Application of Multivariate Statistics to Optimizing Polyolefin Manufacturing

Niket Sharma and Y.A. Liu
Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061, U.S.A.

Abstract

In this chapter, we delve into the sophisticated realm of multivariate statistical methods, focusing on Principal Component Analysis (PCA) and Projection to Latent Structures (PLS), as pivotal tools for unraveling the complexity of process data analytics. By anchoring these statistical techniques within the framework of polyethylene manufacturing processes, we aim to illuminate their exceptional utility and novelty in addressing the multifaceted challenges inherent in process optimization and quality control.

The discourse begins by introducing PCA, not merely as a statistical tool, but as a fundamental cornerstone for the analytical examination of process variables. Through a meticulously designed workshop, we demonstrate the application of PCA in dissecting the intricate web of variables influencing the quality and conversion rates of Low-Density Polyethylene (LDPE) production in a two-zone tubular reactor. The integration of Aspen ProMV as a practical tool for PCA applications exemplifies the seamless bridge between statistical theory and industrial application, emphasizing the method's accessibility and relevance to both academia and industry.

Transitioning to PLS, the chapter articulates its differentiation from PCA by its ability to simultaneously handle datasets comprising both process variables (X) and product quality variables (Y), offering a holistic view of the manufacturing process. Through pragmatic workshops, we showcase PLS's robustness in application to challenges such as melt index prediction and causal analysis in High-Density Polyethylene (HDPE) manufacturing, underscoring its adaptability to complex industrial datasets, including those with measurement time lags.

The exploration extends to the nuanced application of these multivariate statistical methods to batch polymer processes. Here, we introduce a novel batch-wise unfolding approach via multiway PCA and PLS, expanding the frontier of statistical applications in process data analytics.

This chapter transcends the conventional boundaries of statistical applications, highlighting the transformative impact of PCA and PLS in the domain of process data analytics. It aspires to foster a deeper understanding and appreciation of these statistical methods, encouraging their broader adoption and adaptation in optimizing manufacturing processes and enhancing product quality. This contribution not only reaffirms the critical role of advanced statistical techniques in the scientific community but also underscores their practical significance in improving industrial operations and outcomes.

This is a preprint version of a chapter from our book - *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing. Please cite the original work if referenced [31,32].*

9.1 Introduction to Principal Component Analysis (PCA)

This Chapter 9 focuses on the use of multivariate statistics. Sections 9.1 introduces an important multivariate statistics tool in process data analytics, namely, *principal component analysis (PCA).* Section

9.2 presents a hands-on workshop on the application of PCA for analyzing the process variables that affect quality and conversion of LDPE product from a two-zone tubular reactor. We introduce the use of the software tool, Aspen ProMV, for multivariate statistics applications, available to universities at low cost. Section 9.3 introduces *the projection to latent structures or partial least squares (PLS)*. A key difference between PCA and PLS is that PCA involves only datasets of process variables (X) or deals with the X-space; PLS involves datasets of both process variables (X) and product quality variables (Y), or deals with both X-space and Y-space. Section 9.4 presents two hands-on workshops of applying PLS to the LDPE problem of Section 9.2, and to the melt index prediction and causal analysis of a HDPE manufacturing process. Section 9.5 introduces PLS for process data analytics with measurement time lags and includes a hands-on workshop of PLS for a HDPE process for the melt index prediction and causal analysis, including the effect of time lags on melt index measurements. Section 9.6 covers the process data analytics for batch polymer processes and presents a hands-on workshop to demonstrate the multiway PCA and PLS methodology, particularly the batch-wise unfolding approach, for data analytics. Section 9.7 gives the bibliography and suggested further reading

Beginning in late 1980 to early 1990, chemical engineers have been paying an increasing attention to the emerging topics of artificial intelligence, neural computing, multivariate statistics, machine learning and big data analytics, and their applications to bioprocessing and chemical engineering [1 to 5]. MacGregor and others have demonstrated the significant applications of multivariate statistics and big data analytics to optimizing the manufacturing of LDPE, HDPE, Nylon 6 and other polymers [6 to 10]. Multivariate statistical analysis [11 to 13] and its implementation using languages such as Python, R or software such as Aspen ProMV, SAS, JMP, etc. find a growing number of applications to polymer manufacturing. These include: (1) data quality deviation analysis; (2) unit yield analysis; (3) production capacity degradation analysis; (4) offline production optimization (discovery and optimization of key variables); (5) online process monitoring and troubleshooting; and (6) batch process variable analysis.

This section introduces the principal component analysis (PCA), following the multivariate statistical analysis textbooks of Johnson and Wichern [11], and Rencher and Christensen [12], and the excellent online book of Dunn [13], which is continually updated. The online reader is allowed "to freely download, share, adapt, commercialize and attribute" some of the book materials, as long as the reader acknowledges that "Portions of this work are the copyright of Kevin Dunn". That is exactly what we wish to acknowledge here, as we shall use some of the explanations and figures from reference [13] below.

Both textbooks [11,12] include a chapter of matrix algebra relevant to multivariate statistical analysis. Therefore, we have included an *Appendix A, Matrix Algebra in Multivariate Data Analysis and Model-Predictive Control* Using *MATLAB and Python* at the end of this book. This appendix also includes the basic implementation of the relevant matrix operations and principal component analysis in both MATLAB and in Python.

9.1.1 Introduction to Principal Components

We follow [14] to illustrate the concept of principal components. Figure 9.1 shows a 3D image of some process data. When projecting the same data onto a 2D plane in Figure 9.2, we are unable to observe the same 3D relationship. However, we can observe sufficient characteristics of the original 3D image in two dimensions if we can identify two linear combinations of process variables x, y and z in order to capture most of the variations in these three process variables. See Figure 9.3.

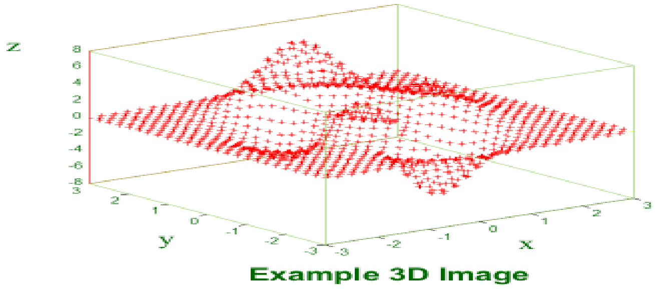


Figure 9.1 The original 3D image of process data. Used with perssion from Aspen Technology, Inc.

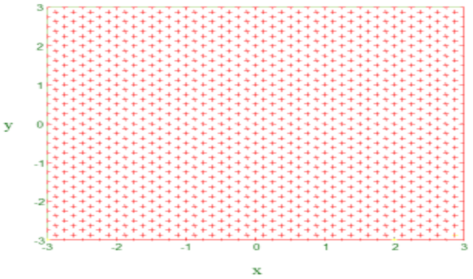


Figure 9.2 Losing the characteristics of the original 3D image when projecting onto the x-y plane. Used with perssion from Aspen Technology, Inc.

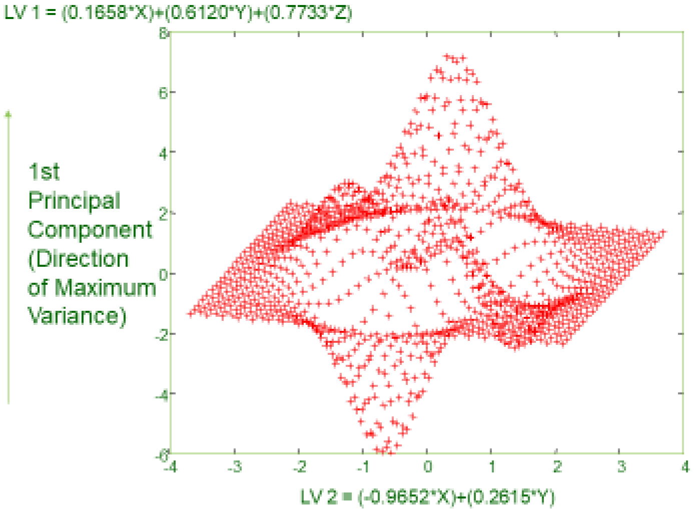


Figure 9.3 Retaining the characteristics of the original 3D image when displaying on the two-dimensional plane of latent variables LV1 and LV2 (or principal components 1 and 2). Used with perssion from Aspen Technology, Inc.

In Figure 9.3, we see the characteristics of the original 3D image on a two-dimensional plane of two linear combinations (LV1 and LV2) of the process variables (x, y and z):

$$LV1 = 0.1658\ x + 0.6120\ y + 0.7733\ z \qquad (9.1a)$$

$$LV2 = -0.9652\ x + 0.2615\ y \qquad (9.1b)$$

We call these linear combinations *the latent variables* or *principal components* of the process variables.

Principal component analysis (PCA) is a data transformation method that rotates data such that the principal axis of the data is in the direction of maximum variation. See Figure 9.4. We follow the interpretation of [15] here. The first latent variable or first principal component of the process data or observations given by Eq. (9.1a) represents the linear combination of the original process variables whose *sample variance* (see Appendix A, Section A.1.6) is greatest among all possible linear combinations. The second latent variable or second principal component represents the linear combination of the original process variables that accounts for a maximum proportion of the remaining variance, subject to being uncorrelated with the first principal component. We can define subsequent principal components similarly.

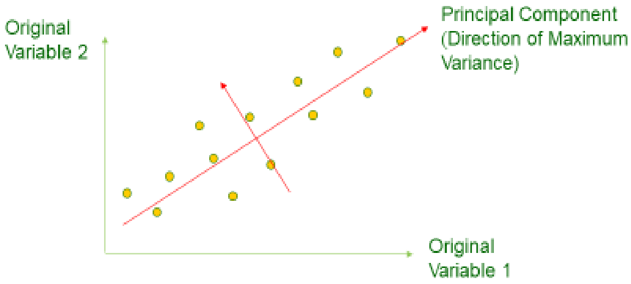


Figure 9.4 An illustration of the principal component that shows the direction of maximum variation of the process data

We can view the rotated data on the new principal axes (components). We call the coordinates of the data in this new coordinate system as *principal component scores*. They are essentially the projection of the data onto the principal axes. As seen in Figure 9.4, the principal components are essentially vectors in the original variable space, and these vectors are called *principal component loadings*. We quantify both the principal component scores and loadings, and their relationship to the original process data matrix in the following section.

9.1.2 Data Preprocessing: Mean-Centered and Scaled Process Data Matrix X, Principal Component Score Matrix T, and Principal Component Loading Matrix P

We follow [13] for the development of the PCA model. Let us consider a J x K *process data matrix* **X**, with K columns of process variables x_k (k=1,2 ,…, K), and with each variable x_k having J observations or measured values, x_{1k}, x_{2k}, x_{3k}, …x_{Jk} (or x_{jk}, j = 1, 2,…, J). In Appendix A, Section A.1.7**,** we introduce *the standardized data matrix,* or *mean-centered and scaled data matrix* **Xs**, and *the correlation coefficient matrix* **R** from the process data matrix **X**.

To correctly carry out PCA, we first preprocess the data. Specifically, we start with a *data standardization* step to convert the process data matrix **X** to a standardized data matrix **Xs** that is mean-centered and scaled by standard deviation [11,13]. For convenience in eliminating the letter "**s"** from a mean-centered and scaled data matrix **Xs**, we assume in the following discussion that *our process data matrix **X** has already gone through a standardization procedure* described in Appendix A, Sections A.1.5

to A.1.7. As we demonstrate in Appendix A, it only takes a single command using Matlab [zscore(X)] or Python [stats.zscore()], to standardize a process data matrix **X**.

9.1.3 Development of PCA Model

We write the standardized data matrix **X** as a matrix of K process variable vectors:

$$\mathbf{X} = [\mathbf{x_1}\ \ \mathbf{x_2} \ldots\ldots\mathbf{x_K}] = \begin{bmatrix} x_{11} & \cdots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{j1} & \cdots & x_{jk} \end{bmatrix} \qquad (9.2)$$

In this matrix, the k-th process variable vector $\mathbf{x_K}$ is a (J x 1) column vector, $[x_{1k}, x_{2k}, x_{3k}, \ldots x_{Jk}]'$, where J is the number of samples or measurements. The transpose of $\mathbf{x_K}$**,** or $\mathbf{x_k}'$, is a (1 x J) observation vector.

Figure 9.5 illustrates the projection of the vector $\mathbf{x_k}$ onto the first principal component vector $\mathbf{p_1}$. The score value $t_{k,1}$ for this observation vector is the distance from the origin along the principal component loading vector, $\mathbf{p_1}$**,** to the point where we find the perpendicular projection onto $\mathbf{p_1}$ [13].

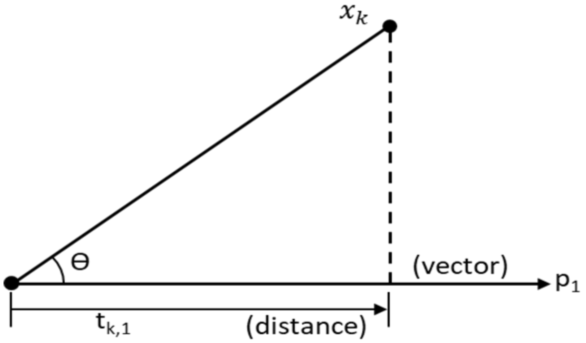


Figure 9.5 The projection of the kth process variable vector $\mathbf{x_k}$ onto the first principal component loading vector $\mathbf{p_1}$. $t_{k,1}$ is the score value of $\mathbf{x_k}$ on $\mathbf{p_1}$.

We can write from geometry that: (1) the cosine of an angle in a right-angled triangle is the ratio of the adjacent side to the hypotenuse; (2) the cosine of the defines the dot product of two vectors. See Eqs. (9.3) and (9.4):

$$\cos\theta = (\text{adjacent length})/(\text{hypotenuse}) = t_{k,1} / \| \mathbf{x_k} \| \qquad (9.3)$$

$$\cos\theta = \mathbf{x_k}'\, \mathbf{p_1} / \| \mathbf{x_k} \|\, \| \mathbf{p_1} \| \qquad (9.4)$$

where ‖ · ‖ represents the length of the enclosed vector, and the length of the principal component loading vector, $\| \mathbf{p_1} \|$ is 1.0. Therefore, we find:

$$t_{k,1} = \mathbf{x_k}'\, \mathbf{p_1} \qquad = x_{k,1}\, p_{1,1} + x_{k,2}\, p_{2,1} + \ldots + x_{k,j}\, p_{j,1} \ldots. + x_{k,J}\, p_{J,1} \qquad (9.5)$$

Likewise, we write

$$t_{k,2} = \mathbf{x_k}'\, \mathbf{p_2}$$

$$= x_{k,1}\, p_{1,2} + x_{k,2}\, p_{2,2} + \ldots + x_{k,j}\, p_{j,2} \ldots. + x_{k,J}\, p_{J,2} \qquad (9.6)$$

Generalizing Eq. (9.5) and (9.6), we write the principal component score vector $\mathbf{t_k}$ resulting from projecting the process data vector $\mathbf{x_k}$ onto A principal component loading vectors, expressed by the (K x A) loading matrix **P**:

$$\mathbf{t_k' = x_k' P} \qquad (9.7)$$
$$(1 \text{ x } A) = (1 \text{ x } K)\ (K \text{ x } A)$$

Lastly, we can represent the projection of the entire process data matrix **X** in terms of a principal component score matrix **T** and a principal component loading matrix **P:**

$$\mathbf{T = X\,P} \qquad (9.8)$$
$$(J \text{ x } A) = (J \text{ x } K)(K \text{ x } A)$$

where J is the number of samples or measurements, A is the number of principal components, and K is the number of process variables.

9.1.4 Prediction Errors from PCA Model

Figure 9.6 illustrates the projection of the original data vector $\mathbf{x_k}$ onto the first principal component vector $\mathbf{p_1}$. The best estimate of $\mathbf{x_k}$ is a vector $\hat{\boldsymbol{x}}_{\boldsymbol{k},\mathbf{1}}$ along the first principal component loading vector $\mathbf{p_1}$ where the original vector is projected. We call this estimate of the data vector, $\hat{\boldsymbol{x}}_{\boldsymbol{k},\mathbf{1}}$. We note the distance along the first principal component loading vector $\mathbf{p_1}$ is the principal component score $t_{k,1}$. Based on vector geometry, we represent the error between $\mathbf{x_k}$ and $\hat{\boldsymbol{x}}_{\boldsymbol{k},\mathbf{1}}$ as an error vector $\mathbf{e_{k,1}}$.

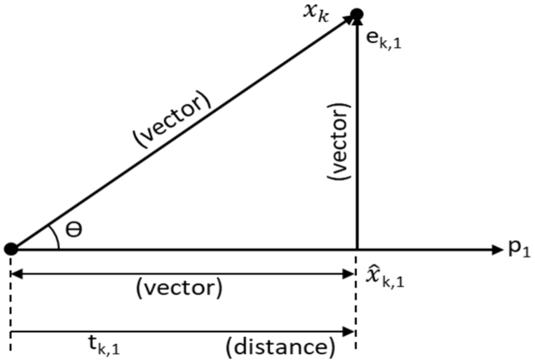


Figure 9.6 The projection of the kth process variable vector $\mathbf{x_k}$ onto the first principal component vector $\mathbf{p_1}$, indicating an estimate of the data vector $\hat{\boldsymbol{x}}_{\boldsymbol{k},\mathbf{1}}$, together with an error vector $\mathbf{e_{k,1}}$. $t_{k,1}$ is the score value of $\mathbf{x_k}$ on $\mathbf{p_1}$.

We write the prediction vector as:

$$\hat{\boldsymbol{x}}'_{\boldsymbol{k},\mathbf{1}} = t_{k,1}\,\mathbf{p'_1} \qquad (9.9)$$

$$(1 \text{ X } K) = (1 \text{ x } 1)(1 \text{ x } K)$$

and the corresponding prediction error vector is:

$$\mathbf{e'_{k,1}} = \mathbf{x'}_k - \hat{\boldsymbol{x}}'_{\boldsymbol{k},\mathbf{1}} \qquad (9.10)$$

$$(1 \text{ x } K) = (1 \text{ x } K) - (1 \text{ x } K)$$

Adding the second principal component vector $\mathbf{p_2}$, we generalize the prediction vector from Eq. (9.9) as:

$$\hat{\boldsymbol{x}}'_{\boldsymbol{k},\mathbf{2}} = t_{k,1}\,\mathbf{p'_1} + t_{k,2}\,\mathbf{p'_2} \qquad (9.11)$$

$$(1 \text{ x } K) = (1 \text{ x } K) + (1 \text{ x } K)$$

where $t_{k,1}$ and $t_{k,2}$ **are** the score values of $\mathbf{x_{k,2}}$ on $\mathbf{p_1}$ and $\mathbf{p_2}$, respectively.

Extending Eq. (9.11) to **A** principal component vectors, we write the projector vector of the original data vector $\mathbf{x_k}$ onto the A principal component loading vectors [$\mathbf{p_1}$ $\mathbf{p_2}$ **……**$\mathbf{p_A}$] or principal component loading matrix **P**, with $\mathbf{t_k}$ being the score vector:

$$\hat{\boldsymbol{x}}'_{\boldsymbol{k},\boldsymbol{A}} = [t_{k,1}\ t_{k,2}\ldots,\ t_{k,A}\ ]\mathbf{P'} = \mathbf{t'_k}\,\mathbf{P'} \qquad (9.12)$$

$$(1 \text{ x } A) = (1 \text{ x } A)\ (A \text{ x } K)$$

We generalize Eq. (9.12) to represent the entire data prediction matrix $\hat{\boldsymbol{X}}$ in terms of the score matrix **T** and the principal component loading matrix **P**:

$$\hat{\boldsymbol{X}} = \mathbf{T}\ \mathbf{P'} \qquad (9.13)$$

$$(J \text{ x } K\ ) = (J \text{ x } A)\ (A \text{ x } K)$$

We define the residual vector $\mathbf{e_{k,A}}$ for the k-th process variable using **A** principal components as the difference between the actual and predicted observations:

$$\mathbf{e'_{k,\,A}} = \mathbf{x'_k} - \hat{\boldsymbol{x}}'_{\boldsymbol{k},\boldsymbol{A}} = \mathbf{x'_k} - \mathbf{t'_k}\,\mathbf{P'} \qquad (9.14)$$

$$(1 \text{ x } A) = (1 \text{ x } A) - (1 \text{ x } A)$$

Referring to Figure 9.7, we define *the row residual or the squared prediction error (SPE)* for k-th process variable as:

$$\text{SPE}_k = (\ \mathbf{e'_{k,\,A}} \cdot \mathbf{e_{k,\,A}})^{\mathbf{1/2}}$$

$$= [\ (x_{k,1} - \hat{\mathrm{x}}'_{\mathrm{k},1})^2 + (x_{k,2} - \hat{\mathrm{x}}'_{\mathrm{k},2})^2 + \ldots\ldots\ldots + (x_{k,A} - \hat{\mathrm{x}}'_{\mathrm{k},\mathrm{A}})^2]^{\mathbf{1/2}} \qquad (9.15)$$

The corresponding vector representation of all SPE_k (k= 1,2 …K) for all K process variables is

$$\mathbf{SPE} = [\ \text{SPE}_1\ \text{SPE}_2\ldots,\ \text{SPE}_k\ ]' \qquad (9.16)$$

We write Eq. (9.14) as a prediction error or residual matrix **E** for all **K** process variables, **J** observations per variable, and **A** principal component loading vectors [$\mathbf{p_1}$ $\mathbf{p_2}$ **……**$\mathbf{p_A}$] or principal component loading matrix **P** as follows:

$$\mathbf{E} = \mathbf{X} - \hat{\boldsymbol{X}} = \mathbf{X} - \mathbf{T}\ \mathbf{P'} \quad \textbf{or} \quad \mathbf{X} = \mathbf{T}\ \mathbf{P'} + \mathbf{E} \qquad (9.17)$$

Figure 9.7 illustrates the relationship between **E, X,** $\hat{\boldsymbol{X}}$**,** and **SPE.**

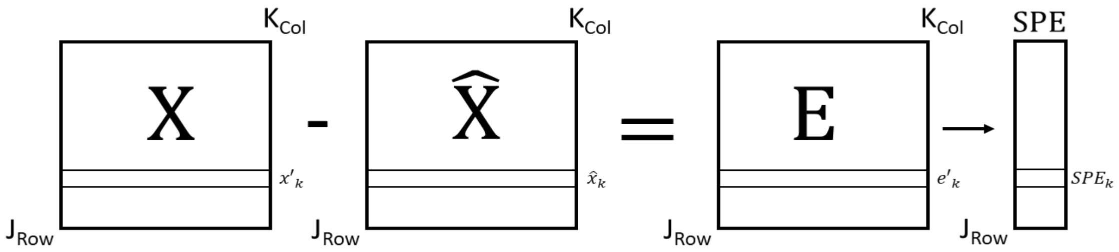


Figure 9.7 An illustration of the relationships among the prediction error matrix **E**, process variable matrix **X**, predicted process variable matrix $\hat{\boldsymbol{X}}$, and squared prediction error matrix **SPE**.

In Figure 9.7, each row of **E** contains the row residual or the prediction error for j-th observation (j= 1,2, …J) for all **K** process variables.

Figure 9.8 shows a similar plot, focusing on the column residual, or the prediction error for each column that represents the k-th process variable (k= 1,2, ...K) in the residual matrix **E** [13].

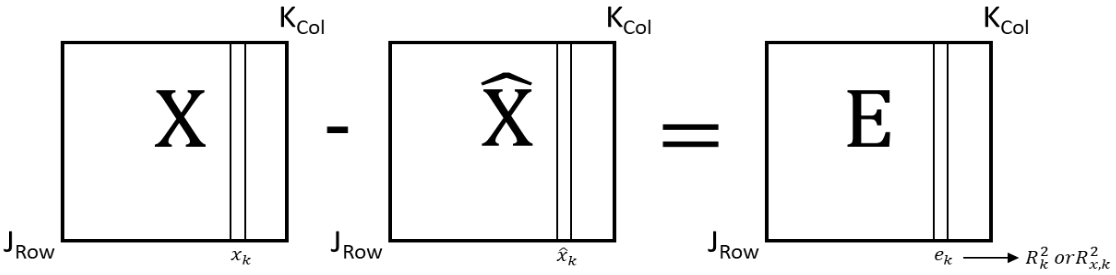


Figure 9.8 An illustration of the relationships among the prediction error matrix **E**, process variable matrix **X**, predicted process variable matrix $\hat{\boldsymbol{X}}$, and column or the prediction error for k-th process variable (column)

Each column of **E** contains the prediction error for one variable. Referring to the discussion of least squares model analysis on pages 165 to 168 of [13], we can find the R^2 value for the k-th process variable (column) as:

$$R_k^2 \;=\; R_{X,k}^2 = 1 - \frac{Var(x_k - \hat{\boldsymbol{x}}_{\boldsymbol{k}})}{Var(x_k)} \;= \frac{Var(e_k)}{Var(x_k)} \tag{9.18}$$

The R_k^2 value for each process variable will increase with every principal component that is added to the model. The minimum value is 0.0 when there is no principal component and $\hat{\boldsymbol{x}}_{\boldsymbol{k}} = 0$. The maximum value is 1.0 when we have added the maximum number of principal components with $x_k \;=\; \hat{\boldsymbol{x}}_{\boldsymbol{k}}$ and $e_k \;= 0$.

We can extend the preceding row residual and column residual concepts to the whole process data matrix **X** and calculate the R^2 value of the entire matrix [13]. *This value is the ratio of the variance of **X** that we can explain with the PCA model over the ratio of variance initially present in **X**.*

$$R^2 = \; 1 - \frac{Var(\boldsymbol{X} - \hat{\boldsymbol{X}})}{Var(X)} \;=1- \frac{Var(E)}{Var(X)} \tag{9.19}$$

By using ML or Python (see Appendix A), or Aspen Technology’s software Aspen ProMV, we can evaluate the R^2 value and identify the number of principal components needed to adequately explain the data variability in **X**. We have demonstrated this aspect in Appendix A and will illustrate this aspect in our hands-on workshop WS9.1, in which Aspen ProMV shows *the* $\boldsymbol{R^2}$ *value as* ***R2*** for different number of principal components.

Lastly, page 380 of Dunn [13] explains the concept of determining the number of principal components to use in a model based on cross-validation (CV), originally proposed by Wold [20]. We follow Dunn's exposition below.

The general idea is to divide the process data matrix **X** into G groups of rows. These rows should be selected randomly but are often selected in order: row 1 goes in group 1, row 2 goes in group 2, and so on. We can collect the rows belonging to the first group into a new matrix called $\mathbf{X}_{(1),}$ and leave behind all the other rows from all other groups, which we will call group $\mathbf{X}_{(-1)}$. So in general, for the g-th group, we can split matrix X into $\mathbf{X}_{(g)}$ and $\mathbf{X}_{(-g)}$. Wold's cross-validation procedure asks to build the PCA model on the data in $\mathbf{X}_{(-1)}$ using A components. Then use data in $\mathbf{X}_{(1)}$ as new, testing data. In other words, we preprocess the $\mathbf{X}_{(1)}$ rows, calculate their score values, $\mathbf{T}_{(1)} = \mathbf{X}_{(1)}\mathbf{P}$, calculate their predicted values, $\hat{\mathbf{X}}_{(1)} = \mathbf{T}_{(1)}\mathbf{P}'$, and their residuals, $\mathbf{E}_{(1)} = \mathbf{X}_{(1)} - \hat{\mathbf{X}}_{(1)}$. We repeat this process, building the model on $\mathbf{X}_{(-2)}$ and testing it with $\mathbf{X}_{(2),}$ to eventually obtain $\mathbf{E}_{(2)}$. After repeating this on G groups, we gather up $\mathbf{E}_1, \mathbf{E}_2, \ldots, \mathbf{E}_G$ and assemble a type of residual matrix, $\mathbf{E}_{A,\mathrm{CV}}$, where the A represents the number of components used in each of the G PCA models. The CV subscript indicates that this is not the usual error matrix, **E**. From this, we can calculate a type of $\boldsymbol{R}\mathbf{2}$ value. We do not call this $\boldsymbol{R}\mathbf{2}$, but it follows the same definition for an $\boldsymbol{R}\mathbf{2}$ value. We will call it $\boldsymbol{Q}\mathbf{2}_A$ instead, where A is the number of components used to fit the G models.

$$\mathrm{Q2}_A = 1 - \mathrm{Var}(\mathrm{E}_{A,\,\mathrm{CV}}) / \mathrm{Var}(\mathrm{X}) \qquad (9.20)$$

Essentially, Q2$_A$ is a measure of how well the process variables will be predicted with new data calculated by cross validation. *I*n our hands-on workshop WS9.1, Aspen ProMV shows *the* Q^2 *value as Q2* for different number of principal components with cross validation.

9.1.5 Hotelling's T^2 value from PCA Model

In Figure 9.6, we illustrate the score value $\mathbf{t}_{k,1}$ of process variable vector $\mathbf{x}_k$ on the first principal component $\mathbf{p}_1$. Let $t_{k,a}$ (k=1,2…K; a=1,2,….A) be the score value of kth process variable $\mathbf{x}_k$ on the a-th principal component, and s_a (a= 1,2,…A) be the variance of the a-th principal component. Then, the Hotelling's $\mathbf{T}^2$ value for the k-th process variable is:

$$\mathrm{T}^2 = \Sigma\, (t_{k,a}/s_a)^2 \qquad (9.21)$$

$\mathbf{T}^2$ value is a positive, scalar number that summarizes all the score values. It represents the distance from the center of the hyperplane of process variables to the projection of the sample onto the hyperplane. For samples that are very close to the sample mean gives a $\mathbf{T}^2$ value of zero [15].

Figure 9.9 illustrates the concept of Hotelling's $\mathbf{T}^2$ value for an example with two principal components (A=2):

$$\mathrm{T}^2 = \frac{t_1^2}{S_1^2} + \frac{t_2^2}{S_2^2} \qquad (9.22)$$

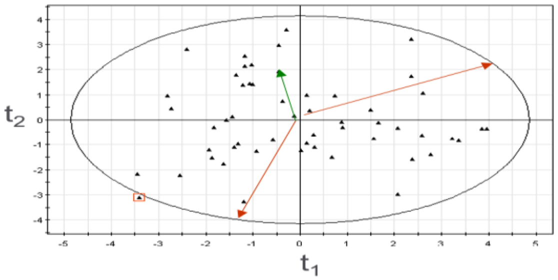


Figure 9.9 An illustration of the concept of Hotelling's T^2 value in a two-latent-variable or a two-principal-component space, t_2 versus t_1

In the figure, the equation for $\mathbf{T^2}$, Eq. (9.21), is that of an ellipse. $\mathbf{T^2}$ expresses how far an observation is from the center of the model in the plane. All points on the ellipse have the same $\mathbf{T^2}$ value.

We note that references [11,15], among others, have presented the detailed development, showing that the variances of principal components s_a (a= 1,2,…A) are actually the eigenvalues of the correlation coefficient matrix **R,** which is introduced in Appendix A, Section A.1.7, and Eq. (A.24), based on the standardized data matrix **Xs**. Additionally, the eigenvectors of **R** correspond to principal component loading vectors $\mathbf{p_a}$ (a= 1,2 …A). Extracting principal components as the eigenvectors of **R** is equivalent to calculating the principal components from the original variables after each has been standardized to have zero mean and unit variance [13], as we discussed in Appendix A, Sections A.1.5 to A.1.7.

In Appendix B of this book, code B.8 and Table B.1 at the end give the Python implementation of the PCA algorithm, together with a list of common parameters and their suggested values.

9.2 Workshop 9.1: PCA of the Process Variables Affecting the Quality and Conversion of LDPE Product from a Two-Zone Tubular Reactor

We demonstrate the development of a PCA model for analyzing the quality and conversion of a two-zone tubular reactor for producing low-density polyethylene (LDPE). The problem comes from references [6,18], and the process data for LDPE are available in [19]. We use Aspen Technology's multivariate statistical analysis software, Aspen ProMV, for this workshop. The LDPE production process is similar to the process defined by Sharma & Liu.

Figure 9.10 shows a schematic diagram of the two-zone reactor, and Table 9.1 defines the 14 process variables (**X**) and 5 product quality variables (**Y**).

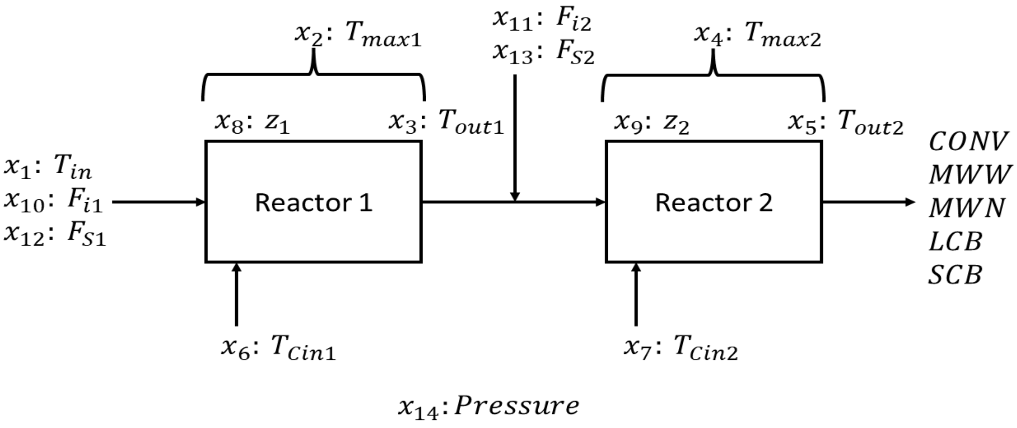


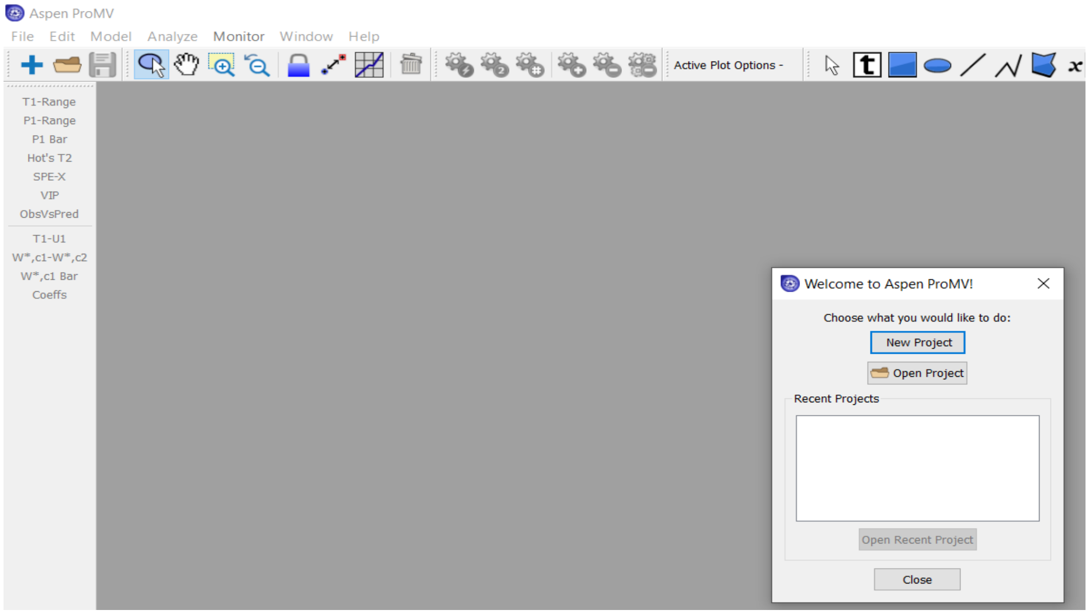
Figure 9.10 A schematic diagram of a two-zone tubular reactor for producing LDPE

Table 9.1 Process and quality variables for workshop 9.1 [18]

Process variables (X)		Quality variables (Y)	
T_{max1},T_{max2}	Maximum temperature of the reaction mixture (K) (subscripts 1 and 2 refer to zones 1 and 2)	Conv	Cumulative conversion of the monomer
T_{out1}, T_{out2}	Outlet temperature of the reaction mixture (K)	MWN	Number-average molecular weight
T_{cin1}, T_{cin2}	Inlet temperature of the coolant (K)	MWW	Weight-average molecular weight
Z_1, Z_2	Axial reactor length of T_{max1} and T_{max2} (% of the reactor length)	LCB	Long chain branching per 1,000 carbon atom
F_{i1}, F_{i2}	Flow rate of the initiator (g/s)	SCB	Short chain branching per 1,000 carbon atom
F_{s1}, F_{s2}	Flow rate of the solvent in the inlet feed and in the intermediate feed (% of the ethylene flow rate)		
T_{in}	Inlet temperature of the reaction mixture (K)		
Press	Reactor pressure (atom)		

*The outlet temperature of the coolant in two zones is fixed.

Step 1. Start Aspen ProMV. Select new projects. See Figure 9.11.



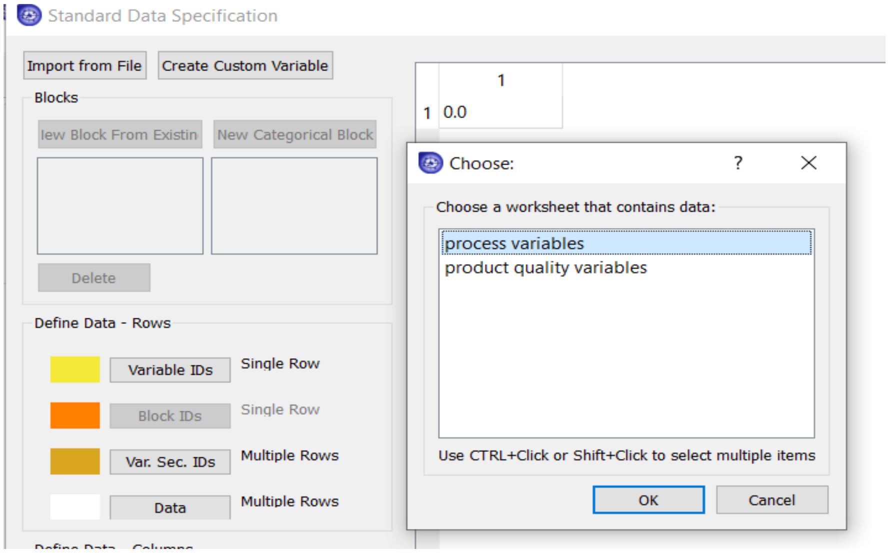
Welcome to Aspen ProMV!

Choose what you would like to do:

Figure 9.11 Choose New project in Aspen ProMV.

Step 2. Load the process data file and save the project file.

Click on Add/Edit Data and Import from File, ***LDPE.xls***. Choose Process Variables only. See Figures 9.12. Figure 9.13 displays a portion of the imported process variable data. By clicking OK twice, we see the Standard Data Specification. See Figure 9.14. We then click OK and see the observation summary of Figure 9.15. Highlight the observation ID column to include all observations and the Include observations button turns “green” to indicate that we have included all observation data. See Figure 9.16. Click OK. Save the project as ***WS9.1_PCA-X.pmvx***. See Figure 9.17.

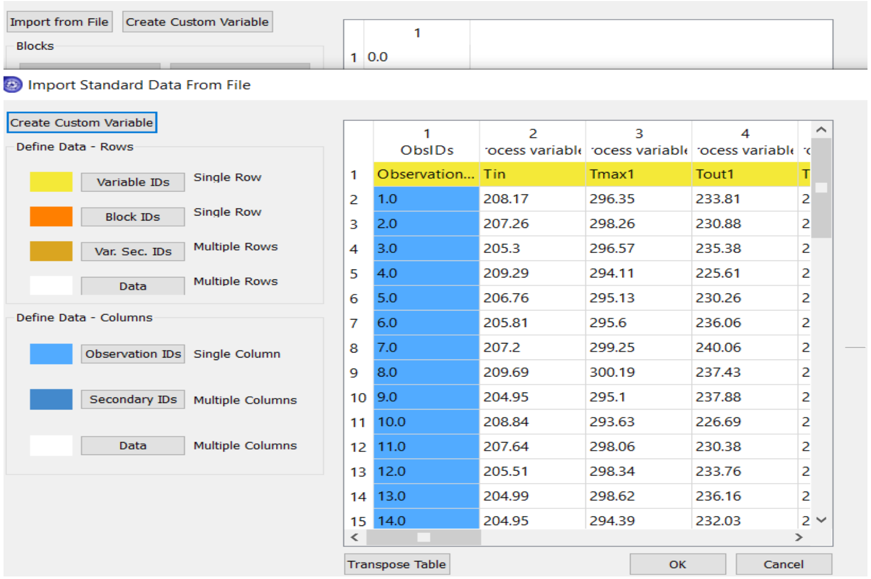


Choose a worksheet that contains data:

process variables
product quality variables

Use CTRL+Click or Shift+Click to select multiple items

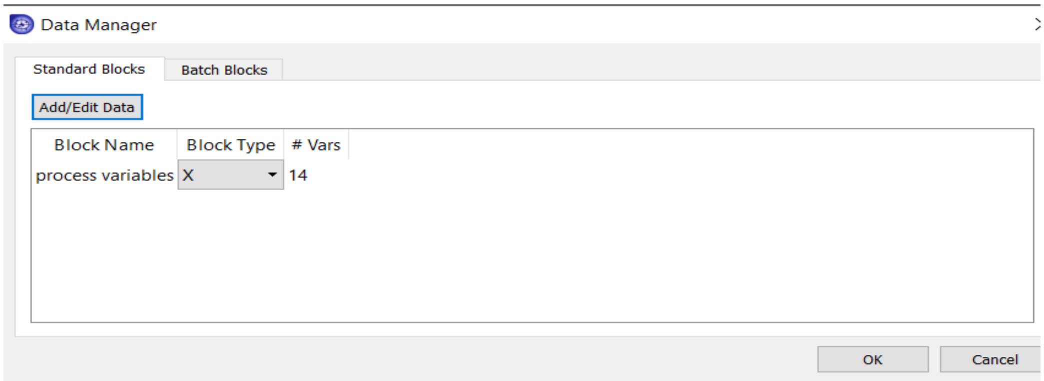
Figure 9.12 Import process data from file, ***LDPE.xls***, and choose process variable worksheet only



	1
1	0.0

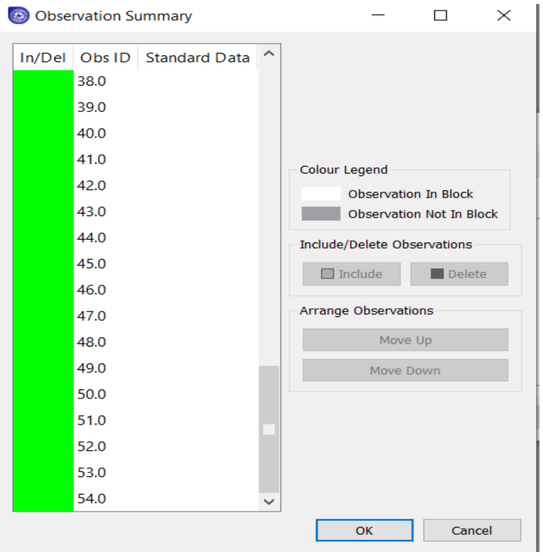
	1 ObsIDs	2 ocess variabl	3 ocess variabl	4 ocess variabl	c
1	Observation...	Tin	Tmax1	Tout1	T
2	1.0	208.17	296.35	233.81	2
3	2.0	207.26	298.26	230.88	2
4	3.0	205.3	296.57	235.38	2
5	4.0	209.29	294.11	225.61	2
6	5.0	206.76	295.13	230.26	2
7	6.0	205.81	295.6	236.06	2
8	7.0	207.2	299.25	240.06	2
9	8.0	209.69	300.19	237.43	2
10	9.0	204.95	295.1	237.88	2
11	10.0	208.84	293.63	226.69	2
12	11.0	207.64	298.06	230.38	2
13	12.0	205.51	298.34	233.76	2
14	13.0	204.99	298.62	236.16	2
15	14.0	204.95	294.39	232.03	2

Figure 9.13 A display of imported process variable data



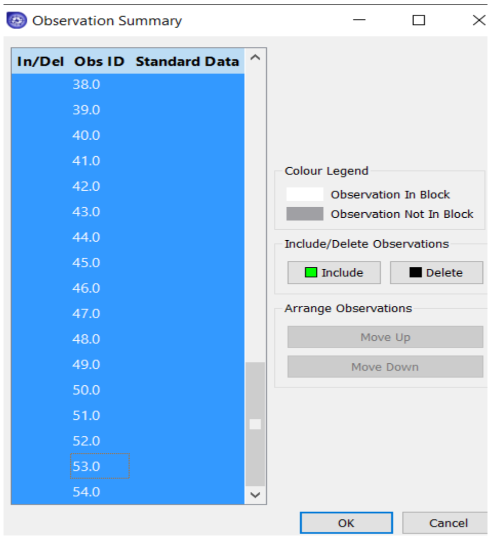
Block Name	Block Type	# Vars
process variables	X	14

Figure 9.14 Specify block type for process variables, X



In/Del	Obs ID	Standard Data
🟩	38.0	
🟩	39.0	
🟩	40.0	
🟩	41.0	
🟩	42.0	
🟩	43.0	
🟩	44.0	
🟩	45.0	
🟩	46.0	
🟩	47.0	
🟩	48.0	
🟩	49.0	
🟩	50.0	
🟩	51.0	
🟩	52.0	
🟩	53.0	
🟩	54.0	

Figure 9.15 A display of observation summary


Observation Summary

In/Del	Obs ID	Standard Data
	38.0	
	39.0	
	40.0	
	41.0	
	42.0	
	43.0	
	44.0	
	45.0	
	46.0	
	47.0	
	48.0	
	49.0	
	50.0	
	51.0	
	52.0	
	53.0	
	54.0	

Colour Legend
- Observation In Block
- ▨ Observation Not In Block

Include/Delete Observations

🟩 Include ■ Delete

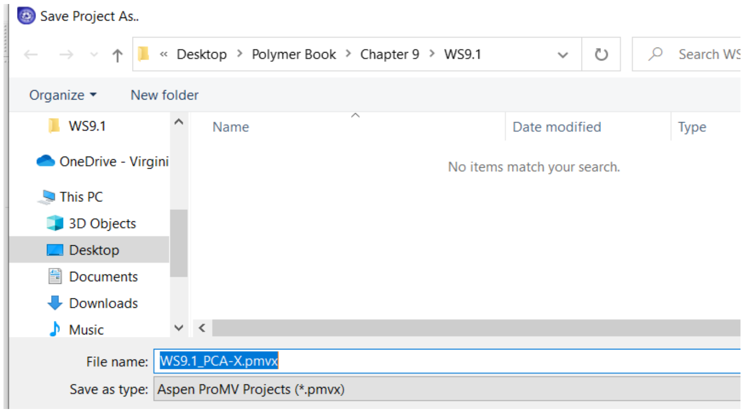
Arrange Observations

Move Up

Move Down

OK Cancel

Figure 9.16 Highlight all observations to include them in the model development


Save Project As..

« Desktop › Polymer Book › Chapter 9 › WS9.1

Search WS

Organize ▾ New folder

- WS9.1
- OneDrive - Virgini
- This PC
- 3D Objects
- Desktop
- Documents
- Downloads
- Music

Name	Date modified	Type

No items match your search.

File name: WS9.1_PCA-X.pmvx

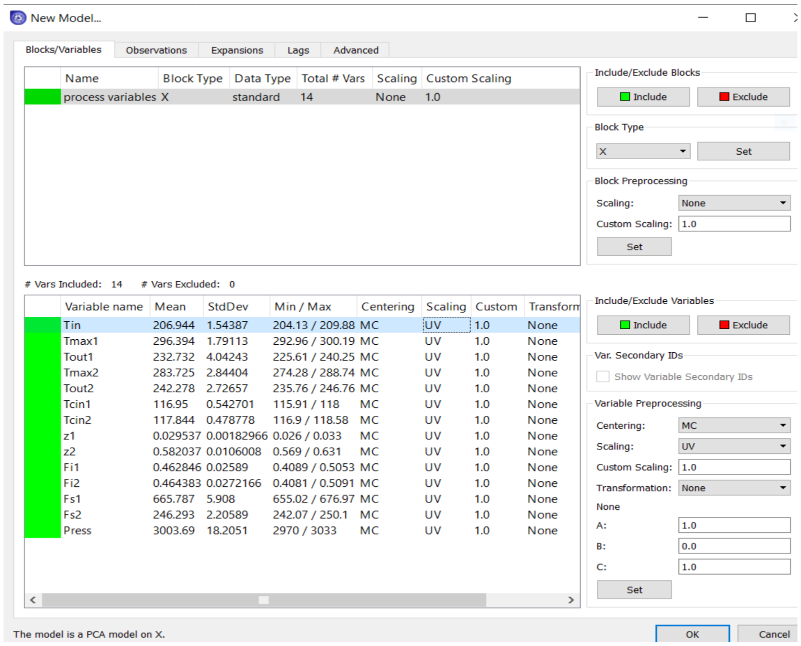
Save as type: Aspen ProMV Projects (*.pmvx)

Figure 9.17 Saving the Aspen ProMV project file as ***WS9.1_PCA-X.pmvx***.

Step 3. Build a PCA model for process variables X.

After saving the project file, we see the New Model dialog. We click on the Blocks/Variables name, “Process Variables”, to display the 14 process variables. Both the Block name and Variable names are in green. See Figure 9.18. In the figure, “MC” and “UV” represent the preprocessing of data to make them Mean-Centered with Unit Variance Scaling, as we discuss in Appendix A, Sections A.1.5 and A.1.7 for standard data matrix. “Custom” in the figure refers to Custom Scaling, that is, the variables will be multiplied by this custom value after we have applied data centering and scaling.

We then click OK, and fill in the model name, ***WS9.1_PCA-X.pmvx.*** See Figure 9.19. Select Model -> Active Model ->Auto Fit -> See Figure 9.20. Figure 9.21 shows the resulting R2 and Q2 values, Eqs. (9.18) to (9.20), versus the number of principal components. We can right-mouse-click on this plot and select “Create Table” to see a table of R2 and Q2 values in the plot, as seen on the right of Figure 9.21.



	Name	Block Type	Data Type	Total # Vars	Scaling	Custom Scaling
🟩	process variables	X	standard	14	None	1.0

Block Type: X

Block Preprocessing

Scaling: None

Custom Scaling: 1.0

Vars Included: 14 # Vars Excluded: 0

	Variable name	Mean	StdDev	Min / Max	Centering	Scaling	Custom	Transform
🟩	Tin	206.944	1.54387	204.13 / 209.88	MC	UV	1.0	None
🟩	Tmax1	296.394	1.79113	292.96 / 300.19	MC	UV	1.0	None
🟩	Tout1	232.732	4.04243	225.61 / 240.25	MC	UV	1.0	None
🟩	Tmax2	283.725	2.84404	274.28 / 288.74	MC	UV	1.0	None
🟩	Tout2	242.278	2.72657	235.76 / 246.76	MC	UV	1.0	None
🟩	Tcin1	116.95	0.542701	115.91 / 118	MC	UV	1.0	None
🟩	Tcin2	117.844	0.478778	116.9 / 118.58	MC	UV	1.0	None
🟩	z1	0.029537	0.00182966	0.026 / 0.033	MC	UV	1.0	None
🟩	z2	0.582037	0.0106008	0.569 / 0.631	MC	UV	1.0	None
🟩	Fi1	0.462846	0.02589	0.4089 / 0.5053	MC	UV	1.0	None
🟩	Fi2	0.464383	0.0272166	0.4081 / 0.5091	MC	UV	1.0	None
🟩	Fs1	665.787	5.908	655.02 / 676.97	MC	UV	1.0	None
🟩	Fs2	246.293	2.20589	242.07 / 250.1	MC	UV	1.0	None
🟩	Press	3003.69	18.2051	2970 / 3033	MC	UV	1.0	None

Variable Preprocessing

Centering: MC

Scaling: UV

Custom Scaling: 1.0

Transformation: None

None

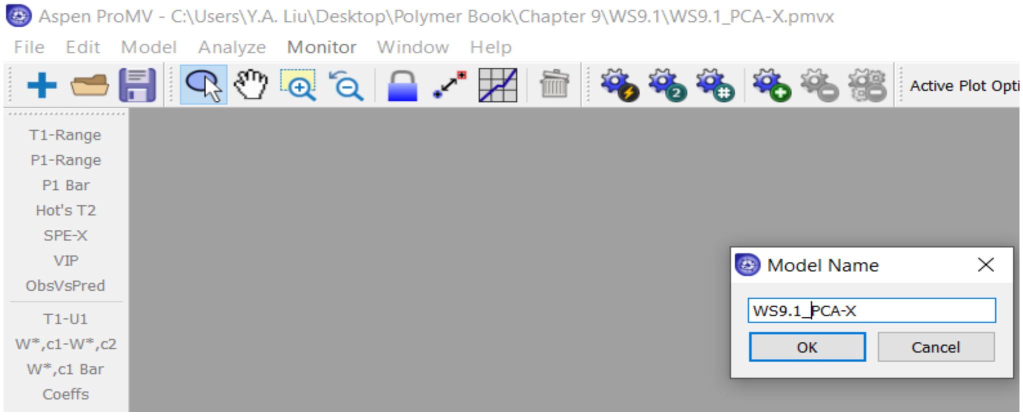
A: 1.0

B: 0.0

C: 1.0

The model is a PCA model on X.

Figure 9.18 Process variables (X) for developing a PCA model on X

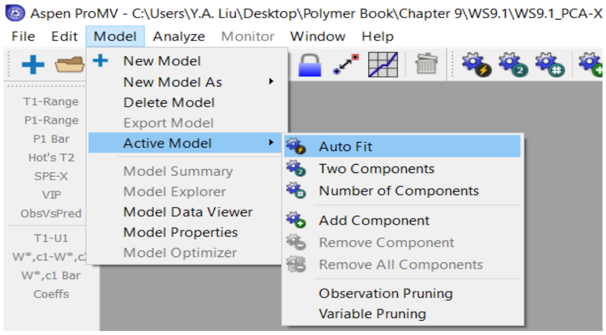


Aspen ProMV - C:\Users\Y.A. Liu\Desktop\Polymer Book\Chapter 9\WS9.1\WS9.1_PCA-X.pmvx

Model Name

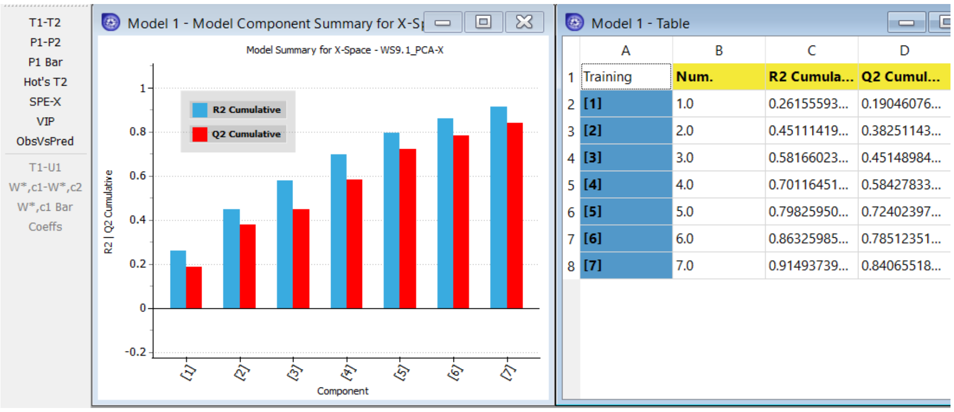
WS9.1_PCA-X

Figure 9.19 Filling in model name, WS9.1_PCA-X.pmvx.



Aspen ProMV - C:\Users\Y.A. Liu\Desktop\Polymer Book\Chapter 9\WS9.1\WS9.1_PCA-X

Figure 9.20 Auto-fitting the PCA model with the number of principal components (A) equal to half of the number of process variables (N=14), A = 14/2 = 7.



	A	B	C	D
1	Training	Num.	R2 Cumula...	Q2 Cumul...
2	[1]	1.0	0.26155593...	0.19046076...
3	[2]	2.0	0.45111419...	0.38251143...
4	[3]	3.0	0.58166023...	0.45148984...
5	[4]	4.0	0.70116451...	0.58427833...
6	[5]	5.0	0.79825950...	0.72402397...
7	[6]	6.0	0.86325985...	0.78512351...
8	[7]	7.0	0.91493739...	0.84065518...

Figure 9.21 R2 and Q2 values versus the number of principal components.

Figure 9.21 shows the cumulative R2 and Q2 values for each model component. The R2 of the final component is the total amount of the variability in the dataset that the model explains, and the Q2 value of the final component is a measure of how well the dataset is predicted by unseen data in cross validation. If the R2 and Q2 values are low, it could mean that there is significant noise in the data, existence of significant outliers, or not enough information in the data to fit an acceptable model. The figure indicates that increasing the number of principal components or latent variables, increses the R2 value, as explained previously in Section 9.1.4. With 7 principal components, an R2 value of 0.9149 says that the PCA model can explain 91.49% of the variability of the dataset for process variables (X). Likewise, with cross validation, a Q2 vale of 0.8406 says the model can explain 84.07% of the variability of the dataset for process variables.

Task 4. Generating PCA plots and their interpretations.

We follow the Aspen ProMV online help section on Interpreting Plots to demonstrate some useful plots and their interpretations.

(1) Model summary (R2 and Q2) plots for a selected maximum number of principal components

Choose the # button in the middle of the top of the screen, and fill in a maximum number of 4 principal components, we get a R2-Q2 plot of Figure 9.22.

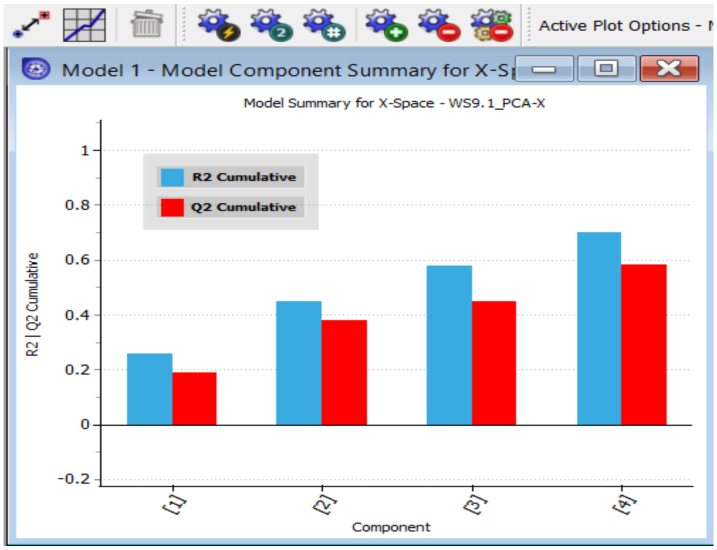
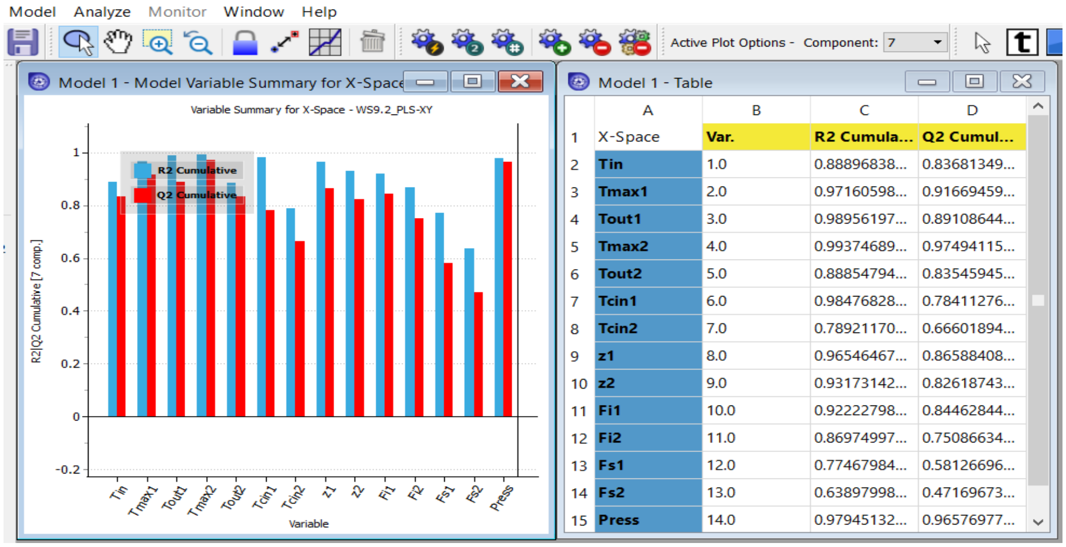


Figure 9.22 R2-Q2 plot with a maximum number of 4 principal components

(2) Variable summary plot

We follow the path: Analyze -> Model -> Variable summary ->Block: Choose X-space, and Component: Choose 7, and see Figure 9.23. We also right-mouse-click on this figure and select "Create Table" to see a table of R2 and Q2 values in the figure, as seen on the right of Figure 9.23. This figure shows the total R2 and Q2 of each X variable in a PCA model. As explained previously with Figure 9.21, if there are many variables and a few variables are not predicted well, this may mean there is no information in the dataset that can well explain these variables, not enough variation in the variables, too much noise, or significant outliers.



	A	B	C	D
1	X-Space	Var.	R2 Cumula...	Q2 Cumul...
2	Tin	1.0	0.88896838...	0.83681349...
3	Tmax1	2.0	0.97160598...	0.91669459...
4	Tout1	3.0	0.98956197...	0.89108644...
5	Tmax2	4.0	0.99374689...	0.97494115...
6	Tout2	5.0	0.88854794...	0.83545945...
7	Tcin1	6.0	0.98476828...	0.78411276...
8	Tcin2	7.0	0.78921170...	0.66601894...
9	z1	8.0	0.96546467...	0.86588408...
10	z2	9.0	0.93173142...	0.82618743...
11	Fi1	10.0	0.92222798...	0.84462844...
12	Fi2	11.0	0.86974997...	0.75086634...
13	Fs1	12.0	0.77467984...	0.58126696...
14	Fs2	13.0	0.63897998...	0.47169673...
15	Press	14.0	0.97945132...	0.96576977...

Figure 9.23 Variable summary plot in the X-space in a PCA model

(3) "Components by variable" plot

We show Figure 9.24 by following the path: Analyze -> Model -> Components by variable ->Block: choose process variables, and Variable: choose Tin (displayed in Figure 9.10, inlet temperature of the reaction mixture). This figure shows the R2 and Q2 values for all of the components for a specific X or Y variable. In this case, it is the X variable, inlet temperature, Tin.

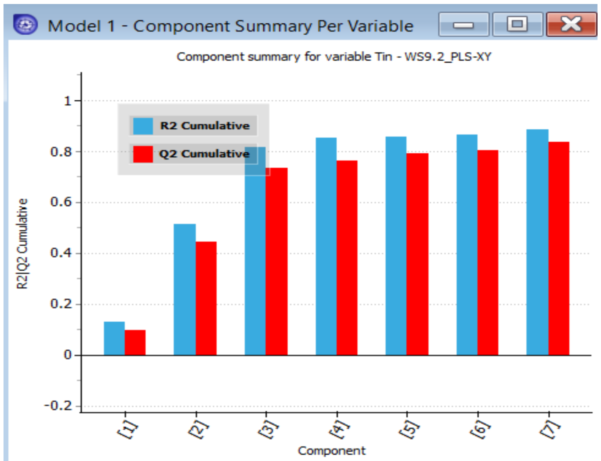


Figure 9.24 R2 and Q2 values for all components for a specific X or Y variable

(4) T1-T2 score plot and P1-P2 loading plot

We discussed in Section 9.1.4 of the score matrix T and the loading matrix P. In Appendix A, Section B.4, we demonstrate the use of ML to generate these matrices. By choosing T1-T2 and P1-P2 buttons on the left side pane, we generate the score plot and loading plot of Figure 9.25. These figures plot the score and loading values of the second principal component, T[2] and P[2], versus those of the first principal component, T[1] and P[1].

The scores are the latent variables, which are the weighted averages of the original process variables, X's. The score plot enables us to find clusters (such as in the middle of the left score plot in Figure 9.25) and outliers (such as observation 54). On the score plot, the inner dashed ellipse represents 95% confidence limit, while the outer solid ellipse represents 99% confidence limit. Observations that fall outside the 95% or 99% confidence intervals may be outliers; however, 5% and 1% of the observations are expected to naturally fall outside of the 95% and 99% confidence intervals, respectively. This plot shows the scores for a PCA model where there are both data clustering and a possible outlier.

The loadings are the model variables that explain the relationship between the X variables in PCA (or X and Y variables in the case of PLS discussed in Sections 9.3 and 9.4) and the latent variables (scores). In the right loading plot of Figure 9.25, variables that are close to the center of the plot are not significant for explaining the variation in the plotted components. By contrast, variables that are far from the center are important for explaining the variation in the dataset. Variables that are close together on the loading plot are correlated and variables that lie on the opposite sides of the plot are negatively correlated.

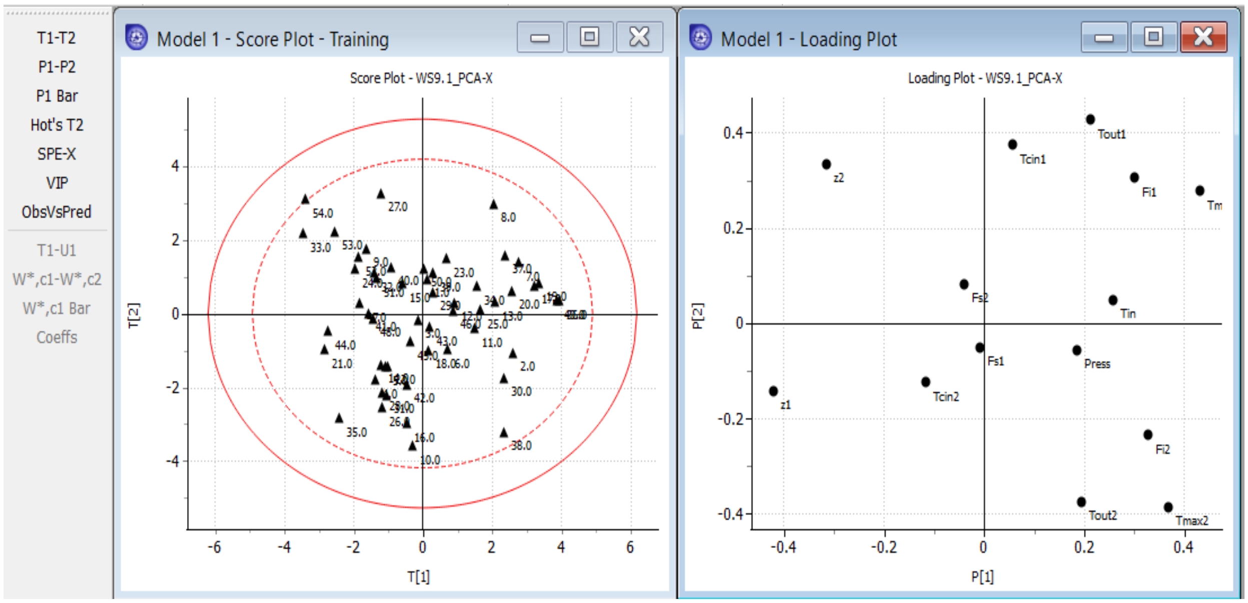


Figure 9.25 T[2] vs T{1} score plot and P[2] vs P[1] loading plot

(5) Hotelling's T^2 plot

Select the Hot's T2 button on the left side pane. We see the Hotelling's T^2 plot of Figure 9.26. This plot measures the deviation of an observation from the origin, that is, from the average operating point. Note the two horizontal lines labelled 0.99 and 0.95 for 99% and 95% confidence limits. We see observation #54 lie above the 95% confidence limit. This is acceptable, as there are generally on average 5 out 100 observations lying outside the 95% confidence limit.

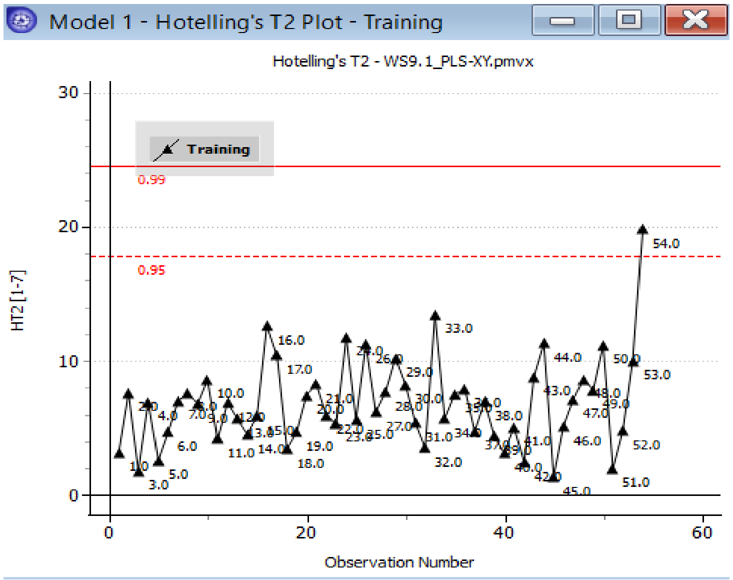


Figure 9.26 The Hotelling's T^2 plot

(6) Row residual or squared prediction error SPE-X plot

We discussed the SPE in Eqs. (9.15) and (9.16) and in Figure 9.7. Choosing SPE-X button on the left side pane gives the SPE-X plot of Figure 9.27. In the plot, we see that observation 54 has the largest SPE value.

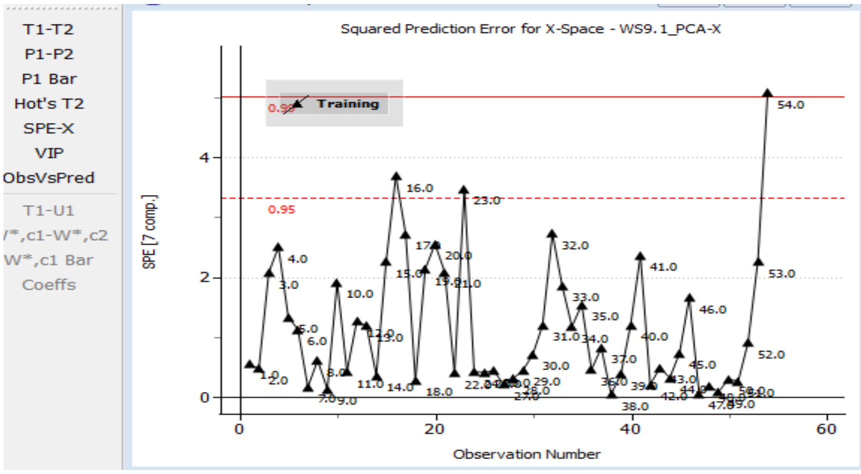


Figure 9.27 The SPE-X plot

(7) Variable importance to projection (VIP) plot

Choosing VIP button on the left side pane gives the VIP plot of Figure 9.28. The VIP plot gives a quantitative metric of the relative importance of a variable to a PCA model. A rule of thumb is that Variables with a VIP value close to or greater than 1 are important. Referring to Figure 9.10 and Table 9.1 for the definitions of process variables, we see that T_{out1}, T_{max2}, T_{cin1}, and T_{cin2} are four most important process variables in the PCA model.

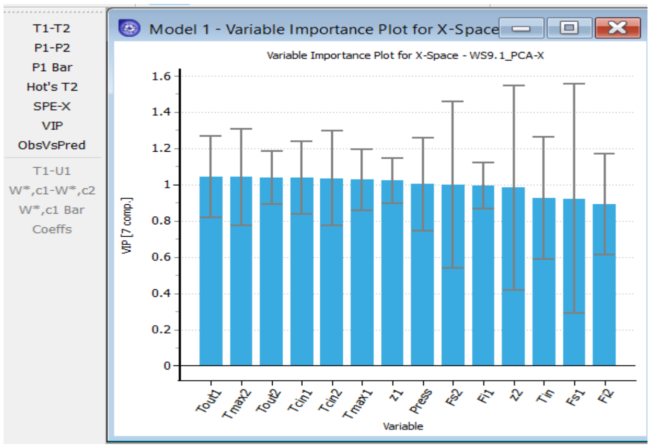
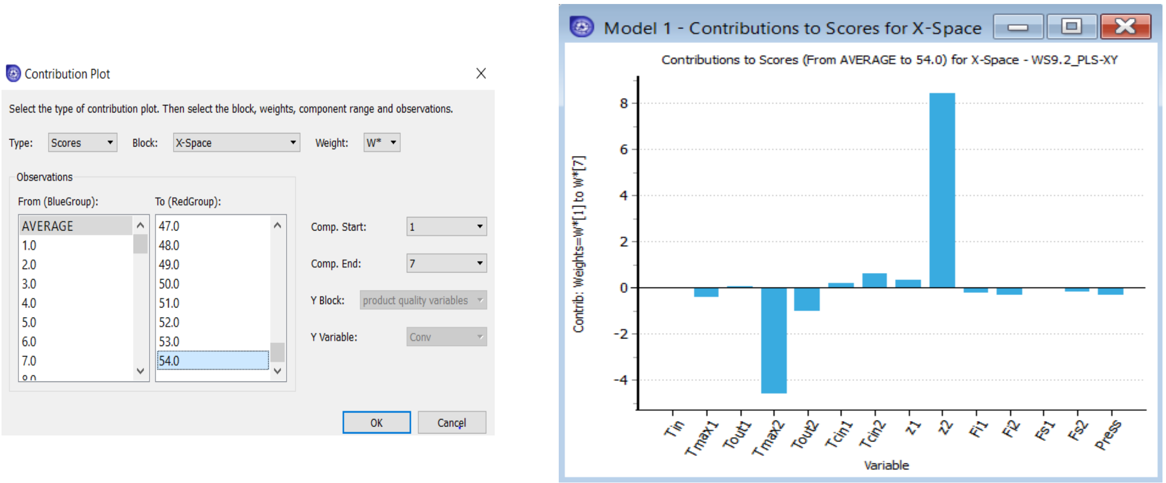


Figure 9.28 Variable importance plot.

(8) Contribution plots

When a single observation is selected on a score plot or Hotelling's T2 plot, the contribution plot shows the difference between that observation and the average observation. Following the path: Analyze -> Contributions ->This opens the contribution plot window. In this window, we specify analysis between our apparent outlier, observation 54, and the average observation using the specifications shown in Figure 9.29, left. These inputs yield the contribution plot on the right side of Figure 9.29. We see that z2 (the axial reactor length at the maximum temperature of the reaction mixture in zone 2, Tmax2, Figure 9.10) is higher than the average, and Tmax2 is lower than the average.



Contribution Plot

Select the type of contribution plot. Then select the block, weights, component range and observations.

Type: Scores   Block: X-Space   Weight: W*

Observations

From (BlueGroup): AVERAGE, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0

To (RedGroup): 47.0, 48.0, 49.0, 50.0, 51.0, 52.0, 53.0, 54.0

Comp. Start: 1

Comp. End: 7

Y Block: product quality variables

Y Variable: Conv

OK
Cancel
Model 1 - Contributions to Scores for X-Space
Contributions to Scores (From AVERAGE to 54.0) for X-Space - WS9.2_PLS-XY
Contrib: Weights=W*[1] to W*[7]
8
6
4
2
0
-2
-4
Tin
Tmax1
Tout1
Tmax2
Tout2
Tcin1
Tcin2
z1
z2
Fi1
Fi2
Fs1
Fs2
Press
Variable

Figure 9.29 Point-to-average contribution point.

This concludes workshop ***WS9.1_PCA-X***, and we save the file as ***WS9.1_PCA-X.pmvx***.

9.3 Partial Least Squares or Projection to Latent Structures (PLS)

9.3.1 Introduction to PLS

When applying data analytics to chemical processes, we often deal with not only process variables and their measurements, but also focus on quality or productivity variables. Let **X** be a JxK process variable matrix with K columns of process variables (k=1 to K), and J rows of measurements per variable (j=1 to K), and **Y** be a JxM process quality variable matrix with M columns of quality variables (m =1 to M), and J rows of measurements per variable (j = 1 to J).

As discussed in Section 9.1.1, PCA rotates the process data such that the principal axis of the data represents the direction of maximum variation. The projection of the data in the new coordinate system of A principal components is called the *principal components or latent variable scores*, which are represented by a JxA score matrix **T**, where J represents the rows of measurements per variable, and A denotes the number of principal components. Additionally, the principal components are vectors in the original variable space, and we call these vectors *principal component or latent variable loadings*, which are represented by a (KxA) principal loading matrix **P,** where K represents the number of process variables and A is the number of principal components.

Since PCA only uses X-data to find the principal component scores, **T**, these components explain variation in X-data, and not necessarily the most predictive of Y-data. In this section, we wish to use both X-data and Y-data simultaneously to identify the latent variables that explain the variation in X and are predictive of Y.

In Figure 9.30, we decompose the standardized J x K process variable matrix **X** and J x M product quality matrix **Y** into their principal component loading vectors ($\mathbf{p}_a$ and $\mathbf{c}_a$, a = 1,2 …A), and principal component score vectors ($\mathbf{t}_a$ and $\mathbf{u}_a$; a = 1, 2, …A). Alternatively, we can express the entire process variable matrix **X** in terms of a principal component loading matrix **P,** a principal component score matrix **T**, and a prediction error or residual matrix **E**, as seen previously in Eq. (9.17). Likewise, we represent the entire product quality matrix **Y** in terms of a principal component loading matrix **C,** a principal component score matrix **U**, and a prediction error or residual error matrix **F**. See Eq. (9.23).

$$\mathbf{X} = \mathbf{T\,P'} + \mathbf{E} = \hat{X} + \mathbf{E} \qquad (9.17)$$

$$(J \times K) = (J \times A)\,(A \times K)$$

$$\mathbf{Y} = \mathbf{U\,C'} + \mathbf{F} = \hat{Y} + \mathbf{F} \qquad (9.23)$$

$$(J \times M) = (J \times A)\,(A \times M)$$

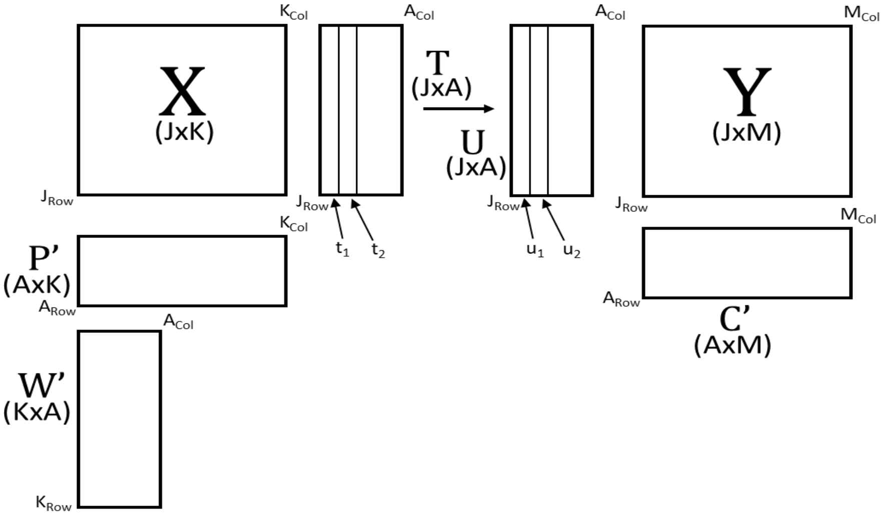


Figure 9.30 An illustration of partial least squares (PLS) regression of both process variable matrix **X** and product quality matrix **Y**, their principal component loading matrices **P** and **C,** and score matrices **T** and **U**, together with the weight matrix **W**ʹ for relating the score matrix **T** with process variable matrix **X** [13].

We can estimate the **X**-scores (i.e., principal component score vectors $\mathbf{t}_a$'s or principal component score matrix **T**) as linear combinations of the original process variable vectors $\mathbf{x}_k$ (k=1,2 …K) with the coefficients, "weights", **w**

$$t_{ja} = \Sigma\, x_{jk}\, w_{ka} \quad \text{or} \quad \mathbf{T} = \mathbf{X\,W} \qquad (9.24)$$

9.3.2 Nonlinear Iterative Partial Least Squares (NIPALS) Algorithm

We follow [13,16,17] to show how to compute the principal components sequentially and to handle missing data, with reference to the steps involved in Figure 9.31 [13].

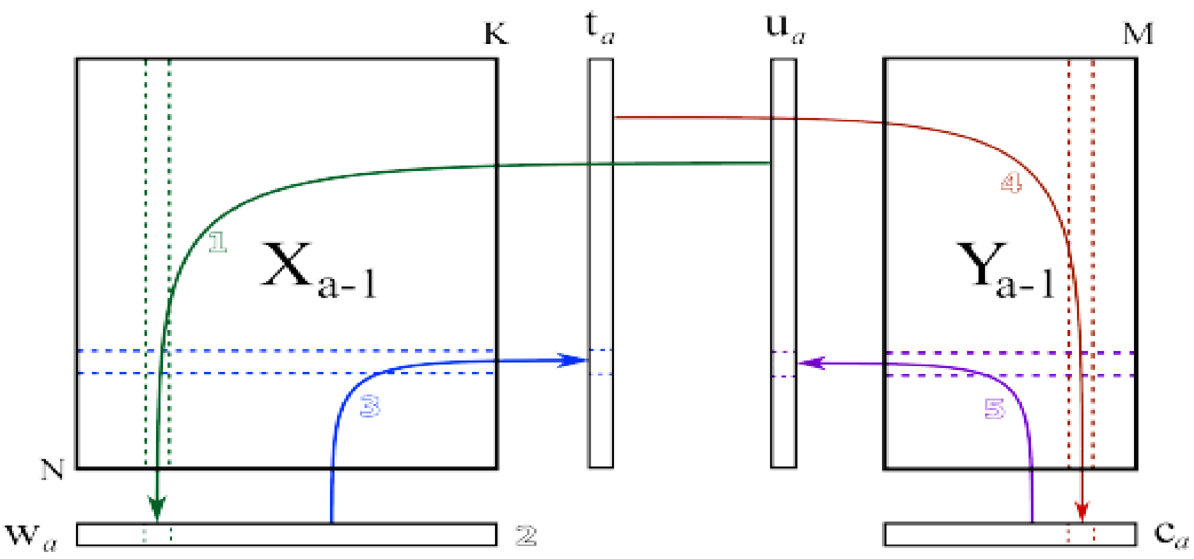


Figure 9.31 An illustration of the steps involved in computing the principal components by the NIPALS algorithm [13].

Step 1. See the number or the numbered arrow direction in Figure 9.31 for the numbered step. Begin with the J x K process variable matrix **X** and J x M process quality matrix **Y**. $\mathbf{X}_a$ **and** $\mathbf{Y}_a$ are both preprocessed versions of the raw data when the number of principal components a equal to 1. Select a column $\mathbf{y}_a$ within the quality matrix **Y** as our initial estimate for a score vector $\mathbf{u}_a$. Regress a data column $\mathbf{x}_a$ within data matrix **X** onto a score vector $\mathbf{u}_a$ within score matrix **U.** Store the regressed slope coefficients in the weight vector $\mathbf{w}_a$. Large weight coefficients reflect that columns in $\mathbf{X}_a$ are strongly correlated with $\mathbf{u}_a$. We do this regression as follows:

$$\mathbf{w}_a = (1/\ \mathbf{u}_a'\mathbf{u}_a)\ \mathbf{Xa}'\ \mathbf{u}_a \qquad (9.25)$$

Step 2. Normalize the weight vector to unit length.

$$\mathbf{w}_a = \mathbf{w}_a /\ (\mathbf{w}_a'\mathbf{w}_a)^{1/2} \qquad (9.26)$$

Step 3. Regress every row in $\mathbf{X}_a$ onto the weight vector $\mathbf{w}_a$. Store the regressed slope coefficients in the score vector $\mathbf{t}_a$. Repeat doing this for all J rows of observations:

$$\mathbf{t}_a = (1/\ \mathbf{w}_a'\mathbf{w}_a)\ \mathbf{X}_a'\ \mathbf{w}_a \qquad (9.27)$$

Step 4. Regress every column in $\mathbf{Y}_a$ onto the vector $\mathbf{t}_a$. Store the regressed slope coefficient in the loading vector $\mathbf{c}_a$. Repeat doing this for all M columns of quality variables:

$$\mathbf{c}_a = (1/\ \mathbf{c}_a'\mathbf{c}_a)\ \mathbf{Y}_a'\ \mathbf{c}_a \qquad (9.28)$$

Step 5. Regress each of the J rows in quality matrix $\mathbf{Y}_\mathbf{a}$ onto to the weight vector $\mathbf{c}_a$. Large weight coefficients indicate rows in $\mathbf{Y}_a$ are strongly correlated with $\mathbf{c}_a$.

$$\mathbf{u}_a = (1/\ \mathbf{c}_a'\mathbf{c}_a)\ \mathbf{Y}_a'\ \mathbf{c}_a \qquad (9.29)$$

The NIPLAS algorithm then continues with a procedure called "*deflation*" to remove variability already explained in $\mathbf{X}_a$ and $\mathbf{Y}_a$. This involve two steps.

Deflation Step 1. Calculate a loading vector for the X space.

We calculate a loading vector $\mathbf{p}_a$ using the X-space scores, following Eqs. (9.7) - (9.8):

$$\mathbf{p}_a = (1/\ \mathbf{t}_a'\mathbf{t}_a)\ \mathbf{x}_a'\ \mathbf{t}_a \tag{9.30}$$

Here, the score vector $\mathbf{t}_a$ is normalized. This loading vector $\mathbf{p}_a$ contains the regression slope of every column in **X**a onto the score vector $\mathbf{t}_a$. In this regression, the score vector $\mathbf{t}_a$ is the x-variable, the column from **X**a is the y-variable.

Deflation Step 2. Remove the predicted variability from X and Y.

Using the score vector $\mathbf{t}_a$ and the loading vector $\mathbf{p}_a$, we follow Eq. (9.19) to calculate the predicted value of $\mathbf{X}_a$, denoted by $\widehat{\boldsymbol{X}}_{\boldsymbol{a}}$:

$$\widehat{\boldsymbol{X}}_{\boldsymbol{a}} = \mathbf{t}_a\ \mathbf{p'}_a \tag{9.31}$$

We then remove this best prediction $\widehat{\boldsymbol{X}}_{\boldsymbol{a}}$ from the $\mathbf{X}_a$, that is, we remove the variability already explained well from the original data matrix $\mathbf{X}_a$:

$$\mathbf{E}_a = \mathbf{X}_a - \widehat{\boldsymbol{X}}_{\boldsymbol{a}} = \mathbf{X}_a - \mathbf{t}_a\ \mathbf{p'}_a \tag{9.32}$$

We define the remaining data matrix as $\mathbf{X}_{a+1}$:

$$\mathbf{X}_{a+1} = \mathbf{E}_a \tag{9.33}$$

In the same way, we remove the variability from the quality data matrix **Y**, using the score vector $\mathbf{t}_a$ and the leading vector $\mathbf{c}_a$.

$$\widehat{\boldsymbol{Y}}_{\boldsymbol{a}} = \mathbf{t}_a\ \mathbf{c'}_a \tag{9.34}$$

$$\mathbf{F}_a = \mathbf{Y}_a - \widehat{\boldsymbol{Y}}_{\boldsymbol{a}} = \mathbf{Y}_a - \mathbf{t}_a\ \mathbf{c'}_a \tag{9.35}$$

$$\mathbf{Y}_{a+1} = \mathbf{F}_a \tag{9.36}$$

The NIPALS algorithm repeats all over again using the deflated matrices for the subsequent iterations.

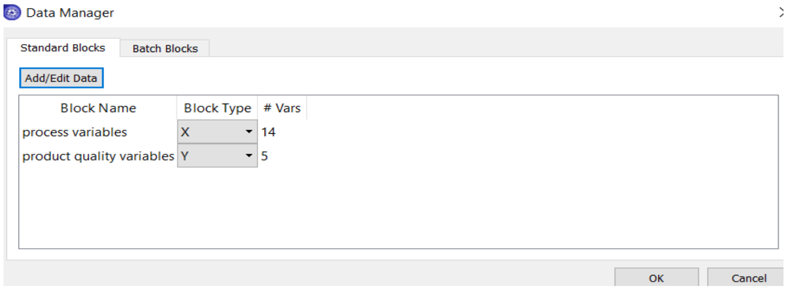
In Appendix B of this book, code B.5 and Table B.1 at the end give the Python implementation of the PLS algorithm, together with a list of common parameters and their suggested values.

9.4 Hands-on Workshops of PLS of LDPE and HDPE Processes

9.4.1 Workshop 9.2: PLS of Process and Quality Variables Affecting the Quality and Conversion of LDPE Product from a Two-Zone Tubular Reactor

The procedure to carry out this workshop is similar to **WS9.1_PCA-X** in Section 9.2. We only highlight the changes to the previous workshop when considering both X and Y spaces (i.e., process and quality variables).

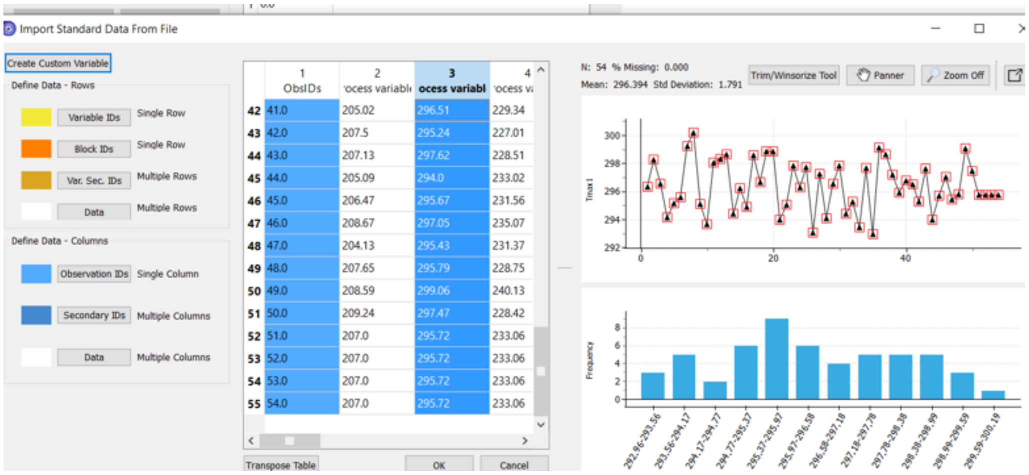
We follow Figure 9.11 to start a new project, and follow Figure 9.12 to import data from file, ***LDPE.xls***, *but to choose both process variables and product quality variables*. We change Figure 9.14 to Figure 9.32.



Block Name	Block Type	# Vars
process variables	X	14
product quality variables	Y	5

Figure 9.32 Specify block types for process variables, X, and product quality variables, Y.

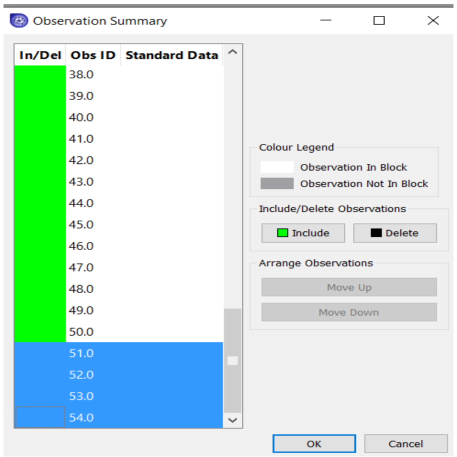
After importing the data file, ***LDPE.xls***, we note that the original reference [18] for the LPDE data indicates that some observations, such as observation IDs from 51 to 54, reflect a gradual increasing level of impurities in the feed ethylene to both zones of the tubular reactor, and they progressively move outside the acceptable region. Additionally, the values for observation IDs from 51 to 54 do not change for most of the 15 process variables. See Figure 9.33, in which the data plot on the right results from our highlighting process variable 3 on the left. Next, we see an observation summary in Figure 9.34 (similar to Figire 9.15), in which we delete observations 51 to 54.



	1 ObsIDs	2 'ocess variabl	3 ocess variabl	4 'ocess v
42	41.0	205.02	296.51	229.34
43	42.0	207.5	295.24	227.01
44	43.0	207.13	297.62	228.51
45	44.0	205.09	294.0	233.02
46	45.0	206.47	295.67	231.56
47	46.0	208.67	297.05	235.07
48	47.0	204.13	295.43	231.37
49	48.0	207.65	295.79	228.75
50	49.0	208.59	299.06	240.13
51	50.0	209.24	297.47	228.42
52	51.0	207.0	295.72	233.06
53	52.0	207.0	295.72	233.06
54	53.0	207.0	295.72	233.06
55	54.0	207.0	295.72	233.06

N: 54 % Missing: 0.000
Mean: 296.394 Std Deviation: 1.791

Figure 9.33 A display of observed values for process variable 3.



Observation Summary

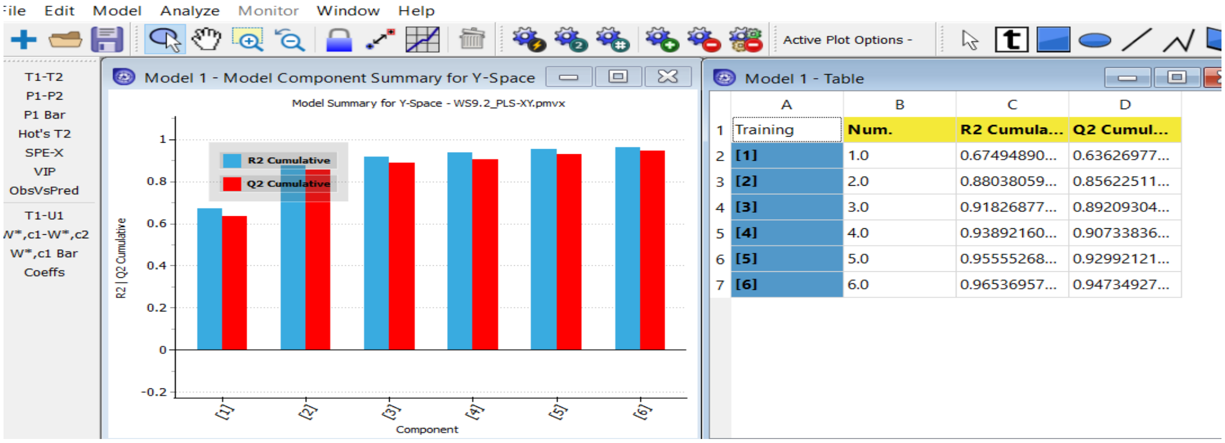
In/Del	Obs ID	Standard Data
🟩	38.0	
🟩	39.0	
🟩	40.0	
🟩	41.0	
🟩	42.0	
🟩	43.0	
🟩	44.0	
🟩	45.0	
🟩	46.0	
🟩	47.0	
🟩	48.0	
🟩	49.0	
🟩	50.0	
	51.0	
	52.0	
	53.0	
	54.0	

Figure 9.34 Delete observations 51 to 54

We save the project file as ***WS9.2_PLS-XY.pmvx***. In the following, we demonstrate the PLS model plots for the quality variable or Y-space, focusing on those new plots that we did not illustrate in Section 9.2, Workshop 9.1 with the PCA model.

(1) PLS model for the Y-space

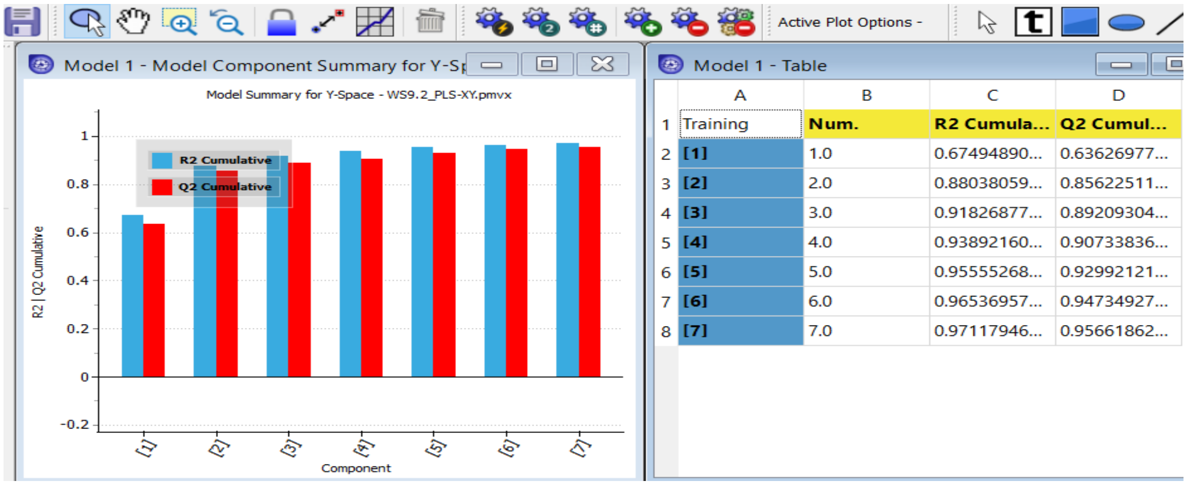
We follow the path: Model -> Active Model ->Auto Fit (follow Figure 9.20), and see the resulting model of Figure 9.35, displaying the R2 and Q2 values, Eq. (9.19) and (9.20), versus the number of principal components. As demonstrated in Figure 9.21, we can right-mouse-click on the R2-Q2 plot and select "Create Table" to see a table of R2 and Q2 values in the plot. An R2 value of 0.9654 indicates that the PLS model for Y-space will explain 96.54% of the variability of the dataset for product quality variables with 6 principal components. A Q2 value of 0.9474 says that with cross validation, the model can explain 94.74% of the variability of the dataset for product quality variables.



	A	B	C	D
1	Training	Num.	R2 Cumula...	Q2 Cumul...
2	[1]	1.0	0.67494890...	0.63626977...
3	[2]	2.0	0.88038059...	0.85622511...
4	[3]	3.0	0.91826877...	0.89209304...
5	[4]	4.0	0.93892160...	0.90733836...
6	[5]	5.0	0.95555268...	0.92992121...
7	[6]	6.0	0.96536957...	0.94734927...

Figure 9.35 R2 and Q2 values of PLS for Y-space with six principal components resulting from auto fit

Following Figure 9.19 and choosing the # button on the top of the screen and filling in a maximum number of 7 principal components, we get a R2-Q2 plot of Figure 9.36. It appears that adding one principal component increases the R2 value from 0.9654 to 0.9712, and Q2 values from 0.9473 to 0.9566. We will use 7 principal components in the example below.

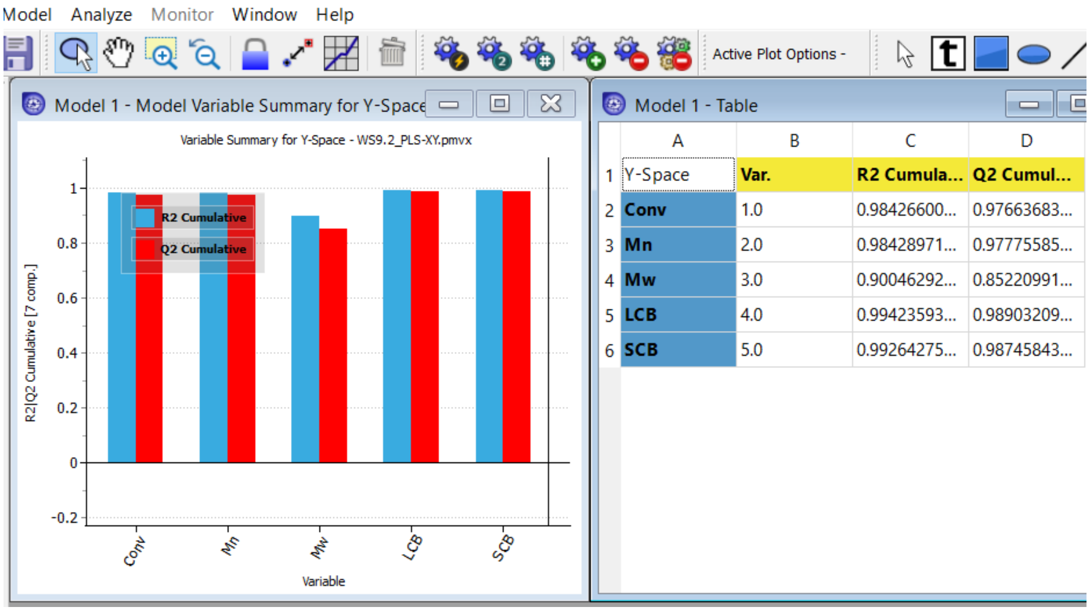


	A	B	C	D
1	Training	Num.	R2 Cumula...	Q2 Cumul...
2	[1]	1.0	0.67494890...	0.63626977...
3	[2]	2.0	0.88038059...	0.85622511...
4	[3]	3.0	0.91826877...	0.89209304...
5	[4]	4.0	0.93892160...	0.90733836...
6	[5]	5.0	0.95555268...	0.92992121...
7	[6]	6.0	0.96536957...	0.94734927...
8	[7]	7.0	0.97117946...	0.95661862...

Figure 9.36 R2 and Q2 values of PLS for Y-space with seven principal components

(2) Model variable summary for the Y-space

Next, we follow the path: Analyze ->Model -> Variable Summary -> Choose Block: y-space, and Component: 7. We see in Figure 9.37 that except for Mw (or MWW) (weight-average molecular weight), the PLS model for Y-space predicts CONV, Mn (or MWN), LCB and SCB reliability with R2 values above 0.9842.



	A	B	C	D
1	Y-Space	Var.	R2 Cumula...	Q2 Cumul...
2	Conv	1.0	0.98426600...	0.97663683...
3	Mn	2.0	0.98428971...	0.97775585...
4	Mw	3.0	0.90046292...	0.85220991...
5	LCB	4.0	0.99423593...	0.98903209...
6	SCB	5.0	0.99264275...	0.98745843...

Figure 9.37 R2 and Q2 values of PLS for Y-space with seven principal components

(3) T1-T2 score plot and W*C[1] vs W*C[2] loading plot

The T1-T2 score plot used to identify clusters and outliers, previously shown in Figure 9.25, applies to both PCA model for the X-space and PLS model for the X and Y spaces. For the PLS model, it is best to use the W*C[1] vs W*C[2] loading plot, because it also explains the relationship between the X and Y variables.

By choosing W*,c1-W*,c2 button on the left side pane, we generate the preferred PLS loading plot on the right side of Figure 9.38. In the plot, we see 5 quality variables or Y variables in red, and 14 process variables or X variables in black. Referring to the variable definitions in Table 9.1, Section 9.2, we see that the quality variable SCB in red is positively correlated with process variables Fi1 and Tmax1; quality variables Mw, CONV and LCB in red are negatively correlated with process variables Fi2, Tout2 and Tmax2 in black; and quality variable Mn in red is also negatively correlated with process variables z1 and Tcin2 in black.

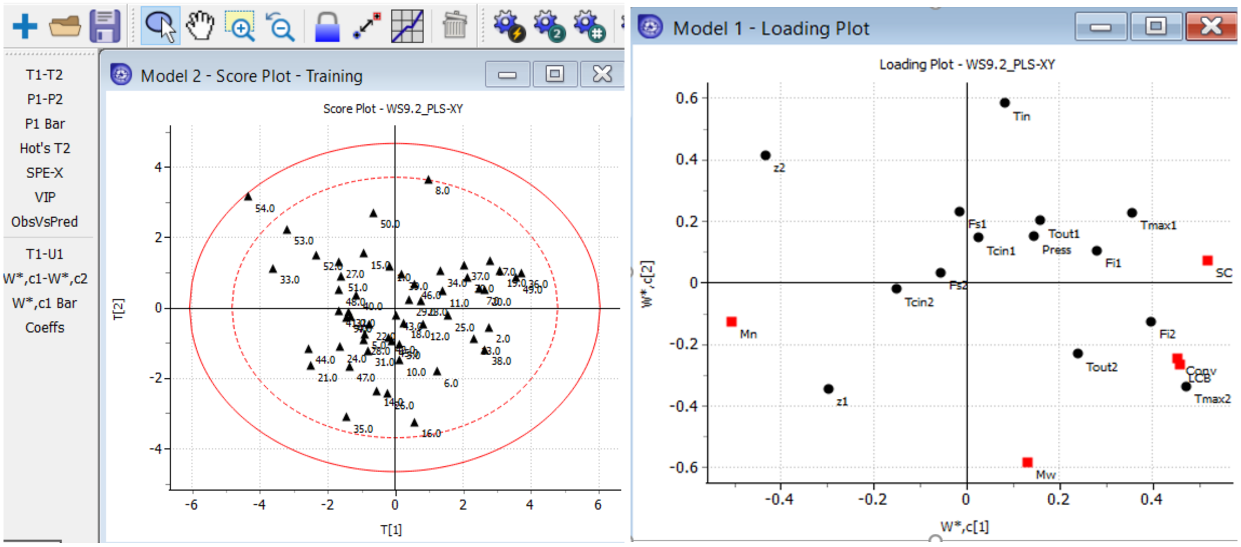


Figure 9.38 T[2] vs T[1] score plot and W*C[1] vs W*C[2] loading plot

(4) Loading bi-plot

A loading bi-plot super-imposes the loadings and scores, such as Figure 9.38 (left and right) for easier interpretation of the relationship between the variables and observations. Following the path, Analyze -> Loading bi-plot ->Worksets: training, Block: X-space; X-axis: component 1; Y-axis: component 2, we generate a loading bi-plot of Figure 9.39.

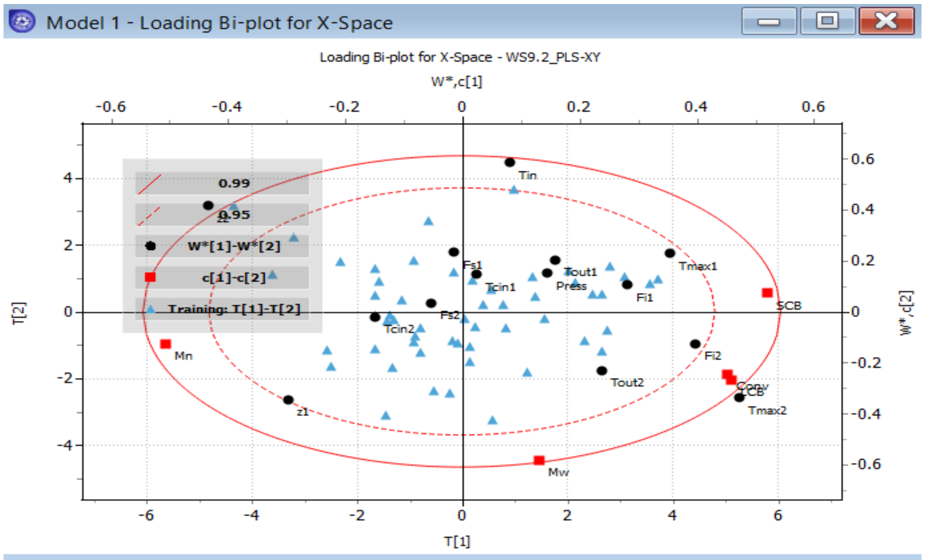
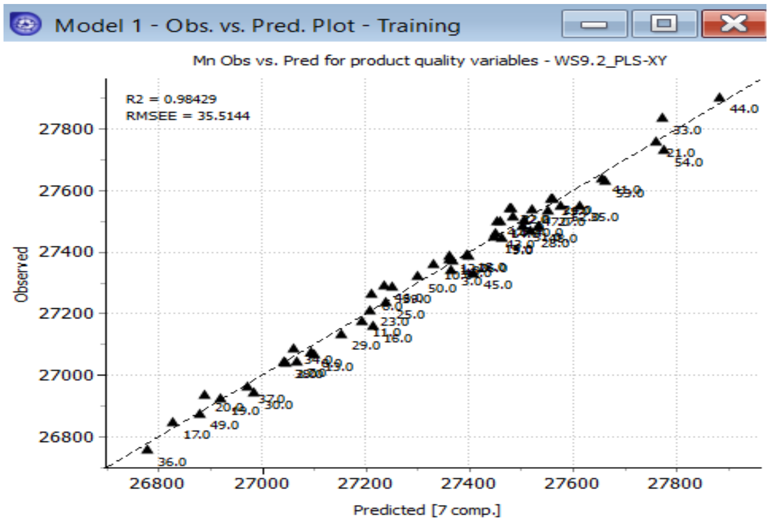


Figure 9.39 Loading bi-plot

(4) Obs vs pred plot

By choosing obs vs pred on the left-side pane and specifying the following: worksets- training; observation - 1; block - product quality variable; variable- Mn; component:7; raw units, we generate the Obs vs pred plot of Figure 9.41. The root-mean-squared-error (RMSE) of 35.5144 is only 0.013% of the average observed value of 27400.



Model 1 - Obs. vs. Pred. Plot - Training

Mn Obs vs. Pred for product quality variables - WS9.2_PLS-XY

R2 = 0.98429
RMSEE = 35.5144

Figure 9.41 Obs. Vs Pred. plot of product quality variable, Mn (number-average molecular weight)

This concludes workshop ***WS9.2_PLS-XY***, and we save the file as ***WS9.2_PLS-XY.pmvx***.

9.4.2. Workshop 9.3 - Polymer Melt Index Prediction and Causal Analysis Using PLS

The objective of this workshop is to demonstrate the application of PLS model for predicting the MI and causal analysis of a HDPE manufacturing process. We consider an industrial slurry HDPE process with two reactors in parallel with the model details as defined by Sharma and Liu [28,43,44], using plant data from LG Petrochemicals in South Korea [21]. See Figure 9.42.

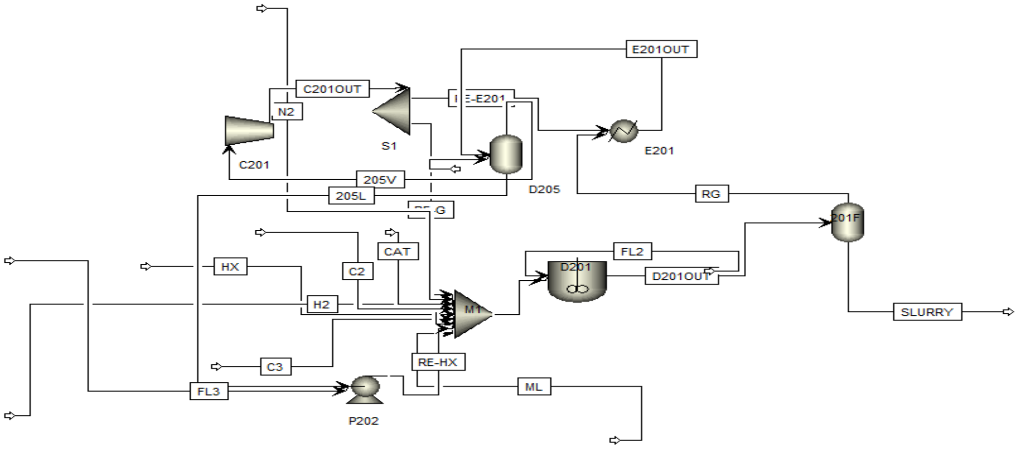


Figure 9.42 Process simulation flowsheet of an industrial parallel slurry HDPE process

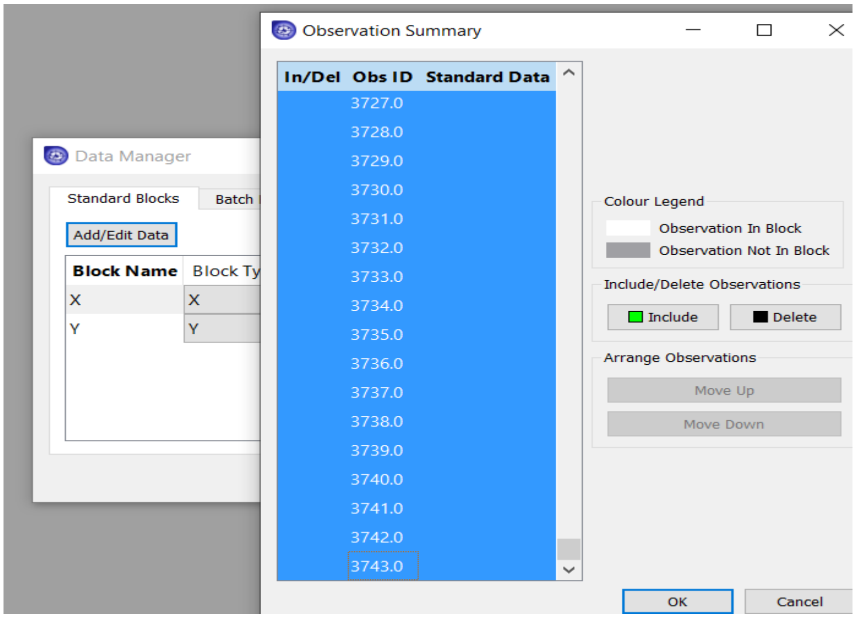
We convert a steady-state simulation model based on Aspen Plus to a dynamic (time-dependent) simulation model using Aspen Plus Dynamics. The resulting dynamic simulation model has similar independent variables as explained before. Both steady-state and dynamic simulation models are developed from first principles such as phase-equilibrium calculations and mass and energy balances. Therefore, they are scientifically consistent models.

Park et. al. [21] correlate the MI data by considering the independent variables shown in Table 9.2. The dataset consists of 5000 observations and 9 main independent process variables and 1 dependent variable, MI, as the quality target. We first make sure the data are in Excel format and the process variable (X) and (Y) data are in different sheets within ***HDPE_XY Data.xlsx.***

Table 9.2 Process and quality variables of the parallel slurry HDPE process

Process and quality variables	Description
C2	Ethylene feed flow rate
H2	Hydrogen feed flow rate
CAT	Catalyst feed flow rate
HX	Hexane solvent feed flow rate
C3	Comonomer feed flow rate
T	Temperature of the reactor
P	Pressure in the reactor
H2/C2	Feed concentration ratio in the reactor of ethylene to hydrogen
C3/C4	Feed concentration ratio of Propylene to Butylene monomer
MI (quality variable)	Melt Index of polymer

We open a new project in Aspen ProMV. Following Figure 9.27 in Workshop 9.2, we import both process (X) and quality (Y) variable datasets, ***HDPE_XY Data.xlsx***, into Aspen ProMV. On the data manager, we choose the X block and highlight all “Obs ID” for X vailables to “Include” all X obsrvations (see Figure 9.43). Clicking on OK in “Observaton Summaey” leads to the “New Model” screen, and choosing the X block generates the details of 9 process variables, including their mean, standard deviation, min/max value, etc. (see Figure 9.44). Likewise, choosing the Y block shows the details of the single quality vailable, MI (see Figre 9.45). We then name the new model as ***WS9.3_PLS-XY.pmvx.***


Data Manager

Standard Blocks | Batch

Add/Edit Data

Block Name	Block Ty
X	X
Y	Y

Observation Summary

In/Del	Obs ID	Standard Data
	3727.0	
	3728.0	
	3729.0	
	3730.0	
	3731.0	
	3732.0	
	3733.0	
	3734.0	
	3735.0	
	3736.0	
	3737.0	
	3738.0	
	3739.0	
	3740.0	
	3741.0	
	3742.0	
	3743.0	

Colour Legend

Observation In Block

▨ Observation Not In Block

Include/Delete Observations

🟩 Include ■ Delete

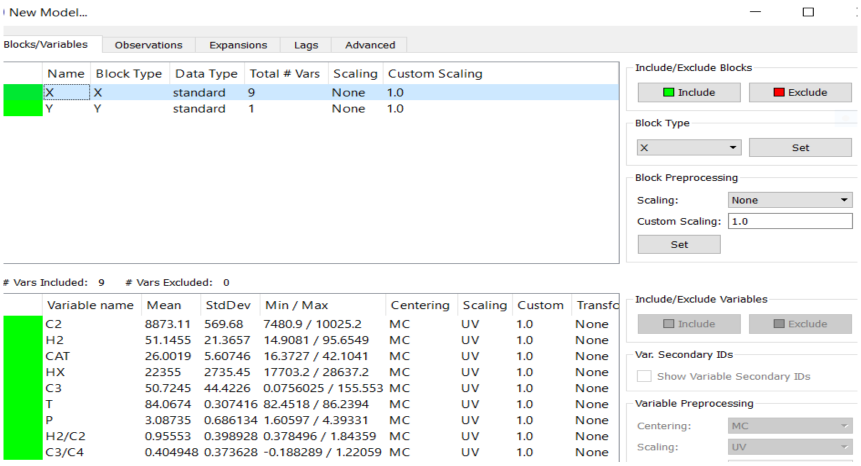
Arrange Observations

Move Up

Move Down

OK Cancel

Figue 9.43 Highlighting all X variable Obs ID in “Observation Summary” to include all observations


New Model...

Blocks/Variables | Observations | Expansions | Lags | Advanced

	Name	Block Type	Data Type	Total # Vars	Scaling	Custom Scaling
🟩	X	X	standard	9	None	1.0
🟩	Y	Y	standard	1	None	1.0

Include/Exclude Blocks: 🟩 Include 🟥 Exclude

Block Type: X Set

Block Preprocessing

Scaling: None

Custom Scaling: 1.0

Set

Vars Included: 9 # Vars Excluded: 0

	Variable name	Mean	StdDev	Min / Max	Centering	Scaling	Custom	Transfo
🟩	C2	8873.11	569.68	7480.9 / 10025.2	MC	UV	1.0	None
🟩	H2	51.1455	21.3657	14.9081 / 95.6549	MC	UV	1.0	None
🟩	CAT	26.0019	5.60746	16.3727 / 42.1041	MC	UV	1.0	None
🟩	HX	22355	2735.45	17703.2 / 28637.2	MC	UV	1.0	None
🟩	C3	50.7245	44.4226	0.0756025 / 155.553	MC	UV	1.0	None
🟩	T	84.0674	0.307416	82.4518 / 86.2394	MC	UV	1.0	None
🟩	P	3.08735	0.686134	1.60597 / 4.39331	MC	UV	1.0	None
🟩	H2/C2	0.95553	0.398928	0.378496 / 1.84359	MC	UV	1.0	None
🟩	C3/C4	0.404948	0.373628	-0.188289 / 1.22059	MC	UV	1.0	None

Include/Exclude Variables: Include Exclude

Var. Secondary IDs

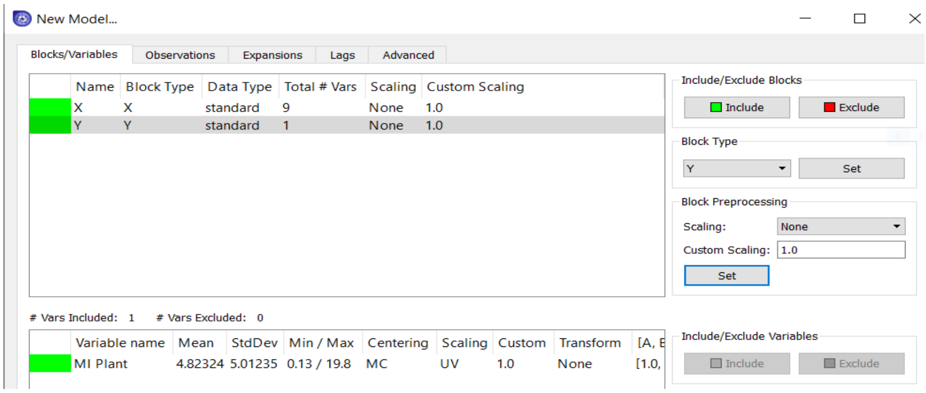
□ Show Variable Secondary IDs

Variable Preprocessing

Centering: MC

Scaling: UV

Figure 9.44 Process variable details in the new model.


New Model...

Blocks/Variables | Observations | Expansions | Lags | Advanced

	Name	Block Type	Data Type	Total # Vars	Scaling	Custom Scaling
🟩	X	X	standard	9	None	1.0
🟩	Y	Y	standard	1	None	1.0

Include/Exclude Blocks: 🟩 Include 🟥 Exclude

Block Type: Y Set

Block Preprocessing

Scaling: None

Custom Scaling: 1.0

Set

Vars Included: 1 # Vars Excluded: 0

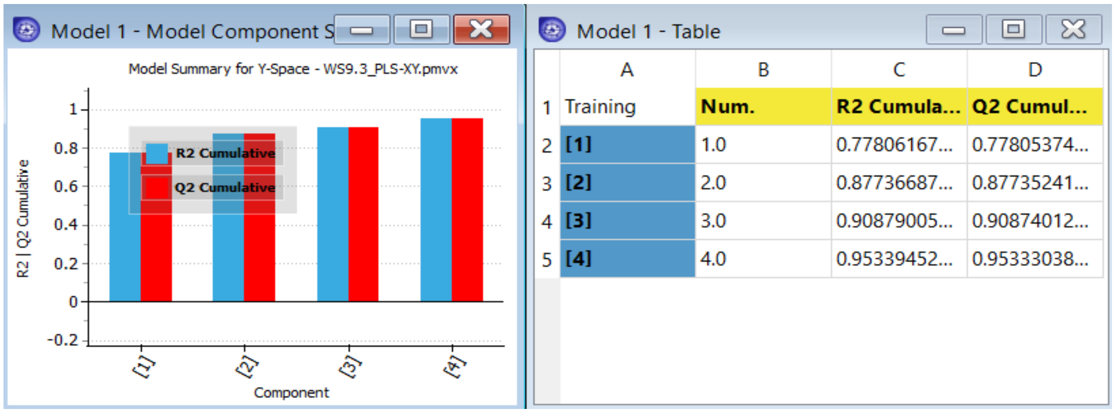
	Variable name	Mean	StdDev	Min / Max	Centering	Scaling	Custom	Transform	[A, E
🟩	MI Plant	4.82324	5.01235	0.13 / 19.8	MC	UV	1.0	None	[1.0,

Include/Exclude Variables: Include Exclude

Figure 9.45 Quality variable details in the new model.

(1) PLS Model for the Y-space

We follow the path: Model ->Active Model -> Auto Fit (follow Figure 9.20), and see the R2 and Q2 values of the resulting model in Figure 9.46. The figure shows that with four principal components, an R2 value of 0.9534 says that the PLS model can explain 95.34% of the variability of the product quality variable, the melt index (MI); a Q2 value of 0.9533 means that with cross validation, the PLS model can explain 95.33% of the data variability.



	A	B	C	D
1	Training	Num.	R2 Cumula...	Q2 Cumul...
2	[1]	1.0	0.77806167...	0.77805374...
3	[2]	2.0	0.87736687...	0.87735241...
4	[3]	3.0	0.90879005...	0.90874012...
5	[4]	4.0	0.95339452...	0.95333038...

Figure 9.46 R2 and Q2 values of PLS for Y-space with four principal components resulting from auto fit.

(2) Obs vs Pred plot

Following Figure 9.41, we generate a Obs. Vs Pred plot in Figure 9.47. The root-mean-squared-error (RMSEE) is 1.08266.

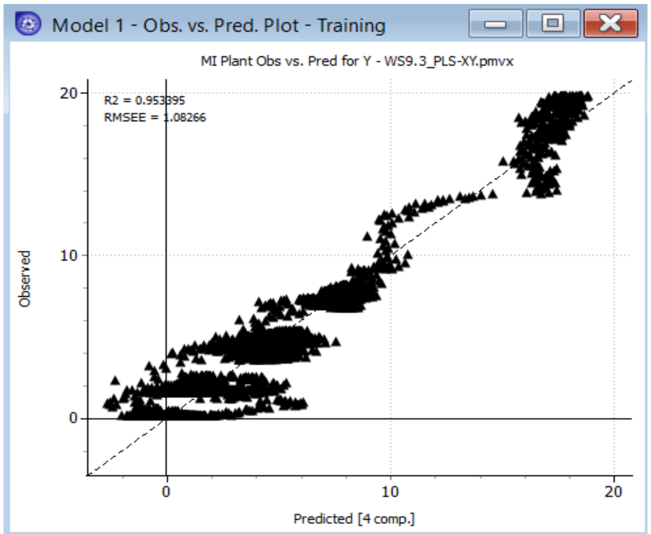


Figure 9.47 The Obs vs pred plot with 4 principal components

(3) Loading bi-plot and VIP plot

Following Figure 9.39, we show a loading bi-plot, super-imposes the T[2] vs T[1] score plot and W*c[1] vs W*c[2] plot in Figure 9.48.

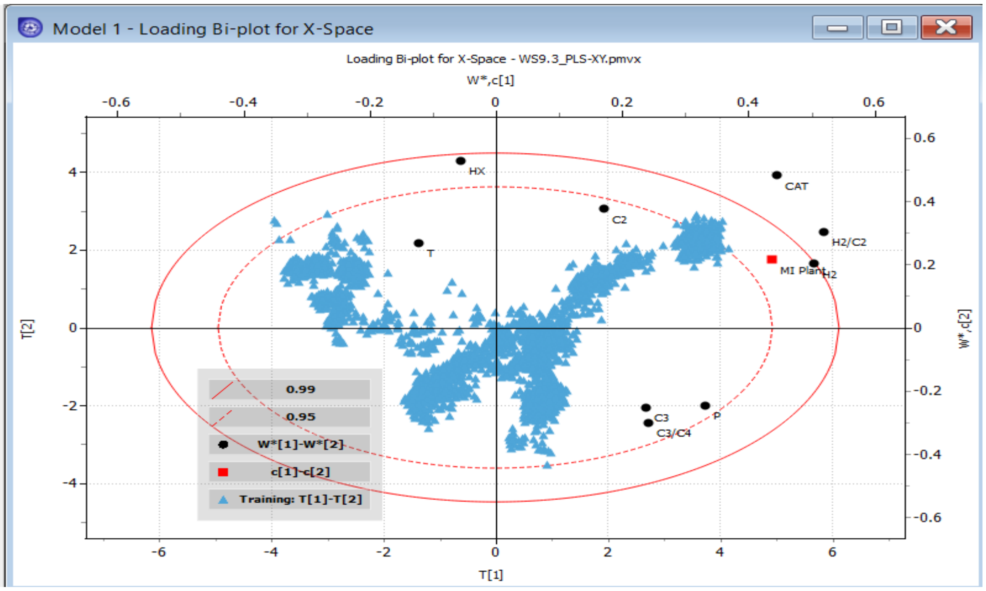


Figure 9.48 Loading bi-plot

We see that from the scores, T[2] vs T[1], process variables CAT and H2/C2 in black are both outside the 99% confidence limit, and are potentially outliers. Additionally, quality variable, MI Plant, in red is positively correlated with process variables H2, H2/C2 and CAT in black (since they lie nearer to each other). We can confirm this strong correlation by selecting the VIP button on the left side pane to generate a variable importance to projection plot, VIP, in Figure 9.49.

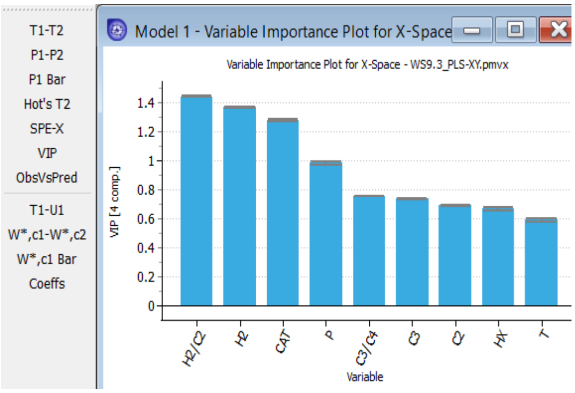


Figure 9.49 Variable importance to projection, VIP plot

(4) Hotelling's T^2 plot

Following Figure 9.26, we show the Hotelling's T^2 plot in Figure 9.50, and want to demonstrate new tools to identify the cause of a selected outlier in our dataset. We right-click within the plot to show the menu to display observation number. We click on "Display Point Tooltips", and then put the mouse on one of the outliers. We see in Figure 9.51 the observation number as 2697.

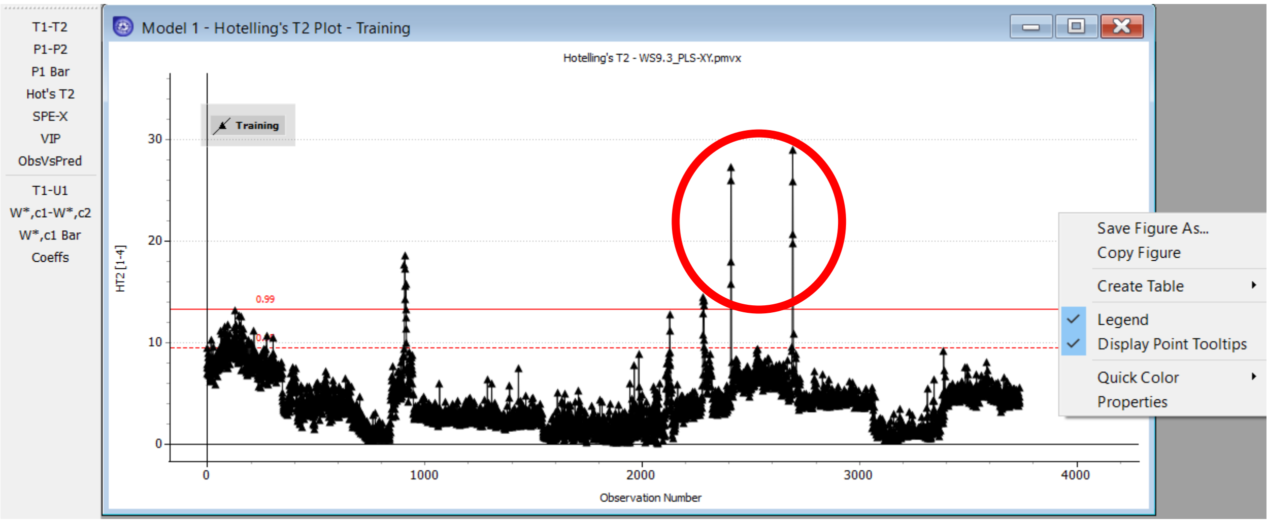


Figure 9.50 The Hotelling's plot and the menu to display observation number., "Display Point Tooltips". The data points within the red circle represent potential outliers outside the 95% confidence limit.

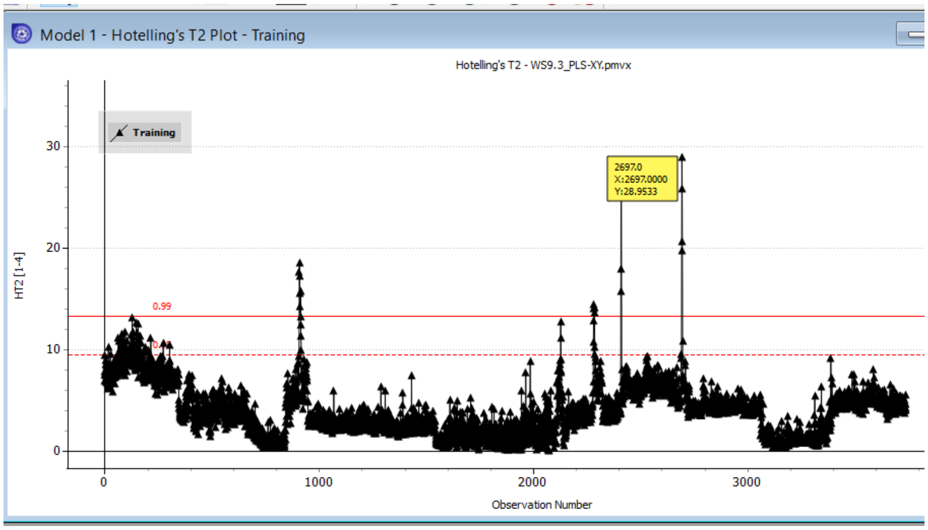
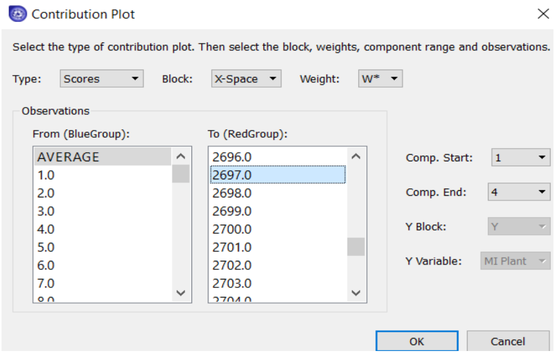


Figure 9.51 Displaying data number 2697 for an outlier located on the far right, top data point

How do we identify the cause of data number 2697 as an outlier? We use the contribution plot below.

(4) Contribution plot

We follow the path: Analyze -> Contributions -> Specify according to Figure 9.52 -> Contribution plot of Figure 9.53.



Contribution Plot

Select the type of contribution plot. Then select the block, weights, component range and observations.

Type: Scores Block: X-Space Weight: W*

Observations

From (BlueGroup):
AVERAGE
1.0
2.0
3.0
4.0
5.0
6.0
7.0

To (RedGroup):
2696.0
2697.0
2698.0
2699.0
2700.0
2701.0
2702.0
2703.0

Comp. Start: 1

Comp. End: 4

Y Block: Y

Y Variable: MI Plant

Figure 9.52 Specifying a contribution plot from average to data point 2697.

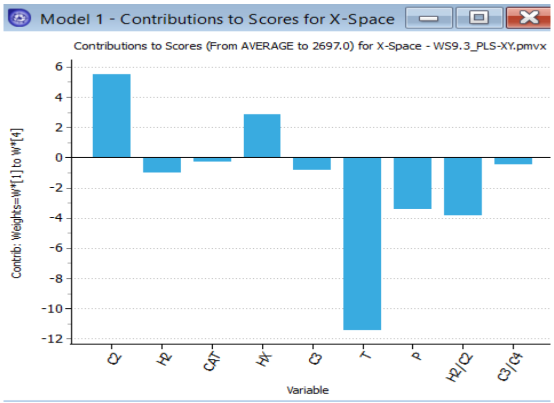


Figure 9.53 Contribution plot indicating temperature of data pint 2697 being much lower than the average value, causing an outlier in the Hotelling's T^2 plot

Next, we start a new project, import the data file, ***HDPX_XY_Data.xlsx***, again. We then follow Figures 9.15-9.16, and Figure 9.34, to remove observation IDs 2412 to 2415, and 2695 to 2698 (potential outliers highlighted within the red circle in Figure 9.50), and save the resulting model file as ***WS9.3-1.PLS-XY.pmvx***. Following the path: Model -> Active Model -> Auto Fit, we generate the model resulting from removing observation IDs 2412 to 2415, and 2695 to 2698. Figure 9.54 shows the corresponding R2-Q2 plot and the Obs vs Pred plot.

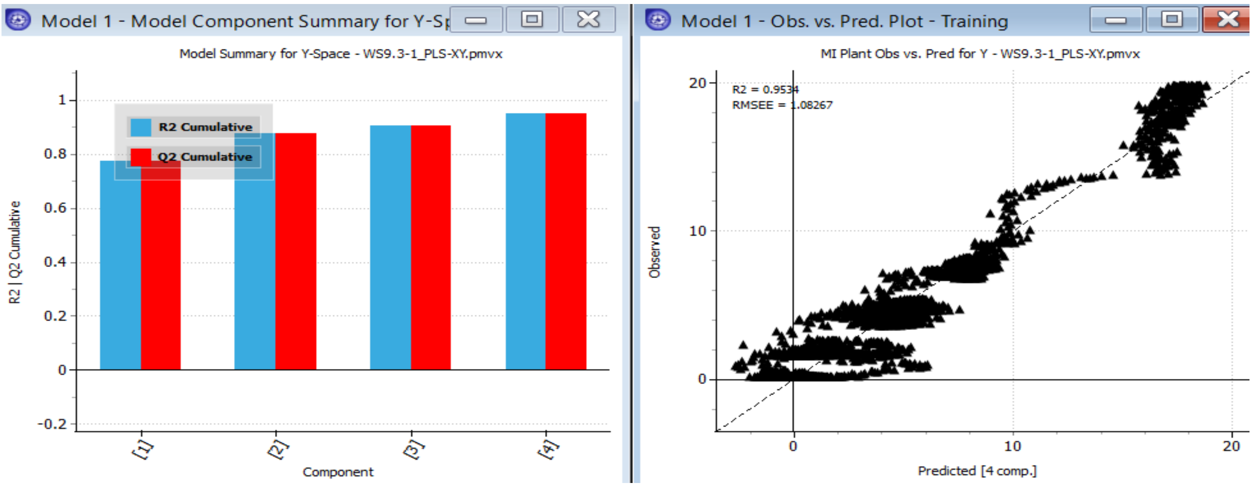


Figure 9.54 The R2-Q2 plot and Obs vs Pred Plot after removing potential outliers.

Comparing Figures 9.47 and 9.54, we find the change of R2 from 0.953395 to 0.9534, and RMSE from 1.08266 to 1.08267 quite insignificant. Therefore, we can stay with the original model. This concludes workshop ***WS9.2_PLS-XY***, and we save the file as ***WS9.2_PLS-XY.pmvx***.

9.5. Workshop 9.4 - Polymer Melt Index Prediction and Causal Analysis with Measurement Time Lags Using PLS

9.5.1 Introduction to PLS with Measurement Time Lags

In many chemical processes, there is some lag between the time when the quality variable like MI at the process outlet is measured and the process variable measurements. The output in a dynamic process is related to the past process variable inputs and past outputs as well. To handle the autocorrelation data, we mimic the concept of auto-regressive moving average exogenous (ARMAX) time series models by forming the data matrix with previous observation in each observation vector. The time series model which relates quality (dependent) variable y at present time to past quality variable y's and process (independent) variable x's.

The model equation is represented below:

$$\mathrm{y_t} = \beta_1 y_{t-1} + \ \beta_2 y_{t-2} + \cdots + \gamma_1 x_{t-1} + \gamma_2 x_{t-2} + e_t \qquad (9.37)$$

This eventually means that we need to use a lagged value of the quality variable to account for the time lags. Thus, we consider the autocorrelation in the data in ProMV by introduction of the lag of variable order. This time series modeling technique is also referred as PLS with observation time lags. When applied the technique to batch processes with time lags, Chen and Liu [21] refer the method as batch dynamic partial least squares (BDPLS).

When a quality (Y) variable in a PLS model contains measurement time lags, we introduce a lagged quality variable to the Y block to which it belongs. Following Aspen ProMV online help, we show in Figure 9.55 an example of a Y data block with a single quality variable that is lagged three time units. In the figure, we add three lagged quality variables. The resulting quality data block with lags (called LagsY block) now has three more variables due to time lags, but three fewer observations. We define a lagged variable with the original name with the suffix _L#, where # represents the lag value for that particular value.

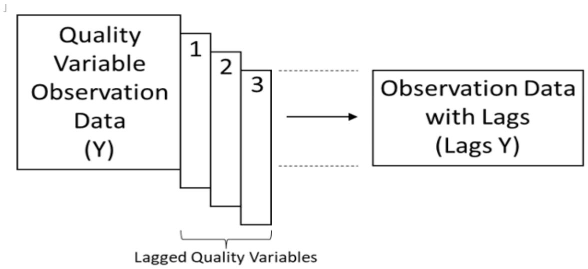


Figure 9.55 An illustration of a single quality variable that is lagged three time units.

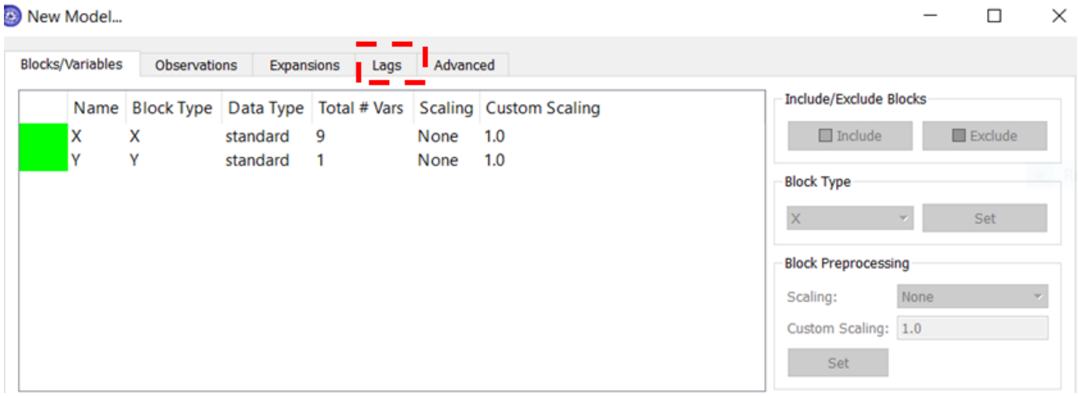
In the following, we demonstrate how to apply PLS with observation time lags using Aspen ProMV.

9.5.2 Workshop 9.4 - Application of Aspen ProMV to Polymer Melt Index Prediction and Causal Analysis with Measurement Time Lags Using PLS

We use the same industrial HDPE process in Workshop 9.3 in Section 9.4.2, and the same industrial dataset, ***HDPE_XY_Data.xlsx***.

We load the data using the same procedure. In this case we introduce a lag of order 1 in both the input process variables and the process output MI, so that the MI at the current time is function of the historical value of process variables and past MI value.

We follow the steps from Figures 9.12 to 9.18, but import both process and quality variable data (X and Y spaces) as in Figure 9.32, and save the file as ***WS9.4_PLS-X and LagsY.pmvx***. In the New Model screen, we pay attention to the "Lags" button. See Figure 9.56.



	Name	Block Type	Data Type	Total # Vars	Scaling	Custom Scaling
🟩	X	X	standard	9	None	1.0
🟩	Y	Y	standard	1	None	1.0

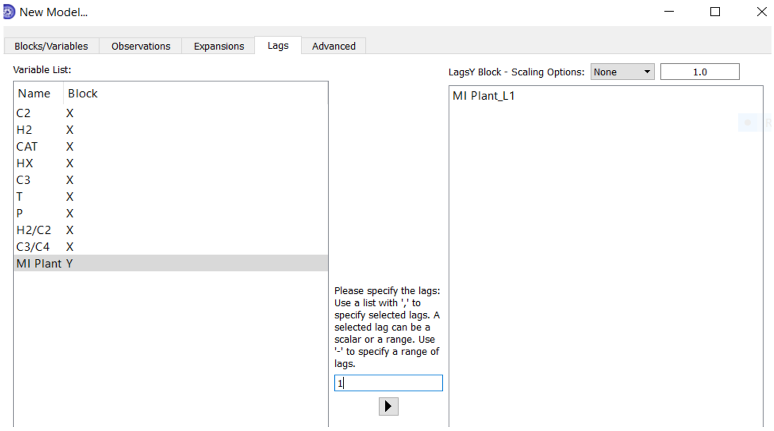
Block Type: X

Scaling: None

Custom Scaling: 1.0

Figure 9.56 The "Lags" button in the New Model screen

Referring to Figure 9.57, we choose quality variable, Plant MI, specify a lag of 1 time unit, and use the arrow key to move the data to the LagsY block on the right. We then save the model file as ***WS9.4_PLS-X and LagsY.pmvx.***



Variable List:

Name	Block
C2	X
H2	X
CAT	X
HX	X
C3	X
T	X
P	X
H2/C2	X
C3/C4	X
MI Plant	Y

LagsY Block - Scaling Options: None 1.0

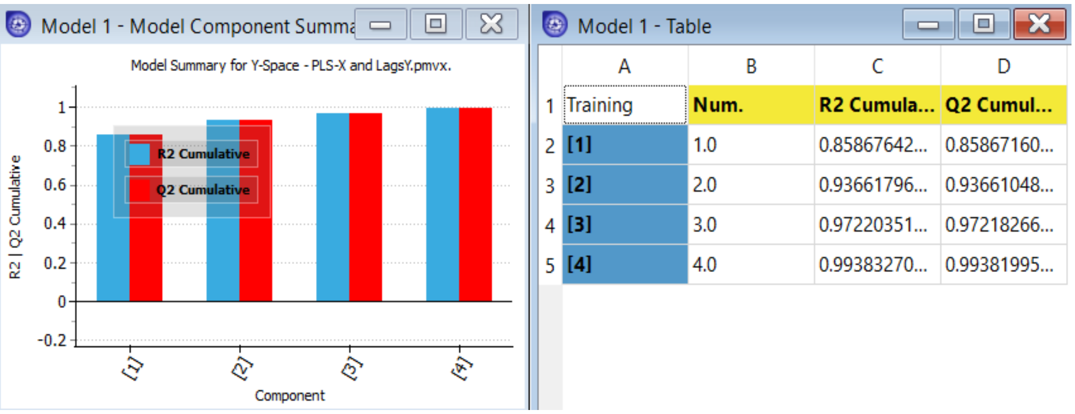
MI Plant_L1

Please specify the lags: Use a list with ',' to specify selected lags. A selected lag can be a scalar or a range. Use '-' to specify a range of lags.

1

Figure 9.57 Specifying a lag of one time-unit for the quality variable, MI Plant.
The new variable is named MI Plant_L1.

We build a PLS model following the path: Model ->Active Model ->Auto Fit (Figure 9.20) and see the resulting PLS model with time lag in Figure 9.58. An R2 value of 0.9938 says that the PLS model with time lag can explain 99.38% of the variability of the quality variable, Plant MI (melt index); a Q2 value of 0.9938 says that with cross validation, the model can explain 99.38% of the data variability. From Figure 9.46, we see that the corresponding R2 and Q2 values without time lag are 0.9534 and 0.9533, respectively. This comparison shows that by introducing the time lag, both R2 and Q2 values increases significantly when compared to those values without time lag.



	A	B	C	D
1	Training	Num.	R2 Cumula...	Q2 Cumul...
2	[1]	1.0	0.85867642...	0.85867160...
3	[2]	2.0	0.93661796...	0.93661048...
4	[3]	3.0	0.97220351...	0.97218266...
5	[4]	4.0	0.99383270...	0.99381995...

Figure 9.58 R2 and Q2 values of PLS for Y-space with time lag with four principal components resulting from auto fit.

Following Figures 9.41 and 9.47, we generate an Obs vs Pred plot in Figure 9.59. It is significant to note that by adding a time lag, the PLS model significantly lowers the RMSEE value from 1.08266 without time lag (Figure9.47) to 0.393567 with time lag (Figure 9.59).

Figure 9.60 shows a VIP plot for the PLS model with time lag. By comparing this plot with the corresponding VIP plot without time lag, Figure 9.49, we see that the lagged quality variable, Plant MI, becomes the most important variable for the PLS model with time lag.

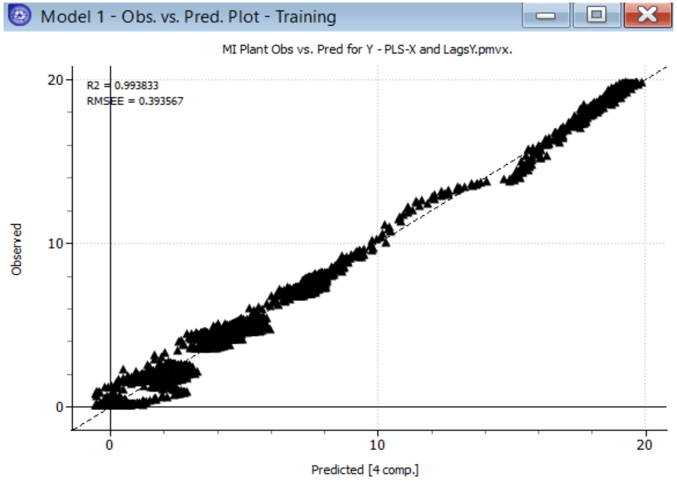


Figure 9.59 The Obs vs Pred plot with 4 principal components and with time lag

Thus, we can actually use the data from PLS model and separately plot the results with the actual plant data. Figure 9.60 demonstrates that predictions from a PLS model with measurement time lag compare well with the time-dependent plant MI data.

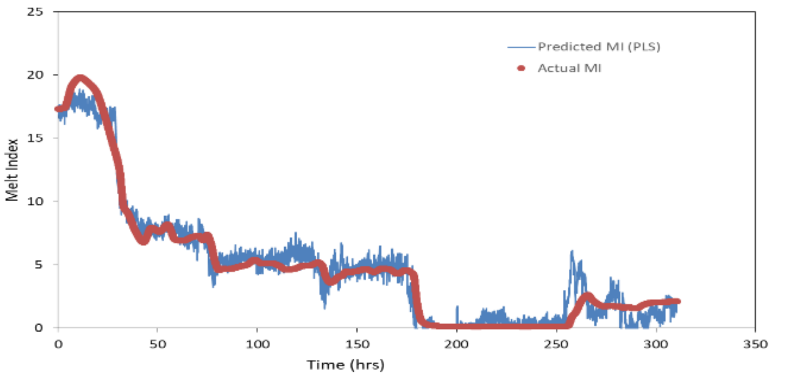


Figure 9.60. Development of a soft sensor of MI based on Causal PLS model

This concludes our workshop 9.4, and we save the resulting simulation file as ***WS9.4_PLS-X and LagsY.pmvx.***

9.6. Multiway PCA and PLS for Batch Processes

9.6.1 Batch-Wise Unfolding and Observation-Wise Unfolding Approaches to Multiway PLS

Our discussion of data analytics in the preceding sections has been mostly for continuous processes. For data analytics of batch processes, we require a different approach. Industrial batch process data with multiple batches have a three-dimensional structure with the three data dimensions, namely, process variables, time, and number of batches. Nomikos and Macgregor [22] explained those three data dimensions as an example of *a multiway approach* to multivariate data analytics, and they specifically demonstrated two approaches when applied to PCA or PLS.

The first approach is *the batch-wise unfolding (BWU) approach* that extracts the batch trajectory observations horizontally in a time-wise manner, as illustrated in Figure 9.61. Each batch becomes a single row of data. In the figure, we have a three-way array of trajectory data (X) of i = 1 to I (number of batches), j = 1 to J (number of process variables), and k = 1 to K (time step of data observation). We also append an initial condition matrix Z and a product quality matrix Y at time k =0 (beginning time) an k=K (ending time), respectively. In BWU, the data are unfolded into a two-way array X (I × J by K), where the rows of the unfolded matrix represent the batches. Each batch becomes a single row of data in the model.

We develop PLS models based on the unfolded data matrices. The BWU principal component score predicts the final state of each batch based on all the time history of that batch to the current time. The resulting principal component scores show differences among batches. The BWU approach is useful to predicting the final product quality, monitoring, control and optimization of batch processes.

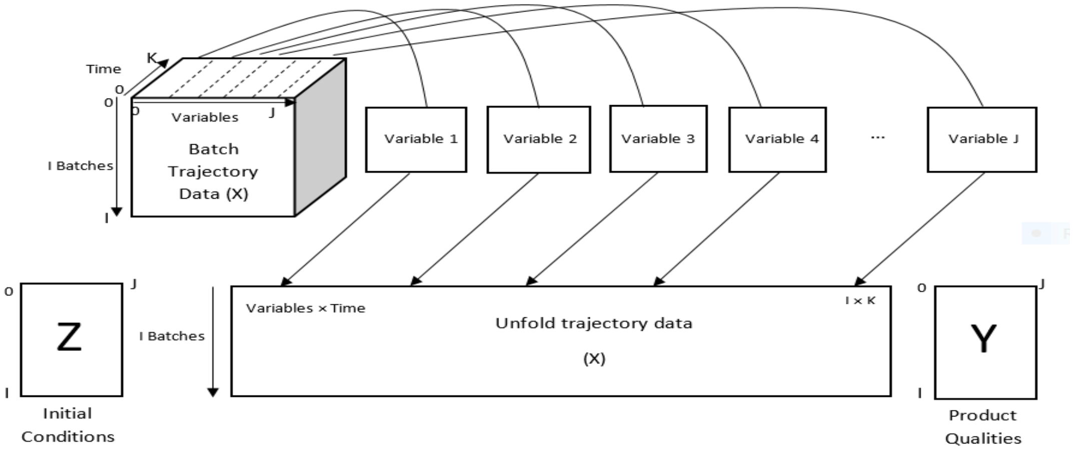


Figure 9.61 An illustration of batch-wide unfolding (BWU) of the 3-way (I x J x K) array of trajectory data (X) of i = 1 to I (number of batches), j = 1 to J (number of process variables), and k = 1 to K (time step of data observation).

Refer to our discussion of PLS in Section 9.3.1, particularly Eqs. (9.17), (9.23) and (9.24), and to Figure 9.30. We show in Figure 9.62 an extension of the PLS structural digram to batch-wise unfolding. In the figure, Z is the initial condition vector, X is the process data matrix, Y is the quality data matrix, T is the principal component score matrix, V^T is the initial condition vector, W is the weight matrix, and C^T is the principal component loading matrix. Based on Figure 9.30 and Eq. (9.24), we write the principal component score matrix T and the predicted product quality matrix $\widehat{\boldsymbol{Y}}$ as:

$$\mathbf{T} = \mathbf{X\ W} \tag{9.24}$$

$$\widehat{\boldsymbol{Y}} = \mathbf{T\ C^T = [X\ W\ ]C^T\ =\ X\ [WC^T] =\ X}\ \boldsymbol{\beta} \tag{9.24a}$$

where **β** is the regression coefficient matrix of PLS. We have a row of coefficients for each Y variable. These coefficients show the relative importance of the X's to each individual Y variable.

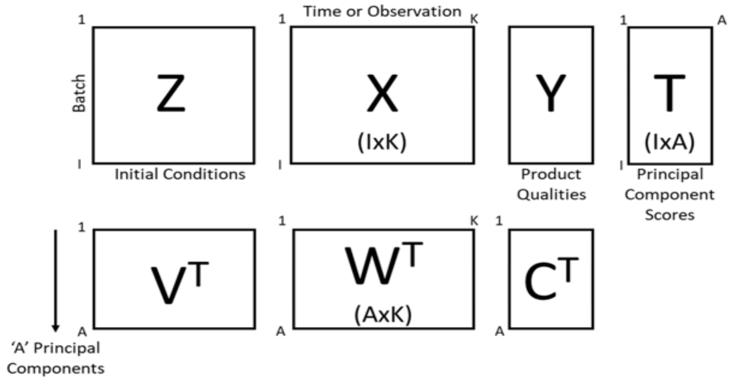


Figure 9.62 PLS of batch-wise unfolded data.

The second approach is *the observation-wide unfolding (OWU) approach* where the process data for each batch are stacked on top of one another, following the way we typically read batch data. The analysis will summarize the instantaneous condition of each batch using the measured values at the current time. The study by Nguyen et al. [30] compares the two batch folding techniques BWU and OWU for analysis of foaming in a fermentation process. Figure 9.63 illustrates the OWU approach, which is useful to reducing the dimension of data collected from batches.

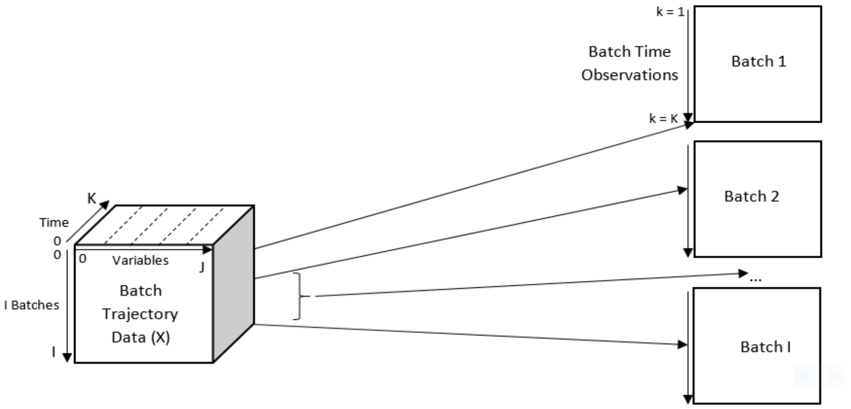
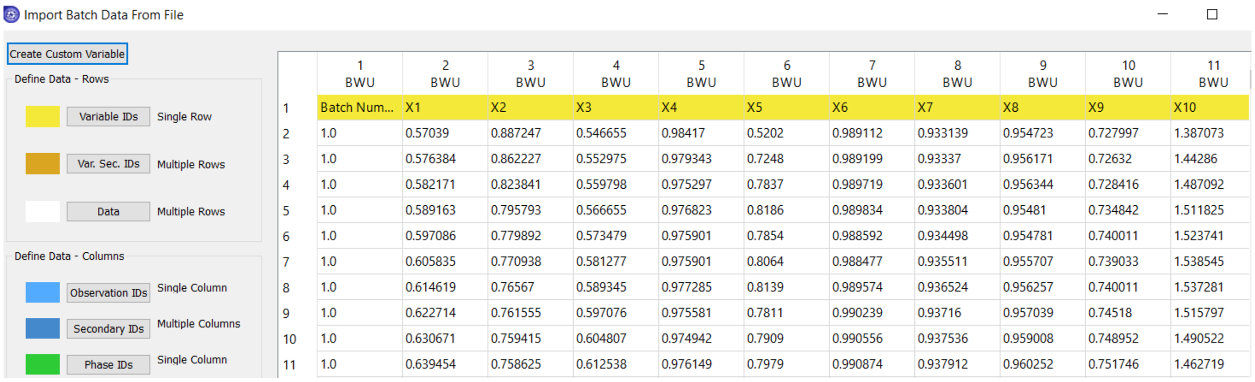


Figure 9.63 An illustration of observation-wise unfolding (OWU of of the 3-way (I x J x K) array of trajectory data (X) of i = 1 to I (number of batches), j = 1 to J (number of process variables), and k = 1 to K (time step of data observation).

9.6.2 Workshop 9.5 - Application of Aspen ProMV to Batch-Wise Unfolding (BWU) Approach to Multiway PCA of Batch Polymerization Data

We consider a polymer batch dataset (***polymer.xls***) provided by Dunn [19] consisting of ten process variables X (j = 1 to 10) in 55 batches (j = 1 to 55). Within each batch, we have 100 time steps of data observations (k = 1 to 100). We use PCA along with BWU analysis to identify the abnormal/bad batches using Aspen Pro MV.

We first load ***polymer.xls*** as follows. Start Aspen ProMV and select "New Project". Within the Data Manager, we choose "Batch Blocks", and then click on "Import Batch Block**"** to upload ***polymer.xls*** into the software. Figure 9.64 shows part of the imported data with 10 process variables in different batches.



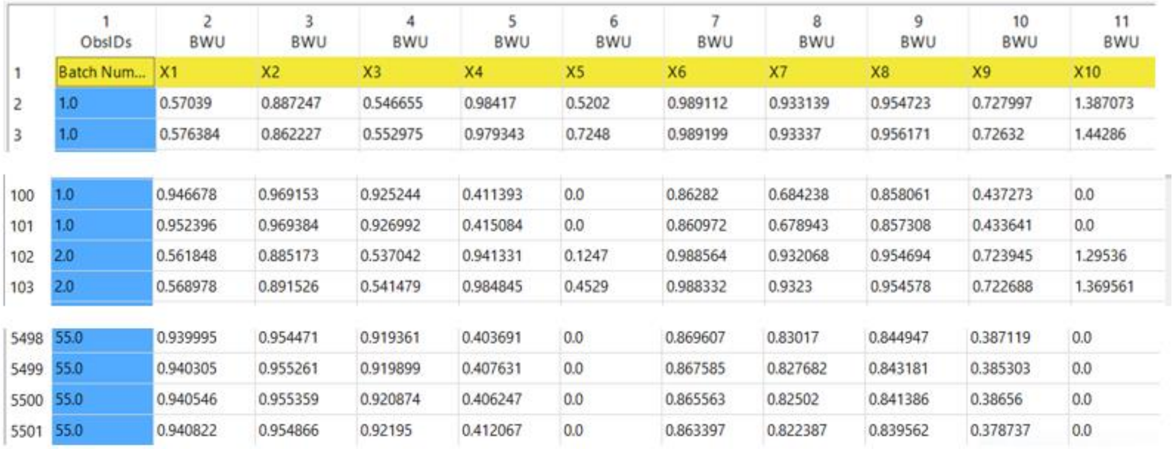
	1 BWU	2 BWU	3 BWU	4 BWU	5 BWU	6 BWU	7 BWU	8 BWU	9 BWU	10 BWU	11 BWU
1	Batch Num...	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
2	1.0	0.57039	0.887247	0.546655	0.98417	0.5202	0.989112	0.933139	0.954723	0.727997	1.387073
3	1.0	0.576384	0.862227	0.552975	0.979343	0.7248	0.989199	0.93337	0.956171	0.72632	1.44286
4	1.0	0.582171	0.823841	0.559798	0.975297	0.7837	0.989719	0.933601	0.956344	0.728416	1.487092
5	1.0	0.589163	0.795793	0.566655	0.976823	0.8186	0.989834	0.933804	0.95481	0.734842	1.511825
6	1.0	0.597086	0.779892	0.573479	0.975901	0.7854	0.988592	0.934498	0.954781	0.740011	1.523741
7	1.0	0.605835	0.770938	0.581277	0.975901	0.8064	0.988477	0.935511	0.955707	0.739033	1.538545
8	1.0	0.614619	0.76567	0.589345	0.977285	0.8139	0.989574	0.936524	0.956257	0.740011	1.537281
9	1.0	0.622714	0.761555	0.597076	0.975581	0.7811	0.990239	0.93716	0.957039	0.74518	1.515797
10	1.0	0.630671	0.759415	0.604807	0.974942	0.7909	0.990556	0.937536	0.959008	0.748952	1.490522
11	1.0	0.639454	0.758625	0.612538	0.976149	0.7979	0.990874	0.937912	0.960252	0.751746	1.462719

Figure 9.64 A part of the imported batch polymerization dataset.

We then choose the batch number column (column1) and click on "Observation IDs" button on the left side pane to designate column 1 to contain Observation IDs. For batch dataset, one "observation" represents a batch. Referring to Figure 9.65, we can explain how the three-way database is displayed.

First, we see that the column to the left of the Observation ID (i.e., batch number) goes from 2 to 5501 (currently displaying columns 2 to 3, 100 to 103, and 5498 to 5501 in the figure), which represents a total of 5500 time steps of data observations, with each observation ID or each batch number containing 100 time steps (that is k = 1 to 100) from 2 to 101 for batch 1, 102 to 201 for batch 2, 202 to 301 for batch 3, …., and 5402 to 5501 for batch 55.

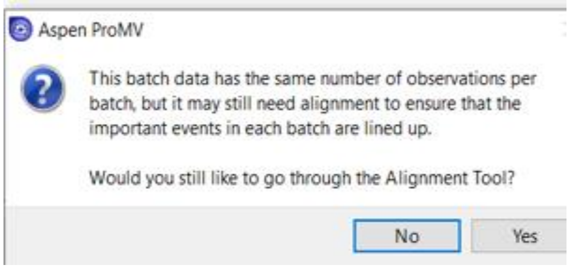
Next, we see column 1 (ObsIDs), Batch Number, varies from 1 to 55 (that is, i = 1 to 55). Lastly, we see Columns 2 to 11 for X1 to X10, representing 10 process variables (that is, j = 1 to 10).



	1 ObsIDs	2 BWU	3 BWU	4 BWU	5 BWU	6 BWU	7 BWU	8 BWU	9 BWU	10 BWU	11 BWU
1	Batch Num...	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
2	1.0	0.57039	0.887247	0.546655	0.98417	0.5202	0.989112	0.933139	0.954723	0.727997	1.387073
3	1.0	0.576384	0.862227	0.552975	0.979343	0.7248	0.989199	0.93337	0.956171	0.72632	1.44286
100	1.0	0.946678	0.969153	0.925244	0.411393	0.0	0.86282	0.684238	0.858061	0.437273	0.0
101	1.0	0.952396	0.969384	0.926992	0.415084	0.0	0.860972	0.678943	0.857308	0.433641	0.0
102	2.0	0.561848	0.885173	0.537042	0.941331	0.1247	0.988564	0.932068	0.954694	0.723945	1.29536
103	2.0	0.568978	0.891526	0.541479	0.984845	0.4529	0.988332	0.9323	0.954578	0.722688	1.369561
5498	55.0	0.939995	0.954471	0.919361	0.403691	0.0	0.869607	0.83017	0.844947	0.387119	0.0
5499	55.0	0.940305	0.955261	0.919899	0.407631	0.0	0.867585	0.827682	0.843181	0.385303	0.0
5500	55.0	0.940546	0.955359	0.920874	0.406247	0.0	0.865563	0.82502	0.841386	0.38656	0.0
5501	55.0	0.940822	0.954866	0.92195	0.412067	0.0	0.863397	0.822387	0.839562	0.378737	0.0

Figure 9.65 A display of parts of the unfolded dataset

On the same window screen of Figures 9.64-9.65, we click on OK, and then choose "No" to align the batch trajectory. See Figure 9.66. We then see the "View/Edit Batch Block" screen, and we highlight column 1 and see the time-dependent change of variable X1 over the 100 time steps within batch 1. See Figure 9.67. We click on "Save".

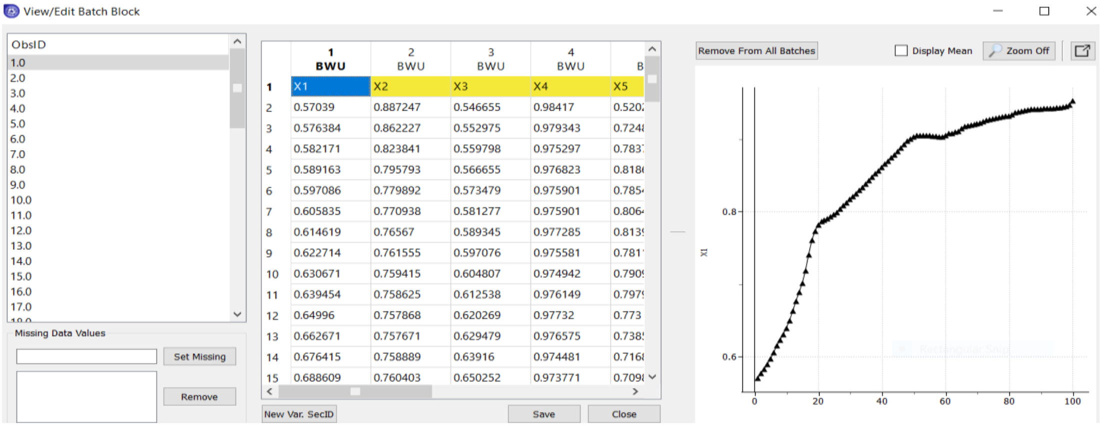


Aspen ProMV

This batch data has the same number of observations per batch, but it may still need alignment to ensure that the important events in each batch are lined up.

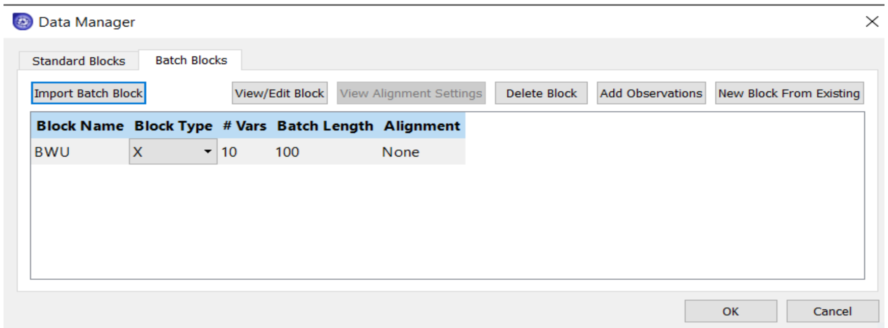
Would you still like to go through the Alignment Tool?

Figure 9.66 Option to align the batch trajectory



	1 BWU	2 BWU	3 BWU	4 BWU	B
1	X1	X2	X3	X4	X5
2	0.57039	0.887247	0.546655	0.98417	0.520
3	0.576384	0.862227	0.552975	0.979343	0.724
4	0.582171	0.823841	0.559798	0.975297	0.783
5	0.589163	0.795793	0.566655	0.976823	0.818
6	0.597086	0.779892	0.573479	0.975901	0.785
7	0.605835	0.770938	0.581277	0.975901	0.806
8	0.614619	0.76567	0.589345	0.977285	0.813
9	0.622714	0.761555	0.597076	0.975581	0.781
10	0.630671	0.759415	0.604807	0.974942	0.790
11	0.639454	0.758625	0.612538	0.976149	0.797
12	0.64996	0.757868	0.620269	0.97732	0.773
13	0.662671	0.757671	0.629479	0.976575	0.738
14	0.676415	0.758889	0.63916	0.974481	0.716
15	0.688609	0.760403	0.650252	0.973771	0.709

Figure 9.67 A display of the time-dependent changes of variable X1 for 100 time steps in batch 1

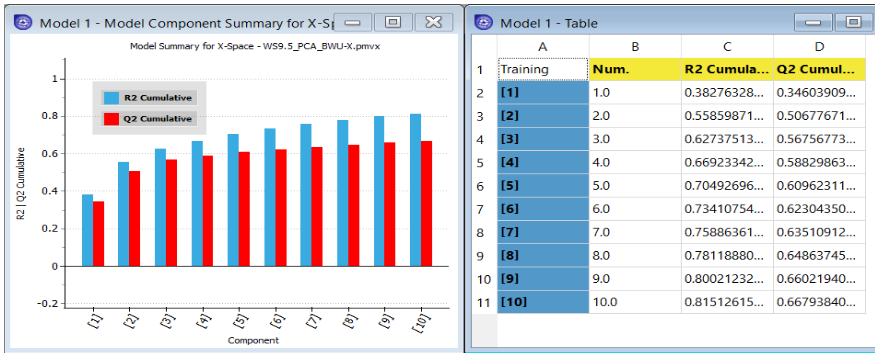


Block Name	Block Type	# Vars	Batch Length	Alignment
BWU	X	10	100	None

Figure 9.68 A summary of imported dataset

We click on OK on the screen of Figure 9.68 and see the Observation Summary (as illustrated previously in Figure 9.15) and then click on OK. We save the resulting file as ***WS9.5_PCA_BWU-X.pmvx***.

Following Figures 9.20, we develop a PCA model of the batch-wide unfolded dataset with 10 principal components (A = 10). Figure 9.69 shows the resulting R2 and Q2 values versus the number of principal components. We note that both R2 and Q2 increase with an increase in the number of principal components. Should we choose to use auto-fitting tool following Figure 9.20, the number of principal components (A) is equal to half of the total number of process variables (j= 1 to 10), that is, A = 5. The corresponding R2 and Q2 values are 0.7049 and 0.6096, respectively.



	A	B	C	D
1	Training	Num.	R2 Cumula...	Q2 Cumul...
2	[1]	1.0	0.38276328...	0.34603909...
3	[2]	2.0	0.55859871...	0.50677671...
4	[3]	3.0	0.62737513...	0.56756773...
5	[4]	4.0	0.66923342...	0.58829863...
6	[5]	5.0	0.70492696...	0.60962311...
7	[6]	6.0	0.73410754...	0.62304350...
8	[7]	7.0	0.75886361...	0.63510912...
9	[8]	8.0	0.78118880...	0.64863745...
10	[9]	9.0	0.80021232...	0.66021940...
11	[10]	10.0	0.81512615...	0.66793840...

Figure 9.69 R2 and Q2 values versus the number of principal components

Following Figure 9.25, we show in Figure 9.70 the score plot, T[2] vs T[1], for the case with 5 principal components. In the plot, we use the button highlighted by an arrow on the top ribbon to select points located close to the 95% confidence limit (dashed ellipse), batch 51; points located between 95% and 99% confidence limits (dashed and solid ellipses), batches 50,52,53 and 55; and point outside the 99% confidence limit, batch 54. These batches represent the apparent outliers or abnormal batches among the 55 batches (i = 1 to 55).

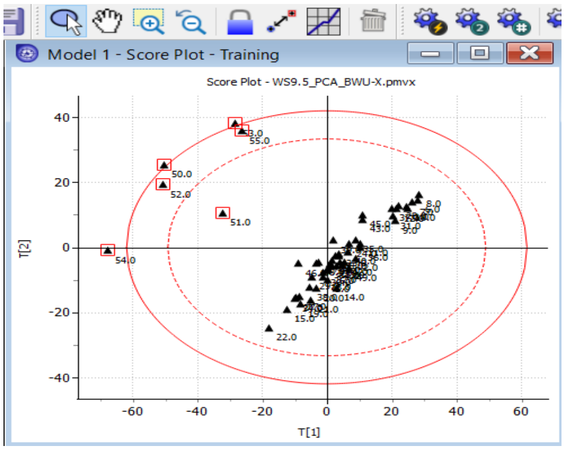


Figure 9.70 Score plot, T[2] vs T[1]

We confirm batches 50 to 55 being abnormal by following Figure 9.26 to draw the Hostelling's T^2 plot in Figure 9.71. Significantly, this plot shows another abnormal batch 49 that is not apparent in Figure 9.70.

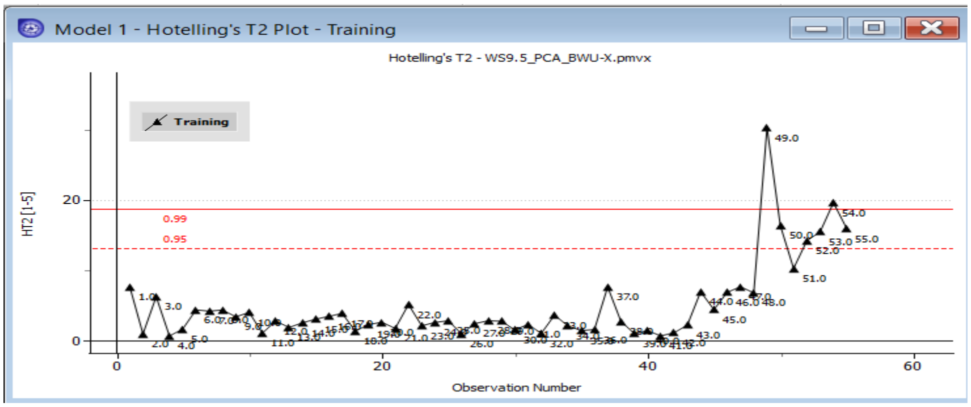


Figure 9.71 The Hostelling T^2 plot

Following Figure 9.27, we show in Figure 9.72 the squared prediction error SPE-X plot, which reveals that batch 51 has the largest SPE-X.

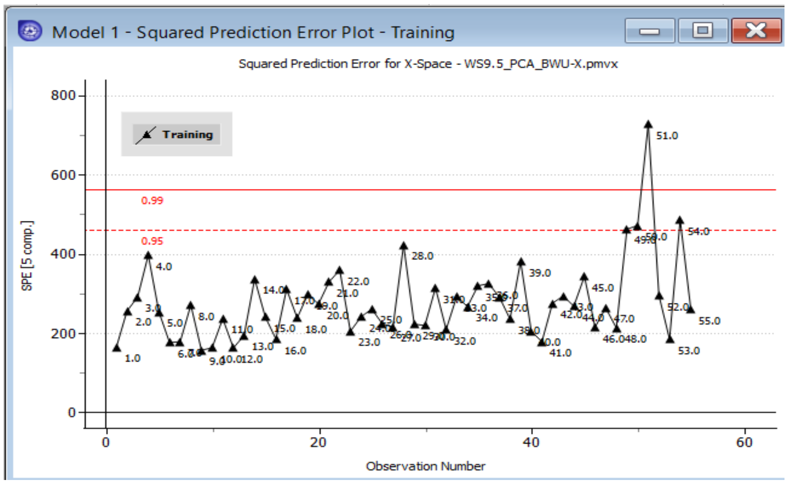
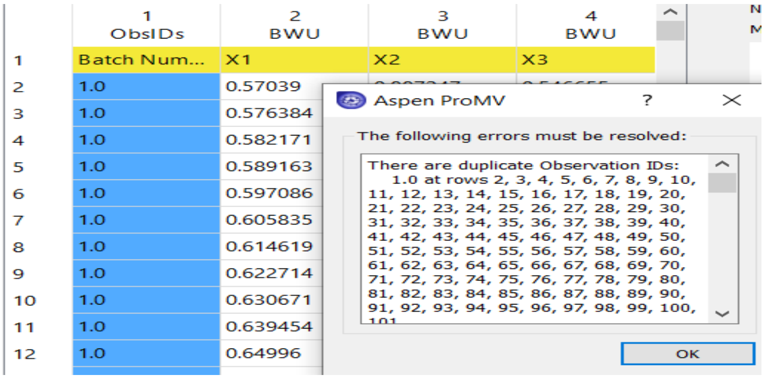


Figure 9.73 The SPE-X plot

We conclude this workshop by finding out what happens when applying the observation-wide unfolding (OWU) approach to the same polymer dataset. We first load ***polymer.xls*** as follows. Start Aspen ProMV and select "New Project". To use OWU on the same dataset, we need to import batch data using "Standard Blocks" within the Data Manager, not "Batch Block" (see Figure 9.14 shown previously). Since the dataset (***polymer.xls***) contains 55 batches with each batch containing 100 time steps, following the same procedure as in the BWU approach will lead to the error of having "duplicate Observation IDs" (see Figure 9.73). We need to do some manipulations of the dataset to prevent the error. We use the Pandas package in Python (see Appendix 9.1) to delete the "Batch Number" column in the Excel sheet and add an index column from one to 5500 observation instances. This column serves as "Observation IDs" for data import by "Standard Blocks". We save the Excel file after data manipulation as ***Polymer_OWU_No duplicate ObsIDs.xls***.



	1 ObsIDs	2 BWU	3 BWU	4 BWU
1	Batch Num...	X1	X2	X3
2	1.0	0.57039		
3	1.0	0.576384		
4	1.0	0.582171		
5	1.0	0.589163		
6	1.0	0.597086		
7	1.0	0.605835		
8	1.0	0.614619		
9	1.0	0.622714		
10	1.0	0.630671		
11	1.0	0.639454		
12	1.0	0.64996		

Aspen ProMV

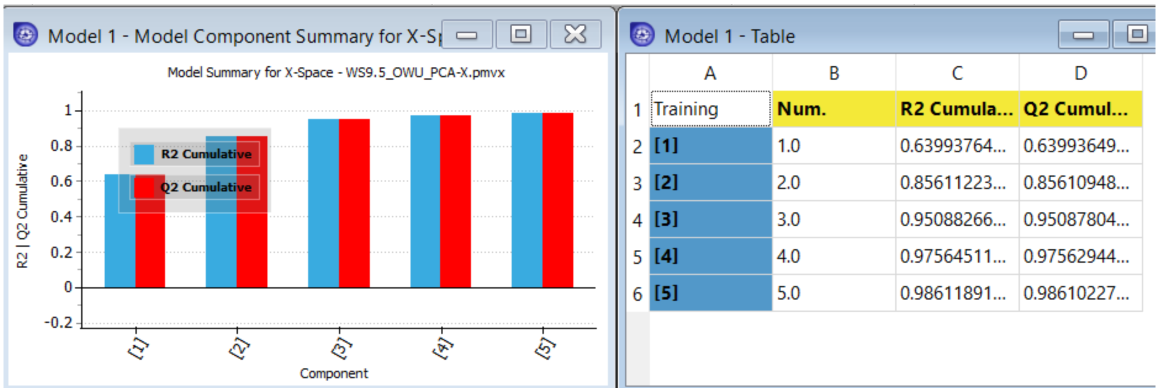
The following errors must be resolved:

```
There are duplicate Observation IDs:
    1.0 at rows 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
61, 62, 63, 64, 65, 66, 67, 68, 69, 70,
71, 72, 73, 74, 75, 76, 77, 78, 79, 80,
81, 82, 83, 84, 85, 86, 87, 88, 89, 90,
91, 92, 93, 94, 95, 96, 97, 98, 99, 100,
101
```

Figure 9.73 Error of having duplicate Observations IDs

We use the Pandas package in Python (see Appendix 9.1) to delete the "Batch Number" column in the dataset Excel sheet and add an index column from one to 5500 observation instances. This column serves as "Observation IDs" for the "Standard Blocks" data import. The file after data manipulation is saved as, ***Polymer_OWU_No duplicate ObsIDs.xls***.

We now follow Figures 9.11 to 21 to develop a PCA model using standard blocks with the modified dataset, ***Polymer_OWU_No duplicate ObsIDs.xls***. Figure 9.74 shows part of the imported dataset, and we do not see the error of having duplicated observation IDs. We save the resulting PCA model as ***WS9.5_OWS_PCA-X.pmvx***. Figure 9.74 shows the resulting R2 and Q2 versus the number of principal components, and Figure 9.75 gives the corresponding score and loading plots. It is unfortunate that the score plot of Figure 9.75 (left) shows *no interpretable trends and no observable outliers with the OWU approach.*This is in contrast to the outliers (batches 50 to 55) depicted in the score plot of Figure 9.70 resulting from applying the BWU approach. The loading plot on the right of Fgure 9.75 shows some variabes located closely together that are correlated, and some variables that lie on the opposite sides of the plot that are negatively correlated.



	A	B	C	D
1	Training	**Num.**	**R2 Cumula...**	**Q2 Cumul...**
2	**[1]**	1.0	0.63993764...	0.63993649...
3	**[2]**	2.0	0.85611223...	0.85610948...
4	**[3]**	3.0	0.95088266...	0.95087804...
5	**[4]**	4.0	0.97564511...	0.97562944...
6	**[5]**	5.0	0.98611891...	0.98610227...

Figure 9.75 R2 and Q2 value versus the number of principal component

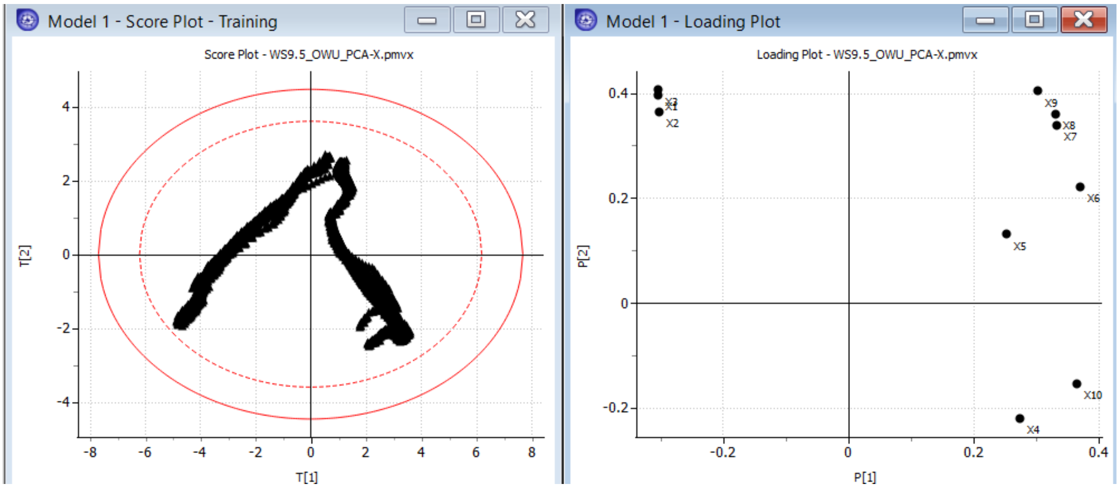
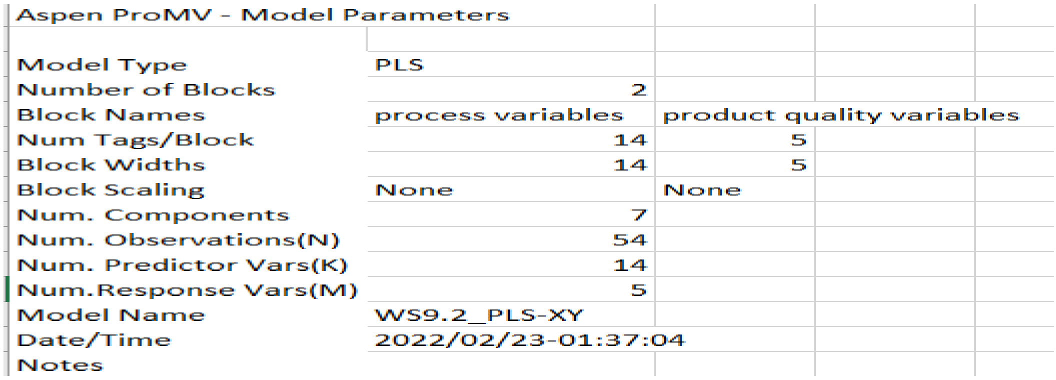


Figure 9.76 T[2] vs T[1] score plot and P[2] vs P[1] loading plot

We conclude that the batch-wise unfolding (BWU) approach is more effective than observation-wise unfolding (OWU) for batch data analytics.

9.7 Implementation of Multivariate Statistics Models

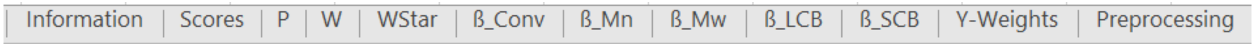
Should the reader wish to extract the equations and coefficients from of developed PCA and PLS models from Aspen ProMV to implement elsewhere, follow the path: Model -> Export Model ->Model List ->Model 1 -> Included in Export -> Training, Batch, Monitoring and Alignment Data -> Excel, e.g., WS9.2_PLS-XY_WS9.2_PLS-XY.xlsx. Figure 9.76 shows an information summary of the model.



Aspen ProMV - Model Parameters		
Model Type	PLS	
Number of Blocks	2	
Block Names	process variables	product quality variables
Num Tags/Block	14	5
Block Widths	14	5
Block Scaling	None	None
Num. Components	7	
Num. Observations(N)	54	
Num. Predictor Vars(K)	14	
Num.Response Vars(M)	5	
Model Name	WS9.2_PLS-XY	
Date/Time	2022/02/23-01:37:04	
Notes		

Figure 9.76 An information summary of the exported model

For the example, we see the following Excel folders of the exported model:



Information	Scores	P	W	WStar	ß_Conv	ß_Mn	ß_Mw	ß_LCB	ß_SCB	Y-Weights	Preprocessing

The various sub-folders in the exported Excel model are as follows.

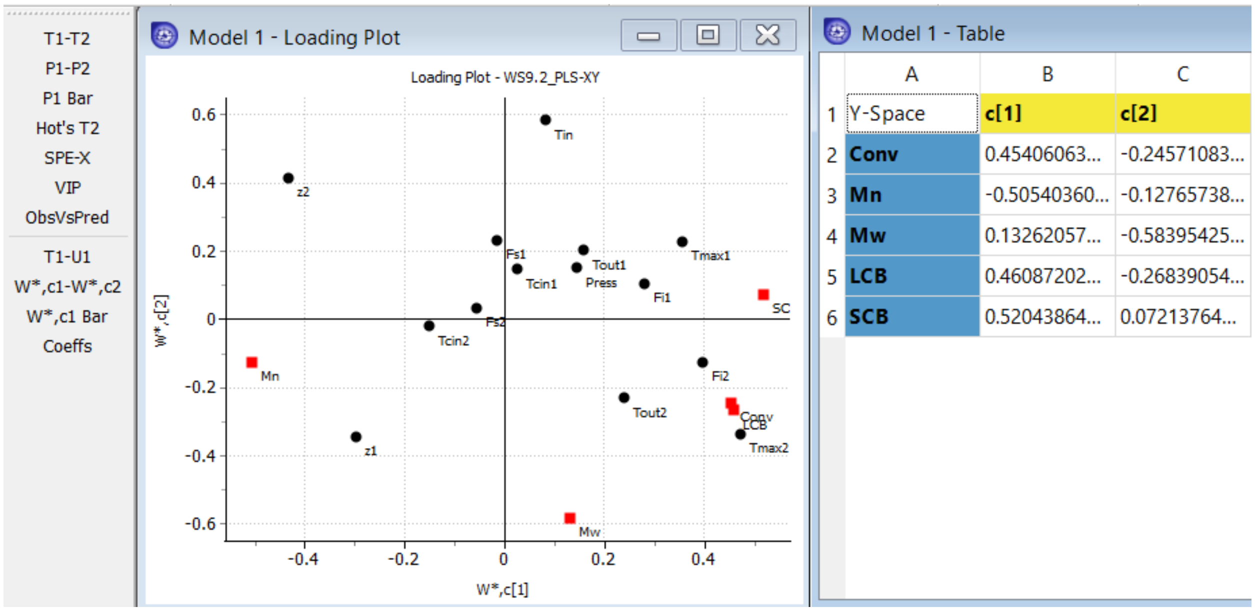
(1) Scores (T) and loadings (P): See Eqs. (9.8) and (9.17); Figures 9.25 and 9.38.

(2) Weights (W): See Eq. (9.24).

(3) Weights (WStar; W*): See Figure 9.38.

(4) Regression coefficients β_Conv, β_Mn, β_Mw, β_LCB and β_SCB: Eq. (9.38)

(5) Y-Weights: See Figure 9.77 below.



	A	B	C
1	Y-Space	c[1]	c[2]
2	Conv	0.45406063...	-0.24571083...
3	Mn	-0.50540360...	-0.12765738...
4	Mw	0.13262057...	-0.58395425...
5	LCB	0.46087202...	-0.26839054...
6	SCB	0.52043864...	0.07213764...

Figure 9.77 Weights for Y-space.

If you do not have Aspen ProMV for outlier or anomaly detection, consider using Python that we introduce in Appendix B, Introduction to Python for Chemical Engineers, and refer to Section 10.1.3, Suggested Resources to Get Started with Machine Learning. Adopt the open-source Scikit-Learn PLS library for model building and getting the model coefficients: https://scikitlearn.org/stable/modules/generated/sklearn.cross_decomposition.PLSRegression.html

Additionally, In Section 10.2.3 and Table 10.5 of Chapter 10, we introduce additional machine learnng-based methods for outlier or anomaly detection, which can be implemented by Python. Two of the popular methods are Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Section 10.2.3.d) and Gaussian mixture model (GMM) (Section 10.2.3.e).

Most of the multivariate statistical models in this chapter and machine learning models in Chapters 10 and 11 use historical data. For online implementation, we need a real-time plant historian, such as Aspen InfoPlus.21 and Aspen Process Explorer, to demonstrate online model deployment. For example, Aspen Technology, Inc. has several software tools, such as Aspen Process Pulse^{TM} and Aspen Scrambler^{TM} to enable the monitoring, controlling, and optimizing processes with real-time visibility of all types of process and spectral data. Interested readers may refer to Sharmin et al. [27] about a PCA-based fault detection scheme for an industrial high-pressure polyethylene reactor using Aspen Process Explorer.

9.8. Conclusion and Suggested Resources for Further Studies

In this chapter, we have showcased the utility of latent variable models like PCA and PLS for causal analysis to identify correct correlations between input and outputs for polymer process application. We identify the Dynamic PCA and PLS model utility in dynamic time series process data by considering the measurement lags. We also demonstrate the methodology for batch-wise unfolding (BWU) and observation-wise unfolding OWU) analyses of batch data.

For further studies, we recommend references [24] to [26] in the bibliogrphy below. We discuss a number of topics together with their relevant references below.

Gracia-Munoz et. al. [24] discussed the issue of time allignment in batch proesses. Specifically, in many batch processes, batches can be of different time durations within certain phases or across the entire batch evoluation. A search of Aspen ProMV online help gives the details and examples of alignment tools and their implementation in batch processes.

Park et. al. [21] and Han et. al. [25] presented interesting case studies of applying PLS and machine learning tools (support vector macines and artificial neural networks) to modeling the melt index of high-density polyethylene (HDPE), styrene-acrylontrille (SAN) and polypropylene (PP) processes operating in Korea.

Chen and Lu [26] integrated auto-regressive moving average (ARMAX) exogenous time series model with PCA model, and called it dynamic PCA (DPCA) that involves the use of time lags which we discussed in Sections 9.5.1 and 9.5.2. They also combined three-way observation-wise unfolding PLS (Section 9.6.1) with time-lagged windows, and called it batch dynamic PLS (BDPLS). They applied both methods to industrial batch polymerization datasets. One of the future idea is to combine the multivariate statistics with science and process knowledge in hybrid methodology for process improvement as shown Sharma & Liu [29,45].

This chapter is published with Wiley publication in the book *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing by Liu & Sharma.* [31,32,33,34,35,36,37,38,39,40,41,42]

9.9 Biobliography

1. Quantrille, T. E.; Liu, Y. A. (1991). *Artificial Intelligence in Chemical Engineering*. Academic Press, Inc., San Diego, CA (now Elsevier, New York).
2. Baughman, D. R.; Liu, Y. A. (1995). *Neural Networks in Bioprocessing and Chemical Engineering*. Academic Press, Inc., San Diego, CA (now Elsevier, New York).
3. Qin, S. J. (2014). Process Data Analytics in the Era of Big Data. *AIChE Journal* , **60**, 3092-3100.
4. Chiang, L.; Lu, B.; Castillo, I. (2017). Big Data Analytics in Chemical Engineering. *Annual Review of Chemical and Biomolecular Engineering*, **8**, 63.
5. Ge, Z.; Song, Z.; Ding, S. X.; Huang, B. (2017). Data Mining and Analytics in the Process Industry: The Role of Machine Learning. *IEEE Access*, ***5***, 20590.
6. Skagerberg, B.; MacGregor, J. F.; Kiparissides, C. (1992). Multivariate Data Analysis Applied to Low-Density Polyethylene Reactors. *Chemometrics and Intelligent Laboratory Systems*. **14,** 341.
7. Kourti, T.; MacGregor, J. F. (1995). Process Analysis, Monitoring and Diagnosis Using Multivaraite Projection Methods. *Chemometrics and Intelligent Laboratory Systems*. **28,** 3.
8. MacGregor, J. F.; Kourti, T.,(1995). Statistical Process Control of Multivariate Processes. *Control Engineering Practice*, **3** , 403.
9. MacGregor, J. F. (1997). Using On-Line Process Data to Improve Quality: Challenges for Statisticians. *International Statistical Review*, **65**, 309.
10. MacGregir, J. F.; Bruwer, M.-J. (2017). Optimizion of Processes and Products Using Historical Data. *Foundations of Computer Aided Process Operations / Chemical Process Control Conference,* St. Antonio, Texas, January. https://docplayer.net/42981979-Optimization-of-processes-products-using-historical-data.htML. Accessed January 15, 2022.
11. Johnson, R. A.; Wichern, D. W. (2013). *Applied Multivariate Statistical Analysis*. 6th Ed., Pearson Education, Inc. New York.
12. Rencher, A. C.; Christiansen, W. F. (2012). *Methods of Multivariate Analysis*, 3rd Ed., Wiley, New York.
13. Dunn, K., Process Improvement Using Data. *Creative Commons Attribution-ShareAlike;* https://learnche.org/pid/. Accessed January 15, 2022.
14. Aspen Technology, Inc. (2003). Training Course on Inferential Property Development and Control with Aspen IQ and DMCplus: Multivariate Statistics.
15. Everett, B.; Hothorn, T. (2011) *An Introduction to Applied Multivariate Analysis with R*. Springer, New York, NY.
16. Wold, S.; Sjostrom, M.; Erikson, L. (2001). PLS-Regression: A Basic Tool for Chemometrics. *Chemometrics and Intelligent Laboratory Systems*. **58**, 109.
17. Geladi, P.; Kowalski, B. R. (1986). Partial Least-Squares Regression: A Tutorial. *Analytica Chimica Acta*, **185**, 1.
18. Dunn, K.; LDPE Dataset. *All OpenMV.net Databases*, https://openmv.net/. Accessed January 16, 2022.
20. Wold, S. (1978). Cross-Validatory Estimation of the Number of Components in Factor and Principal Component Models. *Technometrics*. **20**, 397.

21. Park, T. C.; Kim, T. Y.; Yeo, Y. K. (2010). Prediction of the Melt Flow Index Using Partial Least Squares and Support Vector Regression in High-Density Polyethylene (HDPE) Process. *Korean Journal of Chemical Engineering*, **27**, 1662.

22.Nomikos, P.; MacGregor, J. F. (1994). Monitoring Batch Processes Using Multiway Principal Component Analysis. *AIChE Journal,* **40** , 1361.

23. Wold, S.; Geladi, P.; K. Esbensen; Ohman, J. (1987). Multiway Principal Components and PLS Analysis, *J. Chemometrics*, **1** ,41.

24. Garcia-Munoz, S.; Khouri, T.; MacGregor, J. F.; Mates, A.G.; Murphy, G. (2003). Troubleshooting of an Industrial Batch Process Using Multivariate Methods. *Ind. Eng. Chem. Res*., **42**, 3592.

25. Han, I.-S.; Han, C.; Chung, C.-B. (2005). Melt Index Modeling with Support Vectr Machines, Partial Least Squares, and Artificial Networks. *J. App. Polymer Sci.*, **95**, 967.

26. Chen, J.; Liu, K.C. (2002). On-Line Batch Process Monitoring Using Dynamic PCA and Dynamic PLS Models. *Chem. Eng. Sci.*, **57**, 63.

27. Sharmin, R.; Shah, S. L.; Sundararaj, U. (2008). A PCA Based Fault Detection Scheme for an Industrial High Pressure Polyethylene Reactor. *Macromolecular Reaction Engineering*, **2**, 12.

28. Sharma, N., & Liu, Y. A. (2019). 110th anniversary: an effective methodology for kinetic parameter estimation for modeling commercial polyolefin processes from plant data using efficient simulation software tools. *Industrial & Engineering Chemistry Research*, *58*(31), 14209-14226. https://doi.org/10.1021/acs.iecr.9b02277

29. Sharma, N., & Liu, Y. A. (2022). A hybrid science-guided machine learning approach for modeling chemical processes: A review. *AIChE Journal*, *68*(5), e17609. https://doi.org/10.1002/aic.17609

30. Nguyen, X. D. J., Sharma, N., Liu, Y. A., Lee, Y., & McDowell, C. C. (2023). Analyzing the occurrence of foaming in batch fermentation processes using multiway partial least square approaches. *AIChE Journal*, *69*(12), e18250. https://doi.org/10.1002/aic.18250

31. Liu, Y. A., & Sharma, N. (2023). *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing*. Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831

32. Liu, Y. A., & Sharma, N. .2023. Application of Multivariate Statistics to Optimizing Polyolefin Manufacturing. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing* (Chapter 9, pp. 477-531). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch9

33. Liu, Y. A., & Sharma, N. (2023). Introduction to Integrated Process Modeling, Advanced Control, and Data Analytics in Optimizing Polyolefin Manufacturing. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing* (Chapter 1, pp. 1-40). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch1

34. Liu, Y. A., & Sharma, N. (2023). Selection of Property Methods and Estimation of Physical Properties for Polymer Process Modeling. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing* (Chapter 2, pp. 41-86). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch2

35. Liu, Y. A., & Sharma, N. (2023). Reactor Modeling, Convergence Tips, and Data-Fit Tool. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing* (Chapter 3, pp. 87-114). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch3

36. Liu, Y. A., & Sharma, N. (2023). Free Radical Polymerizations: LDPE and EVA. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing* (Chapter 4, pp. 115-162). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch4

37. Liu, Y. A., & Sharma, N. (2023). Ziegler–Natta Polymerization: HDPE , PP , LLDPE, and EPDM. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 5, pp. 163-265). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch5

38. Liu, Y. A., & Sharma, N. (2023). Free Radical and Ionic Polymerizations: PS and SBS Rubber. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 6, pp. 267-319). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch6

39. Liu, Y. A., & Sharma, N. (2023). Improved Polymer Process Operability and Control Through Steady-State and Dynamic Simulation Models. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 7, pp. 321-379). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch7

40. Liu, Y. A., & Sharma, N. (2023). Model-Predictive Control of Polyolefin Processes. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 8, pp. 381-476). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch8

41. Liu, Y. A., & Sharma, N. (2023). Applications of Machine Learning to Optimizing Polyolefin Manufacturing. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 10, pp. 553-650). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch10.

42. Liu, Y. A., & Sharma, N. (2023). A Hybrid Science-Guided Machine Learning Approach for Modeling Chemical and Polymer Processes. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 11, pp. 651-698). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch11

43. Sharma, N. and Liu, Y., 2019, November. Polyolefin Process Modeling and Monitoring. In *2019 AIChE Annual Meeting*. AIChE.

44. Sharma, N. and Liu, Y., 2020, November. Polyolefin Process Improvement Using Machine Learning. In *2020 Virtual AIChE Annual Meeting*. AIChE

45. Sharma, N., 2022, November. Polyolefin Property Estimation using Process Modeling and Machine Learning in Industry. In *2022 AIChE Annual Meeting*. AIChE.