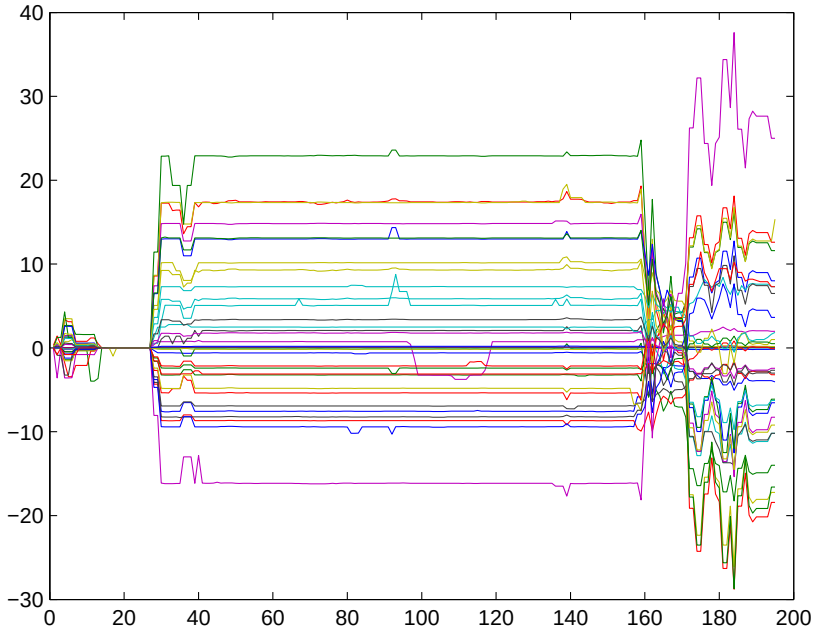

Figure 4.3: Filtered PCA residual signals.

Chapter 5

Application of Alarm Management Techniques to an Industrial Alarm Rationalization Project

5.1 Overview

In this chapter, we describe an industrial alarm rationalization project that is still in progress. Univariate alarm rationalization on bad tags has been finished. First, the top 50 bad alarm tags were identified according to their alarm counts in the historical alarm data. Then, delay timers were designed based on the alarm data of the top bad tags. Filter design and alarm limit tuning were also done based on the process data corresponding to the top bad tags. An oscillation detection technique based on alarm data was also provided to detect oscillation tags. By applying these techniques, more than 60% of historical alarms could be suppressed.

5.2 The Industrial Plant and Its Alarm System

The alarm rationalization project is carried out at an oil-sand extraction plant. The feed of the extraction process includes oil sand slurry, cold water and hot water. The mixture is then separated into three layers by the primary, secondary and tertiary separation cells connected in series. The top layer, namely, the bitumen froth, is the main product. The middle layer, which is

called middlings, is recycled back into the process. The bottom layer, which is mainly composed of sand and clays, is the tailings.

Process variables are measurements on a vast range of flow rates, temperatures, levels, pressures, and so on. Analog alarm tags are generated corresponding to these process variables. Aside from these analog alarm tags, there are many digital alarm tags as well, such as communication alarm tags for the field bus.

Four segments of half-month historical alarm data were examined. We found that the average alarm rate is higher than 1 alarm per second, which is much higher than the recommended value in the ISA standard, during all of these four periods. Although the top 50 bad tags listed in different periods were different, they shared many common tags. Moreover, more than half of the alarms were from the top 3 bad alarm tags. Because of this situation, it is necessary to perform bad alarm tag management as the first stage.

5.3 Delay Timer and Filter Design on Bad Tags

5.3.1 Delay Timer Design

There are two kinds of delay timers: on-delay timer and off-delay timer. Generally speaking, on-delay timers are good at avoiding fleet alarms, and off-delay timers can lock the alarm indication when repeat alarms are raised. The main design parameter of a delay timer is its length.

Based on the historical alarm data, specifically the alarm (ALM) messages and the return to normal (RTN) messages in the alarm log, a practical and effective method to design the length of off-delay and on-delay timers was developed in Section 5.6 in [65]. For a certain tag, time differences between each ALM message and the following RTN message are calculated. When an on-delay timer with length k is implemented, the reduction of the alarm count equals the amount of time differences lower than k. Then, a curve of alarm count deduction with respect to the length of the on-delay timer can be plotted for the on-delay timer design. Analogously, the length of the off-delay timer can be designed based on the time differences between RTN messages and the following ALM messages.

By adopting this method, the top 50 alarm tags during the four periods were analyzed. An interesting range of length of the delay timers is from 1

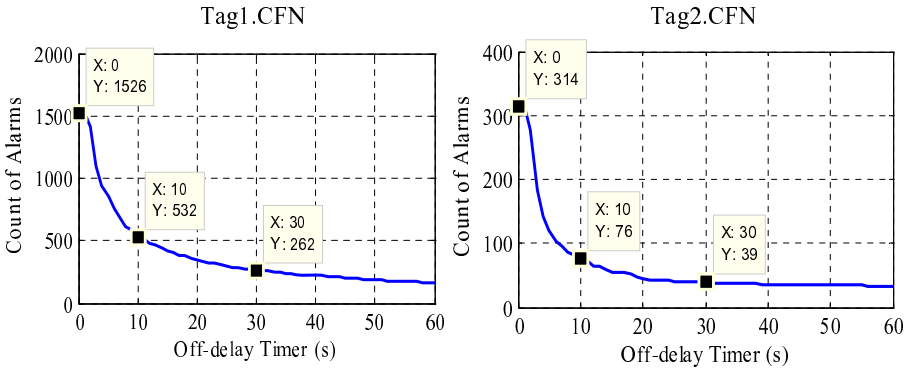

Figure 5.1: Off-delay timer analysis of Tag1.CFN (left) and Tag2.CFN (right).

second to 60 seconds. The detection delay will be too great if an on-delay timer that is too long is applied, and standing alarms may be introduced if the length of the applied off-delay timer is too large. We discovered that many bad tags could be mitigated by applying a 10- to 30-second off-delay timer, especially a group of "change from normal" (CFN) alarm tags. Plots of remaining alarm count v.s. length of off-delay timer for two CFN alarm tags are provided in Fig. 5.1 as examples. The alarm counts of the two tags could be reduced to about 15% with 30-second off-delay timers.

5.3.2 Filter Design and Alarm Limits Tuning

Since filter design and alarm limits tuning are based on process data, we only focused on the analog alarm tags in which delay timers were not effective. For some of these tags, we could only obtain data in the normal situation. If this was the case, we only focused on the distributions of the data in the normal case and their false alarm rates.

Compared with limits tuning, filtering is preferred since changing alarm limits requires much paper-work and demands very solid reasons.

According to the distributions of the data, we found that filtering techniques should be effective for most of the tags. Generalized median filters were selected. This kind of filters is the generalization of moving average filters, as well as the generalization of median filters. The advantages of moving average filters have been analyzed in Chapter 2; they are optimal linear filters under certain conditions. However, we also mentioned that moving average filters are sensitive to outliers. In other words, performance quality decreases when the distributions have heavy tails. Conversely, median filters are good

at rejecting outliers. As a result, generalized median filters, the combination of median filters and moving average filters, inherit the advantages of both median filters and moving average filters. As a result, the generalized median filters are good at suppressing noise as well as rejecting outliers. The design parameters of a generalized filter are two lengths, l and n. Denote the original alarm signal by $x(k)$ and the filtered signal by $y(k)$. The expression of the filtered signal is as follows:

$$y(k) = \frac{\sum_{i=n+1}^{l-n} \widetilde{x}_i}{l-2n},$$

where $\{\widetilde{x}_1, \widetilde{x}_2, \cdots, \widetilde{x}_l\}$ is the sequence, $\{x(k-l+1), x(k-l+2), \cdots, x(k)\}$, sorted in descending order.

The effects of adding generalized median filters are obvious. For example, the low alarm of a flow rate tag was raised 470 times during a period of about 40 hours. The low limit was 1000. Based on the alarm event log, all of these alarms should be false alarms. The signal is shown in Fig. 5.2 and its histogram is shown in Fig. 5.3. This distribution is typical with both noise and outliers. By adding a generalized median filter, both the false alarm rate and the alarm count can be reduced by about 80%.

Generalized median filters reduced the alarm counts significantly for most of the analog alarm tags that we focused on. One exception was a level control tag, which was among the top 5 bad tags during one of the four periods. We found that the distribution of the data had a peak very close to 15 which is the low alarm limit of this tag (the range of the variable is from 0 to 100). Because of this peak, low alarms of this tag were repeatedly raised and cleared. The performance could hardly be improved without alarm limit tuning. We attempted to decrease the limit slightly to 14. The alarm count would decrease dramatically by more than 90%.

5.4 Alarm Data-Based Oscillation Detection

5.4.1 Motivation

Oscillation is an abnormal phenomenon in process industries. As mentioned in [107], "the common causes for oscillation include sticky control valves, oscillatory external disturbances, loop interaction and aggressive control". In our project, we also encountered situations during which oscillation was present.

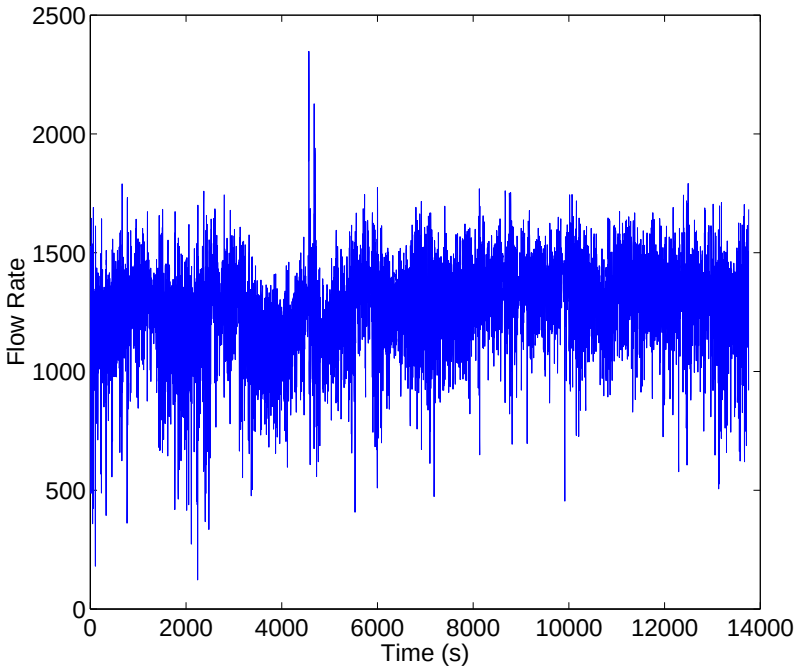

Figure 5.2: Time trend of a flow rate tag.

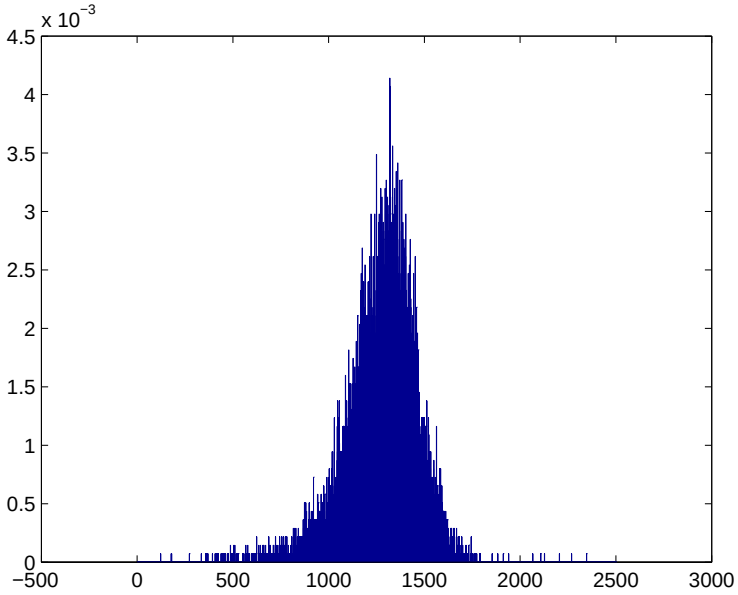

Figure 5.3: Histogram of the original data.

Because of oscillating process signals, alarms were repeatedly raised. The time differences between one ALM message to the next RTN message, and between one RTN message to the following ALM message, were mainly decided by the period and magnitude of the oscillation. A band-stop filter would be effective if the frequency of the oscillation was known. Unfortunately, it is very difficult to predetermine such a frequency band; and in fact, the oscillation frequency of one process variable may vary from time to time. Delay timers are also not a good solution for the repeated alarms caused by an oscillation, since the length of the delay timer has to be very large if the frequency of the oscillation is not very high. Moreover, the cause of an oscillation may be totally different from the cause of the mean value drift on the same process variable. Therefore, the alarm system needs to be able to distinguish between different phenomena.

There is a lot of valuable research on oscillation detection based on process data, such as [74, 107]. However, oscillation detection based on alarm data is also necessary in some circumstances, notably in our project. The reason is twofold. First, compared with the process data of hundreds or even thousands of process variables sampled every minute, the alarm data has a much smaller size and is easier to handle. As a result, detecting oscillating process variables by alarm data and then analyzing the process data of such variables provide a practical solution. Second, it may be impossible to obtain the "real" historical process data from the industrial database. Usually, historical data is stored after down sampling in consideration of computer space limitations. In our project, the down sampled period was 1 minute. Because of the frequency alias, the process data sometimes failed to provide the correct frequency information of an oscillation. For example, in one of the tags that we focused on, one oscillation whose period was about 45 seconds according to alarm data became 3 minutes in the down sampled process signal.

5.4.2 Detection Algorithm

Within alarm data, we can obtain the time stamps of ALM and RTN messages of a certain tag. In an ideal oscillation, the time gaps between two consecutive ALM messages, as well as those between two consecutive RTN messages, should be the same. Moreover, they should equal the period of the oscillation. In practice, such strict equalities do not hold due to the influence of noise. However, in the case that the signal to noise ratio is not small, variation be-

tween the time gaps is not too large. This was the case in our project, since the magnitude of the oscillation must be large enough to exceed the alarm limit. As a result, an alarm data-based oscillation detection algorithm was developed. Denote the time instances of the k-th ALM message and RTN message for a certain alarm as $t_a(k)$ and $t_r(k)$, respectively. Two types of time gaps can be defined as:

$$\begin{aligned} g_a(k) &= t_a(k) - t_a(k-1); \\ g_r(k) &= t_r(k) - t_r(k-1). \end{aligned}$$

Within the period where variations of both $g_a(k)$ and $g_r(k)$ are small, which can be measured by the standard deviation to mean ratio in a sliding window, an oscillation should occur. So, we can define the following two signals:

$$\begin{aligned} O_{\text{ALM}}(k) &= \frac{\sqrt{\sum_{i=k-l+1}^{k} (g_a(i) - \mu_a(k))^2}}{l\mu_a(k)}, \\ O_{\text{RTN}}(k) &= \frac{\sqrt{\sum_{i=k-l+1}^{k} (g_r(i) - \mu_r(k))^2}}{l\mu_r(k)}, \end{aligned} \tag{5.1}$$

where

$$\begin{aligned} \mu_a(k) &= \tfrac{1}{l} \sum_{i=k-l+1}^{k} g_a(i), \\ \mu_r(k) &= \tfrac{1}{l} \sum_{i=k-l+1}^{k} g_r(i). \end{aligned}$$

When both signals are smaller than a prescribed threshold, an oscillation is detected.

5.4.3 Detection Results

The algorithm was run on the historical alarm data we obtained. A sliding window with a length of 5 was used. Two typical oscillation tags and their oscillation periods were captured. Alarms repeated during the oscillation periods, which significantly increased alarm counts. If the oscillation detection method is used, only the first four repeated alarms would be raised, then a single standing oscillation alarm could replace all of those following during that oscillation period.

A segment of one oscillation tag is shown in Fig. 5.4. The red curve denotes the process signal and the blue line indicates the alarm status. From

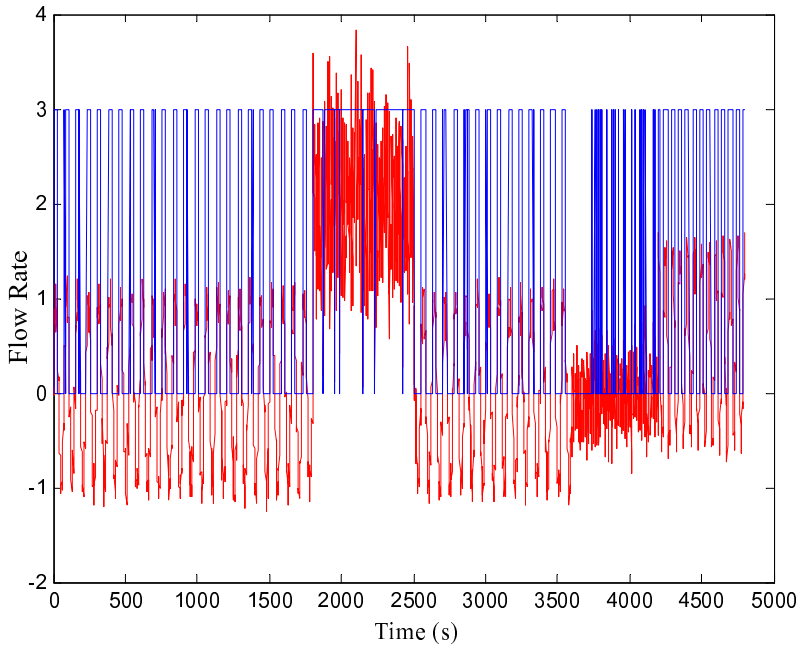

Figure 5.4: Process signal and alarm data of an oscillation tag.

one ALM message to the following RTN message, i.e., when an alarm is raised, the alarm status is assigned a value of 3 (see the blue line). Its value is set to be zero from one RTN message to the next ALM message. The alarm count, namely, the number of ALM messages, is 51. O_{ALM} and O_{RTN} signals are shown in Fig. 5.5, which shows three periods of oscillations. Notice that the horizontal axes of the two curves are not time axes; they indicate the index k in equation (5.1). In Fig. 5.6, two different kinds of repeated alarms of the same tag are shown. Obviously, the first segment is not an oscillation while the second should be. Our algorithm successfully distinguished these two situations.

5.5 Estimated Effect on the Alarm System

In summary, recommendations for the bad tags management stage were as follows: 1) a list of bad tags in which an off-delay timer or an on-delay timer of 30 seconds may be applied was provided; 2) generalized median filters were recommended on several process variables; 3) the low alarm limit of one tag should be slightly decreased; 4) analog alarm tags should be suppressed by design in the case of oscillations; 5) one tag should be shelved because it was

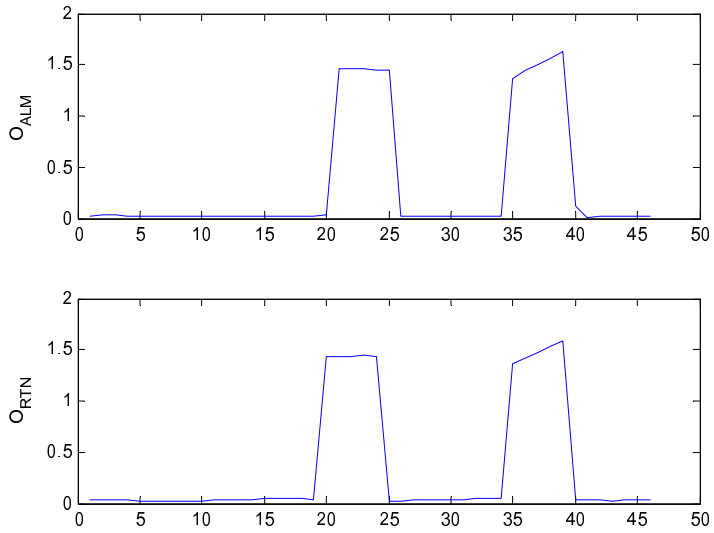

Figure 5.5: O_{ALM} and O_{RTN} signals.

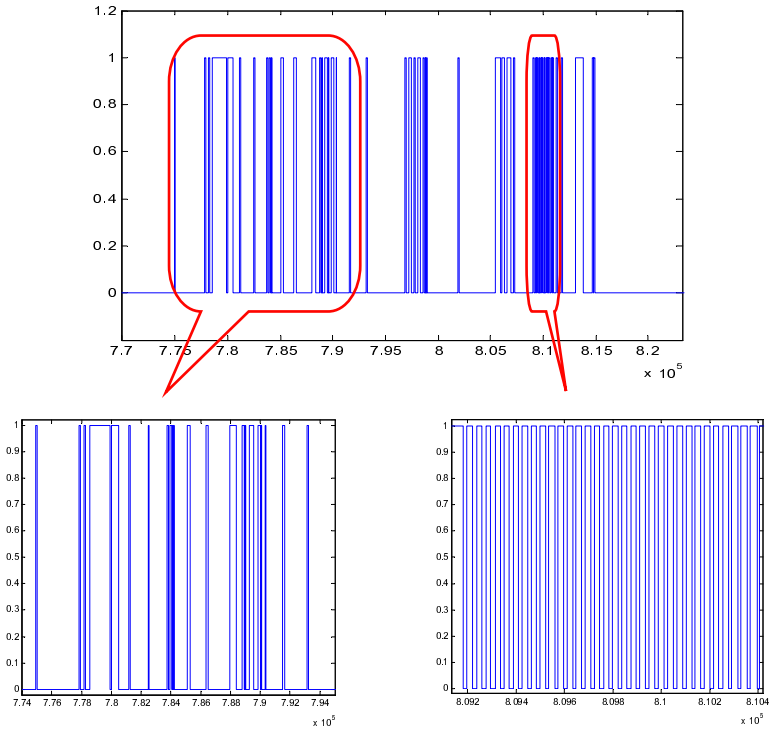

Figure 5.6: Alarm data of an oscillation tag in which both oscillation-induced repeated alarms and other repeated alarms exist.

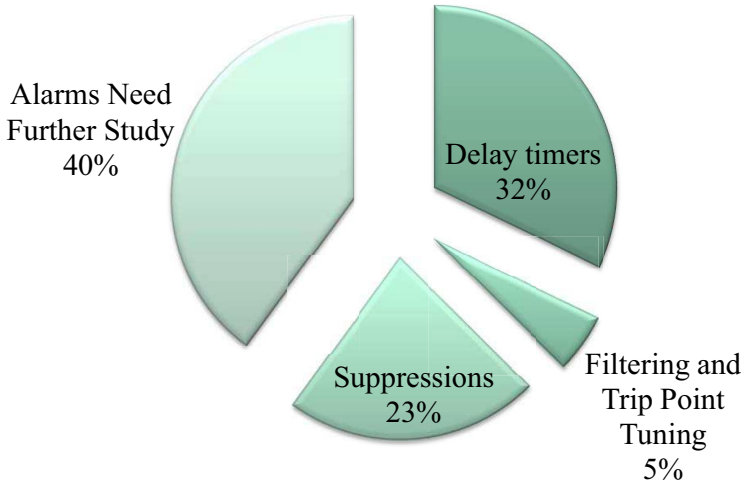

Figure 5.7: Contributions of different techniques on alarm count reduction.

caused by a device for which maintenance was required.

The alarm count could be reduced by about 60% if these recommendations were taken on the historical alarm data. A pie chart in Fig. 5.7 shows the contributions of different techniques on alarm count reduction.

Chapter 6

Concluding Remarks and Future Work

The purpose of the work reported in this thesis is to develop techniques and tools for alarm system design and analysis that can be applied to the improvement of alarm systems. In light of the current "data rich, information poor" situation, we focus on data-driven techniques. Both process data and alarm data are considered; and both univariate data-based and multivariate data-based methods are proposed.

6.1 Major Thesis Contributions

In Chapter 2, we present the design of an optimal alarm filter for maximum alarm accuracy, minimizing a weighted sum of false and missed alarm rates (probabilities). Alarm signal filtering is a univariate process data processing technique. The main function is to improve the quality of the alarm signal. With fixed filter structures, the design of optimal linear alarm filters is studied, and a numerical optimization procedure is proposed. Some key elements in the optimal design include the use of characteristic functions from probability theory to facilitate computation of the objective function, and a differential evolution (DE) algorithm for optimization (the optimization problem is non-convex and with small gradients). A sufficient condition for moving average filters being optimal linear alarm filters is also provided, which gives engineers an idea of when moving average filters can necessarily improve alarm performance.

Chapter 3 is concerned with alarm flood pattern analysis, which is helpful for root cause analysis of historical floods and for incoming flood prediction. In

this chapter, a multivariate alarm data-driven method for alarm flood pattern matching is proposed. A modified Smith-Waterman algorithm considering time stamp information is proposed to calculate a similarity index of alarm floods and uncover the connections between consequence alarms. The effectiveness of the algorithm is validated with a case study on real industrial data from a major refinery process.

Chapter 4 discusses the application of a new statistical analysis technique, principal component pursuit, to process monitoring. A new scaling preprocessing step is proposed to improve quality of data matrices for low coherence. An algorithm is proposed for the optimal scaling vector search. The convergence of the algorithm to a global optimal point is also proved. A residual generator and a post-filter suitable for PCP-based process models are also provided. The post-filtered residual represents the fault signal, which makes the fault detection, isolation, and reconstruction procedure simple and straightforward. A numerical example is also provided to illustrate the PCP-based process modeling and monitoring procedure.

In Chapter 5, an alarm system rationalization project on an industrial oil-sand extraction plant is discussed. Delay timers and filters are designed for the tags with frequent alarms (bad tags). An oscillation detection method based on alarm data is also provided to detect oscillation tags in the historical data set.

6.2 Directions for Future Work

Alarm management is still a relatively new direction in academia. This section focuses on future research directions that could be pursued.

Alarm Flood Analysis

Alarm flood analysis is an area that requires further studies. In this thesis, we provide an alarm data-based pattern matching technique for alarm flood pattern analysis. However, DCSs can also log another type of event, operator actions, aside from alarm events. Alarm flood sequences are greatly correlated with operator actions. Correct actions may stop a flood very quickly, while incorrect actions may further escalate the abnormality. As a result, alarm flood analysis considering both alarm events and operator actions is worth developing. Moreover, the outcome of the analysis can be used not only for

root cause analysis, but also for workflow mining. The best actions against different alarm flood patterns may be discovered when operator action data are incorporated. Another issue is how to combine data-driven methods and expert knowledge-based methods in alarm flood analysis. Computers and human beings are good at different tasks: computers are adept at large scale data processing, while the strength of human is the ability to make decisions on a case by case basis. The alarm flood analysis technique can be significantly improved if data-driven methods and knowledge-based methods come together to create a more superior integrated method.

PCP on Process Monitoring and Abnormality Management

Research on the application of PCP to process monitoring is still in its infancy stage. In this thesis, the relationship between the incoherence index and performance of the PCP algorithm is emphasized, and an optimal scaling vector design scheme is provided to minimize the incoherence. However, the quantitative relationship between the incoherence index and the properties of the PCP algorithm, e.g., its convergence ratio, the probability of obtaining an accurate solution, and so on, still needs further analysis. Moreover, PCP analysis has currently been discussed in the linear static situation. How to extend it to dynamic nonlinear situations is still an open question. Many techniques have already been proposed in the field of PCA to deal with similar issues. Dynamic PCA and nonlinear PCA techniques have been extensively discussed in [67, 68]. As a result, research on PCP may adopt such ideas from the PCA literature, such that dynamic and nonlinear PCP techniques may be developed.

Univariate Alarm Analysis and Design Based on Process and Alarm Data

Generalized median filtering and alarm data-based oscillation detection techniques are verified as effective univariate alarm processing techniques in the real industrial project described in Chapter 5. However, further theoretical analysis is required. Compared with that on moving average filters, theoretical analysis on generalized median filters is more difficult due to its nonlinearity. Alarm data-based oscillation detection methods work well in high signal-noise-ratio cases, but significance level analysis is still necessary for the threshold setting.

Implementation

Another important issue is how to integrate advanced techniques developed in this area and apply them to real industrial processes. Collecting alarm system analysis tools in a toolbox is a good idea. Preliminary attempts have been done in [5]. To implement online techniques, e.g., online alarm flood pattern matching and online oscillation detection, the involvement of a DCS programmer may be necessary. Alternatively, the online techniques may be integrated into existing alarm management platforms such as "Alarm Manager" developed by Honeywell-Marikon. Since the end users of these online tools would be operators, design on HMI is sometimes as important as the techniques themselves. Therefore, topics discussed in [53] are notable for further studying.

Bibliography

[1] Abnormal situation management consortium definition: impact. `http://www.asmconsortium.net/defined/impact/Pages/default.aspx`.

[2] J. Aach and G. Church. Aligning gene expression time series with time warping algorithms. *Bioinformatics*, 17(6):495–508, 2001.

[3] N.A. Adnan, I. Izadi, and T. Chen. On expected detection delays for alarm systems with deadbands and delay-timers. *Journal of Process Control*, 21(9):1318–1331, 2011.

[4] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[5] M.S. Amin. Developing matlab tools for data based alarm management and causality analysis. Master's thesis, University of Alberta, Edmonton, AB, 2012.

[6] A. Amir, Y. Aumann, Benson G., A. Levy, O. Lipsky, E. Porat, S. Skiena, and U. Vishne. Pattern matching with address errors: Rearrangement distances. *Journal of Computer and System Sciences*, 75(6):359–370, 2009.

[7] M. Y. An. Log-concave probability distirbutions: Theory and statistical testing. Technical report, Economics Department, Duke University, Durham, 1995.

[8] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice-Hall, Englewood Cliff, 1993.

[9] M. Bauer, J.W. Cox, M.H. Caveness, J.J. Downs, and N.F. Thornhill. Finding the direction of disturbance propagation in a chemical process

using transfer entropy. *IEEE Transactions on Control Systems Technology*, 15(1):12–21, 2007.

[10] D. Beebe, S. Ferrer, and D. Logerot. The connection of peak alarm rates to plant incidents and what you can do to minimize. *Process Safety Progress*, 32(1):72–77, 2013.

[11] T. Bergquist, J. Ahnlund, and J.E. Larsson. Alarm reduction in industrial process control. In *Proceedings of IEEE International Conference on Emerging Technologies and Factory Automation*, volume 2, pages 58–65, Lisbon, Portugal, Sept. 2003.

[12] M. G. Borgognone, J. Bussi, and H. Guillermo. Principal component analysis in sensory analysis: covariance or correlation matrix? *Food Quality and Preference*, 12(5-7):323–326, 2001.

[13] A. C. Bovik, T. S. Huang, and D. C. Munson. A generalization of median filtering using linear combinations of order statistcs. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 31(6):1342–1350, 1983.

[14] G. Box and A. Luceno. *Statistical Control By Monitoring and Feedback Adjustment*. John Wiley & Sons, New York, 1997.

[15] M.L. Bransdy and J. Jebkinson. The management of alarm system. HSE contract research report 166, HSE Book, 1998.

[16] D. Brook and D.A. Evans. An approach to the probability distribution of cusum run length. *Biometrika*, 59(3):539–549, 1972.

[17] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of ACM*, 58(3), 2011.

[18] E. J. Candes and B. Recht. Exact matrix completion via convex optimization. *Fundations of Computational Mathematics*, 9(6):717–772, 2009.

[19] E. J. Candes and J. Romberg. Sparsity and incoherence in compressive sampling. *Inverse Problems*, 23(3):969–985, 2007.

[20] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[21] R. Candy and J. Taisne. Advanced alarm processing facilities installed on Eskom's energy management system. In *Proceedings on IEEE PES Power System Conference and Exposition*, pages 1–8, Johannesburg, South Africa, July 2007.

[22] F. A. Carlos and S. J. Qin. Reconstruction-based contribution for process monitoring. *Automatica*, 45:1593–1600, 2009.

[23] S. Chakrabarti, S. Sarawagi, and B. Dom. Mining surprising pattern using temproal discription length. In *Proceedings of 24th International Conference on Very Large Databases*, pages 606–617, New York, USA, 1998.

[24] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection for discrete sequences: a survey. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):823–839, 2012.

[25] V. Chandrasekaran, S. Sanghavi, P. A.Parrilo, and A. Willsky. Sparse and low-rank matrix decompositions. In *15th IFAC Symposium on System Identification*, volume 15, pages 1493–1498,, Saint-Malo, France, 2009.

[26] V. Chandrasekaran, S. Sanghavi, P. A.Parrilo, and A. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

[27] Y. Cheng and T. Chen. Application of principal component pursuit to process fault detection and diagnosis. In *Proceedings of American Control Conference 2013*, Washington D.C., USA, June 2013.

[28] Y. Cheng, I. Izadi, and T. Chen. On optimal alarm filter design. In *Proceedings of International Symposium on Advanced Control of Industrial Processes*, pages 139–145, Hangzhou, China, May 2011.

[29] Y. Cheng, I. Izadi, and T. Chen. Optimal alarm signal processing: Filter design and performance analysis. *IEEE Transactions on Automation Science and Engineering*, 10(2):446–451, 2013.

[30] Y. Cheng, I. Izadi, and T. Chen. Pattern matching of alarm flood sequences by a modified Smith-Waterman algorithm. *Chemical Engineering Research and Design*, 91:1085–1094, 2013.

[31] Y. Chetouani. Use of cumulative sum (CUSUM) test for detecting abrupt changes in the process dynamics. 14(1):65–80, 2007.

[32] L. H. Chiang, E. L. Russell, and R. D. Braatz. *Fault Detection and Diagnosis in Industrial Systems*. Sorubger-Verlag, London, Great Britain, 2001.

[33] P. Cisar, E. Hostalkova, and P. Stluka. Data mining techniques for alarm rationalization. In *19th European Symposium on Computer Aided Process Engineering*, pages 1457–1462, Cracow, Poland, 2009.

[34] R. Cole and R. Hariharan. Approximate string matching: a faster simpler algorithm. In *Proceedings of 9th ACM-SIAM Symposium on Discrete Algorithms*, pages 463–472, New Orleans, USA, 1998.

[35] M. Cule and R. Samworth. Theoretical properties of the log-concave maximum likelihood estimator of a multidimensional density. *Electronic Journal of Statistics*, 4:254–270, 2010.

[36] Y Dombb, O. Lipsky, B. Porat, E. Porat, and A. Tsur. The approximate swap and mismatch edit distance. *Theoretical Computer Science*, 411(43):3814–3822, 2010.

[37] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1298–1306, 2006.

[38] D. L. Donoho. For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparest solution. *Communication on Pure and Applied Mathematics*, 59(6):797–829, 2006.

[39] P. Duan, F. Yang, T. Chen, and S.L. Shah. Direct causality detection via the transfer entropy approach. *IEEE Transactions on Control Systems Technology, Article in Press*, 2013.

[40] J. M. Duarte-Carvajalino and G. Sapiro. Learning to sense sparse signals: simulataneous sensing matrix and sparsifying dictionary optimization. *IEEE Transactions on Image Processing*, 18(7):1395–1408, 2009.

[41] EEMUA. *Alarm Systems: A Guide to Design, Management and Procurement*. EEMUA Publication 191. Engineering Equipment and Materials Users' Association, London, 2nd edition, 2007.

[42] M. Elad. Optimized projections for compressed sensing. *IEEE Transactions on Signal Processing*, 55(12):5695–5702, 2007.

[43] Westen Electric. *Statistical Qualit Control Handbook*. Western Electric Corporation, Indianapolis, IN, 1956.

[44] R. Eubank, V. Lariccia, and R. Rosenstein. Testing symmetry about an unkown median, via linear rank procedures. *Journal of Nonparametric Statistics*, 1(4):301–311, 1992.

[45] J. Folmer, D. Pantforder, and B. Vogel-Heuser. An analytical flood reduction to reduce operator's workload. In *HCII'11 Proceedings of the 14th International Conference on Human-computer Interaction: Users and Applications*, volume Part IV, pages 297–306, Orlando, FL, USA, 2011.

[46] J. Folmer and B. Vogel-Heuser. Computing dependent industrial alarms for alarm flood reduction. In *9th International Multi-Conference on Systems, Signal and Devices*, pages 1–6, Chemnitz, Germany, 2012.

[47] X. Ge and P. Smyth. Deformable Markov model templates for time-series pattern matching. In *Proceedings of the 6th ACM SIGKDD*, pages 81–90, Boston, MA, USA, 2000.

[48] C.W.J. Granger. Investigating causal relationships by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

[49] J. A. Hartigan and P. M. Hartigan. The dip test of unimodality. *The Annals of Statistics*, 13(1):70–84, 1985.

[50] D. Hatch and T. Stauffer. Operators on alert: operator response, alarm standards, protection layers keys to safe plants. *In Tech*, 56(9):14–16, 2009.

[51] A. Henningsen and J. P. Kemmerer. Intelligent alarm handling in cement plants. *IEEE Industry Application Magazine*, 1(5):9–15, 1995.

[52] S. Hofmeyr, S. Forrest, and A. Somayaji. Intrusion detection using sequences of system calls. *Journal of Computer Security*, 6(3):151–180, 1998.

[53] S.M. Hoppe, G.R. Essenberg, D.A. Wiegmann, and T.J. Overbye. Three-dimensional displays as an effective visualization technique for power systems monitoring and control. Technical report, University of Illinois, Urbana-Champaign, IL, 2004.

[54] E. Hostalkova and P. Stluka. Alarm rationalization support via correlation analysis of alarm history. In *19th International Congress of Chemical and Process Engineering*, Prague, Czech Republic, 2010.

[55] A. J. Hugo. Estimation of alarm deadbands. In *Proceedings of IFAC Safeprocess 2009*, pages 663–667, Barcelona, Spain, 2009.

[56] I. A. Ibragimov. On the composition of unimodal distributions. *Theory of Probability and its Applications*, 1(2):255–260, 1956.

[57] A. Inselberg. *Parallel Coordinates: VISUAL Multidimensional Geometry and Its Applications*. Springer, 2009.

[58] ISA. *Management of Alarm Systems for the Process Industries*. ANSI/ISA-18.2-2009. The International Society of Automation, Research Triangle Park, 2nd edition, 2009.

[59] J. D. Isom and R. E. LaBarre. Process fault detection, isolation, and reconstruction by principal component pursuit. In *2011 American Control Conference*, number 7, pages 238–243, San Francisco, CA, USA, 2011.

[60] I. Izadi, S. L. Shah, D. S. Shook, and T. Chen. An introduction to alarm analysis and design. In *Proceedings of IFAC Safeprocess 2009*, pages 645–650, Barcelona, Spain, 2009.

[61] I. Izadi, S. L. Shah, D. S. Shook, S. R. Kondaveeti, and T. Chen. A framework for optimal design of alarm systems. In *Proceedings of IFAC Safeprocess 2009*, pages 651–656, Barcelona, Spain, 2009.

[62] I. Izadi, S.L. Shah, and T. Chen. Effective resource utilization for alarm management. In *Proceedings of 49th IEEE Conference on Decision and Control*, pages 6803–6808, Atlanta, USA, 2010.

[63] M.C. Johannesmeyer. Abnormal situation analysis using pattern recognition techniques and historical data. Master's thesis, University of California, Santa Barbara, CA, 1999.

[64] D.S. Kirschen and B.F. Wollenberg. Intelligent alarm processing in power systems. *Proceedings of the IEEE*, 80(5):663–672, 1992.

[65] S.R. Kondaveeti. *Advanced Analysis and Redesign of Industrial Alarm Systems*. PhD thesis, University of Alberta, Edmonton, AB, 2013.

[66] S.R. Kondaveeti, S.L. Shah, and I. Izadi. Application of multivariate statistics for efficient alarm generation. In *Proceedings on IFAC Safeprocess 2009*, pages 657–662, Barcelona, Spain, June 2009.

[67] U. Kruger, J. Zhang, and L. Xie. Developments and applications of nonlinear principal component analysis - a review. *Lecture Notes in Computational Science and Engineering*, 58:1–43, 2008.

[68] W. Ku, R.H. Storer, and C. Georgakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30(1):179–196, 1995.

[69] T. Lane and C. Brodley. Temporal sequence learning and data reduction for anomaly detection. *ACM Transections on Information Systems and Security*, 2(3):295–331, 1999.

[70] J.E. Larsson. Simple methods for alarm sanitation. In *Proceedings of IFAC Symposium on Artificial Intelligence in Real-Time Control*, pages 1–6, Budapest, Hungary, Oct. 2000.

[71] Y. H. Lee and S. A. Kassam. Generalized median filtering and related nonlinear filtering techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(3):672–683, 1985.

[72] E. L. Lehmann and J. P. Romano. *Testing Statistical Hypotheses*. Springer, 3rd edition, 2005.

[73] V. Levenshtein. Binary codes capable of correcting spurious insertions and deletions of ones. *Problems of Inofrmation Transmission*, 1:8–17, 1965.

[74] X. Li, J. Wang, B. Huang, and S. Lu. The DCT-based oscillation detection method for a single time series. *Journal of Process Control*, 20(5):609–617, 2010.

[75] D. Lipman and W. Pearson. Rapid and sensitive protein similarity searches. *Science*, 227(4693):1435–1441, 1990.

[76] O. Lipsky, B. Porat, E. Porat, B. Riva Shalom, and A. Tzur. String matching with up to k swaps and mismatches. *Information and Computation*, 208(9):1020–1030, 2010.

[77] J. Liu, K.W. Lim, W.K. Ho, K.C. Tan, R. Srinivasan, and A. Tay. The intelligent alarm management system. *IEEE Software*, 20(2):66–71, 2003.

[78] X. Liu, N. Masaru, and N. Hirokazu. Evaluation of plant alarm systems by behavior simulation using a virtual subject. *Computers and Chemical Engineering*, 34(3):374–386, 2010.

[79] N. Mabroukeh and C. Ezeife. A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys*, 43(3), 2010.

[80] J.F. MacGregor and T.J. Harris. The exponentially weighted moving variance. *Journal of Quality Technology*, 25(2):106–118, 1993.

[81] K. Min, Z. Zhang, J. Wright, and Y. Ma. Decomposing background topics from keywords by principal component pursuit. In *the 19th ACM International Conference on Information and Knowledge Management*, pages 269–277, New York, NY, USA, 2010.

[82] E. Naghoosi, I. Izadi, and T. Chen. Estimation of alarm chattering. *Journal of Process Control*, 21(9):1243–1249, 2011.

[83] G. Navarro and M. Raffinot. *Flexible Pattern Matching in Strings*. Cambridge University Press, 2002.

[84] S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[85] J. Nishiguchi and T. Takai. IPL2 and 3 performance improvement method for process safety using event correlation analysis. *Computers and Chemical Engineering*, 34(12):2007–2013, 2010.

[86] J. Nishiguchi and H. Tsutsui. A new approach to process alarm reduction using statistical point processes. In *SICE Annual Conference*, pages 443–448, Okayama, Japan, 2005.

[87] L. O'Brien. A good alarm management strategy. *Hydrocarbon Processing*, page 13, March 2009.

[88] E.S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

[89] F. Proschan and R. Pyke. Tests for monotone failure rate. In *Proceedings of Fifth Berkeley Symp. on Math. Statist. and Prob.*, pages 293–312, Berkeley, CA, USA, 1967.

[90] S. J. Qin. Statistical process monitoring: Basics and beyond. *Journal of Chemometrics*, 17(8-9):480–502, 2003.

[91] A. B. Ramirez, H. Arguello, and G. Arce. Video anonaly recovery from compressed spectral imaging. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1321–1324, Santander, Bucaramanga, Colombia, 2011.

[92] C. Ratanamahatana and E. Keogh. Three myths about dynamic time warping. In *Proceedings of SIAM International Conference on Data Mining*, pages 506–510, Newport Beach, CA, USA, 2005.

[93] D.V.C. Reising, J.L. Downs, and D. Bay. Human performance models for response to alarm notifications in the process industries: An industrial case study. In *Proceedings of Annual Meeting of the Human Factors and Ergonomics Society*, volume 5, pages 1189–1193, New Orleans, LA, USA, Sept. 2004.

[94] P.B. Robinson and T.Y. Ho. Average run lengths of geometric moving average charts by numerical methods. *Technometrics*, 20(1):85–93, 1987.

[95] G. Shorack and J. Wellner. *Empirical Processes with Applications to Statisitcs*. John Wiley Sons, 1986.

[96] A. Singhal. *Pattern Matching in Multivariate Time-Series Data*. PhD thesis, University of California, Santa Barbara, CA, 2002.

[97] A. Singhal and D.E. Seborg. Matching pattern from historical data using pca and distance similarity factors. In *Proceedings of American Control Conference*, pages 1759–1764, Arlington, VA, 2001.

[98] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[99] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *5th International Conference on Extending Database Technology*, pages 3–17, Avignon, France, Mar. 1996.

[100] M.S. Srivastava and Y. Wu. Evaluation of optimum weights and average run lengths in EWMA control schemes. *Communications in Statistics-Theory and Methods*, 26(5):1253–1267, 1997.

[101] T. Stauffer. Implement an effective alarm management program. *CEP Magazine*, pages 19–27, July 2012.

[102] R. Storn and K. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[103] J. Taisne. Intelligent alarm processor based on chronicle recognition for transmission and distribution system. In *Proceedings on IEEE PES Power Systems Conference and Exposition*, pages 1606–1611, Atlanta, GA, USA, Oct 2006.

[104] T. Umeda, T. Kuriyama, E. O'Shima, and H. Matsuyama. A graphical approach to cause and effect analysis of chemical processing systems. *Chemical Engineering Science*, 35(12):2379–2388, 1980.

[105] T. Varga, F. Szeifert, and J. Abonyi. Detection of safe operating regions: a novel dynamic process simulator based predictive alarm management approach. *Industrial Engineering Chemical Research*, 49:658–668, 2010.

[106] G. Verdier, H. Hilgert, and J.P. Vila. Adaptive threshold computation for CUSUM-type procedures in change detection and isolation problems. *Computiational Statistics and Data Analysis*, 52(9):4161–4174, 2008.

[107] J. Wang, B. Huang, and S. Lu. Improved DCT-based method for online detection of oscillations in univariate time series. *Control Engineering Practice*, 21(5):622–630, 2013.

[108] X. Wang. A compositive method of neural networks and control charts for monitoring process disturbance based on integrated SPC/EPC. In *Proceedings of CiSE*, pages 1–4, Wuhan, China, Dec. 2009.

[109] W. Weirauch. Networking, alarm management, security among top initiatives. *Hydrocarbon Processing*, page 17, March 2009.

[110] J. Wen, X. Xiao, J. Dong, Z. Chen, and X. Dai. Data normalization for diabetes II metabonomics analysis. In *The 1st International Conference on Bioinformatics and Biomedical Engineering*, pages 682–685, Wuhan, China, 2007.

[111] B.F. Wollenberg. A knowledge-based alarm processor for an energy managemnt system. *IEEE Transactions on Power System*, 1(2):241–247, 1986.

[112] K.P. Wong. Artificial intelligence and neural network applications in power systems. In *Proceedings on IEE Internation Conference on Advances in Power System Control, Operation and Management*, pages 37–46, Hong Kong, Dec. 1993.

[113] J. Wright, A. Yang, A. Ganesh, Y. Ma, and S. Sastry. Robust face recognition via spares representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[114] H. Xu, C. Caramanis, and S. Sanghavi. Robust PCA via outlier pursuit. *IEEE Transactions on Information Theory*. in press.

[115] J. Xu, J. Wang, I. Izadi, and T. Chen. Performance assessment and design for univariate alarm systems based on FAR, MAR and ADD. *IEEE Transactions on Automation Science and Engineering*, 9(2):296–307, 2011.

[116] K. Yamanishi and Y. Maruyama. Dynamic syslog mining for network failure monitoring. In *Proceedings of 11th ACM SIGKDD*, pages 499–508, Chicago, IL, USA, 2005.

[117] F. Yang, S.L. Shah, D. Xiao, and T. Chen. Improved correlation analysis and visualization for industrial alarm data. In *18th IFAC World Congress*, pages 12898–12903, Milano, Italy, August 2011.

[118] Q. Zhao and M. Kinnaert. Statistical properties of CUSUM based fault detection schemes for fault tolerant control. In *Proceedings of 48th IEEE Conference on Decision and Control*, pages 7831–7836, Shanghai, China, Dec. 2009.

[119] Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma. Stable principal component pursuit. In *2010 IEEE International Symposim on Information Theory Proceedings*, pages 1518–1522, Austin, Texas, USA, 2010.

Appendix A

Proofs of Lemmas and the Main Result in Chapter 4

A.1 Proof of Lemma 4.1

Necessity: Let $\mathbf{E}_p = [\ \mathbf{E}_i \quad \mathbf{E}_j\ ]$. We have

$$\mathbf{E}_p^{-1} = \mathbf{E}_p^T.$$

Since

$$\mathbf{E}_i^T \mathbf{V}\mathbf{V}^T \mathbf{E}_j = \mathbf{0},$$

we obtain that

$$\mathbf{E}_p^T \mathbf{V}\mathbf{V}^T \mathbf{E}_p = \begin{bmatrix} \mathbf{C}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_j \end{bmatrix},$$

where

$$\mathbf{C}_i = \mathbf{E}_i^T \mathbf{V}\mathbf{V}^T \mathbf{E}_i,\ \mathbf{C}_j = \mathbf{E}_j^T \mathbf{V}\mathbf{V}^T \mathbf{E}_j.$$

As a result, there exists a unitary matrix $\mathbf{T}$ such that

$$\mathbf{E}_p^T \mathbf{V} = \begin{bmatrix} \mathbf{V}_{ii} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{jj} \end{bmatrix} \mathbf{T}, \tag{A.1}$$

where

$$\mathbf{V}_{ii}\mathbf{V}_{ii}^T = \mathbf{C}_i,\ \mathbf{V}_{jj}\mathbf{V}_{jj}^T = \mathbf{C}_j,$$
$$\mathbf{V}_{ii}^T\mathbf{V}_{ii} = \mathbf{I},\ \mathbf{V}_{jj}^T\mathbf{V}_{jj} = \mathbf{I}.$$

Substituting (A.1) and (4.9) into (4.12) yields

$$\begin{aligned}
&\mathbf{E}_i^T \mathbf{V}_s \mathbf{V}_s^T \mathbf{E}_j \\
&= \mathbf{E}_i^T \mathbf{D}(\mathbf{x}) \mathbf{E}_p \mathbf{E}_p^T \mathbf{V} (\mathbf{V}^T \mathbf{E}_p \mathbf{E}_p^T \mathbf{D}(\mathbf{x})^2 \mathbf{E}_p \mathbf{E}_p^T \mathbf{V})^{-1} \mathbf{V}^T \mathbf{E}_p \mathbf{E}_p^T \mathbf{D}(\mathbf{x}) \mathbf{E}_j \\
&= [\ \mathbf{E}_i^T \mathbf{D}(\mathbf{x}) \mathbf{E}_i \quad \mathbf{0}\ ] \\
&\cdot \begin{bmatrix} \mathbf{V}_{ii} (\mathbf{V}_{ii}^T \mathbf{E}_i^T \mathbf{D}(\mathbf{x})^2 \mathbf{E}_i \mathbf{V}_{ii})^{-1} \mathbf{V}_{ii}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{jj} (\mathbf{V}_{jj}^T \mathbf{E}_j^T \mathbf{D}(\mathbf{x})^2 \mathbf{E}_j \mathbf{V}_{jj})^{-1} \mathbf{V}_{jj}^T \end{bmatrix} \\
&\cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{E}_j^T \mathbf{D}(\mathbf{x}) \mathbf{E}_j \end{bmatrix} = \mathbf{0}
\end{aligned}$$

The proof of sufficiency is similar. We only need to swap $\mathbf{V}$ and $\mathbf{V}_s$, and change $\mathbf{D}(x)$ by $\mathbf{D}(x)^{-1}$ in the necessity proof. $\square$

A.2 Proof of Lemma 4.2

Since $J(\mathbf{y})$ is the maximum value among the n_2 diagonal entries of matrix $\mathbf{V}_s\mathbf{V}_s^T$, it can be proved that

$$\begin{aligned} n_2 J(\mathbf{y}) &\geq \text{trace}\left(\mathbf{V}_s\mathbf{V}_s^T\right) \\ &= \text{trace}\left(\mathbf{V}_s^T\mathbf{V}_s\right) \\ &= \text{trace}(\mathbf{I}_{r\times r}) = r. \end{aligned}$$

So that

$$J(\mathbf{y}) \geq \frac{r}{n_2}.$$

The equality holds only if all of the diagonal entries are the same.

$\mathbf{V}_{s\perp}$ denotes a normalized orthogonal basis of the null space of $\mathbf{V}_s$. Let $\mathbf{v}_s^i$ denote the i-th column of ${\mathbf{V}_\mathbf{s}}^T$, and $\mathbf{v}_{s\perp}^i$ denote the i-th column of ${\mathbf{V}_{\mathbf{s}\perp}}^T$. Since

$$\begin{bmatrix} \mathbf{V}_s & \mathbf{V}_{s\perp} \end{bmatrix} \begin{bmatrix} \mathbf{V}_s^T \\ \mathbf{V}_{s\perp}^T \end{bmatrix} = \mathbf{I},$$

we have

$$\mathbf{v}_s^{iT}\mathbf{v}_s^i = 1 - {\mathbf{v}_{s\perp}^i}^T\mathbf{v}_{s\perp}^i \leq 1,$$

for $i = 1, \cdots, n_2$. Since $\mathbf{v}_s^{iT}\mathbf{v}_s^i$ is the i-th diagonal entry of matrix $\mathbf{V}_s\mathbf{V}_s^T$, all of the diagonal entries of this matrix are less than or equal to 1. As a result $J(\mathbf{y}) \leq 1$. The equality holds only if there exists an index o such that $\mathbf{v}_{s\perp}^o = \mathbf{0}$. Thus

$$\mathbf{v}_s^{oT}\mathbf{v}_s^i + {\mathbf{v}_{s\perp}^o}^T\mathbf{v}_{s\perp}^i = \mathbf{v}_s^{oT}\mathbf{v}_s^i = 0,$$

for $i = 1, \cdots, n_2,\ i \neq o$, which means that

$$\mathbf{e}_o^T\mathbf{V}_s\mathbf{V}_s^T\begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{o-1} & \mathbf{e}_{o+1} & \cdots & \mathbf{e}_{n2} \end{bmatrix} = \mathbf{0}.$$

According to Lemma 1, this equation contradicts Assumption 1. Thus there is no such index o. In other words, $J(\mathbf{y}) < 1$. $\square$

A.3 Proof of Lemma 4.3

1) Since in step 2 to step 6 of Algorithm 2, only the j-th element of $\widehat{\mathbf{y}}$ changes, $\mathbf{A}(\widehat{\mathbf{y}}_0) = \mathbf{A}(\widehat{\mathbf{y}}_1)$, $\mathbf{B}(\widehat{\mathbf{y}}_0) = \mathbf{B}(\widehat{\mathbf{y}}_1)$, and $q(\widehat{\mathbf{y}}_0) = q(\widehat{\mathbf{y}}_1)$. As a result, we can use

the symbols q, $\mathbf{A}$, $\mathbf{B}$, Γ, and γ_t in steps 3 and 4 of Algorithm 2 without any confusion. Consider the following equation with respect to any vector $\mathbf{y} > 0$:

$$\begin{aligned}
\begin{bmatrix} d_j(\mathbf{y}) \\ d_1(\mathbf{y}) \\ \vdots \\ d_{j-1}(\mathbf{y}) \\ d_{j+1}(\mathbf{y}) \\ \vdots \\ d_{n_2}(\mathbf{y}) \end{bmatrix} &= \operatorname{diag}\left(\begin{bmatrix} \mathbf{e}_j^T \\ \mathbf{e}_1^T \\ \vdots \\ \mathbf{e}_{j-1}^T \\ \mathbf{e}_{j+1}^T \\ \vdots \\ \mathbf{e}_{n_2}^T \end{bmatrix} \mathbf{V}(\mathbf{V}^T\mathbf{D}(\mathbf{y})\mathbf{V})^{-1}\mathbf{V}^T\mathbf{D}(\mathbf{y}) \begin{bmatrix} \mathbf{e}_j^T \\ \mathbf{e}_1^T \\ \vdots \\ \mathbf{e}_{j-1}^T \\ \mathbf{e}_{j+1}^T \\ \vdots \\ \mathbf{e}_{n_2}^T \end{bmatrix}^T \right) \\
&= \operatorname{diag}\left(\begin{bmatrix} \mathbf{v}_j^T \\ \mathbf{V}_{\setminus j} \end{bmatrix} (\mathbf{v}_j y_j \mathbf{v}_j^T + \mathbf{V}_{\setminus j}^T \mathbf{D}(\mathbf{y}_{\setminus j})\mathbf{V}_{\setminus j})^{-1} [\ \mathbf{v}_j y_j \quad \mathbf{V}_{\setminus j}^T \mathbf{D}(\mathbf{y}_{\setminus j})\ ] \right) \\
&= \operatorname{diag}\left(\begin{bmatrix} \mathbf{v}_j^T \\ \mathbf{V}_{\setminus j} \end{bmatrix} (\mathbf{v}_j y_j \mathbf{v}_j^T + \mathbf{Q}_j(\mathbf{y}))^{-1} [\ \mathbf{v}_j y_j \quad \mathbf{V}_{\setminus j}^T \mathbf{D}(\mathbf{y}_{\setminus j})\ ] \right)
\end{aligned} \tag{A.2}$$

Since only the j-th element changes in one iteration, we only need to consider the change of $d_i(\mathbf{y})$s with respect to $\widehat{y}_j$. So, the $d_i(\mathbf{y})$s can be denoted as $d_i(\widehat{y}_j)$s without confusion.

If the matrix $\mathbf{V}_{\setminus j}$ is not full column rank, there exists a non-zero column vector $\mathbf{w}$ that $\mathbf{V}_{\setminus j}\mathbf{w} = \mathbf{0}$. Since

$$[\ \mathbf{v}_j \quad \mathbf{V}_{\setminus j}^T\ ] \begin{bmatrix} \mathbf{v}_j^T \\ \mathbf{V}_{\setminus j} \end{bmatrix} = \mathbf{I},$$

we have

$$\mathbf{v}_j(\mathbf{v}_j^T\mathbf{w}) = \mathbf{w}.$$

So $\mathbf{V}_{\setminus j}\mathbf{v}_j = \mathbf{0}$, which means that Assumption 1 is violated. As a result, the matrix $\mathbf{V}_{\setminus j}$ must be full column rank. Then, matrix $\mathbf{Q}_j(\mathbf{y})$ is invertible. The vector $\mathbf{v}_j$ must be non-zero because of the same reason. Thus, we can apply the matrix inversion lemma in (A.2) to get $d_i(\widehat{y}_j)$s.

$$\begin{aligned}
\begin{bmatrix} d_j(\widehat{y}_j) \\ d_1(\widehat{y}_j) \\ \vdots \\ d_{j-1}(\widehat{y}_j) \\ d_{j+1}(\widehat{y}_j) \\ \vdots \\ d_{n_2}(\widehat{y}_j) \end{bmatrix} &= \operatorname{diag}\left(\begin{bmatrix} \mathbf{v}_j^T \\ \mathbf{V}_{\setminus \mathbf{j}} \end{bmatrix} (\mathbf{Q}_j(\widehat{\mathbf{y}}) + \mathbf{v}_j \widehat{y}_j \mathbf{v}_j^T)^{-1} \right. \\
&\qquad \left. \cdot \begin{bmatrix} \mathbf{v}_j \widehat{y}_j \quad \mathbf{V}_{\setminus j}^T \mathbf{D}(\widehat{\mathbf{y}}_{\setminus j})\ \end{bmatrix} \right) \\
&= \operatorname{diag}\left(\begin{bmatrix} \frac{q\widehat{y}_j}{q\widehat{y}_j+1} & * \\ * & \mathbf{A} - \frac{\widehat{y}_j}{q\widehat{y}_j+1}\mathbf{B} \end{bmatrix} \right);
\end{aligned} \tag{A.3}$$

Since $\mathbf{v}_i \neq \mathbf{0}$ for $i = 1, 2, \cdots, n_2$, we have $q > 0$, $\operatorname{diag}(\mathbf{A}) > \mathbf{0}$, and $\operatorname{diag}(\mathbf{B}) \geq \mathbf{0}$. It can be proved that $d_j(\widehat{y}_j)$ is a strictly monotonic increasing

function in its domain $\widehat{y}_j \in (0, +\infty)$; and $d_i(\widehat{y}_j)$ is a monotonic decreasing functions in its domain $\widehat{y}_j \in (0, +\infty)$ for $i = 1, \cdots, j-1, j+1, \cdots, n_2$. Moreover, for any $i = 1, \cdots, j-1$, the solution of $d_j(\widehat{y}_j) = d_i(\widehat{y}_j)$, namely,

$$\frac{q\widehat{y}_j}{q\widehat{y}_j + 1} = \mathbf{e}_i^T \left(\mathbf{A} - \frac{\widehat{y}_j}{q\widehat{y}_j + 1} \mathbf{B} \right) \mathbf{e}_i,$$

is

$$\widehat{y}_j = \frac{\mathbf{e}_i^T \mathbf{A} \mathbf{e}_i}{\mathbf{e}_i^T \mathbf{B} \mathbf{e}_i + q - \mathbf{e}_i^T \mathbf{A} \mathbf{e}_i q} = \gamma_i.$$

For any $i = j+1, \cdots, n_2$, the solution of $d_j(\widehat{y}_j) = d_i(\widehat{y}_j)$, namely,

$$\frac{q\widehat{y}_j}{q\widehat{y}_j + 1} = \mathbf{e}_{i-1}^T \left(\mathbf{A} - \frac{\widehat{y}_j}{q\widehat{y}_j + 1} \mathbf{B} \right) \mathbf{e}_{i-1},$$

is

$$\widehat{y}_j = \frac{\mathbf{e}_{i-1}^T \mathbf{A} \mathbf{e}_{i-1}}{\mathbf{e}_{i-1}^T \mathbf{B} \mathbf{e}_{i-1} + q - \mathbf{e}_{i-1}^T \mathbf{A} \mathbf{e}_{i-1} q} = \gamma_{i-1}.$$

Since $\mathbf{V}_{\setminus j}$ is full column rank, we can prove that $\mathrm{diag}(\mathbf{A}) \leq \mathbf{1}$ by following the same line in the proof of Lemma 2. Moreover, if there exists an index o that $\mathbf{e}_o^T \mathbf{A} \mathbf{e}_o = 1$, then $\mathbf{e}_o^T \mathbf{B} \mathbf{e}_o \neq 0$. Otherwise $d_o(\widehat{y}_j) = 1$ for $\widehat{y}_{0j}$. Thus $J(\widehat{\mathbf{y}}_0) = 1$, which contradicts Lemma 2 stating that $J(\mathbf{y}) < 1$. As a result, $\gamma_t \in (0, +\infty)$ for $t = 1, \cdots, n_2 - 1$. Hence, $0 < \Gamma < +\infty$.

If $\max\limits_{1 \leq i \leq n_2} (d_i(\widehat{y}_{0j})) \neq d_j(\widehat{y}_{0j})$, there exists an index $o \neq j$ that $d_o(\widehat{y}_{0j}) > d_j(\widehat{y}_{0j})$. Thus $\gamma_o > \widehat{y}_{0j}$ if $o < j$, or $\gamma_{o-1} > \widehat{y}_{0j}$ if $o > j$. As a result,

$$\max\{\Gamma, \widehat{y}_{0j}\} = \Gamma > \widehat{y}_{0j}.$$

Therefore,

$$\widehat{y}_{0j} < \frac{\Gamma}{2} + \frac{\widehat{y}_{0j}}{2} = \widehat{y}_{1j} < \frac{\Gamma}{2} + \frac{\Gamma}{2} = \Gamma.$$

On the other hand, if $\max\limits_{1 \leq i \leq n_2} (d_i(\widehat{y}_{0j})) = d_j(\widehat{y}_{0j})$, $\gamma_t \leq \widehat{y}_{0j}$ for $t = 1, \cdots, n_2 - 1$. Thus, $\max\{\Gamma, \widehat{y}_{0j}\} = \widehat{y}_{0j}$. Therefore,

$$\widehat{y}_{1j} = (\max\{\Gamma, \widehat{y}_{0j}\} + \widehat{y}_{0j})/2 = \widehat{y}_{0j}.$$

The proof of property 1) is completed. Moreover, the updated value $\widehat{y}_{1j}$ is in the range $(0, +\infty)$ as long as $\widehat{y}_{0j} \in (0, +\infty)$.

2) Since $\widehat{y}_{1j} \geq \widehat{y}_{0j}$, and $d_i(\widehat{y}_j)$ is a monotonic decreasing function for $i = 1, \cdots, j-1, j+1, \cdots, n_2$, $d_i(\widehat{y}_{0j}) \geq d_i(\widehat{y}_{1j})$ for $i = 1, \cdots, j-1, j+1, \cdots, n_2$. The proof of property 2) is completed.

3) If $\max_{1\le i\le n_2}(d_i(\widehat{y}_{0j})) \neq d_j(\widehat{y}_{0j})$, then $\widehat{y}_{1j} < \Gamma$. Hence, there exists an index $o \neq j$ such that the solution of equation $d_o(\widehat{y}_j) = d_j(\widehat{y}_j)$ is greater than $\widehat{y}_{1j}$. Since $d_j(\widehat{y}_j)$ is monotonically increasing and $d_o(\widehat{y}_j)$ is monotonically decreasing, $d_o(\widehat{y}_{1j}) > d_j(\widehat{y}_{1j})$. From property 1), we have $d_o(\widehat{y}_{0j}) \ge d_o(\widehat{y}_{1j})$, which implies $d_o(\widehat{y}_{0j}) > d_j(\widehat{y}_{1j})$. Consequently,

$$\max_{1\le i\le n_2} d_i(\widehat{y}_{0j}) > d_j(\widehat{y}_{1j}).$$

On the other hand, if $\max_{1\le i\le n_2}(d_i(\widehat{y}_{0j})) = d_j(\widehat{y}_{0j})$, $\widehat{y}_{0j} = \widehat{y}_{1j}$. As a result,

$$\max_{1\le i\le n_2} d_i(\widehat{y}_{0j}) = d_j(\widehat{y}_{0j}) = d_j(\widehat{y}_{1j}).$$

The proof of property 3) is completed.

4) It follows from property 2) that

$$d_i(\widehat{y}_{1j}) \le d_i(\widehat{y}_{0j}) \le \max_{1\le t\le n_2}(d_t(\widehat{y}_{0j})),$$

for $i = 1, \cdots, j-1, j+1, \cdots, n_2$. From property 3), we have,

$$d_j(\widehat{y}_{1j}) \le \max_{1\le t\le n_2}(d_t(\widehat{y}_{0j})).$$

As a result,

$$\max_{1\le t\le n_2}(d_t(\widehat{y}_{0j})) \ge \max_{1\le i\le n_2}(d_i(\widehat{y}_{1j})).$$

Next, we show that $\max_{1\le t\le n_2}(d_t(\widehat{y}_{0j})) = \max_{1\le i\le n_2}(d_i(\widehat{y}_{1j}))$ if and only if there exists an index o such that

$$d_o(\widehat{y}_{1j}) = d_o(\widehat{y}_{0j}) = \max_{1\le t\le n_2}(d_t(\widehat{y}_{0j})). \tag{A.4}$$

$\max_{1\le t\le n_2}(d_t(\widehat{y}_{0j})) = \max_{1\le i\le n_2}(d_i(\widehat{y}_{1j}))$ if and only if there exists an index w such that

$$d_w(\widehat{y}_{1j}) = \max_{1\le i\le n_2}(d_i(\widehat{y}_{1j})) = \max_{1\le t\le n_2}(d_t(\widehat{y}_{0j})). \tag{A.5}$$

Case 1: $d_w(\widehat{y}_{1j}) = d_j(\widehat{y}_{1j})$. Then, equation (A.5) becomes

$$d_j(\widehat{y}_{1j}) = \max_{1\le i\le n_2}(d_i(\widehat{y}_{1j})) = \max_{1\le t\le n_2}(d_t(\widehat{y}_{0j})). \tag{A.6}$$

According to property 3), equation (A.6) holds only if

$$d_j(\widehat{y}_{0j}) = d_j(\widehat{y}_{1j}) = \max_{1\le t\le n_2}(d_t(\widehat{y}_{0j})), \tag{A.7}$$

which is further equivalent to $\widehat{y}_{1j} = \widehat{y}_{0j}$ by using property 1). So that equation (A.6) holds if and only if equation (A.7) holds.

Case 2: $d_w(\widehat{y}_{1j}) \neq d_j(\widehat{y}_{1j})$. Assume that equation (A.5) holds. Then, it follows from property 2) that

$$d_w(\widehat{y}_{0j}) \geq d_w(\widehat{y}_{1j}) = \max_{1 \leq t \leq n_2} (d_t(\widehat{y}_{0j})) \geq d_w(\widehat{y}_{0j}),$$

which implies that

$$d_w(\widehat{y}_{1j}) = d_w(\widehat{y}_{0j}) = \max_{1 \leq t \leq n_2} (d_t(\widehat{y}_{0j})). \tag{A.8}$$

Conversely, if equation (A.8) holds, then according to property 3) we have

$$d_w(\widehat{y}_{1j}) = \max_{1 \leq t \leq n_2} (d_t(\widehat{y}_{0j})) \geq d_j(\widehat{y}_{1j});$$

and according to property 2) we have

$$d_w(\widehat{y}_{1j}) = \max_{1 \leq t \leq n_2} (d_t(\widehat{y}_{0j})) \geq d_i(\widehat{y}_{0j}) \geq d_i(\widehat{y}_{1j})$$

for all $i \neq j$.

Therefore,

$$d_w(\widehat{y}_{1j}) = \max_{1 \leq i \leq n_2} (d_i(\widehat{y}_{1j})),$$

and thus equation (A.5) holds. The proof of property 4) is completed. □

A.4 Proof of Lemma 4.4

According to property 4) of Lemma 3, each inner iteration does not increase the objective function. The scaling step, step 7 of Algorithm 2, does not affect the objective function, i.e., $d_j(\widetilde{\mathbf{y}}) = d_j(\mathbf{y}_{k+1})$ for $j = 1, \cdots, n_2$. Therefore, $J(\mathbf{y}_k) \geq J(\mathbf{y}_{k+1})$. If $J(\mathbf{y}_k) = J(\mathbf{y}_{k+1})$, in each inner iteration $J(\widehat{\mathbf{y}}_0) = J(\widehat{\mathbf{y}}_1) = J(\mathbf{y}_k)$. There are two cases in which this situation may occur:
Case 1: The set $O = \{o | d_o(\mathbf{y}_k) = J(\mathbf{y}_k)\} = \{1, \cdots, n_2\}$. This is the case that the global lower bound $\frac{r}{n}$ has been reached.
Case 2: The set $O = \{o | d_o(\mathbf{y}_k) = J(\mathbf{y}_k)\} \subset \{1, \cdots, n_2\}$. In this case we define two other index sets: $O_\perp = \{1, \cdots, n_2\} \backslash O$, and $O_m = \{o | d_o(\mathbf{y}_{k+1}) = J(\mathbf{y}_k)\}$. For any index $j \in O_\perp$, function $d_j(\widehat{\mathbf{y}})$ may increase only in the j-th inner iteration based on property 2) of Lemma 3. So in the j-th inner iteration, $d_j(\widehat{\mathbf{y}}_0) \leq d_j(\mathbf{y}_k) < J(\mathbf{y}_k) = J(\widehat{\mathbf{y}}_0)$. According to property 3) of Lemma 3,

$d_j(\widehat{\mathbf{y}}_1) < J(\widehat{\mathbf{y}}_0) = J(\mathbf{y}_k)$. According to property 1) of Lemma 3, $\widehat{y}_{0j} < \widehat{y}_{1j}$. Again, because of property 2) of Lemma 3, after the j-th inner iteration, $d_j(\widehat{\mathbf{y}})$ does not increase, so $d_j(\mathbf{y}_{k+1}) < J(\mathbf{y}_k)$. As a result, $O_\perp \bigcap O_m = \emptyset$, and $\mathbf{e}_j^T\mathbf{y}_k < \mathbf{e}_j^T\widetilde{\mathbf{y}}$ for all $j \in O_\perp$. Therefore, $O_m \subseteq O$.

If $O_m = O$, for any $i \in O$, in the i-th inner iteration $d_i(\widehat{\mathbf{y}}_0) = J(\mathbf{y}_k)$, hence $\widehat{y}_{i0} = \widehat{y}_{i1}$. Therefore, $\mathbf{e}_i^T\mathbf{y}_k = \mathbf{e}_i^T\widetilde{\mathbf{y}}$ for all $i \in O$. Make a partition on the indices $\{1, \cdots, n_2\}$ into O and $O_\perp$. We have matrices $\mathbf{E}_O$, and $\mathbf{E}_{O_\perp}$ whose columns are the standard basis vectors corresponding to the index sets O and $O_\perp$, respectively. Let

$$\begin{array}{l}\mathbf{V}_O = \mathbf{E}_O^T\mathbf{V},\ \mathbf{V}_{O_\perp} = \mathbf{E}_{O_\perp}^T\mathbf{V},\\ \mathbf{D}_O(\mathbf{y}) = \mathbf{E}_O^T\mathbf{D}(\mathbf{y})\mathbf{E}_O,\\ \mathbf{D}_{O_\perp}(\mathbf{y}) = \mathbf{E}_{O_\perp}^T\mathbf{D}(\mathbf{y})\mathbf{E}_{O_\perp},\\ \mathbf{R}(\mathbf{y}) = \mathbf{V}^T\mathbf{D}(\mathbf{y})\mathbf{V}.\end{array}$$

It can be proved that

$$\begin{array}{ll}\mathbf{D}_O(\widetilde{\mathbf{y}} - \mathbf{y}_k) = \mathbf{0}, & \mathbf{D}_O(\widetilde{\mathbf{y}}) = \mathbf{D}_O(\mathbf{y}_k),\\ \mathbf{D}_{O_\perp}(\widetilde{\mathbf{y}} - \mathbf{y}_k) > \mathbf{0}.\end{array}$$

So,

$$\begin{aligned}&\mathbf{E}_o^T\begin{bmatrix} d_1(\widetilde{\mathbf{y}}) \\ \vdots \\ d_{n_2}(\widetilde{\mathbf{y}}) \end{bmatrix} = \operatorname{diag}\left(\mathbf{V}_O\mathbf{R}(\widetilde{\mathbf{y}})^{-1}\mathbf{V}_O^T\mathbf{D}_O(\widetilde{\mathbf{y}})\right)\\ &= \operatorname{diag}\Big(\mathbf{V}_O\left(\mathbf{R}(\mathbf{y}_k) + \mathbf{V}_{O_\perp}^T\mathbf{D}_{O_\perp}(\widetilde{\mathbf{y}} - \mathbf{y}_k)\mathbf{V}_{O_\perp}\right)^{-1}\mathbf{V}_O^T\mathbf{D}_O(\mathbf{y}_k)\Big)\\ &= \mathbf{E}_o^T\begin{bmatrix} d_1(\mathbf{y}_k) \\ \vdots \\ d_{n_2}(\mathbf{y}_k) \end{bmatrix} + \operatorname{diag}\Big(\mathbf{V}_O\mathbf{R}(\mathbf{y}_k)^{-1}\mathbf{V}_{O_\perp}^T\left(\mathbf{V}_{O_\perp}\mathbf{R}(\mathbf{y}_k)^{-1}\mathbf{V}_{O_\perp}^T\right.\\ &\quad \left.+\mathbf{D}_{O_\perp}(\widetilde{\mathbf{y}} - \mathbf{y}_k)^{-1}\right)^{-1}\mathbf{V}_{O_\perp}\mathbf{R}(\mathbf{y}_k)^{-1}\mathbf{V}_O\mathbf{D}_O(\mathbf{y}_k)\Big).\end{aligned} \tag{A.9}$$

According to Lemma 1 and Assumption 1,

$$\mathbf{V}_O\mathbf{R}(\mathbf{y}_k)^{-1}\mathbf{V}_{O_\perp}^T \neq \mathbf{0}.$$

Then

$$\mathbf{E}_o^T\begin{bmatrix} d_1(\widetilde{\mathbf{y}}) \\ \vdots \\ d_{n_2}(\widetilde{\mathbf{y}}) \end{bmatrix} \neq \mathbf{E}_o^T\begin{bmatrix} d_1(\mathbf{y}_k) \\ \vdots \\ d_{n_2}(\mathbf{y}_k) \end{bmatrix},$$

which contradicts that $\mathbf{e}_i\mathbf{y}_k = \mathbf{e}_i\widetilde{\mathbf{y}}$ for all $i \in O$. As a result, $O_m \subset O$.

In case 2, the set O_m in the k-th outer iteration is the set O in the $k+1$-th outer iteration. The size of O is at most $n_2 - 1$, and the size of O_m must be strictly smaller than the size of O. Therefore, case 2 can keep at most $n_2 - 2$ consecutive outer iterations. As a result, $J(\mathbf{y}_k) > J(\mathbf{y}_{k+n_2-1})$ unless $J(\mathbf{y}_k) = \dfrac{r}{n_2}$. □

A.5 Proof of Theorem 4.2

For convenience, we define the following domains first:

$$\begin{aligned}
\mathfrak{D}_o &= \{\mathbf{y} | \max_{1\le i\le n_2} \mathbf{e}_i^T\mathbf{y} = 1, \min_{1\le i\le n_2} \mathbf{e}_i^T\mathbf{y} > 0\};\\
\mathfrak{D} &= \{\mathbf{y} | \max_{1\le i\le n_2} \mathbf{e}_i^T\mathbf{y} = 1, \min_{1\le i\le n_2} \mathbf{e}_i^T\mathbf{y} \ge 0\};\\
\mathfrak{D}_1 &= \{\mathbf{y} | \max_{1\le i\le n_2} \mathbf{e}_i^T\mathbf{y} = 1, \min_{1\le i\le n_2} \mathbf{e}_i^T\mathbf{y} = 0\}.
\end{aligned}$$

$\mathfrak{D}_o$ is the original domain of our optimization problem. The objective function and Algorithms 1 and 2 are well defined in this domain. Lemmas 4.1 to 4.2 are proved and thus valid in this domain. However, it is not a compact set. $\mathfrak{D}$ is an extension of $\mathfrak{D}_o$ to its boundary, so that $\mathfrak{D}$ is a compact set. $\mathfrak{D}_1$ is the relative complement of $\mathfrak{D}_o$ in $\mathfrak{D}$.

Let function $\mathbf{g} : \mathfrak{D}_o \to \mathfrak{D}_o$ denote the combination of all operations done in one outer iteration. In other words, $\mathbf{y}_{k+1} = \mathbf{g}(\mathbf{y}_k)$. Hence, the reduction of the objective function in one outer iteration is defined as:

$$\Delta J(\mathbf{y}) = J(\mathbf{y}) - J(\mathbf{g}(\mathbf{y})).$$

For simplicity, $\mathbf{g}^{(t)}$ and $\Delta J^{(t)}(\mathbf{y})$ denote the operations done in t outer iterations and the reduction of the objective function in t outer iterations, respectively. In other words,

$$\mathbf{g}^{(t)}(\mathbf{y}_k) = \mathbf{y}_{k+t};$$

and

$$\Delta J^{(t)}(\mathbf{y}) = J(\mathbf{y}) - J(\mathbf{g}^{(t)}(\mathbf{y})).$$

In the domain $\mathfrak{D}_o$, all of the functions defined above are continuous. However, they may not have limits on the boundary, namely, $\mathfrak{D}_1$.

Consider case 1: there exists a scalar $\epsilon > 0$ such that for any finite index k, there exists a finite index $\widetilde{k}$ greater than k which satisfies:

$$\min(\mathbf{y}_{\widetilde{k}}) \ge \epsilon.$$

Then there must exist a subsequence $\{l_k\}_{k=0}^{\infty}$ of the sequence $\{0, 1, 2, \cdots\}$ with $\min(\mathbf{y}_{l_k}) \ge \epsilon$ for $k = 0, 1, 2, \cdots$. Define the following domain:

$$\mathfrak{G}^{\epsilon} = \{\mathbf{y} | \min(\mathbf{y}) \ge \epsilon, \max(\mathbf{y}) = 1\}.$$

It can be proved that $\mathfrak{G}^{\epsilon} \subset \mathfrak{D}_o$; and $\mathbf{y}_{l_k} \in \mathfrak{G}^{\epsilon}$ for $k = 0, 1, 2, \cdots$.

Assume that $J(\mathbf{y})$ does not converge to r/n_2. According to property 4) of Lemma 4.3, the sequence $\{J(\mathbf{y}_k)\}_{k=0}^{\infty}$ monotonically, yet may not strictly, decreases. So, there must exist a value $\beta > r/n_2$ such that $J(\mathbf{y}_k) \geq \beta$ for $k = 0, 1, 2, \cdots$. The set $\mathfrak{G}_\beta^\epsilon = \{\mathbf{y} | J(\mathbf{y}) \geq \beta, \mathbf{y} \in \mathfrak{G}^\epsilon\}$ is a compact set, and $\mathbf{y}_{l_k} \in \mathfrak{G}_\beta^\epsilon$ for $k = 0, 1, 2, \cdots$. Since the function $\Delta J^{(n_2)}(\mathbf{y})$ is continuous in $\mathfrak{G}_\beta^\epsilon$, the minimum of $\Delta J^{(n_2)}(\mathbf{y})$ in the compact set $\mathfrak{G}_\beta^\epsilon$ exists. According to Lemma 4.4, it must be greater than 0. Denote the minimum by δ, so that $\Delta J^{(t)}(\mathbf{y}) \geq \delta$ for $t \geq n_2$. Since $l_{k+n_2} - l_k \geq n_2$, $J(\mathbf{y}_{l_0}) - J(\mathbf{y}_{l_{\lceil n_2/\delta \rceil}}) \geq 1$, where $\lceil n_2/\delta \rceil$ is the smallest integer that is no less than n_2/δ. As a result, $J(\mathbf{y}_{l_{\lceil n_2/\delta \rceil}}) < J(\mathbf{y}_{l_0}) - 1 < 0$, which clearly contradicts the assumption that $J(\mathbf{y}_k) \geq \beta > r/n_2$ for $k = 0, 1, 2, \cdots$. As a result, the assumption is false, and the algorithm must converge to an global optimal point whose objective function values r/n_2.

Then, consider case 2 in which the sequence $\{\min(\mathbf{y}_k)\}_{k=0}^{\infty}$ converges to 0. In other words, $\{\mathbf{y}_k\}_{k=0}^{\infty}$ converges to a subset of $\mathfrak{D}_1$. We need to extend the definition of $J(\mathbf{y})$ in domain $\mathfrak{D}_1$ and that of $\Delta J^{(t)}(\mathbf{y})$ as well.

For a point in $\mathfrak{D}_1$, a set of indices B is defined by: $B = \{b | y_b = 0\}$, and the other indices belong to the set: $B_\perp = \{1, 2, \cdots, n_2\}/B$. Let $\mathbf{E}_{B_\perp}$ denote the matrix whose columns are the standard basis vectors corresponding to the index set $B_\perp$. We can further divide set B into two subset: B_o and B_r. B_o is the set of all indices $b \in B$ such that the b-th row of matrix $\mathbf{V}$ is in the row space of $\mathbf{E}_{B_\perp}^T \mathbf{V}$. Then, we can define $\mathbf{E}_{B_o}$ and $\mathbf{E}_{B_r}$ similarly. Denote the cardinalities of B_r, B_o and $B_\perp$ by n_{B_r}, n_{B_o} and $n_{B_\perp}$, respectively. Denote the rank of $\mathbf{E}_{B_\perp}^T \mathbf{V}$ by r_1 which is smaller than or equal to $\min(r, n_{B_\perp})$. Then, there exists an invertible matrix $\mathbf{T_B}$ such that

$$\begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{V}\mathbf{T_B} = \begin{bmatrix} \mathbf{V_{11}} & \mathbf{0} \\ \mathbf{V_{21}} & \mathbf{0} \\ \mathbf{V_{31}} & \mathbf{V_{32}} \end{bmatrix},$$

where $\mathbf{V_{11}} \in \mathbb{R}^{n_{B_\perp} \times r_1}$ and $\mathbf{V_{32}} \in \mathbb{R}^{n_{B_r} \times (r - r_1)}$ are full column rank matrices. The objective function of a boundary point along a certain direction $\boldsymbol{\lambda} = [\ \widetilde{\boldsymbol{\lambda}}^T \quad \widehat{\boldsymbol{\lambda}}^T\ ]^T = [\ \tilde{\lambda}_1 \quad \cdots \quad \tilde{\lambda}_{n_{B_o}} \quad \hat{\lambda}_1 \quad \cdots \quad \hat{\lambda}_{n_{B_r}}\ ]^T$ can be defined as:

$$J_B(\mathbf{y}|\boldsymbol{\lambda}) = \lim_{\alpha \to 0^+} J(\mathbf{y}_B(\alpha|\boldsymbol{\lambda})), \tag{A.10}$$

where

$$\mathbf{y}_B(\alpha|\boldsymbol{\lambda}) = \mathbf{y} + \alpha \mathbf{E}_{B_o} \widetilde{\boldsymbol{\lambda}} + \alpha \mathbf{E}_{B_r} \widehat{\boldsymbol{\lambda}},$$

for all direction vectors in the domain:

$$\mathfrak{L}_o = \{\boldsymbol{\lambda} | \max(\boldsymbol{\lambda}) < \infty, \min(\boldsymbol{\lambda}) \geq 0, \|\widehat{\boldsymbol{\lambda}}\|_0 \geq (r - r_1 + 1)\}.$$

Since $J(\mathbf{y}_B(\alpha|\lambda))$ is the maximum of n_2 bounded rational fractions with respect to α, its limit at 0^+ exists. However, the limits along different directions may be different. Similarly, we can define the function $\Delta J^{(t)}$ along direction $\boldsymbol{\lambda}$ on the boundary points:

$$\Delta J_B^{(t)}(\mathbf{y}|\boldsymbol{\lambda}) = \lim_{\alpha \to 0^+} \Delta J^{(t)}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda})). \tag{A.11}$$

It can be derived that

$$\begin{aligned} J(\mathbf{y}_B(\alpha|\boldsymbol{\lambda})) = \max \Bigg(& \operatorname{diag} \Bigg(\begin{bmatrix} \mathbf{V_{11}} & \mathbf{0} \\ \sqrt{\alpha}\mathbf{V_{21}} & \mathbf{0} \\ \sqrt{\alpha}\mathbf{V_{31}} & \mathbf{V_{32}} \end{bmatrix} \\ & \Bigg(\begin{bmatrix} \mathbf{V_{11}}^T \mathbf{D}(\mathbf{E}_{B_\perp}^T \mathbf{y}) \mathbf{V_{11}} & \mathbf{0} \\ \mathbf{0} & \mathbf{V_{32}}^T \mathbf{D}(\widehat{\boldsymbol{\lambda}}) \mathbf{V_{32}} \end{bmatrix} \\ & + \begin{bmatrix} \alpha \left(\mathbf{V_{21}}^T \mathbf{D}(\widetilde{\boldsymbol{\lambda}}) \mathbf{V_{21}} + \mathbf{V_{31}}^T \mathbf{D}(\widehat{\boldsymbol{\lambda}}) \mathbf{V_{31}} \right) & \sqrt{\alpha} \mathbf{V_{31}}^T \mathbf{D}(\widehat{\boldsymbol{\lambda}}) \mathbf{V_{32}} \\ \sqrt{\alpha} \mathbf{V_{32}}^T \mathbf{D}(\widehat{\boldsymbol{\lambda}}) \mathbf{V_{31}} & \mathbf{0} \end{bmatrix} \Bigg)^{-1} \\ & \begin{bmatrix} \mathbf{V_{11}}^T \mathbf{D}(\mathbf{E}_{B_\perp}^T \mathbf{y}) & \sqrt{\alpha} \mathbf{V_{21}}^T \mathbf{D}(\widetilde{\boldsymbol{\lambda}}) & \sqrt{\alpha} \mathbf{V_{31}}^T \mathbf{D}(\widehat{\boldsymbol{\lambda}}) \\ \mathbf{0} & \mathbf{0} & \mathbf{V_{32}}^T \mathbf{D}(\widehat{\boldsymbol{\lambda}}) \end{bmatrix} \Bigg) \Bigg). \end{aligned} \tag{A.12}$$

Hence,

$$\begin{aligned} & J_B(\mathbf{y}|\boldsymbol{\lambda}) \\ & = \max \left(\operatorname{diag} \left(\mathbf{V_{11}} (\mathbf{V_{11}}^T \mathbf{D}(\mathbf{E}_{B_\perp}^T \mathbf{y}) \mathbf{V_{11}})^{-1} \mathbf{V_{11}}^T \mathbf{D}(\mathbf{E}_{B_\perp}^T \mathbf{y}) \right), \right. \\ & \left. \operatorname{diag} \left(\mathbf{V_{32}} (\mathbf{V_{32}}^T \mathbf{D}(\widehat{\boldsymbol{\lambda}}) \mathbf{V_{32}})^{-1} \mathbf{V_{32}}^T \mathbf{D}(\widehat{\boldsymbol{\lambda}}) \right) \right). \end{aligned} \tag{A.13}$$

The objective function (A.13) equals the larger one between two terms. The first term is only related to matrix $\mathbf{V}_{11}$ and the non-zero elements in vector $\mathbf{y}$; and the second term is only related to matrix $\mathbf{V}_{32}$ and the direction vector $\widehat{\boldsymbol{\lambda}}$. Since our algorithm starts from the point $\mathbf{1}$, the algorithm will never approach a boundary point whose objective function (A.13) approaches values greater than $J(\mathbf{1})$ along all directions $\boldsymbol{\lambda} \in \mathfrak{L}_0$. As a result, we do not need to consider such boundary points in our proof. The boundary points whose corresponding matrix $\mathbf{V}_{11}$ is a square matrix are such points, since their objective functions (A.13) equal 1. So that $n_{B_\perp} < r_1$. In this case, because of condition 2) in Theorem 4.2, the rank of any h rows in $\mathbf{V}_{32}$ is $\min(h, r_1)$. So that when more than $r - r_1 + 1$ elements in $\widehat{\boldsymbol{\lambda}}$ approach 0, the value of

$$\max \left(\operatorname{diag} \left(\mathbf{V_{32}} (\mathbf{V_{32}}^T \mathbf{D}(\widehat{\boldsymbol{\lambda}}) \mathbf{V_{32}})^{-1} \mathbf{V_{32}}^T \mathbf{D}(\widehat{\boldsymbol{\lambda}}) \right) \right)$$

approaches 1. Therefore, we can further constrain the domain of direction $\boldsymbol{\lambda}$:

$$\begin{aligned}\mathfrak{L} = \quad & \{\boldsymbol{\lambda} | \max(\boldsymbol{\lambda}) < \infty, \min(\boldsymbol{\lambda}) \geq 0, \\ & \max\left(\operatorname{diag}\left(\mathbf{V_{32}}(\mathbf{V_{32}}^T\mathbf{D}(\widehat{\boldsymbol{\lambda}})\mathbf{V_{32}})^{-1}\mathbf{V_{32}}^T\mathbf{D}(\widehat{\boldsymbol{\lambda}})\right)\right) \leq J(\mathbf{1})\Big\}.\end{aligned}$$

Thus, the minimums of $\Delta^{(t)}J_B(\mathbf{y}|\boldsymbol{\lambda})$ for any finite integer t with respect to $\boldsymbol{\lambda}$ in the domain $\mathfrak{L}$ exist. In case 2, there must exist at least one point and an associated direction $\boldsymbol{\lambda} \in \mathfrak{L}$ in $\mathfrak{D}_1$ such that

$$\Delta J_B^{(t)}(\mathbf{y}|\boldsymbol{\lambda}) = 0, \tag{A.14}$$

for all finite integers t, and the algorithm will ultimately converge to such a point or a set of such points. Any boundary points in which Lemma 4.4 still holds cannot satisfy equation (A.14). So, we need to scrutinize Lemmas 4.2 to 4.4 for the boundary points with their associated directions $\boldsymbol{\lambda}$.

First, notice that if a continuous function $f(\alpha)$ is well defined and equal to or greater than 0 in the domain $(0, +\infty)$, then $\lim_{\alpha\to 0^+} f(\alpha) \geq 0$. As a result, all properties in Lemmas 4.3 and 4.4 using "greater than or equal to" symbols, such as $\widehat{y}_{1j} \geq \widehat{y}_{0j}$, $J(\widehat{\mathbf{y}}_0) \geq d_j(\widehat{\mathbf{y}}_1)$, hold for boundary points. Since $J(\widehat{\mathbf{y}}_0) \geq J(\widehat{\mathbf{y}}_1)$ in every inner iteration, $J(\mathbf{y}) \leq J(\mathbf{1}) < 1$ in the whole optimization process. So, Lemma 4.2 holds automatically.

Then, look into the other properties in Lemma 4.3. There are two cases in the j-th inner iteration: $j \in B_r$ and $j \in B_\perp \cup B_o$.

Considering the case $j \in B_r$ first, we rewrite equation (A.3):

$$\begin{bmatrix} d_j(\widehat{y}_j) \\ d_1(\widehat{y}_j) \\ \vdots \\ d_{j-1}(\widehat{y}_j) \\ d_{j+1}(\widehat{y}_j) \\ \vdots \\ d_{n_2}(\widehat{y}_j) \end{bmatrix} = \operatorname{diag}\left(\begin{bmatrix} \frac{\alpha q \frac{\widehat{y}_j}{\alpha}}{\alpha q \frac{\widehat{y}_j}{\alpha}+1} & * \\ * & \mathbf{A} - \frac{\frac{\widehat{y}_j}{\alpha}}{\alpha q \frac{\widehat{y}_j}{\alpha}+1}\alpha\mathbf{B} \end{bmatrix}\right). \tag{A.15}$$

Suppose j is the w-th index in the set B_r. Let $\mathbf{V}_{31\setminus w}$, $\mathbf{V}_{32\setminus w}$, and $\mathbf{E}_{B_r\setminus w}$ denote the matrices by removing the w-th row from $\mathbf{V}_{31}$, and $\mathbf{V}_{32}$, and $\mathbf{E}_{B_r}$ respectively. Let $\mathbf{v}_{31(w)}$ and $\mathbf{v}_{32(w)}$ denote the w-th column of $\mathbf{V}_{31}^T$ and $\mathbf{V}_{32}^T$, respectively. It will be proved after the j-th iteration that $j \in B_r$, $\lim_{\alpha\to 0^+} \widehat{y}_{1j}/\alpha < \infty$, which means that the inner iteration cannot remove elements from the set B_r.

Let $\widehat{\boldsymbol{\lambda}}_{\backslash w} = \lim_{\alpha\to 0^+} \frac{1}{\alpha}\mathbf{E}_{B_r\backslash w}^T \widehat{\mathbf{y}}_0$, thus $\max(\widehat{\boldsymbol{\lambda}}_{\backslash w}) < \infty$. We have

$$\alpha q(\alpha) = \begin{bmatrix} \sqrt{\alpha}\mathbf{v}_{31(w)} \\ \mathbf{v}_{32(w)} \end{bmatrix}^T \left(\begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{V}_{21} & \mathbf{0} \\ \sqrt{\alpha}\mathbf{V}_{31\backslash w} & \mathbf{V}_{32\backslash w} \end{bmatrix}^T \mathbf{D}\left(\begin{bmatrix} \mathbf{E}_{B_\perp}^T \widehat{\mathbf{y}}_0 \\ \mathbf{E}_{B_o}^T \widehat{\mathbf{y}}_0 \\ \frac{1}{\alpha}\mathbf{E}_{B_r\backslash w}^T \widehat{\mathbf{y}}_0 \end{bmatrix} \right) \cdot \begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{V}_{21} & \mathbf{0} \\ \sqrt{\alpha}\mathbf{V}_{31\backslash w} & \mathbf{V}_{32\backslash w} \end{bmatrix} \right)^{-1} \begin{bmatrix} \sqrt{\alpha}\mathbf{v}_{31(w)} \\ \mathbf{v}_{32(w)} \end{bmatrix}. \tag{A.16}$$

Since $J(\widehat{\mathbf{y}}_0) \leq J(\mathbf{1}) < 1$, $\|\frac{1}{\alpha}\mathbf{E}_{B_r}^T \widehat{\mathbf{y}}_0\|_0 \geq r - r_1 + 1$. Thus, $\|\frac{1}{\alpha}\mathbf{E}_{B_r\backslash w}^T \widehat{\mathbf{y}}_0\|_0 \geq r - r_1$. Moreover, any $r - r_1$ rows in matrix $\mathbf{V}_{32\backslash i}$ are independent under condition 2) in Theorem 4.2. So, the following matrix is invertible and positive definite:

$$\lim_{\alpha\to 0^+} \mathbf{V}_{32\backslash w}^T \mathbf{D}(\frac{1}{\alpha}\mathbf{E}_{B_r\backslash w}^T \widehat{\mathbf{y}}_0)\mathbf{V}_{32\backslash i}.$$

Since the matrix $\mathbf{V}_{11}$ is full rank, and $\mathbf{E}_{B_\perp}^T \widehat{\mathbf{y}}_0 > 0$, the following matrix is also invertible and positive definite:

$$\lim_{\alpha\to 0^+} \mathbf{V}_{11}^T \mathbf{D}(\mathbf{E}_{B_\perp}^T \widehat{\mathbf{y}}_0)\mathbf{V}_{11} + \mathbf{V}_{21}^T \mathbf{D}(\mathbf{E}_{B_o}^T \widehat{\mathbf{y}}_0)\mathbf{V}_{21}.$$

So, $0 < \lim_{\alpha\to 0^+} \alpha q(\alpha) < \infty$. Similarly, we can find that

$$\mathbf{0} \leq \lim_{\alpha\to 0^+} \operatorname{diag}(\mathbf{A}(\alpha)) \leq \mathbf{1},$$
$$\lim_{\alpha\to 0^+} \operatorname{diag}(\alpha\mathbf{B}(\alpha)) \geq \mathbf{0}.$$

Moreover, if there exists an index o that $\lim_{\alpha\to 0^+} \mathbf{e}_o^T \mathbf{A}(\alpha)\mathbf{e}_o = 1$, then $\lim_{\alpha\to 0^+} \alpha\mathbf{e}_o^T \mathbf{B}(\alpha)\mathbf{e}_o \neq 0$. The reason is the same as the one provided in Lemma 4.3. Then,

$$\lim_{\alpha\to 0^+} \frac{1}{\alpha}\gamma_t = \frac{\lim_{\alpha\to 0^+} \mathbf{e}_t^T \mathbf{A}(\alpha)\mathbf{e}_t}{(1 - \lim_{\alpha\to 0^+} \mathbf{e}_t^T \mathbf{A}(\alpha)\mathbf{e}_t) \lim_{\alpha\to 0^+} \alpha q(\alpha) + \lim_{\alpha\to 0^+} \alpha\mathbf{e}_t^T \mathbf{B}(\alpha)\mathbf{e}_t} < \infty$$

Then, the proof of Lemma 4.3 on the boundary points is the same except that property 1) should be modified as: $\lim_{\alpha\to 0^+} \frac{\widehat{y}_{1j}}{\alpha} \geq \lim_{\alpha\to 0^+} \frac{\widehat{y}_{0j}}{\alpha}$. The equality holds if and only if $\lim_{\alpha\to 0^+} J(\widehat{\mathbf{y}}_0) = \lim_{\alpha\to 0^+} d_j(\widehat{\mathbf{y}}_0)$.

Then, consider the case that $j \in B_\perp \bigcup B_o$. Similarly, we can define $\mathbf{V}_{11\backslash w}$, $\mathbf{V}_{21\backslash w}$, $\mathbf{E}_{B_\perp\backslash w}$, and $\mathbf{E}_{B_o\backslash w}$ by removing the w-th row from the original matrices. Let $\mathbf{v}_{11(w)}$ and $\mathbf{v}_{21(w)}$ denote the w-th column of $\mathbf{V}_{11}^T$ and $\mathbf{V}_{21}^T$, respectively. If

$j \in B_\perp$,

$$q(\alpha) = \begin{bmatrix} \mathbf{v}_{11(w)} \\ \mathbf{0} \end{bmatrix}^T \left(\begin{bmatrix} \mathbf{V}_{11\setminus w} & \mathbf{0} \\ \mathbf{V}_{21} & \mathbf{0} \\ \sqrt{\alpha}\mathbf{V}_{31} & \mathbf{V}_{32} \end{bmatrix}^T \mathbf{D}\left(\begin{bmatrix} \mathbf{E}_{B_\perp\setminus w}^T \widehat{\mathbf{y}}_0 \\ \mathbf{E}_{B_o}^T \widehat{\mathbf{y}}_0 \\ \frac{1}{\alpha}\mathbf{E}_{B_r}^T \widehat{\mathbf{y}}_0 \end{bmatrix} \right) \cdot \begin{bmatrix} \mathbf{V}_{11\setminus w} & \mathbf{0} \\ \mathbf{V}_{21} & \mathbf{0} \\ \sqrt{\alpha}\mathbf{V}_{31} & \mathbf{V}_{32} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{v}_{11(w)} \\ \mathbf{0} \end{bmatrix} \tag{A.17}$$

Notice that $\mathbf{V}_{11\setminus w}$ is still a full column rank matrix. Otherwise, the first term in objective function (A.13) equals 1. As a result, the following matrix is invertible and positive definite:

$$\mathbf{V}_{11\setminus w}^T \mathbf{D}(\mathbf{E}_{B_\perp\setminus w}^T \widehat{\mathbf{y}}_0)\mathbf{V}_{11\setminus w}. \tag{A.18}$$

So, $0 < \lim_{\alpha\to 0^+} q(\alpha) < \infty$. Similarly, we can find that

$$\begin{aligned} \mathbf{0} \le \lim_{\alpha\to 0^+} \operatorname{diag}(\mathbf{A}(\alpha)) \le \mathbf{1}, \\ \lim_{\alpha\to 0^+} \operatorname{diag}(\mathbf{B}(\alpha)) \ge \mathbf{0}. \end{aligned}$$

Moreover, if there exists an index o that $\lim_{\alpha\to 0^+} \mathbf{e}_o^T \mathbf{A}(\alpha)\mathbf{e}_o = 1$, then $\lim_{\alpha\to 0^+} \mathbf{e}_o^T \mathbf{B}(\alpha)\mathbf{e}_o \neq 0$. The reason is the same as the one provided in Lemma 4.3.

For the case that $j \in B_o$,

$$q(\alpha) = \begin{bmatrix} \mathbf{v}_{21(w)} \\ \mathbf{0} \end{bmatrix}^T \left(\begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{V}_{21\setminus w} & \mathbf{0} \\ \sqrt{\alpha}\mathbf{V}_{31} & \mathbf{V}_{32} \end{bmatrix}^T \mathbf{D}\left(\begin{bmatrix} \mathbf{E}_{B_\perp}^T \widehat{\mathbf{y}}_0 \\ \mathbf{E}_{B_o\setminus w}^T \widehat{\mathbf{y}}_0 \\ \frac{1}{\alpha}\mathbf{E}_{B_r}^T \widehat{\mathbf{y}}_0 \end{bmatrix} \right) \cdot \begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{V}_{21\setminus w} & \mathbf{0} \\ \sqrt{\alpha}\mathbf{V}_{31} & \mathbf{V}_{32} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{v}_{21(w)} \\ \mathbf{0} \end{bmatrix} \tag{A.19}$$

Similarly,

$$\begin{aligned} 0 < \lim_{\alpha\to 0^+} q(\alpha) < \infty \\ \mathbf{0} \le \lim_{\alpha\to 0^+} \operatorname{diag}(\mathbf{A}(\alpha)) \le \mathbf{1}, \\ \lim_{\alpha\to 0^+} \operatorname{diag}(\mathbf{B}(\alpha)) \ge \mathbf{0}. \end{aligned}$$

And if there exists an index o that $\lim_{\alpha\to 0^+} \mathbf{e}_o^T \mathbf{A}(\alpha)\mathbf{e}_o = 1$, then $\lim_{\alpha\to 0^+} \mathbf{e}_o^T \mathbf{B}(\alpha)\mathbf{e}_o \neq 0$. Then, the whole proof procedure is the same as Lemma 4.3. Moreover, according to equation (A.12), $d_j(\widehat{\mathbf{y}}_0) = 0$ if $j \in B_o$. In other words, properties 1) can be simply expressed as $\lim_{\alpha\to 0^+} \widehat{y}_{1j} > \lim_{\alpha\to 0^+} \widehat{y}_{0j}$.

Then, go to Lemma 4.4. $\lim_{\alpha\to 0^+} \widehat{y}_{1j} > \lim_{\alpha\to 0^+} \widehat{y}_{0j}$ for $j \in B_o$. Thus, $B_o \subset O_\perp$. Suppose that $\mathbf{y}_k$ is a boundary point approached along direction $\boldsymbol{\lambda}$. Denote it by $\mathbf{y}_B(\alpha|\boldsymbol{\lambda})$. According to the modified Lemma 4.3 on the boundary points, we have

$$\mathbf{0} < \lim_{\alpha\to 0^+} \mathbf{E}_{O_\perp}^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \begin{bmatrix} \mathbf{E}_{B_\perp}(\widetilde{\mathbf{y}} - \mathbf{y}) \\ \mathbf{E}_{B_o}\widetilde{\mathbf{y}} - \alpha\widetilde{\boldsymbol{\lambda}} \\ \frac{1}{\alpha}\mathbf{E}_{B_r}\widetilde{\mathbf{y}} - \widehat{\boldsymbol{\lambda}} \end{bmatrix} < \infty. \tag{A.20}$$

Let $\mathbf{G}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda}))$ denote the matrix

$$\begin{aligned} \mathbf{G}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda})) = & \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sqrt{\alpha}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{V}\mathbf{R}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda}))^{-1}\mathbf{V}^T \\ & \cdot \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sqrt{\alpha}\mathbf{I} \end{bmatrix} \end{aligned} \tag{A.21}$$

The following equations can be easily verified:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sqrt{\alpha}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{V}_{21} & \mathbf{0} \\ \mathbf{V}_{31} & \mathbf{V}_{32} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{V}_{21} & \mathbf{0} \\ \sqrt{\alpha}\mathbf{V}_{31} & \mathbf{V}_{32} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sqrt{\alpha}\mathbf{I} \end{bmatrix}; \tag{A.22}$$

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{1}{\sqrt{\alpha}}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{V}_{21} & \mathbf{0} \\ \sqrt{\alpha}\mathbf{V}_{31} & \mathbf{V}_{32} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{V}_{21} & \mathbf{0} \\ \mathbf{V}_{31} & \mathbf{V}_{32} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{\alpha}}\mathbf{I} \end{bmatrix}. \tag{A.23}$$

Substitute equations (A.22) and (A.23) into equation (A.21),

$$\begin{aligned} \mathbf{G}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda})) = & \begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{V}_{21} & \mathbf{0} \\ \sqrt{\alpha}\mathbf{V}_{31} & \mathbf{V}_{32} \end{bmatrix} \left(\begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{V}_{21} & \mathbf{0} \\ \sqrt{\alpha}\mathbf{V}_{31} & \mathbf{V}_{32} \end{bmatrix}^T \mathbf{D} \left(\begin{bmatrix} \mathbf{E}_{B_\perp}\mathbf{y} \\ \alpha\widetilde{\boldsymbol{\lambda}} \\ \widehat{\boldsymbol{\lambda}} \end{bmatrix} \right) \right. \\ & \left. \cdot \begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{V}_{21} & \mathbf{0} \\ \sqrt{\alpha}\mathbf{V}_{31} & \mathbf{V}_{32} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{V}_{21} & \mathbf{0} \\ \sqrt{\alpha}\mathbf{V}_{31} & \mathbf{V}_{32} \end{bmatrix}^T \end{aligned} \tag{A.24}$$

Define the following diagonal matrices:

$$\begin{aligned} & \mathbf{K}_B(\alpha|O) \\ = \; & \mathbf{E}_O^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sqrt{\alpha}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{E}_O; \end{aligned} \tag{A.25}$$

$$\begin{aligned} & \mathbf{K}_B(\alpha|O_\perp) \\ = \; & \mathbf{E}_{O_\perp}^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sqrt{\alpha}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{E}_{O_\perp}. \end{aligned} \tag{A.26}$$

We have

$$\begin{aligned}
&\mathbf{K}_B(\alpha|O)\mathbf{E}_O^T \\
&= \mathbf{E}_O^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sqrt{\alpha}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}; \qquad \text{(A.27)}
\end{aligned}$$

$$\begin{aligned}
&\mathbf{K}_B(\alpha|O_\perp)\mathbf{E}_{O_\perp}^T \\
&= \mathbf{E}_{O_\perp}^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sqrt{\alpha}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}. \qquad \text{(A.28)}
\end{aligned}$$

Thus,

$$\begin{aligned}
&\mathbf{V}_O\mathbf{R}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda}))^{-1}\mathbf{V}_{O_\perp}^T \\
&= \mathbf{E}_O^T\mathbf{V}\mathbf{R}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda}))^{-1}\mathbf{V}^T\mathbf{E}_{O_\perp} \\
&= \mathbf{K}_B(\alpha|O)^{-1}\mathbf{E}_O^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \mathbf{G}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda})) \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r} \end{bmatrix} \mathbf{E}_{O_\perp}\mathbf{K}_B(\alpha|O_\perp)^{-1}; \qquad \text{(A.29)}
\end{aligned}$$

$$\begin{aligned}
&\mathbf{V}_{O_\perp}\mathbf{R}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda}))^{-1}\mathbf{V}_{O_\perp}^T \\
&= \mathbf{E}_{O_\perp}^T\mathbf{V}\mathbf{R}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda}))^{-1}\mathbf{V}^T\mathbf{E}_{O_\perp} \\
&= \mathbf{K}_B(\alpha|O_\perp)^{-1}\mathbf{E}_{O_\perp}^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \mathbf{G}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda})) \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r} \end{bmatrix} \mathbf{E}_{O_\perp}\mathbf{K}_B(\alpha|O_\perp)^{-1}; \qquad \text{(A.30)}
\end{aligned}$$

$$\begin{aligned}
\mathbf{D}_{O_\perp}(\widetilde{\mathbf{y}} - \mathbf{y}_B(\alpha|\boldsymbol{\lambda})) = \; &\mathbf{K}_B(\alpha|O_\perp)\mathbf{E}_{O_\perp}^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \mathbf{D}\left(\begin{bmatrix} \mathbf{E}_{B_\perp}(\widetilde{\mathbf{y}} - \mathbf{y}) \\ \mathbf{E}_{B_o}\widetilde{\mathbf{y}} - \alpha\widetilde{\boldsymbol{\lambda}} \\ \frac{1}{\alpha}\mathbf{E}_{B_r}\widetilde{\mathbf{y}} - \widehat{\boldsymbol{\lambda}} \end{bmatrix}\right) \\
&\cdot \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{E}_{O_\perp}\mathbf{K}_B(\alpha|O_\perp); \qquad \text{(A.31)}
\end{aligned}$$

$$\mathbf{D}_O(\mathbf{y}_B(\alpha|\boldsymbol{\lambda})) = \mathbf{K}_B(\alpha|O)\mathbf{E}_O^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \mathbf{D}\left(\begin{bmatrix} \mathbf{E}_{B_\perp}\mathbf{y} \\ \alpha\widetilde{\boldsymbol{\lambda}} \\ \widehat{\boldsymbol{\lambda}} \end{bmatrix}\right) \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{E}_O\mathbf{K}_B(\alpha|O). \qquad \text{(A.32)}$$

Substitute equations (A.29-A.32) into equation (A.9),

$$
\begin{aligned}
&\mathbf{E}_o^T \begin{bmatrix} d_1(\widetilde{\mathbf{y}}) \\ \vdots \\ d_{n_2}(\widetilde{\mathbf{y}}) \end{bmatrix} - \mathbf{E}_o^T \begin{bmatrix} d_1(\mathbf{y}_B(\alpha|\boldsymbol{\lambda})) \\ \vdots \\ d_{n_2}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda})) \end{bmatrix} \\
= \; & -\mathrm{diag}\left(\mathbf{E}_O^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \mathbf{G}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda})) \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{E}_{O_\perp} \right. \\
& \cdot \left(\mathbf{E}_{O_\perp}^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \mathbf{G}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda})) \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{E}_{O_\perp} \right. \\
& \left. + \left(\mathbf{E}_{O_\perp}^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \mathbf{D} \left(\begin{bmatrix} \mathbf{E}_{B_\perp}(\widetilde{\mathbf{y}} - \mathbf{y}) \\ \mathbf{E}_{B_o}\widetilde{\mathbf{y}} - \alpha\widetilde{\boldsymbol{\lambda}} \\ \frac{1}{\alpha}\mathbf{E}_{B_r}\widetilde{\mathbf{y}} - \widehat{\boldsymbol{\lambda}} \end{bmatrix} \right) \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{E}_{O_\perp} \right)^{-1} \right)^{-1} \\
& \cdot \mathbf{E}_{O_\perp}^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \mathbf{G}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda})) \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{E}_O \\
& \left. \cdot \, \mathbf{E}_O^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \mathbf{D} \left(\begin{bmatrix} \mathbf{E}_{B_\perp}\mathbf{y} \\ \widetilde{\alpha\boldsymbol{\lambda}} \\ \widehat{\boldsymbol{\lambda}} \end{bmatrix} \right) \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{E}_O \right).
\end{aligned}
\tag{A.33}
$$

According to equation (A.20),

$$
\lim_{\alpha \to 0^+} \mathbf{E}_{O_\perp}^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \mathbf{D} \left(\begin{bmatrix} \mathbf{E}_{B_\perp}(\widetilde{\mathbf{y}} - \mathbf{y}) \\ \mathbf{E}_{B_o}\widetilde{\mathbf{y}} - \alpha\widetilde{\boldsymbol{\lambda}} \\ \frac{1}{\alpha}\mathbf{E}_{B_r}\widetilde{\mathbf{y}} - \widehat{\boldsymbol{\lambda}} \end{bmatrix} \right) \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{E}_{O_\perp} > 0. \tag{A.34}
$$

According to equation (A.24),

$$
\lim_{\alpha \to 0+} \mathbf{E}_{O_\perp}^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \mathbf{G}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda})) \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{E}_{O_\perp} \geq 0. \tag{A.35}
$$

Since $B_o \subset O_\perp$, the diagonal matrix

$$
\mathbf{E}_O^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \mathbf{D} \left(\begin{bmatrix} \mathbf{E}_{B_\perp}\mathbf{y} \\ \widetilde{\alpha\boldsymbol{\lambda}} \\ \widehat{\boldsymbol{\lambda}} \end{bmatrix} \right) \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{E}_O > 0. \tag{A.36}
$$

Because of equations (A.34-A.36), equation (A.33) is a zero vector if and only

if

$$\lim_{\alpha\to 0^+} \mathbf{E}_O^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T \mathbf{G}(\mathbf{y}_B(\alpha|\boldsymbol{\lambda})) \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{E}_{O_\perp} = \mathbf{E}_O^T \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix}^T$$
$$\cdot \begin{bmatrix} \begin{bmatrix} \mathbf{V}_{11} \\ \mathbf{V}_{21} \end{bmatrix} (\mathbf{V}_{11}^T \mathbf{D}(\mathbf{E}_{\mathbf{B}_\perp}\mathbf{y})\mathbf{V}_{11})^{-1} \begin{bmatrix} \mathbf{V}_{11} \\ \mathbf{V}_{21} \end{bmatrix}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{32}(\mathbf{V}_{32}^T \mathbf{D}(\mathbf{E}_{\mathbf{B}_\perp}\mathbf{y})\mathbf{V}_{32})^{-1}\mathbf{V}_{32}^T \end{bmatrix}$$
$$\cdot \begin{bmatrix} \mathbf{E}_{B_\perp}^T \\ \mathbf{E}_{B_o}^T \\ \mathbf{E}_{B_r}^T \end{bmatrix} \mathbf{E}_{O_\perp} = \mathbf{0}. \tag{A.37}$$

If

$$\mathbf{V}_{21}(\mathbf{V}_{11}^T \mathbf{D}(\mathbf{E}_{\mathbf{B}_\perp}\mathbf{y})\mathbf{V}_{11})^{-1}\mathbf{V}_{11}^T = \mathbf{0}, \tag{A.38}$$

then

$$\begin{aligned} & \operatorname{rank}\left(\begin{bmatrix} \mathbf{V}_{11} \\ \mathbf{V}_{21} \end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix} \mathbf{V}_{11} \\ \mathbf{V}_{21} \end{bmatrix} (\mathbf{V}_{11}^T \mathbf{D}(\mathbf{E}_{\mathbf{B}_\perp}\mathbf{y})\mathbf{V}_{11})^{-1} \begin{bmatrix} \mathbf{V}_{11} \\ \mathbf{V}_{21} \end{bmatrix}^T\right) \\ = \; & \operatorname{rank}\left(\mathbf{V}_{11}(\mathbf{V}_{11}^T \mathbf{D}(\mathbf{E}_{\mathbf{B}_\perp}\mathbf{y})\mathbf{V}_{11})^{-1}\mathbf{V}_{11}^T\right) + \operatorname{rank}\left(\mathbf{V}_{21}(\mathbf{V}_{11}^T \mathbf{D}(\mathbf{E}_{\mathbf{B}_\perp}\mathbf{y})\mathbf{V}_{11})^{-1}\mathbf{V}_{21}^T\right) \\ = \; & \operatorname{rank}(\mathbf{V}_{11}) + \operatorname{rank}(\mathbf{V}_{21}) \end{aligned} \tag{A.39}$$

It can be proved that equation (A.39) holds only if $\mathbf{V}_{21}$ is a zero or empty matrix. If it is a zero matrix, it means that several columns in the outlier free data matrix are zeros, which contradicts Assumption 1. So, $\mathbf{V}_{21}$ must be an empty matrix, i.e., $B_o = \emptyset$, if equation (A.39) holds.

If there exists a partition on set $B_\perp$ with the corresponding matrices $\mathbf{E}_\zeta$ and $\mathbf{E}_\xi$ that

$$\mathbf{E}_\zeta^T \mathbf{V}_{11}(\mathbf{V}_{11}^T \mathbf{D}(\mathbf{E}_{\mathbf{B}_\perp}\mathbf{y})\mathbf{V}_{11})^{-1}\mathbf{V}_{11}^T \mathbf{E}_\xi = \mathbf{0},$$

Denote the numbers of rows of $\mathbf{E}_\zeta$ and $\mathbf{E}_\xi$ by n_ζ and n_ξ, respectively. If $n_\zeta = \operatorname{rank}(\mathbf{E}_\zeta^T \mathbf{V}_{11})$ or $n_\xi = \operatorname{rank}(\mathbf{E}_\xi^T \mathbf{V}_{11})$, the value of the objective function (A.13) is 1. If $n_\zeta > \operatorname{rank}(\mathbf{E}_\zeta^T \mathbf{V}_{11})$ and $n_\xi > \operatorname{rank}(\mathbf{E}_\xi^T \mathbf{V}_{11})$, condition 2) in Theorem 4.2 cannot be satisfied. As a result, $B_\perp \subseteq O$ or $B_\perp \subseteq O_\perp$. Analogously, $B_r \subseteq O$ or $B_r \subseteq O_\perp$.

Now there are only three possibilities of set O: B_r, $B_r \bigcup B_\perp$, and $B_\perp$. Firstly, if $O = B_r$, $J_B(\mathbf{y}|\boldsymbol{\lambda}) = (r - r_1)/n_{B_r}$. However, according to condition 1) in Theorem 4.2, $(r - r_1)/n_{B_r} < r/n_2$, which contradicts Lemma 4.2. So, $O \neq B_r$. Secondly, if $B_o \neq \emptyset$, $B_\perp \subseteq O_\perp$. Otherwise, equation (A.37) holds only if equation (A.38) holds, which is impossible when $B_o \neq \emptyset$. Then, the proof of Lemma 4.4 on boundary points is the same as the original Lemma

4.4. There is only one case left: $B_\perp = O$, $B_r = O_\perp$, and $B_o = \emptyset$. If this is the case, $\mathbf{E}_{B_\perp}^T \widetilde{\mathbf{y}} = \mathbf{E}_{B_\perp}^T \mathbf{y}$. Thus, $\max(\widetilde{\mathbf{y}}) = \max(\mathbf{E}_{B_\perp}^T \mathbf{y}) = 1$. Then, $\mathbf{y}_{k+1} = \widetilde{\mathbf{y}}$. Since we have proven that

$$\lim_{\alpha \to 0^+} \frac{1}{\alpha} \mathbf{E}_{B_r} \widetilde{\mathbf{y}} > \boldsymbol{\lambda},$$

our coordinate descent algorithm diverges from instead of converges to the boundary point in this case.

To sum up, under the condition provided in Theorem 4.2, the case that the algorithm converges to a set of boundary points never happens. As a result, our coordinate descent algorithm always converges to the global optimal point r/n_2 when the condition is satisfied. □