Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 48-8 (2015) 427–432

Soft Sensor Model Maintenance: A Case Study in Industrial Processes ⋆

Kuilin Chen * **Ivan Castillo** ** **Leo H. Chiang** ** **Jie Yu** *

* *Department of Chemical Engineering, McMaster University 1280 Main Street West, Hamilton, Ontario, L8S 4L8, Canada*
** *Analytical Technology Center, The DOW Chemical Company 2301 Brazosport Blvd., Freeport, TX 77541, USA*

Abstract: One of the challenges of utilizing soft sensors is that their prediction accuracy deteriorates with time due to multiple factors, including changes in operating conditions. Once soft sensors are designed, a mechanism to maintain or update these models is highly desirable in industry. This paper proposes an index that can monitor the prediction performance of soft sensor models and provide guidance about when to update these models. In the proposed approach, a Kalman filter based model mismatch index is developed to monitor the prediction performance of soft sensors with the support of traditional process monitoring indexes, T^2 and SPE. Then, the soft sensor model can be updated through partial least squares (PLS) regression by using samples from the off-line training set and new process conditions. The proposed online update method is applied to an industrial process case study and the effectiveness of the proposed approach is demonstrated by comparing with traditional recursive partial least squares (RPLS).

Keywords: Soft sensors, Inferential sensors, Kalman filter, Model mismatch index, PLS

1. INTRODUCTION

In industry, accurate and reliable measurement and prediction of quality variables play an important role in process control, monitoring, stability and improving product quality (Sharmin et al., 2006; Zhang et al., 2010). Soft sensors are widely used to predict quality variables that are difficult to measure online by using real-time plant data (Bosca and Fissore, 2011; Yu, 2012). An advantage of utilizing soft sensor models is that hardware analyzers can be replaced for these models (Fortuna et al., 2007; Lin et al., 2007).

Soft sensors are traditionally based on first principle models as well as Kalman filter and observers (de Assis and Maciel Fiho, 2000; Heineken et al., 2007; Mangold, 2012). Nevertheless, model based soft sensors require in-depth process knowledge and significant effort for model development. Data-driven techniques can also be utilized to develop soft sensors, which rely mainly on plant data and knowledge of the process (Dufour et al., 2005; Facco et al., 2009). The well-known multivariate statistical methods such as principal component regression (PCR) and partial least squares (PLS) gain some success in building linear inferential models for quality estimates from high-dimensional data with collinearity (Kano and Nakagawa, 2008; Kadlec et al., 2009; Lu et al., 2014). However, the prediction accuracy of soft sensors tends to degrade after a period of their online operation due to process fouling, abrasion of mechanical components, drifted operating conditions, process faults and others (Kadlec et al., 2011). The degradation of soft sensor models may result in a series of issues in process operation with undesirable quality of final products. Therefore, solutions for degradation of soft sensor models are highly desirable in industrial practice. The main contribution of this paper is to provide a mechanism to properly update soft sensor models once they are deployed to online operation.

Various techniques are developed for online adaptation to cope with the issue of degradation in soft sensor models. Block-wise moving window techniques are employed to update the soft sensor model sequentially by retraining the model periodically when a given number of new data samples are collected, such as fast moving window principal component analysis and moving window kernel principal component analysis (Wang et al., 2005; Liu et al., 2009). More recently, a PLS based local learning algorithm is developed to construct an adaptive soft sensor model by using the data in a moving window with different process states (Kadlec and Gabrys, 2011). Nevertheless, the effectiveness of moving window based methods is based on the assumptions that the window size and the intervals between updates are set correctly and the process dynamics do not change within the span of one moving window. If the assumptions do not hold, it is very likely that the soft sensors adapt to noise or have very weak adaptation capabilities. Meanwhile, the recursive partial least squares (RPLS) model is developed by updating the model structure recursively at each sampling instance when the new process and quality measurements are available (Helland et al., 1992; Mu et al., 2006). In addition, recursive methods for PLS are further modified by using the

⋆ Correspondence should be given to Ivan Castillo (Castillo2@dow.com).
The authors would like to acknowledge the support from the DOW chemical company and the natural sciences and engineering research council of Canada (NSERC).

2405-8963 © 2015, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.09.005

new process data in either a sample-wise or a block-wise manner to update the current soft sensor model in the form of covariance matrices (Dayal and MacGregor, 1997; Qin, 1998). Compared with moving window techniques, the computational efficiency of recursive adaptation methods is much higher because only covariance or kernel matrix is updated in model adaptation. Nevertheless, choosing an appropriate forgetting factor for previous models is not a trivial task in recursive adaptation methods. Another alternative of performing online parameter estimation of soft sensor models is by utilizing Kalman filters, in which coefficient estimates are calculated by minimizing the noise effects (Rutan, 1990; Teppola et al., 1999). Some effort has been made to employ just-in-time learning (JITL) strategy to construct a local model based on a number of nearest neighbors of the test sample (query data) for adaptive predictions (Ge and Song, 2010). However, JITL methods cannot model the brand new process dynamics between process and quality variables if corresponding data are not stored in the database. The aforementioned dynamic and recursive modeling approaches are not desirable in some industrial applications because the soft sensor model is updated at each sampling instance or each block of sampling instances without considering the necessity of model update. As long as the soft sensor model provides accurate quality predictions, updating the soft sensor model is an unnecessary effort in industry. In addition, updating the soft sensor model at each sampling instance or each block of sampling instances could lead to unstable model parameters with poor interpretability and generalization capabilities.

Table 1. Soft sensor degradation scenarios

Quality error	Soft sensor degradation	Process faults
Lab analysis issue for quality	Normal operation condition with drifted process correlation	Process under abnormal condition
No effective index	Proposed index	T^2 & SPE
Not considered	Model Update	No update

Three relevant scenarios that could lead to soft sensor degradation are listed in Table 1. In this study, it is assumed that quality measurement is correct because identifying abnormalities in quality measurements is a challenging task (Kaneko and Funatsu, 2013). A Kalman filter based model mismatch index is proposed to monitor the soft sensor degradation and provide guidance about when to update the model. The difference from Kalman filter based model update methods (developed in the 90's) is that Kalman filter is utilized to derive a model mismatch index to monitor the soft sensor model performance. With the assistance of traditional T^2 and SPE process monitoring indexes, the soft sensor model will not be updated under abnormal process conditions. After the model update decision is made based on the model mismatch index, the regression coefficients can be updated through PLS regression using samples from the training set and the current process conditions. Compared with the model parameters from recursive/block-wise updating or Kalman filtering, the updated soft sensor model parameters updated by the proposed method are more stable with better interpretability.

The remainder of the paper is organized as follows. Section 2 gives preliminaries about PLS regression. Then the proposed online update method is developed in Section 3. The effectiveness of the proposed method is demonstrated in Section 4 utilizing an industrial case study. Finally, concluding remarks are drawn in Section 5.

2. PRELIMINARIES

2.1 Partial least squares

In this study, the case of a single quality output is considered (PLS1). Given a regressor matrix $\mathbf{X} \in \mathbb{R}^{N\times D}$ consisting of N samples with D selected process variables per sample, and the response matrix $\mathbf{Y} \in \mathbb{R}^{N\times 1}$ of quality outputs, PLS projects $\mathbf{X}$ and $\mathbf{Y}$ onto a lower dimensional subspace defined by a number of latent variables $[\mathbf{t}_1, \ldots, \mathbf{t}_A]$ as follows:

$$\begin{cases} \mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} \\ \mathbf{Y} = \mathbf{T}\mathbf{Q}^T + \mathbf{F} \end{cases} \tag{1}$$

where $\mathbf{T} \in \mathbb{R}^{N\times A}$ (A is the number of latent variables) is the score matrix representing the projections of the variables on the subspace, $\mathbf{P} \in \mathbb{R}^{D\times A}$ represents the loading matrix for $\mathbf{X}$, and $\mathbf{Q} \in \mathbb{R}^{1\times A}$ defines the loading matrix for $\mathbf{Y}$ (Dayal and Macgregor, 1997; Li et al., 2010). $\mathbf{E}$ and $\mathbf{F}$ denote the modeling residuals. Both $\mathbf{X}$ and $\mathbf{Y}$ matrices are scaled to zero mean and unit variance. The projection matrices in PLS are calculated in an iterative way by solving the following optimization problem:

$$\begin{gathered} \max\ \mathbf{w}_a \mathbf{X}_a^T \mathbf{Y}_a \mathbf{q}_a \\ \text{s.t. } ||\mathbf{w}_a|| = 1, ||\mathbf{q}_a|| = 1 \end{gathered} \tag{2}$$

where $\mathbf{w}_a$ and $\mathbf{q}_a$ are loading vectors for $\mathbf{X}_a$ and $\mathbf{Y}_a$, respectively. Denoting $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_A]$, $\mathbf{T}$ cannot be calculated directly from $\mathbf{X}$ using $\mathbf{W}$ because $\mathbf{X}$ is deflated in each iteration. Instead, $\mathbf{T}$ can be computed from $\mathbf{X}$ directly as follows:

$$\mathbf{T} = \mathbf{X}\mathbf{R} \tag{3}$$

where weighting matrix $\mathbf{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_A]$. Each column of $\mathbf{R}$ can be computed in a recursive manner as follows:

$$\begin{aligned} &\mathbf{r}_1 = \mathbf{w}_1 \\ &\mathbf{r}_a = \mathbf{w}_a - \mathbf{p}_1^T \mathbf{w}_a \mathbf{r}_1 - \ldots - \mathbf{p}_{a-1}^T \mathbf{w}_a \mathbf{r}_{a-1}, \quad (a > 1) \end{aligned} \tag{4}$$

where $\mathbf{p}_a$ is the column vector in $\mathbf{P}$. Based on Eqs. (1) and (3), the PLS regression coefficients β_{PLS} between $\mathbf{X}$ and $\mathbf{Y}$ are given by:

$$\beta_{PLS} = \mathbf{R}\mathbf{Q}^T \tag{5}$$

In addition, the number of latent variables for PLS regression is usually determined by cross-validation in order to achieve the optimal prediction performance.

2.2 Recursive partial least squares

Recursive partial least squares (RPLS) is one of the most commonly used methods for adaptive online process modeling, especially when the process variables are highly correlated (Qin, 1998; Wang et al., 2003). In RPLS, the old data $\mathbf{X}$ and $\mathbf{Y}$ can be discounted by updating the covariance matrices as new data become available

$$(\mathbf{X}^T\mathbf{X})_k = \lambda(\mathbf{X}^T\mathbf{X})_{k-1} + \mathbf{x}_k^T \mathbf{x}_k \tag{6}$$

$$(\mathbf{X}^T\mathbf{Y})_k = \lambda(\mathbf{X}^T\mathbf{Y})_{k-1} + \mathbf{x}_k^T \mathbf{y}_k \tag{7}$$

where $\mathbf{x}_k$ and $\mathbf{y}_k$ are the new process and quality variables observed at sampling instance k, $(\mathbf{X}^T\mathbf{X})_k$ and $(\mathbf{X}^T\mathbf{Y})_k$ are