Feature Extraction for Soft Sensing and Process Monitoring in Steam Generators

by

Seraphina Jinyeong Kwak

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

Master of Science

in

Process Control

Department of Chemical and Materials Engineering

University of Alberta

Abstract

Data is becoming more valuable as there are still many uncertainties and hidden information that have yet to be discovered. For this reason, the application of data analysis and machine learning in the industry is becoming more popular. For example, SAGD (steam assisted gravity drainage) is a type of oil extraction process where high-pressure steam is used to heat the bitumen underground. Optimizing the steam generation is one of the ways to improve the SAGD process as steam is an important part of the SAGD process. One method that may be used to optimize this process is the feature extraction analysis.

Feature extraction analysis is a method that tries to extract valuable information from a given dataset. Essentially, it projects the given dataset to another subspace such that a particular statistical property is amplified and noises are minimized. In this thesis, data analysis is explored to optimize the SAGD process. The first chapter defines the problem and feature extraction methods are introduced.

In the second chapter, a grey box model is used to develop a soft sensor to predict the steam quality out of a steam generator in the real SAGD process. The core model structure is based on energy balance and data analytic methods is used to further improve the predictability strength by applying a Kalman filter and online bias updating technique. Later on, feature extraction methods are further explored to improve the developed soft sensor. Finally, cointegration analysis (CA), which is a type of feature extraction method, is modified to monitor fouling accumulation inside the steam generator tubes. The difficulty of predicting fouling buildup in a steam generator using process knowledge alone is addressed. Since fouling buildup involves

complex chemical phenomena, a data analysis approach is proposed that can be easily applied in the industry. In the proposed method, PCA is paired with CA to develop a practical solution to predict fouling buildup.

Acknowledgements

First of all, I would like to show my sincere gratitude to my professor, Prof. Biao Huang, for his support and guidance. He was patient while teaching me and encouraging me to explore and expand my knowledge in process data analytics. Without him, I would not be able to be where I am. It truly has been an honor to work with him and the CPC (Computer Process Control) group. The CPC group had so many bright and admirable people. In particular, Yanjun Ma and Rahul Raveendran have helped me tremendously with my research and understand process data analytics from the ground up. They have helped me revising my work, and gave me ideas to explore. I also have not forgotten Mohammad Rashedi, Dr. Hariprasad Kodamana, Dr. Fadi Ibrahim, Agustin Vicente, Lei Fan, Mengqi Fang, Shabnam Sedghi, David Scott, Alireza Kheradmand, Wenhen Shen, Zheyuan Liu, Anahita Sadeghian and many else.

In addition to the support of my academic research, I got tremendous support for my industrial projects from Yamei Liu and Sandeep Sheth. They taught and helped me understand how to link research with industry. Without their support, I would not have been able to understand the importance of research within the industry nor understand how to apply research to industry.

I would also like to take this opportunity to thank my co-workers and friends, who encouraged me throughout my graduate program: Hossein Shahandeh, Anuja Samuel, Joan Javillonar, Uriel Guerrero, Bryan Stock, Scott Guilcher, Eliyya Shukeir, Fei Qi and many else. Last but not least, I would like to express my sincere gratitude to my family and my parents for their selfless love and support. Without them, I

would not be where I currently am.

Also, I would like to acknowledge and thank Natural Sciences and Engineering Research Council of Canada for the financial support of the Industrial Research Chair program.

I could not have completed this thesis if I did not have support from all the kind people around me. Their support and guidance helped me made the journey of completing my thesis.

Contents

*part of this chapter has been submitted for publication as "**Kwak, S.**, Ma, Y., & Huang, B. (2018). Extracting Nonstationary Features for Process Data Analytic and Application in Fouling Detection" in Computers & Chemical Engineering.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

In chemical engineering, predictive models are widely used to predict quality variables as soft sensors. Predictive models can either be black box models, white box models or grey box models. A black box model is a purely data-driven model. It does not require prior knowledge about the process but requires the training dataset to represent the system. On the other hand, a white box model is solely based on process knowledge. Therefore, for the model development of a white box model, the quality of the training dataset is not a major concern. But building predictive models solely based on engineering/process knowledge has limitations. First of all, it does not consider the uncertainties in the system. Second, some chemical processes, such as fouling accumulation, are too complicated to develop a white box model.

With an abundant source of available data, the application of statistical and machine learning methods has become popular in recent years in chemical engineering. For example, a SAGD (steam assisted gravity drainage) process is a type of in-situ bitumen extraction process that uses heat from steam to reduce the viscosity of bitumen to bring it up to surface. The steam that is used for the process is generated by steam generators and the energy used in steam generators is responsible for more than 90% of the total energy requirements in the SAGD process. Therefore, it is desirable to optimize the OTSG (once-through steam generators) operation as much

as possible.

Some typical steam generators used in the SAGD process are OTSG and HRSG (heat recovery steam generators). Two of the many ways to optimize OTSG are: 1. predict steam quality out of the steam generator with high accuracy so that advanced control can be implemented for steam quality, and 2. monitor and predict the fouling accumulation inside the tubes to ensure normal OTSG operations. Steam quality is the mass fraction in the saturated mixture that is vapor. For example, 80% steam quality would mean that 80% is steam and 20% is water in the saturated mixture (by weight). Typically, steam quality is sampled manually and manual samples give the most reliable measurement, but the sampling frequency is not fast enough to be useful for control or optimization. Due to its complex chemical mechanism however, it is very difficult to predict fouling buildup despite the fact fouling/scaling accumulation in steam generators has been studied for many years. Consequently, the goal of this research is to maximize the use of available process data - both known and hidden information - to solve engineering problems in the industry. In this thesis, grey box and black box models will be investigated and developed to predict steam quality and fouling accumulation in the steam generator used in the SAGD process respectively.

1.2 Preliminaries

As this thesis mainly deals with feature extraction analysis and soft sensors, they are briefly introduced in this section.

1.2.1 Feature Extraction Analysis

With the ability to store a large amount of data, data analytics and machine learning have become a popular choice to solve engineering problems. One method of machine learning is feature extraction analysis. In essence, feature extraction methods try to extract informative features that contain valuable information within the data. Through some form of mapping, it extracts a set of new features from the original

dataset. This method is particularly useful when dealing with a multivariate system as it can be used to reduce the dimension of the system, consequently focusing only on the selected features. Also, feature extraction transforms the data such that redundancy within the original dataset is reduced. Below is a brief explanation of some feature extraction methods that are commonly used in chemical engineering.

1.2.2 Brief overview of PCA

PCA (principal component analysis) is one of the most commonly used feature extraction methods in process data analysis. It was first invented by Pearson in 1901 [1] and independently developed by Hotelling in 1933 [2]. In essence, PCA tries to extract features based on variation by applying eigendecomposition. It transforms the data and projects it into a coordinate system (commonly known as the latent space) so the components (also known as scores) are orthogonal to each other and ordered based on variance. Thus the first component will try to find the direction where the variance of the system is maximized:

$$\text{maximize}_{\theta_{11},\ldots,\theta_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \Big(\sum_{j=1}^{p} \theta_{j1} x_{ij} \Big)^2 \right\} \quad \text{subject to } \sum_{j=1}^{p} \theta_{j1}^2 = 1 \tag{1.1}$$

where θ is the parameter (loading vector), x_{ij} is the data point, and the first principal component can be expressed as:

$$z_{il} = \theta_{11} x_{i1} + \theta_{21} x_{i2} + \ldots + \theta_{p1} x_{ip} \quad \text{where i = 1,...,n} \tag{1.2}$$

Equation (1.2) is commonly known as scores or latent variables and they define the latent space. Since PCA tries to find the direction where the variance of the system is maximized, it can consequently be used as a dimension reduction method: it can represent the data in a lower dimension that contains most of the variation. Also, it can be observed that there is no temporal interpretation in conventional PCA: conventional PCA does not consider sequential correlations of data samples. To incorporate temporal information, SFA (slow feature analysis) and CA (cointegration analysis) also have been considered.

1.2.3 Brief overview of SFA

Like PCA, SFA (slow feature analysis) is another feature extraction method that uses eigendecomposition to solve the feature extraction problem. Unlike PCA, SFA focuses on the dynamics and tries to extract slowly varying features and the dimension of the new coordinate system needs to be predefined. Although the original dataset does not need to be linear, SFA assumes that the features extracted are linear.

SFA tries to find the slowly varying feature by minimizing the temporal variation within the latent space:

$$\text{minimize } \mathbb{E}(\dot{z}_i^2) \tag{1.3}$$

Since any constant value would give the optimal solution, equation (1.3) must be satisfied while:

$$\mathbb{E}(z_i) = 0 \tag{1.4}$$

$$\mathbb{E}(z_i^2) = 1 \tag{1.5}$$

$$\mathbb{E}(z_i z_j) = 0 \tag{1.6}$$

where

$$\dot{z}_i(t) = z_i(t) - z_i(t-1) \tag{1.7}$$

Equation 1.6 ensures that obtained slow features are orthogonal to each other, thus decorrelated. By minimizing the temporal variation of the latent score, we try to find the smooth and simple underlying feature from the dataset, which tends to be important in process data analysis.

1.2.4 Brief overview of CA

CA (cointegration analysis) was developed by Engle and Granger [3] in econometrics and Chen et al [4] first introduced it in chemical engineering as statistical process monitoring methods. The main objective of CA is to extract the most stationary feature within a nonstationary dataset so that we can understand the cointegrated

relationship between the variables. Assuming that the dataset can be represented as a VAR (vector autoregressive) model, and then applying first-order difference, the system can be represented as:

$$\begin{aligned} X_t - X_{t-1} = [\Pi_1 - \mathrm{I}][X_{t-1} - X_{t-2}] + [\Pi_1 + \Pi_2 - \mathrm{I}][X_{t-2} - X_{t-3}] \\ + ... + [\Pi_1 + \Pi_2 + .. + \Pi_k - \mathrm{I}]X_{t-k} + e_t \end{aligned} \tag{1.8}$$

$$\Delta X_t = \Gamma_1 \Delta X_{t-1} + \Gamma_2 \Delta X_{t-2} ... + \Gamma_{k-1} \Delta X_{t-k+1} + \Pi X_{t-k} + e_t \tag{1.9}$$

where e_t is is white noise with zero mean and finite variance, $e_t \sim N(0, \Lambda)$ and the parameter Π maps the nonstationary space to the stationary space. Π can be further decomposed into α and β:

$$\Pi = \alpha \beta^T \tag{1.10}$$

where $\beta^T X_t$ becomes stationary even if X_t is nonstationary. Johansen shows how to determine β by applying eigendecomposition [5].

1.2.5 Brief overview of PLS

PLS (partial least squares) is different from the feature extraction methods mentioned above in the sense that PLS is a supervised learning method while the other methods are unsupervised methods. As an unsupervised method, PCA, SFA, and CA do not use any information from the response variable (Y). Consequently, there is no guarantee that the extracted features would best describe the response variables when using unsupervised methods. PLS, on the other hand, maps both X and Y to latent space such that the covariance is maximized.

$$X = TP^T + E \tag{1.11}$$

$$Y = UQ^T + F \tag{1.12}$$

1.2.6 Soft sensor

In the process industry, measurement of key variables, such as steam quality and fouling buildup, is of great importance. The availability of online measurements of

these variables is crucial for control, optimization, and monitoring. However, hardware instruments do not always perform at a satisfactory level. In many cases, manual measurements are not sufficiently frequent enough for advanced process control. Consequently, analyzers are usually used but suffer from reliability issues and poor economic efficiency. As a result, estimating key process variables using inferential sensors (soft sensors) has become a favorable alternative solution for the online measurement of key variables. Soft sensors have shown that they are a valuable alternative to obtain critical values and popular soft sensor modeling techniques have been discussed in [6].

In [6], the authors briefly mentions about the different types of soft sensor models that can be developed, namely the white box (first principle) model, grey box (hybrid) model, and black box (data-driven) model. White box models are based on the chemical and physical properties of the process. As a result, it is heavily dependent on detailed and in-depth process knowledge, which may not be always available. The black box model, on the other hand, is data-based so as long as there is data available, in-depth process knowledge is not required. However, they are more susceptible to the typical properties of process data, such as missing data, drifting phenomenon, outliers, and co-linearity. Manual pre-processing and basic process knowledge is necessary to work around these properties while developing a soft sensor. To mitigate these issues, semi-automated/ recursive soft sensors have been developed. A grey box model is a mixture of both white and black box models and it tries to reduce disadvantages of black box models. The most common application of soft sensors is the online prediction of a quality variable, but they can also be used for process monitoring and fault detection by using multivariate statistics such as PCA (principal component analysis). In addition to PCA, there are other popular multivariate statistical methods that are used to develop data-driven soft sensors, such as PLS (partial least squares), SVM (support vector machines), and ANN (artificial neural networks).

In [7], Shima et al. further discusses soft sensor development but focuses on how Bayesian methods can be incorporated to further improve soft sensors. It also divides different types of soft sensors into a white box, black box, and grey box models, and emphasizing importance of pre-processing the data to deal with missing datasets, outliers and co-linearity. Some of the advantages of Bayesian methods that were mentioned in the paper are the ability to obtain grey box models by incorporating process knowledge in a Bayesian scheme via prior distributions over model parameters, functional forms, and constraints. It can also handle incomplete and non-Gaussian distribution data.

1.3 Thesis Outline

The remainder of this thesis is structured as follows:

Chapter 2 explains the development of the grey-box steam quality soft sensor and evaluates the performance in detail. As this model has been developed for the industry, the main focus was to develop a model that is robust yet accurate enough to predict steam quality. The core model structure was based on energy balance and it was developed to maximize the use of the available process variables while simple enough to be implemented in the DCS. In order to utilize the available data as much as possible, statistical techniques are also implemented to improve predictability. The developed model is then compared with the existing steam quality analyzer to show performance improvement.

In Chapter 3, PCA (principal component analysis), SFA (slow feature analysis) and CA (cointegration analysis) are selected to extract informative features to further improve the developed steam quality soft sensor. The importance of this chapter is to maximize the information and extract the hidden features within the system for model development. The different feature methods were selected to determine which type of feature(s) is most informative. The proposed model can be implemented in any grey-box models. To validate the model, its correlation with the lab samples and

predictability are compared with the grey-box model that was developed in Chapter 2.

Feature extraction methods are further explored in Chapter 4. In this chapter, residual CA is proposed to extract the most nonstationary feature within the system. In chemical engineering, the process is inevitably nonstationary due to equipment aging and accumulation in the system. Consequently, understanding the nonstationary features within the system becomes desirable. Multiple numerical simulations were conducted to validate that the proposed model can indeed extract the most nonstationary feature(s) in the system. To show the importance of this model, residual CA is implemented in the two-layer feature extraction method to predict and monitor fouling accumulation in OTSG tubes, which is a very difficult process to predict due to the complex chemical phenomenon.

Finally, Chapter 5 summarizes the thesis and recommends future research directions.

1.4 Thesis Contributions

contribution of this thesis are as follows:

1. Developed a grey-box steam quality soft sensor to predict steam quality out of once-through steam generators (OTSG) that are used in SAGD processes, where the core model structure was based on energy balance and statistical techniques were used to enhance the predictability.

2. Improved the developed grey-box steam quality soft sensor by applying feature extraction techniques on the bias terms and process variables.

3. Developed a method to extract nonstationary features from process data.

4. Proposed a method to monitor and predict fouling accumulation in OTSG tubes by using a two-layer feature extraction method using PCA and modified CA.

Chapter 2

Developing steam quality soft sensor with an energy balance based model

2.1 Introduction

Steam-assisted gravity drainage (SAGD) is an enhanced oil recovery (EOR) method that uses heat from high-pressure steam to reduce the viscosity of oil sand underground so that it is easier to bringing up to surface. Compared to other EOR methods, SAGD has higher bitumen production rates. The steam used for the SAGD process can be produced by once-through steam generators (OTSG). OTSG uses combustion energy from fuel gas (FG) to change the water (BFW) into wet steam (WS). Fuel gas can be either sweet gas, makeup gas, or a mixture of both. To maximize efficiency, BFW is divided into a certain number of passes. The energy used in OTSGs is responsible for more than 90% of the total energy requirements in the SAGD process [8]. Therefore, it is desirable to optimize the OTSG operation as much as possible. Figure 2.1 is a process flow diagram of a typical OTSG. One of the ways to optimize OTSG operation is by measuring steam quality with high accuracy (measurements can be used to control the steam generator with tight operating limits). Steam quality is the mass fraction in the saturated mixture that is vapor. For example, 80% steam quality would mean that 80% is steam and 20% is water in the saturated mixture (by weight).

Unfortunately, hardware instruments often give not-so-accurate measurements. It is common to sample quality variables in the field (which gives reliable measurements), but they are not sampled frequently enough to be useful for process control. Therefore, inferential sensors (soft sensors) have become a promising alternative solution for measuring key variables.

Inferential sensors can either be based on a black-box model, white box model or grey box model. A black box model is purely data-driven. It does not require prior knowledge about the process but requires the training dataset to be representative of system operations. On the other hand, a white box model is solely based on process knowledge.

However, many industrial processes are complex such that it requires in-depth knowledge to develop a model, and the quality of data is often not good enough to develop accurate black box models. To mitigate these shortfalls, grey box models have been considered as a more practical solution. A grey box model is a mixture of black box and white box models where it tries to amplify advantages and minimize their disadvantages. For example, a simpler model structure can be developed from process knowledge with a few assumptions. With readily available data, parameters for the models can be identified. In addition to parameter identification, statistical techniques can be used to further compensate for the inadequacy of the assumptions and extract hidden information from the process data to capture the essential information.

The majority of the soft sensor development in this chapter will be based on hybrid modeling where the core model structure is based on energy balance. The parameter identification and other soft sensing techniques will be based on process data. Figure 2.1 is a brief process diagram of an OTSG.

Figure 2.2 is a diagram that briefly explains the soft sensing algorithm. Developing a reliable steam quality sensor ultimately leads to enhancing the performance of the OTSG and achieving its optimization. A typical procedure to develop a soft sensor

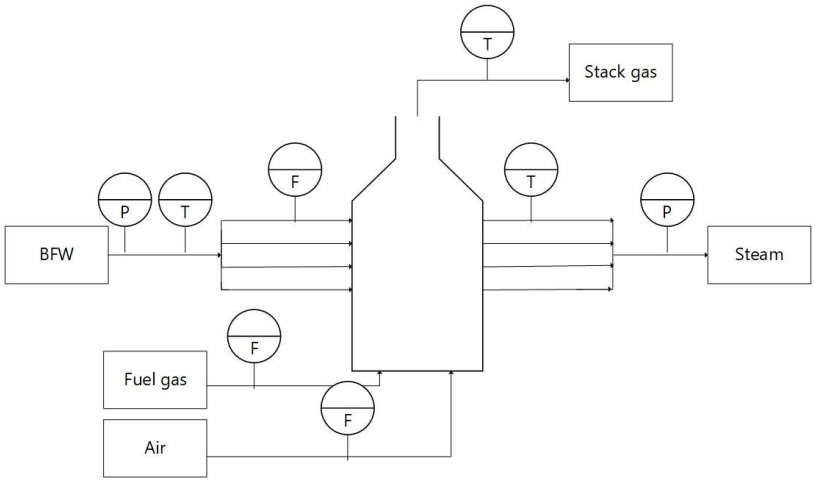

Figure 2.1: OTSG schematic

is as follows.

1. Problem statement: As with many engineering problems, the first step is to identify the problem that needs to be solved and learn about the process. In this chapter, the problem that needs to be solved is to develop a steam quality soft sensor. The process required to be learned is the steam generation process via OTSG.

2. Data collection and validation: After identifying the problem, appropriate data should be collected and validated. For this thesis, a year-long data from six OTSGs with a one-minute sampling interval was obtained. The quality of data is assessed to determine if they could be used for modeling.

3. Model identification: Based on the knowledge about the process, a model structure can be developed. Based on the available data and the knowledge about the steam generation process via OTSG, an energy balance based model is developed.

4. Soft sensing techniques: To strengthen the prediction power, soft sensing techniques like robust index, filtering, and online bias updating are used. A robust index is used to increase the robustness of the soft sensor by indicating poten-

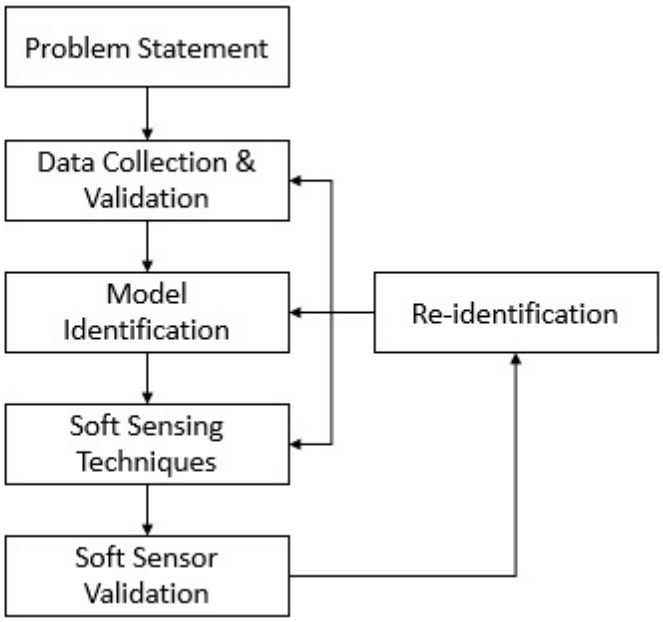

Figure 2.2: Soft Sensing Algorithm

tial problems in advance. Kalman filter is used as a filtering method to reduce the variability and the online bias updating technique is used as an adaptation mechanism.

5. Soft sensor validation: The developed soft sensor needs to be validated to see if all the requirements have been met. The performance should also be validated online to ensure there is no significant discrepancy between offline performance and online performance.

6. Re-identification: During soft sensor validation step, offline and/or online, it is possible to find problems and fail to meet the requirements. In such cases, fine-tuning of the soft sensor is required.

In the remainder of this chapter, section 2.2 explains the previous method. Section 2.3 describes the model structure derived from process knowledge. Section 2.4 describes the soft sensing techniques based on statistical methods. Section 2.5 shows the performance of the developed soft sensor and section 2.6 describes the future work

and some concluding remarks.

2.2 Preliminaries: Existing energy balance steam quality soft sensor

A grey box steam quality soft sensor based on energy balance has been previously developed in [9]. In [9], the authors showed the successful application of the soft sensor in the industry, where online bias update technique and online outlier detection techniques were incorporated to compensate for the error and increase the robustness of the model respectively.

2.2.1 Model development

The model developed in [9] assumes that there is no heat loss between the individual passes and the recombined outlet. By assuming that there is no heat loss, the energy balance at the recombined outlet becomes:

$$Q(t) = Q_a(t) + Q_b(t) \tag{2.1}$$

where

$$Q_a(t) = \rho F_f(t) C_p\big(T_r(t) - T_f(t)\big) \tag{2.2}$$

$$Q_b(t) = \rho F_f \Delta H X(t) \tag{2.3}$$

Similar equation can be developed for individual passes:

$$Q_i(t) = Q_{ai}(t) + Q_{bi}(t) \tag{2.4}$$

where

$$Q_{ai}(t) = \rho F_i(t) C_p\big(T_i(t) - T_f(t)\big) \tag{2.5}$$

$$Q_{bi}(t) = \rho F_i \Delta H X_i(t) \tag{2.6}$$

Taking the energy balance between the individual passes and the recombined outlet, equation (2.1) and (2.4) can be recombined to determine $X_i(t)$:

$$X_i(t) = k_1 u_1(t) + k_2 u_2(t) + k_3 u_3(t) \tag{2.7}$$

where k_1, k_2, k_3 are scaling parameters that will be estimated with the following physical properties:

$$k_1 = \xi_i$$

$$k_2 = \frac{100\xi_i C_p}{\Delta H}$$

$$k_3 = \frac{-100 C_p}{\Delta H}$$

and u_1, u_2, u_3 are process inputs:

$$u_1(t) = \frac{F_f(t)}{F_i(t)} X(t)$$

$$u_2(t) = \frac{F_f(t)}{F_i(t)} [T_r(t) - T_f(t)]$$

$$u_3(t) = [T_i(t) - T_f(t)]$$

The final steam quality soft sensor prediction for the individual passes thus becomes:

$$\hat{Y}_i(t) = X_i(t) + \beta(t) \tag{2.8}$$

where $X_i(t)$ is the model prediction obtained from equation (2.7), and $\beta(t)$ is the online bias term used to offset the model error which has the following formulation:

$$\beta(t) = \alpha[Y_i(t-1) - X_i(t-1)] + (1-\alpha)\beta(t-1) \tag{2.9}$$

where $\alpha \in [0, 1]$ is the weighting factor. In their work, α, k_1, k_2, k_3 have been determined using the prediction error method (PEM). To determine the outliers, Hampel's method was used, which is later discussed in detail in section 2.4.1.

2.3 Model derivation and assumptions

Process knowledge-based models usually start from the laws of conservation. For the proposed application in this thesis, the conservation of thermal energy was used. Since recombined manual steam quality samples were not available however, a different approach from [9] was taken.

To develop a simple model with available process variables, some assumptions are made while developing the model structure. Statistical techniques can be used to compensate for the assumptions made. This section explains how the model structure was made along with its associated assumptions.

2.3.1 Model derivation

A general energy balance equation can be established between the total energy input (Q_{in}) and total energy output (Q_{out}). The energy output can be expressed as the summation of sensible heat ($Q_{out}^{(s)}$) and latent heat ($Q_{out}^{(l)}$). When there is a phase change and assuming no heat loss, the energy consumption of the system can be expressed as:

$$Q_{in} = Q_{out} = Q_{out}^{(s)} + Q_{out}^{(l)}. \tag{2.10}$$

The energy supplied from a furnace or a steam generator can be determined by the combustion energy equation [10].

$$Q_{in} = \dot{m}_{fuel} \cdot LHV \cdot \eta_{combustion} \tag{2.11}$$

With the fuel gas flowrate (F_{FG}) available and assuming that LHV and efficiency stays constant, equation (2.11) could be simplified as a linear function of F_{FG}:

$$Q_{in} = k^{'} \cdot F_{FG} + b^{'} \tag{2.12}$$

where k' and b' are unknown parameters. Assumptions made to develop equation (2.12) will be discussed further in section 2.3.2.

Sensible heat is defined as the energy required to raise the temperature of a system to its boiling point:

$$Q_{out}^{(s)} = \dot{m} \cdot C_p \cdot (T_{out} - T_{in}) \tag{2.13}$$

where

$$\dot{m} = \dot{m} \cdot x + \dot{m}(1 - x) \tag{2.14}$$

and x is the mass fraction of steam and $0 \leq x \leq 1$. To simplify the model structure, the liquid phase of the steam mixture is ignored in the heat balance model and the detailed discussion is provided later. Hence, equation (2.13) can be simplified into

$$Q_{out}^{(s)} \approx \dot{m} \cdot x \cdot C_p \cdot (T_{out} - T_{in}). \tag{2.15}$$

Latent heat is the energy required to change water into steam without changing temperature:

$$Q_{out}^{(l)} = \dot{m} \cdot x \cdot \Delta h \tag{2.16}$$

Combining equations (2.15) and (2.16), the total output energy can be expressed as:

$$Q_{in} = \dot{m} \cdot x \cdot C_p \cdot (T_{out} - T_{in}) + \dot{m} \cdot x \Delta h. \tag{2.17}$$

Combining equation (2.12) and equation (2.17):

$$k^{'} \cdot F_{FG} + b^{'} = \dot{m} \cdot x \cdot C_p \cdot (T_{out} - T_{in}) + \dot{m} \cdot x \cdot \Delta h \tag{2.18}$$

Solving for steam quality, equation (2.18) can be simplified and hence the model structure for steam quality soft sensor becomes:

$$\hat{X}_M = \frac{k' \cdot F_{FG} + b'}{\dot{m} f_{C_p}(T_{in}, Pin) \cdot (T_{out} - T_{in}) + f_{\Delta h}(T_{out})} \tag{2.19}$$

where f_{C_p}, and $f_{\Delta h}(T_{out})$ are known functions, and F_{FG}, $\dot{m}$, T_{in} , P_{in} and T_{out} are measurements available from the hardware sensors. In order to compensate for the error from unmeasurable variables and simplified assumptions, the following equation is proposed as the soft sensor model:

$$\hat{X}_M = K \cdot \frac{F_{FG}}{\dot{m} f_{C_p}(T_{in}, P_{in}) \cdot (T_{out} - T_{in}) + f_{\Delta h}(T_{out})} + b \tag{2.20}$$

where K is the gain parameter and b is the bias term. Bias can be offline estimated and online updated. The estimation of parameters is required in the model as compensation of errors since many assumptions were made. To compensate for the assumptions, K and b are determined from data and these parameters will attempt to

capture the factors that were not included in the model such as efficiency ($\eta_{combustion}$) and heating value (LHV). Latent heat is the energy required to change water into steam without changing temperature:

2.3.2 Assumptions

Since the model is based on energy balance (knowledge-based), it is usually more versatile than data-driven based models. However, in the developed model above, the following assumptions had to be made to maintain a simple model structure and compensate for missing information.

Assumption 1: Representation of the energy input

The amount of heat required to change water into the saturated mixture is provided by the combustion energy of fuel gas and air. To calculate the actual amount of heat produced from fuel gas, however, its exact physical properties must be known. If the fuel gas is a purchased sweet gas, obtaining its physical properties should not be so difficult. The problem arises when the fuel gas contains makeup gas where its physical properties are difficult to obtain. To develop a simple model, it has been assumed that LHV and steam generator efficiency stays constant. The flow rate of fuel is used to represent the energy input and the regression parameters are used to compensate for the inadequacy of these assumptions.

Assumption 2: Ignoring of the liquid phase

It can be observed in equation (2.19) that the liquid phase of the saturated mixture has been ignored. The steam quality used in SAGD operations typically ranges from 70 to 100 % [11]. In a volumetric fraction, if the steam quality is at 70%, it is mostly steam. Therefore, while developing the model, it was assumed that process variables measuring the saturated mixture represent saturated steam. Consequently, the proposed method is limited to high quality saturated mixtures.

Assumption 3: Using inlet temperature to calculate heat capacity

Specific heat capacity at a constant pressure (C_p) is defined as the heat required to raise the temperature of a substance by one degree. Mathematically, it can be expressed as

$$C_p = (\frac{dH}{dT})_p \tag{2.21}$$

where H represents heat and T is temperature. Even when there is a temperature variation in the system, C_p is often used as a constant term, where the average temperature is used to calculate its value, since it is considered that the variation due to changing temperature is usually insignificant [12]. To reduce the variation of the prediction as much as possible, various ways of calculation have been studied.

Figure 2.3 shows that the change in C_p with respect to the temperature is a quadratic function and its variance increases as temperature increases. Therefore, simply taking the average temperature to calculate this value is questionable. Most importantly, although the difference was quite small, the performance of the EB model prediction was best when the inlet temperature was used to determine C_p. For these reasons, inlet temperature will be used to determine C_p. Table 2.1 summarizes the correlation coefficients of the steam quality in pass 1 from six different OTSGs.

Table 2.1: Correlation Coefficient between the predicted steam quality and its reference value

	OTSG A	OTSG B	OTSG C	OTSG D	OTSG E	OTSG F
Inlet	0.84	0.77	0.62	0.70	0.54	0.77
Averaged	0.85	0.76	0.61	0.69	0.54	0.76
Outlet	0.84	0.73	0.57	0.62	0.53	0.71

2.4 Soft sensing techniques

In order to increase the robustness and accuracy of the model, the developed soft sensor uses the following statistical techniques for further processing:

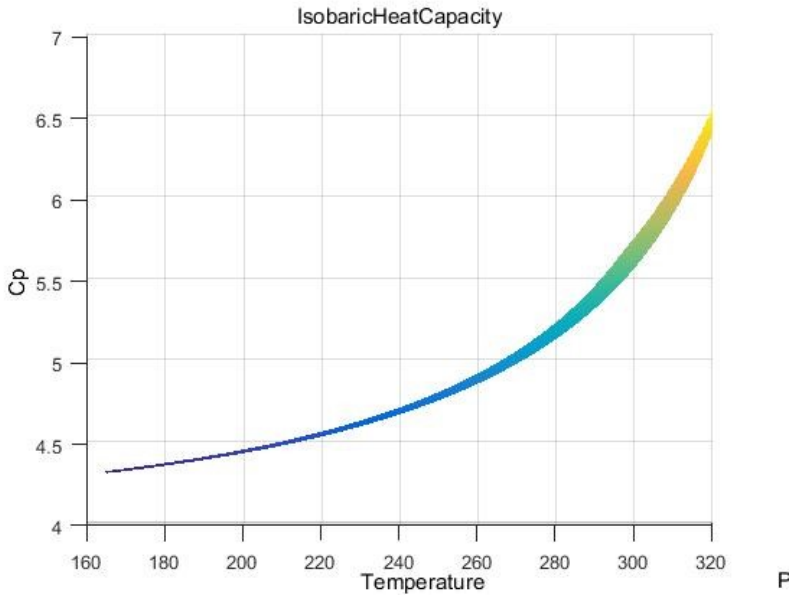

Figure 2.3: Specific heat capacity at constant pressure vs. Temperature

1. A robust layer to protect soft sensor prediction to check the reliability of the process measurements.

2. A filtering layer to reduce variance in the soft sensor prediction.

3. An online bias updating method to remove bias in soft sensor prediction.

2.4.1 Robust layer

Outliers from the process measurements can significantly impact soft sensor prediction. For this reason, robust layers to detect outliers online should be part of the soft sensor algorithm. To calculate, Hampel's method was used:

1. First bound: $Med(X) \pm 3 \cdot MAD(X)$

2. Second bound: $Med(X) \pm 6 \cdot MAD(X)$

where Med(X) is the median value of a given process measurement, and MAD(X) is the median absolute deviation, which is an indicator of measurement variance:

$$MAD(X) = Med(|X - Med(X)|) \tag{2.22}$$

Compared to the 3σ method, which is another common outlier detection method that determines the outliers by comparing data with 3 standard deviations, Hampel's method is more robust to outliers [9]. Figure 2.4 shows an example of the robust layers calculated by using Hampel's method, where y-axis has been removed for proprietary reason. To calculate the upper layer, the high operating region was used. Conversely, the low operating region was used to determine the lower layer. With these four layers, each variable can be classified as normal (N), mild outlier (M) or severe outlier (S). Based on this classification, the robust index can be assigned. Table 2.2 summarizes how the roust index is assigned for given process measurement.

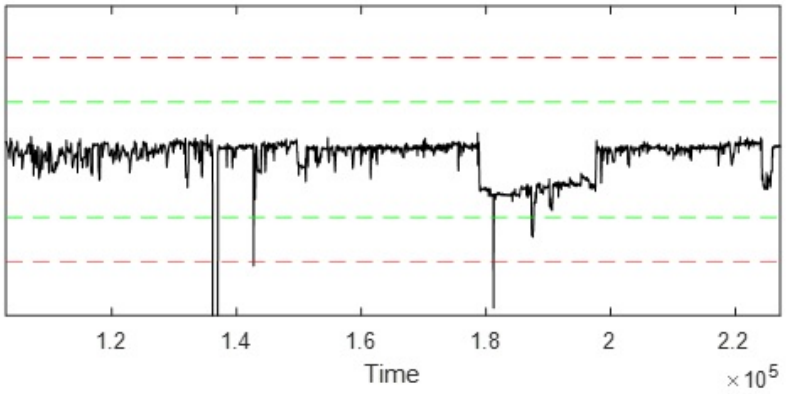

Figure 2.4: Robust boundary for a process variable

Table 2.2: Models used to generate features

Robust Index (RI)	Determination	Description
2	Number of severe outliers ≥ 1	Soft sensor has stopped working
1	Number of mild outliers ≥ 2	The prediction may not be accurate
0	Otherwise	The prediction is in good condition

2.4.2 Correction factor

Using the robust index, the soft sensor can distinguish if the process variables are reliable enough to be used for the calculations. After having a model prediction by using equation (2.20), a state estimation method is used to reduce the variance of

the prediction and online bias update technique is used to mitigate for the drifting phenomena in the process.

State Estimation

State estimation is a useful technique that applies to systems subject to disturbances and can be used for filtering and smoothing. In the proposed application of steam quality prediction, a simplified Kalman filter is applied. Comparing to the fixed weight filters such as low-pass filters [13] or wavelet filters [14, 15], the Kalman filter can adjust the weights automatically for each state sample.

Kalman filter can also be used to predict new values in the state space system [16]. The original model for a dynamic system is as follows [17]:

$$\begin{cases} S(t) = A \cdot S(t-1) + w(t), & w(t) \sim N(0, \sigma_p^2), \\ X(t) = H \cdot S(t) + v(t), & v(t) \sim N(0, \sigma_o^2), \end{cases} \tag{2.23}$$

where $S(t)$ is the state at time t and $X(t)$ is the observation at time t. Based on the linear model considered in equation (2.23), the current state $S(t)$ can be estimated from the previous state and the current observation sample. More importantly, the uncertainties of this model are described in a probabilistic way, where Normal distribution is assumed for the noise terms, $w(t)$ and $v(t)$. Thus, the state value can be solved optimally through this probabilistic model.

To apply Kalman filter to steam quality prediction, model output ($\hat{X}_M$ from equation (2.20)) will be considered as an observation. Soft sensor output (steam quality) can be an indicator of the steam generation process. Therefore, it should show dynamic behavior and can be described by using the state $S(t)$ in the first equation of model (2.23). The simplification used in this application is that the parameters, A and H, will be set as one. The justification of this application is as follows.

1. Based on the prior knowledge, the steam quality is controlled with a predetermined setpoint and the process has strong inertia. This means that the

current steam quality value should be near the steam quality value at the previous time step. Consequently, A will be set as one.

2. The model prediction is calculated based on observed process variables. Since the model prediction is based on first principle knowledge, model output value should also be close to the actual steam quality value. Therefore, H = 1 is based on the intuitive interpretation.

It can be observed that only the variance parameters σ_p and σ_o in the Normal distributions in equation (2.23) are left for identification. In practice, cross-validation methods can be used to identify the optimal values of these two variance parameters from the collected process data.

Bias Update

After the prediction is smoothed through Kalman filter, online bias updating is used to further improve the prediction quality. Chemical processes often contain drifting bias. Some examples of the causes of the drifts are fouling and equipment aging. To capture this phenomenon, an online bias updating technique will be used:

$$\hat{X}_{ss}(t) = \hat{X}_k(t) + \beta(t) \tag{2.24}$$

where $\hat{X}_k(t)$ is the model output after the filter. The $\beta(t)$ term is the bias term that is updated online based on the manual field measurements. As mentioned in section 2.1, quality variables are often sampled in the field. Although the manual field measurements are not sampled frequently enough to be used for control purposes, they represent the actual value of steam quality most accurately. For this reason, manual field readings will be used for bias updating. To control the speed of bias updating, weight factor, $\alpha \in [0, 1]$, will be used:

$$\beta(t+1) = \beta(t) + \alpha(X_{ref}(t) - \hat{X}_M(t)) \tag{2.25}$$

During normal operations (RI=0), small weight factor will be used to give more weight to the model output. Conversely, during the initialization of the soft sensor

or transient stage, a higher weight factor will be used. By having different weight factors, it is possible to avoid abrupt changes in the soft sensor prediction during normal operation that may be misleading but still allow quick capture during start-ups or plant upsets. Based on the training data, $\alpha = 0.1$ was used during normal operations and $\alpha = 0.7$ for initialization. To further control bias updating, the value of $\beta(t+1)$ is restricted to a maximum value of 1 during normal operation and 10 during initialization and when the manual field measurement is within its designated range. Although manual field measurement represents steam quality most accurately, it still contains noise and not all of its measurements are trustable. The additional restrictions on bias updating technique allow a good balance between the prediction made by the model structure and manual field readings.

2.5 Performance analysis

After all parts of the soft sensing algorithms have been considered, a statistical analysis must be carried out to evaluate the performance of the proposed soft sensors before online implementation. Two evaluation indices, Mean Absolute Error (MAE) and standard deviation are calculated for the proposed and the existing soft sensors. In addition to the statistical summary, visualization will be used to illustrate the features of soft sensing algorithms.

Figure 2.5 compares the steam quality prediction with the existing steam quality analyzer and figure 2.6 compares the variance. From figure 2.5, it can be observed that the proposed method captures the steam quality better with smaller variance. From figure 2.6, it is evident that the proposed method improved the ability to capture steam qualities.

2.5.1 Mean absolute error (MAE) comparison

Table 2.3 summarizes offline MAE comparison for pass one of six OTSGs. To ensure a fair comparison, the following conditions are required for the manual sampled steam

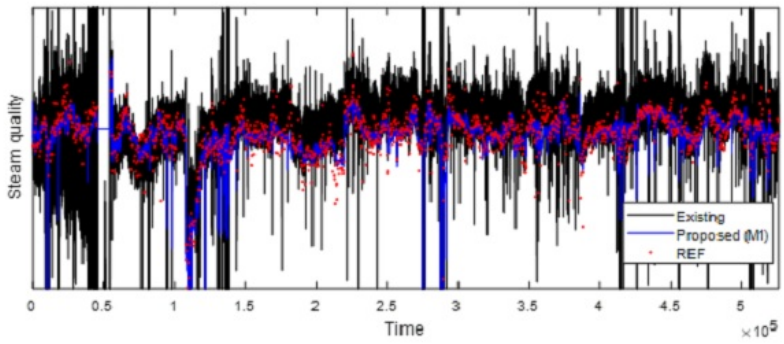

Figure 2.5: Time trend of steam quality prediction at OTSG B, pass 1

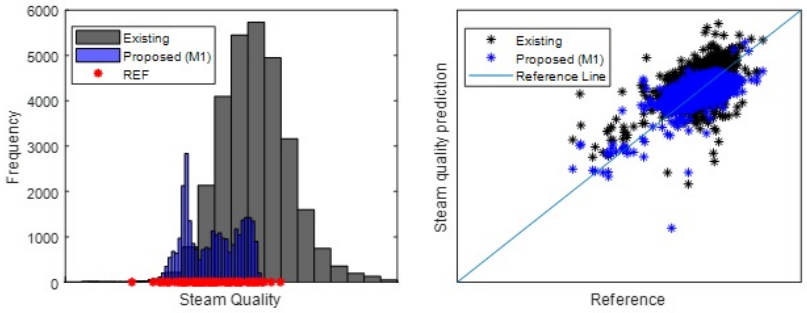

Figure 2.6: Variance comparison (left) and correlation comparison (right) at OTSG B, pass 1

quality to mitigate possible error while collecting the samples:

1. Manual field reading is a valid number (i.e., ignore NaN, ignore steam quality measurements that were recorded during turnaround period).

2. Manual field reading ranges from 60% to 90%.

3. Validation dataset is used.

Table 2.3: MAE Comparison

	Existing	EBSS	% reduction
OTSG A	1.01	0.59	42%
OTSG B	0.70	0.56	20%
OTSG C	0.67	0.56	16%
OTSG D	0.67	0.53	22%
OTSG E	1.16	0.59	49%
OTSG F	1.25	0.56	55%

2.5.2 Standard deviation comparison

During steady-state operations, where the steam quality is within a small variation range, the prediction should not vary too widely. 3σ represents the magnitude of variability of predictions. Table 2.4 summarizes 3σ for all six OTSGs at pass 1. The ranges were chosen to ensure the process is as steady as possible. Since each OTSGs are considered as independent systems, the time frame where it is most steady may be different for different OTSGs. Consequently, different data range was used for different OTSGs.

Table 2.4: 3σ Comparison

	Existing	EBSS	% reduction
OTSG A	2.93	1.13	61%
OTSG B	1.34	1.49	-11%
OTSG C	1.33	0.85	36%
OTSG D	1.64	0.73	55%
OTSG E	1.55	0.88	43%
OTSG F	1.35	0.96	29%

From 2.3 and 2.4, it can be observed that the proposed method generally improved the performance, except for OTSG B. Therefore, it can be concluded that the performance of the developed model is promising and can be used to further improve OTSG optimization. However, seeing the poorer in performance for OTSG B indicates that there is room for improvement.

2.5.3 Limitation of the Proposed Method

Since EB soft sensor is based on first principle methods, they are generally more versatile than data-driven models. However, due to the assumptions made while developing the models, it may have the following limitations:

1. Outlet stream has to be mostly steam: One of the assumptions made while developing the soft sensors was that the liquid phase of the mixture is unmeasurable. For this assumption to be valid, the outlet stream must be mostly

steam. As a result, the proposed soft sensors may lose performance in predicting the steam quality when the outlet of the OTSGs might have a significant amount of liquid. For this reason, the current mitigation of this downfall is to give more weight to an online bias updating technique using manual samples.

2. State estimation parameters should be re-tuned for faster rate: The state estimation parameters were trained based on one-minute data. Consequently, if the proposed soft sensors are to be implemented for faster sampling rate data, the parameters should be retrained to ensure performance.

3. Bias adjustment is required for a startup: During startup, it is common for process measurements to show large bias. To capture such changes introduced in the process during startup, bias should be updated more quickly during process startups. Currently, the proposed soft sensors are designed so that bias update has higher weight during startups.

4. Efficiency and LHV value stay constant: The model structure is from an energy balance and its development has made multiple assumptions. Two regression parameters were obtained from data to compensate for the inadequacy of the assumptions, but they may not be sufficient to capture all the dynamics that happen in the steam generator. For the linear regression parameters to have better compensation effect, efficiency and LHV values should stay constant. One way to improve on this deficiency is to extract additional information from the readily available dataset and add it as inputs.

2.6 Conclusion

In this chapter, the energy balance based soft sensor was developed to predict the steam quality of the steam generated by OTSG. The model structure was based on the energy balance but statistical techniques such as regression, robust index,

state estimation and online bias update were used to compensate for inadequate assumptions, making it as a hybrid (grey box) model. A linear model structure was chosen for its practicality and simplicity, but it may not be sufficient to capture all dynamics within the system and introduces room for improvement for the model.

Compared to an existing steam quality analyzer, the proposed energy balance soft sensor improved MAE by 34% on average and variance by 36%, However OTSG B showed a decline in performance when comparing the variance. Therefore, it can be concluded that although the proposed soft sensors can be used to predict steam quality, improvements can be made to further enhance the developed soft sensor and further optimize OTSG control. In the next chapter, we will consider such improvement through better bias update mechanism.

Chapter 3

Applying Feature Extraction Method on the Bias Estimation

3.1 Introduction

Previously, an energy balance steam quality soft sensor was developed to determine the steam quality from a once-through steam generator (OTSG). It was a grey model where the model structure was based on energy balance, model parameter was obtained from the training dataset and statistical techniques were used to enhance the quality of the prediction. To develop a simple and practical model, several assumptions were made such as constant efficiency and low heating value (LHV).

Although the developed model showed performance improvements compared to the existing soft sensor, it still had room for improvement. For example, while validating the soft sensor, it was observed that the online bias update trends showed similar characteristics for the different passes. Also, several assumptions were made while developing the hybrid model, there may exist valuable hidden information in the process variables that could be further extracted and used to improve the performance of the soft sensor. Therefore, the objective of this chapter is to develop a method that extracts information from the bias or the process variable to further improve the energy balance based soft sensor.

In section 3.2, the motivation of the chapter will be discussed. In section 3.3, the proposed method to extract information from the bias will be explained and the

results will be discussed in section 3.8. Conclusion will be provided in section 3.9.

3.2 Motivation

Motivations of the proposed method are as follows:

1. Attempt to develop a model that is not dependent on online bias updating technique.

2. Maximize information extracted from the available process variables.

Section 3.2.1 and 3.2.2 will explain both motivations in detail.

3.2.1 Attempt to develop model that is not dependent on online bias updating technique

Although the real values of the output can be determined by manual sampling, the frequency of sampling can vary and typically requires manual work. For example, one operating company may sample twice a shift whereas another may only sample once a month. Therefore, it is ideal to develop a model that is only dependent on input variables. Even if sampling is done twice a day, it is still not frequent enough for control and optimization. Therefore, it will be attempted to extract process information from the bias term and map it back such that the final model is only dependent on process variables, and also to develop a model that maximizes the information extracted from the available inputs.

3.2.2 Maximize information from the available process variables

Due to the changes in the process such as adjustments in the feed quality or fouling in the equipment, industrial processes exhibit some form of time-variant behavior. Such time-variant behaviors cause estimates for the soft sensor to deviate from actual ones over time and give biased results. To capture such behavior, a bias updating term

has been incorporated in the soft sensor design. In practice, the bias term in the soft sensor gets updated to reduce the error between the predictive model and the manual samples. In equation (3.1), $\hat{X}_M(t)$ is the model output after applying the filtering method to reduce the variance, and the bias term gets updated only when there is a manual sample available at time t. Consequently, the bias term is time-dependent.

$$\hat{X}_{ss}(t) = \hat{X}_M(t) + \beta(t) \tag{3.1}$$

$$\beta(t) = \begin{cases} (1-\alpha)\beta(t) + \alpha(X_{ref} - \hat{X}_M(t)), & \text{if } X_{ref} \text{ available} \\ \beta(t), & \text{otherwise} \end{cases} \tag{3.2}$$

The bias correction method is commonly used in soft sensing techniques to further improve the soft sensor prediction and various methods for online bias updating techniques have been investigated in many papers. For example Mu et al [18] and Ni et al [19] uses weighting factors between 0 and 1 for the online bias updating, where the weighting factor was determined by trial and error. Xie et al [20] used prediction error method to determine the optimal parameters for the weighting factors. Shardt and Huang discuss how the online bias update term is used to compensate for the slow or drifting disturbance. In their work, they also discuss how to tune the soft sensor's bias update term when there is measurement delay, multirate sampling, or both in both open loop case and closed loop case [21, 22].

In the previous chapter, the bias term β was updated by using the manual sample data to capture the deviations. The rate of the update was controlled by a tuning parameter, α. The bias term gets updated when the manual samples are available. For the energy balance based steam quality soft sensor that was developed in chapter 1, α is relatively small ($\alpha = 0.2$) during the normal operating mode. When at transient state such as start-ups, $\alpha = 0.7$ so that we can capture the unsteady state much faster. Since the bias term only gets updated when there is a manual sample available at time t, it is also dependent on the sampling frequency of the manual

samples. For example, the soft sensor when there is a manual sample available every six hours will be more dependent on the bias term than the soft sensor when there is a manual sample available every 30 days.

While validating the energy balance model, it was observed that the bias term for the different passes shares a similar characteristic, in particular for passes 1, 2, 3, 5 and 6. Therefore, this may indicate that the bias update is to to correct for the missing information from the process. In other words, the bias terms may contain information about the system. Figure 3.1 shows the time trend of the bias term from the different passes using an energy balance steam quality soft sensor.

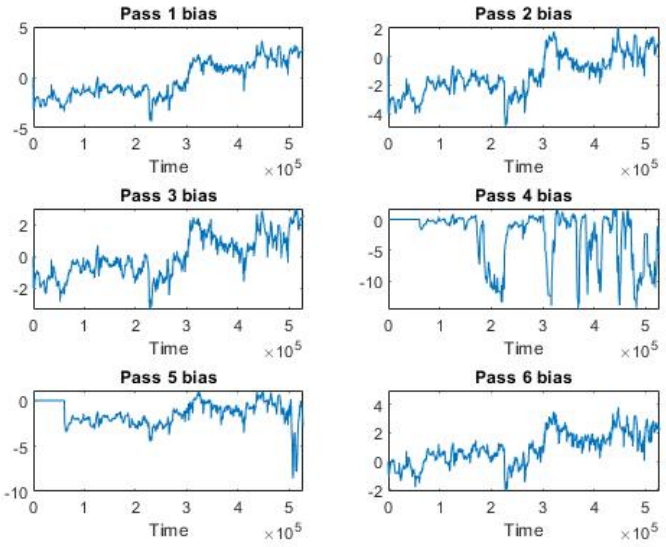

Figure 3.1: Time trend of the bias correction term

One of the major assumptions made for the energy balance was that the fuel gas flow rate is sufficient to capture the dynamics of the energy into the system. In reality, however, the energy provided in the system is more complicated, especially when the property of the fuel gas changes. Also, the model did not consider energy loss. With such assumptions made, it is not surprising that the bias correction was used to correct the missing information from the system and not just disturbance/error from the transmitters.

Since bias most likely contains information about the system, we can potentially improve the performance of the soft sensor if we can extract information from the bias. For this reason, it was decided to improve the performance of the soft sensor by extracting features from the bias.

3.3 Proposed Method

Bias correction is the last step in the soft sensor which is used to improve the prediction. If the bias trend from the different passes shares a similar characteristic, then using the extracted features from the bias corrections should further improve the soft sensor performance. However, since the bias correction is time-dependent and we are updating the next bias term to be used, we cannot directly use the extracted features from bias for online application. Consequently, we instead need to find a way to map it back to the process variables.

Feature extraction methods are commonly used in machine learning to extract valuable information from the system. In particular, since feature extraction methods sort out the latent variables in the order of importance, feature extraction methods are widely used for feature selection and dimension reduction [23]. In essence, feature extraction methods project the data into a latent space using a loading matrix. As the loading matrix is used to connect the original system (data) with the latent space, we use the loading matrix to determine the latent variables which define the latent space. By selecting the adequate latent variable / proper number of dimensions, we can filter out the noise from the data and define a new system (in latent space). By applying feature extraction methods to the bias and the process variables, we can further improve the prediction by adding valuable information as one of the inputs.

The common trend of the bias term from the different passes suggests that the bias term is also correcting to amend the missing information from the process data. Therefore, by selecting features from the bias terms to extract valuable information, and then mapping the process variables with the selected features, we can potentially

find good parameters for the process variables to improve the performance of the soft sensor. The proposed method is summarized below:

1. Apply bias updating to the original (first-principles) energy balance model.
2. Obtain bias corrected time trend β from the multiple passes in the OTSG and see if they have a similar trend.
3. Extract features from multiple bias terms using feature extraction methods.
4. Regress the process variables onto the extracted features to obtain coefficients (loading matrix) for the process variables.
5. Regress process variables and original energy balance model onto manual steam quality soft sensor.

Figure 3.2 shows the general process of the proposed method. In summary, we are extracting features from the bias, regressing the process variables to the extracted features to determine the optimal coefficients for the process variables to further improve the performance of the soft sensor. For online implementation and verification, only the model without an online bias updating technique will be used to determine its predictability as an infinite step ahead predictor. To verify if the information from the bias is more useful than applying the feature extraction method directly to the process variables, the effect of using feature extracted directly from the process variables has also been investigated. Figure 3.3 shows this method.

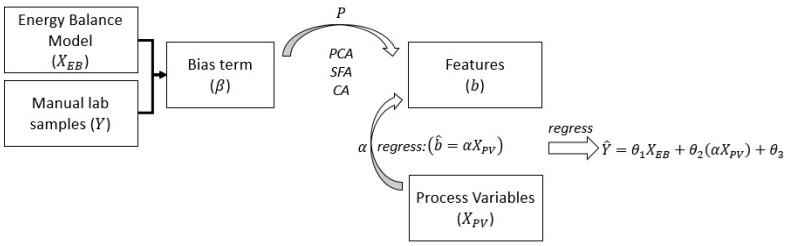

Figure 3.2: Schematic of the proposed method

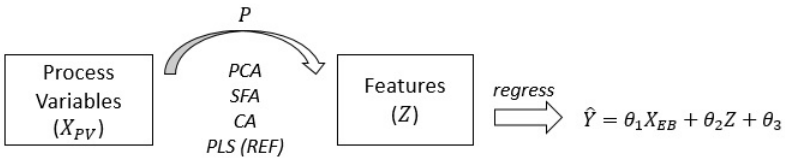

Figure 3.3: Schematic of the comparing method

To validate and compare the improvement of adding the extracted features as one of the inputs, the original energy balance model is used as a benchmark:

$$\hat{X}_M = \frac{F_{NG}}{F_i(C_p\Delta T + \Delta H)} \tag{3.3}$$

where F_{NG} is the fuel gas flowrate, F_i is the boiler feedwater flowrate from the individual passes, and C_p is the specific heat capacity. ΔT is the temperature difference between the individual inlet flowrate and recombined outlet flowrate and ΔH is the heat of vaporization. Detailed information on how this model was formulated and the corresponding assumptions have been explained in Chapter 1 in detail. Three deterministic feature extraction methods are chosen and an explanation of each method follows.

3.4 Overview of PCA

PCA (principal component analysis) is one of the most commonly used feature extraction methods in process data analysis. Also, it is commonly used for a normalization technique called sphering [24]. The sphering technique using PCA is applied in SFA, which will be discussed shortly.

It was first invented by Pearson in 1901 [1] and independently developed by Hotelling in 1933 [2]. In essence, PCA tries to extract features based on the variation. It transforms the data and projects it into a coordinate system (commonly known as the latent space) so the components (also known as scores) are orthogonal to each other and ordered based on variance. Thus the first component will try to

find the direction where the variance of the system is maximized:

$$\text{maximize}_v (Xv)^T (Xv) \quad \text{subject to } v^T v = 1 \tag{3.4}$$

In scalar format:

$$\text{maximize}_{v_{11},\ldots,v_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \Big(\sum_{j=1}^{p} v_{j1} x_{ij} \Big)^2 \right\} \quad \text{subject to } \sum_{j=1}^{p} v_{j1}^2 = 1 \tag{3.5}$$

where v is the parameter (loading vector), and x_{ij} is the data point. Since PCA tries to find the direction where the data varies the most, it essentially tries to find the eigenvectors and orders them such that the largest eigenvalue of the covariance matrix corresponds to the direction where it is desired to project the data onto. PCA solves equation (3.5) by applying singular value decomposition on the covariance matrix of the centered data. Since the covariance matrix is symmetric:

$$\Sigma_x = \frac{X^T X}{N-1} \tag{3.6}$$

Applying SVD on Σ_x:

$$\Sigma_x = V \Lambda V^T \tag{3.7}$$

where V shows the direction in which the data varies the most and it is ordered by the magnitude of the eigenvalues:

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$$

The first principal component can be determined by mapping the original data based on the eigenvectors:

$$Z = XV \tag{3.8}$$

In a scalar format:

$$z_{il} = v_{11} x_{i1} + v_{21} x_{i2} + \ldots + v_{p1} x_{ip} \quad \text{where i = 1,...,n} \tag{3.9}$$

Equation (3.9) is commonly known as scores or latent variables and they define the latent space. For p dimension system, p number of latent variables can be obtained.

Since PCA tries to find the direction where the variance of the system is maximized, it can consequently be used as a dimension reduction method: it can represent the data in a lower dimension that contains most of the variation. Also, it can be observed that there is no temporal interpretation in PCA: PCA is independent of time. To incorporate temporal information, SFA (slow feature analysis) and CA (cointegration analysis) have been considered.

3.5 Overview of SFA

Like PCA, SFA (slow feature analysis) is another feature extraction method that uses eigendecomposition to solve the feature extraction problem. Unlike PCA, SFA focuses on the dynamics and tries to extract slowly varying features.

Suppose we are given data matrix X ($N \times P$) with N representing the number of sample points, $t = 1, 2, ..., N$, and P representing the dimension of the system. So at a given time point:

$$x(t) = [x_1(t), x_2(t), ..., x_P(t)] \tag{3.10}$$

Assume that the input vector $x(t)$ is mapped to a latent subspace by using the following input-out function $g(x)$:

$$g(x) = [g_1(x), g_2(x), ..., g_J(x)] \tag{3.11}$$

The latent subspace can be expressed as:

$$y_j(t) = g_j(x(t)), j = 1, ..., m \tag{3.12}$$

The mapping $g_j(x(t))$ is typically assumed as a linear combination of some basis functions

$$y_j(t) = g_j(x(t)) = w_j^T h(x(t)) = w_j^T z(t) \tag{3.13}$$

where $z(t)$ is the expanded signal: $z(t) = h(x(t))$. In [25], the authors mention how $h(x)$ can be a set on nonlinear function. In this thesis, $z(t)$ will be derived linearly.

The objective of SFA is to find the slowly varying feature by trying to minimize the temporal variation within the latent space:

$$\text{minimize } \mathbb{E}(\dot{y}_i^2) \tag{3.14}$$

where

$$\dot{y}_i(t) = y_i(t) - y_i(t-1) \tag{3.15}$$

Since any constant value would give the optimal solution, equation (3.14) must be satisfied while subject to:

$$\mathbb{E}(y_i) = 0 \tag{3.16}$$

$$\mathbb{E}(y_i^2) = 1 \tag{3.17}$$

$$\mathbb{E}(y_i y_j) = 0, \forall i \neq j \tag{3.18}$$

Equation (3.18) ensures that obtained slow features are orthogonal to each other, thus decorrelated. By minimizing the temporal variation of the latent score, we try to find a smooth, simple underlying feature from the dataset, which tends to be important in process data analysis [26].

If expanded, equation (3.14) becomes

$$\text{minimize } \mathbb{E}(\dot{y}_i^2) = w_j^T \mathbb{E}(\dot{z}\dot{z}) w_j \tag{3.19}$$

If the expanded signal, $z(t)$, was chosen such that it has zero mean and unit variance:

$$\mathbb{E}(y_i) = w_j^T \mathbb{E}(z) = 0 \tag{3.20}$$

$$\mathbb{E}(y_i^2) = w_j^T \mathbb{E}(zz^T) w_j = w_j^T w_j = 1 \tag{3.21}$$

$$\mathbb{E}(y_i y_j) = w_i^T \mathbb{E}(zz^T) w_j = w_i^T w_j = 0, \forall i \neq j \tag{3.22}$$

In other words, if the weight vectors are constrained to be orthonormal, the three constraints will be satisfied.

A normalization method called sphering or whitening will be used to generate signal $z(t)$ such that it has zero mean and identity covariance. One of the ways to whiten data is by applying PCA [24].

$$\Sigma_x = \frac{X^T X}{N-1} = U \Lambda U^T \tag{3.23}$$

where X has been zero-mean centered.

$$\Sigma_x^{-1} = U \Lambda^{-1} U^T = U \Lambda^{-1/2} \Lambda^{-1/2} U^T \tag{3.24}$$

$$\Sigma_x^{-1/2} = \Lambda^{-1/2} U^T \tag{3.25}$$

$$z = \Lambda^{-1/2} U^T X \tag{3.26}$$

Since z has zero mean and identity covariance matrix, the objective function now can be simplified to:

$$\min_{w_j} w_j^T \mathbb{E}(\dot{z}\dot{z}^T) w_j \tag{3.27}$$

such that

$$w_j^T w_j = 1$$

$$w_i^T w_j = 0, \forall i \neq j$$

Solving the optimization problem such that the weight vectors are set of orthonormal vectors:

$$J = \min_{w_j} w_j^T \mathbb{E}(\dot{z}\dot{z}^T) w_j - \lambda(w_j^T w_j - I) \tag{3.28}$$

$$\frac{dJ}{dw_j} = 2\mathbb{E}(\dot{z}\dot{z}^T) w_j - 2\lambda w_j = 0 \tag{3.29}$$

$$\mathbb{E}(\dot{z}\dot{z}^T) w_j = \lambda w_j \tag{3.30}$$

From equation (3.30), it can be seen that w_j can be obtained by solving eigenvalue decomposition problem on matrix $\dot{z}\dot{z}^T$, where the weight factors are ordered such that:

$$\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_J$$

For nonlinear expansion, $h(x)$ can be chosen such that $z(t)$ becomes expanded signal. For example, for quadratic SFA:

$$h(x) = [x_1, ..., x_p, x_1x_1, ..., x_1x_p] \tag{3.31}$$

3.6 Overview of CA

CA (cointegration analysis) was developed by Engle and Granger [3] in econometrics and Chen et al [4] first introduced it in chemical engineering for statistical process monitoring. The detailed derivation of CA is shown in the next chapter, but a brief explanation of the algorithm is summarized below.

The main objective of CA is to extract the most stationary feature within a nonstationary dataset so that we can understand the cointegrated relationship between the variables. Assuming that the dataset can be represented as a VAR (vector autoregressive) model, and then applying first-order difference, the system can be represented as:

$$\begin{aligned} X_t - X_{t-1} = [\Pi_1 - \mathrm{I}][X_{t-1} - X_{t-2}] + [\Pi_1 + \Pi_2 - \mathrm{I}][X_{t-2} - X_{t-3}] \\ + ... + [\Pi_1 + \Pi_2 + .. + \Pi_k - \mathrm{I}]X_{t-k} + e_t \end{aligned} \tag{3.32}$$

$$\Delta X_t = \Gamma_1 \Delta X_{t-1} + \Gamma_2 \Delta X_{t-2} ... + \Gamma_{k-1} \Delta X_{t-k+1} + \Pi X_{t-k} + e_t \tag{3.33}$$

where e_t is white noise with zero mean and finite variance, $e_t \sim N(0, \Lambda)$ and the parameter Π maps the nonstationary space to the stationary space. Π can be further decomposed into α and β:

$$\Pi = \alpha\beta^T \tag{3.34}$$

where $\beta^T X_t$ becomes stationary even if X_t is nonstationary. Johansen shows how to determine β by applying eigendecomposition [5]. To apply CA on bias however, we instead try to find the most nonstationary feature by slightly modifying CA. Explanation of this modification will be discussed in detail in Chapter 4.

3.7 Overview of PLS

PLS (partial least squares) is different from the unsupervised feature extraction methods mentioned above in the sense that PLS is a supervised learning method while PCA, SFA, and CA are unsupervised methods. Unsupervised methods do not use any information from the response variable (Y), which in this case is the manual samples of the steam quality. Consequently, there is no guarantee that the extracted features using the unsupervised methods would best describe the response variables. PLS, on the other hand, maps both X ($N \times P$) and Y ($N \times M$) to latent space such that the covariance between the scores extracted from both is maximized.

$$X = TP^T + E \tag{3.35}$$

$$Y = UQ^T + F \tag{3.36}$$

where X(N $\times$ P) is a regressor matrix, Y (N $\times$ M) is the matrix containing the response variables, T is the latent score matrix for X, U is the latent score matrix for Y, P is the projection matrix that is used to map X, U is the projection matrix for Y, E is the residual matrix for X and F is the residual matrix for Y.

The objective function for PLS can be defined as:

$$\max \quad \frac{W_x^T X^T Y W_y}{\sqrt{W_x^T W_x}\sqrt{W_y^T W_y}} \tag{3.37}$$

which is equivalent to:

$$\max \quad W_x^T X^T Y W_y, \tag{3.38}$$

subject to

$$W_x^T W_x = 1$$

$$W_y^T W_y = 1$$

Solving the optimization problem:

$$J = W_x^T X^T Y W_y - \lambda_x W_x^T W_x - \lambda_y W_y^T W_y \tag{3.39}$$

$$\frac{dJ}{dW_x} = X^T Y W_y - 2\lambda_x W_x = 0 \tag{3.40}$$

$$\frac{dJ}{dW_y} = W_x^T X^T Y - 2\lambda_y W_y = 0 \tag{3.41}$$

Since 2 can be absorbed in the eigenvalues, equations (3.40) and (3.41) can be simplified to:

$$W_x^T S_{xy} W_y = \lambda_x W_x^T W_x \tag{3.42}$$

$$W_y^T S_{yx} W_x = \lambda_y W_y^T W_y \tag{3.43}$$

respectively, where

$$S_{xy} = X^T Y$$

$$S_{yx} = S_{xy}^T = Y^T X$$

Since $W_x^T W_x = 1$ and $W_y^T W_y = 1$, it is easy to see that:

$$\lambda_x = \lambda_y = \lambda$$

Therefore, the optimization problem can be simplified to:

$$\begin{bmatrix} 0 & S_{xy} \\ S_{yx} & 0 \end{bmatrix} \begin{bmatrix} W_x \\ W_y \end{bmatrix} = \lambda \begin{bmatrix} W_x \\ W_y \end{bmatrix} \tag{3.44}$$

Compared to the ordinary least squares (OLS) method, PLS can handle colinearity better as the regressors are orthogonalized. If x_j and x_i are highly correlated, the parameters, $\theta(j)$ can become very unstable, which is one of the main drawbacks of OLS. PLS projects Y and X onto a subspace to obtain latent variables T and U respectively such that the elements of T and U are independent. Consequently, when estimating the parameter θ, the regressors do not affect each other's parameters.

There are two commonly used algorithms to compute PLS: NIPALS (nonlinear iterative partial least squares) or SIMPLS (simple partial least squares). In this thesis, PLS obtained via NIPALS will be explained.

As the name suggests, NIPALS is an iterative algorithm and it tries to find the scores factor by factor, $T = [t_1, t_2, ..., t_A]$ and $U = [u_1, u_2, ..., u_A]$ where $A \leq \min(\text{P,M})$.

The first t_1 and u_1 are weighted sum of the mean-centered variables.

$$t_1 = X_0 w_1 \tag{3.45}$$

$$u_1 = Y_0 q_1 \tag{3.46}$$

where X_0 and Y_0 are the mean-centered input and output matrices respectively, w_1 and t_1 are the weight vectors that can be obtained by applying SVD onto $X^T Y$. After obtaining the first set of scores, residual matrices are calculated as:

$$X_1 = X_0 - t_1(t_1 X_0)(t_1^T t_1)^{-1} \tag{3.47}$$

$$Y_1 = Y_0 - t_1(t_1 Y_0)(t_1^T t_1)^{-1} \tag{3.48}$$

From the equations, it can be observed that the residual matrices have been obtained by regressing all variables on t_1.

Introducing p to represent the loading vector for t_1 on X, equation (3.47) can be rewritten as,

$$X_1 = X_0 - t_1 p_1^T \tag{3.49}$$

where

$$p_1 = X_0 t_1 (t_1^T t_1)^{-1}$$

Similarly, equation (3.48) can be rewritten as,

$$Y_1 = Y_0 - b_1 t_1 q_1^T \tag{3.50}$$

where b is the regression coefficient for the inner relationship:

$$b_1 = u_1^T t_t (t_1^T t_1)^{-1} \tag{3.51}$$

SVD is applied again on the residual matrix to determine the next set of latent scores. The NIPALS algorithm continues until the dimension of the subspace, A, has been determined.

3.7.1 Summary of Methods

In summary, the methods reviewed above will be used as the feature extraction methods, which is either applied directly on the process variables or the bias term. To determine which method is better, the original energy balance model will be used as a benchmark. To have a simpler notation, each method will be distinguished with subscripts and its description is as follows:

1. Benchmark: The original energy balance model: $\hat{Y}_1 = \theta_{1,1}\hat{X}_M + \theta_{2,1}$

2. Applying PCA on the process variables and adding the features as the inputs along with the energy balance model: $\hat{Y}_2 = \theta_{1,2}\hat{X}_M + \theta_{2,2}Z_{PCA,PV} + \theta_{3,2}$

3. Applying SFA on the process variables and adding the features as the inputs along with the energy balance model: $\hat{Y}_3 = \theta_{1,3}\hat{X}_M + \theta_{2,3}Z_{SFA,PV} + \theta_{3,3}$

4. Applying CA on the process variables and adding the features as the inputs along with the energy balance model: $\hat{Y}_4 = \theta_{1,4}\hat{X}_M + \theta_{2,4}Z_{CA,PV} + \theta_{3,4}$

5. Applying PLS on the process variables (as inputs) and manual samples (as outputs) and adding the features as the inputs along with the energy balance model: $\hat{Y}_5 = \theta_{1,5}\hat{X}_M + \theta_{2,5}(\alpha_{PCA,bias}X_{PV}) + \theta_{3,5}$

6. Applying PCA on the bias term, regress the process variables onto the extracted features to obtain model coefficients, and then use the process variables along with the energy balance model as the model inputs: $\hat{Y}_6 = \theta_{1,6}\hat{X}_M + \theta_{2,6}(\alpha_{PCA,bias}X_{PV}) + \theta_{3,6}$

7. Applying SFA on the bias term, regress the process variables onto the extracted features to obtain model coefficients, and then use the process variables along with the energy balance model as the model inputs $\hat{Y}_7 = \theta_{1,7}\hat{X}_M + \theta_{2,7}(\alpha_{SFA,bias}X_{PV}) + \theta_{3,7}$

8. Applying CA on the bias term, regress the process variables onto the extracted features to obtain model coefficients, and then use the process variables along with the energy balance model as the model inputs $\hat{Y}_8 = \theta_{1,8}\hat{X}_M + \theta_{2,8}(\alpha_{CA,bias}X_{PV}) + \theta_{3,8}$

3.8 Results and Discussion

To validate and compare the model, correlation and mean absolute error (MAE) are calculated and compared on the validation data. Pass 4 manual sampling points vary a lot compared to other passes, which has resulted in a larger error. For this reason, pass 4 was not used to determine which method is the best. Up to three features were used for feature extraction methods and the results are summarized in Table 3.1 to 3.6. The top two results from each pass are bolded.

3.8.1 Correlation Analysis Comparison

To determine the best model, the correlation between the different models and the manual samples was compared. The results for using one feature is summarized in Table 3.1, two features in Table 3.2, and three features in Table 3.3. From the tables, it can be seen that when using only one feature, using SFA to extract features from the process variables and CA to extract the feature from the bias term seems to give the best result. Shardt and Huang mention how online bias term is used to capture the slow/drifting disturbance [21, 22]. Since CA was used to extract the nonstationary feature, it is likely that the extracted feature contains the drifting phenomenon of the system. Cointegration biased model and the benchmark model have been compared and plotted in Figure 3.4 as an example to see which model correlates better with the manual sample readings. From the figure, it can be observed that the Cointegration bias model has a better correlation with the manual samples than the benchmark model. When applying the feature extraction method on the process variables directly, SFA gave the best result. This may indicate and confirm

how the important features in the system are the slowly varying trend in chemical engineering [26].

On the other hand, when using two or three features, it is difficult to see which feature extraction method is superior to the other. When using two features, it generally seems using CA on either process variables or the bias term seems to give better correlation with the manual samples, which may indicate that nonstationary/drifting features embedded in the system contain valuable information. The quality variables that are being attempted to be predicted are steam quality of the steam generator, and it is widely known that steam generators suffer from fouling accumulation (in addition to equipment aging). If features extracted from CA was able to improve the correlation between the manual samples, then it may suggest that the process variables contain important nonstationary feature of the system. When using three features, results are even more spread out, which may suggest as more features are used, it starts to capture more information, regardless of its characteristics: it becomes difficult to distinguish important nonstationary features from a slowly varying feature and vice versa. Regardless of how many features were used, extracting features from bias or directly from the process variables did not show any evident distinction. Also, extracting features using PCA did not seem to improve the correlation compared to the benchmark model, which may suggest that features that reflect variance of the system are not as informative as the features that represent nonstationarity or slowness.

3.8.2 MAE (mean absolute error) Comparison

Online bias update is heavily dependent on the sampling frequency of the manual samples, and manual samples typically require manual work. Consequently, it would be ideal to develop a model that is solely dependent on readily available process variables that are typically measured using instrumentation. To determine if an infinite step ahead predictor could be developed, MAE was also compared. Results

Table 3.1: Model comparison based on correlation (using 1 feature)

	Pass 1	Pass 2	Pass 3	Pass 4	Pass 5	Pass 6
$\hat{Y}_1$ (benchmark)	0.66	0.73	0.76	0.14	0.70	0.78
$\hat{Y}_2$ (PCA, PV)	0.69	0.68	0.74	0.27	0.66	0.65
$\hat{Y}_3$ (SFA, PV)	0.62	**0.80**	**0.84**	0.07	**0.71**	**0.82**
$\hat{Y}_4$ (PLS, PV)	0.41	0.35	0.40	0.14	0.30	0.33
$\hat{Y}_5$ (CA, PV)	**0.84**	0.73	0.71	0.19	0.70	0.67
$\hat{Y}_6$ (PCA, bias)	0.74	0.69	0.74	-0.26	0.66	0.70
$\hat{Y}_7$ (SFA, bias)	0.56	0.58	0.61	0.11	0.65	0.60
$\hat{Y}_8$ (CA, bias)	**0.87**	**0.79**	**0.80**	-0.20	**0.71**	**0.79**

Table 3.2: Model comparison based on correlation (using 2 features)

	Pass 1	Pass 2	Pass 3	Pass 4	Pass 5	Pass 6
$\hat{Y}_1$ (benchmark)	0.66	0.73	**0.76**	0.14	0.70	**0.78**
$\hat{Y}_2$ (PCA, PV)	0.56	0.56	0.56	0.11	0.62	0.57
$\hat{Y}_3$ (SFA, PV)	**0.74**	0.63	0.70	0.12	**0.71**	0.69
$\hat{Y}_4$ (PLS, PV)	0.47	0.33	0.38	0.30	0.38	0.37
$\hat{Y}_5$ (CA, PV)	**0.81**	**0.78**	**0.83**	0.08	0.69	**0.79**
$\hat{Y}_6$ (PCA, bias)	0.69	0.68	0.74	0.07	0.66	0.69
$\hat{Y}_7$ (SFA, bias)	0.60	0.76	0.64	-0.18	0.69	0.74
$\hat{Y}_8$ (CA, bias)	0.64	**0.78**	0.74	-0.02	**0.70**	0.77

are summarized in Table 3.4 to 3.6. For visual comparison, Figure 3.5 was plotted, which compares the benchmark model ($\hat{Y}_1$) with the model that used CA to extract a feature from the bias terms ($\hat{Y}_8$). From the figure, it can be observed that the proposed model ($\hat{Y}_8$) can capture the widespread of manual samples around 4×10^5 - 5×10^5 better. From the tables, it can be seen that adding features improves the results, but it is debatable if they are sufficient to develop a soft sensor without an online bias updating technique. Compared to the added complexity, the results seem unrewarding, especially because with the online bias updating technique, MAE for all the passes was lower than one.

Table 3.3: Model comparison based on correlation (using 3 features)

	Pass 1	Pass 2	Pass 3	Pass 4	Pass 5	Pass 6
$\hat{Y}_1$ (benchmark)	0.66	0.73	**0.76**	0.14	**0.70**	0.78
$\hat{Y}_2$ (PCA, PV)	0.51	0.50	0.53	0.21	0.57	0.48
$\hat{Y}_3$ (SFA, PV)	0.71	0.63	0.70	0.14	0.61	0.79
$\hat{Y}_4$ (PLS, PV)	0.66	0.49	0.55	0.18	0.53	0.53
$\hat{Y}_5$ (CA, PV)	**0.84**	0.79	0.53	0.04	0.67	**0.81**
$\hat{Y}_6$ (PCA, bias)	0.60	0.72	**0.76**	-0.09	0.64	0.73
$\hat{Y}_7$ (SFA, bias)	**0.73**	**0.81**	0.75	-0.01	0.69	**0.81**
$\hat{Y}_8$ (CA, bias)	0.67	**0.80**	0.75	0.01	**0.70**	0.79

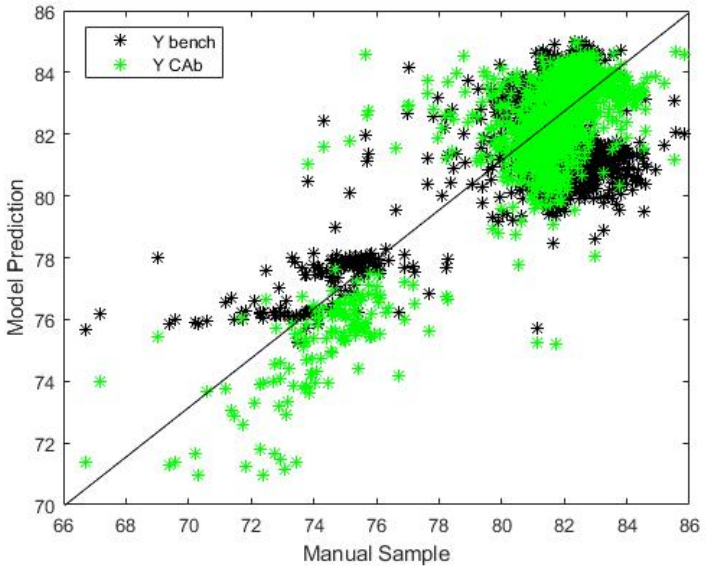

Figure 3.4: Correlation comparison for Pass 1

3.9 Summary

In this chapter, feature extraction methods (PCA, CA and SFA) were explored to improve the steam quality soft sensor that was developed in Chapter 1. Feature extraction methods were applied either directly to the process variables or on the online bias terms. Regardless of the method, adding features generally improved the results. In particular, when using only one feature, the feature extracted from the process variables using SFA generally gave the best result while features extracted from the bias terms using CA generally gave the best result. This aligns with previous

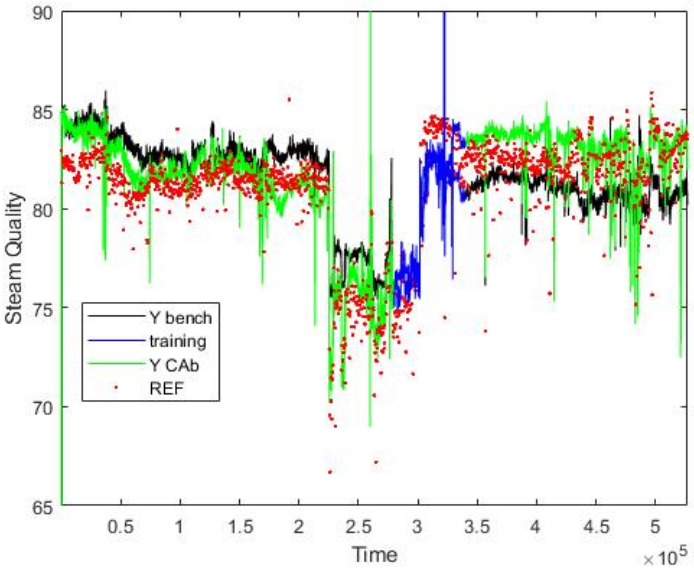

Figure 3.5: Time Trend Comparison for Pass 1

Table 3.4: Model comparison based on MAE (using 1 feature)

	Pass 1	Pass 2	Pass 3	Pass 4	Pass 5	Pass 6
$\hat{Y}_1$ (benchmark)	1.79	1.54	**1.28**	7.62	1.47	**1.20**
$\hat{Y}_2$ (PCA, PV)	1.45	**1.41**	1.33	7.86	1.42	1.43
$\hat{Y}_3$ (SFA, PV)	1.85	**1.23**	**1.07**	8.62	**1.27**	**1.01**
$\hat{Y}_4$ (PLS, PV)	2.36	2.49	2.60	7.68	2.69	2.49
$\hat{Y}_5$ (CA, PV)	**1.29**	1.60	1.50	8.46	1.42	1.45
$\hat{Y}_6$ (PCA, bias)	1.51	1.77	1.39	7.86	1.41	1.60
$\hat{Y}_7$ (SFA, bias)	2.07	2.15	1.86	7.61	1.51	1.91
$\hat{Y}_8$ (CA, bias)	**1.16**	1.45	1.46	8.36	**1.25**	1.53

statements how slowly varying features are important in process engineering while the online bias term is used to capture the disturbance in the system. In addition to comparing correlation, MAE was also compared to see if an infinite step ahead predictor can be developed, but it was concluded that adding extracted features as inputs is not sufficient to create an infinite step ahead predictor.

Table 3.5: Model comparison based on MAE (using 2 features)

	Pass 1	Pass 2	Pass 3	Pass 4	Pass 5	Pass 6
$\hat{Y}_1$ (benchmark)	1.79	1.54	**1.28**	7.62	1.47	**1.20**
$\hat{Y}_2$ (PCA, PV)	1.73	1.85	1.96	8.16	1.56	1.77
$\hat{Y}_3$ (SFA, PV)	1.61	1.85	1.68	9.07	**1.28**	1.66
$\hat{Y}_4$ (PLS, PV)	2.23	2.87	2.81	8.21	2.63	2.56
$\hat{Y}_5$ (CA, PV)	**1.16**	**1.24**	**1.14**	8.62	**1.27**	**1.17**
$\hat{Y}_6$ (PCA, bias)	**1.64**	1.79	1.44	7.71	**1.43**	1.60
$\hat{Y}_7$ (SFA, bias)	2.00	1.58	1.79	8.59	1.36	1.66
$\hat{Y}_8$ (CA, bias)	1.89	**1.51**	1.65	8.01	1.30	1.60

Table 3.6: Model comparison based on MAE (using 3 features)

	Pass 1	Pass 2	Pass 3	Pass 4	Pass 5	Pass 6
$\hat{Y}_1$ (benchmark)	1.79	**1.54**	**1.28**	7.62	**1.47**	**1.20**
$\hat{Y}_2$ (PCA, PV)	1.98	1.95	1.96	8.50	1.70	1.82
$\hat{Y}_3$ (SFA, PV)	**1.49**	1.84	1.68	9.17	1.57	1.50
$\hat{Y}_4$ (PLS, PV)	1.67	2.47	2.44	8.87	2.22	2.23
$\hat{Y}_5$ (CA, PV)	**1.08**	1.81	1.90	8.67	1.50	**1.39**
$\hat{Y}_6$ (PCA, bias)	1.81	1.75	**1.39**	8.31	1.56	1.59
$\hat{Y}_7$ (SFA, bias)	1.91	1.87	1.55	7.94	1.49	1.51
$\hat{Y}_8$ (CA, bias)	1.86	**1.44**	1.64	7.78	**1.31**	1.54

Chapter 4

Extracting Nonstationary Features for Process Data Analytics and Application in Fouling Detection *

4.1 Introduction

With the ability to store a large amount of data, data-driven methods and latent variable models are becoming more popular and are being used for predictive modeling and process monitoring. Compared to conventional methods, latent variable models have the advantage of extracting representative latent features to capture the essential information, which makes them more robust to low-quality observations. As one critical component of time series data, the nonstationary trend is the desired variation to be extracted in this study. In particular, nonstationary features can be used to represent the cumulating behaviour in process data. To extract the feature that represents cumulating behavior, a novel feature extraction method is proposed based on a cointegration analysis. Through its maximum likelihood estimation, nonstationary trends can be extracted, and the effectiveness of the proposed model is validated with numerical examples. By combining with principal component analysis, its advantages are highlighted in an industrial application of monitoring fouling buildup in steam generators.

*part of this chapter has been submitted for publication as "**Kwak, S.**, Ma, Y., & Huang, B. (2018). Extracting Nonstationary Features for Process Data Analytic and Application in Fouling Detection" in Computers & Chemical Engineering.

4.2 Background

Predictive models play an important role in the chemical process. In particular, when influential variables cannot be measured with transmitters easily, predictive models are formulated to provide informative estimation from readily available process measurements. For example, soft sensors [20] use predictive models to give more granular estimations of quality variables, which can be used to optimize the process. Examples of quality variables are steam quality out of steam generators and water content in water-oil emulsions. The predictive model can either be the first principle based, data-driven, or a mixture of both. First principle based models essentially use conservation laws such as mass and energy balance to describe the process. Consequently, they usually assume an ideal, steady state process and can be sensitive to condition changes. Also, to develop theoretical models, we need in-depth knowledge about the process and availability of all the related measurements.

Data-driven methods, on the other hand, are less dependent on the detailed knowledge about the process. As a result, when the process is too complicated to develop a first-principles model, data-driven methods become attractive and useful in predicting important quality variables [6, 27, 20, 7]. Examples of data-driven models include principal component analysis (PCA) [28], partial least squares (PLS)[29] and slow feature analysis (SFA) [25]. The similarity among these methods is that raw observations are projected to a latent space to formulate "informative" features. For example, PCA projects the observations based on the variance of the system. PLS uses latent space to extract correlated variations between inputs and the desired outputs. SFA tries to separate slowly varying features from quickly varying observations.

A common assumption made in the aforementioned data-driven methods is that the system is stationary. In other words, the statistical properties (such as mean and variance) of the observed data are independent of time. Due to this assumption, the latent scores ($s_{1:T}$) are usually assumed with zero mean and unit variance: $\mathbb{E}_t(s_t) = 0$

and $\mathbb{E}_t(s_t \cdot s_t) = 1$. In reality, however, chemical processes can be nonstationary due to different operating modes, equipment aging, accumulation within the system, and so on. In some applications, understanding the most nonstationary feature within the system is valuable. Accumulation in chemical processes, such as fouling buildup in heat exchanging equipment, is an example of a nonstationary trend. Fouling buildup in heat exchanging equipment has been a long-lasting issue in chemical engineering. Due to its complexity, however, it is difficult to monitor fouling buildups directly in real time. In some simplified cases, the fouling buildup can be monitored through some primitive calculations. For example, monitoring the differential pressure across the tube can indicate fouling buildup: if differential pressure increases over time, it may indicate that tube diameter is decreasing, which is most likely caused by fouling buildup on the tube surface. However, since differential pressure contains information other than reducing tube diameter, monitoring differential pressure alone cannot give consistent estimates. Indeed, there are fouling predictive models that give a better understanding of fouling buildup [30, 31] and many of the models try to predict fouling resistance:

$$R_f = \frac{1}{U_{dirty}} - \frac{1}{U_{clean}} \tag{4.1}$$

where R_f is the fouling resistance, U_{clean} represents the overall heat transfer coefficient when the tubing is clean and U_{dirty} is the overall heat transfer coefficient when the tubing has fouling buildup. Change in the fouling resistance would indicate the effect of fouling in the system over time. Ebert and Panchal [31] introduced the "threshold concept" to predict fouling in crude oil processing:

$$\frac{dR_f}{dt} = ARe^{-\beta} \exp\big(-\frac{E}{RT_f}\big) - C\tau_w, \tag{4.2}$$

where A, β and C are model parameters. However, the application of the fouling resistance predictive models on the industrial heat exchangers is challenging. For example, the overall heat transfer coefficient depends on the heat exchanger configuration and

chemical process which makes its calculation complex. Also, the parameters in equation (4.2) can only be determined experimentally. As a result, it is inevitable to tune the estimate parameters for an online application. Since fouling data is collected sparsely, determining these parameters is difficult.

To understand the complex fouling phenomenon, a data-driven method is proposed in this work. By extracting the nonstationary features from historical process measurements, the fouling predictive model uses latent features to capture the cumulating trend. Consequently, we are no longer limited by rigorous physical assumptions and difficulties from sparse samples. Cointegration analysis (CA) is a method actively used in econometrics that tries to find the relationship between stationarity and nonstationarity within the system. One particular advantage of CA is that it preserves the dynamics that exist over a long period. By preserving the dynamic information, we can potentially distinguish the stationary features from the nonstationary features. The concept of CA was first introduced by Engle and Granger [3], where they used CA to show that consumption and income are cointegrated. For process data analytics, CA allows us to extract the stationary relationship between variables while the system is nonstationary. CA was first presented in chemical engineering by Chen et al [4], where it was used for process monitoring purposes. CA based process monitoring focuses on nonstationary variables within the system and identifies faulty variables within the system. Based on the cointegration relationship, the faults can be detected when the cointegration relationship is disrupted and the noise term becomes nonstationary. Successful application of CA in chemical engineering can be found in [32, 33], and [34], where CA was again used for fault detection and diagnosis of nonstationary processes. The existing adaptation of CA focuses on extracting stationary features to study the cointegrated relationship between process variables. However, nonstationary trends are our interest. Therefore, a novel CA based feature extraction method is proposed to extract the most nonstationary feature to determine the fouling.

In this chapter, a method to extract the most nonstationary feature in the system is presented. In section 4.4, the method to extract the most nonstationary feature is explained. In sections 4.5 and 4.6, numerical simulation and industrial examples for estimating fouling buildup are discussed respectively. Finally, in section 4.7, concluding remarks with some future works are listed.

4.3 Preliminaries: Cointegration Analysis (CA)

If a nonstationary system X becomes stationary by differencing it by d times, it is said to be integrated of order d, which can be also represented as $X \sim I(d)$. Cointegration analysis was originally proposed to show that nonstationary variables can have a stationary relationship as long as the nonstationary variables are integrated with the same order [3]. Johansen proposed a method to use a vector autoregressive (VAR) model to determine the cointegrated relationship between nonstationary variables. VAR model expresses a dynamic system based on its own lagged values:

$$X_t = \Pi_1 X_{t-1} + \Pi_2 X_{t-2} + ... + \Pi_k X_{t-k} + e_t \tag{4.3}$$

where X is $(N \times P)$ data matrix with P representing the dimension of the system and N representing the number of sample points, $t = 1, 2, ..., N$. k is the order of the VAR model and e_t is white noise with zero mean and finite variance, $e_t \sim$ N(0,Λ). The nonstationary data investigated by CA is assumed with first order nonstationarity, which means their first order differences are stationary ($X \sim I(1)$) [35]. The following vector error-correction (VEC) model can be transformed to show the connection between the stationary feature and the nonstationary feature within the system:

$$\begin{aligned} X_t - X_{t-1} = [\Pi_1 - \mathrm{I}][X_{t-1} - X_{t-2}] + [\Pi_1 + \Pi_2 - \mathrm{I}][X_{t-2} - X_{t-3}] \\ + ... + [\Pi_1 + \Pi_2 + .. + \Pi_k - \mathrm{I}]X_{t-k} + e_t \end{aligned} \tag{4.4}$$

$$\Delta X_t = \Gamma_1 \Delta X_{t-1} + \Gamma_2 \Delta X_{t-2} ... + \Gamma_{k-1} \Delta X_{t-k+1} + \Pi X_{t-k} + e_t \tag{4.5}$$

In a more compact form:

$$Z_{0t} = \Gamma Z_{1t} + \Pi Z_{kt} + e_t \tag{4.6}$$

where

$$Z_{0t} = \Delta X_t = [X_t - X_{t-1}]$$

$$Z_{1t} = [\Delta X_{t-1}, \Delta X_{t-2}, ..., \Delta X_{t-k+1}]$$

$$\Delta X_{t-1} = X_{t-1} - X_{t-2}, ..., \Delta X_{t-k+1} = X_{t-k+1} - X_{t-k}$$

$$\Gamma = [\Gamma_1, \Gamma_2, \Gamma_{k-1}]$$

$$\Pi = [\Pi_k + \Pi_{k-1} + ... - \mathrm{I}]$$

$$Z_{kt} = X_{t-k}$$

Based on the preliminary assumptions, the data sample Z_{kt} is from nonstationary sequences. On the other hand, the first order differences, Z_{0t} and Z_{1t}, are stationary. Thus, the matrix Π, which connects the nonstationary space to the stationary space, is the critical parameter. Depending on its rank, there are three possible cases:

1. rank(Π) = 0, implies $\Pi = 0$ and there's no cointegrated relationship.

2. rank(Π) = P implies all variables are cointegrated, hence the co-integrated system is stationary.

3. rank(Π) = r < P implies there are r cointegrated variables, or r co-integrated stationary variables.

It can be observed that the third case is the main interest, which indicates that Π can be further decomposed into two column full r-rank matrices:

$$\Pi = \alpha \beta^T \tag{4.7}$$

where β is a P-by-r matrix (β^T thus being β transpose), consisting of r column vectors as the cointegration vector and α is a P-by-r matrix, including some r column vectors as the distributing weights [5]. In cointegration analysis, β allows $\beta^T X_t$ to be a stationary feature even though X is nonstationary. Therefore, the goal of CA is to find β. To determine β, the maximum likelihood estimation method has been utilized [5]. Based on the probability density function for all N samples, ΔX can be expressed as:

$$\begin{aligned} p(\Delta X|\Gamma,\Pi,\Lambda) &= \prod_{t=1}^{N} p(\Delta X_t) \\ &= \prod_{t=1}^{N}\prod_{i=1}^{P} p(\Delta x_i) \qquad (4.8) \\ &= (2\pi)^{-NP/2}|\Lambda|^{-N/2}\exp(-\frac{1}{2}\sum_{t=1}^{N} e_t^T\Lambda^{-1}e_t) \end{aligned}$$

$$e_t = Z_{0t} - \Gamma Z_{1t} - \Pi Z_{kt} \qquad (4.9)$$

The problem is now formulated as the maximization of likelihood (4.8). Before investigating the parameter β, the optimal solution is first discussed for the parameter Γ, Π and the covariance matrix for the noises, Λ.

Before manipulating the above likelihood, the connection between Γ and Π can be used to reduce the computational complexity. Also, the fact that Z_{1t} is perpendicular to e_t is used to derive the following equation from equation ((4.6)):

$$M_{01} = \Gamma M_{11} + \Pi M_{k1} \qquad (4.10)$$

where

$$M_{01} = \sum_{t=1}^{N} Z_{0t}Z_{1t}^T$$

$$M_{11} = \sum_{t=1}^{N} Z_{1t}Z_{lt}^T$$

$$M_{k1} = \sum_{t=1}^{N} Z_{kt}Z_{1t}^T$$

Thus, the parameter Γ can be represented in terms of Π by using covariance matrices from data:

$$\Gamma = M_{01}M_{11}^{-1} - \Pi M_{k1}M_{11}^{-1} \tag{4.11}$$

Substituting (4.11) for Γ and after rearranging it, equation ((4.10)) becomes:

$$e_t = R_{0t} - \Pi R_{kt} \tag{4.12}$$

where

$$R_{0t} = Z_{0t} - M_{01}M_{11}^{-1}Z_{1t}$$

and

$$R_{kt} = Z_{kt} - M_{k1}M_{11}^{-1}Z_{1t}$$

From the definition of matrices M_{01}, M_{11}, and M_{k1}, it can be seen that R_{0t} is the residual regressing Z_{0t} onto Z_{1t} and R_{kt} is the residual regressing Z_{kt} onto Z_{1t}. Based on this new formulation of the noise term in equation (4.12), the optimal solution for parameter Π can be determined through least square estimation:

$$S_{0k} = \Pi S_{kk} \tag{4.13}$$

$$\Pi = S_{0k}S_{kk}^{-1} \tag{4.14}$$

where

$$S_{0k} = \sum_{t=1}^{N} R_{0t}R_{kt}^T$$

and

$$S_{kk} = \sum_{t=1}^{N} R_{kt}R_{kt}^T$$

Similarly, based on the objective function in equation (4.8), the optimal solution for Λ can be obtained as:

$$\Lambda = \sum_{t=1}^{T} e_t e_t^T \tag{4.15}$$

Substituting the solution for Π into equation (4.12), the final expression of Λ can be written as:

$$\Lambda = \sum_{t=1}^{T}(R_{0t} - S_{0k}S_{kk}^{-1}R_{kt})(R_{0t} - S_{0k}S_{kk}^{-1}R_{kt})^T \tag{4.16}$$

and simplified as:

$$\Lambda = S_{00} - S_{0k}S_{kk}^{-1}S_{k0} \tag{4.17}$$

Now, we can use the assumption in equation (3.34) to determine the parameter β. After substituting (3.34) into (4.14), the optimal solution of Π, α and Λ can be determined for a fixed β:

$$\Pi(\beta) = S_{0k}\beta(\beta^T S_{kk}\beta)^{-1}\beta^T \tag{4.18}$$

$$\Lambda(\beta) = S_{00} - S_{0k}\beta(\beta^T S_{kk}\beta)^{-1}\beta^T S_{k0} \tag{4.19}$$

Substituting the expressions above into the objective function in equation ((4.8)), we can simplify the likelihood function by using the property of trace:

$$\begin{aligned} L(\beta) \propto & |\Lambda|^{-P/2}\exp\left(-\frac{1}{2}\sum_{t=1}^{N}(R_{0t} - \Pi R_{kt})^T\Lambda^{-1}(R_{0t} - \Pi R_{kt})\right) \\ &= |\Lambda|^{-P/2}\exp\left(-\frac{1}{2}tr\big(\Lambda^{-1}\sum_{t=1}^{N}(R_{0t} - \Pi R_{kt})(R_{0t} - \Pi R_{kt})^T\big)\right) \\ &= |\Lambda|^{-P/2}\exp\left(-\frac{1}{2}tr\Big(\Lambda^{-1}(S_{00} - S_{0k}\beta(\beta^T S_{kk}\beta)^{-1}\beta^T S_{k0})\Big)\right) \\ &= |\Lambda|^{-P/2} \end{aligned} \tag{4.20}$$

From the equation, it can be seen that maximizing the likelihood with respect to β is equivalent to maximizing the determinant Λ. Johansen shows how MLE of β^T is equivalent to solving the eigenvalue equation [†]:

$$|\lambda S_{kk} - S_{k0}S_{00}^{-1}S_{0k}| = 0 \tag{4.21}$$

[†] $max_\beta|S_{00} - S_{0k}\beta(\beta^T S_{kk}\beta)^{-1}\beta^T S_{k0}| \Longleftrightarrow max_\beta \frac{|S_{00}||\beta^T(S_{00} - S_{k0}(S_{00})^{-1}S_{0k})|\beta}{|\beta^T S_{kk}\beta|}$

where

$$S_{k0} = S_{0k}^T$$

From equation ((4.21)), it can be observed that the solution to β is simply solving an eigenvalue decomposition problem, where the eigenvector corresponding to the highest eigenvalue is the most cointegrated feature (i.e. the most stationary feature) and the eigenvectors can be ordered based on their level of stationarity. Based on the conintegrated features, the dynamic relationship within the nonstationary variables is expressed as:

$$Z = \beta^T X \tag{4.22}$$

where $\beta(P \times r) = [\beta_1, \beta_2, ..., \beta_r]$ and Z is stationary.

Assumptions and the associated equations are summarized in Figure 4.1.

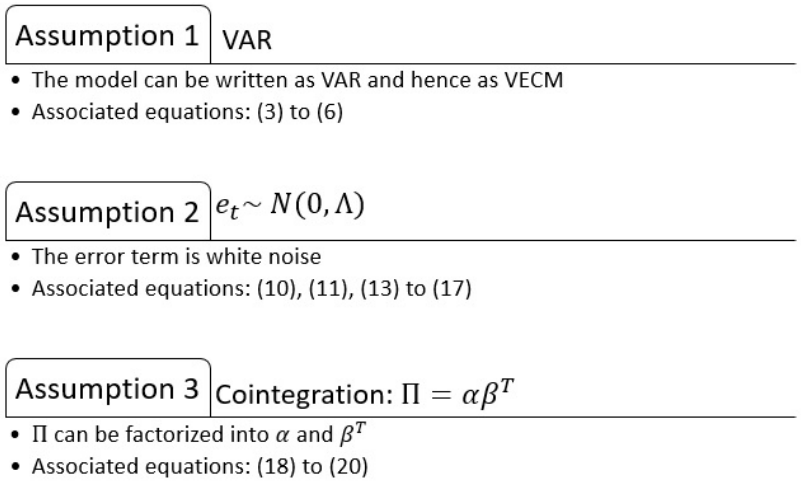

Figure 4.1: Summary of CA

4.4 Proposed model: Residual CA (r-CA)

CA has been studied in chemical engineering for process monitoring and fault diagnosis for nonstationary processes [32, 33, 34]. They use projection matrix, $\beta = [\beta_1, ..., \beta_r]$, to study the stationary features within the nonstationary processes and

the stationary features are ordered based on the descending order of their associated eigenvalues. Theoretically, in traditional CA, the rank of the system should equal the number of stationary features. Hence in the maximum eigenvalue test proposed by Johansen, it essentially looks at the number of nonzero eigenvalues to determine the number of cointegrated vectors. In reality, however, it is most likely to have a full rank system due to measurement noise: by assuming that measurement noise is independent of each other, we cannot have perfect collinearity. Consequently, the system will have full rank and the number of nonstationary features will be represented with eigenvalues close to zero. Therefore, if there exists a sharp decrease in the eigenvalues, we commonly associate features corresponding to those eigenvalues after the sharp jump as the features that describe the nonstationarity in the system. Based on this consideration, the features are ordered in a manner where stationary features come first in the principal space (r < P) and the latter features in the residual space $(P - r)$ are therefore by definition the nonstationary features. In addition to the stationary features, understanding the nonstationary features is important in chemical engineering as it can explain what causes the system to be nonstationary, such as fouling buildup, equipment aging, and different engineering modes. Extracting the nonstationary features that are unique in the system and cannot become stationary through co-integration with other variables allow us to understand the dynamics that exist over time. For these reasons, rather than using CA to study the principal space of CA, we instead focus on the residual space or nonstationary features and call this method as residual cointegration analysis (r-CA).

In statistical analysis, residual space has been widely used such as independent component analysis (ICA) [36] and slow feature analysis (SFA) [25]. ICA essentially tries to decompose the system into statistically independent features. By assuming that the desired features are non-Gaussian and that noise is Gaussian, ICA obtains the desired features by removing Gaussian noise from the system. After Gaussian noise is removed, the left-over features in the "residual space" are considered to be

important. SFA is another method that extracts slowly varying features from the quickly varying system. The SFA algorithm achieves its objective by first extracting the fastest varying feature from the system. The latter features in the residual space are thus the slowly varying features, which are the desired features.

In addition to ICA and SFA, knowledge about the most nonstationary features is valuable in chemical engineering and features in the residual space have been used in many applications. The most common usage of residual space is SPE (squared prediction error) statistics in process monitoring methods [36, 33, 37, 38, 34, 32, 4, 39]. For the residual space of cointegration analysis method, the nonstationary features are valuable since they can reveal the information about the undesired attributes within the chemical process. Examples of the important nonstationary features in chemical processes include: fouling buildup in heat exchanging equipment, residue buildup in reactor walls and corrosion. In particular, understanding the fouling feature is desirable as it can be used to plan the fouling removal effectively.

Like many statistical methods such as PCA, ICA, and SFA, it is important to normalize the inputs for r-CA to avoid misleading results while extracting features and give equal chances to each process variable. If the system is not normalized, it may distribute uneven and incorrect weights in the loading matrix when the eigenvalues are being determined. The number of latent features used to represent the system can be determined by plotting the eigenvalues associated with the features. Since the proposed model essentially determines the desired features based on eigenvalue decomposition, the percentage of these eigenvalues can tell the desired information. The algorithm for the r-CA is as follows:

1. Given a nonstationary system, write in vector error correction (VEC) form as shown in Equation (4.5), where Π in the equation can further be divided into:

$$\Pi = \alpha\beta^T$$

2. Determine the β by maximizing the likelihood function in Equation (4.20),

which is equivalent to maximizing the determinant Λ

3. Assuming that the measurement noise is independent of each another, Π will have full rank and no eigenvalues will equal zero. However, the stationary features will be associated with large eigenvalues (principal space) and the non-stationary features will be associated with eigenvalues close to zero (residual space).

4. Focusing on the residual space, the nonstationary features can be mapped from the original system if we allow β to involve nonstationary features (residual space)

Application of the r-CA to predict fouling buildup in once-through steam generator (OTSG) tubes will be discussed in Section 4.6. To verify if the reverse order of eigenvalues can be used to extract the most nonstationary feature, numerical simulation was conducted first in Section 4.5.

4.5 Numerical simulations and property investigations

To validate if the r-CA can truly capture the most nonstationary feature(s) within the system, numerical simulation was conducted. In section 4.5.1, r-CA was used to extract the most nonstationary feature(s) within a system. In section 4.5.2, the effect of the following items on r-CA was also investigated:

- Effect of the number of samples (N): when performing any statistical analysis, sample size plays an important role. To study the effect of sample size in r-CA model, sample size was varied (N=50, N=100, N=1000).

- Effect of assuming wrong VAR order: to determine if the specified order (lag length) is valid, AIC (Akaike information criterion) and/or BIC (Bayesian information criterion) can be calculated. Once the order is determined, it should

be ensured that the error term is white noise. To understand the significance of wrong order assumption, multiple numerical simulations with wrong CA order specified were performed. Orders in Table 4.4 and 4.5 are validated using AIC and BIC.

- Effect of $X \sim I(1)$ assumption: if a nonstationary time series X becomes stationary by differencing it d times, X is then integrated by order d and it is written as $X \sim I(d)$. One assumption made for CA and in r-CA is that first order difference of the system is stationary, which is denoted as $X \sim I(1)$. To evaluate the sensitivity of this assumption, r-CA was used to extract second order nonstationary features to see its performance for $X_t \sim I(d > 1)$.

4.5.1 Case 1: Nonstationary feature extraction

Numerical simulation has been conducted to verify if the r-CA can extract the nonstationary feature(s) from a given dataset. 50 Monte Carlo simulations were performed to randomly simulate features with different dynamics. Table 4.1 summarizes the models used to generate the features: three models were used to generate stationary features and one model was used to generate the nonstationary feature (random walk). The objective of this numerical simulation is to extract the most nonstationary feature. To study if r-CA can extract more than one nonstationary feature, the third model in Table 4.1 was replaced with another random walk model later on so that the system contained two first order random walks. For comparison, PCA (principal component analysis) was used to see if it can also extract the most nonstationary feature. PCA is a widely used unsupervised learning method where feature extraction is based on variance. The first feature extracted from PCA (first principal component) is the direction where the variance of the system is maximized. The second principal component is orthogonal to the first principal component and captures the direction where the variance is smaller than the first but larger than the latter principal components. PCA was chosen as a benchmark for comparison since it is one of the most

widely used machine learning methods. Also, although PCA should be applied to stationary data, in theory, the first principal component shows the direction where the data varies the most. Hence it should capture the nonstationary feature to some extent.

Table 4.1: Models used to generate features

	1st order	2nd order	3rd order
model 1	$\frac{1}{1-0.5z^{-1}}e_t$	$\frac{1}{1-0.7z^{-1}+0.06z^{-2}}e_t$	$\frac{1}{1-1.3z^{-1}+0.39z^{-2}-0.027z^{-3}}e_t$
model 2	$\frac{1}{1-0.9z^{-1}}e_t$	$\frac{1}{1+0.5z^{-1}+0.06z^{-2}}e_t$	$\frac{1}{1+0.4z^{-1}-0.39z^{-2}-0.126z^{-3}}e_t$
model 3	$\frac{1}{1-0.7z^{-1}}e_t$	$\frac{1}{1-1.3z^{-1}+0.42z^{-2}}e_t$	$\frac{1}{1+1.4z^{-1}+0.41z^{-2}-0.056z^{-3}}e_t$
model 4	$\frac{1}{1-z^{-1}}e_t$	$\frac{1}{1-z^{-1}}e_t$	$\frac{1}{1-z^{-1}}e_t$

$$\mathbf{X} = \mathbf{Z} \times \mathbf{P} \tag{4.23}$$

where for a system with single nonstationary feature

$$\mathbf{Z} = [\text{model 1, model 2, model 3, model 4}]$$

and for a system with two nonstationary features

$$\mathbf{Z} = [\text{model 1, model 2, model 4, model 4}]$$

The $\mathbf{P}$ in Equation (4.23) is randomized uniformly by Monte Carlo simulations. To visually present the performance of the r-CA, one Monte Carlo simulation result was randomly selected and the results are shown in Figures 4.3 and 4.4. To compare the performance, the correlation coefficient was averaged, and the distribution of the correlation coefficients obtained from each Monte Carlo simulation is also plotted. To validate that $\mathbf{X}$ is nonstationary, the ADF test can be done. To determine how many cointegrated relationships exist in $\mathbf{X}$, Johansen test is performed.

Figure 4.2 shows the time trend of $\mathbf{X}$ and $\Delta\mathbf{X}$ from one of the simulations. Table 4.2 summarizes the average correlation for a single nonstationary feature system, where the number of samples was set as 100. The specified order in the table represents the order of r-CA. From the table, it can be seen that r-CA was able to capture

the nonstationary feature better than PCA. Figure 4.3 compares the extracted nonstationary features from PCA and r-CA with the original nonstationary feature which is a random walk stochastic process (model 4). The figure also clearly shows how r-CA is superior to PCA at capturing the nonstationary feature. For the system with two nonstationary features, the correlation was calculated based on the vector regressed by the two nonstationary features. Figure 4.4 shows the feature extracted from r-CA and the surface of the two nonstationary features. From figure 4.4, it can be seen that the feature extracted using r-CA has less error than PCA. This is more evident in the correlation comparison in table 4.3. In summary, r-CA was able to extract the nonstationary feature in both where the system has one nonstationary feature and where the system has two nonstationary features.

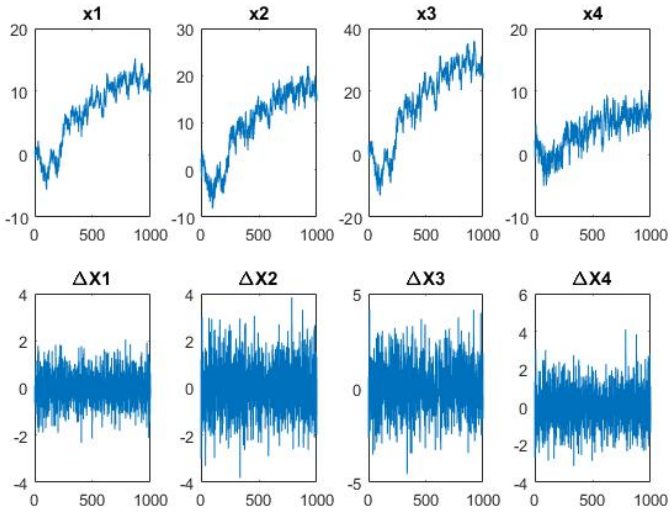

Figure 4.2: Top:X, bottom:ΔX, N=1000, order=2 (time trend)

Table 4.2: Correlation comparison between PCA and r-CA. N=100, system with single nonstationary feature

	Order = 1	**Order = 2**	**Order = 3**
r-CA	1.0	0.99	0.94
PCA	0.87	0.82	0.62

To compare the performance from all 50 Monte Carlo simulations, correlation coefficients from all the simulations are plotted as a histogram. The top plot in Figure

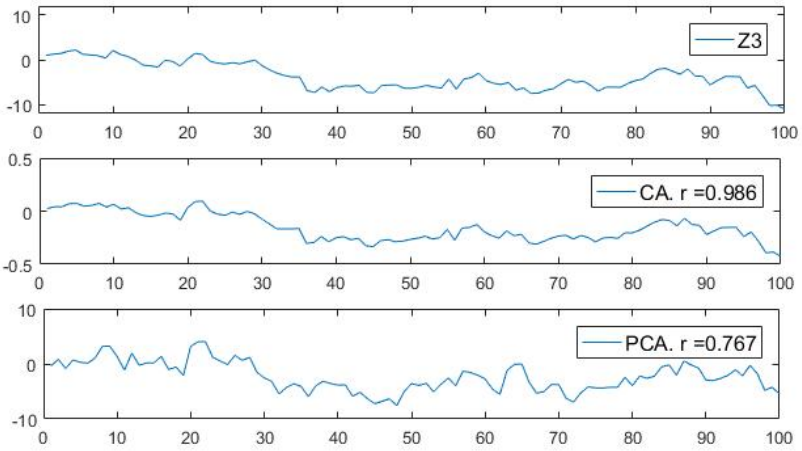

Figure 4.3: Top: Z3=$\frac{1}{1-z^{-1}}e_t$, middle: feature extracted from r-CA, bottom: feature extracted from PCA (time trend)

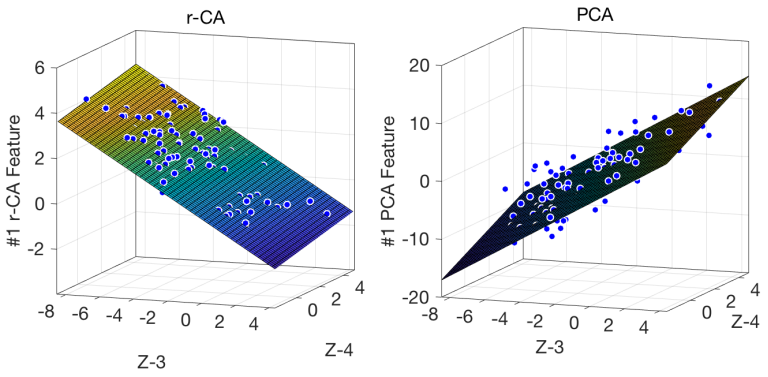

Figure 4.4: Numerical simulation: System order = 1; Left: feature extracted from r-CA; Right: feature extracted from PCA

4.5 shows the distribution of the correlation coefficient from r-CA and the bottom plot shows the distribution of the correlation coefficient extracted from PCA. From the figure, it is evident that r-CA is superior at extracting the nonstationary feature. In particular, it can be seen that all 50 simulations gave very similar correlation coefficient results for r-CA, indicating that r-CA indeed gave consistent results to extract the nonstationary feature. Similarly, figure 4.6 compares the correlation coefficient distribution from the system with 2 nonstationary features. This figure reiterates how r-CA is better than PCA at capturing the nonstationary feature.

Table 4.3: Correlation comparison between PCA and r-CA. N=100, system with 2 nonstationary features

	Order = 1	**Order = 2**	**Order = 3**
r-CA	0.99	0.99	0.93
PCA	0.87	0.91	0.77

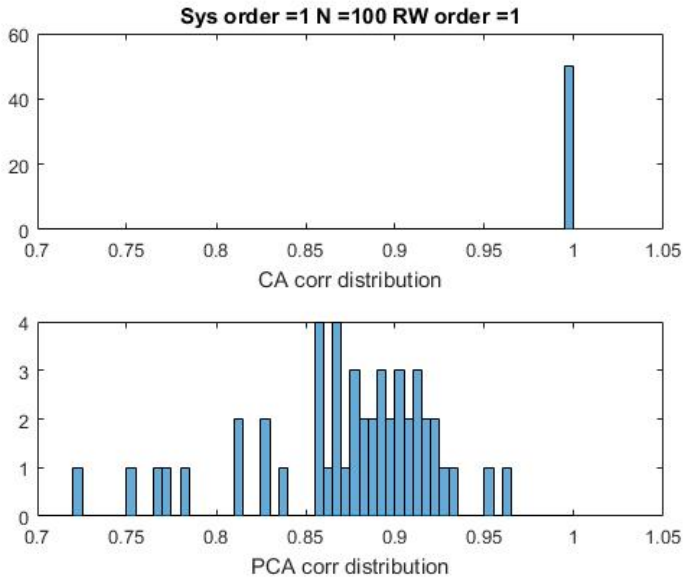

Figure 4.5: Correlation coefficient distribution; 1 nonstationary variable

4.5.2 Case 2: Effects of different assumptions

To study the different effects from the different assumptions, 50 Monte Carlo simulations were again performed to randomly generate different loading matrices and r-CA was used to extract the most nonstationary feature (random walk, model 4). To compare the performance, the correlation coefficient between the extracted feature and random walk data was calculated for each simulation. The average values of 50 simulations are summarized in Table 4.4 for the effect of sample sizes, Table 4.5 shows the effect of $X_t \sim I(d > 1)$, and Table 4.6 shows the effect of wrong CA order. The number of the nonstationary feature was set as one.

Sampling size: Except for the 3rd order with N=50, the feature extracted from r-CA matches very well with the random walk in terms of correlation coefficients regardless of the sample size. On the other hand, the performance of PCA was more

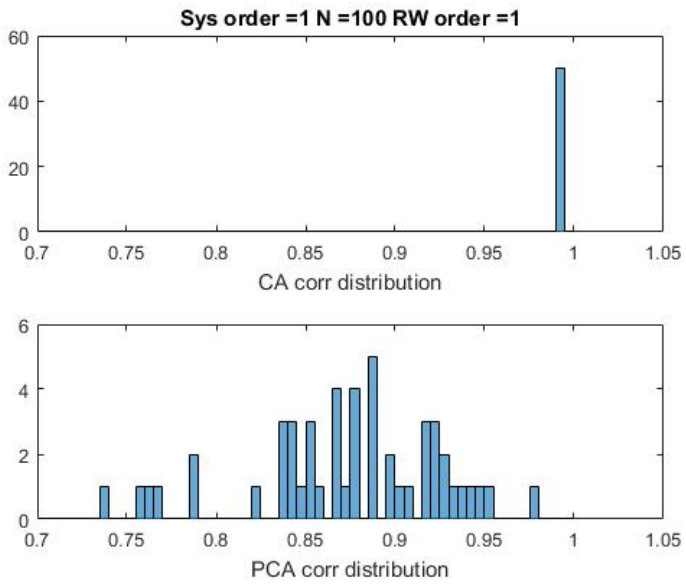

Figure 4.6: Correlation coefficient distribution; 2 nonstationary variables

dependent on the sample size. Although the performance of r-CA does increase with the sampling rate, the incremental improvement does not behave in the same way as PCA.

Higher order random walk: For extracting the most nonstationary feature using r-CA, the $X \sim I(1)$ assumption does not always have a significant impact on the performance. Although r-CA assumes that the system is first order stationary, results in terms of the correlation coefficient for second-order stationary systems also appear to be acceptable. Even if the original data have higher order nonstationarity, using the proposed method give acceptable result in terms of correlation coefficient so this method could be considered for higher order nonstationary data.

r-CA lag order: The bold numbers in Table 4.6 indicate the correct order number. Except for the 3rd order system, r-CA outperforms PCA, even with the wrong order. Regardless, the correct order assumption does give better results so using AIC and/or BIC to determine the correct order can improve the results. Orders in Table 4.4 and 4.5 have been validated using AIC and BIC.

Table 4.4: Effect of sampling size on r-CA

	Sampling size					
	N=50		N=100		N=1000	
System order	PCA	r-CA	PCA	r-CA	PCA	r-CA
1st order	0.61	0.96	0.87	1.0	0.98	1.0
2nd order	0.45	0.81	0.82	0.99	0.98	1.0
3rd order	0.14	0.58	0.62	0.94	0.95	1.0

Table 4.5: Effect of $X \sim I(d > 1)$. N=100

	Random walk order			
	1st order		2nd order	
System order	PCA	r-CA	PCA	r-CA
1st order	0.87	1.0	1.0	0.47
2nd order	0.82	0.99	1.0	0.99
3rd order	0.62	0.94	1.0	0.92

4.6 Industrial application results

In this section, r-CA was applied on industrial data along with PCA to predict fouling buildup in steam generators. Fouling buildup in once-through steam generator (OTSG) tubes were used as a case study. Fouling in heat exchanging equipment such as OTSGs has been a long-standing problem in chemical engineering. Due to its complexity, however, building predictive models for fouling buildup is very [40, 30, 41]. It should be stressed however that the method to be developed is not restricted to OTSGs: it can be applied to other steam generators or heat exchangers as well.

4.6.1 Process and data description

Steam-assisted gravity drainage (SAGD) is a type of in-situ methods where bitumen underground is extracted by injecting steam into the reservoir. During the steam generation process, steam generators accumulate impurities (scales/fouling). Figure 4.7 summarizes the general water cycle within the SAGD process.

In a typical industrial scale process, scales/fouling are removed physically when the system is offline by a process called "pigging." By understanding/ knowing how much impurity is formed inside the tubes, shutdowns can be planned more effectively

Table 4.6: Effect of lag order on r-CA. N=100

	r-CA order					
	1st order		2nd order	3rd order	5th order	7th order
System order	PCA	r-CA	r-CA	r-CA	r-CA	r-CA
1st order	0.87	**1.0**	0.99	0.99	0.97	0.99
2nd order	0.82	0.98	**0.99**	0.98	0.98	0.94
3rd order	0.62	0.85	0.95	**0.94**	0.56	0.56

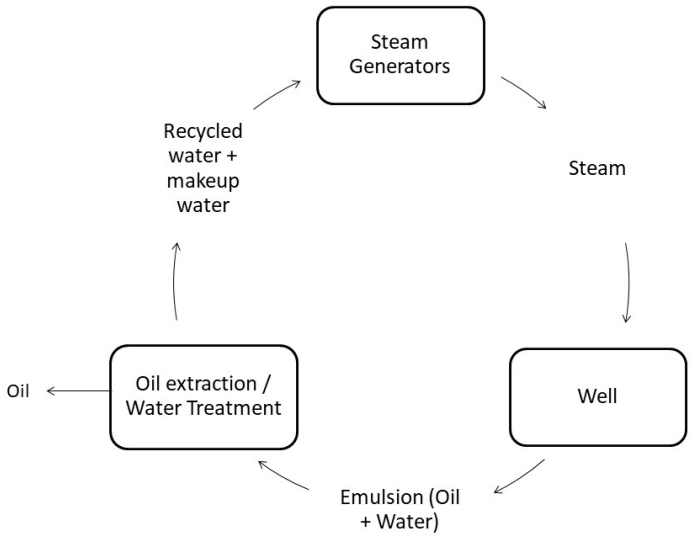

Figure 4.7: Water cycle within SAGD process

to schedule fouling removal. Pigging is done typically every 100 - 500 days for SAGD operations. In other words, in order to have a sufficient amount of pigging data and develop a predictive model based on supervised learning methods, we need to collect data for decades. As an alternative, lab experiments can be done to obtain an adequate amount of fouling data, and then perform regression analysis. However, it will be difficult to scale up the exact same model in large scale plants and achieve the same results as the lab experiment. For these reasons, the unsupervised learning method was used to extract features from commonly measured process variables that may represent fouling. Available fast rate process variables for OTSGs are listed in Table 4.7. The sampling interval was 10 minutes for all variables.

To effectively apply r-CA to extract informative information, a two-layer system

Table 4.7: Available process variables

Variable	Unit
Temperature of boiler feed water	
Pressure of boiler feed water	kPa
Boiler feed water flow rate of individual passes	t/hr
Outlet temperature of recombined stream	
Outlet pressure of recombined stream	kPa
Differential pressure of individual passes	kPa
Fuel gas flowrate	kg/hr

is proposed. In the first layer, PCA was applied to extract informative features with slow varying fluctuations, thus removing large fluctuation part of data. This is based on the understanding that scaling is a slowly increasing process. In the second layer, r-CA was used to extract the most nonstationary trend from the slowly varying features, which captures a slowly increasing trend. Should r-CA be applied directly to the raw measurements, the extracted features can be flooded with noise/disturbance due to common process upset or operating condition changes. In order to extract the most informative feature as much as possible, PCA was essentially used as a filtering method. PCA is one of the most widely used unsupervised methods. PCA extracts feature based on variance. In other words, features that capture the most variance of the system are in the PCA principal space. Therefore, applying PCA on the data to remove the most dominant variation features can remove undesired fluctuations. Another advantage of using PCA prior to using r-CA is that since r-CA uses the least squares for regression, applying PCA will help with the regression by making the regressor orthogonal to each other. We can, therefore, avoid the issues of collinearity in regression. The algorithm for the industrial application of the modified CA is summarized below:

1. Identify the informative nonstationary variables by unit root tests.

2. Apply PCA on the training dataset where high variance/fluctuations represent disturbances/noise. In this way, first principal component typically contains the

undesirable features (disturbance/noise). Use the rest principal components will be retained to capture the slowly-increasing fouling.

3. Apply r-CA to the latent variables. Since undesired fluctuations have been removed via PCA, the extracted nonstationary feature contains the desired dynamic information.

In summary, in the first layer, PCA will be applied where the data has large fluctuations, and the first principal component will be removed and the rest p-1 principal components are retained to capture slowly-increasing trends. In the second layer, r-CA will be applied to extract the slowly accumulating fouling in the steam generators.

Before applying r-CA, the stationary variables should be separated from the non-stationary variables since stationary variables may mask the cointegrated relationship between the nonstationary variables. As a result, when r-CA is applied to a system with stationary variables included, the obtained features may not represent the relationship accurately. Nonstationary variables can be detected by unit root tests such as the Augmented Dickey-Fuller test (ADF) [32]. For the industrial application of r-CA, relevant process variables were selected based on prior knowledge. Figure 4.8 summarizes the proposed method:

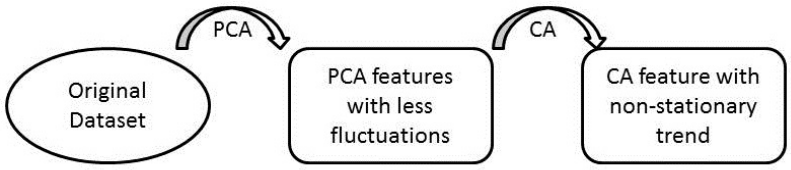

Figure 4.8: PCA-rCA algorithm

For this case study, results from three OTSGs are demonstrated. For proprietary reason, the actual amount of scales collected from each pigging incident is not disclosed and all data is normalized.

4.6.2 Results and discussion

Since the thorough first-principle knowledge about fouling buildup in OTSGs tubes is still an unsolved problem, the r-CA order was determined by numerical experiments. Figure 4.9 shows the extracted fouling buildup feature. The blue trend is the extracted feature, the pink vertical lines are shutdowns, and red vertical lines are the shutdowns where pigging was performed. From the figure, it can be observed how the trends are generally increased and suddenly drops after fouling/scaling buildup inside the tubes is removed. The sudden drop/fluctuations in the latter part of the process are caused by the step changes and/or shutdowns in input variables. Table 4.8 summarizes the percent error between the estimated fouling buildup and the actual amount of fouling buildups measured after each pigging. From the table, it can be seen that the estimated fouling buildup amount is comparable with the actual fouling buildup amount, indicating a promising result from the proposed method.

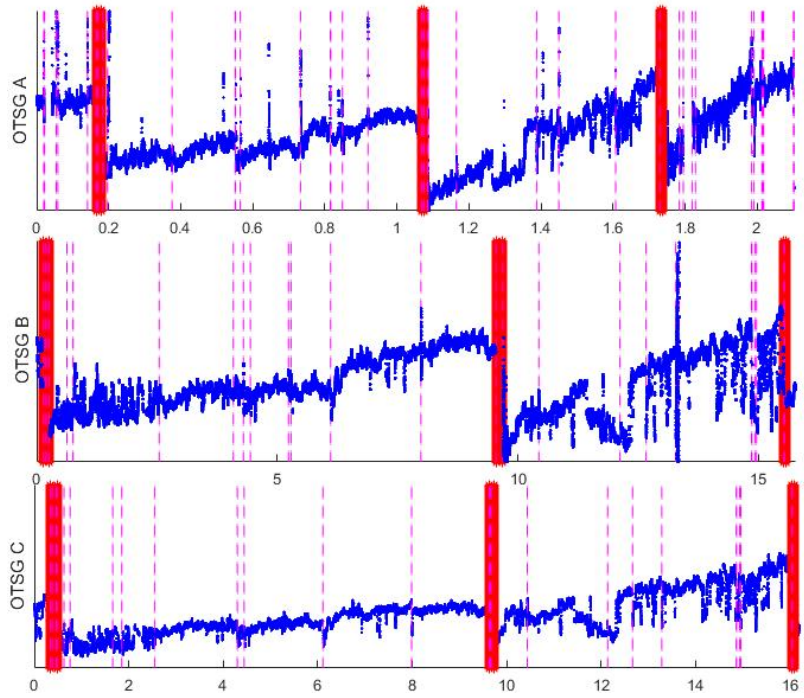

Figure 4.9: PCA-rCA results

Table 4.8: prediction percent error

	1	2
OTSG A	0.02	0.2
OTSG B	0.64	16.7
OTSG C	15.6	8.3

4.7 Conclusion

In this paper, a novel application of CA is proposed to extract the most nonstationary feature in data. Extracting the nonstationary feature is valuable in chemical processes since it can be used to analyze accumulation in the system such as fouling buildup in heat exchanging equipment, residue buildups in reactors and so on. The performance of the proposed method has been validated in various numerical simulations. Associated industrial application to monitor fouling buildup in OTSG tubes has also been presented. From the numerical simulations, it can be observed that the proposed method is capable of extracting the nonstationary feature from data. From the industrial application example, the proposed method was able to predict the general increasing trend of fouling buildup in OTSG tubes with approximately 2- 17% error from the fouling accumulation obtained by the actual amount scalings measured after the pigging. It can thus be concluded that the method presented in this study is promising in monitoring fouling accumulation behaviour from process data.

Chapter 5

Conclusion

5.1 Conclusion

This chapter summarizes the work and recommends future work for potential improvements.

5.1.1 Summary

In this work, the focus was on applying statistical methods to develop predictive models that are difficult to measure with instrumentation. While applying chemical engineering fundamentals, we also investigated data-driven methods to improve the predictability of quality variables in chemical processes such as steam quality and fouling accumulation in steam generators. Contribution of this thesis is summarized below:

In Chapter 2, the grey box model based soft sensor was developed to predict steam quality in a steam generator for SAGD (steam-assisted gravity drainage) process. The core model structure was based on energy balance, and model parameter identification and statistical techniques were used to enhance the predictive power. As the steam generator process is the most energy-intensive part of the SAGD process, knowing the steam quality with high accuracy can help with steam generator optimization. The developed model was then compared with an existing steam quality analyzer to show the improvements in detail.

To further improve the steam quality soft sensor performance, feature extraction methods were explored in Chapter 3. In this chapter, it was attempted to extract features from readily available process variables and also the online bias trends. The motivation of extracting features from the bias term was because the bias terms from the different passes in the steam generator showed a similar trend, which is a strong indication that the online bias correction may be used to correct the missing process information. In addition to improving the performance of the soft sensor, the possibility of developing an infinite step ahead predictor was also investigated: if the extracted features significantly improve the performance, a soft sensor that is not dependent on online bias update could be developed, which results in less manpower by having fewer manual samples. PCA (principal component analysis), SFA (slow feature analysis), and CA (cointegration analysis) were selected as feature extraction methods to extract informative features from the bias terms, which were mapped back to obtain appropriate coefficients for the process variables, which were used as additional inputs for the soft sensor. When compared with the original steam quality soft sensor, the additional features both improved correlation and reduced mean absolute error.

In Chapter 4, feature extraction methods were further investigated to solve the difficult problem of fouling accumulation in heat exchanger tubes. Fouling has been a long time problem in chemical engineering for many years. Unfortunately, due to the complicated fouling buildup phenomenon, it is very difficult to develop a predictive model solely based on chemical engineering fundamentals. To solve this problem, two-layer feature extraction method was used, where PCA (principal component analysis) was used for the first layer for de-noising purpose and residual CA (cointegration analysis) was used to extract the nonstationary trend. The results showed that the proposed method was able to capture the slowly accumulating trend and could predict the general fouling buildup trend, given only the readily available process variables in the industry.

5.1.2 Future Work

In this thesis, various feature extraction methods were explored and modified to improve the predictability of steam quality soft sensor and to predict fouling buildup in steam generators. To further improve the performance, some recommendations are listed below:

1. Apply the stochastic method in chapter 3: in chapter 3, feature extraction methods were applied to further improve the steam quality soft sensor. The chosen feature extraction methods, however, were all deterministic. To further improve this method, it is recommended to explore probabilistic feature extraction methods. Compared to the deterministic methods, probabilistic methods have the advantage of being able to deal with missing process data and dealing with the multimode process. In chemical engineering, due to field instrumentation and/or server connection issues, missing process information is a common problem in chemical processes. In addition, continuous processes typically operate in multiple modes. Since probabilistic methods can handle missing information and the multimode process, the proposed method in Chapter 3 can be further improved.

2. Develop probabilistic / Bayesian CA for fouling detection: two layer feature extraction method using PCA for de-noising and CA for extracting slow accumulating features were used to predict fouling buildup. For further improvement, it is recommended to explore stochastic methods such as probabilistic CA or Bayesian CA. Fouling is a very slow process and requires to be monitored for a long duration, typically in years. Consequently, fouling predictive models should ideally be able to deal with multi-mode processes and missing information: since monitoring fouling accumulation is done is a long time span, it most likely suffers from missing information from aging instrumentation and multimode continuous processes. By applying stochastic CA method on monitoring

fouling accumulation, it is expected to achieve better performance

3. Apply el-1 filtering method to improve the fouling detection method: el-1 filtering method is a modification of el-2 filtering method. Typically, fouling is removed from steam generator tubes by a process called pigging. If multiple pigging is performed, fouling accumulation will portray a piece-wise trend, where the trend is slowly accumulating due to the fouling buildup, then suddenly drops as a result of pigging. Therefore, by applying el-1 filtering method in addition to CA, the predictability of fouling accumulation may improve.

Bibliography

[1] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[2] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[3] Robert F Engle and Clive WJ Granger. Co-integration and error correction: representation, estimation, and testing. *Econometrica: journal of the Econometric Society*, pages 251–276, 1987.

[4] Qian Chen, Uwe Kruger, and Andrew YT Leung. Cointegration testing method for monitoring nonstationary processes. *Industrial & Engineering Chemistry Research*, 48(7):3533–3543, 2009.

[5] Søren Johansen and Katarina Juselius. Maximum likelihood estimation and inference on cointegration—with applications to the demand for money. *Oxford Bulletin of Economics and statistics*, 52(2):169–210, 1990.

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

[7] Shima Khatibisepehr, Biao Huang, and Swanand Khare. Design of inferential sensors in the process industry: A review of bayesian methods. *Journal of Process Control*, 23(10):1575–1596, 2013.

[8] Omid Ashrafi, Philippe Navarri, Robin Hughes, and Dennis Lu. Heat recovery optimization in a steam-assisted gravity drainage (sagd) plant. *Energy*, 111:981 – 990, 2016.

[9] Li Xie, Yu Zhao, Daniel Aziz, Xing Jin, Litao Geng, Errol Goberdhansingh, Fei Qi, and Biao Huang. Soft sensors for online steam quality measurements of otsgs. *Journal of Process Control*, 23(7):990–1000, 2013.

[10] K Nihed, K KTahar, and BB Ammar. A thermal analysis and optimization of a combined cycle by several technologies. *Am J Energy Res*, 2:35–41, 2014.

[11] Kwan-Woong Gwak and Wisup Bae. A review of steam generation for in-situ oil sands projects. *Geosystem Engineering*, 13(3):111–118, 2010.

[12] Frank Incropera and David DeWitt. Introduction to heat transfer. 1985.

[13] JF Kaiser and WA Reed. Data smoothing using low-pass digital filters. *Review of Scientific Instruments*, 48(11):1447–1457, 1977.

[14] Marc Antonini, Michel Barlaud, Pierre Mathieu, and Ingrid Daubechies. Image coding using wavelet transform. *IEEE Transactions on image processing*, 1(2):205–220, 1992.

[15] Jin Lin and MJ Zuo. Gearbox fault diagnosis using adaptive wavelet filter. *Mechanical systems and signal processing*, 17(6):1259–1269, 2003.

[16] Yuri AW Shardt. *Statistics for Chemical and Process Engineers*. Springer, 2015.

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

[18] Shengjing Mu, Yingzhi Zeng, Ruilan Liu, Ping Wu, Hongye Su, and Jian Chu. Online dual updating with recursive pls model and its application in predicting crystal size of purified terephthalic acid (pta) process. *Journal of Process Control*, 16(6):557–566, 2006.

[19] Wangdong Ni, Ke Wang, Tao Chen, Wun Jern Ng, and Soon Keat Tan. Gpr model with signal preprocessing and bias update for dynamic processes modeling. *Control Engineering Practice*, 20(12):1281–1292, 2012.

[20] Li Xie, Yu Zhao, Daniel Aziz, Xing Jin, Litao Geng, Errol Goberdhansingh, Fei Qi, and Biao Huang. Soft sensors for online steam quality measurements of otsgs. *Journal of Process Control*, 23(7):990 – 1000, 2013.

[21] Yuri AW Shardt and Biao Huang. Tuning a soft sensor's bias update term. 1. the open-loop case. *Industrial & Engineering Chemistry Research*, 51(13):4958–4967, 2012.

[22] Yuri AW Shardt and Biao Huang. Tuning a soft sensor's bias update term. 2. the closed-loop case. *Industrial & Engineering Chemistry Research*, 51(13):4968–4981, 2012.

[23] Samina Khalid, Tehmina Khalil, and Shamila Nasreen. A survey of feature selection and feature extraction techniques in machine learning. In *2014 Science and Information Conference*, pages 372–378. IEEE, 2014.

[24] Agnan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, 2018.

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

[26] Chao Shang, Fan Yang, Xinqing Gao, and Dexian Huang. Extracting latent dynamics from process data for quality prediction and performance assessment via slow feature regression. In *2015 American Control Conference (ACC)*, pages 912–917. IEEE, 2015.

[27] Petr Kadlec, Ratko Grbić, and Bogdan Gabrys. Review of adaptation mechanisms for data-driven soft sensors. *Computers & Chemical Engineering*, 35(1):1 – 24, 2011.

[28] Ian T Jolliffe. Principal component analysis and factor analysis. In *Principal component analysis*, pages 115 –128. Springer, 1986.

[29] Svante Wold, Michael Sjöström, and Lennart Eriksson. Pls-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, 58(2):109 –130, 2001.

[30] M Kreider and G White. Thermal performance degradation and heat-transfer fouling. In *Steam Generators for Nuclear Power Plants*, pages 365–404. Elsevier, 2017.

[31] Takuya Kuwahara, Christianto Wibowo, Akiya Kuboyama, Mitsutaka Nakamura, and Yoshihiro Yamane. Fouling monitoring in thermosiphon reboiler. *Heat Transfer Engineering*, 36(7-8):780–786, 2015.

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

[33] 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.

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

[35] Søren Johansen. Statistical analysis of cointegration vectors. *Journal of economic dynamics and control*, 12(2-3):231–254, 1988.

[36] Jong-Min Lee, ChangKyoo Yoo, and In-Beum Lee. Statistical process monitoring with independent component analysis. *Journal of Process Control*, 14(5):467–485, 2004.

[37] Chunhui Zhao and Furong Gao. Critical-to-fault-degradation variable analysis and direction extraction for online fault prognostic. *IEEE Transactions on Control Systems Technology*, 25(3):842–854, 2017.

[38] Wenqing Li, Chunhui Zhao, and Furong Gao. Linearity evaluation and variable subset partition based hierarchical process modeling and monitoring. *IEEE Transactions on Industrial Electronics*, 65(3):2683–2692, 2018.

[39] Rahul Raveendran, Hariprasad Kodamana, and Biao Huang. Process monitoring using a generalized probabilistic linear latent variable model. *Automatica*, 96:73–83, 2018.

[40] D Ian Wilson, Edward M Ishiyama, and Graham T Polley. Twenty years of ebert and panchal—what next? *Heat Transfer Engineering*, 38(7-8):669–680, 2017.

[41] Mostafa M Awad. Fouling of heat transfer surfaces. In *Heat Transfer-Theoretical Analysis, Experimental Investigations and Industrial Systems*. InTech, 2011.