
Inferential control with the aid of modified QPLS-based soft sensor for an industrial FCCU fractionator

Xuemin Tian, Ling Tu and Minghui Yang

College of Information and Control Engineering,
China University of Petroleum (Hua Dong),
Dongying, Shandong 257061, China
E-mail: tianxm@hdpu.edu.cn
E-mail: tuling393@sohu.com
E-mail: yangmhui@sohu.com

Sheng Chen*

School of Electronics and Computer Science,
University of Southampton,
Southampton SO17 1BJ, UK
E-mail: sqc@ecs.soton.ac.uk
*Corresponding author

Abstract: A modified quadratic partial least squares (MQPLS) algorithm based on non-linear constrained programming is proposed, in which a sequential unconstrained minimisation technique is employed to calculate the outer input weights and the parameters of inner relationship. Other existing quadratic partial least squares (QPLS) algorithms are also reviewed and compared with the proposed MQPLS in the applications to two datasets, one being an artificial dataset and the other being the real data from an industrial fluidised catalytic cracking unit (FCCU) main fractionator. It is shown that the MQPLS not only can explain better the underlying variability of the data but also achieves improved modelling and predictive performance over the existing QPLS algorithms. An inferential control system is implemented on the distributed control system for an industrial FCCU main fractionator, in which the soft-sensor is built based on the MQPLS algorithm to estimate the diesel oil solidifying point online and the controller is established via a constrained dynamic matrix control algorithm. Experimental results obtained demonstrate that the inferential control system with the aid of the MQPLS soft sensor works much better than the original tray temperature control system and it realises well the boulder control of diesel oil solidifying point.

Keywords: non-linear systems; neural networks; non-linear partial least squares; NPLS; soft sensing; inferential control; fractionator; dynamic matrix control; DMC.

Reference to this paper should be made as follows: Tian, X., Tu, L., Yang, M. and Chen, S. (2010) 'Inferential control with the aid of modified QPLS-based soft sensor for an industrial FCCU fractionator', *Int. J. Modelling, Identification and Control*, Vol. 11, Nos. 1/2, pp.59–70.

Biographical notes: Xuemin Tian is a Professor of Process Control at China University of Petroleum (Hua Dong). He received his BEng from Huadong Petroleum Institute and his Master from Beijing University of Petroleum. From September 2001 to June 2002, he was a Visiting Professor at University of California in Santa Barbara. His research interests are in modelling, advanced process control and optimisation for petro-chemical process.

Ling Tu obtained her Master degree in Control Theory and Engineering from China University of Petroleum in 2005. She is currently a Research Assistant in College of Information and Control Engineering, China University of Petroleum. Her research interests cover chemical process modelling and simulation, soft measurement, as well as inferential control.

Minghui Yang received his BEng in 2001 and his Master in 2004, both in Control Theory and Engineering, from China University of Petroleum. He is currently a Lecturer in the College of Information and Control Engineering, China University of Petroleum. His research interests include theory and application of control system, especially chemical process modelling and simulation, advanced process control.

Sheng Chen received his BEng from Huadong Petroleum Institute, Dongying, China, in January 1982 and his PhD from the City University, London, UK, in September 1986, both in Control Engineering. He was awarded the DSc by the University of Southampton, Southampton, UK, in 2005. He has been with the School of Electronics and Computer Science, the University of Southampton, since September 1999. His research interests include wireless communications, machine learning, finite-precision digital controller design and evolutionary computation.

1 Introduction

Many variables, which characterise the ‘quality’ of the final product in chemical processes, are often difficult to measure in real-time, and hence, cannot be used directly in a feedback configuration for process control. Most online quality analysers, such as gas chromatographs and near-infrared analysers, suffer from large measurement delays as well as high investment and maintenance costs. Under these circumstances, a common alternative is to set up soft sensors to infer the product properties (primary variables) by employing some auxiliary measurements (secondary variables), thus building an inferential control scheme.

Statistic regression techniques have been extensively used in establishing soft sensing models from historical data. Among many regression techniques, the partial least squares/projection to latent structure (PLS) method has been proved to be a powerful tool for the problems where data are noisy and highly correlated as well as where the numbers of observations are limited (Berglund and Wold, 1997; MacGregor et al., 1991). The power of PLS lies in the fact that it projects the input-output data onto a latent space to extract a number of principle components with an orthogonal structure, while capturing most of the variance in the original data. Therefore, the PLS can overcome the limitations of the traditional least squares (LS) regression. When dealing with highly correlated multivariate data, the traditional LS method often results in high variance in parameter estimation or even singular solution.

However, many industrial processes exhibit significant non-linear behaviours. As a linear regression method, the PLS is inappropriate for modelling non-linear systems. Hence, various non-linear PLS (NPLS) methods have been proposed in the literature which extend the PLS model structure to capture non-linearities of the systems. A successful step towards NPLS modelling is the quadratic PLS (QPLS) proposed by Wold et al. (1989). In the QPLS, second order polynomial (quadratic) regression is used to fit the function between each pair of input and output score vectors, namely, the inner relation of PLS. Other ‘generic’ NPLS methods, such as the spline PLS (SPLS) (Wold, 1992), the neural networks PLS (NNPLS) (Qin and McAvoy, 1992; Andersson et al., 1996; Wilson et al., 1997; Liu et al., 2000; Wang and Yu, 2004) and the fuzzy PLS (FPLS) (Bang et al., 2002) have been developed to model generic non-linearities. As their respective names suggested, the SPLS uses spline function (quadratic or cubic) as the inner model and the NNPLS uses neural networks as the inner model, while the FPLS uses Takagi-Sugeno-Kang

fuzzy model as the inner model. All the above-mentioned NPLS algorithms have been developed from the non-linear iterative PLS (NIPALS) algorithm (Geladi and Kowalski, 1986), which is referred to as the ‘engine’ of the PLS methodology.

The problem of the input weight updating in the NPLS structure is firstly considered by Wold et al. (1989) and the benefits achievable by applying an updating procedure to the parameters of the NPLS model is also proved. This has motivated further research for the input weight updating procedure in the NPLS model. By modifying the input weight updating procedure of Wold et al. (1989), Baffi et al. (1999a, 1999b, 2000) have derived an error-based input weight updating approach. The first contribution of this paper is to formulate the weight and parameter updating procedure as a constrained non-linear optimisation problem. Sequential unconstrained minimisation technique (SUMT) is then utilised to calculate the outer input weights and the parameters of quadratic inner relation. The proposed updating method overcomes the drawbacks of the need for pseudo-inverse and high computational burden that exist in the original error-based input weight updating approach (Baffi et al., 1999a, 1999b, 2000). The new updating method is combined with the original QPLS in this paper, leading to a modified QPLS (MQPLS) algorithm. It is worth mentioning that this new weight updating approach is applicable to any NPLS algorithm.

Control of industrial fluidised catalytic cracking unit (FCCU) main fractionators has been a classical and difficult control problem. Traditionally, the product quality control is achieved by tray temperature control, which has found wide-ranging applications in chemical plants when online analysers or product quality soft sensors are unavailable. The second contribution of this paper is to design an inferential controller aided by the MQPLS-based soft sensor. We apply the MQPLS-based soft sensor to predict the diesel oil solidifying point, which in turn enables us to design an inferential controller for close-loop quality control of the fractionator. Many works (Kano et al., 2000, 2003) have proposed the cascade inferential control system in which the set point of tray temperature controller is given by the output of quality inferential controller. However, in such a control scheme, the inner temperature controller has a greater influence on the performance of the whole system and its complex structure brings some major difficulties to operators. We provide a new inferential control scheme which is capable of switching between the tray temperature controller and the quality inferential controller without causing disturbance. The proposed inferential controller is established via a simplified dynamic matrix control (DMC)

algorithm (Zhan and Grassi, 1999; Wang and Li, 2003; Zhao et al., 2003; Biswas et al., 2007).

The paper is organised as follows. In Section 2, the NPLS is reviewed and the error-based input weight updating procedure due to Baffi et al. (1999a, 1999b, 2000) is introduced. Section 3 proposes a new weight and parameter updating method. In particular, combining this new updating procedure with the QPLS algorithm leads to the MQPLS algorithm. A comparative case study for various QPLS algorithms is also given in this section. Section 4 introduces an inferential controller for an industrial FCCU main fractionator, in which the soft-sensor is build based on the MQPLS algorithm to estimate the diesel oil solidifying point and the controller is established using a simplified DMC algorithm. Section 5 offers our conclusions.

2 Non-linear partial least squares

Consider the generic case where N samples have been collected on M regression variables (the independent variables) $\mathbf{x}_m = [x_m(1)x_m(2)\cdots x_m(N)]^T$, $1 \leq m \leq M$ and K response variables (the dependent variables) $\mathbf{y}_k = [y_k(1)y_k(2)\cdots y_k(N)]^T$, $1 \leq k \leq K$. The data can be arranged into the $N \times M$ matrix $\mathbf{X} = [\mathbf{x}_1 \mathbf{x}_2 \cdots \mathbf{x}_M]$ and the $N \times K$ matrix $\mathbf{Y} = [\mathbf{y}_1 \mathbf{y}_2 \cdots \mathbf{y}_K]$. Typically, the matrix \mathbf{X} relates to the process variables (inputs), whilst the matrix \mathbf{Y} corresponds to the quality or reference variables (outputs). Before any regression algorithm is implemented, both input and output variables, \mathbf{x}_m and \mathbf{y}_k , are preprocessed to having zero mean and unit variance. Proper preprocessing prevents the score vectors from being biased towards variables with large magnitudes.

The PLS algorithm decomposes the input and output data matrices \mathbf{X} and \mathbf{Y} by projecting them onto the directions of the input weight \mathbf{w}_l and output weight \mathbf{c}_l to extract L pairs of the input score vectors \mathbf{t}_l and output score vectors \mathbf{u}_l , where $1 \leq l \leq L$ and $L < \min\{M, K\}$ is the number of score vectors extracted. The decomposition, referred to as the PLS outer relation, is given by (Berglund and Wold, 1997; Baffi et al., 1999a)

$$\mathbf{X} = \sum_{l=1}^L \mathbf{t}_l \mathbf{p}_l^T + \mathbf{E} \quad (1)$$

$$\mathbf{Y} = \sum_{l=1}^L \hat{\mathbf{u}}_l \mathbf{q}_l^T + \mathbf{F} \quad (2)$$

where \mathbf{p}_l and \mathbf{q}_l are known as the input and output loading vectors, respectively, \mathbf{E} and \mathbf{F} are the resulting residual matrices, respectively, while $\hat{\mathbf{u}}_l$ denotes an estimate of \mathbf{u}_l , which is determined by the PLS inner relation.

An appropriate number of components, L , required to describe the data structure, is generally determined by means of cross validation. A typical cross-validation method

for determining the data structure is to select L by minimising the predictive error sum of squares (PRESS) (Stone, 1974; Myers, 1990). Usually, most of the variances of the input and output variables can be accounted for by the first few score vectors, whilst the residuals are typically associated with the random noise in the datasets.

The traditional linear PLS defines a linear inner relation, namely, performs an ordinary LS regression between the pair of score vectors

$$\mathbf{u}_l = b_l \mathbf{t}_l + \mathbf{e}_l \quad (3)$$

with

$$b_l = \frac{\mathbf{t}_l^T \mathbf{u}_l}{\mathbf{t}_l^T \mathbf{t}_l} \quad (4)$$

The estimator $\hat{\mathbf{u}}_l$ is simply defined by $\hat{\mathbf{u}}_l = b_l \mathbf{t}_l$.

2.1 The basic principle of NPLS

To model the data with non-linear structure, the linear inner relation (3) is inadequate. Consider the NPLS with the generic non-linear inner relation

$$\mathbf{u}_l = \mathbf{f}(\mathbf{t}_l; \mathbf{b}_l) + \mathbf{e}_l \quad (5)$$

where \mathbf{b}_l denotes the parameter vector that defines the non-linear vector mapping \mathbf{f} . Different NPLS algorithms adopts different non-linear functions \mathbf{f} . For example, the QPLS algorithm (Wold et al., 1989; Baffi et al., 1999a) employs the second order polynomial (quadratic) regression for the inner mapping

$$\mathbf{u}_l = b_{0,l} \mathbf{1} + b_{1,l} \mathbf{t}_l + b_{2,l} \mathbf{t}_l^2 + \mathbf{e}_l \quad (6)$$

In this case, the parameter vector that specifies this quadratic mapping is $\mathbf{b}_l = [b_{0,l} \ b_{1,l} \ b_{2,l}]^T$. The basic NPLS algorithm is now summarised.

Denote $\mathbf{X}^{(0)} = \mathbf{X}$ and $\mathbf{Y}^{(0)} = \mathbf{Y}$, the original data matrices. In the l th step, the PLS decomposition extracts the pair of the input and output score vectors \mathbf{t}_l and \mathbf{u}_l by projecting $\mathbf{X}^{(l-1)}$ and $\mathbf{Y}^{(l-1)}$ onto the directions of \mathbf{w}_l and \mathbf{c}_l . The decomposition satisfies the following two conditions:

- 1 \mathbf{t}_l and \mathbf{u}_l capture the variances of $\mathbf{X}^{(l-1)}$ and $\mathbf{Y}^{(l-1)}$, respectively, as much as possible
- 2 the covariance between \mathbf{t}_l and \mathbf{u}_l is maximised.

The PLS outer relation is simply

$$\mathbf{X}^{(l-1)} = \mathbf{t}_l \mathbf{p}_l^T + \mathbf{E}_l \quad (7)$$

$$\mathbf{Y}^{(l-1)} = \hat{\mathbf{u}}_l \mathbf{q}_l^T + \mathbf{F}_l \quad (8)$$

After identifying the parameters of the inner relation (5), i.e., \mathbf{b}_l , the estimator of \mathbf{u}_l is calculated as

$$\hat{\mathbf{u}}_l = \mathbf{f}(\mathbf{t}_l; \mathbf{b}_l) \quad (9)$$

and the residual matrices are given by

$$\mathbf{X}^{(l)} = \mathbf{E}_l = \mathbf{X}^{(l-1)} - \mathbf{t}_l \mathbf{p}_l^T \quad (10)$$

$$\mathbf{Y}^{(l)} = \mathbf{F}_l = \mathbf{Y}^{(l-1)} - \hat{\mathbf{u}}_l \mathbf{q}_l^T \quad (11)$$

The procedure is repeated until $l = L$.

When the non-linear inner relation is chosen as the quadratic mapping (6), this NPLS algorithm becomes the QPLS algorithm. For a more comprehensive review, the reader is referred to the paper by Geladi and Kowalski (1986).

2.2 The error-based NPLS

Wold et al. (1989) described in their paper that ‘the present algorithm (QPLS) is fairly complicated and converge slowly when the data lack structure. Hopefully, this situation may be improved by better algorithms’. Wold et al. (1989), in fact, have identified the main drawbacks of merging non-linear regression techniques with the linear PLS framework. The main issue is concerned with the updating of input weights, as the value of $\mathbf{f}(\mathbf{t}_l; \mathbf{b}_l)$ depends on the value of \mathbf{w}_l . They then went on to consider the input weight updating in the NPLS structure. Although the weight updating procedure proposed by Wold et al. (1989) might be considered to be somewhat cumbersome in its approach, it does raise quite a few issues that are useful in solving the problem. After reviewing and discussing the input weight updating procedure proposed by Wold et al. (1989), Baffi et al. (1999a) developed three variants of input weight updating procedures and finally concluded that the error-based approach is the best method of the three procedures. This error-based input weight updating procedure is summarised in the following.

The mismatch or error, \mathbf{e}_l , between the value of \mathbf{u}_l given by $\mathbf{u}_l = \mathbf{Y} \mathbf{q}_l$ and the value of $\hat{\mathbf{u}}_l$ given by the non-linear mapping, $\hat{\mathbf{u}}_l = \mathbf{f}(\mathbf{t}_l; \mathbf{b}_l)$, can be denoted by

$$\mathbf{e}_l = \mathbf{u}_l - \hat{\mathbf{u}}_l \quad (12)$$

Through application of the first-order series expansion, \mathbf{e}_l can be written as

$$\mathbf{e}_l = \mathbf{u}_l - \hat{\mathbf{u}}_l = \mathbf{u}_l - \mathbf{f}|_0 = \frac{\partial \mathbf{f}}{\partial \mathbf{w}_l} \Big|_0 \Delta \mathbf{w}_l \quad (13)$$

where $\mathbf{f}|_0$ indicates that the calculation is based on the previous input weight value \mathbf{w}_l . By defining the partial derivative matrix

$$\mathbf{Z}_l = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{w}_l} \Big|_0 \right] \quad (14)$$

the mismatch can be written as $\mathbf{e}_l = \mathbf{Z}_l \Delta \mathbf{w}_l$ and the correction $\Delta \mathbf{w}_l$ can be regressed directly as follows

$$\Delta \mathbf{w}_l = (\mathbf{Z}_l^T \mathbf{Z}_l)^{-} \mathbf{Z}_l^T \mathbf{e}_l \quad (15)$$

where $(\mathbf{Z}_l^T \mathbf{Z}_l)^{-}$ denotes the pseudo-inverse of the matrix $\mathbf{Z}_l^T \mathbf{Z}_l$. Then the input weight is updated according to

$$\mathbf{w}_l = \mathbf{w}_l + \Delta \mathbf{w}_l \quad (16)$$

A new input score vector \mathbf{t}_l is calculated according to $\mathbf{t}_l = \mathbf{X} \mathbf{w}_l$. If \mathbf{t}_l converges, the updating procedure is over; otherwise the procedure is repeated.

Table 1 The error-based NPLS algorithm

Step	Summary
0	Preprocess \mathbf{X} and \mathbf{Y}
1	Set the output score vector \mathbf{u}_l to a column of \mathbf{Y}
2	Calculate the input weight vector \mathbf{w}_l : $\mathbf{w}_l = \frac{\mathbf{u}_l^T \mathbf{X}}{\mathbf{u}_l^T \mathbf{u}_l}$
3	Normalise \mathbf{w}_l to unit length: $\mathbf{w}_l = \frac{\mathbf{w}_l}{\ \mathbf{w}_l\ }$
4	Calculate the input score vector \mathbf{t}_l : $\mathbf{t}_l = \mathbf{X} \mathbf{w}_l$
5	Fit the non-linear inner relation: $\mathbf{b}_l \leftarrow \text{fit}[\mathbf{u}_l = \mathbf{f}(\mathbf{t}_l; \mathbf{b}_l) + \mathbf{e}_l]$
6	Calculate the non-linear prediction of \mathbf{u}_l : $\hat{\mathbf{u}}_l = \mathbf{f}(\mathbf{t}_l; \mathbf{b}_l)$
7	Calculate the output loading vector \mathbf{q}_l : $\mathbf{q}_l = \frac{\hat{\mathbf{u}}_l^T \mathbf{Y}}{\hat{\mathbf{u}}_l^T \hat{\mathbf{u}}_l}$
8	Normalise \mathbf{q}_l to unit length: $\mathbf{q}_l = \frac{\mathbf{q}_l}{\ \mathbf{q}_l\ }$
9	Calculate the new output score vector \mathbf{u}_l : $\mathbf{u}_l = \mathbf{Y} \mathbf{q}_l$
10	Update the input weight vector \mathbf{w}_l : $\mathbf{w}_l = \mathbf{w}_l + \Delta \mathbf{w}_l$ according to the error-based input weight updating procedure of Subsection 2.2
11	Normalise \mathbf{w}_l to unit length: $\mathbf{w}_l = \frac{\mathbf{w}_l}{\ \mathbf{w}_l\ }$
12	Calculate the new input score vector \mathbf{t}_l : $\mathbf{t}_l = \mathbf{X} \mathbf{w}_l$
13	Check the convergence of \mathbf{t}_l : If yes, go to Step 14; else go to Step 5
14	Fit the non-linear inner relation: $\mathbf{b}_l \leftarrow \text{fit}[\mathbf{u}_l = \mathbf{f}(\mathbf{t}_l; \mathbf{b}_l) + \mathbf{e}_l]$
15	Calculate the non-linear prediction of \mathbf{u}_l : $\hat{\mathbf{u}}_l = \mathbf{f}(\mathbf{t}_l; \mathbf{b}_l)$
16	Calculate the input loading vector \mathbf{p}_l : $\mathbf{p}_l = \frac{\mathbf{t}_l^T \mathbf{X}}{\mathbf{t}_l^T \mathbf{t}_l}$
17	Calculate the input residual matrix \mathbf{E}_l : $\mathbf{E}_l = \mathbf{X} - \mathbf{t}_l \mathbf{p}_l^T$
18	Calculate the output residual matrix \mathbf{F}_l : $\mathbf{F}_l = \mathbf{Y} - \hat{\mathbf{u}}_l \mathbf{q}_l^T$
19	If $l < L$, set $\mathbf{X} = \mathbf{E}_l$, $\mathbf{Y} = \mathbf{F}_l$ and repeat Steps 1 to 18 with $l = l + 1$; else stop

Source: Baffi et al. (1999a, 1999b, 2000)

By combining this error-based input weight updating procedure with the NPLS structure, Baffi et al. (1999a, 1999b) derived the error-based NPLS algorithm, which we summarise in Table 1. When the non-linear inner relation is chosen as the quadratic mapping (6), we refer to this error-based NPLS algorithm as the error-based QPLS algorithm. Without an input weight updating procedure, the first pair of input and output score vectors as well as the weight vector of the NPLS model remains the same as those of the linear PLS. By applying this input weight updating procedure, these quantities are different from those of the linear PLS and the final model has improved fitting and predictive capability (Baffi et al., 1999a, 1999b).

3 Modified NPLS

Although the above error-based NPLS algorithm offers more accurate modelling capability than the original NPLS algorithm without an input weight updating procedure, there are three aspects of this error-based input weight updating procedure which are worth investigating. Firstly, \mathbf{Z}_l is rank deficient, if the input dimension M is larger than the number of samples N and/or the partial derivatives of the inner relation are linearly correlated or alternatively they are correlated with the inner relation $f(\bullet; \bullet)$. Under these two conditions, the correction $\Delta \mathbf{w}_l$ cannot be obtained directly from equation (15) using the normal matrix inverse and numerical techniques are required to evaluate the pseudo-inverse $(\mathbf{Z}_l^T \mathbf{Z}_l)^-$, which is computationally expensive. Secondly, \mathbf{w}_l is updated iteratively until the input score vector is converged (refer to Table 1, Steps 4 to 13), which imposes a significantly increased computation burden. Thirdly, by applying the error-based input weight updating procedure, the NPLS model can capture larger output cumulative variance, but smaller input variance. This problem is also pointed out by Bang et al. (2002).

In this paper, we propose a new updating procedure for the outer input weight \mathbf{w}_l and inner parameter \mathbf{b}_l by modifying and extending the error-based method for updating \mathbf{w}_l , proposed by Baffi et al. (1999a, 1999b, 2000).

3.1 A new updating procedure with quadratic inner mapping

The objective of the error-based input weight updating procedure due to Baffi et al. (1999a, 1999b, 2000) is to find appropriate input weights of the PLS outer relation and this task is embedded in the iterative loop for finding parameters of the non-linear inner relation, see Table 1, Steps 5 to 13. The joint task of finding the optimal outer input weights and parameters of the non-linear inner mapping can in fact be formulated as a joint optimisation that minimises the regression sum of squared errors of the non-linear inner relation, which is a constrained non-linear programming problem. We will restrict to the case of the quadratic inner mapping (6). In this QPLS case, the optimal input weights

and polynomial coefficients of the quadratic inner mapping are obtained by solving the following non-linear constrained optimisation problem

$$\begin{aligned} & \min_{\mathbf{w}_l, \mathbf{b}_l} (\mathbf{u}_l - \hat{\mathbf{u}}_l)^T (\mathbf{u}_l - \hat{\mathbf{u}}_l) \\ & \text{s.t. } \|\mathbf{w}_l\| = 1 \\ & \quad \mathbf{t}_l = \mathbf{X} \mathbf{w}_l \\ & \quad \hat{\mathbf{u}}_l = b_{0,l} \mathbf{1} + b_{1,l} \mathbf{t}_l + b_{2,l} \mathbf{t}_l^2 \end{aligned} \quad (17)$$

where $\mathbf{b}_l = [b_{0,l} \ b_{1,l} \ b_{2,l}]^T$. We adopt a SUMT to transform the constrained optimisation problem (17) into a series of unconstrained non-linear programming problems. Then the Hooke-Jeeves' method (Hooke and Jeeves, 1961; Rao, 1996) is employed to solve these unconstrained non-linear programming problems. The initial values of \mathbf{w}_l and \mathbf{b}_l are supplied by the QPLS algorithm.

Table 2 The proposed MQPLS algorithm

Step	Summary
0	Preprocess \mathbf{X} and \mathbf{Y}
1	Set the output score vector \mathbf{u}_l to a column of \mathbf{Y}
2	Calculate the input weight vector \mathbf{w}_l : $\mathbf{w}_l = \frac{\mathbf{u}_l^T \mathbf{X}}{\mathbf{u}_l^T \mathbf{u}_l}$
3	Normalise \mathbf{w}_l to unit length: $\mathbf{w}_l = \frac{\mathbf{w}_l}{\ \mathbf{w}_l\ }$
4	Calculate the input score vector \mathbf{t}_l : $\mathbf{t}_l = \mathbf{X} \mathbf{w}_l$
5	Fit the quadratic inner relation: $\mathbf{b}_l \leftarrow \text{fit}[\mathbf{u}_l = b_{0,l} \mathbf{1} + b_{1,l} \mathbf{t}_l + b_{2,l} \mathbf{t}_l^2 + \mathbf{e}_l]$
6	Calculate the non-linear prediction of \mathbf{u}_l : $\hat{\mathbf{u}}_l = b_{0,l} \mathbf{1} + b_{1,l} \mathbf{t}_l + b_{2,l} \mathbf{t}_l^2$
7	Solve the optimisation problem (17) to obtain the optimal input weight vector \mathbf{w}_l and the parameter vector \mathbf{b}_l of the quadratic inner mapping according to the procedure of Subsection 3.1
8	Calculate the new input score vector \mathbf{t}_l : $\mathbf{t}_l = \mathbf{X} \mathbf{w}_l$
9	Calculate the input loading vector \mathbf{p}_l : $\mathbf{p}_l = \frac{\mathbf{t}_l^T \mathbf{X}}{\mathbf{t}_l^T \mathbf{t}_l}$
10	Normalise \mathbf{p}_l to unit length: $\mathbf{p}_l = \frac{\mathbf{p}_l}{\ \mathbf{p}_l\ }$
11	Calculate the new non-linear prediction of \mathbf{u}_l : $\hat{\mathbf{u}}_l = b_{0,l} \mathbf{1} + b_{1,l} \mathbf{t}_l + b_{2,l} \mathbf{t}_l^2$
12	Calculate the output loading vector \mathbf{q}_l : $\mathbf{q}_l = \frac{\hat{\mathbf{u}}_l^T \mathbf{Y}}{\hat{\mathbf{u}}_l^T \hat{\mathbf{u}}_l}$
13	Normalise \mathbf{q}_l to unit length: $\mathbf{q}_l = \frac{\mathbf{q}_l}{\ \mathbf{q}_l\ }$
14	Calculate the new output score vector \mathbf{u}_l : $\mathbf{u}_l = \mathbf{Y} \mathbf{q}_l$
15	Calculate the input residual matrix \mathbf{E}_l : $\mathbf{E}_l = \mathbf{X} - \mathbf{t}_l \mathbf{p}_l^T$
16	Calculate the output residual matrix \mathbf{F}_l : $\mathbf{F}_l = \mathbf{Y} - \hat{\mathbf{u}}_l \mathbf{q}_l^T$
17	If $l < L$, set $\mathbf{X} = \mathbf{E}_l$, $\mathbf{Y} = \mathbf{F}_l$ and repeat Steps 1 to 16 with $l = l + 1$; else stop

3.2 The MQPLS

Combining this new updating procedure for the outer input weights and parameters of the quadratic inner mapping with the QPLS algorithm leads to a new NPLS algorithm, which we refer to as the MQPLS algorithm. The detailed steps of this MQPLS algorithm is summarised in Table 2. Compared with the error-based QPLS algorithm of Table 1, computational requirements of the proposed MQPLS algorithm are significantly reduced, because \mathbf{w}_l and \mathbf{b}_l do not need to be calculated iteratively.

3.3 A comparative case study – simulated non-linear function

The simulated non-linear function has the four uncorrelated random input variables x_i , $1 \leq i \leq 4$, which are uniformly distributed in $[-0.25, 0.25]$ and the output is given by

$$y = \exp(2x_1 \sin(\pi x_4)) + \sin(x_2 x_3) \quad (18)$$

This non-linear function was employed by Baffi et al. (1999a) to compare the performance of their error-based QPLS algorithm with that of the original QPLS algorithm. In this paper, we use this simulated non-linear function in a case study to compare the performance of the four PLS algorithms, namely, the linear PLS, the original QPLS, the error-based QPLS and the proposed MQPLS.

Table 3 Cumulative variances captured (%) by the linear PLS, QPLS, error-based QPLS and MQPLS for the simulated non-linear function

Number of latent variables L	Linear PLS		Original QPLS	
	X	Y	X	Y
1	25.25	2.97	25.41	49.91
2	49.56	2.97	50.73	51.06
3	75.43	2.97	75.00	53.10
4	100.0	2.97	100.0	71.57
Number of latent variables L	Error-based QPLS		MQPLS	
	X	Y	X	Y
1	25.24	79.07	25.51	79.08
2	50.02	94.17	50.33	94.21
3	75.45	95.31	75.45	95.40
4	94.42	95.63	100.0	95.73

The total number of samples is 500. Four hundreds of these data samples are used for model training and the remaining 100 are used as a test dataset. No noise is added to the variables in this study. The linear PLS, original QPLS, error-based QPLS and MQPLS algorithms were applied to the training dataset. The cumulative variances of the \mathbf{X} and \mathbf{Y} blocks captured by each model are given in Table 3. Due to the fact that the four inputs are uncorrelated and uniformly distributed between -0.25 and 0.25 , the amount of the variance of \mathbf{X} explained by each input score vector is approximately the same. This is confirmed by the results

shown in Table 3. Figures 1 to 4 show the different abilities of the four models to fit the non-linear inner relation, where the curve in each graph, consisting of symbols +, denotes the inner regression mapping obtained by each algorithm.

Figure 1 Scatter plot of the scores and the inner model obtained by the linear PLS for the simulated non-linear function (see online version for colours)

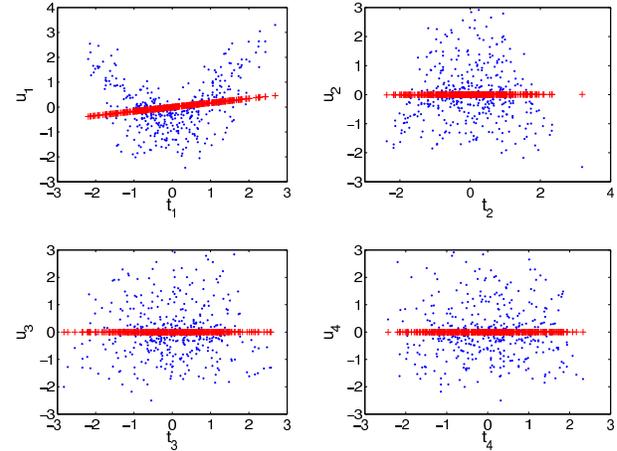
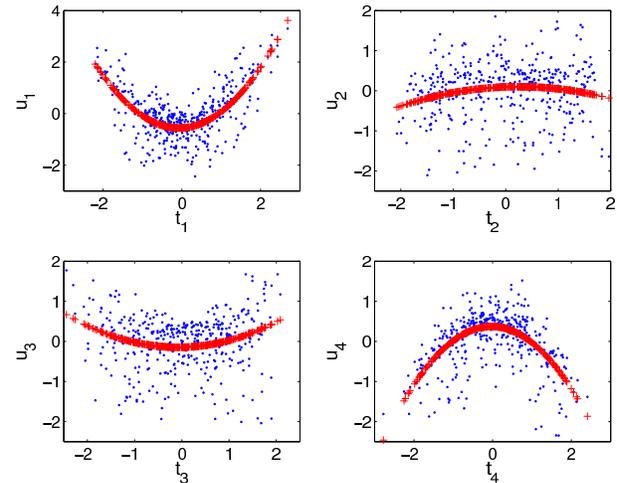


Figure 2 Scatter plot of the scores and the inner model obtained by the original QPLS for the simulated non-linear function (see online version for colours)



It is easily concluded that the linear PLS is incapable of fitting a non-linear inner model, as illustrated clearly in Figure 1. Furthermore, the cumulative variance of \mathbf{Y} captured by the linear PLS model is only 2.97%. It is also evident that improvement can be achieved over the original QPLS by implementing an updating procedure in the QPLS algorithm, as clearly shown in the cases of the error-based QPLS and the proposed MQPLS. The greater the spread in \mathbf{u}_l , the poorer the fit of the inner relation and vice versa. Note that the output score vectors of the error-based QPLS and MQPLS are less spread than those of the QPLS. In addition, from the point of view of the cumulative variance captured, the error-based QPLS is effective in explaining the cumulative variance of \mathbf{Y} block (95.63%) but less so in

explaining the cumulative variance of \mathbf{X} block (94.42%). In comparison, the output cumulative variance explained by the MQPLS is marginally better (95.73%) than that of the error-based QPLS and, moreover, the input cumulative variance explained by the MQPLS is better (100%).

Figure 3 Scatter plot of the scores and the inner model obtained by the error-based QPLS for the simulated non-linear function (see online version for colours)

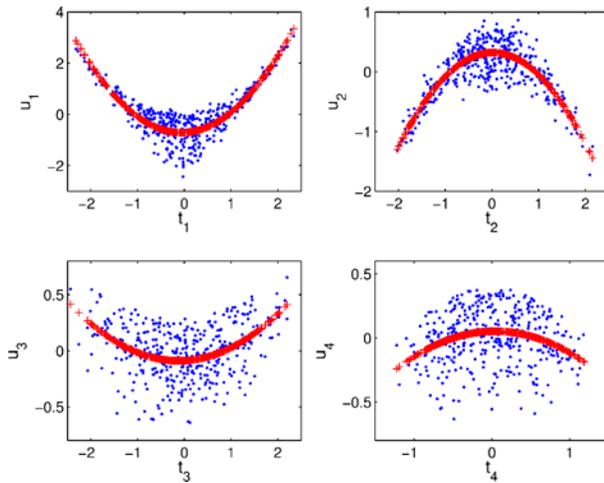
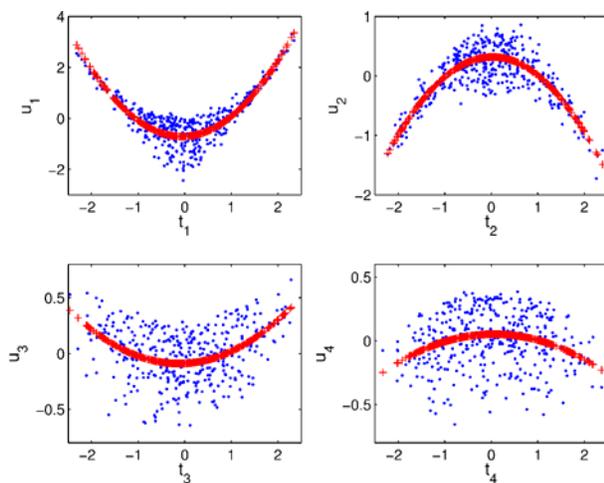


Figure 4 Scatter plot of the scores and the inner model obtained by the MQPLS for the simulated non-linear function (see online version for colours)



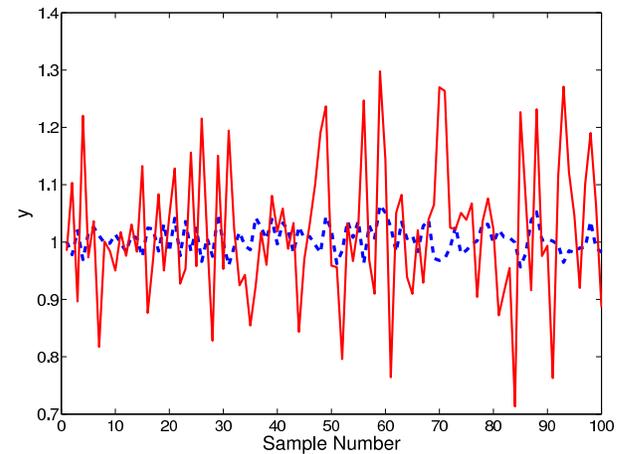
Figures 5 to 8 depict the actual and predicted values for the test dataset based on the four models obtained by the linear PLS, original QPLS, error-based QPLS and MQPLS algorithms, respectively. Table 4 compares the test root mean squared predictive error (RMSPE) performance of the four models and the run times required by the four algorithms under the same computational platform to obtain these four models. It can be seen that the model obtained by the MQPLS algorithm, whose RMSPE is only 0.0282, is the most accurate one among the four models. The time consumed by an algorithm represents the computational complexity of the algorithm. It can be concluded that the

MQPLS only requires about one third computational complexity of the error-based QPLS.

Table 4 Test RMSPE and run time consumed by the linear PLS, original QPLS, error-based QPLS and MQPLS for the simulated non-linear function

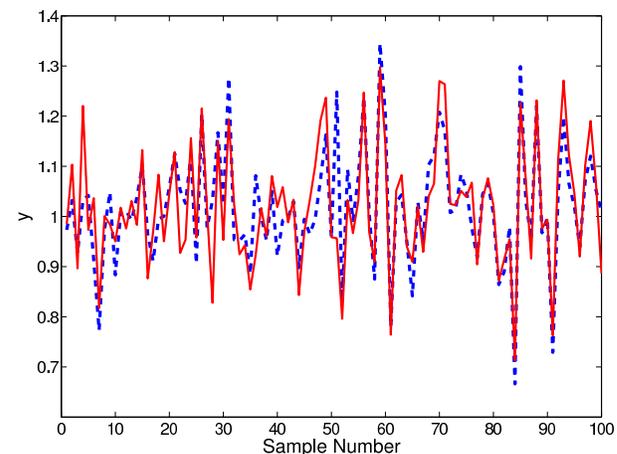
	<i>Linear PLS</i>	<i>QPLS</i>	<i>Error-based QPLS</i>	<i>MQPLS</i>
RMSPE	0.1254	0.0673	0.0389	0.0282
Time(s)	0.4840	0.8130	14.969	4.6720

Figure 5 Test results by the linear PLS for the simulated non-linear function (see online version for colours)

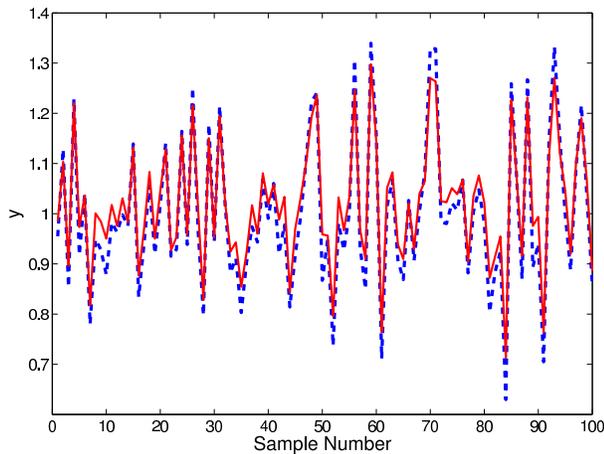


Note: Solid: actual output and dashed: model predicted output.

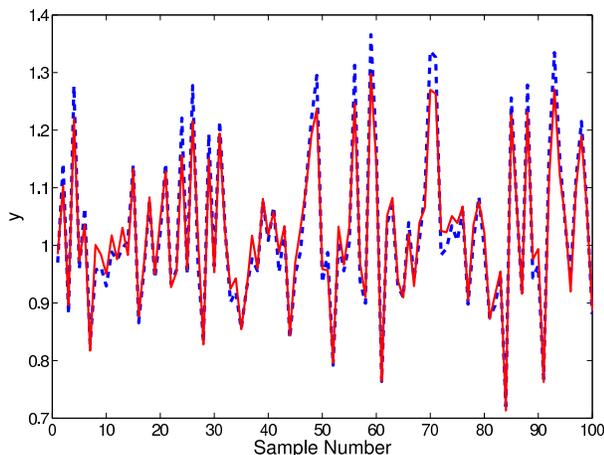
Figure 6 Test results by the original QPLS for the simulated non-linear function (see online version for colours)



Note: Solid: actual output and dashed: model predicted output.

Figure 7 Test results by the error-based QPLS for the simulated non-linear function (see online version for colours)

Note: Solid: actual output and dashed: model predicted output.

Figure 8 Test results by the MQPLS for the simulated non-linear function (see online version for colours)

Note: Solid: actual output and dashed: model predicted output.

4 Inferential control of a FCCU fractionator

An FCCU converts heavy distillates like gas oils or residuals into gasoline, diesel oil and middle distillates using cracking catalyst and is one of the key processes in modern petroleum refining.

4.1 Soft sensor for diesel oil solidifying point

The four algorithms, the linear PLS, original QPLS, error-based QPLS and MQPLS, are applied to establish the soft sensors on an industrial FCCU main fractionator to predict the diesel oil solidifying point (primary variable). Through analysing the underlying chemical process, 15 process variables are chosen as the secondary variables and measured online at one minute intervals, while the actual

product quality is only available from the lab with a frequency of two hours. The chosen secondary variables include the top pressure, the top temperature, the flow rate, the temperature of the second reflux, etc. A dataset including 720 samples are gathered from the plant, which have been chosen to be representative of the range of known operating conditions. The outliers have been removed beforehand. The data is split into a training dataset and a test dataset. Every fifth observation is placed in the test dataset, totalling 144 samples and the remaining 576 observations form the training dataset. The optimal number of components, L , is calculated by cross-validation using the PRESS statistics, as mentioned in Section 2.

Slight non-linearity is found in the first pair of the components of the data gathered, which indicates that a quadratic inner model is appropriate. The training performance of the four algorithms are summarised in Table 5, where the cumulative variances of the \mathbf{X} and \mathbf{Y} matrices captured by each model are given. Figures 9 to 12 compares the actual output, the diesel oil solidifying point ($^{\circ}\text{C}$), of the test dataset with the model predictions obtained by the four algorithms, respectively. Table 6 lists the test mean squared predictive error (MSPE) performance of the four models obtained by the four algorithms. It is clear from Table 6 that the three QPLS-type algorithms better capture the non-linear characteristics in the FCCU data than the linear PLS algorithm. In particular, the performance of the MQPLS algorithm is the most accurate among the four algorithms. The prediction of the MQPLS model is therefore used as a reference for the plant operator.

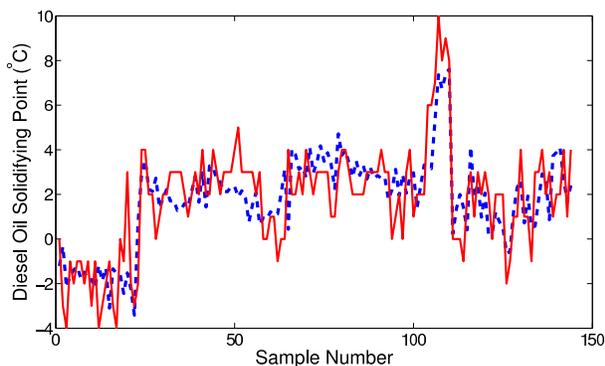
Table 5 Cumulative variances captured (%) by the linear PLS, QPLS, error-based QPLS and MQPLS for the real dataset of an industrial FCCU fractionator

Number of latent variables L	Linear PLS		Original QPLS	
	X	Y	X	Y
1	69.75	56.37	74.40	72.47
5	70.85	62.14	93.61	75.51
10	92.45	65.06	99.40	78.24
15	100.0	68.85	100.0	78.62
Number of latent variables L	Error-based QPLS		MQPLS	
	X	Y	X	Y
1	29.42	78.61	34.56	82.70
5	37.17	80.24	53.56	84.96
10	52.08	91.73	87.26	92.57
15	62.58	92.55	88.83	93.79

Table 6 Test MSPE by the linear PLS, original QPLS, error-based QPLS and MQPLS for the real dataset of an industrial FCCU fractionator

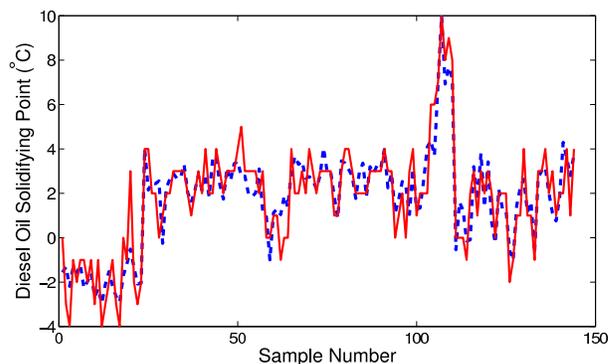
	Linear PLS	QPLS	Error-based QPLS	MQPLS
MSPE	1.4687	1.31197	1.1651	1.0847

Figure 9 Test results by the linear PLS for the real dataset of an industrial FCCU fractionator (see online version for colours)



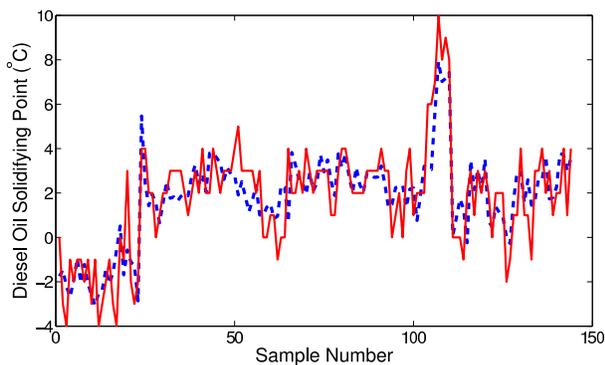
Note: Solid: actual output and dashed: model predicted output.

Figure 12 Test results by the MQPLS for the real dataset of an industrial FCCU fractionator (see online version for colours)



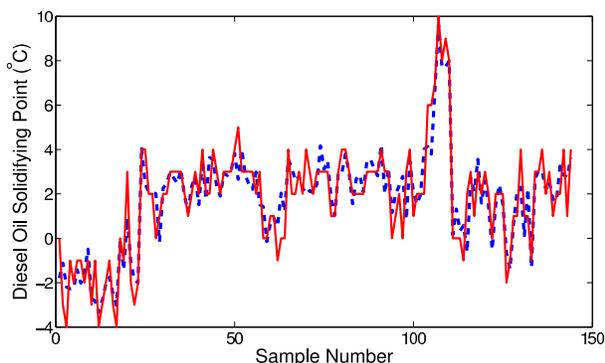
Note: Solid: actual output and dashed: model predicted output.

Figure 10 Test results by the original QPLS for the real dataset of an industrial FCCU fractionator (see online version for colours)



Note: Solid: actual output and dashed: model predicted output

Figure 11 Test results by the error-based QPLS for the real dataset of an industrial FCCU fractionator (see online version for colours)

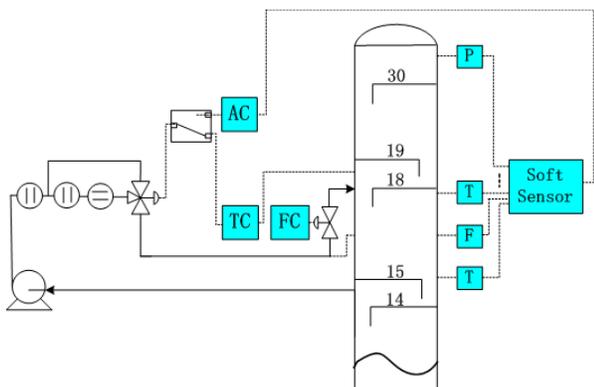


Note: Solid: actual output and dashed: model predicted output.

4.2 Predictive inferential control scheme

The product quality control of fractionators is a classical and difficult problem. Tray temperature control is a traditional control scheme that is widely used in chemical plants for product quality control. It is very simple and does not rely on any online product quality analysers or soft sensors. Because the soft sensor based on the MQPLS algorithm to predict the diesel oil solidifying point is available, more intelligent inferential controller for quality close-loop control can be implemented. Typically, a cascade inferential control system is built in which the set point of the tray temperature controller is given by the output of the quality inferential controller (Kano et al., 2000, 2003). However, such a control scheme has complex structure which may cause some serious difficulties to the plant operator. Furthermore, in such a cascade control system, the inner temperature controller has a greater influence on the performance of the whole system and the benefits of the quality inferential control is not fully realised. In this paper, we propose a new alternative inferential control scheme that switches between the conventional tray temperature controller and the quality inferential controller without causing disturbance to the operation of the plant. The configuration of the proposed inferential control system is shown in Figure 13. The temperature controller (TC in Figure 13) still uses the original tray temperature controller. The inferential controller (AC in Figure 13) adopts the constrained single-value DMC algorithm (Wang and Li, 2003; Zhao et al., 2003; Biswas et al., 2007), which is widely applied in chemical and related industries and has a particular advantage in dealing with long time delays of chemical processes.

Figure 13 Schematic diagram of the FCCU main fractionator with predictive inferential control system (see online version for colours)



The adopted DMC algorithm is an internal model control strategy based on the discrete-time step response model. For any assumed set of the present and future control moves $\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+P)$, the future outputs corrected by the error between the predicted output value $\hat{y}(k)$ and the actual plant output value $y(k)$ are labelled as $y_c(k+1), y_c(k+2), \dots, y_c(k+P)$. If only the P th step error correction is considered, the present control move $\Delta u(k)$ is computed by solving the following constrained optimisation

$$\begin{aligned} \min_{\Delta u(k)} & \left\{ J = (y_r(k+P) - y_c(k+P))^2 \right\} \\ \text{s.t.} & |\Delta u(k)| \leq u_{\max} \\ & u_{\min} \leq u(k) \leq u_{\max} \end{aligned} \quad (19)$$

where $y_r(k+P)$ is the set objective value.

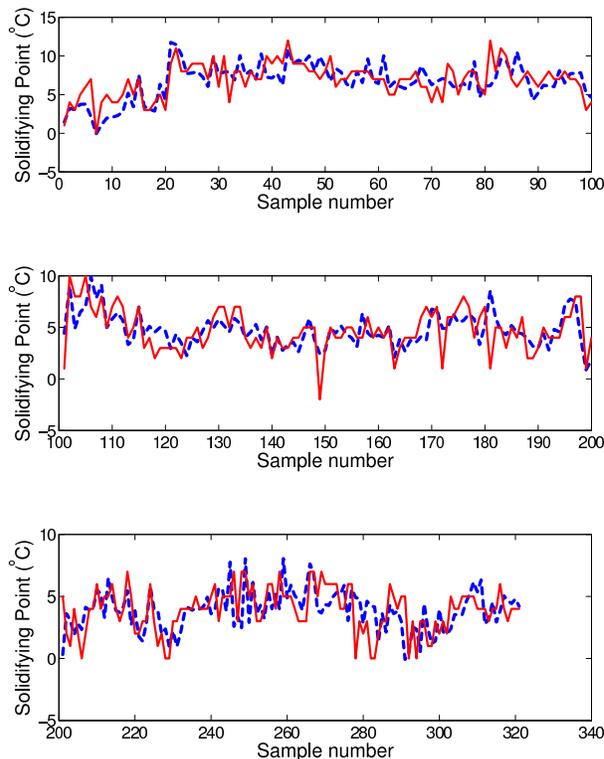
This kind of DMC design scheme can make use of the original system module and is easy to be realised on the distributed control system (DCS) as well as simple to operate by the plant operator. However, there are still some issues that require particular attention in practice. When the step response of the process is made for the DMC, it must be ensured that the step response starts from some steady state. Also, the inferential control system should take into account the abnormalities from the DCS and the process to guarantee safety of the process operation. Because the operating conditions of chemical plants are often changing, predictive model updating is another key issue.

4.3 Control experimental results

The designed MQPLS-based soft sensor and the predictive inferential controller have been realised using the control language programming and are made operational on the DCS of the plant to control the solidifying point of diesel oil. The results of an experiment using the MQPLS-based soft sensor to predict the diesel oil solidifying point ($^{\circ}\text{C}$) are depicted in Figure 14, where the solid-curve data are the actual solidifying point gathered from the laboratory and the dashed-curve data are the predicted value by the

MQPLS-based soft sensor. The MSPE of the MQPLS-based soft sensor in this experiment is 0.9522 and the predicted result is regarded as satisfactory to be used as the set point of the inferential controller.

Figure 14 Experimental results of predicting the diesel oil solidifying point ($^{\circ}\text{C}$) using the MQPLS-based soft sensor (see online version for colours)



Note: Solid: actual solidifying point and dashed: predicted by the soft sensor.

Figure 15 shows the results of control performance before and after switched to the predictive inferential control system. From Figure 15 it can be seen clearly that the control variance decreases significantly after switching from the tray temperature control system to the inferential control system. Closed-loop response of the predictive inferential control system is illustrated in Figure 16, where the set point of solidifying point is changed from -7.5°C to -6°C . It can be seen that the predictive inferential control system can response quickly to step change in the set point. Good closed-loop response is of benefit to the boulder quality control. The experimental results obtained have indicated that the inferential control system achieves a better performance than the traditional tray temperature control system.

Figure 15 Control performance before and after switched to the predictive inferential control system (see online version for colours)

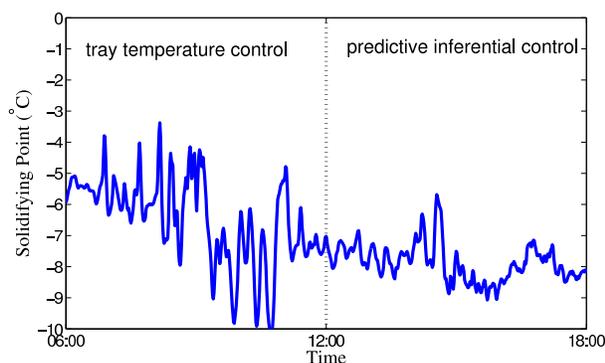
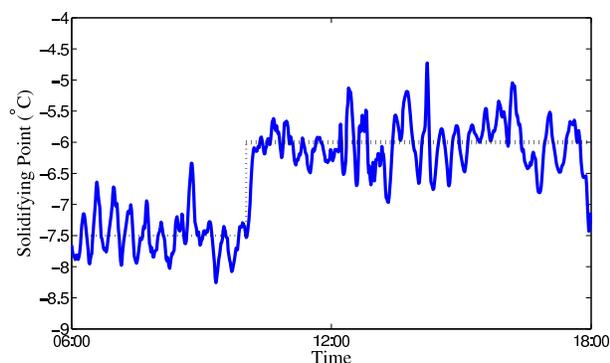


Figure 16 Closed-loop response of the predictive inferential control system (see online version for colours)



5 Conclusions

The contribution of this paper has been two-fold. Firstly, the error-based outer input weight updating procedure within the QPLS framework due to Baffi et al. (1999a, 1999b) has been studied and a new updating procedure based on non-linear programming has been formulated for obtaining the optimal outer input weights and parameters of the quadratic inner mapping, which leads to the proposed MQPLS algorithm. In comparison with the error-based QPLS algorithm, the new MQPLS algorithm has been shown to have advantages of capturing higher percentages of both the input and output cumulative variances, avoiding the expensive pseudo-inverse of matrix and reducing the computational complexity. Secondly, to realise online measurement, a soft sensor has been built based on the MQPLS algorithm to estimate the diesel oil solidifying point of an industrial FCCU main fractionator. This has enabled us to develop a predictive inferential controller based on the constrained single-value DMC algorithm for the quality control of the FCCU fractionator. Our proposed new control scheme can switch between the conventional tray temperature controller and the predictive inferential controller aided by the MQPLS-based soft sensor. Experimental results obtained on an industrial FCCU main fractionator have demonstrated that the proposed predictive

inferential control system performs better than the traditional tray temperature control system.

When the operating conditions of a plant changes significantly, such as in the event of large disturbances entering the process, prompt model updating is essential for a predictive inferential controller to continue functioning well. Further research will investigate effective adaptation of both the inferential MQPLS model and the step response model.

Acknowledgements

This work was supported by China National 863 Program (2007AA04Z193), Shandong Province Nature Science Foundation (No. Y2007G49), and the UK Royal Society and Royal Academy of Engineering.

References

- Andersson, G., Kaufmann, P. and Renberg, L. (1996) 'Non-linear modelling with a coupled neural network – PLS regression system', *J. Chemometrics*, Vol. 10, Nos. 5–6, pp.605–614.
- Baffi, G., Martin, E.B. and Morris, A.J. (1999a) 'Non-linear projection to latent structures revisited: the quadratic PLS algorithm', *Computers and Chemical Engineering*, Vol. 23, No. 3, pp.395–411.
- Baffi, G., Martin, E.B. and Morris, A.J. (1999b) 'Non-linear projection to latent structures revisited (the neural network PLS algorithm)', *Computers and Chemical Engineering*, Vol. 23, No. 9, pp.1293–1307.
- Baffi, G., Martin, E.B. and Morris, A.J. (2000) 'Non-linear dynamic projection to latent structures', *Chemometrics and Intelligent Laboratory Systems*, Vol. 52, No. 1, pp.5–22.
- Bang, Y.H., Yoo, C.K. and Lee, I.B. (2002) 'Nonlinear PLS modeling with fuzzy inference system', *Chemometrics and Intelligent Laboratory Systems*, Vol. 64, No. 2, pp.137–155.
- Berglund, A. and Wold, S. (1997) 'INLR, implicit non-linear latent variable regression', *J. Chemometrics*, Vol. 11, No. 2, pp.141–156.
- Biswas, P.P., Ray, S. and Samanta, A.N. (2007) 'Multi-objective constraint optimizing IOL control of distillation column with nonlinear observer', *J. Process Control*, Vol. 17, No. 1, pp.73–81.
- Geladi, P. and Kowalski, B.R. (1986) 'Partial least-squares regression: a tutorial', *Analytical Chimica Acta*, Vol. 185, pp.1–17.
- Hooke, R. and Jeeves, T.A. (1961) 'Direct search', *Journal of the ACM*, Vol. 8, pp.212–229.
- Kano, M., Miyazaki, K., Hasebe, S. and Hashimoto, I. (2000) 'Inferential control system of distillation compositions using dynamic partial least squares regression', *J. Process Control*, Vol. 10, Nos. 2–3, pp.157–166.
- Kano, M., Showchaiya, N., Hasebe, S. and Hashimoto, I. (2003) 'Inferential control of distillation compositions: selection of model and control configuration', *Control Engineering Practice*, Vol. 11, No. 8, pp.927–933.
- Liu, J., Min, K.G., Han, C.H. and Chang, K.S. (2000) 'Robust nonlinear PLS based on neural networks and application to composition estimator for high-purity distillation columns', *Korean J. Chemical Engineering*, Vol. 17, No. 2, pp.184–192.

- MacGregor, J.F., Skagerberg, B. and Kiparissides, C. (1991) 'Multivariate statistical process control and property inference applied to low density polyethylene reactors', *Proc. IFAC Symposium ADCHEM'91*, October, Toulouse, France, Pergamon Press, Oxford, pp.131–135.
- Myers, R.H. (1990) *Classical and Modern Regression with Applications*, 2nd ed., PWS-KENT, Boston.
- Qin, S.J. and McAvoy, T.J. (1992) 'Nonlinear PLS modeling using neural networks', *Computers and Chemical Engineering*, Vol. 16, No. 4, pp.379–391.
- Rao, S.S. (1996) *Engineering Optimization – Theory and Practice*, 3rd ed., John Wiley & Sons.
- Stone, M. (1974) 'Cross validation choice and assessment of statistical predictions', *J. Royal Statistics Society Series B*, Vol. 36, pp.111–147.
- Wang, D. and Li, C. (2003) 'Self-adaptive dynamic matrix control of high-speed machining servo control', *Int. J. Advanced Manufacturing Technology*, Vol. 21, Nos. 10–11, pp.733–738.
- Wang, H.Z. and Yu, J.S. (2004) 'Application study on non-linear dynamic FIR modeling using hybrid SVM-PLS method', *Proc. 5th World Cong. Intelligent Control and Automation*, June, Hangzhou, China, pp.3479–3482.
- Wilson, D.H.J., Irwin, G.W. and Lightbody, G. (1997) 'Nonlinear PLS modelling using radial basis functions', *Proc. 1997 American Control Conf.*, June, Albuquerque, New Mexico, pp.3275–3276.
- Wold, S. (1992) 'Nonlinear partial least square modeling II. Spline inner relationship', *Chemometrics and Intelligent Laboratory Systems*, Vol. 14, Nos. 1–3, pp.71–84.
- Wold, S., Kettaneh-Wold, N. and Skagerberg, B. (1989) 'Nonlinear PLS modeling', *Chemometrics and Intelligent Laboratory Systems*, Vol. 7, Nos. 1–2, pp.53–65.
- Zhan, Q. and Grassi, V.G. (1999) 'Dynamic matrix control of a speciality chemical plant with different plantwide control structures', *Proc. 1999 American Control Conf.*, June, San Diego, California, pp.632–636.
- Zhao, Z., Xia, X.H., Wang, J.H., Gu, J. and Jin, Y.H. (2003) 'Nonlinear dynamic matrix control based on multiple operating models', *J. Process Control*, Vol. 13, No. 1, pp.41–56.