Multivariate Modeling in Chemical Toner Manufacturing Process

by

Hassan Khorami

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Chemical Engineering

Waterloo, Ontario, Canada, 2013

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.
I understand that my thesis may be made electronically available to the public.

Abstract

Process control and monitoring is a common problem in high value added chemical manufacturing industries where batch processes are used to produce wide range of products on the same piece of equipment. This results in frequent adjustments on control and monitoring schemes. A chemical toner manufacturing process is representative of an industrial case which is used in this thesis. Process control and monitoring problem of batch processes have been researched, mostly through the simulation, and published in the past . However, the concept of applying the subject to chemical toner manufacturing process or to use a single indicator for multiple pieces of equipment have never been visited previously.

In the case study of this research, there are many different factors that may affect the final quality of the products including reactor batch temperature, jacket temperature, impeller speed, rate of the addition of material to the reactor, or process variable associated with the pre-weight tank. One of the challenging tasks for engineers is monitoring of these process variables and to make necessary adjustments during the progression of a batch and change controls strategy of future batches upon completion of an existing batch. Another objective of the proposed research is the establishment of the operational boundaries to monitor the process through the usage of process trajectories of the history of the past successful batches.

In this research, process measurements and product quality values of the past successful batches were collected and projected into matrix of data; and preprocessed through time alignment, centering, and scaling. Then the preprocessed data was projected into lower dimensions (latent variables) to produce latent variables and their trajectories during successful batches. Following the identification of latent variables, an empirical model was built through a 4-fold cross validation that can represent the operation of a successful batch.

The behavior of two abnormal batches, batch 517 and 629, is then compared to the model by testing its statistical properties. Once the abnormal batches were flagged, their data set were folded back to original dimension to form a localization path for the time of abnormality and process variables that contributed to the abnormality. In each case the process measurement were used to establish operational boundaries on the latent variable space.

Acknowledgements

First and foremost, I would like to thank Dr. Ali Elkamel to take me in as a graduate student. I'd also like to acknowledge him for his constant assistance, valuable advice, and encouragement throughout my graduate studies. I can't thank him enough as this thesis took so many turns and he kept putting me in touch with researchers around the University of Waterloo and the Petroleum Institute.

Many thanks to my colleagues in Xerox Corporation. I would like to thank my managers, Ed Langlois for his continuous support throughout my career at Xerox and Dan McNeil for his words of encouragement and push for tenacity. I am thankful to Dr. Hwee Ng for his effort on the confidentiality agreement, Dr. Santiago Faucher for his advice during early stages of my studies, and Dr. Chieh-Min Cheng for his feedback.

I wish to thank University of Waterloo and the Department of Chemical Engineering for the support I received during my studies and for providing me the opportunity to conduct my thesis research. This degree is truly a realization of a long time dream, and I am so grateful to have been part of such a well recognized university.

I would like to extend my sincere thanks to Dr. Luis Ricardez Sandoval and Dr. Mazda Biglari from the Department of Chemical Engineering for their valuable advice and comments on the thesis. I am also thankful to Dr. Uwe Kruger of The Petroleum Institute for his valuable advice and for showing me new frontiers in the filed of Statistical Process Control. I am also thankful to my friends and colleagues in the Department of Chemical Engineering, specially Baoling Chen for her kind help, supports and encouragements.

Dedication

To my family who have supported me throughout my life, have never left my side, and are very special.

A special feeling of gratitude to my wonderful mother whose words of encouragement and love have been inspirational to me. I will always appreciate all you have done for me.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background

Batch processes are widely used in production of high value added chemicals including pharmaceuticals, tailor made polymers, biochemical and semiconductors manufacturing (Hong, Zhang and Morris 2011). One of the issues in production of a batch process is process control and monitoring. Monitoring and control a batch process is difficult because of complex reaction kinetic, combination of non linear and linear process variables, and designation of a specific piece of equipment in production of wide range of products.

Process system engineering (PSE) can be used to improve the operation of batch processes. These improvements can be applied in the design phase where the nominal operation is defined. Mechanistic models are used to develop the design models and estimate the operating parameters (Munoz 2004). In the production stage, scheduling (through optimization) can be used as another source of process improvement (Leung 2009). Despite the setting of operational parameters during the design stage, during the production of a batch another set of undesired disturbances may enter to the process. Examples of these undesired disturbances are the variations in the raw materials, malfunctioning of process instruments, and varying times of each batch due to the first two unwanted disturbances.

1.2 Process Description

Controlled batch copolymerization and aggregation of submicron polymer lattices are an important process in the production of high quality Chemical Toners. Companies such as Xerox and Konica-Minolta have developed processes where copolymers of Nano micron size particles are synthesized and then combined with pigment and other petroleum derived additives into micron sized aggregates. These aggregates are then processed further and made into functional dry toner powder for xerographic application. Over the years Chemical Toner has been transferred from the laboratory to the production scale. While the process has been optimized experimentally and the control variables are understood, there is no work on mathematical modeling the manufacturing processes. Such fundamental models could help existing production plants or future lab scale optimizations and may eventually find value as standard controls.

A toner manufacturing process that offers tailor made chemical toners and developers is a representative in this case. Multiple pieces of equipment are used to charge different raw materials to the reactor. Each piece of equipment including the reactor has different process variables associated with it. Once the reaction is complete and product is discharged, the reactor is purged and the final product goes under quality control testing.

The charging of raw material to reactor has six steps including the initial material charge to the reactor, initiator addition, first reactant charge to reactor from reactor feed tank, second reactant charge to reactor from reactor feed tank, reactor hold time and batch cooling, and last step is the purge of remaining material through the reactor. Since no reaction takes place in the sixth step and all process variables remain constant during the purge of material from the reactor, the last step of the process was excluded from this study.

Variables for each batch could be organized in to two sets. First set includes the process variables trajectories throughout the batch. Second set of variables include the quality variables that represent the quality of finished product and measured at the end of a batch. Two data types of X, and Y are defined to hold these sets of data. X holds the batch process variables and matrix Y holds the product final quality variables. Table (1.1) provides a more detailed description of the variables used for data analysis.

Figure (1.1) is the graphic representation of the process including associated vessels and instruments.

Table 1-1: List of variables measured per batch

Data Type	**Variable Name**	**Description (Unit)**
X	Process Variable 1	Reactor Batch Temperature (°C)
X	Process Variable 2	Reactor Jacket Temperature (°C)
X	Process Variable 3	Reactor Jacket Supply Temperature (°C)
X	Process Variable 4	Reactor Jacket Return Temperature (°C)
X	Process Variable 5	Reactor Pressure (Kpa)
X	Process Variable 6	Reactor Agitator Speed (rpm)
X	Process Variable 7	Reactor Condenser Temperature (°C)
X	Process Variable 8	Reactor Feed Tank Temperature (°C)
X	Process Variable 9	Reactor Material Addition Rate (Kg/min)
X	Process Variable 10	Reactor Feed Tank Pressure (Kpa)
Y	Quality Variable 1	Value of the first quality test
Y	Quality Variable 2	Value of the second quality test
Y	Quality Variable 3	Value of the third quality test
Y	Quality Variable 4	Value of the fourth quality test

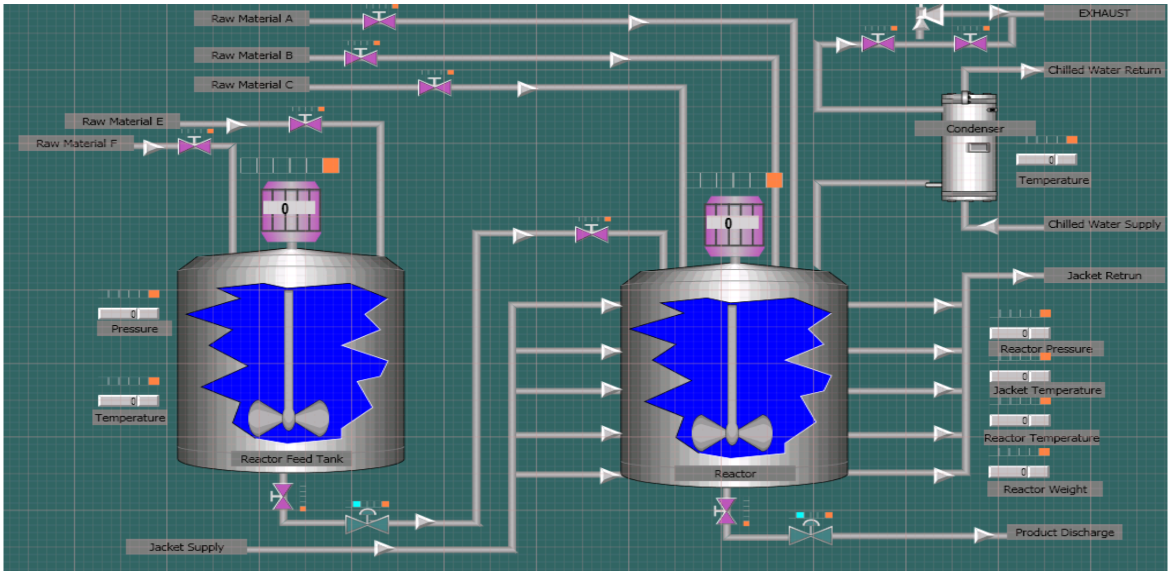


Figure 1-1: Process Flow Diagram

Figures (1.2) through (1.4) provide a visual description of the 5 different stages of the process. Due to the confidential nature of the process, wherever the presentation of the raw data was required, the data of Y-axis has been fit to the scale of 0 to 1.

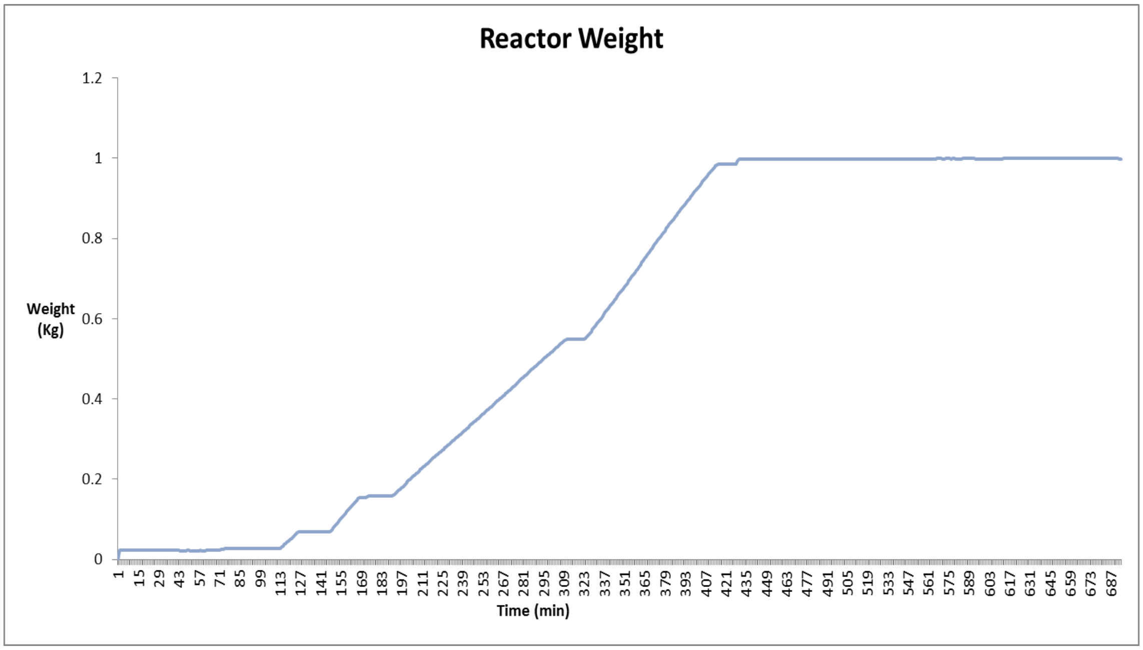


Figure 1-2: Evolution of reactor weight during 5 stages of reaction.

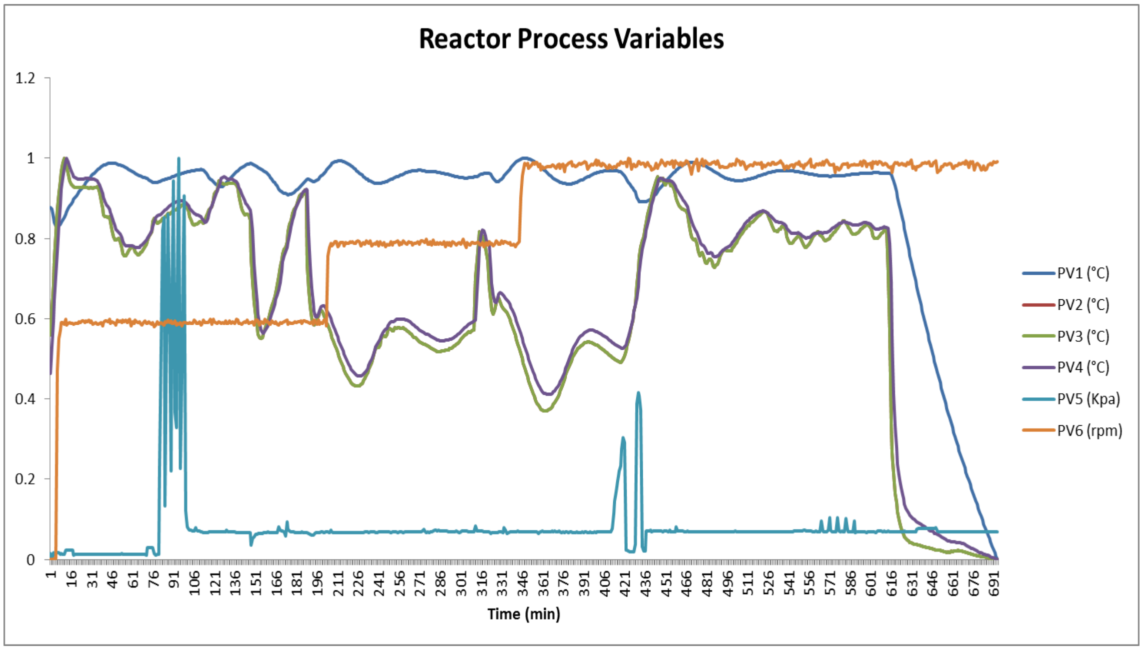


Figure 1-3: Evolution of reactor process variables during 5 stages of reaction.

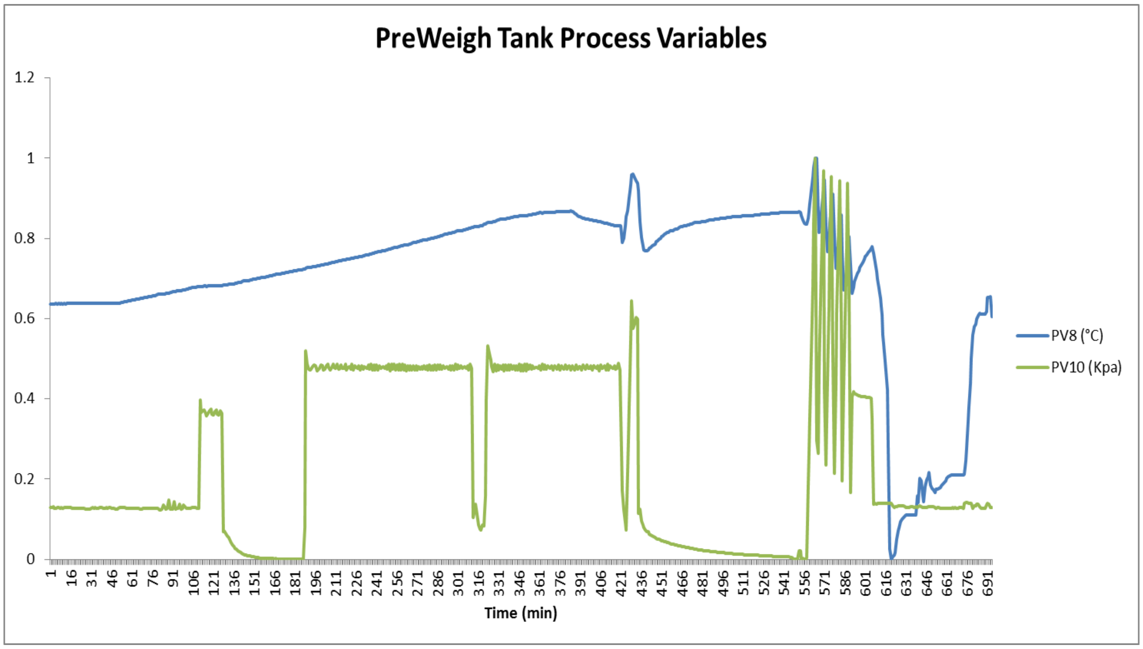


Figure 1-4: Evolution of Pre-weigh Tank process variables during 5 stages of reaction.

1.3 Research Objectives

The goal of this thesis is to turn the mathematical modeling techniques from literature and research papers to a mathematical model that is built from the history of the past successful batches. The model can be applied for process monitoring and control of developer and chemical toner manufacturing processes. Therefore, providing an online monitoring technique for multi-vessel processes and to facilitate recognition of abnormal process behaviour and faulty batches. The specific goals to be addressed are:

- Detailed literature review of linear and non linear mathematical modeling techniques to determine the appropriate modeling approach.
- Data preprocessing to ensure each process variable has equal representation in the model and it is not over looked.
- Model cross validation to prevent over fitting of the model but still providing reliable model prediction.
- Introduction of batches with abnormal process behaviour or off specification quality products to ensure the model is capable of flagging the time of abnormality and process variable associated with abnormal behaviour.
- Establishment of the operational boundaries from the history of the past successful batches to provide an online upper and lower limit approach for individual process variables.

1.4 Thesis Overview

A brief summary of propose techniques in this book and each chapter is given below:

Chapter 2 is the review of mathematical modeling techniques applicable to chemical toner process including latent variable modeling and population balance modeling.

Chapter 3 introduces data centering, scaling and alignment techniques for batch data collected from the process.

Chapter 4 is about improvement on the principal component analysis, multi-way principal component analysis, model cross validation, and fault detection and diagnosis in the batch processes. This is accomplished by applying these techniques on data collected from the process.

Chapter 5 introduces least square regression, partial least square regression and multi-way partial lest square regression techniques. This chapter also presents the projection to latent structure equation to introduce the operation policy and boundaries of process variables.

Chapter 6 is a summary of all the research done in this thesis and provides recommendations for future research on this topic.

Chapter 2
Literature Review

This chapter reviews the literature of two modeling approaches and draws comparisons between stochastic modeling versus a mechanistic modeling in chemical toner and developer manufacturing process.

Section one is a review of Latent Variable Modeling. This modeling technique, in a simpler term could be referred to as process component analysis, PCA. PCA is a method in which a set of variables are converted, compressed, to a set of new variables that that are uncorrelated to the original data and orthogonal to each others. Thus number of principal components, latent variables, is less than or equal to the number of original variables. In PCA the first principal component contains the highest possible variability within the dataset. Subsequent principal components contain the next highest possible variance in the dataset, given that principal components are orthogonal ,uncorrelated, to each other.

Section two is a review for population balance modeling. The origin of population balance could be traced to Boltzmann equation (developed more than a century ago), or it could be considered a relatively new subject that has been explored in various application that engineers has more recently put it into use. The recent trend shows the fact that methodology of population balances can not be avoided in the study of particulate processes. Chemical reaction in a dispersed phase system exists in conjunction with the process producing the dispersion and population balance is capable of addressing the evolution of dispersion. Besides, reviewing the application of population balance modeling in the particulate systems, this section focuses on systems that experience changes in the number of particles including appearance or disappearance anywhere within the particle space. Appearance and disappearance of particles can be linked to particles breakage and/or aggregation process. The last part of this section focuses on the rate of particle breakage/aggregation and the efficiency of the models representing these processes. Thus, it should be considered that population balance equation can be applied in aggregation process only by modeling the efficiency of the aggregation process.

2.1 Latent Variable Modeling

Batch processes are increasingly used in high value added product manufacturing industries including pharmaceuticals, agriculture, chemicals, biochemical and composite manufacturing. Monitoring and optimization of these processes are very important to assure they produce high quality products consistently. Process System Engineering (PSE) can play a key role in developing techniques to improve the monitoring and optimization of batch processes (Munoz 2004). These improvements can be achieved at the design or production phases of a batch process. Once the batch is in the production phase, the improvements such as optimization and controls are applied to scheduling and production itself respectively to maximize the throughput of a batch process.

Furthermore, in certain time during the life cycle of a batch process, the operation of the batch might be revised to purposefully change the desired specification of the final product.

Usually, once a batch reaches the production stage, undesired disturbances (explained in section 1.1) enter the process and the original design may not be suitable to point out the disturbances. Besides, the nature of disturbances changes from the design or pilot stage to the production stage. Improving a batch process to deal with the new sources of uncertainty could be a challenging task due to the nature of the unwanted disturbances and complex behavior of most batch processes.

One of the challenges in the production of batch processes are process monitoring and control. Other issues are scheduling, planning, operational policies and quality control decisions. These issues have been discussed by (MacGregor, Penlides and Hamielec 1984), (Stephanopoulos 1990), and (Kozub 1992). Lack of online sensors to measure quality variables, finite duration of batch processes, existence of significant non-linearities, and lack of steady state operation are among main challenges limiting the ability to provide adequate control and monitoring of batch processes. Batch processes usually contain batch to batch variations. These variations can be caused by process variables deviation from their trajectories, errors originating from charging raw materials, or differences originating from variations in impurities of the raw materials (Nomikos and MacGregor 1995). Abnormal conditions developed during a batch can lead to the production of poor quality products, specially if the problem isn’t detected and addressed. Since there aren’t many online sensors to measure the final quality of a product, even the current sensors on the market are very expensive to own and operate, most batch processes operate under open-loop conditions. Once a batch is finished, several quality tests are performed on a sample of the final product to determine the quality of the batch. In some cases the measurements resulted from quality tests are used to adjust the recipe for the next batch. The current practice in the industry to achieve quality targets is sequencing every stage of a batch process through process automation software. Monitoring defined as checking that the sequence is followed and important reaction process variables follow acceptable trajectory.

Statistical Process Control (SPC) Charts in batch processes can be traced to Shewhart charts (Nomikos and MacGregor 1995). One of the difficulties in applying SPC originates from the dynamic nature of batch processes. Usually, the SPC method covers the quality measurement obtained at the end of a batch thus monitors batch to batch variation. Nowadays, computers are connected to batch processes and can collect data during production of a batch. Temperature, Pressure, and flow rates are examples of process variables collected during a batch. While it is typical to measure many process variables during production of a batch, this does not translate to reaction aspects taking place independently. Because at anytime during a batch only few events influence progression of a batch. The most challenging factor is handling large number of process variables including time alignment, highly correlated and non linear structure of batch data, and the finite time nature of batch processes. Besides, the relationship among process variables at any time during progression of a batch have the same importance as the past history of these process variables. One of the common practices in the analysis of this type of data is multivariate statistical projection methods which are based on Principal Components Analysis (PCA) and Partial Least Squares (PLS). In these methods the data are compressed to extract the information from them.

Some of the schemes for multivariate procedures for monitoring continuous processes are of (Kresta 1991) and (MacGregor, Jaeckle, et al. 1994). In these methods the variation in the trajectories of historical reference distribution of normal batches are characterized by projection of the data into a lower dimensional space, (data compression), principal component space. The principal component space summarizes process variables and their time histories for successful batches.

Figure (2.1) is a visual representation that shows the matrix of original data, X, is reduced to a matrix of summary variables T (principal components).

Figure 2-1: Projection of data into lower dimension.

Once the trajectories of history of data from normal batches are characterized in a lower dimension, it provides a fingerprint for each batch and an empirical model which projects operation of a successful batch. This approach is similar to statistical process control approach where process behavior is characterized through the usage of data obtained when process is running normal and under control. Then, behavior of the new batch is compared to the model and its statistical properties to test a null hypothesis. The hypothesis states that the projection of process variables throughout the new batch follow the same pattern of normal batch operation explained by the history of the previous successful batches. This technique has culminated to the introduction of multivariate statistical process control charts that are similar to Shewhart charts in interpretation but they more powerful to detect small changes in batch process. This powerful ability can be used as part of the monitoring policy to shrink the control limits, detect faulty behaviours, and prevent their future appearance. Therefore it facilitates production of batches with more consistent quality products (Nomikos and MacGregor 1994). Although this is a non-directional approach that detects any deviation from the history of previous normal process behavior, it will be less powerful than knowledge based approaches like state estimation to detect abnormalities that are built into the models. The advantage of multivariate approach is that to develop the monitoring procedure, the only information is required is the database of previous successful batches. In the Shewhart charts, when deviation from normal quality target is detected, it is up to engineers and process specialists to use their knowledge of process to provide fault diagnosis, and respond appropriately. Whereas, multivariate statistical procedures including principal component analysis and partial least square regression provide more diagnostic information about process abnormality (Kourti, Nomikos and MacGregor 1995) and (Nomikos and MacGregor 1995).

To perform mathematical procedures related to principal component analysis and partial least squares, the matrix of data containing process variable values should have the same lengths. This means that batch processes should have the same time durations but batch processes have different batch times which are caused by variations among raw materials, operating conditions, or process instruments. An example would be a batch process that is set to stay at a temperature set point of 100 °C until certain molecular weight is reached for the polymer being formed inside a reactor. If the processing time of the first batch is assumed to be 1 hour, subsequent batches will have longer processing times because each batch increases fouling's inside the reactor. The processing times keep increasing until the reactor is cleaned, and the first batch after the cleaning will have a processing times of 1 hour. Since batch processes have different times, they add extra complexity to process data analysis. Dealing with batch data with different durations is discussed by (Munoz 2004) and (Nomikos and MacGregor 1995) through introduction of time alignment and indicator variable respectively.

Another concern in using process and quality variable values to perform mathematical procedures related to principal component analysis and partial least squares is treatment of original values. Table (1.1) indicates that process variables have different measurement units. They also have different operating range. Using process and quality variables to build a mathematical model without performing preprocessing will result in values with higher operating range and numbers to have more influence in the model. Thus preventing the model from establishing a finger print of past successful batches based on all critical process values involved in the batch process. Centering and scaling introduced by (Bro and Smilde 2003) can address concerns about treatment of process data in the data set used in this thesis. They stated that their data preprocessing technique can be used for multi-way data analysis.

In centering, the data set is converted from an interval scale to a ratio scale. Therefore, centering reduces rank of a model and increases the fit to the data simultaneously. Another benefit of centering is the prevention of numerical problems. In some principal component analysis algorithms, the larger the values, the longer it takes to calculate largest eigenvalues and compute their convergence rates. In this regard, centering is seen as a projection step that is used to project process and quality values to a model. To build a mathematical model, model parameters need to be estimated. Centering is a convenient method of estimating parameters of a model where the offsets are limited but in a real world industrial example, the subject of this thesis, the offsets are not limited. Elimination of the offsets basically can not be accepted in this research because it prevents the ability of a model to predict abnormal behaviours.

Scaling is a way to change the weights of process data to be used in a model. However, scaling does not the relationship between data or the structure of a model (Bro and Smilde 2003). In other words, scaling explains the offsets without changing the relationship between the data.

2.2 Population Balance Modeling

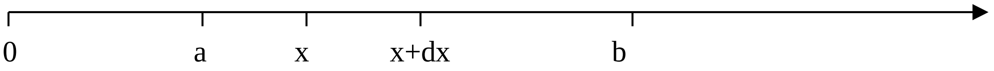
As a particle distribution theory and predictive multidimensional modeling technique, population balance is used in predicting the shape and density of the particle distribution. Therefore, distribution function is characterized through a solution of a differential equation. The parameters which affect the distribution formation are then plugged in the differential equation, thus making the theory predictively and descriptively useful.

Studying the particulate systems and the behavior of the particles population in their surrounding environment can be synthesized from the particle behaviour on its local environment. The term population can be explained by the density of a specific variable (e.g., number of particles, mass or volume of particles). Population balance can be used to model a wide variety of disperse phase systems including solid –liquid (crystallization), gas – liquid, gas – solid, and liquid- liquid dispersions. (Randolph and Larson 1988). Among the wide application of population balance, this study mostly focuses on the distribution of particle populations in the aggregation and their effect on the system. Another feature of interest in this research is the study of particles constantly created and destroyed through the process of breakage and aggregation that could be used in the population balance modeling of the system. External and internal coordinates are two major characteristics of particles in particulate systems. Quantitative characterization are provided by internal coordinates which are properties attached to each particle regardless of its position.(e.g., particle size). External coordinates on the other hand, indicate the location of the particle in the physical space. (e.g., description of a well mixed particulate process). The joint space of the internal and external coordinate is known as particle phase space that provides a through description of the properties of the distribution. To formulate a population balance equation, it is assumed that number density of the particles at every point in the particle space state exists. By integrating the number density function of the desired region, the number of particles in a desired region can be obtained. The population balance equation is a method of predicting appearance or disappearance rate of particles from a system. The particle formation is birth process and disappearance is the death process. Examples of birth of new particles are breakage, splitting, aggregation and nucleation. Breakage and aggregation are also contributing factor to the death process for particles that either break or aggregate with other particles in the particle space system.

2.2.1 Population Balance Equation

As mentioned earlier the idea behind population balance is to formulate a number balance equation for particles of each type, and the particle type is described by one or more state variables that are usually continuous. If a single variable is used, it is referred to as scalar state variable. For types by more than one variables, they are referred to as states. Examples of state variables are particle size, age, temperature and concentration. Physical space variables may also be included.

To start, it is assumed that a particle described by a single state variable in which x is the scalar property of a particle.



Therefore, there is an interest in knowing how many particles are there with the specific property of x.

Thus by definition of the density function, n(x,t) is the number of particles per unit volume in the state space therefore, n(x,t) is defined as the number density of particles at time t.

n(x,t)dx = number of particles at time t between x and x+dx

next definition is N(t) as the total number of particles in some finite sub-region of particle phase space x. so total number of particles in the system of all x's is

$$N(t) = \int_0^\infty n(x,t)dx \qquad (2.1)$$

The number of particles between a and b are calculated by

$$\int_a^b n(x,t)dx$$

n(x,t) changes because of particle move along (e.g, crystallization or growth) or against (e.g, dissolution) the length line or disappearance. To derive the population balance equation in the number density function, it is considered that the particles are embedded in a continuum which deforms in with the kinetic field responsible by vector $\dot{X}(x)$. $\dot{X}(x)$ is the rate of change of x = velocity of x.

The mass balance equation: Accumulation = Input – Output + Net generation

Therefore, the Population Balance Equation (PBE) for the sub-region of x between a and b which moves convectively with the particle phase space velocity can be written as (Ramkrishna 2000):

$$\frac{d}{dt}\int_a^b n(x,t)\, dx = n(a,t)\ \dot{X}(a) - n(b,t)\dot{X}(b) + \int_a^b h(x,t)dx \qquad (2.2)$$

$$\int_a^b [\frac{\partial\, n(x,t)}{\partial t} + \frac{\partial}{\partial x}\{\dot{X}(x)\, n(x,t)\} - h(x,t)]\, dx = 0 \qquad (2.3)$$

$$\frac{\partial\, n(x,t)}{\partial t} + \frac{\partial}{\partial x}\{\dot{X}(x)\, n(x,t)\} = h(x,t) \qquad (2.4)$$

Halbert and Katz (1964) derived the above equation which is considered the simplest form of a PBE in which $h\,(x,t)$ is the net generation. The possibility of particle appearance or disappearance anywhere in the internal particle phase ,also defined by birth and death functions, of the particle distribution should be considered. Thus $h\,(x,t)$ is the result of subtraction of death rate from birth rate. Halbert and Katz equation shouldn't be considered by itself a population balance model. The modeling should consider identifying the nature of the function $\dot{X}(x)$ and $h\,(x,t)$ for a given situation.

Now that a simple population balance equation is derived, more in depth application of a population balance equation in form of pure aggregation process can be reviewed. In pure aggregation the population can be considered uniformly distributed in space thus external coordinate are not involved. The aggregation frequency of the particle pairs (particles x and x') is identified by $a(x,x')$. B and D are birth and Death functions (Ramkrishna 2000).

$$h(x,t) = B(x,t) - D(x,t) \qquad (2.5)$$

The birth function represents the rate of the formation of particles of x and is derived by assumption that particles of size $(x - x')$ aggregate with particles of size x' to produce particles of size$x$. Thus $0 < x' < x \quad for\ a(x-x',x')$ and each pair in the set is considered twice thus the birth function should be divided by 2 to avoid double counting.

$$B(x,t) = \frac{1}{2}\int_0^x a(x-x',x')\, n(x-x',t)\, n(x',t)\, dx' \qquad (2.6)$$

$$D(x,t) = \int_0^\infty a(x,x')\, n(x,t)\, n(x',t)\, dx' \qquad (2.7)$$

The PBE equation is then re arrange to include the number density function (Ramkrishna 2000).

$$\frac{\partial n(x,t)}{\partial t} = \left[\frac{1}{2}\int_0^x a(x-x',x')\, n(x-x',t)\, n(x',t)\, dx'\right] - n(x,t)\int_0^\infty a(x,x')\, n(x',t)\, dx' \; (2.8)$$

Aggregation is formed between two particles but in crowded space several adjacent particles can colloid and aggregate simultaneously. One of the main instruments in the population balance modeling of aggregation processes is aggregation frequency. Aggregation frequency is the probability of a pair of particles aggregating per unit time. Since population balances assumes well mixed solutions with no external coordinates in the population density, the modified interpretation of aggregation frequency in population balance modeling can be used (Ramkrishna 2000). Certain distance between particles is a requirement in the aggregation between two particles, thus the aggregation is about the relative motion between particles. So aggregation frequency model should consider relative motion between particles.

$$a(x,x') = \int_{\Omega r} a(x,0;x',r'-r)\, dVr'-r \qquad (2.9)$$

Where $a(x,0;x',r'-r)dt$ is probability of particles of size x' at the relative location r'-r encountering particle with the size of x located at the origin during the next interval of dt and is model specific.

Brownian motion is an example of particles by random motion. In the following equation k is boltzman constant, μ is viscosity of the suspension and T is temperature.

$$a(x,x') = \frac{2kT}{\mu}\left(x^{-\frac{1}{3}} + x'^{-\frac{1}{3}}\right)\left(x^{\frac{1}{3}} + x'^{\frac{1}{3}}\right) \qquad (2.10)$$

The above equation, Brownian coalescence frequency, explains high rate of agglomeration between particles because of vigorous diffusion of the smaller particles toward their larger particles. In Brownian coalescence frequency the particles move independently from each other despite the fact that they are in close distance of each other. That said, because of inter-particle forces of intervening suspensions, this equation does not factor correlation between the movements of particles. In some situations, collision bouncing off particles to different directions. Thus, an aggregation efficiency should be factored in to complete the modeling of the aggregation frequency. In fact what have been used so far as aggregation frequency could be replaced as collision frequency. Collision frequency yields to aggregation frequency by including the aggregation efficiency.

$$Aggregation\, Frequency \;= Collision\, Frequency * Aggregation\, Efficiency \qquad (2.11)$$

In summary, to calculate the aggregation frequency, both collision frequency and aggregation efficiency should be modelled. The aggregation efficiency is the probability that two particles collision is an act of aggregation and formation of a single particle. This fact denotes that collision frequency and aggregation frequency are rate functions but aggregation efficiency isn't. (Ramkrishna 2000)

2.3 Literature Review Summary and Thesis Direction

Despite the fact that population balance modeling provides more deterministic behaviour of a process compared to predictive approach of latent variable modeling, population balance modeling has remained a tool mostly used within academic community. Though population balance modeling has remarkable active research and provides better understanding of internal and external coordinates involved in the process and can better explain multivariable input and output of the process (Díez 2004). There are several reasons for the lack of industrial applications of PBEs.

- A simple PBE equation on the assumption of complete mixing may not lead to a realistic model but factoring real distribution of the particles results in more complex models which makes the model mathematically challenging and difficult to use in industrial applications.
- It is hard to derive a PBE control strategy for a batch process due to the distributed nature of the population balance models and limited number of the manipulated variables in batch operations.
- Deterministic nature of population balance modeling requires extensive experimental efforts to gather good data for population density distribution.

Since most of the objectives of this thesis is to pave the way for modeling, optimization, process monitoring, and product development in Chemical Toner Processes- it is more realistic to focus on latent variable modeling methodology rather than population balance modeling. In chemical toner process, latent variable modeling would capture the essence of the process and provides a realistic prediction of process behaviour. Latent variable modeling shall be implemented such that represent realistically the operation of the industrial unit and yet mathematically simple enough to be used in online application. The model shall be used for the purpose of design and implementation of an automatic control system which could be used in the real time Chemical Toner Processes.

That said the author suggests that in order to use the population balance modeling, one could still refer to the manufacturing data and use population balance models. In order to accomplish this, a population balance models that most could resemble our process should be chosen from the literature, (Ramkrishna 2000) then the model should be applied to the manufacturing data and model constraints should be built. The final stage of the model building would include model cross validation with data unused for model building. A technique used in the cross validation technique described in chapter 2 can be applied to population balance modeling.

This thesis contains techniques for modeling, monitoring, prediction, fault localization of a batch process in its production stage.

- Such techniques are data preprocessing, alignment, Model Validation, Multi-way Principal Component Analysis, and Multi-way Projection to Latent Structure.

- Two different types of data sets are used for preprocessing of the batch data. First set is the historical batch data of 36 batches identified as product 4 in this thesis, and the second data set consist of 27 batches, identified as product 2. In this process, the solution is prepared in a pre-weight holding tank and is charged to the reactor. The charging of raw material to the reactor has five steps including the initial reactant transfer, initiator addition, first material charge to reactor, second material charge to reactor and last step is the purge of remaining material through the reactor. Then the reactor is put under constant steering and the batch is cooled down. After the batch completion, the material is transferred to another holding tank and awaits quality approval before being used in other types of products.

This thesis will illustrate the latent variable methodology by modeling the above process. Then, it will illustrate how Multi-way Principal Component Analysis (MPCA) is used to perform analysis on completed batches to distinguish abnormal batches from successful batches. This analysis can be deployed for improving operational boundaries and to understand contributing factors in batch-to-batch variation.

A base recipe will be chosen, with completed successful batches to create a reference database for the model. The final quality of these batches were similar to the variations observed in successful production runs. The final quality measurements characterize what quality zones are acceptable for a product. A normal batch should be within the mean for each quality variable with 3 standard deviations as the margin of error. Additionally, two more batches which have product quality outside of this specified zone (abnormal batches) will be chosen. These batches will be helpful to investigate MPCA’s ability in detecting the cause for process deviation occurring at different times. The final stage of PCA modeling is fault localization for purpose of online monitoring with PCA and projection to latent structure. We’ll also illustrate the operation conditions for each process variable in order to make consistent quality products. The final stage in the thesis is delivering the original source code (MATLAB) executable version of the model. As mentioned earlier, this model shall be used as a ground work for the implementation of an automatic process control systems of Chemical Toner Process.

Chapter 3
Data Preprocessing and Alignment Techniques

As discussed in previous chapter, section (2.1), treatment of process and quality variables, raw data, is necessary to perform mathematical procedures related to principal component analysis and partial least squares. Treatment techniques surveyed from literature review were centering, scaling, and synchronization. This chapter reviews and integrates the application of methodology introduced by (Bro and Smilde 2003) for centering and scaling in principal component analysis. They authored the paper on the two way data analysis however they expanded the outcome to be applicable for a multi-way data analysis which will be the focus in this research. The chapter also utilizes the information obtained from the alignment of batch trajectories (Munoz 2004) .

In centering ,as a projection step, data are projected into a predefined space in a given mode. Generally, model building from data includes two major steps: the first step is to assume a structural model and the second step is to choose a method for parameter estimation. In other words, Centering deals with creating a model including its offset. Scaling covers the second part. In scaling, a different way of model fitting is employed. Centering can be considered a convenient method in parameter estimation in models that have specific offsets, but this technique can not be generalized to all offset categories.

Scaling is a method that uses a weighted least squares loss function to fit models. It changes the weights of data that are used to fit into a model but does not change model's structure. Therefore, compared to centering, it has and it has less influence on the model (Bro and Smilde 2003).

Analysis of batches with varying duration is one of the issues with batch process analysis. In order to synchronize or to align a set of batch trajectories, certain transformation should be performed on each batch trajectory so that once the alignment is complete, batch trajectories line up and have similar evolution. Of course the number of samples, time stamps or batch duration, should be equal to perform principal component analysis.

3.1 Data Centering

To understand how a data set is centered, it is essential to review its goal and how centering works. Objective and qualitative reasons to perform centering are provided by (Harshman and Lundy 1984). If usual offsets exist within the data or constructing a model of the usual offsets result in a reasonable model, then centering can be considered. Therefore, the purpose of centering is to have interval-scale data behaving like ratio-scale data (this data type is mostly used in multivariate models). Consequently, centering can increase the fitting of a model to the data, specifically remove the offsets, or prevent numerical problems.

To reduce rank of a model: centering is reasonable in a case that a model of raw data is represented by (n + 1) components in the matrix of raw data X (I, J), but a model built with centered data requires (n) components. A model representing centered data contains n (I + J) + J parameters. If the centering is deployed on the first mode, then the J parameters can be connected to the calculated averages. It would be meaningless to fit a with model with (n+1) components to the matrix of raw data because it would produce (n + 1)(I + J) parameters.

Increase the fit to the data: sometimes centering does not reduce the rank of the proper model. But if introducing extra parameter would significantly improve the fit of the centred data, then introducing extra parameters is useful. The offsets produced by centering could be considered as half a component more than what is required by the model. In this half extra component, the scores or the loadings are equal to one. Scores are considered as centering across the first mode and loadings as centering across the second mode. Thus a model that is fitted with n components would not adequately represent the data compared to a model fitted with n components and offsets. The latter is less adequate fit than a model represented by n + 1 components.

Remove of offsets: centering is helpful in removal of some offsets but in certain circumstances such as process monitoring the offsets are of interest.

Preventing numerical problems: to minimize problems raised from algorithm solutions, it is helpful to centre the data in some algorithms. For instance, when constructing a model by principal component analysis, the rate of convergence in the iteration method is related to the ratio of the two largest eigenvalues. Therefore, centering of specific modes is beneficial to reduce the numerical problem, because the optimization problem relates to a different model that has different properties (Bro and Smilde 2003).

If X (I, J) is the matrix of raw data, m is the vector that holds the average of the jth element, and l is an I vector of ones - the centered data can be represented in the matrix of A (I,J) through the following mathematical expression.

$$A = X - lm' \quad (2.1)$$

The following figures provide comparison of the raw data for the variables within the matrix X prior and after being treated by mean centre calculation. Comparison of Figures (3.1) and (3.2) indicates that centering has decreased the numerical values of process variables 1 through 4, and decreased the numerical values of Process Variables 5,7,8 and 9 up to the centre. Therefore, converting interval-scale data to behave as ratio-scale data. This will result in process variables with higher numerical values to have the same influence on a model as process values with lower numerical values. Unfortunately, it can be noticed that the ratios of the operating ranges of the process variables have changed which indicates that centering has changed the structure of the data.

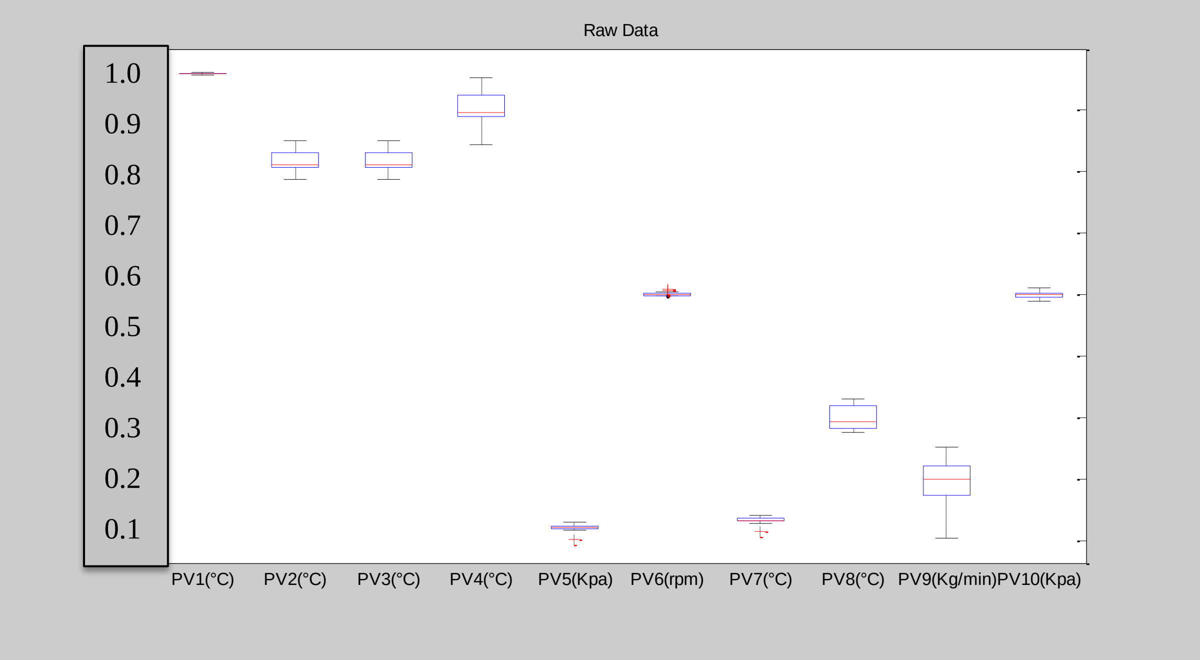


Figure 3-1: Boxplot of raw data

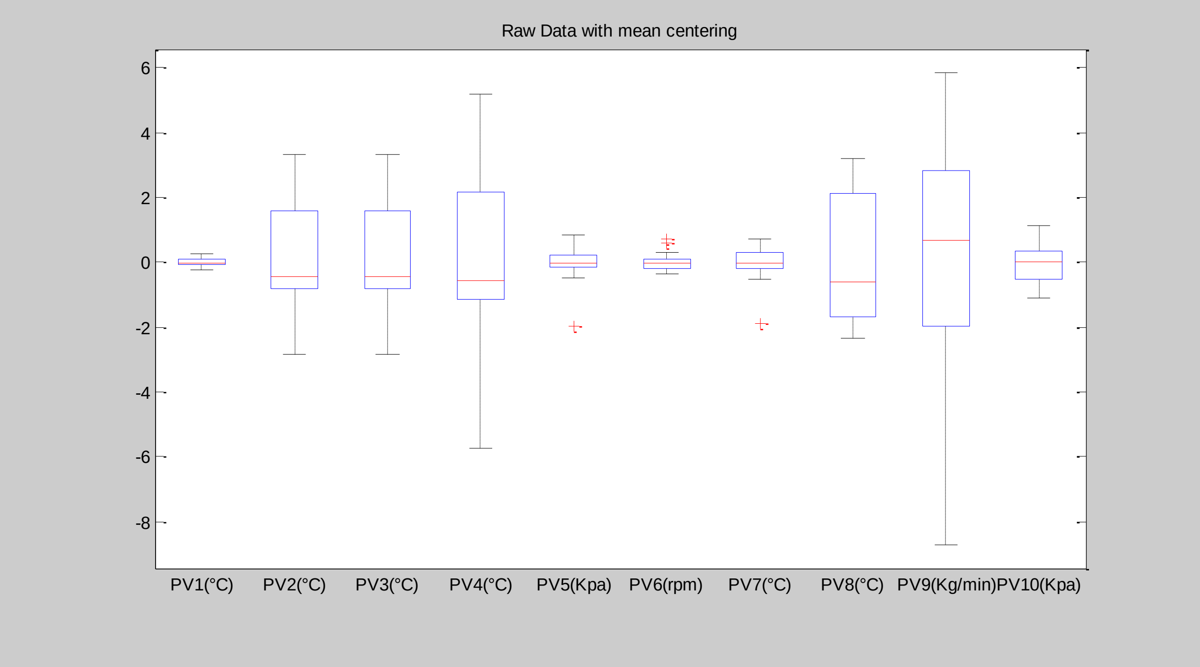


Figure 3-2: Boxplot of mean centered data

3.2 Data Scaling

One of the application of scaling is changing the weights of data that are used to fit into a model. Thus application is different from centring because, model's structure stays intact upon scaling. Therefore; it has less influence on models than centering (Westerhuis, Kourti and MacGregor 1999). Some of the scaling issues are discussed by (Paatero and Tapper 1994).

Scaling is used to adjust differences, accommodate heteroscedasticity, or to permit various size of subset of data.

To adjust differences: scaling commonly is used to (auto-scaling defined by centering on the first mode and then scaling by standard deviation on the second mode) have identical variance for each variable. Thus equal variance is observed by all variables. Subsequently fitting of a model is performed to explain the highest variation within the data. Therefore, each variable is represented equally by the model. One of the applications of this scaling approach is where variables have various measurement unit (Bro and Smilde 2003).

Accommodate heteroscedasticity: if the errors of a least squares model are homoscedastic, independent and Gaussian, (from maximum likelihood perspective) it can be concluded that the fitted model is statistically optimal. In the case of unequal variances of distributions, regardless of them being the same within a certain variable, the fitting procedure could be performed by data scaling within variable mode. Therefore, by applying inverse of the standard deviation to each variable, the data set is scaled and the fitted model is optimal with respect to maximum likelihood.

Different size of subset data: data sets contain subsets of varying sizes; it is an advantage to scale subsets of data separately to allow them to influence the model. For instance, in a reactor with 3 types of process variables measured; the first variable type is 500 measured infrared wavelengths (range is 0 to 1), second variable is pressure, and the third is temperature. The sum of the variances of the infrared spectra is significant compared to temperature and pressure knowing to the noticeable difference in the recorded entries (500, 1, and 1 respectively).The model is inclined to focus on the infrared data if scaling is not performed to adjust for this difference. Explaining the temperature and pressure variables does not proceed to a perfect model, unless the model has sufficient complexity to fit all data subsets instantly or the temperature and pressure data provide similar pattern as the infrared data. If the infrared, pressure and the temperature readings are considered to have equal importance on the process, then scaling all three subsets of data to an equal total variance produces a model that represents this assumption. Therefore, from information perspective, scaling ensures that necessary information is entered in the model, regardless of the variance observed in the various subsets of data (also known as sources of information).

The interpretation of a model and its parameter are not changed by scaling because, as mentioned earlier, scaling is a way of introducing a loss function other than a least square loss function normally used. Scaling is deployed through multiplication of each row or column of the raw data matrix of X (I, J) by a scalar. If scaling is performed on the first mode, then the scaled data is represented by A through the following equation:

$$A = WX \quad (2.2)$$

Where W(I,I) is diagonal matrix with the scaling parameter for the ith row on its jth diagonal element (Bro and Smilde 2003). An example of this type of scaling is used in standard normal variant correction. (barnes, Dhanoa and Lister 1989). If we scale on the second mode, then the scaled data is represented by A through the following equation:

$$A = XW \quad (2.3)$$

Where W(J,J) is a diagonal matrix with the scaling parameter for the ith row on its jth diagonal element (Bro and Smilde 2003). This type of scaling is used throughout this thesis where the weight of a column is the inverse of the standard deviation of the variable (the very same column).

The following figure provides comparison of the raw data for the variables within the matrix X prior and after being mean centred and treated to by scaling to unit variance (dividing centred data to unit variance). It can be noticed that the ratio of the operating ranges of the process variables is returned to its original structure. For instance, process variable 1 and 9 have the same weight on a model and similar variations within their values although, process variable 9 would have significantly higher weight on a model built with only centred data.

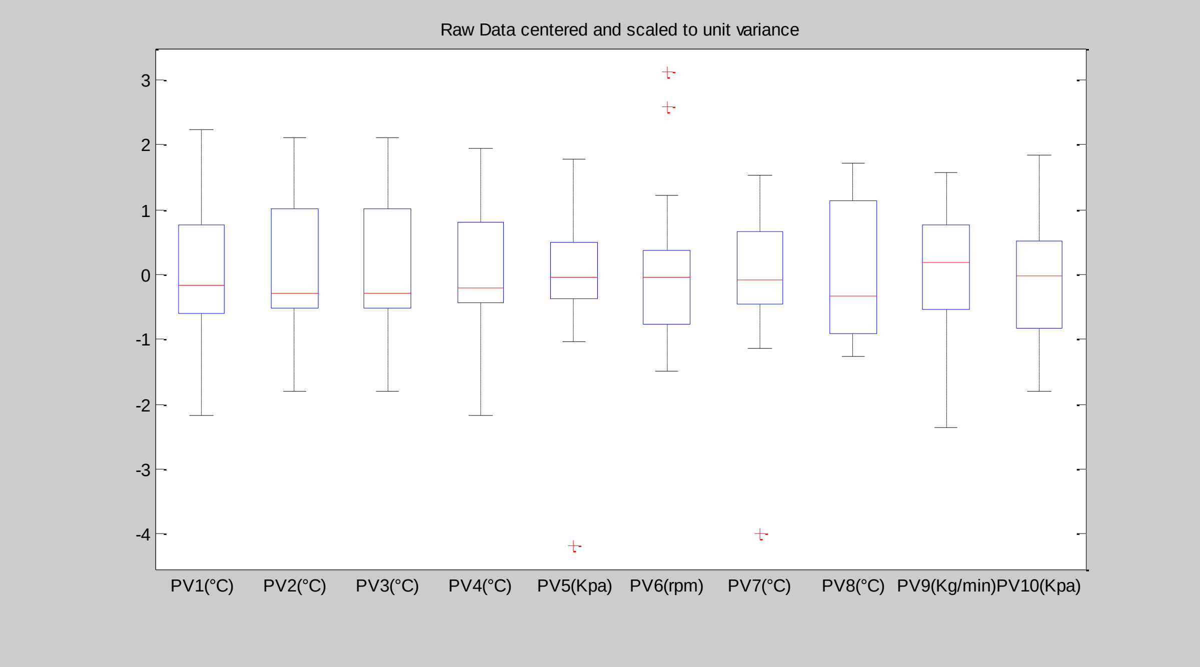


Figure 3-3: Boxplot of data mean centered and scaled to unit variance

3.3 Data Alignment

One of the issues with batch process analysis is handling batches with different duration. This is referred to as the alignment or synchronization problem (Munoz 2004). To perform alignment or synchronize on a data set of batch trajectories, certain transformation needs to be performed on individual trajectories such that once the synchronization operations is complete, batch trajectories are aligned and have identical evolvement. Of course the number of samples should be equal to perform principal component analysis.

In some batch processes the operating conditions are different for each batch thus making the alignment unavoidable. An indicator variable was proposed by (Nomikos and MacGregor 1995) to resample batch trajectories for alignment. For instance, in monitoring the conversion rate for every 1% increment, in the absence of obvious indicator variables, the use of dynamic time wrapping approach was proposed by (Kassidas, MacGregor and Taylor 1998) to obtain samples of the batch data. The authors suggested treating the information that resulted from the alignment as a new piece of information for each batch. In the presence of a consistently increasing or decreasing variable in a batch process that always starts and ends in each batch with different values, the alignment is possible by re-sampling at specific intervals of the variable. Reaction conversion information (Neogi and Schlags 1998), and the cumulative weight of an important monomer added to the batch during reaction time (Kourti, Lee and MacGregor 1996) are example variables. In the absence of such an indicator variable for the entire duration of the batch, each stage of the batch should be looked into to find an indicator variable to allow a stage by stage synchronization.

Often, the transition between stages of a batch is determined by discrete events. The events are triggered by control system or operators as an acknowledgment of completion of a specific stage of a batch or start up of the next stage. A mathematical filter was proposed by (Kaistha and Moore 2001) to use batch trajectories to identify the batch events but in real world, the events are already known and do not require identifying them from the batch trajectories.

In this research, prior knowledge about the operation of the process was used to define 5 stages in the batch, Fig (3.4), and then these stages were used to align batch trajectories. Stage 1 runs from the beginning of the initial solution to the reactor to the end of it. Stage 2 is the initiator addition. Stage 3 covers from the end of initiator addition to the reactor to the end of first reactant transfer to the reactor. Stage 4 covers the second part of the reactant charge to the reactor. Stage 5 is the reactor hold time and batch cooling.

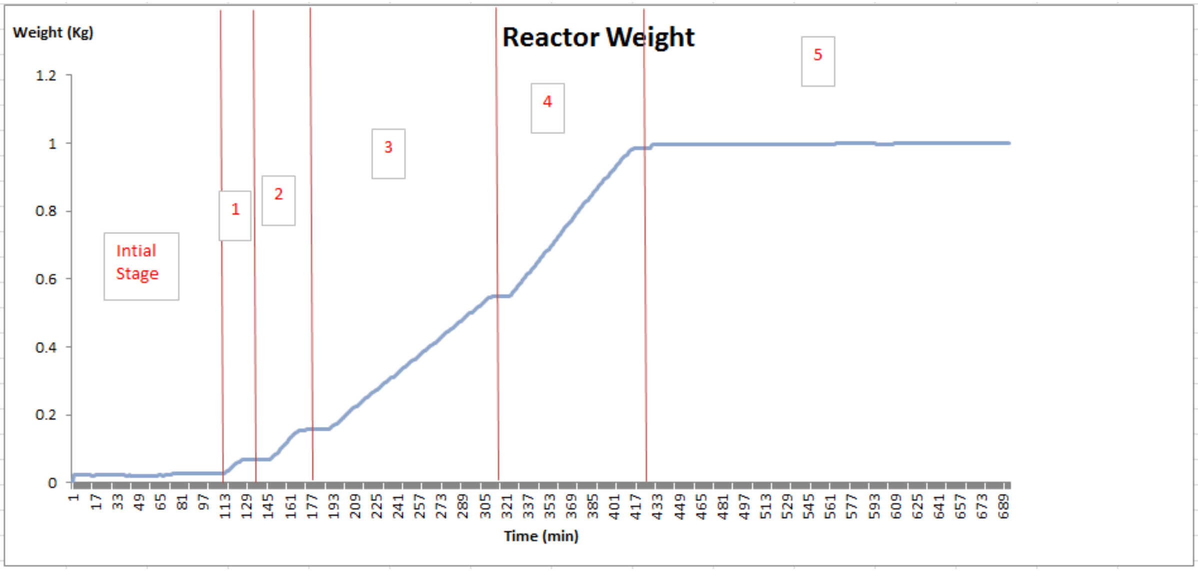


Figure 3-4: Different Stages of the batch

According to Figure (3.4), material addition rate to the reactor and tank level can be used as indicators throughout different stages of the reaction. The first indicator (material addition rate) is the process value of the flow control valve during the material addition from the pre-weigh tank to the reactor. The second indicator (tank level) is the reactor level during the batch. The first indicator is part of the pre-weigh tank but initiator is added to the reactor from a source other than the pre-weigh tank. Thus, the first indicator can not be chosen for the stage 2. During the material addition from the pre-weigh tank, the material is added on a controlled flow rate from the pre-weight tank to the reactor. This indicator, although not monotonically increasing or decreasing, can be used as the basis of the start and stop of each stage. The value output indicates the start or stop of each stage. For initiator addition, the material addition to the reactor is done through the initiator tank, therefore this stage is aligned using (Munoz 2004) techniques on aligning the batch data through tank level. For each batch, the value of tank level is slightly different at the end of the stage, thus it is not appropriate to use a fixed tank level among all batches. Therefore, it can be assumed that for a batch i, the initiator stage is at 0% completion when the tank is at initial level (time zero and l-initial). At the end of the stage, when the tank level reaches l-final, the tank level will be at 100% completion. Each batch is re-sampled at level increments given by $\Delta\, l = (lfinal - linitial\,)/(n-1)$ to have n samples from 0% to 100%. This alignment technique assures an equal number of samples,(matrix of raw data have equal rows), and a proper "line up" of batches within the stage. Some of the variables for all batches are plotted prior and after the alignment. Figures (3.5) through (3.12) illustrate the difference between prior and after aligning the data. One may notice non alignment results in figures (3.11) and (3.12). These graphs show the Process Variable 8and Process Value 9. Upon investigation, it was discovered that the once the contents of pre-weigh tanks is transferred to the reactor (end of stage 4), the tank is used to prepare the solution for the next batch. Therefore, non alignment is observed 400 minutes into the data bank of a batch. Appreciating alignment technique that flagged the abnormality in the databank, the data for indicated two variables are plotted in figures (3.13) and (3.14) to show that variables are aligned from stage 1 through the end of stage 4.

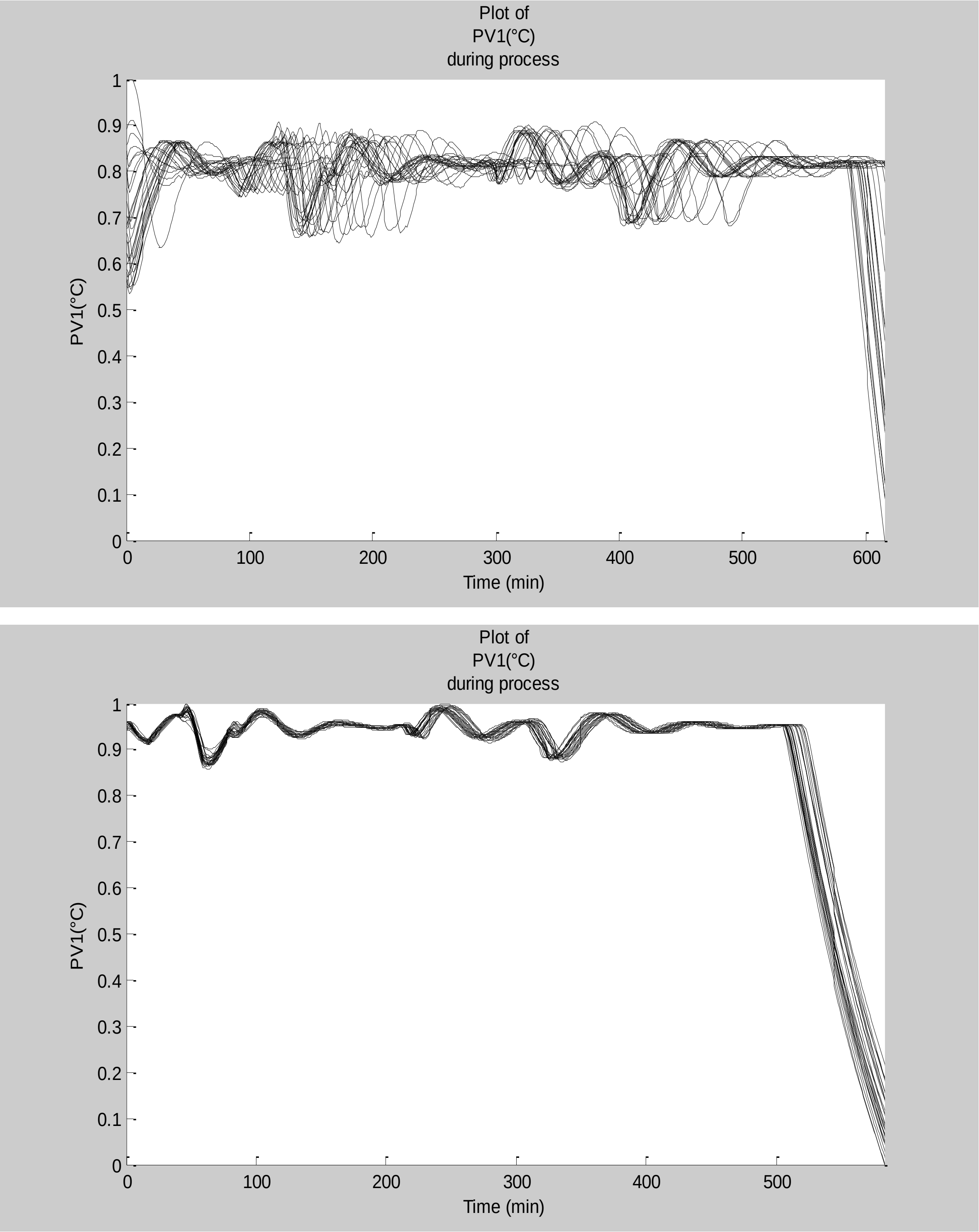


Figure 3-5: Non Aligned & Aligned raw data for Process Variable 1

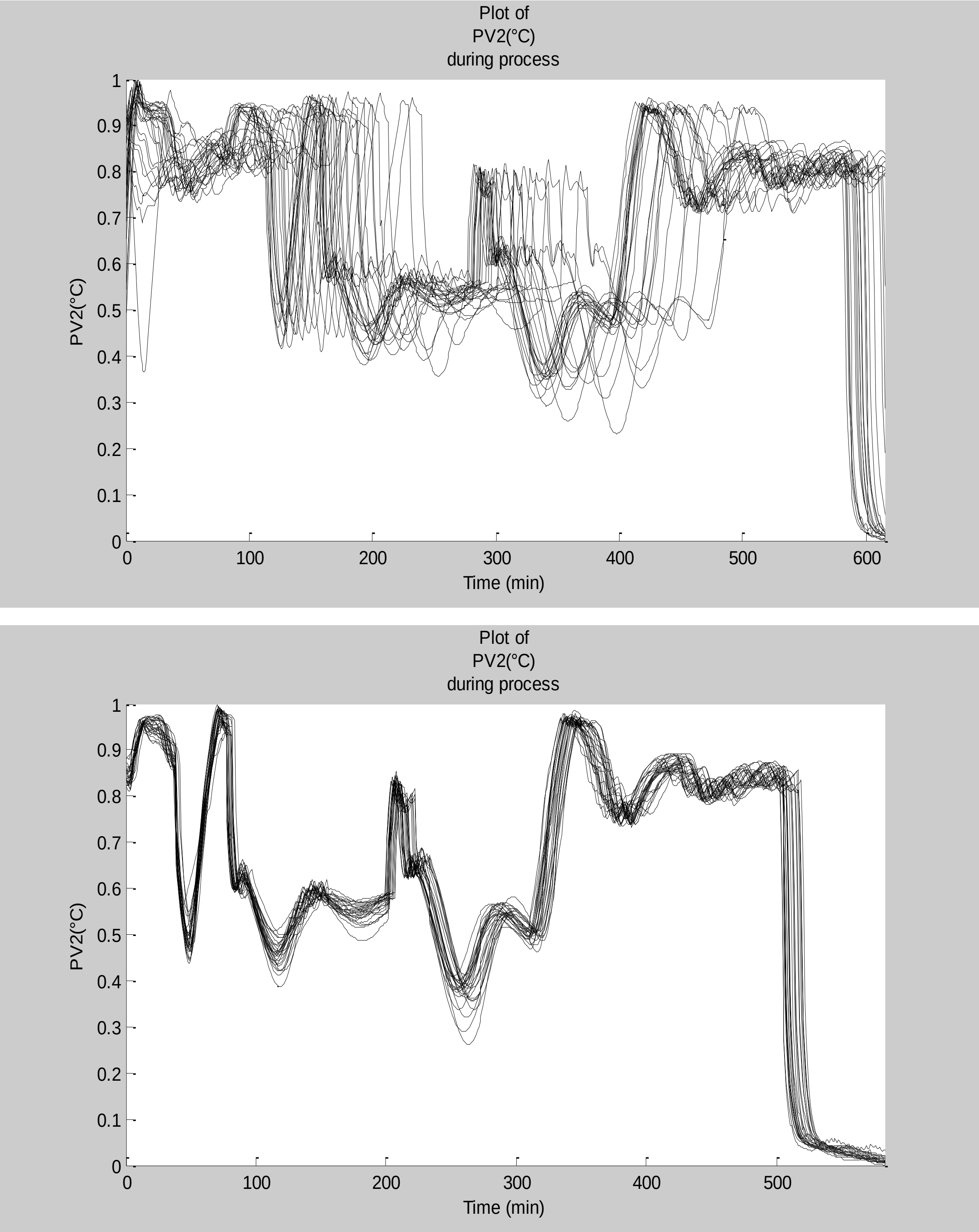
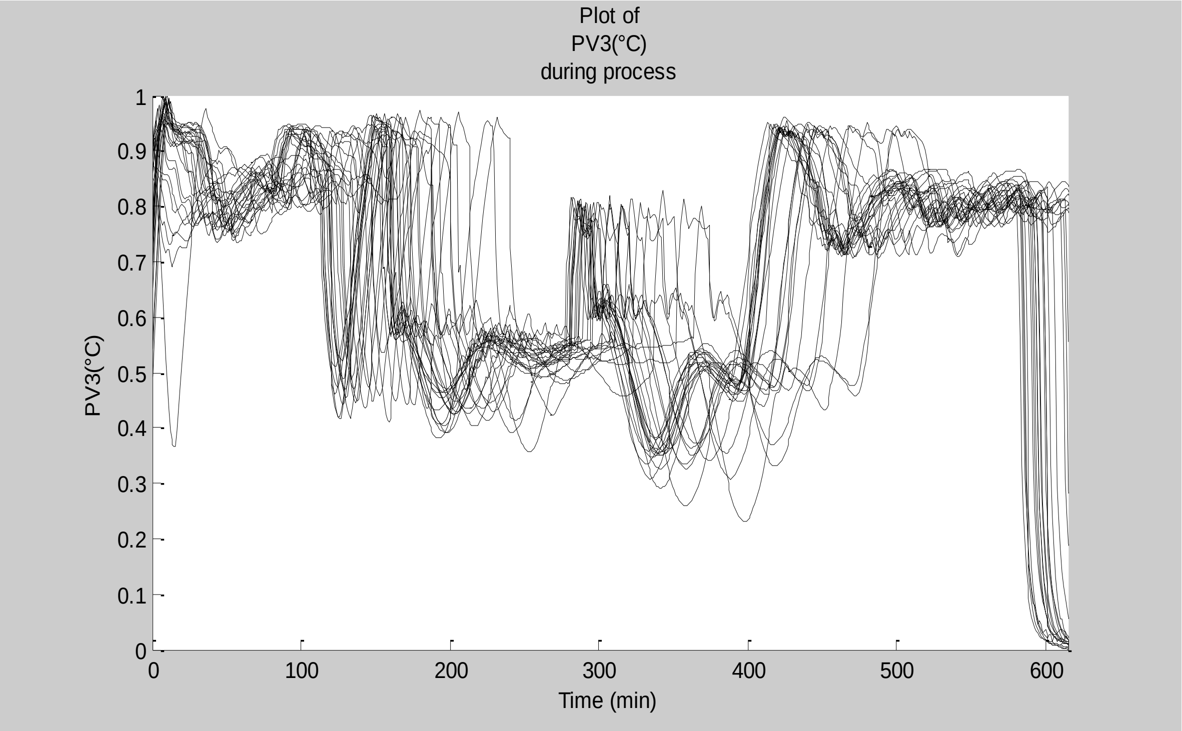


Figure 3-6: Non Aligned & Aligned raw data for Process Variable 2



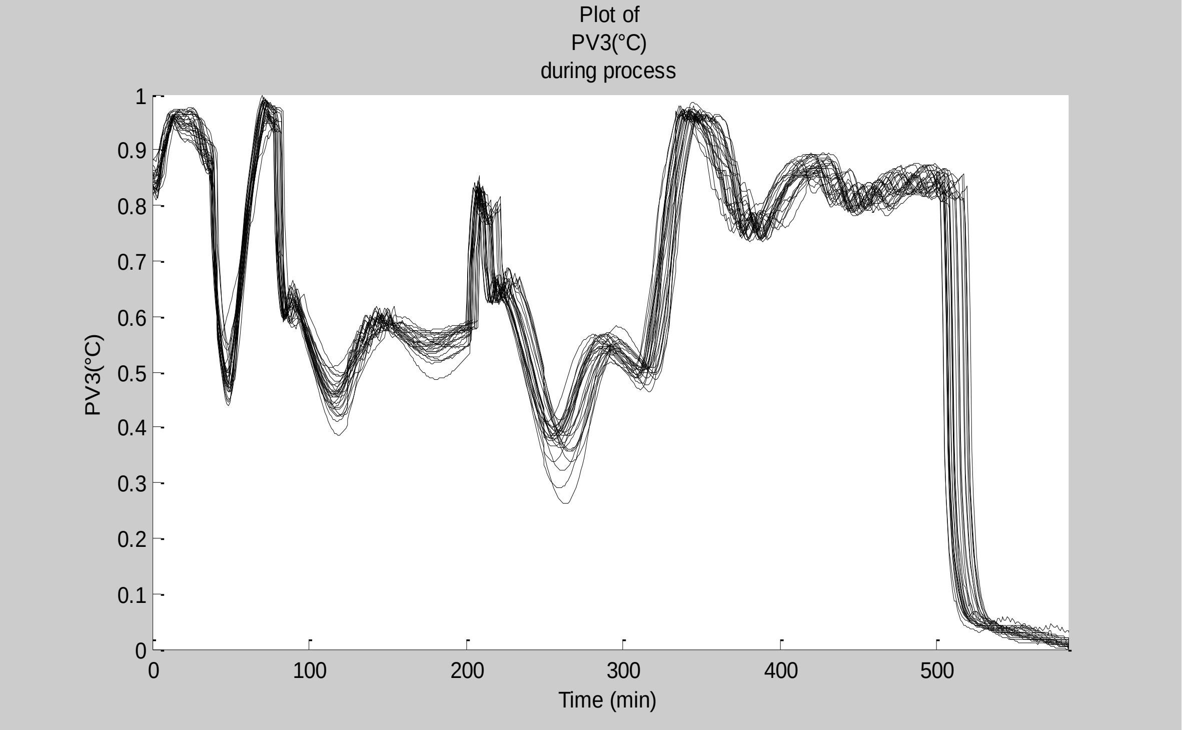
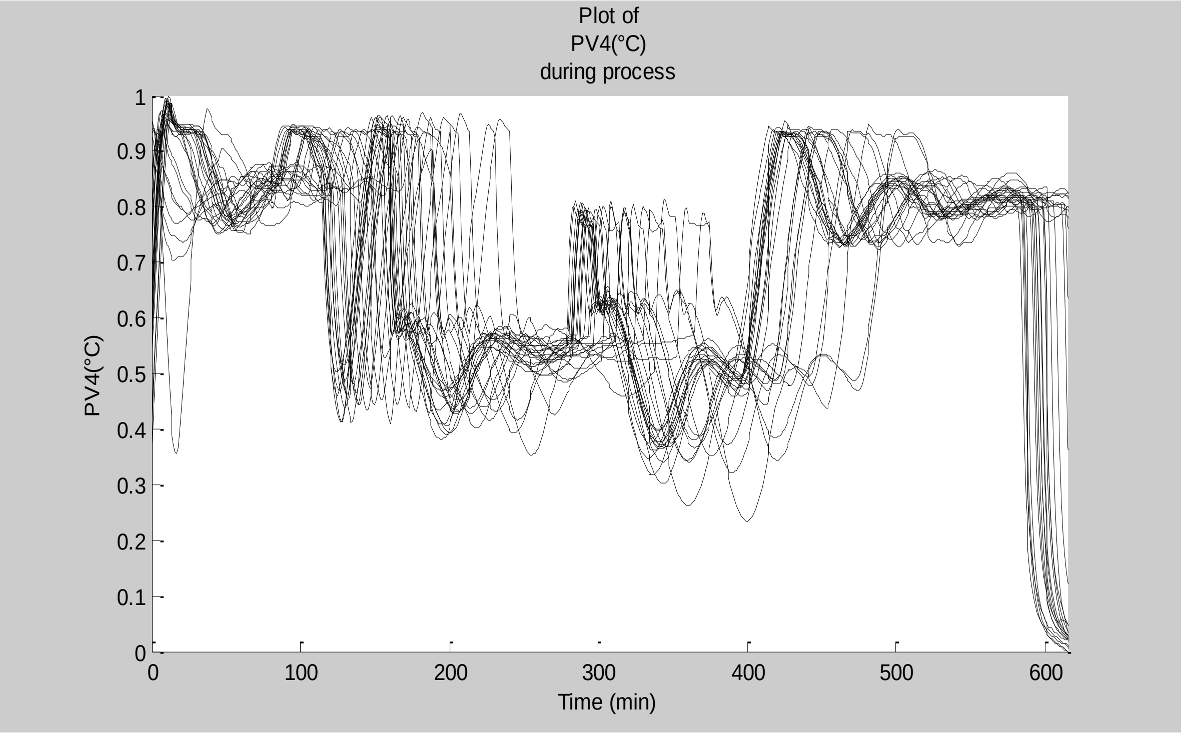


Figure 3-7: Non Aligned & Aligned raw data for Process Variable 3



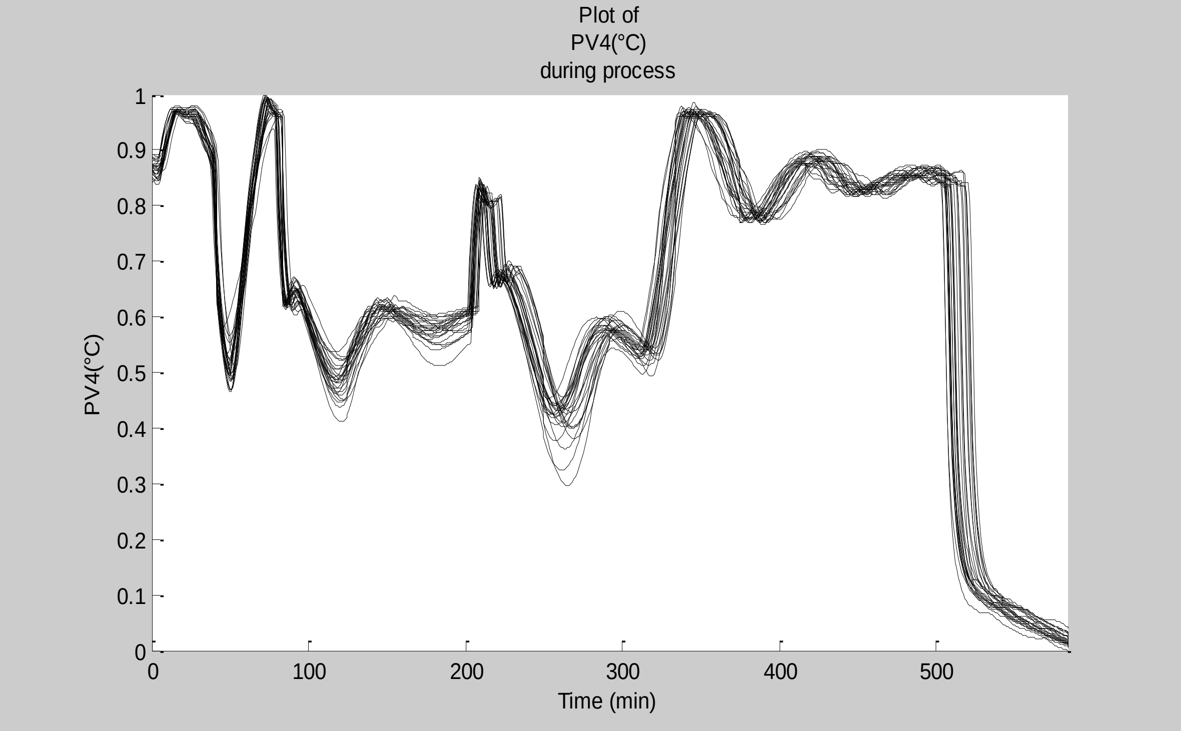
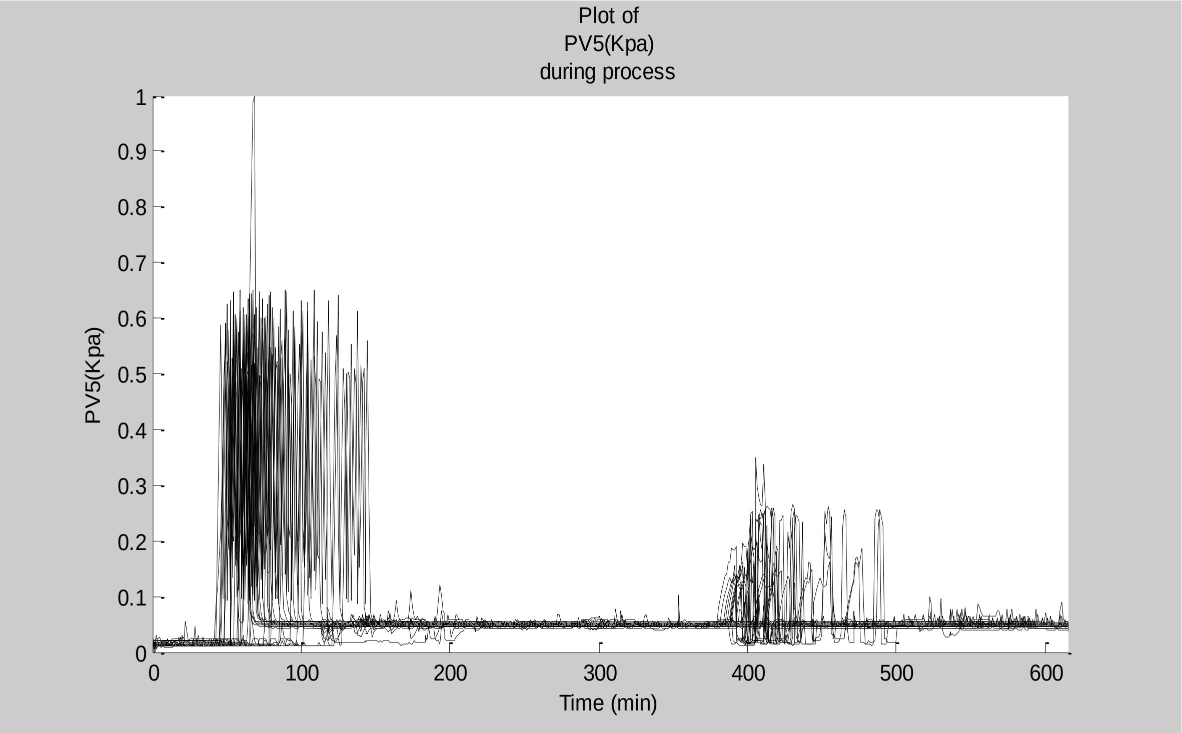


Figure 3-8: Non Aligned & Aligned raw data for Process Variable 4



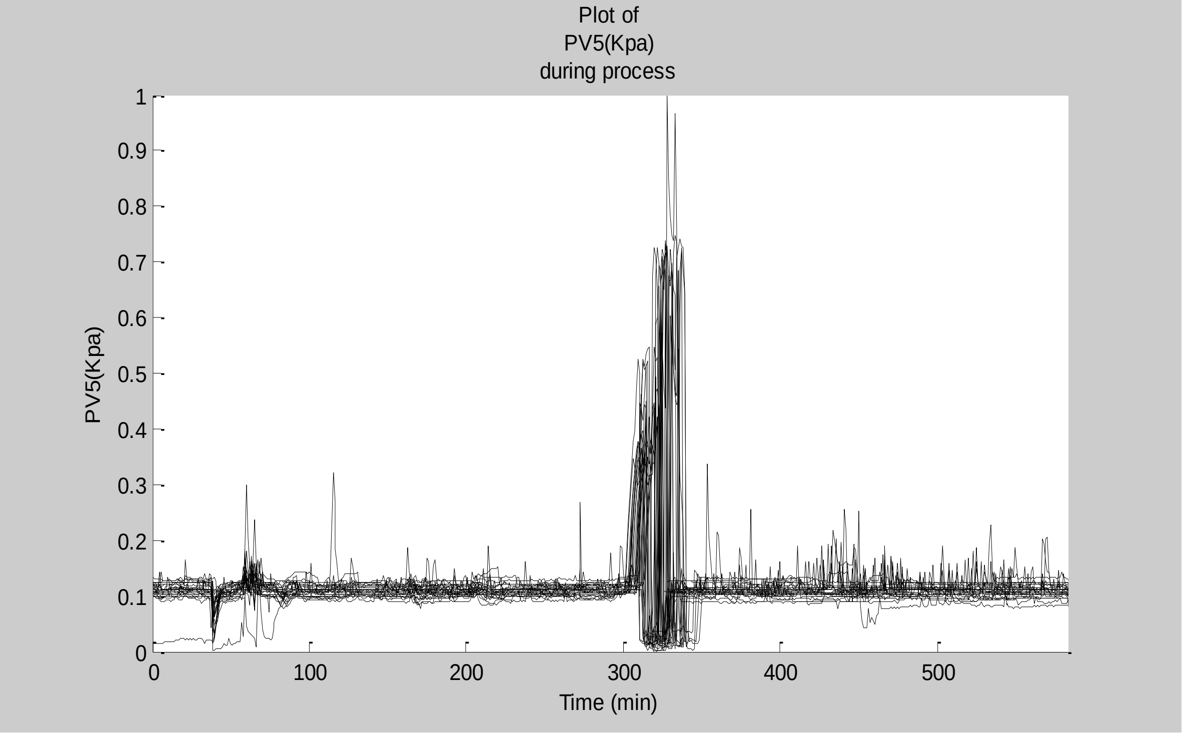


Figure 3-9: Non Aligned & Aligned raw data for Process Variable 5

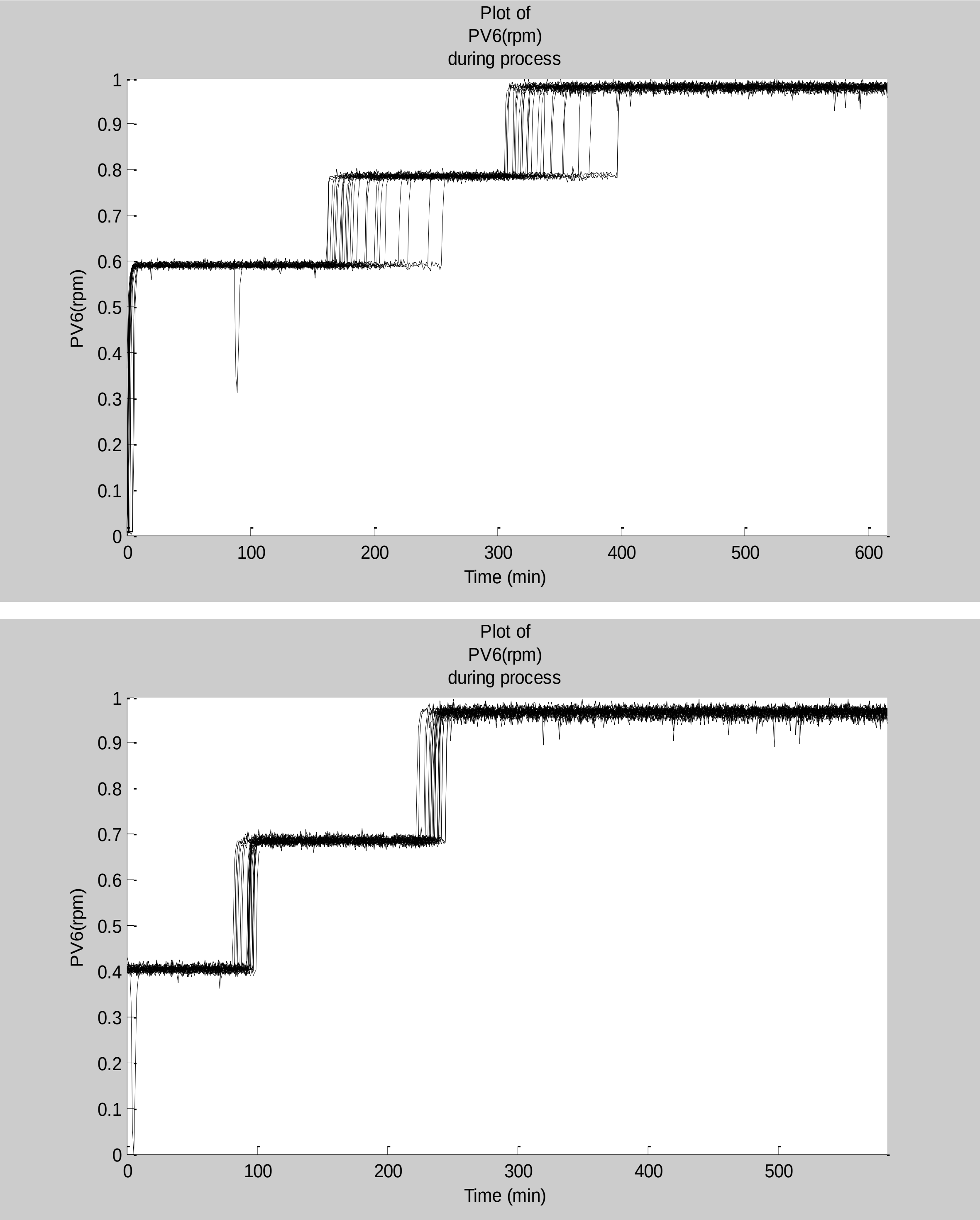


Figure 3-10: Non Aligned & Aligned raw data for Process Variable 6

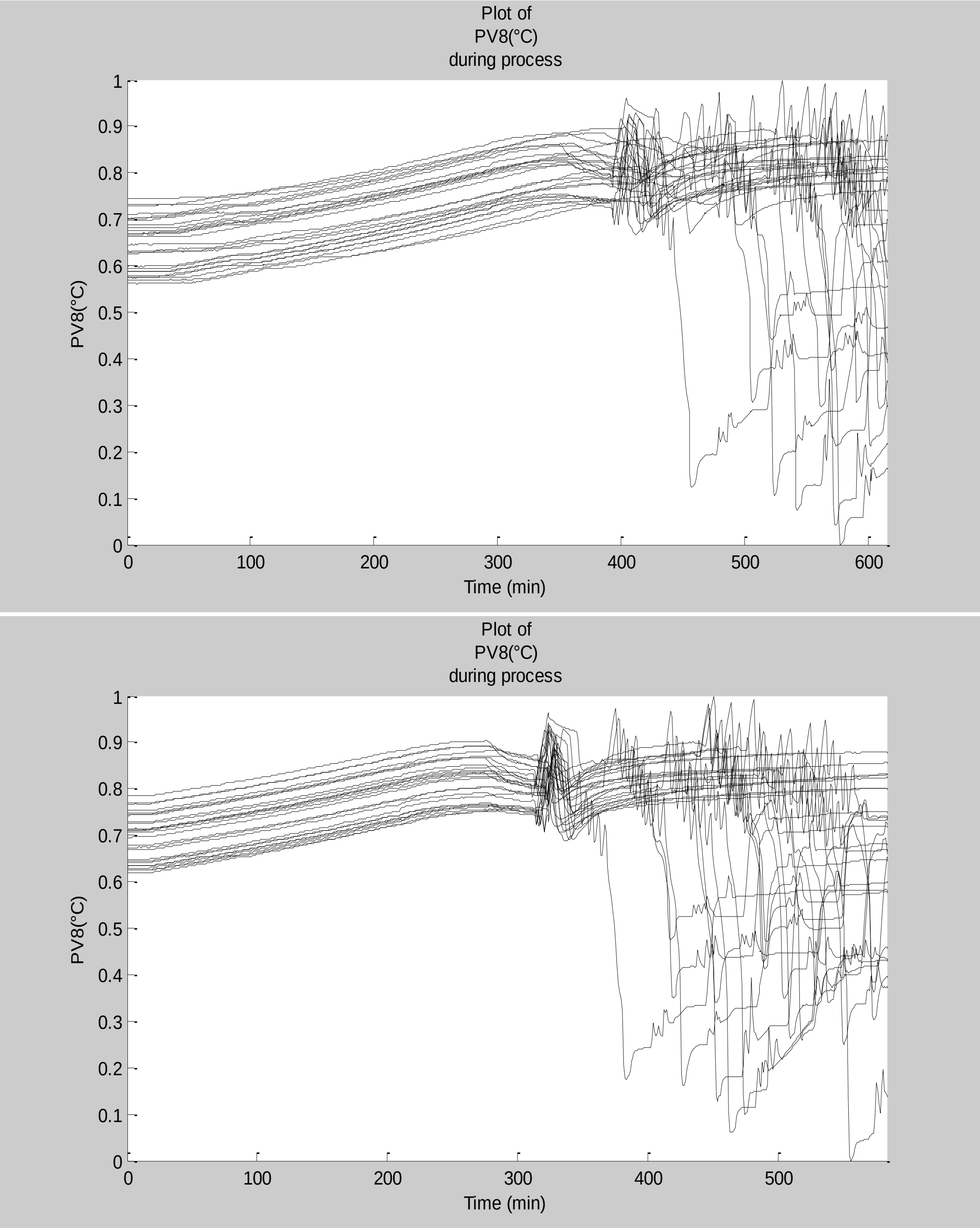


Figure 3-11: Non Aligned & Aligned raw data for Process Variable 8

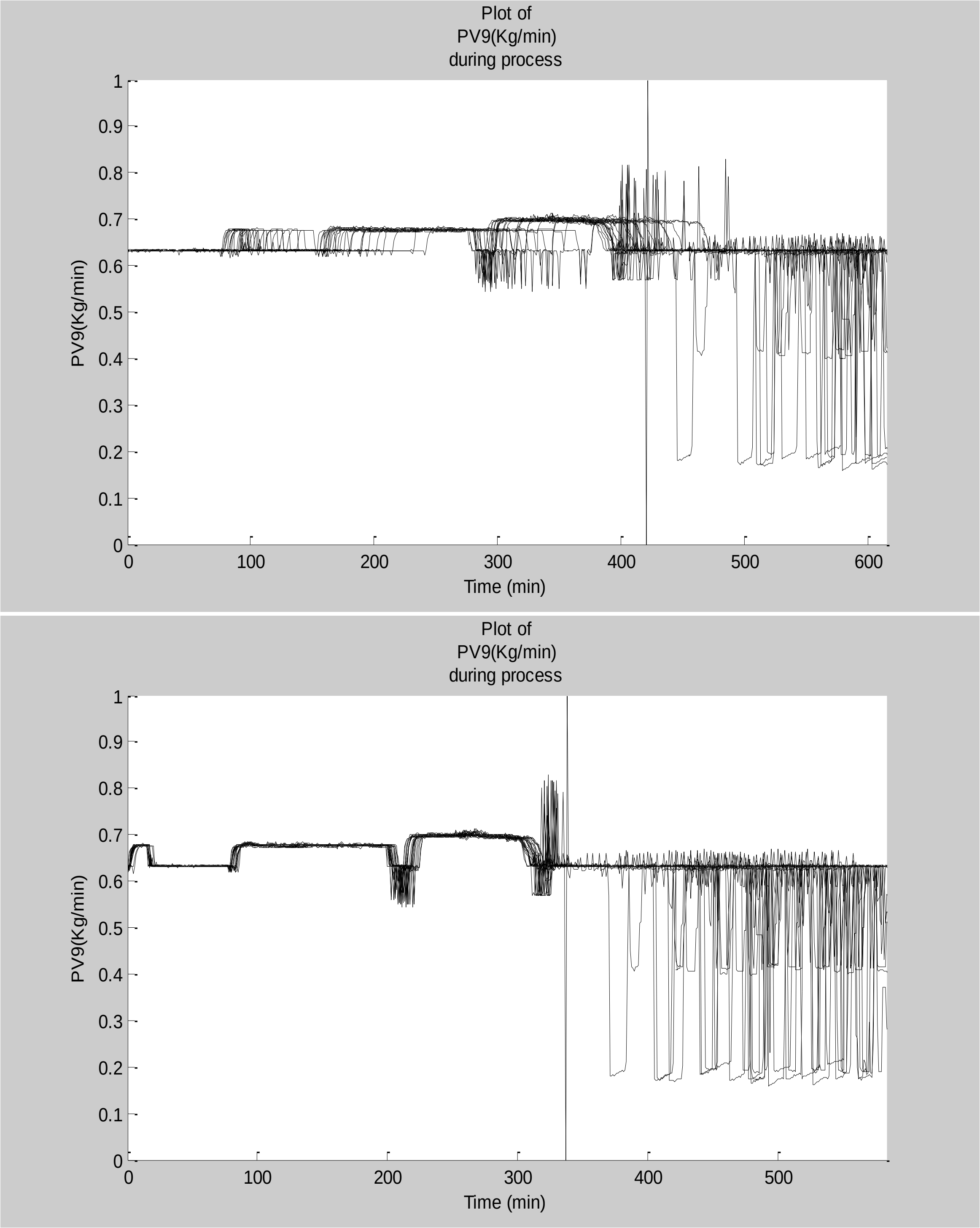


Figure 3-12: Non Aligned & Aligned raw data for Process Variable 9

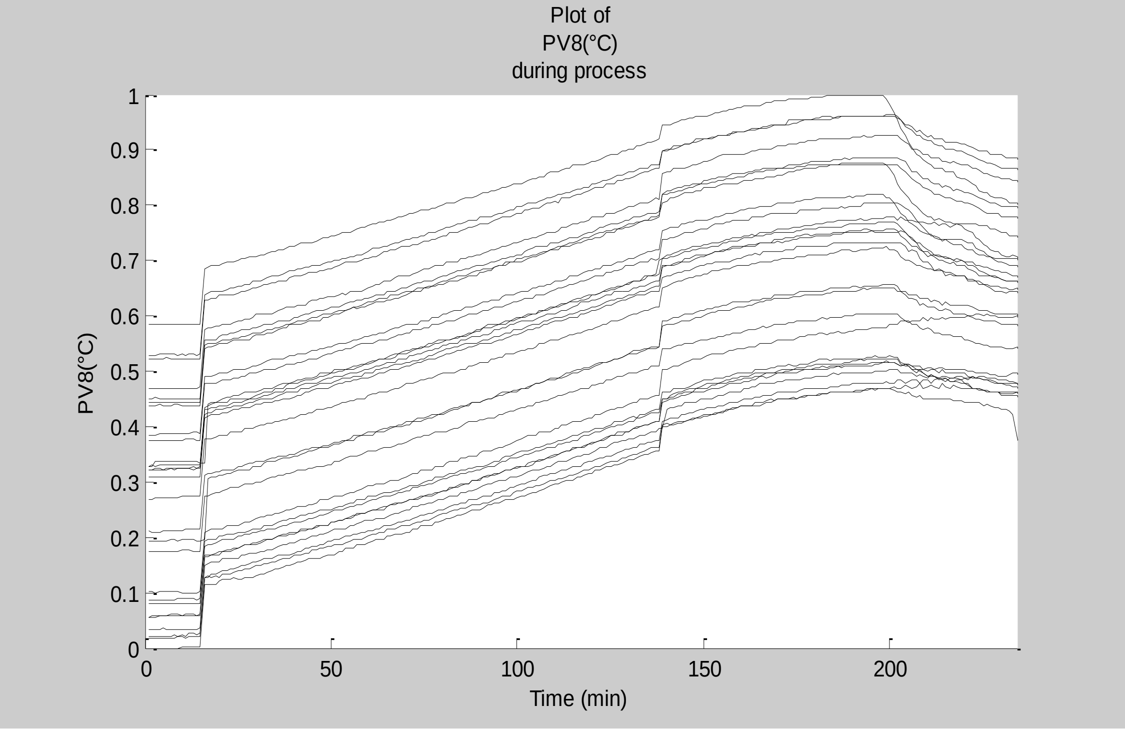


Figure 3-13: Realignment of Process variable 8

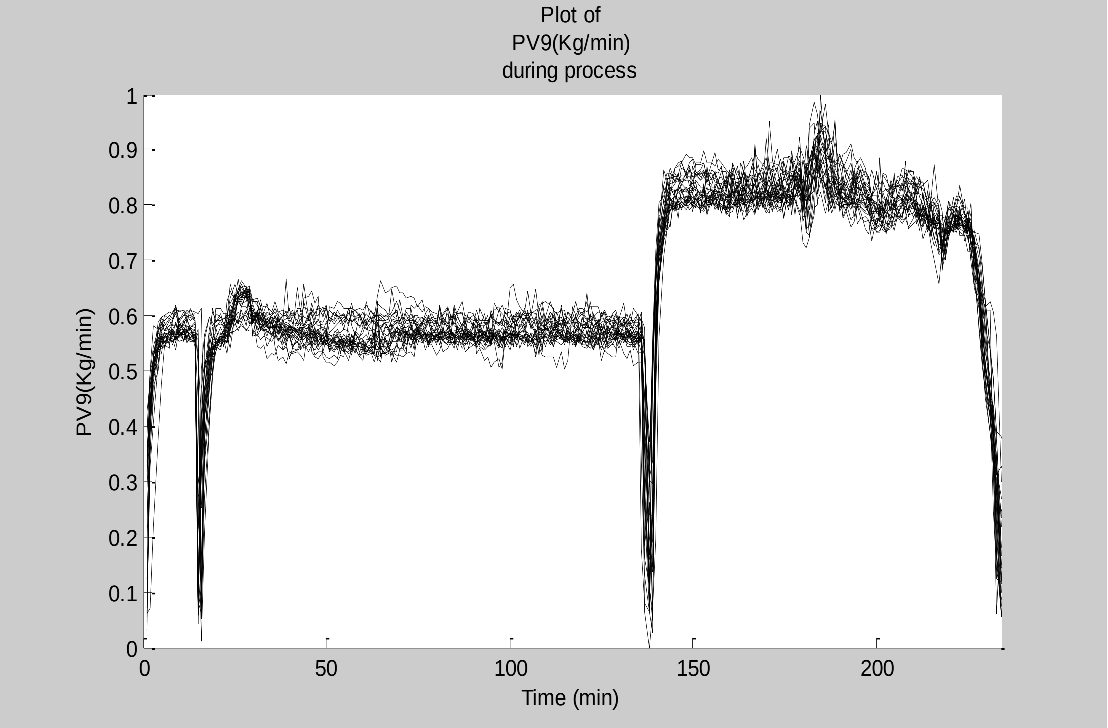


Figure 3-14: Realignment of Process variable 9

3.4 Summary

Centering, scaling, and time alignment were explored as important features in data preprocessing of Chemical Toner batch manufacturing process. It was noted that centering influence the structure of a model whereas scaling explores fitting of a model. Centering was used as a two step procedure to remove the offsets before multi-linear terms are introduced to dataset. Scaling was also explored as a way to change the objection function of a model will be built in next chapter. In this research equal weights assumed for all process variables but individuals having extra knowledge of process could give different weights to process variables. Incorrect centering and scaling will introduce untrue variation and these variations are highly dependant on the data set. The untrue variation would lead to models that are not optimal. Proper combination of centering and scaling was expressed as a two way case. The final formula of a two step centering scaling can be written as:

$$X(centered) = X - X(mean)$$

$$X(centered\ \&\ Scaled) = \frac{X(cenetred)}{StdX(centered)}$$

The second formula is a reminder of equation used for calculating Z values for normal distribution:

$$Z = \frac{X - \ \mu}{\sigma}$$

It may be misleading to assume that the entire data set of process variables after data preprocessing is normally distributed but one should consider the fact that according to figure (3.7) each column of the process values have different mean and standard deviation. Therefore data set can not be considered normally distributed.

The chapter also included handling batches with varying duration which is one of the issues with batch process analysis. The alignment technique enabled us to lay out the same batch progression trajectory for model building. Later in chapter 4, it will be demonstrated that stacking aligned batches is a key factor in fault localization and propagation path in Chemical Toner batch analysis. As the aligned data can be a confirmation of findings of this research from multivariate analysis.

Chapter 4
Process Modeling by Multi-way PCA

In literature review summary, it was discussed that latent variable modeling is an appropriate choice to capture a finger print of a batch process and to provide prediction of process behaviour based on the history of the past successful batches. This chapter reviews the application of Multi-way Principal Component Analysis, a methodology introduced by (Nomikos and MacGregor 1995) for batch process analysis and expanded by (Hong, Zhang and Morris 2011) to include fault propagation path through progressive principal component analysis. These application are applied to two different data sets that were synchronized, centred and scaled in chapter 3.

4.1 Introduction

Consider a rc matrix in which rows are filled sample numbers and columns with variables. Decomposition of matrix X leads to the sum of c outer products of vectors ($r > c$). The first principal component is expressed by, $t_1 = Xp_1$, in which the direction of the highest variability within the dataset is represented by the first loading vector p_1. The second principal component is expressed by $t_2 = Xp_2$, in which the direction of the next highest variability within the dataset is represented by the second loading vector p_2 which is orthogonal to p_1 (MacGregor and Kourti 1995) and (Zhang, Martin and Morris 1997). Thus X can be explained by the linear combination of vectors, t and p through the following equation:

$$X = TP^T = \sum_{i=1}^{c} t(i)\ P(i)^T \qquad (4.1)$$

In equation (4.1), $t(i)$ is the ith score vector and $p(i)$ is the ith loading vector. Once matrix, X, is projected on the vectors, the results are explained by score vectors. Thus value of $t(i)$ represents the variation of X in the direction of $p(i)$. Often, the first few principal components explain the majority of the variation within the data because of the correlation among the variables. Therefore, the first n principal components are expressed as

$$\dot{X} = TP^T \sum_{i=1}^{n} t(i)\ P(i)^T \qquad (4.2)$$

Three methods of computing PCA are Eigenvalue Decomposition (EVD), Singular Value Decomposition (SVD), and Non Iterative Partial Least Square Analysis (NIPALS). Both EVD and SVD have disadvantages to NIPALS. EVD and SVD calculate all principal components at once, and also aren’t capable of handling missing data. In EVD method, calculating large matrixes of data can be difficult. It also is prone to numerical overflow for very large datasets for large matrixes and, it negates the intended benefit of EVD. The only advantages of SVD and EVD to NIPALs are that they drive all the properties of PCA. They also are slightly more accurate since the error is spread over all components. NIPALS gives another insight into what the loadings and scores mean by looking at orthogonality between components. NIPALs method handles missing data and it can be accurate by selecting fewer components. The following steps explains the NIPALS algorithm for sequential calculation of principal components:

1- Unfold $X(I,J,K)$ into $X(I,JK)$
2- Centre and Scale X
3- Select an arbitrary column of X as t
4- $p = X't$
5- Normalize the loadings $p = p/|p|$. This is done to rescale p to magnitude of 1
6- $t = Xp$
7- Move to next step if t has converged, otherwise return to step 4
8- Calculate the matrix of residuals. $E = X - tp'$
9- $X = E$
10- Move to step 3 to extract the next principal component.

Figure (4.1) illustrates the NIPALS algorithm in a flowchart format.

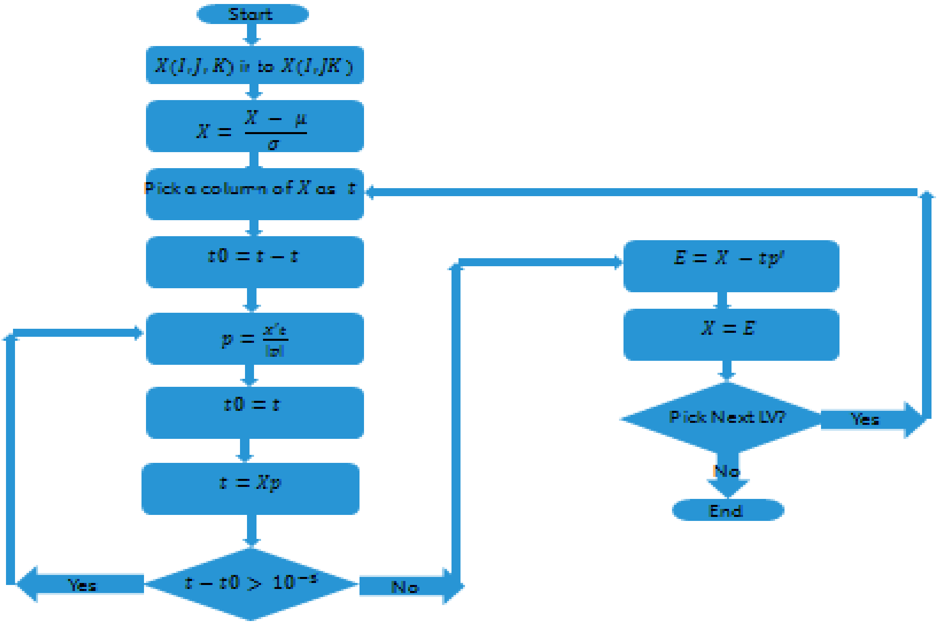


Figure 4-1: NIPALS flowchart for PCA

If the variables are highly correlated, small number of principal components can explain most variation within the data. In batch processes most of the measurement variables are highly cross correlated with one another and auto correlated over the time. The batches are then compared with an multi-way Principal Component Analysis by plotting their score plots of principal components and also their sum of square of errors also known as Standard Prediction Error (SPE) or Q Statistics Q^2.

4.2 Multi-way Principal Component Analysis

Multivariate analysis can be applied to the monitoring of industrial processes to increase their productivity and efficiency (Hong, Zhang and Morris 2011). Batch processes are common in high-value-added chemical manufacturing industries including pharmaceuticals, biochemical, and agriculture. Process monitoring of batch processes are difficult due to complex reactions involved, non-steady state, and batch to batch variations. Besides, control of batch processes is a difficult task because slight shift in process variable may result a product with poor quality. Process monitoring of batch processes by multivariate statistical process control technique through multi-way principal component analysis (MPCA) was developed and implemented by (Nomikos and MacGregor 1995).

Multi-way PCA (MPCA) is considered a data-driven technique in monitoring of batch processes. Data in continuous processes are projected into matrixes with two dimensions, whereas in the batch processes the data are projected into a three-dimensional matrix because the batch number constitutes a dimension. The MPCA method covers this extra dimension.

Ideally, online measurements of process variables can be fed into MPCA to develop real-time monitoring. It is capable of handling highly correlated data including batch process variables because it reduces dimension of data by using principal components methodology. The dimensions of batch data are batch number, process variables, and sampling time of each process variable. In order to apply PCA to data obtained from a batch process, data arrays should be re-organized from three-dimensions to two dimensions through a procedure called unfolding. Several multi dimensional techniques have been proposed for re-organizing batch data arrays into the sum of a fewer vectors and matrixes, and to summarize the variation of the data in the reduced dimension of the these spaces. Two different techniques for data unfolding are discussed by (Nomikos and MacGregor 1994) and (Wold, et al. 1998). MPCA has proved to be a very effective and easy to understand method for analyzing batch data because of its simplicity and well defined properties. In a case where MPCA is used where one of the dimensions is a factor like time, it will give variables a strong dynamic behaviour.

The Nomikos and Mac Gregor (N&M) unfolding approach arranges the data such that rows represent the batch numbers and columns represent process variables at various times during the specified batch number. On the other hand, Wold approach unfolds the data so that rows contain sampling times at different batches and columns contain process variables. According to N&M technique, average value of each column produces average trajectory of process variable associated with the column that is removed through data scaling explained in the previous chapter. However, the Wold approach uses average values that represent each process variable through duration of batches. Therefore, preprocessing performed by N&M method removes nonlinearities of data because it removes of mean trajectories and the results tend to be normally distributed. But, preprocessing by Wold methodology keep nonlinearities within the raw data. Furthermore, MPCA model built by N&M technique has different loadings (weights) for each process variable during a batch. This approach is useful to address nonlinear and time varying behavior of batch processes while modeling but MPCA technique proposed by Wold has the same loadings (weights) for individual process variables throughout a batch. Consequently, the N&M modeling technique is more applicable in batch process monitoring. But Wold approach facilitates the study of variation of each process variable throughout the batch. Later in this chapter, the Wold approach will be used to establish a fault localization path.

Consider data obtained from I batches and J process variables are measured at K time samples within each batch. Batch data array, three-dimensional matrix, can be unfolded according to Figure (4.2). Matrices of scores and loadings are computed upon the construction of a PCA model with the unfolded data. Two statistical indicators, SPE (also known as Q) and T^2, are computed by scores and loadings of the PCA model.

$$T^2 = \sum_{i=1}^{n} \frac{t^2(i)}{\sigma^2(i)} \qquad Hotelling's\ T^2\ Statistics \ (4.3)$$

Figure (4.2) describes the unfolding of the matrix batch data for our data set consisting of 36 batches with 10 variables and 228 time units in two different unfolding methods.

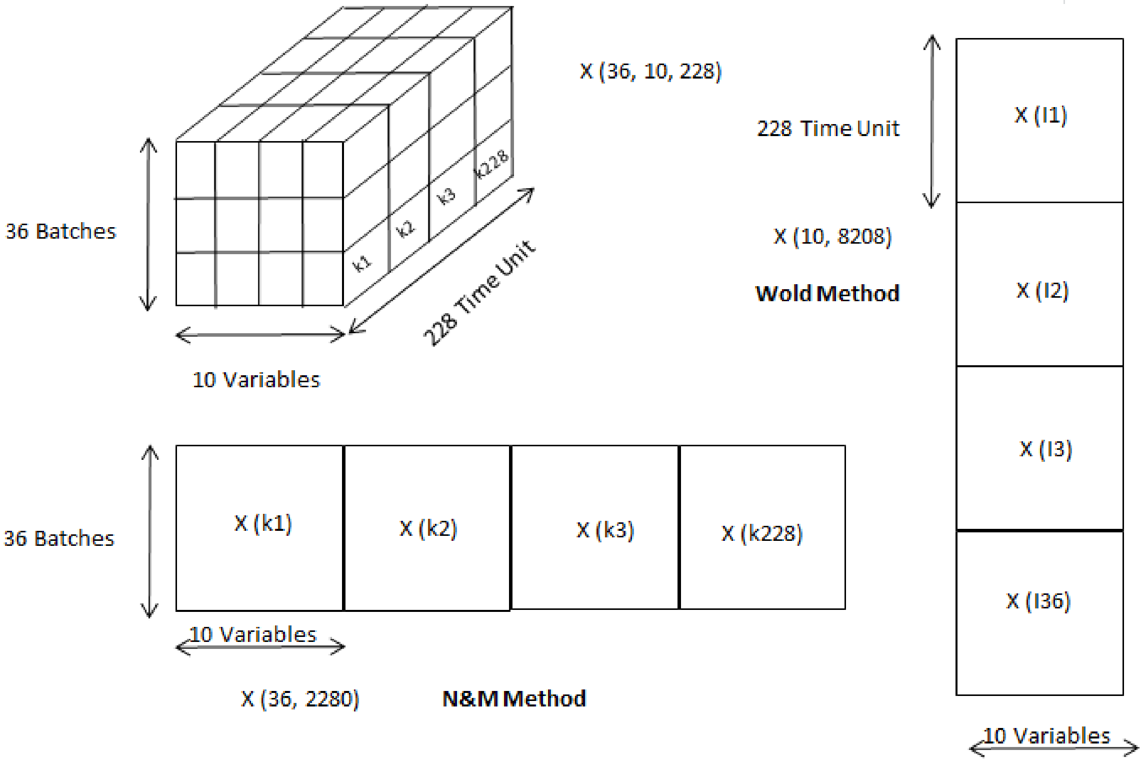


Figure 4-2: Unfolding batch data from 3D to 2D

In Equation (4.4), n represents the number of principal components, and $\sigma(1)$ to $\sigma(n)$ are the standard deviation of T. The standard deviations can also be connceted to the square root of the eigenvalues of the covariance matrix $X'X$ (Hong, Zhang and Morris 2011).

$$\sigma(i) = \sqrt{\lambda(i)} \qquad (\, i = 1,2,3, \ldots\ldots, n) \qquad (4.4)$$

$$SPE(i) = \sum_{j=1}^{j} SPE\,(ij) \qquad Q\ Statistics\ (4.5)$$

$$SPE(ij) = \sum_{k=1}^{K} (x(ij,k) - \hat{x}(ij,k))^2 \ \ (4.6)$$

In the above equations, $SPE(i)$ explains the squared prediction error of batch i, $x(ij,k)$ contains value of variable j in the batch i at the time k. $\hat{x}(ij,k)$ is the model prediction for $x(ij,k)$. $SPE(ij)$ represents sum of the SPE containing variable j in the ith batch. When a fault that is caused by significant changes in the value of a process variables occurs, it can be detected through a significant increase in T^2. The other statistic,Q, is squared prediction error that indicates the predictability of data by PCA model. In new batches, the prediction of the PCA model is calculated, and then compared with the historical values of process variables. Large Q statistics is an indication of a fault that affects the correlation structure among process variables (Hong, Zhang and Morris 2011).

4.3 Model Cross Validation

In order to understand if the MPCA model is acceptable in describing the data bank of 36 batches and is best quantifying hypothesized relationships between process variables obtained from NIPALS analysis, we require applying a model validation technique. The validation process can be analyzed the goodness of fit of the regression and checking whether the model's predictive performance when applied to data that was not used in model building.

Coefficient of determination R^2, a number that describes the variation within the data set, does not guarantee that the model is the best fit to the data because a high R^2 can occur by misspecification of the functional form of a relationship or in the presence of outliers that distort the true relationship between components of a model. Another problem with selecting R^2 as a measure of model validation is that is increased by adding more variables into the model. Therefore, a model fit based on high R^2, is not a good approach on how well a model predict. It is to over-fit a model by including too many variables or degrees of freedom to inflate coefficient of determination.

Cross-validation is a model evaluation method that assesses the generalized impact of a statistical analysis to an independent data set. There are several methods to cross validate an statistical model. These methods are Holdout Method, K-fold Cross Validation, and Leave-One-Out Cross Validation.

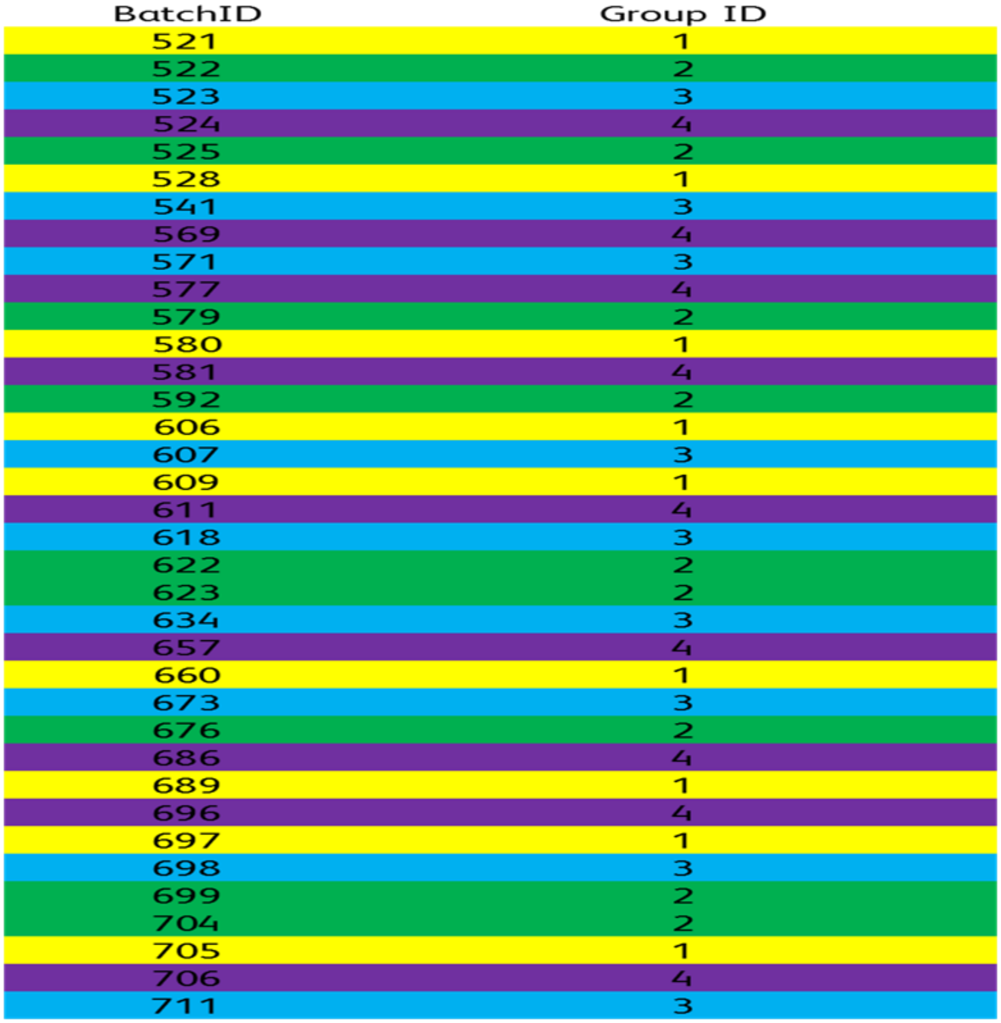
Handout Method: hand out method also called 2-fold cross-validation is the most convenient form of cross validation techniques. The data set is split into two equal training and testing sets, and then an empirical model is built fitting the training data set. Following the model building, the training dataset is projected to the model to predict the output of the model. The errors of the projection are then accumulated in the forms of sums of squares to provide the mean absolute test errors which will be used to assess the model. Faster computing times and model simplicity are the advantages of this method whereas high variance in the evaluation can count against applying handout method. The evaluation depends on the individual data points in the training and testing sets. Therefore, depending how the division is made, the evaluation errors may be significantly different.

K-fold Cross Validation: an improvement to 2-fold cross validation, the dataset is divided into k sets and then the hold out method is repeated k times. Each time, one dataset is set aside as a testing set and other data sets are put together as the training set. Upon splitting, the average errors among all trials are calculated. The advantage of this technique to hand out method is that as the number of k increases the variance in the resulting estimates decreases because the importance of how the data set is divided into k sub groups decreases. In this method each data points gets projected to the testing set once. The disadvantage of this method is that the computation time is much longer than hand out method because the training algorithm has to be rebuilt for every repetition.

Leave-One-Out Cross Validation: in this method one observation is used as the testing data and the rest of the observations used as the training data set. This is an extended K-fold Cross validation such that the number of repetitions, K, is equal to the number of data points; N. the variation error in projecting the testing data to the model is calculated similarly to K-fold Cross Validation. The advantage of this method is that the error results are reasonably reliable even when the number of observation isn't high. The disadvantage of this method is that as the number of observation increases the computing time increases significantly and it becomes very costly to cross validate a model through this technique.

Considering the data size (36 batches) and computing time, in this research, it was decided to perform a K-fold Cross Validation with, K = 4, with our data set. This approach is a combination of techniques used by (Lin, et al. 2006) and (Abdi and Williams 2010) with differences arising from the size of data set.

Firstly, the data set is split into 4 randomly selected groups. The group classification for 36 batches in the data set is illustrated in figure (4.3). To better visualize the technique, yellow, green, blue, and magenta colours are assigned to groups 1 through 4 respectively.



BatchID	Group ID
521	1
522	2
523	3
524	4
525	2
528	1
541	3
569	4
571	3
577	4
579	2
580	1
581	4
592	2
606	1
607	3
609	1
611	4
618	3
622	2
623	2
634	3
657	4
660	1
673	3
676	2
686	4
689	1
696	4
697	1
698	3
699	2
704	2
705	1
706	4
711	3

Figure 4-3: Classification of data set for K-fold Cross Validation

Then a model was fit with one latent variable with groups 2,3, and 4 as the training data using equation (4.7), and R^2 was calculated using equation (4.8)

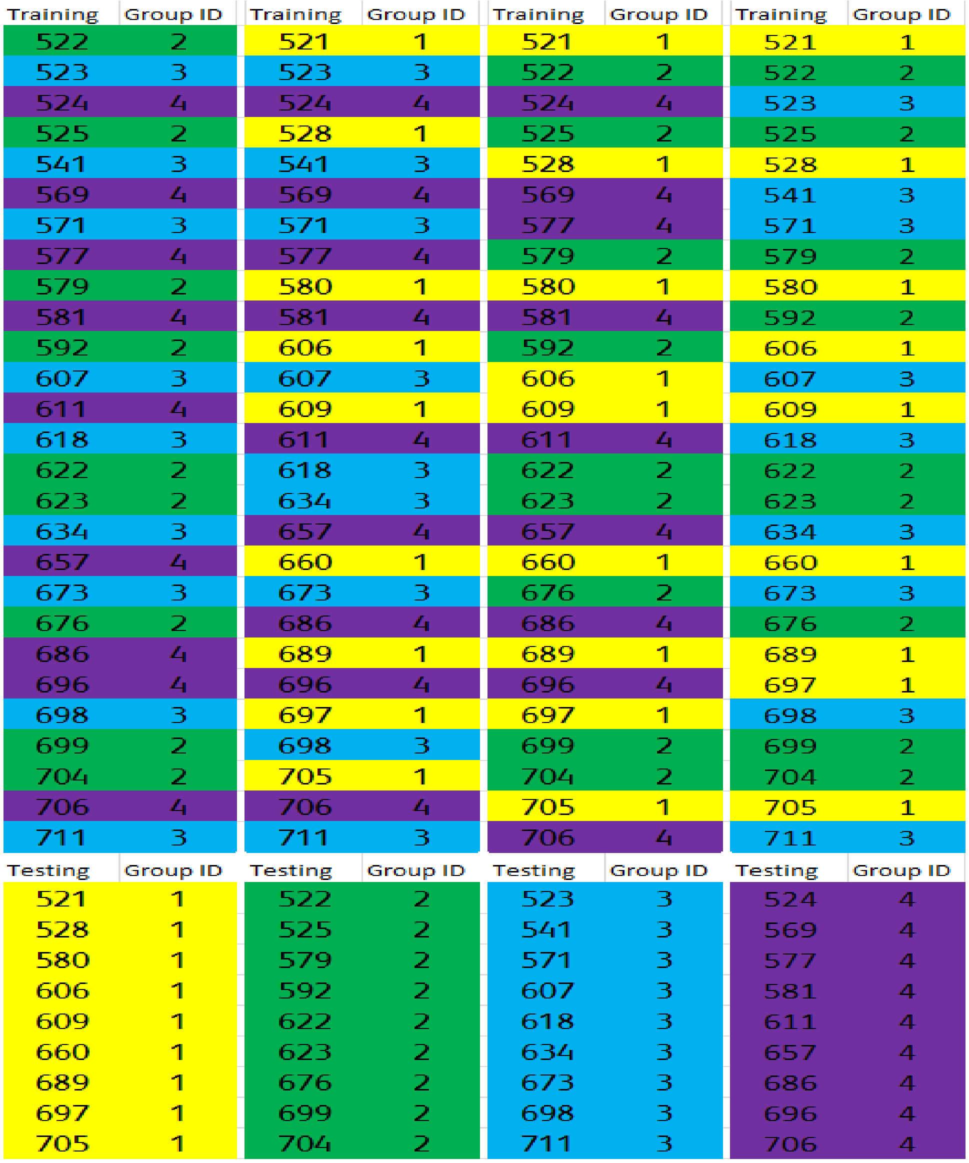
$$X = TP' + E \quad (4.7)$$

$$R^2 = 1 - \frac{Var\ (E)}{Var\ (X)} \qquad (4.8)$$

Once the model was in place, group 1 was projected to the model as the testing group and calculated the norm of residuals as illustrated by (Abdi and Williams 2010) using equation (4.9)

$$E_1 = ||X_1 - \widehat{X_1}||^2 \quad (4.9)$$

The above procedure was repeated with other groups 2,3, and 4 as the remaining testing groups. Figure (4.4) illustrates the 4-fold Cross Validation technique on data set.



Training	Group ID	Training	Group ID	Training	Group ID	Training	Group ID
522	2	521	1	521	1	521	1
523	3	523	3	522	2	522	2
524	4	524	4	524	4	523	3
525	2	528	1	525	2	525	2
541	3	541	3	528	1	528	1
569	4	569	4	569	4	541	3
571	3	571	3	577	4	571	3
577	4	577	4	579	2	579	2
579	2	580	1	580	1	580	1
581	4	581	4	581	4	592	2
592	2	606	1	592	2	606	1
607	3	607	3	606	1	607	3
611	4	609	1	609	1	609	1
618	3	611	4	611	4	618	3
622	2	618	3	622	2	622	2
623	2	634	3	623	2	623	2
634	3	657	4	657	4	634	3
657	4	660	1	660	1	660	1
673	3	673	3	676	2	673	3
676	2	686	4	686	4	676	2
686	4	689	1	689	1	689	1
696	4	696	4	696	4	697	1
698	3	697	1	697	1	698	3
699	2	698	3	699	2	699	2
704	2	705	1	704	2	704	2
706	4	706	4	705	1	705	1
711	3	711	3	706	4	711	3

Testing	Group ID	Testing	Group ID	Testing	Group ID	Testing	Group ID
521	1	522	2	523	3	524	4
528	1	525	2	541	3	569	4
580	1	579	2	571	3	577	4
606	1	592	2	607	3	581	4
609	1	622	2	618	3	611	4
660	1	623	2	634	3	657	4
689	1	676	2	673	3	686	4
697	1	699	2	698	3	696	4
705	1	704	2	711	3	706	4

Figure 4-4: 4-fold Cross Validation of data set

For each cross validation repetition, norm of residuals and the Prediction Error Sums of Residual were calculated using equation (4.10)

$$\begin{aligned} PRESS = {} & E_1 + E_2 + E_3 + E_4 \\ & = ||X_1 - \widehat{X_1}||^2 + ||X_2 - \widehat{X_2}||^2 + ||X_3 - \widehat{X_3}||^2 + ||X_4 - \widehat{X_4}||^2 \quad (4.10) \end{aligned}$$

Once $PRESS$ and R^2 were calculated for each component, a model was fitted with the calculated component (latent variable). Thus 4 fold cross validation technique was repeated for each subsequent latent variable derived from the data set. The $PRESS$ was then plotted following (Lin, et al. 2006) recommendations to measure the predictability of the model, however values of R^2were recorded to measure the amount of variability explained by the model. Figures (4.5) and (4.6) show the cumulative $PRESS$ and R^2 for number of latent variables added to the model.

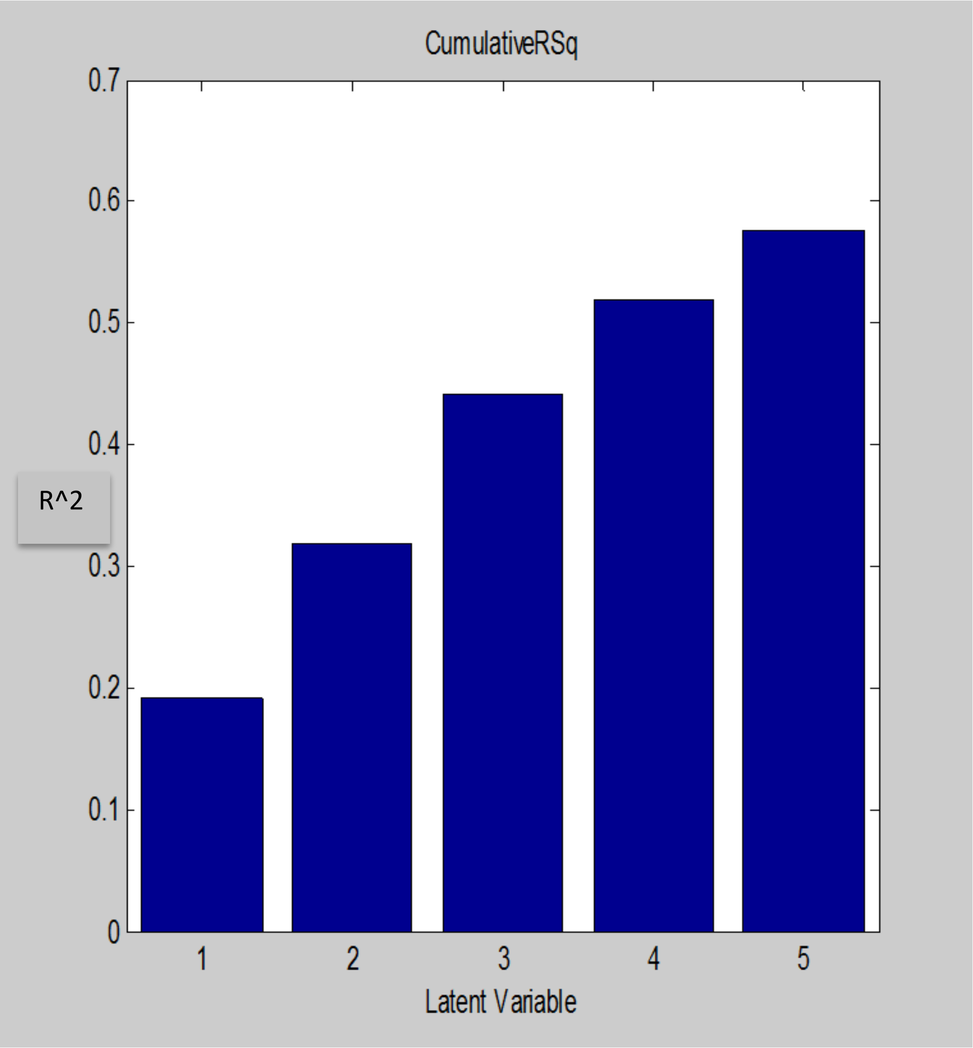


Figure 4-5: Cumulative R-square value

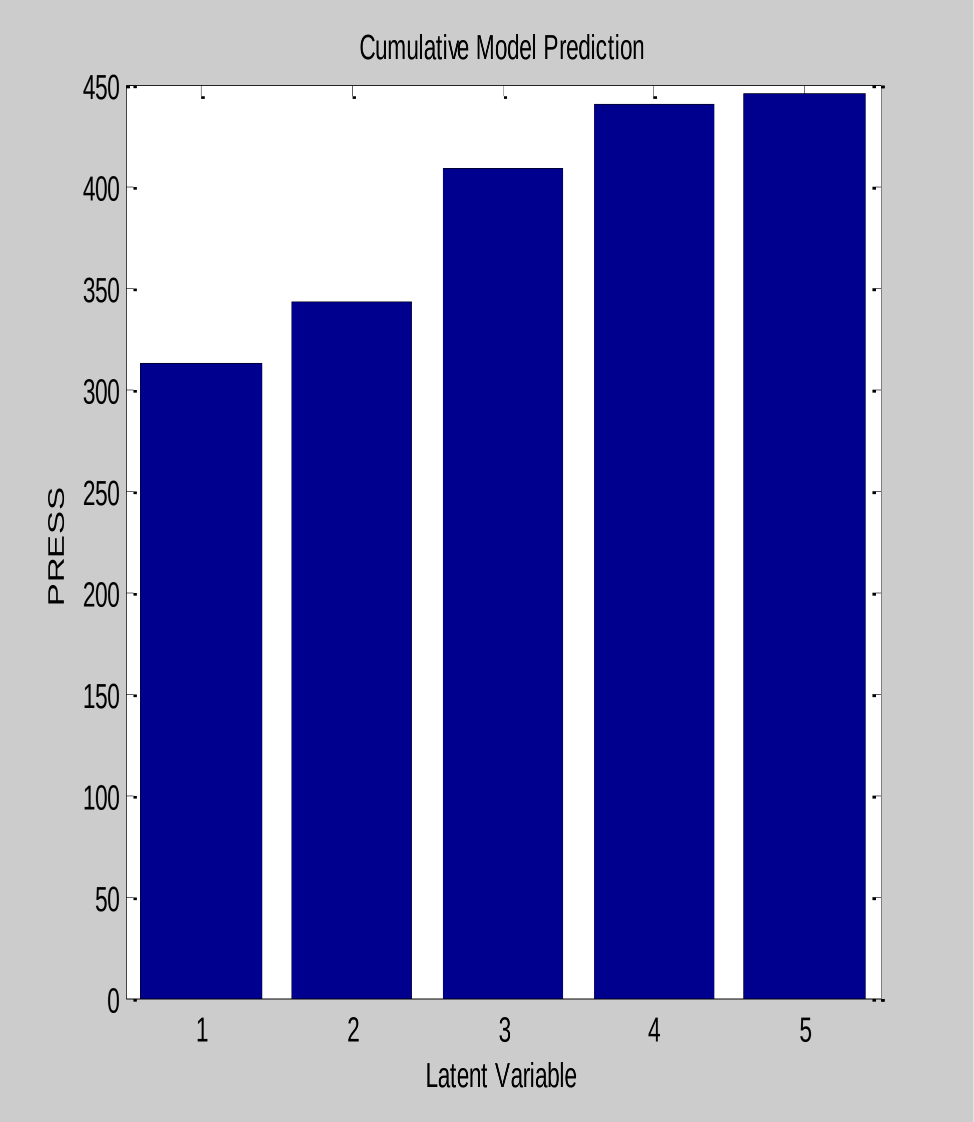


Figure 4-6: Cumulative PRESS value

As illustrated in the above figures, R^2 value is 58% indicating the model explains 58% variability within the process data however, the 5th latent variable addition to the model have only explained 6% of variability within the process data. A quick review of $PRESS$ indicates that the 5th latent variable addition to the model have only increased 1.3% of predictably of data set. Given the number of data points,82080, involved in this analysis (36 batches, 10 variables and 228 time units), it can be concluded that a model with 5 latent variable would have the best predictability of the process data with minimal chance of over fitting. Yet the 58% value of R^2indicates that there still some values of importance that might not be explained by the model. Thus justifying the need to consider analyzing E in equation (4.7) .

4.4 Fault Localization

A MPCA was performed on the data bank of 36 batches of process variables and result was plotted. Then the process variables data of batch 517 which has out of specification quality (the quality results for this batch is discussed in section 5.2) were projected to the model. Figure (4.7) represents the projection of trajectories of process values of each batch onto the reduced plane defined by the first two latent variables. The cut of point of the NIPALs algorithm, demonstrated by the cross validation technique in the previous section, is at 5 principal components. Figure (4.7) indicates that process values of batch 517 do not belong to the same cluster as the model.

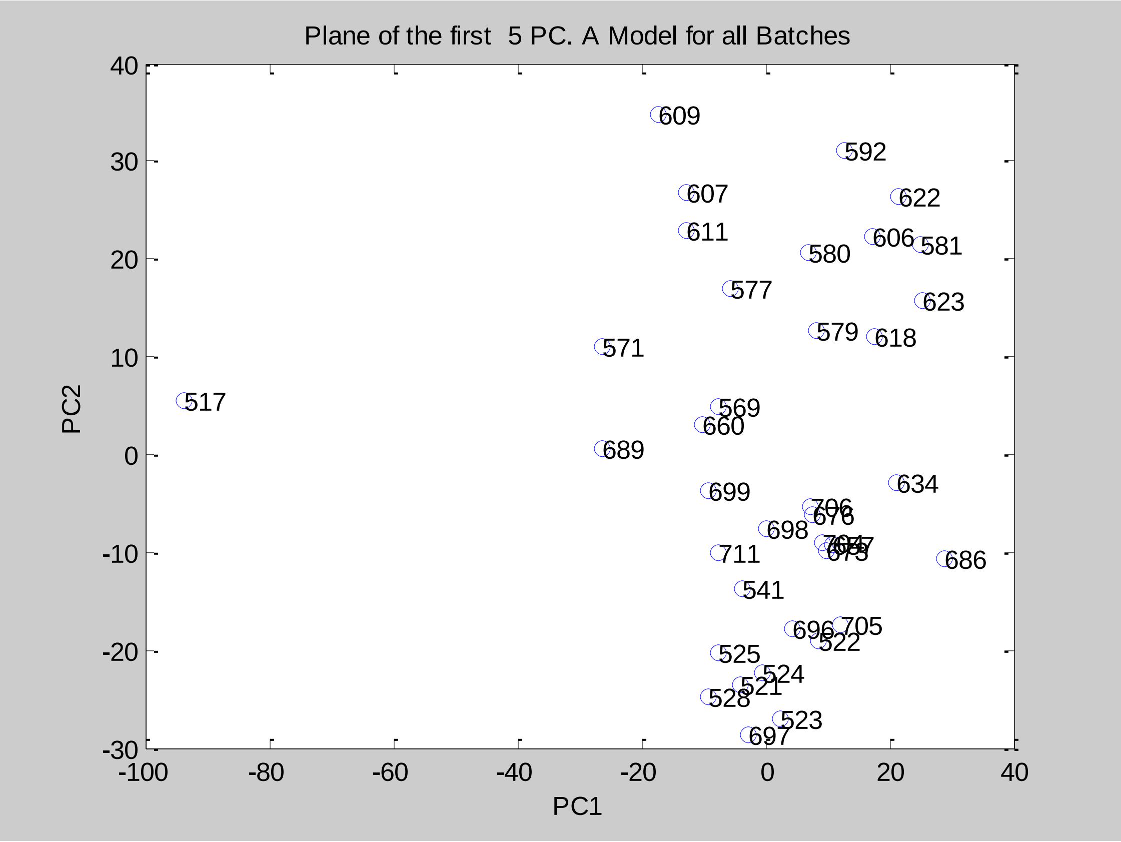


Figure 4-7: Plot of the first two latent variables for process

Hotelling's number, T_i^2, is the multivariate analog of the two samples $t - test$ in univariate statistics. Figure (4.8) represent the first two principal components with the 95% confidence limit with hotteling's distribution. T_i^2 is the summary of all calculated components within the row i of r principal components and is expressed in the flowing equation in which s_r is the standard deviation of the score column of r. Calculation of Hotelling's number is shown in equation (4.11). Visiting Figure (2.1) that visualizes how principal component analysis projects the original data into a lower dimension, it can be concluded that equation (4.11) is another step in the dimension reduction. Thus enabling the explanation of the model with one single value which is Hotelling's number. Subsequently, as described by equation (4.5) the prediction error of the model can be explained by SPE. Therefore, behaviour of a new batch can be explained by only two values derived from the history of past successful batches.

$$T_i^2 = \sum_{i=1}^{r=r} (\frac{t_{i,r}}{s_r})^2 \quad (4.11)$$

In the case study, the calculation of T_i^2 for each batch is reduced to the following equations (equation of an ellipse) where i is the number of batches, r is the number of principal components, and ∝ is the confidence limit.

$$T_{2,,0.05}^2 = \frac{t_1^2}{s_1^2} + \frac{t_2^2}{s_2^2} \quad (4.12)$$

$$T_{r,\alpha}^2 = \frac{(i-1)(i+1)r}{i(i-r)} \quad (4.13)$$

Figure (4.8) shows that for the first two latent variables, batch 517 fall significantly outside the cluster of other batches which indicate an abnormal process condition during the batch run. All batches that behave similarly should cluster in the same zone of the space described by the latent variables. In this case, since the variations explained by the first two latent variables are close to each other, review figure (4.5), the ellipse of figure (4.8) is similar to a circle.

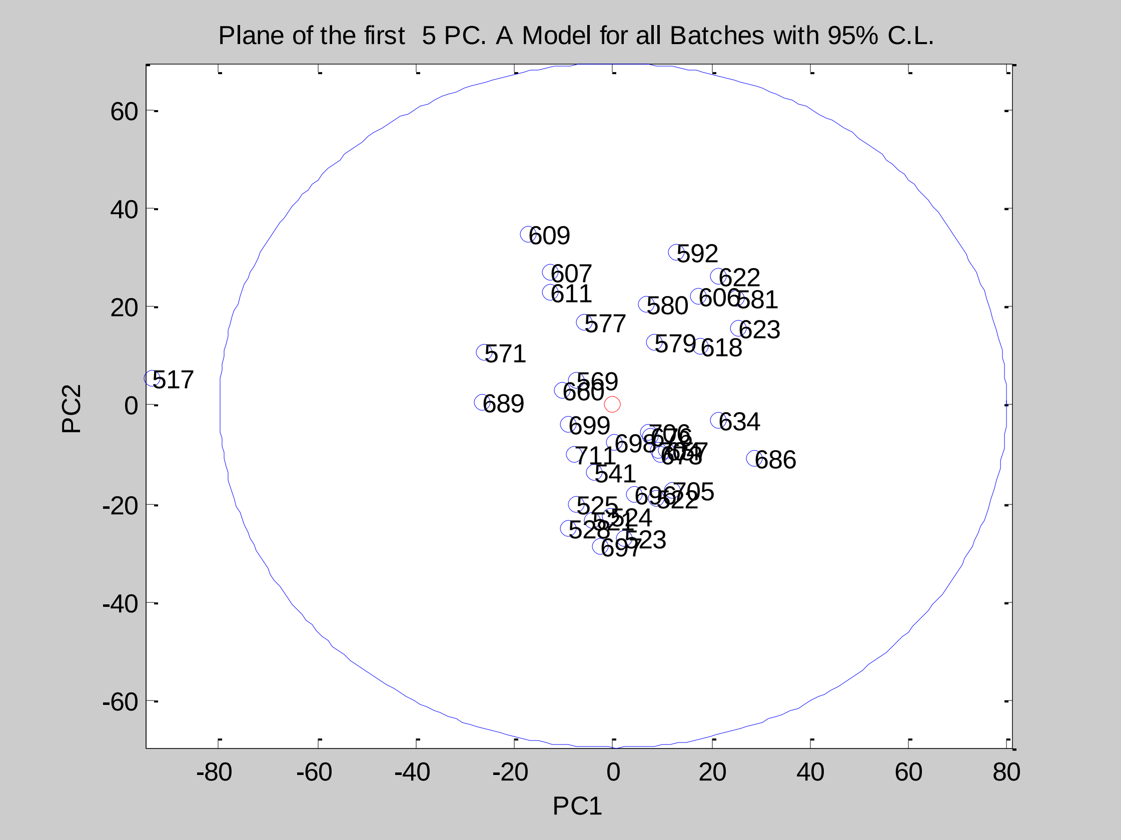


Figure 4-8: Plot of the first two latent variables including 95% confidence

Because the model is best explained by 5 latent variables, there is a need to graph the results on a fashion that would better explain the data. Equation (4.11) is best suited to graph the results of models that have three or more latent variables. Therefore, $r = 5$ in the equation. Figure (4.9) is the projection of the batch 517 as a time series graph with the green line showing the 95% limit of the model. The figure indicates that all 10 process values associated with each batch were condensed into one single value, and this value for batch 517 was put against the batches that were in the model. Subsequently, indicating that batch 517 was an abnormal batch from process standpoint.

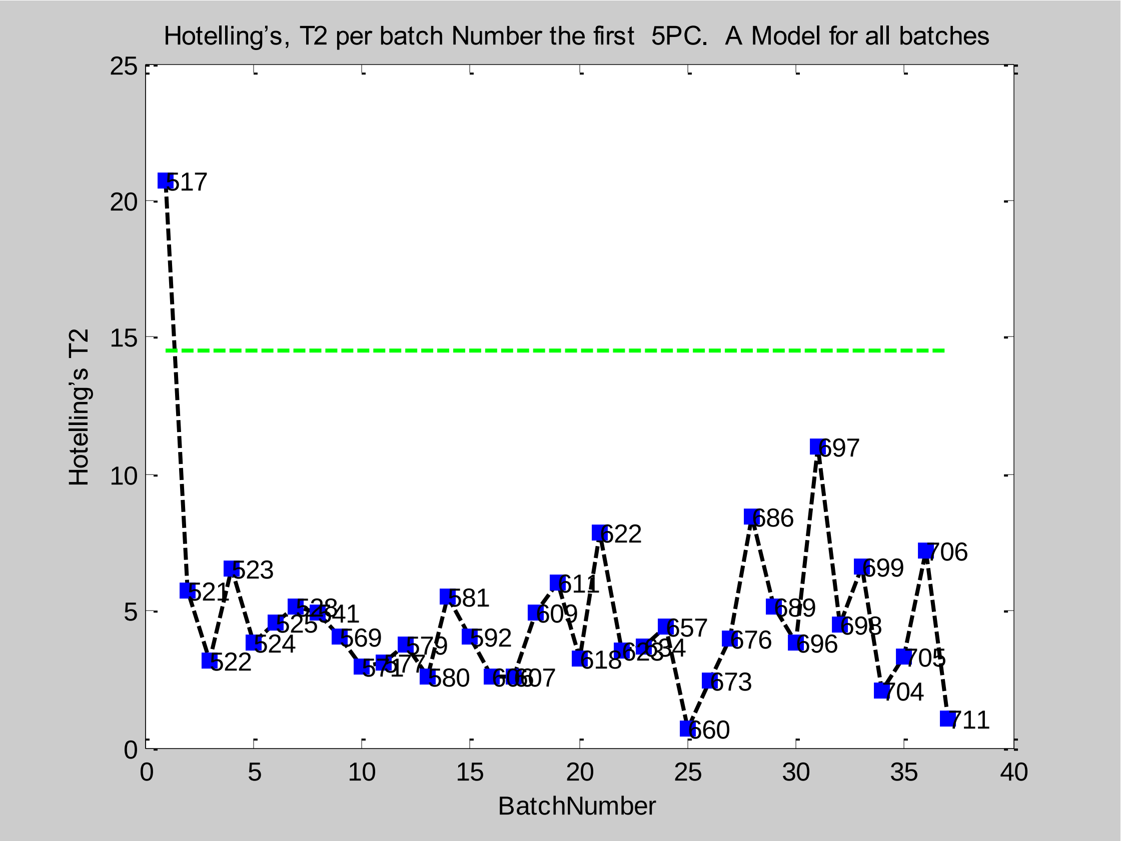


Figure 4-9: Plot of batch 517 projection to the model

To prove model's capability to detect any process related issues whether explained by Hotelling's number or SPE, process data of another batch, 629, were projected to the model (abnormal quality results for this batch is discussed in section 5.2). Figure (4.10) represents the projection of batch 629 in the model of the process data. It can be noticed that the model again capture batch 629 as the one that had abnormal process condition. Comparing figures (4.9) and (4.10), batch 629 exhibits higher process variability than batch 517.

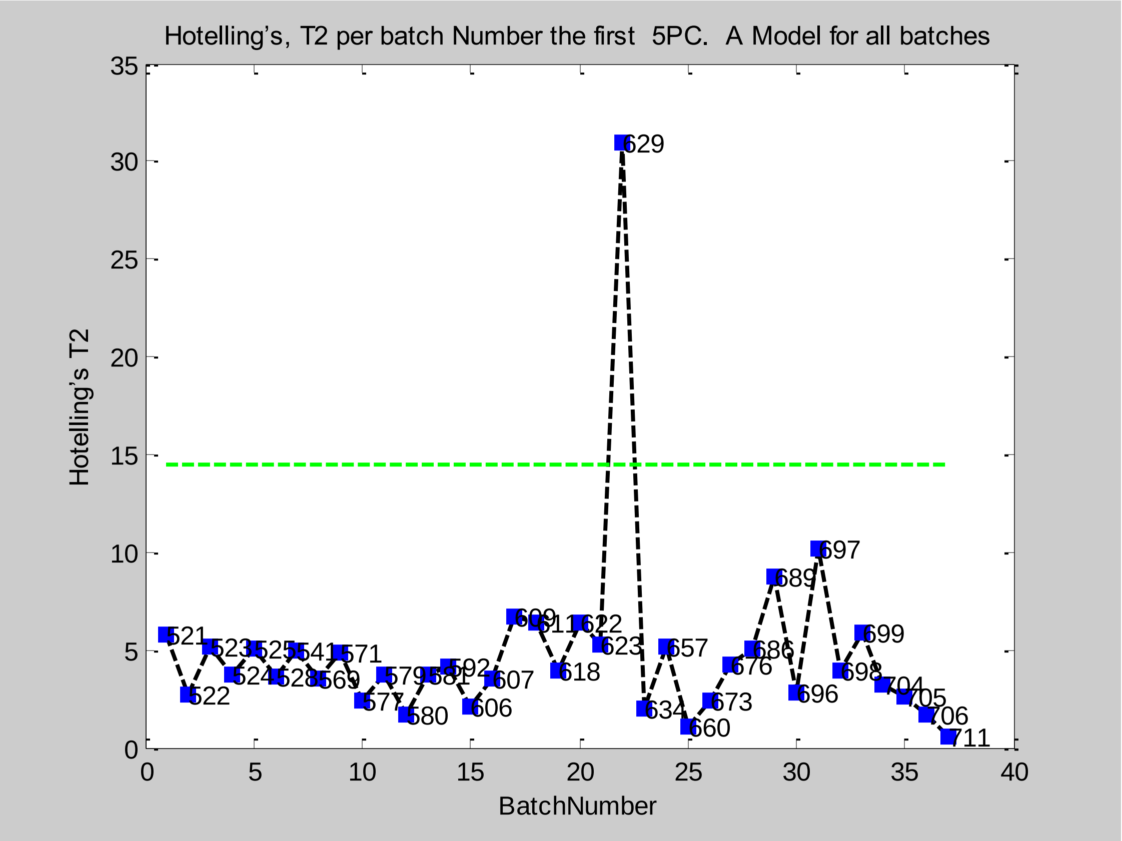


Figure 4-10: Plot of batch 629 projection to the model

Now that the multivariate model has shown the ability to recognize the abnormal batch, the focus will be shifted on localizing the fault and establishing a propagation path for occurrence of the fault. To progress through this path, the faulty batch should be extracted from the matrix of data $X(I + 1, JK)$ which in this case is $X(37, 2280)$. Once the abnormal batch is extracted from the dataset, it has to be reorganized unfolded to the original format. Figure (4.11) illustrates the folding technique.

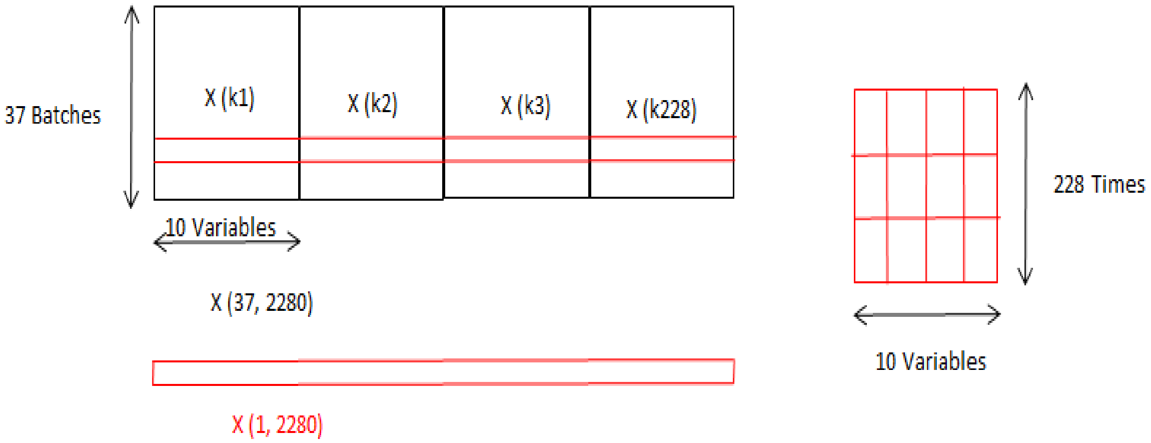


Figure 4-11: extracting and folding the abnormal batch

(Hong, Zhang and Morris 2011) proposed a technique to localize fault in simulated semi batch fed penicillin process, PenSim software, using progressive PCA modeling. In this method, upon observation of abnormal behaviour, process variables connected to abnormality are flagged through their contribution plots. Time series plot of the SPE is used to identify the time information that abnormalities occurred. Then the information about the time of abnormality was used to localize the faults. This information is then used to detect and trace the origin of the fault.

To determine if SPEor prediction values (TP') are smaller than normal, control limits of each value at specific sampling times are needed. These values have a quadratic form of the prediction errors which could be calculated using multi-normal distribution (Hong, Zhang and Morris 2011). According to (Box 1954), chi-squared distribution can be used to estimate the sum of squared variables with multi-normal distribution. According to (Nomikos and MacGregor 1995) control limit, $lim_{\propto}$,,with $\propto$ significance level is calculated by equation (4.14).

$$lim_{\propto} = g\,\chi^2_{h,\alpha} \qquad (4.14)$$

In this equation, χ is chi-squared distribution, h is the degree of freedom and g is the weight. h, and g will be calculated by variance, v, and mean, m, of the distribution according to equation (4.15). Thus variance and mean of the chi-squared distribution are equal to variance and mean of the SPE or prediction values at individual time point so the result of $lim_{\propto}$, is shown by equation (4.16) . Therefore, equations (4.14) through (4.16) provide calculation of 95% control limits for a time series type plot.

$$g = \left(\frac{v}{2m}\right) \ \& \ h = 2\left(\frac{m^2}{v}\right) \qquad (4.15)$$

$$lim_{\propto} = g\,\chi^2_{h,\alpha} \ = (\frac{v}{2m})\,\chi^2_{(2\frac{m^2}{v},\alpha)} \quad (4.16)$$

Batches 517 and 629 were unfolded and the results were plotted, figures (4.12) through (4.15) for both variable and time contributions according to figure (4.11).

Figures (4.12) and (4.13) show the overall variation within all variables of batches 517 and 629 during the batch progression (overall time contribution plot) with dotted green line being the 95% control limit that arises from the history of all batches and being calculated by equation (4.16). It shows that for most of the reaction time, batch 517 had a significantly higher trajectories than that of normal batches within the model. however, batch 629 had higher trajectories than that of normal batches only in the first half of reaction time. Figures (4.14) and (4.15) show the variation within the variables (overall variable contribution plot) of batches 517 and 629 throughout the batch time. Now, there is enough information to which variables are contributing factors to the abnormal conditions. However, there is still a need to figure which variable has contributed to the a specific time abnormality.

To accomplish this two arbitrary time values of abnormal conditions from figures (4.12) and (4.13), 110 minutes from aligned time on batch 517 and 75 minutes from aligned time on batch 629, are chosen. Then variable contribution at these time points are projected to the overall variable contribution of all batches within the model. Batches 517 and 629 are individually put against, figure (4.16), as the model to establish the control limits according to equation (4.16) for variable contributions. The results are then plotted in figures (4.17) and (4.18) with the dotted line being the 95% of the limit for a specific variable contribution to the abnormality condition for times 110 and 75 respectively. The figures show that in batch 517, process variables 1 through 4 have significantly higher deviation from the model; and for batch 629 these deviation were observed for process variables 1 through 4 and Process Variable 7.

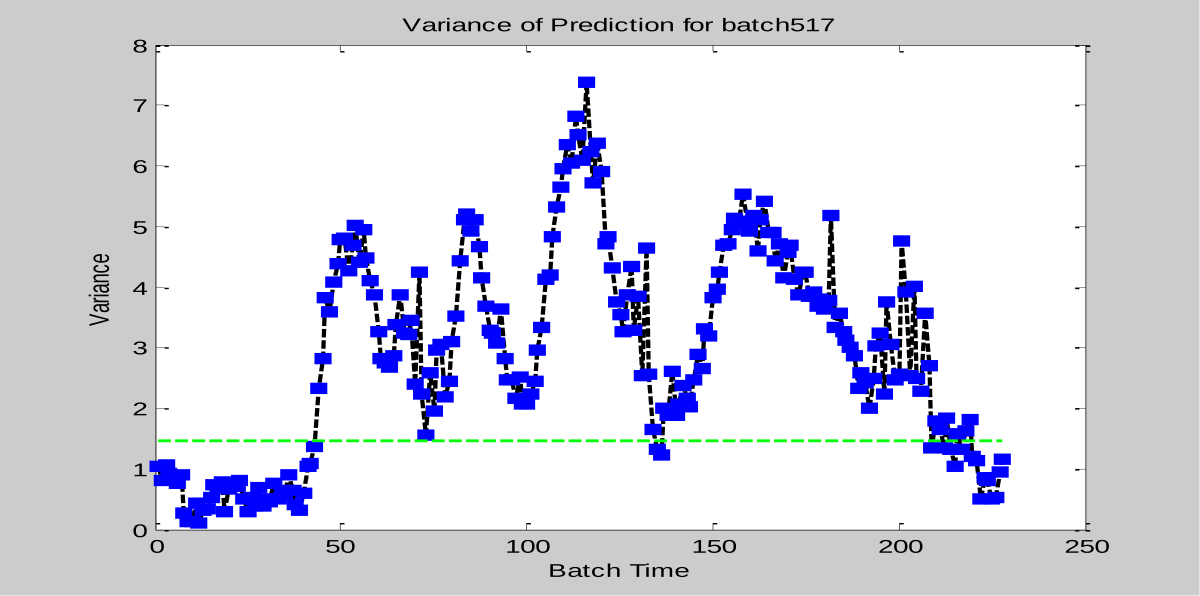


Figure 4-12: Overall Time contribution plot for batch 517

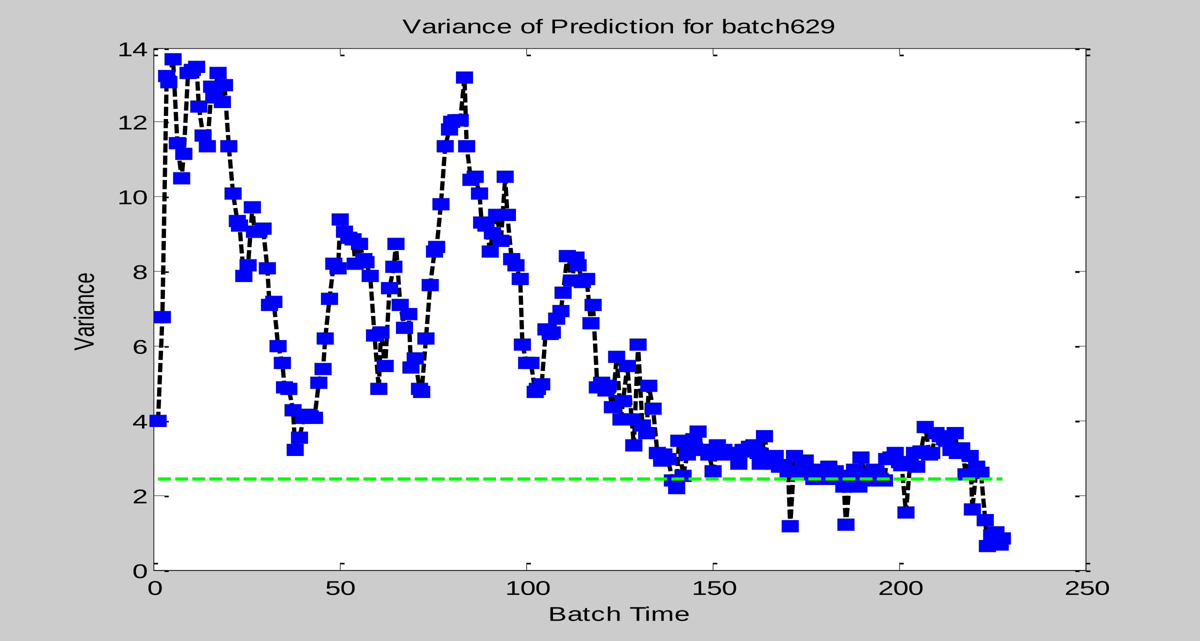


Figure 4-13: Overall Time contribution plot for batch 629

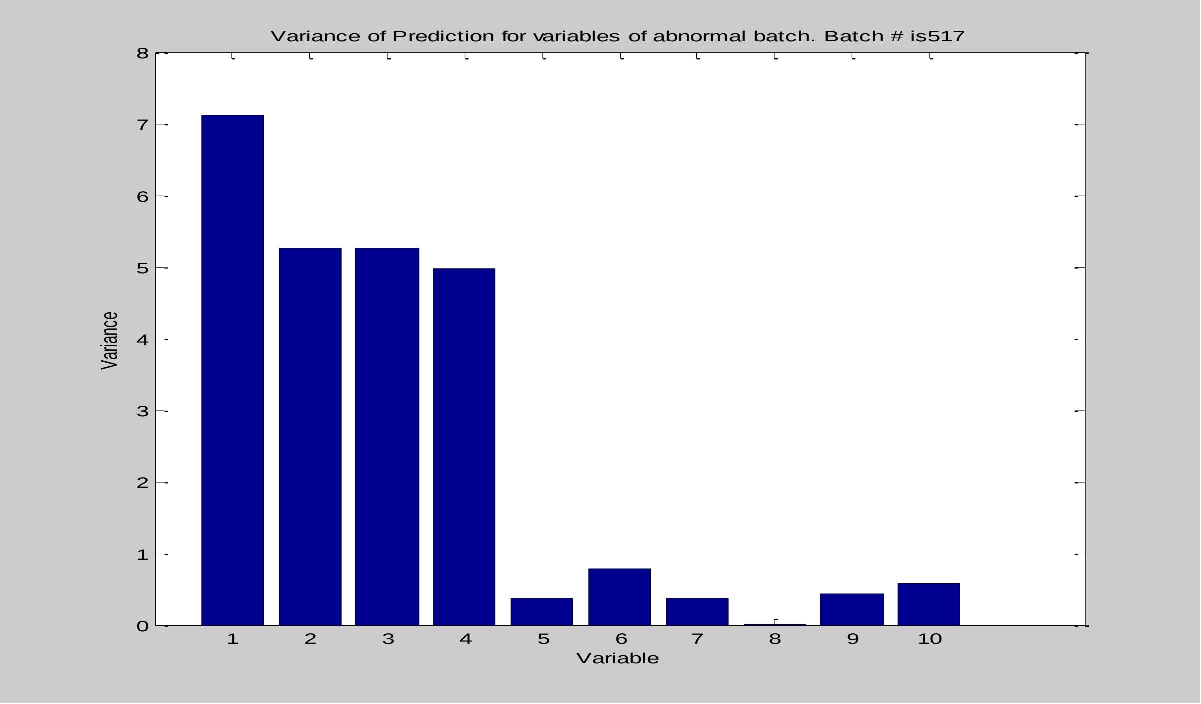


Figure 4-14: Variable contribution plot for batch 517

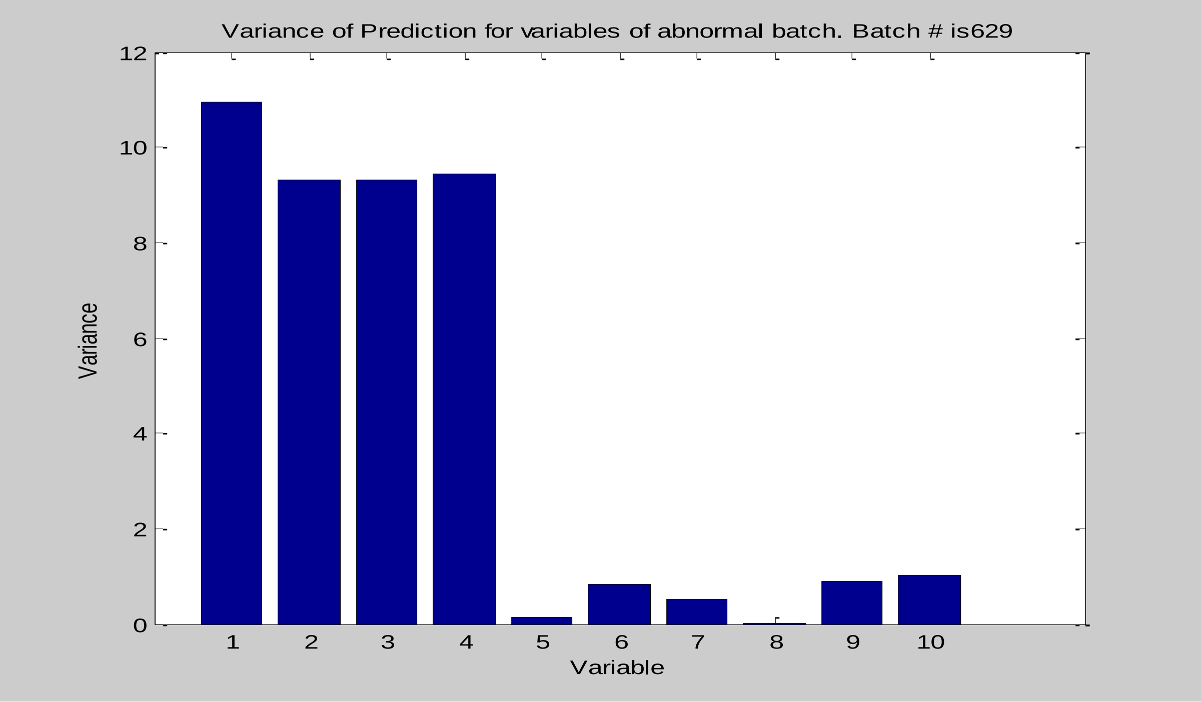


Figure 4-15: Variable contribution plot for batch 629

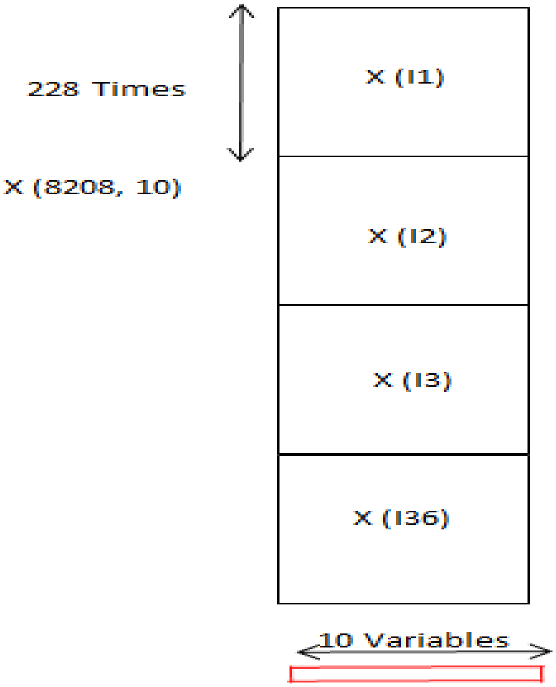


Figure 4-16: Introducing batches 517 and 629 individually to the model.

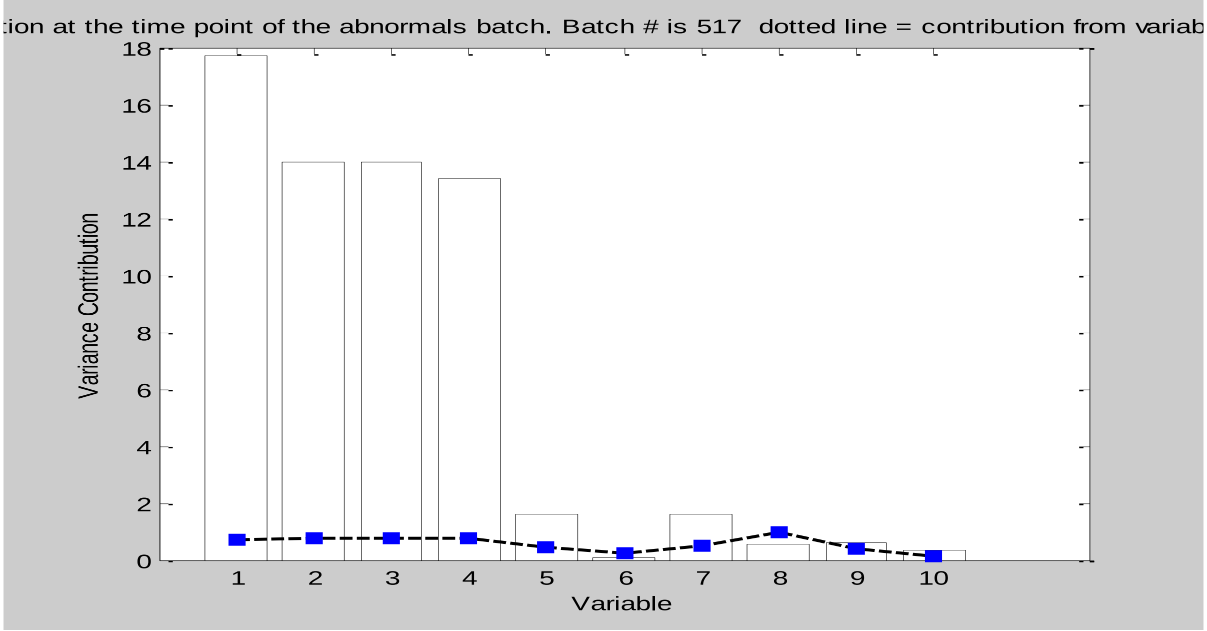


Figure 4-17: Variable contribution plot for batch 517 against the control limit of the model

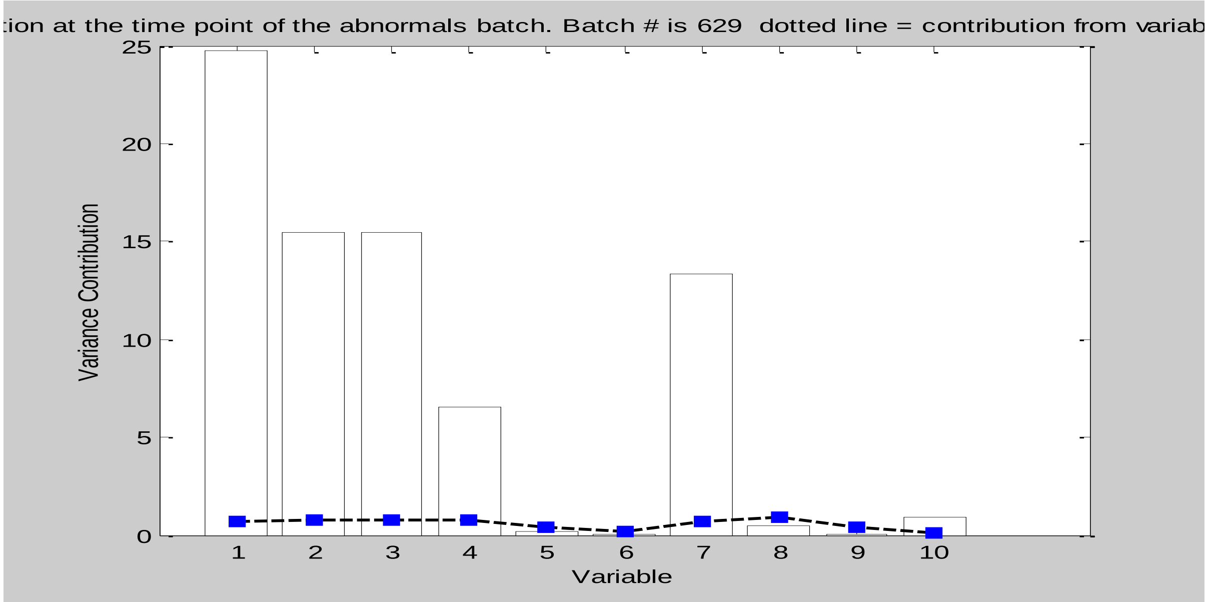


Figure 4-18: Variable contribution plot for batch 629 against the control limit of the model

Figures (4.19) and (4.20) display process variable1 trends in a non aligned mode (data are zoomed in for better visual representation) whereas figures (4.21) and (4.22) show the process variable 1 trends in aligned mode. The blue line states the times of abnormality which are 267 and 247 time unit in the non aligned mode for batches 517 and 629 respectively. The non aligned time unit are equivalent to 110 and 75 minutes in the aligned mode for batches 517 and 629 respectively. One might conclude from figures (4.19) and (4.20) that batches 517 and 629 had similar variable trajectories as the other batches in the model; however, figures (4.21) and (4.22) show the differences of process progression over time between batches 517 and 629 and the rest of the batches within the model.

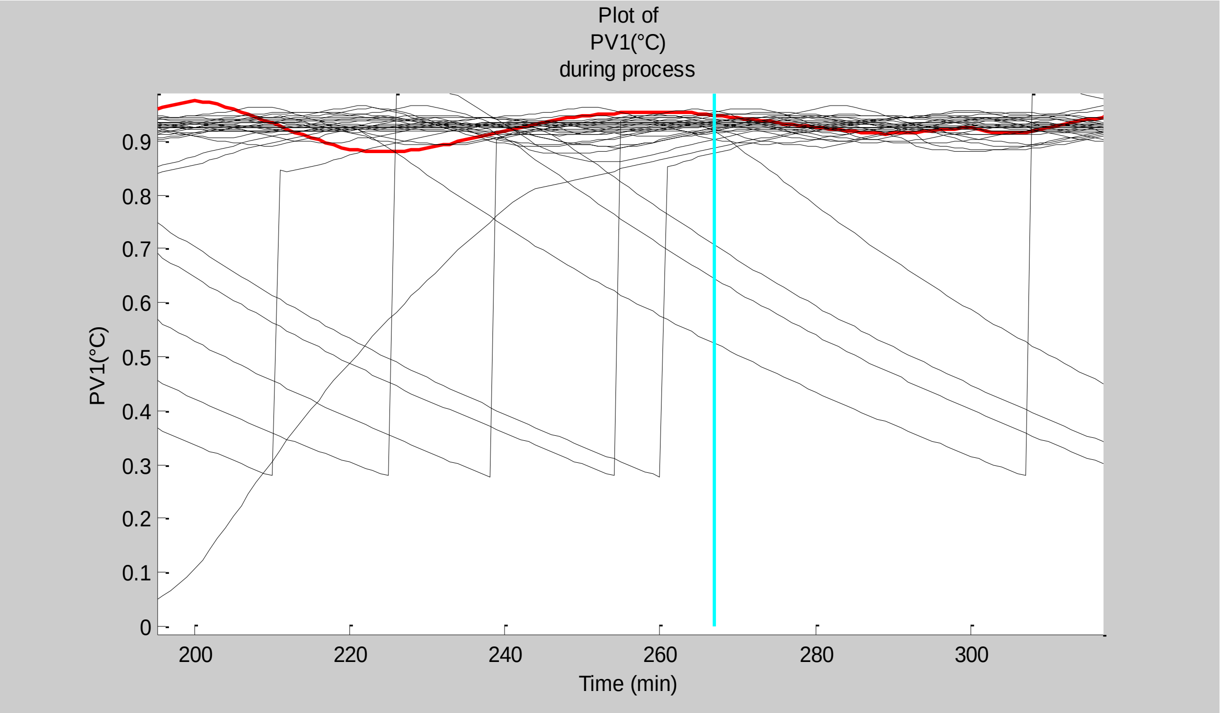


Figure 4-19: Plot of batch 517 process variable 1 for non aligned data

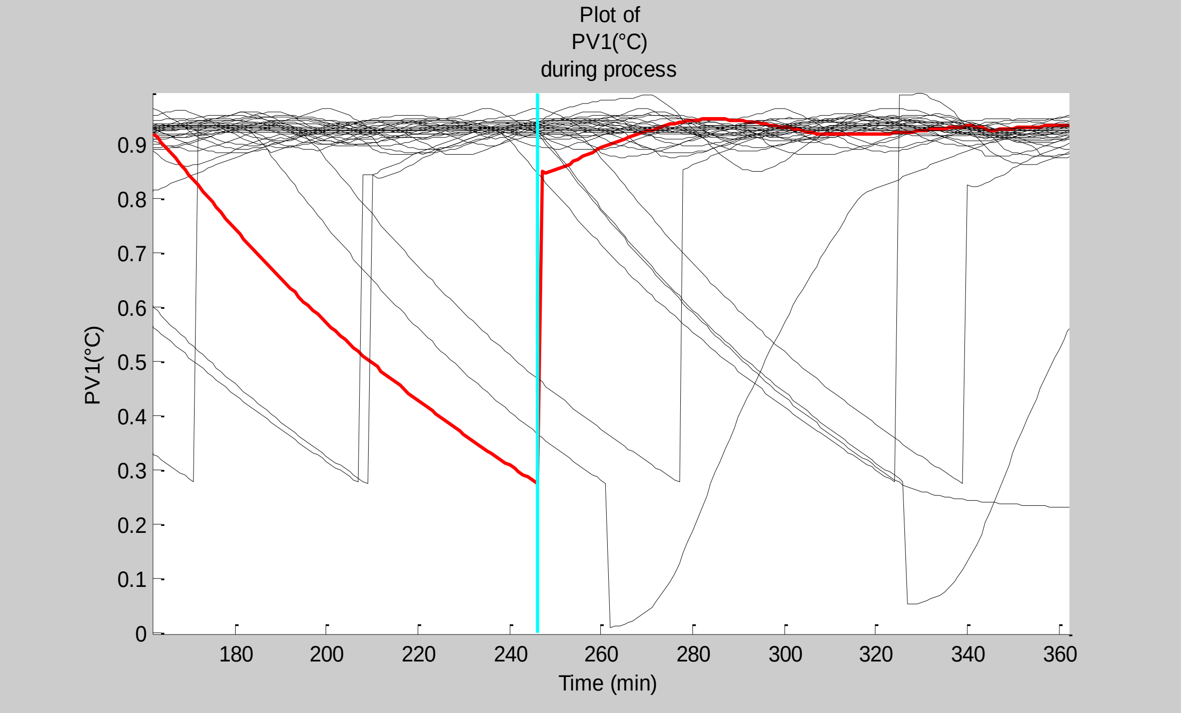


Figure 4-20: Plot of batch 629 process variable 1 for non aligned data

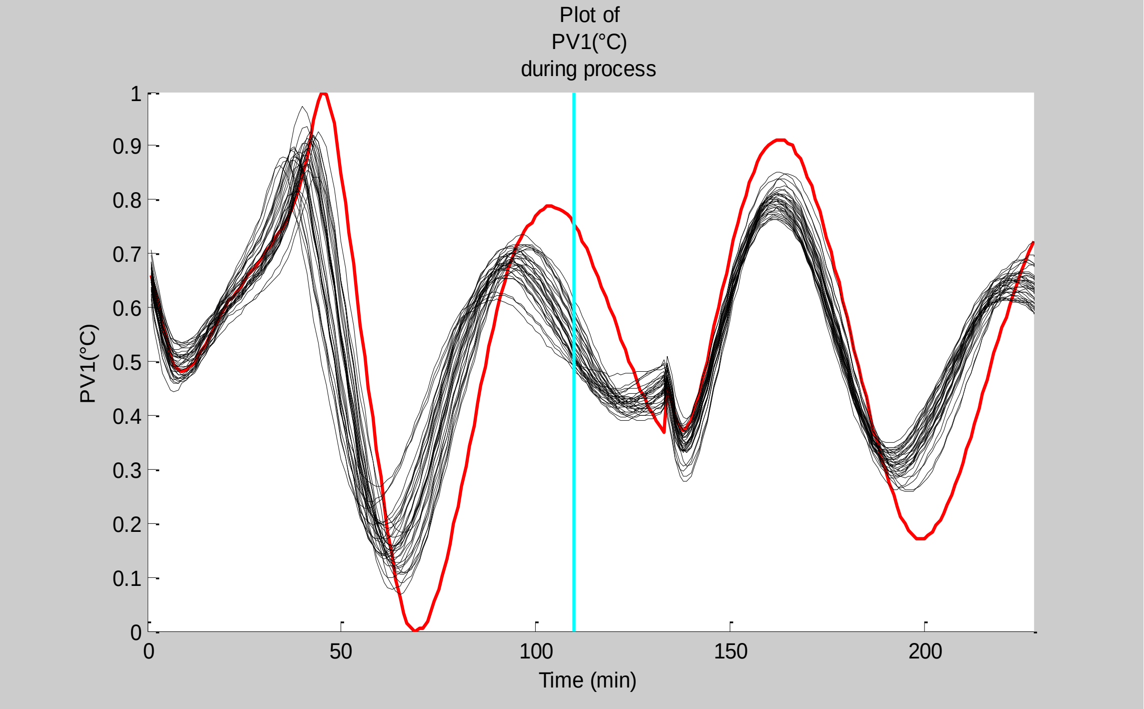


Figure 4-21: Plot of batch 517 process variable 1 for aligned data

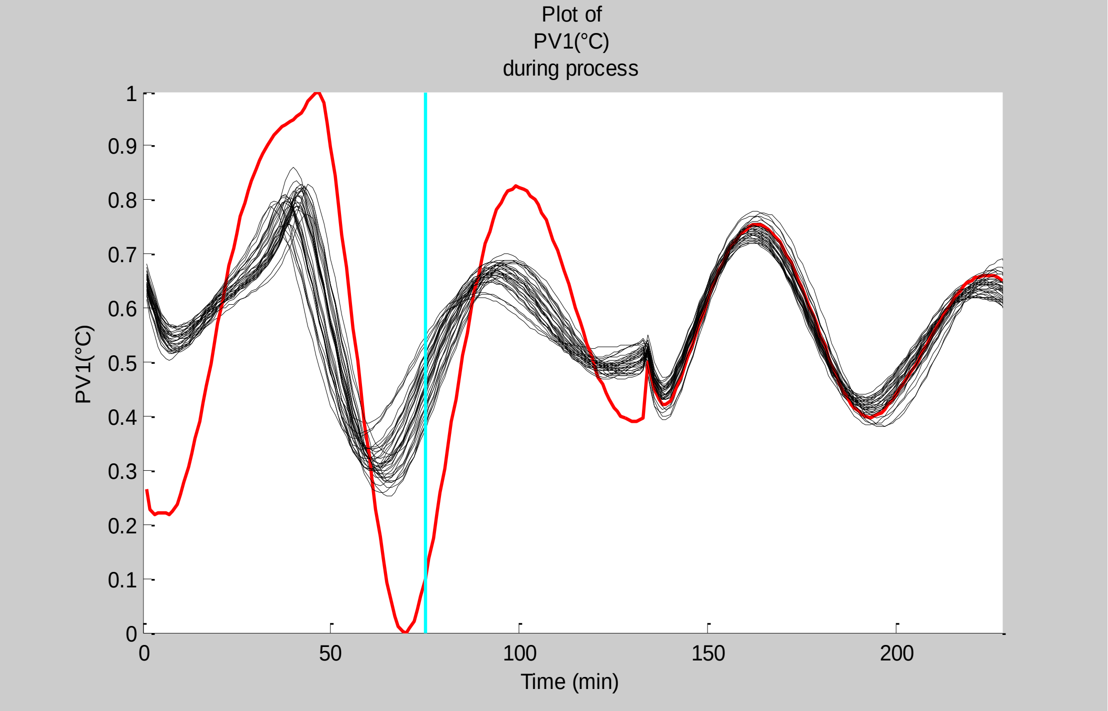


Figure 4-22: Plot of batch 629 process variable 1 for aligned data

A wrong conclusion is that the difference in the process variable1 from batch 517 arises by non alignment of batch 517 with the rest of the model. In other words, if this batch was aligned properly, it would present the same pattern of progression for process variable 1 trajectory. To address this argument, two process variables that did not have higher variable contribution than the model were chosen. Process Variable 8 and Process Variable 9 for batch 517 and 629 are plotted against the model in figure (4.23) and the patterns show perfect alignment.

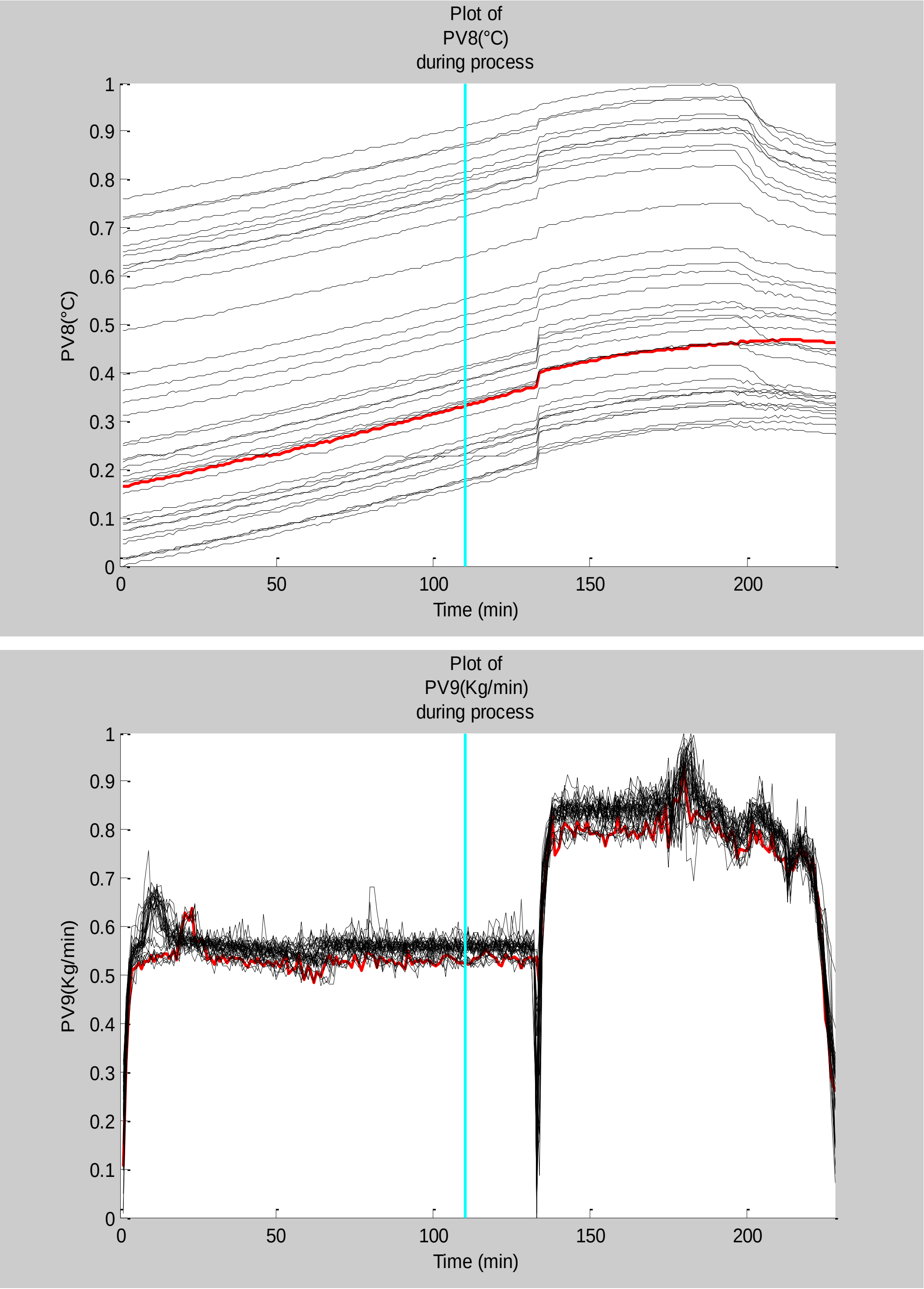


Figure 4-23: Plots of Process Variable 8 and Process Variable 9

A wrong conclusion is that only the alignment of batch data would be enough to understand the variation of an abnormal batch from the historical trajectories of normal batches. Although, this assumption might be considered as enough reasoning to prevent latent variable modeling on batch data set, it should be noted that analysis of Chemical Toner historical batch data set through only aligned batches may not always lead into the variable contributions to abnormality. The abnormal batches in the examples covered so far (product type 4), happen to have significant variation only on certain process variables.

To provide an example where the fault detection shows the ability to capture the variability in the operation of the abnormal batch, a different product (product type2) batch data were modeled. Once the model was built and validated, MPCA analysis was performed on the model. The latent variable model showed, figure (4.24), that the batch 379 of product 2 has a higher standard prediction error to the rest of the model. Once the abnormal batch was flagged, the time series plot of batch 379 was plotted, figure (4.25). A time of abnormality, 17 minutes through the batch, was chosen and variable contribution plot of batch 379 at 17 minutes in the batch was plotted, figure (4.26). According to figure (4.26) Process Variable 10, Process Variable 1, Process Variable 3, and Process Variable 4 had higher trajectories than the model of the normal batches. Looking at the aligned batch data for these process variables, one can easily identify the significant variations, at 17 minutes, on the Process Variable 10, figure (4.27). However, none of other 4 process variables can be identified through the aligned data despite the fact that they show significant contribution to abnormality according to figure (4.26) . Figures (4.28) and (4.29) are the aligned data for Process Variable 2 and Process Variable 3. The blue line shows the process value at 17 minutes through the batch and the red line is batch 379. Deviations of process variables 2 and 3 can not be identified through aligned data. However, these deviations are identifiable through latent variable modeling results in figure (4.26). This is a proof on a statement was made earlier in this section that relying only on data alignment would not reveal all abnormalities in process variable trajectories during progression of a batch.

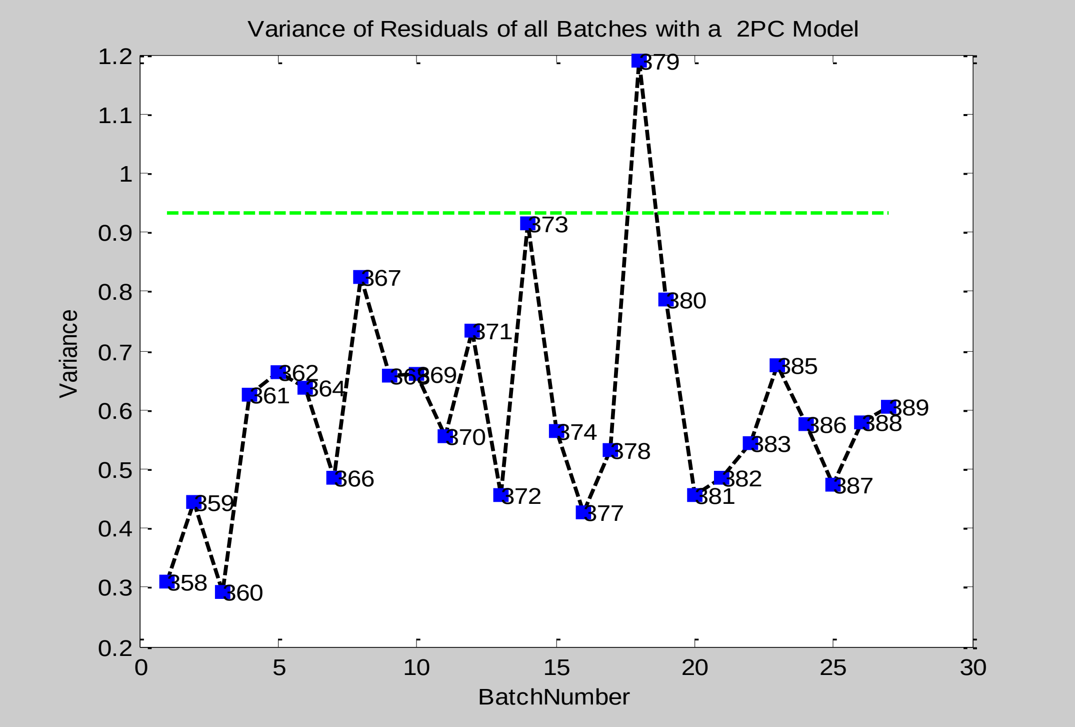


Figure 4-24: Plot of batch 379 projection to the SPE model

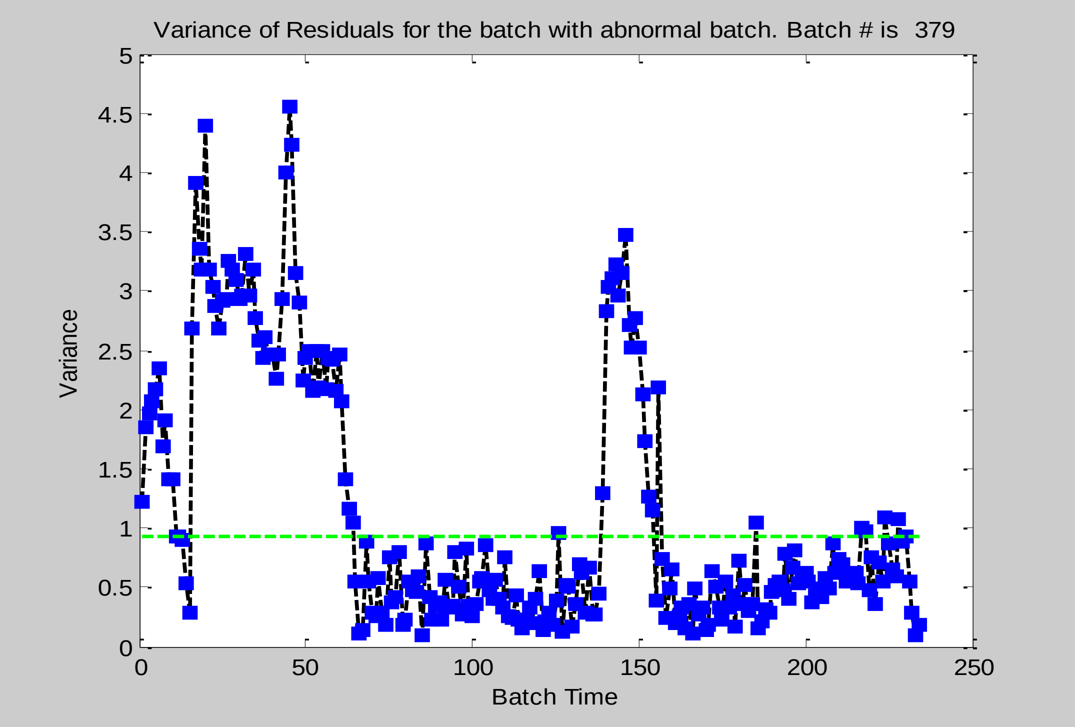


Figure 4-25: Time contribution plot for batch 379

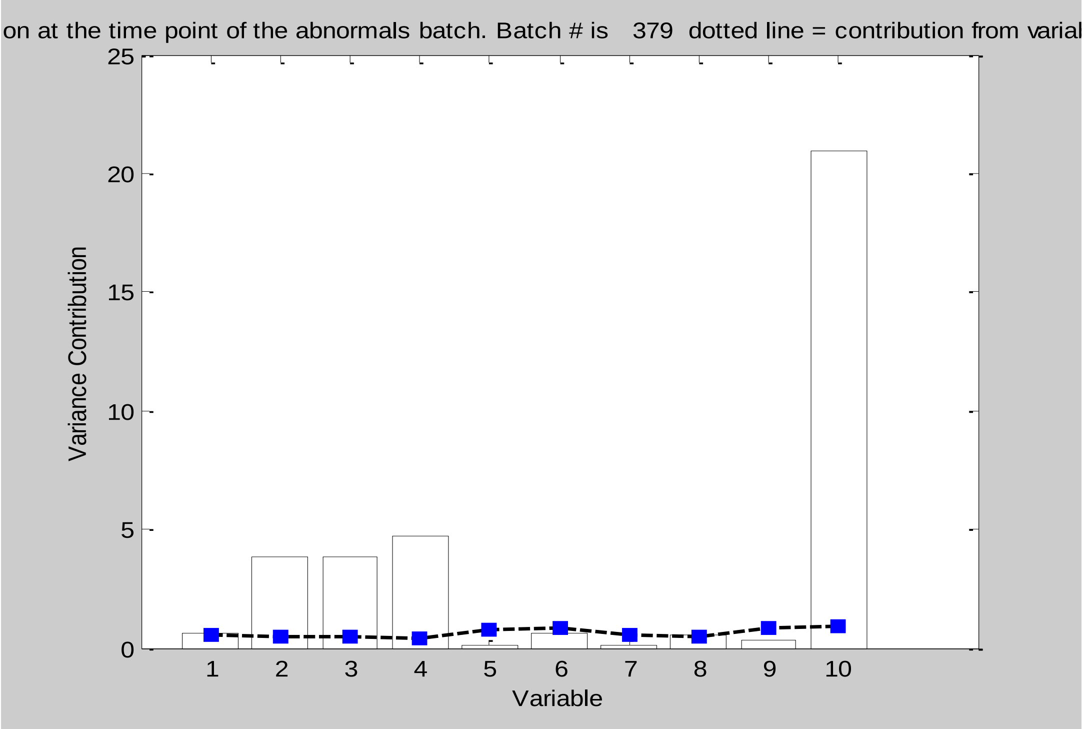


Figure 4-26: Variable contribution plot for batch 379 against the control limit of the model

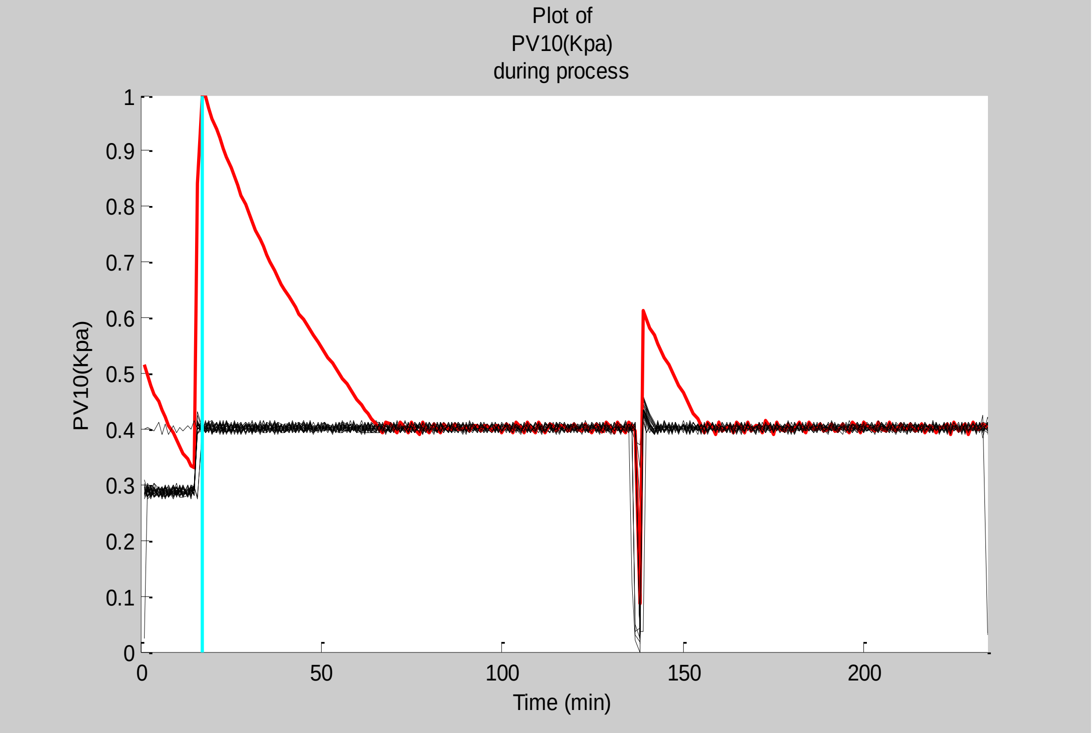


Figure 4-27: Plot of batch 379 Process Variable 10 for aligned data

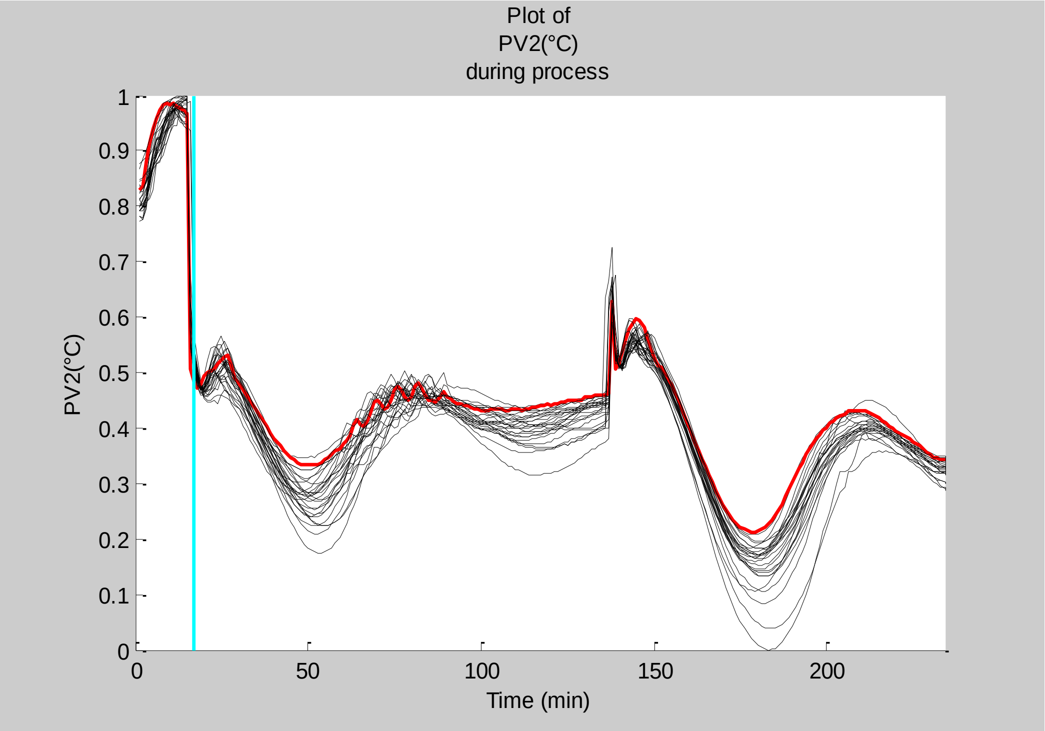


Figure 4-28: Plot of batch 379 process variable2 for aligned data

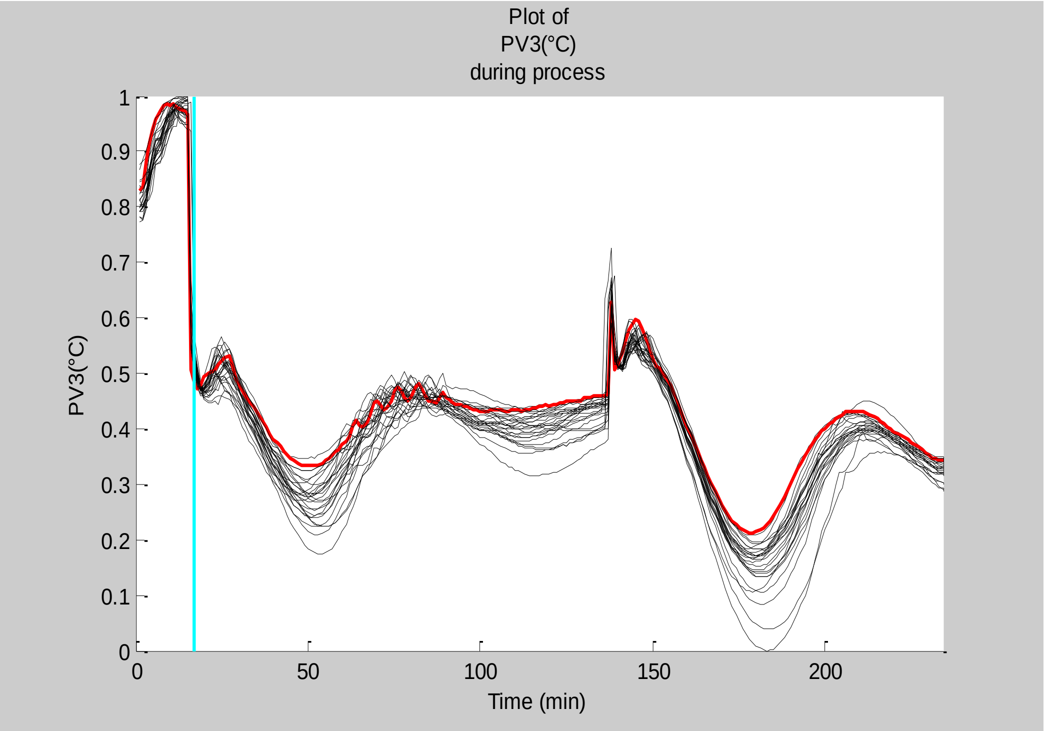


Figure 4-29: Plot of batch 379 Process Variable3 for aligned data

4.5 Summary

Latent variable modeling was used to analyze process values of different bathes of two different product types of Chemical Toner processes to identify root causes for a high level of deviation from target quality measurements. The analysis identified that the operating policies (batch progression) of the batches were the major contributors to the poor quality products. These approaches also isolated the contribution of a specific process variable at a given time during the batch, fault localization. Process variables trajectories such as process variable 2 and process variable 3 profiles were contributing factors to poor product quality. This study illustrates the ability of multivariate modeling to analyze history of batch data and establish the operation policy for online monitoring of Chemical Toner processes.

This study also provided examples that even aligning data of abnormal batch along the normal batches of historical data set were not able to localize the abnormal condition at a desired time point during the batch. The later part of the study highlighted the power of multivariate modeling in detecting and localizing the faulty process behaviour.

Chapter 5
Product Quality Modeling by Multi-way PLS

5.1 Introduction

Partial Last Square is a projection method for linear modeling of the relationship between a set of response variables Y and a set of predictor variables X. In 1982, Wold derived the PLS algorithm, which has been used by other researches since its introduction (Geladi and Kowalski 1986), (Nomikos and MacGregor 1995), and (Munoz 2004).

The PLS geometric interpretation is closely related to the geometry of PCA. PLS can be interpreted as performing PCA on the covariance of X and $Y(Y'X)$. Therefore, the decomposition is not affected only by the variance of the X and Y matrices also by the correlation between them. PLS tries to accomplish two tasks simultaneously. It searches to find the directions of maximum variability in the x and y-space, and at the same time it tilts these directions so the direction in the x-space has maximum correlation with the direction of y space.

In this research Multi-way Partial Least Squares (MPLS) is used in the process reaction example to incorporate the product quality data Y that are available once the batch is completed.

5.2 Multivariate Classification on the Quality Matrix

According to (Duchesne and MacGregor 2004) since the quality is a multivariate property, the quality of product has to be analyzed by multivariate techniques. To do this, first multivariate product classification of data is built using NIPALS algorithm for PCA used in chapter 3.

PCA model is built for both data sets of batches in reaction with 2 latent variables. In the data set of 36 batches (product type 4), the first and second latent variable captured 53.83% and 13.92% variability within data respectively. Thus enabling the PCA model to explain 67.75% of the variability within the dataset. The plane of the latent variables (Figure 5.1) shows two clustering of data and the ellipse shows the 95% confidence region in the model containing 36 observations.

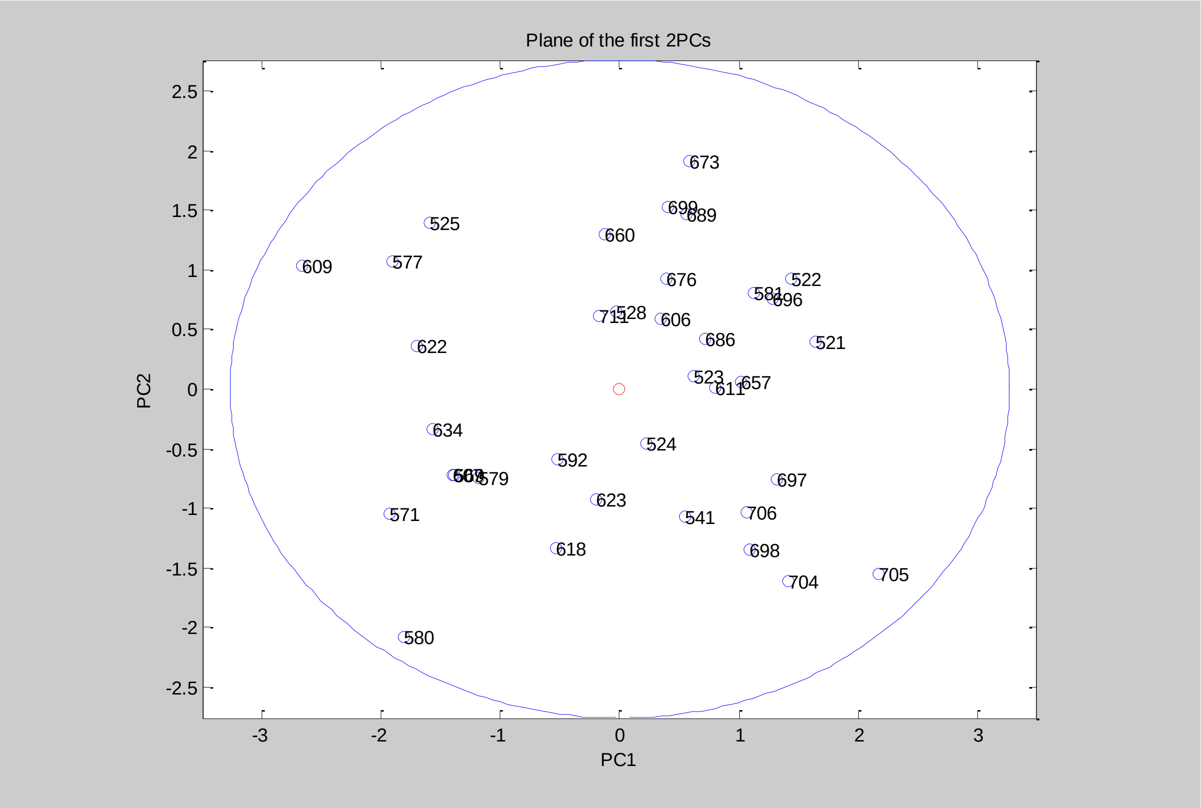


Figure 5-1: Latent Variable overall quality variables.

Once the model is built, the abnormal batches are introduced to the model. According to table (5.1) batch 517 and 629 have abnormal quality.

Table 5-1: Quality measurements of batches 517 & 629

	Model Lower Limit	Model Upper Limit	Target	517 Result	629 Result
Variable 1	1.81	2.59	2	2.45	2.27
Variable 2	10.5	35.4	30	40.3	46.7
Variable 3	2.85	3.64	3	3.5	3.31
Variable 4	1.7	4	3.5	2	2.2

These batches constitute an off spec results for variable 2 however, there are some variations on other variables. The results of these batches are projected to the latent variable space of the data set to have an understanding of their overall quality.

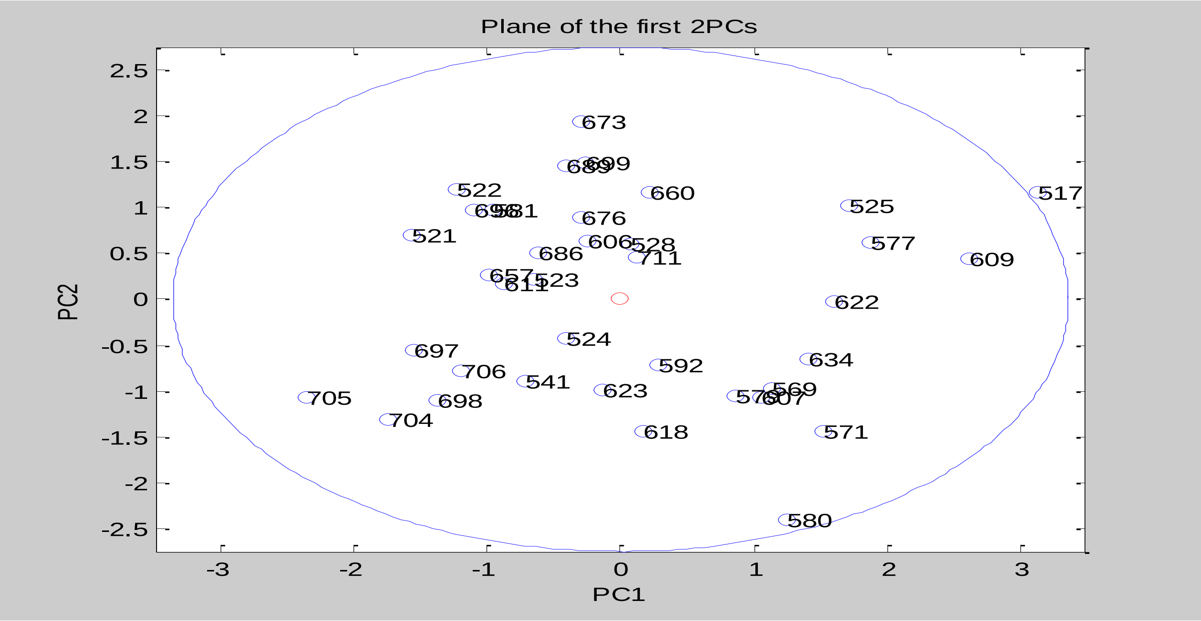


Figure 5-2: Hoteling's overall quality variables with abnormal batch

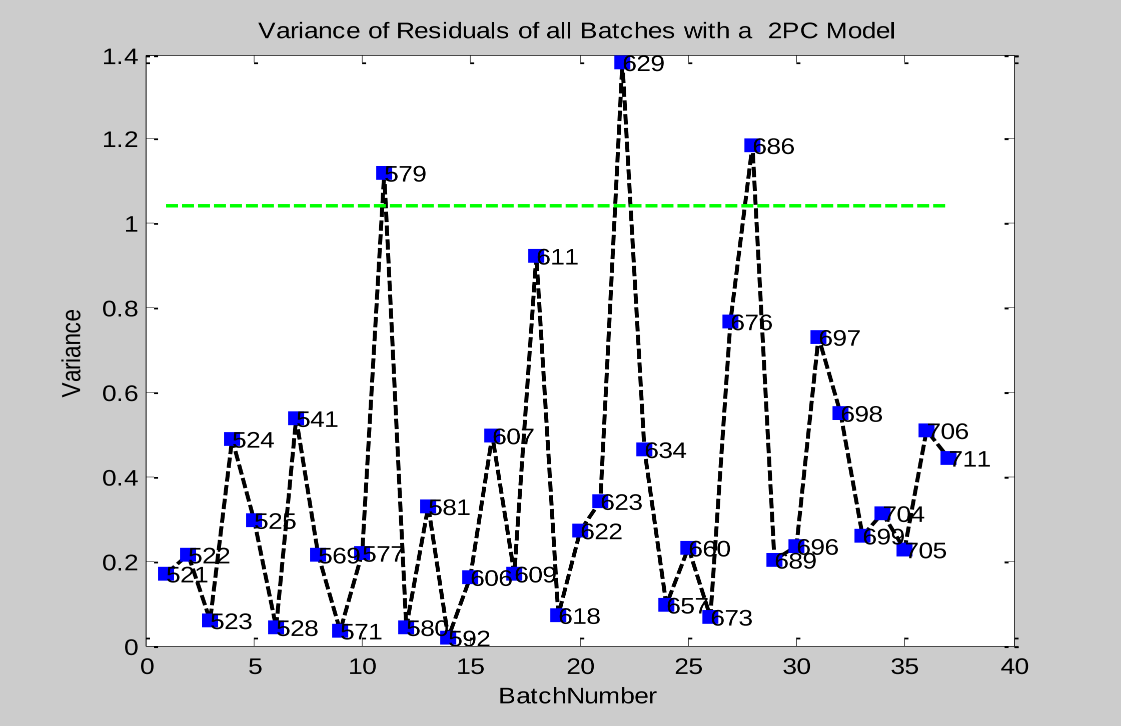


Figure 5-3: SPE's overall quality variables with abnormal batch

According to figure (5.2) batch 517 can be detected as having quality issue from its hoteling's values but batch 629's abnormality is detected through its projection to SPE in figure (5.3).

The plot of the latent variables will be used to define what variables and into what degree are the contributing factors to an abnormal batch. The contribution plot (Figure 5.4) also explains what are the contributing factors to the out of spec batch.

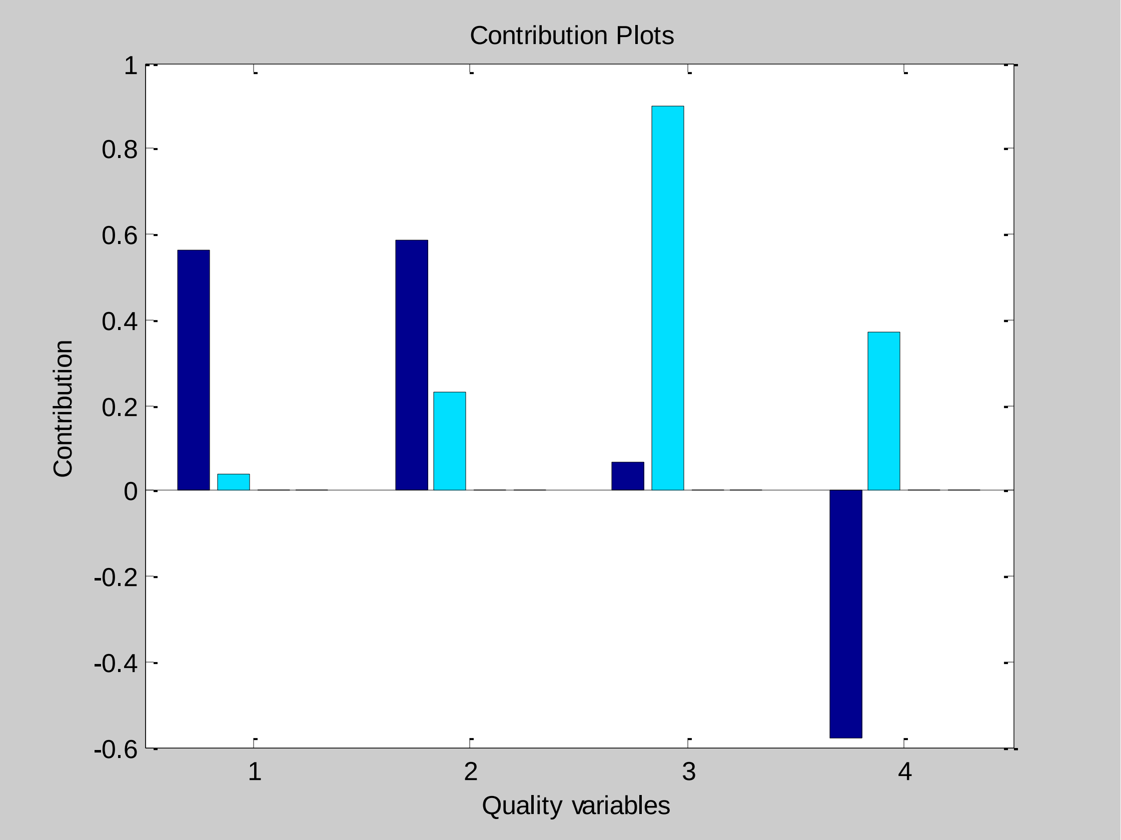


Figure 5-4: Latent Variable overall contribution of quality variables

In the above figure, the first and second latent variables are represented with dark blue and cyan bars respectively. Figure (5.4) shows that in the overall data set of batches, variables 3 and 4 have the highest variation within the model but most of the variations within the first latent variable is explained by variables 1 and 2. Figure (5.5) is the time series representation of figure (5.2) that projects overall quality variables of batches within the model. Figure (5.5) shows that batch 517 has failed the overall quality target. However, figure (5.2) indicates that the contributing factors to the failure of batch 517 were high variations within the first latent variable (marked as PC1 on the graph). Then reviewing figure (5.5) indicates that the contributing factors to the first latent variable were quality variable 1 and quality variable 2.

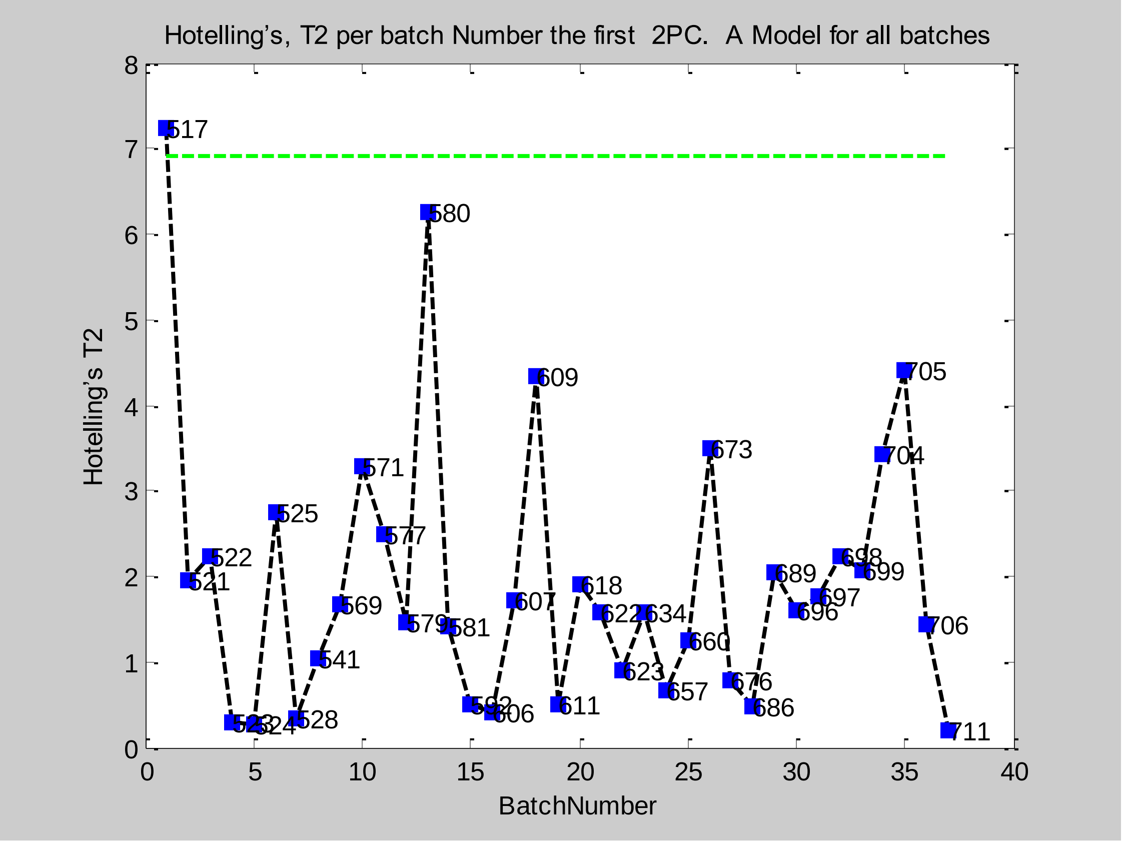


Figure 5-5: Hostelling's of quality variables in time series fashion

5.3 MPLS Analysis Using NIPALS

The linear and compressed combination of X and Y are latent variables T and U respectively. The relation between the latent variables in both x and y-space enables the PLS model to provide proper predictions. The number of variables included in the PLS model is the essential criteria for the predictability of a regression model (Nomikos and MacGregor 1995). PLS uses the latent variables to address the regression problem, thus it gives the minimum number of variables that is necessary. There are several algorithms proposed to extract the PLS components. In this thesis NIPALs algorithm is used to keep consistency with the algorithm used in chapter 3. The following steps explains the NIPALS algorithm for sequential computing of the dominant latent variables:

1- Unfold $X(I,J,K)$ into $X(I,JK)$
2- Centre and Scale X and Y
3- Select an arbitrary column of Y as u
4- $w = X'u$
5- Normalize the loadings $w = w/|w|$. This is done to rescale p to magnitude of 1

6- $t = Xw$

7- $q = Y't/(t't)$

8- $u = Yq/(q'q)$

9- If u has converged, go to next steep, otherwise go to step 4

10- Calculate the matrix of residuals. $p = X't/(t't)$, $E = X - tp'$, and $F = Y - tq'$

11- $X = E$ and $Y = F$

12- Move to step 3 to extract the next latent variable.

Figure (5.6) illustrates the NIPALS algorithm in a flowchart format.

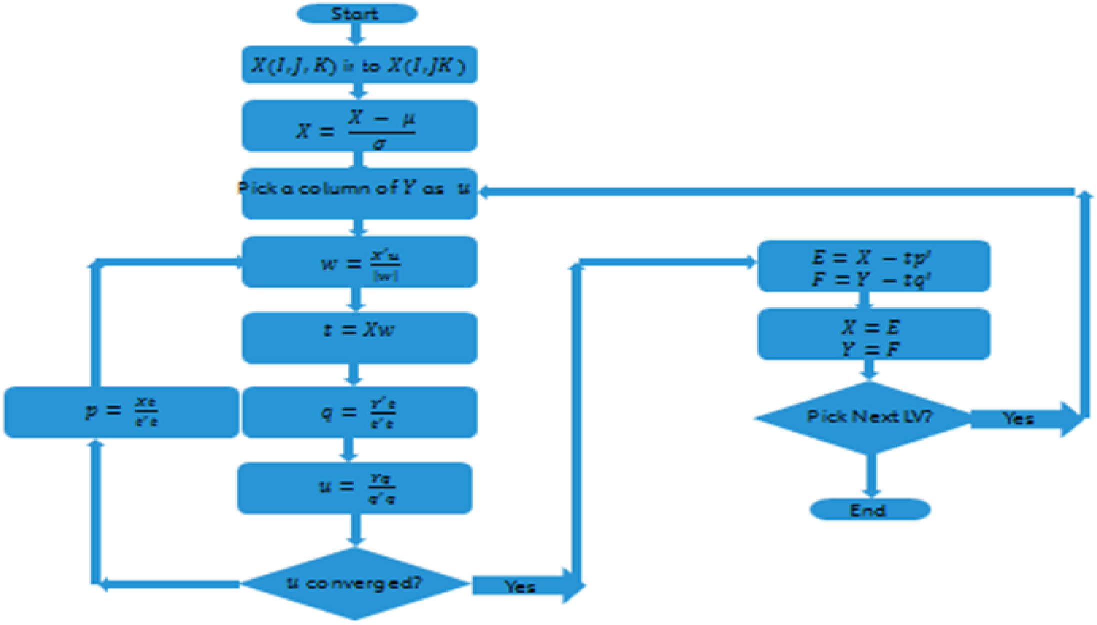


Figure 5-6: NIPALS flowchart for PLS

The NIPALs algorithm for PLS only selects one pair of latent variables (t, u) at a time and use the residual matrixes for the calculation of the remaining pairs. This algorithm applied to the dataset of the 36 batches of reaction and the first two sets of latent variables were plotted against each other . When the latent variables are highly correlated, the observations in the latent variables space is approximately near to the diagonal of the graph. Figures (5.7) and (5.8) show that the variation explained by the first two latent variables in the PLS model in the X space is sufficiently correlated with the corresponding variation in the Y space.

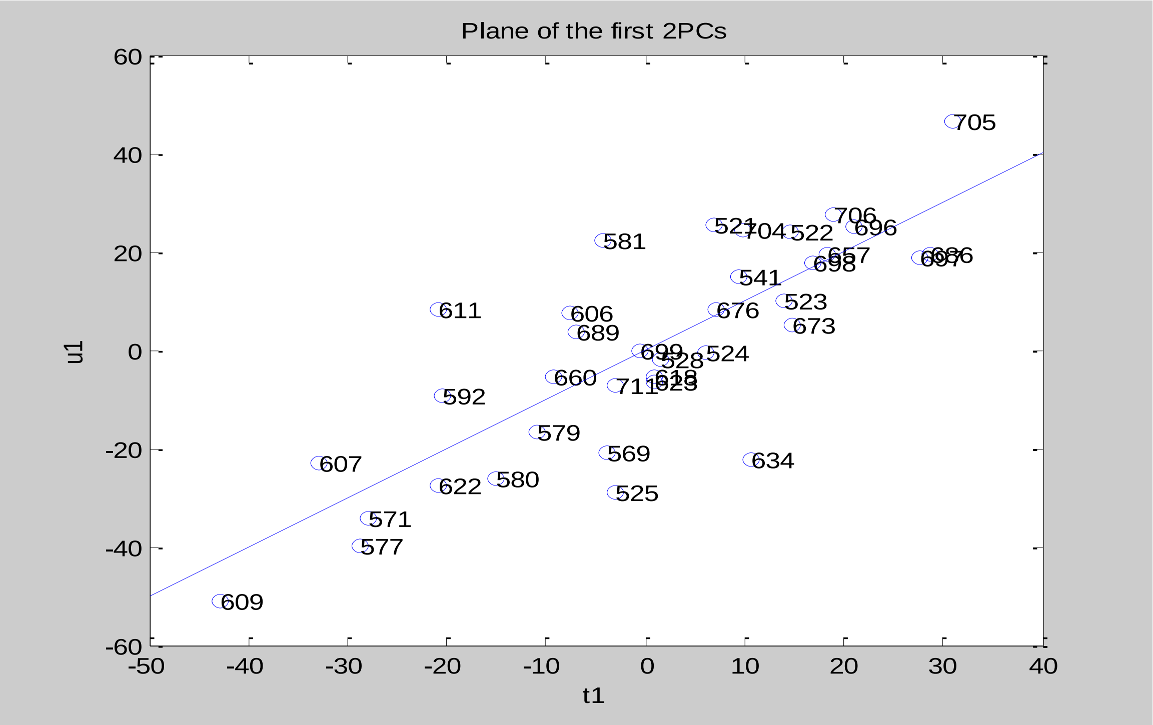


Figure 5-7: plot of the first pair of latent variables

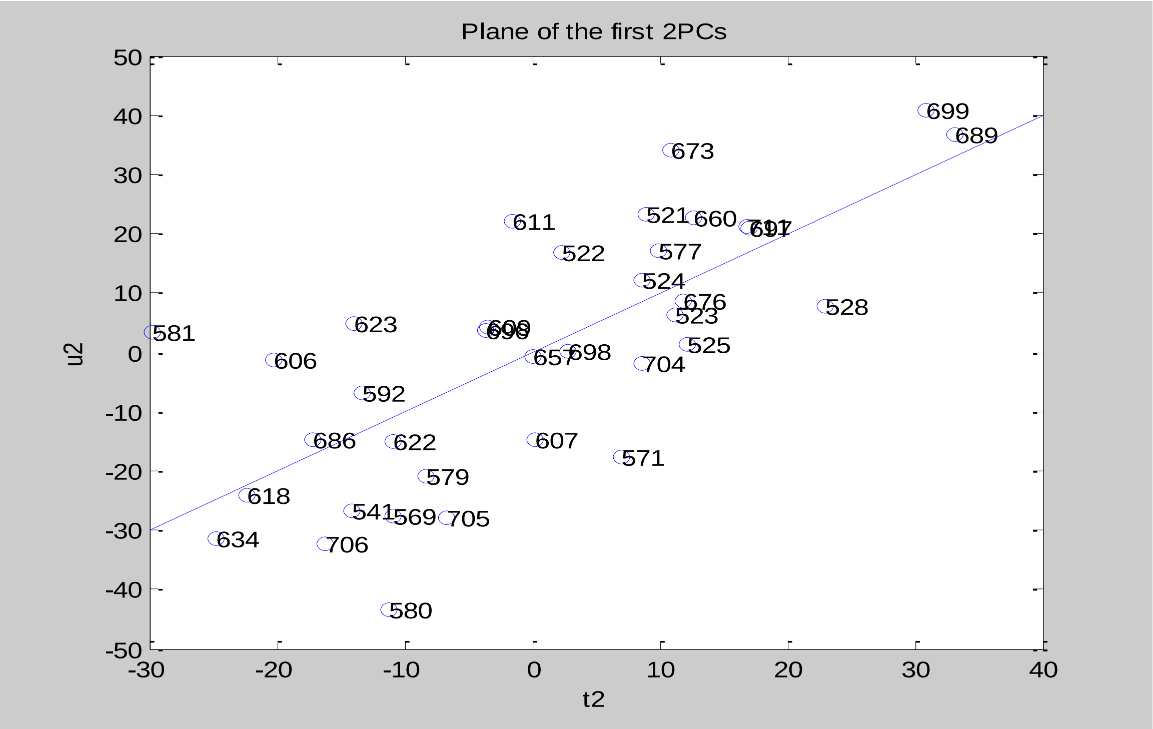


Figure 5-8: Plot of the second pair of the latent variables

Once the multivariate model of data is built, the operational boundaries of the latent variable space can be defined. Equation (5.1) defines the relationship between the latent variable t, its loading q, and residuals of F.

$$Y = tq' + F \quad (5.1)$$

Reviewing the database of the NIPALS algorithm provides a path to define the lower and upper limits of the latent variables t and u. Table (5.2) highlights the operational boundaries defined by the model for latent variables.

Table 5-2: Operational boundaries of latent variables

	T1	T2	U1	U2
Lower Limit	-42.85	-29.71	-51.14	-43.50
Upper Limit	30.89	33.08	46.63	40.81

These boundaries are used to identify off spec batches on the latent variable space. Batch 517 is introduced to the NIPALS model of batch data set with the operational boundaries defined in table (5.2). Table (5.3) shows that batch 609 quality variables were both on the upper limits for variables 2 and 3 but the variable 4 was on the lower limit.

Table 5-3: Quality measurements of batch 609

	Model Lower Limit	Model Upper Limit	Target	609 Result
Variable 1	1.81	2.59	2	2.35
Variable 2	10.5	35.4	30	35.4
Variable 3	2.85	3.64	3	3.42
Variable 4	1.7	4	3.5	1.7

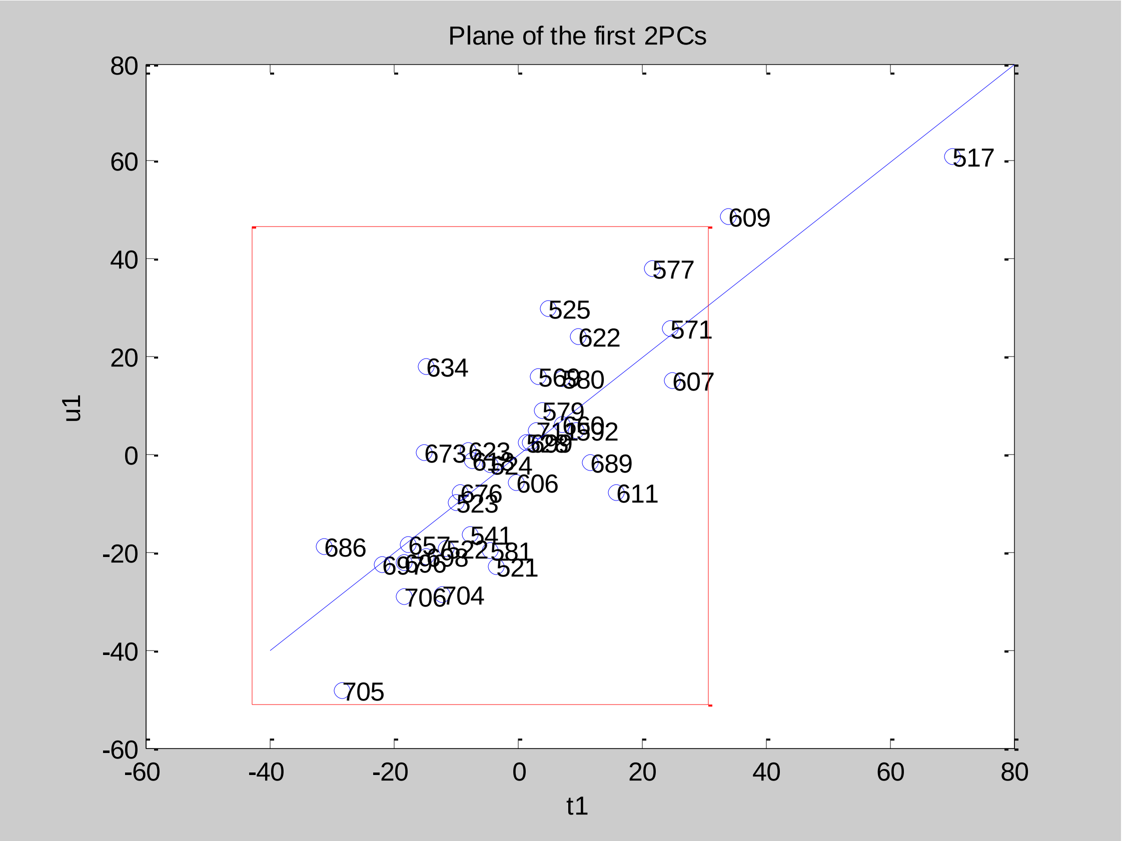


Figure 5-9: Projection of the first pair of the latent variables from batch 517 to the model

Figure (5.9) illustrates the projection of batch 517 to the model with operational boundaries are set with a red line in a square. Batch 517 clearly shows off spec and outside of the permitted boundaries of other batches within the model. Batch 609 is also fell outside the operational boundary limit. Upon investigation, it was discovered that batch 609 had barely passed the quality variable limits set by the model.

A review of the Process Variable 1 in figure (5.10) shows that batch 609 had slightly less deviation in Process Variable 1 than 517. As seen in figure (5.10), batch 609 (in yellow) had progressed like other batches. However, it has a lower minimum value of Process Variable 1 than other batches in the model at 65 minutes time during the process.

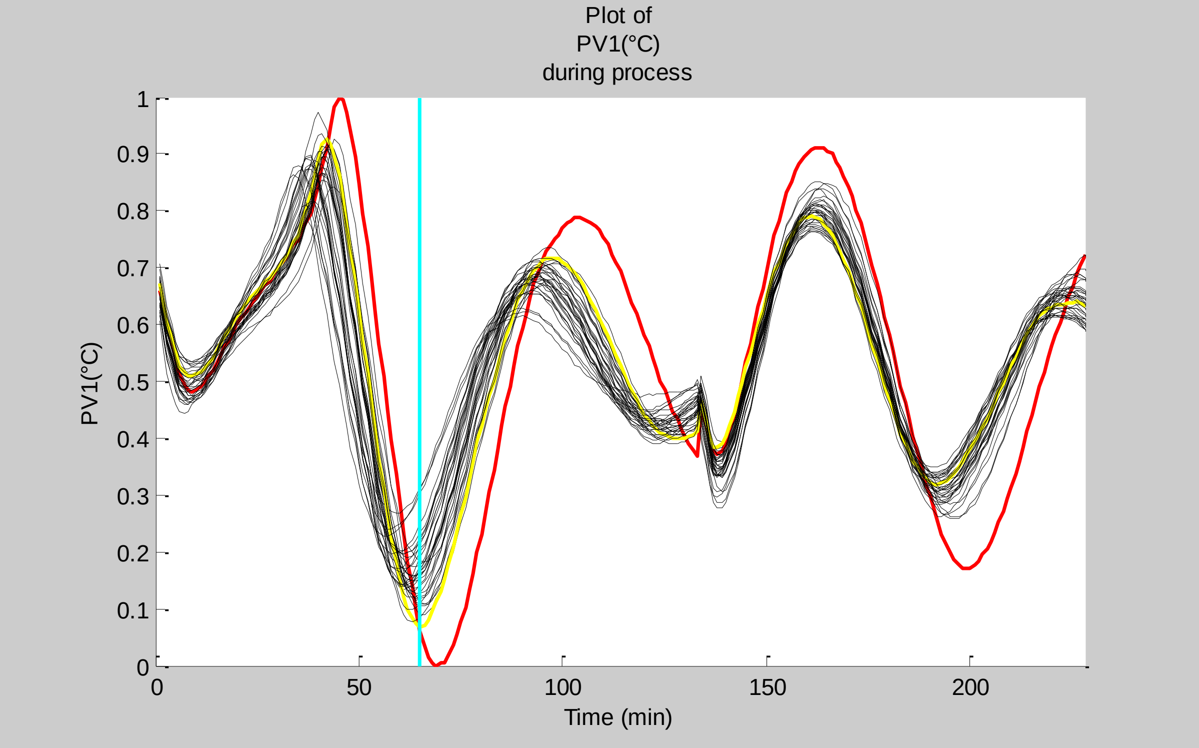


Figure 5-10: Projection of Process Variable 1 for batches 517 and 609

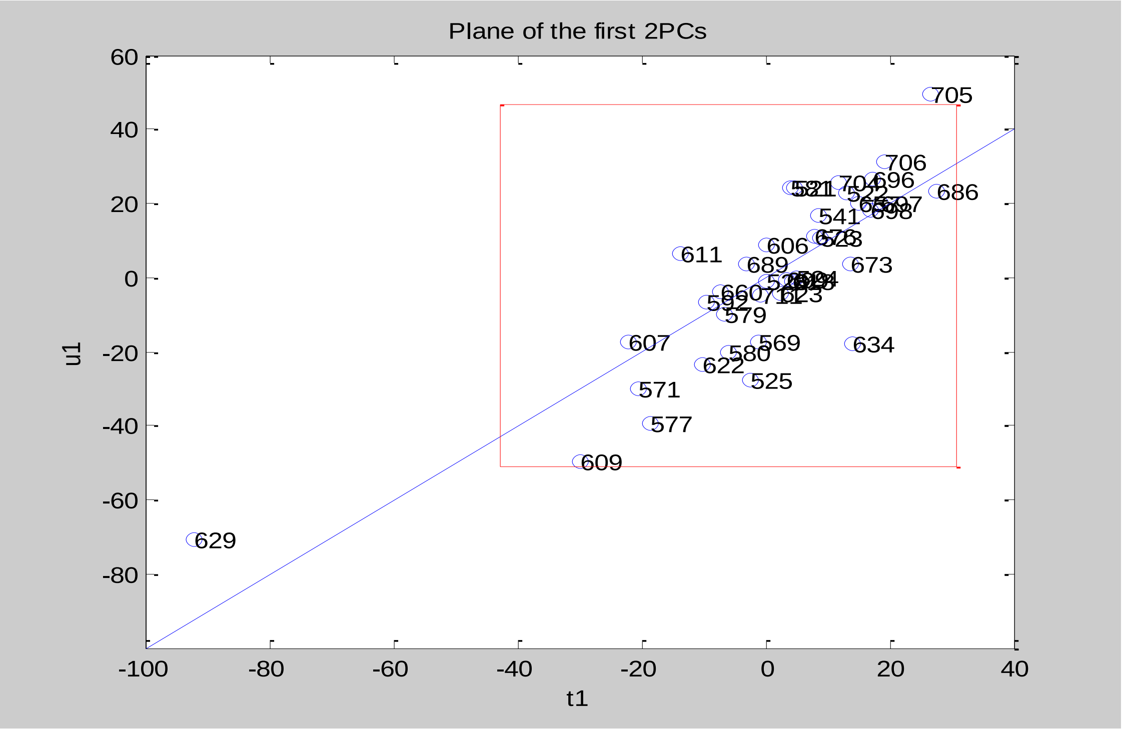


Figure 5-11: Projection of the first pair of the latent variables from batch 629 to the model

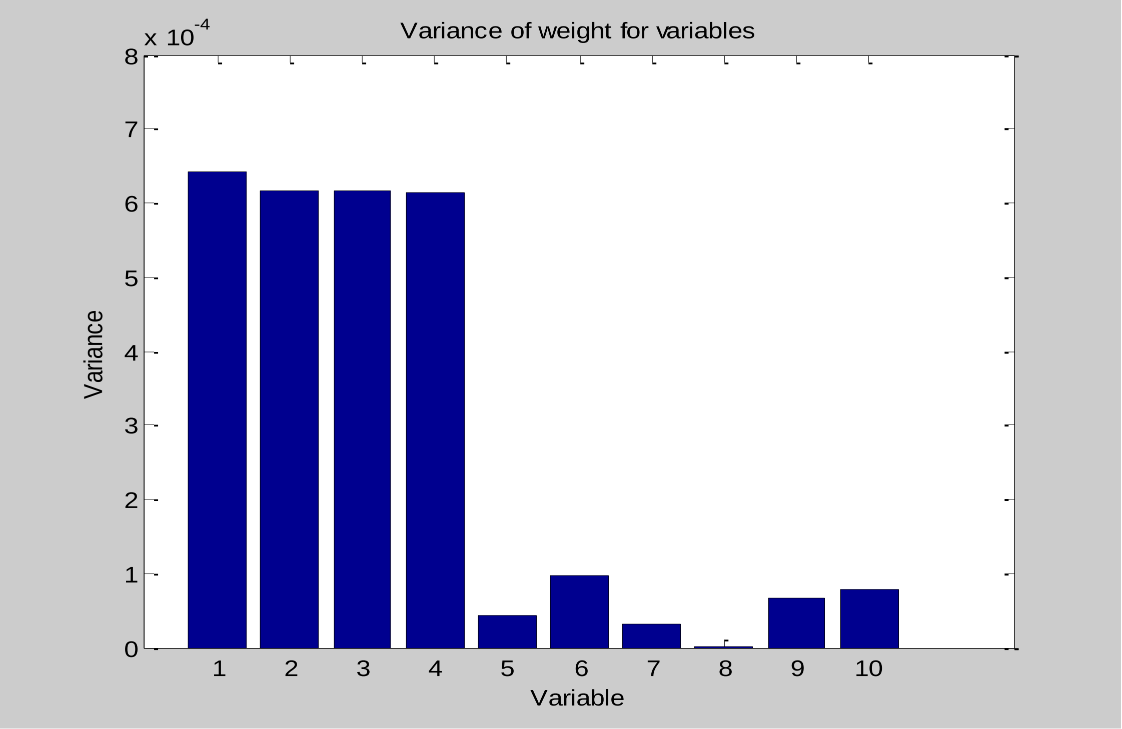


Figure 5-12: Projection of the weights of process variables for the first pair of latent variables

Figure (5.11) illustrates the projection of batch 629 to the model with operational boundaries are set with a red line in a square. Batch 629 clearly shows off spec and outside of the permitted boundaries of other batches within the model. The combined weight matrix, W, for both quality variables and process variables was calculated according to figure(5.6) and folded according to figure (4.16). The graph in figure (5.12) illustrated that the majority of the variation came from process variables 1 through 4.

5.4 Summary

Multivariate Partial Least Squares method was combined with the NIPALS algorithm to correlate the process variable data in the Chemical Toner Manufacturing process to the quality variables. Once the model was built, its operational boundaries for each latent variable were established and the off spec batch was identified through the use of the operational boundary.

The monitoring intention in a batch process is the driving force in choosing between MPLS and MPCA. MPCA uses only process operation measurements and can compress the process data and flag any deviation from average process trajectory, whereas MPLS correlates the quality data to the process measurements. A situation may occur that deviation from average process trajectory, Process Variable 10 in this study, may not have a direct impact on the final quality of the product. Thus, MPLS model wont be able to detect the deviation of the Process Variable 10. In general, it is a good practice to address any deviation in the process trajectories once they occur to prevent a permanent process malfunction. That said MPCA had the advantage of capturing all deviation from average process trajectories regardless of their relation to product quality.

Chapter 6
Conclusion and Recommendation for Future Work

In this study instead of using detailed engineering knowledge about the process, multivariate modeling techniques were utilized to monitor progress of chemical toner manufacturing process. This approach used the information collected from the historical data base of previous batches that are readily available in this computer monitored chemical process. Multi-way Principal Component Analysis MPCA and Multi-way Partial Least Square methods were used in accordance to the methodology used in the statistical process control to track the progression of the existing batch and flag abnormality arising from process variables. Once the abnormalities were detected, the information were used to predict the impact on the quality of final products.

Data preprocessing techniques including centering and scaling coupled with batch time synchronization were deployed to give an equal weight to all process variables involved in this process. One Important aspect of this type of data preprocessing is that, individuals with prior knowledge about the reaction kinetics of the process can assign different factors to process variables. Thus magnifying the impact of deviations among critical process variables.

Multivariate modeling techniques used in this study had tremendous impact in reducing dimension of data by producing two summary variables, T^2 and SPE, to monitor the progression of the entire process. To further align multivariate modeling approach with the concept of statistical process control, control limits for the monitoring charts were extracted from the literature and applied to this study. The power of multivariate modeling technique used in this study is in its ability to utilize unsteady state trajectory data from all process variables associated with a batch; and to bring to account both deviations from the history of measured process variables at any given time.

A novel feature of this research is the validation of the multivariate model with k-fold cross validation technique. In the past studies of multivariate analysis in batch processes, multivariate models were only projected on a two dimensional space to identify abnormal batches. A model built on the latter approach lacks sufficient predictability. To compensate lack of predictability, the previous researchers, used the matrix of loadings along the graphs of individual process variables to identify process variables that were contributing factors to abnormal batches. Combination of multivariate model with k-fold cross validation provided optimal predictability in the model. As observed in this study, the historical data base of past successful batches contain massive amount of data which makes it challenging to derive a statistical model with high value of R^2. That said, only high value of R^2 does not necessary result in optimal predictability. To accomplish this challenge, Prediction Error Sums of Squares, $PRESS$, were introduced and a trade off point (how many latent variable would represent the process) was made with R^2 to reach maximum predictability from the model.

Another topic that was explored in this research was the establishment of operational boundaries of the process in Multi-way Partial Least Square Method. Operational boundaries based in minimum and maximum accepted values of latent variables establish a range that can identify the abnormal batches on the MPLS space.

Like all empirical modeling techniques of chemical processes, assumptions of comparable runs and observable events must be true for the multivariate modeling technique to work. This means, that individual models must be built for different product types manufactured in the plant. Although the frame work of models including MATLAB code are similar, some efforts required to set up separate models for individual products.

Another draw back of this work which is recommended to be the basis for future work is to use the model to provide multivariate control on the process. To put multivariate modeling in use as control strategy, both online multivariate monitoring and batch online synchronization tools are required. In online multivariate monitoring, the future values of process variables for the existing batch run should be filled with the average values of the past historical trajectories. These values would be replaced by actual process values as the existing batch progress. But a more sophisticated approach is required to address the issue of online batch synchronization. As batch run times vary from batch to batch because of variations and disturbances in the process, constant realignment of batch trajectories are required to provide an online batch synchronization technique. Therefore, the future work on this research should more focused on the online batch synchronization and multivariate control.

Appendix A: MATLAB Code for Data Alignment

```
%% Create Matrix for I batches, J variables and K time stamps
imax = 37;
jmax = 10;
kmax = 610;
Faulty_SPE_BatchID = 629;
SPE_Time = 17;
Faulty_Prediction_BatchID = 629;
Prediction_Time = 246;
ReScaley = 0; % 1 to rescale the y axes
A = eye(imax*kmax,jmax);
A1 = eye(1,jmax);
[~, ~, DataBase] = xlsread('DataBase.xls','Data');  % Read the database of
the batch files
DataBase =DataBase(2:end,1:3);
%% Import the data from spreadsheet
for i = 1:imax
    Name = DataBase(i,2);
    FileName = char(Name);
    DataPoints(i) = DataBase(i,3);
    [~, ~, Rawini] = xlsread(FileName,'Data');
    Raw = Rawini(2:end,2:end);
    %Create output variable
    Data = cell2mat(Raw);
    A1 = cat(1,A1,Data);
end
%% Create matrix of data and establish plot access
Faulty_SPE_BatchIndex = find(cell2mat(DataPoints) == Faulty_SPE_BatchID);
Faulty_Prediction_BatchIndex = find(cell2mat(DataPoints) ==
Faulty_Prediction_BatchID);
A1(1,:) = [];
A=A1;
Y_Min_Array = min(A,[],1);  % find min of each variable
Y_Max_Array = max(A,[],1);  % find max of each variable
%% plot the data each variable, j, is plotted in a separate figure
% there should be jmax figures and each figure has imax lines
for j = 1:jmax
figure(j);
clf;
%%  define axes min and max
X_Min = 0;
X_Max = kmax;
   if ReScaley == 1
      Y_Min = Y_Min_Array(:,j);
      Y_Max = Y_Max_Array(:,j);
      Y_Range = Y_Max - Y_Min;
      Y_Range = Y_Range*2;
      Y_Min = Y_Min - Y_Range;
      Y_Max = Y_Max + Y_Range;
           if Y_Min <0
              Y_Min = 0;
           end
     else
```

```
      Y_Min = Y_Min_Array(:,j);
      Y_Max = Y_Max_Array(:,j);
   end
axis([X_Min X_Max Y_Min Y_Max]);
%%  plot the j for all batches
hold on;

    for i = 1:imax
        startrow = ((i*kmax)-kmax+1);
        endrow =  i*kmax;
        startcolumn = j;
        endcolumn = j;

         if i == Faulty_SPE_BatchIndex
           LineColour = 'r';
           Width = 2;
           Size = 2;
        elseif i == Faulty_Prediction_BatchIndex
           LineColour = 'r';
           Width = 2;
           Size = 2;
        else
           LineColour = 'k';
           Width = 1;
           Size = 1;
         end

        jArray = A(startrow:endrow,startcolumn:endcolumn);
        %seperate variable(j)for each batch (i)

        plot(jArray,LineColour,'LineWidth',Width,...
                               'MarkerSize',Size);
        xlabel('Time');
        ylabel(Rawini(1,j+1));
        title(['Plot of' ,Rawini(1,j+1),'during process']);
    end


        LineColour = 'c';

        plot([SPE_Time SPE_Time],[Y_Min
Y_Max],LineColour,'LineWidth',1,...
                'MarkerSize',1);
        plot([Prediction_Time Prediction_Time],[Y_Min
Y_Max],LineColour,'LineWidth',1,...
                'MarkerSize',1);
hold off;
end
```

Appendix B: MATLAB Code for Data Preprocessing

The following code illustrates the technique used to rearrange the raw data of batches including rearrangement from 3D to 2D, mean centering, and auto scaling.

```
%% Create Matrix for I batches, J variables and K time stamps
imax = 36;
jmax = 10;
kmax = 228;
A = eye(imax,jmax*kmax);
A1 = eye(1,jmax*kmax);
[~, ~, DataBase] = xlsread('DataBase.xls','Data');
% Read the database of the batch files
DataBase =DataBase(2:end,1:3);
%% Import the data from spreadsheet
for i = 1:imax
    Name = DataBase(i,2);
    FileName = char(Name);
    DataPoints(i)  = DataBase(i,3);
    [~, ~, Rawini] = xlsread(FileName,'Data');
    Raw = Rawini(2:end,2:end);
    %Create output variable
    Data = cell2mat(Raw);
    jkArray = Data(1,:);
%Allocate imported array to column variable names
for k = 2:kmax
    Strk=int2str(k);
    K ='K';
    TimeStamp = strcat(K, Strk);
    TimeArray = Data(k,:);
    jkArray = cat(2,jkArray,TimeArray);
end
    A1 = cat(1,A1,jkArray);
end
%% Normalize Data
%Delete the first Row and put A1 back into A
A1(1,:) = [];
A=A1;
% The data matrix with normalization
%% Create boxplot of raw data for the first jmax columns
Variables = Rawini(1,2:jmax+1);
RawData = A;
ExampleRawData = RawData(:,1:10);
figure(1);
clf;
boxplot(ExampleRawData,Variables);
title(['Raw Data']);
%% Create boxplot of mean centred data
meanx=mean(A);
MeanRawData = (A-meanx(ones(imax,1),:));
ExampleMeanRawData = MeanRawData(:,1:10);
```

```
figure(2);
clf;
boxplot(ExampleMeanRawData,Variables);
title(['Raw Data with mean centering']);
%%
stdx=std(A);
X=(A-meanx(ones(imax,1),:))./stdx(ones(imax,1),:);
ScaledRawData = X;
ExampleScaledRawData = ScaledRawData(:,1:10);
figure(3);
clf;
boxplot(ExampleScaledRawData,Variables);
title(['Raw Data centered and scaled to unit variance']);
```

Appendix C: MATLAB Code for Modeling Process Measurement

The following code illustrates the NIPALS technique used to perform PCA on raw data of batches including rearrangement from 3D to 2D, mean centering, and auto scaling. NIPALS code of this section was published by (Cao 2008).

```
%% Create Matrix for I batches, J variables and K time stamps
imax = 36;
jmax = 10;
kmax = 228;
PC_Cutoff = 5; % Define the number of principal components
A = eye(imax,jmax*kmax);
A1 = eye(1,jmax*kmax);
[~, ~, DataBase] = xlsread('DataBase.xls','Data');  % Read the database of
the batch files
DataBase =DataBase(2:end,1:3);
%% Import the data from spreadsheet
for i = 1:imax
    Name = DataBase(i,2);
    FileName = char(Name);
    DataPoints(i) = DataBase(i,3);
    [~, ~, Raw] = xlsread(FileName,'Data');
    Rawini = Raw;   % keep track of what was in the raw matrix
    Raw = Raw(2:end,2:end);
    %Create output variable
    Data = cell2mat(Raw);
    jkArray = Data(1,:);
%Allocate imported array to column variable names
    for k = 2:kmax
    Strk=int2str(k);
    K ='K';
    TimeStamp = strcat(K, Strk);
    TimeArray = Data(k,:);
    jkArray = cat(2,jkArray,TimeArray);
    end
    A1 = cat(1,A1,jkArray);
end
%% Normalize Data
%Delete the first Row and put A1 back into A
A1(1,:) = [];
A=A1;
% The data matrix with normalization
meanx=mean(A);
stdx=std(A);
X=(A-meanx(ones(imax,1),:))./stdx(ones(imax,1),:);
Xini=X; % Keep Xini to make sure the calculation is correct
B=X'*X;
%% NIPALS Algorithm   % (Cao 2008)
% T ( Scores) and P (loadings)
T=zeros(imax,jmax*kmax);
P=zeros(jmax*kmax);
% tol for convergence
```

```
tol=1e-6;
for r=1:jmax*kmax
    %find the column which has the maximum norm
    [dum,idx]=max(sum(X.*X));
    t=A(:,idx);
    %storage to judge convergence
    t0=t-t;
    %iteration if not converged
    while norm(t-t0)>tol
          normtol(r)= norm(t-t0);
        %iteration to approach the eigenvector direction
        p=X'*t;
        %normalize the vector
        p=p/norm(p);
        %save previous t
        t0=t;
        %t is a product of eigenvalue and eigenvector
        t=X*p;
    end
    %subtracing PC identified..... X is also the matrix of residuals
    X=X-t*p';
    T(:,r)=t;
    P(:,r)=p;
    %Calculate R Square
    VarE = var(X,0,2);
    VarX = var((T*P'+X),0,2);
    RSq(r)= 1-((VarE)'/(VarX)');
    NormX(r) = norm(X);
    if r >=PC_Cutoff
        break
    end
end
Xfin=T*P'+X; % Keep Xfin to make sure the calculation is correct. (Xfin =
Xini).
             % X is the matrix of residuals
Prediction_Y = T*P';
%% Create data Point labels
DataPoint_Count = (1:imax)'; %transpose vertically
DataPoint_String = num2str(DataPoint_Count);
%DataPoints = cellstr(DataPoint_String);

%% Variance of Residuals of all Batches
VarEfin = var(X,0,2); % calculate Variance of the matrix of residuals. a
replacement of Q square.
%SPE control limit calculations: weight, g, and degree of freedom, df,
%caculated using mean and variance of distribution
%g = v/2*m , df = 2*m^2/v
%Lim(a)=g*chi2inv(a,df) where "a" is confidence limit

VarEfin_Weight = var(VarEfin)/(2*mean(VarEfin));
VarEfin_df = (2*(mean(VarEfin)^2))/var(VarEfin);
VarEfin_Chi = chi2inv(0.95,VarEfin_df);
VarEfin_Lim = VarEfin_Weight*VarEfin_Chi;
```

```
figure(11);
clf;
plot(VarEfin,'--ks','LineWidth',2,...
                'MarkerEdgeColor','b',...
                'MarkerFaceColor','b',...
                'MarkerSize',5);
bar(VarEfin,1);
xlabel('BatchNumber');
ylabel('Variance');
title(['Variance of Residuals of all Batches with a
',num2str(PC_Cutoff),'PC Model']);
text(DataPoint_Count,VarEfin,DataPoints);
hold on;
plot([1 imax],[VarEfin_Lim VarEfin_Lim],'--g','LineWidth',2,...
                'MarkerSize',5);
hold off;
%% Plot Data in 2D Principal Component Space with 95% Confidence Interval
%Hotelling's for 2 PC plane: T^2 = Hotelling, t = PC, s = std of t
%T^2 =(t1^2/s1^2)+(t2^2/s2^2)
%[t1^2/((s1^2)*Hotelling)]+[t2^2/((s2^2)*Hotelling)]= 1
%T^2 (A,a) = (((N-1)*(N+1)*A)/(N*(N-A)))*Fa(A,N-A)
%Fa(A,N-A) = (finv(1-a,A,N-A))
%For A Components
%On N Observations
%For 100(1-a)% confidence limit
%range to plot over
sdtT = std(T,0,1);
Ellip_a_ini=sdtT(:,1);
Ellip_b_ini= sdtT(:,2);
Ellip_C=[0,0];
Ellip_N = 50;
Ellip_theta = 0:1/Ellip_N:2*pi+1/Ellip_N;
%Assign Hotelling's
Hotelling_95 = (((imax-1)*(imax+1)*r)/(imax*(imax-r)))*(finv(0.95,r,imax-
r));
%Recalculate a & b
Ellip_a_fin = sqrt(Hotelling_95)*Ellip_a_ini;
Ellip_b_fin = sqrt(Hotelling_95)*Ellip_b_ini;
% Parametric equation of the ellipse
state(1,:) = Ellip_a_fin*cos(Ellip_theta);
state(2,:) = Ellip_b_fin*sin(Ellip_theta);
% Coordinate transform (since ellipse is axis aligned)
EllipX = state;
EllipX(1,:) = EllipX(1,:) + Ellip_C(1);
EllipX(2,:) = EllipX(2,:) + Ellip_C(2);
% Plot
figure(201);
clf;
plot(EllipX(1,:),EllipX(2,:));
hold on;
plot(Ellip_C(1),Ellip_C(2),'rO');
axis equal;
plot(T(:,1), T(:,2),'O');
```

```
xlabel('PC1');
ylabel('PC2');
title(['Plane of the first  ',num2str(PC_Cutoff),' PC. A Model for all
Batches with 95% C.L.']);
text(T(:,1), T(:,2),DataPoints);
hold off;
figure(202);
clf;
plot(T(:,1), T(:,2),'O');
xlabel('PC1');
ylabel('PC2');
title(['Plane of the first  ',num2str(PC_Cutoff),' PC. A Model for all
Batches']);
text(T(:,1), T(:,2),DataPoints);
%% Plot 2D Principal Component as time series with 95% Confidence Interval
%Hotelling's as the time series T(i)^2 = Simga[(t(i,r)/sr)^2] where
% r is the number of PCs and i is the number of batches
sdtT = std(T,0,1);
for  i_1 = 1:imax
        Ti = 0;
    for Tr = 1:r
        Ti= (T(i_1,Tr)/sdtT(:,Tr)).^2 + Ti;
    end

    Hotelling_Seq(i_1)= Ti;
end
figure(21);
clf;
plot(Hotelling_Seq,'--ks','LineWidth',2,...
                'MarkerEdgeColor','b',...
                'MarkerFaceColor','b',...
                'MarkerSize',5);
xlabel('BatchNumber');
ylabel('Hotelling’s T2');
title(['Hotelling’s, T2 per batch Number the first
',num2str(PC_Cutoff),'PC.  A Model for all batches']);
text(DataPoint_Count,Hotelling_Seq,DataPoints);
hold on;
plot([1 imax],[Hotelling_95 Hotelling_95],'--g','LineWidth',2,...
                'MarkerSize',5);
hold off;
%%
X_Sq = X.^2; % calculate SSQ of the matrix of residuals. a replacement of
Q square.
% double check the calculation for Variacne of the matrix of residuals
%SPE control limit calculations: weight, g, and degree of freedom, df,
%caculated using mean and variance of distribution
%g = v/2*m , df = 2*m^2/v
%Lim(a)=g*chi2inv(a,df) where "a" is confidence limit
SSQ_X = sum(X_Sq,2);
SSQ_X_Weight = var(SSQ_X)/(2*mean(SSQ_X));
SSQ_X_df = (2*(mean(SSQ_X)^2))/var(SSQ_X);
SSQ_X_Chi = chi2inv(0.95,SSQ_X_df);
```

```
SSQ_X_Lim = SSQ_X_Weight*SSQ_X_Chi;
figure(101);
clf;
plot(SSQ_X,'--ks','LineWidth',2,...
                'MarkerEdgeColor','b',...
                'MarkerFaceColor','b',...
                'MarkerSize',5);
xlabel('BatchNumber');
ylabel('Variance');
title(['Sum of Square of Residuals for ',num2str(PC_Cutoff),'PCs']);
text(DataPoint_Count,SSQ_X,DataPoints);
hold on;
plot([1 imax],[SSQ_X_Lim SSQ_X_Lim],'--g','LineWidth',2,...
                'MarkerSize',5);
hold off;
%% SPE contribution for the abnormal batch
%Calculate the Q statistics for a specific batch and produce
%a time series plot of SPE for that batch. it also produces variable
%contribution plot for the same batch
Faulty_SPE_BatchID = input('Insert the batch ID with abnormal SPE: ');
Faulty_SPE_BatchIndex = find(cell2mat(DataPoints) == Faulty_SPE_BatchID);
Faulty_SPE_Array1 = X(Faulty_SPE_BatchIndex,:);
Faulty_SPE_Matrix = eye(1,jmax);
for  k_1 = 1:kmax
        Faulty_SPE_Array2= Faulty_SPE_Array1(:,1:jmax);
        Faulty_SPE_Matrix = cat(1,Faulty_SPE_Matrix,Faulty_SPE_Array2);
        Faulty_SPE_Array1(:,1:jmax) = [];
end
Faulty_SPE_Matrix(1,:) = [];
Faulty_SPE_Time = var(Faulty_SPE_Matrix,0,2);
Faulty_SPE_Var = var(Faulty_SPE_Matrix,0,1);
figure(12);
clf;
plot(Faulty_SPE_Time,'--ks','LineWidth',2,...
                'MarkerEdgeColor','b',...
                'MarkerFaceColor','b',...
                'MarkerSize',5);
xlabel('Batch Time');
ylabel('Variance');
title(['Variance of Residuals for the batch with abnormal batch. Batch #
is  ',num2str(Faulty_SPE_BatchID)]);
hold on;
plot([1 kmax],[VarEfin_Lim VarEfin_Lim],'--g','LineWidth',2,...
                'MarkerSize',5);
hold off;
figure(13);
clf;
bar(Faulty_SPE_Var);
plot(Faulty_SPE_Var,'--ks','LineWidth',2,...
                'MarkerEdgeColor','b',...
                'MarkerFaceColor','b',...
                'MarkerSize',5);
xlabel('Variable');
```

```
ylabel('Variance');
title(['Variance of Residuals for variables of abnormal batch. Batch # is
',num2str(Faulty_SPE_BatchID)]);
%% SPE contribution for the abnormal batch. Determine if the variable
contributes to the fault
SPE_Matrix = eye(1,jmax);
for i_1 = 1:imax
SPE_Array1 = X(i_1,:);
    for  k_2 = 1:kmax
        SPE_Array2= SPE_Array1(:,1:jmax);
        SPE_Matrix = cat(1,SPE_Matrix,SPE_Array2);
        SPE_Array1(:,1:jmax) = [];
    end
end
SPE_Matrix(1,:) = [];
SPE_TotalVar = var(SPE_Matrix,0,1);
SPE_TotalMean = mean(SPE_Matrix,1);
SPE_TotalVar_Weight = SPE_TotalVar./(2*(SPE_TotalMean));
SPE_TotalVar_df = (2*(SPE_TotalMean.^2))./SPE_TotalVar;
SPE_TotalVar_Chi = chi2inv(0.95,SPE_TotalVar_df);
SPE_TotalVar_Lim = SPE_TotalVar_Weight.*SPE_TotalVar_Chi;
SPE_Time = input('Insert the time of abnormality for the batch with
abnormal residuals: ');
SPE_Time_Array = Faulty_SPE_Matrix(SPE_Time,:);
figure(14);
clf;
bar(SPE_Time_Array.^2,'w');
hold on;
plot(SPE_TotalVar,'--ks','LineWidth',2,...
                'MarkerEdgeColor','b',...
                'MarkerFaceColor','b',...
                'MarkerSize',5);
hold off;
xlabel('Variable');
ylabel('Variance Contribution');
title(['bar = contribution at the time point of the abnormals batch. Batch
# is   ',num2str(Faulty_SPE_BatchID), '  dotted line = contribution from
variables of all batches']);
%% Plot model prediction of Data. Prediction contribution for the abnormal
batch
Var_Prediction = var(Prediction_Y,0,2); % calculate Variance of the matrix
of predictions (T*P').
%Prediction control limit calculations: weight, g, and degree of freedom,
df,
%caculated using mean and variance of distribution
%g = v/2*m , df = 2*m^2/v
%Lim(a)=g*chi2inv(a,df) where "a" is confidence limit

Var_Prediction_Weight = var(Var_Prediction)/(2*mean(Var_Prediction));
Var_Prediction_df = (2*(mean(Var_Prediction)^2))/var(Var_Prediction);
Var_Prediction_Chi = chi2inv(0.95,Var_Prediction_df);
Var_Prediction_Lim = Var_Prediction_Weight*Var_Prediction_Chi;
figure(203);
```

```
clf;
plot(Var_Prediction,'--ks','LineWidth',2,...
                'MarkerEdgeColor','b',...
                'MarkerFaceColor','b',...
                'MarkerSize',5);
bar(Var_Prediction,1);
xlabel('BatchNumber');
ylabel('Variance');
title(['Variance of predictions for the  ',num2str(PC_Cutoff),'PC
model']);
text(DataPoint_Count,Var_Prediction,DataPoints);
hold on;
plot([1 imax],[Var_Prediction_Lim Var_Prediction_Lim],'--
g','LineWidth',2,...
                'MarkerSize',5);
hold off;
%% Prediction contribution for the abnormal batch
%Calculate the prediction statistics for a specific batch and produce
%a time series plot of SPE for that batch. it also produces variable
%contribution plot for the same batch
Faulty_Prediction_BatchID = input('Insert the batch ID with abnormal
Prediction: ');
Faulty_Prediction_BatchIndex = find(cell2mat(DataPoints) ==
Faulty_Prediction_BatchID);
Faulty_Prediction_Array1 = Prediction_Y(Faulty_Prediction_BatchIndex,:);
Faulty_Prediction_Matrix = eye(1,jmax);
for  k_1 = 1:kmax
        Faulty_Prediction_Array2= Faulty_Prediction_Array1(:,1:jmax);
        Faulty_Prediction_Matrix =
cat(1,Faulty_Prediction_Matrix,Faulty_Prediction_Array2);
        Faulty_Prediction_Array1(:,1:jmax) = [];
end
Faulty_Prediction_Matrix(1,:) = [];
Faulty_Prediction_Time = var(Faulty_Prediction_Matrix,0,2);
Faulty_Prediction_Var = var(Faulty_Prediction_Matrix,0,1);
figure(22);
clf;
plot(Faulty_Prediction_Time,'--ks','LineWidth',2,...
                'MarkerEdgeColor','b',...
                'MarkerFaceColor','b',...
                'MarkerSize',5);
xlabel('Batch Time');
ylabel('Variance');
title(['Variance of Prediction for
batch',num2str(Faulty_Prediction_BatchID)]);
hold on;
plot([1 kmax],[Var_Prediction_Lim Var_Prediction_Lim],'--
g','LineWidth',2,...
                'MarkerSize',5);
hold off;
figure(23);
clf;
bar(Faulty_Prediction_Var);
```

```
plot(Faulty_Prediction_Var,'--ks','LineWidth',2,...
                'MarkerEdgeColor','b',...
                'MarkerFaceColor','b',...
                'MarkerSize',5);
xlabel('Variable');
ylabel('Variance');
title(['Variance of Prediction for variables of abnormal batch. Batch #
is',num2str(Faulty_Prediction_BatchID)]);
%% Prediction contribution for the abnormal batch. Determine if the
variable contributes to the fault
Prediction_Matrix = eye(1,jmax);
for i_1 = 1:imax
Prediction_Array1 = Prediction_Y(i_1,:);
    for  k_2 = 1:kmax
        Prediction_Array2= Prediction_Array1(:,1:jmax);
        Prediction_Matrix = cat(1,Prediction_Matrix,Prediction_Array2);
        Prediction_Array1(:,1:jmax) = [];
    end
end
Prediction_Matrix(1,:) = [];
Prediction_TotalVar = var(Prediction_Matrix,0,1);
Prediction_TotalMean = mean(Prediction_Matrix,1);
Prediction_TotalVar_Weight =
Prediction_TotalVar./(2*(Prediction_TotalMean));
Prediction_TotalVar_df =
(2*(Prediction_TotalMean.^2))./Prediction_TotalVar;
Prediction_TotalVar_Chi = chi2inv(0.95,Prediction_TotalVar_df);
Prediction_TotalVar_Lim =
Prediction_TotalVar_Weight.*Prediction_TotalVar_Chi;
Prediction_Time = input('Insert the time of abnormality for the batch with
abnormal Prediction: ');
Prediction_Time_Array = Faulty_Prediction_Matrix(Prediction_Time,:);
figure(24);
clf;
bar(Prediction_Time_Array.^2,'w');
hold on;
plot(Prediction_TotalVar,'--ks','LineWidth',2,...
                'MarkerEdgeColor','b',...
                'MarkerFaceColor','b',...
                'MarkerSize',5);
hold off;
xlabel('Variable');
ylabel('Variance Contribution');
title(['bar = contribution at the time point of the abnormals batch. Batch
# is ',num2str(Faulty_Prediction_BatchID), '  dotted line = contribution
from variables of all batches']);
```

Appendix D: MATLAB Code for Modeling Product Quality

```
%% Create Matrix for I batches, J variables and K time stamps
imax = 36;
jmax = 10;
kmax = 228;
PC_Cutoff = 2; % Define the number of principal components
A = eye(imax,jmax*kmax);
A1 = eye(1,jmax*kmax);
[~, ~, DataBase] = xlsread('DataBase.xls','Data');
% Read the database of the batch files
DataBase =DataBase(2:end,1:3);
%% Import the data from spreadsheet for X
for i = 1:imax
    Name = DataBase(i,2);
    FileName = char(Name);
    DataPoints(i) = DataBase(i,3);
    [~, ~, Raw] = xlsread(FileName,'Data');
    Rawini = Raw;    % keep track of what was in the raw matrix
    Raw = Raw(2:end,2:end);
    %Create output variable
    Data = cell2mat(Raw);
    jkArray = Data(1,:);
%Allocate imported array to column variable names
for k = 2:kmax
    Strk=int2str(k);
    K ='K';
    TimeStamp = strcat(K, Strk);
    TimeArray = Data(k,:);
    jkArray = cat(2,jkArray,TimeArray);
end
    A1 = cat(1,A1,jkArray);
end
%% Normalize Data
%Delete the first Row and put A1 back into A
A1(1,:) = [];
A=A1;
% The data matrix with normalization
meanx=mean(A);
stdx=std(A);
X=(A-meanx(ones(imax,1),:))./stdx(ones(imax,1),:);
Xini=X; % Keep Xini to make sure the calculation is correct
%% Import the data from spreadsheet for Y
y =1;
Stry=int2str(y);
Y ='Y';
YFileName = strcat(Y, Stry,'.xls');
[~, ~, YRaw] = xlsread(YFileName,'Sheet1');
YRawini = YRaw;    % keep track of what was in the raw matrix
YRaw = YRaw(2:end,3:end);%Create output variable
YData = cell2mat(YRaw);
C=YData;
meanx=mean(C);
```

```
stdx=std(C);
Y=(C-meanx(ones(imax,1),:))./stdx(ones(imax,1),:);
Yini=Y; % Keep Yini to make sure the calculation is correct
%% Make a copy of the original X & Y
Xini = X;
Yini = Y;
%% Determine the size of the matrices
imax = size(X,1); % number of batches
jmaxkmax = size(X,2); % The number of columns (jmax*kmax)
ymax = size(Y,2); % The number of Y vriables
tol = 1e-6; % The tolerance for the convergence test
%% Initialize the outputs
%P = zeros(jmaxkmax,r); % The loadings of X
%Q = zeros(ymax,r); % The principal components of Y
%W = zeros(jmaxkmax,r); % The X weights
%T = zeros(imax,r); % The Principal Components of X
T=zeros(imax,jmax*kmax);
U=zeros(imax,ymax);
W=zeros(jmaxkmax,ymax);
P = zeros(jmaxkmax);
Yres = Y;
%% Determine each of the principal components
for r=1:ymax
    %find the column which has the maximum norm
    [dum,idy]=max(sum(Y.*Y));
    u=C(:,idy);
    %storage to judge convergence
    u0=u-u;
    %iteration if not converged
    while norm(u-u0)>tol
          normto1(r)= norm(u-u0);% keeep track of the changes in the norm
        %iteration to approach the eigenvector direction
        w=X'*u;
        %normalize the vector
        w=w/norm(w);
        t = X * w;
        q = (Y'*t)/(t'*t);
        u0=u;
        u = (Y*q)/(q'*q);

    end
    %subtracing PC identified..... X is also the matrix of residuals
    p =(X'*t)/(t'*t);
    X = X - t*p';
    Y = Y - t*q';
    T(:,r)=t;
    U(:,r)=u;
    P(:,r)=p;
    W(:,r)=w;
    Q(:,r)=q;

    if r >=PC_Cutoff
        break
    end
end

T(:,r+1:jmax*kmax)=[];
U(:,r+1:ymax)=[];
P(:,r+1:jmax*kmax)=[];

Xfin=T*P'+X; % Keep Xfin to make sure the calculation is correct. (Xfin =
Xini).
             % X is the matrix of residuals
Yfin=T*Q'+Y; % Keep Yfin to make sure the calculation is correct. (Yfin =
Yini).
             % X is the matrix of residuals
%% Create data Point labels
DataPoint_Count = (1:imax)'; %transpose vertically
DataPoint_String = num2str(DataPoint_Count);
DataPoints = YRawini(2:end,2);
%%
figure(1);
clf;
plot(T(:,1), U(:,1),'O');
xlabel('t1');
ylabel('u1');
title(['Plane of the first ',num2str(PC_Cutoff),'PCs']);
text(T(:,1), U(:,1),DataPoints);
%%
figure(2);
clf;
plot(T(:,2), U(:,2),'O');
xlabel('t2');
ylabel('u2');
title(['Plane of the first ',num2str(PC_Cutoff),'PCs']);
text(T(:,2), U(:,2),DataPoints);
%%
figure(3);
clf;
plot(T(:,1), U(:,1),'O');
xlabel('t1');
ylabel('u1');
title(['Plane of the first ',num2str(PC_Cutoff),'PCs']);
text(T(:,1), U(:,1),DataPoints);
hold on;
lsline
hold off;
%%
figure(4);
clf;
plot(T(:,2), U(:,2),'O');
xlabel('t2');
ylabel('u2');
title(['Plane of the first ',num2str(PC_Cutoff),'PCs']);
text(T(:,2), U(:,2),DataPoints);
hold on;
```

```
lsline
hold off;
```

Bibliography

Abdi, Herve, and Lynne Williams. "Principal Component Analysis." *Wiley Interdisciplinary Reviews: Computational Statistics* 2 (2010): 433-459.

barnes, R.J, M.S Dhanoa, and S.J Lister. "Standard Normal Variate transformation and de-trending of near-infrared diffuse reflectance." *Appl. Spectrosc* 43 (1989): 772-777.

Box, G.E.P. "some theorems on quadratic forms applied in the study of analysis of variance problems." *The Annals of Mathematical Statistics* , 1954: 290-302.

Bro, Rasmus, and Age Smilde. "Centering and scaling in component analysis." *J. Chemometrics* 17 (2003): 16-33.

Cao, Yi. *MATLAB CENTRAL.* February 19, 2008. http://www.mathworks.com/matlabcentral/fileexchange/18760-partial-least-squares-and-discriminant-analysis.

Díez, Marta Dueñas. *Population balance modeling and passivity-based control of particulates processes,applied to the Silgrain process.* Extended Abstract, European Federation of Chemical Engineering, 2004.

Duchesne, C, and J.F MacGregor. "Establishing Multivariate Specification Regions for Incoming Materials." *Journal of Quality Technology,* 36 (2004): 78-94.

Geladi, P, and B Kowalski. "Partial Least Squares Regression." *Analytica Chimica Acta* 185 (1986): 1-17.

Harshman, RA, and ME Lundy. "Data Preprocessing and the extended PARAFAC model." (In Research Methods for Multimode Data Analysis) 1984.

Hong, Jeong Jin, Jie Zhang, and Julian Morris. "Fault Localization in Batch Processes through Progressive Principal Component Analysis Modeling." *Industrial & Engineering Chemistry Research* , 2011.

Kaistha, N, and C Moore. "Extraction of Event Times in Batch Profiles for Time Synchronization and Quality Predictions." *I&ECH* 40 (2001): 252-260.

Kassidas, A, J.F MacGregor, and P Taylor. "Synchronization of Batch Trajectories Using Dynamic Time Wraping ." *AIChE* 44 (1998): 864-875.

Kourti, T, J Lee, and J.F MacGregor. "Industrial Applications of Projection Methods for Multivariate Statistical Process Control." *Computers and Chemical Engineering* 20 (1996): 745 - 750.

Kourti, T, P Nomikos, and J.F MacGregor. "Analysis, Monitoring and Fault Diagnosis of Batch Processes Using Multi-Block and Multi-Way PLS." *Journal of Process Control*, 1995.

Kozub, D.J., and J.F. MacGregor. *State Estimation for Semi-Batch Polymerization Reactors.* Vol. 47. Chemical Engineering, 1992.

Kresta, J., J.F. MacGregor, and T.E. Marlin. *MultivariatStatistical Monitoring ofProcess Operating Performance.* Vol. 69. Canadian Journal ofChemical Engineering, 1991.

Leung, Michelle. *Production Scheduling Optimization of a Plastics Compounding Plant with Quality Constraints.* 2009.

Lin, Bao, Bodil Recke, Jorgen Knudsen, and Sten Jorgensen. "A systematic approach for soft sensor development." *Computers & Chemical Engineering* 31 (2006): 419-425.

MacGregor, J. F., and T. Kourti. "Statistical process control of multivariate processes." *Control Engineering Practice*, 1995.

MacGregor, J.F, A Penlides, and A.H Hamielec. "Control of Polymerization Reactors." (Polymer Process Engineering) 2 (1984): 179-206.

MacGregor, J.F, C Jaeckle, C Kiparissides, and M Koutoudi. "Monitoring and Diagnosis of Process Operating Performance by Multi-Block PLS Methods with an Application to Low Density Polyethylene Production." (AIChE Journal) 40 (1994): 826-838.

Munoz, S.G. "Batch Process Improvement using latent variable methods." PhD Thesis , Chemical Engineering , McMaster University , Hamilton , 2004.

Neogi, D, and C Schlags. "Mutivariate Statistical Analysis of an Emulsion Batch Process." *I & ECR* 37 (1998): 3971-3979.

Nomikos, P, and J.F MacGregor. "Multivariate PSC Charts for Monitoring Batch Processes." *Technometrics* 37 (1995).

Nomikos, P., and J. F MacGregor. "Monitoring of batch processes using multi-way principal component analysis." (AIChE) 1994.

Nomikos, P., and J. F. MacGregor. "Multivariate SPC charts for monitoring batch processes." (Technometrics) 1995.

Nomikos, P., and J. F. MacGregor. "Multiway partial least squares in monitoring batch process." *Chemometrics and Intelligent Laboratory Systems*, 1995.

Paatero, P, and U Tapper. "Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values." *Environmetrics* 5 (1994): 111-126.

Ramkrishna, Doraiswami. *Population Balances, Theory and Applications to Particulate Systems in Engineering.* Academic Press, 2000.

Randolph, A.D, and M.A Larson. *Theory of Particulate Processes.* Academic Press, 1988.

Stephanopoulos, G., G. Henning, and H. Leene. *A Modeling Language for Process Engineering.* 1990.

Westerhuis, J, T Kourti, A Kassidas, P Taylor, and J.F MacGregor. "Synchronizing the Trajectories of the Process Variables for on-Line Monitoring of Batch Runs With Unequal Duration." *SSC6, Porsgrunn, Norway*. 1999.

Westerhuis, J.A, T Kourti, and J.F MacGregor. "Comparing alternative approaches for multivariate statistical analysis of batch process data ." *J.Chemometrics* 13 (1999): 397-413.

Wold, S, N Kettaneh, H Friden, and A Holmberg. "Modelling and diagnostics of batch processes and analogous kinetic experiments." *chemometrics and intelligent laboratory systems* 44 (1998): 331 - 340.

Zhang, J., E. B. Martin, and A. Morris. "Process monitoring using non-linear statistical techniques." *Journal of Chemical Engineering* , 1997.