The Thesis Committee for Vinay Kumar

Certifies that this is the approved version of the following thesis:

Multi-state PLS based Data-driven Predictive Modeling for Continuous Process Analytics

APPROVED BY SUPERVISING COMMITTEE:

Supervisor: _____

Robert H. Flake

Co-Supervisor: _____

Thomas F. Edgar

Multi-state PLS-based Data-driven Predictive Modeling for Continuous Process Analytics

by

Vinay Kumar, B.E.

Thesis

Presented to the Faculty of the Graduate School

of the University of Texas at Austin

in Partial Fulfilment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

May 2012

Dedication

This thesis is dedicated with love to my Spiritual Guru, Parents, Wife, Family and Friends who kept my spirits up in my most difficult and challenging times. Without their unconditional support, this thesis would have not been completed.

Acknowledgements

I would like to heart fully thank My Guru and Mentor Dr. Thomas F Edgar for giving me the opportunity to work under him on this project and for his valuable guidance, encouragement and support during my time in his research group. Working under him was a life changing experience for me.

This thesis would not have been complete without the support of Dr. Ricardo Dunia, University of Texas, Austin, Terry Blevins, Mark Nixon and Willy Wojsznis from Emerson Process Management. I would like to thank them for their valuable time and support. Their expert advice and patent information enabled me to deliver my best in this project.

I would also like to thank Dr. Robert H Flake from ECE Department for supervising this project and for his encouragement and advice all through my Graduate studies.

I want to thank all my research group members Gill, Kody, Wess, Doug, Anh, Jong, Ivan, Kriti, Ben, Ramiro and Xiao. They have been a great source of knowledge and I have learnt many things from them through the control group seminars in our research group. I would also like to thank ex-Edgar group member Ninad Patwardhan for helping me with my questions.

I would like to thank my lovely wife Karuna, my Mom, Dad and my extended family for unconditionally loving and supporting me during difficult times in my life. Thanks to my sister Rekha for keeping me enthusiastic and motivated for being able to complete this work, Thanks to my friends Sameer and Virag for making my Masters and stay at Austin an enjoyable journey.

May 2012

Abstract

Multi-state PLS-based Data-driven Predictive Modeling for Continuous Process Analytics

Vinay Kumar, M.S.E.

The University of Texas at Austin, 2012

Supervisor: Robert .H. Flake

Co-Supervisor: Thomas .F. Edgar

Today's process control industry, which is extensively automated, generates huge amounts of process data from the sensors used to monitor the processes. These data if effectively analyzed and interpreted can give a clearer picture of the performance of the underlying process and can be used for its proactive monitoring. With the great advancements in computing systems a new genre of process monitoring and fault detection systems are being developed which are essentially data-driven.

The objectives of this research are to explore a set of data-driven methodologies with a motive to provide a predictive modeling framework and to apply it to process control. This project explores some of the data-driven methods being used in the process control industry, compares their performance, and introduces a novel method based on statistical process control techniques.

To evaluate the performance of this novel predictive modeling technique called Multi-state PLS, a patented continuous process analytics technique that is being developed at Emerson Process Management, Austin, some extensive simulations were performed in MATLAB. A MATLAB Graphical User Interface has been developed for implementing the algorithm on the data generated from the simulation of a continuously stirred blending tank. The effects of noise, disturbances, and different excitations on the performance of this algorithm were studied through these simulations. The simulations have been performed first on a steady state system and then applied to a dynamic system .Based on the results obtained for the dynamic system, some modifications have been done in the algorithm to further improve the prediction performance when the system is in dynamic state. Future work includes implementing of the MATLAB based predictive modeling technique to real production data, assessing the performance of the algorithm and to compare with the performance for simulated data.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The process control industry is driven by the goals of profitability, and strives continuously for methods for improved controllability and monitoring. Standard process controllers (PID, MPC, etc.) have been employed to keep the process in control and within product specification limits. Increasingly complex process systems are causing large set of process variables to be created and thus arises the need for their efficient monitoring. Two critical aspects of any process is **a)** process monitoring and **b)** process prediction. Process monitoring essentially consists of continuously monitoring any kind of anomalies also called **faults** in the running system and then removing them.

Recent significant advancements in industrial measurement technologies and process control technologies such as programmable logic controllers (PLC) and distributed control systems (DCS) have made it possible to collect huge amount of process data, which can include several process variables such as temperature, flow, etc. These variables at any time can reflect the operating performance of the running plant. Thus, proper understanding and analysis of the accumulated data is critical to the profitable running of any process plant.

Process monitoring classification and one of its classes, data-driven methods will be discussed in detail in section 1.1

1.1 Process Monitoring

A process can be broadly divided in three categories: 1) continuous 2) batch or 3) a mixture of both continuous and batch. A process is traditionally controlled with PID controllers or more recently by model predictive controllers to compensate for the external and internal disturbances. Faults can be considered as special events occurring during the monitoring of a process. Early and proper fault detection, particularly in continuous processes, is crucial for its performance and operation and thus maintains its economic viability. The presence of multiple input and output variables in conjunction with the stochastic noise make the system a multivariate one and therefore poses great challenges in fault detection. The process monitoring techniques are an amalgamation of statistical theory, pattern recognition and system identification theory. Process monitoring methods, which principally include fault detection, fault identification and removal techniques have been classified into three main types: namely, quantitative model-based, qualitative model-based and process history or data-driven based. V. Venkatasubramanian et al. [46] have provided a good classification of the various fault diagnosis algorithm types and is shown in Figure 1.

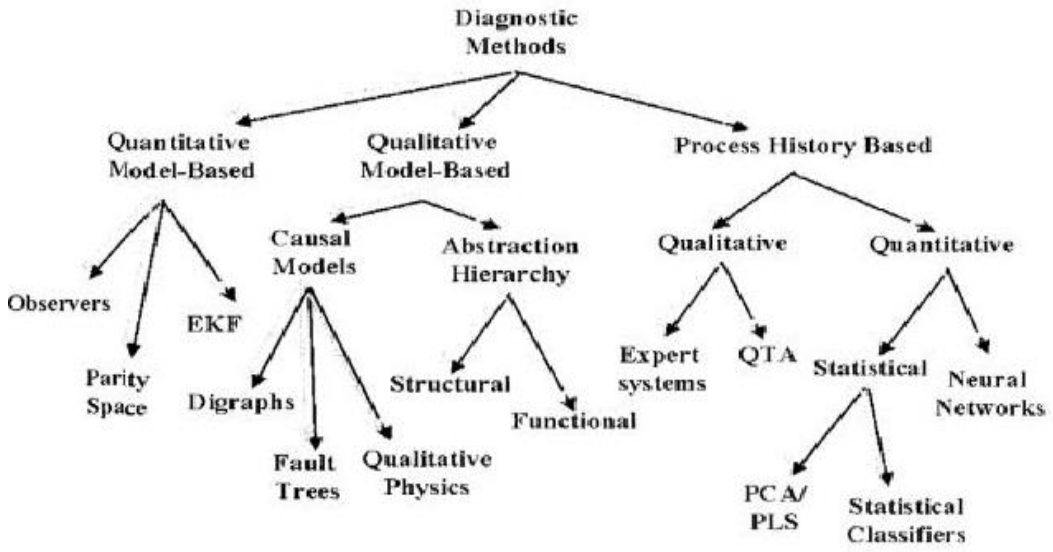


Figure 1.1. General Classification of Fault detection methods

The **Quantitative model** based methods use mathematical models, which are based on the first principles equations governing the underlying process. For a successful implementation of these methods, the system must have enough sensors to provide ample information for the mathematical model to be validated. The main advantage of this approach is the actual physical representation of process and its easy realization. The disadvantage includes the fact that this method does not perform well for large systems with multiple input and outputs. As seen in Figure 1, some typical examples of quantitative model based methods include observer based, parity space and extended Kalman filter methods.

The **Qualitative model** based methods are used when full mathematical details of the underlying system are not available. These methods too like the quantitative models are not well suited for large systems with multiple inputs and outputs. Typical examples include causal models, abstraction hierarchy, etc.

The **Process history** based methods also called the data-driven methods have been successful in many ways where the qualitative and quantitative methods have failed.

These methods utilize process data, which are derived from large numbers of sensors and actuators installed throughout the plant. data-driven methods generate models based on the historical process data and thus are used for process monitoring, fault detection and prediction. Their advantage lies in the fact that since they employ techniques that lower the dimensionality of the underlying system, their application to multi input-output systems is easy. These methods also involve statistical process control techniques, which are applied on the huge data sets generated in process plants. Typical methods used in data-driven process monitoring methods includes expert systems, neural networks, PCA/ PLS and other statistical concepts.

1.2 Data-driven Methods In Process Monitoring- A Survey

Multivariate monitoring techniques have been developed to take historic data including a number of process variables and to look at the normal range of each process variable and the relationship between them. The general hypothesis assumed in these methods is that a historical data set defines a characteristic region of the n-dimensional space that represents normal operation. Any new data outside this region is considered an outlier and thus is characterized as a fault.

In order to predict the response variable, we need to generate a model for it in terms of the predictor variables. The model, which gives a good relation between the variables, can be linear, quadratic, etc. The best model suited and used in the process industry is the linear model. Due to more than one response and predictor variables in process control systems, the regression methods are of multivariate nature and purely based on the statistical mathematics.

1.2.1 A Brief Introduction of Principal Component Analysis (PCA)

The origin of principal component analysis (PCA) dates back to Pearson (1901) and Hotelling (1933). The central idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of a large number of interrelated variables while retaining as much as possible of the variation present in the data set. This is achieved by transforming to a new set of variables, the principal components (PCs), which are uncorrelated and ordered so that the first few retain most of the variation present in all of the original variables (Jolliffe, 2002).

If we have X as a matrix (n x m) of the predictor variables which are pre-processed by applying mean centring and scaling by the standard deviation, then we can get the matrix of variance-covariance (m x m) as $X^T * X$. This matrix defines the eigenvectors wherein the maximum variation of X occurs.

Then PCA compresses the information matrix in

$\mathbf{X = [x_1^T, x_2^T , x_3^T ... x_k^T]}$ x_k= row vector corresponding to the process variable measurement at k^{th} time

$\mathbf{T= [t1, t2, t3 ... t_a]}$ t_a = column vector of the projection of all measurement on to the a th eigenvector.

With $P = [p_1, p_2 \ldots p_a]$ as the eigenvector of covariance matrix X^T*X which is orthogonal, i.e., $P^T*P= I$, where I is the identity matrix.

After extracting all the eigenvectors as per PCA definition, X can be reconstructed as

$$\mathbf{X= T * P^T} \qquad (1.0)$$

and if a reduced number of eigenvectors are chosen which give the maximum variability in X as a<k, then the model in eq. (1.0) takes the form as

$$\mathbf{X= T * P^T + E}$$

where E is the residual error, which cannot be explained by the model.

With its simple yet effective use, PCA has been applied in many fields including the process control industry. There are numerous publications that cite the successful use of PCA technique in fault detection and process control. Broadly, these techniques have been applied in two ways: linear PCA techniques or nonlinear PCA techniques.

1.2.2 Applications of PCA in Process Monitoring

Recent developments in PCA techniques have furthered the classical PCA application originally proposed by Wold, S., et al. [48]. The basic PCA technique has been augmented with problem specific features to make it more effective. Below is a brief list of various versions of PCA applied in the industry more recently.

Li, W. et al. [30] have used recursive PCA for adaptive process monitoring, which has been proposed keeping in mind the time varying nature of industrial process and by using a time variant PCA and recursively updating the mean, covariance and correlation structures, and number of principal components used, which are the building blocks of PCA. They have proposed two algorithms for the illustration of the recursive PCA and have successively applied it on the rapid thermal annealing process in semiconductor processing.

Many recent publications have focused on the extension of the classical PCA to compensate for the dynamic changes in the steady state continuous processes like grade changes, etc. This can be thought of as autocorrelation among the process variables.

A good example of the above-mentioned fact regarding the enhanced version of PCA has been provided by Treasure, et al. [44] in Dynamic multivariate statistical process control using subspace identification. In order to tackle the

various process changes like throughput changes, they use subspace model identification (SMI) with the use of dynamic PCA.

Their approach is to get a low dimensional "state space-based model" of the process and then to apply the PCA method to reduce the number of variables and use the contribution charts to detect the abnormality. The method takes further the application of SMI to define the T^2 statistics earlier applied by Negiz, and Cinar [34], Norvilas et al. [35] and capture the T^2 statistics for both the process and state variables, which gives rise to the error in variables (EIV) approach. The method is demonstrated by applying it to simulated CSTR and getting a twofold model dimension reduction with SMI. It needs to be mentioned here that a similar approach of using state space for monitoring has also been provided by Li and Qin [29] and Wang and Qin [47].

Recent research has also directed its effort in the development of varieties of alternate PCA techniques, which is important to mention in the context of this report, namely, Multiscale PCA, and Multiblock PCA.

Multiscale PCA is normally applied to small and complicated processes and is a technique based on the wavelet transformations where individual signals are decomposed in different scales or frequencies and the decomposed scale is used in process monitoring. The decomposed scales are then used to capture the correlation among the process variables. An earlier mention of this method is found in Bakshi et al. [3] and Misra et al. [33]. Yoon and MacGregor [52] have given a further related extension to multiscale PCA. The method proposes an algorithm for decomposing the process signals and proposing a scale contribution of T^2 to a fault and time variant alternative of the same. The same approach is applied to the SPE and the limit for both statistics is

calculated at a particular significance level for fault detection and identification. A more recent application example is found in Xia and Pan [51].

Multiblock PCA applies to large processes that are composed of a multitude of units and thus many variables per unit. The method decomposes the whole unit into blocks of variables, which are highly coupled within block, and least coupled different blocks. This method makes it easier to diagnose and detect a fault in a larger multi-unit process. Apart from the block statistics monitoring, the method also proposes the use of superblock or the statistics of the entire process. An earlier example of the method is found in MacGregor et al. [32]. A more recent application of Multiblock PCA in semiconductor process is found in Cherry and Qin [9], Zhiqiang et al. [56] in two-level multiblock statistical monitoring for plant-wide processes, as well as Perk and Cinar [36].

A good review article of multivariate process control charts is by Bersimis et al. [4]. The author provides a good review of all the work related to PCA and various multivariate methods being used in process control. As per the author, Tsung et al. [45] presented a method focused on process control schemes that are based on a combination of the process outputs and automatic control actions using adaptive PCA. Chiang et al. [10] discussed the use of discriminant analysis, PCA, and PLS for fault diagnosis in chemical processes. Norvilas et al.[35] have developed an intelligent process monitoring and fault-diagnosis environment by interfacing multivariate statistical process control monitoring techniques and knowledge-based systems for monitoring multivariate process operation. Lane et al. [27] proposed an extension to PCA, which enables the simultaneous monitoring of a number of product grades or recipes. Kano et al. [22] proposed a novel statistical monitoring method, which was based on PCA, called moving PCA, in

order to improve process-monitoring performance. The aim of this method is to identify changes in the correlation structure. Chen and Liu [7] proposed on-line batch process monitoring using dynamic PCA and dynamic PLS models.

Finally, Arteaga and Ferrer [1] dealt with the missing-data problem in the estimation of latent variables scores from an existing PCA model. Badcock et al. [2] proposed two alternative projection techniques that focus on the temporal structure of multivariate data. Ramaker et al. [38], using simulation, studied the effect of the size of the training set and number of principal components on the false-alarm rate in statistical process monitoring

1.2.3 Brief Summary of Nonlinear PCA Methods

The PCA methods discussed previously in this article work best for linear processes, but for nonlinear processes, a different approach is required. As mentioned in Zhang et al. [54], the main difference between PCA and nonlinear PCA is the introduction of nonlinear mappings between the original and reduced dimensional space. A linear principal component minimizes the sum of the orthogonal deviations between a straight line and the data while the nonlinear approach summarizes the data by a smooth curve, which is determined by the nonlinear relationships between all the variables. The paper uses the approach of principal curves, demonstrates some fault simulation, and shows how the SPE statistics can be monitored to detect fault in a nonlinear process. A classical example of the use of non-linear PCA technique is given in Dong and McAvoy [14].

As mentioned earlier, the process industry uses various nonlinear PCA methods also based on neural networks. Much literature is available where various nonlinear PCA approaches have been used. Cho et al. [11] have developed a new method of fault

identification for process monitoring using kernel principal component analysis. This paper takes forward the idea of the kernel PCA developed by Schölkopf et al. [40]. The method extends the input space to arbitrary high dimension space and then finds the principal components in that feature space. Fault identification is then achieved with the help of the contribution plots. Lee et al. [28] and Choi et al. [12] show a good use of Kernel PCA.

1.3 Process Variable Prediction

The method of process variable or the response variable quality prediction of analyzing the predictor variables using statistical tools is commonly called "Regression Analysis". The use of regression analysis in process control has been attractive due to its inherent advantages as described below:

1. Sometimes the response variable is very expensive to measure and the predictor variable measurement is easily available, thus the advantage of expressing response variable in terms of predictor variables.
2. The response variable measurement in case of a hazardous process or process with extreme conditions of temperature and pressure is difficult and the prediction of this kind of variable with easy to measure controllable predictor variables with regression analysis is attractive.

In order to predict the response variable, we need to generate a model for it in terms of the predictor variables. The model, which gives a good relation between the variables, can be linear, quadratic, etc. The best model suited and used in the process industry is the linear model. Due to more than one response and predictor variables in process control systems, the regression methods are of multivariate nature and purely based on the statistical mathematics. The process control industry has seen an

increased use of sophisticated multivariate process control techniques, which are an extension of univariate process control techniques earlier used and which include statistical analysis as their most important part, and thus called multivariate statistical process control (MSPC), and multivariate statistical quality control (MSQC).

1.3.1 Background Information of MSPC/MSQC

The field of MSPC/MSQC is based on solid statistical fundamentals, which are then applied to control to predict the behaviour of an industrial process. With the increase in computing power, these methods, in conjunction with computational software, have benefited the process industry in terms of enhanced productivity and profitability.

As discussed earlier, apart from the control and monitoring of a chemical process in terms of fault detection and identification, the aspect of correctly predicting the important outputs of the process and creating a relevant model of the process for future is equally relevant and important for the industry.

The three basic methods used in the industry for prediction and model generation are multiple linear regression (MLR), principal component regression (PCR) and partial least squares regression or projection to latent structures (PLS). We will briefly discuss the theory of each method and then concentrate on the latest developments in PCR and PLS methods that are being applied in the industry, as they are the most frequently used methods in the industry due to their inherent advantages.

1.3.2 Multiple Linear Regression:

The classical first order linear regression model has the form as

$$Y_i = \beta_0 + \beta_1 x_{1i} + \varepsilon_i, \tag{1.1}$$

where Y_i is the value of the response variable in the ith observation or sample, β_0 is the intercept constant, β_1 is the slope constant , X_i is the i^{th} observation of the independent variable, and ε_i is the random error term which is thought of having the mean of 0 and variance σ^2. The same theory can be extended to multilinear regression, where the model is given as

$$Y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \ldots + \beta_k x_{ki} + \varepsilon_i, \tag{1.2}$$

and β_0 is the intercept factor and β_k is the slope factor associated with the k th variable.

MLR gives the best result when the predictor variables are very uncorrelated, but this situation is not possible practically in industrial process where there are many interrelated variables. Moreover, a slight addition of noise can cause the factors in the model to change and thus the old model is no longer valid. This parametric sensitivity problem is the reason MLR is not generally used in the process industry.

1.3.3 Principal Component Regression:

In order to overcome the problems of dealing with correlated variables, the noise suppressing and dimension reduction method of PCA is used. As discussed earlier in this article, PCA maps X variables (predictor variables) to a reduced subspace, which is defined by the eigenvectors (also called the loadings), of the X space and thus helps to develop a linear model that produces orthogonal values (also called the scores) that have the maximum covariance with the X data. The mathematical expression for PCA has been discussed earlier.

PCR can be thought of an extension of PCA for the modelling of Y (response variable) performed from the X (response variable) where the variables X are regressed on the score matrix T of predictor variables as

$$\mathbf{Y = T^* Q + E} \tag{1.3}$$

As defined in PCA, taking the equation (1) we get

$$\mathbf{Y = X^*P^*Q + E} \tag{1.4}$$

On performing the regression, we get the regression coefficients as

$$\mathbf{B_{PCR} = P^*Q} \tag{1.5}$$

1.3.4 Partial Least Squares Regression:

PLS regression is a technique, which extends the methodology of PCR and thus creates a better performing model. The technique differs from PCR in the sense that apart from decomposing the predictor variables into scores and loadings it also decomposes the response variables into loadings and scores and then performs regression. PLS maximizes the covariance between the predictor variables and the response variables and thus yields a model, which predicts the best values for the future measurements. Early literature for PLS can be found in Geladi and Kowalski (1986). If we have **X** and **Y** as the predictor and response variable matrices, in PLS both the matrices are decomposed as

$$\mathbf{X= T^*P^T + E} \tag{1.6}$$

Where **T** is the score matrix, **P** is the loading matrix, and **E** is the residual matrix of **X** data set and

$$\mathbf{Y = U * Q^T + F} \quad (1.7)$$

where **U** is the score matrix, **Q** is the loading matrix, and **F** is the residual matrix of **Y** data set. The score matrix **T** of **X** is found by rotating **X** by a weighting vector **w** so that the vectors give as much information about **Y** as possible and their covariance structure is maximized. This can be given mathematically as

For $T = \mathbf{Xw}$ the vector **T** is found so that $|\mathbf{Y^T * t}|^2$ **is** maximized.

Because of the effectiveness of the PLS method, it is the technique most used in the process control industry and we present here a review of some recent relevant work related to PLS and its variants for process control.

One foundational paper for modeling with PLS/PCR is by Burnham et al. [6], where they have discussed the regression modeling of various types like latent variable multivariate regression (LVMR), and reduced rank errors-in-variables multivariate regression (EIVR). The paper discusses all of the aspects related to modeling, viz. handling missing data and prediction region with an experimental design.

Kohonen et al. [26] have applied the Multiblock technique in their work on the massive spectral data, which is partitioned into smaller blocks according to types or the particular part of the process. The paper presents and compares four methods: PLS regression (same as classical PLS), Covproc method (that combines PLS and classical regression methods which has been introduced as a pre-processing method to find the best values of the weights **w** so that the goodness of fit and cross validation is as large as possible), Priority regression (by giving each variable priority

number , selecting the highest priority number finding the model taking weights of rest of variables as zero, continuing the same process for all priority number variables), and the Multiblock method (by creating blocks of predictor variables and response variables and finding respective weights that maximally describe Y). More details on the Covproc method can be found in Reinikainen et al. [39].

Kourti [24] has mentioned a recent popular technique, which is being used as an inferential modelling, process monitoring, and control method using inexpensive digital cameras and Multivariate image analysis (MIA). The method as proposed, extracts subtle information from the image that is related to the product quality and uses the information for prediction, monitoring, and control. There have been literatures available applying the multivariate image analysis in the process industry from Bharati and McGregor [5], Liu et al. [31], and Yu et al. [53].

A different approach in monitoring transitions in steady state chemical plants in the form of start up, shut down, feedstock change, and grade change has been taken by Sundarraman et al. [42] by the use of trend analysis. They compare the real time trend with the dictionary trends and apply dynamic feature synchronization algorithm as an expert system to track the location of the transition.

Recent research methods regarding the prediction models with PLS have been compared by Zhang et al. [55]. This paper discusses the formulation of prediction uncertainty confidence interval. To find the standard deviation of the prediction error, they compared four methods, namely ordinary least squares type method, Linearization-based methods, resampling based methods, and u-deviation method. The paper also discusses three approaches like Naïve, Generalised, and Pseudo degree of freedom approach for calculating the degree of freedom consumed by the PLS model.

Ergon [15] has discussed in his paper a rather new approach, Informative PLS score - loading plots for process understanding and monitoring, where for scalar response variables, all PCR/PLSR models can be reduced to an equivalent model with two components only. The Idea of using just 2 components makes it easier for the process operator to monitor just a single plot to detect the fault in the process. The paper also provides an industrial example with the 2PLS algorithm and plots a score-loading plot for the monitoring of the example data.

In another interesting treatment of MSPC technique, Chen et al. [8] have discussed a method based on kernel density estimation to apply it on the T^2 and Q statistics and generate a single monitoring chart that is easy to monitor.

A recent work by Li, et al. [29] has treated a fault as a slowly time-varying auto correlated process, and the fault is estimated from the observations based on fault reconstruction. They have introduced a new index to integrate the fault detection and prognosis based on wavelet de-noising technology and vector autoregressive model.

Fujiwara et al. [16] have developed soft sensors based on correlation just–in-time modeling. This work, in order to prove that the Recursive PLS and the classical just-in-time methods are insufficient when the process characteristics change abruptly, proposes a new method of correlation based on just-in-time modeling to take care of the changing process characteristics.

Kaneko et al. [21] in a very recent paper have focused on industrial polymer process grade change transition detection through the use of soft sensors. They have constructed models to detect the completion of the transition of a grade in a polymer process, so that predicted value conforms to that of actual value after transition. The authors have used the K-nearest neighbour and support vector machines.

Sijmen de Jong [13] has proposed a novel algorithm SIMPLS as an alternative PLS method to the existing NIPALS algorithm .As per the author**,** "The construction of deflated data matrices as in the nonlinear iterative partial least squares (NIPALS)-PLS algorithm is avoided. For univariate, y SIMPLS is equivalent to PLS1 and for multivariate **Y,** there is a slight difference between the SIMPLS approach and NIPALS-PLS2. In practice the SIMPLS algorithm appears to be fast and easy to interpret as it does not involve a breakdown of the data sets".

1.4 A Comparative study of MLR, PCR PLS with Hypothetical Data

1.4.1 Simulation setup

We use a simple blending process to perform simulation (Seborg et al. [41]) and thus do a comparative study of the three process variable prediction techniques being used in the process control industry currently.

A continuous stirred tank is shown in Fig 1.1. The control objective is to blend the two inlet streams to produce an outlet stream with a desired chemical composition. We have made the following assumptions for this simulation:

1. A general version of blending system is considered for simulation.
2. The blending system is a continuously stirred tank with constant liquid density.

3. Volume of blending tank can vary and there is no overflow line.
4. Exit flow rate is not necessarily equal to the sum of inlet flow rates.
5. The blending tank is perfectly mixed.

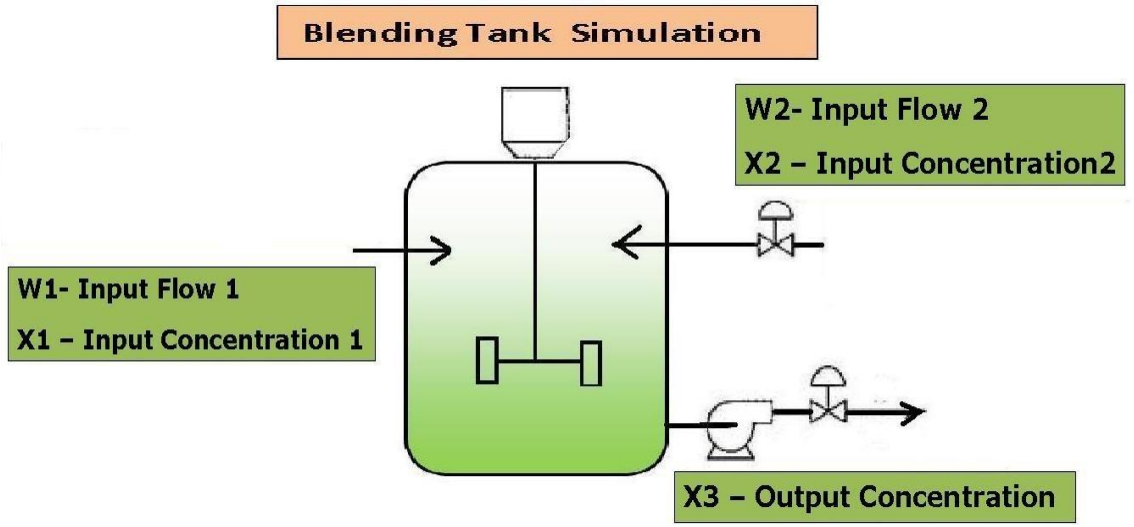


Figure 1.2: The blending system used in the simulation

As an illustrative example, we consider the above blending tank. It is a more general form of the blending system as the overflow line specifically present in blending systems has been omitted. The input streams have flow rates of W1 and W2 kg/min and concentrations of X1 and X2 respectively. The dimensions of tank are taken to be Height=2.5 m, Diameter =2 m. The density of the liquid is ρ=800 kg/m^3.

It is also assumed that the process has been operating for a long period at steady state with flow rates of W1 =110 kg/min and W2 =55 kg/min and mass fractions of X1 =0.16 and X2=0.9.

An unsteady mass balance for the above blending system has the form

$$\left\{\begin{matrix}\text{rate of accumulation}\\ \text{of mass in the tank}\end{matrix}\right\} = \left\{\begin{matrix}\text{rate of}\\ \text{mass in}\end{matrix}\right\} - \left\{\begin{matrix}\text{rate of}\\ \text{mass out}\end{matrix}\right\}$$

The mass of the liquid can be expressed as a product of liquid volume **V** and the density **ρ.** Thus, the mass balance equation can be written as

Mass balance equation

$$\frac{d(V\rho)}{dt} = W_1 + W_2 - W_3 \tag{1.8}$$

$$\frac{\rho A d(h)}{dt} = W_1 + W_2 - W_3 \tag{1.9}$$

The unsteady component balance can also be derived in an analogous manner. Since we have earlier considered the assumptions for the blending tank that it is perfectly mixed, there are no concentration gradients in the tank and the composition of the exit stream is equal to the tank composition.

For the perfect mixing assumption, the rate of accumulation is $\frac{d(V\rho x_3)}{dt}$ where x_3 is the mass fraction of the outlet stream. Thus, we have

$$\frac{d(V\rho x_3)}{dt} = W_1 x_3 + W_2 x_3 - W_3 x_3 \tag{1.10}$$

The steady state model can be derived by equating the left hand side of the equation to zero and we obtain

$$x_3 = \frac{w_1}{w_{1+}w_2}(x_1) + \frac{w_2}{w_{1+}w_2}(x_2) \tag{1.11}$$

For the purpose of input data simulation, a generic case has been considered by generating various values for the predictor variables x_1, x_2, W_1, W_2. To run the simulation white noise has been added to each variable to simulate measurement and process noise.

1. x_1 (Mass concentration of A) has been simulated as an input to the Blending with constant concentration of 0.16 and with a random noise of mean 0 and variance .25

2. x_2 (Mass concentration of pure A) has been simulated as an input to the Blending with constant concentration of 0.9 and with a random noise of mean 0 and variance .0001.

3. w_1 (Flow rate of x_1) has been simulated as a constant value of 110 kg/min with an added random noise of mean 0 and variance 9 .

4. w_2 (Flow rate of x_2) has been simulated as a constant value of 55 kg/min with an added random noise of mean 0 and variance 4.

5. x_3 (Flow rate of x_3) has been calculated from the formula (4) above.

The plot of all the predictor variables is given below in figures 1.3 to 1.8

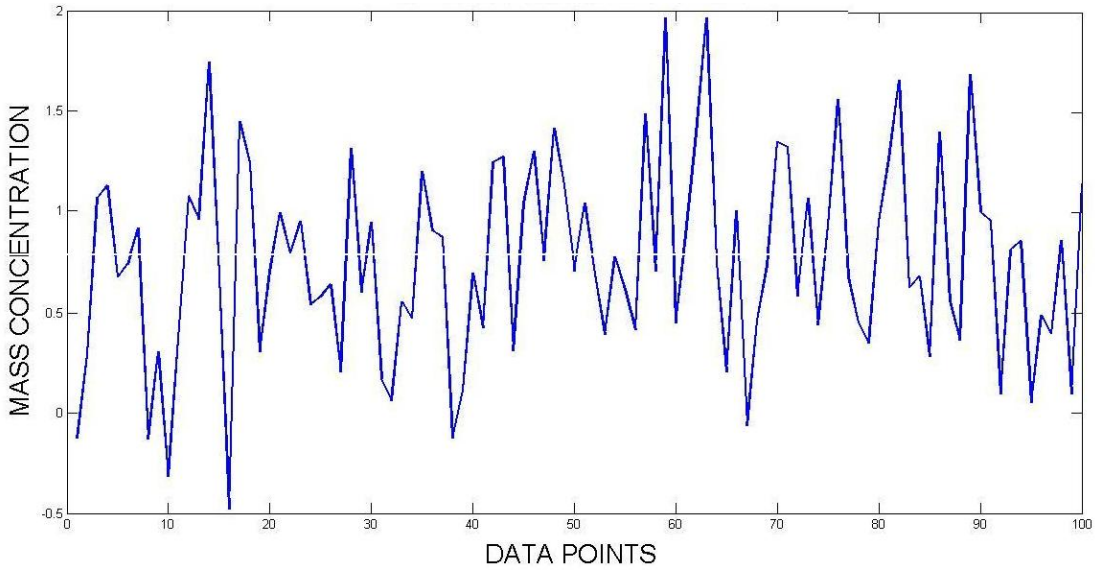


Figure 1.3. Plot of X_1 (mass concentration) predictor variable.

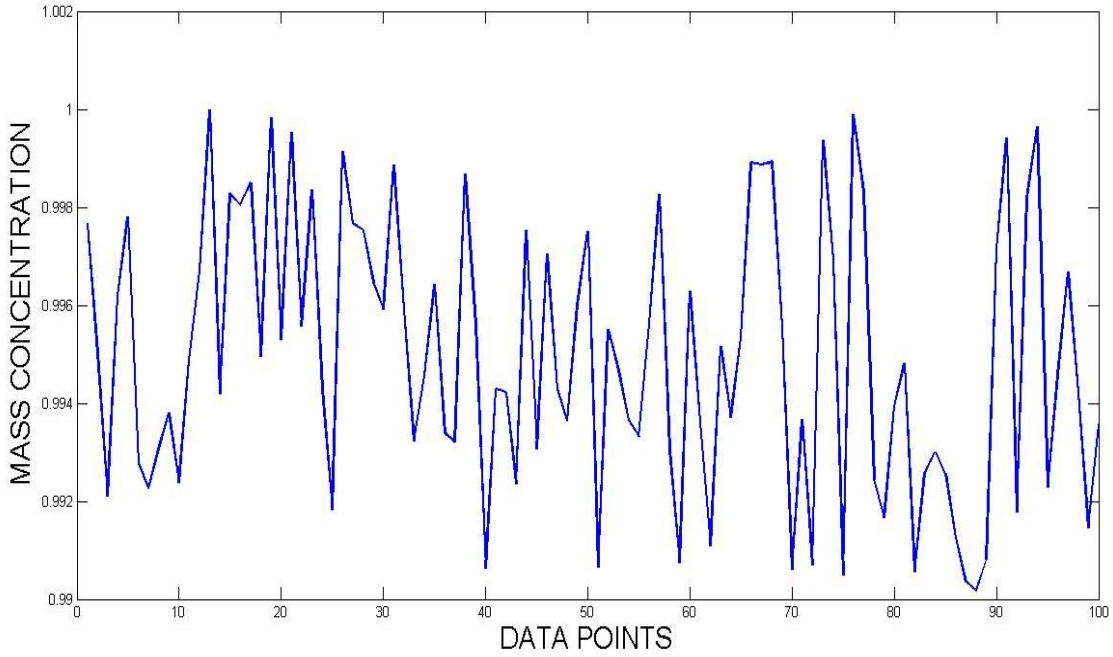


Figure 1.4. Plot of X_2 (mass concentration) predictor variable.

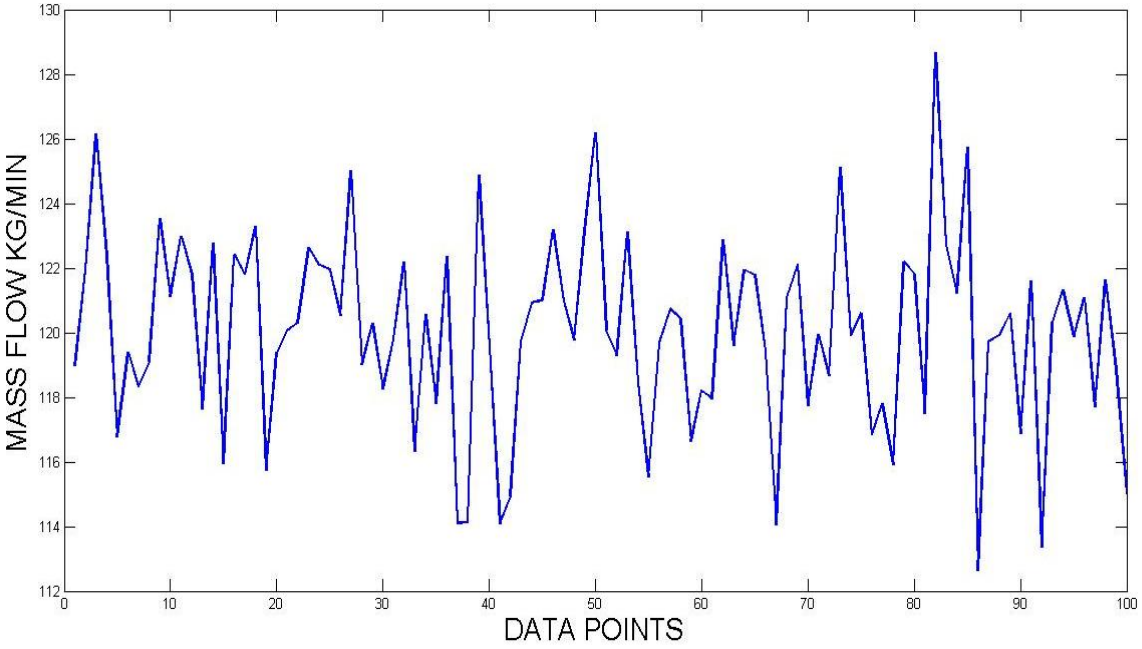


Figure 1.5. Plot of W_1 (mass flow) predictor variable.

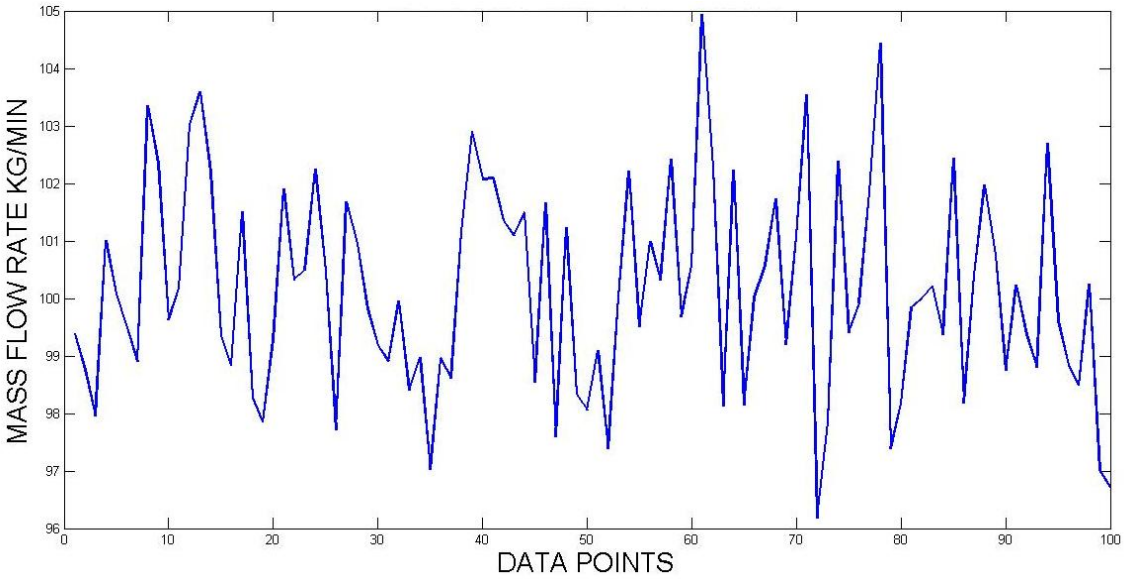


Figure 1.6. Plot of W_2 (mass flow) predictor variable

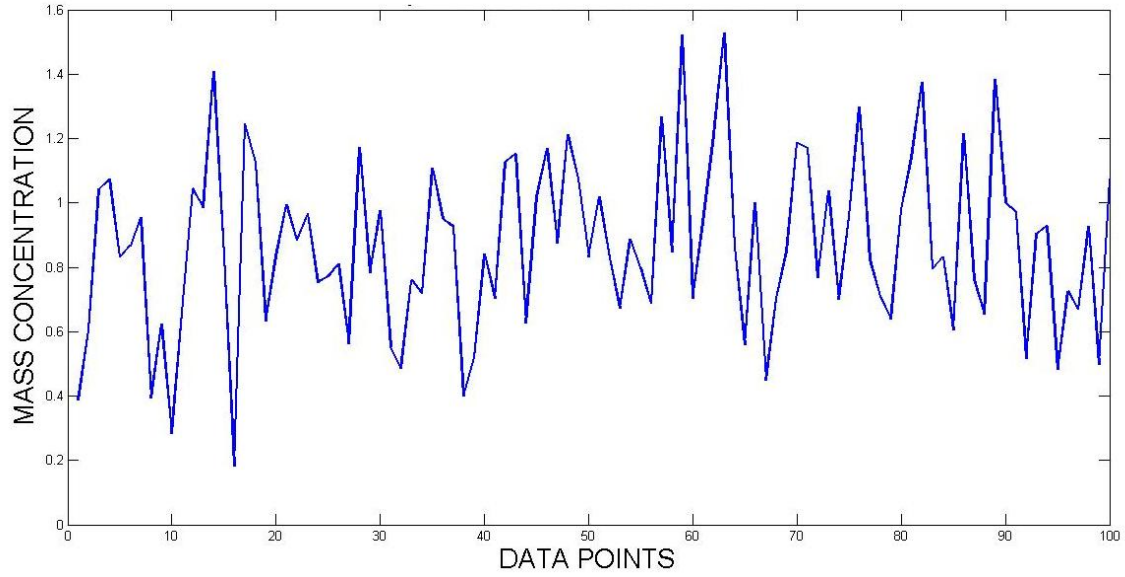


Figure1.7. Plot of X_3 (mass concentration) response variable

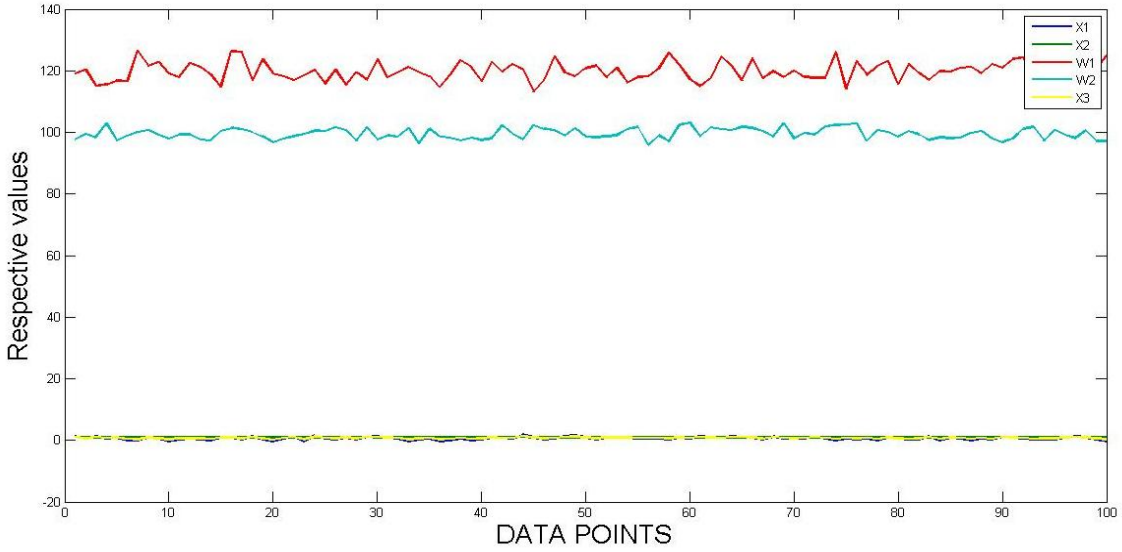


Figure 1.8. Plot of X_1, X_2, W_1, W_2 and X_3 with their respective values

The blending tank predictor variables and the response variables after grouping in training set(40 data points) and test set are analysed with the regression techniques of MLR,PCR PLS(NIPALS and SIMPLS), and the prediction of the test sets analyzed for each technique to compare their performance with respect to the same data set.

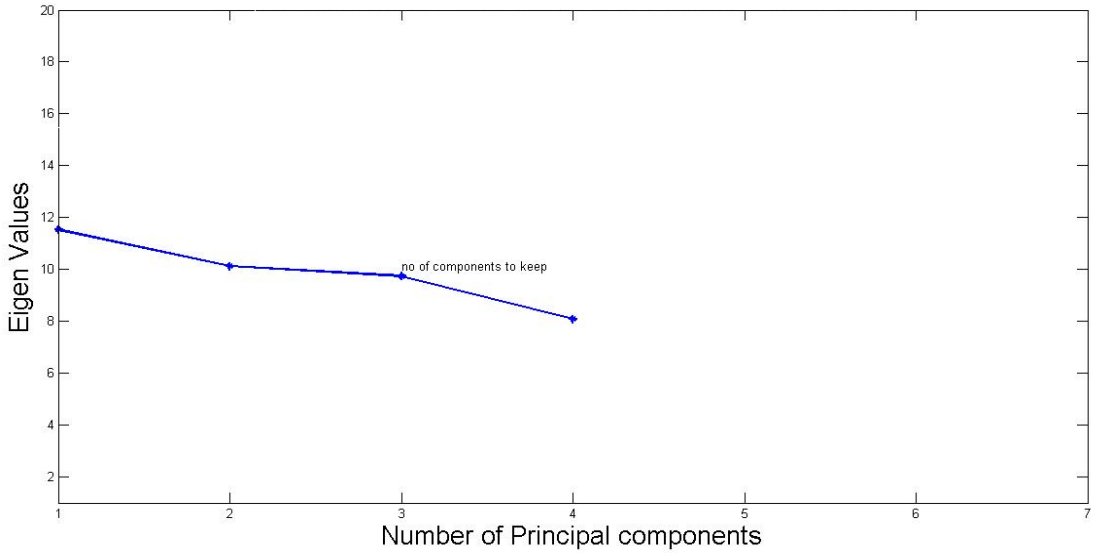


Figure 1.9. Scree Plot for optimum number of principal components.

The above figure depicts the number of principal components that are to be considered important for the generation of the model.

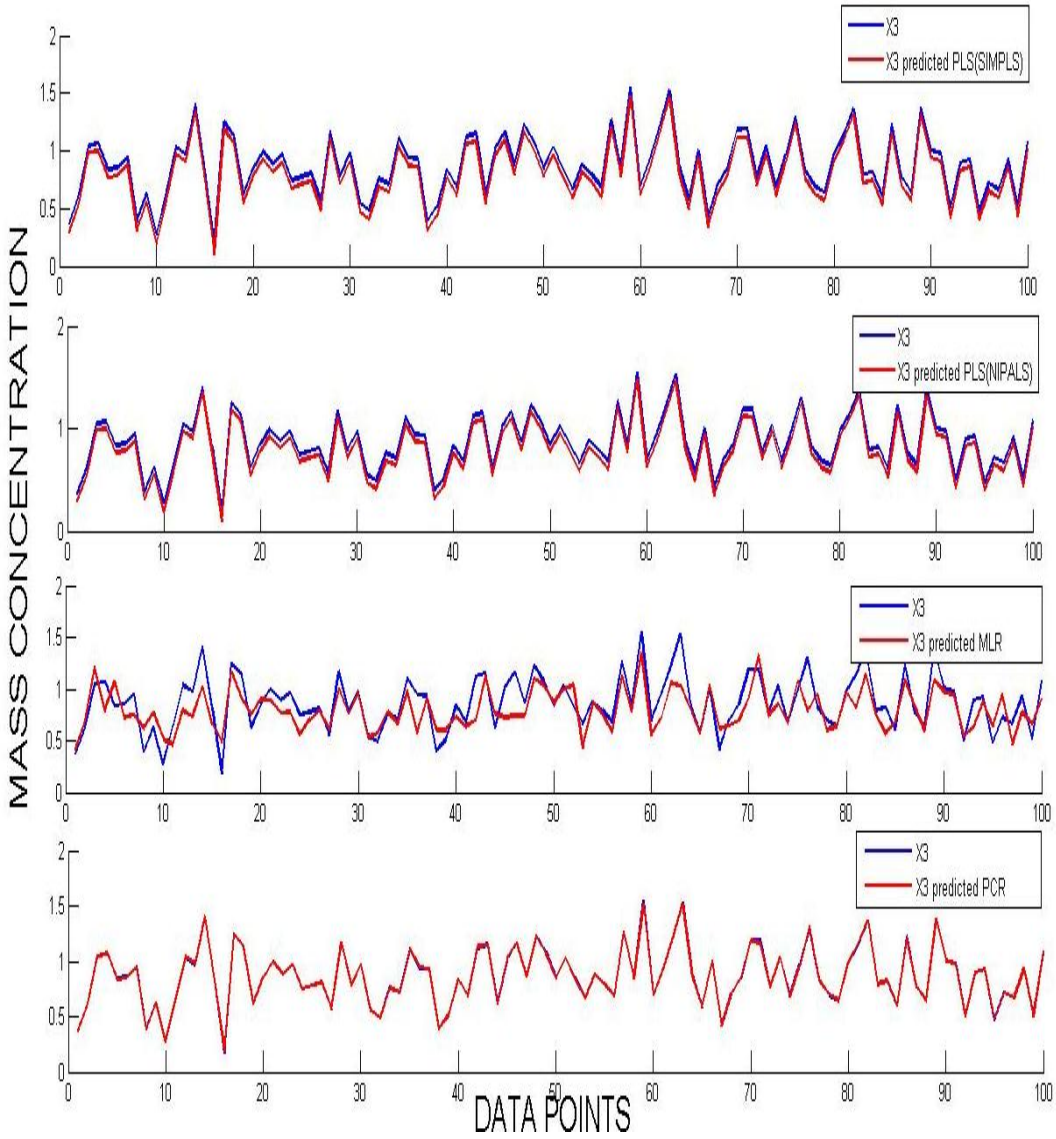


Figure 1.10. Comparison of X3 prediction with measured value for PLS, MLR AND PCR.

It can be seen that PCR and PLS techniques perform best in predicting the output and MLR fails to provide a good prediction.

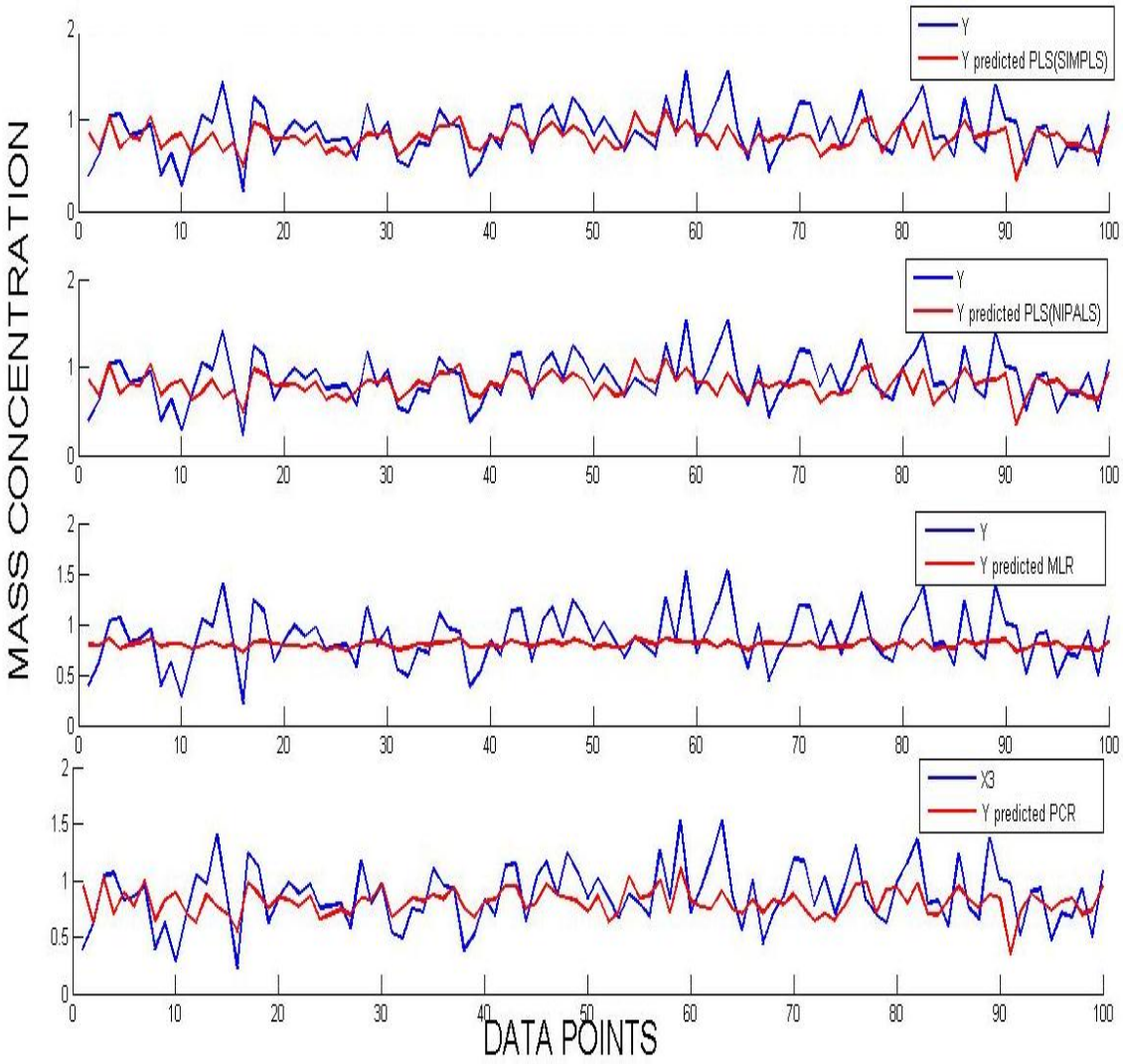


Figure 1.11 Comparison of X3 prediction with measured value for PLS, MLR AND PCR with noise added in the response variable

It can be seen that in the presence of noise again PCR and PLS techniques perform best in predicting the output in presence of noise and MLR completely fails to provide a prediction.

1.4.2 Conclusion

The above result obtained after performing the various regression techniques of PLS (NIPALS & SIMPLS), MLR and PCR clearly depicts the superiority of the PLS and PCR techniques in predicting the values of the response variable due to their numerical conditioning algorithm. Moreover, it is seen that the model prediction in PCR is indistinguishable from the data .The results of NIPALS and SIMPLS algorithm are the same.

The above figure is the result obtained after performing the regression techniques of PLS (NIPALS & SIMPLS), MLR and PCR, with noise added in the response variable which simulates the presence of output sensor measurement noise. The new result corroborates the earlier observation and shows that in the presence of noise the PLS and PCR techniques give better results than the MLR technique. The MLR model is unable to match the data and thus ineffective in the presence of noise

Chapter 2

Multistate-PLS based Predictive Modeling– A novel method

The previous chapter gave a brief idea of the various techniques, or their variants being used in the process control industry for the purpose of process variable prediction. We also presented a simulation test of the three important techniques namely principal component regression (PCR), multiple linear regression (MLR), and partial least squares regression (PLS). As per the case study, it was found that PLS technique gave the best results among the three.

In this chapter we discuss in detail about the two common methods that implement the PLS technique and introduce the novel algorithm of Multi-state PLS, a patented technique being developed at Emerson Process Management, Austin, Texas.

2.1 PLS Regression

After the brief introduction to PLS in section 1.3.4, we will now discuss in detail the PLS regression method and its implementation algorithm. We discussed earlier that PLS regression is an extension of the process monitoring techniques of PCA, PCR and a generalization of the process variable prediction technique of MLR. Just to re-iterate, regression methods have been long used in process control industry in order to generate a model from the predictor variables (also called input variable and denoted as X matrix) in order to predict the response variable (also called the output variable and denoted as Y vector). PLS regression has an inherent advantage over MLR in that it can model a dataset with strong collinearity or correlation and also simultaneously model several response variables Y**.**

As per S. Wold et al [49], “PLS regression was first introduced in 1975 by Herman Wold for the modelling of complicated data sets in terms of blocks of matrices”. It included the a way to estimate the parameters in the models called NIPALS(Non-linear Iterative Partial Least Squares) which also led to the coining of the acronym PLS (Partial Least Squares) . The word Iterative in the acronym NIPALS refers to the fact that each parameter is estimated iteratively and can be considered as the slope of a simple least squares regression between a matrix column or row as the Y variable and another parameter vector as the X-variable. The details of the algorithm are presented in the next section. Further the partial in the PLS acronym indicates that this is a partial regression, since the X-vector is considered fixed in this estimation. The authors of the PLS and its algorithm in order to give it a more descriptive meaning, also at times call it as Projection to Latent Structures. The latent structure refers to the fact that, any system or process is actually influenced by just a few underlying variables as opposed to all of them. The aim of PLS regression is to estimate these unknown number of latent variables.

A variant of NIPALS algorithm, implementing PLS regression is the SIMPLS algorithm referred in section 1.3.4 in chapter 1. As mentioned by of De Jong [13] the PLS factors in this algorithm are calculated directly as linear combinations of original variables maximizing the covariance criterion and simultaneously obeying the orthogonality and normalization restrictions. It differs from the NIPALS approach in a way, by avoiding the construction of the deflated matrices for X and Y. The advantage of this is seen in the better processing speed of the algorithm as compared to NIPALS. We discuss the implementation of SIMPLS algorithm in section 2.3.

2.2 NIPALS (Nonlinear Iterative Partial Least Squares) Regression Algorithm

The NIPALS algorithm before its implementation starts with a pre-processing of both the X and Y matrices. This includes the following two steps:-

1. Transformation: This step is carried out to make the distribution of X and Y symmetrical, if they differ at all. The most common technique applied in transformation is that of a logarithmic transformation.
2. Scaling: This step includes scaling each of the variables of X and Y to unit variance by dividing them by their standard deviations. Further, they are centred by subtracting each variable from their respective averages. This causes each variable to have the same weight.

Following are the steps that implement the **NIPALS** algorithm as introduced by Wold et al(1984) in their paper.

A. Initialize a vector **u,** by making it equal to one of the columns of the output variable **Y.** Thus we have **y = u.**

B. Obtain the **x** weights **w** as:

$$\mathbf{w = X'u / u'u} \tag{2.1}$$

and normalize **w** to have a norm **|| w|| =1.**

C. Calculate **X** scores **t** as **t=Xw.**

D. Obtain the **Y** weights as

$$\mathbf{c = Y't / t't} \tag{2.2}$$

E. Finally calculate the updated **Y** scores as

$$\mathbf{u = Yc / c'c} \tag{2.3}$$

F. The algorithm is considered to be converged if it is found that the percentage change in the value of score **t** is significantly small ,which can be also written as

Algorithm converged if $\| \mathbf{t}_{old} - \mathbf{t}_{new} \| / \| \mathbf{t}_{new} \| < \theta$ where θ is of the order 10^{-6} or 10^{-8} and the execution moves to the next step **G**. In the case when algorithm is not converged, the execution returns back to step **B** and again continues till convergence is reached.

G. Deflate the values of **X** and **Y** of the current component as

$$\mathbf{p = X't / t't} \quad (2.4)$$

$$\mathbf{X = X - tp' \text{ and}} \quad (2.5)$$

$$\mathbf{Y = Y - tc'} \quad (2.6)$$

and further use these deflated values as X and Y for the next component.

The PLS regression model (generally implemented by NIPALS as discussed above), is developed from a training set of say *M* observations of P X- variables denoted by x_k $(k=1...K)$ and *N* Y-variables y_n $(m=1...N)$. Thus these two training data from the matrix **X** and **Y** of respective dimensions (M x P) and (K x N). The predictions for new observations are calculated from their X –data.

2.3 SIMPLS Regression algorithm

The SIMPLS algorithm too, before its implementation starts with a pre-processing of both the X and Y matrices as mentioned in the implementation of NIPALS algorithm in section 2.2.

The SIMPLS algorithm decomposes the X and Y matrix as below

$$\mathbf{X = TP' + E = \sum t_h p_h + E} \quad (2.8)$$

$$Y = UQ' + F = \sum u_h q_h + F \quad (2.9)$$

Where **T** and **U** are the scores, **P** and **Q** are loadings and **E** and **F** are the residuals of **X** and **Y** respectively.

The steps followed in the algorithm are as below:

A. For each ***h = 1….c*** we calculate

$$A_0 = X'Y \quad (2.10)$$

$$M_0 = X'X \text{ and} \quad (2.11)$$

$$C_0 = I \quad (2.12)$$

B. Calculate q_h the dominant eigenvector of $A'_h A_h$.

C. Calculate the weight matrix **W** by calculating

$$w_h = A_h q_h \quad (2.13)$$

$$C_h = w'_h M_h w_h \quad (2.14)$$

$$w_h = \frac{w_h}{\sqrt{C_h}} \quad (2.15)$$

D. Calculate the following and store the resulting p_h as a column in **P** vector.

$$p_h = M_h w_h \quad (2.16)$$

E. Calculate the following and store the resulting q_h as a column in **Q** vector.

$$q_h = A'h\, w_h \quad (2.17)$$

F. Calculate the values of

$$v_h = p_h, \quad (2.18)$$

$$v_h - v_{h/} \,||\, v_h|| \quad (2.19)$$

G. Lastly, Calculate the matrices for the next iteration as

$$C_{h+1} = C_h\, C_h - v_h\, v'_h \quad (2.20)$$

$$M_{h+1} = M_h - p_h\, p'_h \quad (2.21)$$

$$A_{h+1} = C_h\, A_h \quad (2.22)$$

Thus, the SIMPLS results are computed as

$$T = XW \quad (2.23)$$

$$U = WQ' \quad (2.24)$$

2.4 Multi-State PLS Algorithm- A novel method

In the previous sections, 2.2 and 2.3 we have discussed the standard algorithms NIPALS and SIMPLS which are used to implement the PLS regression. Now we focus on the main objective of this research, which implements a patented novel process prediction algorithm, being developed at Emerson Process Management, Austin, Texas, USA. This algorithm is used in conjunction with NIPALS to calculate the PLS prediction parameters.

2.4.1 Background

The work for this algorithm started in 2010 when at Emerson Process Management; needed to provide an add-on to their existing Neural block of DeltaV DCS system, which was catering to the Batch Analytics applications including process variable prediction and process monitoring solution. This add-on was to provide support for the quality parameter prediction associated with continuous processes. The new add-on allows user to select various prediction algorithm of neural network (NN), multiple linear regression (MLR) or partial least squares regression (PLS). One of the major constraints on the application of NN, MLR, PLS algorithms to the continuous process was that the underlying technology will have to be based on the deviation of the process measurements from their mean values, which was a typical scenario in continuous process. Since an increase in plant production rate or a change in product grade can cause mean values to shift, it was required that the continuous data analytics application for the prediction model building must account for changes in the mean value of the measurements. Thus the main objectives to develop the continuous process analytics were:

1. Automatically modify the mean values of the measurements used in the analytics to compensate for changes in the production rate or product grade.
2. Compensate the deviation value for the time required to transition from different throughputs and product grades.

In general, a continuous process is operated at a constant throughput and occasionally used to make a single product grade. For this case, the mean values associated with the measurements will essentially be nearly constant and the traditional techniques of NN, MLR and PLS is generally sufficient. However, in many cases the throughput of the continuous process needs to be changed frequently in order to meet the required inventory levels or demands. Thus, the frequent change in the required output composition will require changes in all the variables responsible for a particular output composition.

A continuous reactor is a good example of the above-mentioned fact, where, the target for the output composition is changed frequently to allow for the manufacturing of different grades. To shift the output composition, it is often necessary to change the operating point of one or more related process inputs.

The above-mentioned facts and considerations have motivated the development of an algorithm called the "Multi-state PLS based process variable prediction", the details of which will be discussed in the following sections.

2.4.2 State concept and definition

After presenting a brief background about the need and motivation of the algorithm, now we discuss the most important part of the Multi-state PLS based process variable prediction algorithm, which is the concept of States. The state concept utilized in this algorithm is different from the state space analysis, which is used in the modern control theory.

The state as defined in this algorithm refers to a state parameter, which is chosen among the various variables associated with the process. That variable is chosen to correspond to the changes because of a major disturbance to process operation. In other words, such a variable is selected to be the state parameter, which is representative of the underlying process. The parameter, which is selected as the state parameter, is also divided into user defined "states" (maximum of 5 states). These states are actually five equal regions in which the whole operating range of the state parameter is divided.

2.4.3 Implementation of Multi-state PLS Algorithm

The Multi-state PLS algorithm is implemented in two phases:

1. Offline model development phase: This is the phase where historical process data will be processed upon with some statistical techniques and a model will be developed based on the multi-state methods.

2. Online calculation and prediction phase: This is the phase where a new observation will be applied to the model that was previously developed and the response variable prediction will be made based on the value generated by the model.

Steps involved in the Offline model development phase (Input and Output):

A. Depending upon the user selected number of states, the range of variation in the state parameter is calculated and divided into that many equal segments called the "States".

B. For the period of time that the process operated within a particular state, an average value of the state parameter is calculated and stored as a part of the model.

State - Parameter	1	2	3	4	5
State Range	25-35	35-45	45-55	55-65	65-75
No. Sample in Range	210	340	150	85	30
State Parameter-Average	30	40	50	60	70
Process Variable 1	30	40	50	60	70
Process Variable 2	35	45	55	65	75
Process Variable 3	2.5	2.5	2.5	2.5	2.5
Process Variable 4	-1	-1	-1	-1	-1
Process Variable 5	20	28	38	50	65

Table 2.1: Illustration of typical state parameter operating range

Table 2.1 shows a typical illustration of the state parameter and its average calculation when the operating range goes from 25 to 75(Terry et al [43])

C. The input data matrix consisting of all the input variables denoted by **X** matrix is processed and the state parameter range denoted as State-Span is calculated as

$$\textbf{State Span} = \frac{(\textbf{State}_{\textbf{Hi}} - \textbf{State}_{\textbf{Low}})}{\textbf{Number of states}} \qquad (2.25)$$

where State_{Hi} = Highest value of state parameter in data set.

$\text{State}_{\text{Low}}$ = Lowest value of state parameter in data set.

D. The input data matrix is again pass through to calculate the State vector and the Average value of State parameter in each state as below:

For each sample, if the State parameter value is less than the $\textbf{State}_{\textbf{Hi}}$ the state is calculated by formula:

$$\textbf{State} = \textbf{Round}\left\{(\textbf{State\_Parameter}_{\textbf{current}} - \textbf{State}_{\textbf{Low}})/\textbf{State Span} + \textbf{1}\right\} \qquad (2.26)$$

Otherwise the state is calculated as

State = Num_State

where State-parameter $_{current}$ = State parameter value for the current Sample.

Round ( ) = Rounding of the value obtained to the nearest integer. The average matrix is obtained which calculates the average value of state parameter and also each input variable in the input data matrix for each state.

E. The next step is to calculate the mean matrix for each sample and each variable in the input data matrix. The mean for each sample and input variable is calculated based on the value of the state of each sample and the previously stored average values for each state. The following formula shows the mean calculation for each sample of all variables:

The formula for the mean calculation for each sample of input variables depends upon checking two conditions on the state parameter for that sample, and thus is different for both conditions.

Condition 1:

If $\mathbf{State\_Parameter_{current} < Average_{current\_state}}$

where, $State\_Parameter_{current}$ = State parameter value for the current Sample.

And $Average_{current\_state}$ = Average value of the State parameter to which current sample belongs to.

Further, before calculating the mean value we calculate two more statistics, which are **Span** and **Fraction** given as:

$$\mathbf{Span = Average_{current\_state} - Average_{(current\_state-1)}} \quad (2.27)$$

where, $Average_{(current\_state-1)}$ = Average value of state parameter one state less than the current state

$$\mathbf{Fraction} = \frac{\mathbf{State{-}Parameter_{current}{-}Average_{(current\_state-1)}}}{\text{Span}} \quad (2.28)$$

The mean formula for Condition 1 above is given as:

$$\mathbf{Mean_{current-variable}} = \mathbf{Fraction} * (\mathbf{Average_{current\_state}} - \mathbf{Average_{(current\_state-1)}}) + \mathbf{Average_{(current\_state-1)}} \quad (2.29)$$

where $\text{Mean}_{\text{current\_variable}}$ = Calculated mean value for each variable at the current Sample

Condition 2:

Otherwise If $\mathbf{State\_Parameter_{current}} > \mathit{Average}_{\mathbf{current\_state}}$

We again calculate the **span** and **fraction** for condition two given as:

$$\mathbf{Span} = \mathbf{Average_{(current\_state+1)}} - \mathbf{Average_{current\_state}}$$

where, $\text{Average}_{(\text{current\_state}+1)}$ = Average value of state parameter one state more than the current state

$$\mathbf{Fraction} = \frac{\mathbf{State{-}Parameter_{current}{-}Average_{(current\_state)}}}{\text{Span}} \quad (2.30)$$

The mean formula for Condition 2 above is given as:

$$\mathbf{Mean_{current\_variable}} = \mathbf{Fraction} * (\mathbf{Average_{(current\_state+1)}} - \mathbf{Average_{(current\_state)}}) + \mathbf{Average_{(current\_state)}} \quad (2.31)$$

where $\text{Mean}_{\text{current\_variable}}$ = Calculated mean value for each variable at the current Sample

F. For each variable in the data matrix **X,** deviation from mean is calculated for every sample and saved as:

$$\mathbf{Deviation_{variable}} = \mathbf{X_{variable}} - \mathbf{Mean_{variable}} \quad (2.32)$$

where $\text{Deviation}_{\text{variable}}$ = deviation vector for each variable in the input data **X,** X_{variable} = Input vector of each variable in the input data matrix **X,** and $\text{Mean}_{\text{variable}}$ = Mean vector for each variable in the input data matrix **X.**

G. Standard deviation for each variable vector calculated in equation 2.32 is calculated and saved as:

$$\mathbf{Standard-deviation_{variable}} = \mathbf{STDEV}(\mathbf{Deviation_{variable}}) \qquad (2.33)$$

where STDEV = Function that calculates the standard deviation.

H. The normalized data matrix for the input data matrix is then calculated and saved as:

$$\mathbf{X_{Model}} = \frac{\mathbf{Deviation_{variable}}}{\mathbf{Standard-deviation_{variable}}} \qquad (2.34)$$

where X_{Model} = The final data matrix to be used in calculating PLS prediction parameters.

I. Steps **E TO H** are again repeated for the output vector denoted by **Y** and finally the following vector is obtained:

$$\mathbf{Y_{Model}} = \frac{\mathbf{Deviation_{variable}(Y)}}{\mathbf{Standard-deviation_{variable}(Y)}} \qquad (2.35)$$

J. Finally the PLS model parameters are calculated by applying the NIPALS algorithm to X_{Model} and Y_{Model} as:

$$\boldsymbol{\beta}_{\mathbf{PLS}} = \mathbf{NIPALS}\ (\mathbf{X_{Model}}, \mathbf{Y_{Model}})$$

where β_{PLS} = vector of PLS parameters and NIPALS ( ) = NIPALS function applied.

Steps involved in the Online calculation and prediction:

During on-line operation, the mean values of parameters used in continuous analytics will be calculated at each instance in time. The future predicted value of response variable **Y** is calculated using the instantaneous values of X ,PLS parameters obtained during model development, state parameters and values that were determined off-line for the state low and high end of range, state span, and average parameter values for each operating state .

The steps involved in online calculation and response variable prediction are as below:

A. The test data input matrix and output vector are obtained and step **A** to step **I** are applied to both input and output test data in a similar way as done in the offline model development section.
B. The final X matrix and Y vector obtained after normalization are denoted as $\mathbf{X_{Test}}$ and $\mathbf{Y_{Test}}$.
C. The predicted response variable is then obtained as :

$$\mathbf{Y_{Predicted}} = \mathbf{X_{Test}} * \boldsymbol{\beta}_{\mathbf{PLS}}$$

In order to get the predicted response variable in proper units the value obtained earlier is de-normalized by the standard deviation and mean of the response variable that was obtained during the offline model generation as:

$$\mathbf{Y_{Predicted}} = \mathbf{Y_{predicted}} * \mathbf{Standard-deviation(Y)} + \mathbf{Mean(Y)}$$

where Standard − deviation(Y) = standard deviation vector of Y obtained in offline model

Mean(Y) = mean vector of the response variable Y obtained in offline model

Now that we have discussed the implementation of multi-state PLS algorithm, in the next chapter we will be applying this algorithm to the blending tank model we discussed in section 1. Chapter 3 will also discuss the results obtained by applying the MATLAB based implementation of Multi-state PLS to the simulated data for the blending system.

Chapter 3

MATLAB® Graphical User Interface based Implementation of Multistate PLS Algorithm

The Multi-state PLS based data-driven predictive modeling algorithm was discussed in detail in the previous chapter. In this chapter, we now focus on the implementation of the algorithm on the simulation of the blending system discussed earlier using MATLAB® and SIMULINK®. The details of the Blending system can be reviewed in Section 1.4.1.

3.1 The Graphical User Interface

In order to implement the algorithm and make it user interactive, it was decided to employ the MATLAB-based graphical user interface (GUI) software in conjunction with SIMULINK. Figure 3.1 shows the layout of the GUI that was developed. The GUI has been divided into two major parts, the data generation parameters and the PLS prediction parameters each marked by 1 and 2 respectively on Figure3.1. In order to explain the function of each button in the GUI, these parameters have been further divided into sub parts from **A to J** .We now discuss the function of each part of the GUI.

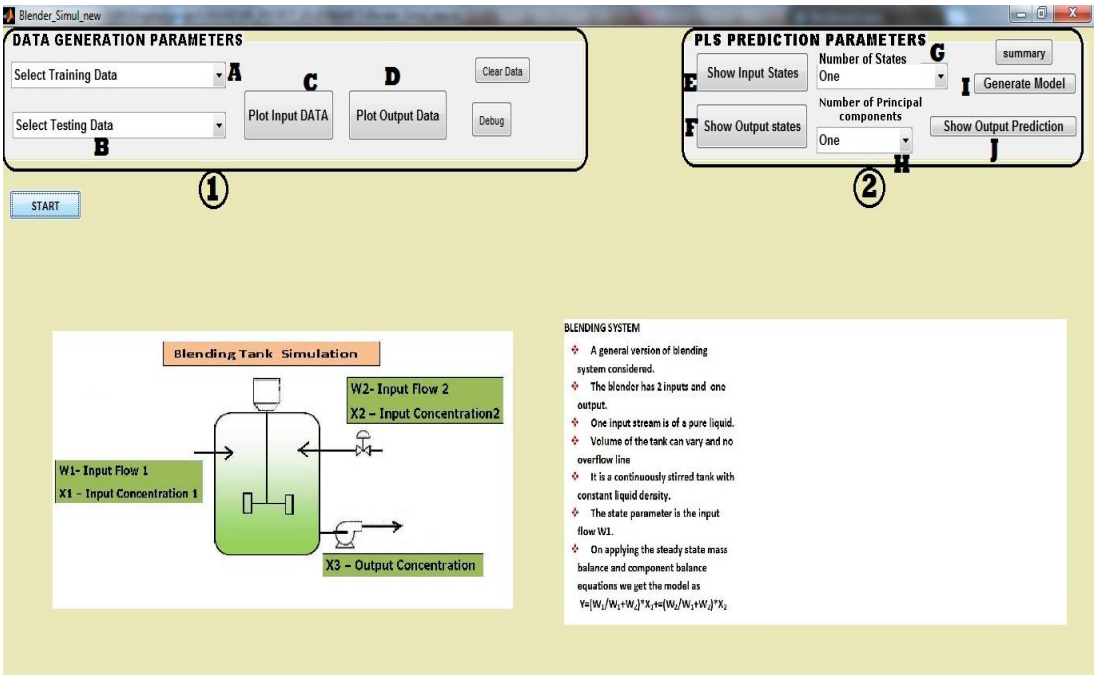


Figure 3.1. The GUI used for implementing the algorithm

1. **Data generation parameters (denoted by ①in Figure 3.1)-** This part of the GUI is used by the user to generate and at the same time view the input data which are used for model building and model testing
   - **Region 'A'**- This drop down menu is used to generate the simulated data denoted by **training data,** for the four inputs to the blending tank namely; input flows **W1**, **W2** and input fluid concentrations **X1, X2** and is used to train the Multi-state PLS model. Figure 3.2 shows the drop down menu**.**

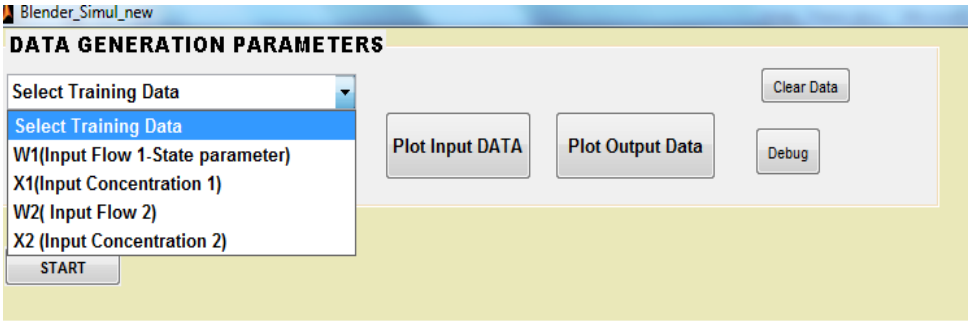


Figure 3.2. Data generation options for training data

- **Region 'B'**- This drop down menu is used to generate the simulated data denoted by **testing data,** for the four inputs to the blending tank namely; input flows **W1**, **W2** and input fluid concentrations **X1, X2** and is used to test the performance of the multi-state PLS model generated from the training data.

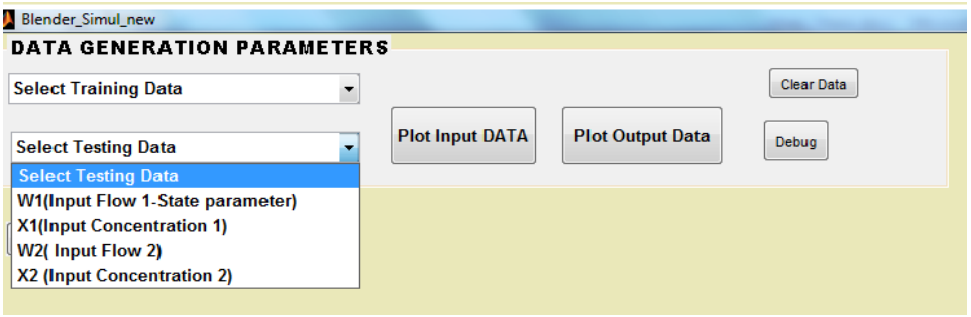


Figure 3.3. Data generation options for testing data

- **Region 'C'**- When a proper selection in region A and B has been made, this button when pressed generates both the training and testing data to be used in generating the model and plots the data, which has been generated for the user to view. The user can view all the four inputs that

have been generated for the model generation with the repeated use of this button

- **Region 'D'**- Once the input data has been generated, the output data is obtained by reading input data in a SIMULINK model , and calculating a steady state or dynamic model .The details of the SIMULINK model can be seen in Figure 3.12

2. **PLS prediction parameters (denoted by ② in Figure 3.1)-** This part of the GUI is used by the user to set the parameters used by the multistate-PLS model.
   - **Region 'E'**- The model building process starts with this button , which when pressed calculates the states of the user defined state variable (input flow **W1** in this simulation) . This button also displays the data points in the state variable as assigned to various states.
   - **Region 'F'**- This button repeats the process of calculating the states for the output variable and also displays the data points as assigned to given number of states, which is defined by the user in the drop down selection **'G'.**
   - **Region 'G' and 'H'**- These drop down selections prompt the user to select the number of states and the number of principal components to be used in the PLS algorithm .
   - **Region 'I' –** Once all the parameters in the PLS model have been selected the button "Generate Model" generates the multi–state model for the training data earlier selected by the user.
   - **Region 'J'-** Once the user has generated the model, this button denoted by "Show output prediction" applies the testing data to the model and calculates the predicted output variable as generated by the PLS model. The user can also see the plot of the predicted output variable and the actual output variable in the training data. Further, the root mean square error

(RMSE) is calculated between the predicted and actual output variable, which gives the measure of the efficiency of the Multi-state PLS model.

3.1 The Implementation

Previous section described in detail the outlook of the graphical user interface and its various parts. We now present the various steps graphically, that the user will take to test the performance of the multi-state PLS algorithm implemented by our GUI.

Step 1. The user selects all the inputs from the drop down menus and generates their data. The selection of W1 (state variable) gives the user an additional option of selecting the excitation mode for the state variable .User can select from these from options namely;

a. Positive step change- The state variable W_1 is generated with a positive step change from a lower average value of 110 to an upper average value of 160 kg/min with added noise.
b. Negative step change- The state variable W_1 is generated with a negative step change and varies from an upper average value of 160 to a lower average value of 110 kg/min with added noise.
c. Random change- In this option the state variable W_1 is changed randomly with added noise.
d. Integrated moving average- This option changes the state variable W_1 in accordance to the integrated moving average equation given by:

$\mathbf{X_{t+1=}\ X_t + \varepsilon_t}$ where ε_t is random noise with mean 0 and standard deviation σ^2

The above-mentioned options are displayed in Figure **3.4** to **3.7.**

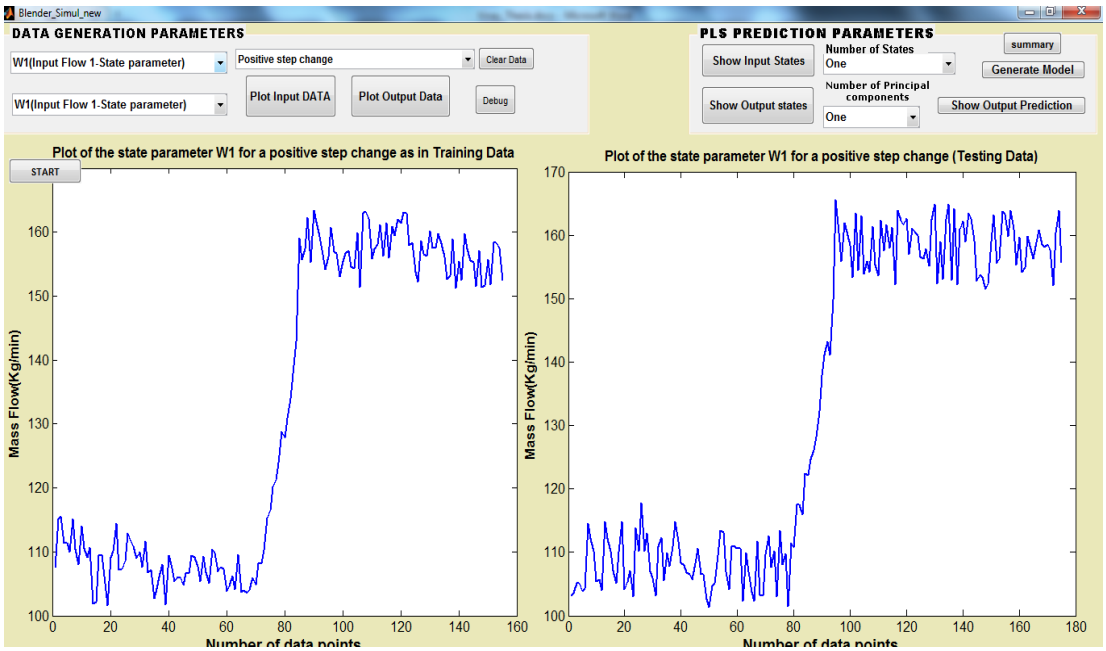


Figure 3.4. State variable W_1 generated with positive step change

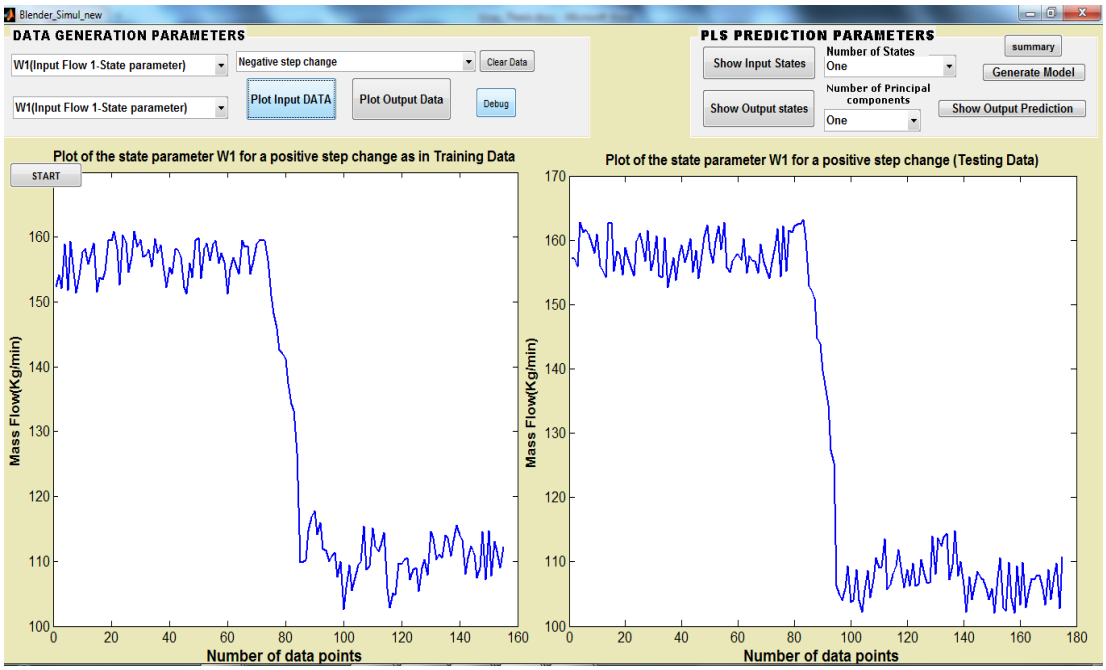


Figure 3.5. State variable W_1 generated with negative step change

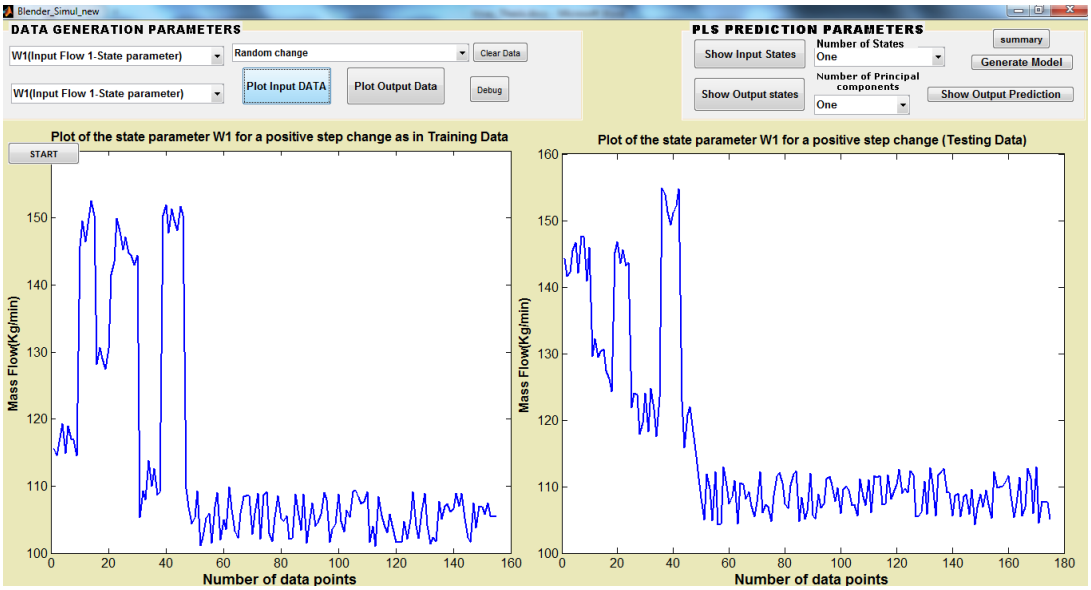


Figure 3.6. State variable W_1 generated with random change

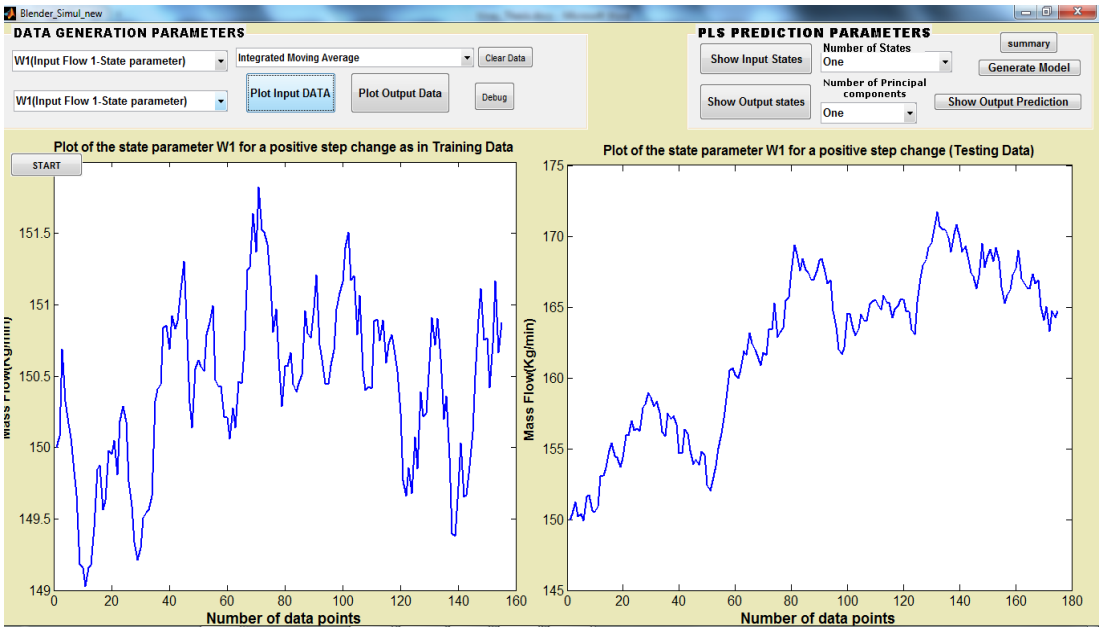


Figure 3.7. State variable W_1 generated with integrated moving average

Step 2. Similar to step 1, the user selects the other three inputs and generates the training and testing data for the model building. Figures **3.8** to **3.10** depict pictorially the steps taken by the user to generate the other three input data sets.

The data for X_1, W_2 and X_2 are generated as below.

X_1 Data – The input mass concentration X_1 is generated with an average value of 0.16 and some noise added to it.

W_2 Data- The second fluid input to the Blending tank W_2 is generated with an option of ratio control, due to which it follows the other input flow W_1 in a similar fashion with a scaled down version.

X2 Data-The mass concentration of the second input fluid is generated with an average value of 0.93 with added noise.

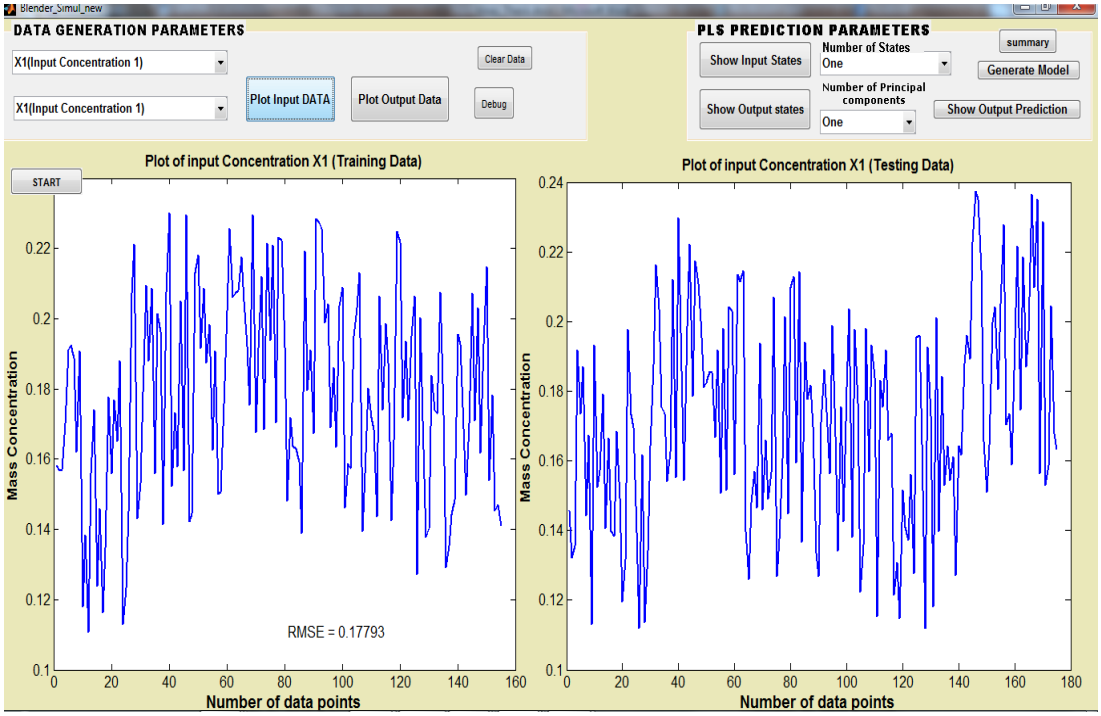


Figure 3.8. Input fluid X_1 with its generated data for training and testing data

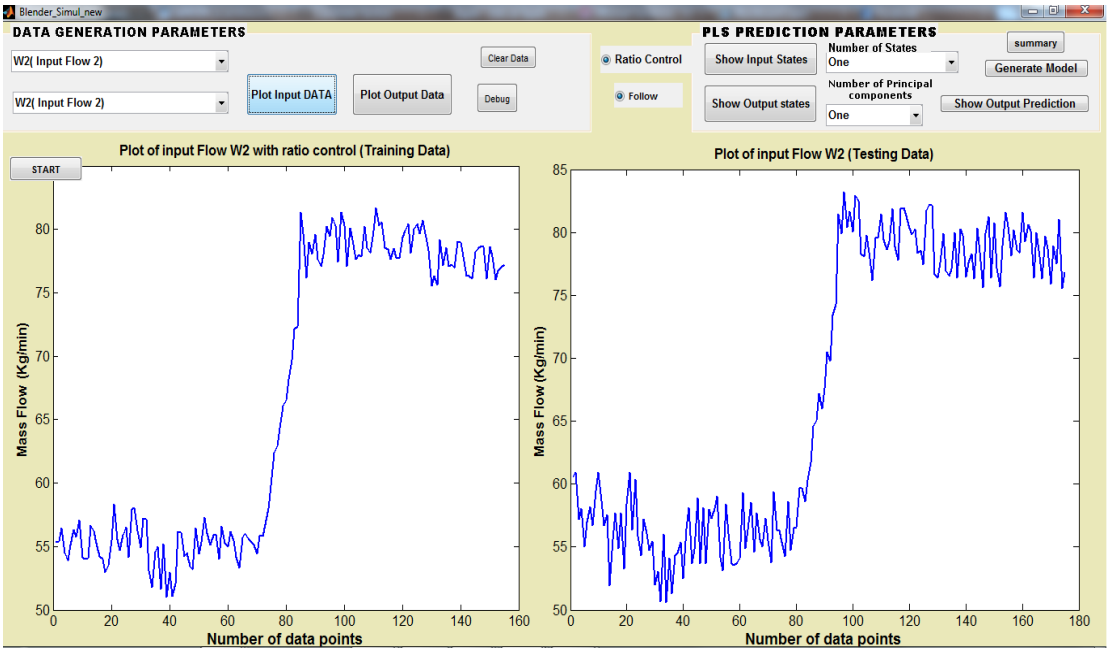


Figure 3.9. Input fluid W_2 with its generated data for training and testing data in ratio control mode

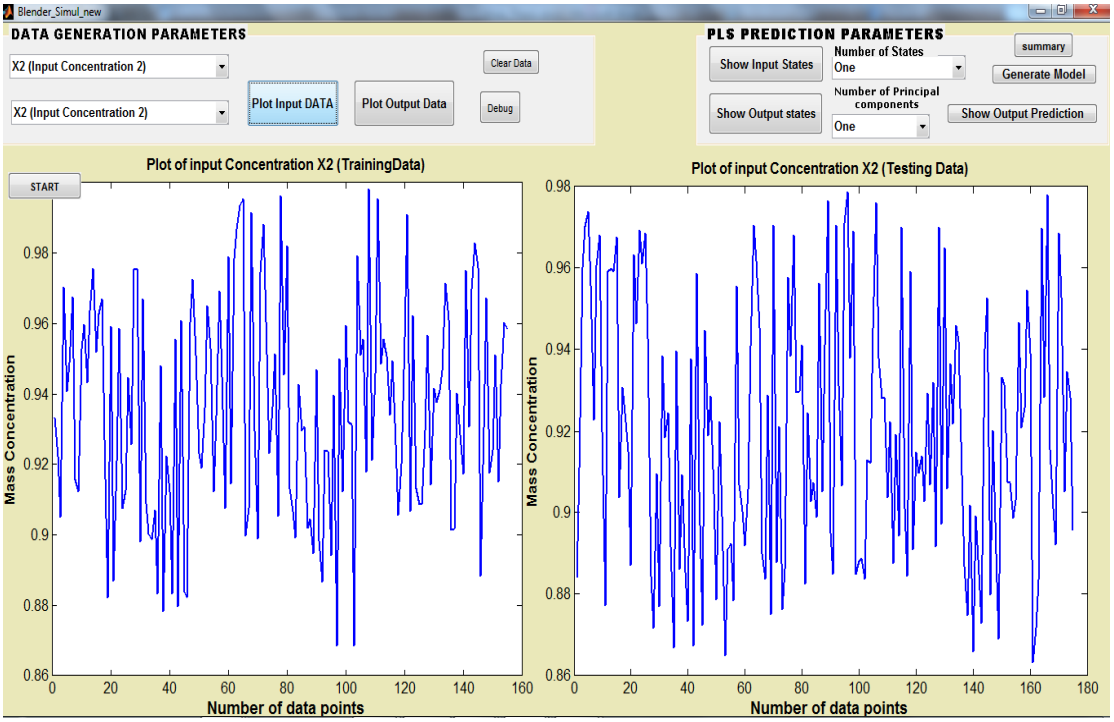


Figure 3.10. Input fluid X_2 with its generated data for training and testing data

Step 3. Once the user has selected and generated the input data, the output data can be generated by pressing the button Plot output data. Figure **3.11** depicts the output pictorially. The output data is generated with a SIMULINK model as shown in Figure **3.12.**

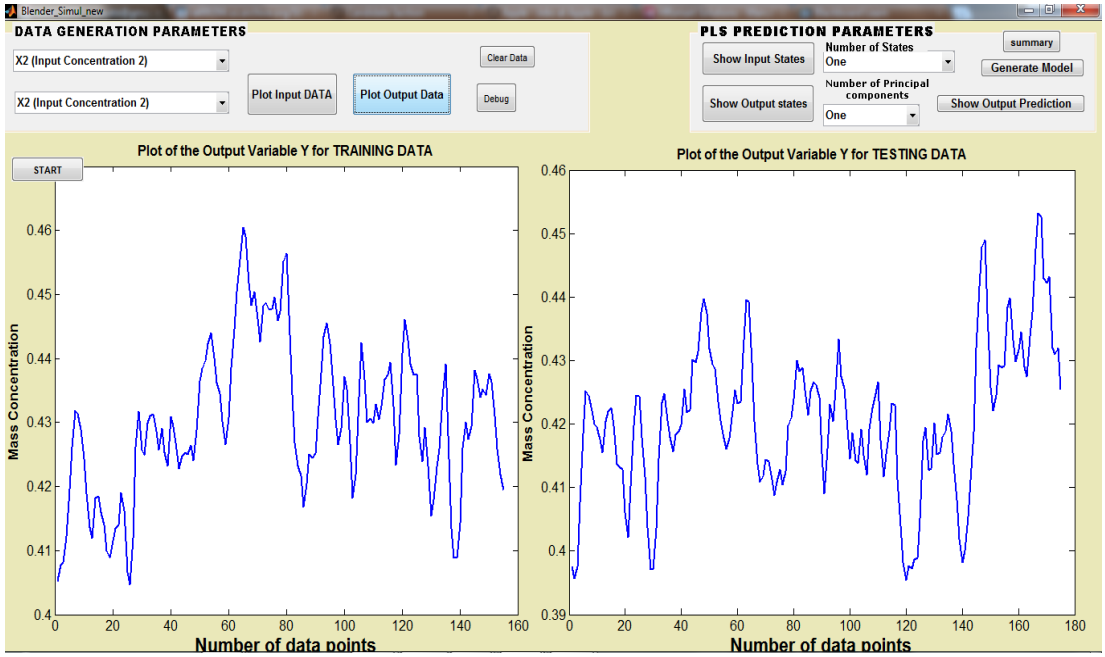


Figure 3.11. Output data Mass concentration for training and testing data

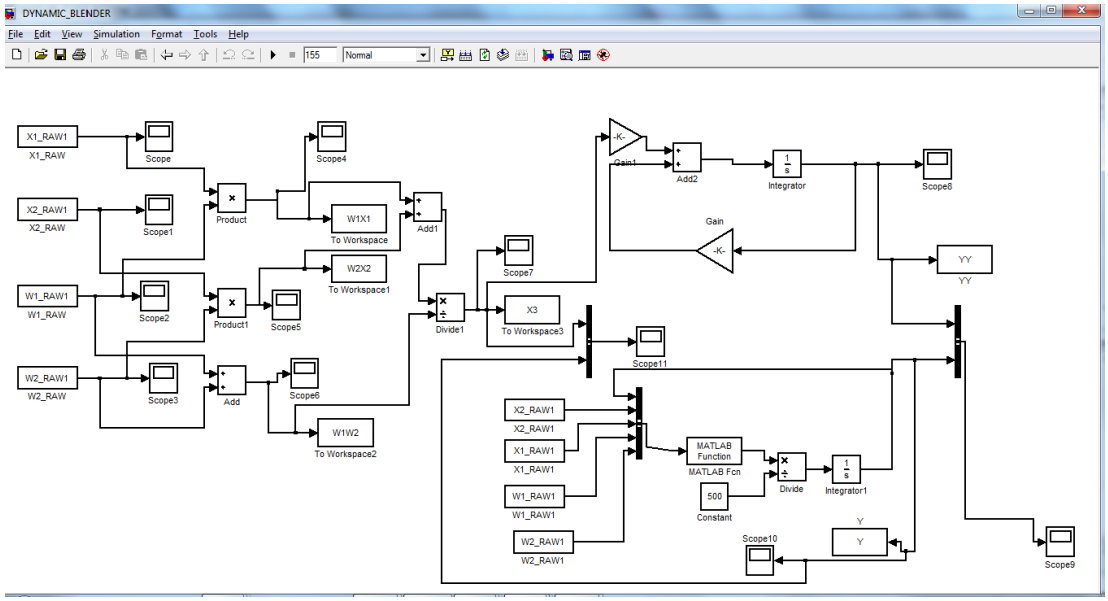


Figure 3.12. SIMULINK model for the output data generation

Step 4. In the next step user sets the parameters for the PLS prediction algorithm and by pressing the buttons “Show input state” and “Show output state” the state vector is obtained both for state variable and the output variable .Figure **3.13** and **3.14** show how the state variable and output variables have been divided into states when the user selects four states and two principal components.

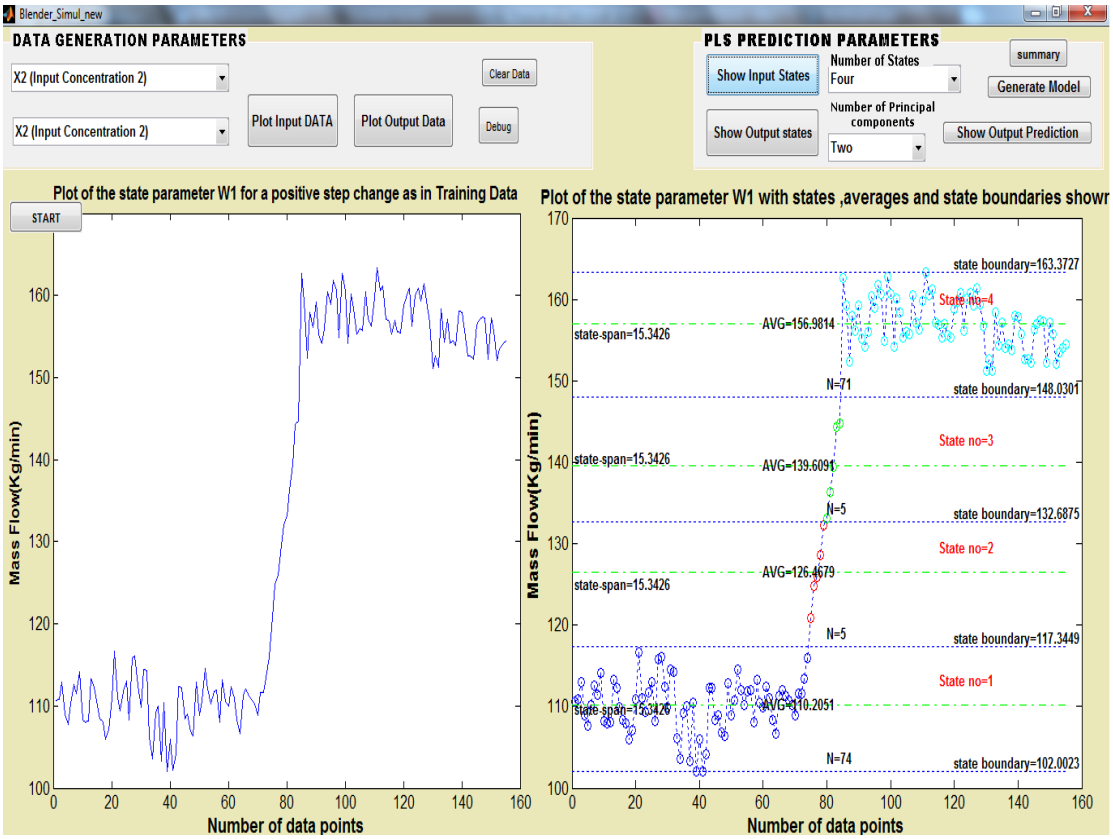


Figure 3.13. State variable W_1 divided into four states

Step 5. The next step that the user performs is to click the button "Generate Model ", which builds the multi-state PLS model based on the inputs selected by the user and the resulting output generated.

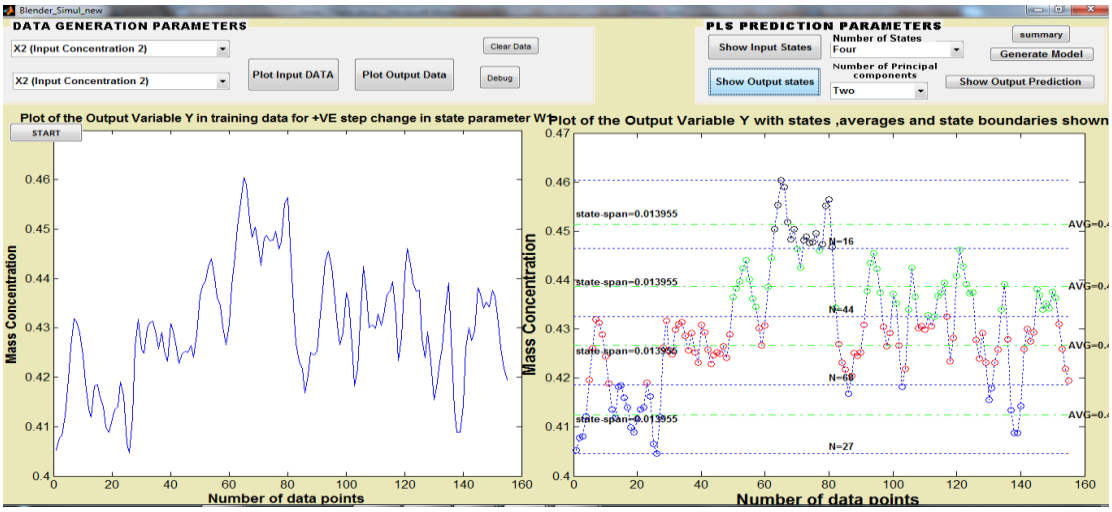


Figure 3.14. Response variable Y divided into 4 states

Step 6. This is the step where the user obtains the prediction result for the testing data based on the model generated for training data by pressing button " Show output prediction " . The result is the plotting of predicted and actual output for the response variable showing the RMSE value between the predicted and actual output. Figure **3.15** depicts the prediction results.

Step 7. With this final step user can summarize the whole result in terms of RMSE by pressing the button "Summary", which shows the tabulated information about the simulation carried out. Figure 3.16 shows the summary table generated for the results for the PLS model generated.

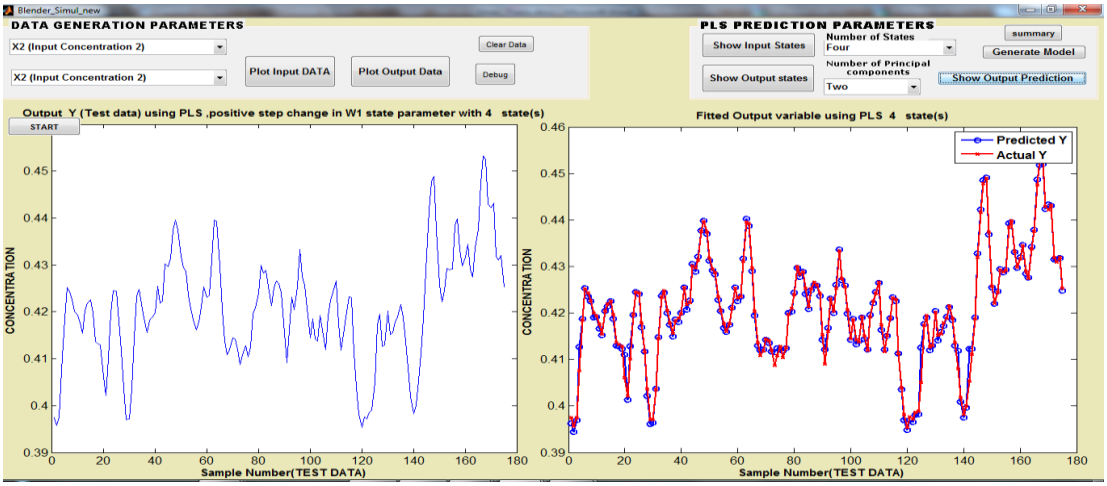
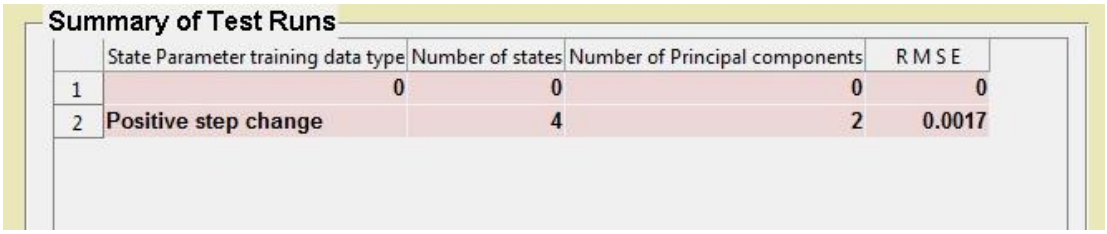


Figure 3.15. Response variable Y showing prediction and actual vales



Summary of Test Runs

	State Parameter training data type	Number of states	Number of Principal components	R M S E
1	0	0	0	0
2	Positive step change	4	2	0.0017

Figure 3.16. Summary of the results obtained after PLS prediction results

Figure 3.15 clearly shows the very accurate prediction of the test data for the output variable as applied to the testing data set earlier discussed. Referring to the Figure **3.16** indicates that t he RMSE for this data set is calculated to be 0.0017, which is a reasonable assumption.

The user can carry these above seven steps out and a very good prediction result can be obtained by using PLS model. The implementation can be repeated and again the prediction results can be obtained for a different training and testing data for input and output variable.

Chapter 4

Conclusions and Future Work

With the increased focus on data-driven techniques and their more and more usage in the applications of manufacturing and process control, this thesis and research work adds a new perspective into the above-mentioned techniques. A novel multi-state PLS based process variable prediction algorithm was presented with a detailed literature review of the various existing data-driven statistical process control techniques.

Further, the algorithm was implemented on a simulation framework for a simple blending system and the algorithm was found to be working impressively good on the data set that was simulated for both steady-state and dynamic system. The Root Mean Square Error (RMSE) between the predicted value (from the algorithm) and the actual measured value for the output concentration of the simulated blending tank was found to be on an average of the order of 0.0010, depicting the effectiveness of the multi-state PLS based data-driven algorithm. Further to add, it was concluded that the algorithm performed very well even in the conditions where the testing data was totally different from the training data and even where the testing data was out of the range of the training data thus indicating the robust nature of this novel algorithm

This research also introduced a unique way of performing the simulations with the usage of MATLAB® Graphical Unser Interface (GUI). The use of GUI enhanced the flexibility and intuitiveness of the algorithm implementation by providing the user an option to select four different excitation modes for the state parameter as mentioned in the algorithm. In addition to displaying the plots of the data sets simulated for both inputs and outputs the GUI also implemented the algorithm by

creating the multi-state PLS model in the background and showing the prediction results. In addition, the option was provided in the GUI to archive the results for different iterations so that a comparison can be done to infer new insights.

The future work related to this thesis and research project includes the implementing and testing of multi-state PLS algorithm on an industrial data in addition to the previously done test on a simulated blending system.

The testing with an industrial data is expected to give some additional insights into the working of the algorithm, which can be used to further make some modifications in the implementation.

References

[1].Arteaga .Francisco; Ferrer .Alberto,(2002), "Dealing with missing data in MSPC: several methods, different interpretations, some examples". Journal of Chemometrics Volume 16, Issue 8-10, Pages 408-418.

[2].Badcock, Julie; Jonathan, Philip; Bailey, Trevor C.; Krzanowski, Wojtek J,(2004) "Two Projection Methods for Use in the Analysis of Multivariate Process Data With an Illustration in Petrochemical Production", Technometrics, Vol. 46, No. 4, November 2004, Pages 392-403.

[3].Bakshi, R, Bhavik.(1998), "Multiscale PCA with Application to Multivariate Statistical Process Monitoring", AIChE Journal Volume 44, Issue 7, Pages 1596–1610.

[4].Bersimis S; Psarakis S. and Panaretos J.,(2007), Multivariate Statistical Process Control Charts: An Overview, International Quality and Reliability Engineering, volume 23,Issue 5, Pages 517–543.

[5].Bharati,M;MacGregor,J,F;Champagne,M,(2004)," Using near infra-red multivariate image analysis technique to predict pulp properties,TAPPI Journal 3(5), Pages 8-14

[6].Burnham,J,Alison;MacGregor,F,John;Viveros,Roman,(1999),"Latent variable multivariate regression modeling" Chemometrics and Intelligent Laboratory Systems Volume 48,Issue 2, Pages 167–180.

[7].Chen Junghui and Liu Kun-Chih,(2002), "On-line batch process monitoring using dynamic PCA and dynamic PLS models" Chemical Engineering Science Volume 57, Issue 1, Pages 63-75.

[8].Chen Q;Kruger U; Meronk M., Leung A.Y.T,(2004), "Synthesis of T2 and Q statistics for process monitoring ", Control Engineering Practice 12, Pages 745–755.

[9].Cherry, G, A., Qin, S.J.,(2006),"Multiblock principal component analysis based on a combined index for semiconductor fault detection and diagnosis". IEEE Transactions on Semiconductor Manufacturing, Volume 19, Issue 2,Pages 159 - 172

[10]. Chiang Leo H; Russell Evan L; Braatz Richard D,(2000)"Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis", Chemometrics and Intelligent Laboratory Systems, Volume 50, Issue 2,, Pages 243-252.

[11]. Cho Ji-Hoon; Lee Jong-Min; Choi Wook Sang; Lee Dongkwon and Lee In-Beum ,(2005) "Fault identification for process monitoring using kernel principal component analysis" Chemical Engineering Science Volume 60 ,Issue 1 Pages 279-28.

[12]. Choi, S.W., Lee, C., Lee, J.M., Park, J.H., Lee, I.B., (2004). "Fault detection and identification of nonlinear processes based on kernel PCA". Chemometrics and Intelligent Laboratory Systems, Pages 341–350.

[13]. De Jong, S,(1993), "SIMPLS: An alternative approach to partial least squares regression",Chemometrics and Intelligent Laboratory Systems Volume 18, Issue 3, Pages 251-263

[14]. Dong .D. and McAvoy .J.T.(1996), "Non-linear principal component analysis based on principal curves and neural networks," Computer. Chem. Eng., vol. 20, no. l, Pages 65–78.

[15]. Ergon Rolf,(2004),"Informative PLS score-loading plots for process understanding and monitoring", Journal of Process Control 14, Pages 889–897.

[16]. Fujiwara Koichi; Kano Manabu; Hasebe Shinji and Takinami Akitoshi,(2009), "Soft-Sensor Development Using Correlation Based Just-in-Time Modeling", AIChE Journal Volume 55, Issue 7.Pages 1754–1765,

[17]. Geladi Paul and Kowalski Bruce R., (1986), "Partial least-squares regression: a tutorial", Analytica Chimica Acta Volume 185, 1986, Pages 1-17.

[18]. Hoskuldsson Agnar,(1988),"PLS Regression Methods", Journal of Chemometrics Volume 2, Issue 3, Pages 211–228.

[19]. Yu Honglu, MacGregor JF, "Monitoring flames in an industrial boiler using multivariate image analysis, AIChE Journal 50, Pages 1474-1483.

[20]. Kadlec Petr, Grbic Ratko, Gabrys Bogdan,(2011), "Review of adaptation mechanisms for data-driven soft sensors", Computers and Chemical Engineering 35, Pages 1–24 .

[21]. Kaneko Hiromasa; Arakawa Masamoto and Funatsu Kimito, (2010) "Novel soft sensor method for detecting completion of transition in industrial polymer processes", Computers & Chemical Engineering,(Article in press).

[22]. Kano Manabu; Hasebe Shinji, Hashimoto Iori,(2001) ,Hiromu Ohno, "A new multivariate statistical process monitoring method using principal component analysis", Computers and Chemical Engineering 25, Pages 1103–1113.

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

[24]. Kourti Theodora,(2005), “Application of Latent variable methods to process control and multivariate statistical process control in the industry”, International Journal of Adaptive control and signal processing, 19, Pages 213-246.

[25]. Kourti Theodora, Macgregor John (1995), “Process analysis, monitoring and diagnosis using multivariate projection methods”, Chemometrics and Intelligent Laboratory Systems 28, Pages 3-21.

[26]. Kohonen Jarno; Reinikainen, Satu-Pia;Aaljoki Kari, Perkiö Annikki; Väänänen Taito; Höskuldsson Agnar,(2007), “Multi-block methods in multivariate process control”, Journal of Chemometrics Special Issue: Conferentia Chemometrica 2007 Volume 22, Issue 3-4, Pages 281–287.

[27]. Lane S; Martin E. B; Kooijmans R. and Morris A. J,(2001),“Performance monitoring of a multi-product semi-batch process, Journal of Process Control 11, Pages 1-11.

[28]. Lee, J.-M., Yoo, C.K., Choi, S.W., Vanrolleghem, P.A., Lee, I.-B.,(2004). “Nonlinear process monitoring using kernel principal component analysis”. Chemical Engineering Science 59, Pages 223–234.

[29]. Li Gang; Qin Joe .S; Ji Yindong; Zhou .Donghua,(2010), “Reconstruction based fault prognosis for continuous processes”, Control Engineering Practice 18, Pages 1211–1219.

[30]. Li Weihua; Yue H. Henry; Valle-Cervantes Sergio; Qin S. Joe,(2000) “Recursive PCA for adaptive process monitoring ”,Journal of Process Control 10, Pages 471-486.

[31]. Liu J. Jay, MacGregor F John,(2008) “Froth-based modeling and control of flotation processes”, Minerals Engineering 21, Pages 642–65.

[32]. MacGregor, J. F., C. Jaeckle, C. Kiparissides, and M. Koutoudi, (1994) "Process Monitoring and Diagnosis by Multiblock PLS Methods," AIChE J., 40(5), Pages 826-838.

[33]. Misra Manish; H. Yue Henry; Qin S. Joe; Ling Cheng,(2002), "Multivariate process monitoring and fault diagnosis by multi-scale PCA", Computers and Chemical Engineering 26 (2002), Pages 1281–1293.

[34]. Negiz,A;Cinar,A, (1997) ,Statistical monitoring of multivariable continuous processes with state-space models, AIChE Journal 43 (8), Pages 2002–2020.

[35]. Norvilas, Aras; Negiz, Antoine; DeCicco, Jeffrey and Çinar Ali, (2000), "Intelligent process monitoring by interfacing knowledge-based systems and multivariate statistical monitoring". Journal of Process Control Volume 10, Issue 4, Pages 341-350.

[36]. Perk Sinem and Cinar Ali,(2007), "Agent based monitoring, fault detection ,diagnosis, and control of spatially distributed processes."8TH International IFAC symposium on Dynamics and control of process systems Vol 1.Pages 285-290

[37]. Piovoso M.J; Hoo Karlene.A, "Control Systems Applications".

[38]. Ramaker .Henk-Jan; van Sprang . N. M Eric.; Westerhuis A .Johan and Smilde K Age; (2004),"The effect of the size of the training set and number of principal components on the false alarm rate in statistical process monitoring",Chemometrics and Intelligent Laboratory Systems Volume 73, Issue 2, Pages 181-187.

[39]. Reinikainen Satu-Pia; Höskuldsson Agnar,(2003), "COVPROC method: strategy in modeling dynamic systems ", Journal of Chemometrics Volume 17, Issue 2, Pages 130–139.

[40]. Schölkopf, Bernhard; Smola, Alexander and Müller, Klaus-Robert,(1998), "Kernel principal component analysis", Lecture Notes in Computer Science, , Volume 1327, Pages 583-588.

[41]. Seborg, Edgar, Mellichamp,"Process Dynamics and control, ,2nd edition, Wiley Publications).

[42]. Sundarraman Anandakrishnan; Srinivasan Rajagopalan, (2003) ,"Monitoring transitions in chemical plants using enhanced trend analysis ", Computers and Chemical Engineering 27, Pages 1455-1472.

[43]. Terry Blevins; Willy Wojsznis (2012) ,"Continuous Data Analytics – Concept Document", Emerson Internal Document.

[44]. Treasure Richard J; Kruger Uwe and Jonathan E.Coopers,(2004), "Dynamic multivariate statistical process control using subspace identification", Journal of Process Control Volume, Pages 279-292.

[45]. Tsung Fugee; Wang Kaibo, (2010) "Adaptive Charting Techniques: Literature Review and Extensions", FRONTIERS IN STATISTICAL QUALITY CONTROL 92010, Part 1, Pages 19-35.

[46]. Venkatasubramanian Venkat; Rengaswamy Raghunathan; Yin Kewen; Kavuri Surya N(2003), "A review of process fault detection and diagnosis Part I: Quantitative model-based methods", Computers and Chemical Engineering 27, Pages 293-311

[47]. Wang Jin and Qin Joe S, "A new subspace identification approach based on principal component analysis" (2002), Journal of Process Control, Volume 12, Issue 8, December 2002, Pages 841-855.

[48]. Wold, S; Esbensen, K., and Geladi, P.(1987). "Principal component analysis", Chemometrics and Intelligent Laboratory Systems, 2, Pages 37–52

[49]. Wold, S; Sjöström, M., and Eriksson, L.(2001). "PLS Regression: A basic tool of chemometrics", Chemometrics and Intelligent Laboratory Systems, volume 58 Issue 2, Pages 109–130.

[50]. Wold, S; Ruhe, A;Wold, H ;Dunn III, W.J, "The collinearity problem in linear regression, The partial least squares approach to generalized inverses", SIAM J. Sci. Stat. Comput. 5" (1984) 735–743

[51]. Xia Luyue ;Pan Haitian,(2010), "Improved multi-scale principal components analysis with applications to process monitoring", International Conference on Intelligent Control and Information Processing (ICICIP), 2010 , Pages 222 - 226 .

[52]. Yoon S, MacGregor JF. Principal-component analysis of multiscale data for Process monitoring and fault diagnosis. AICHE Journal 2004; 50: Pages 2891– 2903.

[53]. Yu Honglu and MacGregor F John; Haarsma Gabe and Bourg Wilfred (2003), "Digital Imaging for Online Monitoring and Control of Industrial Snack Food Processes", *Ind. Eng. Chem. Res.*, 42 (13), Pages 3036–3044.

[54]. Zhang J; Martin E.B U; Morris A.J, (1997) "Process monitoring using non-linear statistical techniques" Chemical Engineering Journal 67, Pages 181–189 .

[55]. Zhang Lin; Garcia-Munoz Salvador,(2009), "A comparison of different methods to estimate prediction uncertainty using Partial Least Squares (PLS): A practitioner's perspective, Chemometrics and Intelligent Laboratory Systems 97, Pages 152–158.

[56]. Zhiqiang Ge and Song Zhihuan, (2009), Two-level multiblock statistical monitoring for plant-wide processes, Korean J. Chem. Eng., 26(6), Pages 1467-1475.

Vita

Vinay Kumar was born in the capital city of India-New Delhi. He has had his upbringing across multiple cities in India namely Bangalore, Jabalpur and Bhopal. He completed his Bachelor's in Electronics & Communication Engineering from Oriental Institute of Science and Technology, Bhopal affiliated to Rajiv Gandhi Technical University the central University of the state of Madhya Pradesh, India. After the completion of his Bachelor's degree, he worked briefly for 1 year in Business Development for HCL Infosystems Ltd, and then joined National Aluminium Company, India as a Graduate Engineer Trainee in the Automation and IT division. After an achievement driven 7-year period, He left NALCO as Assistant Manager to pursue higher studies and joined The University of Texas at Austin as a Master's student in Electrical & Computer Engineering Department in fall 2010.

Email: vnayk7@gmail.com

www.webspace.utexas.edu/vk3325

This thesis was typed by the author.