Journal of Chemical and Natural Resources Engineering, Special Edition:58-71

ESTIMATION OF FATTY ACID COMPOSITION USING PLS-BASED MODEL

A.AHMAD[1], L.W. PIANG[1]

ABSTRACT

Given sufficiently rich and appropriate pre-treated data, Partial Least Square (PLS) models are able to accurately represent process dynamics. However, when applied to a fatty acid distillation process over broader ranges of operating conditions, the model was found not adequate. To incorporate nonlinear estimation feature, neural networks were incorporated to form Neural Network (NNPLS) and a further modified method known as Nested NNPLS. The results obtained proved that the Nest-NNPLS model provided the best estimation capability and should therefore be developed further as a potential candidate for on-line estimation of chemical product composition.

Key Words: Partial Least square, inferential estimation, data scaling.

1.0 INTRODUCTION

Distillation control system must continuously hold product compositions as well as other important process variables at their set points to satisfy desired operational objectives. In particular, control of product composition is difficult because the variable of interest cannot be measured economically on-line. Furthermore, the analyzers must also be equipped with additional facilities to protect against vibration, and harsh weather conditions. This makes the cost to escalate. More importantly, all on-line analyzers suffer from measurement lag and sampling delay and if the overall duration is significantly large, which is often the case, effective control cannot be established. The problem is further intensified by fouling of the sampling line, gradually accumulated with time. This scenario practically hampers the use of on-line analyzer in composition control of a distillation column.

As an alternative strategy, product compositions in industrial distillation columns are controlled by fixing the operating conditions so that thermodynamics equilibrium is fixed. Temperatures at selected locations are then used to infer the desired product compositions. While this idea is effective for binary system, inaccuracies are introduced when applied to multi-component mixtures. Following some adjustments aided by practical experiences, acceptable controls are often obtained, especially when the plant is in its nominal condition.

One viable solution is to make use of inferential estimation strategy that utilises measurable secondary variables, such as temperature and pressure to capture the behaviour of product quality. This is typically done using one of the available process modelling tools. Theoretically, if an adequately accurate model is developed, consistent process behaviour can be reproduced, thus providing reasonable estimates of the product compositions at any desired frequency required by the control loops. This paper discusses the application of Partial Least Square Regression (PLS) in estimating the composition of C_{12} fatty acid in industrial scale distillation process.

2.0 PARTIAL LEAST SQUARE

[1] Department of Chemical Engineering, Faculty of Chemical and Natural Resources Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia.
Correspondence to : Arshad Ahmad (arshad@fkkksa.utm.my)

Partial least squares regression is one of the multivariate analysis methods. It is a linear system identification method that projects the input-output data down into a latent space, extracts a number of principal factors with an orthogonal structure, while capturing most of the variance in the original data. In line with this commonly used acronym, it is also called Projection to Latent Structures.

2.1 Model Structure

The schematic diagram of the PLS model is shown in Figure 1. The standard computation method of PLS model is the Non-linear Iterative Partial Least Squares (NIPALS) algorithm. There is an alternative method which is known as SIMPLS algorithm [1]. In this work, the NIPALS computational method will be discussed and used for model development.

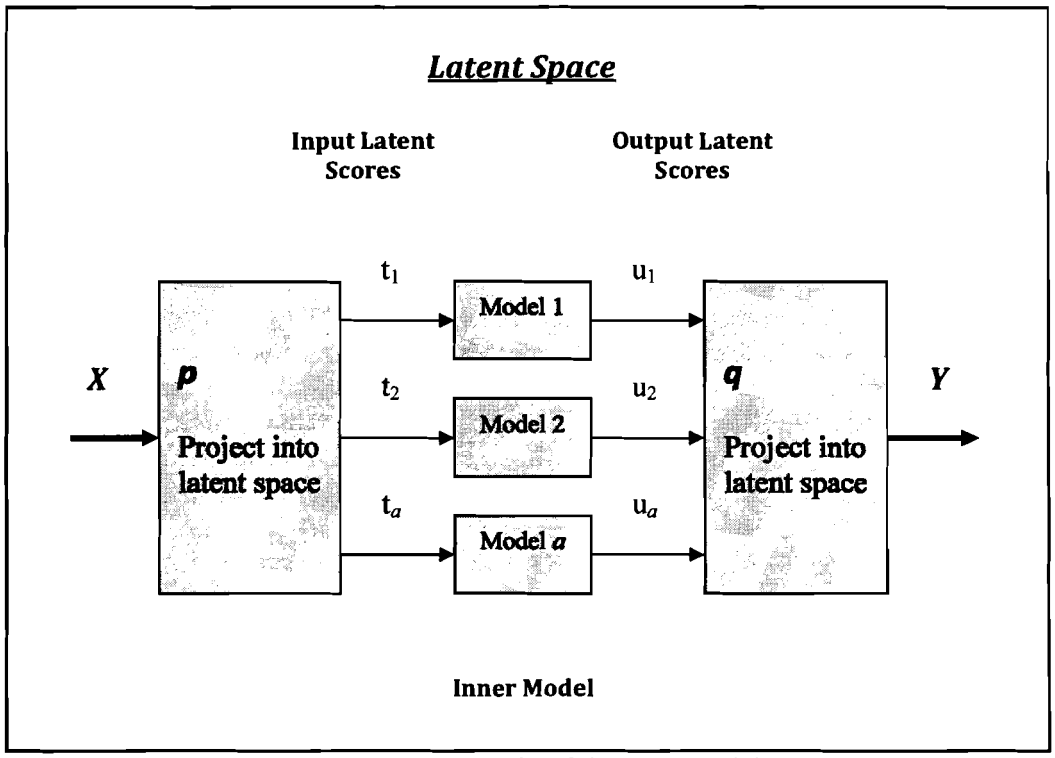

Figure 1 Schematic of the PLS model [3]

A PLS model consists of outer and inner relations. The outer relations are matrices of independent and dependent variables represented by **X** and **Y**, respectively. Assume that there are n output variables, y_i (i = 1, 2, ..., n) and m input variables, x_i (i = 1, 2, ..., m) with j sets of data. The matrices of **X** and **Y** can be formulated as follows:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & & & \\ x_{j1} & x_{j2} & \cdots & x_{jm} \end{bmatrix} \quad (1)$$

$$\mathbf{Y} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & & & \\ y_{j1} & y_{j2} & \cdots & y_{jn} \end{bmatrix} \tag{2}$$

The input **X** is projected into the latent space by the input-loading factor, **p** to obtain the input latent scores, **t**. Similarly, the output latent scores, **u** is obtained by projecting the output **Y** into latent space through the output-loading factor, **q**. Decomposition of **X** and **Y** are similar to principal components analysis and can be presented in matrix form (Equation 3 and 4.)

Outer relations:
$$\mathbf{X} = \mathbf{t}\mathbf{p}^{\mathrm{T}} + \mathbf{E} \tag{3}$$
$$\mathbf{Y} = \mathbf{u}\mathbf{q}^{\mathrm{T}} + \mathbf{F} \tag{4}$$

The matrices **E** and **F** are residuals of **X** and **Y**, respectively. **X** and **Y** are linked with a linear regression equation called inner relation to capture the relationship between the inputs and output latent scores. The notation of the inner relation is written in Equation 5 with $\hat{\mathbf{u}}$ is the predicted output latent scores, b is the regression coefficient, and **f** is the residual matrix.

Inner relation:
$$\hat{\mathbf{u}} = \mathbf{t}\, b + \mathbf{f} \tag{5}$$

The procedure of determining the scores and loadings factor is carried out sequentially from the first to the ath dimension. Scores and loading vectors for each dimension is calculated from the previous residual matrices as shown in Equation 6 and 7, where initially $\mathbf{E}_0 = \mathbf{X}$ and $\mathbf{F}_0 = \mathbf{Y}$.

For **X**,
$$\mathbf{E}_a = \mathbf{E}_{a-1} - \mathbf{t}_a \mathbf{p}_a^{\mathrm{T}} \tag{6}$$
For **Y**,
$$\mathbf{F}_a = \mathbf{F}_{a-1} - \mathbf{u}_a \mathbf{q}_a^{\mathrm{T}} \tag{7}$$

Calculation of the inner and outer relations is performed until the last dimension a, or when residual matrices are below certain threshold. Details on the implementation of the algorithm follow the procedure described by Geladi and Kowalski [2].

2.2 Model Development

The model development (or conveniently termed as training) of a PLS-based model is as summarised in Figure 2. Firstly, data is quality input-output data are generated. This is followed by some pre-processing tasks depending on the source of data itself. Scaling is normally required, for example, data is scaled around zero mean and unit variance. The data is then split into training and validation sets. Then the model is trained using least-squares method, once completed, evaluated based on some selected performance criteria. Adjustments are made if the model is found not adequate.

The mean and standard deviation values for each variable were calculated based on the entire set of data through Equation 8.

$$x_{ms} = (x - \bar{x}) / \sigma_x \tag{8}$$

Here, x indicates the original data, x_{ms} is the mean-scaled data, $\bar{x}$ is the mean value and σ_x is the standard deviation. Following the scaling stage, the input and output data were ready to be used for model training and validation. Training is carried out using "least-squares method" and the calculation is done for each dimension, a. Details of this algorithm are available elsewhere [2],[3].

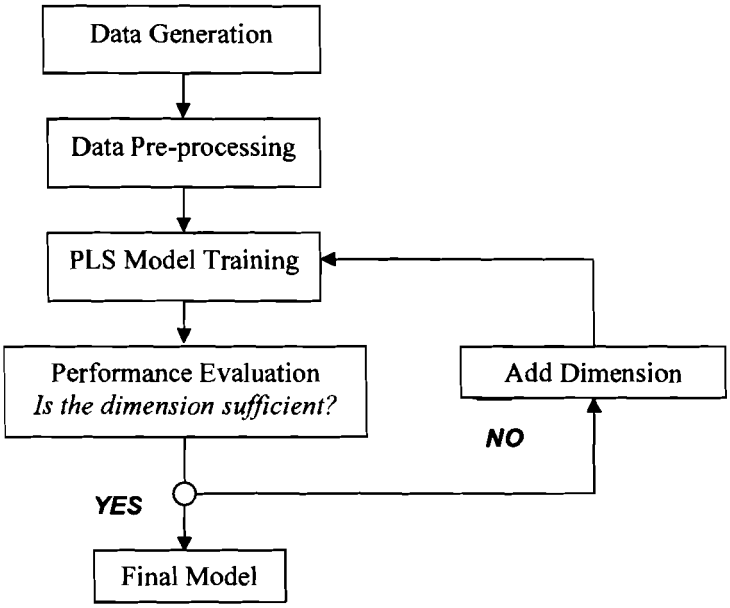

Figure 2 : Algorithm of developing PLS-based model

2.3.1 Model Validation

As mentioned earlier, fittings of inner and outer models are carried out sequentially until the last dimension. These are often determined using cross-validation to avoid over-fitting. Although the number of dimensions that gives the lowest residual on the test data is desirable, the model may also fit noise data set which may results in poor prediction. In order to obtain a reliable model, it is important to balance the number of dimensions against the residual error. The procedure of cross-validation is summarised as follows [4]:

i. Split the data set into a training set and a testing set.
ii. Construct a number of mappings using the training set.
iii. Validate the updated model with the testing set.
iv. Repeat the training process until the prediction error of the testing set reaches a minimum and start to increase.

After the construction of estimation model from reference data set in which both input and output variables are made available, the model is ready for prediction of new output variables when only values of input measurement are given. The new input variables are transformed into matrix $\mathbf{X}_{new}$ after scaling around zero mean and unit variance. The new predicted outputs $\hat{\mathbf{Y}}$ are determined as follows:

$$\hat{\mathbf{Y}} = \mathbf{X}_{new}\boldsymbol{\alpha} \tag{9}$$

Here, $\boldsymbol{\alpha}$ is a prediction coefficient determined using input weight ($\mathbf{w}$) and loading factors ($\mathbf{p}$ and $\mathbf{q}$).

$$\alpha = \mathbf{w}(\mathbf{p}^T\mathbf{w})^{-1}\mathbf{q}^T \tag{10}$$

The performance of the estimation model is evaluated based on the mean squared error of prediction (MSE) and explained prediction variance (EPV) in percent, which are calculated based on the following equation [5]:

$$\text{MSE} = \frac{1}{N}\sum_{n=1}^{N}(x(n) - \hat{x}(n))^2 \tag{11}$$

$$\text{EPV} = \left\{1 - \frac{\sum_{n=1}^{N}(x(n) - \hat{x}(n))^2}{\sum_{n=1}^{N}(x(n) - \bar{x})^2}\right\} \times 100 \tag{12}$$

Here, x is a measurement value, $\hat{x}$ is the estimated value, $\bar{x}$ is the mean value of measurements, and N is the number of measurements.

MSE and EPV are used to indicate the accuracy of the estimation over actual value and the statistical properties of the estimation model, respectively. EPV of **X** indicates how much of the **X** block is used in the estimation model and EPV of **Y** indicates how good the model is.

3.0 RESULTS AND DISCUSSION

The study was implemented on an industrial scale fatty acid distillation column. All required data were generated from a dynamic simulation of the column using Hysys software [6]. The model was first trained using a training data in order to obtain associate score factors and followed by model validation. The MSE for each dimension Figure 3.

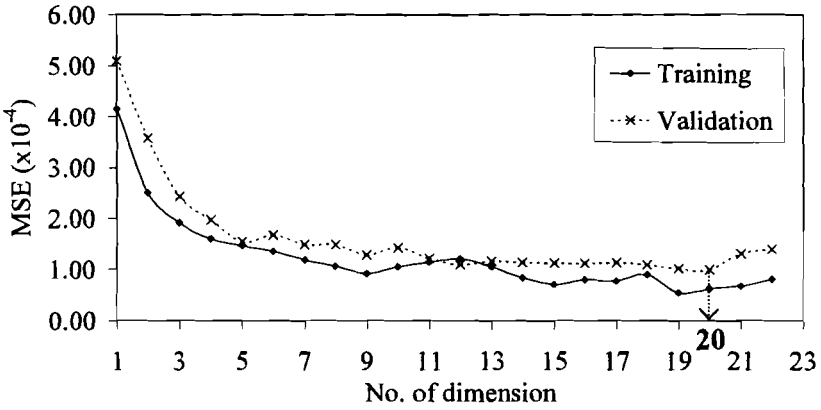

Figure 3 MSE of training and validation data using PLS model

The number of dimension indicates how much iteration that is required for the residual matrices to reach certain threshold. It was determined using cross-validation technique, where the training was stopped when the prediction error of the testing set reached a minimum and started to increase. It was noted the prediction error reached the *early stopping criteria* of cross-validation before the 20th dimension but it did not give the

optimum performance. This was due to the problem of local minima. In order to avoid convergence at local minima, the iteration was allowed to continue a model with the lowest MSE for the validation data was obtained. This was taken as the optimum configuration as summarised in Table 1.

Table 1 Training and validation results of PLS model

No. of dimensions	20
Cumulative variances explained (%) in X-Block	95.8020
Cumulative variances explained (%) in Y-Block	99.3281
Training MSE	0.6227
Validation MSE	0.9915

The result in Table 1 shows that the model used about 95.8% of the original input data, X, to estimate the output, and successfully captured about 99.3% of the original output data, Y. Comparison between the estimation and the actual output is given in Figure 4. Here, solid line and dotted line represent the actual and estimated outputs respectively. It is noted that the estimated output reasonably matched the actual composition in both training and validation set. The calculated MSE for training and validation data were 6.227×10^{-5} and 9.915×10^{-5}, respectively. Based on these results, the PLS-based estimation model was considered successfully constructed.

3.1 MODEL EVALUATION

Most forms of data-based models should be able to reproduce the training data (e.g., Figure 4a) accurately. Similarly, if other data is introduced and the data is of similar pattern and within similar ranges of operating condition, the estimation is also expected to be reasonably accurate (e.g., Figure 4b). However, since the model is to be used in various broader operating conditions while carrying out process estimation task, scrutiny has to be made to ascertain that the proposed model has the capability.

3.1.1 Test-Cases Description

The test-cases considered were four sets of data with total 1000 minutes simulation time, created with output data (C_{12} mole fraction) within 0.91 to 0.99. The characteristic of these data sets are summarised in Table 2.

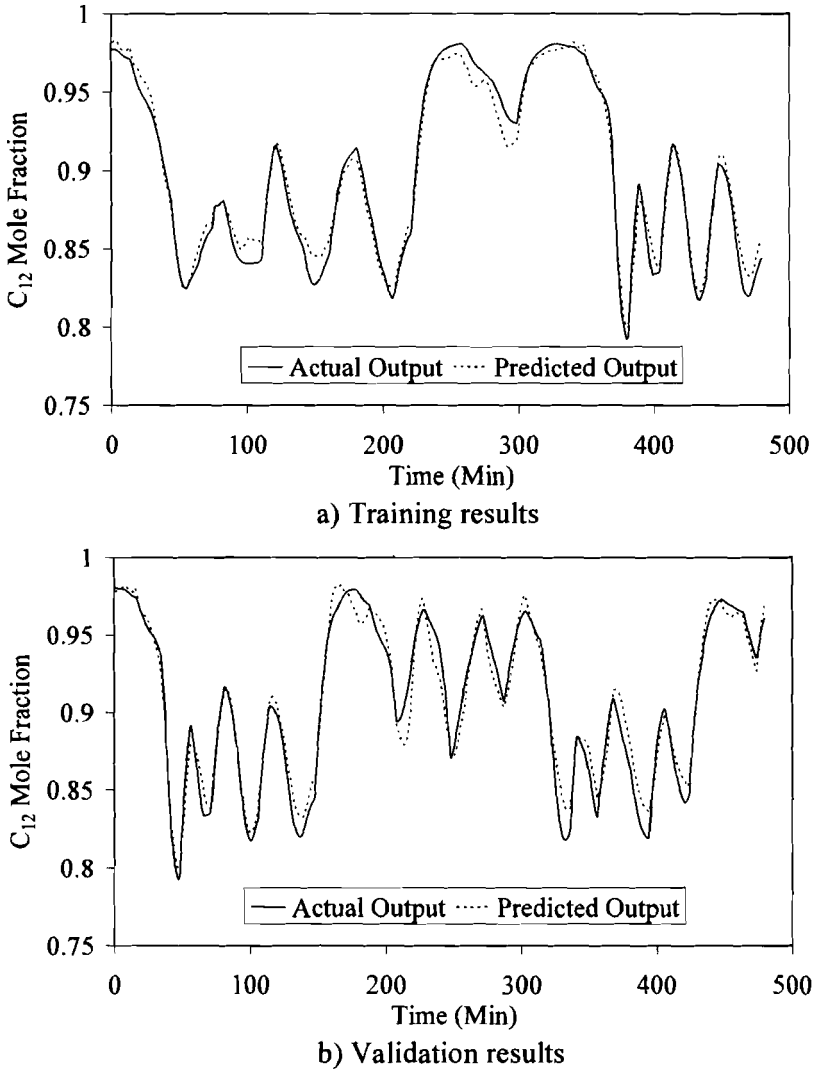

Figure 4 Training and validation results using PLS model

Table 2 Characteristic of data set for process estimation

Data set	Characteristic
A1	Moderate oscillation
A2	Mild oscillation
A3	Moderate oscillation with output range within 0.91 and 0.95
A4	Moderate oscillation with output range within 0.95 and 0.99

3.1.2 Model Performance

Estimation results for all sets of data are as plotted in Figure 5. The performance of the estimation model vary depending on the characteristic of the operating data. Good estimation was obtained only in one out of four sets of data, which was the normal data set with mild oscillation (Data A2). The estimated output managed to track the movement of actual data smoothly and an MSE of 6.2427×10^{-6} was obtained. In other

cases, *i.e.* in the case of Data A1, A3 and A4, the model failed to provide reasonable accuracy. Although the predicted output fell within the range of output values, the corresponding curves did not match the actual trend. The MSE of these data were considerable high, i.e., 9.8170×10^{-5} (Data A1), 2.4811×10^{-4} (Data A3), and 6.8711×10^{-5} (Data A4).

Sub-standard prediction obtained can be the results of many issues. One prominent factor in the perspective of chemical process modelling is the fact that most processes are nonlinear, while PLS model is not apt in nonlinear estimation. An appropriate non-linear estimation model is therefore necessary to provide better prediction of C_{12} composition.

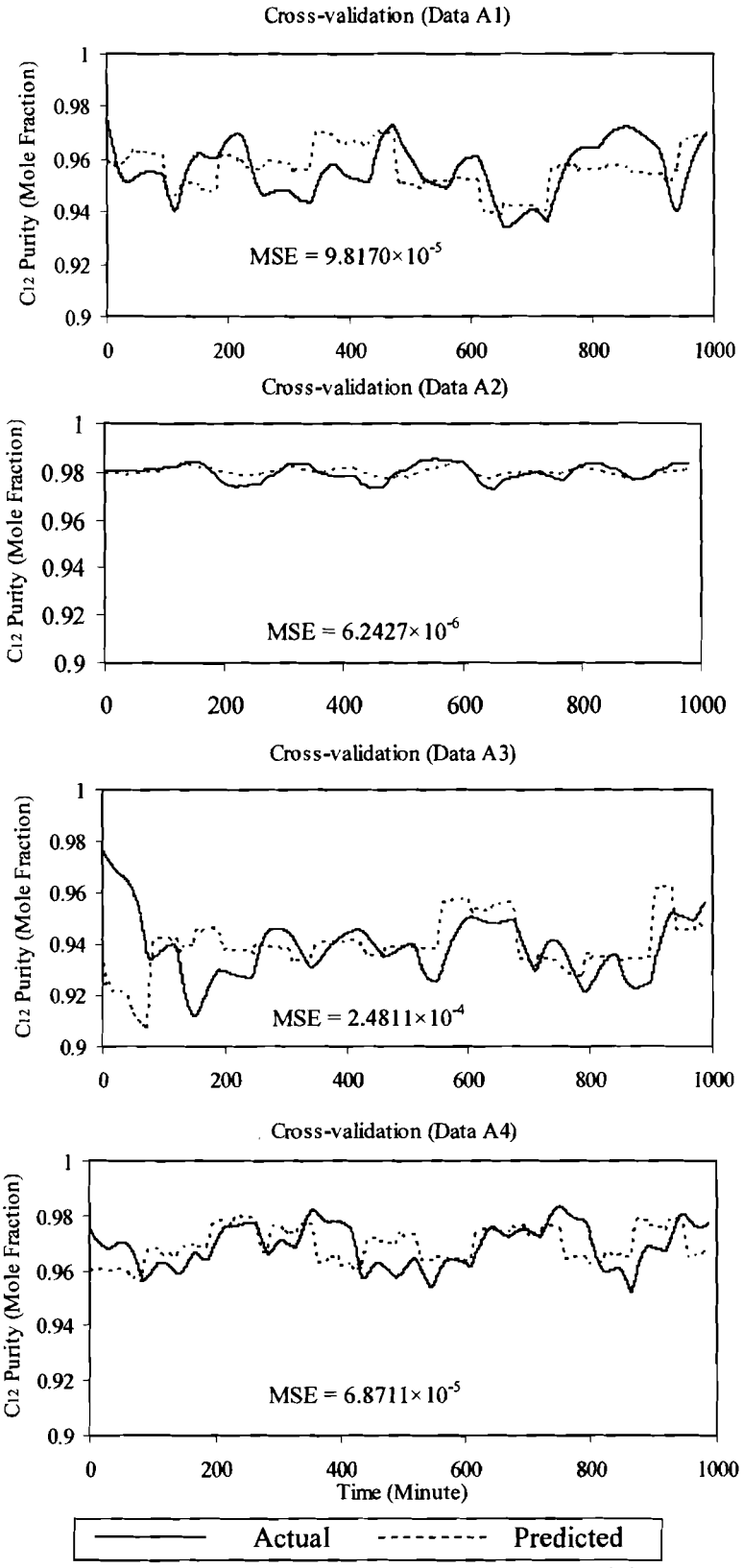

Figure 5 Estimation results using PLS model

3.2 PLS MODEL ENHANCEMENT

Extensions of PLS to include nonlinear capabilities can be implemented in variety of ways. In this case, a neural network model is integrated with a conventional PLS and the model is named neural network PLS (NNPLS). With such a combination, the model should be able to handle both correlated variables and non-linear behaviour owing to the non-linear mapping advantage from neural network and multivariate analysis.

3.2.1 Neural Net – PLS Model (NNPLS)

The schematic diagram of NNPLS model adopted in this study is shown in Figure 6. The structure is similar to what has been proposed by previously [7]. The outer relations are kept linear to transform the original data into score factors (**u** and **t**), and neural network is used in the inner relation as written in Equation 13 with $\mathcal{N}(\bullet)$ representing the non-linear function of a neural network.

$$\hat{\mathbf{u}}_a = \mathcal{N}(\mathbf{t}_a) + \mathbf{r}_a \tag{13}$$

In developing the model, data being used to train the network are pre-processed by PLS outer transform. Transformation of data is required to decompose a multivariate regression problem into a number of univariate sub-problems. Thus, only SISO networks are required. This is advantageous because SISO networks can be trained more conveniently compared to MIMO model. Furthermore, the number of local minimum is expected to be fewer since the network is smaller.

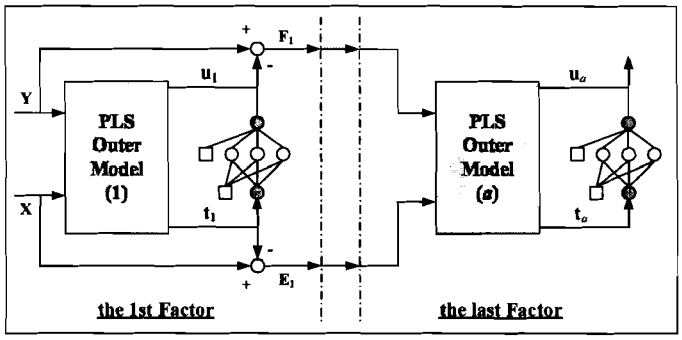

Figure 6 A schematic illustration of NNPLS model [7]

3.2.2 Nested Neural Net – PLS Model (Nest-NNPLS)

Further improvement to the NNPLS is proposed here by extending the nested-PLS [10] into a nested NNPLS (Nest-NNPLS). The method is built on the work of Baffi and co-workers ([8],[9]) to overcome the multi-colinearity issue encountered with the error-based weight updating procedure. Unlike other PLS models, the nested algorithm includes both inner and outer PLS algorithm that link internally. The loading factors, **t**, **u**, **p** and **q** are extracted from the outer algorithm while the inner algorithm derive the weight vectors, **w** for the outer PLS algorithm. The implementation structure of the nested PLS algorithm is illustrated in Figure 7.

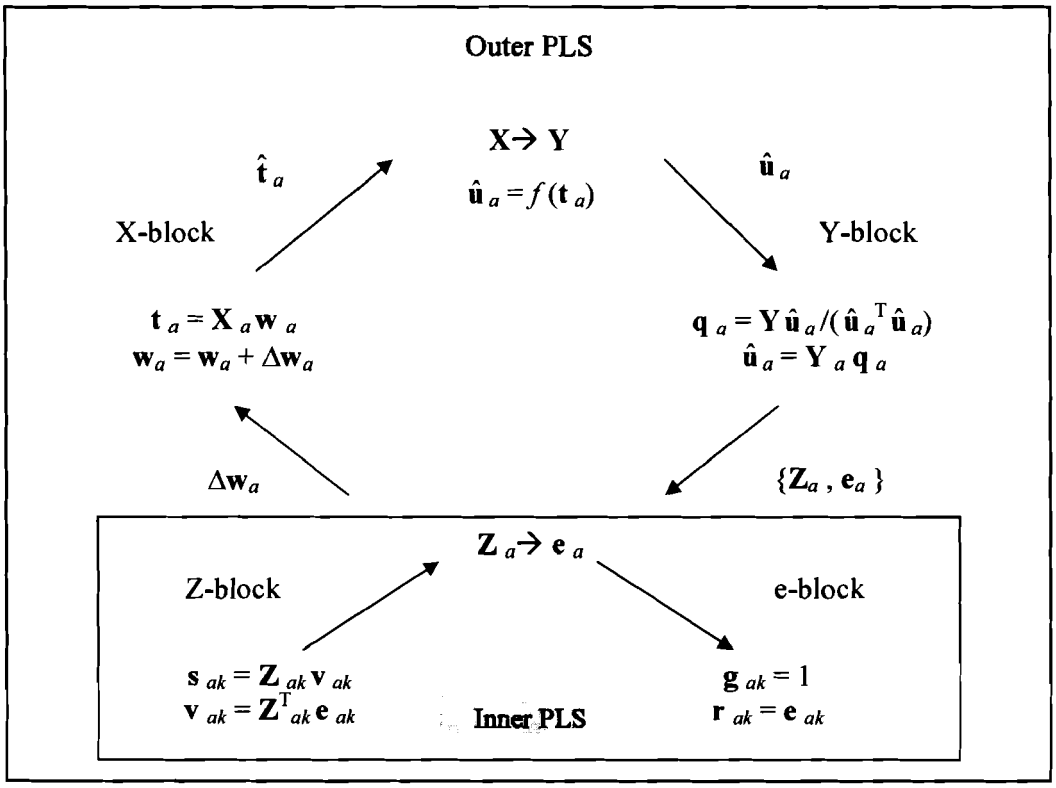

Figure 7 : Illustration of the nested PLS structure [10]

The training approach used of the outer PLS was similar to the conventional linear PLS, but for the inner PLS, systematic procedures proposed by Li et al. [10] were modified to incorporate neural network applications. Details are provided in the work of Lim [11].

Table 3 compares the training and validation results of the various PLS-based models. Encouraging results were obtained with MSE's below 2×10^{-4}. Note that the linear model used more than 95% of the original input data to capture about 99.3% of output data. On the other hand, around 98% of the outputs were explained using about 2% of input data for the case of Nest-NNPLS. Based on the training results, the MSE for the NNPLS is slightly higher than the other two models.

Table 3 Training and validation results of nested PLS model

Model	No. of dimension	MSE ($\times10^{-4}$)		X % [a]	Y % [b]
		Training	Validation		
PLS	20	0.6227	0.9915	95.8020	99.3281
NNPLS	5	1.3826	1.8559	71.4093	39.3188
Nest-NNPLS	6	0.7508	0.8970	2.0201	97.4741

[a] Relative cumulative variances explained (%) in X-Block
[b] Relative cumulative variances explained (%) in Y-Block

When the models were tested on the four test-sets, the results confirmed that both the NNPLS and Nest-NNPLS outperformed the standard PLS model. The summary of these results are presented in Table 4. Nest-NNPLS model demonstrated the best results

with 5.215×10^{-5} as the average MSE. The predicted output of all data sets using nested NNPLS was plotted in Figure 8.

Table 4 Off-line estimation results using non-linear PLS

Error of prediction (MSE×10^{-4})	**PLS**	**NNPLS**	**Nest-NNPLS**
Data A1	0.9817	0.6133	0.5378
Data A2	0.0624	0.0445	0.0417
Data A3	2.4811	1.3639	1.1204
Data A4	0.6871	0.4113	0.3859
Average	1.3833	0.7962	0.5215

4.0 CONCLUSION

In this study, the development of an inferential estimator using the nested NNPLS model has been described. The model was able to estimate the composition of C_{12} fatty acid from the distillate stream of the light-cut column using secondary variables such as stage temperatures, reflux flow rate, pump-around flow rate, feed flow rate and feed temperature. Further modification was made to improve the capability of the model in on-line implementation. Various cases with different operating conditions were designed in order to evaluate the accuracy and robustness of the model. Reasonable prediction results were obtained after several enhancements. Following these extensions, the final inferential estimation model was capable of providing fast and accurate estimation of product quality, and can be concluded as a reasonable substitute device for composition measurement. Such techniques are useful for process monitoring and control in process plants.

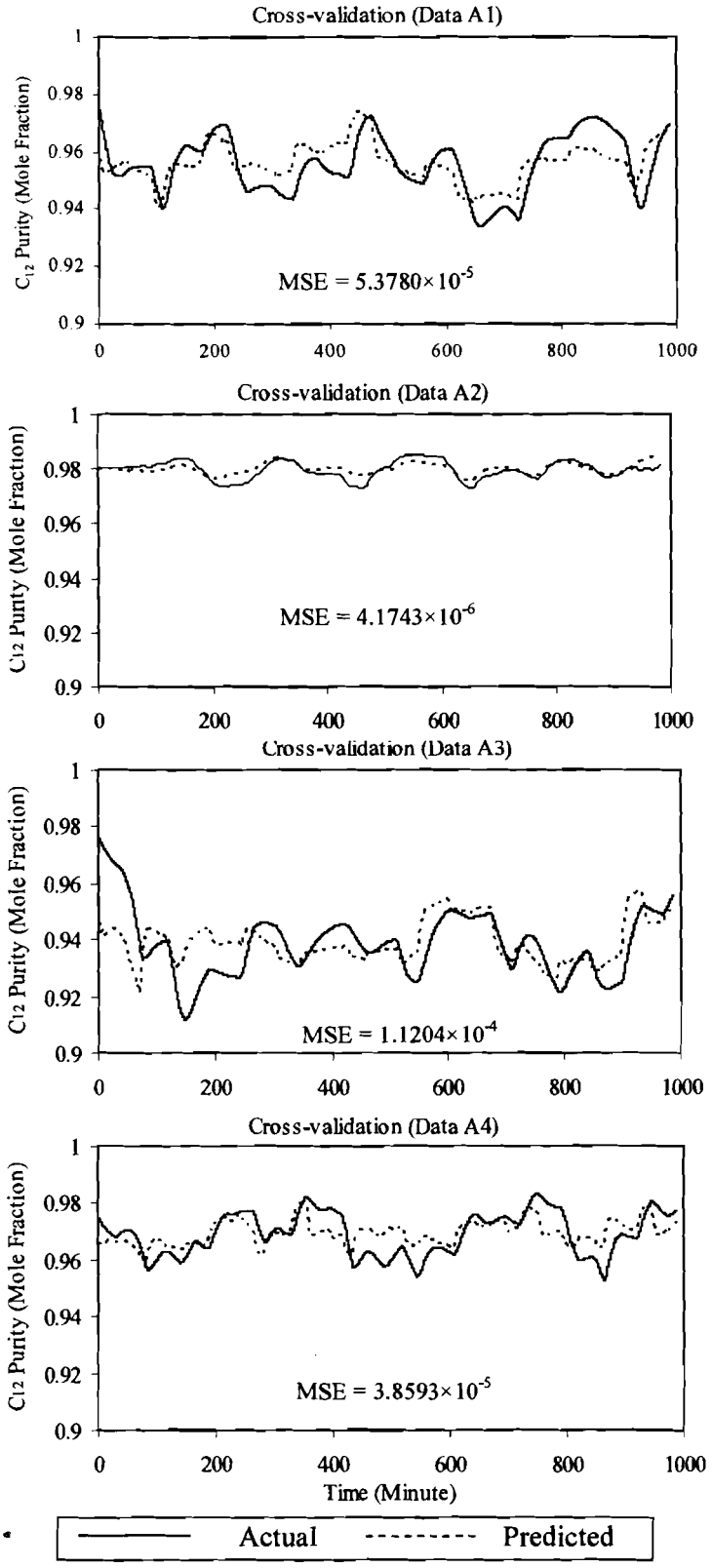

Figure 8 Estimation results using Nest-NNPLS model

REFERENCES

[1] De Jong, S. 1993. SIMPLS: An alternative pproach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems.* 18: 251-263.

[2] Geladi, P., B.R. Kowalski. 1986. Partial Least-Squares Regression: A Tutorial. *Analytica Chimica Acta.* 185: 1-17.

[3] Adebiyi, O.A., A.B. Corripio. 2003. Dynamic Neural Networks Partial Least Squares (DNNPLS) Identification of Multivariable Processes. *Computers & Chemical Engineering.* 27(2): 143-155.

[4] Pollard, J.F., M.R.Broussard, D.B. Garrison, K.Y.San,1992. Process Identification Using Neural Networks. *Computers and Chemical Engineering.* 16(4): 253-270.

[5] Kano, M., K.Miyazaki, S. Hasebe, I. Hashimoto, I. 2000. Inferential Control System of Distillation Compositions Using Dynamic Partial Least Squares Regression. *Journal of Process Control.* 10(2): 157-166.

[6] Ahmad, A., T.S. Wong, L.Y. Ling, 2001. Dynamic Simulation for a Palm Oil Fractionation Process. *Proceeding of the Regional Symposium on Chemical Engineering.* October 29-31. Bandung, Indonesia. M012-M018.

[7] Qin, S.J., T.J. McAvoy. 1992. Nonlinear PLS modelling Using Neural Networks. *Computers & Chemical Engineering.* 16(4): 379-391.

[8] Baffi, G., E.B. Martin, A.J. Morris, A.J. 1999. Non-linear Projection to Latent Structures Revisited (the Quadratic PLS Algorithm). *Computers & Chemical Engineering.* 23(3): 395-411.

[9] Baffi, G., E.B. Martin, A.J. Morris. 1999. Non-linear Projection to Latent Structures Revisited (the Network PLS Algorithm). *Computers & Chemical Engineering.* 23(9): 1293-1307.

[10] Li, B., P.A. Hassel, A.J. Morris, A.J. E.B. Martin, E.B. 2003. A Non-linear Nested Partial Least-Squares Algorithm. *Computational Statistical & Data Analysis.*

[11] Lim, W.P. 2005. Inferential Estimation and Control of Chemical Process Using Partial Least Squares Based Model. M.Eng. Thesis, Universiti Teknologi Malaysia.