Dynamic Latent Structured Data Analytics for Process Modeling and Monitoring

by

Bo Xu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Chemical Engineering

Waterloo, Ontario, Canada, 2023

© Bo Xu 2023

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Bin Liu Associate Professor, Dept. of Manage. Sci., University of Strathclyde
Supervisor(s):	Hector Budman Professor, Dept. of Chem. Eng., University of Waterloo Qinqin Zhu Assistant Professor, Dept. of Chem. Eng., University of Waterloo
Internal-External:	Zhao Pan Assistant Professor, Dept. of Mech. Eng., University of Waterloo
Internal Member:	Mario Ioannidis Professor, Dept. of Chem. Eng., University of Waterloo
Internal Member(s):	Nasser Mohieddin Abukhdeir Associate Professor, Dept. of Chem. Eng., University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Chapter 3 has been published as a paper in the journal "Industrial & Engineering Chemistry Research". The author developed and implemented the documented methodologies, obtained the numerical results, and completed the writing of the article. Qinqin Zhu was involved in the conceptualization, supervision, and writing of the work. All authors read and approved the final manuscript.

Chapter 4 has been published as a paper in the journal "Journal of Process Control". The author developed and implemented the documented methodologies, obtained the numerical results, and completed the writing of the paper. Qinqin Zhu was involved in the conceptualization, supervision, and writing of the work. All authors read and approved the final manuscript.

Chapter 5 has been published as a paper in the journal "IEEE Transactions on Automation Science and Engineering". The author implemented the documented methodologies, obtained the numerical results, and completed the writing of the article. Qinqin Zhu was involved in the supervision, conceptualization, and writing of the work. All authors read and approved the final manuscript.

Chapter 6 is based on a paper that has been submitted and is currently under review. The author developed and implemented the documented methodologies, obtained the numerical results, and completed the writing of the paper. Qinqin Zhu was involved in the supervision, conceptualization, and writing of the work.

Abstract

With the advent of Industry 4.0, variable relations in modern industrial processes are increasingly complex due to their high dimensions and complex auto-correlations and cross-correlations. Multivariate statistical methods such as principal component analysis (PCA), partial least squares (PLS), slow feature analysis, and canonical correlation analysis (CCA) are widely employed to exploit variable relations for process data analytics, process monitoring, and fault diagnosis. However, it is not appropriate to directly apply these multivariate statistical methods to cross-correlated data from a dynamic process, since these methods are only able to model static components of the data. Subsequently, numerous adaptations of multivariate statistical methods have been proposed to address the intricate interdependencies inherent in dynamic process data. Among these, dynamic inner methods have emerged as the most advanced and efficient approaches in recent years. However, it is pertinent to acknowledge that these dynamic inner methods still exhibit certain limitations owing to the intricate properties of time series data.

In this research, several novel algorithms are proposed to address the aforementioned issues.

First, basis functions are adapted into dynamic inner PLS, referred to as dynamic weighted PLS (DWPLS), to deal with redundant information in neighboring samples caused by high sampling rates in industrial processes. Nowadays, however, because of the improvement of sensory technologies, redundant information exists among adjacent samples with high sampling rates. The existing dynamic algorithms ignore the redundant information, causing suboptimal modeling and monitoring schemes. In this paper, a dynamic weighted PLS (DWPLS) method is proposed to deal with the aforementioned issue. DWPLS maximizes the covariance between the quality latent scores and a weighted representation of lagged process scores. The relations of lagged process scores are learned through a weighted summation of basis functions. The monitoring scheme of DWPLS is also constructed. Case studies with two processes, the Tennessee Eastman process, and a distillation process, are conducted to show the advantages of DWPLS over existing methods in terms of regression and fault detection performance.

Further, to fully exploit quality information, a novel dynamic auto-regressive latent variable model (DALVM) is proposed to capture both auto- and cross-correlations simultaneously from high-dimensional time series data. Some dynamic supervised learning algorithms are designed to extract their dynamic cross-correlations, but auto-correlations among quality variables are rarely considered, which, however, can provide additional valuable information. This article proposes a novel dynamic auto-regressive latent variable model (DALVM) to capture both auto and cross-correlations from high-dimensional time series data. DALVM is designed to maximize the covariance between the current quality score and the weighted sum of past quality and process scores, and an auto-regressive exogenous inner model is developed for consistency purposes. Further, a concurrent anomaly detection system is developed based on DALVM, referred to as ConDALVM, which conducts subsequent decompositions in the extracted latent spaces. ConDALVM realizes comprehensive monitoring for both static and dynamic anomalies in process and quality spaces.

The superiority of the methods is demonstrated through a numerical simulation and two industrial processes.

Then to continue to improve the flexibility and performance of DALVM, an efficient dynamic auto-regressive CCA (EDACCA) is also developed, with an emphasis on missing data imputation. an efficient dynamic auto-regressive canonical correlation analysis (EDACCA) method is proposed with a modified auto-regressive exogenous model to extract dynamics in both auto-correlations and cross-correlations. The flexibility and efficiency of EDACCA are improved with the design of weighting parameters and the economic singular value decomposition. EDACCA is further adapted for multi-step ahead (MS) prediction and missing data imputation. Two industrial processes are employed to evaluate the prediction performance and imputation performance of EDACCA.

After that, the exponentially weighted moving average model is adapted into dynamic SFA to capture dynamic relations among data with great interpretability. Slow variations of process variables retain normal operating conditions and important patterns of manufacturing processes, and slow feature analysis (SFA) is widely adopted to study slowly varying signals. However, SFA only extracts first-order dynamics, and most of its multi-step dynamic extensions focus on predictability instead of slowness. Then, a novel exponentially weighted SFA (EWSFA) is proposed to extract slow features by following the decaying nature of temporal signals. Exponentially decaying weights are learned in EWSFA to model dynamic relations between current and historical samples, and its computation efficiency is further improved with the aid of singular value decomposition. An EWSFA-based monitoring scheme is developed to detect anomalies in both slow and fast variations, and a novel dynamic index is designed to identify the occurrence of operating condition changes. The effectiveness of EWSFA is demonstrated through two industrial case studies.

Finally, for future works, the power of neural networks will be effectively incorporated into the design of dynamic algorithms to deal with complex nonlinearities.

Acknowledgements

First of all, I would like to thank my thesis supervisor Professor Hector Budman and Professor Zhao Pan. Cordial thanks for their invaluable support during this period. Without their assistance, I would have been at a loss on how to navigate through it. Their guidance and help have been absolutely pivotal to my successful doctoral graduation. I would also like to thank my uncle Professor Zhiwen Qi who provided me with very good guidance for how to be a good researcher.

During my four-year graduate studies and life, I received lots of useful suggestions, friendship, love, and help. Here I can only name a few. I would like to thank Xin Shen and Shuji Chang.

Finally, I would like to thank my father Zhengxian Xu, my mother Zaoxia Qi, and my girlfriend Qiyue Wang, their moral support really help me a lot. I really miss them!

Dedication

I dedicate this thesis to my parents and my girlfriend for their support and love. I love you all dearly.

Table of Contents

Examining Committee Membership ii

Author's Declaration iii

Statement of Contributions iv

Abstract v

Acknowledgements vii

Dedication viii

List of Figures xiii

List of Tables xv

1 Introduction 1

1.1 Unsupervised Dynamic Latent Variable Analytics 5

1.2 Supervised Dynamic Latent Variable Methods 7

2 Background and Literature Review 10

2.1 Supervised static Learning Methods 10

2.1.1 Partial Least Squares 10

2.1.2 Canonical Correlation Analysis 11

2.2 Supervised Dynamic Latent Variable Models 12

2.2.1 Dynamic Inner PLS 12

2.2.2 Supervised Dynamic inner CCA 13

2.3 Unsupervised static Learning Methods 13

2.3.1 Principal Component Analysis . . . 14
2.3.2 Slow Feature Analysis . . . 14
2.4 Unsupervised Dynamic Latent Variable Models . . . 15
2.4.1 Dyanmic Inner PCA . . . 15
2.4.2 Unsupervised Dynamic Inner CCA . . . 16
2.4.3 Dynamic Inner Slow Feature Analysis . . . 17
2.5 Nonlinearity Handling . . . 17

3 Dynamic Weighted Partial Least Squares 19
3.1 Overview . . . 19
3.2 Introduction . . . 19
3.3 Prelimilary . . . 21
3.3.1 Static Partial Least Squares . . . 21
3.3.2 Dynamic Inner Partial Least Squares . . . 22
3.4 Dynamic Weighted Partial Least Squares . . . 23
3.4.1 Basis Functions . . . 23
3.5 The DWPLS Algorithm . . . 28
3.6 Process and Quality Monitoring with DWPLS . . . 31
3.7 Case Studies . . . 32
3.7.1 Case Study on TEP . . . 32
3.7.2 Case Study on A Distillation Process . . . 38
3.8 Summary . . . 39

4 Dynamic Auto-Regressive Latent Variable Models 40
4.1 Overview . . . 40
4.2 Introduction . . . 40
4.3 Preliminaries . . . 42
4.3.1 Partial Least Squares . . . 42
4.3.2 Dynamic-Inner Partial Least Squares . . . 43
4.3.3 DiPLS-based Monitoring . . . 44
4.4 Methodology . . . 44
4.4.1 Problem Statement and Motivation . . . 44
4.4.2 Dynamic Auto-Regressive Latent Variable Model . . . 46

4.4.3 Comprehensive Online Anomaly Detection . . . 48
4.5 Experiments and Discussion . . . 52
4.5.1 Toy example . . . 53
4.5.2 Tennessee Eastman Process . . . 55
4.5.3 Three-Phase Flow Facility . . . 59
4.6 Conclusion . . . 63

5 An Efficient Dynamic Auto-Regressive CCA for Time Series Imputation with Irregular Sampling 65
5.1 Overview . . . 65
5.2 Introduction . . . 65
5.3 Preliminaries . . . 68
5.3.1 Canonical Correlation Analysis . . . 68
5.3.2 Supervised Dynamic inner CCA . . . 68
5.4 Methodology . . . 70
5.4.1 Traditional Dynamic Auto-Regressive CCA . . . 70
5.4.2 Efficient Dynamic Auto-Regressive CCA . . . 71
5.4.3 Multi-Step-Ahead Prediction with EDACCA . . . 74
5.4.4 Missing Data Imputation with EDACCA-based MS Prediction . . . 76
5.5 Case Studies . . . 77
5.5.1 Tennessee Eastman Process . . . 77
5.5.2 Wind Process . . . 79
5.6 Summary . . . 82

6 Multi-Step Dynamic Slow Feature Analysis for Anomaly Detection 86
6.1 Overview . . . 86
6.2 Introduction . . . 86
6.3 Preliminaries . . . 88
6.3.1 Slow Feature Analysis . . . 88
6.3.2 Dynamic Inner Slow Feature Analysis . . . 89
6.4 Methodology . . . 90
6.4.1 Problem Statement and Motivation . . . 91
6.4.2 Optimization Problem of EWSFA . . . 93

6.4.3 The EWSFA Algorithm . . . 94
6.4.4 Efficient EWSFA . . . 95
6.4.5 EWSFA-based Monitoring Scheme . . . 96
6.5 Case Studies . . . 99
6.5.1 Eastman Chemical Company . . . 99
6.5.2 Three-Phase Flow Facility . . . 103
6.6 Summary . . . 108

7 Conclusions and Futhure Work **111**
7.1 Conclusions . . . 111
7.1.1 Methods to Solve the Redundant Information . . . 111
7.1.2 Methods to Fully Exploit Quality Information . . . 113
7.1.3 Development of Interpretability for Time Series . . . 118
7.1.4 Suitability of Each Method . . . 120
7.2 Future Work . . . 120

References **122**

List of Figures

3.1 An Example of Bézier Series 26
3.2 Selected Series and Weight Variations of the First Two Latent Variables for TEP 34
3.3 Prediction Performance of DWPLS_{PB} for TEP 35
3.4 Fault Detection Results with DWPLS_{PB} 37
3.5 Block Diagram of the Distillation Process 38
3.6 Prediction Performance of DWPLS_{PB} for the Distillation Process 39

4.1 Modeling Performance of DALVM for Numerical Study 54
4.2 Modeling Performance of DiPLS for Numerical Study 54
4.3 Modeling Performance of PLS for Numerical Study 55
4.4 Auto-Correlations and Cross-Correlations of Quality Residuals for DALVM 56
4.5 Auto-Correlations and Cross-Correlations of Quality Residuals for DiPLS . 56
4.6 Dynamic Modeling Performance for TEP 57
4.7 Modeling Performance of $\hat{\mathbf{Y}}_{xg}$ and $\hat{\mathbf{Y}}_{yg}$ for TEP 58
4.8 PCA-based Process and Quality Monitoring for IDV(4) 60
4.9 Monitoring Performance of ConDALVM for IDV(4) 60
4.10 PCA-based Process and Quality Monitoring for IDV(5) 61
4.11 Monitoring Performance of ConDALVM for IDV(5) 61
4.12 PCA-based Process and Quality Monitoring for F1_T2 63
4.13 Monitoring Performance of ConDALVM for F1_T2 63
4.14 Monitoring Performance of DiPLS for F1_T2 64

5.1 Dynamic Modeling Performance of Different Models for TEP 77
5.2 Dynamic Cross-Correlations ($\mathbf{Y}_{xg}$) and Auto-Correlations ($\mathbf{Y}_{yg}$) of EDACCA, DACCA and DAPLS 83
5.3 Computing Time of EDACCA, DACCA and DAPLS for TEP (Semi-log Unit) 84

5.4 Dynamic Modeling Performance of Different Models for Wind Process . . . 84
5.5 t-SNE Visualization for EDACCA and LR Imputation Methods 85

6.1 Input Signals with Dynamics Order $d = 2$ 92
6.2 SFA LVs and Their FFT 92
6.3 The Flowchart of EWSFA-based Monitoring Scheme 97
6.4 The Process Diagram of Eastman Chemical Company Plant 99
6.5 Slow Features Extracted by DiSFA and EWSFA 100
6.6 Correlation Coefficients of SFs of DiSFA and EWSFA 101
6.7 FFT for the First Four SFs of EWSFA (log scale) 102
6.8 FFT for the First Four SFs of DiSFA (log scale) 102
6.9 The Process Diagram of Three-Phase Flow Facility Plant 103
6.10 Details of Some Important Variables in Fault 1.2 104
6.11 EWSFA-based Monitoring Results for Fault 1.2 (semi-log scale) 105
6.12 DiSFA-based Monitoring Results for Fault 1.2 (semi-log scale) 106
6.13 DiPCA-based Monitoring Results for Fault 1.2 (semi-log scale) 107
6.14 Variable Flow Rate Top Separator Output in Fault 3.3 107
6.15 EWSFA-based Monitoring Results for Fault 3.3 (semi-log form) 108
6.16 DiSFA-based Monitoring Results for Fault 3.3 (semi-log form) 109
6.17 DiPCA-based Monitoring Results for Fault 3.3 (semi-log form) 110

List of Tables

3.1 Selected Model Parameters for TEP 33
3.2 MSEs of Different Models for TEP 33
3.3 The Effect of Parameter Points of DWPLS_{PB} on Prediction Performance . 35
3.4 The Effect of Control Points of DWPLS_{BB} on Prediction Performance . . . 36
3.5 FDRs and FARs for Quality-Relevant Faults with DiPLS and DWPLS_{PB} . 36
3.6 FARs for Quality-Irrelevant Disturbances with DiPLS and DWPLS_{PB} . . . 36
3.7 MSEs of Different Models for the Distillation Process 38

4.1 Description of Subspace Abbreviations 51
4.2 Fault Detection Indices for ConDALVM 51
4.3 MSEs of DALVM, DiPLS and PLS for Numerical Study 53
4.4 MSEs of DALVM, DiPLS and PLS for TEP 57
4.5 MSEs of DALVM, DiPLS and PLS for TEP 58
4.6 FDRs and FARs for Quality-Relevant Faults (%) 59
4.7 FARs for Quality-Irrelevant Disturbances (%) 59
4.8 FDRs and FARs for Quality-Relevant Faults (%) 62
4.9 FARs for Quality-Irrelevant Disturbances (%) 62

5.1 Prediction Results of Different Models for TEP 78
5.2 MSEs of $\hat{\mathbf{Y}}_{xg}$ and $\hat{\mathbf{Y}}_{yg}$ for TEP 78
5.3 Prediction Results of Different Models for Wind Process 79
5.4 MSEs of Multi-Step Ahead Prediction Models 80
5.5 MSEs of Different Imputation Methods for MCAR 80
5.6 Parameter Selection for MNAR 81
5.7 MSEs of Different Imputation Methods for MNAR 81

6.1 R^2 of SFs of DiSFA and EWSFA 99

Chapter 1

Introduction

The advent of the Internet of Things (IoT) has fostered the emergence of intelligent and wireless sensors, alongside the rapid expansion of 5G networks, mobile devices and smart devices for process monitoring. Consequently, there has been an exponential increase in the volume of data being collected and stored. Simultaneously, advancements in data analytics have provided increasingly powerful tools for extracting valuable insights from this vast array of information. As a result, numerous industries, such as chemicals, pharmaceuticals, petrochemicals, microelectronics, energy, power grids, and transportation, have expanded their efforts toward deriving systematic understanding and useful insights from diverse data sources [164, 119, 107, 215, 120]. However, the surging complexity and scale of data present significant challenges which require new scientific tools for data analysis. The profound development in data science is a direct response to the need for effective techniques and methodologies to address the complexities posed by the diverse and massive datasets encountered in various industries.

Operational data obtained from processes and manufacturing systems often possess high-dimensional characteristics, exhibiting dynamic features and encompassing multiple sampling rates. Furthermore, these datasets encompass a mixture of continuous and categorical quantities. Industrial processes and equipment are meticulously designed based on well-established mechanistic principles, which serve as the foundation for their intended functionality. While data analytics and machine learning offer remarkable capabilities in capturing intricate relationships within the data, the application of data science to engineering systems presents a challenge for effectively integrating mechanistic models with data analytics [7, 16].

Typically, during regular operations, first-principles models prove to be reliable and trustworthy. However, in situations that deviate from the predefined operating conditions, such as emerging scenarios or anomalies, the value of data as a valuable asset for real-time troubleshooting and decision-making becomes apparent. As a consequence, the field of industrial systems engineering is undergoing a paradigm shift towards a close and synergistic interaction between models and data [145, 106, 106]. In this new framework, mechanistic models are employed to handle expected situations, while data-driven approaches are leveraged to tackle unexpected circumstances. This integrated approach enables indus-

trial systems to enhance their operational performance and responsiveness, enabling swift adaptation and informed decision-making in diverse and dynamic operational scenarios.

While machine-learning models occasionally yield more accurate predictions compared to first-principles models, there exists a prevailing tendency to assign greater trust in the latter due to their interpretability. Hence, the interpretability of data-driven models has emerged as a desirable attribute since in the event of a failure in the data-driven decision-making process, human decision-makers must ultimately assume the responsibility for making critical decisions. Although machine learning has demonstrated notable achievements in domains like image processing and natural language processing, its widespread adoption in process and manufacturing industries necessitates the fulfillment of specific criteria. The paper, proposed by Qin and Chiang [125], outlines several indispensable attributes that are deemed essential for the sustained adoption of machine learning in industrial settings as follows:

(1) Compatibility with First-Principles Models or Process Knowledge: It is imperative for machine learning methods to match predictions from first-principles models or process knowledge that adhere to the fundamental laws of physics and chemistry. This compatibility ensures that the derived solutions align with the underlying principles governing the industrial processes.

(2) Effective Handling of Time-Varying Uncertainties: Machine learning approaches must demonstrate proficiency in effectively handling uncertainties that typically exhibit time-varying characteristics. Robust techniques are required to account for fluctuations and variations in the operational conditions and environmental factors that impact the industrial processes.

(3) Generation of Interpretable Solutions: A crucial aspect of machine learning models is their ability to generate solutions that are interpretable by decision-makers. The is vital for instilling trust and for allowing stakeholders to fully comprehend and make informed choices based on the provided insights.

Specialized analytics and domain expertise are essential for extracting optimal benefits from industrial systems data. In systems engineering, data-driven models serve three primary objectives: prediction, interpretation, and decision-making. These objectives are crucial in harnessing the full potential of industrial systems data. The first objective involves utilizing predictive analytics to forecast critical variables based on other variables present in the data. By leveraging sophisticated analytical techniques, such as machine learning algorithms, these models enable to predict crucial system parameters. The second objective entails extracting valuable features or novel insights from the data, facilitating effective visualization and diagnostics. This process aids in uncovering previously unseen patterns, relationships, and knowledge within the data, enhancing the understanding of the system under analysis. The third objective focuses on utilizing data-driven models to support decision-making processes. By integrating domain knowledge with the insights derived from the data, these models provide a robust foundation for making informed and data-backed decisions regarding system design, operation, maintenance, and optimization.

Overall, the utilization of specialized analytics in conjunction with domain expertise is imperative to maximize the benefits derived from industrial systems data. By employing data-driven models, the field of systems engineering can leverage prediction, interpretation, and decision-making capabilities to enhance system performance and facilitate efficient and informed decision-making processes.

Real-time process control and operational data typically exhibit complex dynamics and uncertainties, presenting as high-dimensional time series. These time series data are often characterized by temporal dependencies, both in terms of cross-correlations and auto-correlations. Consequently, there is a pressing need to develop techniques for dynamic dimensional reduction. Latent variable methods have been proposed for modeling the dynamics within the reduced latent subspace. This research thesis primarily aims to establish a connection between dynamic data analytics and dynamic systems theory, specifically focusing on the domains of modeling, prediction, and monitoring. By bridging these two disciplines, the objective is to advance our understanding and application of methodologies that effectively capture and analyze the intricate dynamics present in real-world datasets, leading to improved modeling, accurate prediction, and reliable monitoring techniques.

With the widespread adoption and integration of advanced high-precision sensors, high-performance computers, and distributed control systems, a substantial volume of industrial time series processing data has been acquired [124]. This proliferation of data has motivated the development of different mathematical tools for mining useful information about the monitored processes[122, 166, 108]. Amidst these approaches, principal component analysis (PCA), partial least squares (PLS), and canonical correlation analysis (CCA) have emerged as potent techniques for capturing the interdependencies between process data ($\mathbf{X}$) and quality data ($\mathbf{Y}$) [178, 180, 63, 158]. These methodologies play a pivotal role in enabling multivariate statistical process monitoring and fault diagnosis within the process analytics domain [74]. Notably, reduced rank regression (RRR) is a relatively lesser-known method in the process analytics community [73] that focuses on optimizing the prediction of variance in the output data. Throughout this thesis, we shall refer to process data as input data and quality data as output data or labeled data. These terminologies will be used interchangeably in the ensuing discussions.

Owing to its capability to handle high-dimensional and correlated data effectively, PCA is widely employed as an unsupervised method to extract latent variables from process data by maximizing the variances of the latent variables. These extracted latent variables possess explicit statistical interpretations, making PCA-based monitoring well-suited and widely adopted in industrial settings for anomaly detection. Nevertheless, as discussed in the work of Zhu and Qin [222], PCA-based process monitoring may encounter issues such as nuisance or false alarms since it lacks the incorporation of information from $\mathbf{Y}$ (the output data) during the modeling phase. In contrast, PLS and CCA are supervised algorithms that offer effective means to explore the relationships between $\mathbf{X}$ (process data) and $\mathbf{Y}$ (quality data). By leveraging the information from both $\mathbf{X}$ and $\mathbf{Y}$, these methods provide enhanced performance and greater accuracy in modeling and monitoring processes.

PLS (Partial Least Squares) is a latent variable extraction method that operates by projecting the input and output scores using outer models to maximize the covariance

among scores, thus establishing a connection between variables $\mathbf{X}$ and $\mathbf{Y}$. During the inner modeling phase, PLS aims to minimize the regression error, which competes with the objectives of the outer model. This gives rise to several limitations. Notably, not all of the latent variables extracted by PLS are relevant for modeling or detecting variations in quality data, as they may encompass irrelevant or orthogonal components. Moreover, even to predict a one-dimensional quality variable, multiple latent components are often required [126]. To address these issues, subsequent research studies have proposed various approaches. These include orthogonalized PLS [156], total PLS [214], and concurrent PLS [132]. These methodologies aim to overcome the aforementioned drawbacks and improve the efficiency and effectiveness of the PLS algorithm in real-world applications. By mitigating the inclusion of irrelevant components and optimizing the extraction of latent variables, these extensions of PLS offer promising solutions to enhance the predictive power and interpretability of the model.

Canonical Correlation Analysis (CCA) is a method that seeks to maximize the multidimensional correlation between input and output variables $\mathbf{X}$ and $\mathbf{Y}$. This emphasis on correlation enhances its predictive capability compared to PLS. Nonetheless, CCA does not take into account the variances of the input and output variables, potentially limiting its effectiveness in certain scenarios. In response to this limitation, Concurrent CCA (CCCA) was introduced as an extension. CCCA incorporates a subsequent Principal Component Analysis (PCA) decomposition in both process and quality spaces, enabling it to capture variance-related information more effectively [221]. By combining the strengths of CCA's correlation-maximizing ability with the variance-oriented approach of PCA, CCCA offers a promising avenue to improve the modeling and predictive performance in various applications.

In a recent study, Zhu and Qin introduced a novel approach called Latent Variable Regression (LVR) as an alternative method for extracting latent structured relationships between process and quality variables [222]. LVR differs from previous techniques like CCA and PLS in that it focuses on minimizing regression errors between process and quality scores in both the inner and outer models. This dual objective allows LVR to take into account both the correlation between X and Y and the variance of Y, resulting in enhanced predictive power compared to CCA and PLS when using a fixed number of latent variables. LVR can be viewed as a compromise between the characteristics of PLS and CCA, similar in spirit to the Reduced Rank Regression [135]. By striking a balance between correlation-based approaches and those that consider variance, LVR demonstrates promising potential in accurately capturing the underlying structure of the data and facilitating more effective predictions in various applications. The incorporation of both inner and outer regression objectives in LVR enables it to leverage the benefits of different modeling aspects, leading to an advanced method for latent variable extraction and regression analysis.

The algorithms discussed above are predicated on the assumption of a static underlying relationship between input and output variables. However, in real-world processes, quality measurements often exhibit dependencies on past data, owing to the existence of inherent system dynamics. Consequently, there is a pressing need to devise data modeling methodologies that can adeptly capture the dynamic relations among process variables ($\mathbf{X}$) and quality variables ($\mathbf{Y}$). The consideration of temporal dependencies and the incorporation

of past data are vital aspects in data modeling to ensure a comprehensive understanding of the evolving nature of real-world processes. Neglecting the dynamics can lead to sub-optimal models and inadequate predictions, thus compromising decision-making effectiveness in various applications.

For example, dynamic modeling versions ofPCA, PLS, and CCA have been proposed for dealing with dynamic processes. Then, these dynamic latent variable (DLV) models can be divided into two parts which are unsupervised DLV analytics and supervised DLV methods. These methods are briefly reviewed in the following sections.

1.1 Unsupervised Dynamic Latent Variable Analytics

For unsupervised DLV analytics, the dynamic latent variables are extracted in a reduced dimensional subspace based on the lagged auto-covariance matrices of the high-dimensional time series. To address the dynamics present in multidimensional data, a straightforward extension involves applying Principal Component Analysis (PCA) to an augmented data matrix that includes lagged measurements. This approach is known by various names, including Extended Empirical Orthogonal Function (EEOF) proposed in Weare and Nasstrom [168], Singular Spectrum Analysis (SSA) introduced by Vautard and Ghil [163], the Method of Delays (MOD) presented by Broomhead and King [25], and Dynamic Principal Component Analysis (DPCA) developed by Ku, Storer, and Georgakis [84]. Indeed, these methods have some notable disadvantages that should be considered. These limitations may impact its practical applicability and interoperability: (1)Increased complexity with lags: One significant drawback is that as the number of lags increases, the dimension of the loading vectors and the number of parameters also grow dramatically. This can lead to computational challenges and increased model complexity, potentially making it harder to handle large-scale datasets efficiently; (2) Lack of explicit representation: the method does not provide an explicit representation between the latent variables and the original measurement variables. This can hinder the interpretability of the model, as it becomes challenging to understand the relationships between the latent variables and the underlying process and quality variables; (3) Neglecting dynamic content: The focus of the method on variance rather than dynamic content of the data is a limitation. By primarily emphasizing variance, the model may not fully capture the time-dependent patterns and trends present in the data, which can limit its ability to provide insights into the underlying correlation structure and dynamics. To overcome these drawbacks, a series of dynamic inner latent variable methods are proposed.

In the pursuit of extracting dynamic latent variables (DLVs) that exhibit the highest covariance with their past predictions, a novel algorithm called Dynamic-Inner PCA (DiPCA) was introduced by Dong and Qin [50]. DiPCA utilizes this covariance to establish a rank order of the DLVs based on their prediction power. However, it is important to note that while DiPCA maximizes the covariance between the latent variables and their predictions, it does not necessarily optimize the R^2 value of the latent variable models. Subsequently, to address this concern, Dong and Qin proposed an alternative approach named Dynamic-Inner Canonical Correlation Analysis (DiCCA) [49]. DiCCA aims to maximize

the canonical correlation, which can lead to more favorable optimization results compared to DiPCA with respect to the R^2 value of the latent variable models. In addition, Dong, Liu, and Qin developed an efficient implementation of DiCCA based on Singular Value Decomposition (SVD) that exhibits numerical robustness, enhancing the practical applicability of the method [48]. Both DiPCA and DiCCA methods generate DLVs successively, one after another, and construct latent relations that do not interact with other exogenous factors. These non-interacting DLV relations are further interpreted in this section with the framework of the Kalman filter theory. This allows for the marginal description of the DLVs, shedding light on their dynamic characteristics and providing valuable insights into their behavior and evolution over time.

SFA, as a time series model, focuses on the invariance between neighboring samples by minimizing their difference, which can capture dynamics directly to some extent. SFA is widely used for object detection, action recognition, and blind source separation [174, 20, 204, 181, 157]. Considering its ability to extract slow variations from noises, SFA is adapted in manufacturing processes to differentiate normal deviations in operating conditions from dynamic anomalies [147]. Since it is common to change process objectives based on the market demands in modern manufacturing processes, identifying actual anomalies is essential to reduce nuisance alarms and thus improve the system's reliability. SFA extracts the first-order dynamics, and augmentation technique is also employed in SFA to extract multi-order dynamics [147, 206, 207, 143]. However, similar to other LVMs, augmented data exploitation leads to unclear dynamic interpretation in the extracted latent variables (LVs). Inspired by the dynamic inner LVMs, dynamic inner SFA (DiSFA) was also designed [153, 184] to build a serial correlation between the current sample and a weighted combination of past samples. DiSFA assigns optimal unconstrained weights to historical samples to maximize the similarity with the current sample. Thus, instead of emphasizing the slowness as in SFA, the nature of DiSFA is to maximize the predictability of LVs. Further, nonlinearity is not considered or effectively integrated in the aforementioned variants of SFA.

Based on the aforementioned methodologies, it can be deduced that for the present state-of-the-art dynamic extensions of Latent Variable Models (LVMs) continue to face certain limitations in establishing a robust framework for formulating dynamic relationships among the latent variables. Moreover, when solving the dynamic model, there remains a susceptibility to potential rank-deficient issues. To address these issues, in Chapter 6, a multi-step dynamic SFA, namely exponentially weighted SFA (EWSFA), is proposed to explicitly extract multi-order dynamics in modern manufacturing processes. EWSFA minimizes the difference between the LV of the current sample and the predicted LV formed by the weighted combination of past samples. Different from dynamic inner LVMs, the exponentially weighted moving average [71] is used to weight past samples to account for the decaying nature of time series. The extracted dynamics are easy to interpret and follow the decaying nature of time series data.

1.2 Supervised Dynamic Latent Variable Methods

In the field of supervised Dynamic Latent Variable (DLV) methods, the latent variables demonstrate dynamic inter-dependencies with the quality label variables, thus necessitating the achievement of suitable trade-offs. Notably, these latent variables may constitute a combination of dynamic and static LVs, setting them apart from the DLVs derived through unsupervised DLV analytics. This distinction is readily apparent when comparing them with their static counterparts. Specifically, the label-relevant latent subspace and the complement residual subspace may encompass a blend of dynamic and static LVs. This nuanced observation underscores the intricate nature of DLV models in supervised settings, emphasizing the need to discern and leverage their dynamic and static components appropriately.

The early exploration of employing dynamic latent variables for predictive purposes can be traced back to the work of Weare and Nasstrom [168]. Their approach utilized DPCA (Dynamic Principal Component Analysis) components as empirical orthogonal functions to predict the associated labels. Subsequently, Ricker [137] and Qin, McAvoy [133], as well as Qin, McAvoy [131], adopted Partial Least Squares (PLS) and neural net PLS techniques to construct dynamic predictive models based on time-lagged process data. Due to their simplicity and ease of implementation, these methodologies have gained widespread adoption in supervised dynamic modeling over the preceding decades. A nonlinear variant of dynamic PLS (DPLS) was introduced by Baffi, Morris, and Martin [15] within the framework of an autoregressive model with exogenous inputs (ARX). This nonlinear approach was effectively applied to predictive control tasks. Furthermore, extensions of DPLS to accommodate batch processes and nonlinear processes were proposed, utilizing an augmented process matrix by Chen and Liu [30], Fletcher, Morris, Montague, and Martin [58], as well as Hu, Zhao, and Liang [70]. Stock and Watson proposed an alternative supervised dynamic modeling method that employed a preliminary Principal Component Analysis (PCA) [155]. Their approach involved conducting PCA on the process data to extract lagged latent factors, which were subsequently linked to quality variables using ordinary least squares. This method can be viewed as a dynamic extension of PCR (Principal Component Regression) but shares some of the limitations inherent in static PCR. The structure of work presented above constitutes significant contributions to the field of supervised dynamic modeling, offering various methodologies that have been widely utilized and studied for their respective merits and drawbacks.

The aforementioned augmentation methodologies have demonstrated efficacy in capturing dynamic variations between process and quality variables. However, a notable drawback is their propensity to introduce a substantial increase in the dimensionality of the process matrix, resulting in a significant computational burden and unwieldy models. Addressing this concern, an enhanced dynamic Partial Least Squares (PLS) approach was proposed by Kaspar and Ray [79]. This improved method aims to maximize the fraction of explained quality variance at each stage of the modeling process. Subsequently, Lakshminarayanan, Shah, and Nandakumar devised an alternative dynamic PLS technique by directly modifying the inner relation of PLS [88]. Here, the process and quality scores were linked through a dynamic component, such as an autoregressive model with exogenous inputs (ARX) or

Hammerstein model. This formulation proves effective in yielding satisfactory solutions, particularly when input signals primarily exhibit low-frequency content. However, it is noteworthy that the outer model in this context is derived using the static PLS algorithm.

Li, Liu, Qin, and Zhou introduced an innovative approach termed Dynamic Partial Least Squares (DPLS) that enhances the compactness of the dynamic latent variable structure by maximizing the covariance between the quality scores and a weighted representation of process scores [94]. Notably, this novel formulation involves a two-step process, where the inner dynamic model is initially established, leading to a refined and concise representation. More recently, a dynamic inner Partial Least Squares (DiPLS) method was proposed by Dong and Qin to concurrently estimate consistent inner and outer models [51]. DiPLS achieves this by maximizing the covariance between the quality scores and a weighted combination of process scores, which incorporates a dynamic inner model. The resulting models are explicit and compact, thereby facilitating their interpretation as dynamic latent factors. In addition to DPLS and DiPLS, analogous methodologies have been extended to Canonical Correlation Analysis (CCA) and Latent Variable Regression (LVR). Dong and Qin [49] and Zhu et al. [224] independently developed similar techniques for CCA and LVR, respectively, to further advance the field of dynamic latent variable modeling. These advancements represent notable contributions to the realm of dynamic modeling, offering sophisticated and interpretable methodologies that enhance the understanding and analysis of dynamic latent factors across various domains and applications.

However, there are still challenges in the field of supervised dynamic latent variable methods. Owing to the elevated sampling rates commonly employed in industrial processes, there exists a potential for the presence of redundant and collinear information in consecutively sampled data. Utilizing such data directly for modeling and monitoring purposes can lead to sub-optimal performance. To address this issue, Chapter 3 presents a novel approach termed dynamic weighted Partial Least Squares (DWPLS). The DWPLS method is designed to effectively handle the redundant information present in adjacent samples, offering a promising solution for improved modeling and monitoring in industrial processes. DWPLS reduces the dependency in adjacent samples by designing a series of basis functions of polynomial and Gaussian types.

In order to enhance the interpretability and reliability of the models, our thesis will persistently emphasize the refinement of multivariate statistical techniques. Nevertheless, within the scope of the aforementioned methodologies, an unaddressed aspect pertains to the auto-correlations existing between current and past quality variables. Exploiting these correlations could significantly enhance the efficacy of the modeling and monitoring processes. To address this crucial aspect, Chapter 4 introduces a novel approach referred to as the Dynamic Auto-Regressive Latent Variable Model (DALVM). This model effectively captures the dynamic behavior of quality variables by incorporating information from not only past process samples but also past quality samples. By integrating these latent dependencies, the DALVM offers a promising means of improving the overall performance of our analysis. While these models exhibit varying degrees of success, certain issues still necessitate resolution. Primarily, when confronted with high rates of missing data, especially in scenarios where dissimilar sampling rates are employed during data collection, they fail to achieve satisfactory performance.

In order to tackle this issue comprehensively, Chapter 6 introduces an enhanced missing data imputation strategy based on a multi-step ahead prediction method. Initially, a novel approach called the Efficient Dynamic Auto-Regressive CCA (EDACCA) is proposed to establish dynamic relationships between the current quality sample and a weighted combination of past processes and quality data. The introduction of two weighting parameters enhances the model's flexibility, while various techniques, including singular value decomposition, are incorporated to bolster its efficiency. Moreover, in order to mitigate the impact of erroneous data and address irregular sampling rates, a multi-step-ahead (MS) prediction model is devised, building upon the foundation of EDACCA. This MS prediction model is subsequently adapted for the purpose of handling irregular missing data imputation, thus providing a robust and comprehensive solution to the prevailing issue.

Following the above, the novel contributions of this work can be summarized as follows:

a. Chapter 3 proposed dynamic weighted latent variable methods to address the redundancy between neighbor samples by adapting basis functions (DWPLS).

b. Chapter 4 designed dynamic auto-regressive models to make full use of both process and quality information and to achieve consistency of their outer and inner models (DALVM).

c. Chapter 5 developed efficient dynamic auto-regressive models for multi-step-ahead (MS) prediction and missing data caused by irregular sampling rates (EDACCA).

d. Chapter 6 developed an efficient extension of slow feature analysis to capture multi-step dynamics (EWSFA). The extracted dynamics are easy to interpret and follow the decaying nature of time series data.

The remainder of this thesis is arranged as follows. Chapter 3 describes the details of the proposed dynamic weighted PLS modeling method and its corresponding monitoring strategy. Case studies with the Tennessee Eastman Process (TEP) and a distillation process are presented to demonstrate their advantages. Chapter 4 proposes a novel dynamic latent variable model, dynamic auto-regressive latent variable model (DALVM) to fully exploit the dynamics between process and quality data. Concurrent decomposition techniques are designed for DALVM for comprehensive monitoring purposes. A numerical simulation, the Tennessee Eastman process (TEP), and the three-phase flow facility (TPFF) are used to illustrate the efficacy of the proposed algorithm. Chapter 5 further developed an efficient dynamic auto-regressive extension of CCA to realize a comprehensive decomposition and modeling of quality data space. The TEP and a wind energy generation process, are employed to demonstrate the effectiveness of the proposed method. Chapter 6 gives more details about the proposed exponentially weighted slow feature analysis (EWSFA) algorithm and its comprehensive dynamic monitoring framework. Case studies on two industrial processes are presented to illustrate the superiority of the proposed methods for oscillation analysis and monitoring performance. Chapter 7 illustrates the conclusion and future works.

Chapter 2

Background and Literature Review

2.1 Supervised static Learning Methods

Within machine learning, supervised learning algorithms have played an important role for classifying infrequent quality data with respect to process inputs [28, 40, 66]. Among them, multivariate statistical analysis methods are widely studied and applied in the process systems engineering community due to their additional ability of dimension reduction and interpretability, and their capability for detection and diagnosis [63, 69, 216]. Some popular methods include partial least squares (PLS) [63], canonical correlation analysis (CCA) [158], and latent variable regression (LVR) [222]. Studies involving the application of these methods are briefly reviewed in the following sub-sections.

2.1.1 Partial Least Squares

PLS was first proposed by Wold et al. [175] to exploit the relation between process and quality data by maximizing the covariance between their latent variables. Assume that $\mathbf{X} \in \mathbb{R}^{n\times m}$ and $\mathbf{Y} \in \mathbb{R}^{n\times p}$ are the input and output data respectively, where m and p are the number of process and quality variables, and n is the number of samples. The objective of PLS can be represented as [63]

$$\begin{aligned} \max_{\mathbf{q},\mathbf{w}} \quad & J = \mathbf{q}^\top \mathbf{Y}^\top \mathbf{X}\mathbf{w} \\ \text{s.t.} \quad & \|\mathbf{w}\| = 1, \|\mathbf{q}\| = 1 \end{aligned} \tag{2.1}$$

where $\mathbf{w}$ and $\mathbf{q}$ are the process and quality weighting vectors respectively, $\mathbf{t} = \mathbf{X}\mathbf{w}$ is the latent score vector of process variables, and $\mathbf{u} = \mathbf{Y}\mathbf{q}$ is the score vector for quality variables. It is noted that the description of variables as input/output and process/quality are used interchangeably in this work.

A linear regression model is constructed between quality and process scores in the inner model of PLS by

$$\mathbf{u} = b\mathbf{t} + \mathbf{r} \tag{2.2}$$

where $\mathbf{r}$ is the regression error. With the iterative procedure, l sets of latent variables can be extracted by PLS, and $\mathbf{X}$ and $\mathbf{Y}$ can be decomposed as

$$\begin{aligned} \mathbf{X} &= \mathbf{T}\mathbf{P}^\top + \mathbf{E} \\ \mathbf{Y} &= \mathbf{T}\mathbf{B}\mathbf{Q}^\top + \mathbf{F} \end{aligned} \tag{2.3}$$

where $\mathbf{T} \in \mathbb{R}^{n\times l} = [\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_l]$, $\mathbf{P} \in \mathbb{R}^{m\times l} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_l]$, $\mathbf{Q} \in \mathbb{R}^{q\times l} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_l]$, $\mathbf{B} = \mathrm{diag}[b_1, b_2, \cdots, b_l]$, l is the number of latent variables, and $\mathbf{E}$ and $\mathbf{F}$ are the residuals of process and quality variables respectively.

For online application, the PLS model can be used to predict quality variables $\hat{\mathbf{y}}$ from a new sample $\mathbf{x}$ by

$$\hat{\mathbf{y}} = \mathbf{Q}\mathbf{R}^\top\mathbf{x} \tag{2.4}$$

where $\mathbf{R} \in \mathbb{R}^{m\times l} = \mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}$ [41, 43].

While the PLS model compresses the data into a reduced space relating quality variables to inputs, its extracted latent scores still contain quality-irrelevant variations. Correspondingly, the variance of the residuals may be very large, which imposes difficulty for subsequent monitoring tasks. To overcome these issues, total and concurrent modeling strategies were proposed to decompose input/output variables' spaces further to achieve improved performance [214, 132, 57, 200, 208].

2.1.2 Canonical Correlation Analysis

Canonical correlation analysis (CCA) algorithm was proposed by Hotelling [68] to capture the correlation between two datasets. Given $\mathbf{X} \in \mathbb{R}^{n\times m}$ and $\mathbf{Y} \in \mathbb{R}^{n\times p}$, CCA aims to maximize the correlation between $\mathbf{X}$ and $\mathbf{Y}$ as follows

$$\begin{aligned} \max_{\mathbf{q},\mathbf{w}} \quad & J = \mathbf{q}^\top\mathbf{Y}^\top\mathbf{X}\mathbf{w} \\ \text{s.t.} \quad & \|\mathbf{X}\mathbf{w}\| = 1, \|\mathbf{Y}\mathbf{q}\| = 1 \end{aligned} \tag{2.5}$$

Singular value decomposition (SVD) or eigendecomposition method is utilized to solve the objective in Eq. (2.5), and the details can be found in [68]. A regularized CCA (rCCA) was proposed by Zhu et al. to improve its robustness in strongly colinear scenarios [221].

After performing CCA, $\mathbf{X}$ and $\mathbf{Y}$ are decomposed into

$$\begin{cases} \mathbf{X} = \mathbf{T}\mathbf{P}^\top + \mathbf{E} \\ \mathbf{Y} = \mathbf{T}\mathbf{Q}^\top + \mathbf{F} \end{cases} \tag{2.6}$$

where $\mathbf{T}$, $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{E}$ and $\mathbf{F}$ have the same meaning as those in Eq. (2.3). Similarly, for a new collected sample $\mathbf{x}$, the quality variables $\mathbf{y}$ can be predicted by

$$\hat{\mathbf{y}} = \mathbf{Q}\mathbf{R}^\top\mathbf{x} \tag{2.7}$$

which can be used further for online monitoring and diagnosis [221, 32, 223].

2.2 Supervised Dynamic Latent Variable Models

The aforementioned PLS and CCA rely on an implicit assumption that only static relations exist in the data. However, for real-world industrial processes, the dynamic relations involved in process and quality variables will reduce the credibility and efficiency of these models. Therefore, their dynamic extensions were proposed to deal with dynamic systems. Qin and McAvoy [131] and Ricker [137] applied the augmentation method directly for PLS, and it is a straightforward method in which the input matrix is combined with a number of lagged values of quality and process variables. However, it is hard to interpret the relations extracted by these methods and the computation costs were increased unnecessarily.

Alternatively, Lakshminarayanan et al. [88] introduced time series models between process and quality latent variables to build the dynamic inner model. Then Li et al. [94] proposed a novel dynamic PLS to maximize the covariance between the quality score and the weighted lagged process scores. Dong and Qin [51] achieved improved modeling performance by designing dynamic inner and outer models which have consistent dynamic formulations, this novel algorithm is referred to as dynamic inner PLS (DiPLS).

2.2.1 Dynamic Inner PLS

Qin et al. [127] compared the reported dynamic variants of PLS and concluded that DiPLS provides good modeling performance because of its consistent dynamic outer and inner models [51]. The objective of the DiPLS outer model is

$$\begin{aligned} &\max_{\mathbf{q},\mathbf{w},\boldsymbol{\beta}} \quad \mathbf{q}^\top \mathbf{Y}_s^\top \mathbb{Z}_s [\boldsymbol{\beta} \otimes \mathbf{w}] \\ &\text{s.t.} \quad \|\mathbf{q}\| = 1, \ \|\boldsymbol{\beta}\| = 1, \ \|\mathbf{w}\| = 1 \end{aligned} \tag{2.8}$$

where

$$\begin{aligned} \mathbb{Z}_s \in \mathbb{R}^{N\times(s+1)m} &= [\mathbf{X}_s, \mathbf{X}_{s-1}, \cdots, \mathbf{X}_0] \\ \mathbf{Y}_s \in \mathbb{R}^{N\times p} &= [\mathbf{y}_s, \mathbf{y}_{s+1}, \cdots, \mathbf{y}_{s+N}]^\top \\ \boldsymbol{\beta} \in \mathbb{R}^{s+1} &= [\beta_0, \beta_1, \cdots, \beta_s]^\top \end{aligned} \tag{2.9}$$

$\mathbf{X}_i \in \mathbb{R}^{N\times m} = [\mathbf{x}_i, \mathbf{x}_{i+1}, \cdots, \mathbf{x}_{i+N}]^\top$ $(i = 0, 1, \cdots, s)$ is a subset of process matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$, $\otimes$ is the Kronecker product, and s is the dynamic order. The objective of Eq. (2.8) is to maximize the covariance between current quality variables and current and past process variables. To align with the denotations in Eq. (2.1), we have $n = N + s + 1$.

Its consistent dynamic inner model is designed to relate the quality score and past process scores according to the following linear regression expression.

$$u_k = \beta_0 t_k + \beta_1 t_{k-1} + \cdots + \beta_s t_{k-s} + r_k \tag{2.10}$$

where r_k is the regression error, $t_k = \mathbf{x}_k^\top \mathbf{w}$ and $u_k = \mathbf{y}_k^\top \mathbf{q}$ are the principal component projection of $\mathbf{x}_k$ and $\mathbf{y}_k$ respectively, and $\mathbf{w}$ and $\mathbf{q}$ have the same meanings as in Eq. (2.1).

Deflation must be performed on $\mathbf{X}$ and $\mathbf{Y}_s$ for the extraction of next sets of latent variables. After training the DiPLS model, the quality variations can be predicted by

$$\hat{\mathbf{y}} = \mathbf{Q}\hat{\mathbf{u}}_s \tag{2.11}$$

where $\hat{\mathbf{u}}_s$ is obtained from $\mathbf{t}$ based on dynamic inner modeling, and $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_l]$ is the key model parameter relating process variables with quality data. More details of DiPLS can be found in [51].

2.2.2 Supervised Dynamic inner CCA

Supervised DiCCA (sDiCCA) was proposed to improve the dynamic modeling performance of CCA [49]. Let denote $\mathbf{x}_k$ and $\mathbf{y}_k$ as the process variables and quality variables collected at time k, their corresponding latent variables are

$$\begin{aligned} t_k &= \mathbf{x}_k^\top \mathbf{w} \\ u_k &= \mathbf{y}_k^\top \mathbf{q} \end{aligned} \tag{2.12}$$

sDiCCA assumes the following weighted relations between u_k and t_k in its inner model.

$$u_k = \beta_0 t_k + \beta_1 t_{k-1} + \cdots + \beta_s t_{k-s} + r_k \tag{2.13}$$

where β_i $(i = 0, 1, \ldots, s)$ are the weights, and r_k is the regression error. The output latent variable $\hat{u}_k$ can then be predicted by

$$\hat{u}_k = \sum_{i=0}^{s} \beta_i t_{k-i} = \sum_{i=1}^{s} \beta_i \mathbf{x}_{k-i}^\top \mathbf{w} \tag{2.14}$$

To achieve dynamic consistency, the outer model of sDiCCA is designed to maximize the correlation between the process and quality scores. More details can be found in Chapter 5.

2.3 Unsupervised static Learning Methods

Unsupervised latent variable methods are adept at reducing high-dimensional data to low-dimensional latent variables in the absence of labeled or target data, denoted as Y. Taking the example of PCA, the maximization of the variance of latent variables serves as an intrinsic objective, owing to the absence of alternative target information. Different unsupervised learning techniques are briefly reviewed below.

2.3.1 Principal Component Analysis

A major feature of latent variable methods is their ability to reduce dimensions and focus on relevant features in the reduced-dimensional subspace. Principal component analysis (PCA) projects the data to a vector of latent scores $\mathbf{t} = \mathbf{X}\mathbf{p}$, and makes the scores have the largest variance, that is,

$$\begin{aligned} \max_{\mathbf{p}} \quad & J = \mathbf{p}^\top \mathbf{X}^\top \mathbf{X}\mathbf{p} \\ \text{s.t.} \quad & \|\mathbf{p}\| = 1 \end{aligned} \tag{2.15}$$

The above problem can be solved by the following

$$\mathbf{X}^\top \mathbf{X}\mathbf{p} = \lambda \mathbf{p} \tag{2.16}$$

where the vector $\mathbf{p}$ represents the eigenvector of the matrix product $\mathbf{X}^\top\mathbf{X}$, corresponding to its largest eigenvalue. It is noteworthy that this particular matrix is directly proportional to the sample covariance matrix, rendering it commonly denoted as the covariance matrix in colloquial parlance. The vector $\mathbf{p}$ is referred to as the loading vector for the principal components. After the number of principal components is detected, the first l loading vectors can be represented as $\mathbf{P} \in \mathbb{R}^{m\times l} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_l]$ which are the eigenvectors of $\mathbf{X}^\top\mathbf{X}$ corresponding to the l largest eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_l$ in descending order. The principal component score matrix and the residual matrix after extracting l principal components can be represented as follows

$$\begin{aligned} \mathbf{T} &= [\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_l] = \mathbf{X}\mathbf{P} \\ \widetilde{\mathbf{X}} &= \mathbf{X} - \mathbf{T}\mathbf{P}^\top \end{aligned}$$

2.3.2 Slow Feature Analysis

SFA was proposed by Wiskott and Sejnowski [174] to extract slowly varying features from high dimensional temporal input signals. Given an m-dimensional input signal $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_m(t)]^\top$, SFA seeks for l sets of transformation functions $\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_l(\mathbf{x})]^\top$, and transforms $\mathbf{x}(t)$ into l LVs $\mathbf{s}(t) = [s_1(t), s_2(t), \ldots, s_l(t)]^\top$ by optimizing

$$\min_{g_j(\mathbf{x})} \quad \langle \dot{s}_j^2 \rangle_t \tag{2.17}$$

$$\text{s.t.} \quad \langle s_j \rangle_t = 0, \quad \langle s_j^2 \rangle_t = 1, \quad \langle s_i s_j \rangle_t = 0 \quad \forall i \neq j \tag{2.18}$$

where the angle bracket denotes temporal averaging $\langle f \rangle_t \equiv \frac{1}{t_1 - t_0}\int_{t_0}^{t_1} f(t)dt$, and $\dot{s}_j$ is the first-order derivative of $s_j(t)$. SFA defines $\langle \dot{s}_j \rangle_t$ in Eq. (2.17) as an indicator for *slowness* of LVs, and LVs are extracted with an ascending order of $\langle \dot{s}_j \rangle_t$. The constraints in Eq. (2.18) ensure that the extracted LVs have zero mean and unit variance, and are decorrelated from previous LVs. It is noted that LVs s_j $(j = 1, 2, \ldots, l)$ are also called slow featuress (SFs) in SFA, and both LVs and SFs are used interchangeably in this work.

2.4 Unsupervised Dynamic Latent Variable Models

Unsupervised dynamic LVMs have been also proposed recently. While supervised LVM is used to obtain dynamic data relationships between two variable sets, the unsupervised counterpart seeks to extract one or more latent variables that represent optimal linear combinations of a single variable set, with a focus on maximizing either auto-covariance or auto-correlation. The goal is that the residual error that remains after extracting the most predictable latent variables from the data will be close to white noise. In this sense, the unsupervised dynamic LVMs extract a set of principal time series from a multi-dimensional time series.

2.4.1 Dyanmic Inner PCA

The DiPCA algorithm, which facilitates the extraction of principal time series, offers several key advantages:

1. Enhanced Predictability: the algorithm has the ability to predict dynamic components based on past data. This predictability enhances the overall reliability of the algorithm.

2. Enhanced Data Interpretation and Diagnosis: The extracted dynamic components derived through DiPCA serve as effective tools for highlighting essential dynamic features in the data. These features might be challenging to discern from the original data directly. Consequently, DiPCA aids in better data interpretation and facilitates accurate diagnosis.

3. Reduced Temporal Correlation: An important benefit of DiPCA is its capability to effectively extract dynamic components from which it is easier to explain the resulting prediction errors.

In general, the goal of DiPCA is to extract dynamics in a latent variable t_k so that the current value can be predicted from the past using the following auto-regressive (AR) model,

$$t_k = \beta_1 t_{k-1} + \beta_2 t_{k-2} + \cdots + \beta_s t_{k-s} + r_k \tag{2.19}$$

where the latent variable is a linear combination of the original variables such as $t_k = \mathbf{x}_k^\top \mathbf{w}$

Since the objective of the dynamic inner PCA algorithm is to maximize the covariance between the extracted data and the prediction, its objective can be represented as follows [50]

$$\begin{aligned} &\max_{\mathbf{w},\boldsymbol{\beta}} \quad \mathbf{w}^\top \mathbf{X}_{s+1}^\top \mathbb{Z}_s [\boldsymbol{\beta} \otimes \mathbf{w}] \\ &\text{s.t.} \quad \|\mathbf{w}\| = 1, \ \|\boldsymbol{\beta}\| = 1 \end{aligned} \tag{2.20}$$

where

$$\begin{aligned} \mathbb{Z}_s \in \mathbb{R}^{N\times(s)m} &= [\mathbf{X}_s, \mathbf{X}_{s-1}, \cdots, \mathbf{X}_1] \\ \boldsymbol{\beta} \in \mathbb{R}^s &= [\beta_1, \beta_2, \cdots, \beta_s]^\top \end{aligned} \tag{2.21}$$

This DiPCA can also be viewed as a whitening filter applied to the data. After all DiPCA components are extracted, the prediction errors are essentially white as virtually all the dynamic relationships in data are extracted. An important notion of this whitening filter is that it has a reduced number of latent variables compared to the number of variables in that data, and is appropriate for modeling the common case of highly collinear data from real-world problems. This solution is different from a full-dimensional vector autoregressive (VAR) model that requires inverting a covariance matrix, which can be ill-conditioned due to highly correlated data. Furthermore, the DiPCA latent variables have a clear objective and can provide useful features for data-based interpretation, visualization, and diagnosis.

2.4.2 Unsupervised Dynamic Inner CCA

CCA has shown its advantages in prediction and time series modeling over PLS [221], and its unsupervised dynamic inner counterpart is also designed, which is referred to as dynamic inner CCA (DiCCA) [49]. The optimization objective of DiCCA is

$$\begin{aligned} \max_{\mathbf{w},\boldsymbol{\beta}} \quad & \mathbf{w}^\top \mathbf{X}_{s+1}^\top \mathbb{Z}_s[\boldsymbol{\beta} \otimes \mathbf{w}] \\ \text{s.t.} \quad & \|\mathbf{X}_{s+1}\mathbf{w}\| = 1, \ \|\mathbb{Z}_s[\boldsymbol{\beta} \otimes \mathbf{w}]\| = 1 \end{aligned} \tag{2.22}$$

where

$$\begin{aligned} \mathbb{Z}_s \in \mathbb{R}^{N\times(s)m} &= [\mathbf{X}_s, \mathbf{X}_{s-1}, \cdots, \mathbf{X}_1] \\ \boldsymbol{\beta} \in \mathbb{R}^s &= [\beta_1, \beta_2, \cdots, \beta_s]^\top \end{aligned} \tag{2.23}$$

$\mathbf{X}_i \in \mathbb{R}^{N\times m} = [\mathbf{x}_i, \mathbf{x}_{i+1}, \cdots, \mathbf{x}_{i+N}]^\top$ $(i = 1, 2\cdots, s+1)$. It is noted that the matrix subscripts in Eq. (2.22) are different from those in Eq. (2.8), since DiCCA was initially designed as an unsupervised learning method; that is, DiCCA maximizes the autocorrelations between current process score and a weighted combination of past process scores.

The inner model of DiCCA is designed to capture the dynamics between current process scores and past process scores, which is

$$t_k = \beta_1 t_{k-1} + \beta_2 t_{k-2} + \cdots + \beta_s t_{k-s} + r_k \tag{2.24}$$

where t_k is the process score for current sample, and t_i $(i = k-1, k-2, \ldots, k-s)$ denotes past process scores. Compared with DiPLS, $\boldsymbol{\beta}$ does not need to be re-estimated in the inner model since $\boldsymbol{\beta}$ is solved coincidentally in the outer model. More details can be found in [49].

2.4.3 Dynamic Inner Slow Feature Analysis

To improve dynamic modeling performance, multi-step SFA methods were proposed, such as multi-step dynamic SFA [103] and dynamic inner SFA [153, 184]. Among them, DiSFA has achieved improved performance since it retains consistent dynamic relations. Inspired by the work of DiCCA [48], Xu et al. [184] proposed DiSFA that capture dynamics from discrete time series signals $\mathbf{x}(k)$ by optimizing the following objective.

$$\min_{\mathbf{w}_j} \quad \left\langle (s_j - \hat{s}_j)^2 \right\rangle_t \tag{2.25}$$

$$\text{s.t. } \langle s_j \rangle_t = 0, \quad \langle s_j^2 \rangle_t = 1, \quad \langle \hat{s}_j^2 \rangle_t = 1, \quad \langle s_i s_j \rangle_t = 0 \quad \forall i \neq j \tag{2.26}$$

where $\hat{s}_j(t)$ is predicted from LVs of the past d samples,

$$\hat{s}_j(t) = \sum_{i=1}^{d} \beta_i s_j(t-i) \tag{2.27}$$

where $s_j(t-i)$ is the j^{th} LV of sample $\mathbf{x}(t-i)$, β_i are their corresponding weights, and d is the time lag. The detailed solution of DiSFA can be referred to [153, 184], and the optimization goals of DiSFA and DiCCA are exactly the same. The additional constraint $\langle \hat{s}_j^2 \rangle_t = 1$ in Eq. (2.26) imposes a limit for the weight parameters $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_d]^\top$, and optimal unconstrained values can be assigned to β_i, which implies that the main objective of DiSFA in Eq. (2.25) is to maximize the *predictability* of the LVs.

After performing DiSFA, a new sample $\mathbf{x}(k)$ can be decomposed as follows

$$\mathbf{x}(t) = \mathbf{P}\mathbf{G}(q^{-1})\mathbf{s}(t) + \mathbf{e}(t) \tag{2.28}$$

where $\mathbf{P} \in \mathbb{R}^{m \times l} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_l]$; $\mathbf{G}(q^{-1})\mathbf{s}(t) = \sum_{i=1}^{d} \boldsymbol{\beta}_i \mathbf{s}(t-i)$ is the transfer function of the dynamic model, and $\mathbf{e}(t)$ represents the residuals.

2.5 Nonlinearity Handling

With the development of process control systems and the complexity of industrial processes, nonlinearity is a big challenge to deal with, and neural networks (NNs) are widely used to capture nonlinearity in industrial data. A neural network can be seen as an artificial network that consists of artificial neurons or nodes.

Deep neural networks (DNNs) have shown their superiority in many fields such as image classification, financial market prediction, and chemical process fault diagnosis [92, 38, 44, 182]. Many variants of deep learning methods are also proposed based on DNN, such as stacking automatic encoders, convolution neural networks, deep probabilistic neural networks, and deep fuzzy neural networks [97, 90, 29, 45].

Autoencoder (AE), a typical unsupervised NN algorithm, includes two important parts which are the encoder and decoder. Encoder aims to encode a set of data with nonlinear

activation functions for dimensional reduction and yields code. Decoder then tries to reconstruct the original data with minimal error. It is noted that when the activation function is linear and the number of hidden layers is one, AE will reduce to PCA, and thus AE can be regarded as a generalized PCA. Extensions of AE were also proposed to improve its robustness. Ng [110] proposed a sparse AE in which the sparsity constraint is introduced to the hidden layer and an additional regularization is applied to exploit the inner structure of data. Vincent et al. [165] proposed the denoising AE (DAE) corrupt input with methods of stochastic mapping. DAE improves the model's robustness and ensures consistent results even when slight disturbances occur. Dropout strategy was also adapted into AE to drop out some hidden units in a hidden layer, which is powerful when training resources are not enough to train large amounts of neural networks [154]. Rifai et al. [139] proposed a deterministic neural network architecture, contractive AE, by designing a Jacobian matrix for activation functions to improve its robustness over the traditional AEs. Moreover, the generative model, variational AE [81, 136], was proposed to deal with missing data by introducing a parameterized probability density function.

Neural networks and deep learning methods have shown their potential and promising outcomes in different applications. However, their interpretability is relatively poor which reduces their applicability in practical industrial processes [53, 4].

Chapter 3

Dynamic Weighted Partial Least Squares

3.1 Overview

[1] To address redundant information existing among adjacent samples with high sampling rates, a dynamic weighted PLS (DWPLS) method is proposed to deal with the aforementioned issue. DWPLS maximizes the covariance between the quality latent scores and a weighted representation of lagged process scores. The relations of lagged process scores are learned through a weighted summation of basis functions. The monitoring scheme of DWPLS is also constructed. Case studies with two processes, the Tennessee Eastman process, and a distillation process, are conducted to show the advantages of DWPLS over existing methods in terms of regression and fault detection performance.

3.2 Introduction

In the past decades, supervised modeling and monitoring have achieved great success in modern industries to improve operating safety and production efficiency [61, 148, 134, 223]. The collected data in these processes are massive, highly dimensional, and highly correlated due to the complexity of physical and chemical conditions and process operations, leading to the ill performance of traditional regression methods. To extract valuable information efficiently from the high dimensional data with high correlation, multivariate statistical methods such as partial least squares (PLS) and canonical correlation analysis (CCA) are widely employed [173, 121, 221]. Among them, PLS is an efficient latent variable algorithm to extract cross-correlations between process variables ($\mathbf{X}$) and quality variables ($\mathbf{Y}$) [175, 42, 114]. Two sub-models are included in PLS: firstly, in order to acquire the maximum covariance of latent variables in the outer model, compute the loading vectors for

[1] Adapted from Xu, Bo, & Zhu, Q. (2020). Online quality-relevant monitoring with dynamic weighted partial least squares. Industrial & Engineering Chemistry Research, 59 (48), 21124-21132

process and quality variables. After those calculations, an inner linear model is then well-constructed between process scores and quality scores. Deflation is also being performed to mitigate the effect of the extracted latent variables from $\mathbf{X}$ and $\mathbf{Y}$. It is noted that input/output variables and process/quality variables are used interchangeably in the paper.

An implicit assumption of PLS is that the relation between process and quality variables is static. Hence, it is not applicable to most industrial processes where dynamics are involved, and the dynamic covariance cannot be exploited fully and accurately by static PLS. Thus, dynamic variants of PLS are proposed to deal with dynamic systems. For the purpose of extracting the dynamic relation between process and quality variables, Ricker [137] constructed the input data matrix with lagged values of the process variables. Qin and McAvoy [133] modeled the dynamics by applying PLS on the data matrix which includes lagged values of both process and quality variables. However, these models involve substantial increments in the dimension of $\mathbf{X}$ that can lead to a high computation cost.

Alternatively, Kaspar and Ray [79] employed a discrete transfer function to represent the dynamic transformation of $\mathbf{X}$, which eliminated the dynamic components in $\mathbf{X}$, and then a static model was established between filtered $\mathbf{X}$ and $\mathbf{Y}$ where lagged variables no longer exist in the outer relations. After that, a dynamic inner model was built between process and quality scores. Qin and McAvoy [130] built a dynamic inner model by implementing the single-input-single-output neural network. Lakshminarayanan et al. [88] proposed a dynamic inner model by introducing time series models such as auto-regressive models with exogenous variables. However, these methods just include a dynamic inner model and lack modification about the static outer model.

Li et al. [94] put forward a dynamic PLS to handle the dynamics by using a weighted vector to combine lagged process data, and an explicit dynamic outer model was developed between quality data and weighted process data. Qin and McAvoy [131] used lagged values of process variables only or both process and quality variables as input data in the dynamic outer model, and a single-input-single-output neural network was used simultaneously which built a dynamic inner model. All of these methods modified the inner or outer relation only or paid no attention to the consistency formulations of dynamic inner and outer relations. Recently, Dong and Qin [51] focused on the consistency of the dynamics of inner and outer models, and developed an improved dynamic PLS (DiPLS). In DiPLS, dynamics are incorporated in both inner and outer models. The monitoring scheme based on DiPLS was developed by using past process information to improve the ability of monitoring and fault diagnosis. State-space models with subspace identification techniques can also be employed for dynamic monitoring purposes, which focuses on extracting state variables that are inherently auto-correlated from dynamic data [142, 109]. All these dynamic methods can also be extended to process batch processes [39, 30].

Due to the high sampling rates in industrial processes, redundant and collinear information may be involved in successive sampled data, and modeling and monitoring with these data directly may result in sub-optimal performance. In this work, inspired by the work of dynamic weighted CCA (DWCCA) [217], a dynamic weighted PLS (DWPLS) method is proposed to deal with the redundant information in neighboring samples. DWPLS reduces the dependency in adjacent samples by designing a series of basis functions such as

polynomial and Gaussian bases. In the inner model, the quality scores are represented by a lagged vector auto-regressive model. To maintain the consistency in dynamic inner and outer models, the outer model is controlled to maximize the covariance between the output latent scores and a weighted combination of lagged input scores which is built by a weighted summation of basis functions. The monitoring framework for DWPLS is also designed to cope with anomaly detection. The effectiveness of DWPLS over DiPLS is substantiated through the Tennessee Eastman process (TEP) and a distillation process in terms of process modeling and anomaly detection performance. It is noted that the idea of basis functions can also be applied to other algorithms such as state space methods and slow feature analysis methods [174].

The rest of this chapter is arranged in the following organization. Firstly, the static PLS and DiPLS algorithms are briefly reviewed. Then the proposed DWPLS algorithm is developed in the following section. The design of the basis function, the objective of DWPLS, and its procedure are presented in detail. Further, a process monitoring scheme based on DWPLS along with the fault detection indices are shown. The case studies' section discusses the effectiveness of the algorithm based on the Tennessee Eastman process and a distillation process. Both case studies include the comparison of prediction performance between DiPLS and DWPLS with different basis functions, the influence of the selection of parameter points, and the effectiveness of the predictive monitoring schemes. Conclusions are summarized in the last section.

3.3 Prelimilary

3.3.1 Static Partial Least Squares

PLS algorithm was proposed by Wold et al. [177] to extract a series of latent variables from process and quality variables which can maximize the covariance between latent variables, and handle collinearity issues in the data. Define $\mathbf{Y} \in \mathbb{R}^{n\times q}$ and $\mathbf{X} \in \mathbb{R}^{n\times m}$ as the input and output data respectively, where n is the number of samples, q is the number of quality variables and m is the number of process variables. Then the objective of PLS is [63, 176]

$$\begin{aligned} \max_{\mathbf{q},\mathbf{w}} \quad & J = \mathbf{q}^\top \mathbf{Y}^\top \mathbf{X}\mathbf{w} \\ \text{s.t.} \quad & \|\mathbf{w}\| = 1, \ \|\mathbf{q}\| = 1 \end{aligned} \tag{3.1}$$

where $\mathbf{w}$ and $\mathbf{q}$ are the weighting vectors of $\mathbf{X}$ and $\mathbf{Y}$ respectively. Eq. (3.1) is referred to as the outer modeling, whereas latent score vectors $\mathbf{u}$ and $\mathbf{t}$ can be defined as $\mathbf{u} = \mathbf{Y}\mathbf{q}$ and $\mathbf{t} = \mathbf{X}\mathbf{w}$. The details of the algorithm can be found in Wold's work [175].

For calculating the inner model, a linear regression model is proposed to build the relation between $\mathbf{t}$ and $\mathbf{u}$ as follows

$$\mathbf{u} = b\mathbf{t} + \mathbf{r} \tag{3.2}$$

The effect of the extracted components can be removed from $\mathbf{X}$ and $\mathbf{Y}$ by deflating them as follows

$$\begin{aligned} \mathbf{X} &:= \mathbf{X} - \mathbf{t}\mathbf{p}^\top \\ \mathbf{Y} &:= \mathbf{Y} - b\mathbf{t}\mathbf{q}^\top \end{aligned} \tag{3.3}$$

where $\mathbf{p} = \frac{\mathbf{X}^\top \mathbf{t}}{\mathbf{t}^\top \mathbf{t}}$. From the deflated $\mathbf{X}$ and $\mathbf{Y}$, subsequent latent variables can be extracted until enough latent variables have been extracted.

Then, $\mathbf{X}$ and $\mathbf{Y}$ can be expressed as

$$\begin{aligned} \mathbf{X} &= \mathbf{T}\mathbf{P}^\top + \mathbf{E} \\ \mathbf{Y} &= \mathbf{T}\mathbf{B}\mathbf{Q}^\top + \mathbf{F} \end{aligned} \tag{3.4}$$

where $\mathbf{P} \in \mathbb{R}^{m\times l} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_l]$, $\mathbf{T} \in \mathbb{R}^{n\times l} = [\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_l]$, $\mathbf{Q} \in \mathbb{R}^{p\times l} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_l]$, $\mathbf{B} = \mathrm{diag}[b_1, b_2, \cdots, b_l]$, and $\mathbf{E}$ and $\mathbf{F}$ are the residuals of input and output, respectively. Hold-out and cross-validation are two normal methods to determine the parameters of the algorithm.

For the purpose of online prediction, the corresponding output $\hat{\mathbf{y}}$ can be calculated from the new sample $\mathbf{x}$ as follows

$$\hat{\mathbf{y}} = \mathbf{Q}\mathbf{R}^\top \mathbf{x} \tag{3.5}$$

where $\mathbf{W} \in \mathbb{R}^{m\times l} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_l]$, and $\mathbf{R} \in \mathbb{R}^{m\times l}$ can be expressed as $\mathbf{R} = \mathbf{W}(\mathbf{P}^\top \mathbf{W})^{-1}$.

3.3.2 Dynamic Inner Partial Least Squares

It is obvious that the aforementioned PLS can achieve good performance in static processes only, which, however, rarely exist in industries. In order to extract the dynamics between process scores t_k and quality scores u_k sampled at time k, a dynamic relation is assumed as follows

$$u_k = \beta_0 t_k + \beta_1 t_{k-1} + \cdots + \beta_s t_{k-s} + r_k \tag{3.6}$$

where $u_k = \mathbf{y}_k^\top \mathbf{q}$, $t_k = \mathbf{x}_k^\top \mathbf{w}$, and s is the time lag.

Accordingly, the objective of the outer model is represented as

$$\begin{aligned} &\max_{\mathbf{q},\mathbf{w},\boldsymbol{\beta}} \quad \mathbf{q}^\top \mathbf{Y}_s^\top \mathbf{Z}_s[\boldsymbol{\beta} \otimes \mathbf{w}] \\ &\text{s.t.} \quad \|\mathbf{q}\| = 1, \ \|\boldsymbol{\beta}\| = 1, \ \|\mathbf{w}\| = 1 \end{aligned} \tag{3.7}$$

where

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{s+N}]^\top \\ \mathbf{Y}_s &= [\mathbf{y}_s, \mathbf{y}_{s+1}, \cdots, \mathbf{y}_{s+N}]^\top \\ \mathbf{Z}_s &= [\mathbf{X}_s, \mathbf{X}_{s-1}, \cdots, \mathbf{X}_0] \\ \mathbf{X}_i &= [\mathbf{x}_i, \mathbf{x}_{i+1}, \cdots, \mathbf{x}_{i+N}]^\top, \quad i = 0, 1, \cdots, s \\ \boldsymbol{\beta} &= [\beta_0, \beta_1, \cdots, \beta_s]^\top \end{aligned} \tag{3.8}$$

and $\otimes$ is the Kronecker product. It is noted that the dimensions of the data are re-defined as $\mathbf{X} \in \mathbb{R}^{(N+s+1)\times m} = [\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{N+s}]^\top$ and $\mathbf{Y} \in \mathbb{R}^{(N+s+1)\times q} = [\mathbf{y}_0, \mathbf{y}_1, \cdots, \mathbf{y}_{N+s}]^\top$ in Eq. (3.7). When s=0, the model reduces to static PLS. The model parameters $\mathbf{q}$, $\mathbf{w}$ and $\boldsymbol{\beta}$ in Eq. (3.7) can be solved iteratively with Lagrange multipliers. A modification of the dynamic relation given by Eq. (3.6) has been proposed for the inner model of the DiPLS algorithm, whose details can be found in Dong and Qin's work [51]. DiPLS has shown its advantages over other dynamic algorithms, and thus the proposed DWPLS algorithm is developed based on DiPLS.

3.4 Dynamic Weighted Partial Least Squares

Due to the development of advanced sensory devices and other auxiliary technologies, the data sampling rates in modern industrial processes can be set to a large value, leading to a high redundancy and collinearity in adjacent samples. In these cases, dynamic latent variables methods may require very large amount of lagged data to work properly. Therefore, it is necessary to design efficient dynamic algorithms to address the strong collinearity in neighboring samples.

A potential way is to make reasonable and restrictive assumptions about the pattern of the weights $\mathbf{w}$ in Eq. (3.6), since neighboring samples may have similar effects on output scores thus not contributing new information while exacerbating the effect of noise. Thus, a dynamic weighted partial least squares (DWPLS) algorithm is proposed in this chapter to address the aforementioned issue. In DWPLS, each element of $\boldsymbol{\beta}$ in Eq. (3.7), β_z ($z \in \{0, 1, 2, \cdots, s\}$) is modeled with basis functions as follows

$$\beta_z = \sum_{j=0}^{\gamma+1} \phi_j(z) b_j, \quad z \in \{0, 1 \cdots, s\} \tag{3.9}$$

where ϕ_j (j$\in \{0, 1, \cdots, \gamma+1\}$) are selected basis functions which will be further discussed in the following subsection, and $\mathbf{b} = [b_0, b_1, \cdots, b_{\gamma+1}]^\top$ is the new weighted vector for DWPLS. Due to the redundancy in the data, γ is usually smaller than s.

Remark 1 *Eq. (3.9) represents how past and current samples influence current data, so the time variation of basis functions is in the range of* 0 *to* s*, and the samples outside the range of* 0 *to* s *are assumed to have no impact on current data.*

3.4.1 Basis Functions

The linear combination of basis functions can be employed to represent every continuous or discrete function in the function space, and its general form can be represented as [72]

$$y(z|\mathbf{b}) = b_0 + \sum_{j=1}^{\gamma+1} \phi_j(z) b_j \tag{3.10}$$

Many existing basis functions can be employed in Eq. (3.10), such as polynomial basis and Fourier basis. In this subsection, we review and adapt three of the most common bases for DWPLS algorithm.

Polynomial Basis Functions

the polynomial function is very popular in areas such as Cryptography due to its simplicity, and the standard polynomial basis $\boldsymbol{\phi}(z)$ is formed by

$$\{1, z, z^2, \cdots, z^{\gamma+1}\} \tag{3.11}$$

To introduce more flexibility in polynomial-based DWPLS (DWPLS$_{\text{PB}}$), the following formulation of ϕ_j ($j \in \{0, 1, \cdots, \gamma+1\}$) is adopted based on Lagrange interpolating polynomials instead in our work [10]

$$\begin{aligned}
\phi_0(z) &= \frac{(z-z_1)(z-z_2)\cdots(z-z_{\gamma+1})}{(z_0-z_1)(z_0-z_2)\cdots(z_0-z_{\gamma+1})} \\
\phi_1(z) &= \frac{(z-z_0)(z-z_2)\cdots(z-z_{\gamma+1})}{(z_1-z_0)(z_1-z_2)\cdots(z_1-z_{\gamma+1})} \\
&\vdots \\
\phi_{\gamma+1}(z) &= \frac{(z-z_0)(z-z_1)\cdots(z-z_{\gamma})}{(z_{\gamma+1}-z_0)(z_{\gamma+1}-z_1)\cdots(z_{\gamma+1}-z_{\gamma})}
\end{aligned} \tag{3.12}$$

Noting that these polynomials have the property that

$$\begin{aligned}
&\phi_j(z_j) = 1 \quad (j = 0, \cdots, \gamma+1), \\
&\phi_j(z_k) = 0 \quad (j \neq; j = 0, \cdots, \gamma+1, k = 0, \cdots, \gamma+1)
\end{aligned}$$

where z is in the range of $\{0, s\}$ and $\{z_0, z_1, \cdots, z_{\gamma+1}\}$ are the selected parameter points used to control the shape of the polynomial basis functions. The optimal value of γ is selected by trying different values between $[1, s]$ based on the modeling performance of DWPLS.

To satisfy Remark 1, parameter point z_0 is selected as -1 and $z_{\gamma+1}$ is $s+1$, and weighted parameters b_0 and $b_{\gamma+1}$ should be 0. Thus, the weight parameters of DWPLS$_{\text{PB}}$ can be reformulated as

$$\beta_z = \sum_{j=1}^{\gamma} \phi_j(z) b_j = \boldsymbol{\phi}(z)^\top \mathbf{b}, z \in \{0, 1, \cdots, s\} \tag{3.13}$$

where $\boldsymbol{\phi}(z)$=$[\phi_1(z), \cdots, \phi_\gamma(z)]^\top$.

The general guideline to select the parameter points for polynomial basis functions is omitted here due to page limitation. Readers can refer to Zhu's work [217] for details and illustrative examples.

Gaussian Basis Functions

Gaussian basis functions are always employed in machine learning fields, and it can be regarded by itself as a simple neural network [26]. For Gaussian-based DWPLS (DWPLS_{GB}), the following form is adopted as the basis functions in Eq. (3.9)

$$\phi_j(z) = \exp\left(-\frac{(z-\mu_j)^2}{2\sigma_j^2}\right), z \in \{0, 1, \cdots, s\} \tag{3.14}$$

where μ_j and σ_j^2 ($j \in \{1, 2, \cdots, \gamma\}$) are the mean and variance of the corresponding $\phi_j(\alpha)$, and μ_j should be in the range of $[0, s]$. To conform to Remark 1, $\phi_j(-1)$ and $\phi_j(s+1)$ should be set to zero. Then the weighted parameters $\boldsymbol{\beta}$ are valid over the range of $[0, s]$. An illustrative example of Gaussian basis functions is provided in Zhu's work [217].

Bézier Basis Functions

Bézier basis function is constructed based on Bernstein basis polynomial function, and its shape geometry is determined by the configuration of the control points. Because of the flexibility of Bézier basis function, it is widely applied in complex systems. The mathematical form of Bézier basis function is represented by

$$\mathbf{B}(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i \mathbf{P}_i \tag{3.15}$$

where $0 \leqslant t \leqslant 1$, $\binom{n}{i}$ are the binomial coefficients, and $\mathbf{P}_i$ ($i \in [0, n]$) are the control points. In Bézier-based DWPLS (DWPLS_{BB}), the n^{th} Bézier expansion of $\phi_j(z)$ is adapted into

$$\phi_j(z) = \sum_{i=0}^{s+2} \binom{s+2}{i} (1 - \frac{z+1}{s+2})^{s+2-i} (\frac{z+1}{s+2})^i \mathbf{P}_{ij}, \quad z \in \{-1, 0, \cdots, s+1\} \tag{3.16}$$

where the control points $\mathbf{P}_{ij}$ ($i \in [0, s+2]$, $j \in [1, \gamma]$ and $\gamma < s$) should be determined. Given Remark 1, the start point $\mathbf{P}_{0j}$ is set as (-1,0) and the end point $\mathbf{P}_{(s+2)j}$ is (s+1,0), which make the $\phi_j(-1)$ and $\phi_j(s+1)$ equal to zero.

The control points of DWPLS_{BB} are determined by their abscissas and ordinates. In Eq. (3.16), for simplicity, the abscissa of each control point $\mathbf{P}_{ij}$ is selected as $i-1$ ($i \in \{0, 1, \cdots, s+2\}$), and the ordinates of $\mathbf{P}_{ij}$ can be obtained recursively as follows.

a. Select a starting point θ and a step difference d.

b. For $j = 1$, $\mathbf{P}_1 = [\mathbf{P}_{01}, \mathbf{P}_{11}, \cdots, \mathbf{P}_{(s+2)1}]$ is designed as

$$\mathbf{P}_1 = [(-1, 0), (0, \theta), (1, \theta - d), \cdots, (i, \theta - id), \cdots, (s, \theta - sd), (s+1, 0)]$$

where the ordinate is reduced by d for each point when $i \in [1, s+1]$.

c. For $j > 1$, $\mathbf{P}_{ij}$ can be obtained from $\mathbf{P}_{i(j-1)}$ by

   i. keeping the start point, end point and abscissa of every control points unchanged;

   ii. extracting the ordinate of the second last point and setting it as the ordinate of the second point; and

   iii. shifting the ordinates of the rest points backward for one position.

   which is mathematically represented as

$$\mathbf{P}_2 = [(-1,0),(0,\theta - sd),(1,\theta),\cdots,(i,\theta-(i-1)d),\cdots,(s,\theta-(s-1)d),(s+1,0)];$$
$$\vdots$$
$$\mathbf{P}_j = [(-1,0),(0,\theta-(s-j-2)d),\cdots,(j-1,\theta),\cdots,(i,\theta-(i-j+1)d),\cdots,(s,\theta-(s-j+1)d),(s+1,0)];$$
$$\vdots$$
$$\mathbf{P}_\gamma = [(-1,0),(0,\theta-(s-\gamma-2)d),\cdots,(\gamma-1,\theta),\cdots,(i,\theta-(i-\gamma+1)d),\cdots,(s,\theta-(s-\gamma+1)d),(s+1,0)];$$

A simple example is provided here to show the details of $\boldsymbol{\beta}$, in which $s = 8$, $\gamma = 4$, $b_1 = b_2 = b_3 = 1$, and the starting point θ and step difference d of control points are selected as 2 and 0.3 respectively, leading to

$$\mathbf{P}_1 = [(-1,0),(0,2),(1,1.7),(2,1.4),(3,1.1),(4,0.8),(5,0.5),(6,0.2),(7,-0.1),(8,-0.4),(9,0)];$$
$$\mathbf{P}_2 = [(-1,0),(0,-0.4),(1,2),(2,1.7),(3,1.4),(4,1.1),(5,0.8),(6,0.5),(7,0.2),(8,-0.1),(9,0)];$$
$$\mathbf{P}_3 = [(-1,0),(0,-0.1),(1,-0.4),(2,2),(3,1.7),(4,1.4),(5,1.1),(6,0.8),(7,0.5),(8,0.2),(9,0)];$$
$$\mathbf{P}_4 = [(-1,0),(0,0.2),(1,-0.1),(2,-0.4),(3,2),(4,1.7),(5,1.4),(6,1.1),(7,0.8),(8,0.5),(9,0)];$$

The corresponding variations of ϕ_j and $\boldsymbol{\beta}$ are shown in Figure 3.1.

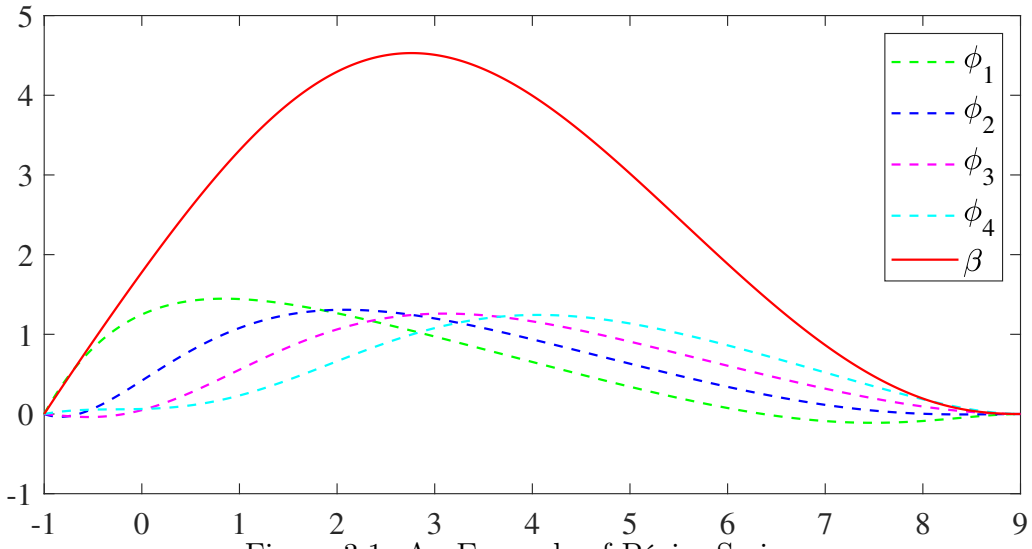


Figure 3.1: An Example of Bézier Series

Remark 2 *For each basis function, the number of control points and the location of control points can be selected thus explaining the flexibility resulting from the use of this basis function. The selection of basis functions depends on the specific process data. Since lagged number s is generally not known in advance, it is necessary to estimate the distribution with different basis functions and varying control points, to choose the best among them.*

Properties of Different Basis Functions for High-Order Interpolation

High-order interpolants are essential in various scientific and engineering applications, and their properties can significantly impact the accuracy and stability of the interpolation process. This subsection explores the distinct properties of stability associated with three commonly used basis function types: polynomial basis functions, Gaussian basis functions, and Bézier basis functions.

- Local vs. Global Support:
  - Polynomial basis functions exhibit global support, meaning they exert influence across the entire interpolation domain.
  - Gaussian basis functions primarily possess local support, affecting only neighboring data points.
  - Bézier basis functions provide local control, influencing regions defined by their control points.
- Control and Precision:
  - Polynomial basis functions offer fine control over interpolation but can exhibit oscillations and susceptibility to Runge's phenomenon [59] for high orders.
  - Gaussian basis functions enable natural control over local data patterns, making them suitable for capturing complex local structures.
  - Bézier basis functions are intuitive for shaping and controlling interpolation but may not guarantee global smoothness.
- Smoothness:
  - The smoothness of polynomial interpolants depends on the polynomial order, with higher orders potentially leading to less smooth interpolations.
  - Gaussian basis functions generally yield smoother interpolants, particularly when neighboring functions overlap appropriately.
  - Bézier interpolants can achieve smoothness through control point adjustments but may not ensure global smoothness.
- Conditioning:
  - Interpolation matrices for high-degree polynomials can be ill-conditioned, introducing numerical instability.

  - Gaussian basis functions often yield well-conditioned interpolation problems, enhancing numerical stability.
  - Bézier interpolation matrices tend to be well-conditioned, promoting numerical stability.

- Computational Efficiency:
  - Computational cost increases significantly for high-order polynomial interpolations.
  - Gaussian basis functions may necessitate specialized techniques like radial basis function (RBF) interpolation, which can be efficient but may require careful parameter tuning.
  - Bézier interpolation is computationally efficient and straightforward for low to moderate orders.

In summary, the selection of basis functions for high-order interpolation should align with the specific requirements of the problem at hand. Polynomial basis functions offer versatility but may suffer from stability concerns at high orders. Gaussian basis functions and Bézier basis functions provide alternative options with distinct characteristics, including local control and smoothness. Thus, the choice should be guided by the inherent properties of the data and the desired characteristics of the interpolant of each basis function.

3.5 The DWPLS Algorithm

In DWPLS, each weighting parameter β_z in Eq. (3.7) is modeled by a weighted sum of basis functions, which is able to be arranged into a matrix form as follows

$$\boldsymbol{\beta} = \boldsymbol{\Phi}^\top \mathbf{b} \tag{3.17}$$

in which $\boldsymbol{\phi}(z) \in \mathbb{R}^\gamma = [\phi_1(z), \phi_2(z), \cdots, \phi_\gamma(z)]^\top$ and $\boldsymbol{\Phi} \in \mathbb{R}^{\gamma \times (s+1)} = [\boldsymbol{\phi}(0), \boldsymbol{\phi}(1), \cdots, \boldsymbol{\phi}(s)]$. Thus, the objective of DWPLS can be expressed as

$$\begin{aligned} &\max_{\mathbf{q},\mathbf{w},\mathbf{b}} \quad \mathbf{q}^\top \mathbf{Y}_s^\top \mathbb{Z}_s [(\boldsymbol{\Phi}^\top \mathbf{b}) \otimes \mathbf{w}] \\ &\text{s.t.} \quad \|\mathbf{q}\| = 1, \ \|\boldsymbol{\Phi}^\top \mathbf{b}\| = 1, \ \|\mathbf{w}\| = 1 \end{aligned} \tag{3.18}$$

where the symbols share the same meanings as in Eq. (3.7), and their denotations are provided here again for better readability.

$$\begin{aligned} &\mathbb{Z}_s \in \mathbb{R}^{N \times (s+1)m} = [\mathbf{X}_s, \mathbf{X}_{s-1}, \cdots, \mathbf{X}_0] \\ &\mathbf{Y}_s \in \mathbb{R}^{N \times p} = [\mathbf{y}_s, \mathbf{y}_{s+1}, \cdots, \mathbf{y}_{s+N}]^\top \\ &\mathbf{X}_i \in \mathbb{R}^{N \times m} = [\mathbf{x}_i, \mathbf{x}_{i+1}, \cdots, \mathbf{x}_{i+N}]^\top \quad (i = 0, 1, \cdots, s) \end{aligned} \tag{3.19}$$

$\mathbf{X}_i \in \mathbb{R}^{N \times m} = [\mathbf{x}_i, \mathbf{x}_{i+1}, \cdots, \mathbf{x}_{i+N}]^\top$ $(i = 0, 1, \cdots, s)$ is a subset of process matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$, $\otimes$ is the Kronecker product, and s is the dynamic order.

It is noted that both auto-correlation and cross-correlation are extracted with Eq. (3.18).

Lagrange multipliers λ_q, λ_w and λ_q are employed to solve the objective in Eq. (3.18) as follows.

$$\mathcal{L} = \mathbf{q}^\top \mathbf{Y}_s^\top \mathbb{Z}_s[(\mathbf{\Phi}^\top \mathbf{b}) \otimes \mathbf{w}] + \frac{1}{2}\lambda_q(1 - \mathbf{q}^\top \mathbf{q}) + \frac{1}{2}\lambda_w(1 - \mathbf{w}^\top \mathbf{w}) + \frac{1}{2}\lambda_q(1 - (\mathbf{\Phi}^\top \mathbf{b})^\top (\mathbf{\Phi}^\top \mathbf{b})) \quad (3.20)$$

For Kronecker product, there exists the following relations

$$(\mathbf{I}_s \otimes \mathbf{w})(\mathbf{\Phi}^\top \mathbf{b}) = [(\mathbf{\Phi}^\top \mathbf{b}) \otimes \mathbf{I}_m]\mathbf{w} = [(\mathbf{\Phi}^\top \mathbf{b}) \otimes \mathbf{w}] \quad (3.21)$$

where $\mathbf{I}$ represents an identity matrix and its subscript represents the dimension.

Taking derivatives of $\mathcal{L}$ with respective to $\mathbf{q}$, $\mathbf{b}$ and $\mathbf{w}$ and setting them as zero lead to

$$\frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathbf{Y}_s^\top \mathbb{Z}_s[(\mathbf{\Phi}^\top \mathbf{b}) \otimes \mathbf{w}] - \lambda_q \mathbf{q} = 0 \quad (3.22)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = [(\mathbf{\Phi}^\top \mathbf{b}) \otimes \mathbf{I}_m]^\top \mathbf{Z}_s^\top \mathbf{Y}_s \mathbf{q} - \lambda_w \mathbf{w} = 0 \quad (3.23)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}} = \mathbf{\Phi}(\mathbf{I}_s \otimes \mathbf{w})^\top \mathbf{Z}_s^\top \mathbf{Y}_s \mathbf{q} - \lambda_b \mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{b} = 0 \quad (3.24)$$

As shown in the above equations, the explicit expressions of $\mathbf{q}$, $\mathbf{w}$ and $\mathbf{b}$ cannot be derived. Thus an iterative approach is adopted to solve the problem as follows.

(1) Initialize $\mathbf{b}$ as a random vector, and normalize it by $\mathbf{b} = \mathbf{b}/\|\mathbf{\Phi}^\top \mathbf{b}\|$.

(2) Calculate $\mathbf{q}$, $\mathbf{w}$ and $\mathbf{b}$ with the respective equations until convergence is achieved.

$$\mathbf{q} = \mathbf{Y}_s^\top \sum_{z=0}^{s} \boldsymbol{\phi}(z)^\top \mathbf{b} \mathbf{X}_{s-z} \mathbf{w}, \quad \mathbf{q} = \mathbf{q}/\|\mathbf{q}\|;$$

$$\mathbf{w} = \sum_{z=0}^{s} \boldsymbol{\phi}(z)^\top \mathbf{b} \mathbf{X}_{s-z}^\top \mathbf{Y}_s \mathbf{q}, \quad \mathbf{w} = \mathbf{w}/\|\mathbf{w}\|;$$

$$\text{Form } \mathbf{T}_s = [\mathbf{X}_s \mathbf{w} \quad \mathbf{X}_{s-1} \mathbf{w} \quad \cdots \quad \mathbf{X}_0 \mathbf{w}];$$

$$\mathbf{b} = (\mathbf{\Phi}\mathbf{\Phi}^\top)^{-1} \mathbf{\Phi} \mathbf{T}_s^\top \mathbf{Y}_s \mathbf{q}, \quad \mathbf{b} = \mathbf{b}/\|\mathbf{\Phi}^\top \mathbf{b}\|.$$

(3) Calculate

$$\begin{aligned} \mathcal{J} &= \mathbf{q}^\top \mathbf{Y}_s^\top \mathbb{Z}_s[(\mathbf{\Phi}^\top \mathbf{b}) \otimes \mathbf{w}] \\ &= \mathbf{q}^\top \mathbf{Y}_s^\top \sum_{z=0}^{s} \boldsymbol{\phi}(z)^\top \mathbf{b} \mathbf{X}_{s-z} \mathbf{w} \end{aligned}$$

After extracting the outer modeling relations, the latent scores $\mathbf{t}$ and $\mathbf{u}$ of the process and quality data are calculated by

$$\begin{aligned}\mathbf{u} &= [u_0\ u_1\ \cdots\ u_{s+N}]^\top = \mathbf{Y}\mathbf{q}\\ \mathbf{t} &= [t_0\ t_1\ \cdots\ t_{s+N}]^\top = \mathbf{X}\mathbf{w}\end{aligned}$$

And the corresponding $\mathbf{u}_s = [u_s\ u_{s+1}\ \cdots\ u_{s+N}]^\top$ and $\mathbf{t}_i = [t_i\ t_{i+1}\ \cdots\ t_{i+N}]^\top$ for $\mathbf{X}_i$ and $\mathbf{Y}_s$ can be represented as

$$\begin{aligned}\mathbf{t}_i &= \mathbf{X}_i\mathbf{w}\\ \mathbf{u}_s &= \mathbf{Y}_s\mathbf{q}\end{aligned} \tag{3.25}$$

where $i = 0, 1, \cdots, s$.

To model the dynamics consistently in outer and inner models, the following linear dynamic regression between output scores and input scores is developed for DWPLS

$$\mathbf{u}_s = \alpha_0\mathbf{t}_s + \alpha_1\mathbf{t}_{s-1} + \cdots + \alpha_s\mathbf{t}_0 + \mathbf{r}_s \tag{3.26}$$

where $\mathbf{r}_s$ is the modeling error of the linear dynamic regression. The modeling parameters $\boldsymbol{\alpha} = [\alpha_0\ \alpha_1\ \cdots\ \alpha_s]^\top$ can be solved with ordinary least squares as follows

$$\boldsymbol{\alpha} = (\mathbf{T}_s^\top\mathbf{T}_s)^{-1}\mathbf{T}_s^\top\mathbf{u}_s \tag{3.27}$$

where $\mathbf{T}_s = [\mathbf{t}_s\ \mathbf{t}_{s-1}\ \cdots\ \mathbf{t}_0]$. Thus, $\mathbf{u}_s$ can be predicted by $\boldsymbol{\alpha}$ and $\mathbf{T}_s$

$$\hat{\mathbf{u}}_s = \mathbf{T}_s\boldsymbol{\alpha} \tag{3.28}$$

With outer and inner modeling, one set of model parameters is extracted, and deflation is conducted on $\mathbf{X}$ and $\mathbf{Y}_s$ to remove their effect, which can be employed to extract the other latent variables. The deflation step can be presented as

$$\begin{aligned}\mathbf{X} &:= \mathbf{X} - \mathbf{t}\mathbf{p}^\top\\ \mathbf{Y}_s &:= \mathbf{Y}_s - \hat{\mathbf{u}}_s\mathbf{q}^\top\end{aligned} \tag{3.29}$$

where $\mathbf{p} = \mathbf{X}^\top\mathbf{t}/\mathbf{t}^\top\mathbf{t}$. Subsequent latent variables can then be extracted with the deflated $\mathbf{X}$ and $\mathbf{Y}_s$. The complete DWPLS algorithm is summarized in Algorithm 1.

With DWPLS, the predicted output $\hat{\mathbf{y}}_k$ at time k can be calculated from the new process data $[\mathbf{x}_k, \mathbf{x}_{k-1}, \cdots, \mathbf{x}_{k-s}]$.

$$\hat{\mathbf{y}}_k = \mathbf{Q}\hat{\mathbf{u}}_k \tag{3.30}$$

where $\hat{\mathbf{u}}_k = \alpha_0\mathbf{t}_k + \alpha_1\mathbf{t}_{k-1} + \cdots + \alpha_s\mathbf{t}_{k-s}$, $\mathbf{t}_k = \mathbf{R}^\top\mathbf{x}_k$, $\mathbf{R} = \mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}$, $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_l]$, $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_l]$, $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_l]$, and $\hat{\mathbf{u}}_k$ can be obtained from $\mathbf{T}$ based on Eq. (3.28).

Algorithm 1 The DWPLS Algorithm

Initialize $\mathbf{X}$ and $\mathbf{Y}$ to unit variance and zero-mean.
Choose the expression of basis functions ϕ_j.
Set $\mathbf{b}$ as a random vector, normalize $\mathbf{b}$ by $\mathbf{b}/\|\mathbf{\Phi}^\top\mathbf{b}\|$, and set $\mathbf{u}_s$ as the first column of $\mathbf{Y}_s$.
Choose model parameters with hold-out or cross-validation method.
Select the corresponding parameter points to calculate the weighted matrix $\mathbf{\Phi}$ in Eq. (3.17).
for $k = 1$ to l **do**
  while $\mathbf{u}_s$ has not converged **do**
    $\mathbf{w} = \sum_{z=0}^{s} \boldsymbol{\phi}(z)^\top \mathbf{b}\mathbf{X}_{s-z}^\top \mathbf{u}_s$, where $\boldsymbol{\phi}(z)=[\phi_1(z), \cdots, \phi_\gamma(z)]^\top$, and normalize $\mathbf{w}$ by $\mathbf{w} = \mathbf{w}/\|\mathbf{w}\|$;
    $\mathbf{t} = \mathbf{X}\mathbf{w}$, and form $\mathbf{T}_s = [\mathbf{t}_s \ \mathbf{t}_{s-1} \ \cdots \ \mathbf{t}_0]$, where $\mathbf{t}_{s-z}$ $(z = 0, 1, \cdots, s)$ can be obtained with Eq. (3.25) ;
    $\mathbf{q} = \mathbf{Y}_s^\top \sum_{z=0}^{s} \boldsymbol{\phi}(z)^\top \mathbf{b}\mathbf{t}_{s-z}$, and normalize $\mathbf{q}$ by $\mathbf{q} = \mathbf{q}/\|\mathbf{q}\|$;
    $\mathbf{u}_s = \mathbf{Y}_s\mathbf{q}$;
    $\mathbf{b} = (\mathbf{\Phi}\mathbf{\Phi}^\top)^{-1}\mathbf{\Phi}\mathbf{T}_s^\top\mathbf{u}_s$, and normalize $\mathbf{b}$ by $\mathbf{b}/\|\mathbf{\Phi}^\top\mathbf{b}\|$;
  end while
  Solve $\boldsymbol{\alpha}$ in the inner model by $\boldsymbol{\alpha} = (\mathbf{T}_s^\top\mathbf{T}_s)^{-1}\mathbf{T}_s^\top\mathbf{u}_s$;
  Calculate the predicted output scores $\hat{\mathbf{u}}_s = \mathbf{T}_s\boldsymbol{\alpha}$;
  Deflate $\mathbf{X}$ and $\mathbf{Y}_s$ by $\mathbf{X} = \mathbf{X} - \mathbf{t}\mathbf{p}^\top$ and $\mathbf{Y}_s = \mathbf{Y}_s - \hat{\mathbf{u}}_s\mathbf{q}^\top$, where $\mathbf{p} = \frac{\mathbf{X}^\top\mathbf{t}}{\mathbf{t}^\top\mathbf{t}}$.
end for

3.6 Process and Quality Monitoring with DWPLS

DWPLS decomposes the input space into the dynamic principal subspace and the residual subspace. The variations in the dynamic principal subspace are related to the output, while the information contained in the residual subspace is quality-irrelevant.

Mathematically, the input sample $\mathbf{x}_k$ is decomposed by DWPLS into $\hat{\mathbf{x}}_k$ in the dynamic principal subspace and $\tilde{\mathbf{x}}_k$ in the residual subspace.

$$\mathbf{x}_k = \hat{\mathbf{x}}_k + \tilde{\mathbf{x}}_k \tag{3.31}$$

where $\hat{\mathbf{x}}_k = \mathbf{P}\mathbf{R}^\top\mathbf{x}_k = \mathbf{P}\mathbf{t}_k$ and $\tilde{\mathbf{x}}_k = (\mathbf{I} - \mathbf{P}\mathbf{R}^\top)\mathbf{x}_k$.

Assuming that the normal data follows a Gaussian distribution, the principal component subspace can be monitored with T^2 index through $\mathbf{t}_k$, while Q index can be employed to detect anomalies in the residual subspace [74]. The T^2 index is defined as

$$T^2 = \mathbf{t}_k^\top\mathbf{\Lambda}^{-1}\mathbf{t}_k = \mathbf{x}_k^\top\mathbf{R}\mathbf{\Lambda}^{-1}\mathbf{R}^\top\mathbf{x}_k \tag{3.32}$$

where $\mathbf{\Lambda} = \frac{1}{(N+s+1)-1}\mathbf{T}^\top\mathbf{T}$, and it contains the main eigenvalues of the training data. T^2 can be approximated by an F distribution with l and $N - l$ degrees of freedom as follows

$$\frac{N(N-l)}{l(N^2-1)}T^2 \sim F_{l,N-l,\alpha} \tag{3.33}$$

For a given significance level α, the corresponding control limit T_α^2 of T^2 be obtained by

$$T_\alpha^2 = \frac{l((N+1)^2-1)}{(N+1)((N+1)-l)} F_{l,N-l,\alpha} \tag{3.34}$$

Then if the T^2 index is below the control limit, the sample is regarded as normal; otherwise, the sample is faulty.

On the other hand, Q index is designed to detect the faults which are irrelevant to the output, and it can be calculated with $\tilde{\mathbf{x}}_k$.

$$Q = \|\tilde{\mathbf{x}}_k\|^2 = \mathbf{x}_k^\top (\mathbf{I} - \mathbf{P}\mathbf{R}^\top)^\top (\mathbf{I} - \mathbf{P}\mathbf{R}^\top)\mathbf{x}_k \tag{3.35}$$

Jackson and Mudholkar [75] proposed a method to calculate the upper control limit δ_α^2 for Q index, which is

$$\delta_\alpha^2 = \theta_1 \left(\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right)^{1/h_0} \tag{3.36}$$

where

$$\begin{aligned} \theta_i &= \sum_{j=l+1}^{m} \lambda_j^i, \qquad i = 1, 2, 3 \\ h_0 &= 1 - \frac{2\theta_1 \theta_3}{3\theta_2^2} \end{aligned} \tag{3.37}$$

and c_α is the normal deviate related with confidence level. If the Q index of the sample $\mathbf{x}_k$ is smaller than δ_α^2, it is considered as normal; otherwise, a quality-irrelevant fault is detected with confidence level $(1-\alpha) \times 100\%$.

3.7 Case Studies

3.7.1 Case Study on TEP

The Tennessee Eastman process (TEP) [54] is a benchmark process designed by Eastman Chemical company which reflects many typical characteristics of controlled chemical processes, and it can be used to evaluate various process monitoring and control technologies. The following reactions occur in the process.

$$\begin{aligned} \mathbf{A}(\mathrm{g}) + \mathbf{C}(\mathrm{g}) + \mathbf{D}(\mathrm{g}) &\rightarrow \mathbf{G}(\mathrm{liq}) \\ \mathbf{A}(\mathrm{g}) + \mathbf{C}(\mathrm{g}) + \mathbf{E}(\mathrm{g}) &\rightarrow \mathbf{H}(\mathrm{liq}) \\ \mathbf{A}(\mathrm{g}) + \mathbf{E}(\mathrm{g}) &\rightarrow \mathbf{F}(\mathrm{liq}) \\ 3\mathbf{D}(\mathrm{g}) &\rightarrow 2\mathbf{F}(\mathrm{liq}) \end{aligned}$$

where two liquid products **G** and **H** and a liquid byproduct **F** are produced by four gas reactants **A**, **C**, **D** and **E**, and **B** is an inert. The diagram and details of TEP are described in Downs and Vogel's work [54].

In the following subsections, TEP is employed to demonstrate the modeling and monitoring performance of the proposed DWPLS model.

Modelling Performance Comparison

The available data of TEP can be separated into two blocks, which are 12 manipulated variables XMV(1-12) and 41 measurements XMEAS(1-41). In this case study, 11 manipulated variables XMV(1-11) and 22 measurements XMEAS(1-22) are selected as process variables, which are sampled every 3 minutes. XMEAS(38) is used as a quality variable, and it is sampled every 15 minutes. In the original data, due to the low sampling rate, XMEAS(38) is repeated five times to match the sampling rates of process variables. Thus, for a fair comparison, only 1/5 of the data points are used as training data in our work.

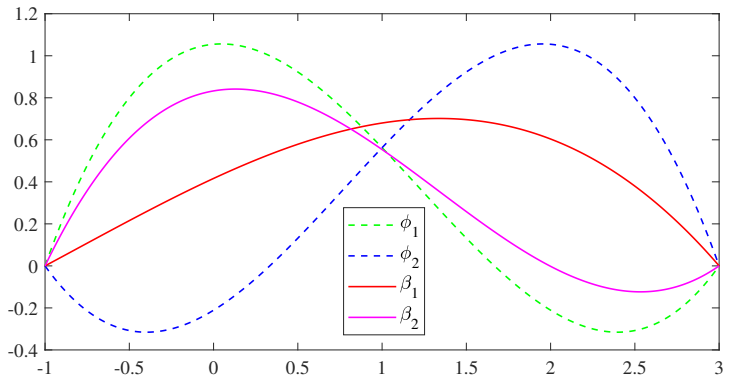
Table 3.1: Selected Model Parameters for TEP

Algorithm	l	s	γ	Parameter Points
DiPLS	2	7	-	-
DWPLS_{PB}	2	2	2	$\mathbf{z} = [-1; 0.33; 1.67; 3]^\top$
DWPLS_{GB}	2	2	2	$\boldsymbol{\mu}$=[0.5;1.5]; $\boldsymbol{\sigma}^2$=[0.67;1]
DWPLS_{BB}	2	2	2	$\mathbf{P}_1$=[(-1,0);(0,2);(1,1.9);(2,1.8); (3,0)]; $\mathbf{P}_2$=[(-1,0);(0,1.8);(1,2);(2,1.9); (3,0)]

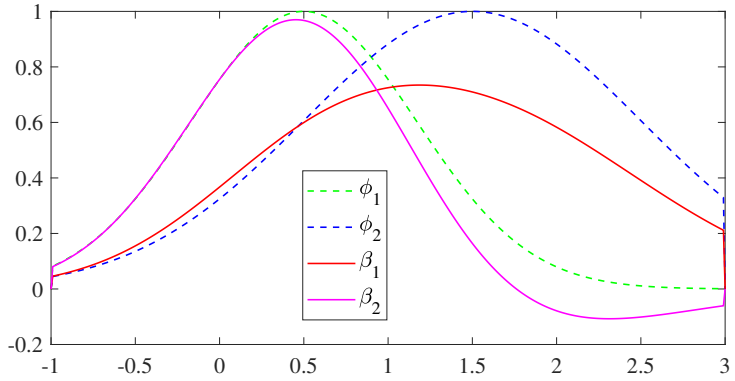
Table 3.2: MSEs of Different Models for TEP

	DiPLS	DWPLS_{PB}	DWPLS_{GB}	DWPLS_{BB}
MSEs	0.682	0.6182	0.6239	0.6212

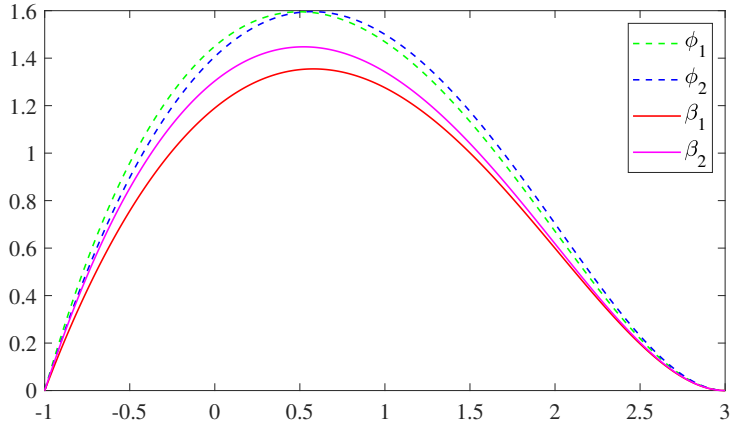
The modeling performance of DiPLS, DWPLS_{PB}, DWPLS_{GB} and DWPLS_{BB}, where the latter three refer to Polynomial, Gaussian basis and Bézier basis functions respectively, are compared in this subsection. The parameter points, lag number and number of latent variables of each algorithm are selected through the hold-out method, which is summarized in Table 3.1. The details of the selected basis functions and the first two constructed latent variables are plotted in Figure 3.2 for DWPLS_{PB}, DWPLS_{GB} and DWPLS_{BB} respectively. Their mean squared errors (MSEs) are presented in Table 3.2. As shown in the table, DWPLS obtains better prediction performance than DiPLS. Among the variants of DWPLS, DWPLS_{PB} achieves the best performance, and its prediction is shown in Figure 3.3. While the variations in the actual process seem very noisy, the variations generated by using the Gaussian basis functions are relatively simplistic. Regarding the use of Bézier bases, the number of control points will influence the flexibility of their shape, and its performance is constrained by the small value of s. Compared with the previous two bases, polynomial bases are more flexible thus providing the best modeling performance in this case.



(a) Polynomial Series



(b) Gaussian Series



(c) Bézier Series

Figure 3.2: Selected Series and Weight Variations of the First Two Latent Variables for TEP

Influence of Parameter Points

Since polynomial basis functions and Bézier basis functions are more flexible and their modeling power is better than Gaussian basis functions, we use them as an example to explore how parameter points influence the modeling performance of DWPLS.

The weighted parameters $\mathbf{b}$ of DWPLS$_{\text{PB}}$ are determined by the parameter points in

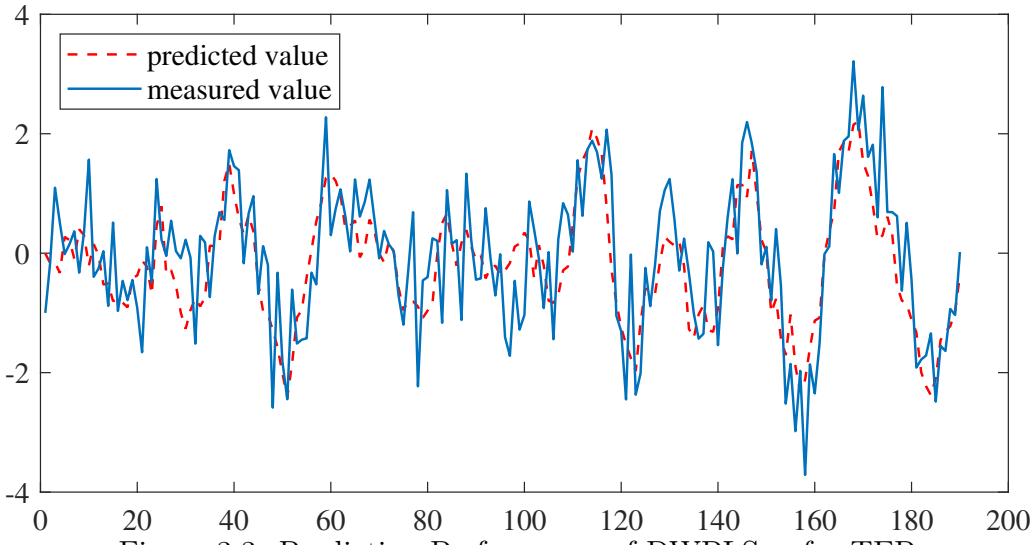


Figure 3.3: Prediction Performance of DWPLS_{PB} for TEP

$\mathbf{z}$. Thus, the selection of the parameter points is very important. Several distributions are tested in this case study, such as distributing them evenly, closer to -1, and closer to $s+1$. The details of how the distribution of $\mathbf{z}$ affects the modeling performance are shown in Table 3.3. As observed from the table, distributing $\mathbf{z}$ evenly leads to the best performance of DWPLS_{PB}.

As discussed in previous sections, the control points of DWPLS_{BB} are determined by their abscissas and ordinates. Different step differences of ordinates between two neighboring points are tested in this case study, and the details are shown in Table 3.4. The results from the table show that DWPLS_{BB} performs better when the step difference of ordinates between two neighboring control points is small.

Table 3.3: The Effect of Parameter Points of DWPLS_{PB} on Prediction Performance

Distribute $\mathbf{z}$	l	s	γ	Parameter Points	MSEs
more close to -1	2	2	2	$\mathbf{z}$=[-1;-0.33;0.33;3]	0.7146
evenly	2	2	2	$\mathbf{z}$=[-1;0.33;1.67;3]	0.6182
more close to $s+1$	2	2	2	$\mathbf{z}$=[-1;1.67;2.33;3]	0.6338

Monitoring Performance on TEP

Since DWPLS_{PB} performs best in terms of modeling, it is adopted in this subsection to illustrate the monitoring performance.

15 known disturbances in TEP are tested in this case study [54]. Based on the PCA-based quality monitoring method, the 15 disturbances can be divided into two subgroups: quality-related faults and quality-irrelevant disturbances. The threshold is selected as 15%: if the proportion of faulty samples exceeds 15%, the disturbance is regarded as quality relevant one; otherwise, it is quality irrelevant. The separation result and their

Table 3.4: The Effect of Control Points of DWPLS_{BB} on Prediction Performance

d	l	s	γ	Parameter Points	MSEs
0.1	2	2	2	$\mathbf{P}_1$=[(-1,0);(0,2);(1,1.9);(2,1.8); (3,0)]; $\mathbf{P}_2$=[(-1,0);(0,1.8);(1,2);(2,1.9); (3,0)]	0.6212
0.7	2	2	2	$\mathbf{P}_1$=[(-1,0);(0,2);(1,1.3);(2,0.6); (3,0)]; $\mathbf{P}_2$=[[(-1,0);(0,0.6);(1,2);(2,1.3); (3,0)]];	0.6247
0.9	2	2	2	$\mathbf{P}_1$=[(-1,0);(0,2);(1,1.1);(2,0.2); (3,0)]; $\mathbf{P}_2$=[(-1,0);(0,0.2);(1,2);(2,1.1); (3,0)];	0.6273
1.1	2	2	2	$\mathbf{P}_1$=[(-1,0);(0,2);(1,0.9);(2,-0.2); (3,0)]; $\mathbf{P}_2$=[(-1,0);(0,-0.2);(1,2);(2,0.9); (3,0)];	0.6323

Table 3.5: FDRs and FARs for Quality-Relevant Faults with DiPLS and DWPLS_{PB}

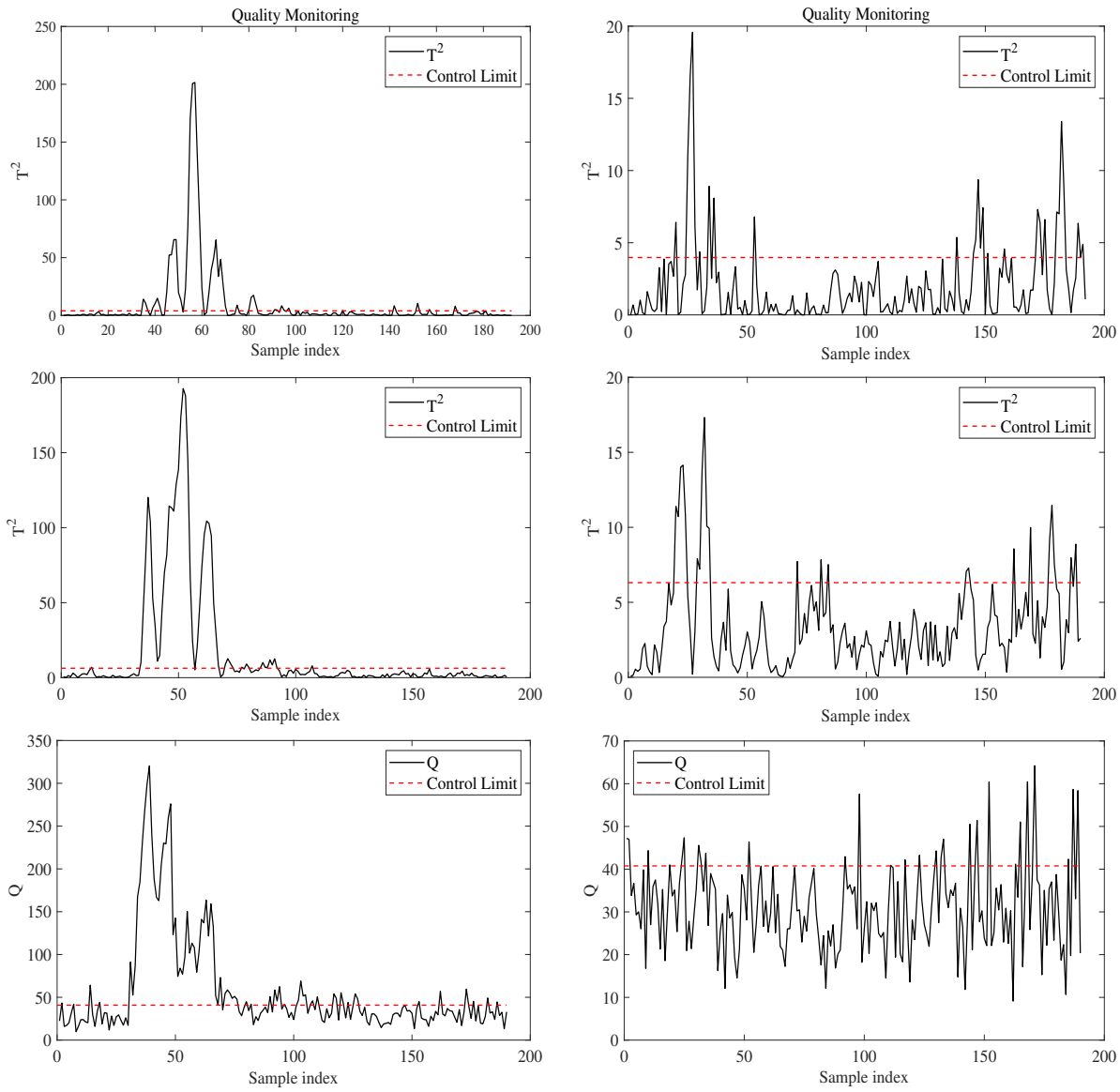
Fault Type	FDRs(DiPLS)	FDRs(DWPLS_{PB})	FARs(DiPLS)	FARs(DWPLS_{PB})
IDV(1)	91.78	86.3	66.67	69.23
IDV(2)	96.3	100	76.69	74.23
IDV(5)	77.27	70.45	8.22	13.01
IDV(6)	100	100	17.95	17.95
IDV(7)	91.55	83.1	39.5	26.89
IDV(8)	85.04	85.83	41.27	49.21
IDV(10)	58.9	67.12	29.06	28.21
IDV(12)	88.55	92.37	44.07	45.76
IDV(13)	96.6	97.96	13.95	18.6
Average	87.33	88.01	37.49	38.12

Table 3.6: FARs for Quality-Irrelevant Disturbances with DiPLS and DWPLS_{PB}

Fault Type	FARs(DiPLS)	FARs(DWPLS_{PB})
IDV(0)	3.78	8.11
IDV(3)	9.32	9.32
IDV(4)	6.98	5.81
IDV(9)	6.75	9.2
IDV(11)	4.94	12.35
IDV(14)	5.78	16.76
IDV(15)	4.71	9.41
Average	6.04	9.13

corresponding fault detection rates (FDRs) and false alarm rates (FARs) of DWPLS_{PB} and DiPLS are summarized in Table 3.5 and Table 3.6. As shown from the tables, DWPLS_{PB} achieves comparable monitoring results with DiPLS, with slightly better fault detection rates.

For better visualization, IDV(5) and IDV(9) are chosen as examples to further illustrate the monitoring performance of DWPLS_{PB}. The monitoring results are presented in Figure 3.4, where for each disturbance, the top plot shows the quality monitoring results by performing PCA on the quality data, and the bottom two plots are the monitoring results obtained with DWPLS_{PB}. IDV(5) introduces a step change in condenser cooling water



(a) IDV(5) (b) IDV(9)

Figure 3.4: Fault Detection Results with DWPLS_{PB}

inlet temperature, and as shown in the quality monitoring results of IDV(5) in the top plot of Figure 3.4a, the disturbance will affect quality variable XMEAS(38) transiently. DWPLS_{PB} based quality relevant monitoring successfully captures the trend and identifies it as a transient quality-relevant fault case.

IDV(9) is a random disturbance to D feed temperature in Stream 2 [54], and as shown in the top plot of Figure 3.4b, this disturbance will not affect quality variables in most cases. The monitoring results of DWPLS_{PB} match well with the quality monitoring results of PCA.

3.7.2 Case Study on A Distillation Process

In this case study, data collected from a distillation process is employed to compare the performance of DiPLS and DWPLS further [23]. Figure 3.5 is the sketch of the process, and three key distillation columns are involved, which are the primary column, feed column, and secondary column. Since the primary column is of higher importance than the other two columns, the 22 variables in the primary column are selected as the input, and the make/reflux ratio is the output in our comparison.

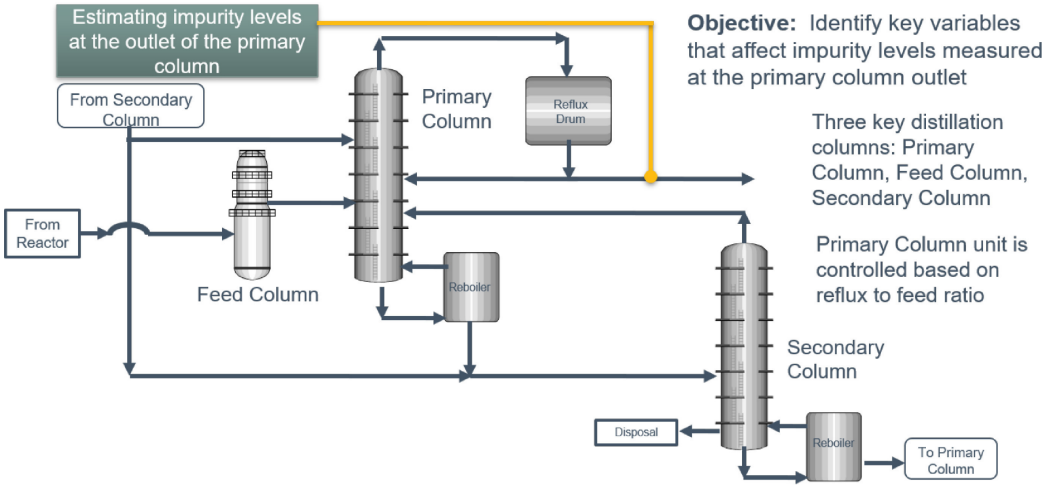


Figure 3.5: Block Diagram of the Distillation Process

DiPLS, DWPLS_{PB}, DWPLS_{GB} and DWPLS_{BB} are applied on the collected dataset, and the selected parameters and their prediction performance in terms of MSEs are summarized in Table 3.7. As shown in the table, all variants of DWPLS have better performance than DiPLS with smaller MSEs, with DWPLS_{PB} performing the best, which is consistent with the observations from the TEP case study. The prediction performance of DWPLS_{PB} is shown in Figure 3.6. Different from the case study of TEP, DWPLS_{GB} performs better than DWPLS_{BB} in this case. This difference shows that different kinds of basis functions will match the properties of different data/processes. Thus, as stated in Remark 2, different basis functions should be tested to estimate the distribution of the data, to choose the optimal one.

Table 3.7: MSEs of Different Models for the Distillation Process

Algorithm	l	s	γ	MSEs
DiPLS	15	25	-	0.2496
DWPLS_{PB}	15	14	2	0.2345
DWPLS_{GB}	15	14	2	0.2353
DWPLS_{BB}	15	15	5	0.2417

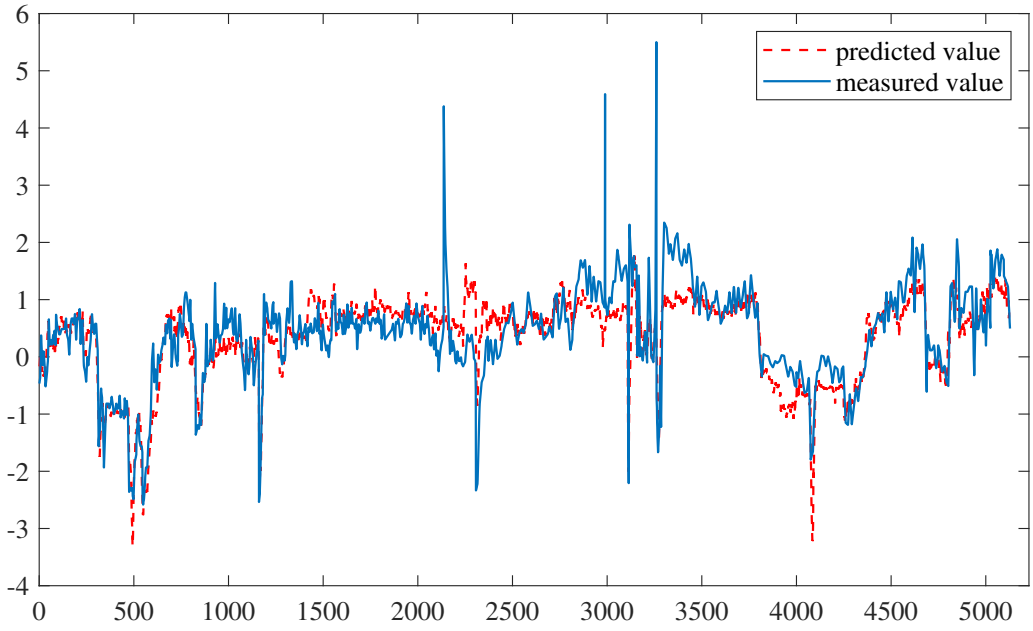


Figure 3.6: Prediction Performance of DWPLS_{PB} for the Distillation Process

3.8 Summary

In this chapter, a dynamic weighted PLS is proposed to capture the dynamic relations between process variables and quality variables. Since DWPLS models the weight parameters with basis functions such as polynomial, Gaussian, and Bézier bases, it can handle the redundancy in successive samples and achieves better modeling performance which is demonstrated through the Tennessee Eastman process and a distillation process. DWPLS-based quality-relevant monitoring scheme is also developed, whose comparable detection performance relative to DiPLS is illustrated through the Tennessee Eastman process. Based on the properties of DWPLS, it will be more sensitive to the quality-relevant dynamic relations existing in the process. Thus, it performs better for the obvious quality-relevant faults but it demonstrates comparatively diminished proficiency in detecting faults of a subtler nature, where the quality-related implications are less overt. The selection of parameter points for different bases and their effects are also demonstrated in this work.

Chapter 4

Dynamic Auto-Regressive Latent Variable Models

4.1 Overview

[1] To fully exploit the auto-correlations among quality variables, a novel dynamic auto-regressive latent variable model (DALVM) is proposed for capturing both auto and cross-correlations from high-dimensional time series data. DALVM is designed to maximize the covariance between the current quality score and the weighted sum of past quality and process scores, and an auto-regressive exogenous inner model is developed for consistency purposes. Furthermore, a concurrent anomaly detection system is developed based on DALVM, referred to as ConDALVM, which conducts subsequent decompositions in the extracted latent spaces. ConDALVM realizes comprehensive monitoring for both static and dynamic anomalies in process and quality spaces. The superiority of the proposed methods is demonstrated through a numerical simulation and two industrial case studies.

4.2 Introduction

Over past decades, supervised learning algorithms play an important role when quality data are difficult to obtain in a timely manner[28, 40, 66]. Among them, considering the high-dimensional and highly auto and cross-correlated characteristics of industrial data, multivariate statistical analysis methods are widely studied and applied in the process systems engineering community due to their additional ability of dimension reduction. The capability of these techniques in improving operating safety and production efficiency [63, 69, 216] has been shown in different studies. Some traditional multivariate methods include partial least squares (PLS) [63], latent variable regression (LVR) [17], and canonical correlation analysis (CCA) [158].

[1] Adapted from Xu, Bo, & Zhu, Q. (2021). Concurrent auto-regressive latent variable model for dynamic anomaly detection. Journal of Process Control, 108, 1-11

PLS extracts the latent structure by maximizing the covariance between process and quality latent scores. Li et al. [95] explored the geometric properties of PLS to perform quality-relevant and quality-irrelevant monitoring. Since the latent scores extracted by PLS still contain quality-irrelevant variations, the variance of the residuals may be very large thus leading to inaccurate fault detection. To overcome this issue, corresponding total and concurrent monitoring strategies were proposed to decompose process spaces into quality relevant and irrelevant sub-spaces to achieve more accurate monitoring performance [214, 132, 57, 200, 208]. It is noted that input/output variables and process/quality variables are used interchangeably in this paper.

Latent variable models rely on an implicit assumption that only static relations exist in the data. However, for real-world industrial processes, the dynamic relations involved in process and quality variables will reduce the credibility and efficiency of these models. Therefore, dynamic extensions were proposed to deal with dynamic systems. Qin and McAvoy [131] and Ricker [137] applied the augmentation method in which the input matrix is combined with a number of lagged values of quality and process variables. However, these methods pay no attention to the structure of the model, and they cannot capture well the dynamic relations well and increase the computation cost unnecessarily.

Alternatively, Qin and McAvoy [130] implemented a neural network to build a dynamic inner model. Lakshminarayanan et al. [88] introduced time series models between process and quality latent variables to build the dynamic inner model. Then Li et al. [94] proposed a novel dynamic PLS to maximize the covariance between the quality score and the weighted lagged process scores. Dong and Qin [51] achieved improved modeling performance by designing consistent dynamic inner and outer models, which is referred to as dynamic inner PLS (DiPLS). An auto-regressive model was also introduced in their models, but the time lags in process and quality scores were constrained to be the same, leading to the degeneration of modeling performance. Dynamic versions of CCA and LVR [224, 49] were also proposed. For example, a dynamic weighted PLS was developed in the current thesis to improve dynamic modeling by addressing redundant information caused by high sampling rates in [183].

In addition to the aforementioned statistical methods, other supervised machine learning methods such as support vector machines (SVM) and k-nearest neighbor (KNN) are also adapted in the process systems engineering field [162, 197]. Through training samples, SVM maximizes the width of the gap between two different kinds of samples, and thus it is widely applied for anomaly detection. Manghnani and Thirumaran [160] improved SVM with a crossover-based binary gravitational search method. Furthermore, to consider the dynamic relations between industrial data, non-linear extensions of SVMs are proposed. To classify non-linear data, several kinds of kernel functions are used in SVM, including polynomial kernel, Radial Basis Function kernel, Sigmoid kernel, and Gaussian kernel [213, 196, 201, 212].

KNN can also be used for classification and regression, and several variants were proposed to improve its modeling performance. To utilize the local class distribution information, it is assumed that some samples contain more important information than other samples. Accordingly, a weighting strategy was proposed for KNN that assigns different

weights to each local class [55]. Local information-based KNN and global information-based KNN have been proposed to fully exploit the information in the collected data [152, 27].

Due to the great potential of neural networks, they are also widely used and adopted for supervised nonlinear and dynamic modeling. For solving certain real-world problems, deep neural networks (DNNs) have exhibited great performance in many fields [92].Several DNN architectures have been proposed, such as stacking automatic encoders, convolution neural networks, deep probabilistic neural networks, and deep fuzzy neural networks [97, 90, 29, 45]. However, due to their complex architecture, the interpretability of DNNs is generally poor which reduces their reliability in industrial practice [53].

To ensure the models' interpretability and improve their reliability, we will focus in this work on the improvement of multivariate statistical methods. Since auto-correlations between the current and past quality scores still remain unexploited in multivariate methods, they will be used for improving the modeling and monitoring performance [219]. In this chapter, we propose a novel dynamic auto-regressive latent variable model (DALVM) to capture the dynamics of quality variables with both the past process samples and the past quality samples. In the outer model, DALVM maximizes the covariance of the quality score and the weighted combination of both past process and quality scores. To align with the dynamic outer model, an auto-regressive exogenous (ARX) model is designed in the inner model to predict the quality scores. Further, to achieve comprehensive static and dynamic monitoring of controlled processes, a concurrent DALVM (ConDALVM) monitoring system is proposed to conduct further decompositions on the extracted latent subspaces. The corresponding monitoring indices are defined for each decomposed subspace, with each serving a particular purpose. A numerical toy example simulation, the Tennessee Eastman process (TEP), and the three-phase flow facility (TPFF) are used to illustrate the efficacy of the proposed algorithm.

The remainder of this chapter is organized as follows. The preliminaries of static PLS and DiPLS are briefly reviewed in Section 4.3. Section 4.4 proposes the dynamic auto-regressive latent variable model and the concurrent DALVM for comprehensive temporal modeling and monitoring. A numerical simulation and two industrial processes are used to verify the superiority of the proposed algorithms in Section 4.5. Finally, Section 4.6 concludes this chapter.

4.3 Preliminaries

4.3.1 Partial Least Squares

Wold et al. [175] firstly proposed PLS to extract the static relations of the process and quality scores by maximizing the covariance between latent variables. Assume that $\mathbf{X} \in \mathbb{R}^{n\times m}$ and $\mathbf{Y} \in \mathbb{R}^{n\times q}$ are the input and output data respectively, where m and q are the number of process and quality, and n is the number of samples. Then the objective of PLS

is [63]

$$\begin{aligned} \max_{\mathbf{q},\mathbf{w}} \quad & J = \mathbf{q}^\top \mathbf{Y}^\top \mathbf{X}\mathbf{w} \\ \text{s.t.} \quad & \|\mathbf{w}\| = 1, \|\mathbf{q}\| = 1 \end{aligned} \tag{4.1}$$

where $\mathbf{w}$ and $\mathbf{q}$ are the process and quality weighting vectors respectively, $\mathbf{t} = \mathbf{X}\mathbf{w}$ is the latent score vector of process variables and $\mathbf{u} = \mathbf{Y}\mathbf{q}$ is the score vector for quality variables.

Then a linear regression model is constructed between quality and process scores in the inner model by

$$\mathbf{u} = b\mathbf{t} + \mathbf{r} \tag{4.2}$$

where $\mathbf{r}$ is the regression error. With iterative procedure, $\mathbf{X}$ and $\mathbf{Y}$ can be decomposed as

$$\begin{aligned} \mathbf{X} &= \mathbf{T}\mathbf{P}^\top + \mathbf{E} \\ \mathbf{Y} &= \mathbf{T}\mathbf{B}\mathbf{Q}^\top + \mathbf{F} \end{aligned} \tag{4.3}$$

where $\mathbf{T} \in \mathbb{R}^{n\times l} = [\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_l]$, $\mathbf{P} \in \mathbb{R}^{m\times l} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_l]$, $\mathbf{Q} \in \mathbb{R}^{q\times l} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_l]$, $\mathbf{B} = \text{diag}[b_1, b_2, \cdots, b_l]$, l is the number of latent variables and $\mathbf{E}$ and $\mathbf{F}$ are the residuals of process and quality variables respectively.

In online modeling, quality variables $\hat{\mathbf{y}}$ can be predicted from a new sample $\mathbf{x}$ by

$$\hat{\mathbf{y}} = \mathbf{Q}\mathbf{R}^\top \mathbf{x} \tag{4.4}$$

where $\mathbf{R} \in \mathbb{R}^{m\times l} = \mathbf{W}(\mathbf{P}^\top \mathbf{W})^{-1}$ [41, 43].

4.3.2 Dynamic-Inner Partial Least Squares

The traditional PLS only extracts static relations between $\mathbf{X}$ and $\mathbf{Y}$, and it is not suitable for real-world dynamic industrial processes. Qin et al. [127] summarized the dynamic variants of PLS, and among these methods, DiPLS obtains better modeling performance because of it has a consistent formulation for dynamic outer and inner models [51].The objective of its outer model is

$$\begin{aligned} \max_{\mathbf{q},\mathbf{w},\boldsymbol{\beta}} \quad & \mathbf{q}^\top \mathbf{Y}_s^\top \mathbb{Z}_s [\boldsymbol{\beta} \otimes \mathbf{w}] \\ \text{s.t.} \quad & \|\mathbf{q}\| = 1, \ \|\boldsymbol{\beta}\| = 1, \ \|\mathbf{w}\| = 1 \end{aligned} \tag{4.5}$$

where

$$\begin{aligned} \mathbb{Z}_s \in \mathbb{R}^{N\times(s+1)m} &= [\mathbf{X}_s, \mathbf{X}_{s-1}, \cdots, \mathbf{X}_0] \\ \mathbf{Y}_s \in \mathbb{R}^{N\times p} &= [\mathbf{y}_s, \mathbf{y}_{s+1}, \cdots, \mathbf{y}_{s+N}]^\top \\ \boldsymbol{\beta} \in \mathbb{R}^{s+1} &= [\beta_0, \beta_1, \cdots, \beta_s]^\top \end{aligned} \tag{4.6}$$

$\mathbf{X}_i \in \mathbb{R}^{N\times m} = [\mathbf{x}_i, \mathbf{x}_{i+1}, \cdots, \mathbf{x}_{i+N}]^\top$ $(i = 0, 1, \cdots, s)$, $\otimes$ is the Kronecker product, and s is the dynamic order. To align with the denotations in Eq. (4.1), we have $n = N + s + 1$.

The consistent dynamic inner model of DiPLS is

$$u_k = \beta_0 t_k + \beta_1 t_{k-1} + \cdots + \beta_s t_{k-s} + r_k \tag{4.7}$$

where r_k is the regression error, and $t_k = \mathbf{x}_k^\top \mathbf{w}$ and $u_k = \mathbf{y}_k^\top \mathbf{q}$ are the principal component projection of $\mathbf{x}_k$ and $\mathbf{y}_k$ respectively, $\mathbf{w}$ and $\mathbf{q}$ have the same meaning as in Eq. (4.1). Consistency is ensured by including the effect of the coefficients of the inner model into the objective function used for teh calculation of the outer model.

Deflation should be performed on $\mathbf{X}$ and $\mathbf{Y}_s$ for the extraction of next sets of latent variables. After the decomposition of DiPLS, the prediction of quality variations is obtained by

$$\hat{\mathbf{y}} = \mathbf{Q}\hat{\mathbf{u}}_s \tag{4.8}$$

where $\hat{\mathbf{u}}_s$ is obtained from $\mathbf{t}$ based on dynamic inner modeling. Additional details about DiPLS can be found in [51].

4.3.3 DiPLS-based Monitoring

After performing DiPLS algorithm, the input space is decomposed into a dynamic principal subspace and a residual subspace. In online modeling, given a new collected sample $\mathbf{x}_k$, it can be decomposed as [47]

$$\begin{aligned}
\mathbf{x}_k &= \hat{\mathbf{x}}_k + \tilde{\mathbf{x}}_k \\
\hat{\mathbf{x}}_k &= \mathbf{P}\mathbf{R}^\top \mathbf{x}_k = \mathbf{P}\mathbf{t}_k \\
\tilde{\mathbf{x}}_k &= (\mathbf{I} - \mathbf{P}\mathbf{R}^\top)\mathbf{x}_k
\end{aligned}$$

where $\hat{\mathbf{x}}_k$ contains the most quality relevant information and $\tilde{\mathbf{x}}_k$ contains the information which is not relevant with quality variables.

It is assumed that the information in the dynamic principal subspace is related to quality variables and the quality-irrelevant variations are contained in the residual subspace. Then T^2 index can be defined as $T^2 = \mathbf{t}_k^\top \mathbf{\Lambda}^{-1} \mathbf{t}_k$ and Q index is designed as $Q = \|\tilde{\mathbf{x}}_k^2\|$. T^2 and Q together with their control limits define the monitoring framework for DiPLS [74].

4.4 Methodology

4.4.1 Problem Statement and Motivation

In real-world industries, product quality is of key importance. However, quality-related variables are often difficult to measure directly online and/or they cannot be measured in a timely manner. On the other hand, supervised machine learning algorithms are widely adopted to predict quality variables. However, since most collected industrial data are

dynamic and thus are in the form of high-dimensional and highly correlated time series, the traditional static methods such as PLS can not extract their dynamics accurately. DiPLS considers the dynamic cross relations between process and quality variables, and it can model the system dynamics to some extent. However, one issue involved in DiPLS is that the collected quality information is not fully exploited as shown in Eq. (4.7), which can provide extra valuable information once they are available. Thus, in addition to the dynamic relations between process and quality scores, the valuable auto-regressive relations among quality data should also be considered for obtaining more accurate modeling performance.

$$\begin{aligned} u_k = \beta_0 t_k \quad & +\beta_1 t_{k-1} + \ldots + \beta_s t_{k-s} \\ & +\delta_1 u_{k-1} + \ldots + \delta_d u_{k-d} + r_k \end{aligned} \tag{4.9}$$

where β_i $(i = 0, 1, \ldots, s)$ and δ_j $(j = 1, \ldots, d)$ are weighted parameters. s and d are the dynamics order for process and quality variables respectively, and where $g = \max(s, d)$. It is noted that different lag numbers are adopted to account for different dynamics in process and quality variables. Based on Eq. (4.9), quality score $\hat{u}_k$ is predicted by

$$\begin{aligned} \hat{u}_k &= \sum_{i=0}^{s} \beta_i \mathbf{x}_{k-i}^\top \mathbf{w} + \sum_{j=1}^{d} \delta_j \mathbf{y}_{k-j}^\top \mathbf{q} \\ &= [\mathbf{x}_k^\top, \mathbf{x}_{k-1}^\top, \ldots, \mathbf{x}_{k-s}^\top] (\boldsymbol{\beta} \otimes \mathbf{w}) \\ &\qquad + [\mathbf{y}_{k-1}^\top, \ldots, \mathbf{y}_{k-d}^\top] (\boldsymbol{\delta} \otimes \mathbf{q}) \end{aligned} \tag{4.10}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{s+1} = [\beta_0, \beta_1, \ldots, \beta_s]^\top$ and $\boldsymbol{\delta} \in \mathbb{R}^d = [\delta_1, \delta_2, \ldots, \delta_d]^\top$.

Eq. (4.10) describes the dynamic relations for a single quality score. To generalize this relation for the whole dataset $\{\mathbf{x}_k\}_1^{N+g+1}$ and $\{\mathbf{y}_k\}_1^{N+g+1}$, the following matrices are defined,

$$\begin{aligned} &\mathbb{Z}_x \in \mathbb{R}^{N \times (s+1)m} = [\mathbf{X}_g, \mathbf{X}_{g-1}, \ldots, \mathbf{X}_{g-s}] \\ &\mathbb{Z}_y \in \mathbb{R}^{N \times (dq)} = [\mathbf{Y}_{g-1}, \mathbf{Y}_{g-2}, \ldots, \mathbf{Y}_{g-d}] \\ &\mathbf{X}_i \in \mathbb{R}^{N \times m} = [\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+N}]^\top, i = \{0, 1, \ldots, g\} \\ &\mathbf{Y}_i \in \mathbb{R}^{N \times q} = [\mathbf{y}_i, \mathbf{y}_{i+1}, \ldots, \mathbf{y}_{i+N}]^\top, i = \{0, 1, \ldots, g\} \end{aligned}$$

The corresponding process and quality scores can be expressed by the above matrices by $\mathbf{t}_i = \mathbf{X}_i \mathbf{w}$ and $\mathbf{u}_i = \mathbf{Y}_i \mathbf{q}$ $(i = \{0, 1, \ldots, g\})$. Then Eq. (4.10) can be formulated into the following matrix form:

$$\hat{\mathbf{u}}_g = \mathbb{Z}_x (\boldsymbol{\beta} \otimes \mathbf{w}) + \mathbb{Z}_y (\boldsymbol{\delta} \otimes \mathbf{q}) \tag{4.11}$$

By involving past and current information about process and quality variables, Eq. (4.11) fully exploits the dynamic information in the collected data.

Algorithm 2 The DALVM Algorithm

1. Initialization: Transform the columns of $\mathbf{X}$ and $\mathbf{Y}$ into zero mean and unit variance, select the first column of $\mathbf{Y}$ as $\mathbf{u}$, and initialize $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ as unit random vectors.

2. Dynamic outer modeling: Based on the outer structure of DALVM and Eqs. (4.14) - (4.17), iterate the following operations to obtain the parameters in the outer model until convergence is achieved.

   i. $\mathbf{X}_\beta = \sum_{i=0}^{s} \beta_i \mathbf{X}_{g-i}$, $\mathbf{Y}_\delta = \sum_{i=1}^{d} \delta_i \mathbf{Y}_{g-i}$, and form $\mathbf{U}_d = [\mathbf{u}_{g-1}, \mathbf{u}_{g-2}, \ldots, \mathbf{u}_{g-d}]$;

   ii. $\mathbf{w} = \mathbf{X}_\beta^\top \mathbf{u}_g$, $\mathbf{w} = \mathbf{w}/\|\mathbf{w}\|$;

   iii. $\mathbf{t} = \mathbf{X}\mathbf{w}$, and construct $\mathbf{T}_s = [\mathbf{t}_g, \mathbf{t}_{g-1}, \ldots, \mathbf{t}_{g-s}]$;

   iv. $\mathbf{q} = \mathbf{Y}_g^\top \mathbf{T}_s \boldsymbol{\beta} + \mathbf{Y}_g^\top \mathbf{U}_d \boldsymbol{\delta} + \mathbf{Y}_\delta^\top \mathbf{u}_g$, $\mathbf{q} = \mathbf{q}/\|\mathbf{q}\|$;

   v. $\mathbf{u}_g = \mathbf{Y}_g \mathbf{q}$;

   vi. $\boldsymbol{\beta} = \mathbf{T}_s^\top \mathbf{u}_g$, $\boldsymbol{\beta} = \boldsymbol{\beta}/\|\boldsymbol{\beta}\|$;

   vii. $\boldsymbol{\delta} = \mathbf{U}_d^\top \mathbf{u}_g$, $\boldsymbol{\delta} = \boldsymbol{\delta}/\|\boldsymbol{\delta}\|$.

3. Dynamic inner modeling: Perform ARX on Eq. (4.19) to obtain the inner model parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, and calculate $\hat{\mathbf{u}}_g = \mathbf{T}_s \boldsymbol{\alpha} + \mathbf{U}_d \boldsymbol{\gamma}$.

4. Deflation: Deflate $\mathbf{X}$ and $\mathbf{Y}$ by

$$\begin{aligned} \mathbf{X} &:= \mathbf{X} - \mathbf{t}\mathbf{p}^\top \\ \mathbf{Y}_{0:g-1} &:= \mathbf{Y}_{0:g-1} - \hat{\mathbf{u}}_{0:g-1}\mathbf{q}^\top \\ \mathbf{Y}_{g:n} &:= \mathbf{Y}_g - \hat{\mathbf{u}}_g \mathbf{c}^\top \end{aligned}$$

5. Repeat Step 2 - 4 until l latent scores are extracted, where l is determined by cross-validation.

4.4.2 Dynamic Auto-Regressive Latent Variable Model

To obtain the weights and latent variables in Eq. (4.11), a dynamic auto-regressive latent variable model (DALVM) is proposed in this subsection. DALVM aims to maximize the covariance between $\hat{\mathbf{u}}_g$ and $\mathbf{u}_g$, according to the following optimization problem:

$$\begin{aligned} \max_{\mathbf{w},\mathbf{q},\boldsymbol{\beta},\boldsymbol{\delta}} \quad & J = \mathbf{q}^\top \mathbf{Y}_g^\top \left[\mathbb{Z}_x \left(\boldsymbol{\beta} \otimes \mathbf{w}\right) + \mathbb{Z}_y \left(\boldsymbol{\delta} \otimes \mathbf{q}\right)\right] \\ \text{s.t.} \quad & \|\mathbf{w}\| = 1, \quad \|\mathbf{q}\| = 1, \quad \|\boldsymbol{\beta}\| = 1, \quad \|\boldsymbol{\delta}\| = 1 \end{aligned} \tag{4.12}$$

Remark 3 *If $d = 0$, DALVM reduces DiPLS, and if $s = 0$, DALVM becomes a pure ARX model. If both s and d are 0, DALVM is simplified as a static PLS.*

To obtain the model parameters in Eq. (4.12), the following equation is constructed with the Lagrange multipliers λ_w, λ_q, λ_β and λ_δ in the dynamic outer modeling.

$$\begin{aligned}\mathcal{L} = & \mathbf{q}^\top \mathbf{Y}_g^\top \left[\mathbb{Z}_x \left(\boldsymbol{\beta} \otimes \mathbf{w}\right) + \mathbb{Z}_y \left(\boldsymbol{\delta} \otimes \mathbf{q}\right)\right] \\ & + \frac{\lambda_w}{2}\left(1 - \mathbf{w}^\top \mathbf{w}\right) + \frac{\lambda_q}{2}\left(1 - \mathbf{q}^\top \mathbf{q}\right) \\ & + \frac{\lambda_\beta}{2}\left(1 - \boldsymbol{\beta}^\top \boldsymbol{\beta}\right) + \frac{\lambda_\delta}{2}\left(1 - \boldsymbol{\delta}^\top \boldsymbol{\delta}\right)\end{aligned} \tag{4.13}$$

Given the constraints in Eq. (4.12), differentiating $\mathcal{L}$ with respect to $\mathbf{w}$, $\mathbf{q}$, $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ and setting them to zero lead to

$$\mathbf{q} \propto \mathbf{Y}_g^\top \mathbf{T}_s \boldsymbol{\beta} + \mathbf{Y}_g^\top \mathbf{U}_d \boldsymbol{\delta} + \mathbf{Y}_\delta^\top \mathbf{u}_g; \quad \mathbf{q} = \mathbf{q}/\|\mathbf{q}\| \tag{4.14}$$

$$\mathbf{w} \propto \mathbf{X}_\beta^\top \mathbf{u}_g; \quad \mathbf{w} = \mathbf{w}/\|\mathbf{w}\| \tag{4.15}$$

$$\boldsymbol{\beta} \propto \mathbf{T}_s^\top \mathbf{u}_g; \quad \boldsymbol{\beta} = \boldsymbol{\beta}/\|\boldsymbol{\beta}\| \tag{4.16}$$

$$\boldsymbol{\delta} \propto \mathbf{U}_d^\top \mathbf{u}_g; \quad \boldsymbol{\delta} = \boldsymbol{\delta}/\|\boldsymbol{\delta}\| \tag{4.17}$$

where the Kronecker relations $\boldsymbol{\beta}\otimes\mathbf{w} = \left(\mathbf{I}_{s+1} \otimes \mathbf{w}\right)\boldsymbol{\beta} = \left(\boldsymbol{\beta} \otimes \mathbf{I}_m\right)\mathbf{w}$ and $\boldsymbol{\delta}\otimes\mathbf{q} = \left(\mathbf{I}_d \otimes \mathbf{q}\right)\boldsymbol{\delta} = \left(\boldsymbol{\delta} \otimes \mathbf{I}_p\right)\mathbf{q}$ are employed. $\mathbf{X}_\beta$, $\mathbf{T}_s$, $\mathbf{Y}_\delta$ and $\mathbf{U}_d$ are defined as

$$\begin{aligned}\mathbf{X}_\beta &\equiv \mathbb{Z}_x \left(\boldsymbol{\beta} \otimes \mathbf{I}_m\right) = \sum_{i=0}^{s} \beta_i \mathbf{X}_{g-i} \\ \mathbf{T}_s &\equiv \mathbb{Z}_x \left(\mathbf{I}_{s+1} \otimes \mathbf{w}\right) = \left[\mathbf{t}_g, \mathbf{t}_{g-1}, \ldots, \mathbf{t}_{g-s}\right] \\ \mathbf{Y}_\delta &\equiv \mathbb{Z}_y \left(\boldsymbol{\delta} \otimes \mathbf{I}_q\right) = \sum_{i=1}^{d} \delta_i \mathbf{Y}_{g-i} \\ \mathbf{U}_d &\equiv \mathbb{Z}_y \left(\mathbf{I}_d \otimes \mathbf{q}\right) = \left[\mathbf{u}_{g-1}, \mathbf{u}_{g-2}, \ldots, \mathbf{u}_{g-d}\right]\end{aligned}$$

Thus, Eqs. (5.17) - (4.17) define the outer structure of DALVM.

To ensure consistency of the outer and inner models, an auto-regressive exogenous model is constructed in inner model to capture dynamic relations among $\mathbf{u}_g$, $\mathbf{T}_s$ and $\mathbf{U}_d$,

$$\mathbf{u}_g = \mathbf{T}_s \boldsymbol{\alpha} + \mathbf{U}_d \boldsymbol{\gamma} + \mathbf{r}_g \tag{4.18}$$

where $\boldsymbol{\alpha} = \left[\alpha_0, \alpha_1, \ldots, \alpha_s\right]^\top$, $\boldsymbol{\gamma} = \left[\gamma_1, \gamma_2, \ldots, \gamma_d\right]^\top$, and $\mathbf{r}_g$ is the modeling error. Based on the structure of the inner model, an ARX model is used to estimate $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$. The mathematical expression of the ARX model is [195]

$$\begin{aligned}\mathbf{y}(t) = & a_1 \mathbf{y}(t-1) + \ldots + a_{n_a} \mathbf{y}(t-n_a) \\ & + b_1 \mathbf{u}(t-n_k) + \ldots + b_{n_b} \mathbf{u}(t-n_b-n_k+1) \\ & + \mathbf{e}(t)\end{aligned}$$

where $\mathbf{y}, \mathbf{u}$ and $\mathbf{e}$ are discrete output, discrete input and model error respectively. n_a and n_b are the number of poles and zeros of model. n_k is the number of input samples

that occur before the input affects the output, also called the dead time in the system. In DALVM, $n_k = 0$, $n_a = d$, $n_b = s$, $\mathbf{y} = \mathbf{u}$, $\mathbf{u} = \mathbf{t}$, $a_i = \gamma_i$ $(i = 1, 1, \ldots, d)$ and $b_{j+1} = \alpha_j$ $(j = 0, 1, \ldots, d)$. Then the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ can be estimated by using prediction error method [195].

After the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ are obtained, the prediction of $\mathbf{u}_g$ can be calculated by

$$\hat{\mathbf{u}}_g = \mathbf{T}_s\boldsymbol{\alpha} + \mathbf{U}_d\boldsymbol{\gamma} \tag{4.19}$$

Afterward, deflation is performed on $\mathbf{X}$ and $\mathbf{Y}$ to remove the effect of the extracted latent variables. $\mathbf{X}$ is deflated by

$$\mathbf{X} := \mathbf{X} - \mathbf{t}\mathbf{p}^\top \tag{4.20}$$

where $\mathbf{p} = \mathbf{X}^\top\mathbf{t}/\mathbf{t}^\top\mathbf{t}$.

The deflation of quality variables is divided into two parts, $\mathbf{Y}_{0:g-1} \equiv \{\mathbf{y}_i\}_{i=0}^{g-1}$ and $\mathbf{Y}_g \equiv \{\mathbf{y}_i\}_{i=g}^{n}$. Similar to static PLS, it is straightforward to deflate $\mathbf{Y}_{0:g-1}$ as

$$\mathbf{Y}_{0:g-1} := \mathbf{Y}_{0:g-1} - \hat{\mathbf{u}}_{0:g-1}\mathbf{q}^\top \tag{4.21}$$

where $\hat{\mathbf{u}}_{0:g-1} = b\mathbf{t}_{0:g-1}$, and $b = \mathbf{u}_{0:g-1}^\top\mathbf{t}_{0:g-1}/\mathbf{t}_{0:g-1}^\top\mathbf{t}_{0:g-1}$.

For $\mathbf{Y}_g$, since $\|\mathbf{Y}_g - \hat{\mathbf{u}}_g\mathbf{c}^\top\|$ should be minimized, its deflation is

$$\mathbf{Y}_g := \mathbf{Y}_g - \hat{\mathbf{u}}_g\mathbf{c}^\top \tag{4.22}$$

where $\mathbf{c} = \mathbf{Y}_g^\top\hat{\mathbf{u}}_g/\hat{\mathbf{u}}_g^\top\hat{\mathbf{u}}_g$.

With iterative extraction, the auto and cross-temporal relations between $\mathbf{X}$ and $\mathbf{Y}$ can be exploited. The details of DALVM algorithm are summarized in Algorithm 2.

4.4.3 Comprehensive Online Anomaly Detection

Quality-relevant dynamics relations are fully exploited in DALVM thus permitting detection of anomalies from high dimensional dynamic data. On the other hand, it is still important to detect abnormal static variations in quality and process-related variables to ensure full assessment of the process status. Therefore, in this subsection, a comprehensive online static and dynamic anomaly detection system is designed based on concurrent DALVM.

Concurrent Dynamic Auto-Regressive LVM

Inspired by CPLS [132], subsequent decompositions to the extracted latent structures are proposed for DALVM, which is referred to as concurrent DALVM (ConDALVM). A concurrent design has been previously reported for static processes but the decomposition of the dynamic and auto-regressive variations has not been addressed in previous studies.

In the proposed concurrent DALVM, both dynamic and static variations from process and quality spaces are exploited, and based on these a monitoring scheme is developed for comprehensive monitoring.

ConDALVM decomposes the data into several subspaces: (1) the dynamic process latent scores extracted by DALVM are further decomposed by a singular value decomposition (SVD) to construct a dynamic covariance subspace (DCVS); (2) the dynamic auto-regressive quality subspace (DAQS) is constructed for the auto-regressive quality part; (3) the remaining quality part is further separated into static quality principal subspace (SQPS) and static quality residual subspace (SQRS) with a subsequent PCA decomposition; (4) the process variations are decomposed with DiPCA into dynamic process principal subspace (DPPS), static process principal subspace (SPPS) and static process residual subspace (SPRS). For ease of understanding, a list of subspace abbreviations are summarized in Table 4.1. The details of the variations in each subspace are summarized in Table 4.2, and the detailed procedure of ConDALVM is as follows:

a. Perform DALVM on the scaled $\mathbf{X}$ and $\mathbf{Y}$ to obtain model parameters $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$, $\mathbf{Q}$, $\mathbf{C}$ and $\mathbf{R}$.

b. The predictable quality variations $\hat{\mathbf{Y}}_g$ can be divided into two parts.

$$\begin{cases} \hat{\mathbf{Y}}_{xg} = & [\mathbb{Z}_x (\boldsymbol{\alpha} \otimes \mathbf{R})] \mathbf{C}^\top \\ \hat{\mathbf{Y}}_{yg} = & [\mathbb{Z}_y (\boldsymbol{\gamma} \otimes \mathbf{Q})] \mathbf{C}^\top \end{cases} \tag{4.23}$$

where $\hat{\mathbf{Y}}_{xg}$ is the predictable output and $\hat{\mathbf{Y}}_{yg}$ is the auto-regressive quality part.

i Perform SVD on $\hat{\mathbf{Y}}_{xg}$ to extract the dynamic relations that are predictable from process variables.

$$\hat{\mathbf{Y}}_{xg} = \mathbf{U}_c \mathbf{D}_c \mathbf{V}_c^\top \equiv \mathbf{U}_c \mathbf{Q}_c^\top \tag{4.24}$$

where $\mathbf{Q}_c = \mathbf{V}_c \mathbf{D}_c$ and the non-zero sigular values in $\mathbf{D}_c$ are allocated in a descending order. Since $\mathbf{V}_c$ is orthogonal, $\mathbf{U}_c$ can be written as

$$\mathbf{U}_c = \hat{\mathbf{Y}}_{xg} \mathbf{V}_c \mathbf{D}_c^{-1} \equiv \mathbf{X}_\alpha \mathbf{R}_c \tag{4.25}$$

where $\mathbf{X}_\alpha = \mathbb{Z}_x (\boldsymbol{\alpha} \otimes \mathbf{I}) = \sum_{i=0}^{s} \alpha_i \mathbf{X}_{s-i}$ and $\mathbf{R}_c = \mathbf{R}\mathbf{C}^\top \mathbf{V}_c \mathbf{D}_c^{-1}$

ii The quality auto-regressive variations $\hat{\mathbf{Y}}_{yg}$ can be expressed as

$$\hat{\mathbf{Y}}_{yg} = [\mathbb{Z}_y (\boldsymbol{\gamma} \otimes \mathbf{Q})] \mathbf{C}^\top \equiv \hat{\mathbf{U}}_\gamma \mathbf{C}^\top \tag{4.26}$$

where $\hat{\mathbf{U}}_\gamma = [\mathbb{Z}_y (\boldsymbol{\gamma} \otimes \mathbf{Q})]$.

c. Perform PCA on the unpredictable output $\mathbf{Y}_c = \mathbf{Y} - \hat{\mathbf{Y}}_{xg} - \hat{\mathbf{Y}}_{yg}$ to separate large variations from noise.

$$\mathbf{Y}_c = \mathbf{T}_y \mathbf{P}_y^\top + \mathbf{Y}_s \tag{4.27}$$

d. Perform DiPCA on $\mathbf{X}$ to extract both static and dynamic process variations

$$\mathbf{X} = \hat{\mathbf{T}}_{d,x}\mathbf{P}_{d,x}^{\top} + \mathbf{T}_{s,x}\mathbf{P}_{s,x}^{\top} + \mathbf{E}_s \tag{4.28}$$

where $\hat{\mathbf{T}}_{d,x}\mathbf{P}_{d,x}^{\top}$ is the dynamic component which can be predicted from the past process scores, $\mathbf{T}_{s,x}\mathbf{P}_{s,x}^{\top}$ is the static component, and $\mathbf{E}_s$ represents the static residuals.

With ConDALVM, $\mathbf{x_k}$ and $\mathbf{y_k}$ are decomposed by

$$\begin{aligned} \mathbf{x}_k =& \mathbf{P}_{d,x}\hat{\mathbf{t}}_{d,x,k} + \mathbf{P}_{s,x}\mathbf{t}_{s,x,k} + \mathbf{e}_{s,x,k} \\ \mathbf{y}_k =& \mathbf{Q}_c\mathbf{u}_c + \mathbf{C}\hat{\mathbf{u}}_\gamma + \mathbf{P}_y\mathbf{t}_y + \tilde{\mathbf{y}} \end{aligned} \tag{4.29}$$

where

$$\hat{\mathbf{t}}_{d,x,k} = \sum_{i=1}^{s} \mathbf{\Theta}_{x,i}\mathbf{t}_{d,x,k-i} \tag{4.30}$$

$$\mathbf{t}_{s,x,k} = \mathbf{P}_{s,x}\mathbf{e}_{x,k} \tag{4.31}$$

$$\mathbf{e}_{s,x,k} = \left(\mathbf{I} - \mathbf{P}_{s,x}\mathbf{P}_{s,x}^{\top}\right)\mathbf{e}_{x,k} \tag{4.32}$$

are scores and residual vectors in DPPS, SPPS and SPRS respectively, where $\mathbf{t}_{d,x,k} = \mathbf{R}_x\mathbf{x}_k$ and $\mathbf{e}_{x,k} = \mathbf{x}_k - \mathbf{P}_{d,x}\hat{\mathbf{t}}_{d,x,k}$, and $\mathbf{P}_{d,x}$, $\mathbf{P}_{s,x}$, $\mathbf{\Theta}_x$ and $\mathbf{R}_x$ are DiPCA parameters [50].

Similarly, we calculate

$$\mathbf{u}_c = \mathbf{R}_c^{\top}\mathbf{x}_{\alpha,k} \tag{4.33}$$

$$\hat{\mathbf{u}}_\gamma = \sum_{i=1}^{d} \gamma_i \mathbf{y}_{k-i}\mathbf{Q} \tag{4.34}$$

$$\mathbf{t}_y = \mathbf{P}_y^{\top}\tilde{\mathbf{y}}_c \tag{4.35}$$

$$\tilde{\mathbf{y}} = \left(\mathbf{I} - \mathbf{P}_y\mathbf{P}_y^{\top}\right)\tilde{\mathbf{y}}_c \tag{4.36}$$

which are corresponding scores and residual vectors in DCVS, DAQS, SQPS and SQRS sub-spaces respectively, where $\mathbf{x}_{\alpha,k} = \sum_{i=0}^{s}\alpha_i\mathbf{x}_{k-i}$, and $\tilde{\mathbf{y}}_c = \mathbf{y}_k - \mathbf{Q}_c\mathbf{u}_c - \mathbf{C}\hat{\mathbf{u}}_\gamma$.

Eqs. (4.30)-(4.36) represent all the dynamic and static relations decomposed by ConDALVM.

Remark 4 *For the subscripts in the above variables, k is the timestamp of the sample, d and s denote dynamic and static variations separately, and x and y represent process and quality vectors respectively.*

Definition of Monitoring Indices

The combined index ϕ, Hotelling's T^2 index and Q index are used in ConDALVM-based monitoring scheme [74]. The scores and residuals given in Eqs. (4.30)-(4.36) are employed to detect anomalies in different subspaces.

The dynamic quality-relevant scores $\mathbf{u}_c$ in Eq. (4.33) are orthonormalized with variance matrix $\mathbf{\Lambda}_c$, which is monitored by the T_c^2 statistic,

$$T_c^2 = \mathbf{x}_{\alpha,k}^\top \mathbf{R}_c \mathbf{\Lambda}_c^{-1} \mathbf{R}_c^\top \mathbf{x}_{\alpha,k} \tag{4.37}$$

$\hat{\mathbf{u}}_\gamma$ in Eq. (4.29) are auto-regressive quality variations which can be predicted from past quality variables $\hat{\mathbf{Y}}_{yg}$, and its monitoring index $T_{d,y}^2$ is defined as

$$T_{d,y}^2 = \hat{\mathbf{u}}_\gamma^\top \mathbf{\Lambda}_{d,y}^{-1} \hat{\mathbf{u}}_\gamma \tag{4.38}$$

where $\mathbf{\Lambda}_{d,y}$ is the variance of $\hat{\mathbf{u}}_\gamma$.

For the unpredictable quality principal components $\mathbf{t}_y$ and residuals $\tilde{\mathbf{y}}_c$, the corresponding statistics are

$$T_y^2 = \mathbf{t}_y^\top \mathbf{\Lambda}_y^{-1} \mathbf{t}_y \tag{4.39}$$

$$Q_y = \|\tilde{\mathbf{y}}_c\|^2 \tag{4.40}$$

where $\mathbf{\Lambda}_y$ is the variance of $\mathbf{t}_y$.

Table 4.1: Description of Subspace Abbreviations

Subspace	Description
DCVS	dynamic covariance subspace
DAQS	dynamic auto-regressive quality subspace
SQPS	static quality principal subspace
SQRS	static quality residual subspace
DPPS	dynamic process principal subspace
SPPS	static process principal subspace
SPRS	static process residual subspace

Table 4.2: Fault Detection Indices for ConDALVM

Subspace	Index	Control Limits	Properties
DCVS	T_c^2	$\chi_{\alpha,l}^2$	quality; dynamic
DAQS	$T_{d,y}^2$	$\chi_{\alpha,l_{d,y}}^2$	quality; auto-regressive
SQPS	T_y^2	χ_{α,l_y}^2	quality; static
SQRS	Q_y	$g_y \chi_{\alpha,h_y}^2$	quality; residuals
DPPS	$\phi_{d,x}$	$g_{d,x} \chi_{\alpha,h_{d,x}}^2$	process; dynamic
SPPS	$T_{s,x}^2$	$\chi_{\alpha,l_{s,x}}^2$	process; static
SPRS	$Q_{s,x}$	$g_{s,x} \chi_{\alpha,h_{s,x}}^2$	process; residuals

Similarly, $\phi_{d,x}$, $T^2_{s,x}$ and $Q_{s,x}$ are used to monitor dynamic process-relevant variations, static process-relevant variations and process residuals respectively, and are defined as follows:

$$\phi_{d,x} = \hat{\mathbf{t}}^\top_{d,x,k} \mathbf{\Phi}_{d,x} \hat{\mathbf{t}}_{d,x,k} = \frac{T^2_{d,x}}{\chi^2_{d,x}} + \frac{Q_{d,x}}{\delta^2_{d,x}} \tag{4.41}$$

$$T^2_{s,x} = \mathbf{t}^\top_{s,x,k} \mathbf{\Lambda}^{-1}_{s,x} \mathbf{t}_{s,x,k} \tag{4.42}$$

$$Q_{s,x} = \|\mathbf{e}_{s,x,k}\|^2 \tag{4.43}$$

where $\mathbf{\Phi}_{d,x} = \frac{\mathbf{P}_{t,x}\mathbf{\Lambda}^{-1}_{d,x}\mathbf{P}^\top_{t,x}}{\chi^2_{d,x}} + \frac{\mathbf{I}-\mathbf{P}_{t,x}\mathbf{P}^\top_{t,x}}{\delta^2_{d,x}}$, $\mathbf{\Lambda}_{d,x}$ is the variance of $\hat{\mathbf{t}}_{d,x,k}$, and $\mathbf{\Lambda}_{s,x}$ is the variance of $\mathbf{t}_{s,x,k}$. In $\mathbf{\Phi}_{d,x}$, loading scores $\mathbf{P}_{t,x}$ is obtained by performing PCA-based monitoring on $\hat{\mathbf{t}}_{d,x,k}$. $\chi^2_{d,x}$ and $\delta^2_{d,x}$ are the corresponding control limits calculated in PCA-based monitoring [194].

Control Limits

The control limits for the aforementioned indices are designed based on [74, 22, 194], which are summarized in Table 4.2. In this Table l, $l_{d,y}$, l_y and $l_{s,x}$ are the number of latent variables for $\mathbf{u}_c$, $\hat{\mathbf{u}}_\gamma$, $\tilde{\mathbf{y}}_c$ and $\mathbf{e}_{x,k}$ respectively, and g_y, h_y, $g_{d,x}$, $h_{d,x}$, $g_{s,x}$ and $h_{s,x}$ are calculated based on [74, 22, 194] for SQRS, DPPS and SPRS respectively. In the ConDALVM-based monitoring system the following rules are used:

1. If $T^2_c > \chi^2_{\alpha,l}$, a dynamic quality-relevant fault that can be predicted from process variables is detected with $(1-\alpha)\times 100\%$ confidence.
2. If $T^2_{d,y} > \chi^2_{\alpha,l_{d,y}}$, an auto-regressive quality-relevant fault is detected with $(1-\alpha)\times 100\%$ confidence.
3. If $T^2_y > \chi^2_{\alpha,l_y}$ or $Q_y > g_y\chi^2_{\alpha,h_y}$, an unpredictable static quality-relevant fault is detected with $(1-\alpha)\times 100\%$ confidence.
4. If $\phi_{d,x} > g_{d,x}\chi^2_{\alpha,h_{d,x}}$, a dynamic process-relevant disturbance is detected with $(1-\alpha)\times 100\%$ confidence.
5. If $T^2_{s,x} > \chi^2_{\alpha,l_{s,x}}$ or $Q_{s,x} > g_{s,x}\chi^2_{\alpha,h_{s,x}}$, a static process-relevant disturbance is detected with $(1-\alpha)\times 100\%$ confidence.
6. If all monitoring indices are within thresholds, the sample is classified as normal.

Different alarming levels are attached for detected anomalies with quality-relevant indices (T^2_c, $T^2_{d,y}$, T^2_y and Q_y) being assigned a higher attention.

4.5 Experiments and Discussion

In this section, a toy example and two industrial processes are employed to illustrate the modeling and monitoring effectiveness of the proposed DALVM and ConDALVM.

4.5.1 Toy example

The following numerical scenario is generated [219].

$$\begin{cases} \mathbf{t}_k = \mathbf{A}_1\mathbf{t}_{k1} + \mathbf{A}_2\mathbf{t}_{k2} + \mathbf{f}_k \\ \mathbf{x}_k = \mathbf{P}\mathbf{t}_k + \mathbf{e}_k \\ \mathbf{y}_k = \mathbf{C}_1\mathbf{x}_k + \mathbf{C}_2\mathbf{x}_{k-1} + \mathbf{C}_3\mathbf{x}_{k-2} \\ \qquad +0.3\mathbf{y}_{k-1} + 0.4\mathbf{y}_{k-2} + \mathbf{v}_k \end{cases} \tag{4.44}$$

where

$$\mathbf{A}_1 = \begin{bmatrix} 0.677 & 0.581 & 0.932 \\ 1.281 & -0.534 & -1.600 \\ -1.508 & 0.999 & 0.753 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 0.716 & -0.065 & 1.119 \\ 1.113 & -0.537 & -0.169 \\ -0.557 & -1.075 & 0.233 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 0.577 & 0.286 & 0.161 \\ 0.366 & 0.046 & 0.906 \\ 0.589 & 0.465 & 0.994 \\ 0.357 & 0.345 & 0.740 \\ 0.404 & 0.685 & 0.226 \end{bmatrix}, \mathbf{C}_1 = \begin{bmatrix} 0.745 & 1.994 & 0.031 \\ 0.493 & 0.773 & 0.057 \\ 0.732 & 1.015 & 0.301 \\ 0.474 & 1.156 & 0.631 \\ 0.565 & 1.231 & 0.983 \end{bmatrix}^\top$$

$$\mathbf{C}_2 = \begin{bmatrix} 0.918 & 0.623 & 0.519 \\ 0.018 & 1.128 & 0.105 \\ 0.578 & 0.407 & 0.786 \\ 0.429 & 0.441 & 0.594 \\ 0.081 & 1.128 & 0.504 \end{bmatrix}^\top, \mathbf{C}_3 = \begin{bmatrix} 0.290 & 1.428 & 0.129 \\ 0.576 & 1.393 & 0.911 \\ 0.823 & 0.057 & 1.484 \\ 1.148 & 1.037 & 0.911 \\ 1.459 & 0.130 & 0.530 \end{bmatrix}^\top$$

where the noises $\mathbf{f}_k$, $\mathbf{e}_k$ and $\mathbf{v}_k$ follow the normal distribution as $\mathbf{f}_k \in \mathbb{R}^3 \sim \mathcal{N}(0, 0.5^2)$, $\mathbf{e}_k \in \mathbb{R}^5 \sim \mathcal{N}(0, 0.5^2)$ and $\mathbf{v}_k \in \mathbb{R}^3 \sim \mathcal{N}(0, 0.5^2)$, and the latent component $\mathbf{t}_k$ is generated from $\mathbf{t}_{k1}$ and $\mathbf{t}_{k2}$ which follow the normal distribution as $\mathbf{t}_{k1} \in \mathbb{R}^3 \sim \mathcal{N}(0, 2^2)$ and $\mathbf{t}_{k2} \in \mathbb{R}^3 \sim \mathcal{N}(0, 2^2)$. As shown in Eq. (4.44), dynamics exist in both $\mathbf{X}$ and $\mathbf{Y}$, and $\mathbf{Y}$ is affected by both past $\mathbf{X}$ and past $\mathbf{Y}$. p

Table 4.3: MSEs of DALVM, DiPLS and PLS for Numerical Study

Model	$\mathbf{y}_1$	$\mathbf{y}_2$	$\mathbf{y}_3$
DALVM	0.0363	0.0356	0.0339
DiPLS	0.2412	0.2099	0.1608
PLS	1.0683	0.9157	1.0821

In this case study, 1200 samples are generated with Eq. (4.44), where the first 800 samples are selected to train the model, the next 200 samples are used to select the model parameters by using cross-validation method, and the remaining 200 samples are selected as testing data. The results of parameter selection are $l = 3$ for PLS; $l = 3$ and $s = 10$ for DiPLS; and $l = 2$, $s = 2$, and $d = 2$ for DALVM. Their prediction performance on the test data are shown in Figs. 4.1 - 4.3 respectively, and the corresponding mean squared errors (MSEs) are shown in Table 4.3. It is shown that the prediction of DALVM follows the actual variations well, and it obtains the smallest MSEs. In contrast, PLS performs poorly

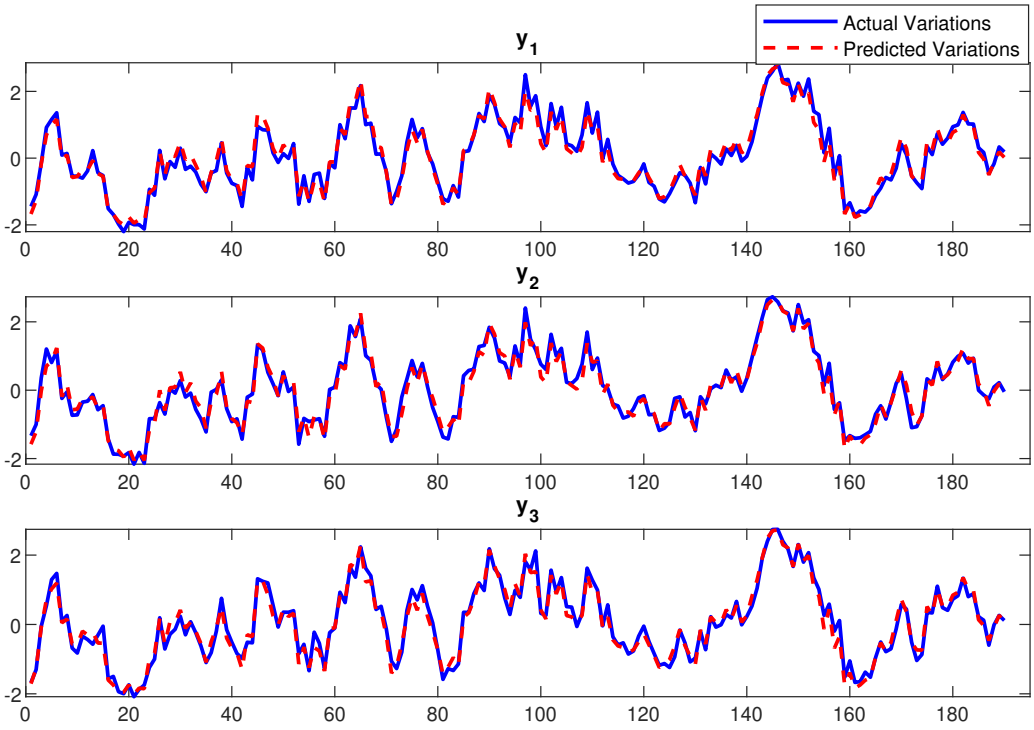


Figure 4.1: Modeling Performance of DALVM for Numerical Study

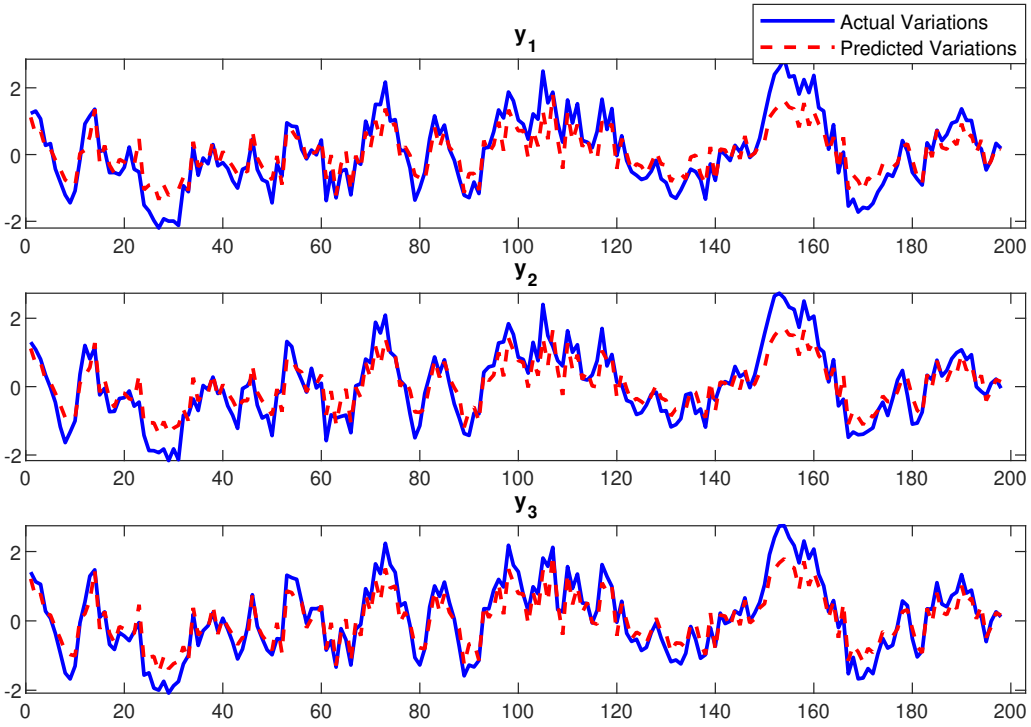


Figure 4.2: Modeling Performance of DiPLS for Numerical Study

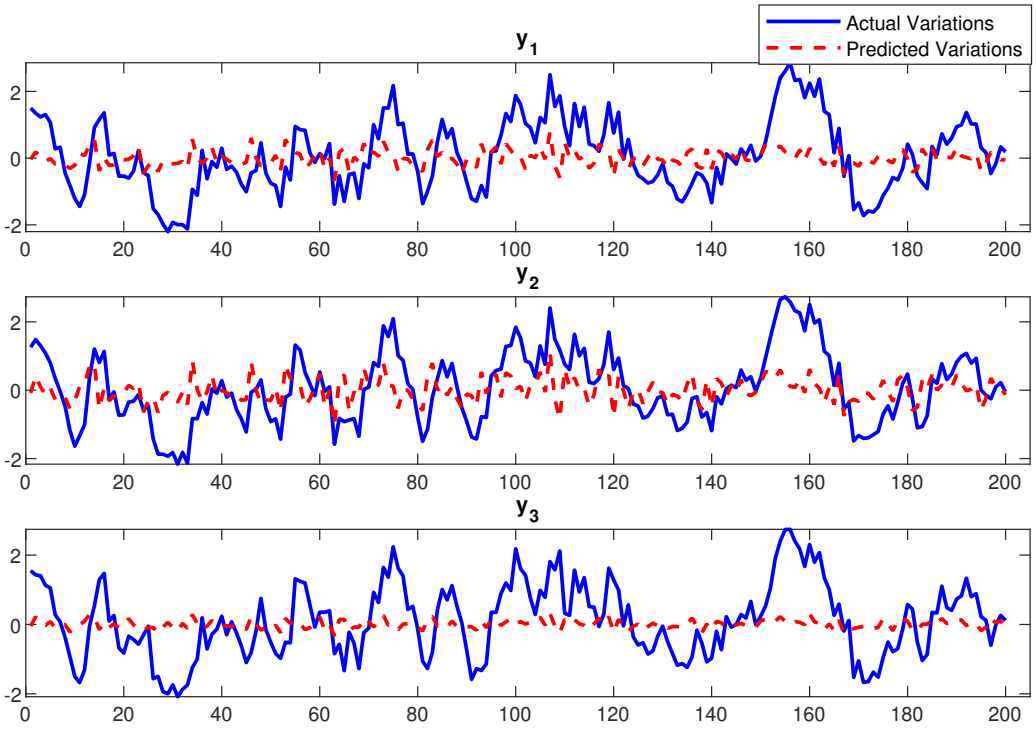


Figure 4.3: Modeling Performance of PLS for Numerical Study

since traditional PLS only extracts static relations. For DiPLS,extra lags are needed to achieve satisfactory results. DALVM outperforms others in prediction performance, and the number of lags associated to quality and process variables are consistent with Eq. (4.44). Figs. 4.4 and 4.5 show the auto-correlations and cross-correlations in the residuals of quality variables for DALVM and DiPLS. Fig. 4.4 shows that dynamic variations do not exist in the residuals since all of them are extracted by DALVM. However, as shown in Fig. 4.5, there are still dynamic variations in the residuals after performing DiPLS. It is observed that compared with DiPLS, DALVM can model the auto-regressive structure in quality variables well, and achieves better dynamic modeling performance.

4.5.2 Tennessee Eastman Process

The TEP data proposed in [54] are widely used to evaluate modeling and monitoring performance of control and monitoring algorithms. In TEP, 53 variables are divided into two parts: 12 manipulated variables (XMV(1-12)) and 41 measured variables (XMEAS(1-41)), and more details about these variables are elaborated in [54]. In our work, XMEAS(1-22) and XMV(1-11) are used as the process variables $\mathbf{X}$, and XMEAS(38) is selected as the quality variable $\mathbf{Y}$. It is noted that XMEAS(1-22) and XMV(1-11) are sampled every 3 minutes, while quality variable XMEAS(38) is sampled every 15 minutes. To balance the sampling discrepancy, only 1/5 samples without duplicate samples are employed in this case study [221].

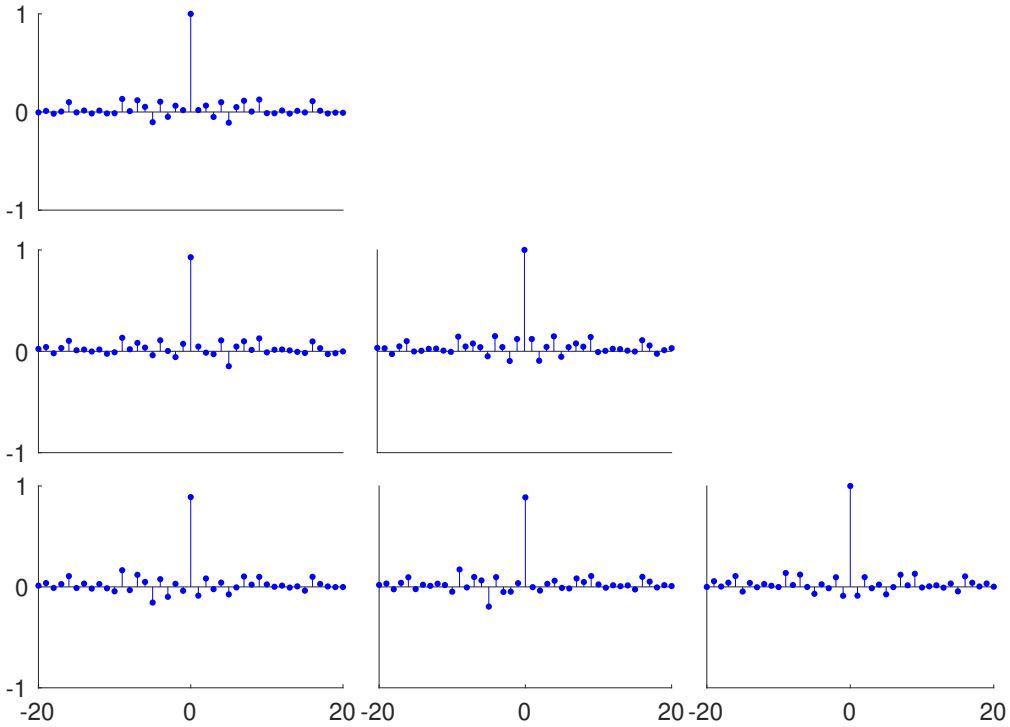


Figure 4.4: Auto-Correlations and Cross-Correlations of Quality Residuals for DALVM

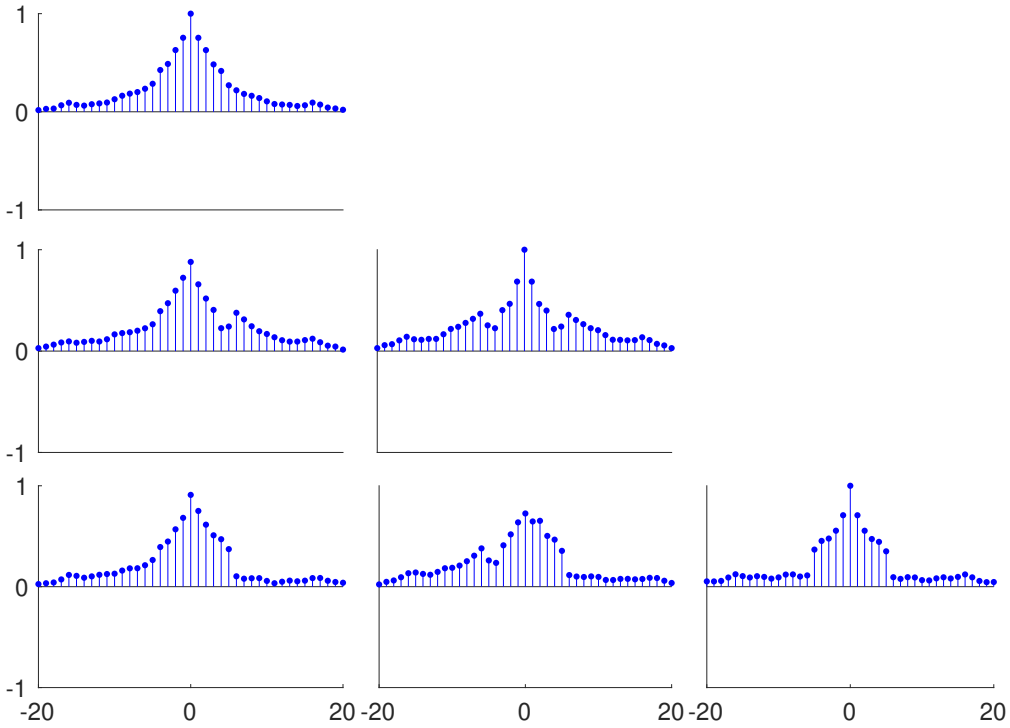


Figure 4.5: Auto-Correlations and Cross-Correlations of Quality Residuals for DiPLS

Table 4.4: MSEs of DALVM, DiPLS and PLS for TEP

	DALVM	DiPLS	PLS
MSEs	0.2462	0.6316	0.9943

Dynamic Modeling Performance

In order to illustrate the time series modeling performance of DALVM, 100 samples are selected as training data and 192 samples are used as test data. After applying the hold-out method, we obtain $l = 3$, $s = 3$, and $d = 1$ for DALVM; $l = 3$ and $s = 5$ for DiPLS; and $l = 32$ for PLS. The actual and predicted quality variations obtained with these methods are shown in Fig. 4.6 and the corresponding MSEs are summarized in Table 4.4. From Fig. 4.6 and Table 4.4, it is observed that DALVM achieves the best prediction performance since it results in the the smallest MSE and the time lag of DALVM is also the smallest. The quality variations predicted by DALVM can match the actual variations better since DALVM is able to model the dynamic cross-correlations and auto-correlations more accurately. To show the contributions of dynamic cross-correlations and auto-correlations, Fig. 4.7 and Table 4.5 are added to show the predicted variations of $\hat{\mathbf{Y}}_{xg}$ and $\hat{\mathbf{Y}}_{yg}$ and to present their corresponding mean squared prediction errors respectively.

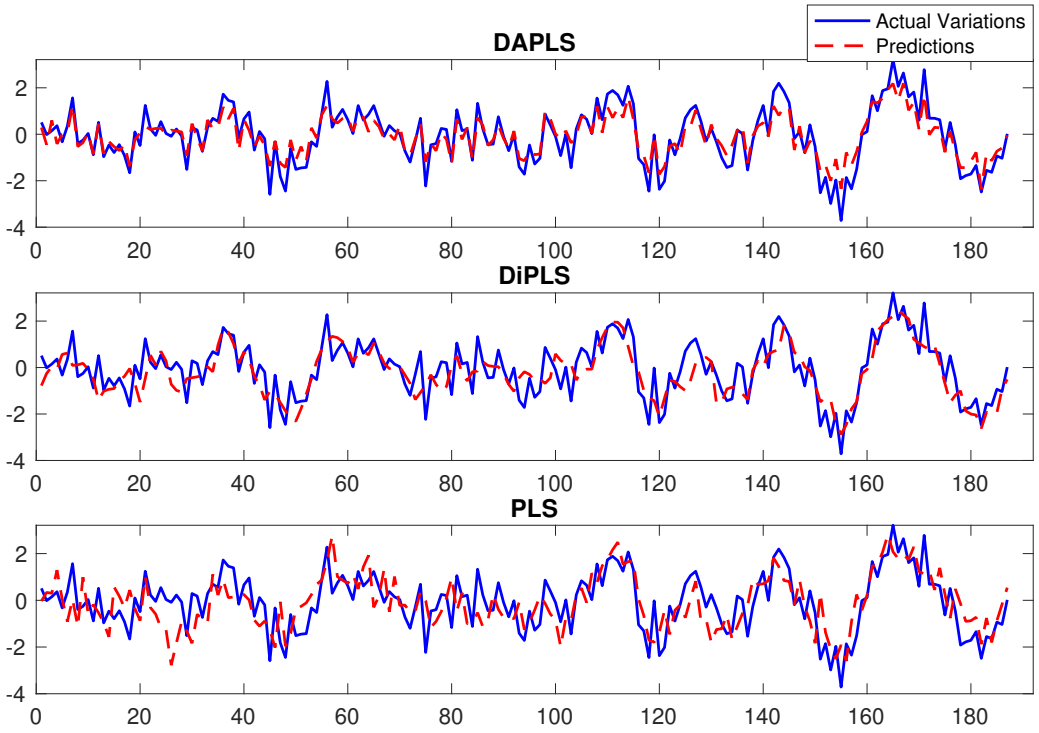


Figure 4.6: Dynamic Modeling Performance for TEP

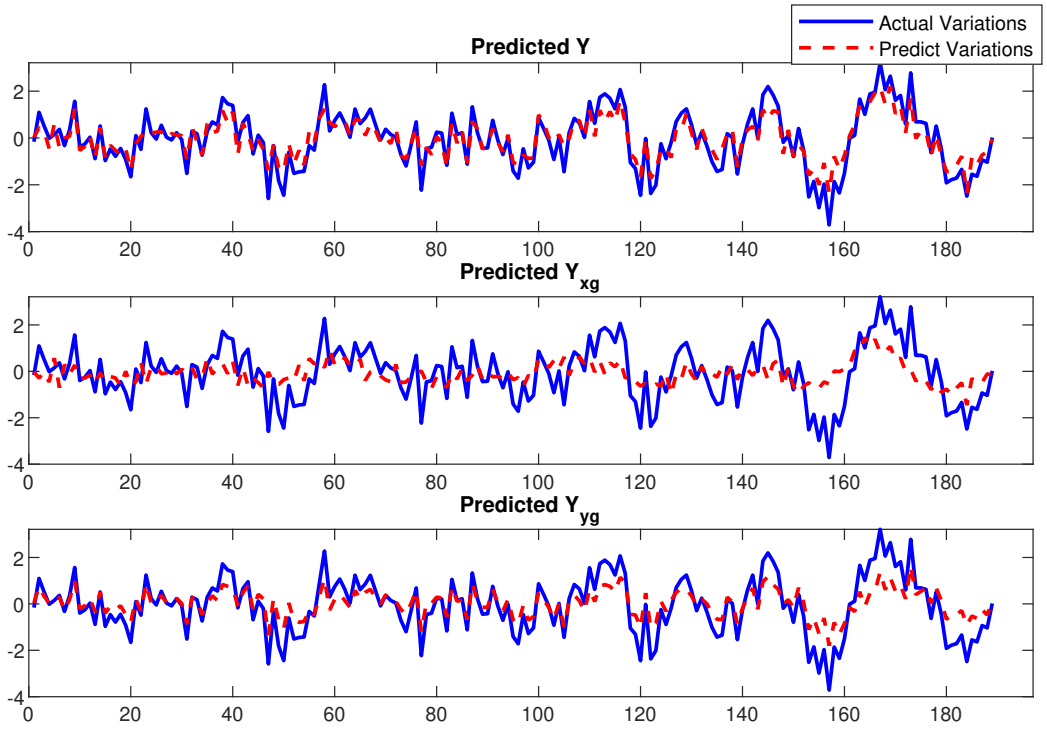


Figure 4.7: Modeling Performance of $\hat{\mathbf{Y}}_{xg}$ and $\hat{\mathbf{Y}}_{yg}$ for TEP

Table 4.5: MSEs of DALVM, DiPLS and PLS for TEP

	$\hat{\mathbf{Y}}$	$\hat{\mathbf{Y}}_{xg}$	$\hat{\mathbf{Y}}_{yg}$
MSEs	0.2462	1.0872	0.5246

Monitoring Performance

15 known disturbances were simulated in [54], and PCA-based *quality monitoring* method is applied to separate them into two groups, quality-relevant one and quality-irrelevant one. The criterion used for separation reasoning is that if the detected faulty samples are less than 15% of the total samples, the corresponding disturbance is classified as a quality-irrelevant disturbance; otherwise, it is a quality-relevant fault. Table 4.6 and Table 4.7 summarize the fault detection rates (FDRs) and false alarm rates (FARs) of DiPLS and ConDALVM for these two groups respectively. It is noted that FDRs and FARs in Tables 4.6 are based on the combination of all quality-relevant indices (T_c^2, $T_{d,y}^2$, T_y^2 and Q_y). From the results in these tables, it can be concluded that compared with DiPLS, ConDALVM achieves higher FDRs with comparable FARs.

Furthermore, IDV(4) and IDV(5) are selected to present the detailed monitoring performance of ConDALVM. A step change occurs in IDV(4) which increases the reactor cooling water inlet temperature, which is a quality-irrelevant disturbance as shown in Fig. 4.8. T_x^2 and Q_x in Fig. 4.8 are the indices for PCA-based *process monitoring*, and T_y^2 is the index for PCA-based *quality monitoring*. Fig. 4.9 shows the variation of each indices in ConDALVM.

As observed from the monitoring performance of ConDALVM in Fig. 4.9, anomalies are detected in process-relevant subspaces with $\phi_{d,x}$, $T^2_{s,x}$ and $Q_{s,x}$ only, which aligns with the actual case, leading to the conclusion that it is process-relevant but quality-irrelevant.

A step change, which affects the condenser cooling water inlet temperature, is introduced to IDV(5), and it is a transient quality-relevant disturbance as seen from the *quality monitoring* results in Fig. 4.10 where the variations of quality-relevant indices reflect the occurrence of a step change. Similar trend is detected with the monitoring indices T^2_c, $T^2_{d,y}$ and T^2_y obtained with ConDALVM in Fig. 4.11, and the fault alarms is only indicated by Q_{sx}, which implies that IDV(5) has a continuous effect on the static process residual subspace.

Table 4.6: FDRs and FARs for Quality-Relevant Faults (%)

Fault Type	DiPLS		ConDALVM	
	FDRs	FARs	FDRs	FARs
IDV(1)	91.78	**66.67**	**100**	68.91
IDV(2)	96.3	76.69	**100**	76.69
IDV(5)	72.72	**11.49**	**97.72**	14.19
IDV(6)	100	17.95	**100**	**17.07**
IDV(7)	91.55	39.5	**97.18**	**23.14**
IDV(8)	85.04	**41.27**	**100**	44.62
IDV(10)	58.9	29.06	**91.78**	**27.73**
IDV(12)	88.55	**44.07**	**99.24**	52.46
IDV(13)	96.6	**13.95**	**100**	17.78

Table 4.7: FARs for Quality-Irrelevant Disturbances (%)

Fault Type	FARs (DiPLS)	FARs (ConDALVM)		
	T^2	T^2_c	$T^2_{d,y}$	T^2_y
IDV(3)	9.32	3.06	**2.45**	3.07
IDV(4)	6.98	**0.57**	1.72	3.45
IDV(9)	6.75	4.85	**2.42**	3.64
IDV(11)	4.94	7.32	**1.83**	**1.83**
IDV(14)	5.78	6.28	**4.51**	4.57
IDV(15)	4.71	3.81	**2.91**	2.91

4.5.3 Three-Phase Flow Facility

The three-phase flow facility (TPFF) is a pressurized system designed to control the flow rate of air, water and oil [141]. In this process, there are 24 different process variables, and in our case study, Variables 1-7, 10, 11-12, 19-21 and 23 are selected as process variables

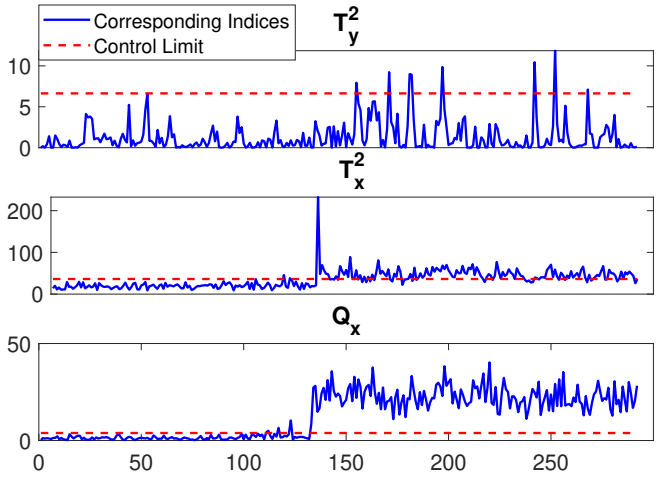


Figure 4.8: PCA-based Process and Quality Monitoring for IDV(4)

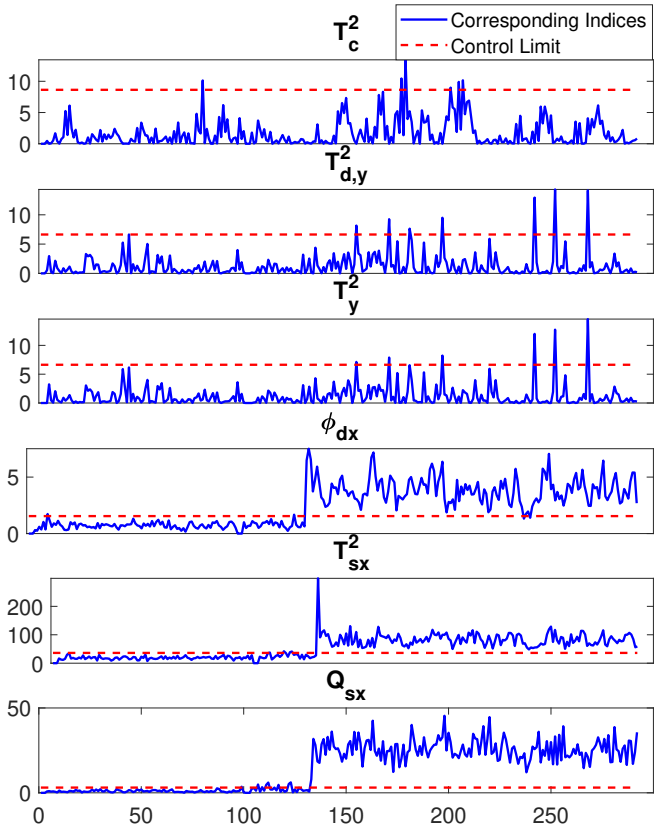


Figure 4.9: Monitoring Performance of ConDALVM for IDV(4)

while Variable 13 is selected as quality variable. Readers can refer to [186, 141] for their physical meanings.

Three data sets (T1, T2 and T3) were acquired from the system as normal data and different faults are introduced in each dataset, leading to 16 fault cases in total. The

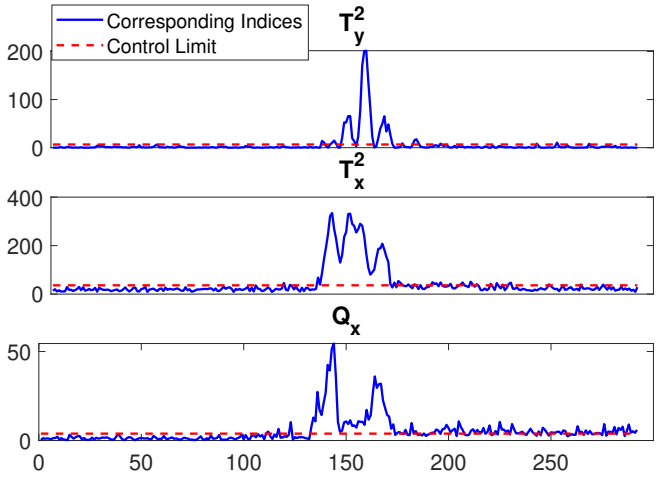


Figure 4.10: PCA-based Process and Quality Monitoring for IDV(5)

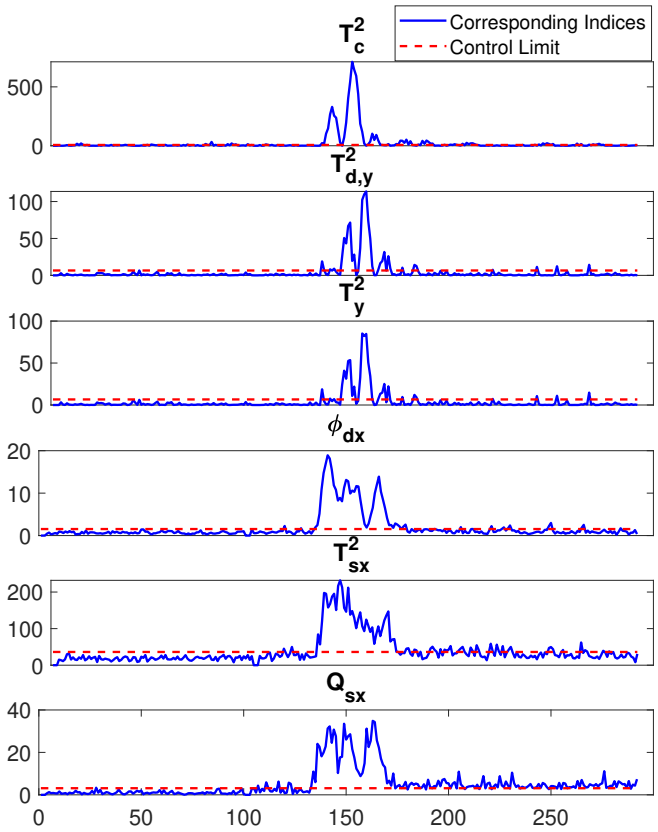


Figure 4.11: Monitoring Performance of ConDALVM for IDV(5)

parameters are determined by the hold-out method as $l = 5$, $s = 1$ and $d = 1$ for DALVM; and $l = 5$ and $s = 5$ for DiPLS. Similar to TEP, the disturbances are divided into quality-relevant faults and quality-irrelevant disturbances, and Table 4.8 and Table 4.9 summarize their FDRs and FARs respectively. It is shown that ConDALVM retains lower FARs than DiPLS for quality-irrelevant disturbances, and for quality-relevant cases, its FDRs are

higher with comparable FARs. Fault case F3_T3, which is caused by the top separator input blockage, directly affects the quality variables. Thus, most information is captured by the auto-regressive part, while $\hat{\mathbf{Y}}_{xg}$ introduces noise into the ConDALVM based model thus leading to a large FAR.

Further, faults 1 in T2 set (F1_T2), where the air line blockage exists, is selected to compare their monitoring performance. As shown in Fig. 4.12, F1_T2 affects the quality variables in the range of Samples [$3143^{\text{th}}, 3793^{\text{th}}$]. In Fig. 4.13, the T_c^2, $T_{d,y}^2$ and T_y^2 calculated with ConDALVM serve to detect the faulty samples successfully, and T_c^2 even forecasts the fault with the aid of process variables. However, DiPLS in Fig. 4.14 fails to detect the fault with T_y^2, and classifies it as a quality-irrelevant fault incorrectly since the auto-regressive dynamic of the quality variable is not captured by DiPLS. Thus, ConDALVM achieves better monitoring performance, especially for faulty samples around 3450^{th} to 3750^{th}.

Table 4.8: FDRs and FARs for Quality-Relevant Faults (%)

Fault Type	DiPLS		ConDALVM	
	FDRs	FARs	FDRs	FARs
F1_T1	99.43	7.56	**100**	**5.97**
F1_T2	18.54	**5.6**	**100**	10.37
F1_T3	38.95	**2.75**	**100**	9.49
F2_T2	100	7.34	**100**	**5.71**
F3_T1	89.44	**6.42**	**100**	14.15
F3_T3	9.47	**7.54**	**100**	77.41
F4_T1	38.73	5.01	**100**	**4.21**
F4_T2	80.79	**4.2**	**100**	6.31
F4_T3	46.93	**4.52**	**100**	5.12
F5_T1	43.76	**2.31**	**100**	7.15
F5_T2	40.34	3.9	**100**	**2.31**
F6_T2	0	**7.31**	**100**	8.68

Table 4.9: FARs for Quality-Irrelevant Disturbances (%)

Fault Type	FARs (DiPLS)	FARs (ConDALVM)		
	T^2	T_c^2	$T_{d,y}^2$	T_y^2
F2_T1	7.54	0.6	**0**	0.76
F2_T3	8.63	1.96	**0**	1.32
F3_T2	11.1	0.03	**0.1**	0.08
F6_T1	8.25	0.63	**0**	0.75

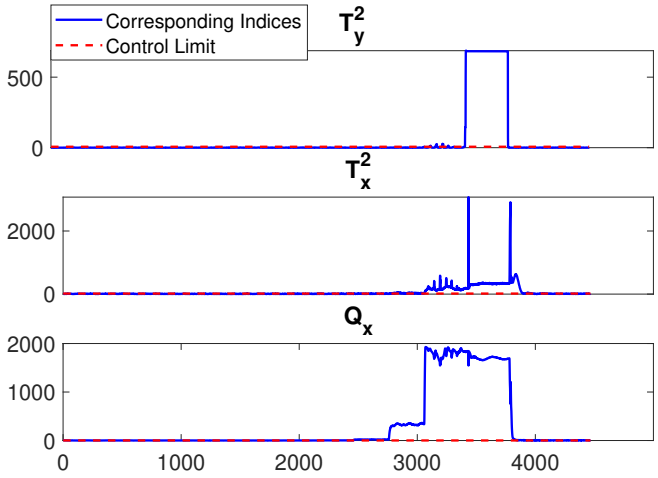


Figure 4.12: PCA-based Process and Quality Monitoring for F1_T2

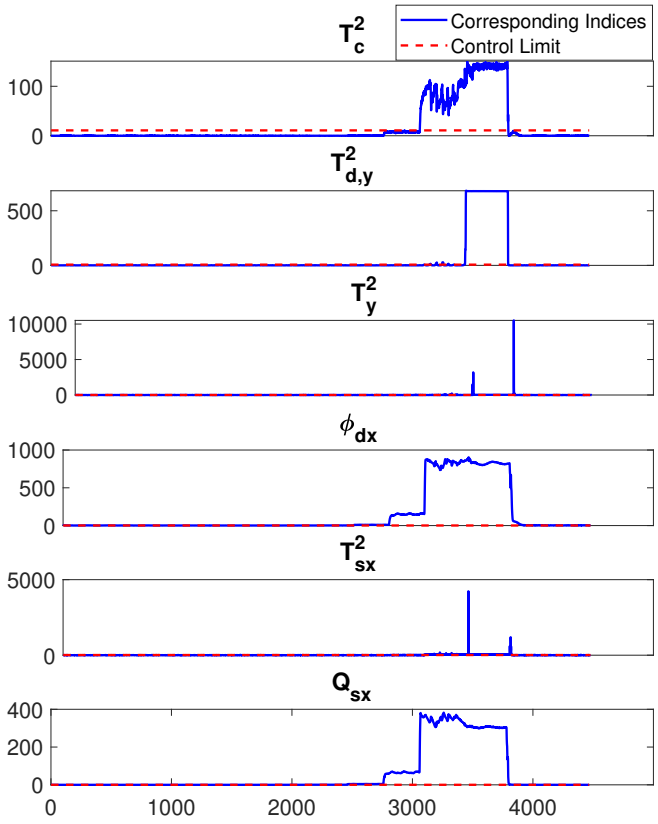


Figure 4.13: Monitoring Performance of ConDALVM for F1_T2

4.6 Conclusion

In this chapter, a dynamic auto-regressive latent variable model is proposed to capture the dynamic relations between the current quality score and the weighted combination of

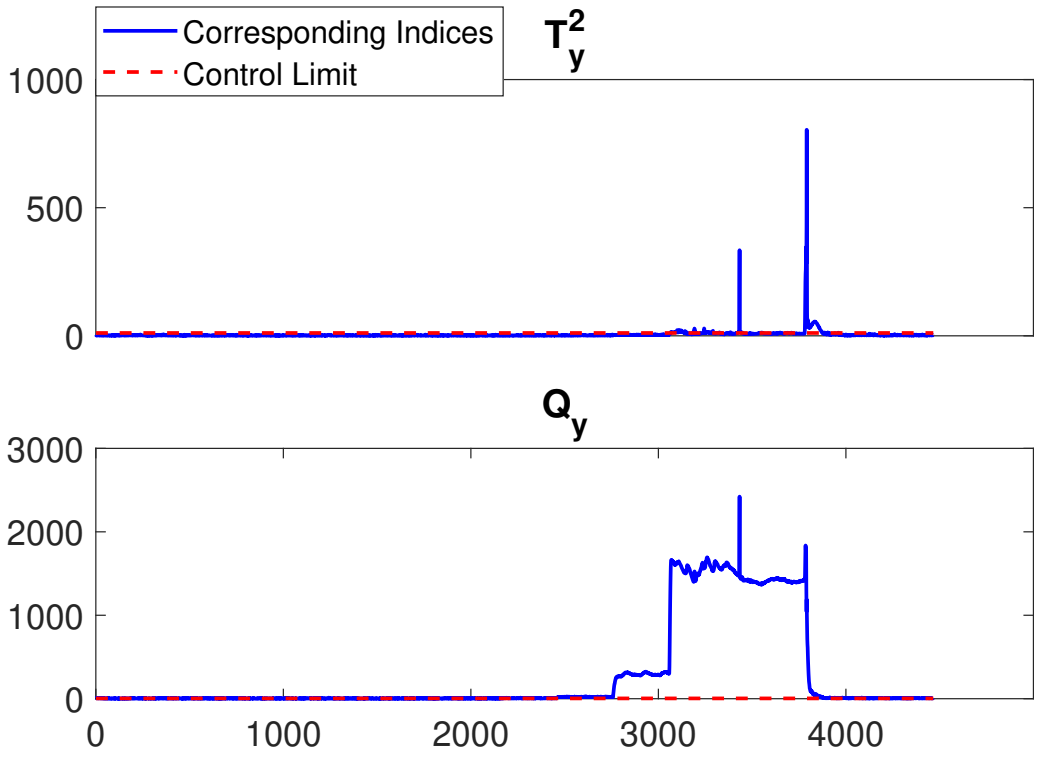


Figure 4.14: Monitoring Performance of DiPLS for F1_T2

the past process and quality scores. Both cross and auto-dynamic correlations of quality variables are extracted in DALVM. To achieve comprehensive static and dynamic modeling and monitoring of process and quality spaces, a further decomposition into subspaces is conducted on the extracted latent spaces of LVM. Both dynamic and static quality-relevant and process-relevant anomalies are explicitly detected in the concurrent DALVM-based monitoring system, with different alarming levels. A toy example, the Tennessee Eastman process, and a three-phase flow facility are used to compare the proposed methods with DiPLS and PLS, demonstrating the superiority of DALVM and ConDALVM for supervised modeling and monitoring.

Chapter 5

An Efficient Dynamic Auto-Regressive CCA for Time Series Imputation with Irregular Sampling

5.1 Overview

[1] To improve the prediction performance and efficiency of DALVM, an efficient dynamic auto-regressive canonical correlation analysis (EDACCA) method is proposed with a modified auto-regressive exogenous model to extract dynamics in both auto-correlations and cross-correlations. The flexibility and efficiency of EDACCA are improved with the design of weighting parameters and the economic singular value decomposition. EDACCA is further adapted for multi-step ahead (MS) prediction and missing data imputation. Two industrial processes are employed to evaluate the prediction performance and imputation performance of EDACCA.

5.2 Introduction

With the advent of Industrial 4.0, automated control systems become increasingly sophisticated and complex in industrial manufacturing processes. Data collected from these processes usually retain the characteristics of high dimensionality, correlation, and dynamics, which impose great challenges for data analytics. Meanwhile, quality data are of major concern in industries since they are directly related to industrial profit, but they are usually difficult to measure in a timely manner [226]. Thus supervised learning methods are preferred to infer quality variations from process variables. For instance, support vector regression (SVR) that was proposed for constructing an accurate regression model between

[1] Adapted from Xu, Bo, & Zhu, Q. (2022). An Efficient Dynamic Auto-Regressive CCA for Time Series Imputation With Irregular Sampling. IEEE Transactions on Automation Science and Engineering

input and output, it requires all training samples to be labeled [13]. To alleviate this requirement, semi-supervised SVR was proposed to use unlabeled samples to improve the prediction accuracy [188]. Liu et al. [98] developed a dynamic weighted probabilistic SVR, which further introduced Bayesian interpretation and a dynamic weighting strategy based on fuzzy similarity analysis to SVR.

Neural networks such as autoencoders (AEs) are also widely used for supervised predictive modeling and process monitoring. Different from traditional unsupervised AEs, a supervised loss in the hidden layer is introduced into supervised AE, and both the supervised loss and the loss for the reconstruction error are minimized [91, 5]. Supervised AEs achieve good modeling and monitoring performance due to their strong nonlinear modeling ability. To construct dynamic neural networks, Wang et al. [167] proposed an evolutionary AE where a dynamic network replaced the traditional nonlinear reconstruction model by introducing a temporal matrix. Although the aforementioned supervised learning methods have achieved various degrees of success, their extracted dynamic relations are hard to interpret. Agarwal et al proposed an explainability approach for auto-encoders based architectures but its accuracy depends on the availability of a large amount of data [3].

Supervised latent variable models (LVMs), on the other hand, retain good interpretability, and they are preferred to exploit large-scale time series data at a relatively low cost [127]. Partial least squares (PLS) and canonical correlation analysis (CCA) are two typical supervised LVMs. PLS extracts two sets of orthogonal score vectors by maximizing their covariance, while CCA finds the linear projections of input and output based on a maximum correlation criterion.

Dynamic extensions of LVM methods are widely studied to improve their modeling performance. Ku et al. [84] constructed an augmented matrix to include dynamics with lagged measurements. The augmentation idea was also extended to nonlinear LVMs [30, 62]. However, dynamics are not explicitly modeled in these methods, leading to poor interpretability of temporal relations. Alternatively, state space models were proposed with first-order autoregressive models [109, 170], but these methods failed to capture dynamics with higher orders since only first-order dynamics were considered. Li et al. [96] proposed a dynamic LVM in which an explicit high-order dynamic inner model was developed. Dong and Qin [51, 49] developed dynamic LVMs with consistent dynamic inner and outer models, and they are termed as dynamic inner PLS (DiPLS) and dynamic inner CCA (DiCCA). Basis functions were further incorporated into these models to address the redundant information contained in neighboring samples [183]. Although current dynamic methods obtain good modeling and monitoring performance, they do not fully exploit quality data, which otherwise can provide unique and important information for modeling. To overcome this issue, Xu and Zhu [185] proposed a dynamic auto-regressive PLS (DAPLS) method by introducing an auto-regressive exogenous model to capture dynamic auto-correlations and cross-correlations.

Due to their ability in extracting relations among variables, LVMs are widely used for missing data imputation to generate more samples for further monitoring and diagnosis. Missing data refers to cases when there are no values collected for some variables in the process, and it can be caused by issues such as sensor failures or irregular sampling rates.

There are several common kinds of missing data: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR) [9]. In manufacturing processes, MCAR is common when some sensors are broken, and the MNAR issue is more frequent than the other two, since the sampling rates of process variables are usually higher than those of quality variables [221]. To deal with these issues, various imputation methods are proposed to replace the missing data with substitute values. The most straightforward method was to substitute the data with the mean of the observed values of the corresponding variable [46]. Some advanced regression methods such as linear regression and random forest regression are also adapted to predict missing data by using observed values [205]. Recently deep learning-based imputation methods also receive wide attention due to their effectiveness. Generative adversarial imputation net (GAIN) is mainly composed of a generator and a discriminator where the generator imputes the missing samples based on observed samples and the discriminator adjusts the generator to minimize the identifiability between imputed and observed samples [189]. Variational autoencoder (VAE) is used as a generative tool to impute the missing samples since it can learn the distributions of observed variables [82]. Denoising autoencoder works well to learn the latent representations and associations of observed variables, and it is also used to generate clean samples from its missing version [64]. These models achieve various degrees of success, but there are still some issues that need to be resolved. First, they cannot obtain satisfactory performance when the missing data rate is high, which is the case when different sampling rates are used during data collection. Further, these methods only focus on maximizing the predictability of the latent representations, and remain temporal relations unexploited.

To address the aforementioned issues, an improved missing data imputation strategy based on a multi-step ahead prediction method is proposed in this work. First, an efficient dynamic auto-regressive CCA (EDACCA) is proposed to construct the dynamic relations between the current quality sample and a weighted combination of past processes and quality data. Two weighting parameters are introduced to improve its flexibility, and several techniques such as singular value decomposition are designed to enhance its efficiency. Further, to avoid the pollution brought by faulty data or to address the irregular sampling rates, a multi-step-ahead (MS) prediction model is designed based on EDACCA, which is subsequently adapted for irregular missing data imputation. Two industrial processes are used to illustrate the performance of EDACCA in terms of modeling and imputation performance. The main contributions of this chapter are summarized as follows.

- A novel dynamic auto-regressive extension of CCA is designed to realize a comprehensive decomposition and modeling of quality data space.
- An economic form of SVD and new constraints are introduced to address the rank deficiency and computational issues caused by matrix inversion.
- A multi-step ahead prediction framework is developed based on EDACCA to predict and control variables in advance, which lays the foundation for the subsequent missing data imputation strategy.

The remainder of this chapter is organized as follows. A brief review of CCA and DiCCA is summarized in Section 5.3. Section 5.4 presents the details of the proposed

EDACCA method and an EDACCA-based MS prediction imputation strategy. Two industrial processes, namely the Tennessee Eastman process and a wind process, are employed to demonstrate the effectiveness of the proposed methods in Section 5.5. Finally, Section 5.6 concludes the chapter.

5.3 Preliminaries

5.3.1 Canonical Correlation Analysis

Given $\mathbf{X} \in \mathbb{R}^{n\times m}$ with n samples and m process variables and $\mathbf{Y} \in \mathbb{R}^{n\times q}$ with q quality variables, CCA aims to maximize the correlation between $\mathbf{X}$ and $\mathbf{Y}$ as follows [68]

$$\begin{aligned} \max_{\mathbf{q},\mathbf{w}} \quad & J = \mathbf{q}^\top \mathbf{Y}^\top \mathbf{X}\mathbf{w} \\ \text{s.t.} \quad & \|\mathbf{X}\mathbf{w}\| = 1, \|\mathbf{Y}\mathbf{q}\| = 1 \end{aligned} \tag{5.1}$$

where $\mathbf{w}$ and $\mathbf{q}$ are the process and quality weighting vectors respectively, and $\mathbf{t} = \mathbf{X}\mathbf{w}$ and $\mathbf{u} = \mathbf{Y}\mathbf{q}$ are the corresponding process and quality latent score vectors. The singular value decomposition (SVD) or eigendecomposition method is utilized to solve the objective in Eq. (5.1) [68]. A regularized CCA was also proposed in [221] to improve its robustness in strongly colinear scenarios.

CCA decomposes $\mathbf{X}$ and $\mathbf{Y}$ into

$$\begin{cases} \mathbf{X} = \mathbf{T}\mathbf{P}^\top + \mathbf{E} \\ \mathbf{Y} = \mathbf{T}\mathbf{Q}^\top + \mathbf{F} \end{cases} \tag{5.2}$$

where $\mathbf{T} \in \mathbb{R}^{n\times l} = [\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_l]$, $\mathbf{P} \in \mathbb{R}^{m\times l} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_l]$, $\mathbf{Q} \in \mathbb{R}^{q\times l} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_l]$, l is the number of latent variables, and $\mathbf{E}$ and $\mathbf{F}$ are the residuals of process and quality variables respectively. For a new collected sample $\mathbf{x}$, the quality variables $\mathbf{y}$ can be predicted by

$$\hat{\mathbf{y}} = \mathbf{Q}\mathbf{R}^\top \mathbf{x} \tag{5.3}$$

$$\mathbf{R} = \mathbf{W}(\mathbf{P}^\top \mathbf{W})^{-1} \tag{5.4}$$

which can be used further for online monitoring and diagnosis [221, 223].

5.3.2 Supervised Dynamic inner CCA

Supervised DiCCA (sDiCCA) was proposed to improve the dynamic modeling performance of CCA [49]. Denote $\mathbf{x}_k$ and $\mathbf{y}_k$ as the process variables and quality variables collected at time k, and their corresponding latent variables are

$$\begin{aligned} t_k &= \mathbf{x}_k^\top \mathbf{w} \\ u_k &= \mathbf{y}_k^\top \mathbf{q} \end{aligned} \tag{5.5}$$

sDiCCA assumes the following weighted relations between u_k and t_k in its inner model.

$$u_k = \beta_0 t_k + \beta_1 t_{k-1} + \cdots + \beta_s t_{k-s} + r_k \tag{5.6}$$

where β_i $(i = 0, 1, \ldots, s)$ are the weights, and r_k is the regression error. The output latent variable $\hat{u}_k$ can then be predicted by

$$\hat{u}_k = \sum_{i=0}^{s} \beta_i t_{k-i} = \sum_{i=1}^{s} \beta_i \mathbf{x}_{k-i}^\top \mathbf{w} \tag{5.7}$$

To achieve dynamic consistency, the outer model of sDiCCA is designed to maximize the correlation between the process and quality scores, and it can be expressed in the following matrix form.

$$\begin{aligned} \max_{\mathbf{q},\mathbf{w},\boldsymbol{\beta}} \quad & \mathbf{q}^\top \mathbf{Y}_s^\top \mathbb{Z}_s[\boldsymbol{\beta} \otimes \mathbf{w}] \\ \text{s.t.} \quad & \|\mathbf{Y}_s \mathbf{q}\| = 1, \ \|\mathbb{Z}_s[\boldsymbol{\beta} \otimes \mathbf{w}]\| = 1 \end{aligned} \tag{5.8}$$

where

$$\begin{aligned} \mathbb{Z}_s \in \mathbb{R}^{N \times (s+1)m} &= [\mathbf{X}_s, \mathbf{X}_{s-1}, \cdots, \mathbf{X}_0] \\ \mathbf{Y}_s \in \mathbb{R}^{N \times p} &= [\mathbf{y}_s, \mathbf{y}_{s+1}, \cdots, \mathbf{y}_{s+N}]^\top \\ \boldsymbol{\beta} \in \mathbb{R}^{s+1} &= [\beta_0, \beta_1, \cdots, \beta_s]^\top \end{aligned} \tag{5.9}$$

and $\mathbf{X}_i \in \mathbb{R}^{N \times m} = [\mathbf{x}_i, \mathbf{x}_{i+1}, \cdots, \mathbf{x}_{i+N}]^\top (i = 0, 1, \cdots, s)$. $\otimes$ is the kronecker product, and $\|\cdot\|$ denotes the L_2 norm. It is noted that Eq. (5.8) is equivalent to

$$\max_{\mathbf{q},\mathbf{w},\boldsymbol{\beta}} \quad \frac{\mathbf{q}^\top \mathbf{Y}_s^\top \mathbb{Z}_s[\boldsymbol{\beta} \otimes \mathbf{w}]}{\|\mathbf{Y}_s \mathbf{q}\| \|\mathbb{Z}_s[\boldsymbol{\beta} \otimes \mathbf{w}]\|}$$

The model parameters in Eq. (5.8) can be obtained with the aid of Lagrange multipliers [49].

Remark 5 *In the original work [49], DiCCA was proposed as an unsupervised method, where u_k in Eq. (5.7) is replaced with the future process score t_{k+1}, and the outer and inner relations are adapted into*

$$\begin{aligned} \max_{\mathbf{w},\boldsymbol{\beta}} \quad & \mathbf{w}^\top \mathbf{X}_{s+1}^\top \mathbb{Z}_s[\boldsymbol{\beta} \otimes \mathbf{w}] \\ \text{s.t.} \quad & \|\mathbf{X}_{s+1} \mathbf{w}\| = 1, \ \|\mathbb{Z}_s[\boldsymbol{\beta} \otimes \mathbf{w}]\| = 1 \end{aligned} \tag{5.10}$$

$$t_{k+1} = \beta_1 t_k + \beta_2 t_{k-1} + \cdots + \beta_s t_{k-s} + r_k \tag{5.11}$$

More details of unsupervised DiCCA can be found in [49].

5.4 Methodology

5.4.1 Traditional Dynamic Auto-Regressive CCA

In Eqs. (5.7) and (5.8), only the weighted relations of process and quality variables are modeled in sDiCCA. Its supervised prediction performance is limited since a large portion of quality information remains unexploited. Alternatively, a dynamic autoregressive CCA (DACCA) method is proposed in this work with the aim to fully exploit valuable information in both process and quality data. In addition to the weighted relation between u_k and process latent variables, an auto-regressive exogenous model is adapted in DACCA to learn the dynamic relations among quality data simultaneously as follows.

$$\begin{aligned} u_k = & \quad \beta_0 t_k + \beta_1 t_{k-1} + \ldots + \beta_s t_{k-s} \\ + & \quad \delta_1 u_{k-1} + \ldots + \delta_d u_{k-d} + r_k \end{aligned} \tag{5.12}$$

where s and d are the dynamics order for process and quality variables respectively. The prediction of u_k is

$$\hat{u}_k = \sum_{i=0}^{s} \beta_i \mathbf{x}_{k-i}^\top \mathbf{w} + \sum_{j=1}^{d} \delta_j \mathbf{y}_{k-j}^\top \mathbf{q}$$

To ensure that both dynamic auto-correlations and cross-correlations are captured, the objective of DACCA is designed to minimize the norm between $\hat{u}_k$ and $\sum_{i=0}^{s} \beta_i \mathbf{x}_{k-i}^\top \mathbf{w}$ and $\sum_{j=1}^{d} \delta_j \mathbf{y}_{k-j}^\top \mathbf{q}$ simultaneously. In a matrix form, it is expressed as

$$\max_{\mathbf{w},\mathbf{q},\boldsymbol{\beta},\boldsymbol{\delta}} \quad \frac{\mathbf{q}^\top \mathbf{Y}_g^\top \mathbb{Z}_x \left(\boldsymbol{\beta} \otimes \mathbf{w}\right)}{\|\mathbf{Y}_g \mathbf{q}\| \|\mathbb{Z}_x \left(\boldsymbol{\beta} \otimes \mathbf{w}\right)\|} + \frac{\mathbf{q}^\top \mathbf{Y}_g^\top \mathbb{Z}_y \left(\boldsymbol{\delta} \otimes \mathbf{q}\right)}{\|\mathbf{Y}_g \mathbf{q}\| \|\mathbb{Z}_y \left(\boldsymbol{\delta} \otimes \mathbf{q}\right)\|}$$

Or, equivalently as

$$\begin{aligned} \max_{\mathbf{w},\mathbf{q},\boldsymbol{\beta},\boldsymbol{\delta}} \quad & \mathbf{q}^\top \mathbf{Y}_g^\top \left[\mathbb{Z}_x \left(\boldsymbol{\beta} \otimes \mathbf{w}\right) + \mathbb{Z}_y \left(\boldsymbol{\delta} \otimes \mathbf{q}\right)\right] \\ \text{s.t.} \quad & \|\mathbf{Y}_g \mathbf{q}\| = 1, \quad \|\mathbb{Z}_x \left(\boldsymbol{\beta} \otimes \mathbf{w}\right)\| = 1, \quad \|\mathbb{Z}_y \left(\boldsymbol{\delta} \otimes \mathbf{q}\right)\| = 1 \end{aligned} \tag{5.13}$$

where

$$\begin{aligned} &\mathbb{Z}_x \in \mathbb{R}^{N \times (s+1)m} = [\mathbf{X}_g, \mathbf{X}_{g-1}, \ldots, \mathbf{X}_{g-s}] \\ &\mathbb{Z}_y \in \mathbb{R}^{N \times (dq)} = [\mathbf{Y}_{g-1}, \mathbf{Y}_{g-2}, \ldots, \mathbf{Y}_{g-d}] \\ &\mathbf{X}_i \in \mathbb{R}^{N \times m} = [\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+N}]^\top, i = \{0, 1, \ldots, g\} \\ &\mathbf{Y}_i \in \mathbb{R}^{N \times q} = [\mathbf{y}_i, \mathbf{y}_{i+1}, \ldots, \mathbf{y}_{i+N}]^\top, i = \{0, 1, \ldots, g\} \\ &\boldsymbol{\beta} \in \mathbb{R}^{s+1} = [\beta_0, \beta_1, \ldots, \beta_s]^\top \\ &\boldsymbol{\delta} \in \mathbb{R}^{d} = [\delta_1, \delta_2, \ldots, \delta_d]^\top \end{aligned}$$

$g = \max(s, d)$ is the higher dynamics order of s and d. The outer model of DACCA can be divided into two parts. The first part is to capture the dynamic cross-correlations between

current quality variables and past and current process variables, while the second part is to extract the auto-correlations between current quality samples and past quality samples. Thus, in DACCA, the output, namely the current quality samples, can be predicted by i. past and current process samples, and ii. past quality samples.

DACCA obtains better predictive modeling performance than most dynamic algorithms. However, several issues are involved in its objective design. Firstly, the constraints imposed in Eq. (5.13) assign equal weights on process and quality data, and the extracted dynamic relations may not be accurate. Further, the model parameters need to be obtained in an iterative way, and it is not efficient since matrix inversion needs to be performed in every iteration where rank-deficient issues may exist. To address these limitations an efficient DACCA (EDACCA) method is proposed to address these issues.

5.4.2 Efficient Dynamic Auto-Regressive CCA

In EDACCA, the objective is to maximize the covariance between $\mathbf{Y}_g\mathbf{q}$ and $\mathbb{Z}_x\left(\boldsymbol{\beta}\otimes\mathbf{w}\right)$ and the correlation between $\mathbf{Y}_g\mathbf{q}$ and $\mathbb{Z}_y\left(\boldsymbol{\delta}\otimes\mathbf{q}\right)$. Then the main objective and constraints in Eq. (5.13) are revised into

$$\begin{aligned}\max_{\mathbf{w},\mathbf{q},\boldsymbol{\beta},\boldsymbol{\delta}}\quad & \mathbf{q}^\top\mathbf{Y}_g^\top\left[k_1\mathbb{Z}_x\left(\boldsymbol{\beta}\otimes\mathbf{w}\right)+k_2\mathbb{Z}_y\left(\boldsymbol{\delta}\otimes\mathbf{q}\right)\right] \\ \text{s.t.}\quad & \|\mathbf{Y}_g\mathbf{q}\|=1,\quad \|\mathbb{Z}_y\left(\boldsymbol{\delta}\otimes\mathbf{q}\right)\|=1,\quad \|\mathbf{w}\|=1,\quad \|\boldsymbol{\beta}\|=1\end{aligned} \tag{5.14}$$

where the constraints for $\mathbf{Y}_g\mathbf{q}$ and $\mathbb{Z}_y\left(\boldsymbol{\delta}\otimes\mathbf{q}\right)$ are designed based on DiCCA as shown in Eq. (5.10). The constraints for $\mathbf{w}$ and $\boldsymbol{\beta}$ aim to restrict weighted parameters for better convergence. k_1 and k_2 are weights of cross-correlations and auto-correlations, and they are selected through joint cross-validation. The incorporation of k_1 and k_2 achieves additional flexibility in the dynamic modeling of process and quality data.

To obtain the model parameters of EDACCA, Lagrange multipliers λ_1, λ_2, λ_3, and λ_4 are introduced, and the following Lagrange function is formulated.

$$\begin{aligned}\mathcal{L}=&\,\mathbf{q}^\top\mathbf{Y}_g^\top\left[k_1\mathbb{Z}_x\left(\boldsymbol{\beta}\otimes\mathbf{w}\right)+k_2\mathbb{Z}_y\left(\boldsymbol{\delta}\otimes\mathbf{q}\right)\right]\\&+\frac{\lambda_1}{2}\left(1-\mathbf{q}^\top\mathbf{Y}_g^\top\mathbf{Y}_g\mathbf{q}\right)+\frac{\lambda_2}{2}\left(1-\mathbf{w}^\top\mathbf{w}\right)\\&+\frac{\lambda_3}{2}\left(1-\boldsymbol{\beta}^\top\boldsymbol{\beta}\right)+\frac{\lambda_4}{2}\left(1-\mathbf{q}^\top\mathbf{Y}_\delta^\top\mathbf{Y}_\delta\mathbf{q}\right)\end{aligned} \tag{5.15}$$

Solving the Lagrange function $\mathcal{L}$ leads to

$$\mathbf{w}\propto\mathbf{X}_\beta^\top\mathbf{Y}_g\mathbf{q}; \tag{5.16}$$

$$\begin{aligned}\mathbf{q}\propto&\left(\mathbf{Y}_g^\top\mathbf{Y}_g+\mathbf{Y}_\delta^\top\mathbf{Y}_\delta\right)^{-1}\\&\left(k_1\mathbf{Y}_g^\top\mathbf{X}_\beta\mathbf{w}+k_2\mathbf{Y}_g^\top\mathbf{Y}_\delta\mathbf{q}+\mathbf{Y}_\delta^\top\mathbf{Y}_g\mathbf{q}\right);\end{aligned} \tag{5.17}$$

$$\boldsymbol{\beta}\propto\mathbf{T}_s^\top\mathbf{Y}_g\mathbf{q}; \tag{5.18}$$

$$\boldsymbol{\delta}\propto\left(\mathbf{U}_d^\top\mathbf{U}_d\right)^{-1}k_2\mathbf{U}_d^\top\mathbf{Y}_g\mathbf{q}; \tag{5.19}$$

where

$$\mathbf{X}_\beta = \mathbb{Z}_x\left(\boldsymbol{\beta} \otimes \mathbf{I}_m\right) = \sum_{i=0}^{s} \beta_i \mathbf{X}_{g-i}$$

$$\mathbf{Y}_\delta = \mathbb{Z}_y\left(\boldsymbol{\delta} \otimes \mathbf{I}_q\right) = \sum_{i=1}^{d} \delta_i \mathbf{Y}_{g-i}$$

$$\mathbf{T}_s = \mathbb{Z}_x\left(\mathbf{I}_{s+1} \otimes \mathbf{w}\right) = [\mathbf{t}_g, \mathbf{t}_{g-1}, \ldots, \mathbf{t}_{g-s}]$$

$$\mathbf{U}_d = \mathbb{Z}_y\left(\mathbf{I}_d \otimes \mathbf{q}\right) = [\mathbf{u}_{g-1}, \mathbf{u}_{g-2}, \ldots, \mathbf{u}_{g-d}]$$

If Eqs. (5.16) - (5.19) are calculated iteratively to obtain model parameters, and matrix inversions must be calculated in each iteration, leading to inefficient computation performance. Furthermore, after one set of model parameters is obtained, deflation must be performed on $\mathbf{X}$ and $\mathbf{Y}$ to remove their effect, which may make $\mathbf{X}$ and $\mathbf{Y}$ rank-deficient. In this case, the matrix inversion $\left(\mathbf{Y}_g^\top \mathbf{Y}_g + \mathbf{Y}_\delta^\top \mathbf{Y}_\delta\right)^{-1}$ in Eqs. (5.17) would lead to ill-conditioned problems.

To deal with the inefficient matrix inversion and handle the ill-conditioning problem, an economic form of SVD, which can remove extra rows or columns of zeros from the diagonal matrix of singular values, is performed to pre-process the input and output/quality matrices as follows:

$$\mathbf{X}_g = \mathbf{U}_{gx}\mathbf{D}_x\mathbf{V}_x^\top \tag{5.20}$$

$$\mathbf{Y}_g = \mathbf{U}_{gy}\mathbf{D}_y\mathbf{V}_y^\top \tag{5.21}$$

where the diagonal matrices $\mathbf{D}_x$ and $\mathbf{D}_y$ only contain nonzero singular values, and $\mathbf{U}_{gx}$, $\mathbf{U}_{gy}$, $\mathbf{V}_x$ and $\mathbf{V}_y$ include their orthogonal singular vectors that correspond to nonzero singular values. Then the latent variables $\mathbf{t}$ and $\mathbf{u}$ can be expressed as

$$\mathbf{t} = \mathbf{X}\mathbf{w} = \mathbf{U}_x\mathbf{w}_u \tag{5.22}$$

$$\mathbf{u} = \mathbf{Y}\mathbf{q} = \mathbf{U}_y\mathbf{q}_u \tag{5.23}$$

where $\mathbf{U}_x = \mathbf{X}\mathbf{V}_x\mathbf{D}_x^{-1}$, $\mathbf{U}_y = \mathbf{Y}\mathbf{V}_y\mathbf{D}_y^{-1}$, $\mathbf{w}_u = \mathbf{D}_x\mathbf{V}_x^\top\mathbf{w}$, and $\mathbf{q}_u = \mathbf{D}_y\mathbf{V}_y^\top\mathbf{q}$. The EDACCA model is developed once the model parameters $\mathbf{w}_u$, $\mathbf{q}_u$, $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ are learned. Thus, combined with the constraints in Eq. (5.14), the model relations in Eqs. (5.16) - (5.19) are updated into

$$\mathbf{w} = k_1\mathbf{X}_\beta^\top\mathbf{u}_g, \quad \mathbf{w} = \mathbf{w}/\|\mathbf{w}\|; \tag{5.24}$$

$$\mathbf{w}_u = \mathbf{D}_x\mathbf{V}_x^\top\mathbf{w}; \tag{5.25}$$

$$\begin{aligned}\mathbf{q}_u = &\left(\mathbf{U}_{gy}^\top\mathbf{U}_{gy} + \mathbf{U}_\delta^\top\mathbf{U}_\delta\right)^{-1} \\ &\left(k_1\mathbf{U}_{gy}^\top\mathbf{U}_\beta\mathbf{w}_u + k_2\mathbf{U}_{gy}^\top\mathbf{U}_\delta\mathbf{q}_u + \mathbf{U}_\delta^\top\mathbf{U}_{gy}\mathbf{q}_u\right);\end{aligned} \tag{5.26}$$

$$\mathbf{q}_u = \mathbf{q}_u/\|\mathbf{U}_y\mathbf{q}_u\|; \tag{5.27}$$

$$\mathbf{u}_g = \mathbf{U}_{gy}\mathbf{q}_u; \tag{5.28}$$

$$\boldsymbol{\beta} = k_1\mathbf{T}_s^\top\mathbf{u}_g, \quad \boldsymbol{\beta} = \boldsymbol{\beta}/\|\boldsymbol{\beta}\|; \tag{5.29}$$

$$\boldsymbol{\delta} = \left(\mathbf{U}_d^\top\mathbf{U}_d\right)^{-1}k_2\mathbf{U}_d^\top\mathbf{u}_g, \quad \boldsymbol{\delta} = \boldsymbol{\delta}/\left(\mathbf{u}_g^\top\mathbf{U}_d\boldsymbol{\delta}\right)^{0.5}. \tag{5.30}$$

where

$$\mathbf{U}_\beta = \sum_{i=0}^{s} \beta_i \mathbf{X}_{g-i} \mathbf{V}_x \mathbf{D}_x^{-1}, \mathbf{U}_\delta = \sum_{i=1}^{d} \delta_i \mathbf{Y}_{g-i} \mathbf{V}_y \mathbf{D}_y^{-1} \tag{5.31}$$

$$\mathbf{U}_d = [\mathbf{u}_{g-1}, \mathbf{u}_{g-2}, \ldots, \mathbf{u}_{g-d}] \tag{5.32}$$

$$\mathbf{T}_s = [\mathbf{t}_g, \mathbf{t}_{g-1}, \ldots, \mathbf{t}_{g-s}] \tag{5.33}$$

$\mathbf{u}_i$ $(i = g-1, g-2, \ldots, g-d)$ are sub-vectors of $\mathbf{u} = \mathbf{U}_y \mathbf{q}_u$, and $\mathbf{t}_i$ $(i = g, g-1, \ldots, g-s)$ are from $\mathbf{t} = \mathbf{U}_x \mathbf{w}_u$. Due to the orthogonality of $\mathbf{U}_{yg}$, $\mathbf{U}_{gy}^\top \mathbf{U}_{gy} = \mathbf{I}$ can address local convergence or ill-conditioned problems caused by matrix inversion $\left(\mathbf{Y}_g^\top \mathbf{Y}_g + \mathbf{Y}_\delta^\top \mathbf{Y}_\delta\right)^{-1}$ in Eqs. (5.17). It is noted that Eqs. (5.24) - (5.30) define the outer structure of EDACCA.

To achieve consistency of the inner and outer models, a modified auto-regressive exogenous model is designed in the inner model of EDACCA.

$$\hat{\mathbf{u}}_g = k_1 \mathbf{T}_s \boldsymbol{\alpha} + k_2 \mathbf{U}_d \boldsymbol{\gamma} \tag{5.34}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ are weighting vectors.

After performing inner and outer models, one pair of dynamic latent variables is extracted, and their effect can be removed by deflation on $\mathbf{X}$ and $\mathbf{Y}$. The deflation of $\mathbf{X}$ is

$$\mathbf{X} := \mathbf{X} - \mathbf{t}\mathbf{p}^\top \tag{5.35}$$

where $\mathbf{p} = \mathbf{X}^\top \mathbf{t} / \mathbf{t}^\top \mathbf{t}$.

It is noted that only $\mathbf{Y}_g \equiv \{\mathbf{y}_i\}_{i=g}^n$ can be predicted with an auto-regressive exogenous model, while for $\mathbf{Y}_{0:g-1} \equiv \{\mathbf{y}_i\}_{i=0}^{g-1}$, the previous g samples are not available, and the static model should be adopted to deflate $\mathbf{Y}_{0:g-1}$. Thus, the deflation of $\mathbf{Y}$ is separated into two parts, where $\mathbf{Y}_{0:g-1}$ is deflated by

$$\mathbf{Y}_{0:g-1} := \mathbf{Y}_{0:g-1} - \hat{\mathbf{u}}_{0:g-1} \mathbf{q}^\top \tag{5.36}$$

where $\hat{\mathbf{u}}_{0:g-1} = b\mathbf{t}_{0:g-1}$, $b = \mathbf{u}_{0:g-1}^\top \mathbf{t}_{0:g-1} / \mathbf{t}_{0:g-1}^\top \mathbf{t}_{0:g-1}$, $\mathbf{t}_{0:g-1} \equiv \{\mathbf{t}_i\}_{i=0}^{g-1}$, and $\mathbf{u}_{0:g-1} \equiv \{\mathbf{u}_i\}_{i=0}^{g-1}$.

For $\mathbf{Y}_g$, its deflation is designed to minimize $\|\mathbf{Y}_g - \hat{\mathbf{u}}_g \mathbf{c}^\top\|$, and it is deflated by

$$\mathbf{Y}_g := \mathbf{Y}_g - \hat{\mathbf{u}}_g \mathbf{c}^\top \tag{5.37}$$

where $\mathbf{c} = \mathbf{Y}_g^\top \hat{\mathbf{u}}_g / \hat{\mathbf{u}}_g^\top \hat{\mathbf{u}}_g$.

Then the next sets of latent variables can be extracted iteratively with the deflated matrices. The details of the EDACCA algorithm are summarized in Algorithm 3, and the weight parameters k_1 and k_2, the dynamics order s and d, and the number of latent scores l are determined with a joint cross-validation method.

Algorithm 3 The EDACCA Algorithm

1. Preprocess $\mathbf{X}$ and $\mathbf{Y}$ by initializing them to zero mean and unit variance, select the first column of $\mathbf{Y}$ as $\mathbf{u}$ and initialize $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ as random vectors with the unit norm.

2. Perform economic SVD on $\mathbf{X}_g$ and $\mathbf{Y}_g$ and obtain the unitary matrices for $\mathbf{X}$ and $\mathbf{Y}$.

$$\begin{aligned} \mathbf{X}_g &= \mathbf{U}_{gx}\mathbf{D}_x\mathbf{V}_x^\top, \quad & \mathbf{Y}_g &= \mathbf{U}_{gy}\mathbf{D}_y\mathbf{V}_y^\top \\ \mathbf{U}_x &= \mathbf{X}\mathbf{V}_x\mathbf{D}_x^{-1}, \quad & \mathbf{U}_y &= \mathbf{Y}\mathbf{V}_y\mathbf{D}_y^{-1} \end{aligned}$$

3. Iterate the following steps until convergence is achieved.

   i. Form $\mathbf{U}_\beta$, $\mathbf{U}_\delta$ and $\mathbf{U}_d$ with Eqs. (5.31) - (5.32);

   ii. Obtain model parameters $\mathbf{w}_u$, $\mathbf{q}_u$, $\boldsymbol{\beta}$ and $\boldsymbol{\sigma}$ with the relations in Eqs. (5.24) - (5.30).

4. Construct an auto-regressive exogenous model with Eq. (5.34) to obtain the inner model parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, and $\hat{\mathbf{u}}_g$ can be obtained by $\hat{\mathbf{u}}_g = k_1\mathbf{T}_s\boldsymbol{\alpha} + k_2\mathbf{U}_d\boldsymbol{\gamma}$.

5. Deflate $\mathbf{X}$ and $\mathbf{Y}$ with Eqs. (5.35) - (5.37) respectively.

6. Repeat Steps 2 - 4 until l sets of latent scores are extracted.

Remark 6 *In industrial applications, sometimes important quality variables are required to be predicted and controlled in advance. To achieve this kind of task, the objective of EDACCA shown in Eq. (5.14) can be rewritten as*

$$\begin{aligned} &\max_{\mathbf{w},\mathbf{q},\boldsymbol{\beta},\boldsymbol{\delta}} \quad \mathbf{q}^\top\mathbf{Y}_g^\top\left[k_1\mathbb{X}_e\left(\boldsymbol{\beta}\otimes\mathbf{w}\right) + k_2\mathbb{Y}_e\left(\boldsymbol{\delta}\otimes\mathbf{q}\right)\right] \\ &s.t. \quad \|\mathbf{Y}_g\mathbf{q}\| = 1, \quad \|\mathbb{Y}_e\left(\boldsymbol{\delta}\otimes\mathbf{q}\right)\| = 1, \quad \|\mathbf{w}\| = 1, \quad \|\boldsymbol{\beta}\| = 1 \end{aligned} \tag{5.38}$$

where e is the number of steps in advance, $g = \max(s+e, d+e)$ and

$$\begin{aligned} \mathbb{X}_e &\in \mathbb{R}^{N\times sm} = [\mathbf{X}_{g-e}, \mathbf{X}_{g-1-e}, \ldots, \mathbf{X}_{g-s+1-e}] \\ \mathbb{Y}_e &\in \mathbb{R}^{N\times dq} = [\mathbf{Y}_{g-e}, \mathbf{Y}_{g-1-e}, \ldots, \mathbf{Y}_{g-d+1-e}] \\ \boldsymbol{\beta} &\in \mathbb{R}^s = [\beta_1, \beta_2, \ldots, \beta_s]^\top \\ \boldsymbol{\delta} &\in \mathbb{R}^d = [\delta_1, \delta_2, \ldots, \delta_d]^\top \end{aligned}$$

Compared with traditional EDACCA, only $\mathbb{Z}_x$ and $\mathbb{Z}_y$ are replaced by $\mathbb{X}_e$ and $\mathbb{Y}_e$ separately.

5.4.3 Multi-Step-Ahead Prediction with EDACCA

Eq. (5.12) provides predictions by assuming the availability of the current and past s process data, and the past d quality data. However, different sampling rates may be

adopted for various sensors, leading to the unavailability of some slow sampling variables. Further, the modeling and monitoring performance may get deteriorated if some variables are polluted by faults. Thus, it is necessary to realize multi-step-ahead (MS) predictions to get rid of the regular sampling constraint and the pollution of faulty data. To achieve this goal, EDACCA is extended to an MS prediction scheme, which is designed based on Eq. (5.12) as follows.

1. One Step ahead prediction. To achieve a strict one-step ahead prediction model, the use of the current process sample t_k is removed from Eq. (5.12), and the dynamic relation is rewritten as

$$\hat{\mathbf{u}}_{k|k-1} = k_1 \sum_{i=1}^{s} \mathbf{B}_i \mathbf{t}_{k-i} + k_2 \sum_{i=1}^{d} \mathbf{\Delta}_i \mathbf{u}_{k-i} \tag{5.39}$$

   where k_1 and k_2 have the same meaning as those in Eq. 5.14, and $\mathbf{B}_i$ and $\mathbf{\Delta}_i$ are the dynamic cross-correlation and auto-correlation coefficients. Eq. (5.39) can provide accurate prediction results only when the process latent scores at time step $k-1, k-2, \ldots, k-s$, and quality scores at time step $k-1, k-2, \ldots, k-d$ are available and faulty-free.

2. Two steps ahead prediction. When the previous sample u_{k_1} is not available due to irregular sampling rates, or it is polluted by faults, a two-step ahead prediction model can be designed.

$$\begin{aligned} \hat{\mathbf{u}}_{k|k-2} = k_1 \sum_{i=1}^{s} \mathbf{B}_i \mathbf{t}_{k-i} \\ + k_2 \left(\mathbf{\Delta}_1 \hat{\mathbf{u}}_{k-1|k-2} + \sum_{i=2}^{d} \mathbf{\Delta}_i \mathbf{u}_{k-i} \right) \end{aligned} \tag{5.40}$$

   where u_{k_1} is substituted with the prediction $\hat{\mathbf{u}}_{k-1|k-2}$ from one-step ahead predictive model, and $\mathbf{\Delta}_1$ is the auto-correlation coefficient for $\hat{\mathbf{u}}_{k-1|k-2}$. Note that in Eq. (5.40), only MS predictions of auto-correlations are considered, and it has wide applications when quality variables are sampled with a smaller frequency than process ones.

3. h-step ahead prediction. Similarly, the h ($h > 2$) steps ahead prediction can be calculated as

$$\begin{aligned} \hat{\mathbf{u}}_{k|k-h} = k_1 \sum_{i=1}^{s} \mathbf{B}_i \mathbf{t}_{k-i} \\ + k_2 \left(\sum_{i=1}^{h-1} \mathbf{\Delta}_i \hat{\mathbf{u}}_{k-i|k-h} + \sum_{i=h}^{d} \mathbf{\Delta}_i \mathbf{u}_{k-i} \right) \end{aligned} \tag{5.41}$$

   where $h \leq g$ in this work.

5.4.4 Missing Data Imputation with EDACCA-based MS Prediction

EDACCA can accurately extract the dynamic relations in the data, and the developed models can be subsequently applied for multiple purposes, such as fault monitoring and diagnosis as developed in [185]. Furthermore, to address the irregular sampling issues in practical industrial processes, the developed MS prediction scheme is adopted as a missing data imputation strategy to deal with the missing data issue in quality variables. The general procedure is outlined as follows.

a. *Initialization:* The model parameters k_1, k_2, s, d, and l are selected for EDACCA by using training data without missing values, and the dedicated EDACCA model is trained with the selected parameters.

b. *Imputation:* The EDACCA model imputes the missing quality variables $\hat{\mathbf{y}}_{k,\text{miss}}$ based on the corresponding process variables and past process and quality samples.

$$\hat{\mathbf{y}}_{k,\text{miss}} = \left(k_1 \sum_{i=0}^{s} \mathbf{t}_{g-i}\boldsymbol{\alpha}_i + k_2 \sum_{i=1}^{d} \mathbf{u}_{g-i}\boldsymbol{\gamma}_i \right) \mathbf{C}^\top \tag{5.42}$$

where

$$\begin{aligned} \mathbf{t} &= \mathbf{x}\mathbf{R} \\ \mathbf{u} &= \mathbf{y}\mathbf{C} \\ \mathbf{R} &= \mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1} \\ \mathbf{C} &= [\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_l] \end{aligned}$$

$\mathbf{x}$ and $\mathbf{y}$ can be the original observed samples or the imputed samples. The imputation is performed in sequential order until all quality data are imputed. It is noted that when the sampling discrepancy between process and quality data is large, more than one quality sample might be missing in the collected data. Thus, the Multi-Step (MS) predictive model should be adopted instead to predict some quality data in Eq. (5.42). $\hat{\mathbf{y}}_{k,\text{miss}}$ should be predicted with

$$\begin{aligned} \hat{\mathbf{y}}_{k,\text{miss}} = \Bigg(& k_1 \sum_{i=0}^{s} \mathbf{t}_{g-i}\boldsymbol{\alpha}_i \\ & + k_2 \left(\sum_{i=1}^{h-1} \hat{\mathbf{u}}_{k-i|k-h}\boldsymbol{\gamma}_i + \sum_{i=h}^{d} \mathbf{u}_{k-i}\boldsymbol{\gamma}_i \right) \Bigg) \mathbf{C}^\top \end{aligned} \tag{5.43}$$

where h is determined based on the discrepancy of sampling frequencies between quality and input variables. To make sure that Eq. (5.43) can run normally, the first missing sample that can be imputed should be imputed by one step ahead of the normal EDACCA algorithm based on time parameters s and d.

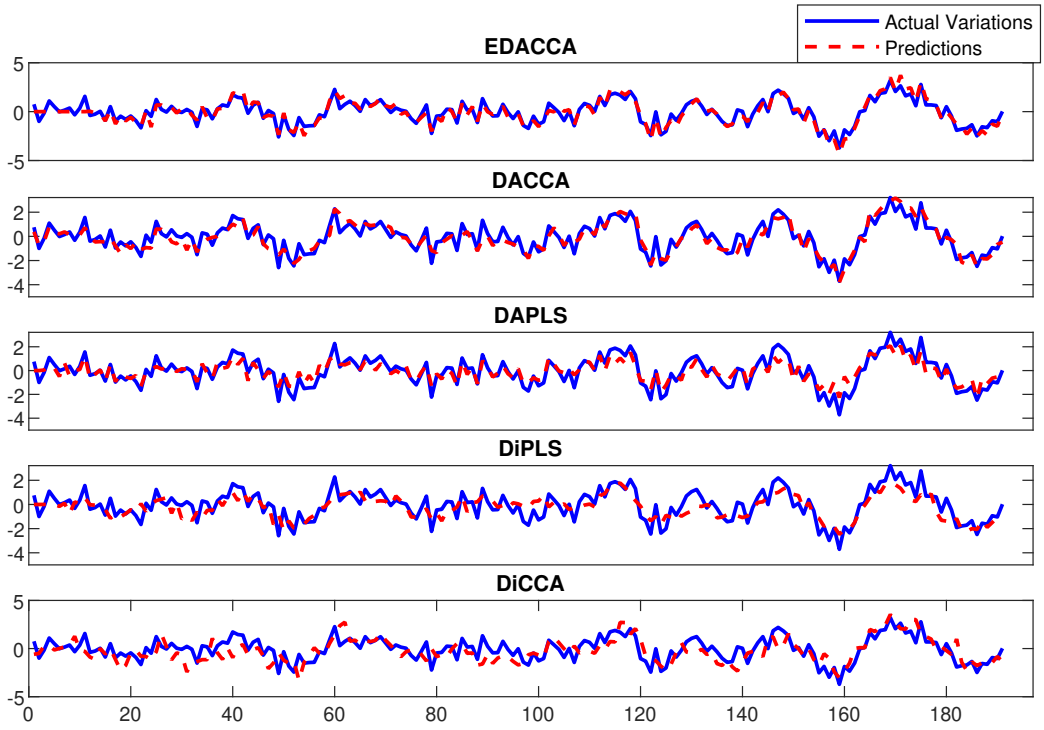


Figure 5.1: Dynamic Modeling Performance of Different Models for TEP

c. *Predictive modeling:* After samples without quality labels are imputed, new training data is obtained, which can be employed for subsequent predictive modeling, monitoring, and diagnosis with other models such as DiPLS [49] or neural networks [192].

Remark 7 *The proposed EDACCA-based imputation method is suitable when different frequencies exist for process and quality variables, which is very common in manufacturing processes since the measurements of quality variables such as product compositions are usually less frequently sampled and have associated delays.*

5.5 Case Studies

In this section, two industrial processes, namely the Tennessee Eastman process (TEP) and a wind process, are used to evaluate the modeling and imputation effectiveness of the proposed EDACCA algorithm.

5.5.1 Tennessee Eastman Process

TEP [54] has been widely used to illustrate the modeling and monitoring performance of different models in the area of process systems engineering. Based on the properties

Table 5.1: Prediction Results of Different Models for TEP

	EDACCA	DACCA	DAPLS	DiPLS	sDiCCA
MSEs	**0.1620**	0.3196	0.2462	0.6316	0.8896
R^2	**0.9479**	0.8945	0.9237	0.7560	0.5976

Table 5.2: MSEs of $\hat{\mathbf{Y}}_{xg}$ and $\hat{\mathbf{Y}}_{yg}$ for TEP

	EDACCA	DACCA	DAPLS	sDiCCA	DiCCA
$\hat{\mathbf{Y}}$	**0.1620**	0.3196	0.2462	0.8896	1.0465
$\hat{\mathbf{Y}}_{xg}$	1.3517	0.9186	1.0872	**0.8896**	-
$\hat{\mathbf{Y}}_{yg}$	**0.1056**	0.3660	0.5246	-	1.0465

of the measurements, the 53 measured variables are divided into 41 measured variables XMEAS(1-41) and 12 manipulated variables XMV(1-12). In this work, to evaluate the prediction performance of EDACCA, XMEAS(38) is used as the quality variable $\mathbf{Y}$, and the process variables $\mathbf{X}$ are composed of XMV(1-11) and XMEAS(1-22). Since $\mathbf{X}$ and $\mathbf{Y}$ have different sampling rates where the quality variable is sampled every 15 minutes and process variables are sampled every 3 minutes, the quality values are replicated four times in the original data [54], which effectively affect the actual dynamic relations between X and Y. To mitigate this effect, the duplicated quality samples are removed along with their corresponding process samples [221]. Thus, 100 samples are used as training data, and 192 samples are used as testing data.

EDACCA, DACCA, DAPLS, DiPLS, and sDiCCA are compared in this case study, and the model parameters are obtained by performing cross-validation: $l = 14$, $s = 12$, $d = 2$, $k_1 = 0.02$ and $k_2 = 0.98$ for EDACCA; $l = 3$, $s = 1$, and $d = 1$ for DACCA; $l = 3$, $s = 3$, and $d = 1$ for DAPLS; $l = 3$ and $s = 5$ for DiPLS; and $l = 2$ and $s = 3$ for sDiCCA. Fig. 5.1 visualizes their prediction results, their mean squared errors (MSEs), and coefficient of determination (R^2) are summarized in Table 5.1. Compared with dynamic inner LVMs such as DiPLS and sDiCCA, EDACCA, DACCA, and DAPLS have obtained better prediction performance. Among them, EDACCA can capture the dynamics more accurately, and it achieves the best predictive modeling performance with the smallest MSE and the largest R^2.

The predictive contributions from the dynamic cross-correlations $\hat{\mathbf{Y}}_{xg}$ and the dynamic auto-correlations $\hat{\mathbf{Y}}_{yg}$ in EDACCA, DACCA, and DAPLS are further studied. Their MSEs are summarized in Table 5.2, which are visualized in Fig. 5.2. The MSEs of supervised sDiCCA and unsupervised DiCCA are also included in Table 5.2 for completeness. For DAPLS, the predictive modeling performance is not satisfactory if only $\hat{\mathbf{Y}}_{xg}$ or $\hat{\mathbf{Y}}_{yg}$ is included, while their combination $\mathbf{Y}$ provides good prediction performance. The MSEs of $\hat{\mathbf{Y}}_{xg}$ and $\hat{\mathbf{Y}}_{yg}$ in DACCA are smaller than DAPLS, but their combined performance is worse. With more flexibility introduced into EDACCA, the model puts more weight on the auto-regressive relations among quality samples, and it achieves the best performance among these models. It is noted that unsupervised DiCCA purely learns the auto-regressive

Table 5.3: Prediction Results of Different Models for Wind Process

	EDACCA	DACCA	DAPLS	DiPLS	sDiCCA
MSEs	**0.0194**	0.0565	0.0219	0.4670	0.4570
R^2	**0.9905**	0.9591	0.982	0.6545	0.6555

relations in quality samples, but its performance is not as good as EDACCA without the filtering effect brought by the process variables. Similarly, only learning the dynamic cross-correlations in sDiCCA does not generate satisfactory outcomes, since quality information is not fully exploited. Thus, the good performance of EDACCA is attributed to its flexibility because of the addition of the weights in its objective function.

Furthermore, to show that EDACCA achieves improvement in computing efficiency over DACCA, 100 rounds of the TEP dataset are selected with 10,000 samples in total, and they are partitioned into 10 groups with an increment of 1,000 samples (more details can be found in the work [138]). EDACCA and DACCA are trained on these 10 groups of datasets respectively, and their computing time is plotted in Fig. 5.3 in a semi-log form for better visualization. The model training is performed in MATLAB on the laptop computer with CPU (Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz) and GPU (NVIDIA GeForce RTX 2070 with Max-Q Design). As shown in the figure, EDACCA is much more efficient than DACCA, because of its handling of the matrix inversion step, and ill-conditioning issues. EDACCA even achieves comparable computing time with DAPLS, which does not involve matrix inversions in its training stage.

5.5.2 Wind Process

In this case study, the wind power data, which are collected from Ontario Independent Electricity System Operator (IESO)[2] at the year 2018, are used to evaluate the performance of EDACCA in terms of prediction performance and missing data imputation. More details about the wind process can be found in [8].

In our work, “total wind power" is selected as the quality variable, and five variables, “temperature", “real humidity", “wind direction", “wind speed" and “pressure", are selected as process variables. To retain the dynamics in the data, 7008 samples collected in the first 292 days are selected as training data, and the remaining 1752 samples are used as testing data.

Predictive Modeling

EDACCA, DACCA, DAPLS, DiPLS and sDiCCA are studied and compared, and their parameters are selected as $l = 3$, $s = 3$, $d = 1$, $k_1 = 0.4$ and $k_2 = 0.6$ for EDACCA; $l = 2$, $s = 3$, and $d = 3$ for DACCA; $l = 3$, $s = 3$, and $d = 1$ for DAPLS; $l = 3$ and $s = 6$ for DiPLS; and $l = 2$ and $s = 7$ for sDiCCA. The models' predictive modeling performance is

[2]https://www.ieso.ca/en/Power-Data/Data-Directory

Table 5.4: MSEs of Multi-Step Ahead Prediction Models

Steps	EDACCA	DiPLS	DiCCA	RFR	LSTM
Original	**0.0194**	0.467	0.457	0.5752	0.5025
1	**0.0112**	0.4951	0.4936	0.5821	0.5252
2	**0.0469**	0.5326	0.5312	0.5873	0.5693
3	**0.1026**	0.5753	0.5747	0.6116	0.6109
4	**0.1849**	0.6266	0.6231	0.6404	0.6683
5	**0.2684**	0.6818	0.6778	0.6767	0.7385

Table 5.5: MSEs of Different Imputation Methods for MCAR

Missing Ratio	EDACCA	GAIN	LR	Dup	DeDup
1%	**0.4607**	0.4804	0.4693	0.4708	0.4702
5%	**0.4618**	0.4970	0.4803	0.4705	0.4736
10%	**0.4611**	0.5247	0.5040	0.4692	0.4686
20%	**0.4614**	0.5253	0.5494	0.4693	0.4844
30%	**0.4622**	0.5771	0.5861	0.4707	0.4864
40%	**0.4557**	0.6769	0.6496	0.4899	0.4997

shown in Fig. 5.4, and their MSEs and coefficient of determination (R^2) are summarized in Table 5.4. It is observed that the prediction performance of EDACCA, DACCA and DAPLS is better than DiPLS and sDiCCA, with EDACCA performing the best, which is consistent with the TEP case study.

Multi-Step-Ahead Prediction

In this part, to further illustrate the prediction effectiveness of EDACCA, the EDACCA-based MS prediction performance is compared with DiPLS, DiCCA [52], random forest regression (RFR) [86] and long short-term memory network (LSTM) [78], with different time steps where $s = 1, 2, \ldots, 5$. Their parameters are selected with a cross-validation method to achieve their optimal performance, and the MS prediction results of these models are summarized in Table 5.4. In general, the uncertainties increase with increasing step h, and as shown in Table 5.4, the MSEs of all methods become larger as h increases. However, for all steps shown in Table 5.4, the predictive performance of EDACCA is better than the other models, indicating that the EDACCA-based multi-steps-ahead model has strong ahead predictive capability.

Missing Data Imputation

Two types of missing data are introduced into the wind power data, which are MCAR and MNAR. MCAR is to mimic the general scenario for missing data, and MNAR is to simulate the case when different frequencies are set for process and quality variables.

Table 5.6: Parameter Selection for MNAR

Step	EDACCA	LR	Dup	DeDup
2	l=3;s=6	l=5;s=10	l=3;s=6	l=3;s=3
3	l=3;s=6	l=5;s=10	l=3;s=6	l=3;s=2
4	l=3;s=5	l=5;s=10	l=3;s=8	l=3;s=2
6	l=3;s=5	l=5;s=10	l=3;s=8	l=5;s=4
12	l=3;s=3	l=3;s=5	l=3;s=5	l=5;s=2

Table 5.7: MSEs of Different Imputation Methods for MNAR

Step	EDACCA	LR	Dup	DeDup
2	**0.4606**	0.4840	0.4656	0.4733
3	**0.4578**	0.4739	0.4624	0.4836
4	**0.4709**	0.4747	0.4775	0.4942
6	**0.4736**	0.4884	0.4906	0.5857
12	**0.5570**	0.6623	0.6519	0.6624

For MCAR, 1%, 5%, 10%, 20%, 30%, and 40% samples are removed from training data randomly, and five imputation methods are compared, including EDACCA-based imputation, generative adversarial imputation nets (GAIN) [189], linear regression (LR) imputation, duplication (Dup) and deduplication (DeDup) method. Dup method imputes the missing data by replicating the previous nearest known values, and the DeDup method simply removes the missing data. GAIN is selected as an example of an adversarial neural network for comparison. Five different imputed data are generated, and DiPLS is performed on these datasets for imputation performance comparison. For each dataset, the model parameters of DiPLS are $l = 3$ and $s = 6$, which are selected with a joint cross-validation method. The MSEs for these imputed datasets are summarized in Table 5.5. For all cases, the EDACCA-based imputation strategy achieves better performance with the smallest MSE. For GAIN, the generator of GAIN generates imputed data without considering the temporal relations in adjacent samples. Though GAIN can generate data that can not be identified by its discriminator, the time series relations will be corrupted, and its performance becomes worse when the missing data ratio increases. LR imputation inserts linear relations into original data randomly which will wrongly predict the original dynamic relations between quality and process variables. Moreover, for EDACCA-based imputation, even if the missing rate increases from 1% to 40%, the imputation performance is consistent.

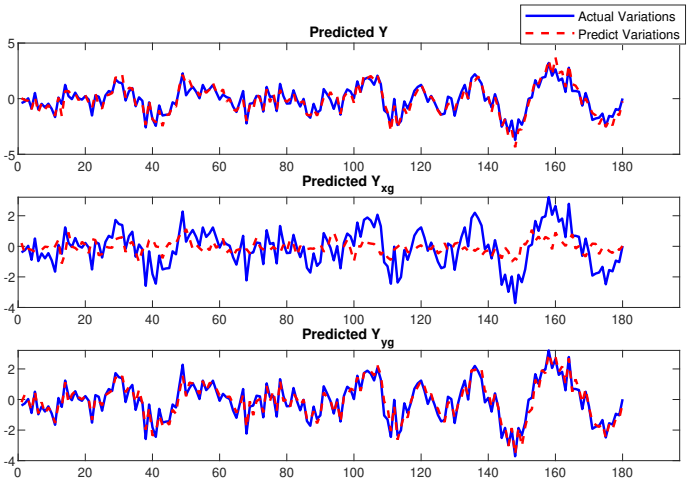
To mimic the MNAR missing data caused by different sampling rates of quality and process variables, the collected quality data are kept in every $\mathcal{S}$ step, where $\mathcal{S}$ is the number of sampling steps. For the wind dataset, the step should be selected as a divisor of 24, and thus 2, 3, 4, 6, and 12 are selected. Then EDACCA, LR, Dup, and DeDup-based imputation methods are conducted on the dataset with MNAR. It is noted that as analyzed in Section 5.4.4 and in the MCAR case study, GAIN achieves inferior performance when the

missing ratio is large, and thus it is not compared here. After performing imputation, four groups of imputed training data are used to train DiPLS respectively. The corresponding parameters selection results are listed in Table 5.6, and their MSEs are summarized in Table 5.7. As shown in the tables, EDACCA can capture the dynamics between process and quality variables, and the corresponding DiPLS model needs fewer latent variables for more accurate predictive modeling, compared with the DiPLS model trained on other imputed datasets. As the step becomes bigger and the ratio of missing data gets larger, the advantage of the EDACCA-based imputation method is more obvious with smaller MSEs. It is concluded that EDACCA imputation will provide more accurate information to compensate for the missing data when the proportion of missing data is large.

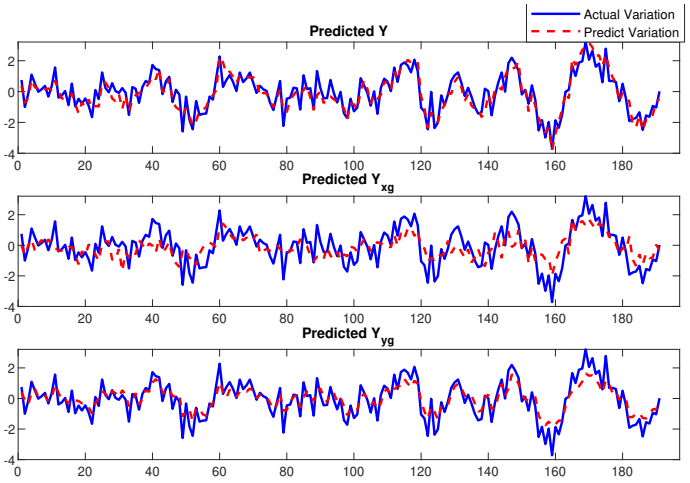
To further illustrate the imputation performance of EDACCA, t-distributed stochastic neighbor embedding (t-SNE) is applied for feature visualization. t-SNE is performed on the augmented matrix, which is constructed by the imputed $\mathbf{Y}$ and actual $\mathbf{Y}$. Fig. 5.5a and Fig. 5.5b visualize the EDACCA imputation and linear regression imputation respectively for $\mathcal{S} = 12$. As shown in Fig. 5.5a, t-SNE can not distinguish $\mathbf{Y}$ imputed by EDACCA and the actual $\mathbf{Y}$, implying that the distribution gap between the imputed data and the actual data is negligible. For linear regression imputation in Fig. 5.5b, however, a large group of imputed $\mathbf{Y}$ can be identified with t-SNE, as shown by the data within a black square in Fig. 5.5b.

5.6 Summary

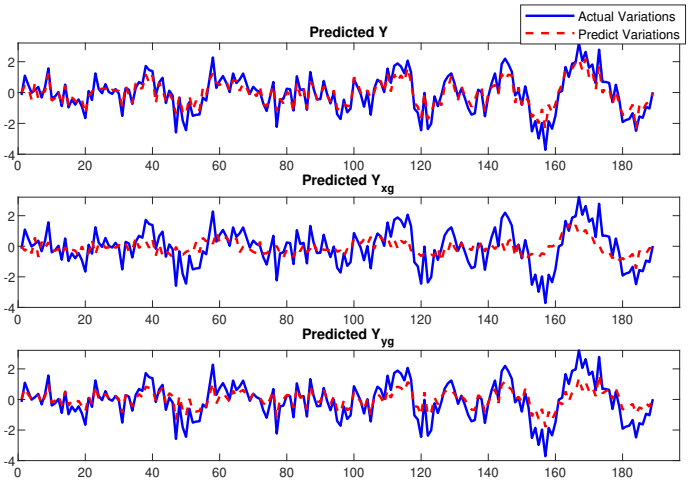
In this chapter, a dynamic regressive CCA method is proposed for dynamic modeling, and its efficiency and stability are improved with full-rank decomposition and SVD techniques. Compared with traditional dynamic extensions of latent variable models, EDACCA can simultaneously extract the cross-correlations between current quality variables and current and past process variables, and the auto-correlations between current and past quality variables. A multi-step ahead prediction method is proposed based on EDACCA, which is adapted further for missing data imputation when irregular sampling frequencies exist in the process. The TEP and wind processes are used to compare EDACCA with some existing methods in terms of predictive modeling and missing data imputation.



(a) EDACCA



(b) DACCA



(c) DAPLS

Figure 5.2: Dynamic Cross-Correlations ($\mathbf{Y}_{xg}$) and Auto-Correlations ($\mathbf{Y}_{yg}$) of EDACCA, DACCA and DAPLS

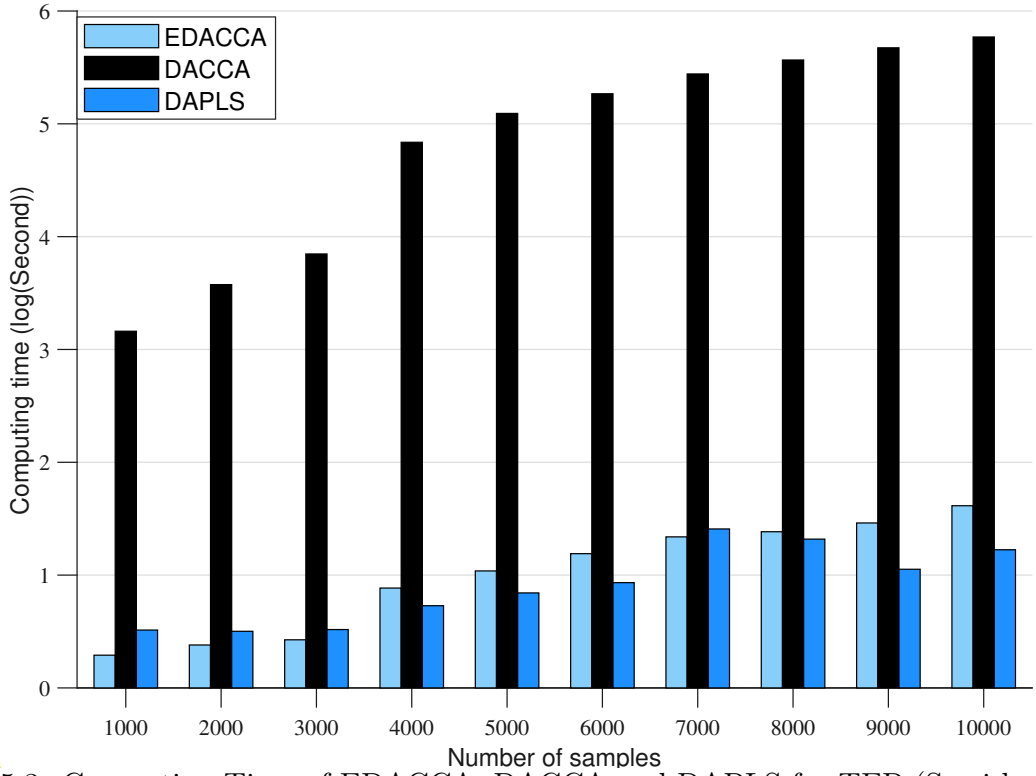


Figure 5.3: Computing Time of EDACCA, DACCA and DAPLS for TEP (Semi-log Unit)

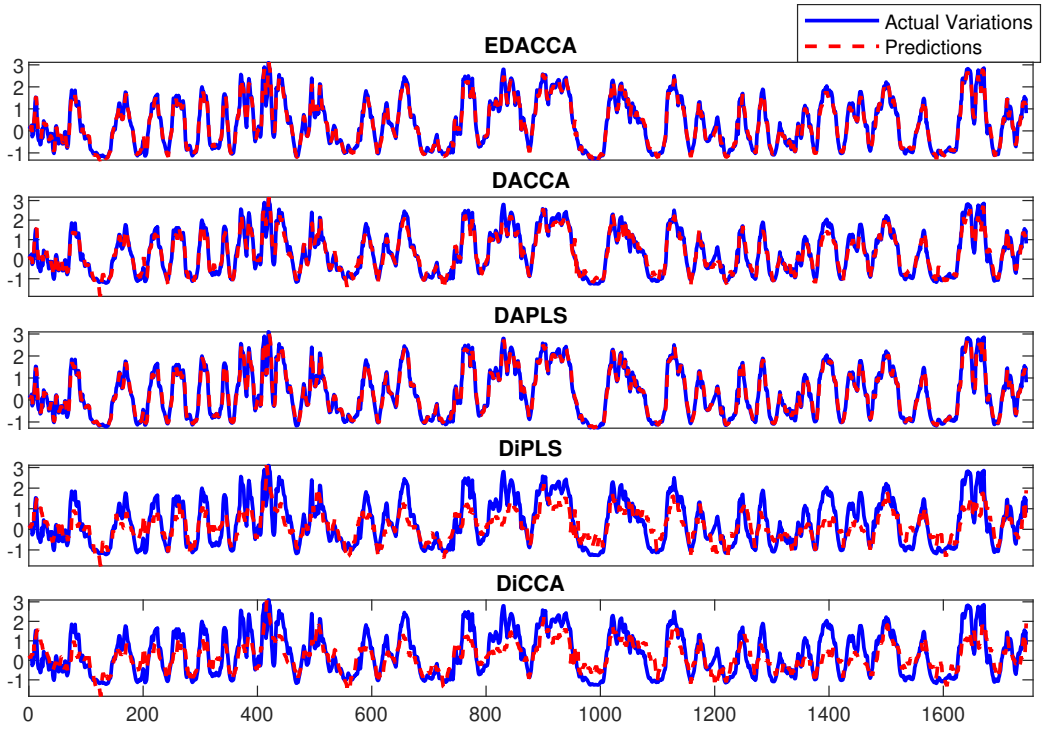
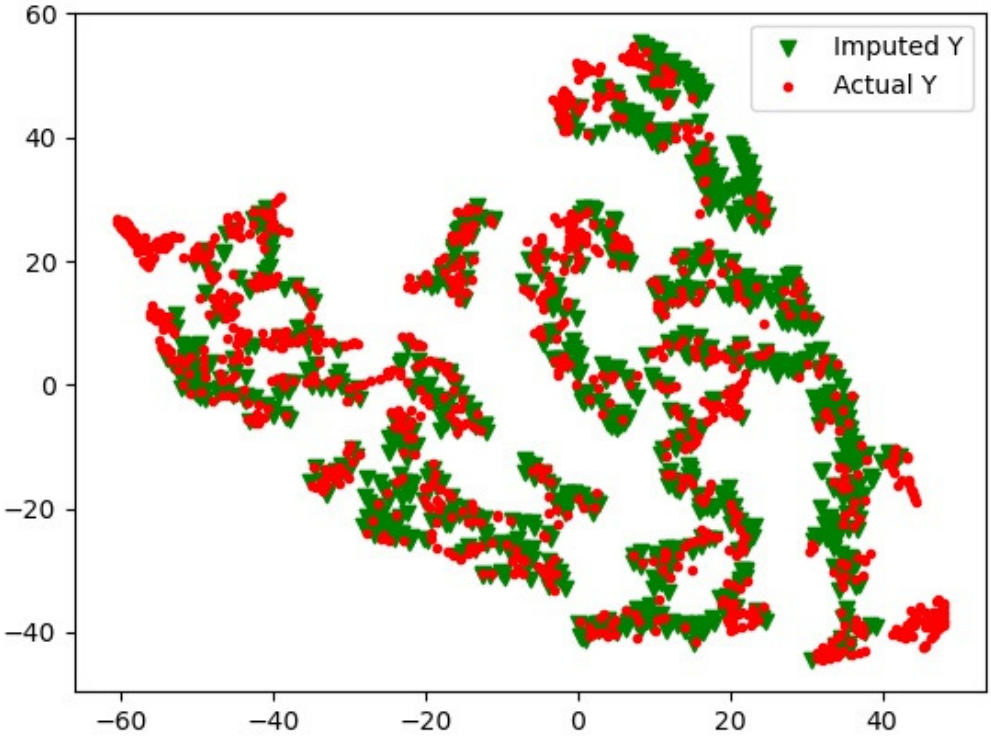
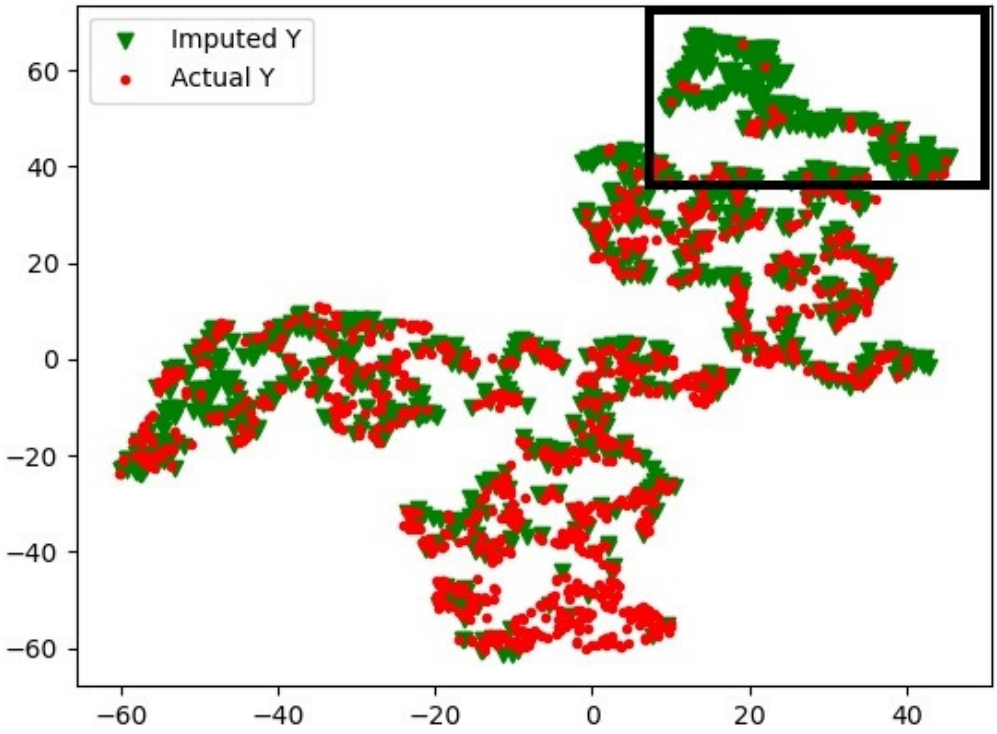


Figure 5.4: Dynamic Modeling Performance of Different Models for Wind Process



(a) EDACCA Imputation



(b) Linear Regression Imputation

Figure 5.5: t-SNE Visualization for EDACCA and LR Imputation Methods

Chapter 6

Multi-Step Dynamic Slow Feature Analysis for Anomaly Detection

6.1 Overview

[1] Current dynamic extensions of latent variable models have good modeling performance. However, most of them focus on predictability instead of slowness which is the invariance between neighboring samples. To consider the slowness in the processes and improve the interpretability, a novel exponentially weighted SFA (EWSFA) is proposed to extract slow features by following the decaying nature of temporal signals. Exponentially decaying weights are learned in EWSFA to model dynamic relations between current and historical samples, and its computation efficiency is further improved with the aid of singular value decomposition. An EWSFA-based monitoring scheme is developed to detect anomalies in both slow and fast variations, and a novel dynamic index is designed to identify the occurrence of operating condition changes. The effectiveness of EWSFA is demonstrated through two industrial case studies.

6.2 Introduction

Accurate modeling is important for developing reliable fault detection and diagnosis methods. With the advancement of smart sensors and data storage techniques, voluminous samples are collected, and data-driven modeling techniques are preferred due to their simplicity and superiority [61, 92, 125, 111]. Among them, dimensionality reduction algorithms are widely studied and used to extract a subset of valuable information from the highly-correlated data based on various criteria, including slow feature analysis (SFA) [174], principal component analysis (PCA) [1], partial least squares (PLS) [63], and latent variable regression (LVR) [218].

[1] Adapted from a paper submitted to AIChE Journal, which is currently under review.

Most of these latent variable methods (LVMs) were originally developed with the assumption that the processes are static, and then, additional dynamic handling techniques were proposed for system dynamics extraction. A straightforward method to incorporate dynamics is to perform LVMs on data augmented with historical samples [30, 58, 70, 100]. However, following data augmentation, the dimension of the weighted vectors and loading vectors will increase dramatically as the time lag increases. For data augmentation methods, it is difficult to obtain an explicit relation between latent variables and historical data. Thus, the extracted relations are hard to interpret. To address this issue, frequency domain methods were also proposed by using Fourier transform to convert the data from the time to the frequency domain [24, 116]. These methods will first capture and construct the dynamic relations between historical data and future data directly, without incorporating current data during the training phase. Subsequently, during the testing stage, they can directly predict future data by using historical data. They require both historical and future time series data for dynamic modeling. Pena et al. [115] avoided the usage of future data with the aid of causal relations. However, dynamics are still confused with cross-correlated variations in these methods. Alternatively, Li et al. [96] explicitly learned the dynamic relations between the current sample and a weighted combination of historical samples. In this method, an outer model is designed to extract the dynamic latent variables by maximizing the auto-covariance whereas an inner model is built to represent the dynamic relationships between extracted latent variables. However, the model retains inconsistent temporal information in the inner and outer models since the outer model only focuses on one-time lag whereas multi-time lags are considered in the inner model. Dong and Qin [50] modified the inner model structure and proposed a dynamic inner PCA to achieve dynamic consistency. The same idea is extended to other LVMs, such as dynamic inner CCA (DiCCA) [48] and dynamic inner LVR [224]. The corresponding monitoring strategies based on the above methods have been also applied for dynamic process monitoring [37, 209, 77]. A more comprehensive review of dynamic LVMs and their monitoring methods can be found in [127, 187].

In contrast, SFA is a time series modeling approach that focuses on the invariance between neighboring samples by minimizing their difference. SFA is widely used for object detection, action recognition, and blind source separation [174, 20, 204, 181, 157]. The idea is based on a premise that slow features generally correspond to very significant anomalies in a process and then it is important to identify them from data that contains important slow features with less relevant fast changing features. Considering its ability to extract slow variations from noises, SFA is adapted in manufacturing processes to differentiate normal deviations in operating conditions from dynamic anomalies [147]. Since it is common to change process objectives based on the market demands in modern manufacturing processes, identifying actual anomalies is essential to reduce nuisance alarms and thus improve the system's reliability. SFA extracts the first-order dynamics and it applies an augmentation technique to extract multi-order dynamics [147, 206, 207, 143]. However, similar to other LVMs, augmented data exploitation leads to unclear dynamic interpretation in the extracted latent variables (LVs). Inspired by the dynamic inner LVMs, dynamic inner SFA (DiSFA) was also designed [153, 184] to build a serial correlation between the current sample and a weighted combination of past samples. DiSFA assigns optimal unconstrained

weights to historical samples to maximize their similarity with the current sample. Thus, instead of emphasizing the *slowness* as in SFA, the nature of DiSFA is to maximize the *predictability* of LVs. Further, nonlinearity is not considered or effectively integrated in the aforementioned variants of SFA.

Following the above, it can be asserted that the current advanced dynamic extensions of SFA still lack a dependable scheme for postulating dynamic relations amongst the slow features and do not address nonlinear behavior. To address these issues, a multi-step dynamic SFA, namely exponentially weighted SFA (EWSFA), is proposed in this work to explicitly extract multi-order dynamics from data. The collected data are first transformed with an autoencoder (AE) or support vector machine (SVM) for nonlinearity extraction, and the reduced-dimensional latent variables generated by the encoder are further processed for dynamic modeling. EWSFA minimizes the difference between the LV of the current sample and the predicted LV formed by the weighted combination of past samples. Different from dynamic inner LVMs, an exponentially weighted moving average [71] is applied to the weights of past samples, which is in line with the decaying nature of time series. The efficiency of EWSFA is further improved with the aid of full-rank decomposition. An EWSFA-based dynamic monitoring framework is also designed, where three monitoring indices are developed to monitor anomalies in slow features, fast features, and operating dynamics respectively. The main contributions of this chapter are summarized as follows.

(1) Proposed an exponentially weighted slow feature analysis (EWSFA) algorithm to capture multi-step dynamics. The extracted dynamics are easy to interpret and follow the decaying nature of time series data.

(2) Improved the computation efficiency of EWSFA to maintain its superior performance when applied to high-dimensional temporal data.

(3) Designed a comprehensive EWSFA-based dynamic monitoring framework. In addition to the monitoring of static and dynamic anomalies, a novel dynamic index is also designed to differentiate the effect of intentional operating condition changes from normal operation.

The remainder of this chapter is organized. A brief review of SFA and DiSFA methods is presented in Section 6.3. Section 6.4 presents the details of the proposed efficient EWSFA modeling and monitoring algorithm. In Section 6.5, two industrial case studies are presented to illustrate the superiority of the proposed methods for the detection of oscillatory responses and monitoring performance. Finally, Section 6.6 concludes the chapter.

6.3 Preliminaries

6.3.1 Slow Feature Analysis

SFA was proposed by Wiskott and Sejnowski [174] to extract slowly varying features from high dimensional temporal input signals. Given an m-dimensional input signal

$\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_m(t)]^\top$, SFA seeks for l sets of transformation functions $\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_l(\mathbf{x})]^\top$, and transforms $\mathbf{x}(t)$ into l LVs $\mathbf{s}(t) = [s_1(t), s_2(t), \ldots, s_l(t)]^\top$ by optimizing

$$\min_{g_j(\mathbf{x})} \ \langle \dot{s}_j^2 \rangle_t \tag{6.1}$$

$$\text{s.t. } \langle s_j \rangle_t = 0, \ \ \langle s_j^2 \rangle_t = 1, \ \ \langle s_i s_j \rangle_t = 0 \ \ \forall i \neq j \tag{6.2}$$

where the angle bracket denotes temporal averaging $\langle f \rangle_t \equiv \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} f(t) dt$, and $\dot{s}_j$ is the first-order derivative of $s_j(t)$. SFA defines $\langle \dot{s}_j \rangle_t$ in Eq. (6.1) as an indicator for *slowness* of LVs, and LVs are extracted with an ascending order of $\langle \dot{s}_j \rangle_t$. The constraints in Eq. (6.2) ensure that the extracted LVs have zero mean and unit variance, and are decorrelated from previous LVs. It is noted that LVs s_j $(j = 1, 2, \ldots, l)$ are also called slow featuress (SFs) in SFA, and both LVs and SFs are used interchangeably in this work.

When $\mathbf{g}(\mathbf{x})$ is chosen to be a linear combination of $\mathbf{x}$, the relation between s_j and $\mathbf{x}$ is expressed as

$$s_j = g_j(\mathbf{x}) = \mathbf{w}_j^\top \mathbf{x} \tag{6.3}$$

where $\mathbf{w}_j = [w_{j1}, w_{j2}, \ldots, w_{jm}]^\top$ is the weighting vector of the j^{th} LV. In a more compact form, we have

$$\mathbf{s} = \mathbf{W}^\top \mathbf{x}$$

where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_l]$. Thus, the optimization objective defined in Eqs. (6.1) and (6.2) is to find the transformation matrix $\mathbf{W}$ such that their corresponding latent variables retain the slowest variations. The values of $\mathbf{W}$ can be obtained with the aid of singular value decomposition and eigendecomposition [174].

Blaschke et al. [19] proved that for discrete time series, SFA is equivalent to maximizing the *correlation* between LVs of neighboring samples $\mathbf{x}(t)$ and $\mathbf{x}(t-1)$, which is

$$\max_{g_j(\mathbf{x})} \ s_j(t) s_j(t-1) \tag{6.4}$$

under the same constraints as in Eq. (6.2). As observed from Eq. (6.4), SFA only considers the one-step auto-correlations and assumes that the *slowness* is captured as long as the neighboring samples are highly correlated. However, since past samples may still have temporal effects on the current sample $\mathbf{x}(t)$, the dynamic relations extracted by the traditional SFA method described above may not be accurate.

6.3.2 Dynamic Inner Slow Feature Analysis

To improve dynamic modeling performance, multi-step SFA methods were proposed, such as multi-step dynamic SFA [103] and dynamic inner SFA [153, 184]. Among them, DiSFA has achieved improved performance since it retains consistent dynamic relations. Inspired

by the work of DiCCA [48], Xu et al. [184] proposed DiSFA to capture dynamics from discrete time series signals $\mathbf{x}(k)$ by optimizing the following objective.

$$\min_{\mathbf{w}_j} \quad \left\langle (s_j - \hat{s}_j)^2 \right\rangle_t \tag{6.5}$$

$$\text{s.t. } \langle s_j \rangle_t = 0, \quad \langle s_j^2 \rangle_t = 1, \quad \langle \hat{s}_j^2 \rangle_t = 1, \quad \langle s_i s_j \rangle_t = 0 \quad \forall i \neq j \tag{6.6}$$

where $\hat{s}_j(t)$ is predicted from LVs of the past d samples,

$$\hat{s}_j(t) = \sum_{i=1}^{d} \beta_i s_j(t-i) \tag{6.7}$$

where $s_j(t-i)$ is the j^{th} LV of sample $\mathbf{x}(t-i)$, β_i are their corresponding weights, and d is the time lag. The detailed solution of DiSFA can be referred to [153, 184], and the optimization goals of DiSFA and DiCCA are exactly the same. The additional constraint $\langle \hat{s}_j^2 \rangle_t = 1$ in Eq. (6.6) imposes a limit for the weight parameters $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_d]^\top$, and optimal unconstrained values can be assigned to β_i, which implies that the main objective of DiSFA in Eq. (6.5) is to maximize the *predictability* of the LVs.

After performing DiSFA, a new sample $\mathbf{x}(k)$ can be decomposed as follows

$$\mathbf{x}(t) = \mathbf{P}\mathbf{G}(q^{-1})\mathbf{s}(t) + \mathbf{e}(t) \tag{6.8}$$

where $\mathbf{P} \in \mathbb{R}^{m \times l} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_l]$; $\mathbf{G}(q^{-1})\mathbf{s}(t) = \sum_{i=1}^{d} \boldsymbol{\beta}_i \mathbf{s}(t-i)$ is the transfer function of the dynamic model, and $\mathbf{e}(t)$ represents the residuals.

The extracted latent variables can be further separated into slow and fast features as $\mathbf{s}(t) = [\mathbf{s}_s(t), \mathbf{s}_f(t)]$, and Shang et al. [147] designed four monitoring indices to detect anomalies in $\mathbf{s}_s(t)$ and $\mathbf{s}_f(t)$. First, T^2 and T_e^2 statistics are designed to detect the static variations in the slow and fast features respectively. Then, two dynamic indices, S^2 and S_e^2, are designed based on the temporal slowness of extracted latent variables for dynamic anomaly detection. More details can be found in [147].

Remark 8 *SFA is a special case of DiSFA with the time lag $d = 1$ [127]. It is also noted that when the linear transformation function Eq. (6.3) is adopted for DiSFA, DiSFA is the same as DiCCA, which maximizes the auto-correlations of LVs and the predicted LVs from past samples.*

6.4 Methodology

The details of the proposed EWSFA algorithm are presented in this section. First, the necessity of the proposed work is established, followed by the formulation of the proposed algorithm. A comprehensive dynamic monitoring method is also developed based on EWSFA.

6.4.1 Problem Statement and Motivation

SFA constructs a new representation of the original data by learning invariant or slowly varying features, and it has been widely used in the area of computer vision for pattern and action recognition [204, 181, 157]. Shang et al. [147] introduced SFA into the process systems engineering field to distinguish actual dynamics anomalies from normal changes in operating conditions, and developed a concurrent SFA-based monitoring strategy. As indicated in Eq. (6.4), SFA is classified as a first-order dynamic analytics method [19, 127], since it mainly focuses on the relation between $\mathbf{x}(t)$ and $\mathbf{x}(t-1)$, while the effect of other past samples $\mathbf{x}(t-i)$ $(i>1)$ are implicitly learned.

Many research efforts have been made to improve temporal modeling performance and explicitly learn the multi-step dynamics. Similar to other latent variable methods, dynamic SFA was initially developed by augmenting the input signals with past samples [147], which makes the dynamics hard to interpret. Repeated signals might also be extracted for augmentation-based methods. For instance, for the samples containing ramp ($\mathbf{x}_1$) and sinusoidal ($\mathbf{x}_2$) variations shown in Figure 6.1, whose actual dynamics order is 2, the augmented SFA unnecessarily extracts the same frequency ($1/e^{-2.2976} \approx 10$) in the 2$^{\text{nd}}$ and 3$^{\text{rd}}$ LVs, as shown in the fast Fourier transform (FFT) in Figure 6.2.

Auto-regressive models have been combined with SFA to extract higher-order dynamics by exploiting the weighted combination of past samples with Eq. (6.7) in the works of [153, 184, 103]. These models, which are trained to maximize the *predictability* of the extracted latent variables, they have achieved some degrees of success in simulated or actual industrial processes. However, the *slowness* properties that the model seeks to infer from adjacent samples are not always obtained, since weights β_i $(i=1,2,\ldots,d)$ will be assigned to past samples only under the constraint $\langle \hat{s}_j^2 \rangle_t = 1$. To ensure that the *slowness* is captured correctly and to guarantee the accurate distinction between normal operating changes and dynamics anomalies, a suitable constraint should be imposed β_i $(i=1,2,\ldots,d)$.

Exponentially weighted moving average (EWMA) filters are a typical quantitative or statistical tool to model a time series [71]. It has been widely used in finance [172, 102]. The main idea of EWMA is that lower weights are assigned for older observations and the weights decay exponentially as the observation time is older. The mathematical formulation of EWMA is

$$\hat{y}_t = \alpha y_t + (1-\alpha)\hat{y}_{t-1} \tag{6.9}$$

where y_t is the current observation, and α is the weight set by the user. Eq. (6.9) is designed as a recursive function to predict current observation by using previous observations, and it can be rearranged as

$$\hat{y}_{t+1} = \alpha[y_t + (1-\alpha)y_{t-1} + (1-\alpha)^2 y_{t-2} + \cdots] \tag{6.10}$$

where the weight of the observation y_{t-i} is $\alpha(1-\alpha)^i$. Since α is set between 0 and 1, the weights fall exponentially as the observation gets older. However, if a small α is selected, the weight of one-time lag will be smaller than two- or more time lag which contradicts the

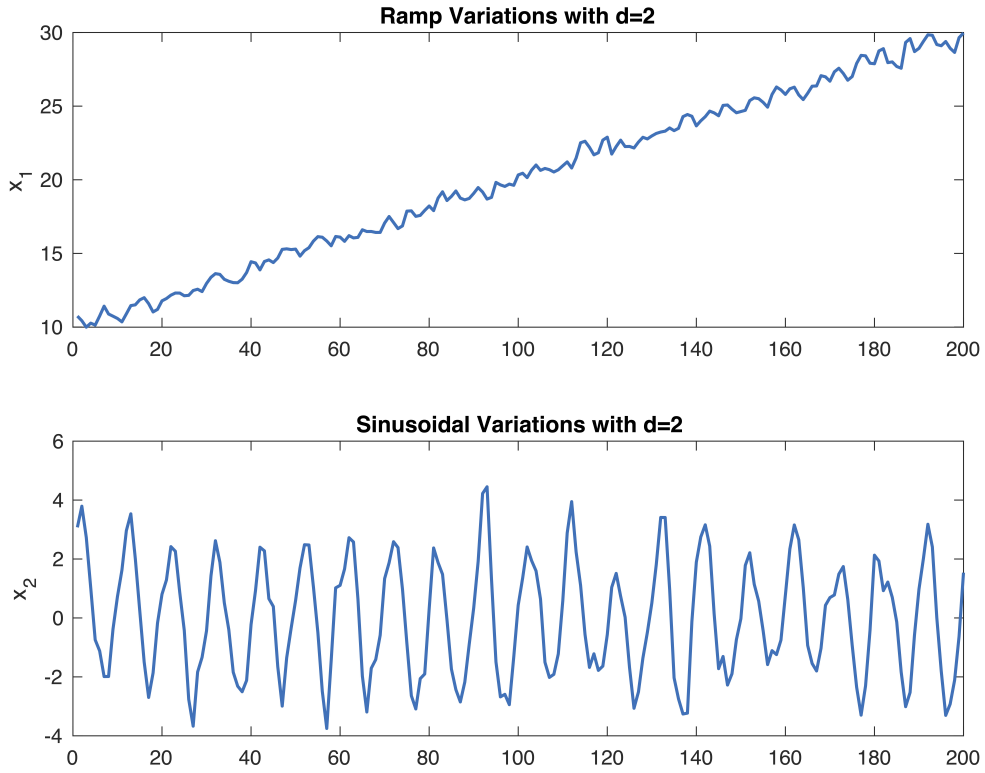


Figure 6.1: Input Signals with Dynamics Order $d = 2$

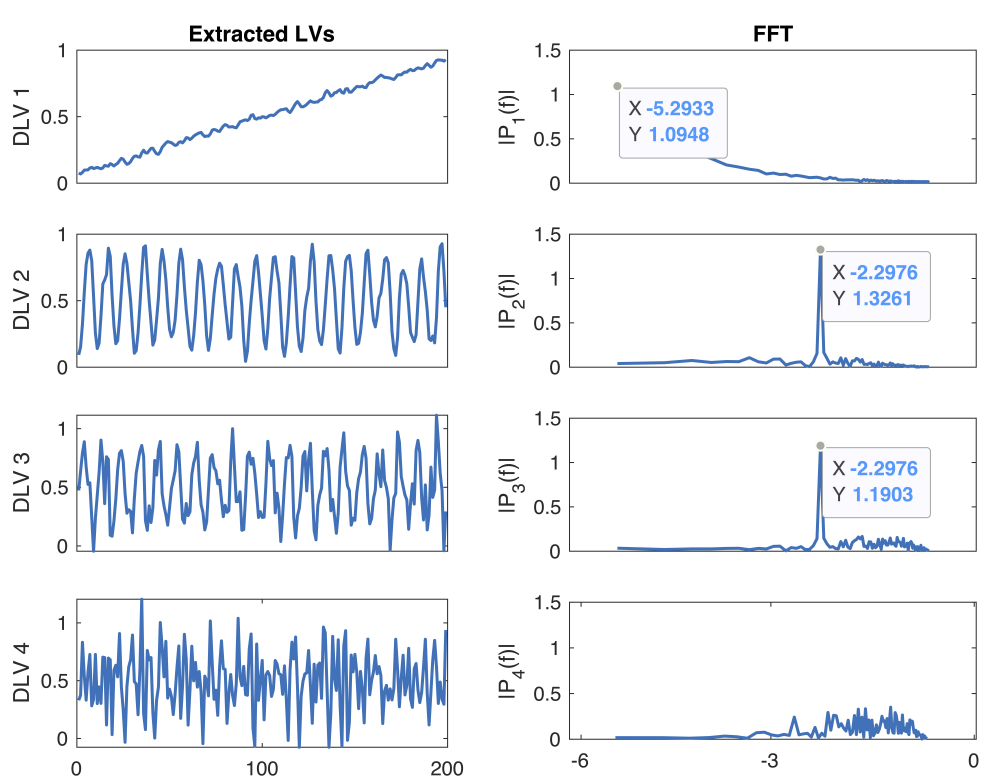


Figure 6.2: SFA LVs and Their FFT

main idea that lower weights are assigned for older observations exactly. In the following section, to ensure the consistent decay of the weights, the main concept of EWMA will be adjusted and adopted in our proposed algorithm.

Another challenge involved with the application of SFA, multi-step SFA, and DiSFA is related to the indices that must be used in these techniques for efficient fault detection. All these methods require four monitoring indices, namely T_d^2, T_e^2, S_d^2 and S_e^2 [147, 153, 184, 103] where T_d^2 and T_e^2 are used to detect deviations from pre-designed operating conditions, while S_d^2 and S_e^2 are for dynamics anomaly detection. However, the differences between these indices in these two groups remain vague, and thus the inferences from these indices are not obvious.

6.4.2 Optimization Problem of EWSFA

In this work, an exponentially weighted SFA (EWSFA) algorithm is proposed to address the aforementioned issues. Denote the whole training data as $\mathbf{X} \in \mathcal{R}^{n\times m} = [\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n)]^\top$, with n samples and each one containing m variables. To ensure that slowly varying signals are kept in the extracted LVs, in addition to the invariance between neighboring samples $s(t)$ and $s(t-1)$, it is also important to guarantee maximum similarity among $s(t)$ and other past series data $\{s(t-1), s(t-2), \ldots, s(t-d)\}$, where d is the dynamics order of the collected samples. The subscript j for j^{th} LV is omitted for ease of representation. Effective integration of the *slowness* features among past temporal samples is provided for. Instead of assigning random or equal weights to $\{s(t-1), s(t-2), \ldots, s(t-d)\}$ as in existing dynamic SFA methods, it is reasonable to assume that more recent samples contribute more to the *slowness* features, Based on this premise a modified version of an exponentially weighted moving average (EWMA) filter is adjusted applied in EWSFA as follows. Mathematically, to capture multi-step dynamics, for a single sample $\mathbf{x}(t)$, the relation between its current LV $s(t)$ and past LVs $\{s(t-1), s(t-2), \ldots, s(t-d)\}$ is expressed as

$$s(t) = \sum_{i=1}^{d} \alpha^i s(t-i) + e(t) \tag{6.11}$$

where α $(0 < \alpha \leq 1)$ is a smoothing constant that satisfies the constraint $\alpha > \alpha^2 > \cdots > \alpha^d$, and it determines the amount of information secured from past samples. $e(t)$ is the modeling error. To ensure a slowly varying trend, the difference between $s(t)$ and $\hat{s}(t) = \sum_{i=1}^{d} \alpha^i s(t-i)$ should be minimized for all samples in $\mathbf{X}$. Accordingly, the optimization goal of EWSFA is formulated as

$$\min_{\mathbf{w}_j} \quad \left\langle (s_j - \hat{s}_j)^2 \right\rangle_{t+1} \tag{6.12}$$

$$\text{s.t.} \quad \langle s_j \rangle_{t+1} = 0, \quad \langle s_j^2 \rangle_{t+1} = 1, \quad \langle \hat{s}_j^2 \rangle_{t+1} = 1, \quad \langle s_i s_j \rangle_{t+1} = 0 \ \ \forall i \neq j \tag{6.13}$$

where $\hat{s}_j(t+1)$ is predicted from LVs of the past d samples as $\hat{s}_j(t+1) = \sum_{i=1}^{d} \alpha^i \mathbf{s}_j(t+1-i)$.

Remark 9 *In conventional SFA, the emphasis is solely on the present time t, which is denoted by the subscript (t). In contrast, our proposed method concentrates on the dynamic interplay between s and $\hat{s}$ at time $t+1$, where $\hat{s}$ is predicted using current and past data.*

For discrete time series, which is the common form of dynamic data for most modern manufacturing processes, the objective of EWSFA in Eq. (6.12) is equivalent to

$$\max_{g(\mathbf{X})} \quad s_j^\top(t+1)\hat{s}_j(t+1) \tag{6.14}$$

$$\text{s.t. } s_j^\top(t+1)s_j(t+1) = 1, \quad \hat{s}_j^\top(t+1)\hat{s}_j(t+1) = 1 \tag{6.15}$$

Remark 10 *Both EWSFA and DiSFA learn multi-step dynamics from time series data. Their key difference is that EWSFA emphasizes both multi-step slowness features, and thus it attaches fewer weights on older samples as in Eq. (6.11). In contrast, DiSFA mainly focuses on predictability.*

6.4.3 The EWSFA Algorithm

The mapping functions $\mathbf{g}(\mathbf{X})$ can be designed based on the nature of collected data. For illustration purposes, a weighted combination form is considered in this work, where $g(\mathbf{x}) := \sum_{h=1}^{H} w_h f_h(\mathbf{x})$. The mapping functions $\mathbf{f} = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_H(\mathbf{x})]^\top$ can be linear or nonlinear functions. Some popular nonlinear algorithms including support vector machine (SVM) and autoencoder (AE) [111, 110] can be used to achieve the required mappings. Denote the resulting matrix as $\mathbf{Z} \in \mathcal{R}^{n\times H} = [\mathbf{z}(1), \mathbf{z}(2), \ldots, \mathbf{z}(n)]^\top$, where $\mathbf{z}(i) = \boldsymbol{f}(\mathbf{x}(i))$. The objective of EWSFA in Eq. (6.14) can then be written as

$$\begin{aligned} \max_{\mathbf{w},\boldsymbol{\beta}} \quad & \mathbf{w}^\top \mathbf{Z}_{d+1}^\top \mathbb{Z}_d[\boldsymbol{\beta} \otimes \mathbf{w}] \\ \text{s.t.} \quad & \|\mathbf{Z}_{d+1}\mathbf{w}\| = 1, \ \|\mathbb{Z}_d[\boldsymbol{\beta} \otimes \mathbf{w}]\| = 1 \end{aligned} \tag{6.16}$$

where

$$\begin{aligned} \mathbb{Z}_d \in \mathbb{R}^{N\times(dH)} &= [\mathbf{Z}_d, \mathbf{Z}_{d-1}, \ldots, \mathbf{Z}_1] \\ \boldsymbol{\beta} \in \mathbb{R}^d &= [\beta_1, \beta_2, \ldots, \beta_d]^\top \end{aligned} \tag{6.17}$$

$\mathbf{Z}_i \in \mathbb{R}^{N\times H} = [\mathbf{z}_i, \mathbf{z}_{i+1}, \ldots, \mathbf{z}_{i+N}]^\top$ $(i = 1, 2 \ldots, d+1)$, $N = n - d - 1$, and $\beta_i = \alpha^i$ $(i = 1, 2 \ldots, d)$. $\otimes$ is the Kronecker product operator.

To obtain the model parameters, the Lagrange multiplier λ_w is introduced, leading to

$$\begin{aligned} \mathcal{L}(\mathbf{w}, \lambda_w) =& \mathbf{w}^\top \mathbf{Z}_{d+1}^\top \mathbb{Z}_d[\boldsymbol{\beta} \otimes \mathbf{w}] \\ &+ \frac{\lambda_w}{2}(1 - \mathbf{w}^\top \mathbf{Z}_{d+1}^\top \mathbf{Z}_{d+1}\mathbf{w}) \end{aligned} \tag{6.18}$$

It is noted that the other constraint $\|\mathbb{Z}_d[\boldsymbol{\beta} \otimes \mathbf{w}]\| = 1$ imposes a bound on the weighting parameter $\boldsymbol{\beta}$, which is not included in Eq. (6.18) on purpose. The need for this constraint will be further elucidated later in the chapter.

Solving the Lagrangian function in Eq. (6.18) with differentiation leads to

$$\left(\mathbf{Z}_{d+1}^{\top}\mathbf{Z}_{d+1}\right)^{-1}\left(\mathbf{Z}_{d+1}^{\top}\mathbf{Z}_{\beta}+\mathbf{Z}_{\beta}^{\top}\mathbf{Z}_{d+1}\right)\mathbf{w}=\lambda_{w}\mathbf{w} \tag{6.19}$$

where $\mathbf{Z}_{\beta}=\mathbb{Z}_{d}[\boldsymbol{\beta}\otimes\mathbf{I}_{H}]=\sum_{i=1}^{d}\beta_{i}\mathbf{Z}_{d-i+1}$. Eq. (6.19) indicates that $\mathbf{w}$ is the eigenvector of the matrix $\left(\mathbf{Z}_{d+1}^{\top}\mathbf{Z}_{d+1}\right)^{-1}\left(\mathbf{Z}_{d+1}^{\top}\mathbf{Z}_{\beta}+\mathbf{Z}_{\beta}^{\top}\mathbf{Z}_{d+1}\right)$ with λ_w as the largest eigenvalue. Combining the preset relation in Eq. (6.11), the model parameters of EWSFA can be obtained by conducting the following relations.

(1) Initialize $\mathbf{w}$ with a random unit vector and $\boldsymbol{\beta}=[\alpha,\alpha^{2},\ldots,\alpha^{d}]^{\top}$, where $\sum_{i=1}^{d}\alpha^{i}=\frac{\alpha(1-\alpha^{d})}{1-\alpha}=1$;

(2) Construct $\mathbf{s}_{i}=[s(i),s(i+1),\ldots,s(i+N)]^{\top}=\mathbf{Z}_{i}\mathbf{w}$ for $i=1,2,\ldots,d+1$;

(3) Denote $\mathbf{S}_{d}=[\mathbf{s}_{d},\mathbf{s}_{d-1},\ldots,\mathbf{s}_{1}]$, and normalize $\mathbf{s}_{d+1}$ by $\mathbf{s}_{d+1}=\mathbf{s}_{d+1}/\|\mathbf{s}_{d+1}\|$;

(4) Obtain $\hat{\mathbf{s}}_{d+1}=\mathbf{S}_{d}\boldsymbol{\beta}$;

(5) Normalize $\boldsymbol{\beta}=\boldsymbol{\beta}/\|\hat{\mathbf{s}}_{d+1}\|$ and $\hat{\mathbf{s}}_{d+1}=\hat{\mathbf{s}}_{d+1}/\|\hat{\mathbf{s}}_{d+1}\|$;

(6) Update $\mathbf{w}=\left(\mathbf{Z}_{d+1}^{\top}\mathbf{Z}_{d+1}\right)^{-1}\left(\mathbf{Z}_{d+1}^{\top}\mathbf{Z}_{\beta}+\mathbf{Z}_{\beta}^{\top}\mathbf{Z}_{d+1}\right)$, where $\mathbf{Z}_{\beta}=\sum_{i=1}^{d}\beta_{i}\mathbf{Z}_{d-i+1}$;

(7) Repeat (2)- (6) until convergence is achieved, and calculate $J_{R^2}=\mathbf{s}_{d+1}^{\top}\hat{\mathbf{s}}_{d+1}$.

As shown in Step (6), the matrix inversion has to be conducted in each iteration before convergence, leading to a high computation burden. Further, with Steps (1) - (7), deflation is applied whereby the first set of model parameters is extracted, and their effect is subtracted from $\mathbf{Z}$ for subsequent LVs extraction. The deflation procedure, however, may make $\mathbf{Z}$ rank-deficient, which must be addressed.

6.4.4 Efficient EWSFA

To reduce the computation cost of EWSFA and to address the potential rank-deficient issue, an economic form of the singular value decomposition (SVD) is adopted to improve its efficiency [48, 118]. Different from conventional SVD, economic SVD removes zeros from the diagonal matrix composed of singular values and their corresponding singular vectors. With the economic SVD, $\mathbf{Z}_{d+1}$ is decomposed into

$$\mathbf{Z}_{d+1}=\mathbf{U}_{d+1}\mathbf{D}\mathbf{V}^{\top} \tag{6.21}$$

where $\mathbf{D}$ is a diagonal matrix containing nonzero singular values only, and $\mathbf{U}_{d+1}$ and $\mathbf{V}$ are its corresponding orthonormal singular matrices. Then the LVs $\mathbf{s}=\mathbf{Z}\mathbf{w}$ can be represented as

$$\mathbf{s}=\left(\mathbf{Z}\mathbf{V}\mathbf{D}^{-1}\right)\left(\mathbf{D}\mathbf{V}^{\top}\mathbf{w}\right)=\mathbf{U}\mathbf{w}_{u} \tag{6.22}$$

Algorithm 4 The Efficient EWSFA Algorithm

1. Normalize the collected data $\mathbf{X}$ to zero mean and unit variance.
2. Perform nonlinear mapping on $\mathbf{X}$ with AE or SVM, and form $\mathbf{Z} = \{\mathbf{z}(i) = \mathbf{f}(\mathbf{x}(i))\}_{i=1}^{n}$.
3. Conduct the economic SVD on $\mathbf{Z}_{d+1} = \{\mathbf{z}(i)\}_{i=d+1}^{d+1+N}$: $\mathbf{Z}_{d+1} = \mathbf{U}_{d+1}\mathbf{D}\mathbf{V}^{\top}$.
4. LV extraction: Initialize $\mathbf{w}_u$ with a random unit vector, and $\boldsymbol{\beta} = [\alpha, \alpha^2, \ldots, \alpha^d]^{\top}$, where α satisfies $\frac{\alpha(1-\alpha^d)}{1-\alpha} = 1$. Then repeat the following steps until convergence is achieved.
   a. Calculate $\mathbf{s} = \mathbf{U}\mathbf{w}_u$, where $\mathbf{U} = \mathbf{Z}\mathbf{V}\mathbf{D}^{-1}$, and form $\mathbf{s}_{d+1-i}$ for $i = 0, 1, \ldots, d$;
   b. Denote $\mathbf{S}_d = [\mathbf{s}_d, \mathbf{s}_{d-1}, \ldots, \mathbf{s}_1]$, and normalize $\mathbf{s}_{d+1}$ by $\mathbf{s}_{d+1} = \mathbf{s}_{d+1}/\|\mathbf{s}_{d+1}\|$;
   c. Calculate $\hat{\mathbf{s}}_{d+1} = \mathbf{S}_d\boldsymbol{\beta}$, and normalize $\boldsymbol{\beta}$ and $\hat{\mathbf{s}}_{d+1}$ with its norm $\|\hat{\mathbf{s}}_{d+1}\|$;
   d. Perform eigen-decomposition on Eq. (6.23), and update $\mathbf{w}_u$ as the eigenvector with the largest eigenvalue.
5. Deflation:
$$\mathbf{Z} := \mathbf{Z} - \mathbf{s}\mathbf{p}^{\top} \tag{6.20}$$
   where $\mathbf{p} = \mathbf{Z}^{\top}\mathbf{s}/\mathbf{s}^{\top}\mathbf{s}$;
6. Return to Step 3 until l sets of slow features are extracted.

where $\mathbf{U} \equiv \mathbf{Z}\mathbf{V}\mathbf{D}^{-1}$, and $\mathbf{w}_u \equiv \mathbf{D}\mathbf{V}^{\top}\mathbf{w}$. $\mathbf{U}$ is used to calculate $\mathbf{U}_{d-i+1}$ for $\mathbf{Z}_{d-i+1}$ ($i = 1, 2, \ldots, d$). With the defined denotations, Eq. (6.19) can then be rewritten as

$$\left(\mathbf{U}_{d+1}^{\top}\mathbf{U}_{d+1}\right)^{-1}\left(\mathbf{U}_{d+1}^{\top}\mathbf{U}_{\beta} + \mathbf{U}_{\beta}^{\top}\mathbf{U}_{d+1}\right)\mathbf{w}_u = \lambda_w\mathbf{w}_u \tag{6.23}$$

where $\mathbf{U}_{\beta} = \sum_{i=1}^{d}\beta_i\mathbf{U}_{d-i+1}$. Due to the orthonormality of $\mathbf{U}_{d+1}$, the matrix inversion $\left(\mathbf{U}_{d+1}^{\top}\mathbf{U}_{d+1}\right)^{-1}$ is not needed, and thus $\mathbf{w}_u$ is the eigenvector of $\left(\mathbf{U}_{d+1}^{\top}\mathbf{U}_{\beta} + \mathbf{U}_{\beta}^{\top}\mathbf{U}_{d+1}\right)$ corresponding to the largest eigenvalue.

The details of the efficient EWSFA are summarized in Algorithm 4, and the monitoring scheme based on efficient EWSFA will be deployed in the subsequent subsection.

6.4.5 EWSFA-based Monitoring Scheme

Based on the nature of EWSFA, a monitoring scheme is designed to detect faulty samples and dynamics anomalies. The complete set of LVs $\mathbf{S} \in \mathbb{R}^{n\times m}$ can be divided into the extracted slow features $\mathbf{S}_s \in \mathbb{R}^{n\times l}$ and the fast features $\mathbf{S}_f \in \mathbb{R}^{n\times(m-l)}$, where $\mathbf{S} = [\mathbf{S}_s, \mathbf{S}_f]$.

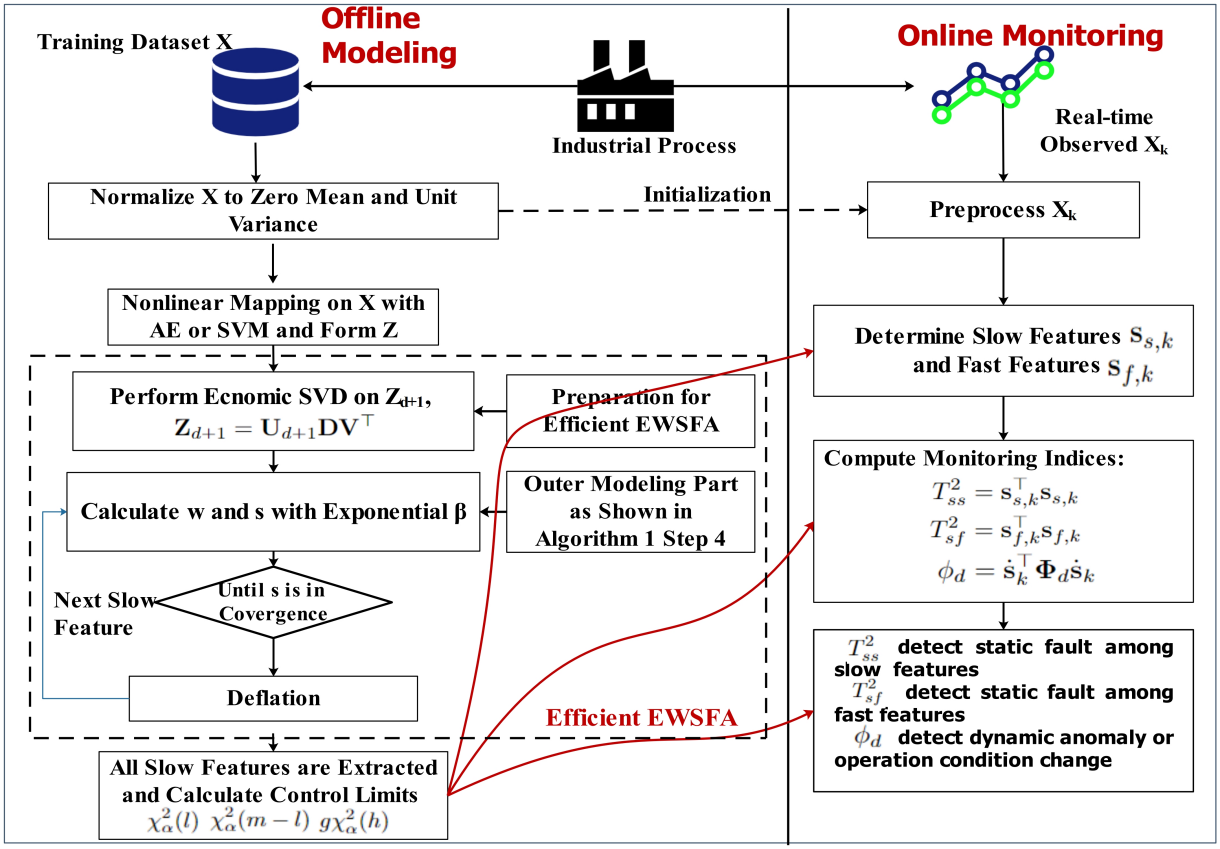


Figure 6.3: The Flowchart of EWSFA-based Monitoring Scheme

In our work, the number of slow features l is determined based on the *slowness* of the extracted features, which is defined as $\mathcal{S} \equiv \langle \dot{\mathbf{s}}^2_{d+1} \rangle_t = \langle (\mathbf{s}_{d+1} - \hat{\mathbf{s}}_{d+1})^2 \rangle_t$. It is noted that based on the proposed optimization goal in Eqs. (6.12) - (6.15), the range of $\mathcal{S}$ is $[0, 4]$: $\mathcal{S}$ is large ($\mathcal{S} \geq 2$) when $\mathbf{s}_{d+1}$ and $\hat{\mathbf{s}}_{d+1}$ is negatively correlated or not correlated, while it is smaller than 2 when the *slowness* is kept between them. In this work, to ensure the invariance between $\mathbf{s}_{d+1}$ and $\hat{\mathbf{s}}_{d+1}$, a threshold of 0.5 is selected; that is, the slowness magnitude of the extracted l slow features in $\mathbf{S}_s$ is smaller than 0.5, while the features in $\mathbf{S}_f$ have *slowness* values larger than this threshold.

The following Hotelling's T^2 statistics are designed for the slow and fast features respectively [76].

$$T^2_{ss} = \mathbf{s}^\top_{s,k} \mathbf{s}_{s,k} \sim \chi^2_\alpha(l) \tag{6.24}$$

$$T^2_{sf} = \mathbf{s}^\top_{f,k} \mathbf{s}_{f,k} \sim \chi^2_\alpha(m-l) \tag{6.25}$$

where $\mathbf{s}_k = [s_{1,k}, s_{2,k}, \cdots, s_{m,k}]^\top = \mathbf{R}^\top \mathbf{x}_k$, $\mathbf{R} = \mathbf{W}(\mathbf{P}^\top \mathbf{W})^{-1}$, $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_m]$, and $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_m]$. $\mathbf{s}_{s,k}$ and $\mathbf{s}_{f,k}$ are independent of each other, and assuming that they follow the Gaussian distribution, the T^2_{ss} statistic for $\mathbf{s}_{s,k}$ is approximated with a χ^2 distribution with l degrees of freedom, and the T^2_{sf} statistic for $\mathbf{s}_{f,k}$ follows a χ^2 distribution with $m-l$ degrees of freedom. T^2_{ss} is designed to monitor the static variations in the dominant slowest features, and T^2_{sf} is used to monitor the static variations in the fast features.

Furthermore, dynamic monitoring statistics are also proposed to detect dynamic anomalies or changes in operating conditions. In EWSFA, the invariance between the current sample and the past samples is

$$\dot{\mathbf{s}}_k = \mathbf{s}_k - \hat{\mathbf{s}}_k \tag{6.26}$$

where $\hat{\mathbf{s}}_k = \sum_{i=1}^{d} \boldsymbol{\beta}_i^\top \mathbf{s}_{k-i}$. T_{ds}^2 and Q_{ds} statistics are used to monitor the dynamic anomalies associated with $\dot{\mathbf{s}}_k$ in the principal and residual subspaces respectively.

$$T_{ds}^2 = \dot{\mathbf{s}}_k^\top \mathbf{P}_d \boldsymbol{\Lambda}^{-1} \mathbf{P}_d^\top \dot{\mathbf{s}}_k \tag{6.27}$$

$$Q_{ds} = \dot{\mathbf{s}}_k^\top \left(\mathbf{I} - \mathbf{P}_d \mathbf{P}_d^\top \right) \dot{\mathbf{s}}_k \tag{6.28}$$

where $\boldsymbol{\Lambda} = \frac{1}{N-1} \dot{\mathbf{S}}^\top \dot{\mathbf{S}}$ is the covariance matrix of $\dot{\mathbf{S}}$, and $\mathbf{P}_d$ is corresponding loading matrix as defined in Eq. (6.20) which contains the main variations, which can be obtained by performing PCA on $\dot{\mathbf{S}}$. Similar to T_{ss}^2 and T_{sf}^2, T_{ds}^2 and Q_{ds} can be approximated with χ^2 distributions, and their control limits are denoted as τ_{ds}^2 and δ_{ds}^2 respectively. Alternatively, a combined index ϕ_d can be derived by combining T_{ds}^2 and Q_{ds} [194] as follows

$$\phi_d = \dot{\mathbf{s}}_k^\top \boldsymbol{\Phi}_d \dot{\mathbf{s}}_k \tag{6.29}$$

where $\boldsymbol{\Phi}_d = \frac{T_{ds}^2}{\tau_{ds}^2} + \frac{Q_{ds}}{\delta_{ds}^2} = \frac{\mathbf{P}_d \boldsymbol{\Lambda}^{-1} \mathbf{P}_d^\top}{\tau_{ds}^2} + \frac{\mathbf{I} - \mathbf{P}_d \mathbf{P}_d^\top}{\delta_{ds}^2}$. The control limit of ϕ_d is then defined as

$$\phi_{d,\text{lim}} = g \chi_\alpha^2(h) \tag{6.30}$$

where

$$g = \frac{\text{tr}((\boldsymbol{\Lambda}\boldsymbol{\Phi}_d)^2)}{\text{tr}(\boldsymbol{\Lambda}\boldsymbol{\Phi}_d)}, \quad h = \frac{[\text{tr}(\boldsymbol{\Lambda}\boldsymbol{\Phi}_d)]^2}{\text{tr}((\boldsymbol{\Lambda}\boldsymbol{\Phi}_d)^2)}$$

$\text{tr}(\cdot)$ is the trace of a matrix, and α is the confidence level.

As for any unsupervised learning algorithm, in EWSFA the control limits are established with normal data, and in the online monitoring stage, T_{ss}^2, T_{sf}^2 and ϕ_d can be obtained for each new sample $\mathbf{x}_k$.

- If $T_{ss}^2 \geq \chi_\alpha^2(l)$, a fault that affects the static relations among the slow features is detected with $(1-\alpha) \times 100\%$ confidence level;
- If $T_{sf}^2 \geq \chi_\alpha^2(m-l)$, a fault that affects the static relations among the fast features is detected with $(1-\alpha) \times 100\%$ confidence level;
- If $\phi_d \geq g\chi_\alpha^2(h)$, an operating condition change is identified, or a dynamic anomaly is detected.

The details of the EWSFA-based monitoring scheme are summarized in Figure 6.3.

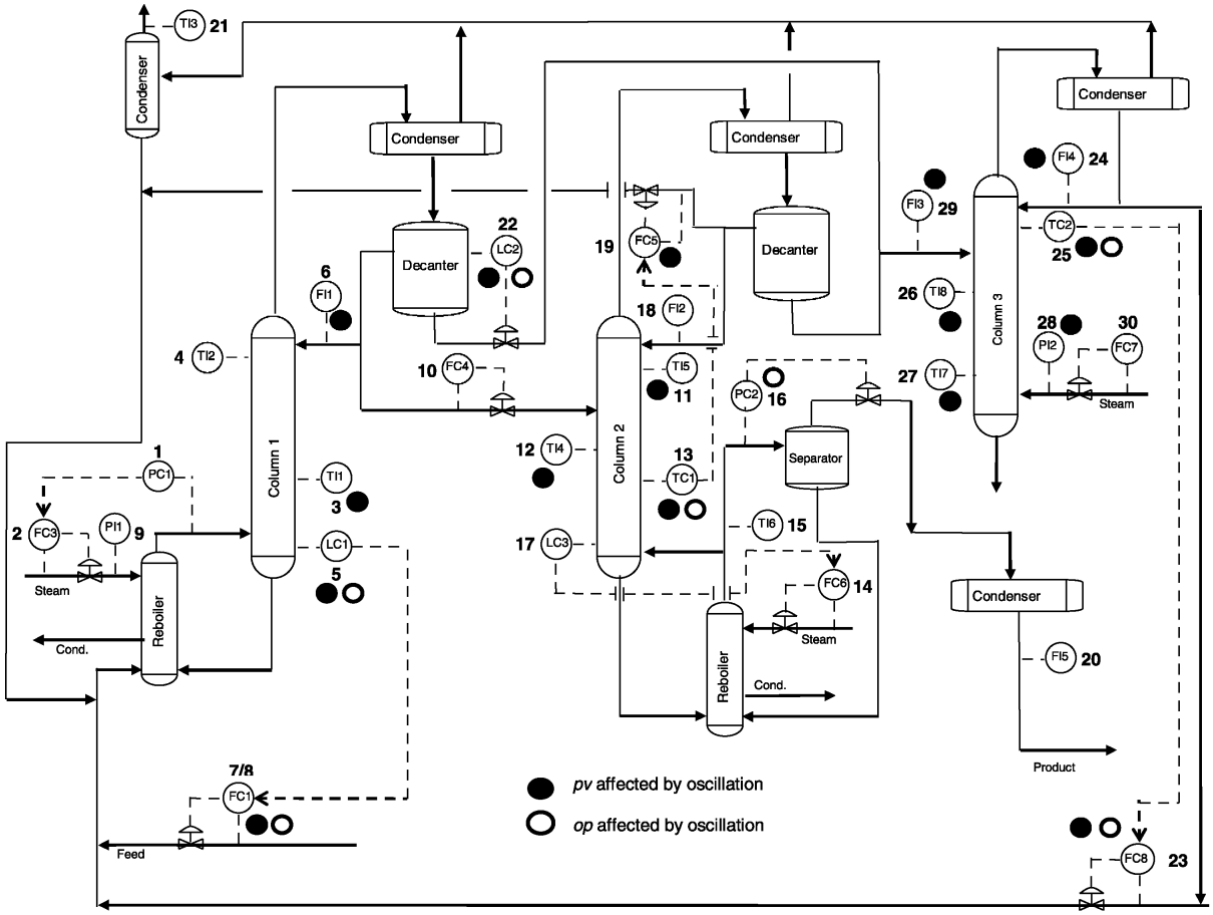


Figure 6.4: The Process Diagram of Eastman Chemical Company Plant

Table 6.1: R^2 of SFs of DiSFA and EWSFA

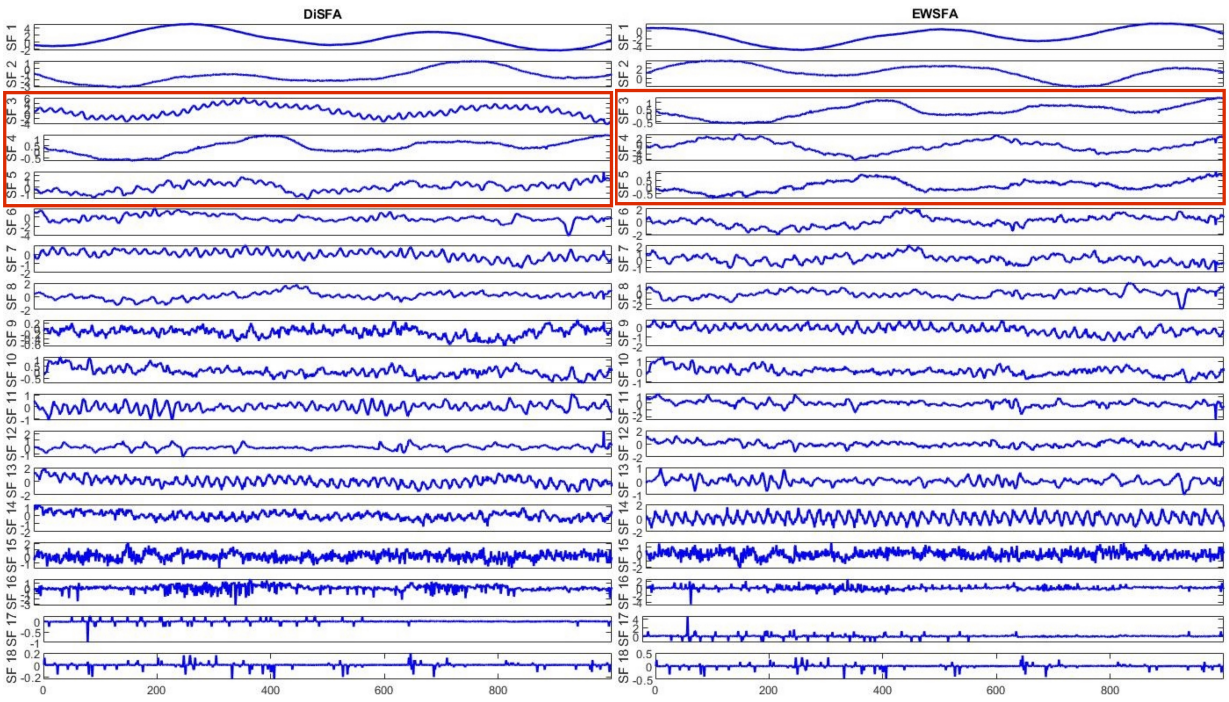
i^{th} Feature	1	2	3	4	5	6	7	8	9
EWSFA	1	0.9986	0.9973	0.9933	0.9851	0.9811	0.9664	0.9497	0.8988
DiSFA	0.9996	0.9988	0.9953	0.9951	0.9861	0.9639	0.9522	0.9513	0.923
i^{th} Feature	10	11	12	13	14	15	16	17	18
EWSFA	0.8833	0.837	0.8184	0.7178	0.6273	0.1346	0.041	0.0254	0.0015
DiSFA	0.8858	0.8268	0.7986	0.7862	0.697	0.2741	0.1299	0.1038	0.0161

6.5 Case Studies

6.5.1 Eastman Chemical Company

Process Description

In this subsection, a real-world industrial dataset provided by the Eastman Chemical Company [159] is used to compare the modeling performance of the proposed EWSFA algorithm over DiSFA. 8640 samples were collected in this process with each contain-



(a) DiSFA (b) EWSFA

Figure 6.5: Slow Features Extracted by DiSFA and EWSFA

ing 60 variables. Based on plant-wide oscillation signals, 18 oscillation variables, which are LC1.PV, LC1.OP, FC1.SP, FC1.PV, FC2.SP, FC2.PV, TI4.PV, TC1.PV, TC1.OP, FC5.OP, FC5.PV, LC2.PV, LC2.OP, FC8.SP, FC8.PV, TC2.OP, TI8.PV, FI3.PV, are selected as process variables where F, L, P, and T represent flow, level, pressure, and temperature respectively[193].

Numerical experiment results

The variations of the first 18 slow features (SFs) extracted by DiSFA and EWSFA are shown in Figures 6.5a and 6.5b respectively. Figure 6.6 depicts the correlation coefficients and similarity between the spectral features (SFs) extracted from EWSFA and DiSFA. The height of each bar in the figure reflects the degree of correlation between the corresponding SFs. A taller bar indicates a high degree of correlation, implying that the corresponding SFs are similar. For instance, in Figure 6.6, the bar positioned at (1,1) displays a large height, indicating that the first SFs of DiSFA and EWSFA exhibit congruent slowly varying signals. On the other hand, as shown in Figure 6.6, the SFs extracted by DiSFA and EWSFA are generally different. For example, in Figures 6.5a and 6.5b, the first two SFs of DiSFA and EWSFA exhibit the same slowly varying signals, which are also verified in Figure 6.6. However, in addition to the slow variations, a high oscillating signal is also involved in SF3 and SF5 of DiSFA, which indicates that DiSFA cannot guarantee a correct extraction order of *slowness*. EWSFA in Figure 6.5b successfully overcomes this issue by imposing a decaying of the weights for increasingly older samples. Table 6.1 shows the R^2 of SFs of DiSFA and EWSFA respectively. In Table 6.1, it shows that the R^2 of the first several

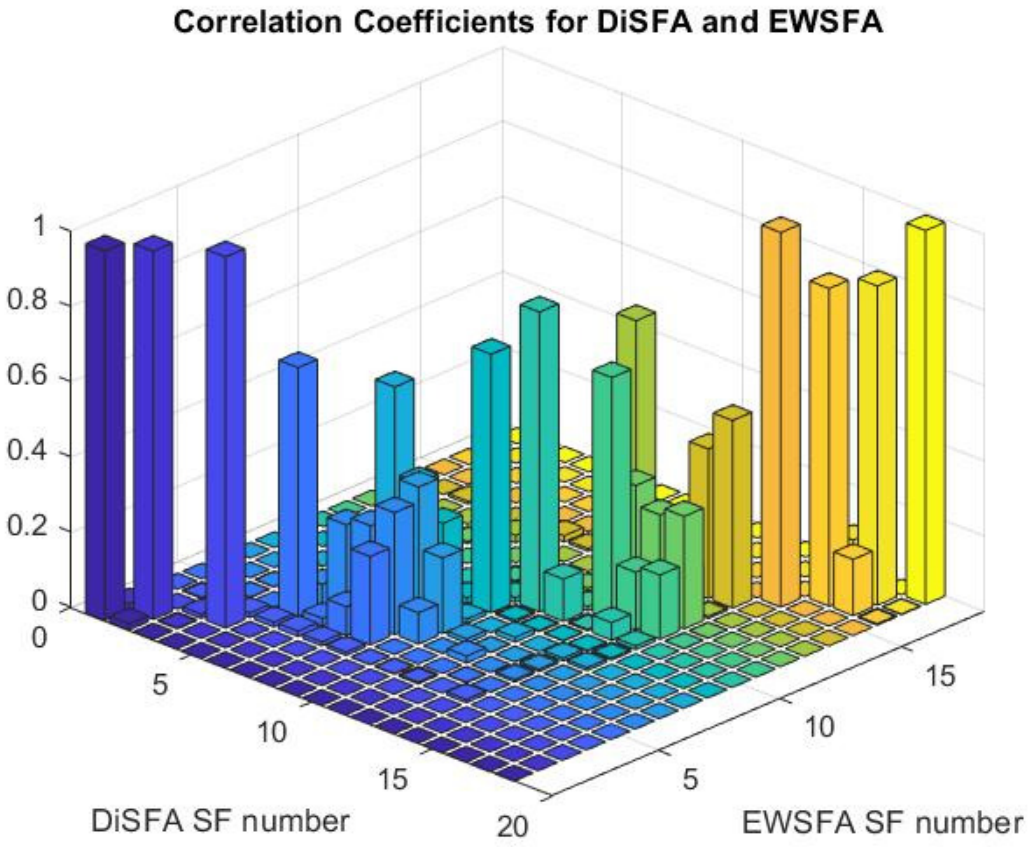


Figure 6.6: Correlation Coefficients of SFs of DiSFA and EWSFA

SFs of EWSFA is higher than DiSFA. Generally, the first several SFs contain the most important information about the process. Thus, the sensitivity of EWSFA is higher.

The fast Fourier transformation (FFT) is performed to further identify and visualize the oscillation period of the plant. Figure 6.7 shows the FFT results for the first four SFs of EWSFA. The oscillation period of SF1 is identified by the main peak, and its main cycle period is $1/e^{-5.8861} * 20s \approx 7200s$, which is consistent with the process description [159]. The FFT results of the first four SFs of DiSFA are summarized in Figure 6.8. The FFT of the first two SFs shows the same results as in EWSFA. However, for SF3, as outlined with the red rectangle in Figure 6.8, some peaks at high-frequency positions exist, implying that fast variations are also involved in the extracted SF in addition to the slowly varying signal. This corroborates that EWSFA is better able to capture dynamic slow features from the process with greater accuracy as compared to DiSFA. This is due to the utilization of the EWMA filtering that ensures that more recent samples have a greater impact on the *slowness* features of the industrial process.

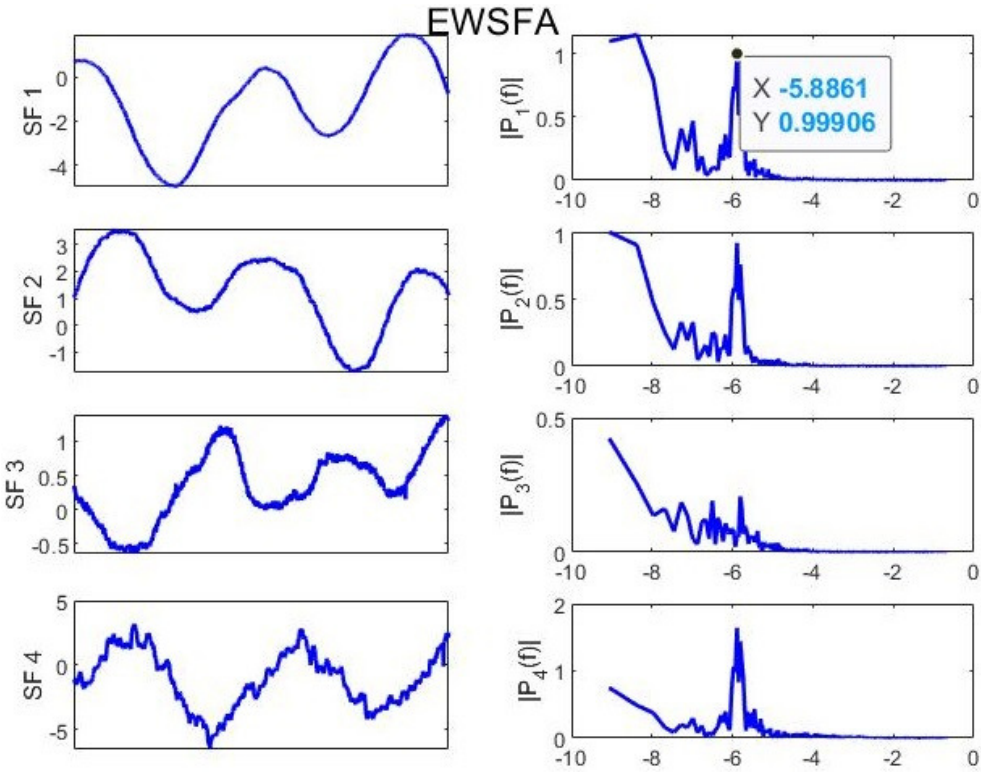


Figure 6.7: FFT for the First Four SFs of EWSFA (log scale)

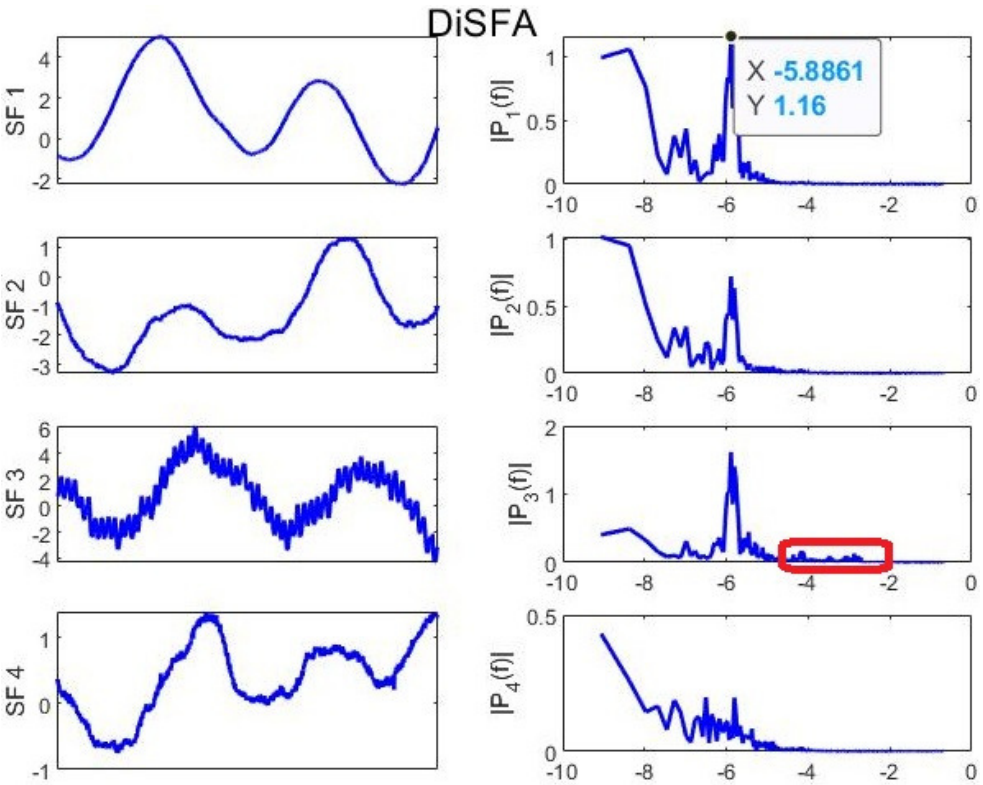


Figure 6.8: FFT for the First Four SFs of DiSFA (log scale)

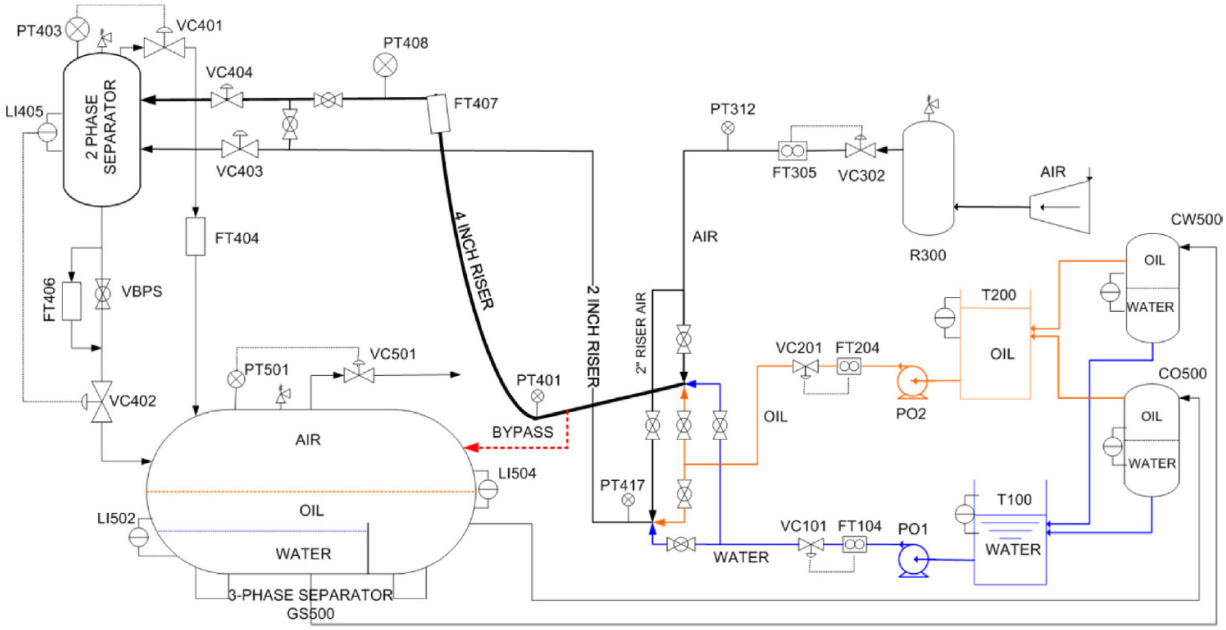


Figure 6.9: The Process Diagram of Three-Phase Flow Facility Plant

6.5.2 Three-Phase Flow Facility

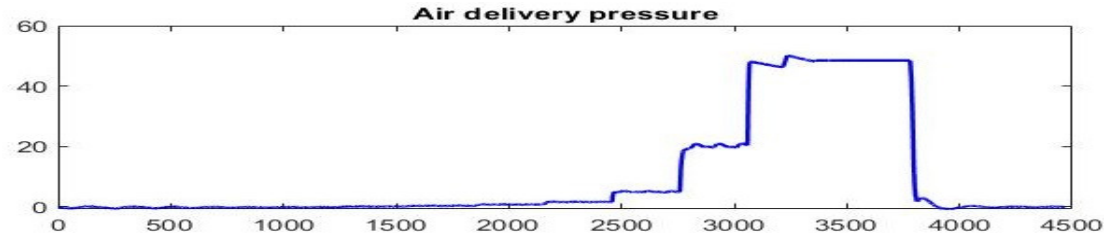
Process Description

A three-phase flow facility simulator developed by Cranfield University is used for demonstration of the proposed algorithm. The process includes measurements and control of the flow rates of air, water, and oil [141]. The system involves a two-phase separator, a three-phase separator, and two oil-water separators which are connected with pipelines. The Emerson Process Management company has designed a Fieldbus-based supervisory, control, and data acquisition system to manage the entire process [93].

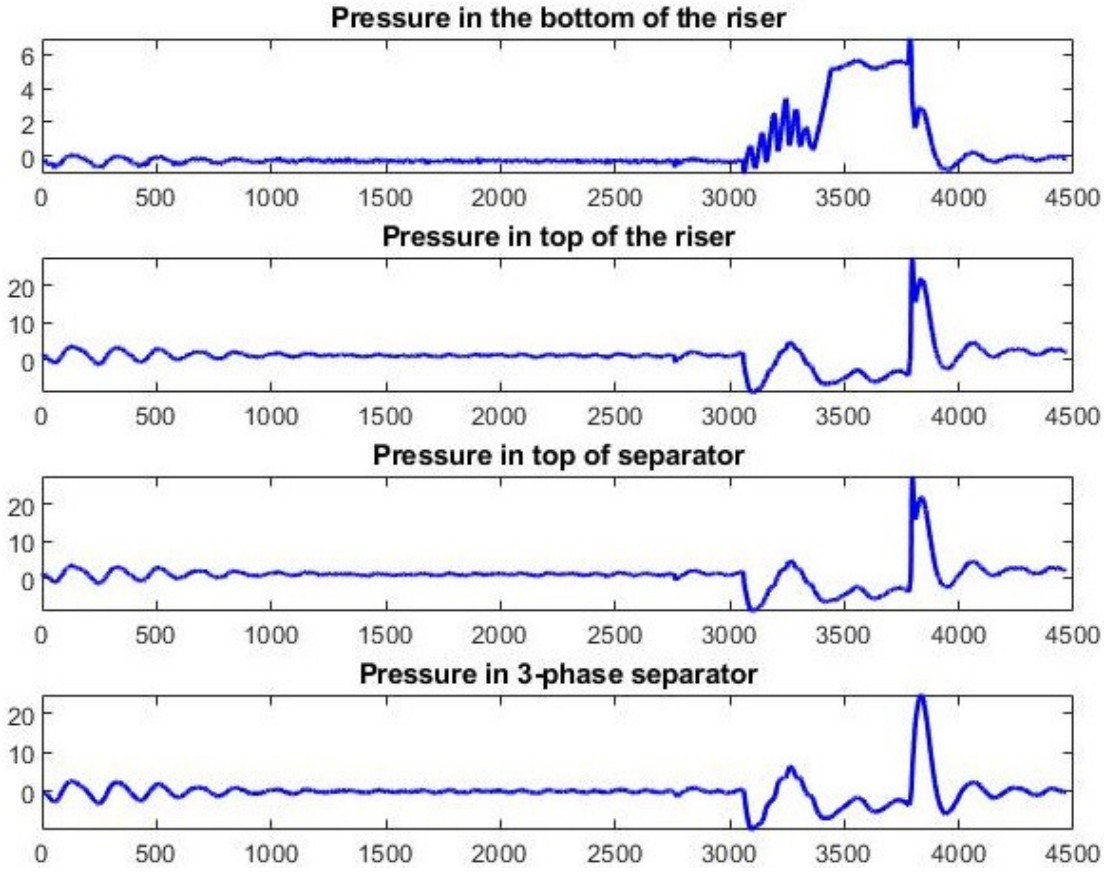
In this process, 24 process variables are measured at a sampling rate of 1 Hz, and three normal data sets (T_1, T_2 and T_3) are available in the process, where 20 different operating conditions with four air flow rate set points and five water flow rate set points are considered. Additionally, six faulty cases are simulated, and the first 23 process variables are used in faulty cases 1 to 5, while all of the 24 process variables are involved in faulty case 6. Faulty cases 1-4 occur for three different operating conditions, while faulty cases 5 and 6 occur for two different operating conditions, contributing to a total of 16 sets of faulty data. It is noted that the variable “Level gas-liquid 3 phase separator" is excluded in the following analysis since its variations are quite different in the training and testing stages.

Numerical experiments results

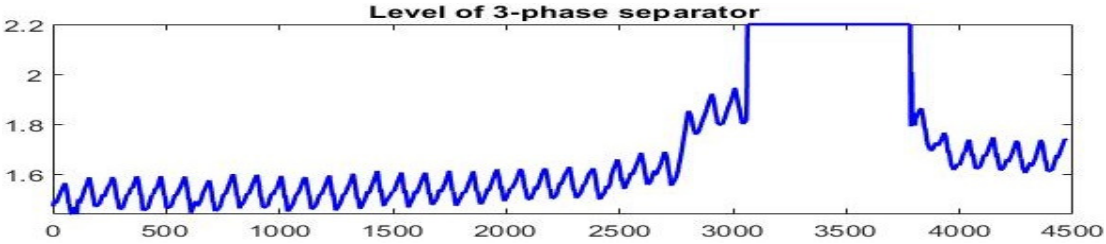
In this case study, DiSFA and DiPCA (dynamic inner principal component analysis [50]) are chosen to illustrate the advantages of EWSFA-based monitoring scheme. It is noted that



(a) Variable Air Delivery Pressure in Fault 1.2



(b) Variables about the Pressure of Devices in Fault 1.2



(c) Variable Level of 3-phase Separator in Fault 1.2

Figure 6.10: Details of Some Important Variables in Fault 1.2

there are three indices designed for DiPCA which are different from DiSFA and EWSFA. For DiPCA, the dynamic index Φ_d is designed through dynamic residuals, and static indices T_x^2 and Q_x are designed through static residuals. More details about DiPCA-monitoring

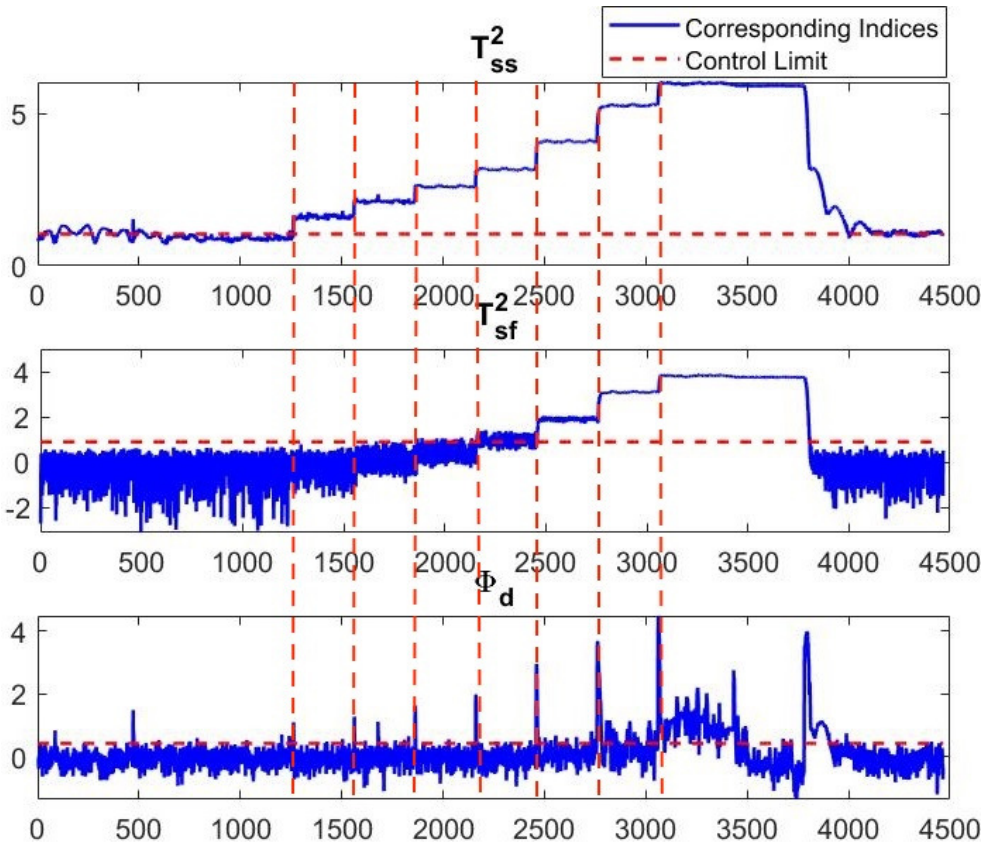


Figure 6.11: EWSFA-based Monitoring Results for Fault 1.2 (semi-log scale)

can be found in [50]. For the simulated data, normal operation data sets are used as training data and the faulty data are selected as testing data. Based on the aforementioned rules of parameter selection, the number of SFs l is selected as 10 and the dynamic order d is set as 3. For simplicity, Fault 1.2 and 3.3 are selected for detailed illustration. It is noted that the monitoring figures for TPFF data are plotted in a semi-log form for better visualization.

An airline blockage is introduced gradually in Fault 1.2 and is removed at the end. Figure 6.10a shows the air delivery pressure variations in the testing data, and it indicates that several obvious step changes are introduced starting at around 1500$^{\text{th}}$ sample. The EWSFA-based, DiSFA-based, and DiPCA-based monitoring results are presented in Figures 6.11-6.13 respectively. As shown in Figure 6.11, EWSFA detects the anomalies in slow features with T^2_{ss} at around 1262$^{\text{th}}$ sample, which is even before the change is visually observed in Figure 6.10a. Compared with T^2_{sf}, T^2_{ss} can capture all step changes, while T^2_{sf} can only capture the anomalies when their fault magnitudes become large. Further, some variables, such as the ones in Figure 6.10b, retain oscillations in the beginning period, which are detected in slow features with T^2_{ss} but not in fast features. Similar patterns are observed after the 3777$^{\text{th}}$ sample. The T^2_{ss} index indicates that the process does not return to normal yet after 3777$^{\text{th}}$ sample, which is consistent with the actual variations in the variable level of the 3-phase separator as shown in Figure 6.10c. However, the fault effect is not detected in the fast variations with T^2_{sf} index. Thus, it can be concluded that T^2_{ss} is more sensitive than T^2_{sf}, and it can reflect the main process variations. For the dynamic

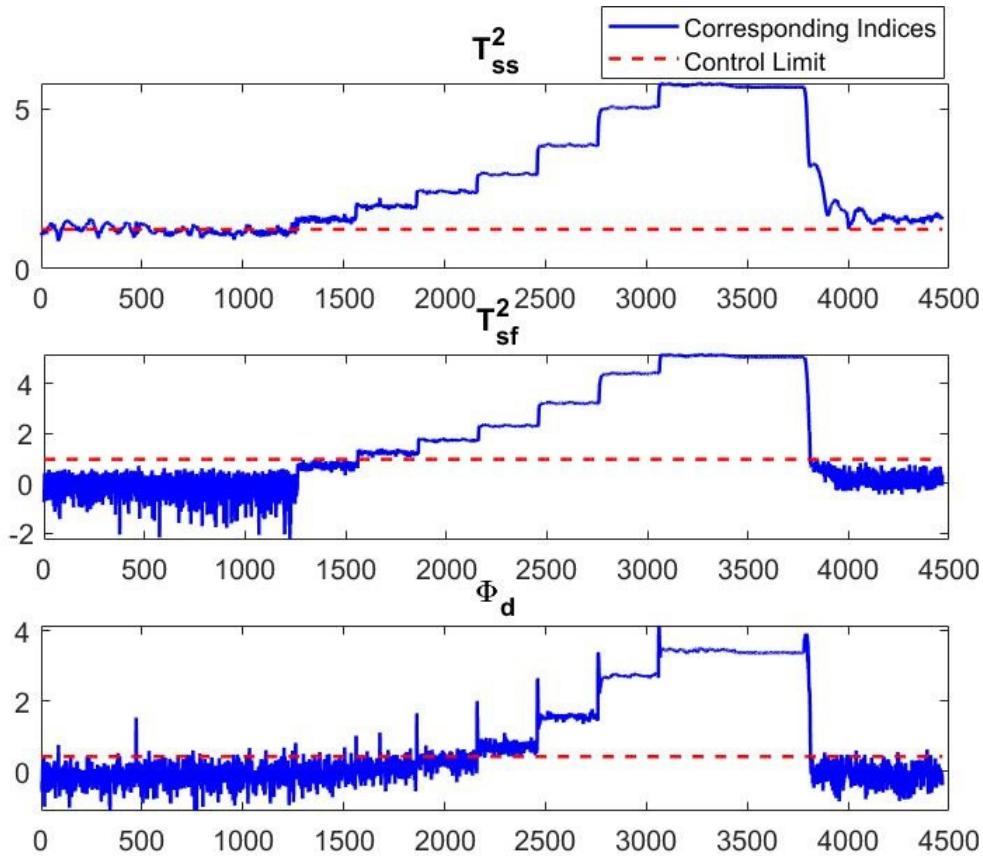


Figure 6.12: DiSFA-based Monitoring Results for Fault 1.2 (semi-log scale)

monitoring index ϕ_d in Figure 6.11, only several peaks are detected, which are in line with the occurrences of the step changes, and these detected anomalies imply the change of operating conditions.

Compared with EWSFA, T^2_{ss} of DiSFA in Figure 6.12 also achieves good monitoring performance. However, DiSFA cannot separate slow variations from fast ones as well, and some slow variations are still left in the fast features, leading to indistinguishable monitoring performance of T^2_{ss} and T^2_{sf}. As shown in Figure 6.12, there is no peak in the dynamic index Φ_d for step change appears at 1262^{th} sample, and for most of the step changes, Φ_d still raises the false alarms even when the process is steady. Thus, Φ_d of DiSFA fails to capture the process dynamics. As shown in Figure 6.13, combining the monitoring results of T^2_x and Q_x of DiPCA, it is shown that DiPCA cannot detect the anomalies from 1262^{th} to 1500^{th} samples. For Φ_d, it cannot detect any dynamics of step changes before 2750^{th} samples and still generates false alarms even when the process is steady. Therefore, Φ_d of DiPCA can not capture the process dynamics accurately as EWSFA.

The other fault case, Fault 3.3, introduces a top separator input blockage, and it will affect "Flow rate top separator output" mainly as shown in Figure 6.14. From Figure 6.14, the top separator input blockage starts at 596^{th} and ends at 9566^{th}. Figures 6.15-6.17 show the monitoring results of EWSFA, DiSFA, and DiPCA respectively. As shown in Figure 6.15, T^2_{ss} and T^2_{fs} can detect the anomalies successfully. Furthermore, as shown in Figure 6.14, the system does not return to normal immediately after the fault is removed

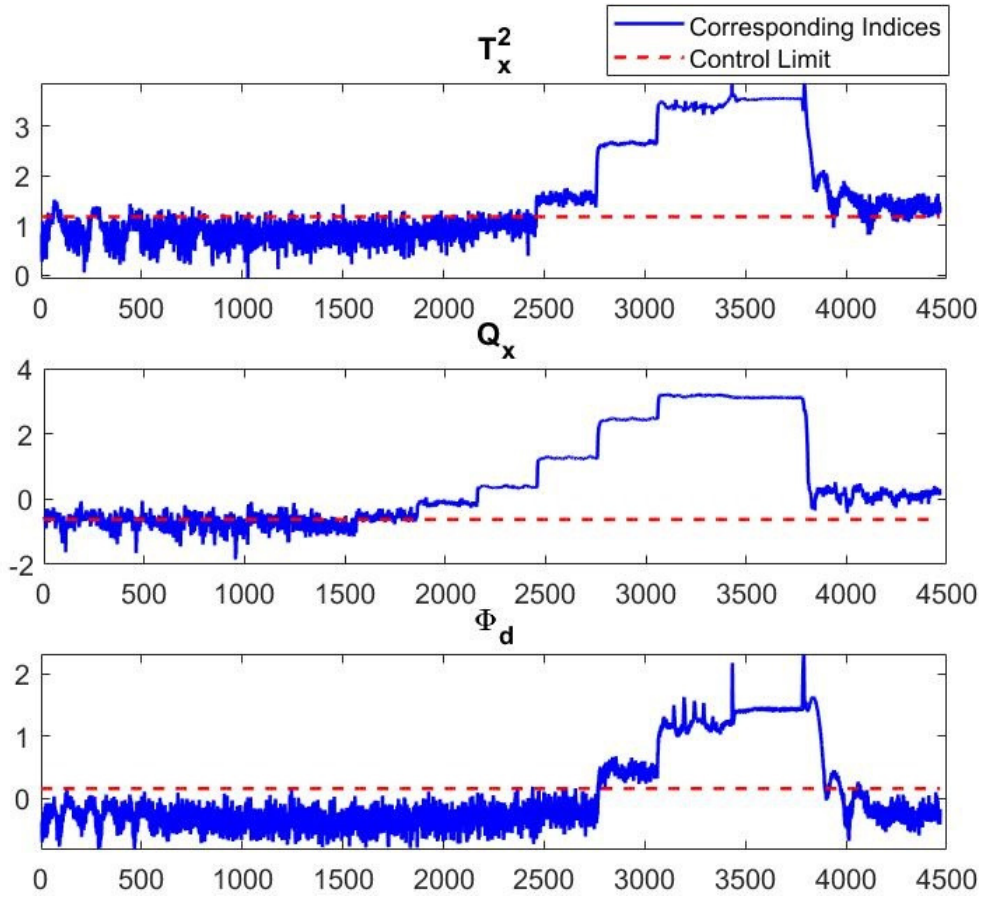


Figure 6.13: DiPCA-based Monitoring Results for Fault 1.2 (semi-log scale)

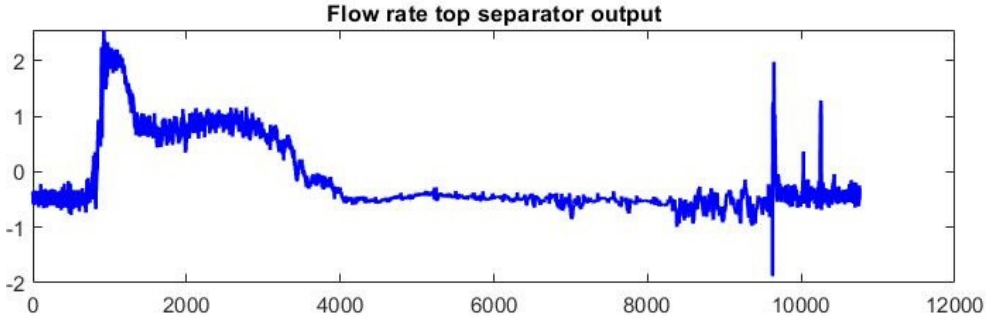


Figure 6.14: Variable Flow Rate Top Separator Output in Fault 3.3

which can be captured and detected by T_{ss}^2 but not T_{fs}^2. This difference between T_{ss}^2 and T_{fs}^2 gives a further verification that T_{ss}^2 is more sensitive than T_{sf}^2. As aforementioned, the operating condition changes are detected with the dynamic index Φ_d after the fault is introduced. In this fault, it is also identified that the system reaches a new steady-state at samples between 4200^{th} and 8000^{th} as shown in Figure 6.14. Then, the monitoring difference between EWSFA and DiSFA, DiPCA of Fault 3.3 is mainly reflected on the dynamic index Φ_d. In DiSFA and DiPCA, samples 4200^{th} to 8000^{th} are still out of control, and DiSFA and DiPCA inaccurately predict that dynamic anomalies still exist when the system reaches the new steady state. Also, they fail to detect the large pulses in the

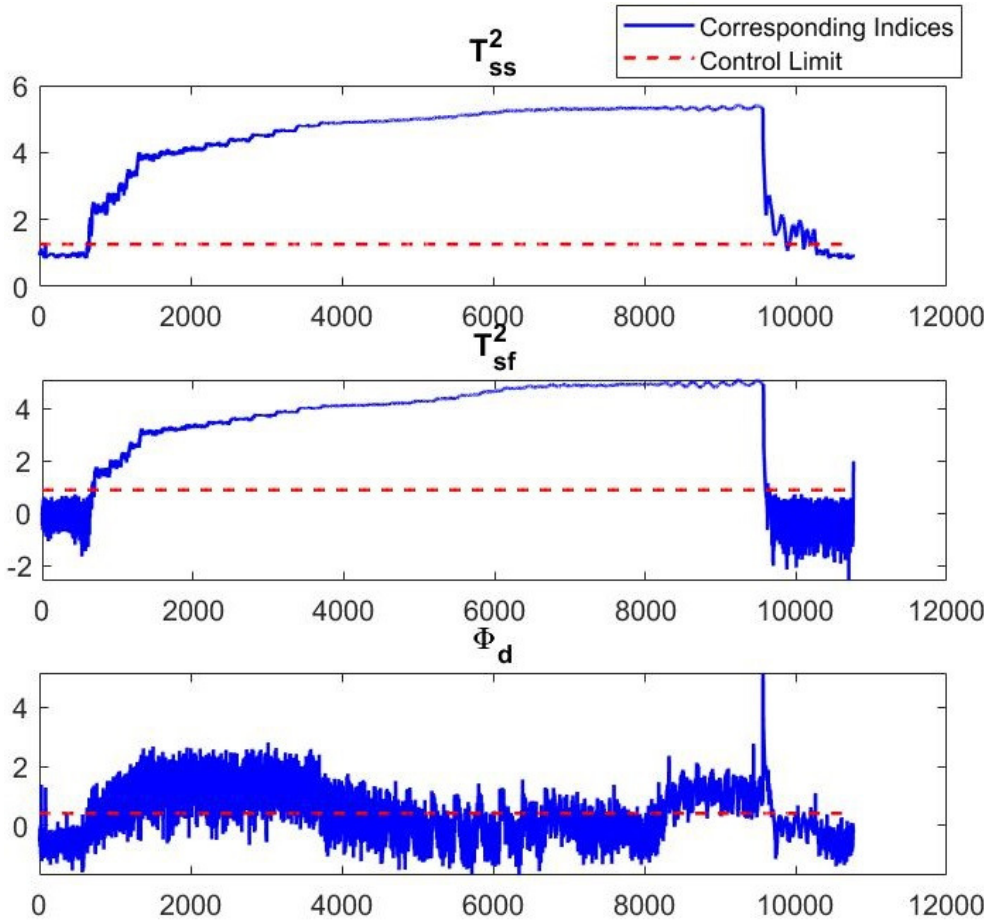


Figure 6.15: EWSFA-based Monitoring Results for Fault 3.3 (semi-log form)

end. Therefore, it can be inferred that the EWSFA's dynamic monitoring performance is more precise and responsive than that of DiSFA and DiPCA, particularly when the system reaches a new steady state. This is because the dynamic features extracted by EWSFA encompass more significant information about the process than those extracted by DiSFA and DiPCA.

6.6 Summary

In this chapter, a novel exponentially weighted SFA method is proposed to extract multi-step dynamics in modern manufacturing processes. To ensure that slowly varying signals are extracted, EWSFA aims to minimize the difference between the latent variables of the current sample and historical samples. The exponentially weighted moving average is designed in EWSFA to ensure the decaying nature of temporal variations, and singular value decomposition is adopted to improve the computation effectiveness. Based on the statistics of the extracted latent variables, an EWSFA-based monitoring scheme is designed to detect anomalies in both slow and fast variations, where a novel dynamic index Φ_d is proposed to monitor operating condition changes. Case studies with the Eastman Chemical Company

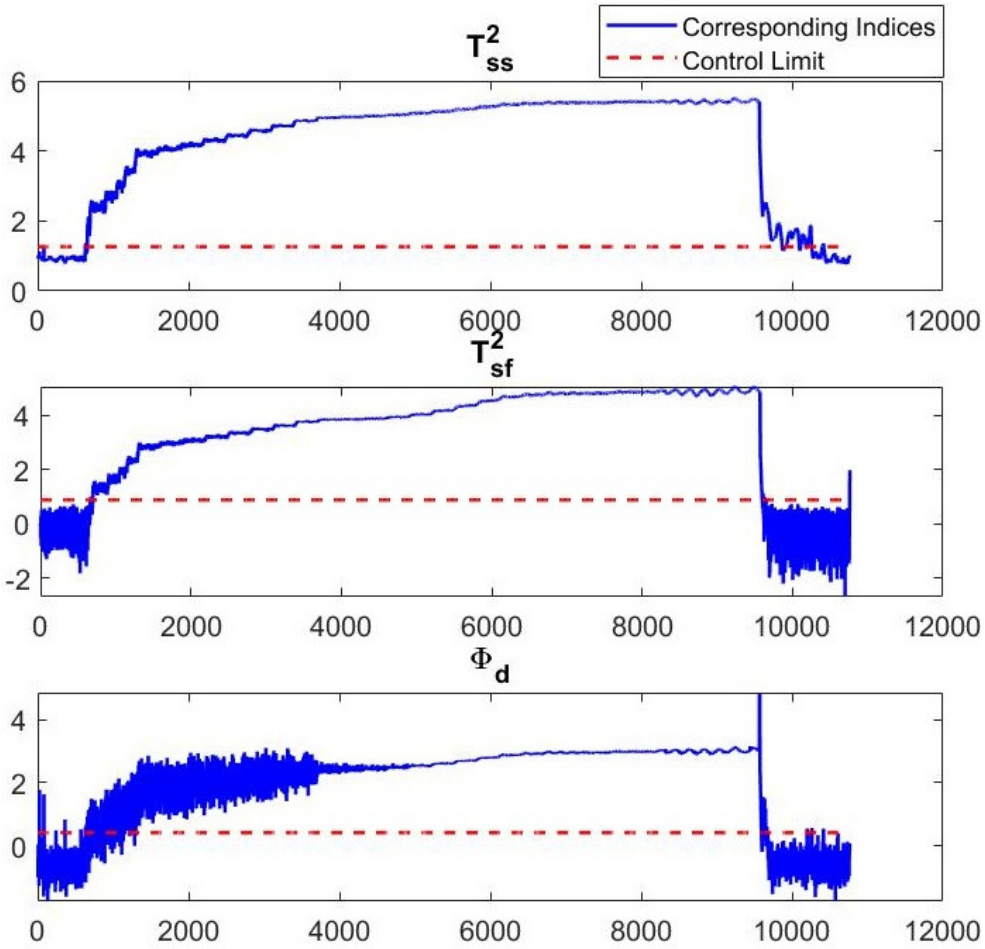


Figure 6.16: DiSFA-based Monitoring Results for Fault 3.3 (semi-log form)

and the Three-Phase Flow Facility illustrate the modeling and monitoring effectiveness of EWSFA as compared to other methods.

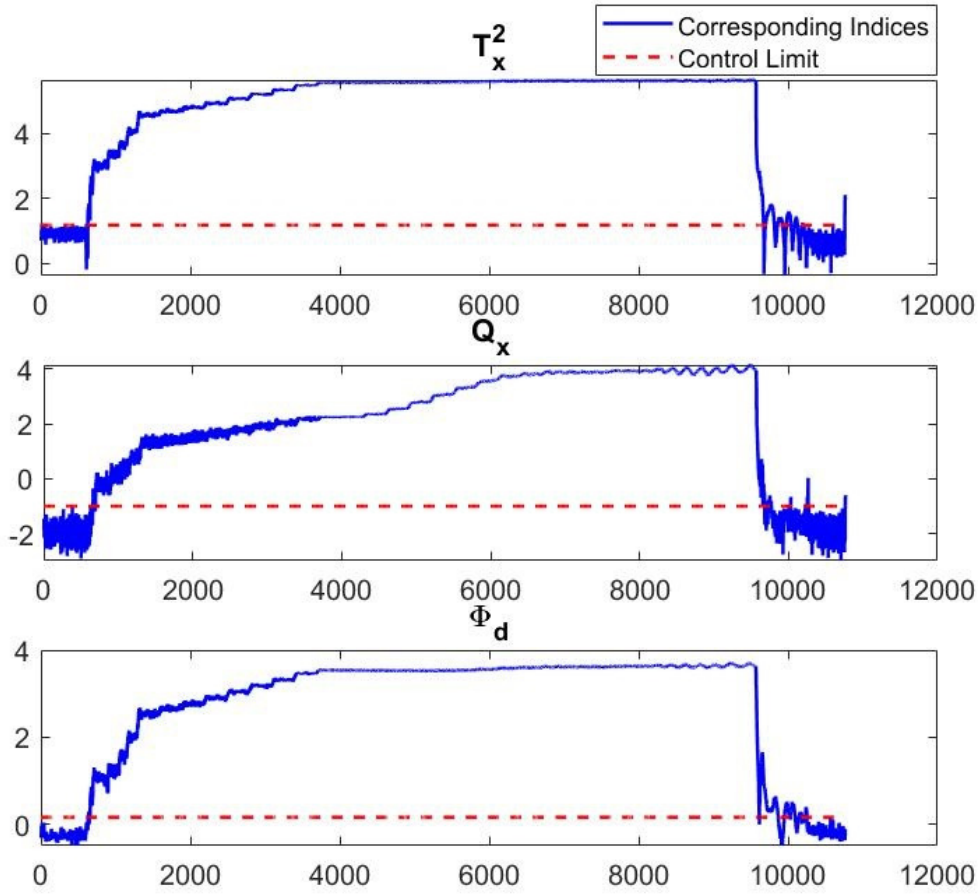


Figure 6.17: DiPCA-based Monitoring Results for Fault 3.3 (semi-log form)

Chapter 7

Conclusions and Futhure Work

7.1 Conclusions

Dynamic inner latent variable models (DiLVN) are a type of machine learning model that can effectively capture the dynamics of the data. The research in this thesis presents novel variants of these methods and validates them through numerical simulations of industrial case studies. The novel methods proposed in this work tackle three main issues: i- treatment of redundant information, ii- full utilization of quality variables' data, and iii- interpretability of the results with respect to dynamic behavior.

7.1.1 Methods to Solve the Redundant Information

The findings of this part of the research can be summarized as follows:

1. Owing to the prevalent high sampling rates observed in industrial processes, it is common for successive sampled data to contain redundant and collinear information. When utilizing these data directly for modeling and monitoring, sub-optimal performance may result. In this research, we propose a novel approach known as Dynamic Weighted Partial Least Squares (DWPLS) method to address the issue of redundant information present in neighboring samples.

   The DWPLS method effectively reduces the inter-dependency between adjacent samples by introducing a series of basis functions, such as polynomial and Gaussian bases. These basis functions play a crucial role in mitigating the impact of redundant information, thereby enhancing the overall modeling and monitoring performance. To ensure consistency between the dynamic inner and outer models, the quality scores are represented using a lagged vector based auto-regressive model within the inner model. This choice allows for the incorporation of temporal dynamics and helps capture the time-varying behavior of the data. Furthermore, in the outer model, the model is designed to maximize the covariance between the output latent scores and a weighted combination of lagged input scores. These lagged input scores are

constructed using a weighted summation of the basis functions, contributing to the comprehensive handling of redundant information and maintaining model consistency.

2. The DWPLS method entails a decomposition of the input space into two distinct subspaces: the dynamic principal subspace and the residual subspace. The dynamic principal subspace captures the variations in the input data that are directly related to the output, thereby carrying essential and informative content for modeling purposes. Conversely, the residual subspace accommodates the information that is deemed quality-irrelevant, as it does not contribute significantly to the understanding and representation of the output. By segregating the input space in this manner, DWPLS effectively separates the relevant signal from the noise or extraneous factors present in the data. This separation enables a more focused and accurate prediction of the quality, as the dynamic principal subspace predominantly drives the system's behavior and influences the output of interest. Through the dynamic principal subspace, DWPLS emphasizes the temporal dynamics and dependencies in the data, which are of paramount importance in capturing the time-evolving characteristics of the underlying process. In contrast, the residual subspace accounts for uncorrelated and insignificant variations, which, when included in the modeling procedure, could lead to suboptimal and misleading results. Thus, the decomposition scheme employed by DWPLS represents a principled approach to enhance modeling efficiency by prioritizing salient features while disregarding noise and redundancy. By elucidating the intrinsic structure of the data and delineating the essential factors affecting the output, DWPLS contributes significantly to improved modeling accuracy and performance in various industrial processes. Its capability to discern and utilize the information residing solely in the dynamic principal subspace underlines the method's effectiveness and relevance in addressing the challenges posed by high sampling rates and redundant information in successive sampled data.

3. The case studies section discusses the effectiveness of the algorithm based on the Tennessee Eastman process and a distillation process which includes the comparison of prediction performance between DiPLS and DWPLS with different basis functions, the influence of the selection of parameter points, and the effectiveness of the predictive monitoring schemes.

   For the Tennessee Eastman process, it is shown that DWPLS demonstrates superior predictive performance in comparison to DiPLS. Notably, within the various DWPLS variants, DWPLS with polynomial basis function emerges as the top-performing method, as evidenced by its prediction and monitoring performance, as quantified by the MSE, FDR, FAR, and corresponding figures. The complexity of industrial processes leads to intricate variations, but when utilizing Gaussian basis functions, the generated variations exhibit a relatively simpler pattern. Moreover, the flexibility of Bézier basis functions is contingent on the number of control points, with its performance being constrained by the small value of s. Comparatively, polynomial bases offer greater flexibility and, consequently, yield the most favorable modeling performance in this specific scenario when contrasted with the aforementioned bases.

Additionally, an investigation into the influence of parameter points on the modeling performance of DWPLS has been conducted.

For the distillation process, it can be concluded that all variants of DWPLS exhibit superior performance compared to DiPLS, as evidenced by their smaller Mean Squared Errors (MSEs), with DWPLS with polynomial basis function demonstrating the most favorable outcomes. This aligns consistently with the observations drawn from the TEP case study, further corroborating the superiority of DWPLS with polynomial basis function. The predictive capabilities of DWPLS with polynomial basis function are visually represented in the corresponding figure. Notably, in contrast to the TEP case study findings, DWPLS with Gaussian basis function surpasses DWPLS with Bézier basis function in this particular instance. This discrepancy emphasizes that distinct types of basis functions may align more effectively with the inherent properties of different datasets or processes. Thus, as articulated in Remark 2, a comprehensive evaluation of diverse basis functions should be conducted for each new case study for accurately estimating the data distribution and identifying the optimal one.

7.1.2 Methods to Fully Exploit Quality Information

Chapter 4 and Chapter 5 present two snovel approaches aimed at maximizing the utilization of quality information. These methodologies are specifically designed for comprehensive monitoring and addressing the issue of missing data imputation respectively.

The findings of DALVM can be summarized as follows:

1. Within the realm of contemporary advanced dynamic inner methods, there exists an untapped potential in the use of auto-correlations between current and past quality scores for improved predictions of quality. This underexplored aspect presents an opportunity for enhancing both modeling and monitoring performance [219]. To address this gap, we present in Chapter 4 a novel approach referred to as the dynamic auto-regressive latent variable model (DALVM). This model is designed to effectively capture the intricate dynamics of quality variables by incorporating information from both past process samples and past quality samples.

   In the context of the outer model, the DALVM is strategically optimized to maximize the covariance between the quality score and a weighted combination of both historical process and quality scores. The resulting alignment of the inner with the dynamic outer model further reinforces the effectiveness of our proposed approach. In conjunction with the dynamic outer model, our framework includes an auto-regressive exogenous (ARX) model integrated within the inner model. This ARX model plays a pivotal role in accurately predicting the quality scores, thereby bolstering the overall performance and reliability of the DALVM. Our work represents a significant step forward in harnessing the potential of auto-correlations in quality scores, ultimately leading to substantial improvements in modeling and monitoring efficacy.

2. Drawing inspiration from the work of CPLS [132], we explore further advancements in decomposing the latent structures extracted by the dynamic auto-regressive latent variable model (DALVM). This enhanced approach, concurrent DALVM (ConDALVM), represents a significant step forward for improved process monitoring. Unlike conventional concurrent designs that primarily focus on static processes, our proposed ConDALVM explicitly addresses the decomposition of dynamic and auto-regressive variations. Through the utilization of ConDALVM, we effectively exploit both dynamic and static variations originating from process and quality spaces. This comprehensive utilization of variations enables us to develop a reliable monitoring scheme. By encompassing dynamic and auto-regressive aspects in our concurrent design, we significantly extend the applicability and effectiveness of the monitoring framework, overcoming the limitations of conventional methods.

3. Within the case study section, we employ a toy example simulation and two industrial processes to exemplify the efficacy of the proposed Dynamic auto-regressive Latent Variable Model (DALVM) and Conditional Dynamic auto-regressive Latent Variable Model (ConDALVM) in terms of modeling and monitoring capabilities.

   The synthetic simulation entails generating 1200 samples using Eq. (1) as specified, with a division of these samples into distinct sets for training, model parameter selection via cross-validation, and testing. Specifically, the first 800 samples are designated for model training, the subsequent 200 samples are employed for model parameter selection through cross-validation, and the final 200 samples are reserved for the testing phase. The analysis of Mean Squared Errors (MSEs) and graphical representations clearly demonstrate that the predictions obtained from DALVM closely align with the actual variations in the data, exhibiting the smallest MSEs among all the methods evaluated. Conversely, the performance of the traditional Partial Least Squares (PLS) approach proves to be inadequate, primarily due to its limitation in capturing only static relationships between variables. As for the DiPLS method, it necessitates the incorporation of additional lag numbers to achieve satisfactory prediction results. In contrast, the DALVM surpasses the other methodologies in terms of prediction accuracy, and remarkably, its lag numbers of quality and process variables align consistently with the functional form represented in Eq. (4.44). The examination of auto-correlations and cross-correlations in the residuals of quality variables for both DALVM and DiPLS provide insightful findings about the process. Specifically, it is evident that the residuals obtained through DALVM do not include significant dynamic content, which corroborates the comprehensive extraction process employed by DALVM. Conversely, the residuals obtained from DiPLS retain dynamic variations even after the application of the method. Notably, this comparison highlights DALVM's superior ability to effectively capture and model the auto-regressive structure inherent in the quality variables, leading to better performance in dynamic modeling as compared to DiPLS.

   Subsequently, the investigation extends to the application of DALVM in the context of two additional industrial processes: 1-the TEP (Test Engineering Platformor Tennessee Eastman Process) and 2- a Three-Phase Flow Facility dataset.

For the TEP dataset, upon conducting a comprehensive comparative analysis of the actual and predicted quality variations among various models, along with the corresponding Mean Squared Errors (MSEs), important conclusions can be drawn. Notably, DALVM stands out by attaining the most superior prediction performance, as evidenced by its smallest MSE and notably minimal required time lag. The predictions generated by DALVM exhibit a remarkable alignment with the actual quality variations, confirming its capacity to correctly capture and model dynamic cross-correlations and auto-correlations with heightened precision. As a result, DALVM proves to be particularly proficient in rendering more accurate representations of the quality variations, outperforming alternative methodologies in this regard.

For both the TEP dataset and a Three-Phase Flow Facility dataset, based on the False Detection Rates (FDRs) and False Alarm Rates (FARs) for both ConDALVM and DiPLS, noteworthy patterns emerge. ConDALVM exhibits lower FARs compared to DiPLS when dealing with quality-irrelevant disturbances, indicating its superior ability to accurately reject false alarms in such scenarios. Conversely, in instances pertinent to quality-relevant disturbances, ConDALVM achieves higher FDRs while maintaining comparable FARs. This suggests that ConDALVM displays a superior ability to detect true positives in quality-relevant cases, despite the presence of a comparable rate of false alarms as observed with DiPLS.

The findings of EDACCA can be summarized as follows:

1. In order to enhance the predictive capabilities of DALVM, an advanced missing data imputation strategy, is proposed, employing a multi-step ahead prediction method based on the application of the Efficient Dynamic Auto-Regressive Canonical Correlation Analysis (EDACCA) method. The EDACCA approach aims to establish dynamic relationships between the current quality sample and a carefully weighted amalgamation of past process and quality data. Notably, the introduction of two weighting parameters enhances the flexibility of the model, while various techniques, including singular value decomposition, are integrated into the calculations to bolster its overall computational efficiency. By leveraging these improvements, the proposed strategy seeks to effectively address missing data challenges and elevate the prediction performance of DALVM.

2. EDACCA offers predictions under the premise of having access to the current and past s process data, as well as the past d quality data. Nevertheless, the deployment of diverse sampling rates across various sensors may result in the absence of data from slower sampling variables. Additionally, the modeling and monitoring performance may suffer in the presence of faults affecting certain variables. As a remedy, the implementation of multi-step-ahead (MS) predictions becomes imperative to circumvent the constraints imposed by regular sampling rates and the deleterious effects of faulty data. In pursuit of this objective, an extension of EDACCA is devised to accommodate an MS prediction scheme, enabling greater adaptability and robustness in handling the challenges associated with variable sampling rates and data pollution from faults. Furthermore, the derived models can be effectively harnessed for diverse

objectives, including fault monitoring and diagnosis, as elucidated in the scholarly work by Xu and Zhu [185]. Moreover, in the context of practical industrial processes, the presence of irregular sampling poses significant challenges. To overcome this challenge, the developed MS prediction scheme is introduced as a missing data imputation approach, thereby offering a viable solution to cope with the prevalent issue of missing data in quality variables.

3. Within the case study section, we conduct a comprehensive evaluation of the modeling and imputation efficacy of the proposed EDACCA framework using two industrial processes: the Tennessee Eastman process (TEP) and a wind process. These industrial processes were carefully selected to rigorously assess the performance and applicability of EDACCA in real-world scenarios. The choice of these distinct processes enables a thorough investigation into EDACCA's ability to accurately model dynamic relations and effectively address missing data imputation challenges, thereby providing valuable insights into its potential applicability across various industrial domains.

4. For TEP, a comparative analysis is conducted among several advanced dynamic algorithms, namely EDACCA, DACCA, DAPLS, DiPLS, and sDiCCA. The model parameters are estimated through the utilization of cross-validation, ensuring robustness and reliability in the assessment. Subsequently, the prediction results of these models, along with their corresponding mean squared errors (MSEs), and coefficients of determination (R^2), are meticulously summarized. The comparison reveals that EDACCA, DACCA, and DAPLS outperform dynamic inner LVMs such as DiPLS and sDiCCA in terms of prediction performance. Notably, EDACCA emerges as particularly noteworthy, as it demonstrates the ability to more accurately capture system dynamics, leading to the best predictive modeling performance among all examined models. This superiority is evidenced by its smallest MSE and largest R^2.

   The predictive contributions from the dynamic cross-correlations $\hat{\mathbf{Y}}_{xg}$ and the dynamic auto-correlations $\hat{\mathbf{Y}}_{yg}$ are further studied. With the added flexibility introduced into EDACCA, the model puts more weight on the auto-regressive relations among quality samples, and it achieves the best performance among these models. It is noted that unsupervised DiCCA purely learns the auto-regressive relations in quality samples, but its performance is not as good as EDACCA without the filtering effect brought by the process variables. Similarly, only learning the dynamic cross-correlations in sDiCCA does not generate satisfactory outcomes, since quality information is not fully exploited. Consequently, the superior performance of EDACCA is attributed to its enhanced flexibility, enabling it to effectively capture and leverage both the dynamics of cross-correlations and auto-correlations.

   Furthermore, to demonstrate the enhanced computing efficiency of EDACCA in comparison to DACCA, we conducted a comprehensive experiment on the Tennessee Eastman Process (TEP) dataset. For this analysis, we meticulously selected 100 rounds of data, comprising a total of 10,000 samples. Subsequently, these samples were partitioned into 10 groups, each containing an increment of 1,000 samples. Both EDACCA and DACCA were individually trained on these 10 groups of datasets,

and their respective computing times were recorded and summarized in the semi-logarithmic form to facilitate clearer visualization. The model training was performed on a laptop computer equipped with a CPU (Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz) and a GPU (NVIDIA GeForce RTX 2070 with Max-Q Design) using the MATLAB environment. The results conclusively reveal that EDACCA exhibits significantly improved computational efficiency when compared to DACCA. This notable advantage can be attributed to EDACCA's adoption of a more efficient matrix inversion approach, that facilitates handling of ill-conditioned matrices and instability issues that may arise during computation. Notably, EDACCA achieves comparable computing time with DAPLS, a model that does not entail matrix inversions in its training stage. This further underscores the computational efficacy of EDACCA, positioning it as a practical and efficient choice for dynamic relation modeling tasks. While many of these calculations can be performed off-line, changes in industrial processes may require model updates thus making fast computation an attractive property of this algorithm.

5. For the wind process, it is observed that the prediction performance of EDACCA, DACCA, and DAPLS is better than DiPLS and sDiCCA, with EDACCA performing the best, which is consistent with the TEP case study. To further illustrate the prediction effectiveness of EDACCA, the EDACCA-based MS prediction performance is compared with DiPLS, DiCCA [52], random forest regression (RFR) [86] and long short-term memory network (LSTM) [78], with different time steps. Their parameters are selected with a cross-validation method to achieve their optimal performance, and the MS prediction results of these models are summarized. Broadly speaking, it is observed that as the time step h increases, the predictive uncertainties escalate, and this is reflected in the MSEs across all the considered methods. Nevertheless, EDACCA method consistently exhibits superior predictive performance in comparison to the alternative models. This pronounced supremacy of the EDACCA-based multi-steps-ahead model corroborates its robust capacity for forward predictions. Such proficiency implies that the EDACCA approach excels in capturing intricate temporal dynamics and provides more accurate forecasts for multiple time steps into the future when contrasted with the benchmark models under examination.

   Two distinct types of missing data have been intentionally introduced into the wind power dataset for experimental purposes: Missing Completely At Random (MCAR) and Missing Not At Random (MNAR). MCAR is to mimic the general scenario for missing data, and MNAR is to simulate the case when different frequencies are set for process and quality variables.

   In the context of the Missing Completely At Random (MCAR) condition, specific proportions of data samples (1%, 5%, 10%, 20%, 30%, and 40%) are deliberately removed at random from the training dataset. The evaluation of missing data imputation methods involves a comparative analysis of five distinct techniques. These methods include: EDACCA-based imputation, generative adversarial imputation nets (GAIN) [189], linear regression (LR) imputation, duplication (Dup) and deduplication (DeDup) method. Across all examined scenarios, the EDACCA-based impu-

tation strategy consistently demonstrates superior performance, yielding the smallest Mean Squared Error (MSE) when compared to the alternative imputation methods. Remarkably, even in the presence of a progressive increase in the missing rate, ranging from 1% to 40%, the imputation performance of the EDACCA-based approach remains consistently stable and robust. This finding corroborates the remarkable efficacy and resilience of the EDACCA-based imputation technique in handling missing data, irrespective of its amount relative to the available data. The observed consistency in performance highlights the method's capacity to effectively capture and leverage the underlying contextual correlations, thereby mitigating the impact of missing values.

In order to replicate the Missing Not at Random (MNAR) phenomenon arising from variations in sampling rates of quality and process variables, the gathered quality data are retained at regular intervals referred to as $\mathcal{S}$ steps, where $\mathcal{S}$ represents the number of sampling steps. For the specific case of the wind dataset, the chosen steps are required to be divisors of 24. Consequently, we select the values 2, 3, 4, 6, and 12 as appropriate intervals. Subsequently, four distinct imputation techniques, namely EDACCA, LR, Dup, and DeDup, are employed to handle the MNAR scenario present in the dataset. As the magnitude of the step increases and the proportion of missing data grows more pronounced, the efficacy of the EDACCA-based imputation method becomes increasingly apparent, as evidenced by its ability to yield smaller Mean Squared Errors (MSEs). These observations lead to the conclusion that, particularly in scenarios where the rate of missing data is substantial, the utilization of the EDACCA imputation approach results in more precise and accurate estimations, thereby facilitating effective compensation for the absent data.

7.1.3 Development of Interpretability for Time Series

The reported dynamic extensions of multivariate statistical methods currently lack a robust and interpretable framework for explaining dynamic relationships among the slow features. Additionally, these extensions face the challenge of potential rank deficiency when resolving the dynamic model. In response to these critical issues, this study introduces a novel approach known as "Exponentially Weighted Slow Feature Analysis" (EWSFA). This method is specifically designed to address the aforementioned limitations by explicitly extracting multi-order dynamics in modern manufacturing processes.

The findings of this part of the research can be summarized as follows:

1. EWSFA minimizes the difference between the LV of the current sample and the predicted LV formed by the weighted combination of past samples. Different from dynamic inner LVMs, the exponentially weighted moving average [71] is used to weight past samples, which is in line with the decaying nature of time series for increasingly older observations. The efficiency of EWSFA is further improved with the aid of full-rank decomposition. Denote the whole training data as $\mathbf{X} \in \mathcal{R}^{n \times m} =$

$[\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n)]^\top$, with n samples and each one containing m variables. To ensure that slowly varying signals are kept in the extracted LVs, in addition to the invariance between neighboring samples $s(t)$ and $s(t-1)$, it is also important to guarantee maximum similarity among $s(t)$ and other past series data $\{s(t-1), s(t-2), \ldots, s(t-d)\}$, where d is the dynamics order of the collected samples. The subscript j for j^{th} LV is omitted for ease of representation. Instead of assigning random or equal weights to $\{s(t-1), s(t-2), \ldots, s(t-d)\}$ as in existing dynamic SFA methods, it is reasonable to assume that more recent samples contribute more to the *slowness*, and thus the idea of exponentially weighted moving average (EWMA) is adjusted and adapted in EWSFA. To reduce the computation cost of EWSFA and address the potential rank-deficient issue, an economic form of the singular value decomposition (SVD) is adopted to improve the computational efficiency [48, 118]. Different from conventional SVD, economic SVD removes zeros from the diagonal matrix composed of singular values and their corresponding singular vectors.

2. Based on the nature of EWSFA, a monitoring scheme is designed to detect faulty samples and dynamics anomalies. The complete set of LVs $\mathbf{S} \in \mathbb{R}^{n \times m}$ can be divided into the extracted slow features $\mathbf{S}_s \in \mathbb{R}^{n \times l}$ and the fast features $\mathbf{S}_f \in \mathbb{R}^{n \times (m-l)}$, where $\mathbf{S} = [\mathbf{S}_s, \mathbf{S}_f]$. In our work, the number of slow features l is determined based on the *slowness* of the extracted features, which is defined as $\mathcal{S} \equiv \langle \dot{\mathbf{s}}_{d+1}^2 \rangle_t = \langle (\mathbf{s}_{d+1} - \hat{\mathbf{s}}_{d+1})^2 \rangle_t$. It is noted that based on the proposed optimization goal in Eqs. (6.12) - (6.15), the range of $\mathcal{S}$ is $[0, 4]$: $\mathcal{S}$ is large ($\mathcal{S} \geq 2$) when $\mathbf{s}_{d+1}$ and $\hat{\mathbf{s}}_{d+1}$ is negatively correlated or not correlated, while it is smaller than 2 when the *slowness* is kept between them. In this work, to ensure the invariance between $\mathbf{s}_{d+1}$ and $\hat{\mathbf{s}}_{d+1}$, a threshold of 0.5 is selected; that is, the slowness of the extracted l slow features in $\mathbf{S}_s$ is smaller than 0.5, while the features in $\mathbf{S}_f$ have *slowness* values larger than the threshold.

3. The present study centers around an in-depth examination of the monitoring performance of the proposed EWSFA. To demonstrate the efficacy of the monitoring scheme based on EWSFA, the TEP and Three-Phase Flow Facility are selected as case studies. These chosen processes serve as illustrative examples to showcase the effectiveness and efficiency of the EWSFA-based monitoring approach in practical industrial settings.

   Based on these two case studies, there are several conclusions: (1) The R^2 of the first several SFs of EWSFA is higher than DiSFA. Generally, the first several SFs contain the most important information about the process. Thus, the predictability of SFs extracted by EWSFA is not weakened and more useful information can be extracted by EWSFA. (2) One can deduce that EWSFA is better able to capture dynamic slow features from the process with greater accuracy compared to DiSFA. This is due to the utilization of EWMA and the assumption that more recent samples have a greater impact on the *slowness* in EWSFA, which aligns well with the characteristics of the real industrial process. (3) It can be inferred that the EWSFA's dynamic monitoring performance is more precise and responsive than that of DiSFA and DiPCA, particularly when the system reaches a new steady state. This is because the dynamic features extracted by EWSFA encompass more significant information about

the process than those extracted by DiSFA and DiPCA.

7.1.4 Suitability of Each Method

To facilitate the selection of appropriate methodologies tailored to specific processes, this section will undertake a comprehensive evaluation of the applicability of each method developed within this thesis.

DWPLS has been introduced as a solution designed to address non-linearity and the presence of redundant information among adjacent samples. Consequently, DWPLS proves particularly well-suited for industrial processes characterized by heightened non-linearity and the presence of redundant information due to factors such as high sampling rates and gradual continuous slow process changes.

Turning our attention to DAPLS and EDACCA, the core premise of these methods centers around the optimal utilization of historical quality data. Thus, their relevance is heightened in scenarios where access to such historical quality data is readily available within industrial processes. Furthermore, DAPLS excels in monitoring industrial processes, while EDACCA emerges as the preferred choice for applications involving the imputation of missing data.

As for EWSFA, it boasts a distinctive dynamic structure well-suited for the analysis of time series data. Its particular strength lies in its capacity to extract gradual variation features from a given process. Consequently, EWSFA is most aptly applied in industrial settings marked by the presence of oscillatory signals. Through its noise-filtering capabilities, EWSFA excels in capturing and elucidating these oscillatory.

7.2 Future Work

As described in previous chapters, DWPLS was proposed to address the redundancy between adjacent samples, and dynamic auto-regressive latent variables methods (DALVM and EDACCA) were proposed to fully exploit process and quality information, which have demonstrated their effectiveness in both numerical simulations and simulated industrial processes. Following the results of this thesis, the following topics are proposed for future study:

1. The probabilistic forms of many traditional latent variable models were proposed to improve their robustness, and probabilistic latent variable models (PLVM) not only provide an explicit explanation of model structure but also retain the ability to draw new samples from a given probability distribution. Traditional PLVM introduced a mixture structure to handle more complex data structures, but the information contained in the past samples is not exploited in these models, and thus dynamics are ignored. To address this issue, past information will be incorporated into PLVM resulting in a dynamic algorithm to be referred to as DPLVM. Further, DPLVM

with a deterministic term (DPLVM-DT) could be developed to improve its modeling accuracy for unsupervised predictive performance.

2. As described in Chapter 2, supervised learning algorithms are widely used when quality data are difficult to obtain in a timely manner. In the outer model of traditional PLS, process and quality latent variables are calculated by maximizing their covariance, and then its inner model is used to capture the linear regression relation between process and quality latent variables. In contrast, probabilistic PLS does not have an inner model, and a part of the process latent variables needs to be used to construct the relations between process and quality latent variables. In future work, in order to consider dynamic variations, the dynamic probabilistic PLS model can be designed to address these issues.

3. Compared with traditional LVMs, neural networks obtain good and robust performance for non-linear processes, whose model structure, however, is not as explicit as for traditional LVMs. In recent years, integrating neural networks and LVMs is a popular research field. Andrew et al. [11] proposed deep canonical correlation analysis (deep CCA), where both $\mathbf{X}$ and $\mathbf{Y}$ are performed with multiple stacked layers of nonlinear transformation, and then regular CCA is performed on pre-treated $\mathbf{X}$ and $\mathbf{Y}$. It has been recognized that deep CCA can capture higher correlation values between $\mathbf{X}$ and $\mathbf{Y}$ and some work has been reported.

   However, the reported methods on this subject are just a direct combination of LVMs and neural networks, and $\mathbf{X}$ and $\mathbf{Y}$ have been compressed twice which inevitably increases the information loss. To address this issue, in future work, we would like to effectively integrate the neural network models into the objectives of static and dynamic LVMs. For static and dynamic PLS, input and output are projected to latent variable spaces where their covariance is maximized. Considering that the objective of AEs is to capture reduced-dimension and main variance of the input data, the data compression step can be replaced by AEs. This design can provide a nonlinear description of the data and develop an explicit interpretable nonlinear model structure.

4. The aforementioned methods should be applied and tested in real-world industrial processes, where a waste treatment process (WTP) from our industrial partner, Anaergia Inc.[1], will be used in my work. The main goal of the whole project with Anaergia Inc. is to optimize the waste-to-resource plants in Anaergia by developing a set of advanced time series, machine learning, and reinforcement learning algorithms, aimed at improving predictive modeling performance and realizing an automatic parameter tuning and optimization mechanism. In order to achieve it, it is important to develop a good understanding of the processes first. Then, since complex dynamics and nonlinearity are involved in these processes, the developed methods in Chapters 3 to 6 can be applied and adapted to model the WTPs.

[1] https://www.anaergia.com/

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] Edgar Acuna and Caroline Rodriguez. The treatment of missing values and its effect on classifier accuracy. In *Classification, clustering, and data mining applications*, pages 639–647. Springer, 2004.

[3] Piyush Agarwal, Mohammad Aghaee, Melih Tamer, and Hector Budman. A novel unsupervised approach for batch process monitoring using deep learning. *Computers & Chemical Engineering*, 159:107694, 2022.

[4] Piyush Agarwal, Melih Tamer, and Hector Budman. Explainability: Relevance based dynamic deep learning algorithm for fault detection and diagnosis in chemical processes. *Computers & Chemical Engineering*, 154:107467, 2021.

[5] Piyush Agarwal, Melih Tamer, M Hossein Sahraei, and Hector Budman. Deep learning for classification of profit-based operating regions in industrial processes. *Industrial & Engineering Chemistry Research*, 59(6):2378–2395, 2019.

[6] Shotaro Akaho. A kernel method for canonical correlation analysis. *arXiv preprint cs/0609071*, 2006.

[7] Mark Alber, Adrian Buganza Tepole, William R Cannon, Suvranu De, Salvador Dura-Bernal, Krishna Garikipati, George Karniadakis, William W Lytton, Paris Perdikaris, Linda Petzold, et al. Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *NPJ digital medicine*, 2(1):115, 2019.

[8] Haneen Alkabbani, Ali Ahmadian, Qinqin Zhu, and Ali Elkamel. Machine learning and metaheuristic methods for renewable power forecasting: A recent review. *Frontiers in Chemical Engineering*, 3:14, 2021.

[9] Paul D Allison. *Missing data*. Sage publications, 2001.

[10] Shirley Almon. The distributed lag between capital appropriations and expenditures. *Econometrica: Journal of the Econometric Society*, pages 178–196, 1965.

[11] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255. PMLR, 2013.

[12] Rebecca R Andridge and Roderick JA Little. A review of hot deck imputation for survey non-response. *International statistical review*, 78(1):40–64, 2010.

[13] Mariette Awad and Rahul Khanna. Support vector regression. In *Efficient learning machines*, pages 67–80. Springer, 2015.

[14] Francis R Bach and Michael I Jordan. Kernel independent component analysis. *The Journal of Machine Learning Research*, 3:1–48, 2003.

[15] G Baffi, J Morris, and E Martin. Non-linear model based predictive control through dynamic non-linear partial least squares. *Chemical Engineering Research and Design*, 80(1):75–86, 2002.

[16] Ruth E Baker, Jose-Maria Pena, Jayaratnam Jayamohan, and Antoine Jérusalem. Mechanistic models versus machine learning, a fight worth fighting for the biological community? *Biology letters*, 14(5):20170660, 2018.

[17] Karen Bandeen-Roche, Diana L Miglioretti, Scott L Zeger, and Paul J Rathouz. Latent variable regression for multiple discrete outcomes. *Journal of the American Statistical Association*, 92(440):1375–1386, 1997.

[18] Christian F Beckmann and Stephen M Smith. Probabilistic independent component analysis for functional magnetic resonance imaging. *IEEE transactions on medical imaging*, 23(2):137–152, 2004.

[19] Tobias Blaschke, Pietro Berkes, and Laurenz Wiskott. What is the relation between slow feature analysis and independent component analysis? *Neural computation*, 18(10):2495–2508, 2006.

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

[21] Magnus Borga, Tomas Landelius, and Hans Knutsson. A unified approach to PCA, PLS, MLR and CCA. 1997.

[22] George EP Box et al. Some theorems on quadratic forms applied in the study of analysis of variance problems, i. effect of inequality of variance in the one-way classification. *The annals of mathematical statistics*, 25(2):290–302, 1954.

[23] B. Braun, I. Castillo, M. Joswiak, R.R. Y. Peng, A. Schmidt, Z. Wang, L. Chiang, and B. Colegrove. Data science challenges in chemical manufacturing. *IFAC world congress proceedings*, 2020.

[24] David R Brillinger. *Time series: data analysis and theory*. SIAM, 2001.

[25] David S Broomhead and Gregory P King. Extracting qualitative dynamics from experimental data. *Physica D: Nonlinear Phenomena*, 20(2-3):217–236, 1986.

[26] DS Broomhead and D Lowe. Multivariable functional interpolation and adaptive networks. *Complex systems*, 2:321–355, 1988.

[27] Jiangzhong Cao, Pei Chen, Qingyun Dai, and Wing-Kuen Ling. Local information-based fast approximate spectral clustering. *Pattern Recognition Letters*, 38:63–69, 2014.

[28] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168, 2006.

[29] Daniel T Chang. Probabilistic deep learning with probabilistic neural networks and deep probabilistic models. *arXiv preprint arXiv:2106.00120*, 2021.

[30] Junghui Chen and Kun-Chih Liu. On-line batch process monitoring using dynamic pca and dynamic pls models. *Chemical Engineering Science*, 57(1):63–75, 2002.

[31] Tao Chen, Elaine Martin, and Gary Montague. Robust probabilistic PCA with missing data and contribution analysis for outlier detection. *Computational Statistics & Data Analysis*, 53(10):3706–3716, 2009.

[32] Zhiwen Chen, Steven X Ding, Kai Zhang, Zhebin Li, and Zhikun Hu. Canonical correlation analysis-based fault detection methods with application to alumina evaporation process. *Control Engineering Practice*, 46:51–58, 2016.

[33] Zhiwen Chen, Ketian Liang, Steven X Ding, Chao Yang, Tao Peng, and Xiaofeng Yuan. A comparative study of deep neural network-aided canonical correlation analysis-based process monitoring and fault detection methods. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[34] Debo Cheng, Shichao Zhang, Zhenyun Deng, Yonghua Zhu, and Ming Zong. kNN algorithm with data-driven k value. In *International Conference on Advanced Data Mining and Applications*, pages 499–512. Springer, 2014.

[35] Leo H Chiang, Richard D Braatz, and Evan L Russell. *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2001.

[36] 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.

[37] Sang Wook Choi and In-Beum Lee. Nonlinear dynamic process monitoring based on dynamic kernel pca. *Chemical engineering science*, 59(24):5897–5908, 2004.

[38] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3642–3649. IEEE, 2012.

[39] Brandon Corbett and Prashant Mhaskar. Data-driven modeling and quality control of variable duration batch processes with discrete inputs. *Industrial & Engineering Chemistry Research*, 56(24):6962–6980, 2017.

[40] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. Supervised learning. In *Machine learning techniques for multimedia*, pages 21–49. Springer, 2008.

[41] Bhupinder S Dayal and John F MacGregor. Improved PLS algorithms. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 11(1):73–85, 1997.

[42] 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.

[43] Sijmen De Jong. SIMPLS: an alternative approach to partial least squares regression. *Chemometrics and intelligent laboratory systems*, 18(3):251–263, 1993.

[44] Matthew Dixon, Diego Klabjan, and Jin Hoon Bang. Classification-based financial markets prediction using deep neural networks. *Algorithmic Finance*, 6(3-4):67–77, 2017.

[45] Young Do Koo, Hye Seon Jo, Man Gyun Na, Kwae Hwan Yoo, and Chang-Hwoi Kim. Prediction of the internal states of a nuclear power plant containment in LOCAs using rule-dropout deep fuzzy neural networks. *Annals of Nuclear Energy*, 156:108180, 2021.

[46] A Rogier T Donders, Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091, 2006.

[47] Yining Dong. *Dynamic Latent Structured Data Analytics*. PhD thesis, University of Southern California, 2016.

[48] Yining Dong, Yingxiang Liu, and S Joe Qin. Efficient dynamic latent variable analysis for high-dimensional time series data. *IEEE Transactions on Industrial Informatics*, 16(6):4068–4076, 2019.

[49] Yining Dong and S Joe Qin. Dynamic-inner canonical correlation and causality analysis for high dimensional time series data. *IFAC-PapersOnLine*, 51(18):476–481, 2018.

[50] Yining Dong and S Joe Qin. A novel dynamic PCA algorithm for dynamic data modeling and process monitoring. *Journal of Process Control*, 67:1–11, 2018.

[51] Yining Dong and S Joe Qin. Regression on dynamic PLS structures for supervised learning of dynamic data. *Journal of Process Control*, 68:64–72, 2018.

[52] Yining Dong and S Joe Qin. New dynamic predictive monitoring schemes based on dynamic latent variable models. *Industrial & Engineering Chemistry Research*, 59(6):2353–2365, 2020.

[53] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[54] James J Downs and Ernest F Vogel. A plant-wide industrial process control problem. *Computers & chemical engineering*, 17(3):245–255, 1993.

[55] Harshit Dubey and Vikram Pudi. Class based weighted k-nearest neighbor over imbalance dataset. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 305–316. Springer, 2013.

[56] Sandra Esteves and Desmond Devlin. Food waste chemical analysis. *WRAP final report. Trefforest, Wales, UK: The Wales Centre of Excellence for Anaerobic Digestion at the University of Glamorgan*, 2010.

[57] Liangjun Feng, Chunhui Zhao, and Biao Huang. A slow independent component analysis algorithm for time series feature extraction with the concurrent consideration of high-order statistic and slowness. *Journal of Process Control*, 84:1–12, 2019.

[58] NM Fletcher, AJ Morris, G Montague, and EB Martin. Local dynamic partial least squares approaches for the modelling of batch processes. *The Canadian Journal of Chemical Engineering*, 86(5):960–970, 2008.

[59] Bengt Fornberg and Julia Zuev. The runge phenomenon and spatially variable shape parameters in rbf interpolation. *Computers & Mathematics with Applications*, 54(3):379–398, 2007.

[60] Ildiko E Frank. A nonlinear PLS model. *Chemometrics and intelligent laboratory systems*, 8(2):109–119, 1990.

[61] Zhiqiang Ge. Review on data-driven modeling and monitoring for plant-wide industrial processes. *Chemometrics and Intelligent Laboratory Systems*, 171:16–25, 2017.

[62] Zhiqiang Ge, Furong Gao, and Zhihuan Song. Batch process monitoring based on support vector data description method. *Journal of Process Control*, 21(6):949–959, 2011.

[63] Paul Geladi and Bruce R Kowalski. Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17, 1986.

[64] Lovedeep Gondara and Ke Wang. Multiple imputation using deep denoising autoencoders. *arXiv preprint arXiv:1705.02737*, 280, 2017.

[65] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[66] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Overview of supervised learning. In *The elements of statistical learning*, pages 9–41. Springer, 2009.

[67] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[68] Harold Hotelling. Relations between two sets of variates. *Biometrika*, pages 321–377, 1936.

[69] Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics*, pages 162–190. Springer, 1992.

[70] Bin Hu, Zhao Zhao, and Jun Liang. Multi-loop nonlinear internal model controller design under nonlinear dynamic PLS framework using ARX-neural network model. *Journal of Process Control*, 22(1):207–217, 2012.

[71] J Stuart Hunter. The exponentially weighted moving average. *Journal of quality technology*, 18(4):203–210, 1986.

[72] Kiyosi Itō. *Encyclopedic dictionary of mathematics*, volume 1. MIT press, 1993.

[73] Alan Julian Izenman. Reduced-rank regression for the multivariate linear model. *Journal of multivariate analysis*, 5(2):248–264, 1975.

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

[75] J. Edward Jackson and Govind S. Mudholkar. Control procedures for residuals associated with principal component analysis. *Technometrics*, 21(3):341–349, 1979.

[76] J Edward Jackson and Govind S Mudholkar. Control procedures for residuals associated with principal component analysis. *Technometrics*, 21(3):341–349, 1979.

[77] Ines Jaffel, Okba Taouali, Mohamed Faouzi Harkat, and Hassani Messaoud. Moving window kpca with reduced complexity for nonlinear dynamic process monitoring. *ISA transactions*, 64:184–192, 2016.

[78] I-Feng Kao, Yanlai Zhou, Li-Chiu Chang, and Fi-John Chang. Exploring a long short-term memory based encoder-decoder framework for multi-step-ahead flood forecasting. *Journal of Hydrology*, 583:124631, 2020.

[79] Michael H Kaspar and W Harmon Ray. Dynamic PLS modelling for process control. *Chemical Engineering Science*, 48(20):3447–3461, 1993.

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

[81] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[82] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[83] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[84] Wenfu Ku, Robert H Storer, and Christos Georgakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and intelligent laboratory systems*, 30(1):179–196, 1995.

[85] Miroslav Kubat. Neural networks: a comprehensive foundation by simon haykin, macmillan, 1994, isbn 0-02-352781-7., 1999.

[86] Ali Lahouar and J Ben Hadj Slama. Hour-ahead wind power forecast based on random forests. *Renewable energy*, 109:529–541, 2017.

[87] Kamakshi Lakshminarayan, Steven A Harp, and Tariq Samad. Imputation of missing data in industrial databases. *Applied intelligence*, 11(3):259–275, 1999.

[88] S Lakshminarayanan, Sirish L Shah, and K Nandakumar. Modeling and control of multivariable processes: Dynamic PLS approach. *AIChE Journal*, 43(9):2307–2322, 1997.

[89] John Law. Principal component analysis, 1987.

[90] Dac-Nhuong Le, Velmurugan Subbiah Parvathy, Deepak Gupta, Ashish Khanna, Joel JPC Rodrigues, and K Shankar. IoT enabled depthwise separable convolution neural network with deep support vector machine for COVID-19 diagnosis and classification. *International journal of machine learning and cybernetics*, pages 1–14, 2021.

[91] Lei Le, Andrew Patterson, and Martha White. Supervised autoencoders: Improving generalization performance with unsupervised regularizers. *Advances in neural information processing systems*, 31:107–117, 2018.

[92] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[93] Kimmo Leppäkoski and Harri Aaltonen. An implementation of compensation of interactions in a deltav automation system.

[94] Gang Li, Baosheng Liu, S Joe Qin, and Donghua Zhou. Quality relevant data-driven modeling and monitoring of multivariate dynamic processes: The dynamic t-pls approach. *IEEE transactions on neural networks*, 22(12):2262–2271, 2011.

[95] Gang Li, S Joe Qin, and Donghua Zhou. Geometric properties of partial least squares for process monitoring. *Automatica*, 46(1):204–210, 2010.

[96] Gang Li, S Joe Qin, and Donghua Zhou. A new method of dynamic latent-variable modeling for process monitoring. *IEEE Transactions on Industrial Electronics*, 61(11):6438–6445, 2014.

[97] Hongbin Liu, Jie Yang, Yuchen Zhang, and Chong Yang. Monitoring of wastewater treatment processes using dynamic concurrent kernel partial least squares. *Process Safety and Environmental Protection*, 147:274–282, 2021.

[98] Jie Liu, Valeria Vitelli, Enrico Zio, and Redouane Seraoui. A novel dynamic-weighted probabilistic support vector regression-based ensemble for prognostics of time series data. *IEEE Transactions on Reliability*, 64(4):1203–1213, 2015.

[99] Qiang Liu, S Joe Qin, and Tianyou Chai. Multiblock concurrent PLS for decentralized monitoring of continuous annealing processes. *IEEE Transactions on Industrial Electronics*, 61(11):6429–6437, 2014.

[100] Qiang Liu, Qinqin Zhu, S Joe Qin, and Tianyou Chai. Dynamic concurrent kernel cca for strip-thickness relevant fault diagnosis of continuous annealing processes. *Journal of Process Control*, 67:12–22, 2018.

[101] Yongjie Liu, Kun Ding, Jingwei Zhang, Yuanliang Li, Zenan Yang, Wenming Zheng, and Xiang Chen. Fault diagnosis approach for photovoltaic array based on the stacked auto-encoder and clustering with iv curves. *Energy Conversion and Management*, 245:114603, 2021.

[102] André Lucas and Xin Zhang. Score-driven exponentially weighted moving averages and value-at-risk forecasting. *International Journal of Forecasting*, 32(2):293–302, 2016.

[103] Xin Ma, Yabin Si, Zeyi Yuan, Yihao Qin, and Youqing Wang. Multistep dynamic slow feature analysis for industrial process monitoring. *IEEE Transactions on Instrumentation and Measurement*, 69(12):9535–9548, 2020.

[104] John F MacGregor, Christiane Jaeckle, Costas Kiparissides, and M Koutoudi. Process monitoring and diagnosis by multiblock PLS methods. *AIChE Journal*, 40(5):826–838, 1994.

[105] John F MacGregor, Honglu Yu, Salvador García Muñoz, and Jesus Flores-Cerrillo. Data-based latent variable methods for process analysis, monitoring and control. *Computers & chemical engineering*, 29(6):1217–1223, 2005.

[106] Azad M Madni, Carla C Madni, and Scott D Lucero. Leveraging digital twin technology in model-based systems engineering. *Systems*, 7(1):7, 2019.

[107] Miguel Molina-Solana, María Ros, M Dolores Ruiz, Juan Gómez-Romero, and María J Martín-Bautista. Data science for building energy management: A review. *Renewable and Sustainable Energy Reviews*, 70:598–609, 2017.

[108] Dimitris Mourtzis, Ekaterini Vlachou, and NJPC Milas. Industrial big data as a result of iot adoption in manufacturing. *Procedia cirp*, 55:290–295, 2016.

[109] Antoine Negiz and Ali Çlinar. Statistical monitoring of multivariable dynamic processes with state-space models. *AIChE Journal*, 43(8):2002–2020, 1997.

[110] Andrew Ng. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

[111] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[112] Paul Nomikos and John F MacGregor. Multivariate SPC charts for monitoring batch processes. *Technometrics*, 37(1):41–59, 1995.

[113] Pabara-Ebiere Patricia Odiowei and Yi Cao. Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations. *IEEE Transactions on Industrial Informatics*, 6(1):36–45, 2010.

[114] Yu C Pan, S Joe Qin, Phi Nguyen, and Michael Barham. Hybrid inferential modeling for vapor pressure of hydrocarbon mixtures in oil production. *Industrial & Engineering Chemistry Research*, 52(35):12420–12425, 2013.

[115] Daniel Peña, Ezequiel Smucler, and Victor J Yohai. Forecasting multiple time series with one-sided dynamic principal components. *Journal of the American Statistical Association*, 2019.

[116] Daniel Peña and Victor J Yohai. Generalized dynamic principal components. *Journal of the American Statistical Association*, 111(515):1121–1131, 2016.

[117] Kaixiang Peng, Kai Zhang, and Gang Li. Quality-related process monitoring based on total kernel PLS model and its industrial application. *Mathematical Problems in Engineering*, 2013, 2013.

[118] Thilo Penzl. Algorithms for model reduction of large dynamical systems. *Linear algebra and its applications*, 415(2-3):322–343, 2006.

[119] António Pesqueira, Maria José Sousa, and Álvaro Rocha. Big data skills sustainable development in healthcare and pharmaceuticals. *Journal of Medical Systems*, 44:1–15, 2020.

[120] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[121] S Joe Qin. Survey on data-driven industrial process monitoring and diagnosis. *Annual reviews in control*, 36(2):220–234, 2012.

[122] S Joe Qin and Thomas A Badgwell. An overview of industrial model predictive control technology. In *AIche symposium series*, volume 93, pages 232–256. New York, NY: American Institute of Chemical Engineers, 1971-c2002., 1997.

[123] S Joe Qin and Thomas A Badgwell. An overview of nonlinear model predictive control applications. In *Nonlinear model predictive control*, pages 369–392. Springer, 2000.

[124] S Joe Qin, Gregory Cherry, Richard Good, Jin Wang, and Christopher A Harrison. Control and monitoring of semiconductor manufacturing processes: Challenges and opportunities. *IFAC Proceedings Volumes*, 37(9):125–136, 2004.

[125] S Joe Qin and Leo H Chiang. Advances and opportunities in machine learning for process data analytics. *Computers & Chemical Engineering*, 126:465–473, 2019.

[126] S Joe Qin and Yining Dong. Data distillation, analytics, and machine learning. *Proceedings of the 2017 CPC/FOCAPO*, 2017.

[127] S Joe Qin, Yining Dong, Qinqin Zhu, Jin Wang, and Qiang Liu. Bridging systems theory and data science: A unifying review of dynamic latent variable analytics and process monitoring. *Annual Reviews in Control*, 2020.

[128] S Joe Qin, Yingxiang Liu, and Yining Dong. Plant-wide troubleshooting and diagnosis using dynamic embedded latent feature analysis. *Computers & Chemical Engineering*, page 107392, 2021.

[129] S Joe Qin and Thomas J McAvoy. Nonlinear PLS modeling using neural networks. *Computers & Chemical Engineering*, 16(4):379–391, 1992.

[130] S Joe Qin and Thomas John McAvoy. *Neural net PLS approach to dynamic modeling: Method and application*. 1991.

[131] S Joe Qin and TJ McAvoy. Nonlinear FIR modeling via a neural net PLS approach. *Computers & chemical engineering*, 20(2):147–159, 1996.

[132] S Joe Qin and Yingying Zheng. Quality-relevant and process-relevant fault monitoring with concurrent projection to latent structures. *AIChE Journal*, 59(2):496–504, 2013.

[133] SJ Qin and TJ McAvoy. A data-based process modeling approach and its applications. In *Dynamics and Control of Chemical Reactors, Distillation Columns and Batch Processes*, pages 93–98. Elsevier, 1993.

[134] Yan Qin and Chunhui Zhao. Comprehensive process decomposition for closed-loop process monitoring with quality-relevant slow feature analysis. *Journal of Process Control*, 77:141–154, 2019.

[135] G Reinsel and R Velu. Multivariate reduced-rank regression (lecture notes in statistcs no. 136), 1998.

[136] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[137] N Lawrence Ricker. The use of biased least-squares estimators for parameters in discrete-time pulse-response models. *Industrial & engineering chemistry research*, 27(2):343–350, 1988.

[138] Cory A Rieth, Ben D Amsel, Randy Tran, and Maia B Cook. Issues and advances in anomaly detection evaluation for joint human-automated systems. In *International Conference on Applied Human Factors and Ergonomics*, pages 52–63. Springer, 2017.

[139] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Icml*, 2011.

[140] Donald B Rubin and Dorothy T Thayer. EM algorithms for ML factor analysis. *Psychometrika*, 47(1):69–76, 1982.

[141] 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.

[142] Evan L Russell, Leo H Chiang, and Richard D Braatz. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemometrics and intelligent laboratory systems*, 51(1):81–93, 2000.

[143] Hussein Saafan and Qinqin Zhu. Comprehensive monitoring with incremental slow feature analysis. In *2021 American Control Conference (ACC)*, pages 917–922. IEEE, 2021.

[144] Farah Saeed, Muhammad Attique Khan, Muhammad Sharif, Mamta Mittal, Lalit Mohan Goyal, and Sudipta Roy. Deep neural network features fusion and selection based on PLS regression with an application for crops diseases classification. *Applied Soft Computing*, 103:107164, 2021.

[145] Michael Schluse, Marc Priggemeyer, Linus Atorf, and Juergen Rossmann. Experimentable digital twins—streamlining simulation-based systems engineering for industry 4.0. *IEEE Transactions on industrial informatics*, 14(4):1722–1731, 2018.

[146] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[147] 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.

[148] 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.

[149] Ning Sheng, Qiang Liu, S Joe Qin, and Tianyou Chai. Comprehensive monitoring of nonlinear processes based on concurrent kernel projection to latent structures. *IEEE Transactions on Automation Science and Engineering*, 13(2):1129–1137, 2016.

[150] Alissa Sherry and Robin K Henson. Conducting and interpreting canonical correlation analysis in personality research: A user-friendly primer. *Journal of Personality Assessment*, 84(1):37–48, 2005.

[151] Yan Shi, Ying Ge, Jie Chang, Hongbo Shao, and Yuli Tang. Garden waste biomass for renewable and sustainable energy production in china: potential, challenges and development. *Renewable and Sustainable Energy Reviews*, 22:432–437, 2013.

[152] Yang Song, Jian Huang, Ding Zhou, Hongyuan Zha, and C Lee Giles. Iknn: Informative k-nearest neighbor pattern classification. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 248–264. Springer, 2007.

[153] Yu Song, Sheng Yang, Chao Cheng, and Pu Xie. A novel fault detection method for running gear systems based on dynamic inner slow feature analysis. *IEEE Access*, 8:211371–211379, 2020.

[154] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[155] James H Stock and Mark W Watson. Forecasting using principal components from a large number of predictors. *Journal of the American statistical association*, 97(460):1167–1179, 2002.

[156] Liang Sun, Shuiwang Ji, Shipeng Yu, and Jieping Ye. On the equivalence between canonical correlation analysis and orthonormalized partial least squares. In *Twenty-First International Joint Conference on Artificial Intelligence*. Citeseer, 2009.

[157] Lin Sun, Kui Jia, Tsung-Han Chan, Yuqiang Fang, Gang Wang, and Shuicheng Yan. DL-SFA: Deeply-learned slow feature analysis for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2632, 2014.

[158] Bruce Thompson. *Canonical correlation analysis: Uses and interpretation*. Number 47. Sage, 1984.

[159] Nina F Thornhill, John W Cox, and Michael A Paulonis. Diagnosis of plant-wide oscillation through data-driven analysis and process understanding. *Control Engineering Practice*, 11(12):1481–1490, 2003.

[160] TinaManghnani and T. Thirumaran. *Computational CBGSA – SVM Model for Network Based Intrusion Detection System.* Springer, Singapore, 2019.

[161] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[162] V. N. Vapnik. *The Nature of Statistical Learning Theory.* The nature of statistical learning theory, 1995.

[163] Robert Vautard and Michael Ghil. Singular spectrum analysis in nonlinear dynamics, with applications to paleoclimatic time series. *Physica D: Nonlinear Phenomena*, 35(3):395–424, 1989.

[164] Venkat Venkatasubramanian. The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE Journal*, 65(2):466–478, 2019.

[165] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[166] Yongjian Wang, Hongguang Li, Jingwen Huang, Chong Su, Bo Yang, and Chu Qi. An improved bar-shaped sliding window cnn tailored to industrial process historical data with applications in chemical operational optimizations. *Industrial & Engineering Chemistry Research*, 58(47):21219–21232, 2019.

[167] Zhen Wang, Chunyu Wang, Chao Gao, Xuelong Li, and Xianghua Li. An evolutionary autoencoder for dynamic community detection. *Science China Information Sciences*, 63(11):1–16, 2020.

[168] Bryan C Weare and John S Nasstrom. Examples of extended empirical orthogonal function analyses. *Monthly Weather Review*, 110(6):481–485, 1982.

[169] Greg Welch, Gary Bishop, et al. An introduction to the Kalman filter. 1995.

[170] 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.

[171] Peter M Williams. Bayesian regularization and pruning using a laplace prior. *Neural computation*, 7(1):117–143, 1995.

[172] Peter R Winters. Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342, 1960.

[173] Barry M Wise and Neal B Gallagher. The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, 6(6):329–348, 1996.

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

[175] Herman Wold. Soft modelling by latent variables: the non-linear iterative partial least squares (NIPALS) approach. *Journal of Applied Probability*, 12(S1):117–142, 1975.

[176] Herman Ole Andreas Wold. *Nonlinear estimation by iterative least square procedures*. 1968.

[177] S. Wold, A. Ruhe, H. Wold, and W. J. Dunn, Iii. The collinearity problem in linear regression. the partial least squares (PLS) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3):735–743, 1984.

[178] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[179] Svante Wold, Nouna Kettaneh-Wold, and Bert Skagerberg. Nonlinear PLS modeling. *Chemometrics and Intelligent Laboratory Systems*, 7(1):53–65, 1989.

[180] Svante Wold, Arnold Ruhe, Herman Wold, and WJ Dunn, Iii. The collinearity problem in linear regression. the partial least squares (PLS) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3):735–743, 1984.

[181] 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, 2013.

[182] Hao Wu and Jinsong Zhao. Deep convolutional neural network model based chemical process fault diagnosis. *Computers & chemical engineering*, 115:185–197, 2018.

[183] Bo Xu and Qinqin Zhu. Online quality-relevant monitoring with dynamic weighted partial least squares. *Industrial & Engineering Chemistry Research*, 2020.

[184] Yuemei Xu, Mingxing Jia, Zhizhong Mao, and Hanqi Li. A novel kernel dynamic inner slow feature analysis method for dynamic nonlinear process concurrent monitoring of operating point deviations and process dynamics anomalies. *Journal of Process Control*, 110:59–75, 2022.

[185] Xu, Bo and Zhu, Qinqin. Concurrent auto-regressive latent variable model for dynamic anomaly detection. *Journal of Process Control*, 108:1–11, 2021.

[186] Jian Yang, Zheng Lv, Hongbo Shi, and Shuai Tan. Performance monitoring method based on balanced partial least square and statistics pattern analysis. *ISA transactions*, 81:121–131, 2018.

[187] Shen Yin, Steven X Ding, Xiaochen Xie, and Hao Luo. A review on basic data-driven approaches for industrial process monitoring. *IEEE Transactions on Industrial Electronics*, 61(11):6418–6428, 2014.

[188] Jaehyun Yoo and H Jin Kim. Target localization in wireless sensor networks using online semi-supervised support vector regression. *Sensors*, 15(6):12539–12559, 2015.

[189] Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *International Conference on Machine Learning*, pages 5689–5698. PMLR, 2018.

[190] Jiaxin Yu, Lingjian Ye, Le Zhou, Zeyu Yang, Feifan Shen, and Zhihuan Song. Dynamic process monitoring based on variational bayesian canonical variate analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[191] Shipeng Yu, Kai Yu, Volker Tresp, Hans-Peter Kriegel, and Mingrui Wu. Supervised probabilistic principal component analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 464–473, 2006.

[192] Wanke Yu and Chunhui Zhao. Broad convolutional neural network based industrial process fault diagnosis with incremental learning capability. *IEEE Transactions on Industrial Electronics*, 67(6):5081–5091, 2019.

[193] Tao Yuan and S Joe Qin. Root cause diagnosis of plant-wide oscillations using granger causality. *Journal of Process Control*, 24(2):450–459, 2014.

[194] H Henry Yue and S Joe Qin. Reconstruction-based fault identification using a combined index. *Industrial & engineering chemistry research*, 40(20):4403–4414, 2001.

[195] Zakiah Mohd Yusoff, Zuraida Muhammad, Mohd Hezri Fazalul Rahiman, and Mohd Nasir Taib. ARX modeling for down-flowing steam distillation system. In *2012 IEEE 8th International Colloquium on Signal Processing and its Applications*, pages 485–490, 2012.

[196] L. Zhang, X Zheng, Q. Pang, and W. Zhou. Fast gaussian kernel support vector machine recursive feature elimination algorithm. *Applied Intelligence*, (6), 2021.

[197] M. L. Zhang and Z. H. Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

[198] Ming-Qing Zhang and Xiong-Lin Luo. Modified canonical variate analysis based on dynamic kernel decomposition for dynamic nonlinear process quality monitoring. *ISA transactions*, 108:106–120, 2021.

[199] Quanshi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *arXiv preprint arXiv:1802.00614*, 2018.

[200] Shumei Zhang, Chunhui Zhao, and Biao Huang. Simultaneous static and dynamic analysis for fine-scale identification of process operation statuses. *IEEE Transactions on Industrial Informatics*, 15(9):5320–5329, 2019.

[201] Y. Zhang, Z. X. Chi, and L. I. Ke-Qiu. A novel SVM based on vague-sigmoid kernel. *Journal of Chinese Computer Systems*, 2008.

[202] Yingwei Zhang, Hong Zhou, and S Joe Qin. Decentralized fault diagnosis of large-scale processes using multiblock kernel principal component analysis. *Acta Automatica Sinica*, 36(4):593–597, 2010.

[203] Yingwei Zhang, Hong Zhou, S Joe Qin, and Tianyou Chai. Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares. *Industrial Informatics, IEEE Transactions on*, 6(1):3–10, 2010.

[204] Zhang Zhang and Dacheng Tao. Slow feature analysis for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 34(3):436–450, 2012.

[205] Zhongheng Zhang. Missing data imputation: focusing on single imputation. *Annals of translational medicine*, 4(1), 2016.

[206] 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.

[207] Chunhui Zhao and He Sun. Dynamic distributed monitoring strategy for large-scale nonstationary processes subject to frequently varying conditions under closed-loop control. *IEEE Transactions on Industrial Electronics*, 66(6):4749–4758, 2018.

[208] Chunhui Zhao and Youxian Sun. Multispace total projection to latent structures and its application to online process monitoring. *IEEE Transactions on Control Systems Technology*, 22(3):868–883, 2013.

[209] Jiale Zheng, Chunhui Zhao, and Furong Gao. Retrospective comparison of several typical linear dynamic latent variable models for industrial process monitoring. *Computers & Chemical Engineering*, 157:107587, 2022.

[210] Junhua Zheng and Zhihuan Song. Mixture modeling for industrial soft sensor application based on semi-supervised probabilistic PLS. *Journal of Process Control*, 84:46–55, 2019.

[211] Junhua Zheng, Zhihuan Song, and Zhiqiang Ge. Probabilistic learning of partial least squares regression model: Theory and industrial applications. *Chemometrics and Intelligent Laboratory Systems*, 158:80–90, 2016.

[212] W. Zheng, Y. Cheng, SK Oh†, and Z. Fu. Multi-radial basis function SVM classifier: Design and analysis. *Journal of Electrical Engineering & Technology*, 13, 2018.

[213] D. X. Zhou and K. Jetter. Approximation with polynomial kernels and SVM classifiers. *Advances in Computational Mathematics*, 25(1-3):323–344, 2006.

[214] Donghua Zhou, Gang Li, and S Joe Qin. Total projection to latent structures for process monitoring. *AIChE Journal*, 56(1):168–178, 2010.

[215] Kaile Zhou, Chao Fu, and Shanlin Yang. Big data driven smart energy management: From big data to big insights. *Renewable and sustainable energy reviews*, 56:215–225, 2016.

[216] Q. Zhu. Latent variable regression for supervised modeling and monitoring. *IEEE/CAA Journal of Automatica Sinica*, 7(3):800–811, 2020.

[217] Qinqin Zhu. Auto-regressive modeling with dynamic weighted canonical correlation analysis. *Journal of Process Control*, 95:32–44, 2020.

[218] Qinqin Zhu. Latent variable regression for supervised modeling and monitoring. *IEEE/CAA Journal of Automatica Sinica*, 7(3):800–811, 2020.

[219] Qinqin Zhu. Dynamic autoregressive partial least squares for supervised modeling. *IFAC-PapersOnLine*, 54(7):234–239, 2021.

[220] Qinqin Zhu, Qiang Liu, and S Joe Qin. Concurrent canonical correlation analysis modeling for quality-relevant monitoring. *IFAC-PapersOnLine*, 49(7):1044–1049, 2016.

[221] Qinqin Zhu, Qiang Liu, and S Joe Qin. Concurrent quality and process monitoring with canonical correlation analysis. *Journal of Process Control*, 60:95–103, 2017.

[222] Qinqin Zhu and S Joe Qin. Latent variable regression for process and quality modeling. In *2019 1st International Conference on Industrial Artificial Intelligence (IAI)*, pages 1–6. IEEE, 2019.

[223] Qinqin Zhu and S Joe Qin. Supervised diagnosis of quality and process faults with canonical correlation analysis. *Industrial & Engineering Chemistry Research*, 58(26):11213–11223, 2019.

[224] Qinqin Zhu, S Joe Qin, and Yining Dong. Dynamic latent variable regression for inferential sensor modeling and monitoring. *Computers & Chemical Engineering*, 137:106809, 2020.

[225] Xiaofeng Zhu, Heung-Il Suk, and Dinggang Shen. Multi-modality canonical feature selection for Alzheimer's disease diagnosis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 162–169. Springer, 2014.

[226] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.