Data Driven Methods for Analysis and Design of Industrial Alarm Systems

by

Muhammad Shahzad Afzal

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Control Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

An alarm system is an integral part of process monitoring and safety. A poorly configured alarm system can sometimes cause more harm than good, by introducing many false and nuisance alarms for the operators. Various standards on alarm system management and rationalization suggest many configuration methods to help in improving the overall performance of alarm monitoring systems. In this thesis, the problem of analyzing and designing alarm systems for both single- and multi-mode processes is considered.

A design procedure of a multivariate alarm system for multi-mode processes is developed. A hidden Markov model based modeling approach is adopted to capture the multi-modality of data and the mode-reachability constraints of a multi-mode process. A monitoring index utilizing the proposed two-step Viterbi algorithm is developed, and for fault isolation, reconstruction based contribution plots are used.

The utility of delay-timers in improving existing univariate alarm systems for multi-mode processes is studied. A mathematical model is developed to calculate analytical expressions for different performance indices (the false alarm rate, missed alarm rate, and expected detection delay). A particle swarm optimization based method is proposed for designing delay-timers, while satisfying the constraints on the performance indices and delay-timer lengths for various modes of the operation of a process.

The analysis and design of time-deadbands for univariate alarm systems

is also considered in this thesis. In particular, a Markov chain process based mathematical model is developed to capture the time-deadband configurations for single mode processes. Analytical expressions for the performance indices are calculated, and design procedures based on process data and alarm data are developed.

Keywords: Alarm systems, Hidden Markov models, Markov processes, Delay-timers, Time-deadbands

Preface

The research work conducted in Chapter 2 was part of the collaboration with Dr. Wen Tan, who was with the University of Alberta as a Research Associate at the time of this work. The ideas presented in Chapter 3 were my original ideas. The work presented in Chapter 4 was a result of collaboration with Dr. Iman Izadi and Ali Bandehkhoda from Isfahan University of Technology, Iran.

- Chapter 2 has been published as: Afzal, M.S., Tan, W., & Chen, T., "Process Monitoring for Multimodal Processes with Mode-Reachability Constraints", *IEEE Transactions on Industrial Electronics*, vol. 64, pp. 4325-4335, 2017.

- Chapter 3 has been published as: Afzal, M.S., & Chen, T., "Analysis and Design of Multimode Delay-Timers", *Chemical Engineering and Research Design*, vol. 120, pp. 179-193, 2017.

- Chapter 4 has been submitted for publication as: Afzal, M.S., Chen, T., Bandehkhoda, A., & Izadi, I., "Analysis and Design of Time-deadbands for Univariate Alarm Systems", *Control Engineering Practices*. A preliminary version of this chapter was presented in American Control Conference (ACC) in Seattle, WA, USA May 24-26, 2017

Acknowledgements

This thesis would not have been possible without the help and support of the people around me, to only some of whom it is possible to give a special mention here. Above all, I would like to pay my utmost gratitude to my supervisor Prof. Tongwen Chen for his constant guidance and support. The discussions we had during our individual and group meetings helped me a lot in carrying out my research work and in writing this thesis. I could not have imagined having a better mentor than him for my Ph.D. program.

I would also like to thank Dr. Wen Tan (North China Electric Power University), and Dr. Iman Izadi (Isfahan University of Technology, Iran) for their suggestions in improving the work, and contributing towards my research. My sincere thanks to Dr. Sirih L. Shah for sharing his ideas during different workshops and meetings, and for his insightful comments on my work.

Last, but not the least, I would like to thank all of my former and current research group members, with a special mention of Wenkai Hu, Ahmad Al-Dabbagh, David Li, Ying Xiong, Ishtiza Azad, and Shiqi Lai for their encouragement, stimulating discussions, and for all the fun we had during the last three years or so.

Contents

List of Tables

List of Figures

List of Acronyms

AIC	Akaike Information Criterion
ALM	Alarm
CSTH	Continuous Stirred Tank Heater
DCS	Distributed Control System
DPCA	Dynamic Principal Component Analysis
EDD	Expected Detection Delay
EEMUA	Engineering Equipment and Materials Users Association
ERP	Enterprise Resource Planning
FAR	False Alarm Rate
FGMM	Finite Gaussian Mixture Model
HMI	Human Machine Interface
HMM	Hidden Markov Model
ISA	International Society of Automation
KDE	Kernel Density Estimation
LRLE	Lossless Run-length Encoding
MAR	Missed Alarm Rate
MES	Manufacturing Execution System
PCA	Principal Component Analysis
PLC	Programmable Logic Controller
PSO	Particle Swarm Optimization
ROC	Receiver Operating Curve

RTN	Return To Normal
SCADA	Supervisory Control and Data Acquisition
SPE	Square Prediction Error

Chapter 1

Motivation and Background

1.1 Industrial process control

Industrial process control is a complex interconnection of components like sensors, actuators and logic controllers. Depending on the scale of a plant, the number of components may vary from a few hundreds to a few thousands. The International Society of Automation (ISA) has put together a standard (ISA-95) that describes a hierarchical model of interconnection between different devices in an industrial process control system. The standard also describes terminologies and device models for each level of hierarchy, and provides a way of comparing production levels of different manufacturing processes. The hierarchical model consists of four main levels, namely, a device level, a control level, a Manufacturing Execution System (MES), and an Enterprise Resource Planning (ERP) level, as shown in Figure 1.1. Plant components such as sensors, actuators, and machines reside on the device level. Feed and production lines are also part of this level, which provide means to workers to stock inputs and collect products from the process. Machine controllers, e.g., programmable logic controllers (PLC), are part of the control level devices, which are connected to the lower level via a high bandwidth connection. Control level devices typically require very little maintenance, and thus are kept in locked areas, accessible only to people responsible for programming them.

An operator monitoring and control room resides at the MES level of the process control system, and it is generally equipped with human machine in-

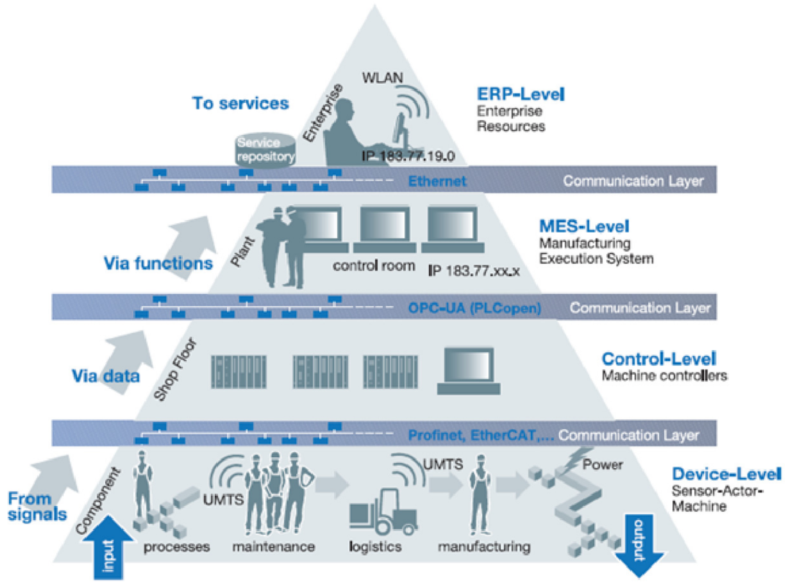


Figure 1.1: Industrial automation pyramid (ANSI/ISA-95) [20]

terface (HMI) and allows the operators to monitor the process through process measurements and alarm signals. Enterprise resources and other repositories belong to the ERP level. This level provides a restricted access to the enterprise documentation and other operating manuals. A Supervisory Control and Data Acquisition system (SCADA) and/or a Distributed Control System (DCS), which overlap between MES and ERP levels, provide interfaces for supervisory and process monitoring services.

1.2 Alarm systems

Industrial processes suffer from faults due to their components malfunctioning or failures, and depending on the criticality of the failures, these faults can even lead to serious incidents. Recently, the likelihood of disrupting the process normal operation has increased drastically due to growing complexity of industrial processes. An alarm system, which is part of the MES level, plays a critical role in alerting operators of the abnormality and enabling them in performing remedial actions. The process of generating alarm signals from

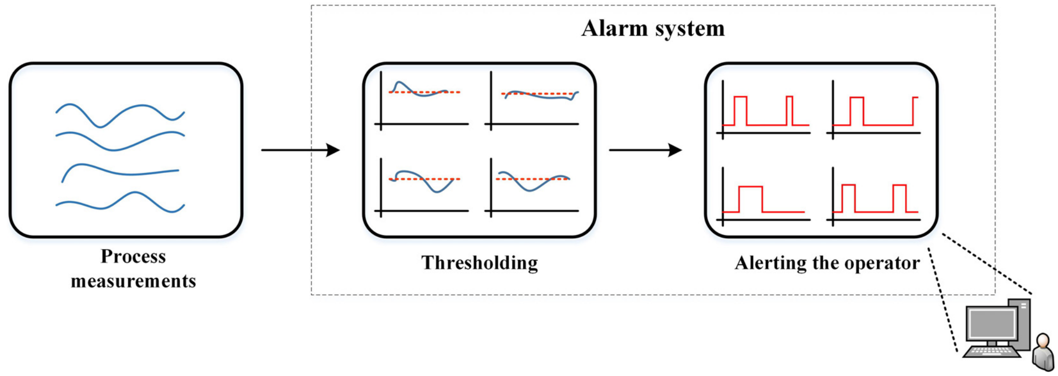


Figure 1.2: Alarm signals generation process

the measurements is shown in Figure 1.2. The inputs of an alarm system are the process measurements, and the alarm signals are generated from the process measurements by checking the variables against the configured thresholds. Based on the method used to configure thresholds, alarm systems can be classified as univariate and multivariate alarm systems. For univariate alarm systems, thresholds are configured on individual process variables, whereas for multivariate alarm systems, thresholds are configured on signals, which are generated based on the set of process measurements using some statistical procedures [24].

Ideally, only one alarm should be raised against an abnormality; however, in industries this is hardly the case. In fact, operators these days see a lot of false and nuisance alarms on their monitoring screens, due to poorly configured alarms, and this number is even higher for multi-mode processes that have various setpoints to cater the production needs, operating requirements, and the availability of input stock. The advent of modern DCS systems has made the matters worse by introducing convenient ways of configuring alarms on process variables [24]. Various surveys have shown that the current status of alarm systems performance from different sectors of industries is far from the benchmarks set by ISA-18.2 and EEMUA 191, which are considered as two de-facto standards for assessing alarm systems performance [21]. Table 1.1 shows the status of different performance measures against the benchmarks.

The performance of alarm systems can be improved by either redesigning

Table 1.1: Alarm system performance survey

Performance measure	Benchmark	Oil & Gas	Petrochemical	Power
Average alarms per hour	≤ 6	36	54	48
Average standing alarms	9	50	100	65
Peak alarms per hour	60	1320	1080	2100
Priority distribution % (low/med/high)	80/15/5	25/40/35	25/40/35	25/40/35

the entire alarm system based on the available process measurements, or by applying various alarm configurations (filters, deadbands, and delay-timers etc.) on the existing alarm system. In ISA-18.2 many alarm configuration methods have been proposed, which can be broadly classified into two main types, namely, basic methods and advanced or enhanced methods [21]. Filters, deadbands, and delay-timers can be categorized as basic alarm configuration methods; whereas techniques like state-based alarming, predictive alarming, and logic based alarming fall under the class of advanced methods. Figure 1.3 gives the taxonomy of alarm configuration methods. In this figure, vertical dots in the last tier show that there is a wide range of filters, and advanced methods available, while only a few are shown here as examples. It is worth mentioning here that these configuration can be used for designing both univariate and multivariate alarm systems.

1.3 Literature survey

In view of the current status of alarm systems, industry personnel have started to put a lot of effort in alarm system management and rationalization. Filters, deadbands and delay-timers are some of the commonly used techniques to get rid of false and nuisance alarms. In addition, some advanced techniques like state-based alarming, logic-based alarming, and predictive alarming are also in practice [19, 26, 27]. For the last few years, researchers from academia have also been engaged with industries to help them in improving their alarm systems. This collaboration has also enriched the published literature on the use and design of various alarm configuration methods [9, 2, 18, 50].

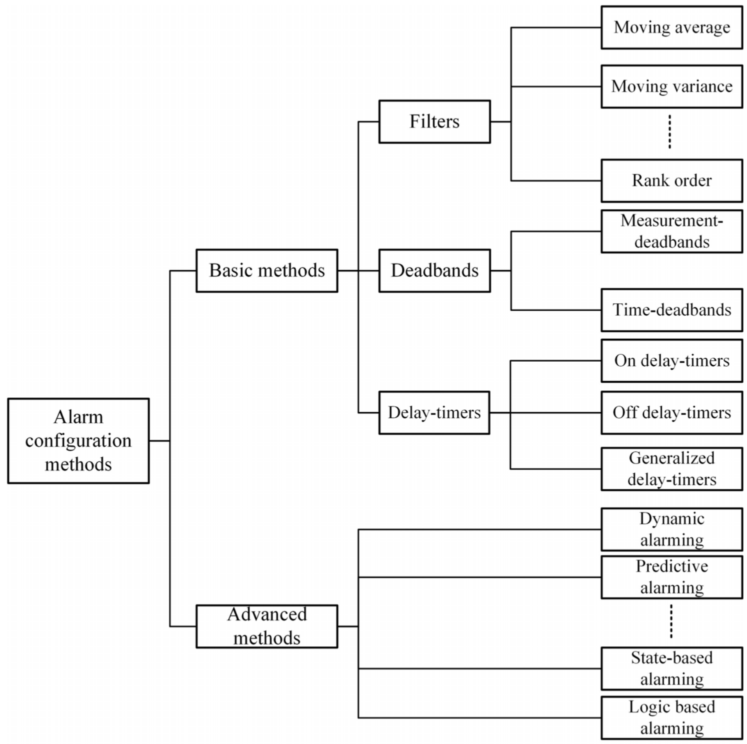


Figure 1.3: Taxonomy of alarm configuration methods

In the literature many papers can be found that deal with the analysis and design of different alarm configuration methods, e.g., the problem of designing optimal alarm filters was studied in [9, 10], and it was found out that the log-likelihood ratio filters gave the optimal performance in terms of alarm system accuracy. Numerical optimization procedures were also proposed for linear and quadratic forms of optimal filters. In [2] the authors have computed the detection delays for both on and off delay-timers, and measurement-deadbands. A design procedure based on the Receiver Operating Curve (ROC) was also proposed. Analytical expressions for run-length distribution based chattering index were computed for delay-timers and measurement-deadbands in [41]. The concept of generalized delay-timers was studied in [1], and the performance of the generalized delay-timers was also compared with the traditional on and off delay-timers. In [54] performance indices, like the False Alarm Rate (FAR), Missed Alarm Rate (MAR), and Expected Detection Delay (EDD) were computed for rank order filters based on univariate alarm systems, and the performance was compared with other filters. In [3] the authors studied the application of delay-timers for multimode processes. The performance indices (FAR, MAR, and EDD) were computed, and a design procedure based on particle swarm optimization was proposed.

A few papers can also be found in the literature that deal with the enhanced configuration methods for alarm systems, e.g., in [40] authors have developed a logic-based alarm system for power distribution unit, by taking into account the information from the breaker operation and the sequence of event recorders. Based on the testing results provided in the paper, the logic-based alarm system showed superior performance. In [42] authors devised a state-based alarm system for a nuclear power plant simulator, and through tests it was observed that the state-based alarming system provided higher usability ratings as compared to the traditional alarm system. The authors of [61] have proposed a method of dynamic alarming based on online removal of chattering and repeating alarms. In this method, alarm durations and time difference between two alarms were considered in the detection of chattering and repeating alarms. In [34] a pattern mining based predictive alarming system was proposed for alarm floods. In this method, a multiple sequence alignment algorithm was developed, and a similarity score was used to detect the similarity of incoming alarm sequences with the mined database. Some more efforts on the advanced alarm configuration methods can be found in [16, 27, 33, 58].

Literature related to model based multivariate alarm system design methods can also be found. In these methods dynamic process models are developed based on parity equations, parameter estimation, or state observers [45, 23]. Process measurements are compared with model and the residual signal is tested against a threshold to check for abnormality detections. Alarm system performance is highly dependent on the model developed for a process, which has become a very challenging and time consuming task due to growing complexity of industrial processes. On the other hand, due to the availability of vast amount of historical data, data drvien methods such as principal component analysis, partial least squares, independent component analysis, and Fisher discriminant analysis are becoming very popular [59, 65]. Extension of data-based methods to multi-mode processes have also been reported in [75, 64, 44].

1.4 Thesis contributions

This thesis focuses on data driven methods for designing univariate and multivariate alarm systems for single mode and multi-mode processes. In the field of alarm system analysis and design, the contributions of this thesis are as follow. In **Chapter 2** a multivariate alarm system design procedure for multi-mode processes is proposed. A hidden Markov model approach is adopted, and mode-reachability constraints between different modes of operation are also considered. In particular, the chapter has the following major contributions:

- An efficient fault detection method is proposed, that can not only handle the multi-mode data, but also capture the mode-reachability constraints.
- A reconstruction based method is proposed to develop contribution plots for fault isolation. The fault isolation method combined with the proposed two-step Viterbi algorithm provides satisfactory results for mode detection in the presence of mode switching restrictions and persistent faults.

In **Chapter 3** the use of delay-timers for multi-mode processes is considered. This chapter has the following major contributions:

- Analytical expressions for different performance indices (the false alarm rate, missed alarm rate, and expected detection delay) are derived for delay-timers when applied on multi-mode processes.
- A particle swarm optimization based algorithm is proposed for the design of delay-timers for multi-mode processes, while satisfying constraints on the performance indices.

In **Chapter 4** time-deadband configurations for univariate alarm systems are studied and analyzed. This chapter has the following major contributions in the field of univariate alarm systems analysis and design:

- A mathematical model is developed for time-deadband configurations based on Markov processes.
- Analytical expressions for the performance indices (the false alarm rate, missed alarm rate, and expected detection delay) are derived.
- Design procedures based on process data and alarm data are proposed.

1.5 Thesis organization

The rest of the thesis is organized as follows. In Chapter 2 a hidden Markov model based multivariate alarm system design method for multi-mode processes is presented. Chapter 3 deals with the analysis and design of delay-timers for multi-mode processes. Chapter 4 is on the use of time-deadbands for univariate alarm system design. Finally concluding remarks and possible future work are provided in Chapter 5. A detailed organization of each chapter is provided in the overview section of each chapter.

Chapter 2

Monitoring of Multi-mode Processes with Mode-reachability Constraints*

FOR increased efficiency and profitability, many processes have multiple modes of operation. Switching between different operating modes is performed according to the standard operating procedures. These procedures are set by considering safety and operating limitations of various subsystems and equipment, and thus put restrictions on the switching of the process modes. In this chapter a hidden Markov model based monitoring method is proposed to capture these operating restrictions and multi-mode process data.

2.1 Overview

Monitoring of a multi-mode process requires that the monitoring system be well equipped to deal with the multi-mode data coming from different operating conditions (modes), which occur due to, e.g., variations in the feedstock, manufacturing requirements, product demand, and operating environment [53]. In addition to multimodality of data, mode-reachability constraints are

*A version of this chapter has been published as: Afzal, M.S., Tan, W., & Chen, T. (2017), Process Monitoring for Multimodal Processes with Mode-Reachability Constraints, IEEE Transactions on Industrial Electronics, vol. 64, pp. 4325-4335 another important and challenging issue for the multi-mode process monitoring. Due to operating and equipment constraints, not every mode is reachable through other modes of operation. For example in a nuclear power plant [37], six plant-wide modes are reported and these modes are highly constrained, e.g., mode 6 (refuelling) can only be reached through mode 5 (cold shutdown), and mode 4 (hot shutdown) is reachable only through mode 3 (zero power). Despite the reachability constraints on multi-mode processes, it is still possible that under some faults or disturbances the process gets drawn towards a constrained mode. For instance, in a Three Mile Island incident, faults in the relief valve and the non-nuclear secondary system led the plant towards a constrained mode, and the operators and the monitoring system mistakenly detected this as a true mode change instead of faults. These faults eventually resulted in a partial meltdown of the uranium core; and the incident was rated as five on a the International Nuclear Event Scale [57]. Thus for efficient process monitoring of multi-mode processes, in addition to multimodality of data, it is important for the monitoring system to capture the mode switching restrictions. Any method insensitive of these constraints will result in many false and missed alarms.

Despite a rich literature of data based monitoring methods, most existing methods are either not capable of handling multi-mode data at all, or have some limitations. Traditional Principal Component Analysis (PCA) based methods work under the assumption that data belongs to uni-mode Gaussian distributions [38, 28], which is not the case for multi-mode processes. Different variants of PCA have been proposed in the literature to handle multimodality of data. In [74], multiple PCA models were used to capture different operating modes of the process and test statistics were calculated corresponding to each of the PCA model. An alarm was generated if a sample did not belong to any of the operating modes. In [73], a probability mixture based PCA model was proposed and a fault detection logic was used which was based on Square Prediction Errors (SPE). The use of multi-mode PCA and dynamic

PCA for multi-mode process monitoring was proposed in [69] and [63], respectively. Some more efforts on PCA based multi-mode process monitoring can be found in [67, 51]. Unfortunately, none of the above mentioned methods can capture the mode-reachability constraints due to lack of modeling parameters. Furthermore, the use of the Bayesian inference strategy for mode detection for Dynamic PCA and multi-mode PCA based methods results in poor mode detection when reachability of some modes in the process are constrained, as shown later in Section 2.4.

Finite Gaussian Mixture Model (FGMM) based monitoring is another method that has been used for multi-mode processes. The main advantage of this method is that a user does not have to specify information about the number of modes beforehand. In [68], a Bayesian Inference Probability based index (BIP) was proposed for FGMM. The Mahalanobis distance was used to calculate the contributions of different Gaussian clusters towards the monitoring index. In [70], FGMM was combined with a hybrid unfolding of a multi-way data matrix to capture the multi-mode data from batch processes. A localized probability index was calculated to detect the fault for each sample based on the operating modes of the process. Similar to PCA, FGMM does not model the mode switching restrictions due to the fact that these methods rely on Bayesian inference strategies for mode detection, which does not give satisfactory results in the case of mode-reachability constraints.

Quite recently, efforts have been made to use the Hidden Markov Models (HMM) for multi-mode process monitoring. An HMM can not only model the multimodality of data but also capture the mode shifting probabilities. In [43], the HMM based statistical pattern analysis method was proposed. In this method, the mode of operation was detected using a traditional window based Viterbi algorithm, and the concept of differential mode vector was introduced, which indicates the mode shift for every sample with respect to the past sample. The use of the standard window based Viterbi algorithm makes the mode detection unsuitable for the case of persistent faults, where faulty samples are also part of the window. In [66], a Bayesian inference probability index was proposed based on the HMM modeling. The posterior probabilities associated to any of the operating modes are calculated using the Mahalanobis distance, which is, insensitive to the mode-reachability constraints. In [48], independent component analysis and the HMM modeling were combined for multi-mode process monitoring. The standard Viterbi algorithm was used for mode detection, and the HMM based I^2 and SPE statistics were developed for fault detection. More literature on the HMM based process monitoring methods can be found in [4, 52, 5]. While aforementioned methods can detect the faults, in most cases fault isolation algorithms are not available; furthermore, the use of the standard Viterbi algorithm for mode detection may result in inaccurate detection when some of the mode transition probabilities are zero [47].

This chapter proposes a new monitoring method based on hidden Markov model. In this method the hidden Markov model is used to capture the multi-mode behavior and mode switching probabilities of the process/subsystem. For the root cause identification, a reconstruction based fault isolation algorithm is developed. The rest of this chapter is organized as follows. Section 2.2 describes all preliminaries related to the proposed methodology. Section 2.3 provides details about the proposed method that includes a two-step Viterbi algorithm, fault detection and fault isolation algorithms. In Section 2.4 various case studies are considered and comparisons with some existing data based methods are also provided.

2.2 Background

In this section, a brief description of a hidden Markov model is presented. Mode-reachability constraints and different ways of representing these constraints are also explained.

2.2.1 Hidden Markov models

A hidden Markov model was firstly introduced in [7] as an extension of Markov chains. An HMM can be thought of as a doubly stochastic process, where one process is used to describe the state transition probabilities, and the other captures the probability of the observation produced during each state. For a complete description of an HMM, the following elements are required [47]:

- *Number of states (Q)*: An HMM has a hidden chain of states and every state is associated with an observation probability density function.

- *State transition matrix (A)*: It describes the probability of jumping from one HMM state to another state:

$$A = [a_{i,j}], \;\; 1 \leq i, j \leq Q \tag{2.1}$$

  where $a_{i,j} = P[q_{t+1} = j | q_t = i]$.

- *Prior probability (π)*: It is the probability of starting the Markov chain in a particular state:

$$\pi_i = P[q_1 = i], \;\; 1 \leq i \leq Q \tag{2.2}$$

- *Observation probability (B)*: It describes the probability of an observation y_t at time t, for every state of an HMM:

$$B = [b_i(y_t)], \;\; 1 \leq i \leq Q \tag{2.3}$$

  where $b_i(y_t) = \Xi_i[y_t, u_i]$, and Ξ_i is a probability density function (probability mass function for discrete observations) associated with state i, with parameters u_i.

Details about three main algorithms (Baum-Welch, forward-backward, and Viterbi algorithm) associated with HMM models can be found in [47].

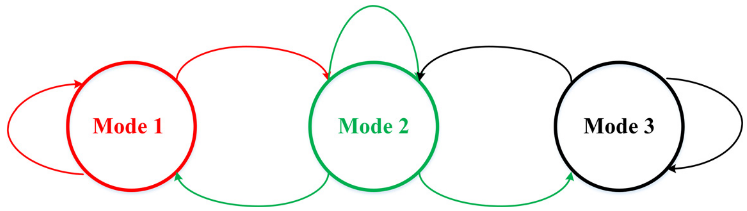


Figure 2.1: Mode-reachability map

2.2.2 Mode-reachability constraints

For multi-mode processes, it is not possible to switch from one operating mode to every other mode. In fact, there are some restrictions that an operator has to obey while switching modes. These restrictions are termed as mode-reachability constraints and can be shown using mode-reachability matrix or mode-reachability map.

A mode-reachability matrix (Ψ) is a $Q \times Q$ matrix, where each element $\psi_{i,j}$ is either 1 or 0. The element $\psi_{i,j}$ is 0 if mode j is not reachable directly through mode i and is 1 otherwise. For example, in a three-mode process, if mode 3 is not reachable through mode 1 and vice versa, the mode-reachability matrix takes the following form:

$$\Psi = \begin{bmatrix} 1 & 1 & \mathbf{0} \\ 1 & 1 & 1 \\ \mathbf{0} & 1 & 1 \end{bmatrix} \tag{2.4}$$

A mode-reachability map is a graphical way of displaying mode-reachability constraints. The mode-reachability map for the above mentioned three-mode system is shown in Figure 2.1.

2.3 Proposed methodology

In this section, the HMM based process monitoring with mode-reachability constraints is proposed. The proposed method has two main steps: offline training of an HMM model and online monitoring using the trained model. Details about these two steps are presented here.

2.3.1 Offline training of an HMM model

Among various types of HMM model (left-right, parallel path, and ergodic), ergodic models for multi-mode process monitoring are selected. An ergodic model has the property that it can reach every other state of the model within finite steps [47], which is directly in line with the property of multi-mode processes, that various modes of operation can be reached directly or indirectly through the current mode of operation. For training purposes, the Matlab toolbox for HMM was modified to incorporate continuous outputs in the model. The selection of various model parameters is done as follows:

- States of the model are selected based on the Akaike information criterion (AIC) [76]. For a complete training data-set, where measurements from every possible mode of the process are available, the model with the number of states equal to the number of modes provides the best AIC value.

- The state transition matrix is initialized randomly but with the constraint that the sum of the elements along each row of the state transition matrix is one.

- For the prior probability vector, the probability of the state corresponding to the start-up mode is set to 1.

- Finally, the observation probability matrix is initialized with appropriate probability density or mass functions, by taking into account the type of process variables.

It is worth mentioning here that use of a mode reachability matrix (Ψ) is dependent on the training data on hand. For the case of time-synchronous training data, mode reachability constraints are always satisfied, and captured as zeros in the state transition matrix, whereas for the case of time-asynchronous data, entries in the state transition matrix have to be manually set to zeros corresponding to the constrained modes. Information about the constrained modes can be collected from the process operation manuals.

2.3.2 Online process monitoring

In this subsection, a detailed description of each step involved in online monitoring is provided. An HMM model, obtained from offline training, is used for this purpose. The proposed method consists of three main steps: operating mode detection using a two-step Viterbi algorithm, fault detection using an HMM based monitoring index, and root cause detection using contribution plots. Interconnection of these stages is shown in Figure 2.2. A two-step Viterbi algorithm is proposed for mode detection. The algorithm uses the current measurement and previous one-step measurement to detect the current mode of operation. The presence of the OR block in Figure 2.2 indicates that the one-step past measurement can be an actual measurement or a reconstructed measurement (y_t^*) generated by the fault isolation block. The use of one of the inputs for mode detection depends on the status of the process at time $t-1$. Once the operating mode has been detected, a monitoring index is calculated to check for any abnormalities in the process. In case of abnormality, potential root causes are identified using a reconstruction based contribution plot.

Mode detection using a two-step Viterbi algorithm

In the context of process monitoring, the existing HMM based methods use the standard Viterbi algorithm for mode detection. The algorithm iteratively finds the state sequence ($\{q_1, q_2, ...q_T\}$) from the observation sequence $\{y_1 y_2 ... y_T\}$ using the following steps [47]:

- Maximization of the partial likelihood term ($\delta_t(i)$), which represents the best score along a single path at time t, given the model parameters (λ): $\delta_t(i) = P[q_1...q_t = i, y_1...y_t|\lambda], 1 \leq i \leq Q, 1 \leq t \leq T$

- Starting from the end time T, determination of the optimal state sequence through back tracking on $\delta_t(i)$.

Back tracking on $\delta_t(i)$ makes the algorithm unsuitable for online state detec-

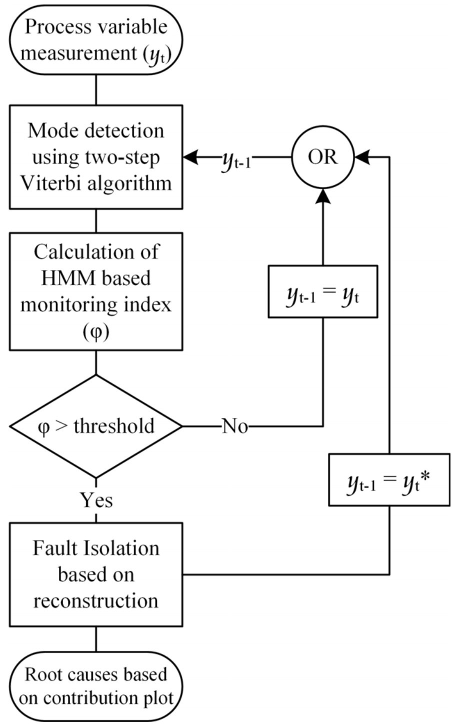


Figure 2.2: Online process monitoring

tion. Furthermore if some of the probabilities in state transition matrix are zero, i.e., $a_{i,j} = 0$ for some i and j, the detected state sequence may not be valid in the event of fault, because of the presence of abnormality in the past data. Consequently, a modified two-step Viterbi algorithm is proposed, which in combination with the fault isolation block, can not only handle the restrictions on state transition probabilities but is also suitable for online implementation. The proposed algorithm follows the same structure, as the standard Viterbi algorithm, however unlike the latter, the two-step Viterbi algorithm uses samples only from current time t and $t-1$. The sample from $t-1$ can be the actual measurement or the reconstructed input (y_t^*), generated by the fault isolation algorithm in case of abnormality at time $t-1$. The proposed two-step Viterbi algorithm, instead of maximizing $\delta_t(i)$ based on complete observation set, maximizes the probability of the expected state pairs $\delta_t(j,i)$, defined as the probability of being in state j at time t and in state i at time $t-1$, given the corresponding observations and the model parameters:

$$\delta_t(j,i) = P[q_t = j\ q_{t-1} = i | y_t y_{t-1}, \lambda], \quad \forall\, i,j \in Q \tag{2.5}$$

where y_t is the current process measurement, y_{t-1} is the process measurement (actual or reconstructed) at time $t-1$, and Q is a set of all possible states. The objective is to find the state at time t by maximizing the $\delta_t(j,i)$, i.e.,

$$q_t = \arg[\max[\delta_t(j,i)]], \ \ \forall\, j \in Q \tag{2.6}$$

By using the definition of conditional probabilities and the Bayes rule (2.5) is written as:

$$\delta_t(j,i) = \frac{P[y_t|q_t{=}j\ q_{t-1}{=}i, y_{t-1}, \lambda] P[q_t{=}j|q_{t-1}{=}i, \lambda] P[y_{t-1}|q_t{=}j, q_{t-1}{=}i, \lambda] P[q_{t-1}{=}i, \lambda]}{P[y_t, y_{t-1}, \lambda]} \tag{2.7}$$

The Markov chain property of memoryless state transitions helps in reducing (2.7) to the following:

$$\delta_t(j,i) = \frac{P[y_t|q_t{=}j, \lambda] P[q_t{=}j|q_{t-1}{=}i, \lambda] P[y_{t-1}|q_{t-1}{=}i, \lambda] P[q_{t-1}{=}i, \lambda]}{P[y_t, y_{t-1}, \lambda]} \tag{2.8}$$

Notice that the terms in the numerator are nothing but the state transition probabilities and observation probabilities, i.e., $P[y_t|q_t = j, \lambda] = b_j(y_t)$, $P[q_t = j|q_{t-1} = i, \lambda] = a_{i,j}$, $P[y_{t-1}|q_{t-1} = i, \lambda] = b_i(y_t)$, and $P[q_{t-1} = i, \lambda] = \pi_{t-1}$. Thus (2.8) takes the following form:

$$\delta_t(j,i) = \frac{b_j(y_t) a_{i,j} b_i(y_{t-1}) \pi_{t-1}}{P[y_t, y_{t-1}, \lambda]} \tag{2.9}$$

Since the denominator of (2.9) is independent of i and j for the given observed data, therefore the maximization of $\delta_t(j,i)$ is only dependent on the numerator of (2.9). Thus the optimization problem can be written as:

$$\max[\delta_t(j,i)] = \max[b_j(y_t) a_{i,j} \vartheta_{t-1}(i)], \ \ \forall\, j \in Q \tag{2.10}$$

Algorithm 1 Online two-step Viterbi algorithm

procedure $\text{Vit}_{\text{ two-step}}(y_{t-1}, y_t, q_{t-1}, \lambda)$
Set $n = q_{t-1}$
Initialization
$\pi_{t-1} = [0...0_{n-1}\ 1\ 0_{n+1}...0_Q]$
$\vartheta_{t-1}(i) = \pi_{t-1} b_i(y_{t-1}), \quad \forall i \in Q$
Recursion
$\delta_t(j,i) = \max_{1\le j\le Q}[b_j(y_t)a_{i,j}]\vartheta_{t-1}(i), \quad \forall i \in Q$
Termination
$q_t = \text{argmax}[\delta_t(j,i)]$
end procedure

where $\vartheta_{t-1}(i) = b_i(y_{t-1})\pi_{t-1}$. A recursion based solution is adopted to maximize $\delta_t(j,i)$. Algorithm 1 outlines the complete procedure for mode detection using the proposed two-step Viterbi algorithm.

For illustration purposes, a multi-mode system with the following mode-reachability matrix is simulated:

$$\Psi = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{2.11}$$

The system is initially in mode 1, and the objective is to shift the operating mode to mode 3 at $t = 700$. Due to the mode-reachability constraints, the system is first shifted to mode 2 at $t = 500$, and then to mode 3. Meanwhile, a fault is introduced at $t = 400$ and cleared at $t = 550$. This fault draws the system towards mode 3. Both traditional Viterbi algorithm and the proposed two-step Viterbi algorithm are used to detect the operating mode, and the results are shown in Figure 2.3. It is evident that during $t = 400$ to 550 the two-step Viterbi algorithm is able to detect the true operating modes irrespective of the fault, due to the use of the reconstructed data during abnormality. The traditional Viterbi algorithm, on the other hand, is not able to nullify the effect of the fault during that time period.

Fault detection

A trained HMM model is used to calculate the likelihood of a new sample belonging to the trained model. Using the information extracted from the

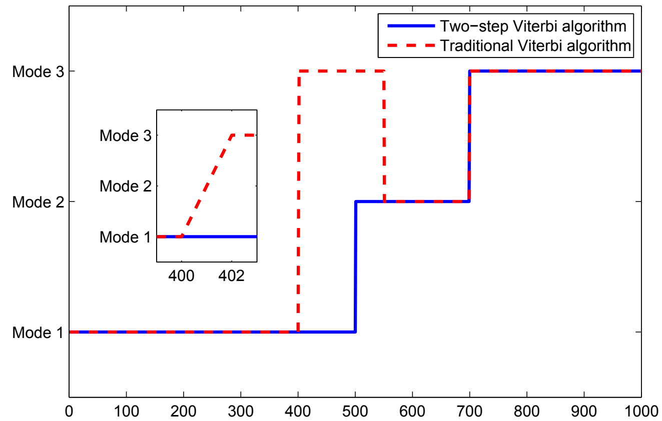


Figure 2.3: Comparison of Viterbi algorithms

two-step Viterbi algorithm, a monitoring index based on the negative of the log likelihood (NLLP) is proposed. For the case when observation probability is multivariate Gaussian distribution, NLLP takes the following form:

$$\varphi_t = -\log[\mathrm{P}[y_t|\lambda]] = \frac{M}{2}\log(2\pi) + \frac{1}{2}\log|\Omega| + \frac{1}{2}(y_t - \eta)^T\Omega^{-1}(y_t - \eta) \quad (2.12)$$

where M is the number of the process variables to be monitored, and η and Ω are the mean vector and covariance matrix of the current operating mode, respectively. Mode information is collected using the two-step Viterbi algorithm:

$$q_t = Vit_{\text{two-step}}(y_{t-1}, y_t, q_{t-1}, \lambda) \quad (2.13)$$

Assuming μ^* and Σ^* are the mean vector and covariance matrix associated with mode q_t, (2.12) can be rewritten as:

$$\varphi_t = -\log[\mathrm{P}[y_t|\lambda]] = \frac{M}{2}\log(2\pi) + \frac{1}{2}\log|\Sigma^*| + \frac{1}{2}(y_t - \mu^*)^T\Sigma^{*-1}(y_t - \mu^*) \quad (2.14)$$

A complete fault detection logic is provided in Algorithm 2. The offline step of the detection algorithm is related to the HMM model training. Depending on the nature of the process, and the number of operating modes, HMM parameters are selected based on the guidelines provided in Subsection 2.3.1.

Algorithm 2 Fault detection algorithm

Offline: *HMM modeling*
1: Set the number of HMM states equal to the number of operating modes.
2: Randomly initialize HMM parameters.
3: Use Baum-Welch algorithm to find the HMM parameters.
4: Find the threshold ($\varphi_{\text{threshold}}$) based on kernel density estimation.
Online: *Fault detection*
For every new sample y_t
5: Find the current operating mode using the two-step Viterbi algorithm: $q_t = Vit_{\text{two-step}}(y_{t-1}, y_t, q_{t-1}, \lambda)$
6: Calculate the test statistic (assuming that μ^* and Σ^* are the mean vector and covariance matrix corresponding to mode q_t):
$\varphi_t = \frac{M}{2}\log(2\pi) + \frac{1}{2}\log|\Sigma^*| + \frac{1}{2}(y_t - \mu^*)^T\Sigma^{*-1}(y_t - \mu^*)$
7: **if** $\varphi_t > \varphi_{\text{threshold}}$ **then**
8: Raise an alarm.
9: **end if**

Since the distribution of the monitoring index ($\varphi(t)$) is not known, a kernel based density estimation (KDE) procedure is used for the selection of appropriate threshold. The KDE is a non-parametric density estimation procedure, where finite data samples are used to estimate the distribution of the data [39]. The monitoring index using the normal training data is calculated, and distribution is estimated based on a Gaussian kernel function.

Fault isolation based on reconstruction

A complete process monitoring method should be able to detect the potential root causes in the event of faults. In this work, contribution plots, a popular unsupervised method for fault isolation, are used for the root cause identification. Contribution plots help in identifying variables which are pushing the monitoring index out of its control limit [62]. In [46, 72, 25, 36], contribution plots based on reconstruction were developed. The basic idea is to minimize the monitoring index by reconstructing the faulty variables using the non-faulty variables. In this chapter, a similar reconstruction approach is used for the proposed fault detection method, and contribution plots are developed.

The proposed monitoring index has the following mathematical form:

$$\varphi = \frac{M}{2}\log(2\pi) + \frac{1}{2}\log|\Sigma^*| + \frac{1}{2}(y - \mu^*)^T\Sigma^{*-1}(y - \mu^*) \tag{2.15}$$

For a given mode, the first **two** terms of (2.15) are independent of y, so the minimization of φ depends only on the last term, namely, $(y-\mu^*)^T\Sigma^{*-1}(y-\mu^*)$. Redefine this term as follows:

$$\omega = z^T G z = (y-\mu^*)^T\Sigma^{*-1}(y-\mu^*) \tag{2.16}$$

Now ω can be minimized by differentiating it with respect to the jth faulty variable and then reconstructing it based on the non-faulty variables:

$$\frac{\partial \omega}{\partial z_j} = 0 \tag{2.17}$$

Equation (2.17) can be rewritten as follows:

$$\tau_j^T G z = 0 \tag{2.18}$$

where τ_j is a column vector of size M, in which only the jth element is one and the rest are zero. For the case of multiple sensor faults (2.18) takes the following form:

$$\tau^T G z = 0 \tag{2.19}$$

where $\tau = [\tau_1 \ \tau_2 \ \cdots \ \tau_k]$ is the collection of all the τ_j's corresponding to k number of the faulty variables. Decomposition of z into faulty and non-faulty variables can be done as: $z = \Lambda z + (I - \Lambda)z$, where Λ is a diagonal matrix of size M, whose elements are 1 corresponding to faulty variables and are 0 otherwise. After decomposition, (2.19) can be written as:

$$\tau^T G \Lambda z = -\tau^T G (I - \Lambda) z \tag{2.20}$$

By letting z_k to be the collection of all the k faulty variables, and observing that $\Lambda z = \tau z_k$, the following reconstruction equation can be obtained:

$$z_k^* = -(\tau^T G \tau)^{-1}\tau^T G (\mathrm{I} - \Lambda) z \tag{2.21}$$

The reconstruction equation in terms of y takes the following form:

$$y_k^* = \mu^* - (\tau^T G \tau)^{-1}\tau^T G (\mathrm{I} - \Lambda)(y - \mu^*) \tag{2.22}$$

Algorithm 3 Fault isolation algorithm

1: **if** $\varphi > \varphi_{\text{threshold}}$ **then**
2:     Find the current operating mode using the two-step Viterbi algorithm.
3:     Set $G = \Sigma^{*-1}$, $\varphi^* = 0$, $y_k = \emptyset$ and $k = 1$
4:     **while** $\varphi^* > \varphi_{\text{threshold}}$ **do**
5:         **for** $j = 1$ to k **do**
6:             Find y_j^* and φ^* using (2.22) and (2.23), respectively.
7:             Calculate $\triangle\varphi = \varphi - \varphi^*$
8:         **end for**
9:         **if** $\varphi^* < \varphi_{\text{threshold}}$ **then**
10:             Exit **while**
11:         **else**
12:             Arrange the variables in decreasing order with respect to $\triangle\varphi$.
13:             Add the variable with maximum $\triangle\varphi$ in y_k.
14:             Set $k = k + 1$ and go back to step 4.
15:         **end if**
16:     **end while**
17: **end if**
18: Find contribution of each variable using (2.24).

Modified data after reconstruction can be found as: $y^* = \tau y_k^* + (I - \Lambda)y$, and the minimized monitoring index φ^* can be obtained as:

$$\varphi^* = \frac{M}{2}\log(2\pi) + \frac{1}{2}\log|\Sigma^*| + \frac{1}{2}(y^* - \mu^*)^T \Sigma^{*-1}(y^* - \mu^*) \tag{2.23}$$

Finally, the contribution of the k faulty variables in reducing monitoring index can be found as:

$$c = [(\tau^T \varphi^* \tau)^{0.5} \tau (y - y^*)]^2 \tag{2.24}$$

The complete fault isolation algorithm based on reconstruction is presented in Algorithm 3. This algorithm reconstructs the faulty variables in such a way that the monitoring index drops below the threshold after reconstruction, and the corresponding contribution of faulty variables is visualized through contribution plots.

2.4 Application examples

Two processes are considered to show the effectiveness of the proposed method. For these processes, modes are defined based on the setpoints, and it is assumed that these setpoints are reflective of the process operational requirements. For comparison purposes, fault scenarios are also tested using Gaussian mixture models, dynamic PCA, and multi-mode PCA. A Bayesian inference strategy is used for mode detection for the multi-mode PCA and Gaussian mixture models [69] [63].

2.4.1 A numerical example

A two-input-two-output system described by the following equation is considered:

$$y = \begin{bmatrix} 1.0 & 0.2 \\ 0.3 & 1.0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \tag{2.25}$$

where e_1 and e_2 are independent Gaussian noises $\mathcal{N}(\mu, \sigma)$, with mean $\mu = 0$ and variance $\sigma = 10^{-3}$, respectively. Depending on the inputs, the system can have one of the following three operating modes:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{cases} \text{Mode 1: } \mathcal{N}(\begin{bmatrix} 1 & 2 \end{bmatrix}, \begin{bmatrix} 1.3 & 0 \\ 0 & 1.5 \end{bmatrix}) \\ \text{Mode 2: } \mathcal{N}(\begin{bmatrix} 5 & 15 \end{bmatrix}, \begin{bmatrix} 1.3 & 0 \\ 0 & 1.5 \end{bmatrix}) \\ \text{Mode 3: } \mathcal{N}(\begin{bmatrix} 12 & 10 \end{bmatrix}, \begin{bmatrix} 1.3 & 0 \\ 0 & 1.5 \end{bmatrix}) \end{cases} \tag{2.26}$$

Mode-reachability constraints are considered by assuming that mode 1 is not directly reachable through mode 3, and vice versa. Thus mode-reachability matrix takes the following form:

$$\Psi = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{2.27}$$

A three-state HMM model with multivariate Gaussian distributions is selected based on AIC criterion, and the following faulty scenarios are considered during mode 1:

i. A bias in y_1 (magnitude of 10 units) is introduced during time 400 - 500.

ii. A bias in y_2 (magnitude of 5 units) is introduced during time 400 - 500.

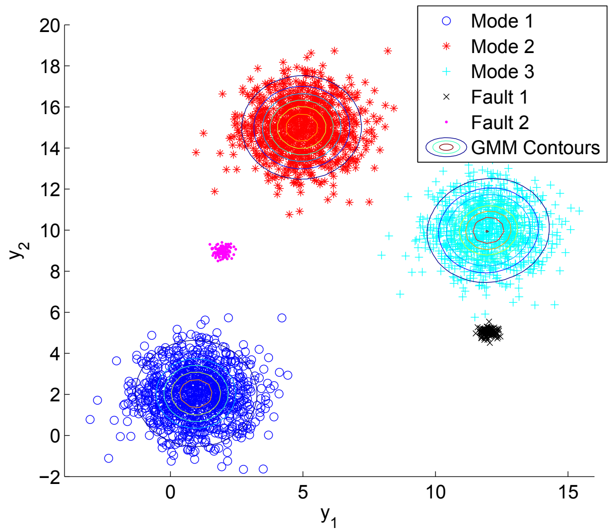


Figure 2.4: Scatter plot of data and Gaussian contours

The scatter plot of the training data, Gaussian contours, and faults are shown in Figure 2.4. It can be seen that under the first faulty scenario the system is drawn towards the constrained mode 3; whereas under the second scenario the system does not violate any of the mode-reachability constraints.

Mode detection results of the Bayesian inference strategy and the proposed two-step Viterbi algorithm for fault 1 are shown in Figure 2.5. During time t (from 400 to 500), the mode is wrongly detected as Mode 3 using the Bayesian inference strategy; whereas the two-step Viterbi algorithm, being sensitive to the mode-reachability constraints, detects the actual mode of operation.

Monitoring results are shown in Figure 2.6 and 2.7, respectively. It can be seen that under the first faulty scenario, where mode-reachability constraints are violated, only the proposed method is able to detect the fault because none of the other methods is able to capture the mode-reachability constraints; whereas for the second scenario the proposed method, FGMM, and multi-mode PCA give comparable results, since in this case no mode-reachability constraints are violated. Dynamic PCA based statistics are not able to capture the multimodality of the data at all.

Results of the fault isolation algorithm for both scenarios are shown in

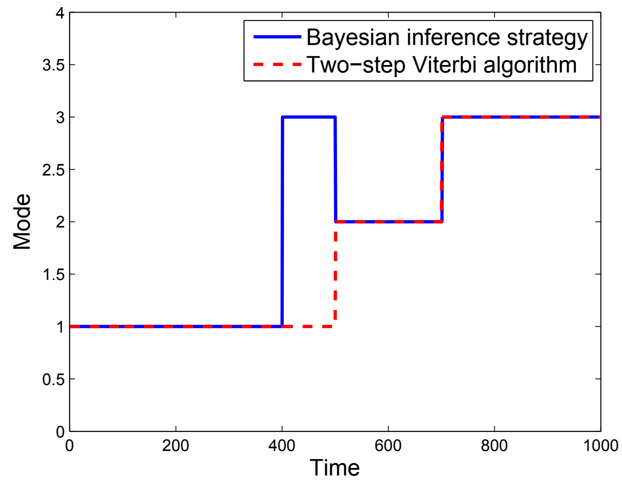
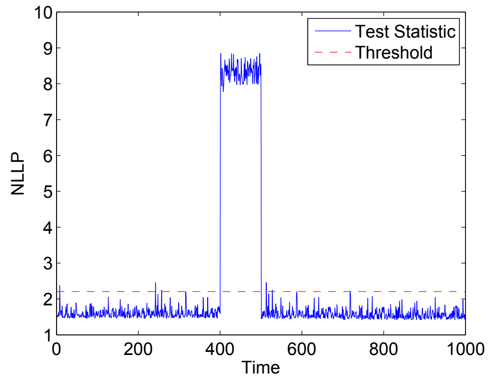


Figure 2.5: Mode detection during faulty scenario 1 (Example 1)

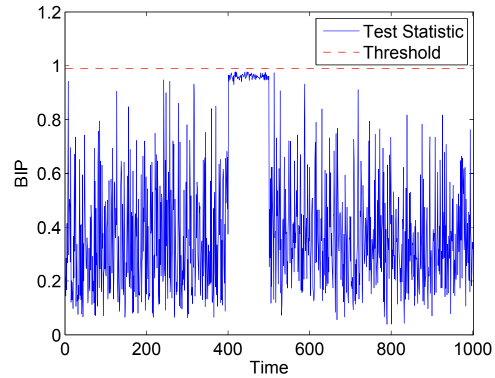
Figure 2.8a and 2.8b, respectively. The contribution plot for fault 1 shows a very high contribution from variable y_1 during time 400 - 500, which has a bias error in variable y_1. Similarly Figure 2.8b shows that variable y_2 has a major contribution (∼100%) during time 400 - 500 in pushing the monitoring index out of its threshold limit.

2.4.2 A continuous stirred tank heater

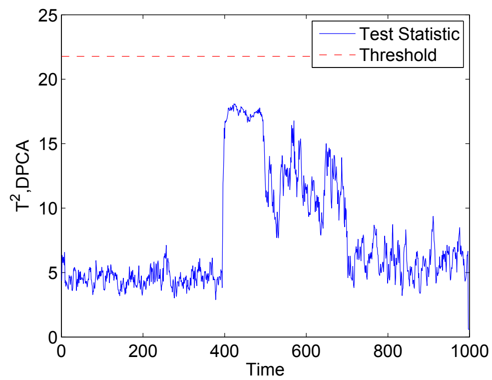
The continuous stirred tank heater (CSTH) is a common process model, whose main purpose is to maintain the flow rate and the temperature of the outlet at the required set point. To achieve this, hot and cold water streams are mixed together, and if required, heated using steam through a heating coil. Controllers are configured in the temperature, level, and flow loops of the system. The overall schematics of the process is shown in Figure 2.9. The laboratory setup of the process is present at the Department of Chemical Engineering, University of Alberta. In this example, the laboratory constraints and the Simulink model developed in [56] is used. Five operating modes are used for testing purposes. These modes are defined in terms of set points (Table 2.1) for level, temperature, and hot water valve opening [69]. These modes are defined in terms of electronic signals (mA); whereas calibration



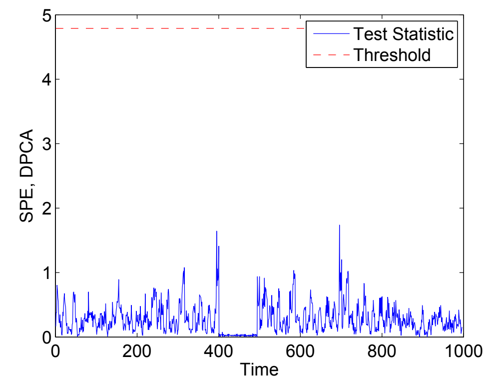
(a) Proposed method



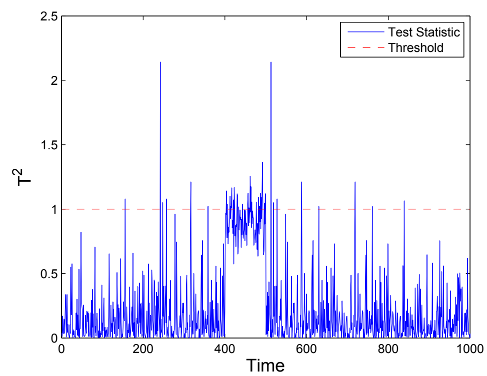
(b) FGMM



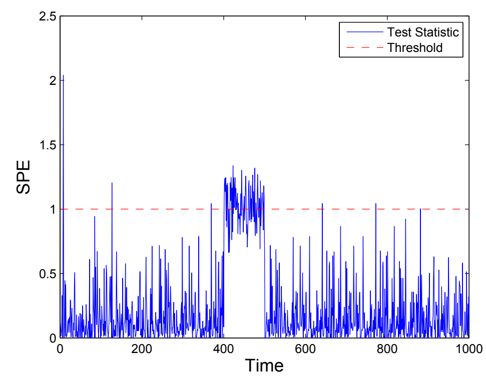
(c) DPCA (T^2)



(d) DPCA (SPE)

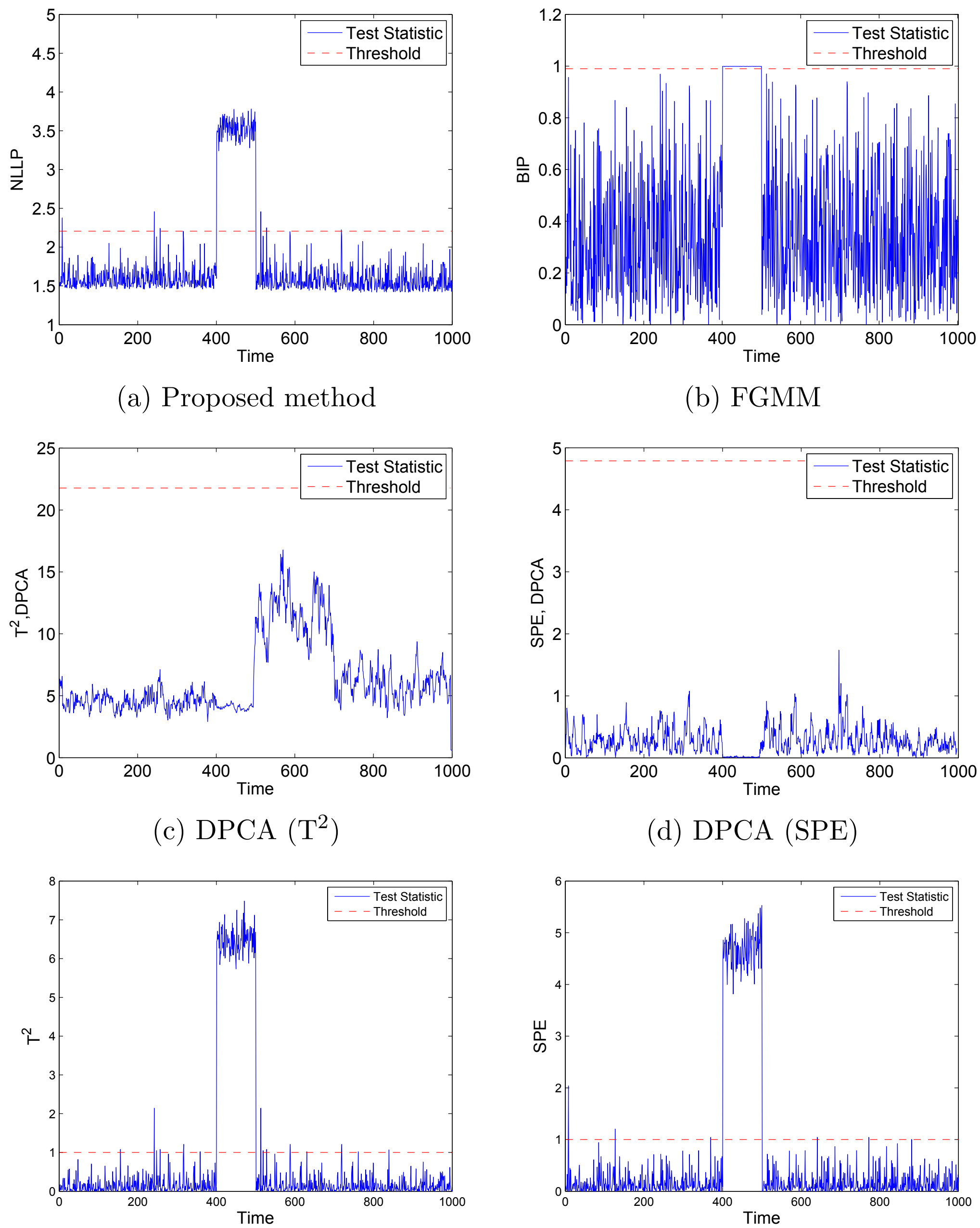


(e) Multi-mode PCA (T^2)



(f) Multi-mode PCA (SPE)

Figure 2.6: Monitoring results for fault 1 for Example 1



(a) Proposed method

(b) FGMM

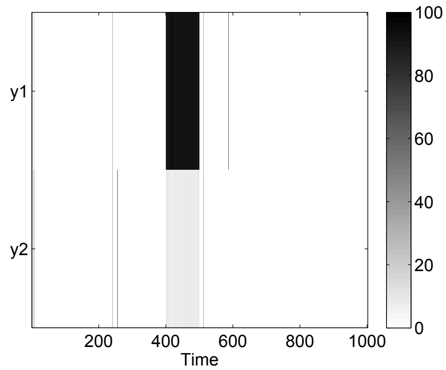
(c) DPCA (T^2)

(d) DPCA (SPE)

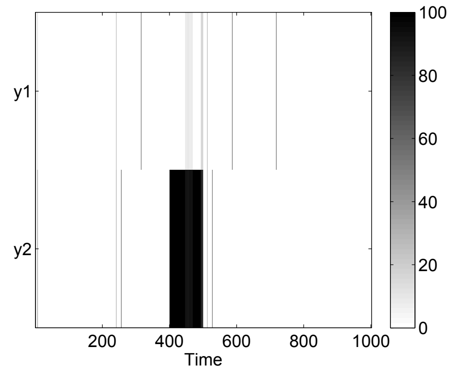
(e) Multi-mode PCA (T^2)

(f) Multi-mode PCA (SPE)

Figure 2.7: Monitoring results for fault 2 for Example 1



(a) Contribution plot (fault 1)



(b) Contribution plot (fault 2)

Figure 2.8: Fault isolation results for Example 1

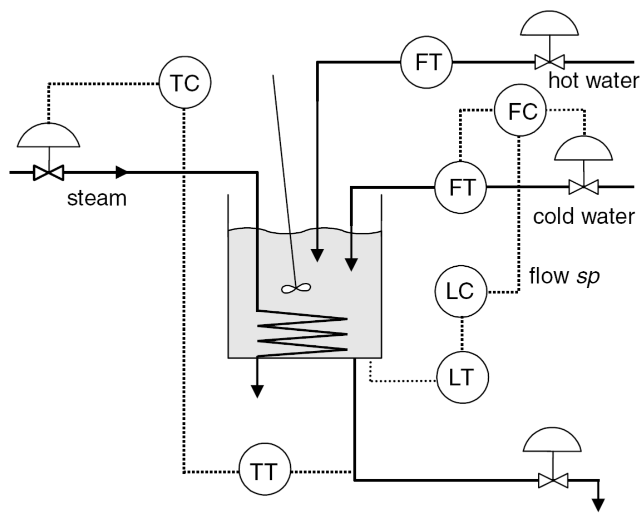


Figure 2.9: Continuous stirred tank heater [56]

information of the sensors can be found in [56], which was determined by recording measurements at several operating points of the sensors.

Some of the modes in the CSTH are constrained due to limitations on the change of the steam flow rate, and hot water valve opening. For example, hot water is one of the shared resources for the setup, and a sudden increase and decrease in hot water demand can cause disturbances for other users [56]. Thus, a direct switch from mode 5 to mode 1 (vice versa) is not recommended, because change in the hot water valve opening from 4 to 5.5 mA requires a sharp increase in the hot water flow rate (0 to 4.8 liters/min), which is not

Table 2.1: Set points for modes of the CSTH [69]

Variable	**Mode 1**	**Mode 2**	**Mode 3**	**Mode 4**	**Mode 5**
Level SP (mA)	12	16	9	12	12
Temperature SP (mA)	10.5	10.5	10.5	13.5	8
HW valve opening (mA)	5.5	5	4.5	6	4

recommended. Furthermore, since the level set point is fixed for both modes (12mA $\approx$ 18 liters), increase in the hot water valve opening can cause an overflow in the tank, as outlet flow of the tank is constant. Similarly, mode 2 is not reachable directly from mode 3, and vice versa, because of constraints on the steam flow rate, and mode 4 is also not reachable directly from mode 5, and vice versa, due to constraints on the hot water flow rate. Under these constraints, the mode-reachability matrix for the CSTH takes the following form:

$$\Psi = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{2.28}$$

From simulations it is observed that the system may suffer from constrained mode change in the event of two-variable faults; whereas for single variable faults, such scenario is not possible. The following two faulty scenarios were considered, while the system was operating in mode 5:

i. A bias error is introduced in the temperature sensor (magnitude of 3.5 mA) at time $t = 400$, and random variation (Gaussian noise) in the cold water flow rate sensor is introduced at time $t = 600$.

ii. Starting from time $t = 400$, a bias error (magnitude of 4 mA) is introduced in the level sensor.

HMM model with five states is selected based on AIC criterion. Mode detection results for the first faulty scenario, using the Bayesian inference strategy and the two-step Viterbi algorithm, are shown in Figure 2.10. From

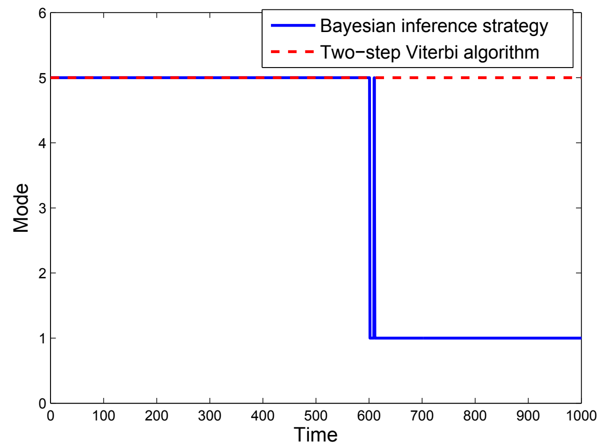
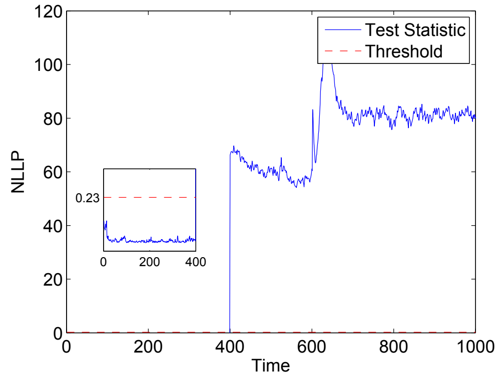


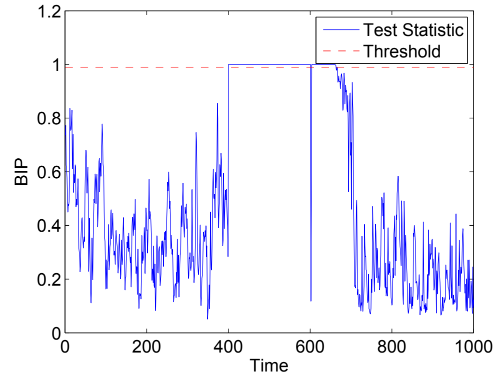
Figure 2.10: Mode detection during faulty scenario 1 (Example 2)

this figure, it can be seen that the Bayesian inference strategy is not able to capture the mode-reachability constraints, and at time $t = 600$ the true mode (mode 5) is not detected; whereas the detected mode using the proposed two-step Viterbi algorithm is always mode 5, irrespective of the fault. Monitoring results for the first faulty scenario are shown in Figure 2.11. Since in this case there is a constrained mode change due to the two-variable fault after time $t = 600$, as expected, only the proposed method is able to detect such a fault efficiently. The FGMM, dynamic PCA and multi-mode PCA based statistics show very poor results in detecting the two-variable fault. For faulty scenario 2, where only a single variable is involved and no constrained mode change occurs, the proposed method, FGMM and multi-mode PCA give comparable results, as shown in Figure 2.12. However, dynamic PCA based statistics again are not able to handle multi-mode data very efficiently.

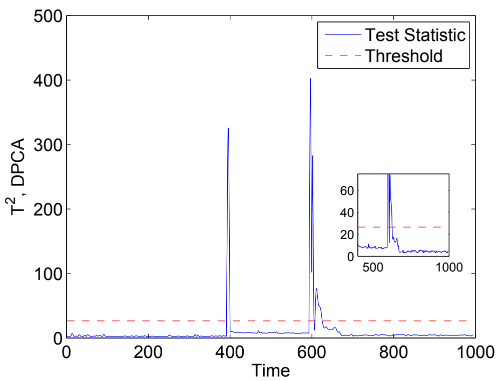
Results of the fault isolation algorithm for both faulty scenarios are shown in Figure 2.13a and 2.13b, respectively. For the first case, initially there is only one faulty variable (the temperature), but after time $t = 600$, a majority of the contribution comes from the second variable (the flow sensor). For the second case, a single faulty variable can be identified by the contribution plot shown in Figure 2.13b.



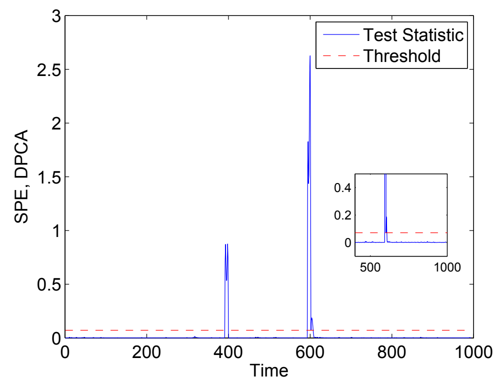
(a) Proposed method



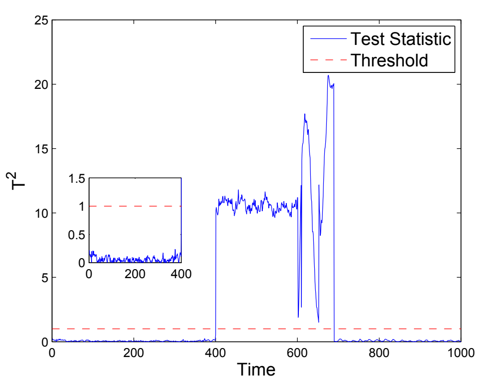
(b) FGMM



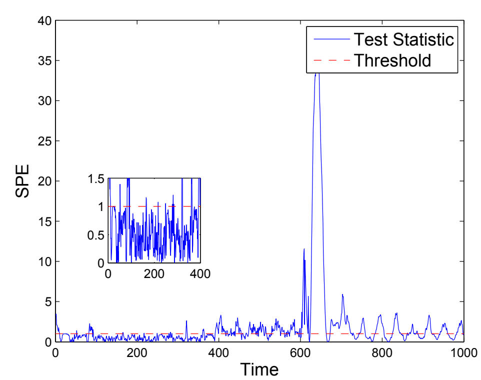
(c) DPCA (T^2)



(d) DPCA (SPE)

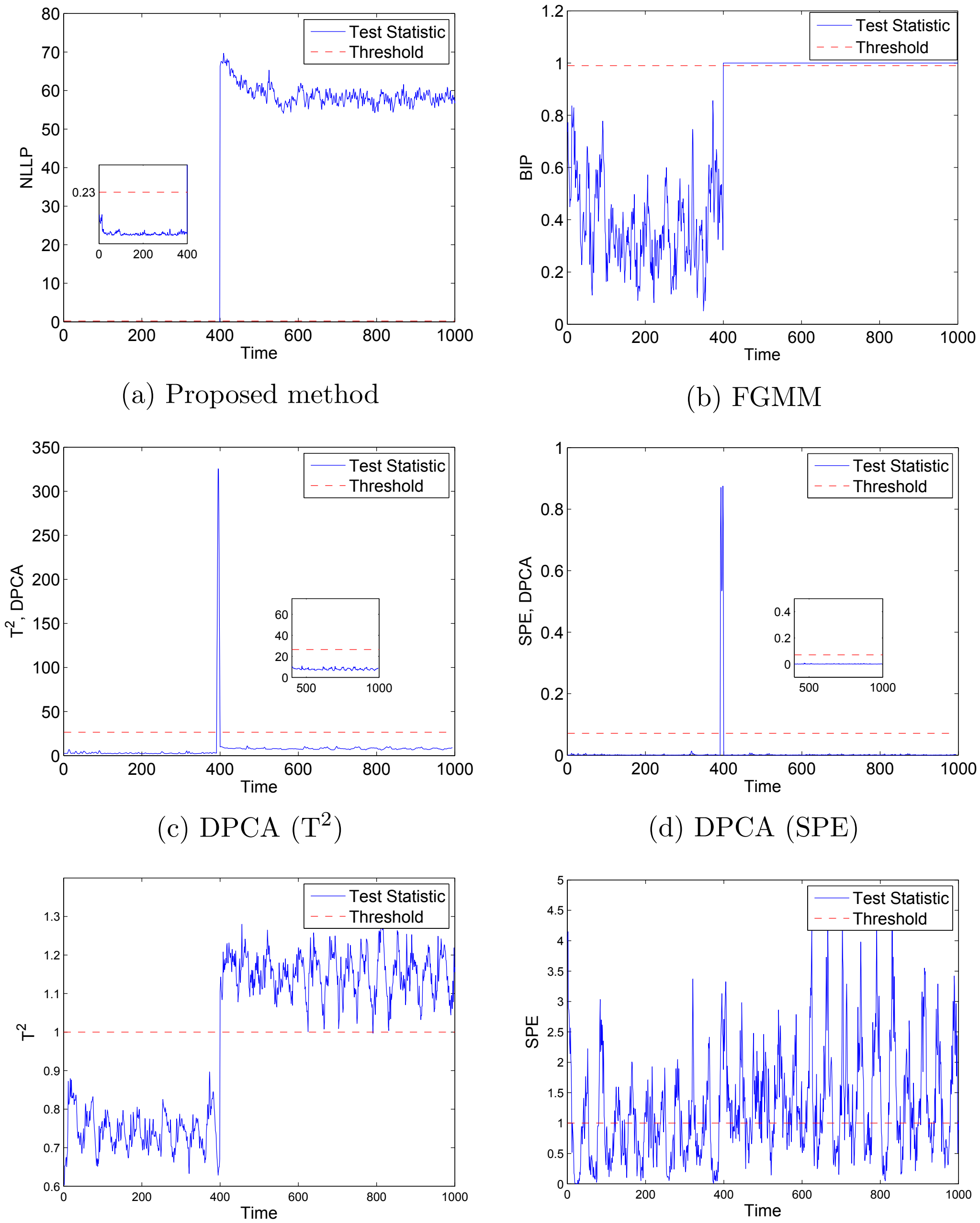


(e) Multi-mode PCA (T^2)



(f) Multi-mode PCA (SPE)

Figure 2.11: Monitoring results for fault 1 for CSTH



(a) Proposed method (b) FGMM

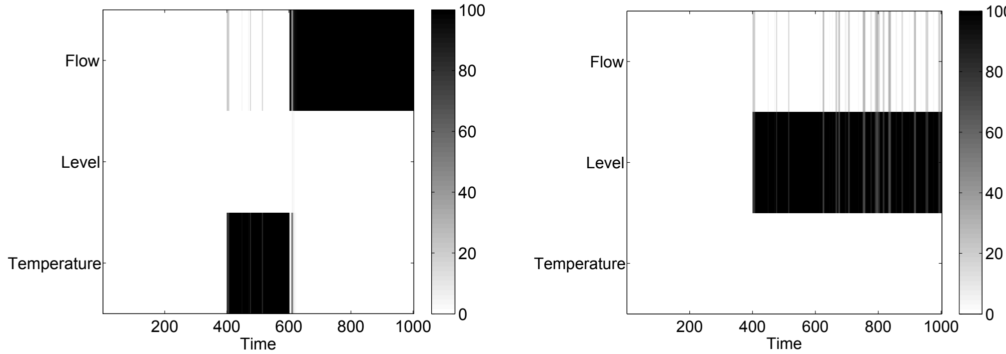
(c) DPCA (T^2) (d) DPCA (SPE)

(e) Multi-mode PCA (T^2) (f) Multi-mode PCA (SPE)

Figure 2.12: Monitoring results for fault 2 for CSTH

2.4.3 Performance comparison

Table 2.2 lists the missed and false alarm rates for the considered faulty scenarios in the above mentioned case studies. From this table it can be



(a) Contribution plot (fault 1) (b) Contribution plot (fault 2)

Figure 2.13: Fault isolation results for Example 2

Table 2.2: Missed and false alarm rates

Missed Alarm Rate (MAR)							
	Fault ID	DPCA(T^2)	DPCA(SPE)	mPCA(T^2)	mPCA(SPE)	FGMM	Proposed
Example 1	Fault 1	100%	100%	76%	52%	100%	**0%**
	Fault 2	100%	100%	**0**%	**0**%	**0%**	**0%**
Example 2	Fault 1	95%	97.67%	51.83%	28.33%	56.50%	**0%**
	Fault 2	98.30%	98.30%	0%	10%	**0%**	**0%**
False Alarm Rate (FAR)							
Example 1	-	0%	0%	1.03%	0.53%	0%	0.3%
Example 2	-	0%	1.67%	1.23%	0.75%	0%	2%

seen that in terms of missed alarm rates the proposed method outperforms FGMM, dynamic PCA and multi-mode PCA based monitoring methods for the type of faults where mode-reachability constraints are violated. For other faults, where mode-reachability constraints are not violated, performance of the proposed method is comparable to the FGMM and multi-mode PCA. Low values of false rates are obtained for both examples.

2.5 Summary

In this chapter, an HMM based multivariate alarm monitoring system was designed for multi-mode processes with mode-reachability constraints. The Viterbi algorithm was first modified to detect the operation modes in the event of faults, and a contribution plot was built by reconstruction of the faulty variables for fault isolation. Application examples showed that the proposed monitoring method can not only handle the multimodality of process data but also capture the mode switching restrictions. In the next chapter, the analysis of delay-timer configurations for multi-mode processes is carried out.

Chapter 3

Analysis and Design of Multi-mode Delay-Timers*

FOR acceptable performance of an alarm system for a multi-mode process, it is inevitable to have appropriate alarm configurations for every mode of the process. Delay-timers, being very effective in reducing false and nuisance alarms, are analyzed and designed for multi-mode processes. A hidden Markov model with Markov chain observations is used to capture the configuration of delay-timers in various modes of the process. Analytical expressions for different performance indices are derived. For the design, a particle swarm optimization based algorithm is proposed.

3.1 Overview

For multi-mode processes, acceptable alarm system performance requires that alarm configurations are updated based on operating modes. These updates may include changing the alarm limits, shelving of certain alarms, reconfiguration of filters, deadbands, or delay-timers. However, DCS alarm capabilities are generally limited to a single mode process, not well suited for processes with multiple operating modes [15]. This has resulted in compromised and less

*A version of this chapter has been published as: Afzal, M.S., & Chen, T. (2017), Analysis and Design of Multimode Delay-Timers, Chemical Engineering and Research Design, vol. 120, pp. 179-193.

efficient designs. For most of the cases, key performance indicators fall well below the standards mentioned in EEMUA and ISA 18.2 [21, 13], thus appropriate design of alarm configurations (filters, deadbands and/or delay-timers) for multi-mode processes is almost inevitable. Consequently, this chapter focuses on the design of delay-timers for multi-mode processes, because of their wide usage in industries, and effectiveness in improving the key performance indicators [30].

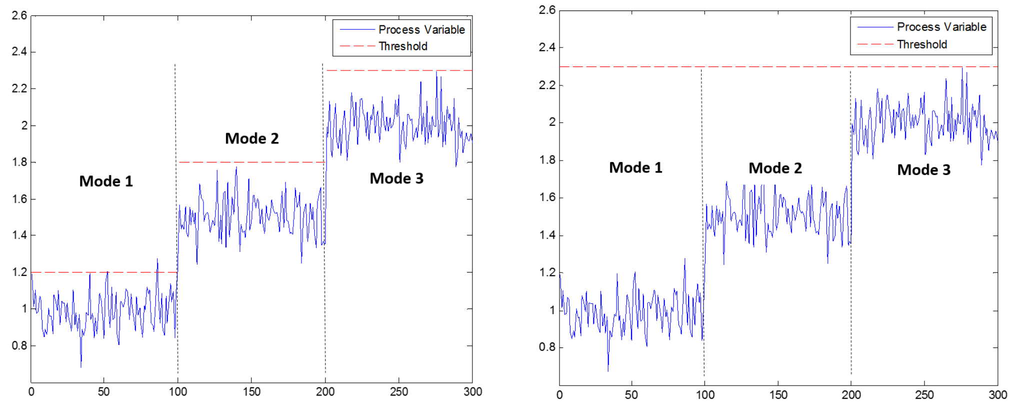
In this chapter multi-mode configuration of delay-timers is considered. Hidden Markov models with Markov chain observations are used to capture the process mode switching probabilities, and delay-timer configurations. The proposed HMM based model is used to analyze and design multi-mode delay-timers. The rest of this chapter is organized as follows. Section 3.2 provides details about the types of alarm systems for multi-mode processes, and preliminary information related to the alarm system performance measures, and hidden Markov models. In Section 3.3, a model for multi-mode delay-timers is developed. Derivation of performance indices, and design methodology is detailed in Sections 3.4 and 3.5, respectively.

3.2 Background

In this section, the types of alarm systems for multi-mode processes are described. Long term behavior of an HMM model is also discussed.

3.2.1 Types of alarm systems for multi-mode processes

For multi-mode processes, two types of alarm systems are used. In the first type, different alarm limits are configured for different modes of the process [17, 60], whereas in the second type, a single threshold setting is used for each mode of the process [67, 75, 74]. In this chapter, these two types of alarm systems are named as multi-threshold alarm systems and uni-threshold alarm systems, respectively. Typical settings of alarm thresholds in multi-threshold and uni-threshold alarm systems are shown in Figure 3.1.



(a) Multi-threshold alarm system

(b) Uni-threshold alarm system

Figure 3.1: Types of alarm systems for multi-mode processes

The problem of designing delay-timers for multi-threshold alarm system can be solved by considering each mode as a single process, and designing delay-timers for each mode separately. Graphical techniques proposed in [2, 30] for single-mode processes can be used for this purpose. For uni-threshold alarm systems, however, it is not possible to separate the design procedure into various modes due to the overlapped threshold, and to the best of authors' knowledge there is no systematic way available to design uni-threshold alarm systems.

3.2.2 Long term behavior of an HMM model

In this chapter hidden Markov models are used to model the delay-timer configurations for multi-mode processes. Under the assumption that the samples from the process measurements are independent and identically distributed, long term behavior of the model can be studied by considering the stationary distribution of the model. For an HMM model with a Markov chain of S number of states, and state transition matrix $A = [a_{i,j}]$, $1 \leq i, j \leq S$, the stationary distribution ϕ is a row vector of size $1 \times S$, and can be found as [76]:

$$\phi = \mathbf{1}(\mathbf{I}_S - A + \mathbf{U}_S)^{-1} \tag{3.1}$$

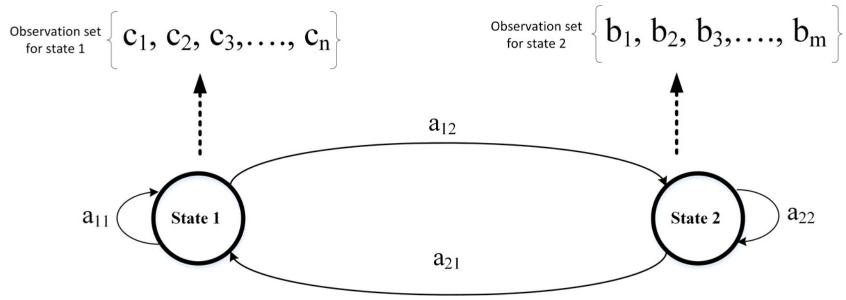


Figure 3.2: Two state HMM with discrete observations

where $\mathbf{1}$ is a row vector of ones of size S, $\mathbf{I}_S$ is a $S \times S$ identity matrix, and $\mathbf{U}_S$ is the square matrix of ones of size S. As as example, a two-state HMM model with discrete observations, is shown in Figure 3.2. State 1 has n number of discrete observations, whereas in state 2, the system has m possible observations. For a two-state HMM model, the stationary distribution ϕ has the following closed form [76]:

$$\phi = \begin{bmatrix} \frac{a_{21}}{a_{12}+a_{21}} & \frac{a_{12}}{a_{12}+a_{21}} \end{bmatrix} \tag{3.2}$$

It is worth mentioning here that in HMM based modeling of multi-mode delay-timers, states of the Markov chain correspond to the modes of the process, and the state transition matrix is used to capture the mode switching probabilities of the process.

3.3 Modeling of multi-mode delay-timers

Before developing the model of multi-mode delay-timers, first consider an example of a single-mode process variable, configured with an on delay-timer of length 3, and an off delay-timer of length 4. Let p_1 be the probability of the process variable to go above the threshold, and p_2 be the probability to fall below the threshold. For the case of uni-threshold alarm systems: $p_2 = 1 - p_1$. Under certain conditions, as described in [2, 30], this configuration can be modeled using a Markov chain process, as shown in Figure 3.3. The resulting model has three no-alarm states (NA, NA_1, and NA_2), and four alarm states

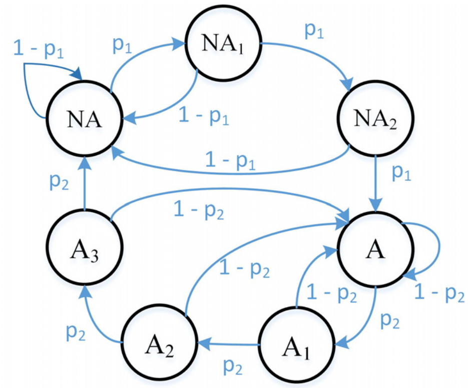


Figure 3.3: Single mode process with on delay-timer ($n = 3$), and off delay-timer ($m = 4$)

(A, A_1, A_2, A_3) corresponding to the on and off delay-timers of lengths 3 and 4, respectively. An alarm is raised if three consecutive samples of the process variable go above the threshold, and the process enters the alarm state, whereas clearance of an alarm requires four consecutive samples to fall below the threshold.

In this chapter, the idea of Markov chain based modeling of delay-timers is extended to the case of multi-mode delay-timers. Hidden Markov models with Markov chain observations are utilized for this purpose. The hidden Markov chain, in the HMM model, captures the mode switching probabilities of the process, and mode based delay-timer configurations are modeled using Markov chain observations. Like standard hidden Markov models, HMM with Markov chain observations is also a doubly stochastic process; however in the latter case, observation at any time t depends on not only the present state, but also the past observation at time $t-1$, i.e.

$$b_i(y_t|y_{t-1}) = \mathbb{P}(\text{observation} = y_t \mid q_t = i,\ y_{t-1}) \tag{3.3}$$

This allows us to incorporate delay-timers in the HMM model. For illustration purposes, consider an example of a two-mode process ($Q = 2$), with the following mode switching probability matrix:

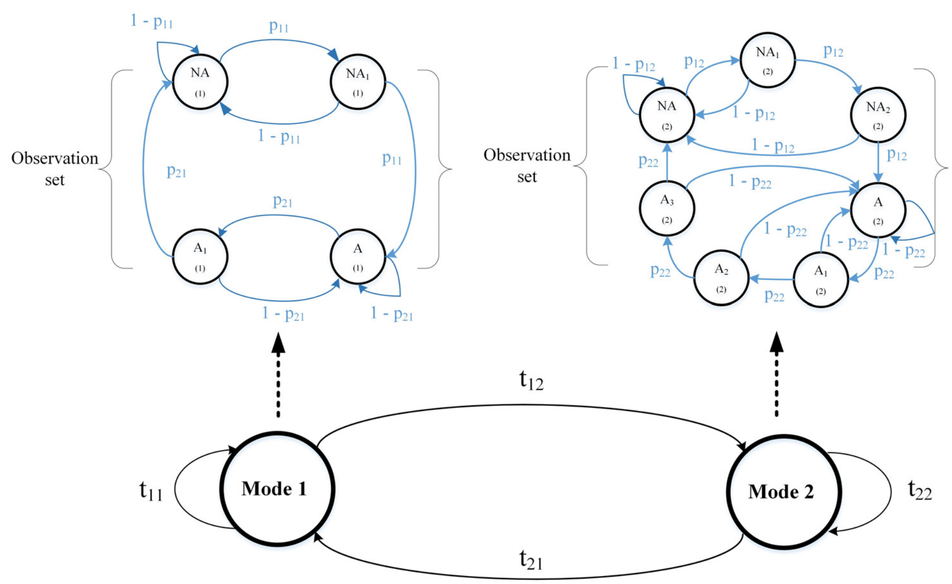


Figure 3.4: Two-mode process (Mode 1: $n_1 = m_1 = 2$, and Mode 2: $n_2 = 3$ $m_2 = 4$)

$$T = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \tag{3.4}$$

where t_{ij} $(1 \leq i, j \leq 2)$ represents the probability of the process to switch from mode i to mode j. Let p_{1i} be the probability of the process variable to go above the threshold in mode i $(1 \leq i \leq 2)$, and p_{2i} is the probability of the process variable to fall below the threshold in the i^{th} mode. If mode 1 of the process is configured with on and off delay-timer of length 2 ($n_1 = m_1 = 2$), and the mode 2 has an on delay-timer of length 3 ($n_2 = 3$), and an off delay-timer of length 4 ($m_2 = 4$), then a two state HMM with Markov chain observations based model can be developed for this configuration, which is shown in Figure 3.4. For the first mode observation set is a Markov chain with two no-alarm states ($\text{NA}_{(1)}$, $\text{NA}_{1(1)}$), representing the on delay-timer, and two alarm states ($\text{A}_{(1)}$, $\text{A}_{1(1)}$), corresponding to the off delay-timer. Similarly, observation set in mode 2 consists of three no-alarm states to model on delay-timer of length 3, and four alarm states for an off delay-timer of length 4.

Like mode switching probability matrix, transition matrix for states of delay-timers (NA's and A's) can also be defined. It captures the probabilities for switching states in observation sets. For the considered two mode process, it can be written as:

$$T_{O_n} = \begin{bmatrix} T_{O_n(1)} & \mathbf{0} \\ \mathbf{0} & T_{O_n(2)} \end{bmatrix}_{11\times 11} \tag{3.5}$$

where

$$T_{O_n(1)} = \begin{bmatrix} 1-p_{11} & p_{11} & 0 & 0 \\ 1-p_{11} & 0 & p_{11} & 0 \\ 0 & 0 & 1-p_{21} & p_{21} \\ p_{21} & 0 & 1-p_{21} & 0 \end{bmatrix},\ T_{O_n(2)} = \begin{bmatrix} 1-p_{12} & p_{12} & 0 & 0 & 0 & 0 & 0 \\ 1-p_{12} & 0 & p_{12} & 0 & 0 & 0 & 0 \\ 1-p_{12} & 0 & 0 & p_{12} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-p_{22} & p_{22} & 0 & 0 \\ 0 & 0 & 0 & 1-p_{22} & 0 & p_{22} & 0 \\ 0 & 0 & 0 & 1-p_{22} & 0 & 0 & p_{22} \\ p_{22} & 0 & 0 & 1-p_{22} & 0 & 0 & 0 \end{bmatrix} \tag{3.6}$$

Steady state behavior of the model can be defined using two stationary distributions: one for the mode switching process, and the other for observation sets. Let δ be the stationary distribution for the mode switching process, and π_n be the stationary distribution of the observation sets, then from (3.2) and (3.1), δ and π_n can be written as:

$$\delta = \mathbb{P}\left[\text{Mode 1 \ Mode 2}\right] = \left[\frac{t_{21}}{t_{12}+t_{21}} \quad \frac{t_{12}}{t_{12}+t_{21}}\right] \tag{3.7}$$

$$\pi_n = \mathbb{P}\left[\mathrm{NA}_{(1)}\ \mathrm{NA}_{1(1)}\ \mathrm{A}_{(1)}\ \mathrm{A}_{1(1)}\ \mathrm{NA}_{(2)}\ \mathrm{NA}_{1(2)}\ \mathrm{NA}_{2(2)}\ \mathrm{A}_{(2)}\ \mathrm{A}_{1(2)}\ \mathrm{A}_{2(2)}\ \mathrm{A}_{3(2)}\right] \tag{3.8}$$

$$\pi_n = \mathbf{1}(\mathbf{I}_{11} - T_{O_n} + \mathbf{U}_{11})^{-1} \tag{3.9}$$

where $\mathbf{I}_{11}$ is an identity matrix of size 11, and $\mathbf{U}_{11}$ is the square matrix of ones of size 11.

For the uni-threshold alarm system, in addition to normal operating modes, there is one abnormal mode, as well, and the process can enter the abnormal mode from any of the normal operating modes. For the considered process, if q_2 is the probability that the process variable remains above the threshold in the abnormal mode, and q_1 is the probability for the process to fall below the threshold, then the transition matrix ($T_{O_{ab}}$) for the states in the observation sets, during the abnormal mode, can be written as:

$$T_{O_{ab}} = \begin{bmatrix} T_{O_{ab}(1)} & \mathbf{0} \\ \mathbf{0} & T_{O_{ab}(2)} \end{bmatrix}_{11\times 11} \tag{3.10}$$

where

$$T_{O_{ab}(1)} = \begin{bmatrix} 1-q_2 & q_2 & 0 & 0 \\ 1-q_2 & 0 & q_2 & 0 \\ 0 & 0 & 1-q_1 & q_1 \\ q_1 & 0 & 1-q_1 & 0 \end{bmatrix}, \quad T_{O_{ab}(2)} = \begin{bmatrix} 1-q_2 & q_2 & 0 & 0 & 0 & 0 & 0 \\ 1-q_2 & 0 & q_2 & 0 & 0 & 0 & 0 \\ 1-q_2 & 0 & 0 & q_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-q_1 & q_1 & 0 & 0 \\ 0 & 0 & 0 & 1-q_1 & 0 & q_1 & 0 \\ 0 & 0 & 0 & 1-q_1 & 0 & 0 & q_1 \\ q_1 & 0 & 0 & 1-q_1 & 0 & 0 & 0 \end{bmatrix} \tag{3.11}$$

Steady state behavior of the mode switching process is still defined using (3.7), however for the observation sets, steady state distribution takes the following form:

$$\pi_{ab} = \mathbf{1}(\mathbf{I}_{11} - T_{O_{ab}} + \mathbf{U}_{11})^{-1} \tag{3.12}$$

Eqs. (3.5), (3.7), (3.9), (3.10), and (3.12) completely defines the delay-timer configurations for the considered two-mode process. Now consider a more general process with Q number of modes, and the following mode switching matrix:

$$T = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1Q} \\ t_{21} & t_{22} & \dots & t_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ t_{Q1} & t_{Q2} & \dots & t_{QQ} \end{bmatrix} \tag{3.13}$$

If the i^{th} mode ($1 \le i \le Q$) is configured with an on delay-timer of length n_i, and an off delay-timer of length m_i, then the resulting HMM based model is shown in Figure 3.5. State transition matrix for the observation sets can be written as:

$$T_{O_n} = \begin{bmatrix} T_{O_n(1)} & & & & & \\ & T_{O_n(2)} & & & & \mathbf{0} \\ & & \ddots & & & \\ & \mathbf{0} & & T_{O_n(i)} & & \\ & & & & \ddots & \\ & & & & & T_{O_n(Q)} \end{bmatrix} \tag{3.14}$$

where T_{O_n} has a dimension of $(n_1 + ... + n_Q + m_1 + ... + m_Q) \times (n_1 + ... + n_Q + m_1 + ... + m_Q)$. $T_{O_n(i)}$ is the transition matrix of the observation set in the i^{th} mode of the process, and is given by:

$$T_{O_n(i)} = \begin{bmatrix} 1-p_{1i} & p_{1i} & 0 & \dots & 0 & & & & & \\ 1-p_{1i} & 0 & p_{1i} & \dots & 0 & & & & & \\ \vdots & \vdots & \vdots & \ddots & \vdots & & & \mathbf{0}_{n_i \times (m_i-1)} & & \\ 1-p_{1i} & 0 & 0 & \dots & p_{1i} & & & & & \\ & & & & 1-p_{2i} & p_{2i} & 0 & & \dots & 0 \\ & & & & 1-p_{2i} & 0 & p_{2i} & & \dots & 0 \\ & & \mathbf{0}_{(m_i-1) \times n_i} & & \vdots & \vdots & \vdots & & \ddots & \vdots \\ & & & & 1-p_{2i} & 0 & 0 & & \dots & p_{2i} \\ p_{2i} & 0 & 0 & \dots & 1-p_{2i} & 0 & 0 & & \dots & 0 \end{bmatrix}_{(n_i+m_i)\times(n_i+m_i)} \tag{3.15}$$

Let δ be the stationary distribution of the hidden Markov chain, capturing the mode switching probabilities of the process, and it can be written as:

$$\delta = \mathbb{P}\left[\text{Mode 1 Mode 2 ... Mode Q}\right] = \mathbf{1}(\mathbf{I}_Q - T + \mathbf{U}_Q)^{-1} \tag{3.16}$$

where $\mathbf{I}_Q$ is the $Q \times Q$ identity matrix, and $\mathbf{U}_Q$ is a square matrix of ones of size Q. The stationary distribution of the observation sets (π_n) in normal modes takes the following form:

$$\pi_n = \left[\pi_{n(1)}\ \pi_{n(2)}\ ...\ \pi_{n(i)}\ ...\ \pi_{n(Q)}\right] \tag{3.17}$$

where π_n is a collection of stationary distributions of observations for each mode of operation, and the i^{th} element of π_n is defined as:

$$\pi_{n(i)} = \mathbb{P}\left[\text{NA}_{(i)}\ \text{NA}_{1(i)}\ ...\ \text{NA}_{n_i-1(i)}\ \text{A}_{(i)}\ \text{A}_{1(i)}\ ...\ \text{A}_{m_i-1(i)}\right] = \mathbf{1}(\mathbf{I}_{(n_i+m_i)} - T_{O_n(i)} + \mathbf{U}_{n_i+m_i})^{-1} \tag{3.18}$$

where $\mathbf{I}_{(n_i+m_i)}$ is a $(n_i + m_i) \times (n_i + m_i)$ identity matrix, and $\mathbf{U}_{(n_i+m_i)}$ is the square matrix of ones of size (n_i+m_i). After some simplifications $\pi_{n(i)}$ reduces to the following row vector:

$$\pi_{n(i)} = \frac{\left[p_{2i}^{m_i}\ p_{1i}p_{2i}^{m_i}\ ...\ p_{1i}^{n_i-1}p_{2i}^{m_i}\ p_{1i}^{n_i}\ p_{2i}p_{1i}^{n_i}\ ...\ p_{2i}^{m_i-1}p_{1i}^{n_i}\right]}{p_{2i}^{m_i}\sum_{j=0}^{n_i-1}p_{1i}^{j} + p_{1i}^{n_i}\sum_{k=0}^{m_i-1}p_{2i}^{k}}, \quad (1 \le i \le Q) \tag{3.19}$$

So far it is assumed that the process is operating under normal modes. However, it is possible for the process to jump into the abnormal mode from any of the normal modes. In this case, a model similar to the normal case can be developed. Except that in this model, probability distributions of raising and clearing of an alarm depends on the abnormal distribution of the process. If it is assumed that q_1 and q_2 are the probabilities for the process variable to fall below and above the threshold, respectively, and the process enters the abnormal mode for the i^{th} normal mode, then the state transition matrix of the observation set can be written as:

$$T_{O_{ab}(i)} = \begin{bmatrix} 1-q_2 & q_2 & 0 & \dots & 0 & & & & & \\ 1-q_2 & 0 & q_2 & \dots & 0 & & & & & \\ \vdots & \vdots & \vdots & \ddots & \vdots & & & 0_{n_i\times(m_i-1)} & & \\ 1-q_2 & 0 & 0 & \dots & q_2 & & & & & \\ & & & & 1-q_1 & q_1 & 0 & & \dots & 0 \\ & & & & 1-q_1 & 0 & q_1 & & \dots & 0 \\ & & 0_{(m_i-1)\times n_i} & & \vdots & \vdots & \vdots & & \ddots & \vdots \\ & & & & 1-q_1 & 0 & 0 & & \dots & q_1 \\ q_1 & 0 & 0 & \dots & 1-q_1 & 0 & 0 & & \dots & 0 \end{bmatrix}_{(n_i+m_i)\times(n_i+m_i)} \tag{3.20}$$

The stationary distribution in this case takes the following form:

$$\begin{aligned} \pi_{ab(i)} &= \mathbb{P}[\mathrm{NA}_{(i)}\ \mathrm{NA}_{1(i)}\ ...\ \mathrm{NA}_{n_i-1(v)}\ \mathrm{A}_{(i)}\ \mathrm{A}_{1(i)}\ ...\ \mathrm{A}_{m_i-1(i)}] \\ &= \frac{\left[q_1^{m_i}\ q_2q_1^{m_i}\ ...\ q_2^{n_i-1}q_1^{m_i}\ q_2^{n_i}\ q_1q_2^{n_i}\ ...\ q_1^{m_i-1}q_2^{n_i}\right]}{q_1^{m_i}\sum_{j=0}^{n_i-1}q_2^{j} + q_2^{n_i}\sum_{k=0}^{m_i-1}q_1^{k}} \end{aligned} \tag{3.21}$$

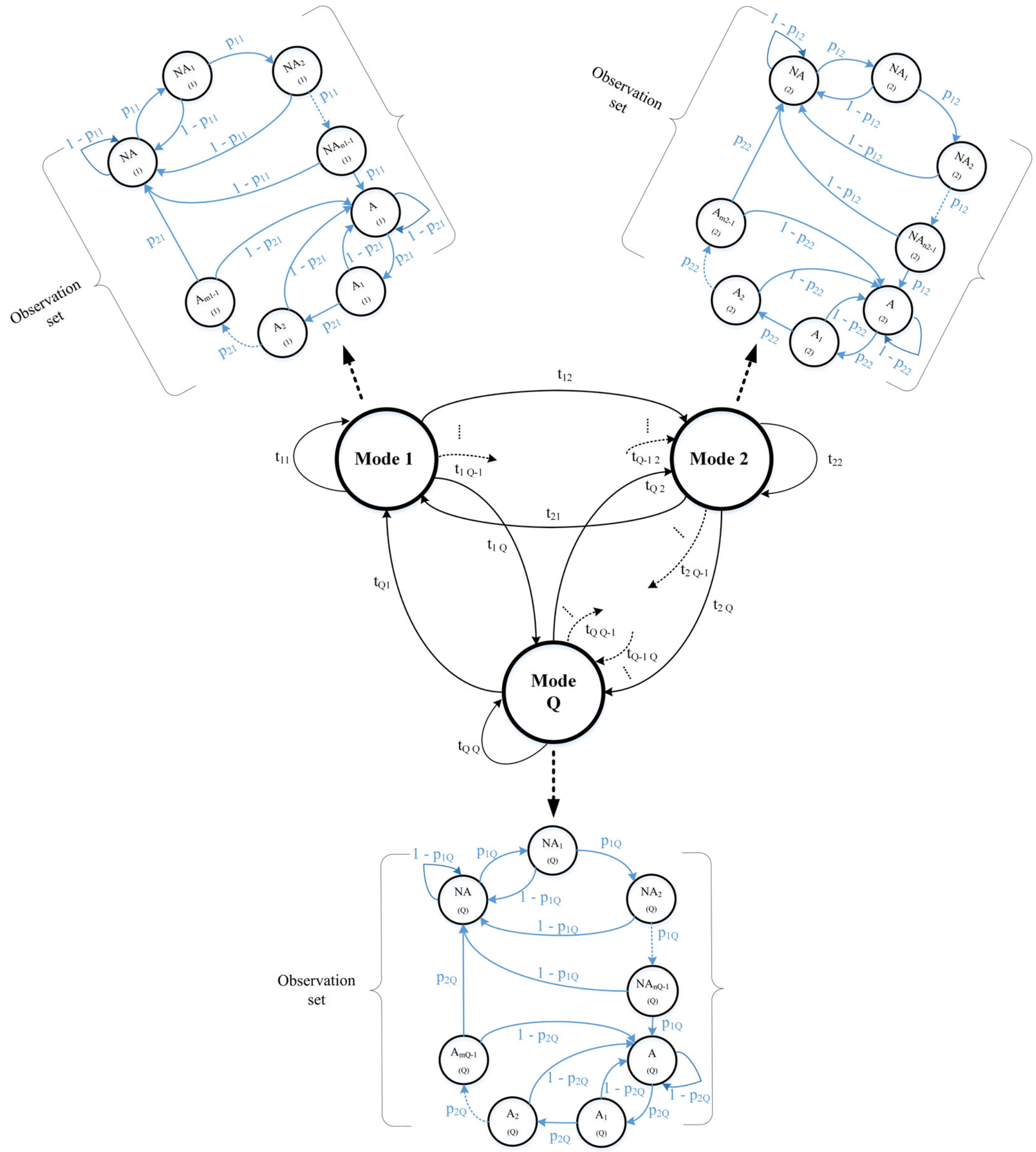


Figure 3.5: HMM model of the Q-mode process with multi-mode delay-timers

3.4 Performance assessment

For performance assessment of multi-mode delay-timers, formulas for three indices, namely, the false alarm rate, missed alarm rate, and expected detection delay, are derived. The HMM based model, developed in the Section 3.3, is utilized for this purpose.

3.4.1 False alarm rate

Give a Q-mode process with τ the threshold and p_n^i the probability density function under the normal mode i, the false alarm rate is defined as the probability of raising an alarm under the normal operation, which can be calculated as:

$$\mathbb{P}(\text{false alarm}) = \sum_{i=1}^{Q} \left[\mathbb{P}(\text{mode} = i) \int_{\tau}^{\infty} p_n^i(x)dx \right] \tag{3.22}$$

where $\mathbb{P}(\text{mode} = i)$ is the stationary distribution of the i^{th} normal mode of the process, and can be calculated using (3.16). Probability of false alarms in the i^{th} mode can be calculated by collecting all the probabilities corresponding to the alarm states from (3.19):

$$\begin{aligned} \mathbb{P}[\text{false alarm} \mid \text{mode} = i] &= \mathbb{P}[A_{(i)}] + \mathbb{P}[A_{1(i)}] + ... + \mathbb{P}[A_{m_i-1(i)}] \\ &= \frac{\sum_{j=0}^{m_i-1} p_{2i}^j p_{1i}^{n_i}}{p_{2i}^{m_i} \sum_{j=0}^{n_i-1} p_{1i}^j + p_{1i}^{n_i} \sum_{k=0}^{m_i-1} p_{2i}^k} \end{aligned} \tag{3.23}$$

Thus the overall false alarm rate for the $Q-mode$ process is given by:

$$\text{FAR} = \sum_{i=1}^{Q} \mathbb{P}(\text{mode} = i)\mathbb{P}(\text{false alarm} \mid \text{Mode} = i) \tag{3.24}$$

$$= \sum_{i=1}^{Q} \left[\delta_i \left(\frac{\sum_{j=0}^{m_i-1} p_{2i}^j p_{1i}^{n_i}}{p_{2i}^{m_i} \sum_{j=0}^{n_i-1} p_{1i}^j + p_{1i}^{n_i} \sum_{k=0}^{m_i-1} p_{2i}^k} \right) \right] \tag{3.25}$$

where δ_i is the i^{th} element of the stationary distribution vector, defined in (3.16).

3.4.2 Missed alarm rate

For a uni-threshold alarm system, where there is only a single abnormal mode, the process can enter the abnormal mode from any of the Q normal modes; thus for calculating the missed alarm rate, the stationary distribution of normal modes (δ_i) needs to be considered, as well. Let p_{ab} be the probability density function of the abnormal mode, then the missed alarm rate is the probability of failure in raising an alarm in the abnormal mode, and it can be calculated as follows:

$$\mathbb{P}(\text{missed alarm}) = \sum_{i=1}^{Q} \delta_i \int_{-\infty}^{\tau} p_{ab}(x)dx \tag{3.26}$$

where τ is the configured threshold for the uni-threshold alarm system. The expression $\int_{-\infty}^{\tau} p_{ab}(x)dx$ can be calculated by collecting all the probabilities corresponding to the no-alarm states from (3.21):

$$\begin{aligned} \mathbb{P}[\text{missed alarm} \mid \text{mode } i \rightarrow \text{abnormal}] &= \mathbb{P}[\text{NA}_{(i)}] + \mathbb{P}[\text{NA}_{1(i)}] + ... + \mathbb{P}[\text{NA}_{n_i-1(i)}] \\ &= \frac{\sum_{j=0}^{m_i-1} q_2^j q_1^{n_i}}{q_2^{m_i} \sum_{j=0}^{n_i-1} q_1^j + q_1^{n_i} \sum_{k=0}^{m_i-1} q_2^k} \end{aligned} \tag{3.27}$$

where the notation (mode $i \rightarrow$ abnormal) means that the process entered the abnormal mode from the i^{th} normal mode. The missed alarm rate of the overall process can be calculated by considering the effect of all modes, and can be written as:

$$\text{MAR} = \sum_{i=1}^{Q} \mathbb{P}(\text{mode} = i)\mathbb{P}(\text{missed alarm} \mid \text{mode } i \rightarrow \text{abnormal}) \tag{3.28}$$

$$= \sum_{i=1}^{Q} \left[\left(\frac{\sum_{j=0}^{m_i-1} q_2^j q_1^{n_i}}{q_2^{m_i} \sum_{j=0}^{n_i-1} q_1^j + q_1^{n_i} \sum_{k=0}^{m_i-1} q_2^k} \right) \delta_i \right] \tag{3.29}$$

3.4.3 Expected detection delay

The detection delay is defined as the number of time samples taken by the alarm system to generate an alarm, after the process has entered the abnormal mode. Mathematically, it is the time difference between abnormality occurrence time (t_{ab}) and alarm generation time (t_{a}):

$$\text{Detection Delay (DD)} = t_{\text{ab}} - t_{\text{a}} \tag{3.30}$$

The mean value of the detection delay is termed as expected detection delay. Based on the proposed HMM model for multi-mode delay-timers, detection delay can be defined as the time taken by the Markov chains in the observation sets to switch from no-alarm states (NA's) to alarm states (A's).

If a Q-mode process enters the abnormal mode at time t_{ab} from the i^{th} normal mode, then the probability distribution of the observation states at time t_{ab} can be written as [35]:

$$\mathbb{P}\big([\mathrm{NA}_{(i)}\ \mathrm{NA}_{1(i)}\ ...\ \mathrm{NA}_{n_i-1(i)}\ \mathrm{A}_{(i)}\ \mathrm{A}_{1(i)}\ ...\ \mathrm{A}_{m_i-1(i)}]\big) = \pi_{n(i)} T_{O_{ab}(i)} \tag{3.31}$$

where $\pi_{n(i)}$ is the stationary distribution of the i^{th} mode of the process, given by (3.19), and $T_{O_{ab}(i)}$, defined in (3.20), is the state transition matrix in the abnormal mode of the process. If $T_{\mathrm{NA}_{ab}}$ is defined as the matrix containing the switching probabilities of the states for no-alarm states only. Then, using the Chapman-Kolmogorov equation [35], it can be shown that t_s samples delay is given by:

$$\mathbb{P}(DD = t_s) = \Delta T_{O_{ab}} T_{\mathrm{NA}_{ab}}^{t_s} \left[0 \dots 0_{n_1}\ 1 \dots 1_{m_1}\ \dots\ 0 \dots 0_{n_Q}\ 1 \dots 1_{m_Q}\right]^T \tag{3.32}$$

where Δ is the stationary distribution of the overall model, and given by:

$$\Delta = [\delta_1 \pi_1\ \delta_2 \pi_2\ ...\ \delta_Q \pi_Q] \tag{3.33}$$

$T_{\mathrm{NA}_{ab}}$ is a matrix that contains switching probabilities corresponding to the no-alarm states only, and can be obtained from $T_{O_{ab}}$ by setting the elements corresponding to the alarm states to zero. It takes the following form:

$$T_{\mathrm{NA}_{ab}} = \begin{bmatrix} T_{\mathrm{NA}_{ab(1)}} & & & & & \\ & T_{\mathrm{NA}_{ab(2)}} & & & & \\ & & \ddots & & 0 & \\ & 0 & & T_{\mathrm{NA}_{ab(i)}} & & \\ & & & & \ddots & \\ & & & & & T_{\mathrm{NA}_{ab(Q)}} \end{bmatrix} \tag{3.34}$$

where

$$T_{NA_{ab}(i)} = \left[\begin{array}{ccccccccc} 1-q_2 & q_2 & 0 & \dots & 0 & & & & \\ 1-q_2 & 0 & q_2 & \dots & 0 & & & & \\ \vdots & \vdots & \vdots & \ddots & \vdots & & \multicolumn{3}{c}{0_{n_i\times(m_i-1)}} & \\ 1-q_2 & 0 & 0 & \dots & q_2 & & & & \\ & & & & 0 & 0 & 0 & \dots & 0 \\ & & & & 0 & 0 & 0 & \dots & 0 \\ & & 0_{m_i\times n_i} & & \vdots & \vdots & \vdots & \ddots & \vdots \\ & & & & 0 & 0 & 0 & \dots & 0 \\ & & & & 0 & 0 & 0 & \dots & 0 \end{array}\right]_{(n_i+m_i)\times(n_i+m_i)} \tag{3.35}$$

The expected detection delay can be calculated by taking the average of the detection delay over t_s:

$$\text{EDD} = \sum_{t_s=0}^{\infty} t_s \, \mathbb{P}(DD = t_s) \tag{3.36}$$

This results in the following equation:

$$\text{EDD} = \Delta T_{O_{ab}} T_{\text{NA}_{ab}} (\mathbf{I} - T_{\text{NA}_{ab}})^{-2} \left[0\dots0_{n_1}\ 1\dots1_{m_1}\ \dots\ 0\dots0_{n_Q}\ 1\dots1_{m_Q}\right]^T \tag{3.37}$$

Derivation for the closed form expression:

For the i^{th} mode, $T_{\text{NA}_{ab(i)}}$ can be written as:

$$T_{\text{NA}_{ab(i)}} = \left[\begin{array}{ccccccccc} 1-q_2 & q_2 & 0 & \dots & 0 & & & & \\ 1-q_2 & 0 & q_2 & \dots & 0 & & & & \\ \vdots & \vdots & \vdots & \ddots & \vdots & & \multicolumn{3}{c}{0_{n_i\times(m_i-1)}} & \\ 1-q_2 & 0 & 0 & \dots & q_2 & & & & \\ & & & & 0 & 0 & 0 & \dots & 0 \\ & & & & 0 & 0 & 0 & \dots & 0 \\ & & 0_{m_i\times n_i} & & \vdots & \vdots & \vdots & \ddots & \vdots \\ & & & & 0 & 0 & 0 & \dots & 0 \\ & & & & 0 & 0 & 0 & \dots & 0 \end{array}\right]_{(n_i+m_i)\times(n_i+m_i)} \tag{3.38}$$

$(\mathbf{I} - T_{\text{NA}_{ab}})^{-1}$ is a diagonal matrix:

$$
(\mathbf{I} - T_{\mathrm{NA}_{ab}})^{-1} = \begin{bmatrix} S_1 & & & & & \\ & S_2 & & & & \\ & & \ddots & & \mathbf{0} & \\ & \mathbf{0} & & S_i & & \\ & & & & \ddots & \\ & & & & & S_Q \end{bmatrix} \tag{3.39}
$$

where S_i takes the following form:

$$
S_i = \begin{bmatrix}
\frac{1}{q_2^{n_i}} & \frac{1}{q_2^{n_i-1}} & \frac{1}{q_2^{n_i-2}} & \dots & \frac{1}{q_2} & 1 & 0 & \dots & 0 \\
\frac{1-q_2^{n_i-1}}{q_2^{n_i}} & \frac{1}{q_2^{n_i-1}} & \frac{1}{q_2^{n_i-2}} & \dots & \frac{1}{q_2} & 1 & 0 & \dots & 0 \\
\frac{1-q_2^{n_i-2}}{q_2^{n_i}} & \frac{1-q_2^{n_i-2}}{q_2^{n_i}} & \frac{1}{q_2^{n_i-2}} & \dots & \frac{1}{q_2} & 1 & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\
\frac{1-q_2}{q_2^{n_i}} & \frac{1-q_2}{q_2^{n_i-1}} & \frac{1-q_2}{q_2^{n_i-2}} & \dots & \frac{1}{q_2} & 1 & 0 & \dots & 0 \\
0 & 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\
0 & 0 & 0 & \dots & 0 & 0 & 1 & \dots & 0 \\
\vdots & \vdots & \vdots & \dots & \vdots & \vdots & & \ddots & \\
0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1
\end{bmatrix} \tag{3.40}
$$

Let $t = (\mathbf{I} - T_{\mathrm{NA}_{ab}})^{-2} \left[0 \dots 0_{n_1} \; 1 \dots 1_{m_1} \; \dots \; 0 \dots 0_{n_Q} \; 1 \dots 1_{m_Q}\right]^T$ be an intermediate term, then it can be shown that:

$$
t = (\mathbf{I} - T_{\mathrm{NA}_{ab}})^{-1} \left[1 \dots 1_{n_1} \; 1 \dots 1_{m_1} \; \dots \; 1 \dots 1_{n_Q} \; 1 \dots 1_{m_Q}\right]^T \tag{3.41}
$$

After substituting $T_{\mathrm{NA}_{ab}}$ in (3.41), some simplifications results in the following:

$$
\begin{aligned}
t = [\tfrac{\sum_{i=0}^{n_1-1} q_2^i + q_2^{n_1}}{q_2^{n_1}} \quad \dots \quad \tfrac{\sum_{i=0}^{n_1-n_1} q_2^i + q_2^{n_1}}{q_2^{n_1}} \quad 1 \quad \dots \quad 1_{m_1} \quad \dots \quad \dots \\
\tfrac{\sum_{i=0}^{n_Q-1} q_2^i + q_2^{n_Q}}{q_2^{n_Q}} \quad \dots \quad \tfrac{\sum_{i=0}^{n_Q-n_Q} q_2^i + q_2^{n_Q}}{q_2^{n_Q}} \quad 1 \quad \dots \quad 1_{m_Q}]^T
\end{aligned} \tag{3.42}
$$

Multiplication of $T_{\mathrm{NA}_{ab}}$ with t results in the following:

$$
\begin{aligned}
T_{\mathrm{NA}_{ab}} t = [\tfrac{\sum_{i=0}^{n_1-1} q_2^i}{q_2^{n_1}} \quad \dots \quad \tfrac{\sum_{i=0}^{n_1-n_1} q_2^i}{q_2^{n_1}} \quad 0 \quad \dots \quad 0_{m_1} \quad \dots \\
\tfrac{\sum_{i=0}^{n_Q-1} q_2^i}{q_2^{n_Q}} \quad \dots \quad \tfrac{\sum_{i=0}^{n_Q-n_Q} q_2^i}{q_2^{n_Q}} \quad 0 \quad \dots \quad 0_{m_Q}]^T
\end{aligned} \tag{3.43}
$$

Post multiplication of $T_{\text{NA}_{ab}}t$ with $T_{O_{ab}}$ results in the following vector:

$$T_{O_{ab}}T_{\text{NA}_{ab}}t = [\frac{\sum_{i=0}^{n_1-1} q_2^i - q_2^{n_1}}{q_2^{n_1}} \quad \dots \quad \frac{\sum_{i=0}^{n_1-n_1} q_2^i - q_2^{n_1}}{q_2^{n_1}} \quad 0 \quad \dots \quad 0_{m_1} \quad \dots \\ \frac{\sum_{i=0}^{n_Q-1} q_2^i - q_2^{n_Q}}{q_2^{n_Q}} \quad \dots \quad \frac{\sum_{i=0}^{n_Q-n_Q} q_2^i - q_2^{n_Q}}{q_2^{n_Q}} \quad 0 \quad \dots \quad 0_{m_Q}]^T \tag{3.44}$$

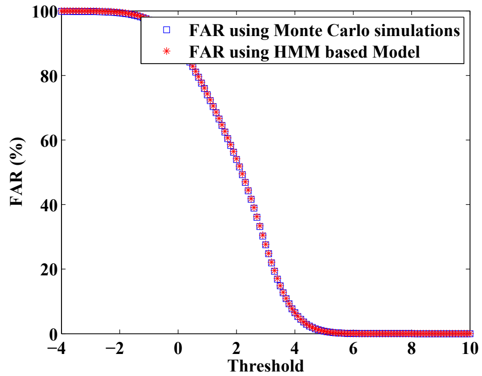
Finally post multiplication of Δ with $T_{O_{ab}}T_{\text{NA}_{ab}}t$ results in the close form expression for EDD:

$$\text{EDD} = \sum_{i=1}^{Q} \frac{\delta_i p_{2i}^{m_i-1}\left(p_{1i}^{n_i} q_1 \sum_{r=0}^{n_i-1} q_2^r + p_{2i}\left(\sum_{j=0}^{n_i-1} p_{1i}^j \sum_{k=0}^{n_i-j-1} q_2^k - q_2^{n_i} \sum_{r=0}^{n_i-1} p_{1i}^r\right)\right)}{q_2^{n_i}\left(p_{2i}^{m_i} \sum_{r=0}^{n_i-1} p_{1i}^r + p_{1i}^{n_i} \sum_{r=0}^{m_i-1} p_{2i}^r\right)} \tag{3.45}$$

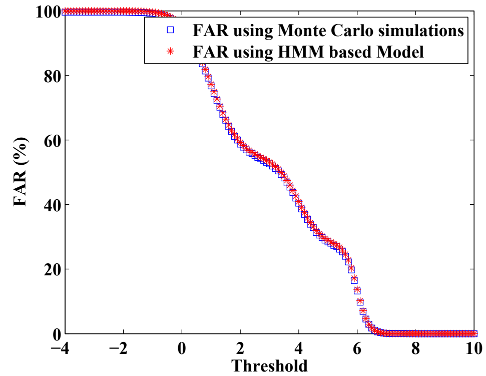
This completes the derivation.

3.4.4 Simulation verification

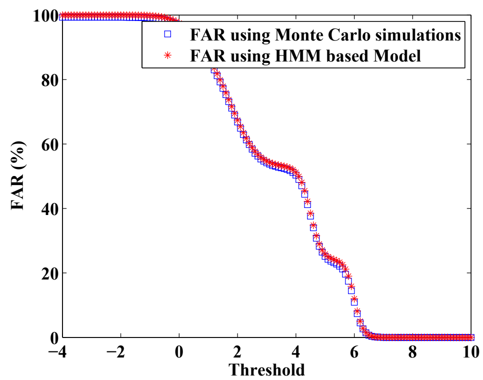
Monte Carlo simulations are carried out to verify the expressions derived for FAR, MAR, and EDD. Processes with different numbers of modes and distributions are considered for the simulation purposes. Gaussian distributions with different means and standard deviations are considered for two and three mode processes; whereas for four mode process, Gamma distributions with different scaling and shaping parameter are considered. To simulate the expected detection delay, the mode of the process is switched from any of the normal modes to the abnormal mode at time t_{ab}, and the expected value of the detection delay is calculated by performing 3000 simulations. Figure 3.6 shows the simulations results of the false alarm rate for different processes. From this figure it can be seen that the false alarm rate curves generated using (3.25), and Monte Carlo simulations are very close. Similarly, curves obtained using analytical expressions for MAR and EDD are very close to the ones generated using Monte Carlo simulations, as shown in Figures 3.7 and 3.8, respectively. This validates the proposed formulas for multi-mode delay-timers, derived using HMM based models.



(a) False alarm rate of a 2 mode process ($n_1 = m_1 = 2,\ n_2 = m_2 = 3$)



(b) False alarm rate of a 3 mode process ($n_1 = m_1 = 2,\ n_2 = m_2 = 3,\ n_3 = m_3 = 4$)

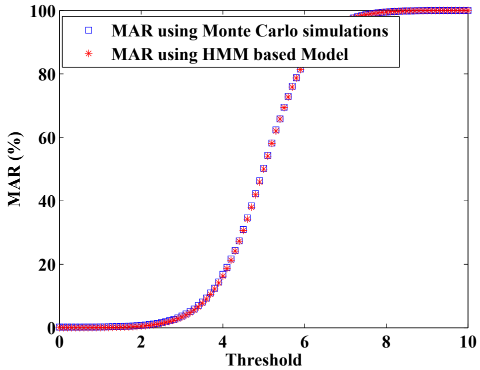


(c) False alarm rate of a 4 mode process ($n_1 = m_1 = 2,\ n_2 = m_2 = 3,\ n_3 = m_3 = 4,\ n_4 = m_4 = 5$)

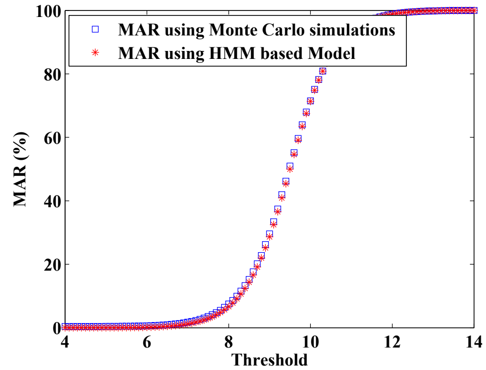
Figure 3.6: False alarm rates of multi-mode delay-timers

3.5 Design of multi-mode delay-timers

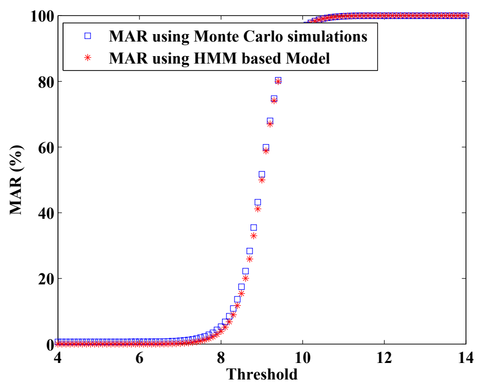
The authors in [2] have proposed an ROC curve based method for single-mode processes, while satisfying upper limits on FAR, MAR, and EDD. For the case of multi-mode processes it is not possible to adopt a similar technique, due to the increased number of design parameters. Consequently in this chapter, a numerical optimization based technique is proposed for designing multi-mode delay-timers.



(a) Missed alarm rate of a 2 mode process $(n_1 = m_1 = 2,\ n_2 = m_2 = 3)$



(b) Missed alarm rate of a 3 mode process $(n_1 = m_1 = 2,\ n_2 = m_2 = 3,\ n_3 = m_3 = 4)$

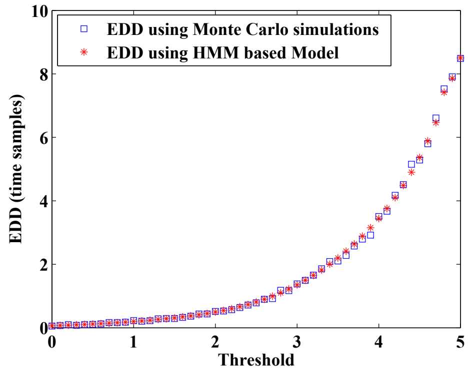


(c) Missed alarm rate of a 4 mode process $(n_1 = m_1 = 2,\ n_2 = m_2 = 3,\ n_3 = m_3 = 4,\ n_4 = m_4 = 5)$

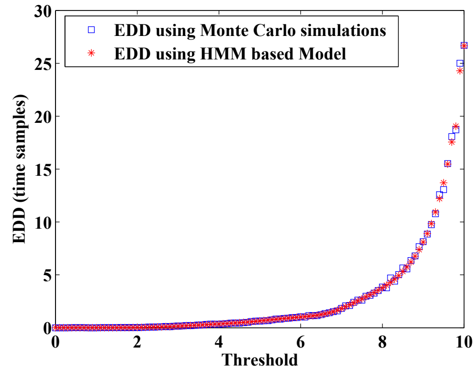
Figure 3.7: Missed alarm rates of multi-mode delay-timers

3.5.1 Problem formulation

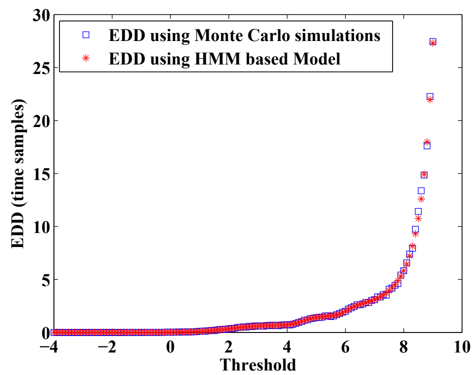
The aim of the design problem is to set threshold (τ), and appropriate lengths of delay-timers for various modes of the process, while upper limits on the FAR, MAR, and EDD are satisfied. Let $x = \{\tau, n_1, ..., n_Q, m_1, ..., m_Q\}$ be the collection of all design parameters, then the design problem can be written as:



(a) EDD of a 2 mode process ($n_1 = m_1 = 2,\ n_2 = m_2 = 3$)



(b) EDD of a 3 mode process ($n_1 = m_1 = 2,\ n_2 = m_2 = 3,\ n_3 = m_3 = 4$)



(c) EDD of a 4 mode process ($n_1 = m_1 = 2,\ n_2 = m_2 = 3,\ n_3 = m_3 = 4,\ n_4 = m_4 = 5$)

Figure 3.8: Expected detection delays of multi-mode delay-timers

$$
\begin{aligned}
&\text{Design} \quad x \\
&\text{s.t.} \begin{cases} \text{FAR} \ \leq a \\ \text{MAR} \leq b \\ \text{EDD} \leq c \end{cases}
\end{aligned} \tag{3.46}
$$

where $\tau \in \mathbb{R}$, and n_i and $m_i \in \{0, 1, 2, 3, ...\}$ $(1 \leq i \leq Q)$, resulting in a mixed-integer design problem. Expressions for FAR, MAR, and EDD are highly non-linear, and it is not possible to solve this design problem analytically. Consequently, it is converted into a minimax problem as follows:

$$
\begin{aligned}
\min \quad & f(x) \\
\text{s.t.} \quad & Ax \leq \kappa
\end{aligned} \tag{3.47}
$$

where $Ax \leq \kappa$ are some reasonable linear constraints on the design variables, and $f(x)$ is defined as:

$$
f(x) = \max \{0, g_1(x), g_2(x), g_3(x)\} \tag{3.48}
$$

$g_1(x) = \text{FAR} - a$, $g_2(x) = \text{MAR} - b$, and $g_3(x) = \text{EDD} - c$. There are some solutions available in the literature to solve this problem [8, 22]. In this chapter, a particle swarm optimization (PSO) based method is adopted, because of its ability to handle the mixed-integer design parameters [11].

3.5.2 Particle swarm optimization

A derivative-free optimization based solution of the problem (3.47) is proposed, as the analytical expressions for the performance indices are complex, and computing derivatives is not a trivial task. Among various derivative-free optimization based algorithms (Nelder-Mead method, Pattern search, Bayesian Optimization etc.) the Particle Swarm Optimization (PSO) provides the faster convergence by combining the global scope of the swarm search with the local convergence of the Nelder-Mead method [14]. PSO is a population based search algorithm, that solves the optimization problem. The algorithm is initialized with a random number of potential solutions, known as particles. In order to achieve the best possible solution, particles traverse the multidimensional search space by updating their positions and velocities based on their neighbouring particles. Equations for the position and velocity updates for the n^{th} iteration are written here [12]:

$$
v^n = v^{n-1} + \alpha^n \beta_1^n (p_{best} - P^{n-1}) + \alpha^n \beta_2^n (g_{best} - P^{n-1}) \tag{3.49}
$$

$$
P^n = P^{n-1} + v^n \tag{3.50}
$$

where v, and P are the velocity and position of the particles, respectively, β_1, β_2 are uniformly distributed random numbers between 0 and 1, p_{best} is the best solution achieved so far by the particle, g_{best} is the global best achieved among all the particles, and α^n is the learning factor of the algorithm, and is updated according to the following equation:

$$\alpha^n = \alpha_{\max} - \frac{\alpha_{\max} - \alpha_{\min}}{n_{\max}} n \tag{3.51}$$

where $\alpha_{\max} = 0.9$, $\alpha_{\min} = 0.4$, and $n_{\max}$ is the maximum number of iterations. For the case, where the particle domain is integers, the velocity update equation takes the following form:

$$v^n = \text{round}(v^{n-1} + \alpha^n \beta_1^n (p_{best} - P^{n-1}) + \alpha^n \beta_2^n (g_{best} - P^{n-1})) \tag{3.52}$$

where round(.) is a function that rounds the element to the nearest integer. Detailed discussion on the algorithm can be found in [12, 29, 49].

3.5.3 Design based on particle swarm optimization

In order to design the threshold, and delay-timers for different modes of the process, the PSO algorithm is utilized. The proposed algorithm consists of three main parts: initialization, computation, and termination. In the initialization section, a large number of particles, as potential solutions, are randomly initialized, while maintaining the constraints on x, an upper limit for the maximum number of iterations, and boundary conditions on the design parameters are set. f_{best} and g_{best} are initialized with suitable values. In the computation section, particles update their velocities and positions, while remaining within the boundary conditions; f_{best} and g_{best} are also tracked in the updating step. Finally in the termination section, design parameters are reported, if the minimum of $f(x)$ has been found. A complete description of the procedure is given in Algorithm 4.

Algorithm 4 Multi-mode delay-timers design

Initialization

Specify the upper limits on FAR, MAR, and EDD (a, b, and c), and set the objective function to be $f(x) = \max\{0, \mathrm{FAR} - a, \mathrm{MAR} - b, \mathrm{EDD} - c\}$

Initialize the particles $(\tau, n_1, ..., n_Q, m_1, ..., m_Q)$ randomly, while satisfying the constraints, and boundary conditions.

Set $i_{\max}$ (maximum number of iterations) to some appropriate large number, and $f(p_{best}) = f(g_{best}) = [200\ 200\ 30]$

Computation

while $i <= i_{\max} \mid\mid f(x) > 0$ **do**

Evaluate the objective function $f(x_i)$ for each particle in the swarm, and compare it with the previous best value obtained by the particle.

    if $f(x_i) < f(p_{best})$ **then**

        $p_{best} = P_i$

        $f(p_{best}) = f(x_i)$

    end if

Compare the minimum value obtained in the i^{th} iteration with the global best g_{best} obtained, so far.

    if $\min(f(x_i)) < f(g_{best})$ **then**

        $g_{best} = arg[\min(f(x_i))]$

        $f(g_{best}) = \min(f(x_i))$

    end if

Use (3.49) and (3.50) to update the continuous variable in the particles (τ), and use (3.52) and (3.50) to update the discrete variables in the particles ($n's$ and $m's$).

If updated positions violate any of the boundary conditions, set the variables to their boundary limits.

$i = i + 1$

end while

Termination

if $f(g_{best}) = 0$ **then**

    Requirements on FAR, MAR, and EDD have met.

    $x_{designed} = g_{best}$

else

    Requirements on FAR, MAR, and EDD have not met.

end if

3.5.4 Design examples

For illustration purposes, three design examples are considered. Each example has different requirements on the FAR, MAR, and EDD.

Example 1: *A two mode process with Gaussian distributions*
In this example, design of multi-mode delay-timers for a two mode process is considered. Description of the process is given below:

$$\text{Normal modes} \begin{cases} \text{Mode 1: } \mathcal{N}(1,2) \\ \text{Mode 2: } \mathcal{N}(3,2) \end{cases} \tag{3.53}$$

$$\text{Abnormal: } \mathcal{N}(4,2) \tag{3.54}$$

$$\text{Mode switching: } T = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \tag{3.55}$$

where $\mathcal{N}(\mu, \sigma)$ is Gaussian distribution with mean μ and standard deviation σ. The following design requirements are set on the process:

$$\begin{aligned} &\text{Design} \quad x = \{\tau, n_1, n_2, m_1, m_2\} \\ &\text{s.t.} \begin{cases} \text{FAR} \leq 3 \\ \text{MAR} \leq 5 \\ \text{EDD} \leq 8 \end{cases} \end{aligned} \tag{3.56}$$

Linear constraints and boundary conditions on the design parameters are as follow:

$$\begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \\ m_1 \\ m_2 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.57}$$

$$4 \leq \quad \tau \quad \leq 6 \tag{3.58}$$

$$1 \leq \quad n_1,\ n_2 \quad \leq 10 \tag{3.59}$$

$$1 \leq \quad m_1,\ m_2 \quad \leq 10 \tag{3.60}$$

Linear constraints, considered here, on the lengths of the delay-timers are arbitrary. However, for actual processes, generally there are certain constraints on the lengths of the delay-timers for various modes, and those constraints can be put into the design requirements using the linear constraints. Algorithm 4 is used to design x. It is found out that the algorithm converges for the given design requirements, and boundary conditions. The following parameters are obtained:

$$x = \{\tau,\ n_1,\ n_2,\ m_1, m_2\} = \{4.0412,\ 4,\ 3,\ 3,\ 1\} \tag{3.61}$$

It is worth mentioning here that the design problem does not have a unique solution, and the algorithm provides one of many possible solutions. It is possible to fine-tune the solution, by specifying the process specific boundary conditions, and constraints.

Example 2: *A three mode process with Gamma distributions*

In this design example a three mode process with gamma distributions is considered. Process specifications are given below:

$$\text{Normal modes} \begin{cases} \text{Mode 1: gamma}(2,2) \\ \text{Mode 2: gamma}(5,1) \\ \text{Mode 3: gamma}(6.5,1) \end{cases} \tag{3.62}$$

$$\text{Abnormal: gamma}(7.5,1) \tag{3.63}$$

$$\text{Mode switching: } T = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.2 & 0.6 & 0.2 \\ 0 & 0.2 & 0.8 \end{bmatrix} \tag{3.64}$$

Design requirements on the multi-mode delay-timers are set to be the following:

$$\begin{aligned} &\text{Design } \ x = \{\tau, n_1,\ n_2,\ n_3\ m_1,\ m_2,\ m_3\} \\ &\text{s.t.} \begin{cases} \text{FAR} \le 5 \\ \text{MAR} \le 5 \\ \text{EDD} \le 5 \end{cases} \end{aligned} \tag{3.65}$$

The following linear constraints and boundary conditions are set on x:

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ m_1 \\ m_2 \\ m_3 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.66}$$

$$8 \leq \quad \tau \quad \leq 10 \tag{3.67}$$

$$1 \leq \quad n_1,\ n_2,\ n_3 \quad \leq 10 \tag{3.68}$$

$$1 \leq \quad m_1,\ m_2,\ m_3 \quad \leq 10 \tag{3.69}$$

For the specified design requirements, and constraints, Algorithm 4 converges to the following x:

$$x = \{\tau,\ n_1,\ n_2,\ n_3,\ m_1, m_2,\ m_3\} = \{7.46,\ 3,\ 4,\ 3,\ 8,\ ,9,\ 1\} \tag{3.70}$$

Example 3: *The Continuous Stirred Tank Heater (CSTH)*

The CSTH is part of a laboratory setup, that is present in the Department of Chemical and Materials Engineering at the University of Alberta. It maintains the temperature of the downstream flow by mixing and stirring the cold and hot water mixture in the tank. Steam flow in the tank is controlled using a steam flow control valve. Outlet flow control valve is used to control the downstream flow rate. The schematic diagram of the process is shown in Figure 2.9. Three process variables (the cold water flow rate, tank level, and temperature) are constantly monitored, and in this chapter the alarm system design for the flow rate is considered. Two operating modes are defined based on the hot water valve opening, and it assumed that the process can switch between these two modes with equal probability [55]. Thus the mode switching matrix (T) can be written as:

$$T = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \tag{3.71}$$

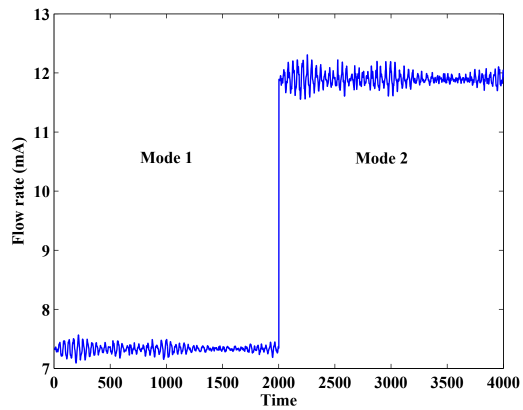


Figure 3.9: Flow rate of the CSTH process

Behavior of the cold water flow rate in the two operating modes is shown in Figure 3.9. Historical data is used to fit the probability distributions for both normal modes, and abnormal mode of the process.

$$\text{Normal modes}\begin{cases}\text{Mode 1: } \mathcal{N}(7.32987, 0.0653087)\\ \text{Mode 2: } \mathcal{N}(11.8971, 0.107156)\end{cases} \tag{3.72}$$

$$\text{Abnormal: } \mathcal{N}(13.02152, 0.4507) \tag{3.73}$$

The following design requirements are set on the design variables:

$$\begin{aligned} &\text{Design} \quad x = \{\tau, n_1, n_2, m_1, m_2\} \\ &\text{s.t.} \begin{cases} \text{FAR} \leq 2 \\ \text{MAR} \leq 2 \\ \text{EDD} \leq 2 \end{cases} \end{aligned} \tag{3.74}$$

In order to decrease the missed alarm count, a relatively larger length of the off delay-timer is to be designed for the mode operating closer to the abnormal mode, i.e., for mode 2, and thus $m_2 \geq m_1$. For on delay-timers, a smaller range is considered in order to decrease the detection delay of the alarm system. This results in the following linear constraints and boundary conditions:

Table 3.1: A performance comparison of design examples

	Example 1		**Example 2**		**Example 3**	
	Analytical	Simulated	Analytical	Simulated	Analytical	Simulated
FAR (%)	1.37	1.40	1.95	1.98	0	0
MAR (%)	3.78	3.76	1.52	1.53	0.23	0.22
EDD	7.63	7.65	3.48	3.45	1.07	1.09

$$\begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \\ m_1 \\ m_2 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.75}$$

$$12.5 \leq \quad \tau \quad \leq 13.5 \tag{3.76}$$

$$1 \leq \quad n_1,\ n_2 \quad \leq 5 \tag{3.77}$$

$$1 \leq \quad m_1,\ m_2 \quad \leq 10 \tag{3.78}$$

Algorithm 4 is utilized for the design of x, and the following parameters are obtained:

$$\text{x} = \{\tau,\ n_1,\ n_2,\ m_1, m_2\} = \{12.8979,\ 2,\ 2,\ 2,\ 7\} \tag{3.79}$$

All three considered examples are simulated with the designed parameters (threshold and delay-timers), and 3000 Monte Carlo simulations are performed to compare the analytical values of FAR, MAR, and EDD. The resulted numbers are listed in Table 3.1. From this table it can be seen that both analytical and simulated values are very close to each other, and this validates the design algorithm.

3.6 Summary

In this chapter HMM based models were developed for the multi-mode delay-timer configurations. Analytical expressions for the false alarm rate, missed alarm rate, and expected detection delay were derived. A particle swarm optimization based algorithm was proposed for designing multi-mode delay-timers. Design examples were considered to show the utility of the proposed method. In the next chapter, time-deadband configurations for single mode processes are studied, while considering univariate alarm system settings.

Chapter 4

Time-deadbands for Univariate Alarm Systems*.

Among various alarm configuration methods, time-deadbands are one of the techniques that is used in industry to get rid of false and nuisance alarms. In this chapter, the utility of time-deadbands in univariate alarm system setting is studied. Mathematical models are developed based on Markov processes, and analytical expression for the false alarm rate, missed alarm rate, and expected detection delay are computed. Systematic procedures are also proposed for designing time-deadbands for single mode processes.

4.1 Overview

Time-deadbands, also known as alarm latches, are one of the types of the deadbands used in industry. A few papers on the quantitative analysis of the measurement-deadbands can be found in the literature, which can help in assessing and designing alarm systems based on measurement-deadbands [2, 41]. While there is also some literature available that deal with time-deadband configurations, e.g., in [30] the authors provided some practical

*A version of this chapter has been submitted for publication as: Afzal, M.S., Chen, T., Bandehkhoda, A., & Izadi, I., Analysis and Design of Time-deadbands for Univariate Alarm Systems, Control Engineering Practices. A preliminary version of this chapter was presented in the American Control Conference (ACC), Seattle, WA, USA May 24-26, 2017

implications on the use of alarm latches based on qualitative analysis, and in [18] the author proposed a time-series prediction based approach to estimate the lengths of time-deadbands, to the best of the authors' knowledge there is not any study available that provides quantitative analysis of the performance of the time-deadband configurations, and proposes systematic design procedures. Consequently, in this chapter we are analyzing time-deadband configurations for univariate alarm systems.

The rest of the chapter is organized as follows. In Section 4.2, background information on the types of deadbands is given. A run-length based encoding scheme for alarm sequences is also introduced in this section. In Section 4.3, a model of the time-deadband configuration is developed. In Section 4.4, definitions of different performance indices are provided, and their expressions for time-deadbands are derived. Section 4.5 provides the description of design procedures for time-deadbands along with illustrative examples. Concluding remarks are given in Section 4.6.

4.2 Background

In this section, different types of deadbands are discussed. A brief description of run-length based encoding of alarm sequences is also part of this section.

4.2.1 Types of deadbands

Two types of deadband configurations can be found in alarm systems: measurement-deadbands and time-deadbands, as shown in Figure 1.3. A measurement-deadband can only be applied on a continuous process variable, and it can be considered as a bi-threshold alarm system, where an alarm is raised when the process variable goes above the upper threshold, and the alarm is cleared when the process variable falls below the lower threshold. Figure 4.1 shows an example of a process variable configured with a measurement-deadband. From this figure it can be seen that there are a few instances when

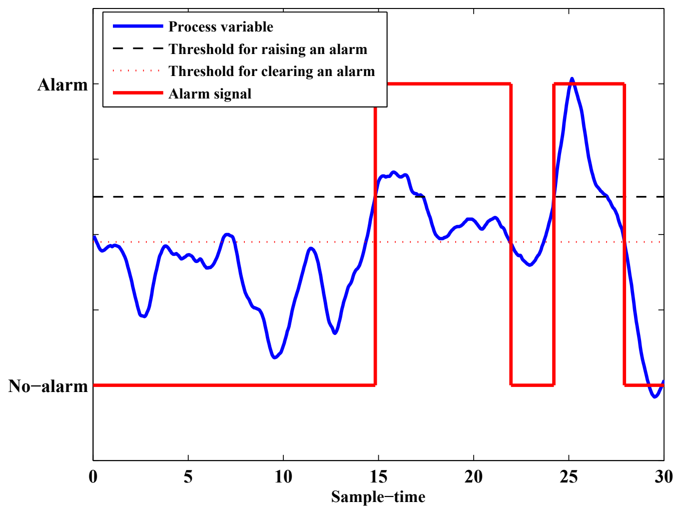


Figure 4.1: Example of a measurement-deadband configuration

the process variable goes above the lower threshold; however alarms are delayed until it crossed the upper threshold. Similarly, clearance of alarms is also delayed until the process variable goes below the lower threshold.

Time-deadbands constitute uni-threshold alarm systems, and can be applied to both continuous and discrete process variables, unlike measurement-deadbands. In general, time-deadband represents two waiting times: (1) Minimum waiting-time before an alarm can be cleared; (2) Minimum waiting-time before an alarm can be raised. In this chapter, these two waiting times are denoted as T_A and T_{NA}, respectively. Whenever a process variable goes above the threshold, an alarm is raised; however the alarm can be cleared only after the minimum waiting-time (T_A) in the alarm state has passed. Similarly, the raise of an alarm is possible only if the minimum waiting-time (T_{NA}) in the no-alarm state has passed. In Figure 4.2 an example of a time-deadband configuration on a continuous process variable is shown. In this example waiting times are configured to be 5 sample-times for both alarm and no-alarm states. From this figure it can be seen that the first alarm is raised at the time instance marked as x_1, and although the variable goes below the threshold (x_2) after a few samples, but the alarm is not cleared because of the configured

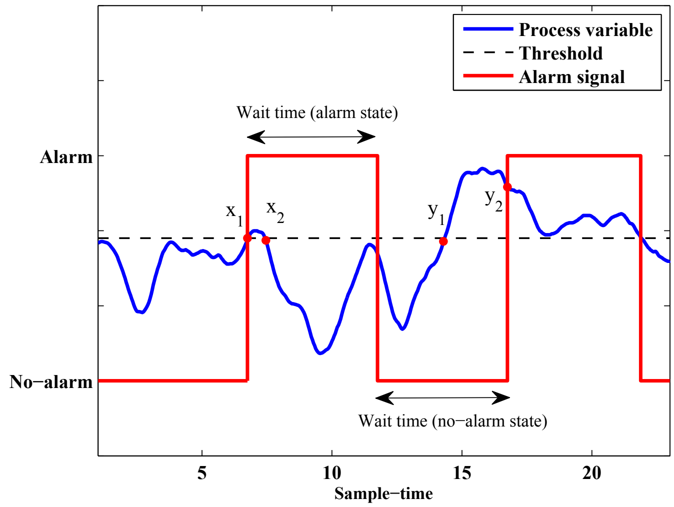


Figure 4.2: Example of a time-deadband configuration

waiting-time in the alarm state. Similarly, at the time instance y_1, the process variable goes above the threshold again; however, raise of an alarm is delayed until y_2, because of the waiting-time in the no-alarm state.

Recommendations on the use of both measurement-deadbands and time-deadbands can be found in ISA 18.2 and EEMUA 191 standards [21] [13]. For benefits of readers, recommendations for time-deadbands based on the types of process variables are listed in Table 4.1. It is worth mentioning here that these recommendations are only used as initial estimates for designing time-deadbands. Actual values for the waiting times are typically decided by considering the criticality of the process variable, and signal to noise ratio of the measurements.

4.2.2 Run-length encoding of alarm sequences

An alarm sequence is a series of ones and zeros, which is generated after comparing the measurements of a process variable against a configured threshold. The design of different types of alarm configurations on a process variable can make use of either actual process measurements or the corresponding alarm sequence. In this chapter both design scenarios for time-deadbands are

Table 4.1: Recommendations for time-deadbands

Variable type	**Time-deadband** (sec.)
Flow	15
Pressure	15
Level	60
Temperature	60
Composition	120

considered, and for the case of alarm sequence based design, run-length based encoding scheme is utilized.

Run-length encoding is a type of lossy or lossless data compression, in which runs (specific sequence of elements occurring throughout dataset) of the data are used to represent the entire dataset [32]. Run-length based encoding has its application in many fields, like image and signal processing, statistical control, and finance. For the case of alarm signals, runs contain sequences of 1's (alarm state) and 0's (no-alarm states), and depending on the definitions of the run many types of run-length encoding schemes can be defined. In the following, a few of these types are described, with the help of an example of an alarm sequence, shown in Figure 4.3. For illustration purposes, lengths of each alarm and no-alarm states are also indicated in the figure.

RTN-RTN run-length based encoding

It is an example of a lossy run-length encoding scheme for alarm signals, in which the length between one RTN point (time instance at which alarm sequence returns to a no-alarm state from an alarm state) to next RTN point is considered as a run. In other words, for *RTN-RTN* run-length encoding scheme, sequences of 0's for the no-alarm state and the following 1's for the alarm state form a run, and it is assumed that the alarm sequence is starting in a no-alarm state. The cases where this assumption is not true, 1's corresponding to the first alarm state are discarded, as is the case for the example shown in Figure 4.3. For this alarm sequence, first *RTN-RTN* run is formed by summing up the lengths of the first no-alarm state, and the second alarm state, i.e., $1+2 = 3$. Overall, *RTN-RTN* run-length encoding will be: *RTN-RTN*= {3, 5, 3, 2, 6, 3}.

ALM-ALM run-length based encoding

It is also an example of a lossy run-length encoding scheme for alarm signals; however, unlike *RTN-RTN* encoding, in this case ALM points (time instance at which alarm sequence jumps to an alarm state from a no-alarm state) are considered to form a run. In particular, the sum of the lengths of an alarm state and the following no-alarm state constitutes a run. For the example shown in Figure 4.3, *ALM-ALM* encoding scheme will be: *ALM-ALM* = {3, 4, 4, 3, 3, 5}.

Lossless run-length encoding (LRLE)

For a lossless run-length encoding scheme, lengths of the alarm states, the no-alarm states are considered separately and synchronously. In this case, instead of representing an alarm sequence with a series of 1's (alarm states) and 0's (no-alarms), lengths of each alarm state and no-alarm state are listed in a time synchronous manner. *LRLE* based representation of the considered example will be: *LRLE* = {2, 1, 2, 2, 3, 1, 2, 1, 1, 2, 4, 1, 2}.

4.3 Modeling of time-deadbands

Consider a process variable x following a distribution $P_n(x)$ during normal operation, and $P_{ab}(x)$ during abnormal operation of the process, as shown in Figure 4.4. Without loss of generality, assume that only high alarm threshold is configured on this variable. Let $p_2 = 1 - p_1$ be the probability of the process variable to go above the threshold during normal operation (shaded

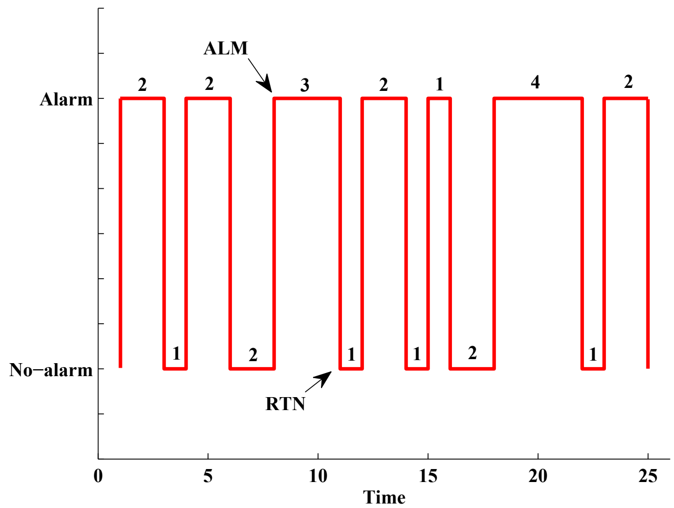


Figure 4.3: Time trend of an alarm sequence

region under the normal distribution), and $q_1 = 1 - q_2$ be the probability of the process variable to fall below the threshold during abnormal operation (shaded region under the abnormal distribution of the process variable). Further, let T_A sample-times be the minimum waiting-time before an alarm can be cleared, and T_{NA} be the minimum waiting-time in the no-alarm state before an alarm can be raised. Then such an alarm configuration can be defined completely using a semi-Markov process with two states, namely, the alarm state (A) and no-alarm state (NA). Transition probabilities between the two states are defined based on the probability distributions of the process variables, and the resting time of each state is dependent on both the probability distributions, and the time-deadband configuration. For the normal operation of the process variable, the following set of equations completely describe the semi-Markov based time-deadband configuration:

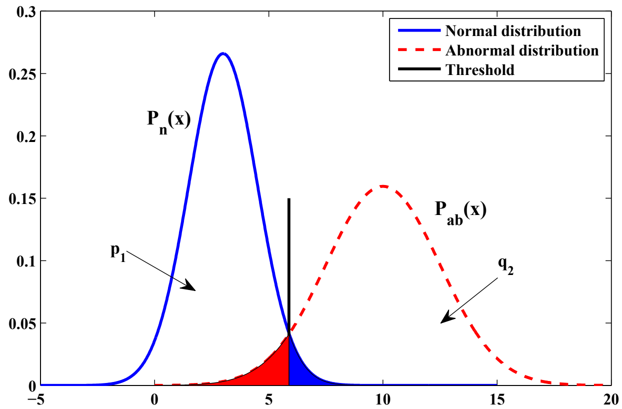


Figure 4.4: Normal and abnormal distributions of a process variable

$$
\begin{cases}
\mathbb{P}\left[S_{[t} = \mathrm{A} \mid S_{[t-d:t-1]} = \mathrm{A}\right] = & \begin{cases} p_2, & d > T_A \\ 1, & d \leq T_A \end{cases} \\
\mathbb{P}\left[S_{[t} = \mathrm{NA} \mid S_{[t-d:t-1]} = \mathrm{A}\right] = & \begin{cases} p_1, & d > T_A \\ 0, & d \leq T_A \end{cases} \\
\mathbb{P}\left[S_{[t} = \mathrm{A} \mid S_{[t-d:t-1]} = \mathrm{NA}\right] = & \begin{cases} p_2, & d > T_{NA} \\ 0, & d \leq T_{NA} \end{cases} \\
\mathbb{P}\left[S_{[t} = \mathrm{NA} \mid S_{[t-d:t-1]} = \mathrm{NA}\right] = & \begin{cases} p_1, & d > T_{NA} \\ 1, & d \leq T_{NA} \end{cases}
\end{cases}
\tag{4.1}
$$

where $\mathbb{P}\left[S_{[t} = i \mid S_{[t-d:t-1]} = j\right]$ represents the probability of the model of switching to state $i \in \{A,\ NA\}$ at time t, given that the state of the model was $j \in \{A,\ NA\}$ during time interval $t-d:t-1$. A similar set of equations can be written for the case when the process is operating under abnormal conditions, by replacing p_1 and p_2 with q_1 and q_2, respectively. Figure 4.5 shows an example of the sample path of a semi-Markov process based model with time-deadband configuration ($T_{NA} = 2,\ T_A = 3$). In this example the probabilities of switching between alarm and no-alarm states are assumed to be $p_1 = 0.8$, and $p_2 = 0.2$. Waiting times in both alarm and no-alarm states are represented as horizontal lines of probabilities equal to 1, whereas the length of the line corresponds to the amount of waiting-time in different states.

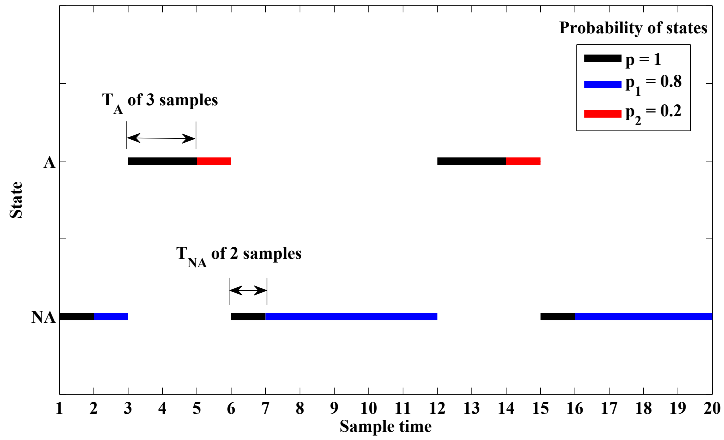


Figure 4.5: A sample-path of a semi-Markov chain for a time-deadband configuration

While a semi-Markov process can completely define the time-deadband configuration, for analysis purposes and to derive performance indices it is required to study the long term behavior of the model, which is not a trivial task for semi-Markov process based models. Fortunately, for the case of time-deadbands, it is possible to convert the semi-Markov model to a standard Markov model, by introducing non-self transitioning states corresponding to the lengths of the waiting times for both the alarm and no-alarm states. Figure 4.6 shows a resultant Markov chain based model of the semi-Markov process shown in Figure 4.5. Three non-self transiting states (A_1, A_2, A_3) correspond to the waiting-time in alarm state ($T_A = 3$), and two non-self transiting states (NA_1, NA_2) represent the $T_{NA} = 2$. A more generic standard Markov model for time-deadband configuration (T_A, T_{NA}) under the normal operation of the process is shown in Figure 4.7. A similar model can be obtained for the abnormal situation by replacing p_1 and p_2 with q_1 and q_2, respectively.

4.3.1 Long-term behavior of the model

Performance of the time-deadband configurations can be assessed by studying the long-term behavior of the Markov chain model. Under the assumption

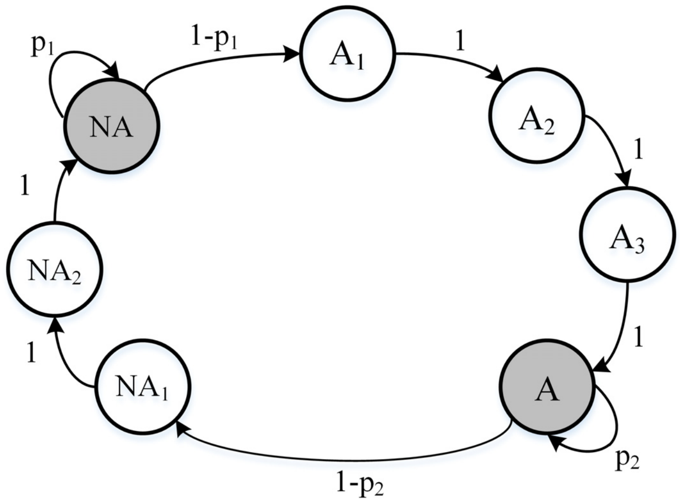


Figure 4.6: The Markov chain model of a time-deadband configuration ($T_A = 3$, $T_{NA} = 2$)

that the samples from the process variable are independent and identically distributed, long-term behavior of the model can be studied by finding the stationary distribution of the model. For the proposed Markov model (Figure 4.7) the state transition matrix can be written as:

$$P_n = \left[\begin{array}{c|c} P_{11} & P_{12} \\ \hline P_{21} & P_{22} \end{array}\right]_{(\alpha+\beta)\times(\alpha+\beta)} \tag{4.2}$$

where $\alpha = 1+T_{NA}$, $\beta = 1+T_A$, and P_{ij} $(1 \leq i, j \leq 2)$ are sub-matrices, which are given by:

$$P_{11} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & p_1 \end{bmatrix}_{\alpha\times\alpha} \quad P_{12} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \\ p_2 & 0 & 0 & \dots & 0 \end{bmatrix}_{\alpha\times\beta} \tag{4.3}$$

$$P_{21} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \\ p_1 & 0 & 0 & \dots & 0 \end{bmatrix}_{\beta\times\alpha} \quad P_{22} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & p_2 \end{bmatrix}_{\beta\times\beta} \tag{4.4}$$

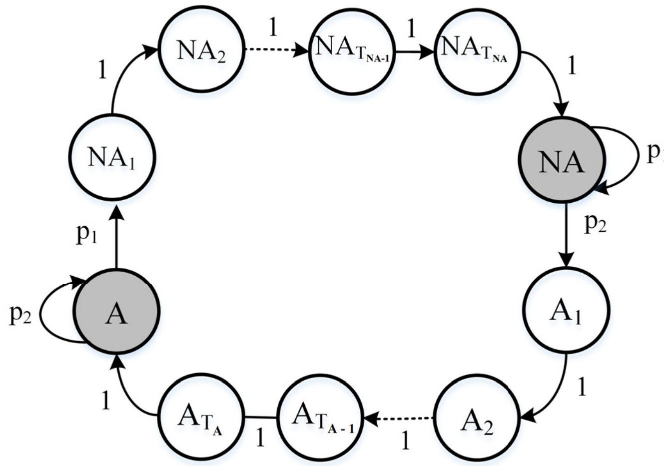


Figure 4.7: The Markov chain model of a time-deadband configuration (T_A, T_{NA})

A similar state transition matrix (P_{ab}) can be obtained by replacing p_1 and p_2 with q_1 and q_2, respectively. The stationary distribution of the model can be found using the following equation [76]:

$$\pi_n = \mathbf{1}_{\alpha+\beta}(\mathbf{I} - P_n + \mathbf{U})^{-1}_{(\alpha+\beta)\times(\alpha+\beta)} \tag{4.5}$$

where $\mathbf{1}$ represents a row vector of ones, $\mathbf{I}$ is an identity matrix, and $\mathbf{U}$ is a square matrix of ones. This results in the following:

$$\pi_n = [\delta_1 \ \ldots \ \delta_{\alpha-1} \ \delta_\alpha \ \delta_{\alpha+1} \ \ldots \ \delta_{\alpha+\beta-1} \ \delta_{\alpha+\beta}] \tag{4.6}$$

where δ_k $(1 \leq k \leq \alpha+\beta)$ represents the probability of the k^{th} state in the long-term run of the model. After some simplifications the stationary distribution of the model reduces to the following vector:

$$\pi_n = \frac{\left[1 \ \ldots \ 1 \ \frac{1}{p_2} \ 1 \ \ldots \ 1 \ \frac{1}{p_1}\right]}{T_A + T_{NA} + \frac{1}{p_1} + \frac{1}{p_2}} \tag{4.7}$$

A similar distribution of the model can be written for the abnormal operation of the process variable:

$$\pi_{ab} = \frac{\begin{bmatrix} 1 & \dots & 1 & \frac{1}{q_2} & 1 & \dots & 1 & \frac{1}{q_1} \end{bmatrix}}{T_A + T_{NA} + \frac{1}{q_1} + \frac{1}{q_2}} \tag{4.8}$$

In the long-run of the process, the stationary distributions of the model can be used to find the probability of the process variable to be present in one of the alarm or no-alarm states.

4.3.2 Discussion on the assumptions

While developing the model a number of assumptions were made. In this subsection the impact of these assumptions for practical cases is discussed.

Distributions of the process data

It is assumed that the distributions of the process data are known for both the normal and abnormal operations. However, in many cases it is not possible to know the distributions of the process data beforehand. In such cases, a Kernel Density Estimation (KDE) based approach can be used to find the estimates of the distributions, given that sufficient historical data is available. Another hurdle in finding the estimates for the normal and abnormal distributions is to distinguish between the normal and abnormal data in the historical data. This problem can be overcome by either referring to the event logs of the process operation or by using some data based methods to distinguish the abnormal data from normal data, e.g., in [71] authors have proposed a correlation directions based method for abnormal data detection.

Existence of the stationary distributions

To study the long-term behavior of the time-deadband configuration, it is assumed that the distributions of the process variable during both the normal and abnormal operation are independent and identically distributed. However, in practice this assumption does not hold true when considering the entire historical data of the process variable, because such data includes a lot of transitions due to mode changes and various other factors pertaining to the process operation. However, if only part of the data is considered by eliminating the transitional changes, an identical distribution for the process variable can be assumed [6]. Furthermore, for the case of univariate alarm systems, the underlying assumption is that the process variable under consideration is independent of the effects of the other variables in the process.

4.4 Performance assessment

For performance assessment of time-deadband configurations, three performance indices, the false alarm rate, the missed alarm rate, and the expected detection delay, are considered.

4.4.1 False alarm rate

The false alarm rate is the probability of raising an alarm while the process is operating under normal conditions. For the case of univariate alarm system, the false alarm rate can be calculated by finding the probability of the process variable to go above the threshold under the normal distribution ($P_n(x)$):

$$\text{FAR} = \int_{\zeta}^{\infty} P_n(x)dx \tag{4.9}$$

where ζ represents the alarm threshold. For time-deadband configurations, the false alarm rate can be calculated by considering the probabilities of all the alarm states ($A_1, A_2, ..., A_{T_A}, A$) in the Markov model, i.e.

$$\text{FAR} = \mathbb{P}(A_1) + \mathbb{P}(A_2) + \ \ldots \ + \mathbb{P}(A_{T_A}) + \mathbb{P}(A) \tag{4.10}$$

The probabilities of all the alarm states can be found from the stationary distribution (π_n), given by (4.7), and after a few simplifications the following expression for the false alarm rate can be obtained:

$$\text{FAR} = \frac{\overbrace{1+1+...+1}^{T_A}+\frac{1}{p_1}}{T_A+T_{NA}+\frac{1}{p_1}+\frac{1}{p_2}} \tag{4.11}$$

$$= \frac{T_A+\frac{1}{p_1}}{T_A+T_{NA}+\frac{1}{p_1}+\frac{1}{p_2}} \tag{4.12}$$

4.4.2 Missed alarm rate

The missed alarm rate is the failure probability of an alarm system in raising an alarm, while the process is in abnormal mode. Theoretically, the probability for a univariate alarm system can be calculated as follows:

$$\text{MAR} = \int_{-\infty}^{\zeta} P_{ab}(x)dx \tag{4.13}$$

where $P_{ab}(x)$ is the distribution of the process variable under abnormal conditions. The missed alarm probability for time-deadbands can be calculated by summing up the probabilities corresponding to the no-alarm states $(NA_1, NA_2, ..., NA_{T_{NA}}, NA)$, i.e.

$$\text{MAR} = \mathbb{P}(NA_1) + \mathbb{P}(NA_2) + \ \dots \ + \mathbb{P}(NA_{T_{NA}}) + \mathbb{P}(NA) \tag{4.14}$$

After plugging in the probabilities of no-alarm states from (4.8), the following analytical expression for the computation of the missed alarm rate is obtained:

$$\text{MAR} = \frac{\overbrace{1+1+...+1}^{T_{NA}}+\frac{1}{q_2}}{T_A+T_{NA}+\frac{1}{q_1}+\frac{1}{q_2}} \tag{4.15}$$

$$= \frac{T_{NA}+\frac{1}{q_2}}{T_A+T_{NA}+\frac{1}{q_1}+\frac{1}{q_2}} \tag{4.16}$$

4.4.3 Expected detection delay

The detection delay is defined as the time taken by the alarm system to raise an alarm after the process has entered the abnormal region of operation.

The mean value of the detection delay is termed as the expected detection delay. Let t_{ab} be the time instance when process enters the abnormal region, and let t_a be the time at which an alarm is raised; then the detection delay in terms of the number of time samples, can be defined as:

$$\text{Detection delay} = t_a - t_{ab} \tag{4.17}$$

For the case of Markov process based model, detection delay can be defined as the time samples taken by the Markov chain in switching from no-alarm states $(NA_1, NA_2, ..., NA_{T_{NA}}, NA)$ to any of the alarm states $(A_1, A_2, ..., A_{T_A}, A)$, which is known as hitting time of the Markov chain [35]. A hitting time of z samples for the developed model can be found as:

$$\mathbb{P}(\text{detection delay} = z) = \pi_n P_{ab} P^z [\overbrace{0\ \ldots\ 0}^{T_{NA}+1}\ \underbrace{1\ \ldots\ 1}_{T_A+1}]^T \tag{4.18}$$

where P is a square matrix of size $\alpha+\beta$, and is obtained from P_{ab} by replacing all the transition probabilities corresponding to the alarm states with zeros. The expected value of the detection delay can be found as:

$$\begin{aligned} \text{EDD} &= \sum_{z=0}^{\infty} z\mathbb{P}(\text{detection delay} = z) \\ &= \pi_n P_{ab} P(I-P)^{-2}[0\ \ldots\ 0\ 1\ \ldots\ 1]^T \qquad (4.19) \end{aligned}$$

Eq. (4.19) can be further simplified by letting $\delta = \pi_n P_{ab}$, and $\gamma = P(I - P)^{-2}[0\ \ldots\ 0\ 1\ \ldots\ 1]^T$, and then the following expressions for δ and γ can be obtained:

$$\delta = \frac{\left[\overbrace{\frac{q_1}{p_1}\ 1\ \ldots\ 1}^{T_{NA}}\ (1+\frac{q_1}{p_2})\ \overbrace{\frac{q_2}{p_2}\ 1\ \ldots\ 1}^{T_A}\ (1+\frac{q_2}{p_1})\right]}{T_A + T_{NA} + \frac{1}{p_1} + \frac{1}{p_2}} \tag{4.20}$$

$$\gamma = \left[\overbrace{\frac{q_2 T_{NA}+1}{q_2} \; \frac{q_2(T_{NA}-1)+1}{q_2} \; \dots \; \frac{q_2+1}{q_2}}^{T_{NA}} \; 1+\frac{q_1}{q_2} \; \underbrace{0\dots0}_{T_A+1}\right]^T \tag{4.21}$$

Finally, the expression for the expected detection delay for time-deadbands takes the following form:

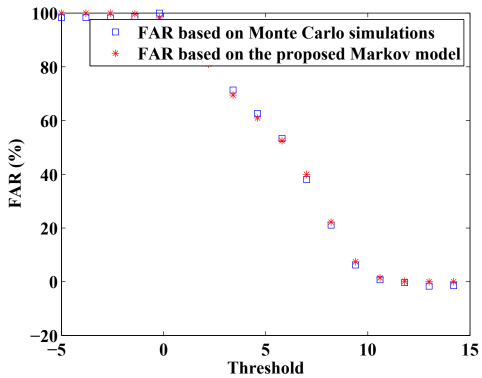
$$\text{EDD} = \frac{\frac{q_1}{p_1}\left(\frac{q_2 T_{NA}+1}{q_2}\right) + \sum_{i=1}^{T_{NA}-1} \frac{q_2(T_{NA}-i)+1}{q_2} + \frac{1}{q_2}\left(1+\frac{q_1}{p_2}\right)}{T_A + T_{NA} + \frac{1}{p_1} + \frac{1}{p_2}} \tag{4.22}$$

4.4.4 Simulation verification

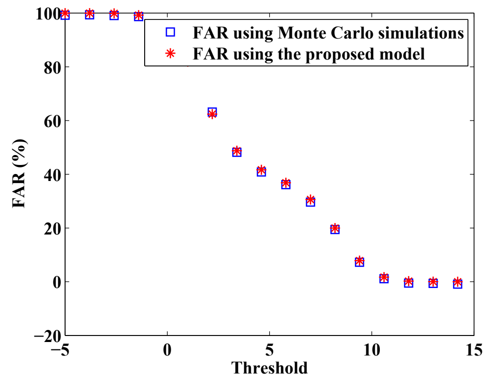
To verify the analytical expressions derived for the false alarm rate, the missed alarm rate, and the expected detection delay, 10,000 Monte Carlo simulations are performed, by considering Gaussian and Gamma distributions of the process variables. Different setting of the time-deadband configurations are also considered. Figure 4.8 shows the simulation results obtained for the false alarm rate. From the figures, it can be seen that curves obtained using the Markov process model, and the Monte Carlo simulations are very close to each other, which validates the proposed analytical expression for the false alarm rate for time-deadbands. Satisfactory results are also obtained while testing the formulas for the missed alarm rate and the expected detection delay, and are shown in Figures 4.9 and 4.10, respectively.

4.5 Design of time-deadbands

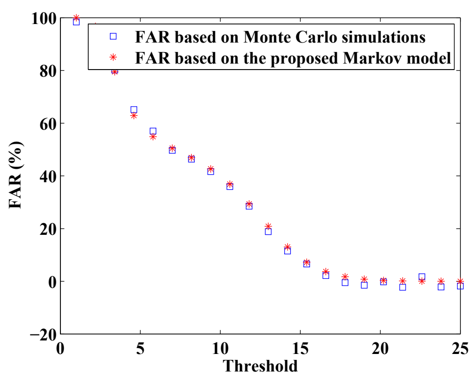
In this section two design methods are proposed for time-deadband configurations. The first method makes use of the process data and the performance indices (FAR, MAR, and EDD) to design a threshold, and time-deadband configuration on the process variable. The second method is based on the use of alarm data, and the objective is to design a time-deadband configuration such



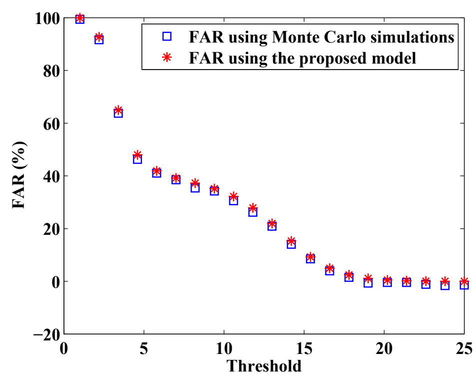
(a) Gaussian distributed variable
($T_{NA} = 3$, $T_A = 5$)



(b) Gaussian distributed variable
($T_{NA} = 10$, $T_A = 6$)

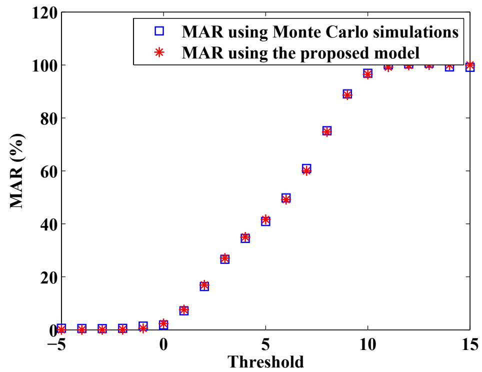


(c) Gamma distributed variable
($T_{NA} = 8$ and $T_A = 8$)

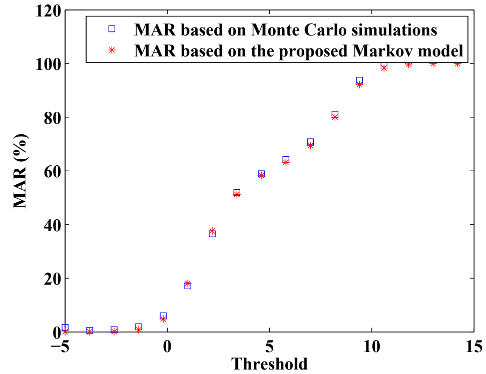


(d) Gamma distributed variable
($T_{NA} = 20$ and $T_A = 12$)

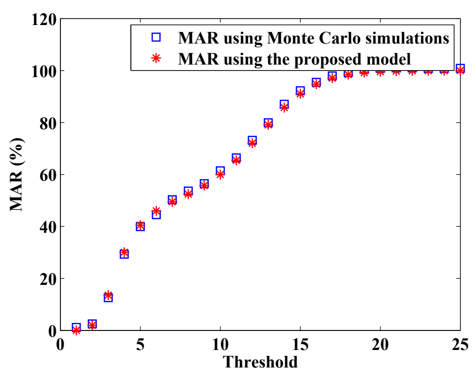
Figure 4.8: Simulation verification of the FAR formula for time-deadbands



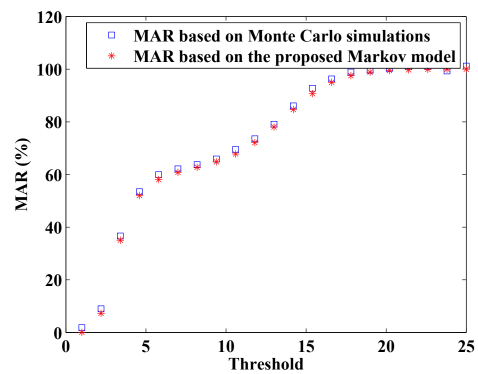
(a) Gaussian distributed variable
($T_{NA} = 3$, $T_A = 5$)



(b) Gaussian distributed variable
($T_{NA} = 10$, $T_A = 6$)

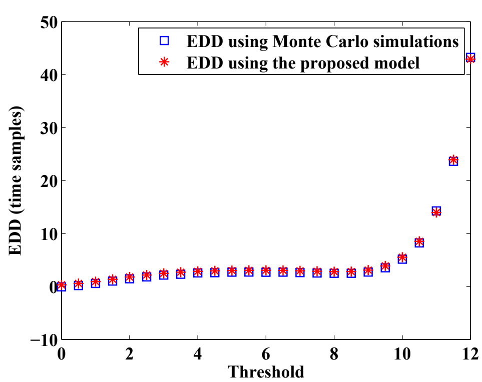


(c) Gamma distributed variable
($T_{NA} = 8$ and $T_A = 8$)

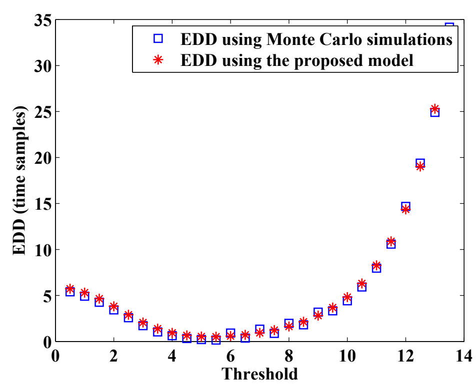


(d) Gamma distributed variable
($T_{NA} = 20$ and $T_A = 12$)

Figure 4.9: Simulation verification of the MAR formula for time-deadbands



(a) Gaussian distributed variable
($T_{NA} = 10$, $T_A = 5$)



(b) Gamma distributed variable
($T_{NA} = 20$ and $T_A = 12$)

Figure 4.10: Simulation verification of the EDD formula for time-deadbands

that alarm count and alarm chattering are reduced, while threshold setting is not altered.

4.5.1 Design based on process data

Let y be the collection of all the design parameters, i.e., $y = \{\text{threshold}, T_{NA}, T_A\}$, then the design problem can be stated as:

$$\begin{aligned} &\text{Design} \quad y \\ &\text{s.t.} \begin{cases} \text{FAR} \leq a \\ \text{MAR} \leq b \\ \text{EDD} \leq c \end{cases} \end{aligned} \tag{4.23}$$

where $a,\ b,\ c$ are the upper allowable limits on the false alarm rate, missed alarm rate, and the expected detection delay, respectively. For this design problem, a Receiver Operating Curve (ROC) based graphical method is utilized. Furthermore, it is assumed that the distributions of the process variable under normal and abnormal situations are known.

For illustration purposes, consider an example of a process variable, which is Gaussian distributed with the mean value of 0 and the standard deviation of 1 during normal operation, and for the abnormal operation the mean value of the data is shifted to 2, while the standard deviation is kept the same. Time-trends of the variable are shown in Figure 4.11, where the first 1000 samples are corresponding to the normal operation, and the last 1000 samples are during the abnormal operation of the process. The objective is to design a time-deadband configuration, and an alarm threshold such that the false and missed alarm rates are less 10%, and the value of expected detection delay is less than 5 samples. Various configurations of the time-deadbands are tested, and the values of the performance indices are calculated using (4.12), (4.16), and (4.22) for the range of threshold. Resulting ROC curves for some selected time-deadband configurations are shown in Figure 4.12. The points nearest to the origin are found by calculating the Euclidean distance for each point on

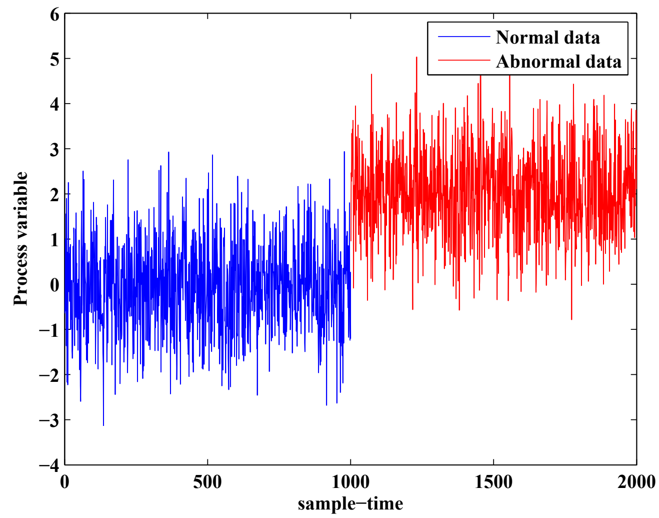


Figure 4.11: Time trend of a process variable

the curve and the origin. These points are listed in Table 4.2 along with the corresponding values of the expected detection delay and the threshold. From this table, it can be seen that the time-deadband configuration of $T_A = 65$ and $T_{NA} = 1$ satisfies the desired performance requirements, and thus can be used for the considered process variable.

4.5.2 Design based on alarm data

Many times operators are reluctant in changing the alarm limits once the process has been commissioned, and is in running state. For such cases, design procedure that involves a change in threshold for the process variable is not suited. At the same time, the operators are interested in improving the alarm system performance by removing chattering alarms, and reducing the alarm count on their monitoring screens. The objective of this design procedure is to reduce the chattering alarms, and thus the alarm count without altering the threshold settings. For quantifying the alarm chattering we have used the run-length distribution based chattering index, proposed in [31].

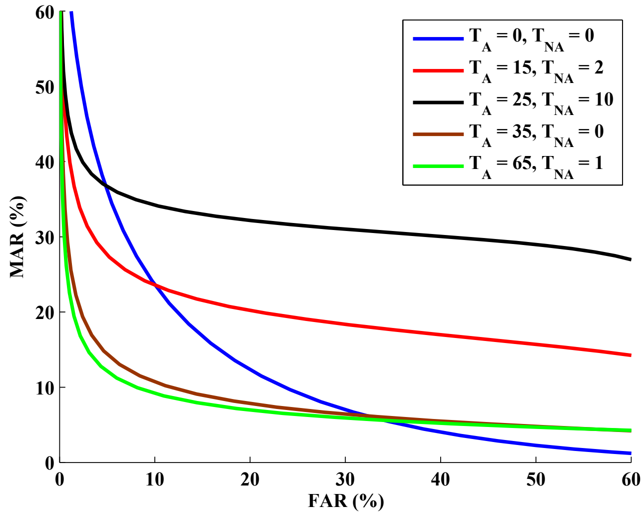


Figure 4.12: ROC curves for the time-deadband configurations

Table 4.2: Performance comparison of the time-deadband configurations

Configuration	**(FAR, MAR)** (%)	**EDD**	**Threshold**
$T_A = 0$, $T_{NA} = 0$	(15.87, 15.87)	0.35	1
$T_A = 15$, $T_{NA} = 2$	(11.43, 22.88)	1.73	2.40
$T_A = 25$, $T_{NA} = 10$	(10.38, 34.10)	2.63	2.60
$T_A = 35$, $T_{NA} = 0$	(8.42, 11.52)	3.43	2.80
$T_A = 65$, $T_{NA} = 1$	(8.17, 9.94)	4.87	3.0

Time-deadband only on no-alarm state

In this case time-deadband is configured only on no-alarm state, i.e., $T_{NA} \neq 0$ and $T_A = 0$. This configuration is useful for the cases where operators can tolerate a waiting-time in the no-alarm state before an alarm can be raised. For design purposes, a range of possible values for T_{NA} are considered, and an algorithm is proposed to compute the percentage alarm reduction after applying a T_{NA} of certain length on the historical alarm data. Algorithm 5 provides the pseudo code of the design procedure for T_{NA}. The algorithm makes use of the *RTN-RTN* run-length based encoded alarm data. While traversing through the encoded alarm data, the alarm count is reduced by one every time if the condition on line 6 of the nested while loop is satisfied.

Algorithm 5 Applying time-deadband only on no-alarm state ($T_{NA} \neq 0$, $T_A = 0$)

Inputs: *RTN-RTN* run-length encoded alarm sequence, T_{NA}
Output: Alarm reduction

```
1:  procedure ALARM REDUCTION(RTN-RTN, T_NA)
    Initialization
2:  n = length(RTN-RTN)                          ▷ initial alarm count
3:  temp = 0                                     ▷ temporary buffer
    Computation
4:  while i ≤ length(RTN-RTN)
5:      temp = RTN-RTN(i)
6:      while T_NA ≥ temp
7:          n = n - 1                            ▷ decrease in alarm count
8:          i = i + 1
9:          temp = temp + RTN-RTN(i)
10:     end while
11:     i = i + 1
12: end while
13: end procedure
```

Time-deadband only on alarm state

In this case operators can tolerate a minimum waiting-time only on the alarm state; however, an alarm should be raised without any delay, i.e., $T_A \neq 0$, and $T_{NA} = 0$. This case is applicable for very critical process variables, where no delay in raising the alarm can be afforded. A design procedure based on *ALM-ALM* run-length encoded alarm data is proposed for designing a recommended length of T_A. The pseudo code of the procedure is shown in Algorithm 6. Similar to Algorithm 5, the alarm count is reduced if condition on line 6 of the nested loop is satisfied.

Time-deadband on both alarm and no-alarm states

This case applicable for the process variables, where operators can afford to have waiting times on both alarm and no-alarm states. A design procedure of a time-deadband configuration with $T_{NA} \neq 0$, and $T_A \neq 0$ is proposed in Algorithm 7. In this case, lossless run-length encoding (LRLE) scheme of the alarm data is considered. Two nested while loops are used to calculate the alarm count reduction. In this algorithm $Mod(.)$ represents the Modulo operation.

Algorithm 6 Applying time-deadband only on alarm state ($T_A \neq 0$, $T_{NA} = 0$)

Inputs: *ALM-ALM* run-length encoded alarm sequence, T_A
Output: Alarm reduction

```
procedure ALARM REDUCTION(ALM-ALM, T_A)
Initialization
n = length(ALM-ALM)                       ▷ initial alarm count
temp = 0                                  ▷ temporary buffer
Computation
while i ≤ length(ALM-ALM)
    temp = ALM-ALM(i)
    while T_A ≥ temp
        n = n - 1                         ▷ decrease in alarm count
        i = i + 1
        temp = temp + ALM-ALM(i)
    end while
    i = i + 1
end while
end procedure
```

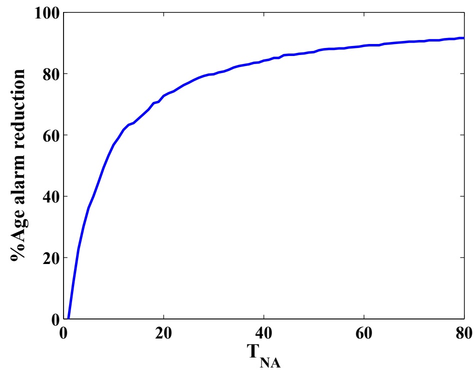
For illustration purposes, two application examples are considered. In the first example alarm data with 5000 samples is collected from a simulated process variable following Gaussian distribution. Number of alarms were observed to be over 600. Algorithms 5, 6, and 7 are used to find the percentage alarm reduction with application of various time-deadband configurations. Figure 4.13 shows the case where one of the time-deadband parameters is fixed to be zero, and Figure 4.14 is obtained when both of the time-deadband parameters (T_{NA} and T_A) are considered. Percentage alarm reduction along with the chattering indices for various time-deadband configurations are listed in Table 4.3.

For the second example, an industrial alarm data for a period of 25 days was collected. A snapshot of a few samples of the alarm sequence is shown in Figure 4.15. For the considered alarm sequence, the alarm count was observed to be 1387, and the chattering index was calculated to be 0.2538 alarms/sec, which is above the cut off value of 0.05 alarms/sec, indicating the problem of severe chattering. All three design scenarios are considered, and the plots for percentage alarm reductions are obtained. Figure 4.16a shows the curve of percentage alarm reduction when only T_{NA} is applied on the alarm data, while keeping the $T_A = 0$. From this figure it can be seen that the alarm

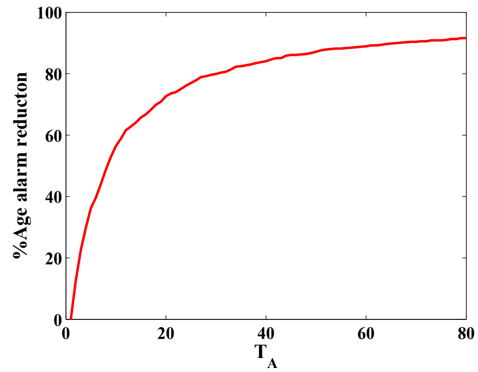
Algorithm 7 Applying time-deadband on both alarm and no-alarm states ($T_A \neq 0$, $T_{NA} \neq 0$)

Inputs: Lossless run-length encoded alarm sequence, T_A, T_{NA}, alarm count
Output: Alarm reduction

```
procedure ALARM REDUCTION(LRLE, T_A, T_NA)
Initialization
n = alarm count                                  ▷ initial alarm count
flag = 0                                         ▷ flag for mode selector
temp = 0                                         ▷ temporary buffer
Computation
while i ≤ length(LRLE)
    temp = LRLE(i)
    if flag == 0                                 ▷ alarm data is in no-alarm state
        while T_NA ≥ temp
            i = i + 1
            temp = temp + LRLE(i)
            if (Mod(i,2) == 1)
                n = n - 1
                i = i + 1
                temp = temp + LRLE(i)
            end if
        flag = 1
        end while
    i = i + 1
    temp = temp - T_NA
    else                                         ▷ alarm data is in alarm state
        while T_A ≥ temp
            i = i + 1
            temp = temp + LRLE(i)
            if (Mod(i,2) == 0)
                n = n - 1
                i = i + 1
                temp = temp + LRLE(i)
            end if
        flag = 0
        end while
    i = i + 1
    temp = temp - T_A
    end if
end while
end procedure
```



(a) Alarm reduction based on T_{NA}



(b) Alarm reduction based on T_A

Figure 4.13: Percentage alarm count reduction (Example 1)

count reduces as the length of T_{NA} is increased. In this figure a marked point indicate that the alarm count reduction is observed to be 83.33% for $T_{NA} = 22$. Similarly, an alternative configuration is proposed by considering only T_A while keeping the $T_{NA} = 0$. Figure 4.16b shows the percentage alarm count reduction for the range of T_A, e.g., T_A of length 27 results in 85.79% reduction in the alarm count. For both of these marked cases, the chattering index is calculated to be 0.031 alarms/sec, which is acceptable according the standards. Figure 4.17 shows the case where the design of both T_A and T_{NA} is considered. The color bar indicates the percentage alarm reduction, when waiting times of particular lengths are configured on both the alarm and no-alarm states. Table 4.3 lists some configurations along with the resultant chattering index, and percentage alarm reduction. From this table it can be seen that the configurations ($T_A = 15$, $T_{NA} = 20$) and ($T_A = 24$, $T_{NA} = 27$) show superior performance as compared to other listed configurations.

It is worth mentioning here that the proposed design procedures provide a set of possible time-deadband configurations to consider. Depending on the type of the process variable, its criticality, and the signal to noise ratio of the process variable, one may choose the one that suits best for the conditions.

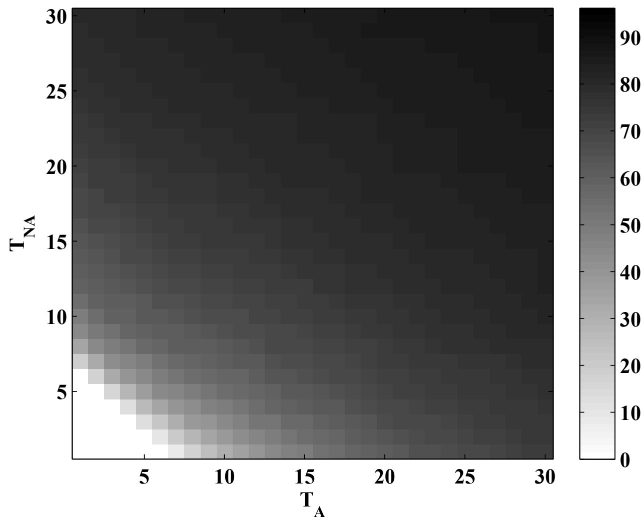


Figure 4.14: Alarm reduction based on both T_{NA} and T_A (Example 1)

4.6 Summary

In this chapter time-deadband configurations for univariate alarm systems have been studied. A Markov process based model was developed under the assumption that the distributions of the process variable are known for both the normal and abnormal operations. Performance indices, namely, the false alarm rate, the missed alarm rate, and the expected detection delay have been calculated by studying the long-term behavior of the Markov model. Design procedures based on the process data, and the alarm data have been developed, to help the operators in achieving acceptable alarm system performance.

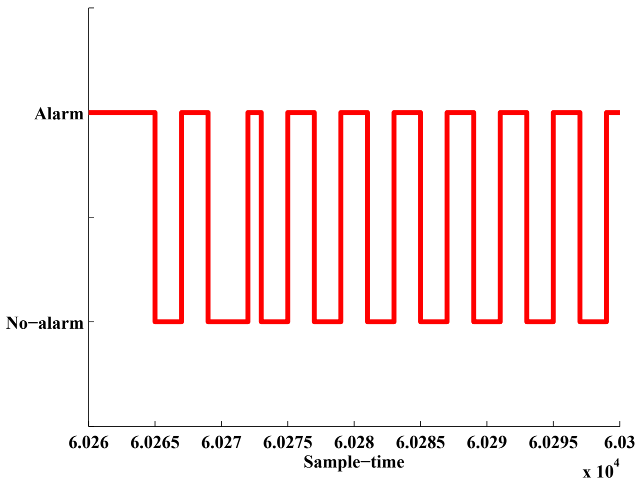
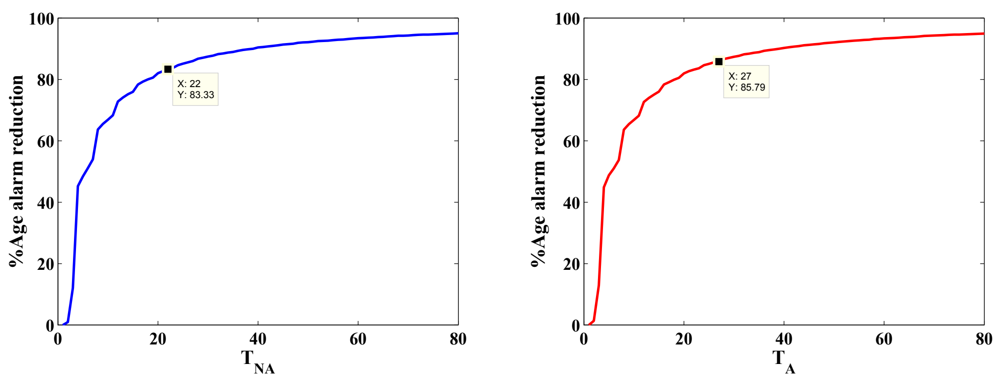


Figure 4.15: Snapshot of the alarm sequence



(a) Alarm reduction based on T_{NA} (b) Alarm reduction based on T_A

Figure 4.16: Percentage alarm count reduction (Example 2)

Table 4.3: Performance comparison of time-deadband configuration

	Configuration	**Alarm count**	**Chattering index**	**Alarm reduction** (%)
Example 1	$T_A = 0$, $T_{NA} = 0$	678	0.2202	–
	$T_A = 0$, $T_{NA} = 30$	132	0.0268	80.53
	$T_A = 40$, $T_{NA} = 0$	105	0.0213	84.51
	$T_A = 10$, $T_{NA} = 15$	152	0.0312	77.58
Example 2	$T_A = 0$, $T_{NA} = 0$	1387	0.2538	–
	$T_A = 15$, $T_{NA} = 20$	118	0.0253	91.49
	$T_A = 6$, $T_{NA} = 11$	277	0.0481	83.63
	$T_A = 5$, $T_{NA} = 4$	464	0.0812	66.55
	$T_A = 24$, $T_{NA} = 27$	106	0.0179	92.36

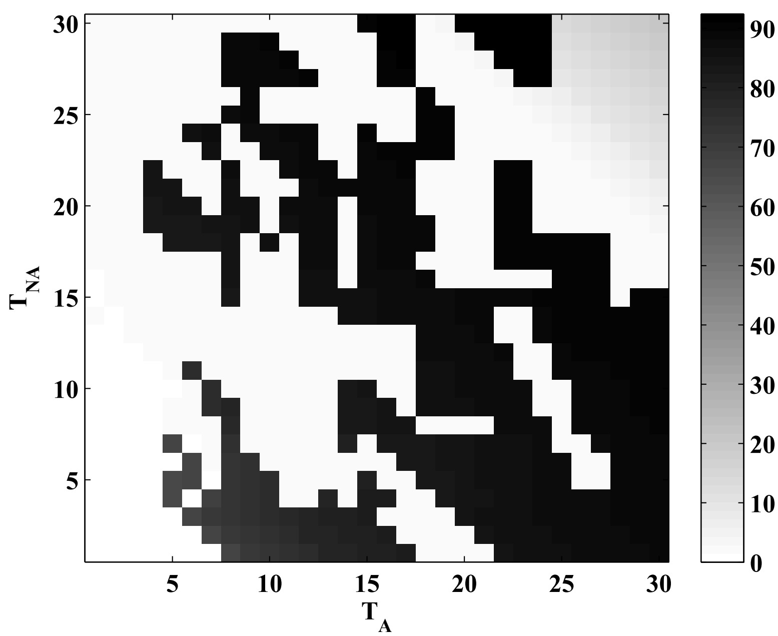


Figure 4.17: Alarm reduction based on both T_{NA} and T_A (Example 2)

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The outcomes of this thesis are summarized as follows:

1. A multivariate alarm system was designed for multi-mode processes. A hidden Markov model approach was adopted to capture the multi-modality of data and the mode-reachability constraints of a multi-mode process. A two-step Viterbi algorithm was proposed to detect the operating mode and a contribution plots based fault isolation scheme was developed. Superior results were obtained for the faulty scenarios under which mode-reachability constraints were violated.

2. The problem of designing delay-timers for multi-mode processes was considered. A hidden Markov model with Markov chain observations was used to model mode changes and delay-timer configurations for various modes. Performance indices (the false alarm rate, missed alarm rate, and the expected detection delay) were calculated, and a particle swarm optimization based algorithm was proposed for designing delay-timers.

3. For univariate alarm systems, the problem of analyzing and designing time-deadbands was considered. Design procedures based on process data and alarm data were proposed. The process data design procedure made use of developed analytical expressions of the performance indices,

and ROC curves were used to make recommendations on the possible lengths of time-deadbands. Run-length encoding schemes for alarm signals were used to develop algorithms for designing time-deadbands based on alarm data.

5.2 Future work

The following are some of the possible future research problems, which are worth investigating:

1. For the design of multivariate alarm systems for multi-mode processes, only steady state behavior of the multi-mode processes was considered. However, for processes that experience long transition periods while switching between modes, the transitional changes should also be part of the model. One possible solution is to use semi-hidden Markov processes in the model, that can be used to capture the transitional changes between different modes of operation, and the time taken by the process in switching from one mode to another. Accurate mode detection along with detection of transitional phase between different modes is another challenging task for such processes.

2. Delay-timers and deadbands are among the most commonly used techniques in industries to remove chattering alarms. In this thesis, both delay-timers and deadbands configurations were studied; however, a performance comparison between these two alarm configurations was not conducted. A possible future work is to perform comparative analysis of various alarm configuration methods including delay-timers and deadbands both qualitatively and quantitatively, to give better recommendations for alarm system performance improvement. Different operating scenarios and modes (e.g., start-up, shutdown etc.) and types of process variables (e.g., flow, temperature, pressure etc.) should also be taken into account.

3. Most of the literature related to alarm system design consider the application of only one alarm configuration technique at a time. The usage of ensemble methods by considering combinations of various alarm configuration techniques (e.g., filters and deadbands or filters and delay-timers etc.) should be explored to achieve better trade-off between performance indices.

Bibliography

[1] Adnan, N. A., Cheng, Y., Izadi, I., and Chen, T. (2013). Study of generalized delay-timers in alarm configuration. *Journal of Process Control*, 23(3):382–395.

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

[3] Afzal, M. S. and Chen, T. (2017). Analysis and design of multimode delay-timers. *Chemical Engineering Research and Design*, 120:179–193.

[4] Alshraideh, H. and Runger, G. (2014). Process monitoring using hidden Markov models. *Quality and Reliability Engineering International*, 30(8):1379–1387.

[5] Atlas, L., Ostendorf, M., and Bernard, G. D. (2000). Hidden Markov models for monitoring machining tool-wear. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 6, pages 3887–3890. IEEE.

[6] Basseville, M. and Nikiforov, I. V. (1993). *Detection of abrupt changes: theory and application*, volume 104. Prentice Hall Englewood Cliffs.

[7] Baum, L. E. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563.

[8] Chen, R. (1985). Solution of minimax problems using equivalent differentiable functions. *Computers & Mathematics with Applications*, 11(12):1165–1169.

[9] Cheng, Y., Izadi, I., and Chen, T. (2011). On optimal alarm filter design. In *International Symposium on Advanced Control of Industrial Processes (ADCONIP)*, pages 139–145. IEEE.

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

[11] Eberhart, R. C. and Shi, Y. (1998). Comparison between genetic algorithms and particle swarm optimization. In *Evolutionary Programming VII*, pages 611–616. Springer.

[12] Eberhart, R. C. and Shi, Y. (2001). Particle swarm optimization: developments, applications and resources. In *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 81–86. IEEE.

[13] Equipment & Materials Users' Association (EEMUA), E. (2007). *Alarm Systems: A Guide to Design, Management and Procurement.* EEMUA Publication 191.

[14] Fan, S.-K. S. and Zahara, E. (2007). A hybrid simplex search and particle swarm optimization for unconstrained optimization. *European Journal of Operational Research*, 181(2):527–548.

[15] Hollifield, B., Habibi, E., and Pinto, J. (2011). *Alarm Management: A Comprehensive Guide.* International Society of Automation.

[16] Hu, W., Wang, J., and Chen, T. (2015). A new method to detect and quantify correlated alarms with occurrence delays. *Computers & Chemical Engineering*, 80:189–198.

[17] Hu, Z., Yin, L., Chen, Z., Gui, W., Yang, C., and Peng, X. (2013). An efficient multi-PCA based on-line monitoring scheme for multi-stages imperial smelting process. *International Journal of Control, Automation, and Systems*, 11(2):317.

[18] Hugo, A. (2009). Estimation of alarm deadbands. In *Proceedings of 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pages 663–667.

[19] Hwang, S.-L., Lin, J.-T., Liang, G.-F., Yau, Y.-J., Yenn, T.-C., and Hsu, C.-C. (2008). Application control chart concepts of designing a pre-alarm system in the nuclear power plant control room. *Nuclear Engineering and Design*, 238(12):3522–3527.

[20] International Society of Automation (ISA) (2000). Enterprise Control System Integration, ANSI/ISA 95.

[21] International Society of Automation (ISA) (2009). Management of Alarm Systems for the Process Industries, ANSI/ISA 18.2.

[22] Inuiguchi, M. and Sakawa, M. (1995). Minimax regret solution to linear programming problems with an interval objective function. *European Journal of Operational Research*, 86(3):526–536.

[23] Isermann, R. (2005). Model-based fault-detection and diagnosis–status and applications. *Annual Reviews in Control*, 29(1):71–85.

[24] Izadi, I., Shah, S., Shook, D., and Chen, T. (2009). An introduction to alarm analysis and design. In *Proceedings of 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pages 645–650.

[25] Jackson, J. E. (2005). *A user's guide to principal components*, volume 587 of *Wiley Series in Probability and Statistics*. John Wiley & Sons.

[26] Jang, G.-S., Suh, S.-M., Kim, S.-K., Suh, Y.-S., and Park, J.-Y. (2013). A proactive alarm reduction method and its human factors validation test for a main control room for SMART. *Annals of Nuclear Energy*, 51:125–134.

[27] Jerhotova, E., Sikora, M., and Stluka, P. (2012). Dynamic alarm management in next generation process control systems. In *IFIP International Conference on Advances in Production Management Systems*, pages 224–231. Springer.

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

[29] Kennedy, J. (2011). Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer.

[30] Kondaveeti, S. R., Izadi, I., Shah, S. L., and Chen, T. (2011). On the use of delay timers and latches for efficient alarm design. In *19th IEEE Mediterranean Conference on Control & Automation*, pages 970–975.

[31] Kondaveeti, S. R., Izadi, I., Shah, S. L., Shook, D. S., Kadali, R., and Chen, T. (2013). Quantification of alarm chatter based on run length distributions. *Chemical Engineering Research and Design*, 91(12):2550–2558.

[32] Kotz, S. and Johnson, N. L. (1988). *Encyclopedia of Statistical Sciences*. Wiley, New York.

[33] Laberge, J. C., Bullemer, P., Tolsma, M., and Dal Vernon, C. R. (2014). Addressing alarm flood situations in the process industries through alarm summary display design and alarm response strategy. *International Journal of Industrial Ergonomics*, 44(3):395–406.

[34] Lai, S. and Chen, T. (2015). A method for pattern mining in multiple alarm flood sequences. *Chemical Engineering Research and Design*, 117:831–839.

[35] Lawler, G. F. (2006). *Introduction to Stochastic Processes*. CRC Press.

[36] Liu, J. (2012). Fault diagnosis using contribution plots without smearing effect on non-faulty variables. *Journal of Process Control*, 22(9):1609–1623.

[37] Livingston, J. (2015). The nuclear electrical engineer, an educational resource for electrical engineers in the nuclear power industry. http://www.nuclearelectricalengineer.com/.

[38] MacGregor, J. and Kourti, T. (1995). Statistical process control of multivariate processes. *Control Engineering Practice*, 3(3):403–414.

[39] Martin, E. and Morris, A. (1996). Non-parametric confidence bounds for process performance monitoring charts. *Journal of Process Control*, 6(6):349–358.

[40] Miao, H., Sforna, M., and Liu, C.-C. (1996). A new logic-based alarm analyzer for on-line operational environment. *IEEE transactions on Power Systems*, 11(3):1600–1606.

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

[42] Nihlwing, C. and Kaarstad, M. (2012). The development and usability test of a state based alarm system for a nuclear power plant simulator. *Proc. NPIC and HMIT*, pages 22–26.

[43] Ning, C., Chen, M., and Zhou, D. (2014). Hidden Markov model-based statistics pattern analysis for multimode process monitoring: An index-switching scheme. *Industrial & Engineering Chemistry Research*, 53(27):11084–11095.

[44] Ning, C., Chen, M., and Zhou, D. (2015). Sparse contribution plot for fault diagnosis of multimodal chemical processes. *IFAC-PapersOnLine*, 48(21):619–626.

[45] Patton, R. and Chen, J. (1997). Observer-based fault detection and isolation: robustness and applications. *Control Engineering Practice*, 5(5):671–682.

[46] Qin, S. J., Yue, H., and Dunia, R. (1997). Self-validating inferential sensors with application to air emission monitoring. *Industrial & Engineering Chemistry Research*, 36(5):1675–1685.

[47] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[48] Rashid, M. M. and Yu, J. (2012). Hidden Markov model based adaptive independent component analysis approach for complex chemical process monitoring and fault detection. *Industrial & Engineering Chemistry Research*, 51(15):5506–5514.

[49] Shi, Y. and Eberhart, R. C. (1999). Empirical study of particle swarm optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC)*, volume 3. IEEE.

[50] Simeu-Abazi, Z., Lefebvre, A., and Derain, J.-P. (2011). A methodology of alarm filtering using dynamic fault trees. *Reliability Engineering & System Safety*, 96(2):257–266.

[51] Song, B., Shi, H., Ma, Y., and Wang, J. (2014). Multisubspace principal component analysis with local outlier factor for multimode process monitoring. *Industrial & Engineering Chemistry Research*, 53(42):16453–16464.

[52] Strachan, I. and Clifton, D. (2009). A hidden Markov model for condition monitoring of a manufacturing drilling process. *IET Condition Monitoring*, pages 803–814.

[53] Tan, S., Wang, F., Peng, J., Chang, Y., and Wang, S. (2011). Multimode process monitoring based on mode identification. *Industrial & Engineering Chemistry Research*, 51(1):374–388.

[54] Tan, W., Sun, Y., Azad, I. I., and Chen, T. (2017). Design of univariate alarm systems via rank order filters. *Control Engineering Practice*, 59:55–63.

[55] Thornhill, N. (2013). The continuous stirred tank heater simulation. http://personal-pages.ps.ic.ac.uk/ nina/CSTHSimulation/index.htm.

[56] Thornhill, N. F., Patwardhan, S. C., and Shah, S. L. (2008). A continuous stirred tank heater simulation model with applications. *Journal of Process Control*, 18(3):347–360.

[57] U.S. Nuclear Regulatory Commission (2014). Background on the three mile island accident. http://www.nrc.gov/reading-rm/doc-collections/fact-sheets/3mile-isle.html.

[58] Varga, T., Szeifert, F., and Abonyi, J. (2009). Detection of safe operating regions: a novel dynamic process simulator based predictive alarm management approach. *Industrial & Engineering Chemistry Research*, 49(2):658–668.

[59] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., and Yin, K. (2003). A review of process fault detection and diagnosis: Part III: Process history based methods. *Computers & Chemical Engineering*, 27(3):327–346.

[60] Wang, F., Tan, S., Peng, J., and Chang, Y. (2012). Process monitoring based on mode identification for multi-mode process with transitions. *Chemometrics and Intelligent Laboratory Systems*, 110(1):144–155.

[61] Wang, J. and Chen, T. (2014). An online method to remove chattering and repeating alarms based on alarm durations and intervals. *Computers & Chemical Engineering*, 67:43–52.

[62] Westerhuis, J. A., Gurden, S. P., and Smilde, A. K. (2000). Generalized contribution plots in multivariate statistical process monitoring. *Chemometrics and Intelligent Laboratory Systems*, 51(1):95–114.

[63] Xu, X., Xie, L., and Wang, S. (2014). Multimode process monitoring with PCA mixture model. *Computers & Electrical Engineering*, 40(7):2101–2112.

[64] Yao, M., Wang, H., and Xu, W. (2014). Batch process monitoring based on functional data analysis and support vector data description. *Journal of Process Control*, 24(7):1085–1097.

[65] Yin, S., Ding, S. X., Haghani, A., Hao, H., and Zhang, P. (2012). A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. *Journal of Process Control*, 22(9):1567–1581.

[66] Yu, J. (2010). Hidden Markov models combining local and global information for nonlinear and multimodal process monitoring. *Journal of Process Control*, 20(3):344–359.

[67] Yu, J. (2012). Local and global principal component analysis for process monitoring. *Journal of Process Control*, 22(7):1358–1373.

[68] Yu, J. (2013). A new fault diagnosis method of multimode processes using Bayesian inference based Gaussian mixture contribution decomposition. *Engineering Applications of Artificial Intelligence*, 26(1):456–466.

[69] Yu, J. and Qin, S. J. (2008). Multimode process monitoring with Bayesian inference-based finite Gaussian mixture models. *AIChE Journal*, 54(7):1811–1829.

[70] Yu, J. and Qin, S. J. (2009). Multiway Gaussian mixture model based multiphase batch process monitoring. *Industrial & Engineering Chemistry Research*, 48(18):8585–8594.

[71] Yu, Y., Zhu, D., Wang, J., and Zhao, Y. (2017). Abnormal data detection for multivariate alarm systems based on correlation directions. *Journal of Loss Prevention in the Process Industries*, 45:43–55.

[72] Yue, H. H. and Qin, S. J. (2001). Reconstruction-based fault identification using a combined index. *Industrial & Engineering Chemistry Research*, 40(20):4403–4414.

[73] Zhang, F. (2004). A mixture probabilistic PCA model for multivariate processes monitoring. In *Proceedings of the American Control Conference (ACC)*, volume 4, pages 3111–3115. IEEE.

[74] Zhao, S. J., Zhang, J., and Xu, Y. M. (2004). Monitoring of processes with multiple operating modes through multiple principle component analysis models. *Industrial & Engineering Chemistry Research*, 43(22):7025–7035.

[75] Zhu, J., Ge, Z., and Song, Z. (2014). Robust modeling of mixture probabilistic principal component analysis and process monitoring application. *AIChE Journal*, 60(6):2143–2157.

[76] Zucchini, W. and MacDonald, I. L. (2009). *Hidden Markov Models for Time Series: An Introduction Using R*. CRC press.