FAULT DETECTION USING PROJECTION PURSUIT REGRESSION (PPR): A CLASSIFICATION VERSUS AN ESTIMATION BASED APPROACH

Shijin Lou, Thomas Duever and Hector Budman

Department of Chemical Engineering, University of Waterloo, Waterloo, ON, Canada

Abstract: Two fault detection approaches are compared using a Projection Pursuit Regression (PPR) algorithm: i- a classification approach where the fault detection PPR model is trained based on the class numbers and ii- an estimation approach where the PPR model is trained to predict the value of the process variable that define the class boundaries and then the corresponding class is identified by comparing the estimated value versus the limits of the fault classes. The comparison is carried on for simple illustration examples, to elucidate the main issues, and for a copolymerization process. The classification approach is found superior provided that the training data closest to the boundaries are located at equidistant locations from these boundaries.

Keywords: Fault Detection, Regression Algorithm

1. INTRODUCTION

One of the goals in fault detection or classification problems is to establish, from measurements, that a specific variable value lies within a certain class defined by a range of values of that variable. Thus, for fault detection problems, the outcomes are a set of discrete values for the variable in question. On the other hand the values of the variables that define the boundary of a class can be continuously predicted from measurements using for example a state estimator. The prediction surface for the variable to be estimated is usually continuous. However, such an estimation model could be easily used for fault detection by comparing the predicted value of a variable to the boundaries defining the different classes or faults and then assessing the class or fault. The additional benefit of having continuous estimates of certain variables is that they could be used for feedback or feed-forward control. This paper is addressing the differences and relative advantages and disadvantages between these two approaches, i.e. the classical fault detection approach where a model is trained to predict directly a class versus the estimation based approach where the model is trained to predict the value of a variable and then the corresponding class is identified from that value. For clarity, the former will be referred to as the classification model whereas the later will be referred to as the estimation-based detection model. Intuitively, it is possible to expect that as the range of values, defining a class for the purpose of fault detection, becomes smaller and smaller, a fault detection model will eventually converge to an estimation model. Does this imply that fault detection is a "rough" version of estimation? Furthermore, will it be always true that a fault detection model will require less experiments for training as compared to the estimation based approach? It will be shown in this manuscript that the answers to these questions is not always affirmative and they are directly related to the level of noise, the linearity of the problem and the specific modelling methodology utilized to obtain the detection or estimation models. Many different modelling techniques have been proposed for estimation or fault detection problems. For example, Kalman filters have been often used for estimation or detection when a mechanistic model is available. Also, a number of empirical techniques have been investigated ranging from Neural Networks (Bakshi, 1999) to multivariate statistical modelling methods such as Partial Least Squares (Yoon and MacGregor, 2004). Projection Pursuit Regression (PPR) is an additional multivariate modelling technique (Friedman, 1981) based on basis functions that are tailored specifically to the particular set of data to be modelled resulting generally in less parsimonious models with lower sensitivity to noise. The authors of this work have conducted extensive research on the use of PPR for class detection and have compared this modelling fault detection methodology with other techniques such as back propagation neural networks and Radial Basis Functions Neural networks. They have found in these studies that PPR provides a good tradeoff between sensitivity to noise and generalization accuracy as compared to other neural network based methodologies. (Lou, 2003). For instance, for a 2-dimensional problem, Lou found that PPR results in approximately 50% of the classification error obtained with a Haar Wavenet-based model and 35% of the classification error obtained with a Backpropagation Neural Network model. Therefore, this study will conduct the comparison between direct detection and estimation-based detection using specifically PPR based models.

This paper will be organized as follows. Section 2 will briefly summarize the PPR algorithm and its application to detection and estimation problems. Section 3 will discuss simple examples that were specifically tailored to elucidate some of the issues discussed in the introduction regarding the comparison between detection and estimation problems using PPR. Section 4 will discuss a more involved chemical engineering problem, the estimation of impurities in a copolymerization process from conversion and temperature measurements. Finally, conclusions are presented in Section 5.

2. PROJECTION PURSUIT REGRESSION (PPR): BRIEF SUMMARY

PPR is a multivariate statistical technique originally proposed by Friedman and Stuetzle (1981). The technique can be viewed as a 3 layer-neural network composed of an input layer, one hidden layer and an output layer. The input layer operates on inputs or independent variables x whereas the output layer produces the outputs or dependent variables y.

Three sets of parameters: projection directions given by weights between the input and the hidden layers $\alpha_k^T = [\alpha_{k1},...,\alpha_{kp}]$, projection 'strengths' given by weights between the hidden and the output layers) $\beta_k = [\beta_{1k},...,\beta_{qk}]$, and the a priori unknown activation functions in the hidden layer $\{f_k\}$, are estimated via the least squares criteria by minimizing the squared error cost function: (Utojo and Bakshi (1999))

$$L = \sum_{i=1}^{q}\left[(y_i - \sum_{k=1}^{m} \beta_{ik} f_k(\alpha_k^T x)\right]^2 \qquad (1)$$

Each response variable, $y_i\ (i = 1,2,...q)$, is modeled as a weighted linear combination of the activation function $\{f_k\}$. Each of these functions is a nonlinear function or 'look-up' table, of a weighted linear combination of the weighted independent variables. The output of a hidden function f_k is decided according to the nearest neighbor or neighbors in the 'look-up' table. Projection Pursuit Regression learns function by function and layer-by-layer cyclically after all the training patterns are presented. Specifically, it applies linear Least Squares to estimate the output-layer weights and the Gauss-Newton nonlinear Least Squares method to estimate the input-layer weights. The optimization algorithm grows the model step-wise as in the Nonlinear Iterative Partial Least Squares (NIPALS) algorithm used for the Partial Least Squares (PLS) method. The main difference between PPR and PLS is that the later uses fixed-shape basis functions either linear or polynomial, while PPR uses adaptive basis functions, which are decided by the training data. The PPR basis functions are computed by smoothing the projected data versus the output by using a variable-span smoother such as the supersmoother (Friedman (1984)). The adaptability of the basis functions in PPR allows it to determine more parsimonious models, i.e., using less basis functions than those modeling tools using fixed basis functions, for the same approximation error. A detailed mathematical description is given by Utojo and Bakshi (1999).

Finally, in the introduction, two different forms of constructing a fault detection algorithm have been discussed, i.e. direct detection of the class or fault versus estimation of the variable value and then testing this value versus the ranges of values that define the classes or faults. The difference between the two methodologies is that for the first case, the output data y is discrete and it is typically given in terms of integer numbers whereas for the second method continuous values of y are used for training. In section 3 and 4 examples are given to compare these two methodologies based on the PPR regression algorithm.

3. SIMPLE ILLUSTRATION EXAMPLES

In this section two simple examples are presented to address the comparison between the direct-classification approach versus the estimation-based detection approach. The examples have been specifically tailored to elucidate the issues especially with regards to sensitivity to measurement noise and nonlinearity of the underlying process for which faults are to be detected.

3.1 Linear example

In this example, a linear process model is represented by the following equation:

$$x=p \qquad (2)$$

Where, x is the process measurement; p is the process variable. The objective of a classification model is to find a specific class or fault based on a measurement x. The classes are defined by the value of p as follows:

$$\text{Class 1: } 0<p\leq 0.5 \quad \text{Class 2: } 0.5<p<1 \quad (3)$$

Unlike the classification model, the goal of the estimation-based model is to establish a direct mapping from x to p, i.e., to predict the true value of p, according to the measurements, x. The estimate of p is then used to decide which class x belongs to. Thus the inputs to the PPR model, referred to as the network input, are the measured values of x and the output from the PPR model y, referred to as the network output, are equal to the class number for the classification model or to the estimated values of p for the estimation-based model.

For this example it is assumed that 3 measurements of x are available x=[0.2 0.6 0.95]. Correspondingly, for the training of a classification model, the PPR model is trained on a data pattern given by y= [1 2 2]. Based on this training data the PPR model is tested for different values of x providing the results shown in Figure 1. Clearly, the PPR model locates the class boundary at x=0.4 instead of x=0.5 that is the actual location of the boundary according to (3) resulting in misclassification of all the point in the range 0.4<x<0.5. The explanation for this misclassification is that the two training data on the two sides of class boundary (x=0.2 and x=0.6) are not symmetric with respect to the actual class boundary $x=0.5$. Since the PPR output calculation is based on the nearest neighborhood concept, the PPR model locates the class boundary at the midpoint between the rightmost point of class 1 (x=0.2) and the leftmost point from class 2 (x=0.6) locating the boundary at x=0.4 with resulting misclassification of testing data.

On the other hand the training data for a PPR estimation-based model are the actual measured values as follows y=[0.2 0.6 0.9] instead of y=[1 2 2] used for the classification model. The estimation-based model finds correctly the straight line relation described by (2) passing through all three training data. Consequently for this case, the estimation model can make accurate prediction, even though the training data on two side of the class boundary are not symmetric with respect to it. In this experiment, the estimation model predicts the testing data accurately, and the classification based on the estimation-based model does not produce any misclassification. This example show that in a noise-free linear problem, a PPR estimation-based model trained with the absolute values of the measurements works better than a PPR classification model trained with the class number values, especially when the training data in the two classes are not symmetric with respect to the actual class boundary.

3.2 Linear Example using Training Data Corrupted by Noise

The system in this example is the same as described by (2) above. In this example, there are also three training data, as in the previous example, with one training data in Class 1 and two in Class 2. The training data pattern for the estimation model is plotted in Figure 2. In this case the training data is corrupted by noise, and consequently is biased from the actual process model represented by the solid straight line in figure 2.

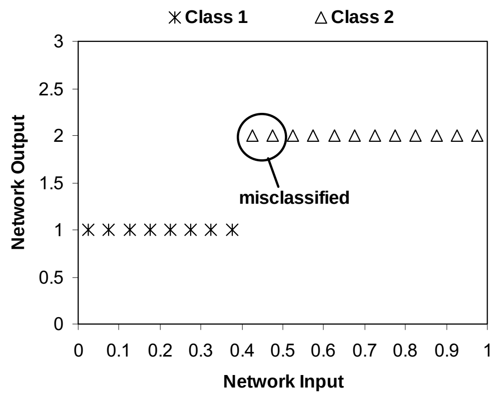


Figure 1. Testing results by a PPR classification model, for a 1-dimensional linear example with noise-free training data.

In the training of a PPR classification model, the inputs are x=[0.2 0.8 0.9] and the outputs for training are the corresponding class numbers as follows y=[1 2 2]. In this case the PPR classification model correctly locates the class boundary at x=0.5 because the rightmost data point from class 1 (x=0.2) and the leftmost data point from class 2 (x=0.8) are now symmetric with respect to the class boundary at x=0.5. Then, the PPR classification model correctly predicts all faults by assigning class one to all measured x<0.5 and class 2 to all measured x>0.5.

For the estimation based model the training data is given by x=[0.2 0.8 0.9] whereas the output data is y=[0.3 0.88 0.9]. In this case, due to noise and the sparseness of the training data a poor PPR estimation-based model is obtained. The prediction of the testing data for different values of x is quite different from their true value as shown in figure 3, resulting in misclassification of 10% of the tested points as illustrated in that figure.

Thus, in a classification problem, the noise in the training data will not affect the classification accuracy, unless the noise level is so significant that it causes data to be assigned to the wrong class. Thus, the noise has no harmful effect on a classification model, if it is small enough such as the training data are still located in the correct classes. This is exactly the situation in this example. Therefore, the classification model makes no misclassification in the testing. This example shows that, due to the discretization of the network outputs, a classification model may be less sensitive to the noise in the training data, as compared to an estimation-based model.

3.3 Nonlinear Example using Noise-free Training Data

This example assumes a nonlinear model, and there is no noise in the training data. The process model can be described by the following model.

$$x = \log_{10}(p) \qquad (4)$$

The classification is decided as follows.

Class 1: $p \leq 3.16$ Class 2: $p > 3.16$

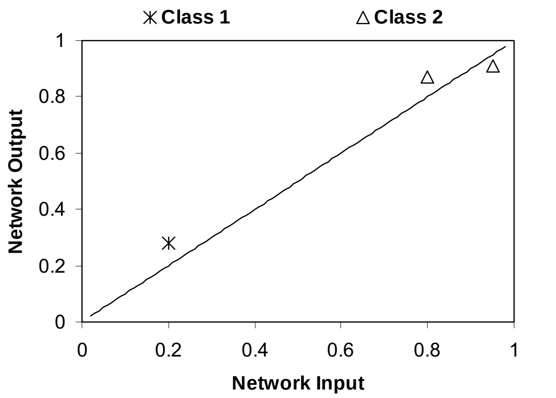


Figure 2. Training data with noise for a PPR estimation model, 1-dimensional linear example

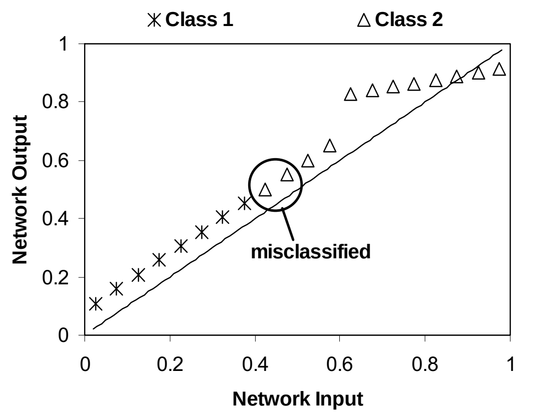


Figure 3. Testing result by a PPR estimation model, 1-dimensional linear example with noisy training data

The training data for the estimation model is presented in Figure 4. The training data in Class 1 and Class 2 are represented by *star* and *triangle* symbols, respectively. The class boundary is located at $x=0.5$ and $p= 3.16$. For these data, the *star* and the *triangle* closest to the class boundary are symmetric with respect to it, in terms of the network input, x. Consequently the PPR classification model accurately predicts the testing data without any misclassification. On the other hand, due to the nonlinearity of the problem and the sparseness of the training data, the estimation-based PPR model misclassifies testing data as shown in Figure 5. The sudden change in the output around the input $x=0.6$ is a consequence of the particular basis functions that the PPR algorithm found for this problem and for the given training data. The training of the estimation model has been done to obtain a training error of zero for the 3 data points in Figure 4. The difference between the estimation and the actual value results in 10% misclassification out of the total data tested. Thus, although PPR is a suitable algorithm to describe nonlinear systems, the resulting estimation model is not accurate due to the sparseness of the training data.

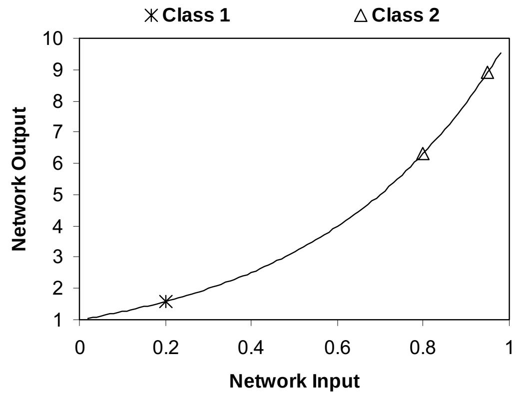


Figure 4. Noise-free training data for a PPR estimation model, 1-dimensional nonlinear example

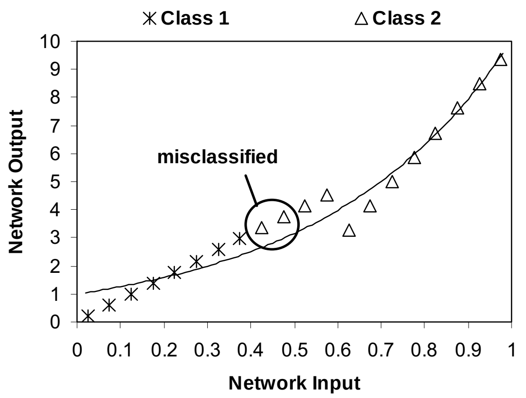


Figure 5. Testing result by a PPR estimation model, 1-dimensional nonlinear example with noise-free training data

This example shows that, in a nonlinear problem, a classification model may need less training data to reach desirable classification accuracy, as compared to an estimation-based model.

3.4 Nonlinear two-dimensional example

The process model investigated here can be mathematically expressed as follows:

$$(5 \cdot x_1)^{0.5} \cdot (1.5 \cdot x_2)^5 = p + v \qquad (5)$$

First all the data are noise free, i.e. v is set to zero. The classification is decided by the value of the process variable, p.

Class 1: $p \leq 0.4$
Class 2: $p > 0.4$

The function is geometrically illustrated in Figure 6. A complete grid is sampled and plotted in the measurement domain in Figure 7. A data point is either represented by a *star* or a *plus*, according to its corresponding class. The boundary between the classes shown in Figure 7 is not straight as in the previous one-dimensional examples. For training, the following four process measurements are sampled: [0.025 0.025], [0.025 0.975], [0.975 0.025], and [0.975 0.975]. The corresponding output to train the classification model are y= [1 2 1 2] and their corresponding output values for training of the estimation-based model are $y=[2.622\times10^{-S}$, 2.366, 1.637×10^{-7}, 14.773] respectively. All, the points in Figure 7 are used for testing of the resulting PPR regression model.

It is possible to show from figure 7 that the selected training data is located approximately symmetrically with respect to the class boundary corresponding to p=0.4, i.e. the training data in class 1 and class 2 are located at similar distances to the class boundary in terms of their x coordinates. The missclassification on the testing data are 16.8% for the estimation model, and 3.2% for the classification model, as summarized in Table 1.

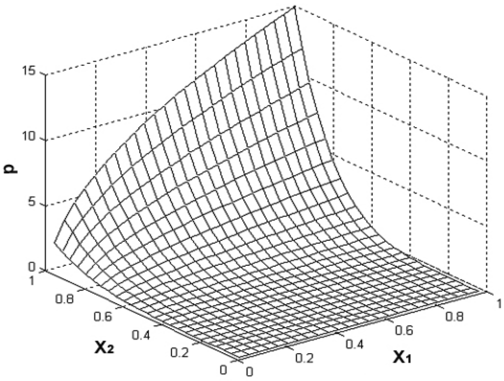


Figure 6. Function surface, 2-dimensional nonlinear example with noise-free data

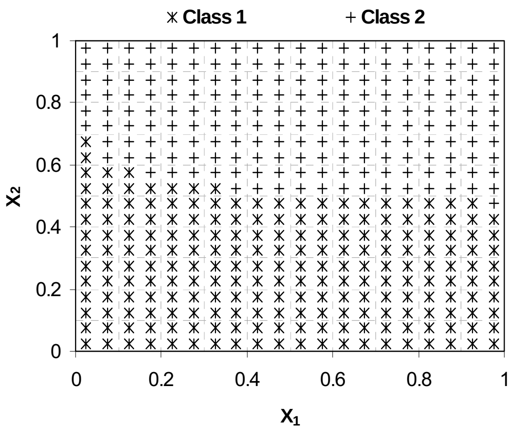


Figure **7**. Testing data set for the 2-dimensional nonlinear example with noise-free data.

Subsequently, the training data is corrupted by random noise. The noise is assumed to be of a magnitude, so that the difference between each measurement (x_1 or x_2) and its true value is smaller than 0.5 times its sampling rate. The training data for the estimation model are: [0.0320 0.0117], [0.0300 0.9579], [0.9301 -0.0556], and [0.9883 0.9791] with the corresponding desired outputs for training the estimation-model y=[0.1489, 2.0961, -0.89646, and 15.499]. The classification model is trained with y=[0 1 0 1].

The misclassification for the testing data is summarized in Table 1. The results in Table 1 verify for the 2 dimensional case the following conclusion: for nonlinear systems and in the presence of measurement noise, a PPR classification model can outperform a PPR estimation-based model, when training data is located on the two sides of class boundary and in symmetrical locations with respect to it. This conclusion is consistent with the results obtained in the one-dimensional case.

Table 1. Comparison of classification and estimation technique in two-dimensional examples

	estimation	**classification**
Misclassification percentage in noise-free data	*0.1675*	*0.0325*
Misclassification percentage in noisy data	*0.215*	*0.04*

	estimation	**classification**
Misclassification percentage in noise-free data	*0.1675*	*0.0325*
Misclassification percentage in noisy data	*0.215*	*0.04*

4. EXAMPLE OF A COPOLYMERIZATION PROCESS

Finally, the comparison between a pure classification model to an estimation-based PPR models is carried out for a fault detection task in a polymerization process. The process is a batch copolymerization of STY/MMA. A detailed mathematical model proposed by Landry (1996) has been used. The model is given by six 1st order ODE's derived from energy, mass and component balances. Reactive impurities are commonly encountered in industrial polymerization processes. Consequently, the objective of the fault detection algorithm is to identify the impurity in ranges of values defining classes as follows:

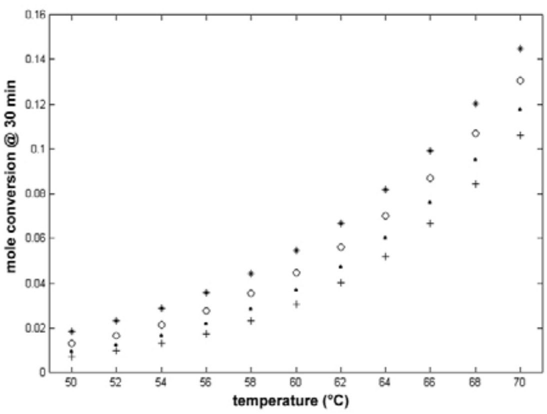
Class 1: $0 \le y < 100\,ppm$
Class 2: $100\,ppm \le y < 300\,ppm$
Class 3: $300\,ppm \le y < 500\,ppm$
Class 4: $500\,ppm \le y < 700\,ppm$

The impurities are detected based on two measurements: the temperature and the mole conversion after 30 minutes of operation. The authors of this work have theoretically shown that the impurity concentration is observable from these two measurements (Lou, 2005). The training data for the case without noise is shown in Figure 8. Based on the observations made for the examples shown in Section 3, the training data points were selected at equidistant locations from the two sides of the class boundaries defined above. The simulation results are summarized in Table 2. The results show that the classification model gives better performance than the estimation model for both the noise free data and data corrupted by Gaussian noise. However, the difference is not as large as expected. To clarify further, the simulated data have been investigated in graphic form. Figure 9 presents the noise-free testing data. In this diagram, the impurity is plotted with respect to the temperature and the mole conversion. Although the overall data pattern is obviously nonlinear, the nonlinearity is not very large.

In general the observations from this more complex example confirms that the PPR classification model tends to outperform a PPR estimation model, when the problem is nonlinear and in the presence of measurement noise. It is expected based on the simple examples shown above, that the improvement of the classification model versus the estimation-based model could be especially significant when the nonlinearity is more pronounced.

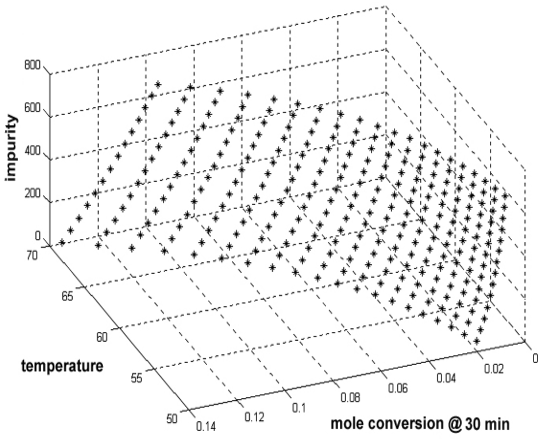
Table 2. Comparison of classification and estimation technique in polymerization examples

	Noise-free data	Data with noise
Estimation	*0.32*	*0.49*
Classification	*0.27*	*0.46*



✱ Class 1; ○ Class 2; • Class 3; + Class 4

Figure 8: Original training data in the process measurement space, isothermal copolymerization example



✱ testing data (not symbolized according to classification)

Figure 9. Noise-free Testing data, 2-dimensional polymerization example

5. CONCLUSIONS

In this work two modelling approaches for fault detection are compared using a PPR algorithm: a classification approach where the model is trained based on the class number versus an estimation based approach where the value of the process variable defining the fault is identified and then the class is identified based on that value. It was found that the classification approach generally outperforms the estimation based approach for nonlinear systems, when the data is sparse and in the presence of measurement noise. This result holds provided that the training data is distributed approximately symmetrically with respect to the class boundaries. Then, the PPR algorithm based on such data correctly locates the class boundary since it uses the nearest neighbourhood concept to calculate the output.

References

[1] Bakshi, B. R.; Stephanopoulos, G.; WaveNet: a Multiresolution Hierarchical Neural Network with Localized Learning, AIChE Journal, vol. 39, no. 1, pp: 57-81, 1993

[2] Bakshi, B. R.; Utojo, U.; A common framework for the unification of neural, chemometric and statistical modeling methods, Analytica Chimica Acta, 384, 227-247, 1999.

[3] Friedman, J. H.; Stuetzle, Werner; Projection Pursuit Regression, Journal of the American Statistical Association, vol. 76, no. 376, p: 817-823, Dec. 1981

[4] Landry, R.; Model discrimination issues in polymer reaction engineering, Master Thesis, University of Waterloo, Canada, 1997

[5] Lou, S. J., Budman, H., Duever, T. A., Comparison of fault detection techniques, Journal of Process Control, v 13, n 5, 2003, p 451-464.

[6] Yoon, S., MacGregor, J. F.; Principal Component Analysis of Multiscale Data for Process Monitoring and Fault Detection, AIChE Journal, 50, 2891-2903, 2004.