Monitoring of Industrial Processes via Non-stationary Probabilistic Slow Feature Analysis Machine Learning Algorithm

by

David Scott

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

Master of Science

in

Process Control

Department of Chemical and Materials Engineering

University of Alberta

Abstract

Multivariate statistical process monitoring (MSPM) methods provide representations of current process conditions utilizing data-driven models that become founded on apriori information. Large scale industrial processes are subject to time-variant conditions that result in non-stationary characteristics. Classical MSPM methods, such as principal component analysis (PCA) and canonical variate analysis (CVA) have difficulties in monitoring non-stationary processes, therefore, in this Thesis, to overcome this predicament, the non-stationary probabilistic slow feature analysis (NS-PSFA) algorithm is proposed. The advantage of the NS-PSFA algorithm, in contrast to classical MPSM methods, is that it can extract underlying insights from non-stationary process data. NS-PSFA extracts information from data as slow features (SFs). The expectation-maximization (EM) algorithm is used to estimate the model parameters of the NS-PSFA algorithm. Then, the estimated model is used to derive control limits for monitoring. Besides, for the online implementation of the proposed method, the Kalman filter is employed to predict SFs. Predicted SFs are utilized to calculate monitoring statistics that indicate process abnormalities. The proposed method has better performance in non-stationary applications such as a continuous stirred tank reactor (CSTR) and a three-phase flow industrial process in comparison with the CVA and PCA methods.

Acknowledgements

I would like to acknowledge the motivation and challenge provided by my Professor, Dr. Biao Huang. Dr. Huang pushed me out of my comfort zone, which allowed me to set personal and professional goals, overcoming limitations I once had. Without his support, there is no chance I would have been as successful as I have been to this point, with a very bright future ahead of me. A second person I would like to thank in the Department of Chemical and Materials Engineering is Marnie Jamieson. Marnie provided me with opportunities to grow and better myself through my Master's program, including being a Teaching Assistant for the Chemical Engineering Capstone Course for two years, as well as helping judge and facilitate the Engineering Commerce Case Competition for two years. Marnie's feedback and her outlook was crucial in multiple decisions I made and her support was of utmost importance in receiving the Alexander Graham Bell Canada Graduate Scholarship - Master's Scholarship - the highest level of Scholarship a Master's student in Canada can achieve.

A big thanks to a Mentor I met in my Undergraduate degree, Jun Liu. Jun encouraged me to be curious and pursue more than a Bachelor's Degree. Without his motivation to dig deeper into data analytics and machine learning, I would have been truly lost and not find my calling: an inspiration from emerging technologies.

Many thanks goes to Mentor's that I met during my time as a Master's student. Specifically, Kris Sommerstad, whom I met volunteering with APEGA and Ray Nelson, a mentor I had made through an APEGA connection. Kris and I bonded over the similar passion of educating youth at a young age, giving them hands-on STEM demonstrations, Squishy Circuits, to help inspire them to develop curiosity regarding science. However, my connection with Kris provided me with many other strengths gained, as well as maturity. Ray provided me with personal and professional life advice to help me think outside the box and discover my talents. Learning that my

top personality traits were an achiever and a learner, helped me prioritize my workload and give my body the right fuel it needed to make myself happy, healthy, and intelligent.

My countless friends, I've met along my journey during my Master's and friends along the way who have helped me. Specifically at the University of Alberta Computer Process Control Engineering Lab, I'd like to thank Alireza Kheradmand, Hossein Shahandeh, Agustin Vicente, Seraphina Kwak (thanks for half a blizzard!), Yashas Mohan, Hareem Shafi, Sanjula Kammammettu, Lei Fan, Mengqi Fang, Yanjun Ma, Rui Nian, Song Bo, Yue Cao, Ranjith Chiplunkar, Rahul Raveendran, Yousef Alipouri and others who helped uplift my professional and personal journey during my Master's program. My friends outside of my Master's, Robert Fedorak, Alexander Gee, Zachary Sinclair, Kyler Guebert, Logan Purdy, and others who provided me motivation and pushed me when I needed extra fuel.

A special thanks goes out to Xintong Li; thanks for helping to open my eyes to a more creative life and cultivating future experiences.

My various Co-founders and group members of the Indigenous Engineering Students' Association and Indigenous Graduate Students' Association. Specifically, Lucy Kootenay, Gabreal Chaumbaud, and Teddy Carter. Thank you for helping me cofound these student groups and making the University of Alberta a more inclusive and culturally inspiring environment for Indigenous Peoples.

My Professors and friends made during my Academic Exchange at Tsinghua University. Many thanks goes to the support and motivation provided by Professor Chao Shang. My friends, namely Liang Cao, Rui Gaoxin, Tony Zhang who helped me make too many amazing memories of my life in China, as well as helping me adjust to Chinese language and culture. My friends who helped my teams win three different international Hackathons: Zhanwei Xu, Hao Huanghong, Shu Li, Rui Gaoxin, and others, without our teamwork and determination, my current opportunities that I am receiving would not be available to me.

Finally, and most importantly, my parents. My father, Barry, instilled hard work, discipline, and determination into my life. My mother, Pam, is likely the kindest and most selfless person I've ever met. She instilled the optimism, creativity, empathy, and kindness that I utilize to uplift myself and others in my life, every day. What I

feel these two gave me was a winning skillset. No matter how bad my day can be, I'll be able to wake up every day being optimistic for the future since I know I'll set my goals, which will be hard, and I will achieve them - swiftly while learning, and having fun.

Contents

List of Tables

List of Figures

Chapter 1

Overview

1.1 Introduction

While operating an industrial process, the detection and mitigation of critical process upsets, equipment faults, and temporary process disturbances are essential to ensure efficient and reliable operations as well as high-quality products [1]. The detection of certain unusual events while operating must be accomplished timely to resolve and mend the problems. However, facing the complexity of large-scale industrial processes, it is challenging to distinguish unusual events as influences on process data occur from underlying complex correlations and interconnections between variables. Hence, disentangling individual complex connections between process variables is imperative to leverage process data for data-driven and data-based process monitoring, including the detection and mitigation of unusual events [2].

Multivariate statistical process monitoring (MSPM) methods have gained attention and importance in fault detection operations. Commonly used MSPM methods comprise of principal component analysis (PCA) [3], independent component analysis (ICA) [4], partial least squares, (PLS) [5], [6], and their respective extensions [7, 8]. These methods have achieved success in recent years for fault detection applications in industrial manufacturing processes; however, they cannot distinguish between operation condition changes and process abnormalities.

By identifying this problem of differentiation, Shang et al. [9] proposed to utilize slow feature analysis (SFA) [10] to establish two novel monitoring statistics tailored to distinguish between operation condition deviations and process abnormalities. In an extension of the use of this added monitoring statistic, other false alarms observed

from different MSPM methods could be identified. As a result, SFA has been studied and employed in various applications [11, 12, 13]. Recently, a developed probabilistic counterpart of SFA called probabilistic slow feature analysis (PSFA) was adopted to provide a compact state-space description for process dynamics [14, 15], which turns out to be advantageous for handling process noise and missing data in regular operations.

In essence, for nominal operation data, classical MSPM methods are defined, and any deviations from the nominal operation are deemed to be abnormal and call for further attention. However, dynamic industrial processes that contain time-variant conditions possess non-stationary behavior. It is challenging for classical MSPM methods to differentiate between normal and non-stationary process conditions. As a result, false alarms occur under normal process variations and serious faults become challenging to differentiate from operation condition changes.

A fundamental limitation of SFA and PSFA is that the underlying distribution of observed data must be *stationary* in that its statistical properties do not vary over time. However, it is often the case that large industrial processes with upstream and downstream process units are affected by non-stationary process dynamics. As a result, numerous process variables are non-stationary, which have statistical properties that vary with time [16]. Under such circumstances, traditional MSPM methods lack the strength to monitor non-stationary processes, which in turn raise false alarms [17]. Several methods have been utilized to tackle modeling of non-stationary dynamic processes including, cointegration analysis (CA) methods [17, 18], data-differencing methods [19], and adaptive modeling strategies [20, 21, 22]. However, these methods are inadequate in the description of stationary process dynamics.

In this Thesis, It is proposed to utilize the non-stationary PSFA (NS-PSFA) algorithm to address both non-stationary and stationary process dynamics. The Augmented Dickey-Fuller (ADF) test first categorizes process variables being non-stationary or stationary, which are used to develop the NS-PSFA model from operational data. Stable auto-regressive processes and random walks model stationarity and non-stationarity, respectively. Then a modified EM algorithm is developed for parameter estimation of the NS-PSFA model. Using the NS-PSFA model, control limits are calculated and used to differentiate between process conditions.

In the online implementation, to obtain monitoring statistics, the SF values are predicted via the Kalman filter. SF values provide a monitoring scheme that can detect abnormalities in non-stationary processes. By synthesizing information from monitoring statistics, process practitioner's decision-making is aided to distinguish between normal and abnormal operations, thereby diminishing false alarms.

The remainder of this Thesis is organized as follows. Chapter 2 briefly revisits SFA and its methodologies. Chapter 3 introduces the NS-PSFA model, parameter estimation strategy via EM algorithm, design of monitoring statistics and a comprehensive monitoring scheme. Two case studies on a simulated example and an industrial three-phase flow process show the effectiveness of the proposed methodology. Chapter 3 visits an industrial application utilizing electric submersible pumps (ESPs). Chapter 4 draws the conclusion and recommendations.

Chapter 2

Introduction to slow feature analysis

2.1 Motivation and Introduction to Utilizing Slow Feature Analysis

In industrial processes, sources of disturbances can be challenging to pinpoint due to hidden underlying features. Hidden underlying features refer to hidden insights contained within data that cannot be easily obtained and utilized while operating a process. By extracting this information from data, and performing analysis, deep insights into a process can be harnessed and utilized after that, generating advantages. A previously well studied technique that has extracted and utilized hidden underlying insights in data is soft-sensors. Soft-sensing is utilizing a process modeling technique to predict hard-to-measure process variables [23], [24], such as density in an oil & gas operation and temperature of the metal in additive manufacturing. While operating an industrial process, difficult to measure variables are frequently required to guarantee product quality and operation efficiency. With this information, industrial practitioners must acquire strategies from process data to unlock competitive advantages from their industrial processes, which can be first done by deriving effective models from their data.

A designer meets two choices in process modeling. First, there is first-principle modeling, which requires intimate domain knowledge about a process [25]. Second, data-driven modeling employs data to generate insights into a process without requiring subject matter expertise. An additional obstacle regarding first-principle model-

ing requires an intimate knowledge of interconnections within process variables, which complicate matters further. Due to the simplicities of data-driven modeling, many industries have adopted this method to secure competitive advantages from their data; thus, data science and machine learning are gaining popularity in recent years [26], [27], [28].

Financial systems are well-known for containing deeply interconnected data between variables [29]. Due to these interconnections, it is challenging to isolate and disentangle data, complicating the capability to generate useful features from data. Machine learning engineers and data scientists aim to overcome issues of high interconnection in data by utilizing exceptional techniques that can decorrelate variables to extract information from the underlying data. Causality refers to the interconnection of data, where a cause and effect relationship exists. For causality, a cause and effect relationship is seen with two process variables; for example, x, and y, where x causes y [30].

In industrial processes, severe fluctuations as disturbances can propagate through the process. Disturbances that remain untreated may affect controlled variables and manipulated variables, both directly and indirectly. For a control system in an industrial process, such as a distributed control system (DCS), the control system can compensate for disturbances. A DCS can adjust manipulated variables in a process to react to disturbances. Manipulated variables are also adjusted to track target set points for a process, such as a liquid level in a process drum, which can have a relationship in achieving another process goal, such as product quality. For processes, variables such as the liquid level of a process drum can affect product quality. Correlation of process data is important and common due to correlation across many variables. Extraction models are used to uncorrelated data with strong interconnection and find independent features of underlying process data. An example of an extraction model is a latent variable (LV) model, such as SFA. SFA is an unsupervised machine learning algorithm utilized, much like PCA, that is used to uncorrelate data [10]. SFA is a useful algorithm for process data analysis and dimensionality reduction [2]. For the development of data-driven soft sensors, there are examples of LV models that are effective in modeling by utilizing dimensionality reduction algorithms. These machine learning algorithms include partial least squares (PLS) [5],

[6], principal component regression (PCR) [31], [32], and their respective extensions, such as probabilistic partial least squares (PPLS) [33].

LV methods require large datasets, such as those from big data, that provide large amounts of data for the training and development of data-driven models. Big data analytics has been an active area of investigation in the machine learning community [2]. As a tool for analytics, machine learning consists of supervised learning and unsupervised learning [34]. Supervised learning assumes a supervisor is available to supervise the learning. Unsupervised learning identifies each pattern without using labels to classify the data [35]. Supervised learning includes algorithms such as support vector machines, neural networks, and classical regression analysis that establish a mapping between an input $\mathbb{X}$ to an output $\mathbb{Y}$ [36].

An important class of machine learning is representation learning. Representation learning establishes a representation of input $\mathbb{X}$ through unsupervised learning, which contributes to creating information for subsequent supervised learning [25, 37]. While utilizing representation learning, it is essentially a LV process to derive a model [27]. The complete advantages of representation learning for industries, in addition to chemical processes, have yet to be fully explored, making it an appealing area for continued research.

Sampling occurs at uniform time intervals and stored to generate features and insights. As a result of fast sampling, data becomes correlated with prior time samples through dynamic connections, including temporal correlations. Temporal correlations are known as auto-correlations, which are indicative of similarities in consecutive data points.

To process data with auto-correlations, we may employ time-series analysis methods, including SFA. In the SFA algorithm, the representation of underlying variations of multivariate process variables is known as slow features (SFs) s [14]. Representation of temporal correlations is illustrated from the following two aspects: [25]

- First, a LV, denoted as $\mathbf{s}(t)$, associated with the current observation, $\mathbf{x}(t)$, and preceding observations $\{\mathbf{x}(t-1), \mathbf{x}(t-2), \cdots\}$.
- $\mathbf{s}(t)$ is correlated to previous values $\{\mathbf{s}(t-1), \mathbf{s}(t-2)\}$.

To extract LVs, input data, $\mathbf{x}(t)$ can be utilized in SFA to extract mutually uncor-

related LVs that have different varying rates. SFA adopts temporal coherence as a heuristic prior to induce LV SFs [10].

2.2 Slow Feature Analysis

The theory and mathematics of the SFA algorithm were established in 2002 by Wiskott and Sejnowski [10]. Computational neuroscience was the first purpose of SFA, to understand causality and interconnections of the visual cortex in the brain [38]. The initial application motivated applications in additional areas including object recognition [39], change detection [40], and nonlinear blind source separation [41]. Fundamental Mathematics and theoretical framework behind classical SFA and its use as an unsupervised dimensionality reduction technique are reviewed [10].

2.2.1 Definition of slowness and notations

Given a stochastic and ergodic signal, $\mathbf{x}(t)$, how slow or fast $\mathbf{x}(t)$ varies is measured as

$$\Delta(\mathbf{x}(t)) \triangleq \langle \dot{\mathbf{x}}^2(t) \rangle_t \tag{2.1}$$

where $\langle . \rangle$ denotes time averaging from N measurement samples $\{\mathbf{x}(t)\}_{t=1}^{\mathrm{N}}$ and $\dot{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}(t-1)$ represents the first order derivative, or time difference of $\mathbf{x}(t)$. The value of $\Delta(.)$ is seen as a measure of how fast the signal $\mathbf{x}(t)$ evolves over time. Given an m dimensional input signal, $\mathbf{x}(t) = [\mathbf{x}_1(t), \cdots, \mathbf{x}_m(t)]^\top$, SFA aims to find a series of SFs $\mathbf{s}(t) = [\mathbf{s}_1(t), \cdots, \mathbf{s}_m(t)]^\top$ where $\mathbf{s}_i(t) = g_i(\mathbf{x}(t))$, such that the value of each $\Delta(s_i)$ is minimal [10]

$$\min_{g_i(.)} \langle \dot{\mathbf{s}}_i^2 \rangle_t, \quad i = 1, \cdots, m \tag{2.2}$$

under the following three constraints

$$\langle s_i \rangle_t = 0, \quad (zero\ mean) \tag{2.3}$$

$$\langle s_i^2 \rangle_t = 1, \quad (unit\ variance) \tag{2.4}$$

$$\forall i \neq j, \quad \langle s_i s_j \rangle_t = 0, \quad (decorrelation\ and\ order) \tag{2.5}$$

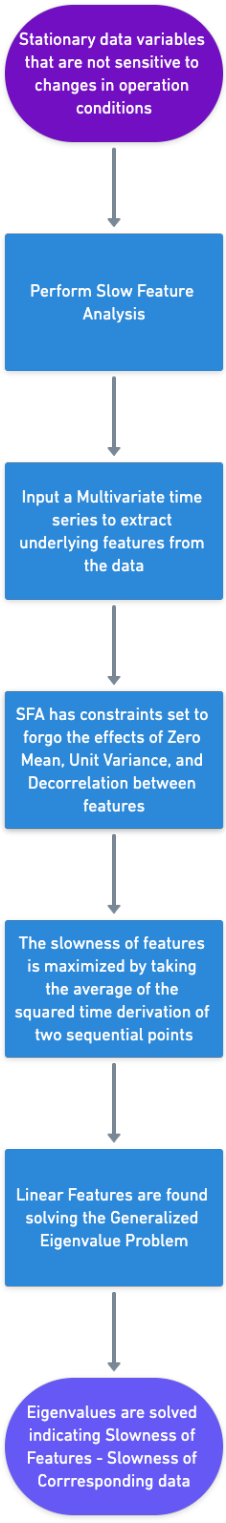


Figure 2.1: Procedure for performing SFA on stationary data

Constraint 2.3 and 2.4 enforce each slow feature, s_i, as zero mean and unit variance to avoid a trivial solution where $s_i \equiv const$. Constraint 2.5 enforces the output of function $\langle g_j \rangle$ to be mutually uncorrelated. A by-product of Constraint 2.5 is that slowness of features is ordered from slowest to fastest, such that s_1 becomes the slowest feature, s_2 as the second slowest, and so forth. The linear SFA (LSFA) algorithm is utilized as a linear extension that can solve for the objective of SFA from the generalized eigenvalue problem. A generalized eigenvalue is from an $[n \times n]$ matrix, which satisfies being in the generalized eigenspace for λ [42].

2.2.2 Linear SFA Algorithm

Linear SFA (LSFA) assumes each SF, $s_i(t)$, is a linear combination of all input variables, $s_i(t) = \mathrm{w}_i^\top \mathbf{x}(t)$. Therefore, mapping the input variables from $\mathbf{x}(t)$ to $\mathbf{s}(t)$ is derived as

$$\mathbf{s}(t) = \mathrm{W}\mathbf{x}(t) \tag{2.6}$$

where $\mathrm{W} = [w_1,\ w_2,\ \cdots,\ w_m]^\top$ is the coefficient matrix to be optimized during linear SFA. To enforce Constraint 2.3 in LSFA, data along each dimension of inputs $\mathbf{x}(t)$, have to be mean-centered in order to satisfy zero mean condition. The optimization problem is solved as a generalized eigenvalue problem utilizing the Lagrange multiplier [10]

$$\mathrm{AW} = \mathrm{BW}\boldsymbol{\Omega} \tag{2.7}$$

where $\mathrm{A} = \langle \dot{\mathbf{x}}\dot{\mathbf{x}}^\top \rangle_t$ denotes the covariance matrix from the first-order derivative of input $\mathbf{x}$, $\dot{\mathbf{x}}$, and $\mathrm{B} = \langle \mathbf{x}\mathbf{x}^\top \rangle_t$ denotes covariance matrix of input $\mathbf{x}$. W contains m generalized eigenvectors from the matrices of A and B, and $\boldsymbol{\Omega} = diag\{\omega_1, \cdots, \omega_m\}$ contains generalized eigenvalues, which happen to be the optimal values of the objective from Equation 2.1 [25].

The schematic of SFA is shown in Figure 2.1. First, stationary variables are utilized for SFA to extract underlying SFs subject to the constraints in Equations 2.3-2.5. The algorithm calculates SFs, which then are utilized to generate linear features from the solution of the generalized eigenvalue problem in Equation 2.7.

The proposed PSFA algorithm is proposed in Equations 2.8 and 2.9, with mathematical details available in Chapter 3.

Consider the following slow feature model:

$$\begin{bmatrix} \mathbf{s}_n(t) \\ \mathbf{s}_s(t) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_p & 0 \\ 0 & \mathbf{F}_s \end{bmatrix} \begin{bmatrix} \mathbf{s}_n(t-1) \\ \mathbf{s}_s(t-1) \end{bmatrix} + \begin{bmatrix} \mathbf{e}_n(t) \\ \mathbf{e}_s(t) \end{bmatrix} \tag{2.8}$$

where $\mathbf{I}_p = \text{diag}\{1, 1, \ldots, 1\}$, $\mathbf{F}_s = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_q\}$, $\mathbf{e}_n(t) \sim \mathcal{N}(0, c_n\mathbf{I}_m)$ with c_n being a constant that is unique and updated for each $\mathbf{s}_n(t)$, $\mathbf{e}_s(t) \sim \mathcal{N}(0, \mathbf{I}_q - \Lambda_s^2)$, $\mathbf{s}_n(t)$ and $\mathbf{s}_s(t)$ being non-stationary and stationary SFs, respectively, and $\mathbf{I}_m$ and $\mathbf{I}_q$ being the identity matrices of size m and q, respectively. Each $\mathbf{s}_n$ and $\mathbf{s}_s$ are first-order auto regressive (AR(1)) processes. Further, consider the relationship between slow features and data:

$$\begin{bmatrix} \mathbf{x}_n(t) \\ \mathbf{x}_s(t) \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{nn} & \mathbf{H}_{ns} \\ 0 & \mathbf{H}_{ss} \end{bmatrix} \begin{bmatrix} \mathbf{s}_n(t) \\ \mathbf{s}_s(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{\varepsilon}_n(t) \\ \boldsymbol{\varepsilon}_s(t) \end{bmatrix} \tag{2.9}$$

where $\boldsymbol{\varepsilon}_n(t) \sim \mathcal{N}(0, \boldsymbol{\Sigma}_n)$ and $\boldsymbol{\varepsilon}_s(t) \sim \mathcal{N}(0, \boldsymbol{\Sigma}_s)$are measurement noises of non-stationary and stationary process variables, with their individual diagonal covariance matrices $\boldsymbol{\Sigma}_n$ and $\boldsymbol{\Sigma}_s$. $\mathbf{H}_{nn}$ and $\mathbf{H}_{ns}$, and $\mathbf{H}_{ss}$ are block-wise emission matrices from nonstationary and stationary PSFs to nonstationary and stationary process variables. The model describes non-stationary process variations as it maps non-stationary SFs to each non-stationary process input $\mathbf{x}_n(t)$ by assuming eigenvalues in Λ_n have a value of one. As non-stationary SFs have no physical influence on stationary inputs $\mathbf{x}_s(t)$, considering existence of unit eigenvalues in Λ_n, the $\mathbf{H}_{sn}$ emission block matrix in Equation 2.9 must be zero. Stationary SFs, however, affect non-stationary variables.

Regardless of non-stationary process data and non-stationary SFs, it is possible to obtain the classical PSFA model. However, classical PSFA is only useful for systems with stationary data and handling missing data. As discussed, an issue with a standard PSFA model for the combined monitoring approach is that it is unable to separate between non-stationary and stationary components. Without the separation of non-stationary and stationary features into a probabilistic model, as derived in Equations 2.8 and 2.9, valuable coupled information is unable to be separated for machine learning algorithm abilities. The classical model for PSFA is derived in Equation 2.10

$$\begin{aligned} \mathbb{S}_j(t) &= \lambda_j \mathbf{s}_j(t-1) + \mathbf{e}_j(t) \\ \mathbb{X}(t) &= \mathbf{H}\mathbf{s}(t) + \mathbf{e}_\mathbf{x}(t) \end{aligned} \tag{2.10}$$

where $\mathbf{e}_j(t) \sim \mathcal{N}(0, 1 - \lambda_j^2)$ and $1 \leq j \leq q$. Additionally, $\mathbf{e}_\mathbf{x} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} = \text{diag}\{\sigma_1^2, \cdots, \sigma_m^2\}$.

Chapter 3

A Probabilistic Framework for Monitoring Non-Stationary Dynamic Industrial Processes

MSPM methods provide sensitive indicators of current process conditions by harnessing the value of massive process data. Large-scale industrial processes are subject to wide-range time-varying operating conditions such that some process variables inevitably exhibit non-stationary dynamics behavior. Non-stationary dynamic behaviors pose significant challenges for the design of MSPM schemes. In this work, a novel non-stationary probabilistic slow feature analysis (NS-PSFA) algorithm is developed to describe both non-stationary and stationary variations that underlie multivariate process measurements during routine operations. Concerning efficient parameter estimation of the probabilistic model from data, an expectation-maximization algorithm is developed and presented. Through modeling non-stationarity and stationarity as a random walk and stable auto-regressive processes, respectively, monitoring statistics are constructed to detect abnormality in non-stationary dynamics, stationary dynamics, and stationary routine conditions. This formulation provides a probabilistic monitoring framework for dynamic industrial processes, which can be beneficial for reducing false alarms and providing meaningful operational information for industrial practitioners. The efficacy and advantages of the proposed monitoring framework are illustrated and validated via two case studies.

3.1 Preliminaries

3.1.1 PSFA model

Traditional latent variable models (LVMs) such as PCA and ICA yield low-dimensional features that are independently and identically distributed, thereby falling short of addressing temporal dynamics. In this respect, SFA and PSFA appear to be advantageous in modeling and monitoring process dynamics. SFA is an unsupervised machine learning algorithm that maps input and output data to understand underlying correlations and connections between process variables [10] and creates slowly-varying latent variables. As a probabilistic counterpart of SFA, PSFA can also be desirably interpreted as an LVM that allows for the following state-space representation [43, 25]:

$$\begin{cases} \mathbf{s}(t) = \mathbf{F}\mathbf{s}(t-1) + \mathbf{e}(t) \\ \mathbf{x}(t) = \mathbf{H}\mathbf{s}(t) + \boldsymbol{\varepsilon}(t) \end{cases} \tag{3.1}$$

where $\mathbf{x}(t) \in \mathbb{R}^m$ denotes the vector of process variables that are measurable, $\mathbf{s}(t) \in \mathbb{R}^q$ denotes the vector of unseen "probabilistic slow features" (PSFs) that explain primary variations within $\mathbf{x}(t)$, and $\mathbf{e}(t) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}),\ \boldsymbol{\varepsilon}(t) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ are Gaussian distributed process noise and measurement noise, respectively. Different from common linear discrete systems (LDSs), PSFA has specific structural properties owing to the following parameterizations of $\mathbf{F}$ and $\boldsymbol{\Gamma}$:

$$\begin{aligned} &\mathbf{F} = \operatorname{diag}\{\lambda_1, \ldots, \lambda_q\},\ 0 < \lambda_q < \cdots < \lambda_1 < 1, \\ &\boldsymbol{\Gamma} = \operatorname{diag}\{1 - \lambda_1^2, \ldots, 1 - \lambda_q^2\}. \end{aligned} \tag{3.2}$$

It obviously follows that $\mathbf{F}^2 + \boldsymbol{\Gamma} = \mathbf{I}_m$. Meanwhile, the noise covariance matrix $\boldsymbol{\Sigma} = \operatorname{diag}\{\sigma_1^2, \ldots, \sigma_m^2\}$ is also diagonal. As a consequence, all PSFs $\{s_j(t)\}$ are independent AR(1) processes, each of which is driven by an independent noise $e_j(t)$. Furthermore, it can be verified that each PSF is stationary with zero mean and unit variance:

$$\mathbb{E}\{s_j(t)\} = 0,\ \operatorname{Var}\{s_j(t)\} = 1,\ 1 \le j \le q. \tag{3.3}$$

The slowly varying nature of $s_j(t)$ arises from its Markov property, and the transition parameter λ_j governs the level of slowness of $s_j(t)$, which is defined as [25]:

$$\Delta(s_j) := \mathbb{E}\left\{[s_j(t) - s_j(t-1)]^2\right\} = 2(1 - \lambda_j). \tag{3.4}$$

Intuitively, a large λ_j implies a strong correlation between $s_j(t)$ and $s_j(t-1)$, showing that $s_j(t)$ tends to be slow varying with a small $\Delta(\cdot)$ value, and vice-versa. PSFs $\mathbf{s}(t)$ can be conceived as modelling "inherent variations" within a large-scale process, based on which variations of process variables $\mathbf{x}(t)$ are induced by a linear mapping plus a measurement noise term $\mathbf{e}_x(t)$. Model parameters of PSFA include transition parameters $\{\lambda_j,\ 1 \leq j \leq q\}$, emission matrix $\mathbf{H} \in \mathbb{R}^{m \times q}$, and measurement noise variances $\{\sigma_i^2,\ 1 \leq i \leq m\}$. Notice that different from classic LVMs where $q \leq m$ must hold, PSFA makes no requirement on the relationship between m and q and over-complete features with $m < q$ are also admissible.

3.1.2 Random Walk model

In order to model non-stationarity, a natural choice is to enforce some transition parameters to be exactly one, which corresponds to the random walk model [44].

From $\mathbb{X}$ let $X_1, X_2, \cdots X_n$ be independent and identically distributed (i.i.d.) with $\mathbb{E}\{X_i\} \leq \infty$. Let $S_0 = 0$. The random walk is a stochastic sequence defined by

$$S_n = \sum_{1}^{n} X_i \tag{3.5}$$

where, $n \geq 1$. Therefore, the process $S_n, n \geq 0$ is called a random walk process [44]. Random walk models have been utilized to model various phenomena, including, general gambling probabilities [44]. Motivated by random walks, a novel NS-PSFA model is developed as an extension of PSFA to handle both stationary and non-stationary process variables.

3.1.3 Non-stationary PSFA

Note that the stationarity of PSFA arises from the constraint on each transition parameter, i.e. $\lambda_j \in (0, 1)$. In the novel NS-PSFA model all $(p+q)$-dimensional features are grouped into *stationary PSFs* $\mathbf{s}_n(t) \in \mathbb{R}^p$ and *non-stationary PSFs* $\mathbf{s}_s(t) \in \mathbb{R}^q$:

$$\begin{bmatrix} \mathbf{s}_n(t) \\ \mathbf{s}_s(t) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_s \end{bmatrix} \begin{bmatrix} \mathbf{s}_n(t-1) \\ \mathbf{s}_s(t-1) \end{bmatrix} + \begin{bmatrix} \mathbf{e}_n(t) \\ \mathbf{e}_s(t) \end{bmatrix}. \tag{3.6}$$

Here, the transition matrix associated with non-stationary PSFs becomes the identity matrix, while the driving noises are independent $\mathbf{e}_n(t) \sim \mathcal{N}(0, \mathbf{\Gamma}_n)$ with covariance

matrix $\mathbf{\Gamma}_n = \text{diag}\{c_1, \cdots, c_p\}$. It is easy to verify that the second-order moment of $\mathbf{s}_n(t)$ varies over time and hence exhibits non-stationarity. The parameterizations of generic PSFA are inherited by stationary PSFs $\mathbf{s}_s(t)$:

$$\begin{aligned} &\mathbf{F}_s = \text{diag}\{\lambda_1, \ldots, \lambda_q\},\ 0 < \lambda_q < \cdots < \lambda_1 < 1, \\ &\mathbf{e}_s(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma}_s),\ \mathbf{\Gamma}_s = \text{diag}\{1 - \lambda_1^2, \ldots, 1 - \lambda_q^2\}. \end{aligned} \tag{3.7}$$

In similar spirit to (3.6), all observed process variables can also be split into non-stationary and stationary variables based on the ADF test [45], which are denoted by $\mathbf{x}_n(t)$ and $\mathbf{x}_s(t)$, respectively. In common industrial practice, controlled variables typically show stationarity due to the closed-loop control, while some disturbance variables and manipulated variables may be prone to external non-stationarity variations. They can be generated by PSFs based on the following relation:

$$\begin{bmatrix} \mathbf{x}_n(t) \\ \mathbf{x}_s(t) \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{nn} & \mathbf{H}_{ns} \\ \mathbf{0} & \mathbf{H}_{ss} \end{bmatrix} \begin{bmatrix} \mathbf{s}_n(t) \\ \mathbf{s}_s(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{\varepsilon}_n(t) \\ \boldsymbol{\varepsilon}_s(t) \end{bmatrix} \tag{3.8}$$

where $\boldsymbol{\varepsilon}_n(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_n)$ and $\boldsymbol{\varepsilon}_s(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_s)$ are measurement noises of non-stationary and stationary process variables, with their individual diagonal covariance matrices $\mathbf{\Sigma}_n$ and $\mathbf{\Sigma}_s$. $\mathbf{H}_{nn}$ and $\mathbf{H}_{ns}$, and $\mathbf{H}_{ss}$ are block-wise emission matrices from nonstationary and stationary PSFs to nonstationary and stationary process variables. Notice that it is practically sound for $\mathbf{s}_n(t)$ to have no influence on $\mathbf{x}_s(t)$ with a zero emission matrix assigned because otherwise $\mathbf{x}_s(t)$ will become non-stationary, which contradicts the assumption. Conversely, it is possible that non-stationary process variables consist of both non-stationary and stationary components.

In summary, the entire NS-PSFA model in Equations (3.6) and (3.8) are following the Gaussian assumption, which leverages a series of independent random walks to describe non-stationarity and a series of stable AR(1) processes to describe stationarity. In industrial production systems, non-stationary variables are typically subject to non-stationary disturbances, whose variations are caused by both non-stationary and stationary PSFs. Stationary variables describe controlled variables and steady disturbances, which are reliant upon stationary PSFs only. In the proposed NS-PSFA algorithm, this leads to much better modeling than PSFA that disregards non-stationarity. As to be demonstrated in the sequel, such a description is particularly useful for monitoring design in the presence of non-stationary process variables.

3.2 Parameter Estimation of Non-stationary PSFA with EM Algorithm

For the NS-PSFA model with dimensions p, q, m given, all parameters that need to be determined from data are denoted as $\theta = \{\mathbf{F}_s, \mathbf{\Gamma}_n, \mathbf{H}_{nn}, \mathbf{H}_{ns}, \mathbf{H}_{ss}, \mathbf{\Sigma}_n, \mathbf{\Sigma}_s\}$. Given a time series of process variables $\mathbb{X} = \{\mathbf{x}(t), 1 \leq t \leq T\}$, a suitable criterion to optimize model parameters is to maximize its *likelihood*, i.e. $P(\mathbb{X}|\theta)$, which yields the maximum likelihood estimation. However, direct optimization of $P(\mathbb{X}|\theta)$ is difficult with respect to θ without closed-form solutions. The EM algorithm has been recognized as an effective tool towards effective parameter estimation of probabilistic models [46]. It iterates between the expectation step (E-step) and the maximization step (M-step) and is able to continually improve the likelihood value until convergence [47]. In fact, the EM algorithm has already been successfully applied to data-driven modeling of LDS, see e.g. [48, 49, 43, 50]. Although the proposed NS-PSFA belongs to the general class of LDS, its special parametric structure prohibits the direct use of off-the-shelf EM algorithms, and hence a tailored procedure is provided below.

Denote by $\mathbb{S} = \{\mathbf{s}(t), 1 \leq t \leq T\}$ the set of PSFs related to all available process measurements. Due to the Markov chain structure of PSFA, the complete data log-likelihood of NS-PSFA can be written as:

$$\begin{aligned}
&\log P(\mathbb{X}, \mathbb{S}|\theta) \\
=& -\frac{T-1}{2}\sum_{j=1}^{m}\log c_j - \frac{1}{2}\sum_{t=2}^{T}\|\mathbf{s}_n(t) - \mathbf{s}_n(t-1)\|^2_{\mathbf{\Gamma}_n^{-1}} \\
&-\frac{T-1}{2}\sum_{j=1}^{q}\log(1-\lambda_j^2) - \frac{1}{2}\sum_{t=2}^{T}\|\mathbf{s}_s(t) - \mathbf{s}_s(t-1)\|^2_{\mathbf{\Gamma}_s^{-1}} \\
&-\frac{T}{2}\log\det\mathbf{\Sigma}_n - \frac{1}{2}\sum_{t=1}^{T}\|\mathbf{x}_n(t) - \mathbf{H}_n\mathbf{s}(t)\|^2_{\mathbf{\Sigma}_n^{-1}} \\
&-\frac{T}{2}\log\det\mathbf{\Sigma}_s - \frac{1}{2}\sum_{t=1}^{T}\|\mathbf{x}_s(t) - \mathbf{H}_{ss}\mathbf{s}_s(t)\|^2_{\mathbf{\Sigma}_s^{-1}} \\
&-\frac{1}{2}\mathbf{s}_n(1)^\top\mathbf{s}_n(1) - \frac{1}{2}\mathbf{s}_s(1)^\top\mathbf{s}_s(1)
\end{aligned} \tag{3.9}$$

where $\mathbf{H}_n = [\mathbf{H}_{nn}\ \mathbf{H}_{ns}]$ is defined for brevity. $\mathbf{s}(1)$ are initial states for both non-stationary and stationary PSFs, which are assumed to follow a standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_{p+q})$. With the complete data log-likelihood being formulated, the

Q-function can be derived by taking conditional expectation of (3.9):

$$Q(\theta, \theta^{\text{old}}) = \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\{\log P(\mathbb{X}, \mathbb{S}|\theta)\} \tag{3.10}$$

where θ^{old} denotes parameters in the previous iteration while executing the EM algorithm.

3.2.1 Maximization step

In the M-step, the Q-function is maximized w.r.t. θ to obtain updated parameters, i.e. $\theta^{\text{new}} = \arg\max_{\theta} Q(\theta, \theta^{\text{old}})$. Next the maximization over each parameter is tackled individually. First, setting the derivative of Q-function w.r.t. c_j to be zero leads to the following updating formula

$$c_{n,j}^{\text{new}} = \frac{1}{T-1} \sum_{t=2}^{T} \mathbb{E}_{\mathbb{X},\theta^{\text{old}}} \left\{ [s_{n,j}(t) - s_{n,j}(t-1)]^2 \right\}. \tag{3.11}$$

As for the optimization of λ_j, the formula in generic PSFA can be directly applied. Specifically, λ_j^{new} can be calculated as the root of the following equation in the range [0,1) [25]:

$$\lambda_j^3 + a_j \lambda_j^2 + b_j \lambda_j + a_j = 0, \tag{3.12}$$

where

$$\begin{aligned} a_j &= -\frac{1}{T-1} \sum_{t=2}^{T} \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\{s_{s,j}(t) s_{s,j}(t-1)\}, \\ b_j &= \frac{1}{T-1} \sum_{t=2}^{T} \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\{s_{s,j}^2(t) + s_{s,j}^2(t-1)\} - 1. \end{aligned} \tag{3.13}$$

Similarly, other parameters may be updated by taking partial derivatives of Q-

function to zero. Desirably these updating policies all admit closed-form expressions:

$$
\begin{aligned}
\frac{\partial Q(\mathbb{X}, \mathbb{S}|\theta)}{\partial \mathbf{H}_n} &= 0 \\
\frac{\partial}{\partial \mathbf{H}_n} \mathbb{E}\left(\sum_{t\ \epsilon \top} \left[(\mathbf{x}_n(t) - \begin{bmatrix} \mathbf{H}_n & b \end{bmatrix} \begin{bmatrix} \mathbf{s}(t) \\ 1 \end{bmatrix})^\top \right.\right. & \\
\left.\left. \boldsymbol{\Sigma}_n^{-1} (\mathbf{x}_n(t) - \begin{bmatrix} \mathbf{H}_n & b \end{bmatrix} \begin{bmatrix} \mathbf{s}(t) \\ 1 \end{bmatrix}) \right] \right) &= 0 \\
\sum_{t\ \epsilon \top} \mathbb{E}\left[(\mathbf{x}_n(t) - \begin{bmatrix} \mathbf{H}_n & b \end{bmatrix} \begin{bmatrix} \mathbf{s}(t) \\ 1 \end{bmatrix})^\top \boldsymbol{\Sigma}_n^{-1} (\mathbf{x}_n(t) - \begin{bmatrix} \mathbf{H}_n & b \end{bmatrix} \begin{bmatrix} \mathbf{s}(t) \\ 1 \end{bmatrix}) \right] &= 0 \\
\sum_{t\ \epsilon \top} \mathbb{E}\left[(\boldsymbol{\Sigma}_n + \boldsymbol{\Sigma}_n^{-1}) (\mathbf{x}_n(t) - \begin{bmatrix} \mathbf{H}_n & b \end{bmatrix} \begin{bmatrix} \mathbf{s}(t) \\ 1 \end{bmatrix}) (-\mathbf{s}^\top(t) \begin{bmatrix} \mathbf{H}_n & b \end{bmatrix}) \right] &= 0 \\
\sum_{t=1}^{T} \mathbb{E}\left[(\boldsymbol{\Sigma}_n + \boldsymbol{\Sigma}_n^{-1}) (\mathbf{x}_n(t) - \begin{bmatrix} \mathbf{H}_1 & b \end{bmatrix} \begin{bmatrix} \mathbf{s}(t) \\ 1 \end{bmatrix}) (-\begin{bmatrix} \mathbf{s}(t) \\ 1 \end{bmatrix}^\top) \right] &= 0 \qquad (3.14) \\
\sum_{t=1}^{T} \mathbb{E}\left[(-\mathbf{x}_n(t) \begin{bmatrix} \mathbf{s}(t) & 1 \end{bmatrix} + \begin{bmatrix} \mathbf{H}_n & b \end{bmatrix} \begin{bmatrix} \mathbf{s}(t) \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{s}(t) \\ 1 \end{bmatrix}^\top \right] &= 0 \\
\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{x}_n(t) \begin{bmatrix} \mathbf{s}(t) & 1 \end{bmatrix} \right] = \sum_{t=1}^{T} \mathbb{E}\left[\begin{bmatrix} \mathbf{H}_n & b \end{bmatrix} \begin{bmatrix} \mathbf{s}(t) \\ 1 \end{bmatrix} \mathbf{s}^\top(t) \right] & \\
\begin{bmatrix} \mathbf{H}_n & b \end{bmatrix}^{\text{new}} = \left(\sum_{t=1}^{T} \mathbb{E}\left\{ \mathbf{x}_n(t) \begin{bmatrix} \mathbf{s}(t) & 1 \end{bmatrix} \right\} \right) & \\
\left(\sum_{t=1}^{T} \begin{bmatrix} \mathbb{E}\{\mathbf{s}(t)\mathbf{s}^\top(t)\} & \mathbb{E}\{\mathbf{s}(t)\} \\ \mathbb{E}\{\mathbf{s}(t)\} & 1 \end{bmatrix} \right)^{-1} &
\end{aligned}
$$

where b is a bias term for the calculation. With the simplified summation of $\mathbf{H}_n^{\text{new}}$ found in Equation 3.18. To generate the noise variance of $\mathbf{H}_n$ for Equation 3.9, the

partial derivative of Equation 3.9 w.r.t. $\sigma_{n,j}^2$ is taken in Equation 3.15 [51]

$$
\begin{aligned}
\frac{\partial Q(\mathbb{X},\mathbb{S}|\theta)}{\partial \sum_n} &= 0 \\
-\frac{\mathrm{T}}{2}\cdot\frac{\partial \log|\sum_n|}{\partial \sum_n} + \frac{\partial}{\partial \sum_n}\mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\Bigg(-\frac{1}{2}\sum_{t\ \epsilon \mathrm{T}}(\mathbf{x}_n(t) - [\ \mathbf{H}_n \quad \mathrm{b}\ ]\begin{bmatrix}\mathbf{s}(t)\\ 1\end{bmatrix}\Bigg)^{\top}\boldsymbol{\Sigma}_n^{-1} \\
\Bigg(\mathbf{x}_n(t) - [\ \mathbf{H}_1 \quad \mathrm{b}\ ]\begin{bmatrix}\mathbf{s}(t)\\ 1\end{bmatrix}\Bigg) &= 0 \\
-\frac{\mathrm{T}}{2}\cdot\frac{\partial \log|\sum_n|}{\partial \sum_n} + \frac{\partial}{\partial \sum_n}\mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\Bigg(-\frac{1}{2}\sum_{t\ \epsilon \mathrm{T}}\Bigg[(\mathbf{x}_n(t) - [\ \mathbf{H}_1 \quad \mathrm{b}\ ]\begin{bmatrix}\mathbf{s}(t)\\ 1\end{bmatrix})^{\top} \\
\boldsymbol{\Sigma}_n^{-1}(\mathbf{x}_n(t) - [\ \mathbf{H}_1 \quad \mathrm{b}\ ]\begin{bmatrix}\mathbf{s}(t)\\ 1\end{bmatrix})\Bigg]\Bigg) &= 0 \\
-\frac{\mathrm{T}}{2}\cdot\frac{\partial \log|\sum_n|}{\partial \sum_n} - \frac{1}{2}\frac{\partial}{\partial \sum_n}\mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\Bigg(\sum_{t=1}^{T}\Bigg[(\mathbf{x}_n^{\top}(t)\boldsymbol{\Sigma}_n^{-1}\mathbf{x}_n(t) - \mathbf{x}_n^{\top}(t)\boldsymbol{\Sigma}_n^{-1}[\ \mathbf{H}_1 \quad \mathrm{b}\ ] \\
\begin{bmatrix}\mathbf{s}(t)\ 1\end{bmatrix} - \begin{bmatrix}\mathbf{s}(t)\\ 1\end{bmatrix}^{\top}[\ \mathbf{H}_1 \quad \mathrm{b}\ ]^{T}\boldsymbol{\Sigma}_n^{-1}\mathbf{x}_n(t) \\
+\begin{bmatrix}\mathbf{s}(t)\\ 1\end{bmatrix}^{\top}[\ \mathbf{H}_n \quad \mathrm{b}\ ]^{\top}\boldsymbol{\Sigma}_n^{-1}[\ \mathbf{H}_n \quad \mathrm{b}\ ]\begin{bmatrix}\mathbf{s}(t)\\ 1\end{bmatrix}\Bigg]\Bigg) &= 0 \\
-\frac{\mathrm{T}}{2}\boldsymbol{\Sigma}_n^{-\top} - \frac{1}{2}\boldsymbol{\Sigma}_n^{-\top}\sum_{t=1}^{T}\mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\Bigg[\mathbf{x}_n(t)\mathbf{x}_n^{\top}(t) - \mathbf{x}_n(t)\begin{bmatrix}\mathbf{s}(t)\\ 1\end{bmatrix}^{\top}[\ \mathbf{H}_n \quad \mathrm{b}\ ]^{\top} \\
-[\ \mathbf{H}_n \quad \mathrm{b}\ ]\mathbf{s}(t)\mathbf{x}_n^{\top}(t) + [\ \mathbf{H}_n \quad \mathrm{b}\ ]\begin{bmatrix}\mathbf{s}(t)\\ 1\end{bmatrix}\begin{bmatrix}\mathbf{s}(t)\\ 1\end{bmatrix}^{\top}[\ \mathbf{H}_1 \quad \mathrm{b}\ ]^{\top}\Bigg]\boldsymbol{\Sigma}_n^{-\top} &= 0 \\
\boldsymbol{\Sigma}_{n,j}^{\text{new}} = \frac{1}{\mathrm{T}}\sum_{t=1}^{T}\Bigg[\mathbb{E}_{\mathbb{X},\theta^{\text{old}}}(x_{n,j}(t)x_{n,j}^{\top}(t)) - \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}(\mathbf{x}_{n,j}(t)\begin{bmatrix}\mathbf{s}(t)\\ 1\end{bmatrix}^{\top}) \\
([\ \mathbf{H}_n \quad \mathrm{b}\ ]^{\top})^{\text{new}} - ([\ \mathbf{H}_n \quad \mathrm{b}\ ])^{\text{new}}\mathbb{E}_{\mathbb{X},\theta^{\text{old}}},\theta^{\text{old}}(\begin{bmatrix}\mathbf{s}(t)\\ 1\end{bmatrix}\mathbf{x}_{n,i}^{\top}(t)) + ([\ \mathbf{H}_n \quad \mathrm{b}\ ])^{\text{new}} \\
E_{\mathbb{X},\theta^{\text{old}}}(\begin{bmatrix}\mathbb{E}\{\mathbf{s}(t)\mathbf{s}^{\top}(t)\} & \mathbb{E}\{\mathbf{s}(t)\}\\ \mathbb{E}\{\mathbf{s}(t)\} & 1\end{bmatrix})([\ \mathbf{H}_n \quad \mathrm{b}\ ]^{\top})^{\text{new}}\Bigg]
\end{aligned}
\tag{3.15}
$$

With the simplified summation $\left(\sigma_{n,j}^2\right)^{\text{new}}$ found in Equation 3.18. Optimization of $\mathbf{H}_{ss}$ requires taking the partial derivative of Equation 3.9 w.r.t. $\mathbf{H}_{ss}$ which occurs in

Equation 3.16 [51]

$$
\begin{aligned}
\frac{\partial Q(\mathbb{X},\mathbb{S}|\theta)}{\partial \mathbf{H}_{ss}} &= 0 \\
\frac{\partial}{\partial \mathbf{H}_{ss}} \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left(\sum_{t\ \epsilon T} \left[(\mathbf{x}_s(t) - \mathbf{H}_{ss}\mathbf{s}_s(t))^\top \mathbf{\Sigma}_s^{-1} (\mathbf{x}_s(t) - \mathbf{H}_{ss}\mathbf{s}_s(t)) \right] \right) &= 0 \\
\sum_{t=1} \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left[(\mathbf{\Sigma}_s + \mathbf{\Sigma}_s^{-1})(\mathbf{x}_s(t) - \mathbf{H}_{ss}\mathbf{s}_s(t))(-\mathbf{s}_s^\top(t)) \right] &= 0 \\
\sum_{t=1}^{T} \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left[\mathbf{x}_s(t)\mathbf{s}_s^\top(t) \right] &= \sum_{t=1}^{T} \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left[\mathbf{H}_{ss}\mathbf{s}_s(t)\mathbf{s}_s^\top(t) \right] \\
\mathbf{H}_{ss}^{\text{new}} &= \left(\sum_{t=1}^{T} \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left[\mathbf{x}_s(t)\mathbf{s}_s^\top(t) \right] \right) \cdot \left(\sum_{t=1}^{T} \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left[\mathbf{s}_s(t)\mathbf{s}_s^\top(t) \right] \right)^{-1}
\end{aligned}
\tag{3.16}
$$

To generate the noise variance of $\mathbf{H}_{ss}$ for Equation 3.16, the partial derivative of Equation 3.9 w.r.t. $\mathbf{\Sigma}_{s,j}$ is taken in Equation 3.17 [51]

$$
\begin{aligned}
\frac{\partial Q(\mathbb{X},\mathbb{S}|\theta)}{\partial \sum_s} &= 0 \\
-\frac{\mathrm{T}}{2} \cdot \frac{\partial \log|\sum_s|}{\partial \sum_s} + & \\
\frac{\partial}{\partial \sum_s} \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left(-\frac{1}{2} \sum_{t\ \epsilon \mathrm{T}} \left[(\mathbf{x}_s(t) - \mathbf{H}_{ss}\mathbf{s}_s(t))^\top \mathbf{\Sigma}_s^{-1} (\mathbf{x}_s(t) - \mathbf{H}_{ss}\mathbf{s}_s(t)) \right] \right) &= 0 \\
-\frac{\mathrm{T}}{2} \cdot \frac{\partial \log|\sum_s|}{\partial \sum_s} + & \\
\frac{\partial}{\partial \sum_s} \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left(-\frac{1}{2} \sum_{t\ \epsilon T} \left[(\mathbf{x}_s(t) - \mathbf{H}_{ss}\mathbf{s}_s(t))^\top \mathbf{\Sigma}_s^{-1} (\mathbf{x}_s(t) - \mathbf{H}_{ss}\mathbf{s}_s(t)) \right] \right) &= 0 \\
-\frac{\mathrm{T}}{2} \cdot \frac{\partial \log|\sum_s|}{\partial \sum_s} - \frac{1}{2} \frac{\partial}{\partial \sum_s} \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left(\sum_{t=1}^{T} \left[(\mathbf{x}_s^\top(t)\mathbf{\Sigma}_s^{-1}\mathbf{x}_s(t) - \mathbf{x}_s^\top(t)\mathbf{\Sigma}_s^{-1}\mathbf{H}_{ss}\mathbf{s}_s(t) \right.\right. & \\
\left.\left. -\mathbf{s}_s^\top(t)\mathbf{H}_{ss}^\top\mathbf{\Sigma}_s^{-1}\mathbf{x}_s(t) + \mathbf{s}_s(t)\mathbf{H}_{ss}^\top\mathbf{\Sigma}_s^{-1}\mathbf{H}_{ss}\mathbf{s}_s(t)) \right] \right) &= 0 \\
-\frac{\mathrm{T}}{2}\mathbf{\Sigma}_s^{-\top} - \frac{1}{2}\mathbf{\Sigma}_s^{-\top} \sum_{t=1}^{T} \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left[\mathbf{x}_s(t)\mathbf{x}_s^\top(t) - (t)\mathbf{s}_s^\top(t)\mathbf{H}_{ss}^\top \right. & \\
\left. -\mathbf{H}_{ss}\mathbf{s}_s(t)\mathbf{x}_s^\top(t) + \mathbf{H}_{ss}\mathbf{s}_s(t)\mathbf{s}_s^\top(t)\mathbf{H}_{ss}^\top \right] \mathbf{\Sigma}_s^{-\top} &= 0 \\
\mathbf{\Sigma}_s^{\text{new}} = \frac{1}{T} \sum_{t=1}^{T} \left[\mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left(\mathbf{x}_s(t)\mathbf{x}_s^\top(t) \right) - 2(\mathbf{H}_{ss}^\top)^{\text{new}} \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left(\mathbf{s}_s(t)\mathbf{x}_s(t) \right) \right. & \\
\left. +(\mathbf{H}_{ss}^\top)^{\text{new}} \mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left(\mathbf{s}_s(t)\mathbf{s}_s^\top(t) \right) (\mathbf{H}_{ss})^{\text{new}} \right] &
\end{aligned}
\tag{3.17}
$$

With the simplified summation $\left(\sigma_{s,j}^2\right)^{\text{new}}$ found in Equation 3.18.

Finally, the closed form expressions of $\mathbf{H}_n^{\text{new}}$, $\mathbf{H}_{ss}^{\text{new}}$, $\left(\sigma_{n,j}^2\right)^{\text{new}}$, and $\left(\sigma_{s,j}^2\right)^{\text{new}}$ are presented in Equation 3.18

$$
\begin{aligned}
&\mathbf{H}_n^{\text{new}} \\
=&\left(\sum_{t=1}^{T}\mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left[\mathbf{x}_n(t)\mathbf{s}(t)^\top\right]\right)\left(\sum_{t=1}^{T}\mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left[\mathbf{s}(t)\mathbf{s}(t)^\top\right]\right)^{-1}, \\
&\mathbf{H}_{ss}^{\text{new}} \\
=&\left(\sum_{t=1}^{T}\mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left[\mathbf{x}_s(t)\mathbf{s}_s(t)^\top\right]\right)\left(\sum_{t=1}^{T}\mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left[\mathbf{s}_s(t)\mathbf{s}_s(t)^\top\right]\right)^{-1}, \\
&\left(\sigma_{n,j}^2\right)^{\text{new}} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left\{[x_{n,j}(t) - (\mathbf{h}_{n,\cdot j}^\top)^{\text{new}}\mathbf{s}(t)]^2\right\}, \\
&\left(\sigma_{s,j}^2\right)^{\text{new}} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\left\{[x_{s,j}(t) - (\mathbf{h}_{ss,\cdot j}^\top)^{\text{new}}\mathbf{s}_s(t)]^2\right\},
\end{aligned}
\tag{3.18}
$$

where $\mathbf{h}_{n,\cdot j}^\top$ is the j-th row of $\mathbf{H}_n$, and $\mathbf{h}_{ss,\cdot j}^\top$ is the j-th row of $\mathbf{H}_{ss}$.

3.2.2 Expectation step

The E-step provides state estimations by evaluating the posterior distribution $P(\mathbb{S}|\mathbb{X},\theta^{\text{old}})$, which lays the basis for the M-step. This can be formally translated to the Kalman filter and smoothing problems [52]. Statistics of $P(\mathbb{S}|\mathbb{X},\theta^{\text{old}})$ to be computed in the E-step are first- and second-order moments:

$$
\begin{cases}
\mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\{\mathbf{s}(t)\} = \hat{\boldsymbol{\mu}}_t, \\
\mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\{\mathbf{s}(t)\mathbf{s}(t-1)^\top\} = \mathbf{J}_{t-1}\hat{\mathbf{V}}_t + \hat{\boldsymbol{\mu}}_t\hat{\boldsymbol{\mu}}_{t-1}^\top \\
\mathbb{E}_{\mathbb{X},\theta^{\text{old}}}\{\mathbf{s}(t)\mathbf{s}(t)^\top\} = \hat{\mathbf{V}}_t + \hat{\boldsymbol{\mu}}_t\hat{\boldsymbol{\mu}}_t^\top
\end{cases}
\tag{3.19}
$$

To this end, Kalman filters are first employed to calculate the posterior distribution $P(\mathbf{s}(t)|\mathbf{x}(1),\ldots,\mathbf{x}(t),\theta^{\text{old}}) \sim \mathcal{N}(\boldsymbol{\mu}_t,\mathbf{P}_t)$ sequentially:

$$
\begin{cases}
\mathbf{P}_{t-1} = \mathbf{F}\mathbf{V}_{t-1}\mathbf{F}^\top + \boldsymbol{\Gamma}, \\
\boldsymbol{\mu}_t = \mathbf{F}\boldsymbol{\mu}_{t-1} + \mathbf{K}_t[\mathbf{x}(t) - \mathbf{H}\mathbf{F}\boldsymbol{\mu}_{t-1}], \\
\mathbf{V}_t = (\mathbf{I}_q - \mathbf{K}_t\mathbf{H})\mathbf{P}_{t-1}, \\
\mathbf{K}_t = \mathbf{P}_{t-1}\mathbf{H}^\top\left(\mathbf{H}\mathbf{P}_{t-1}\mathbf{H}^\top + \boldsymbol{\Sigma}\right)^{-1}
\end{cases}
\tag{3.20}
$$

Parameters in (3.19) are then derived from the Kalman smoother equations:

$$\begin{cases} \hat{\boldsymbol{\mu}}_t = \boldsymbol{\mu}_t + \mathbf{J}_t(\hat{\boldsymbol{\mu}}_{t+1} - \mathbf{F}\boldsymbol{\mu}_t), \\ \hat{\mathbf{V}}_t = \mathbf{V}_t + \mathbf{J}_t(\hat{\mathbf{V}}_{t+1} - \mathbf{P}_t)\mathbf{J}_t^\top, \\ \mathbf{J}_t = \mathbf{V}_t\mathbf{F}^\top\mathbf{P}_t^{-1} \end{cases} \tag{3.21}$$

with initializations of $\hat{\boldsymbol{\mu}}_T = \boldsymbol{\mu}_T, \hat{\mathbf{V}}_T = \mathbf{V}_T$.

3.3 A Probabilistic Monitoring Scheme

3.3.1 Monitoring statistics design

Upon deriving all model parameters θ of NS-PSFA via the EM algorithm, we can proceed with the online implementation of the monitoring scheme, which is based on estimating PSFs with online measurements and then defining monitoring statistics. Because NS-PSFA is essentially an LDS, the Kalman filtering technique can be applied conveniently for online estimation of PSFs [25]:

$$\mathbf{s}(t) = \mathbf{F}\mathbf{s}(t-1) + \mathbf{K}[\mathbf{x}(t) - \mathbf{H}\mathbf{F}\mathbf{s}(t-1)] \tag{3.22}$$

where $\mathbf{K} = \mathbf{P}\mathbf{H}^\top(\mathbf{H}\mathbf{P}\mathbf{H}^\top + \boldsymbol{\Sigma})^{-1}$ is the steady state gain matrix, with $\mathbf{P}$ obtained by solving the algebraic Riccati equation:

$$\mathbf{P} = \mathbf{F}\mathbf{P}\mathbf{F}^\top + \boldsymbol{\Gamma} - \mathbf{F}\mathbf{P}\mathbf{H}^\top(\boldsymbol{\Sigma} + \mathbf{H}\mathbf{P}\mathbf{H}^\top)^{-1}\mathbf{H}\mathbf{P}\mathbf{F}^\top. \tag{3.23}$$

Note that non-stationary and stationary PSFs can be estimated jointly. To develop the monitoring scheme, monitoring statistics are required to provide insight into process conditions. Thanks to the clear interpretation of PSFs, various monitoring statistics can be defined to characterize steady states and process dynamics, respectively. First, stationary PSFs $\mathbf{s}_s(t)$ characterizes stationary variations of routine operations, and devising statistics based on the steady-state distribution of $\mathbf{s}_s(t)$ will be helpful for detecting potential operating condition deviations. To this end, the Hotelling's T^2 statistic and the squared prediction error (SPE) are defined based upon $\mathbf{s}_s(t)$:

$$T_s^2 = \mathbf{s}_s^\top(t)\mathbf{s}_s(t), \tag{3.24}$$

$$\mathrm{SPE}_s = \|\mathbf{x}_s(t) - \mathbf{H}_{ss}\mathbf{F}_s\mathbf{s}_s(t-1)\|^2_{\mathbf{\Phi}_{ss}^{-1}}, \tag{3.25}$$

where $\mathbf{x}_s(t)-\mathbf{H}_{ss}\mathbf{F}_s\mathbf{s}_s(t-1)$ denotes the one step prediction error of stationary process variables. The static covariance matrix $\mathbf{\Phi}_{ss}$ can be obtained as follows:

$$\begin{bmatrix} \mathbf{\Phi}_{nn} & \mathbf{\Phi}_{ns} \\ \mathbf{\Phi}_{sn} & \mathbf{\Phi}_{ss} \end{bmatrix} = \mathbf{\Phi} = \mathbf{H}(\mathbf{F}\mathbf{V}\mathbf{F}^\top + \mathbf{\Gamma})\mathbf{H}^\top + \mathbf{\Sigma}, \tag{3.26}$$

$$\mathbf{V} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}. \tag{3.27}$$

Notice that non-stationary variations typically arises from external disturbances. Monitoring the steady-state of non-stationary variations in $\mathbf{s}_n(t)$ could induce false alarms and is thus unnecessary [17]. This is because in the random walk model the steady-state may show drifting behaviors and has a continuously increasing variance. Conversely, it is rational to monitor its temporal dynamics whose statistical properties do not vary over time. It has been recognized that temporal dynamics acts as an important indicator of control performance, since once controlled variables cannot be stabilized around their setpoints, controllers will maker unusual compensating behaviors [14]. In order to monitor process temporal dynamics, a popular approach is to resort to the distribution of the time-difference of features $\dot{\mathbf{s}}(t)$. To do so, S_s^2 and S_n^2 statistics are constructed to monitor stationary dynamics and non-stationary dynamics, which are based on stationary and non-stationary features, respectively:

$$S_s^2 = \dot{\mathbf{s}}_s(t)^\top \mathbf{\Omega}^{-1}\dot{\mathbf{s}}_s(t), \tag{3.28}$$

$$S_n^2 = \dot{\mathbf{s}}_n(t)^\top \mathbf{\Gamma}_n^{-1}\dot{\mathbf{s}}_n(t). \tag{3.29}$$

where $\mathbf{\Omega} = \mathrm{diag}\{2(1-\lambda_1), \cdots, 2(1-\lambda_q)\}$ is the covariance matrix of $\dot{\mathbf{s}}_s(t)$, as indicated by (3.4). The covariance matrix of $\dot{\mathbf{s}}_s(t)$ is given by $\mathbf{\Gamma}_n$ due to its random walk properties. From all developed statistics, control limits can be set as quantiles of χ^2-distributions because of the Gaussian assumption made upon $\mathbf{s}(t)$.

In summary, there are four monitoring statistics with different physical implications in the proposed monitoring scheme. In monitoring, for the stationary variations, the steady-state and temporal dynamics are monitored, whereas, for the non-stationary variations only monitors dynamics. In particular, the T_s^2 and SPE_s monitoring statistics are based on steady-states of stationary variations and are free from

non-stationary variations, which can be used to identify operating conditions deviations associated with stationary process variables under feedback control. Alternatively, the S_s^2 and S_n^2 statistics are responsible for detecting alterations in stationary and non-stationary temporal dynamics, respectively.

In practical use, the integration of monitoring statistics provides thorough information about process operation status, relieving the uncertainty of alarms, and assists practitioners to properly understand operating status. For example, if T_s^2 and SPE_s exceed control limits thereof, yet S_s^2 and S_n^2 remain plausible, the process is deemed to move to a new steady-state while being under control with sound dynamics. Hence, it is unnecessary to trigger alarms. Conversely, if S_s^2 or S_n^2 issues an alarm, non-stationary, or stationary dynamics are considered to be disrupted, thereby implying the occurrence of control performance changes and a possibly more difficult situation than the previous one. Practitioners can also analyze other situations according to interpretations of statistics together with process-specific knowledge and experience. Hence, the proposed scheme is in sharp contrast to existing LDS-based monitoring methods [50, 53], where monitoring statistics can only indicate some process changes with ambiguous information delivered.

3.3.2 Monitoring scheme design

Figure 3.1 summarizes the schematic of the proposed NS-PSFA-based monitoring framework, which consists of the modeling stage and the implementation stage. In the offline modeling stage, the NS-PSFA model is obtained by first performing the ADF test onto historical data, and then executing the EM algorithm to derive model parameters. During online implementations, the Kalman filter is adopted to obtain estimates of non-stationary and stationary PSFs. PSFs are utilized to generate four monitoring statistics that determine anomalies with different physical meanings.

3.4 Case Studies

3.4.1 Continuous Stirred Tank Reactor Simulation

In this section, a simulated continuous stirred tank reactor (CSTR) process with both non-stationary and stationary dynamics is utilized to showcase the monitoring

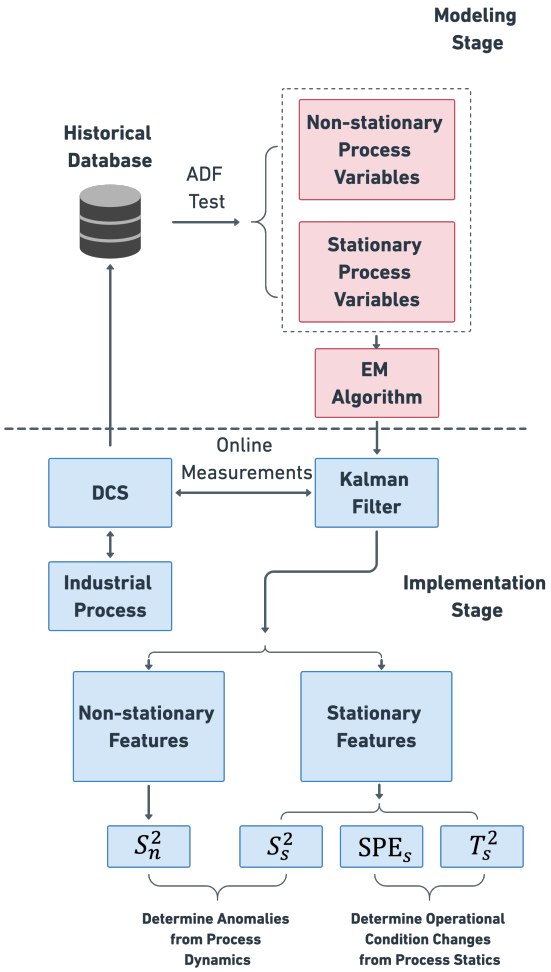


Figure 3.1: Schematic of the probabilistic monitoring framework

capabilities of the proposed method. The CSTR has a single feed stream that contains the reactant, a product stream, and a reactor cooling water stream. The underlying principle can be described by following differential equations [54]:

$$\begin{aligned}\frac{dC_{\mathrm{A}}}{dt} &= \frac{q}{V}(C_{\mathrm{Af}} - C_{\mathrm{A}}) - k_0 e^{-\frac{E}{RT}} C_{\mathrm{A}} + v_1 \\ \frac{dT}{dt} &= \frac{q}{V}(T_{\mathrm{f}} - T) - \frac{\Delta H}{\rho C_{\mathrm{p}}} k_0 \, e^{-\frac{E}{RT}} C_A + \frac{UA}{V\rho C_{\mathrm{p}}}(T_{\mathrm{c}} - T) + v_2\end{aligned} \tag{3.30}$$

where the variables are, q feed flow rate, C_{A} the concentration, T reactor temperature, T_{c} the cooling water temperature, C_{Af} the feed concentration, and T_{f} the feed temperature. For the simulation, $[C_{\mathrm{A}}, T]^\top$ are controlled variables, while $[T_{\mathrm{c}}, q]^\top$ are manipulated variables. Two PID controllers are utilized as $\mathbf{K}_2(K_1 + T_{\mathrm{d}}s + T_1/s)\epsilon$,

with ϵ being the control error $\epsilon = [C_{\mathrm{A}} - C_{\mathrm{A}}^*, T - T^*]^\top$. The default parameter setting has been given in [9] and is omitted here.

In the nominal case, process fluctuations are caused by system noises $\{v_1, v_2\}$, which are modelled as colored Gaussian noises, as well as T_{f}, which is a random walk process $T_{\mathrm{f}}(t) = T_{\mathrm{f}}(t-1) + e_{T_{\mathrm{f}}}(t),\ e_{T_{\mathrm{f}}} \sim \mathcal{N}(0, 0.15^2)$ to describe the non-stationary dynamics in the feed temperature induced by the upstream unit. Hence, there exist non-stationary and stationary variations simultaneously. The sampling time is set as $\Delta t = 10$s to collect five measured process variables $\{C_{\mathrm{A}}, T, T_{\mathrm{c}}, q, T_{\mathrm{f}}\}$. Thanks to the feedback control, T and C_{A} as controlled variables are stationary. Since T_{c} counteracts the non-stationary variations in T_{f} to stabilize the reactor temperature T, non-stationary yet normal variations continually occur in T_{c}. This is also confirmed by the ADF test, which recognizes T_{c} and T_{f} as non-stationary variables and the other variables as stationary ones.

1000 normal samples are collected from nominal process operations to build a training dataset, which is utilized to develop monitoring models. For this work, 4 stationary PSFs and 2 non-stationary PSFs are used in the NS-PSFA model based on the EM algorithm. Concerning comparison, CVA, which has been used extensively in recent years for process monitoring applications, is established. [55]. T^2 and SPE statistics are calculated with the CVA model, and all control limits are calculated with the confidence of 99%.

Next, we investigate the out-of-sample performance of two monitoring schemes based on three deliberately designed test scenarios with distinct mechanisms, each of which includes 1400 samples. In the first scenario, a step change is imposed upon T_{f} at the 400_{th} sample. As the effect of T_{f} is mitigated by manipulating T_{c}, process dynamics, as well as closed-loop control performance, there is no need to issue alarms. Depicted results of NS-PSFA and CVA are in Figure 3.2. Clearly, CVA is prone to considerable false alarms after the occurrence of the step change, whereas NS-PSFA yields much lower false alarm rates (FARs) and better interpretations. On one hand, the compensating behavior of controllers around the 400_{th} sample can be clearly observed; on the other hand, both non-stationary and stationary dynamics recover immediately, as indicated by S_n^2 and S_s^2. Moreover, T_s^2 and SPE_s also imply that steady operating conditions are not affected. In this sense, the proposed

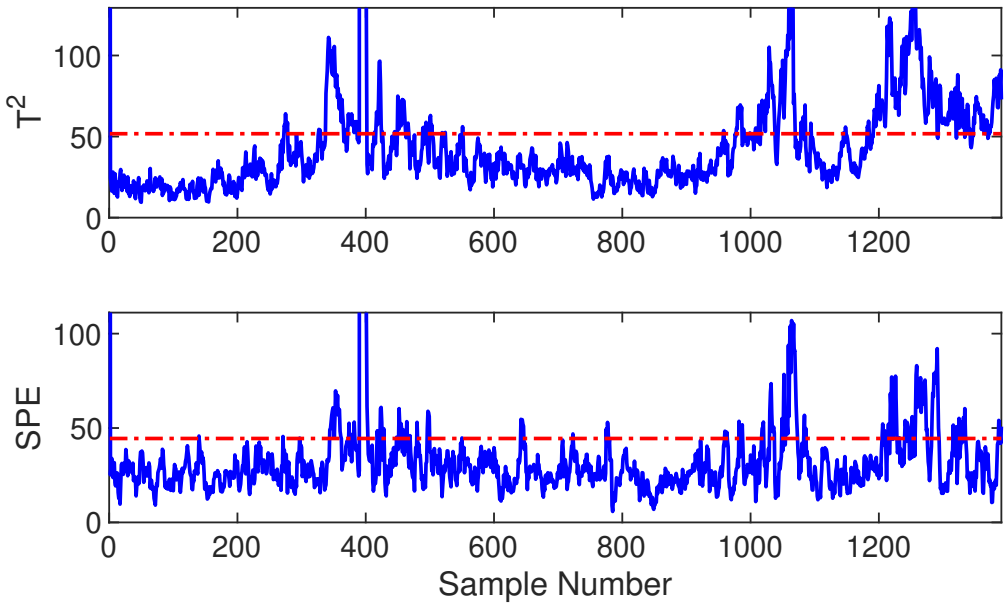


Figure 3.2: Monitoring results for test scenario 1 utilizing CVA

monitoring scheme can well accommodate nominal non-stationary variations under closed-loop control and leads to meaningful information by a synthesis of four monitoring statistics.

The second case is administered by an increase in the variance of the random-walk model of T_{f}, i.e. $e_{T_{\mathrm{f}}} \sim \mathcal{N}(0, 0.4^2)$ after the 400_{th} sample. In this case, non-stationary process dynamics get affected because T_{c} works aggressively and opposite to counteract the increase in non-stationarity of T_{f}, while stationary dynamics is almost intact due to feedback control, which can be inspected from control errors. Figure 3.4 summarizes the monitoring results, where CVA could signify the occurrence of an anomaly. By contrast, NS-PSFA showcases that only non-stationary dynamics are significantly affected since only the S_n^2 index exceeds its control limit frequently, which essentially differs from the first scenario. One can interpret the monitoring result of NS-PSFA as that the dynamics anomaly primarily arises from some external non-stationary disturbances.

In the third case, an internal dynamics anomaly is introduced by enlarging the variances of internal system noises v_1 and v_2. From (3.30), both process dynamics, and the operation condition will be simultaneously disturbed. Monitoring results are

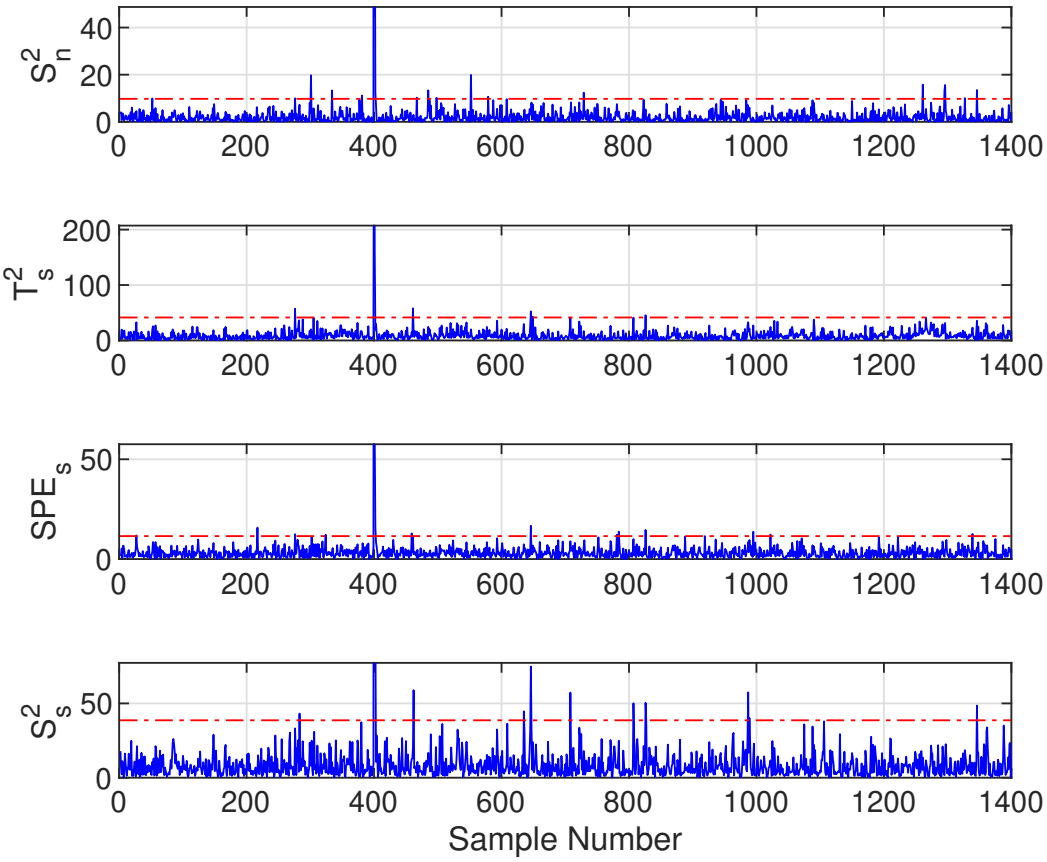


Figure 3.3: Monitoring results for test scenario 1 utilizing NS-PSFA

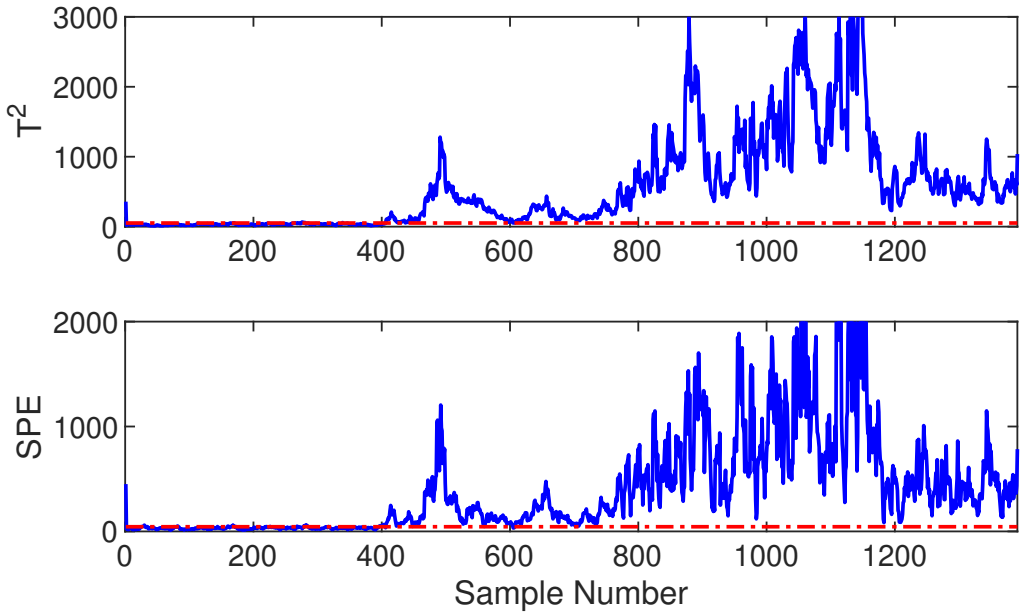


Figure 3.4: Monitoring results for test scenario 2 utilizing CVA

shown in Figure 3.6. Similar to the previous case, CVA can recognize the anomaly; however, one cannot distinguish their difference in detail. Utilizing a combination of interpretable monitoring statistics, NS-PSFA determines that both (non-stationary and stationary) process dynamics and steady states are disrupted, and the root-cause anomaly probably arises from the internal.

3.4.2 Three-phase flow facility process

The Three-phase Flow Facility at Cranfield University was designed to provide controlled and measured flow rates of water, oil, and air to a pressurized system for industrial processes [56]. The test area for the three-phase facility includes separators and connecting pipelines. The facility can be operated under changing process conditions, as it can handle a single phase of air, water, and oil, or a mixture of these three different fluids. The fluid mixtures for the process are to be separated in a three-phase separator. From the three-phase separator, air for the process is vented to the atmosphere, whereas oil and water are transported from the separator to their respective coalesces. In their respective coalesces, fluids are further separated and returned to their respective storage tanks. Additional information on the experimental

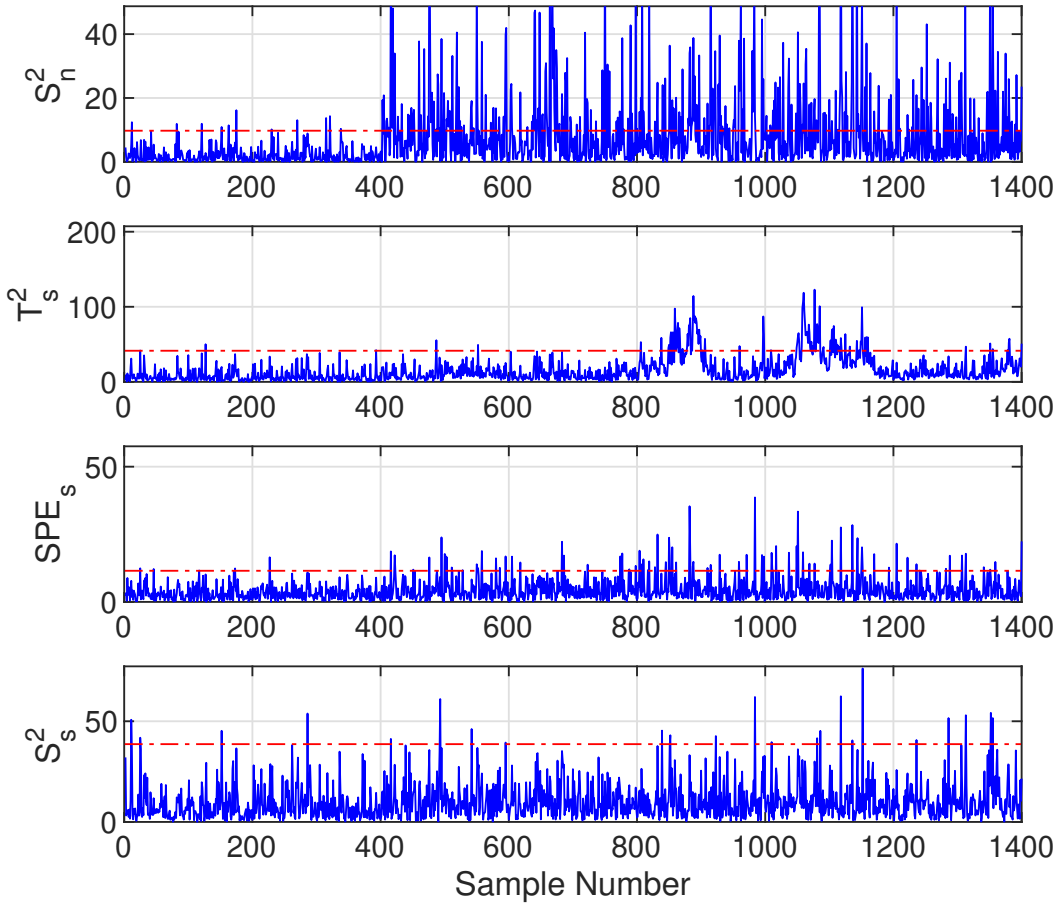


Figure 3.5: Monitoring results for test scenario 2 utilizing NS-PSFA

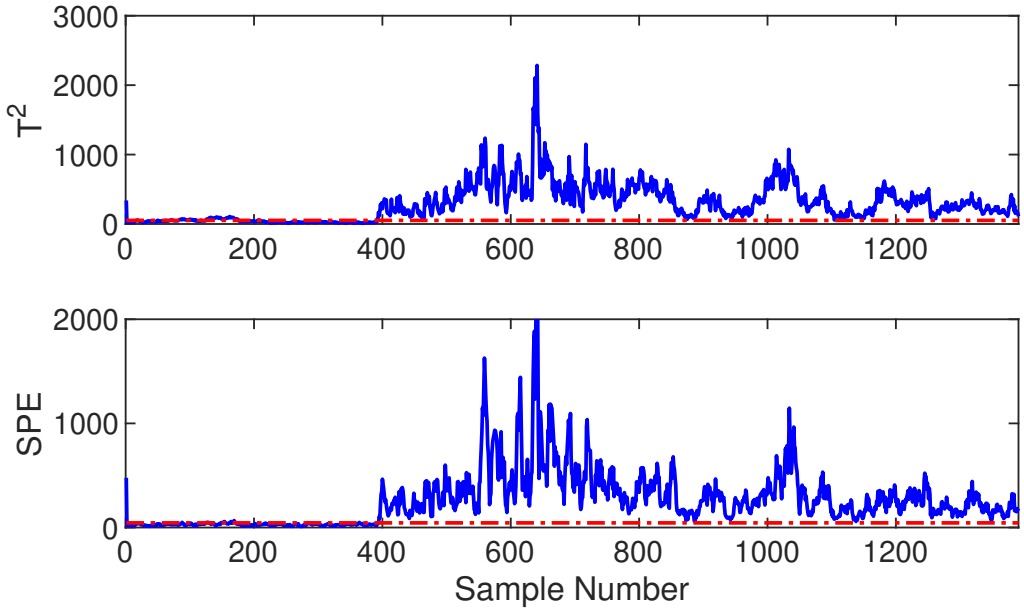


Figure 3.6: Monitoring results for test scenario 3 utilizing CVA

setup may be found in [56].

The process is known to have time-varying non-stationary process characteristics. It contains two process inputs, including air and fluid flow rates, which are deliberately varied to obtain three different datasets standing for nominal operations under changing conditions, denoted by T1, T2, and T3. In total, 20 different combinations of water and airflow rates are introduced in each normal dataset. Switches between these combinations ensure that three normal datasets are representative of typical non-stationary time-varying operating conditions [56].

It has been pointed out that the pressure in the mixture zone 2" line shall not be used for modeling and monitoring; hence 23 out of 24 process variables are used for analysis and monitoring. Meanwhile, six different faulty datasets were created with typical process malfunctions. To train the non-stationary PSFA model, the T3 dataset is utilized because of its representation [56]. Harnessing the ADF test, 13 out of 23 process variables are found to be non-stationary and subject to the influences of varying process condition changes, based on which the EM algorithm is executed. For model development, 15 non-stationary and 7 stationary PSFs are utilized. The EM algorithm converges in 50 iterations between the E-step and M-step. The proposed method is tested against both PCA and CVA to determine if monitoring performance

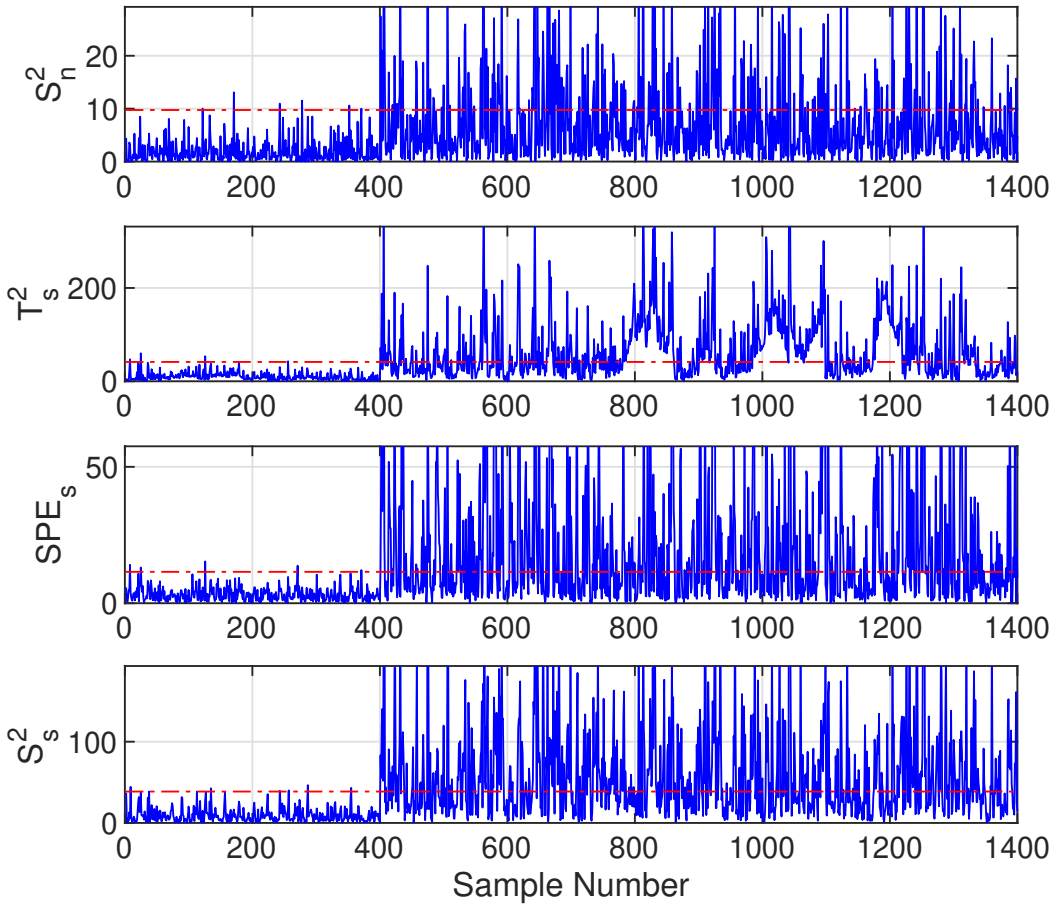


Figure 3.7: Monitoring results for test scenario 3 utilizing NS-PSFA

can be improved by utilizing the proposed NS-PSFA-based monitoring scheme. For comparing the proposed method, the CVA code provided by Ruiz and co-workers [56] is used without modifications. Two monitoring statistics are calculated and used for CVA as well as PCA, namely, T^2 and SPE monitoring statistics. All control limits are computed with confidence of 99%.

Generalization capability on nominal data under changing conditions

If a model fits training data well, but with a poor generalization capability, a high FAR can be induced, making the derived monitoring scheme more sensitive to disturbance and less reliable for online implementation. FARs are calculated for NS-PSFA, CVA, and PCA to determine the performance on out-of-sample normal datasets T1 and T2, as reported in Table 3.1. Notice that each algorithm utilizes different monitoring statistics, with PCA and CVA utilizing T^2 and SPE, and the proposed method utilizing SPE_s, S_s^2, S_n^2, and T_s^2. For the CVA and PCA methods, a general guideline to trigger alarms is to inspect whether any monitoring statistic exceeds the control limit thereof [57], which could occur for both operation condition switches and process anomalies. For the NS-PSFA-based monitoring scheme, it can differentiate between operating condition changes and process anomalies. Specifically, only if the S_s^2 or S_n^2 monitoring statistic goes beyond the limit, the system is indicated to have process anomalies, and alarms shall be issued. Otherwise, when a steady operation condition change is detected, alarms can be removed. Hence, for the NS-PSFA method, FARs of S_s^2 and S_n^2 are computed.

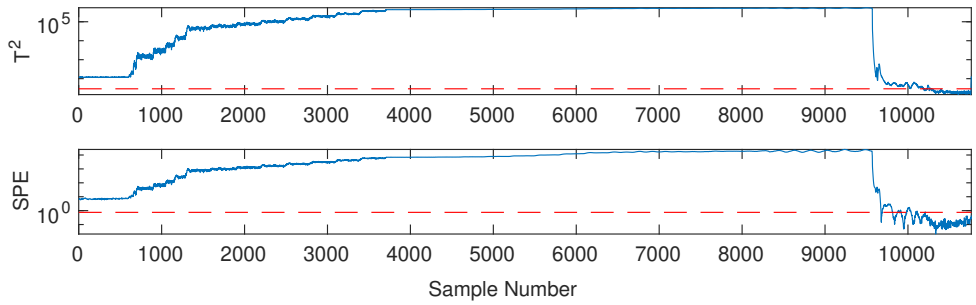
Table 3.1: False Alarm Rates for T1 and T2 Nominal Dataset of PSFA, CVA, PCA Algorithms (%)

Model	Statistics	T1	T2	Average
PSFA	S_s^2	0.30	1.50	0.90
	S_n^2	0.20	1.27	0.73
CVA	T^2	0.01	7.80	3.90
	SPE	0.12	0.09	0.11
PCA	T^2	100.00	100.00	100.00
	SPE	100.00	100.00	100.00

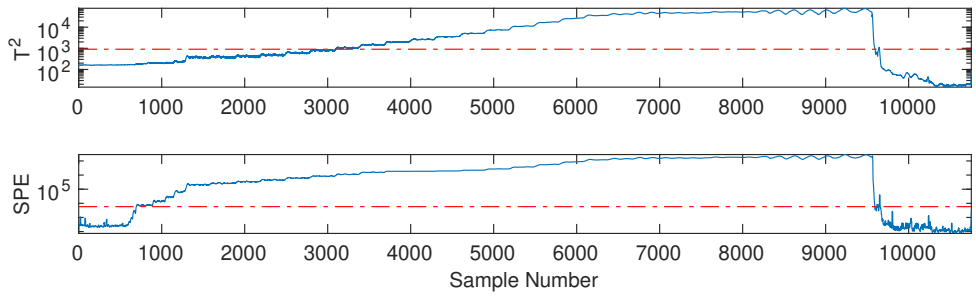
The T1, T2, and T3 datasets are not the equivalent in development. Therefore, it has been challenging to furnish low FARs for a variety of MSPM methods [56]. Utilizing PCA for monitoring does not generalize well on the T1 and T2 datasets, as demonstrated by the algorithm producing a FAR of 100% on the T^2 and SPE monitoring statistics. This result occurs as the PCA model develops on a single model, and any condition that is different from a nominal condition may be recognized as faulty. In contrast, the CVA method yields an average FAR of 3.90% and 0.11% for the T^2 and SPE statics, which shows its much better abilities generalizing on the T1 dataset, however, still produces a high FAR of 7.8% for T^2 when tested on the T2 dataset. For the NS-PSFA monitoring scheme, the average FAR on these two datasets utilizing the S_s^2 and S_n^2 monitoring statistics is less than 1%, which exhibits a satisfactory generalization capability under widely-changing non-stationary process operations.

Sensitivity to pre-specified process faults

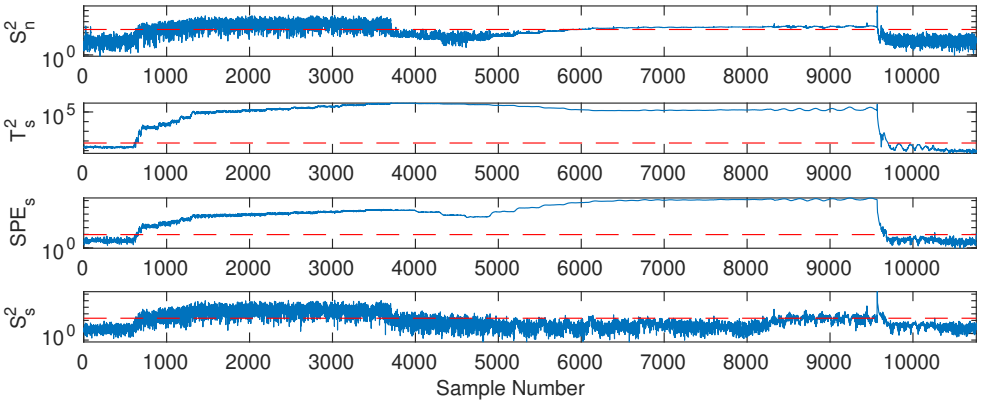
To compare the monitoring scheme based on NS-PSFA with CVA and PCA methods, two typical process faults, which are a separator input blockage and an open direct bypass, respectively, are used. The first fault, namely the case 3 of fault 3, is utilized where the fault begins at the 596_{th} sample and terminates at the 9566_{th} sample. Figure 3.8 shows the monitoring results for the PCA, CVA, and NS-PSFA algorithms, and the T3 dataset is used to train all models. In Figure 3.8(a), PCA is unable to fit process data well, as evidenced by being above the control limit before the fault occurs, which may undermine the entire monitoring system. Monitoring results of CVA showcase that CVA effectively detects the fault as it occurs by both T^2 and SPE statistics. However, they cannot supply other information regarding the type of fault occurrence. When applying the CVA algorithm for online monitoring, it is uncertain whether faults that are detected by operation condition changes, or rather, process anomalies. Monitoring results from the NS-PSFA algorithm contain two additional dynamics-specific monitoring statistics S_s^2 and S_n^2. These two statistics provide additional information when combined with the T_s^2 and SPE_s monitoring statistics. As the T_s^2 and SPE_s statistics quickly detect deviations in a steady-state sense, the S_s^2 and S_n^2 statistics gradually also uncover both non-stationary and sta-



(a)



(b)

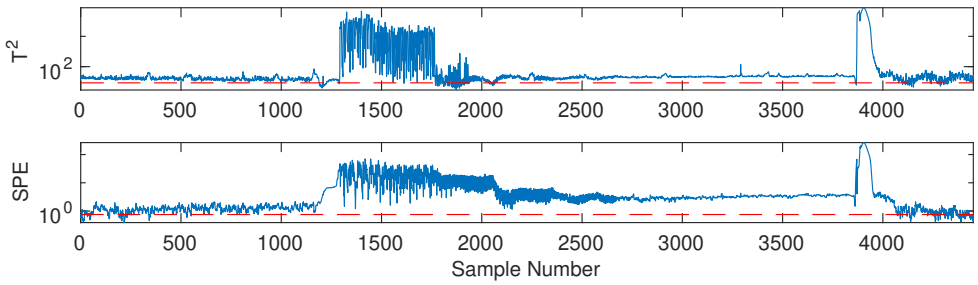


(c)

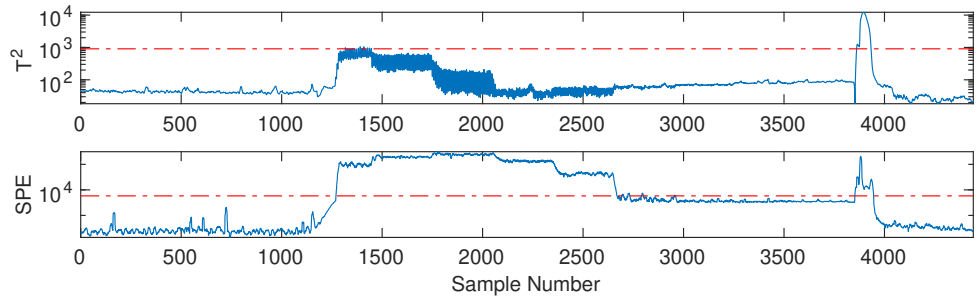
Figure 3.8: Monitoring results utilizing (a) PCA, (b) CVA, (c) NS-PSFA monitoring schemes for Fault 3, Case 3

tionary dynamics anomalies. As time goes by, the process gradually stabilizes, and hence S_s^2 becomes normal; nevertheless, the process moves to a different steady-state, as indicated by T_s^2 and SPE_s, and the non-stationary dynamics remains abnormal, as indicated by S_n^2. Therefore, the NS-PSFA monitoring scheme can provide knowledge about process status by using monitoring statistics with clear physical interpretations.

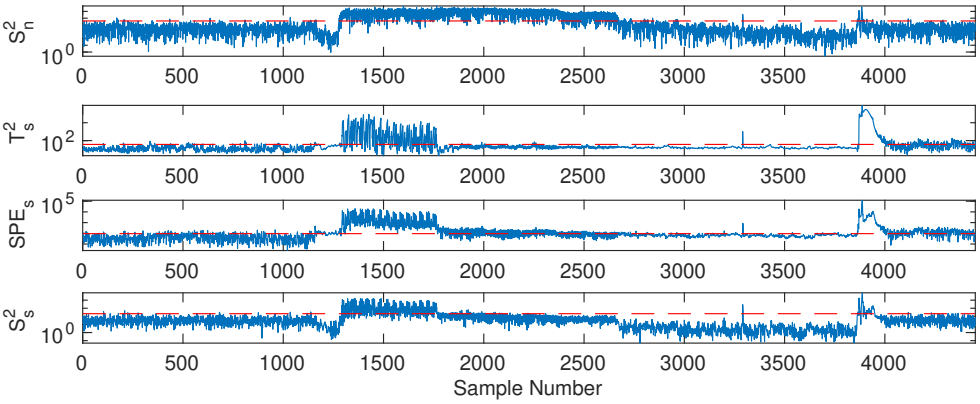
The second scenario (fault 4, case 2) involves an open direct bypass between the mixing point and the three-phase separator by introducing a gradual leakage, which is revealed at the 851_{st} sample and ends at the 3851_{st} sample. Figure 3.9 shows the corresponding monitoring results. Similar to the above case study, PCA is still prone to false alarms at the beginning. The CVA algorithm successfully showcases the fault since the SPE monitoring statistic does not decrease below the control limit during the fault occurrence. However, the T^2 statistic keeps below the threshold despite apparent variations. This result indicates that there exists a model mismatch since the threshold must be carefully re-tuned to give a consistent detection performance. Most importantly, it lacks abilities to uncover additional physical information from the system, such that practitioners receive vague information. As for the NS-PSFA monitoring scheme, the result appears to be much more rational and interpretable in that faulty process operations manifest different characteristics in different stages. After the fault reveals itself significantly (about 1250_{th} sample), all monitoring statistics quickly spike above their control limits. Afterward, all statistics gradually recede, which indicates the compensating behavior of controllers and the entire course of stabilization. Finally, after the 2600_{th} sample, process dynamics return to normal at a newly created working point, as suggested by SPE_s still lying above the threshold. After the fault disappears, around the 3900_{th} sample, the short-lived dynamics induced by compensating behaviors of controllers is observed, based on all monitoring statistics. In this sense, the combination of four monitoring statistics of NS-PSFA provides additional interpretable information regarding the system status, which is in good correspondence with the process mechanism.



(a)



(b)



(c)

Figure 3.9: Monitoring results utilizing (a) PCA, (b) CVA, (c) NS-PSFA monitoring schemes for Fault 4, Case 2

Chapter 4

Industrial applications of NS-PSFA algorithm

Research on electric submersible pumps (ESPs) longevity [58], decrease of failures [59], and maritime optimization [60] is have been explored. However, ESP optimization for technologies particular to Canadian steam-assisted gravity drainage (SAGD) and oil sand processes is unrealized. The NS-PSFA algorithm is employed for industrial application on ESPs as ESPs have recurring operational concerns that can produce early failures. ESPs are an effective agent to test the NS-PSFA algorithm, as the time-series is non-stationary. The objective of the application on ESPs is to employ the NS-PSFA algorithm on data from ESP operation and monitoring scheme to distinguish irregularities. A second goal of the study is to discover if failures of an ESP can be determined before they occur.

4.1 Case Study on Electric Submersible Pumps

Optimizing ESP performance utilizing data-driven modeling is crucial to reducing maintenance costs and inept runtime [61]. There are former studies that outlined constructing a monitoring scheme that collected, transmitted, and analyzed data to recognize and diagnose issues with ESP operations [62]. The NS-PSFA algorithm developed in this Thesis utilizes data-driven modeling, whereas preceding approaches utilized first-principle modeling.

4.1.1 Steam-assisted gravity drainage process with electric submersible pump application

Figure 4.1 is the process schematic for a typical SAGD operation in the petroleum industry, utilizing an ESP. In the SAGD process, fluids are extracted from underground

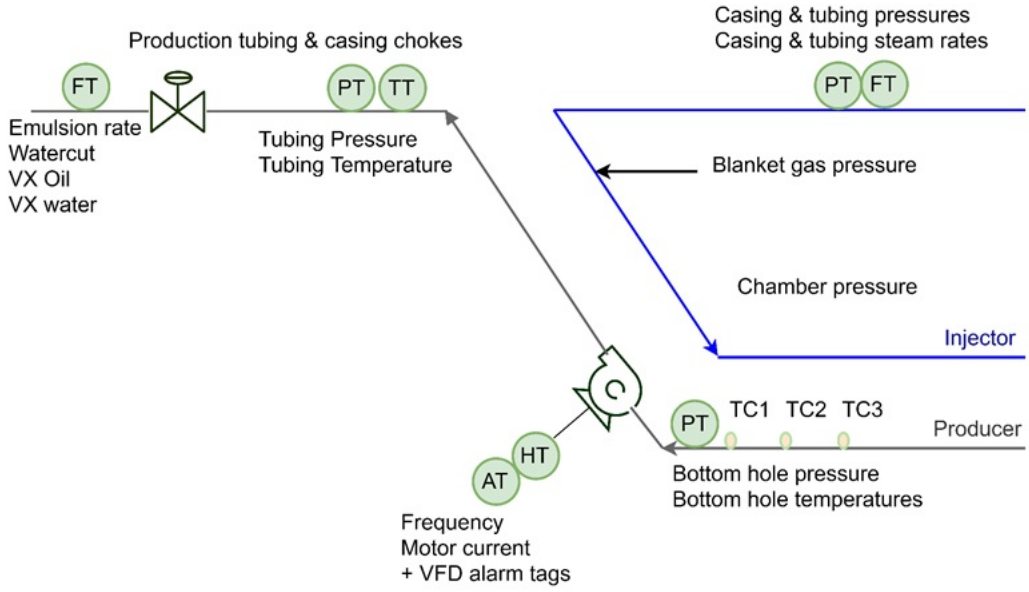


Figure 4.1: Electric submersible pump operating schematic in SAGD operation

to the surface, notably oil and gas, are lifted to the surface by artificial lift. The ESP is a form of artificial lift that provides additional energy to lift fluids to the surface. During SAGD operation, two wells are utilized to extract petroleum fluids to the surface. These two wells are an injector well and a producer well, which respectively, are the top and bottom wells in Figure 4.1. High-pressure-steam (HPS) is pumped from the injector well, into a reservoir. The goal of using HPS is to decrease the viscosity and increase the temperature of fluids inside the reservoir to be extracted. Due to the decrease in viscosity, fluids reach a level of fluidity that allows efficient extraction via ESP. For an SAGD process, the ESP is part of the producer well.

4.1.2 Data preprocessing

Variables chosen from the producer well for modeling the ESP operation utilizing the NS-PSFA algorithm include the bottom hole pressure, ESP frequency, ESP motor current, tubing pressure, and tubing temperature. These five variables are the typ-

ically measured variables in an ESP installation. For each ESP analyzed, these five variables have less than 10% of missing data we have recieved. Data preprocessing is essential before feature engineering and model development.

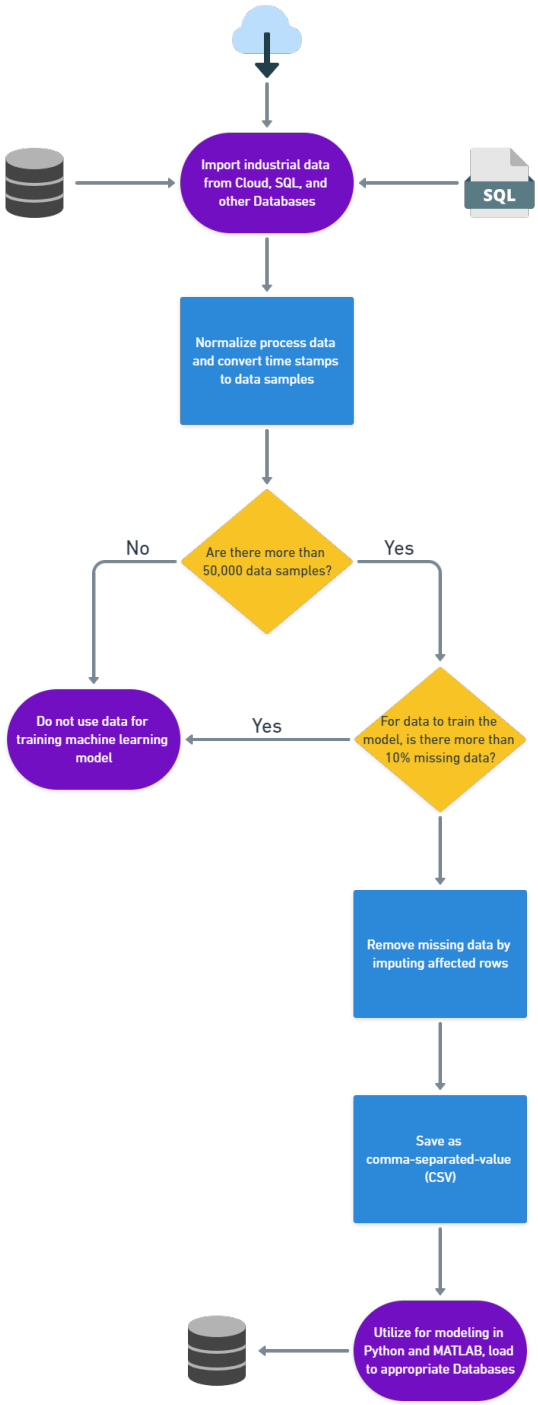


Figure 4.2: NS-PSFA data preprocessing schematic

The schematic of data preprocessing for the process monitoring application of ESPs utilizing the NS-PSFA algorithm is found in Figure 4.2. Industrial data is extracted for each ESP operation from a historical database or a cloud database. Afterward, data preprocessing transpires where data is mean normalized, and process tags are removed, for proprietary reasons. Only ESPs that have more than 50,000 data samples are utilized for training and modeling to ensure each NS-PSFA model has sufficient data for model training and testing. Each ESP dataset that achieves this criterion has data imputed by removing rows that have missing data.

Table 4.1: Data partitioning of training, test, and validation data sets for case studies of ESP 1, ESP 2, and ESP 3 (%)

Equipment	Case Study	Training dataset	Test dataset	Validation dataset
ESP 1	1-1	70.00	30.00	
	1-2	70.00	30.00	
	1-3	40.00	60.00	
ESP 3	2-1	67.00	33.00	
	2-2	22.00	78.00	
	2-3	22.00	78.00	
ESP 3	3-1	40.00	60.00	
	3-2	20.00	60.00	20.00

The modeling pipeline has data separated into training, validation, and test datasets. Once data is partitioned into training, test, and validation data sets, the ADF test is administered employing training data to determine which variables are stationary or non-stationary. Partitioning of non-stationary and stationary variables into categories is based on ADF test results. Initialization of non-stationary and stationary process data in the learning algorithm begins training and optimizing model parameters.

Data from 3 ESP installs are available. Operational data from an industrial sponsor is normalized and utilized. Each ESP install has the sampling time set as $\Delta t = 5\text{m}$ to collect five measured process variables. ESP 1 contains 71,000 data samples, ESP 2 contains 180,000 data samples, and ESP 3 contains 294,000 data samples. In Section 4.2.1, ESP 1 and ESP 3 are utilized to investigate log-likelihood convergence during model training of the NS-PSFA algorithm. ESP 1 has 50,000 data samples utilized to train the NS-PSFA model and ESP 3 has 200,000 data samples utilized

to train its model. Utilizing 50,000 data samples for ESP 1 and 200,000 for ESP 2 is done to determine if different sized data sets reflect changes in the convergence of log-likelihood value. Each model is trained using 80 iterations utilizing 5 PSFs, 1 stationary PSFs and 4 non-stationary PSFs, respectively, based on the EM algorithm. Afterward, ESP 1 utilizes 50,000 data samples and is initialized for five instances to determine if increased initializations of the NS-PSFA algorithm is advantageous in the convergence of log-likelihood.

Section 4.2.2 studies the utilization of the NS-PSFA algorithm to monitor ESP 1, ESP 2, and ESP 3 and realize the goals outlined in the preface of Chapter 4. Each case study for all ESPs utilizes 45 iterations through the EM algorithm. For each case study, the partitioning of datasets is presented in Table 4.1. The presentation of three distinct case studies for ESP 1 is in Section 4.2.2. Case study 1-1 trains the NS-PSFA algorithm utilizing 4 non-stationary PSFs and 1 stationary PSF. Case study 1-2 investigates utilizing 4 non-stationary PSFs and 2 stationary PSFs to model the NS-PSFA model. Determining if stationary monitoring statistics S_s^2, T_s^2, and SPE_s can detect operation condition changes and anomalies with greater accuracy, additional stationary PSFs are employed. Case study 1-3 investigates to ascertain if this trained model can become fine-tuned to operating condition changes and process anomalies. Case study 1-3 is trained utilizing 4 non-stationary PSFs and 1 stationary PSF.

The investigation of three case studies for ESP 2 is in Section 4.2.2. Case study 2-1 trains the NS-PSFA algorithm based on the EM algorithm, where 3 non-stationary PSFs and 2 stationary PSFs are used. Case study 2-2 investigates utilizing 3 non-stationary PSFs and 2 stationary PSFs to model the NS-PSFA model. Results of case study 2 showcase the NS-PSFA model, perhaps over-generalizes process data; thereby, when applied on the system, it loses the ability to determine operating condition changes and anomalies. Thereby, data is partitioned in this fashion to determine if the model results in stronger sensitivities while detecting operation condition changes and anomalies. Case study 2-3 investigates utilizing an additional PSF for NS-PSFA modeling.

The examination of two case studies for ESP 3 is in Section 4.2.2. Case study 3-1 trains the NS-PSFA algorithm utilizing 3 non-stationary PSFs and 2 stationary PSFs are used. Case study 3-2 investigates utilizing 3 non-stationary PSFs and 2

stationary PSFs to model the NS-PSFA model. Validation data is utilized in case study 3-2 for ESP 3, as ESP 3 possesses 40% and 76% more data than ESP 2 and ESP 1, respectively. By possessing a plethora of process data, data is reserved for validation to confirm the fit of the NS-PSFA model before application on monitoring data. Based on the results of ESP case study 3-2, the T_s^2 control limit is adjusted for online monitoring.

Section 4.3 examines a case study where the NS-PSFA algorithm learns parameters from a specific ESP; and then implements the prepared model on a different ESP. The model is prepared to utilize ESP 3 data, where the dataset has been partitioned to possess approximately 50% training data and 50% validation data. For this employment of the NS-PSFA algorithm, 3 stationary PSFs and 2 non-stationary PSFs are employed in the NS-PSFA model. The goal of the application of this pretrained NS-PSFA algorithm on a different ESP application is to achieve application objectives identified in the preface of Chapter 4 when utilizing different time-series data. For online application, a 95% control limit is utilized for the T_s^2 monitoring statistic. Testing of the NS-PSFA monitoring scheme versus the CVA method is conducted. The CVA algorithm utilizes the T^2 and SPE monitoring statistics. The CVA algorithm is equipped to employ a 99% control limit for both monitoring statistics.

The process schematic for modeling the NS-PSFA algorithm for industrial applications is in Figure **??**. First, data is loaded into the operation, then partitioned into different datasets. Subsequently, the ADF test partitions data variables as stationary or non-stationary. The initialization of the EM algorithm occurs by choosing the number of PSFs and parameter guesses. A model is selected, to which the Kalman Filter is applied to extract PSFs. The calculation of monitoring statistics proceeds. The NS-PSFA model then can be adopted for online monitoring.

4.2 Results and Discussion

4.2.1 EM Algorithm model training termination point and initialization

Once training completes through the EM algorithm, proceeding to update model parameters is not advantageous based on the marginal benefit gained when contrasted

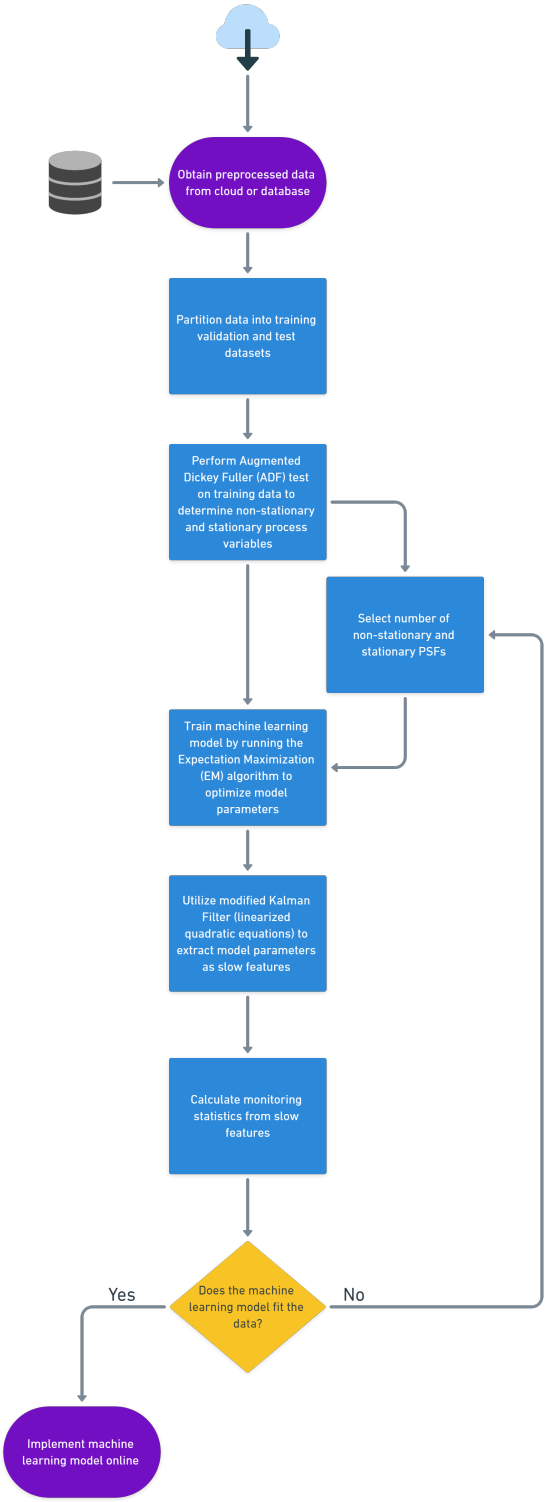


Figure 4.3: NS-PSFA schematic for modeling industrial processes

to the time necessary to train the algorithm on an iteration. Locating the termination point of the EM algorithm for model training of the NS-PSFA algorithm is in Figures 4.4(a) and 4.4(b). For both Figure 4.4(a) and 4.4(b), the black dotted line denotes where the termination point of model training should occur, and the red star denotes at which value the termination occurs. At the termination point of training, each additional iteration afterward provides less than 0.05% gain in raising the log-likelihood value.

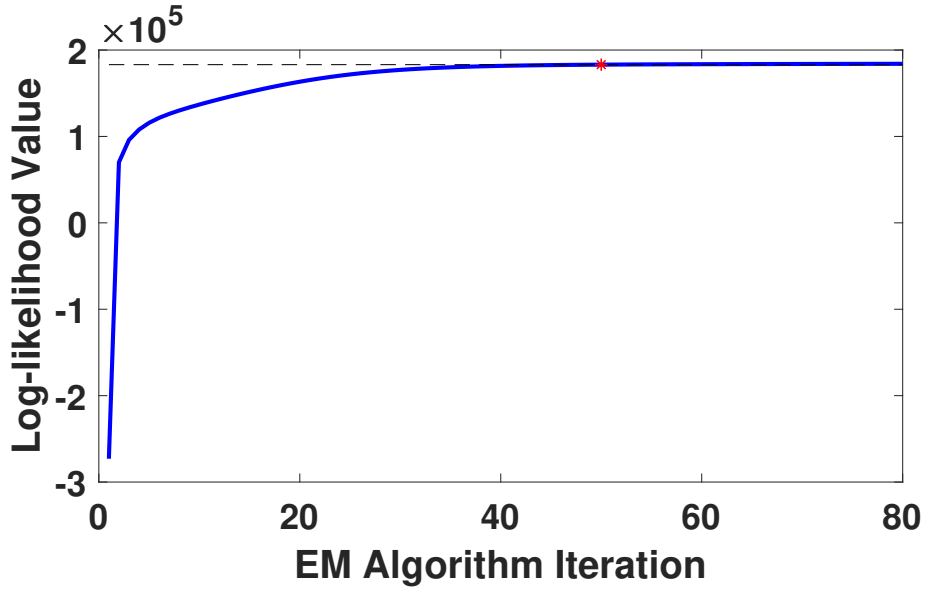
Figures 4.5(a) and 4.5(b) showcase five different initialization results of the NS-PSFA algorithm. The difference in log-likelihood values from the model with the highest log-likelihood value to the model with the lowest log-likelihood value is 34.4%. It is ideal for performing random initializations of the NS-PSFA algorithm as log-likelihood values can converge to higher values.

4.2.2 ESP monitoring

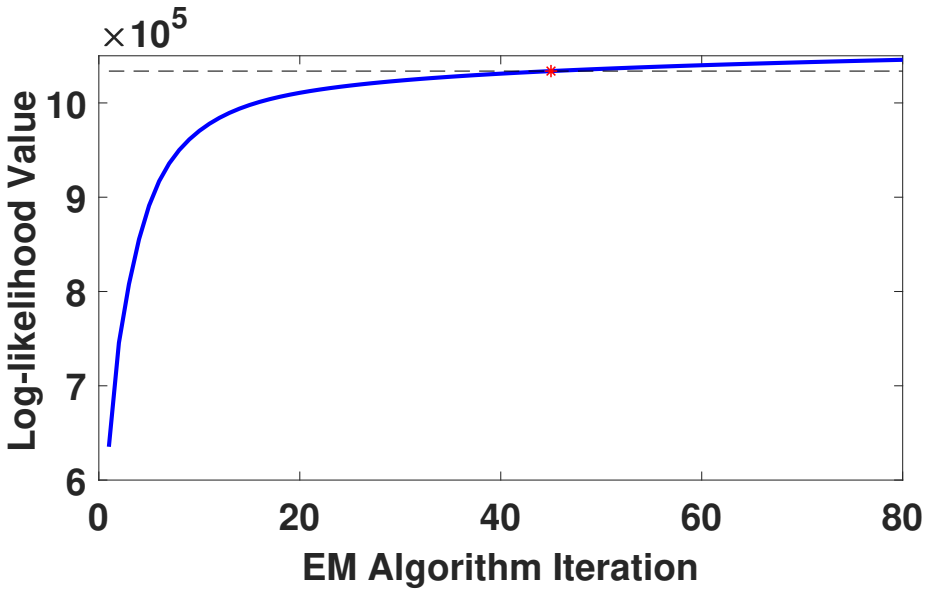
ESP 1 process monitoring

Monitoring results of data for case study 1 of ESP 1 are plotted and seen in Figure 4.6(a) and Figure 4.6(b). During model training, non-stationary dynamics from monitoring statistics S_n^2 discovers two periods of non-stationary irregularities that the model realizes. Respectively, this occurs from data samples 5,000 to 15,000 and 25,000 to 31,000. Once disturbances concede, non-stationary process dynamics stay below calculated 99% control limits. Following the $35{,}000_{th}$ data sample, stationary process dynamics from the S_s^2 monitoring statistic increment in value and oscillate at a value merely below the 99% control limit. Stationary dynamics are affected comparatively quickly, which occurs due to an anomaly.

Stationary dynamics and statics from test data utilizing the monitoring statistics S_s^2 and T_s^2 and SPE_s, are close to the 99% control limit until the $7{,}000_{th}$ data sample. Afterward, from the $8{,}000_{th}$ data sample onward to the $13{,}500_{th}$ data sample, monitoring statistics verify the process is not undergoing operation condition changes or process anomalies. Ensuing, the $13{,}500_{th}$ sample, each of the monitoring statistics exceed their control limits. The T_s^2 monitoring statistic showcases static equilibrium issues, as it surpasses the control limit, particularly following the $13{,}500_{th}$ data sam-

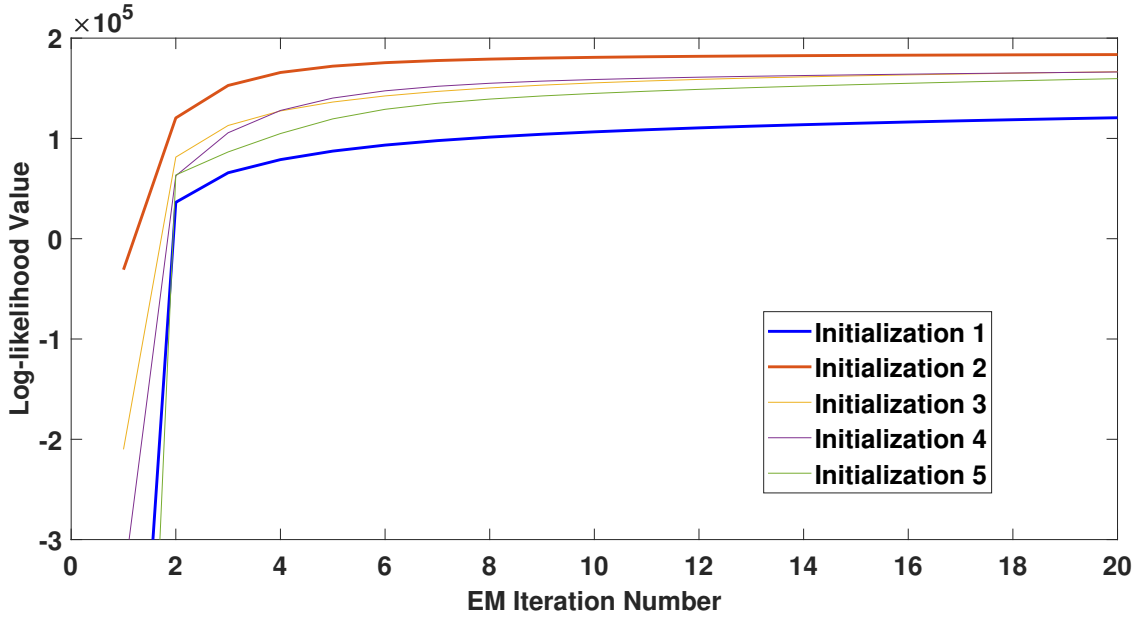


(a)

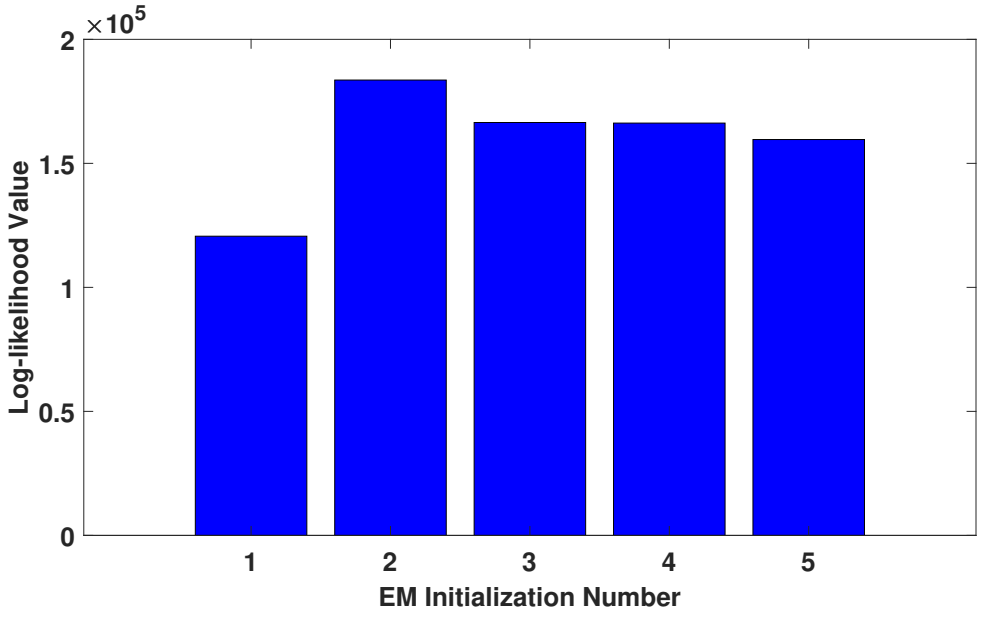


(b)

Figure 4.4: EM model training termination via log-likelihhood curve for (a) ESP 1, (b) ESP 2.

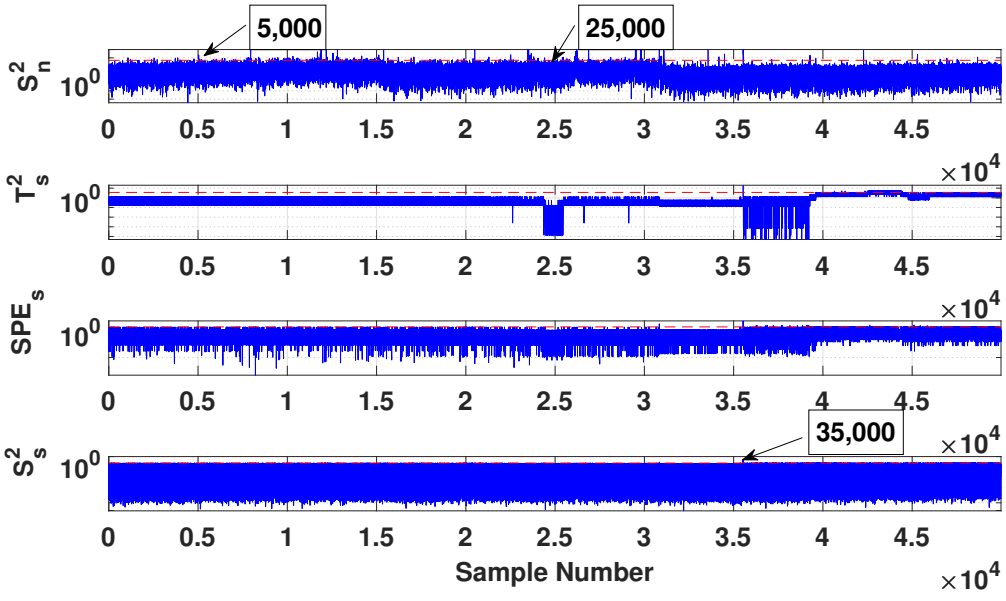


(a)

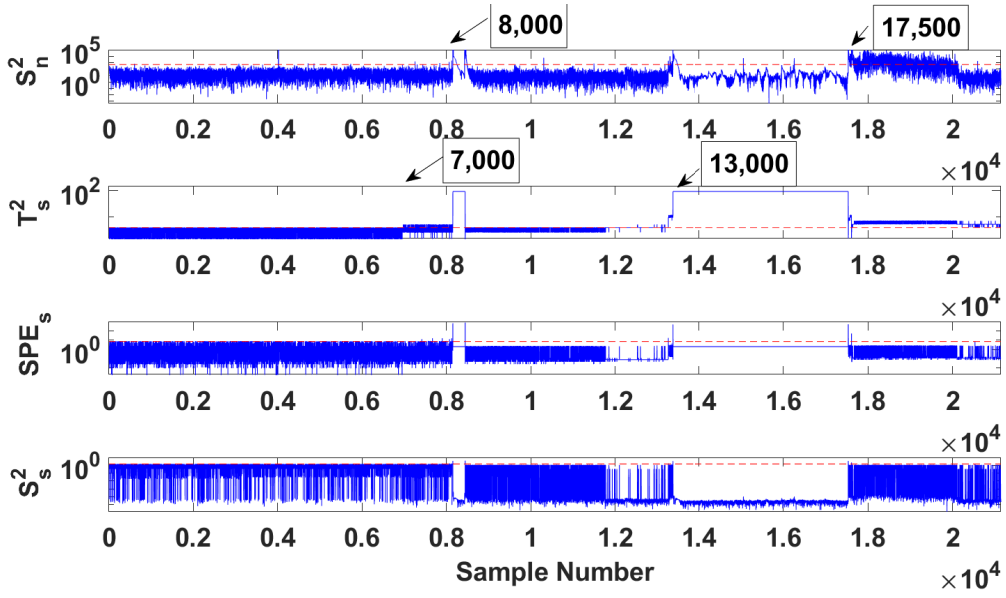


(b)

Figure 4.5: Log-likelihood values from the NS-PSFA algorithm from five iterations on (a) ESP 1 log-likelihood values, (b) ESP 1 final log-likelihood value.



(a)



(b)

Figure 4.6: Monitoring results utilizing NS-PSFA monitoring scheme on ESP 1 case study 1-1 utilizing (a) training data, (b) test data.

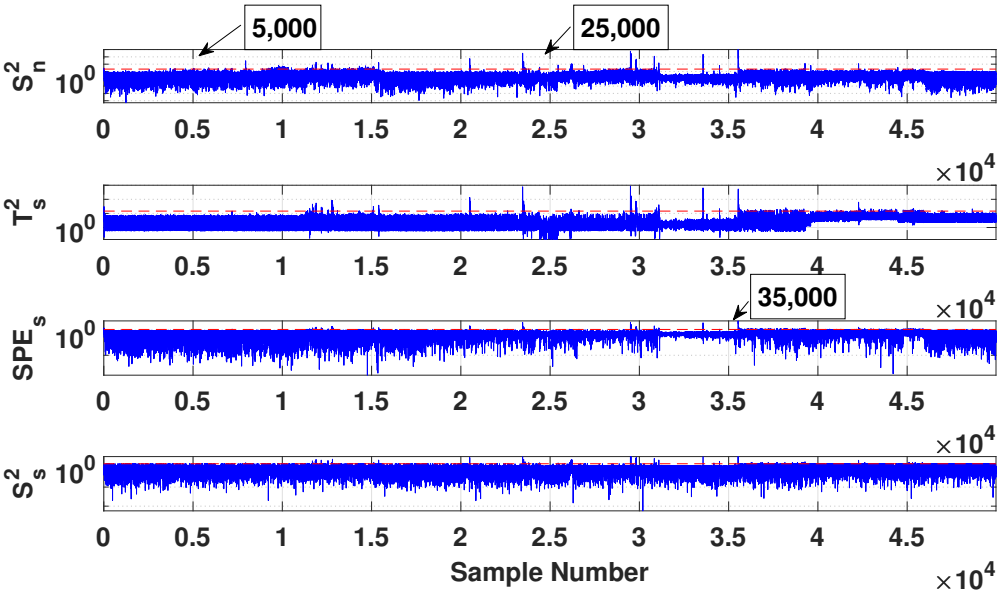
ple. While this occurs, three monitoring statistics exceed their thresholds after this event, with the event suggesting the monitoring threshold of T_s^2 is exceeded; however, others not exceeding indicates an operation condition change. Next, the process undergoes notable faults, indicated by the S_n^2 monitoring statistic increasing in value after the 17,500$_{th}$ sample where each of the S_n^2 and S_s^2 monitoring statistics pass their control limits. These two monitoring statistics are either both over, or periodically over their control limits until the 21,000$_{th}$ data sample, then the ESP fails. An operation condition change from the T_s^2 monitoring statistic and other changes in monitoring statistics correlates with notable faults from the three other monitoring statistics.

Results for utilizing additional stationary PSFs for modeling are shown in Figures 4.7(a) and 4.7(b). Training data in Figure 4.7(a) showcases that additional stationary PSFs remove variations that previously affected non-stationary statics and dynamics between the 5,000$_{th}$ to 15,000$_{th}$ data samples and 25,000$_{th}$ to 31,000$_{th}$ data samples.

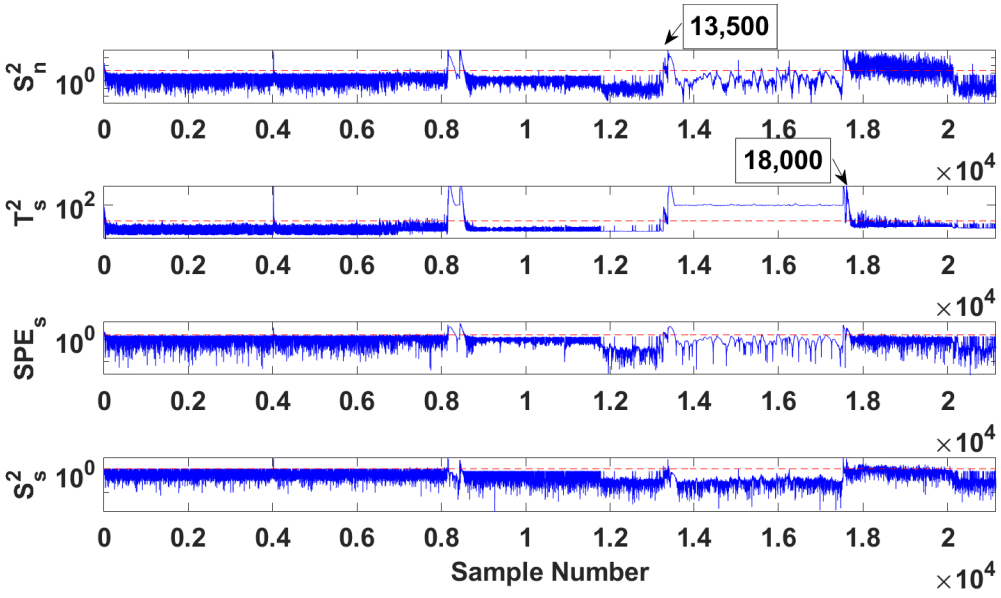
When analyzing the effect of employing fewer PSFs for model development, the T_s^2 and S_s^2 monitoring statistics are uniform throughout the training period. Monitoring statistics from stationary PSFs uncover strong sensitivities to process disturbances at the 35,000$_{th}$ data sample and onward from data as operations change. Following the 35,000$_{th}$ data sample, both T_s^2 and S_s^2 start to top their control limits, displaying stationary PSFs have depicted operation condition changes, that signal irregularities in the process.

From test data, there are operation condition changes discovered from the 13,000$_{th}$ data sample till the 18,000$_{th}$ data sample. Disturbances are recorded following the 18,000$_{th}$ data sample when a process anomaly affects the ESP. Supplementary disturbances noticed by the algorithm are identified around the 8,000$_{th}$ data sample. For stationary monitoring statistics, the main disadvantage in case study 2, as seen in Figure 4.6(b), the T_s^2 monitoring statistic proposed operational changes that occurred before separate monitoring statistics. Beyond this acumen, the algorithm functions likewise to case study one for ESP 1; however, stationary monitoring statistics reveal operation condition changes and anomalies more effectively.

Results are presented in Figure 4.8(a) and Figure 4.8(b) for case study 3 on ESP 1. In Figure 4.8(a), non-stationary fluctuations from monitoring statistic S_n^2 occur from

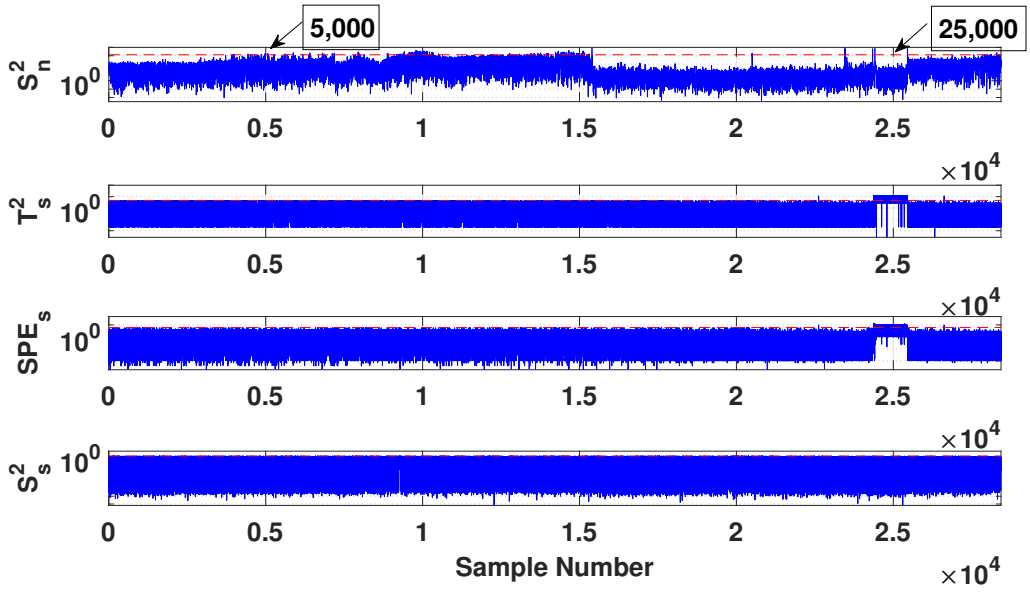


(a)

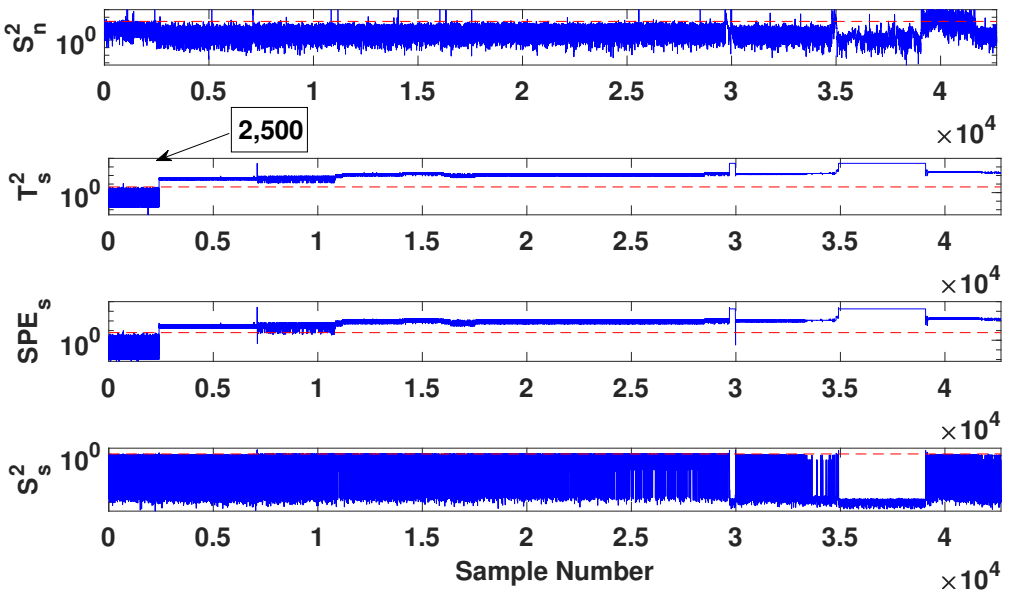


(b)

Figure 4.7: Monitoring results utilizing NS-PSFA monitoring scheme on ESP 1 case study 1-2 utilizing (a) training data, (b) test data.



(a)



(b)

Figure 4.8: Monitoring results utilizing NS-PSFA monitoring scheme on ESP 1 case study 1-3 utilizing (a) training data, (b) test data.

the $5{,}000_{th}$ to $15{,}000_{th}$ data samples, where the control limit is iteratively exceeded. Following the $25{,}000_{th}$ data sample, non-stationary monitoring statistics raise in value, whereas, stationary monitoring statistics values decrease. The T_s^2 values in 4.8(a) depict values of this statistic that are close to the control limit. As a control limit is used to determine when disturbances occur, values that are consistently very close to the control limit value can become problematic.

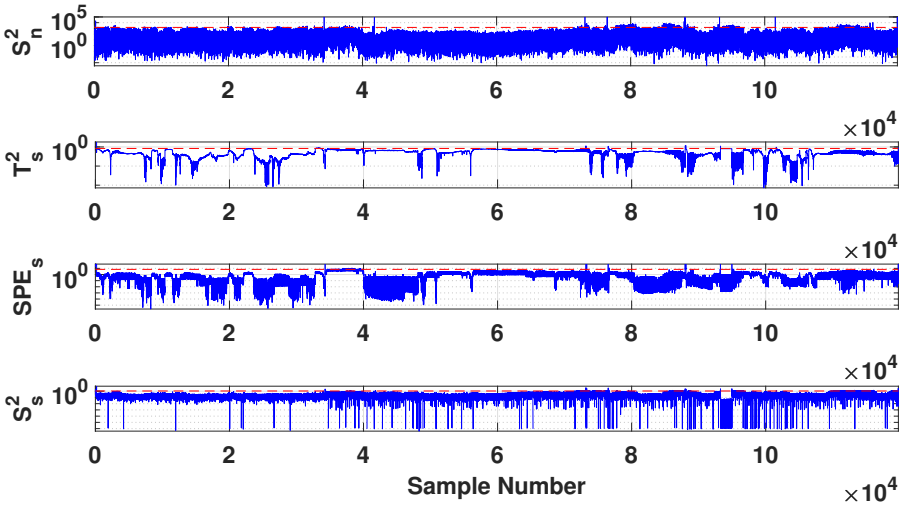
Stationary process data depicts operation condition transitions from testing data in Figure 4.8(b) directly following model deployment. Operation condition changes represented from the SPE_s and T_s^2 stationary monitoring statistics start at the $2{,}500_{th}$ data sample and do not halt until the pump fails. Alarms do not occur as process dynamic monitoring statistics do not exceed control limits in tandem with the T_s^2 and SPE_s monitoring statistics. At specific periods during the industrial process, alarms are signaled as the S_n^2 and S_s^2 monitoring statistics also exceed their control limits.

ESP 2 monitoring

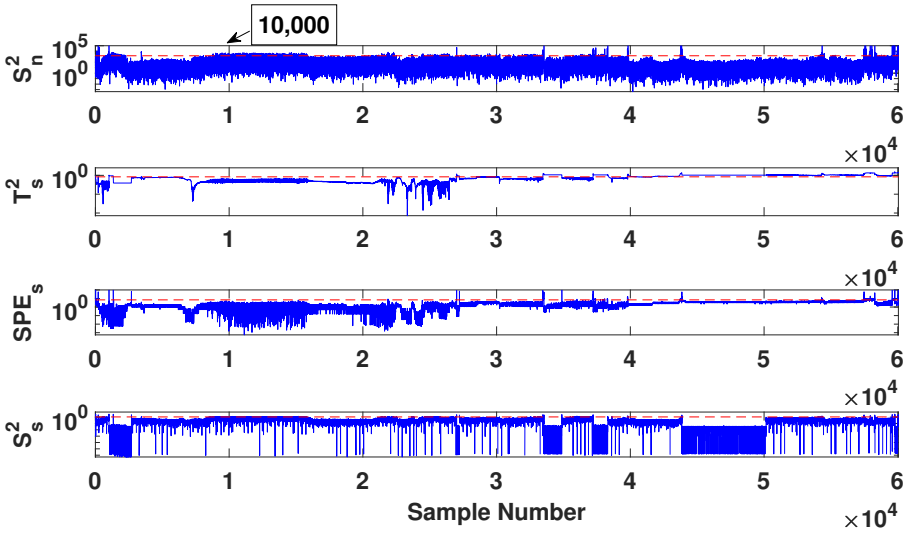
Results for training and test datasets of ESP 2 case study 1 are shown in Figures 4.9(a) and 4.9(b). From training results in Figure 4.9(a), the S_s^2 and S_n^2 statistics exceed control limits at different time periods. The physical depiction this result exhibits is the NS-PSFA model's monitoring scheme can distinguish between process anomalies and operation condition changes with ease. In ESP 2 case study 1, the S_n^2 and S_s^2 monitoring statistics uncover cohesive patterns from data variables that symbolize either operation condition changes or process anomalies.

Test results for monitoring ESP 2 case study 1 in Figure 4.9(b) depict process issues that subsist between the $10{,}000_{th}$ and $40{,}000_{th}$ data samples as seen by the S_n^2 and S_s^2 monitoring statistics overtaking control limits. The main drawback is the NS-PSFA model is conceivably overfitted from model training, as the T_s^2 and SPE_s monitoring statistics do not exceed their control limits in connection with the S_s^2 monitoring statistic.

Results for ESP 2 case study 2 are seen in Figure 4.10(a) and Figure 4.10(b), respectively. From monitoring results in Figure 4.10(b), all four monitoring statistics reveal severe sensitivities to operation condition changes and process anomalies, as evidenced by transcended control thresholds. An upside of a training algorithm utilizing

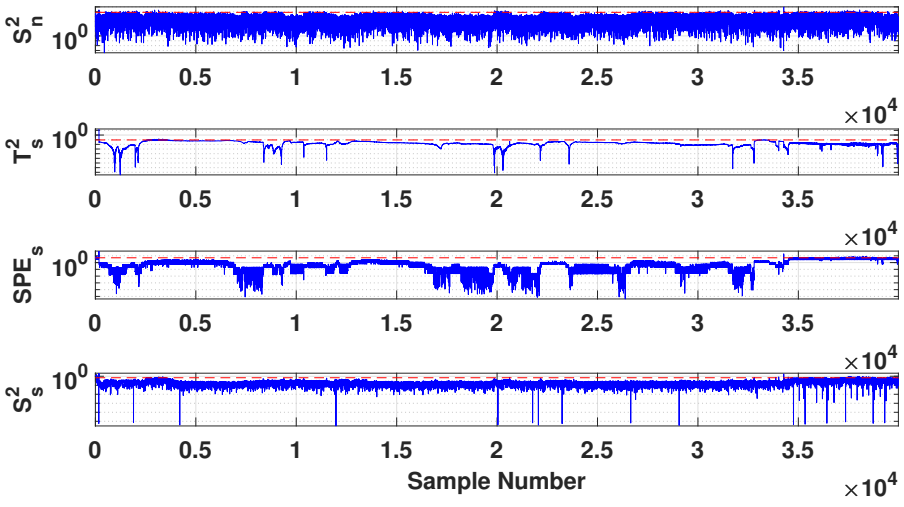


(a)

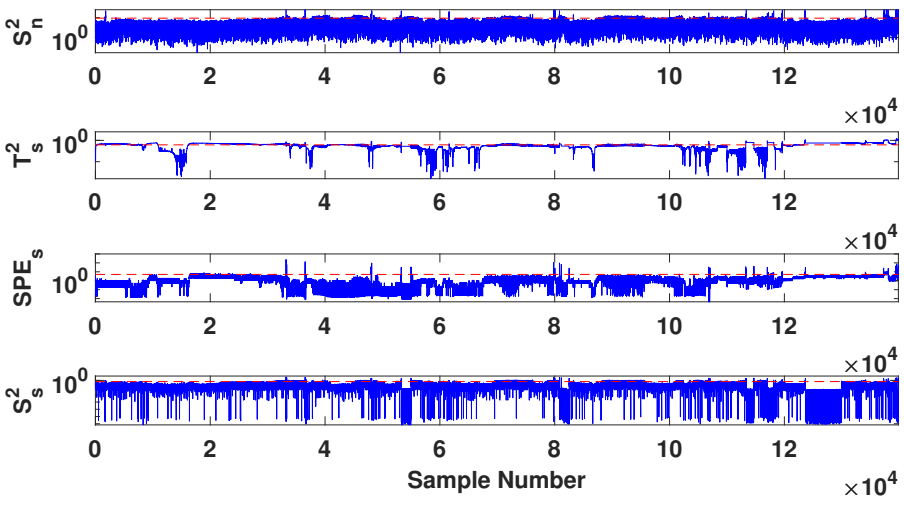


(b)

Figure 4.9: Monitoring results utilizing NS-PSFA monitoring scheme on ESP 2 case study 2-1 utilizing (a) training data, (b) test data.



(a)



(b)

Figure 4.10: Monitoring results utilizing NS-PSFA monitoring scheme on ESP 2 case study 2-2 utilizing (a) training data, (b) test data.

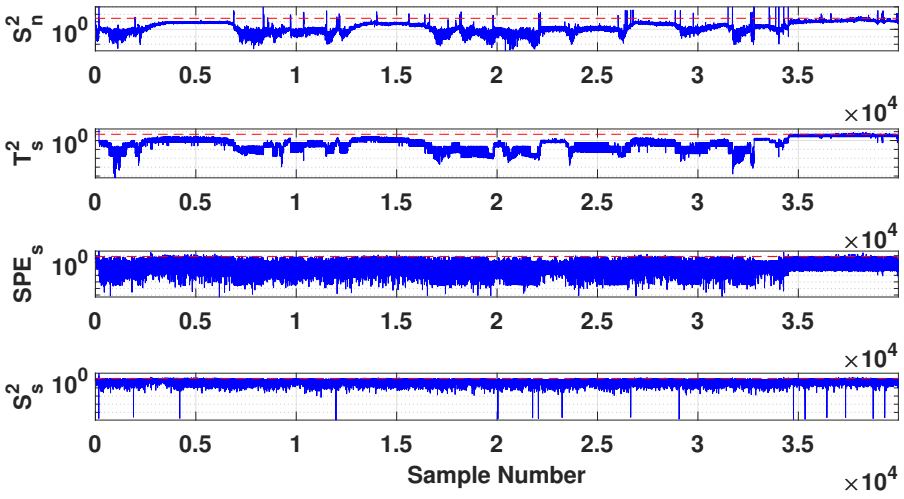
less training data is that it is more adept at processing variations and disturbances. Therefore, there is a more potent sensitivity towards anomalies that promotes repairing them as they transpire. A downside of using a model with a strong sensitivity is that the trained model can be oversensitive to process fluctuations and trigger false alarms, making application challenging. This result is evidenced by the S_n^2 monitoring statistic residing close to the 99% control threshold for the widespread majority of the test data during model application. Employing fewer data can have drawbacks for parameter learning and when training a model.

Results for ESP 2 case study 3 supplementing a stationary PSF for model training has results presented in Figure 4.11(a) and Figure 4.11(b). From monitoring results, the responsiveness of the S_n^2 monitoring statistic decreases. In turn, this reduces alarms raised by S_n^2 during model training.

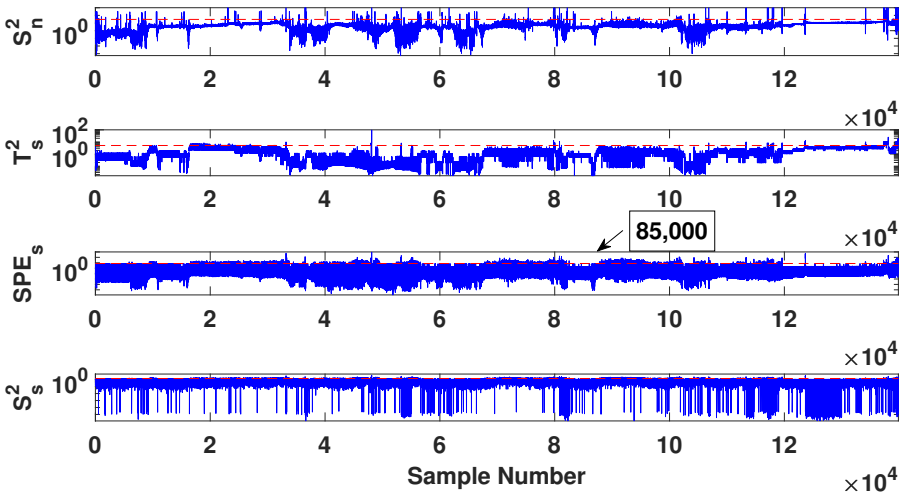
Monitoring results for the trained algorithm for ESP 2 case study 3 are exhibited in Figure 4.11(b). The results from this trained machine learning algorithm, in contrast with the previously trained NS-PSFA model for ESP 2 case study 2 in Figure 4.10(b), describe similar faults from training data. The SPE_s and S_s^2 monitoring statistics discover anomalies and operation condition changes that transpire in the process among the $85{,}000_{th}$ and $120{,}000_{th}$ samples. For ESP 2 case study 3, there does not seem to be any meaningful differences to process dynamics or statics that happen before the ESP fails.

ESP 3 monitoring

Results in Figure 4.12(a) and Figure 4.12(b) showcase results from the trained NS-PSFA model for ESP case study 1. From training data in Figure 4.12(a), the T_s^2 control limit for training data is 900% greater than the mean value of the T_s^2 statistic. This phenomenon can occur due to the utilization of a 99% percentile control limit. If training data is uniform, resulting control limits can pose problems in their effectiveness in online implementation. This ineffectiveness is because outliers can heavily influence the 99% control limit. A second observation from Figure 4.12(a) is monitoring results from all four monitoring statistics appear to be consistent across 120,000 training samples. As data variables change during operation, the model is most likely over-generalized from model training.

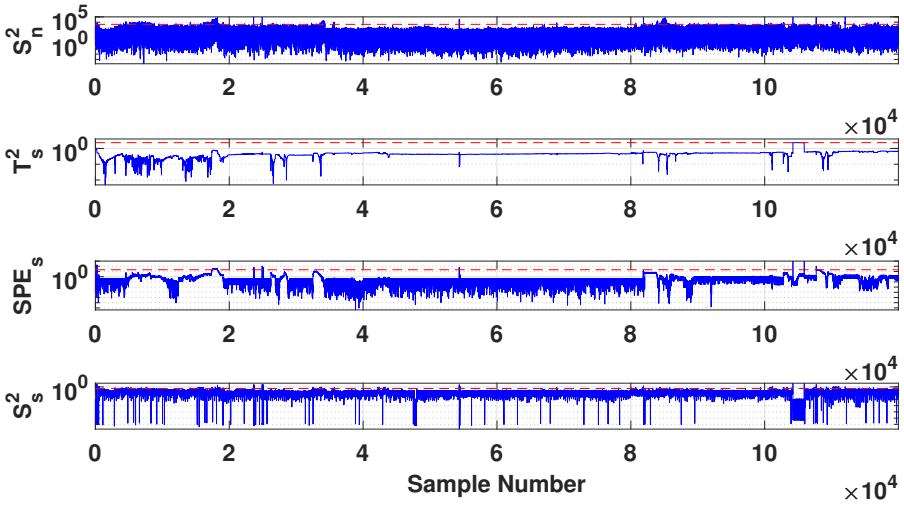


(a)

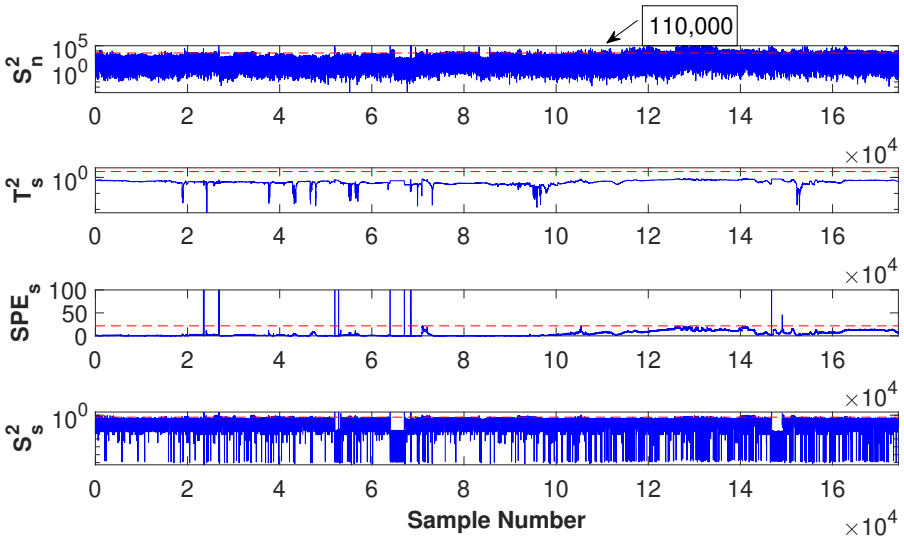


(b)

Figure 4.11: Monitoring results utilizing NS-PSFA monitoring scheme on ESP 2 case study 2-3 utilizing (a) training data, (b) test data.



(a)



(b)

Figure 4.12: Monitoring results utilizing NS-PSFA monitoring scheme on ESP 3 case study 3-1 utilizing (a) training data, (b) test data.

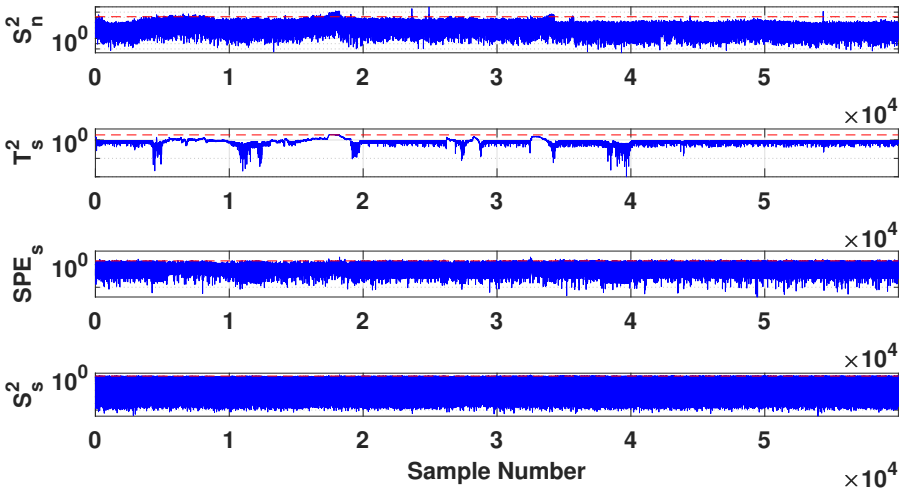
Monitoring results from Figure 4.12(b) depict comparable verdicts from training data results in Figure 4.12(a), concerning uniformity of data. From the first data sample until the final data sample, the T_s^2 and SPE_s monitoring statistics do not surpass their control limits. The S_s^2 monitoring statistic periodically exceeds its control limit, indicating there are physical anomalies. At the $110{,}000_{th}$ data sample, there is a noticeable dynamic change that occurs, affecting non-stationary process data indicated by the S_n^2 monitoring statistic. As alarms are raised for the system as anomalies occur, the monitoring scheme for the application is effective at detecting disturbances.

Results from model training and validation of ESP 3 case study 2 are seen in Figures 4.13(a) and 4.13(b). Results from Figure 4.13(a) exhibit utilizing less training data has pros and cons. In Figure 4.13(a), the monitoring statistic T_s^2 creates concerns for process monitoring, as it rests above mean values from training data. If this control limit were left as a 99% control limit, the T_s^2 monitoring statistic would not be as valuable in the derived monitoring scheme, as it could not be sensitive enough to discover disturbances and more trivial fluctuations in the process. Therefore when deployed online, the T_s^2 monitoring statistic could conceivably not obtain physical interpretations from the model.

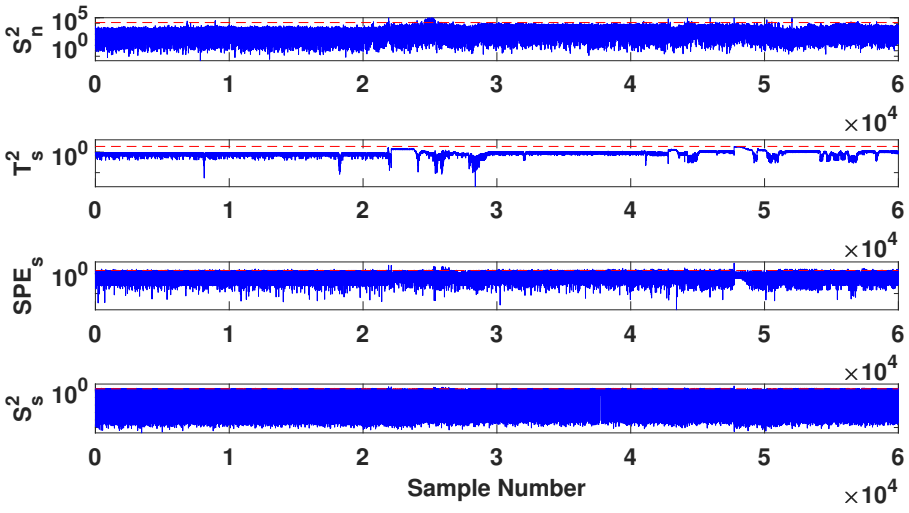
A secondary case study is presented in Figure 4.14(a) and Figure 4.14(b) that demonstrate the effects of utilizing a 95% control limit for the T_s^2 monitoring statistic, instead of utilizing a 99% control limit.

In Figure 4.14(b) results showcase utilizing the 95% control limit for T_s^2 reaps additional benefits for the monitoring scheme in this application, as it can determine whether process operations reside outside normal operating conditions. Similar to results from the previously trained model in Figure 4.13(b), having the S_n^2 monitoring statistic can provide physical benefits in monitoring by determining process anomalies.

Results of model training and online application for ESP 3 case study 3 are plotted in Figures 4.15(a) and 4.15(b). Each monitoring statistic describes operation condition changes and certain anomalies through training data. In particular, an unusual pattern is depicted from S_n^2 monitoring statistic values, as values increase following the $60{,}000_{th}$ data sample, and ultimately, the T_s^2 monitoring statistic values also increase. As T_s^2 monitoring statistic values start to rise repeatedly after the $100{,}000_{th}$

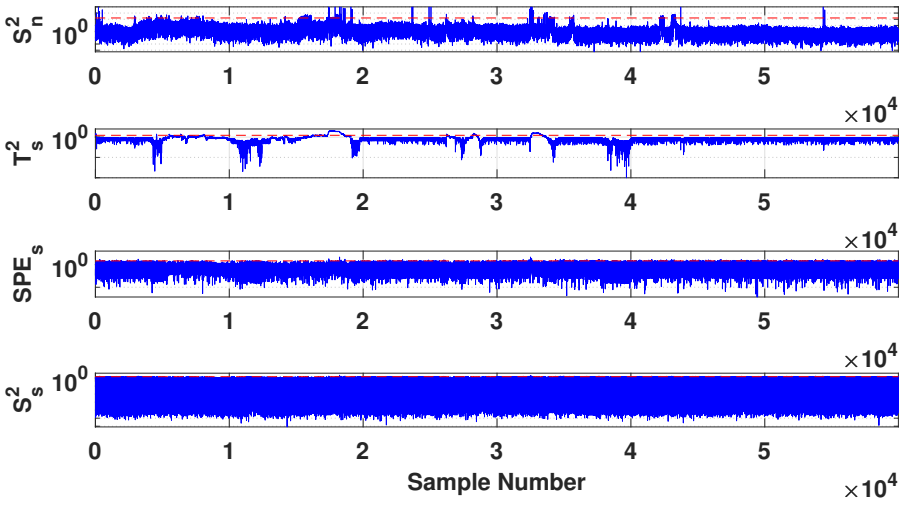


(a)

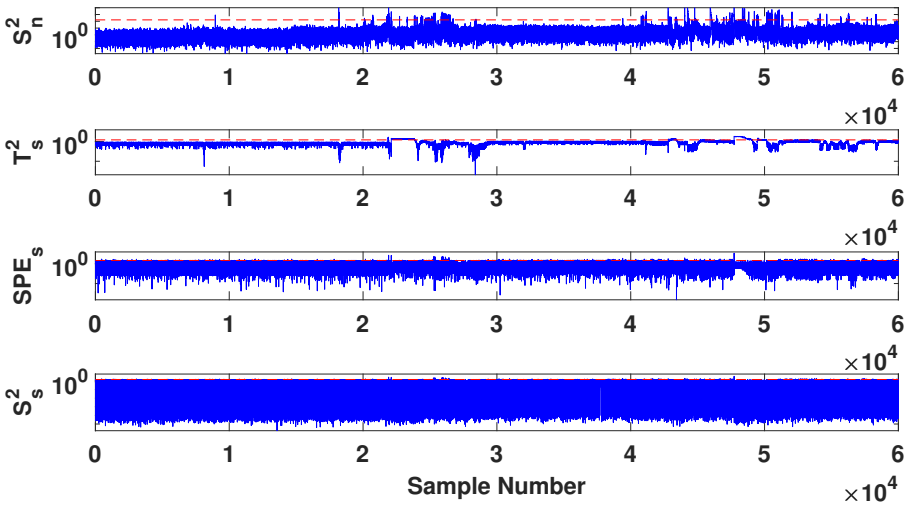


(b)

Figure 4.13: Monitoring results utilizing NS-PSFA monitoring scheme on ESP 3 case study 3-2 utilizing (a) training data, (b) validation data.

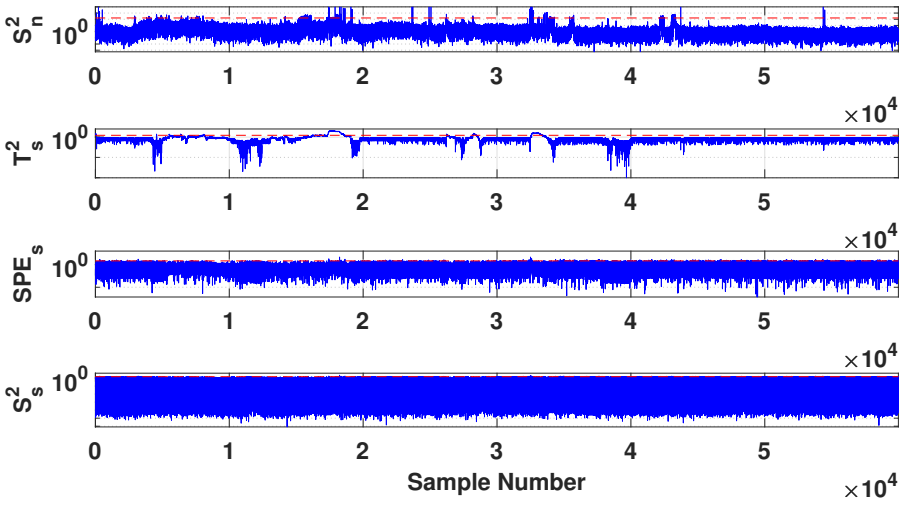


(a)

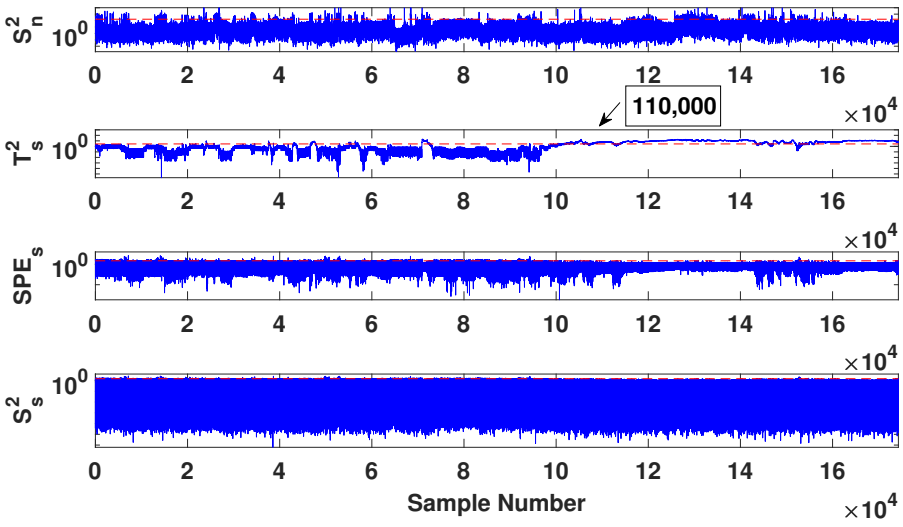


(b)

Figure 4.14: Monitoring results utilizing NS-PSFA monitoring scheme on ESP 3 case study 3-2 utilizing a 95% control limit on T_s^2 (a) training data, (b) validation data.



(a)



(b)

Figure 4.15: Monitoring results utilizing NS-PSFA monitoring scheme on ESP 3 case study 3-2 utilizing a 95% control limit on T_s^2 (a) training data, (b) test data.

data sample, S_n^2 values periodically exceed control intervals, indicating anomalies in the process. This pattern is seen throughout the test data, as the two monitoring statistics maintain a connection.

4.3 Case Study of Trained ESP NS-PSFA Model Applied on other ESP

4.3.1 Training and validating NS-PSFA model

This case study utilizes the NS-PSFA machine learning algorithm to learn parameters and train a model.

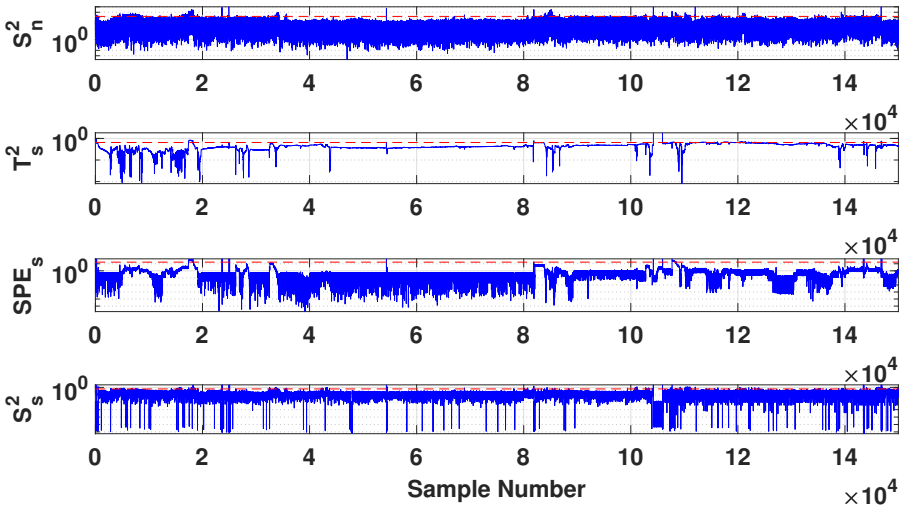


Figure 4.16: Training monitoring results from modeling ESP 3 utilizing NS-PSFA algorithm

Monitoring data from model training and parameter learning is viewed in Figure 4.16, with validation data examined in Figure 4.17. Results for training data showcase that till the 80,000$_{th}$ data sample, the operation does not undergo significant process disturbances that cause alarms, with merely small disturbances exceeding control thresholds. After the 80,000$_{th}$ data sample, control thresholds begin to be surpassed. The S_n^2 monitoring statistic overtakes the control interval from the 80,000$_{th}$ data sample, periodically, till the failure of the ESP.

Validation data displayed in Figure 4.17 showcases application results from the trained model developed and displayed in Figure 4.16. At the 40,000$_{th}$ data sample,

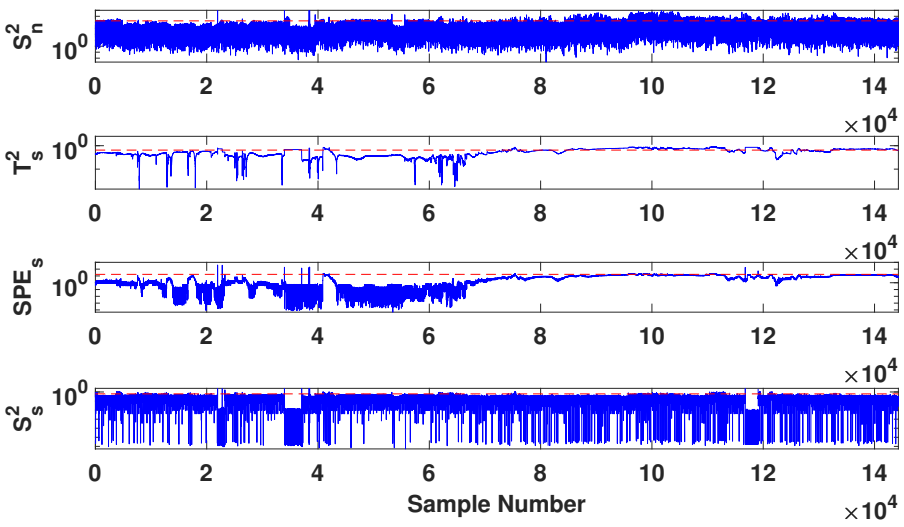


Figure 4.17: Validation monitoring results from modeling ESP 3 utilizing NS-PSFA algorithm

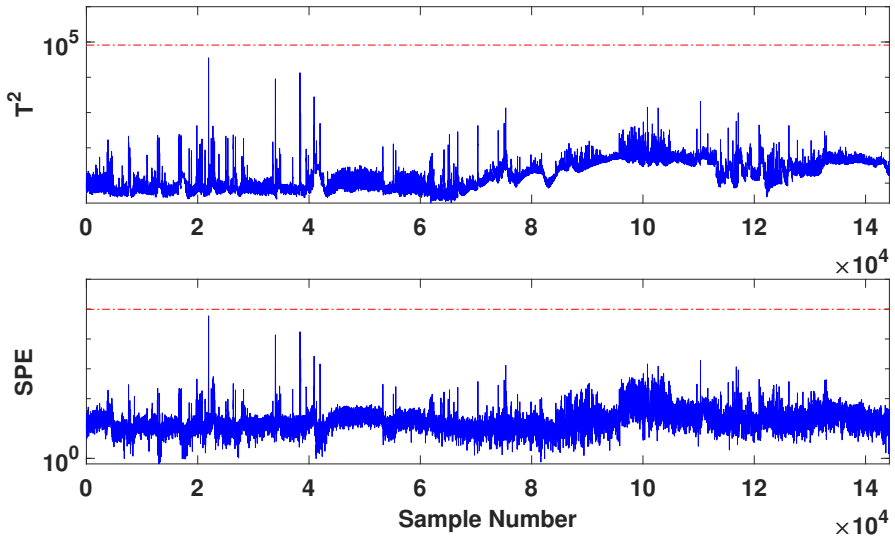


Figure 4.18: Monitoring results from modeling ESP 3 utilizing CVA algorithm

disturbances in the process begin to occur, as a significant spike occurs from the T_s^2 monitoring statistic that exceeds the threshold, thereafter the S_s^2 monitoring statistic exceeds its control intervals in conjunction from the $40{,}000_{th}$ data sample until the $55{,}000_{th}$ data sample. An interesting phenomenon that transpires is the T_s^2 monitoring statistic promptly exceeds its control limit before the S_n^2 monitoring statistic surpasses its control limit. When the T_s^2 monitoring statistic spikes and surpasses its control

limit before other monitoring statistics exceed their control limits, then an alarm could be raised, as a potential mechanism to detect faults. After the $60{,}000_{th}$ data sample, when the S_n^2 monitoring statistics goes beneath its control limit, soon after, the T_s^2 monitoring statistic rises beyond the control limit at the $74{,}000_{th}$ data sample. However, an alarm is not raised in conjunction with the S_n^2 monitoring statistic until the $85{,}000_{th}$ data sample when both statistics are beyond their control limits. Before the spike where the S_n^2 monitoring statistic exceeds its control limit, the T_s^2 values quickly decreases. A relationship between T_s^2 and S_n^2 monitoring statistics, where the T_s^2 monitoring statistic exceeds the confidence threshold before the S_n^2 monitoring statistic does is noticed. Until the process fails after the $140{,}000_{th}$ data sample, both the T_s^2 and S_n^2 monitoring statistics exceed their control thresholds, typically in a parallel fashion. While the disturbance continues to disturb process dynamics for over 60,000 samples, the process continues to deteriorate, as signified by increasing values in T_s^2 monitoring statistic. For this case study, the S_s^2 monitoring statistic does not raise alarms, as it does not detect disturbances affecting dynamics as they occur. Therefore, the NS-PSFA machine learning algorithm is capable of handling non-stationary process data and stationary process data to generate real insights into process disturbances. Due to the formulation and the combination of utilizing four monitoring statistics, the faults can be predicted as they occur from the combination of different monitoring statistics. The second goal of detecting the failure before it occurs is realized, as three monitoring statistics diminish in value before the imminent failure, while the T_s^2 monitoring statistic lingers above its control limit and proceeds to increase in value. This pattern can be continually tested.

Results for CVA process monitoring are exhibited in Figure 4.18. During cross-validation, no alarms occur based on the validation data; however, from validation data utilizing the NS-PSFA model in Figure 4.17, it is discovered that various faults occur before the ESP fails. A drawback of utilizing the CVA method is that there are no distinctions on utilizing stationary and non-stationary data. Physical insights that CVA uncovers from the process are not useful in further monitoring.

4.3.2 Application of NS-PSFA model on different ESP to detect operation condition changes and process anomalies

Monitoring results from ESP 2 are presented in Figure 4.19.

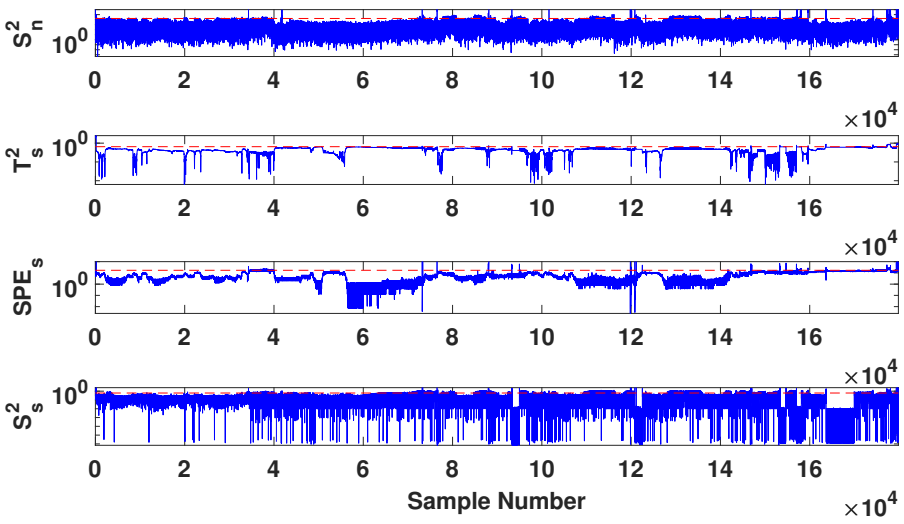


Figure 4.19: Monitoring results for applied ESP 3 NS-PSFA model on ESP 2

The first fault on ESP 2 utilizing the NS-PSFA algorithm and monitoring scheme is depicted at the $35{,}000_{th}$ data sample, when the S_n^2 and S_s^2 monitoring statistics pass their control limits, depicting an anomaly in conjunction with the SPE_s monitoring statistic. In the test application on ESP 2, the T_s^2 monitoring statistic is not perceived to be an important preliminary indicator of operation condition changes. After the $60{,}000_{th}$ data sample, stationary dynamics are affected as implied by the S_s^2 monitoring statistic exceeding the control threshold. This phenomenon occurs during the process, with the S_n^2 and S_s^2, and monitoring statistics indicating significant faults from the $60{,}000_{th}$ to $95{,}000_{th}$ data samples. Through the period of the $60{,}000_{th}$ data sample to the $95{,}000_{th}$ data sample, the T_s^2 monitoring statistic does not exceed the control limit. As the T_s^2 monitoring statistic stays underneath its control limit while the dynamic monitoring statistics stay above their control limits, dynamic monitoring statistics are confirmed for detecting process anomalies. After that, at the $140{,}000_{th}$ data sample, the SPE_s monitoring statistic starts to rise sharply in value. However, the T_s^2 monitoring statistic does not escalate in value. It is remarked that the mon-

itoring statistics S_n^2, and S_s^2 go beneath their control thresholds significantly at the $160{,}000_{th}$ data sample, whereas, the T_s^2 and SPE_s monitoring statistics proceed to rise and approach their highest values. As the phenomenon that before the failure of an ESP, the T_s^2 and SPE_s monitoring statistics continue to rise in value, whereas, other monitoring statistics decrease significantly in value, and an additional alarm could be triggered on a ratio of the T_s^2 and SPE_s monitoring statistic values with S_s^2 and S_n^2 monitoring statistic values.

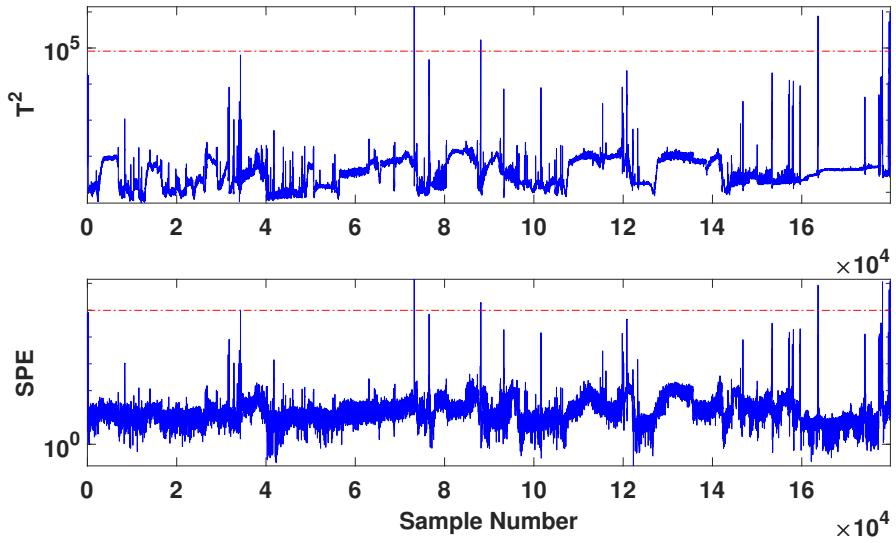


Figure 4.20: Monitoring results for applied ESP 3 CVA model on ESP 2

Figure 4.20 showcases the results of the trained model from ESP 3 data applied to test data on ESP 2. Comparable to results in Figure 4.18, where CVA was applied on validation data from ESP 3, alarms are not frequently raised for the CVA algorithm in comparison to the NS-PSFA algorithm as seen in Figure 4.19. Additionally, alarms that are established by the CVA method are raised spontaneously and are not sustained, which is not a consistent finding from what is determined by the NS-PSFA monitoring scheme. As the CVA method is incapable of determining faults as they occur, this approach cannot accomplish the first goal determined for this case study, whereas the NS-PSFA algorithm can discover faults as they transpire. From the results presented in Figure 4.20, CVA does not identify any problems with the operation before the impending failure; therefore, this issue further negatively affects its monitoring abilities of non-stationary data.

The NS-PSFA method showcases it can be implemented in the ESP application to learn parameters and develop models. Afterward, trained models are capable of achieving the two goals of detecting faults as they occur and generating alarms for the process, as well, ascertaining a mechanism that could detect impending failures before they occur.

Chapter 5

Thesis summary

5.1 Conclusion

In this thesis, a novel NS-PSFA methodology is proposed to model and monitor industrial processes with non-stationary and stationary characteristics. Its unique advantages exist in the aptitude of succinctly modeling and monitoring various operation situations intertwined with both non-stationary and stationary variations, which cannot be handled by conventional monitoring methods. A probabilistic framework formulates the proposed method, which allows for efficient parameter estimation based on the EM algorithm, as well as convenient online feature extraction based on the Kalman filter. Four process monitoring statistics are developed and utilized for monitoring, which incorporates distinct physical interpretations in terms of non-stationary and stationary dynamics anomalies, as well as steady-state deviations. A simulated CSTR tests the abilities of the proposed method and trained models are applied in other industrial processes to demonstrate the applicability of the research with real-world applications. These applications include a non-stationary industrial chemical process and an electrical submersible pump application. The proposed method generalizes satisfactorily for process data that involve changing operating conditions and non-stationary variations and can provide comprehensive information about process status to assist the decision-making of process practitioners better. For each case study, the proposed NS-PSFA algorithm showcases that it can extract physical insights from both non-stationary and stationary data for model training. The various case studies show that the four monitoring statistics can represent operation changes and dynamic disturbances effectively.

5.2 Future Work

Various extensions of the NS-PSFA algorithm are possible. A first one would be to include a case for missing data and using approximations on the distributions utilized. A secondary suggestion is to create an initialization strategy for the non-stationary section that maximizes the log-likelihood value for parameter estimation in the EM algorithm. A third suggestion is to develop a cross-validation strategy that can perform different initializations cohesively, utilizing various values of SFs, speeding up the training and validation process to maximize the parameter estimations.

Bibliography

[1] Chao Shang. *Dynamic Modeling of Complex Industrial Processes: Data-driven Methods and Application Research*. Springer, 2018.

[2] John MacGregor and Ali Cinar. Monitoring, fault diagnosis, fault-tolerant control and optimization: Data driven methods. *Computers & Chemical Engineering*, 47:111–120, 2012.

[3] Leo H Chiang, Evan L Russell, and Richard D Braatz. Fault diagnosis in chemical processes using fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemometrics and intelligent laboratory systems*, 50(2):243–252, 2000.

[4] Jong-Min Lee, S Joe Qin, and In-Beum Lee. Fault detection and diagnosis based on modified independent component analysis. *AIChE journal*, 52(10):3501–3514, 2006.

[5] Eliana Zamprogna, Massimiliano Barolo, and Dale E Seborg. Estimating product composition profiles in batch distillation via partial least squares regression. *Control Engineering Practice*, 12(7):917–929, 2004.

[6] Bhupinder S Dayal and John F MacGregor. Recursive exponentially weighted pls and its applications to adaptive control and prediction. *Journal of Process Control*, 7(3):169–179, 1997.

[7] Dongsoon Kim and In-Beum Lee. Process monitoring based on probabilistic pca. *Chemometrics and intelligent laboratory systems*, 67(2):109–123, 2003.

[8] Jinlin Zhu, Zhiqiang Ge, and Zhihuan Song. Non-gaussian industrial process monitoring with probabilistic independent component analysis. *IEEE Transactions on Automation Science and Engineering*, 14(2):1309–1319, 2016.

[9] Chao Shang, Fan Yang, Xinqing Gao, Xiaolin Huang, Johan AK Suykens, and Dexian Huang. Concurrent monitoring of operating condition deviations and process dynamics anomalies with slow feature analysis. *AIChE Journal*, 61(11):3666–3682, 2015.

[10] Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.

[11] Henning Sprekeler, Tiziano Zito, and Laurenz Wiskott. An extension of slow feature analysis for nonlinear blind source separation. *The Journal of Machine Learning Research*, 15(1):921–947, 2014.

[12] Chunhui Zhao and Biao Huang. A full-condition monitoring method for nonstationary dynamic chemical processes with cointegration and slow feature analysis. *AIChE Journal*, 64(5):1662–1681, 2018.

[13] Lei Fan, Hariprasad Kodamana, and Biao Huang. Semi-supervised dynamic latent variable modeling: I/o probabilistic slow feature analysis approach. *AIChE Journal*, 65(3):964–979, 2019.

[14] Chao Shang, Biao Huang, Fan Yang, and Dexian Huang. Slow feature analysis for monitoring and diagnosis of control performance. *Journal of Process Control*, 39:21–34, 2016.

[15] Feihong Guo, Chao Shang, Biao Huang, Kangcheng Wang, Fan Yang, and Dexian Huang. Monitoring of operating point and process dynamics via probabilistic slow feature analysis. *Chemometrics and Intelligent Laboratory Systems*, 151:115–125, 2016.

[16] Peter J Brockwell, Richard A Davis, and Matthew V Calder. *Introduction to time series and forecasting*, volume 2. Springer, 2002.

[17] Gang Li, S Joe Qin, and Tao Yuan. Nonstationarity and cointegration tests for fault detection of dynamic processes. *IFAC Proceedings Volumes*, 47(3):10616–10621, 2014.

[18] He Sun, Shumei Zhang, Chunhui Zhao, and Furong Gao. A sparse reconstruction strategy for online fault diagnosis in nonstationary processes with no a priori fault information. *Industrial & Engineering Chemistry Research*, 56(24):6993–7008, 2017.

[19] Chunhui Zhao, Fuli Wang, Furong Gao, and Yingwei Zhang. Enhanced process comprehension and statistical analysis for slow-varying batch processes. *Industrial & Engineering Chemistry Research*, 47(24):9996–10008, 2008.

[20] S Joe Qin. Recursive pls algorithms for adaptive data modeling. *Computers & Chemical Engineering*, 22(4-5):503–514, 1998.

[21] Shumei Zhang, Chunhui Zhao, Shu Wang, and Fuli Wang. Pseudo time-slice construction using a variable moving window k nearest neighbor rule for sequential uneven phase division and batch process monitoring. *Industrial & Engineering Chemistry Research*, 56(3):728–740, 2017.

[22] Chao Shang, Fan Yang, Biao Huang, and Dexian Huang. Recursive slow feature analysis for adaptive monitoring of industrial processes. *IEEE Transactions on Industrial Electronics*, 65(11):8895–8905, 2018.

[23] Hiromasa Kaneko and Kimito Funatsu. Estimation of predictive accuracy of soft sensor models based on data density. *Chemometrics and Intelligent Laboratory Systems*, 128:111–117, 2013.

[24] Manabu Kano and Yoshiaki Nakagawa. Data-based process monitoring, process control, and quality improvement: Recent developments and applications in steel industry. *Computers & Chemical Engineering*, 32(1-2):12–24, 2008.

[25] Chao Shang, Biao Huang, Fan Yang, and Dexian Huang. Probabilistic slow feature analysis-based representation learning from massive process data for soft sensor modeling. *AIChE Journal*, 61(12):4126–4139, 2015.

[26] Petr Kadlec, Bogdan Gabrys, and Sibylle Strandt. Data-driven soft sensors in the process industry. *Computers & chemical engineering*, 33(4):795–814, 2009.

[27] S Joe Qin. Process data analytics in the era of big data. *AIChE Journal*, 60(9):3092–3100, 2014.

[28] Chao Shang, Fan Yang, Dexian Huang, and Wenxiang Lyu. Data-driven soft sensor development based on deep learning technique. *Journal of Process Control*, 24(3):223–233, 2014.

[29] Paolo Giudici, Peter Sarlin, and Alessandro Spelta. The interconnected nature of financial systems: direct and common exposures. *Journal of Banking & Finance*, 2017.

[30] Vinicius M Marques, Celso J Munaro, and Sirish L Shah. Detection of causal relationships based on residual analysis. *IEEE Transactions on Automation Science and Engineering*, 12(4):1525–1534, 2015.

[31] Zhiqiang Ge, Biao Huang, and Zhihuan Song. Nonlinear semisupervised principal component regression for soft sensor modeling and its mixture form. *Journal of Chemometrics*, 28(11):793–804, 2014.

[32] Zhiqiang Ge, Biao Huang, and Zhihuan Song. Mixture semisupervised principal component regression model and soft sensor application. *AIChE Journal*, 60(2):533–545, 2014.

[33] Said el Bouhaddani, Hae-Won Uh, Caroline Hayward, Geurt Jongbloed, and Jeanine Houwing-Duistermaat. Probabilistic partial least squares model: Identifiability, estimation and application. *Journal of Multivariate Analysis*, 2018.

[34] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2009.

[35] R Sathya and Annamma Abraham. Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 2(2):34–38, 2013.

[36] Kunhuang Huarng and Tiffany Hui-Kuang Yu. The application of neural networks to forecast fuzzy time series. *Physica A: Statistical Mechanics and its Applications*, 363(2):481–491, 2006.

[37] Y Bengio, A Courville, and P Vincent. Representation learning: a review and new perspectives. arxiv. org. 2012.

[38] Konrad P Körding, Christoph Kayser, Wolfgang Einhäuser, and Peter König. How are complex cell properties adapted to the statistics of natural stimuli? *Journal of neurophysiology*, 91(1):206–212, 2004.

[39] Mathias Franzius, Niko Wilbert, and Laurenz Wiskott. Invariant object recognition with slow feature analysis. In *International Conference on Artificial Neural Networks*, pages 961–970. Springer, 2008.

[40] Chen Wu, Bo Du, and Liangpei Zhang. Slow feature analysis for change detection in multispectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5):2858–2874, 2014.

[41] Tobias Blaschke, Tiziano Zito, and Laurenz Wiskott. Independent slow feature analysis and nonlinear blind source separation. *Neural computation*, 19(4):994–1021, 2007.

[42] Angelika Bunse-Gerstner. An algorithm for the symmetric generalized eigenvalue problem. *Linear Algebra and its Applications*, 58:43–68, 1984.

[43] Richard Turner and Maneesh Sahani. A maximum-likelihood interpretation for slow feature analysis. *Neural computation*, 19(4):1022–1038, 2007.

[44] Sheldon M Ross, John J Kelly, Roger J Sullivan, William James Perry, Donald Mercer, Ruth M Davis, Thomas Dell Washburn, Earl V Sager, Joseph B Boyce, and Vincent L Bristow. *Stochastic Processes*, volume 2. Wiley New York, 1996.

[45] David A Dickey and Wayne A Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366a):427–431, 1979.

[46] Nima Sammaknejad, Yujia Zhao, and Biao Huang. A review of the expectation maximization algorithm in data-driven process identification. *Journal of Process Control*, 73:123–136, 2019.

[47] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[48] Vassilios Digalakis, Jan Robin Rohlicek, and Mari Ostendorf. Ml estimation of a stochastic linear system with the em algorithm and its application to speech recognition. *IEEE Transactions on speech and audio processing*, 1(4):431–442, 1993.

[49] Zoubin Ghahramani and Geoffrey E Hinton. Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science, 1996.

[50] Qiaojun Wen, Zhiqiang Ge, and Zhihuan Song. Data-based linear gaussian state-space model for dynamic process monitoring. *AIChE Journal*, 58(12):3763–3776, 2012.

[51] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[52] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[53] Le Zhou, Gang Li, Zhihuan Song, and S Joe Qin. Autoregressive dynamic latent variable models for process monitoring. *IEEE Transactions on Control Systems Technology*, 25(1):366–373, 2016.

[54] Gang Li, S Joe Qin, Yindong Ji, and Donghua Zhou. Reconstruction based fault prognosis for continuous processes. *Control Engineering Practice*, 18(10):1211–1219, 2010.

[55] Zhiwen Chen, Kai Zhang, Steven X Ding, Yuri AW Shardt, and Zhikun Hu. Improved canonical correlation analysis-based fault detection methods for industrial processes. *Journal of Process Control*, 41:26–34, 2016.

[56] Cristobal Ruiz-Cárcel, Yi Cao, D Mba, Liyun Lao, and RT Samuel. Statistical process monitoring of a multiphase flow facility. *Control Engineering Practice*, 42:74–88, 2015.

[57] Shen Yin, Steven X Ding, Adel Haghani, Haiyang Hao, and Ping Zhang. 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, 2012.

[58] Lawrence AP Camilleri, John Macdonald, et al. How 24/7 real-time surveillance increases esp run life and uptime. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2010.

[59] Supriya Gupta, Luigi Saputelli, Michael Nikolaou, et al. Applying big data analytics to detect, diagnose, and prevent impending failures in electric submersible pumps. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2016.

[60] Zhizhuang Jiang, Bassam Zreik, et al. Esp operation, optimization, and performance review: Conocophillips china inc. bohai bay project. 2007.

[61] Joe Vandevier. Run-time analysis assesses pump performance. *Oil & Gas Journal*, 108(37):76–79, 2010.

[62] Bertrand C Theuveny, Alex Kosmala, Charles Cosad, Francisco Pulido Fernandez, Patrick Pierre Destarac, et al. The challenge of federation of information for automated surveillance of esps: Field examples. In *SPE Latin American and Caribbean Petroleum Engineering Conference*. Society of Petroleum Engineers, 2005.