Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Modeling of Fermentation Processes using Online Kernel Learning Algorithm

Yi Liu*, Diancai Yang, Haiqing Wang*, Ping Li***

**State Key Laboratory of Industrial Control Technology, Institute of Industrial Process Control*
Zhejiang University, Hangzhou, 310027, P. R. China
(Tel: 086-571-8795-1442-810; e-mail: hqwang@ iipc.zju.edu.cn).
***Qingdao Mesnac Co., LTD., Qingdao, 266045, P. R. China (e-mail: yangdc@mesnac.com).*

Abstract: A novel online identification method is developed for nonlinear multi-input multi-output process modeling issue, which is based on kernel learning framework and named as online kernel learning (OKL) algorithm in this paper. This proposed approach can adaptively control its complexity and thus acquire controlled generalization ability. The OKL algorithm performs first a forward increasing for incorporating a "new" online sample and then a backward decreasing for pruning an "old" one, both in a recursive manner. Furthermore, the prior knowledge about process can be easily integrated into the OKL scheme to improve its performance. Numerical simulations on a fed-batch penicillin fermentation process show that the proposed OKL algorithm can learn adaptively the dynamics of the process using relatively small samples, indicating the OKL is an attractive online modeling method for fermentation process.

1. INTRODUCTION

Biochemical processes, such as fermentation processes, encounter many difficulties in modeling and control issues due to the inherent nonlinearity and the exceeding complexity in physiology. To obtain an accurate mathematical model for such a multi-input multi-output (MIMO) bioprocess is a time-consuming and costly task. Furthermore, the lack of suitable online sensors for analyzing the key process variables, such as biomass or production concentrations, limits the effective control and optimization of fermentation processes (Alford, 2006; Thibault *et al*., 1990).

To overcome these obstacles, soft sensors were developed to estimate hard-to-measure process variables from other on-line measurable process variables (Alford, 2006; Tham *et al*., 1991; Thibault *et al*., 1990). Neural networks (NN) have been proved to be able to approximate any nonlinear systems (Narendra and Parthasarathy, 1990) and applied for developing the soft sensors of various biochemical processes (Thibault *et al*., 1990). However, training NN is a time-consuming task and a large number of training examples are necessary. Further, there are still no guarantees of avoidance of the local minima and over-fitting problems. Actually, it is not uncommon only limited measurements from fermentation processes available in practice.

Recently, support vector machine (SVM), which is a novel machine learning method based on statistical learning theory (SLT) and kernel learning (KL) technique, is gaining widespread attention (Schölkopf and Smola, 2002; Suykens *et al*., 2002) and with applications to chemical engineering (Wang *et al*., 2006; Yan *et al*., 2004). Some SVM based soft sensors for fermentation processes have been proposed recently (Desai *et al*., 2006; Li and Yuan, 2006). The results clearly indicate that the SVM is an attractive alternative to NN for the soft sensor applications in bioprocess engineering. However, the soft sensors are built in an offline manner, which implies that a set of samples should be obtained beforehand. Further, offline training algorithm is not suitable for the practical applications such as online system identification and control problems, where the data come in a sequential way and new information of the process is difficult to be directly absorbed into the established model.

In this contribution, an online kernel learning (OKL) with controlled complexity is proposed to learn online the dynamics of the process in a recursive formulation and with application to develop a soft sensor for fermentation processes. After a brief review of SLT and KL, the basic form of OKL is given in section 2. The criterion to add a new sample is presented, and two main stages of OKL are formulated, including the forward incremental learning stage and the backward pruning stage. Application of OKL to a fed-batch penicillin fermentation process is illustrated in section 3 and the conclusions are drawn in the final section.

2. MODEL DEVELOPMENT

The soft sensor model based on KL framework is such a problem where the goal is to learn a mapping f: $X \rightarrow \mathrm{R}$ using a sample sequence $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\} \subset X \times \mathrm{R}$. Generally, it can always assume that the functional $f \in H$ that H is a *Reproducing Kernel Hilbert Space* (RKHS) endowed with a dot product $\langle \cdot, \cdot \rangle_H$. Therefore, as a temporal step of KL based method, one can consider that the input data $\mathbf{x}_i \in X$ is first "mapped" *implicitly* into the *feature space* H by ϕ: $X \rightarrow H$, where ϕ is a nonlinear operator associated with some *positive definite kernel* which satisfies the Mercer theorem, i.e., the so called *kernel trick*: $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_H$ (Schölkopf and

This work was supported by the National Natural Science Foundation of China (No. 20776128) and Alexander von Humboldt Foundation, Germany (Dr. Haiqing Wang). Corresponding author: Haiqing Wang.

Smola, 2002). The functional f is then determined in some optimal sense to yield the soft sensor model wanted.

2.1 Regularized Kernel Learning for Regression

A general form of the "*kernelized*" nonlinear MIMO model proposed by Wang *et al*. (2006) can be formulated as:

$$y_{k,m} = f(\mathbf{w}_{k,m}, \mathbf{x}_k) + e_{k,m} = \mathbf{w}_{k,m}^T \phi(\mathbf{x}_k) + e_{k,m} \quad (1)$$

where $f \in H$ is the wanted model; $y_{k,m}$ denotes the m^{th} output measurement of the wanted model at k instance with m=1, …,M, and M is the number of outputs; and $\mathbf{x}_k$ is a *general input vector* that is usually composed of several measured variables at time k, probably combined with their corresponding delayed forms, and with the delayed outputs. The symbols $\mathbf{w}_{k,m}$ and $e_{k,m}$ denote model parameter vector and process noise of the m^{th} subsystem, respectively.

The problem of inference of a model based on a finite set of observational data is often ill-posed. Typically, a form of *capacity control* is introduced which is often expressed mathematically in the form of *regularization*. Thus, regularized cost functions have been applied in SVM and related methods (Schölkopf and Smola, 2002).

Based on the philosophy of SLT and kernel methods (Schölkopf and Smola, 2002), the following optimization problem, which uses Tihonov regularization, is proposed here to get the solution f in (1):

$$\begin{aligned} &\min\ J(\mathbf{w}_{k,m}) = \frac{1}{2}\|\mathbf{e}_{k,m}\|^2 + \gamma\Omega[\|f\|] \\ &\text{s.t.}\ \ y_{i,m} - \mathbf{w}_{i,m}^T \phi(\mathbf{x}_i) - e_{i,m} = 0 \ \ i = 1, \cdots, k \end{aligned} \quad (2)$$

where $\mathbf{e}_{k,m} = [e_{1,m}, e_{2,m}, \cdots, e_{k,m}]^T$; γ>0 is the regularization parameter to control the smoothness of the solution, and $\Omega(\|f\|)$ is the regularization term (also referred as a penalty term), which is chosen to be convex, such as $\|\mathbf{w}_{k,m}\|^2/2$.

The dual problem is derived for solving the optimization problem above. The Lagrangian for the problem is

$$\begin{aligned} &L(\mathbf{w}_{k,m}, \mathbf{e}_{k,m}, \boldsymbol{\alpha}_{k,m}) = \left(\|\mathbf{e}_{k,m}\|^2 + \gamma\|\mathbf{w}_{k,m}\|^2\right)/2 \\ &-\sum_{i=1}^{k} \alpha_{k,m,i}\left[y_{i,m} - \mathbf{w}_{i,m}^T \phi(\mathbf{x}_i) - e_{i,m}\right] i = 1, \cdots, k \end{aligned} \quad (3)$$

where $\boldsymbol{\alpha}_{k,m} = \left[\alpha_{k,m,1}, \cdots, \alpha_{k,m,k}\right]^T$ are Lagrange multipliers. The conditions for optimality are given by

$$\begin{cases} \dfrac{\partial L}{\partial \mathbf{w}_{k,m}} = 0 \to \mathbf{w}_{k,m} = \dfrac{1}{\gamma}\displaystyle\sum_{i=1}^{k} \alpha_{k,m,i}\phi(\mathbf{x}_i) \\ \dfrac{\partial L}{\partial e_{i,m}} = 0 \to \alpha_{k,m,i} = e_{i,m} \qquad i = 1, \cdots, k \\ \dfrac{\partial L}{\partial \alpha_{k,m,i}} = 0 \to y_{i,m} - \mathbf{w}_{i,m}^T \phi(\mathbf{x}_i) - e_{i,m} = 0 \ \ i = 1, \cdots, k \end{cases} \quad (4)$$

After elimination of the variables $\mathbf{w}_{k,m}$ and $\mathbf{e}_{k,m}$, one gets the following solution

$$\left[\mathbf{K}_k/\gamma + \mathbf{I}_k\right]\boldsymbol{\alpha}_{k,m} = \mathbf{y}_{k,m} \quad (5)$$

where $\mathbf{y}_{k,m} = \left[y_{1,m}, \cdots, y_{k,m}\right]^T$ and $\mathbf{I}_k \in \mathrm{R}^{k \times k}$ is a unit matrix; $\mathbf{K}_k$ is a kernel matrix, and the "*kernel trick*" (Schölkopf and Smola, 2002) applied here is

$$\mathbf{K}_k(i, j) = \left\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \right\rangle, \forall i, j = 1, \cdots, k \quad (6)$$

Then the KL model estimation of the m^{th} subsystem at time k+1 can be obtained

$$f(\mathbf{w}_{k,m}, \mathbf{x}_{k+1}) = \frac{1}{\gamma}\sum_{i=1}^{k} \alpha_{k,m,i}\phi(\mathbf{x}_i)\phi(\mathbf{x}_{k+1}) = \frac{1}{\gamma}\boldsymbol{\alpha}_{k,m}^T \mathbf{k}_{k+1} \quad (7)$$

where $\mathbf{k}_{k+1}(i) = \left\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_{k+1}) \right\rangle, \forall i = 1, \cdots, k$ is a kernel vector.

In summary, the development of a soft sensor model amounts to solving a set of linear equations in the high dimensional feature space introduced by the kernel transform, which is similar to some basic SVM methods, e.g., LS-SVM (Suykens *et al*., 2002). Nevertheless, noting that the KL regression model is not sparse, this adverse factor may cause the solving of (7) troublesome, just as remarked by Wang *et al*. (2006).

The length of parameter vectors $\mathbf{w}_{k,m}$, which can be considered as the order of the KL-based soft sensor model, is equal to the number of the sampling data used. It means that, with the identification continuing online, it may become computationally infeasible during learning and lead to excessive computation times when making prediction for the new sample. Furthermore, the projections of the input samples at different time might be linear dependent in the feature space and thus may cause the solving of $\boldsymbol{\alpha}_{k,m}$ numerical unstable (Wang *et al*., 2006).

Notice that the number of support vectors in SVM learning can be kept as low as possible to avoid the over-fitting problem. And sparseness is generally regarded as good practice in the learning machine. Once the KL regression model has sparse solution, the predictions for the new inputs depend only on the kernel function evaluated at a subset of the training samples.

2.2 Sparseness: Combining the Prior Knowledge

A simple sparsity strategy that can adaptively control the complexity of the KL based soft sensor model is formulated in this section to overcome the embarrassment mentioned above. In this paper, the samples adopted into the KL regression model are referred to as "nodes", just as proposed by Wang *et al*. (2006). The main motivation is to find as few *key nodes* as possible, which can be utilized to establish a suitable soft sensor model with good generalization ability.

There are two main strategies to obtain the sparsity: pre-sparsity and post-sparsity. The former is to control the complexity of the learning machine and suitable for online learning; and the latter is to increase the speed/efficiency of later testing, which is always adopted in batch learning. An interesting pre-sparsity strategy for online KL has been proposed recently by Wang *et al*. (2006), which used a

so-called "*space angle index*" to judge whether the mapped features are approximately linear independent.

A simpler sparsity approach is proposed here. The criterion for adding a new pair of sample [$\mathbf{x}_k$, $\mathbf{y}_k$] to the learning machine is as follows:

$$\left|e_{k,m}\right| = \left|y_{k,m} - f(\mathbf{w}_{k-1,m}, \mathbf{x}_k)\right| > \delta_m, m = 1, \cdots, M \tag{8}$$

where δ_m is a predefined small positive value, named as *Prediction Error Bound* (PEB). The basic idea of this complexity-controlled strategy for the learning machine (that is also the soft sensor model) has the advantage of simplicity and intuition.

If $\left|e_{k,m}\right| > \delta_m, m = 1, \cdots, M$ holds, which means that the approximation error between the actual output and the prediction of the learning machine is significant (for all output measurements of the wanted model at time k), the KL based soft sensor model is not accurate enough and should be improved, so [$\mathbf{x}_k$, $\mathbf{y}_k$] will be introduced as a new node, and in this scenario the soft sensor model will be updated. Otherwise, the soft sensor model of the wanted model at time k is always satisfied, and there is not necessary to add the node in accordance with the well-known "Parsimony Principle". Consequently, the soft sensor model is unaltered. The criterion for adding a new node does not adopt the colinearity concept as used in Wang *et al*. (2006), however, it directly utilizes the prediction error.

Generally speaking, the scales of M outputs of the process are different. Further, for a special subsystem, the output scale at different period of time is probably not the same. Consequently, another advantage of this criterion is that it is easier to combine the prior knowledge of the process into the learning machine. Each output will choose a predefined PEB δ_m according to prior knowledge of the process, that is $\boldsymbol{\delta}_M = [\delta_1, \cdots, \delta_M]^T$. If the prediction of process variable requires more precision, δ_m can be set smaller and vice versa. A smaller δ_m gets more nodes, and a larger one yields a more parsimonious but less precise model. Thus the value of δ_m can be easily selected for the general modeling problems.

This simple sparsity approach makes the complexity of the learning machine restrained; furthermore, from a practical point of view, the computation load is usually very small. Note that this sparsity belongs to pre-sparsity approach (Wang *et al*., 2006) and is different from the basic idea of SVM, where the sparsity is obtained after optimization and the SVM solutions are known as non-maximally sparse (Schölkopf and Smola, 2002).

2.3 Growing: Forward Recursive Learning

The forward learning stage is to grow the nodes with new incoming process information. The initial model only has one node that is the first sample pair. Then assume at time k the OKL based soft sensor model has N_k (at least one) nodes due to the sparsity criterion, and then gets the following equation

$$\left[\mathbf{K}_{N_k} / \gamma + \mathbf{I}_{N_k}\right] \boldsymbol{\alpha}_{N_k,m} = \mathbf{y}_{N_k,m} \tag{9}$$

Note that the above equation has related terms similarly defined in (5), however, the nodes are different. For simplicity, the quantities are defined as $\mathbf{H}_{N_k} = \mathbf{K}_{N_k} / \gamma + \mathbf{I}_{N_k}$ and $\mathbf{P}_{N_k} = \mathbf{H}_{N_k}^{-1}$, then the solution can be expressed as

$$\boldsymbol{\alpha}_{N_k,m} = \mathbf{P}_{N_k} \mathbf{y}_{N_k,m} \tag{10}$$

When a new node is added into the OKL model, (9) becomes

$$\begin{bmatrix} \mathbf{H}_{N_k} & \mathbf{V}_{N_{k+1}} \\ \mathbf{V}_{N_{k+1}}^T & v_{N_{k+1}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_{N_k,m} \\ \alpha_{N_{k+1},m} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{N_k,m} \\ y_{N_{k+1},m} \end{bmatrix} \tag{11}$$

where $\mathbf{V}_{N_{k+1}} = \left[K\left(\mathbf{x}_{N_1}, \mathbf{x}_{N_{k+1}}\right), \cdots, K\left(\mathbf{x}_{N_k}, \mathbf{x}_{N_{k+1}}\right)\right]^T / \gamma$ is the corresponding kernel vector of the new node, and $v_{N_{k+1}} = K\left(\mathbf{x}_{N_{k+1}}, \mathbf{x}_{N_{k+1}}\right) / \gamma + 1$ is a scalar.

Applying the Sherman-Morrison-Woodbury formula (Golub and Van Loan, 1996) to (11) yields the following inverse relation between matrices computation of the inverse $\mathbf{P}_{N_{k+1}}$ and $\mathbf{P}_{N_k}$

$$\mathbf{P}_{N_{k+1}} = \begin{bmatrix} \mathbf{P}_{N_k} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \mathbf{r}_{N_{k+1}} \mathbf{r}_{N_{k+1}}^T z_{N_{k+1}} \tag{12}$$

where $\mathbf{r}_{N_{k+1}} = \left[\mathbf{V}_{N_{k+1}}^T \mathbf{P}_{N_{k+1}}, -1\right]^T$ is a column vector and $z_{N_{k+1}} = 1 / \left(v_{N_{k+1}} - \mathbf{V}_{N_{k+1}}^T \mathbf{P}_{N_{k+1}} \mathbf{V}_{N_{k+1}}\right)$ is a scalar.

The updated algorithm of the forward incremental learning stage is efficient. Whenever a new node is available, the direct computation of the inverse of the matrix $\mathbf{H}_{N_{k+1}}$ requires about $O\left(N_{k+1}^3\right)$ operations, whereas the recursive algorithms only need $O\left(N_{k+1}^2\right)$ operations. The improvement on computing speed is extremely noticeable when the number of nodes N_{k+1} becomes large.

2.4 Pruning: Backward Recursive Learning

Most chemical processes, especially the fermentation processes, are time-variant in nature. The soft sensor model should be able to capture the time-variant characteristic of process by deleting the nodes with old information of the model. Another intention is to keep the model as simple as possible in order to improve the learning speed and save memory space.

The aim of backward decremental learning, also referred as *pruning*, is to recursively delete the old information (Suykens *et al*., 2002; Wang *et al*., 2006). The issue of pruning methods in batch learning of SVM has received a great deal of attention, however, with little research for online learning (Suykens *et al*., 2002). Let the symbol N as the node length. Assume at time k the node growth of the OKL-based soft sensor model is finished and N_k is larger than N. The simplest pruning approach is to delete the first node, for it is

considered as the oldest one and with the least information for learning machine (Wang *et al*., 2006). However, there is no guarantee on the rationality of this intuitional pruning approach. From the optimality conditions in (11) one can infer that the nodes with small Lagrange multipliers also have small error.

Similar with the pruning method proposed by Suykens *et al*. (2002), the nodes with small Lagrange multipliers are deleted. However, Suykens *et al*. (2002) removed the support vectors with small Lagrange multipliers (less than some threshold value) and *retrained* the learning machine, thus it is not suitable for online learning due to the intensive computations.

In our approach, only one node is pruned at a time, furthermore, a recursive update algorithm is adopted to avoid the computation of the matrix inverse $\mathbf{P}_{N_k}$. The pruning procedure includes two steps. Once $N_k > N$ holds, first find out the smallest Lagrange multiplier as follows

$$\arg\min \left| \alpha_{N_k,1,i} \cdots \alpha_{N_k,M,i} \right|, i = 1, \cdots, N_k \tag{13}$$

where $\boldsymbol{\alpha}_{N_k,m,i} = \left[\alpha_{N_k,1,i}, \cdots, \alpha_{N_k,M,i} \right]^T$. Then, when the l^{th} node is pruned from the OKL identification model, the update rule is formulated as

$$\begin{cases} P_{N,i,j} & \leftarrow P_{i,j} - P_{l,l}^{-1} P_{i,l} P_{l,j} \quad \forall i, j = 1, \cdots, l-1 \\ P_{N,i-1,j-1} & \leftarrow P_{i,j} - P_{l,l}^{-1} P_{i,l} P_{l,j} \quad \forall i, j = l+1, \cdots, N_k \end{cases} \tag{14}$$

where $P_{i,j}$ and $P_{N,i,j}$ stands for the item at the i^{th} row and j^{th} column of $\mathbf{P}_{N_k}$ and $\mathbf{P}_N$, respectively. If l equals one, only the second formula of (14) is applied. This updated procedure has been performed for the incremental SVM algorithm (Cauwenberghs and Poggio, 2001). According to (14), $\mathbf{P}_N$ can be efficiently updated from $\mathbf{P}_{N_k}$ without explicitly computing the matrix inverse.

In summary, the modeling algorithmic includes a forward growing stage and a backward deleting stage, both of which perform the recursive algorithms to avoid the direct computation of the inverse of the matrix; consequently, the OKL-based soft sensor model has small computation scale.

3. CASE STUDY: PENICILLIN FERMENTATION PROCESS

3.1 Biomass and Penicillin Concentrations Estimation

The production of secondary metabolites such as antibiotics has been the subject of many studies due to its academic and industrial importance. However, a main obstacle to the implementation of control strategies is the lack of reliable sensors for online measurement of the key variables (Alford, 2006). Thus, the proposed OKL based modeling method is applied here to the soft sensor development of a benchmark process, the fed-batch penicillin fermentation process (PenSim) (Birol *et al*., 2002) to investigate its validity.

Taking both the previous studies (Li and Yuan, 2006; Massimo *et al*., 1992) and PenSim (Birol *et al*., 2002) into consideration, the four process variables can provide the pertinent information: dissolved oxygen concentration (note as C_L), carbon dioxide concentration (CO_2), biomass concentration (X) and penicillin production concentration (P). The fermentation age (note as t) is also of relevance due to the fermentation process is operated as a fed-batch form (Massimo *et al*., 1992). Consequently, the inputs and outputs of the OKL model are below:

$$\begin{cases} \mathbf{x}_k = \left[t, C_{\mathrm{L}k-1}, CO_{2k-1}, X_{k-1}, P_{k-1} \right]^T \\ \mathbf{y}_k = \left[X_k, P_k \right]^T \end{cases} \tag{15}$$

The soft sensor model can be elaborately constructed when the prior knowledge is integrated into the sparsity criterion. The penicillin fermentation process is characterized by nonlinear dynamics and *multistage* characteristics. Typically, penicillin cultivation process has two operational phases. The first phase is conducted as batch operation while the other phase is conducted as fed-batch operation. In general, the system switches to the fed-batch mode after about 45 h, and then a constant glucose feed is used during the fed-batch operation (Birol *et al*., 2002). Another prior knowledge is that the scales of biomass and penicillin production concentrations are different. Note δ_X and δ_P as the PEB of the outputs $\mathbf{y}_k=[X_k, P_k]^T$, respectively. Hence, a simple sparsity criterion is formulated in (16).

$$\left[\delta_X, \delta_P \right]^T = \begin{cases} \left[\delta_{X1}, \delta_{P1} \right]^T & 0 \le t \le 45\ h \\ \left[\delta_{X2}, \delta_{P2} \right]^T & t > 45\ h \end{cases} \tag{16}$$

To investigate the effect of the sparsity criterion on the model precision, different values of δ_X and δ_P are considered in the simulation. Three scenarios, noting as small, normal and large, are expressed as follows

$$\left[\delta_{X1}, \delta_{X2}, \delta_{P1}, \delta_{P2} \right] = \begin{cases} \left[0.02, 0.05, 0, 0.002 \right] & \text{small} \\ \left[0.05, 0.1, 0, 0.005 \right] & \text{normal} \\ \left[0.1, 0.15, 0, 0.05 \right] & \text{large} \end{cases} \tag{17}$$

Notice that δ_{P1} is always set zero because in the batch mode there is no penicillin production. A smaller PEB can achieve a more accurate soft sensor model, however, the value of PEB would not set too small due to the complexity of model.

A total of 10 reference batches are generated under the nominal operations except that the duration time (Birol *et al*., 2002). The duration time is set 300 h but not 400 h. This is in accord with the industrial penicillin fermentation process (Li and Yuan, 2006). Various operating conditions are considered to exhibit batch-to-batch variation in the fermentation process. Further, to investigate the modeling performance of OKL with small samples, the samples are available only every 2 h. Thus the learning machine can only be fed with about 150 sample pairs at each batch.

3.2 Results and Discussion

The simulation environment is Matlab V7.1 with CPU main frequency 2.4GHz and 256M memory. The Gaussian kernel: $K(\mathbf{x}_1, \mathbf{x}_2) = \exp[-\|\mathbf{x}_1 - \mathbf{x}_2\|/\sigma^2]$ is utilized in all simulations below (Schölkopf and Smola, 2002).

After PEB is predefined, only two additional parameters: [γ, σ^2] are to be chosen. The regularization parameter γ=10^{-5} and the kernel parameter σ^2=50 adopted here are chosen by simulation. Note that there is no rigorous parameter selection theory aiming for industrial application available. Fortunately, these two parameters both work well in a wide range. The regularization parameter γ can be first set because it is much smaller than σ. Thus, the parameters can be easily tuned, and the value provided here is just one of many parameter pairs that turn out satisfied results and no optimality is guaranteed.

At first, the first three batches are taken into consideration for simplicity. Thus altogether 450 sample pairs (150*3) are fed into the learning machine. When PEB is set normal, the running time of the whole procedure is only 0.344 s, and altogether 49 nodes are selected out, about 10.9% of the total training data. The prediction of biomass concentration is shown in Fig. 1. Once the prediction error is over the upper bound or under the lower bound, the nodes are increasing by adding the new sample pair. In this case, the node length N=100, thus no pruning strategy is necessary. The prediction of penicillin production concentration is not shown here due to the limitation of the length of the paper. For details about this soft sensor model are tabulated in Tab. 1.

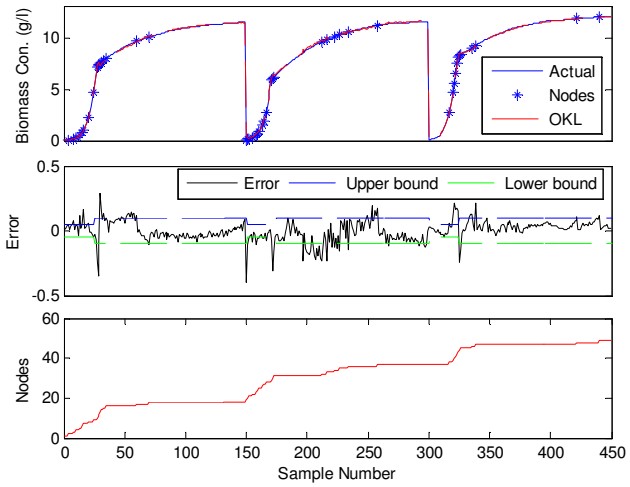

Fig. 1. Online prediction of biomass concentration (1st batch to 3rd batch) with normal PEB

Table 1. Effect of PEB on the performance of OKL modeling

Con.	PEB	RMSE	MAE	Nodes (Sparsity)	Time (s)
X	Large	0.2846	0.9563	20 (4.4%)	0.226
P		0.0199	0.1015		
X	Normal	0.0796	0.3957	49 (10.9%)	0.344
P		0.0040	0.0337		
X	Small	0.0602	0.3662	98 (21.8%)	0.578
P		0.0036	0.0325		

Root Mean Square Error (RMSE) and Maximum Absolute Error (MAE) provided here are two main performance indices of the model precision. As can be seen from Fig. 1 and Tab. 1, the OKL based soft sensor is accurate and has fast learning ability. When PEB is set large, the estimation of biomass concentration is depicted in Fig. 2. Based on Fig. 2 and Tab. 1, we can draw that even a relatively large PEB is adopted, the soft sensor model is accepted to some extent; only 20 nodes (4.4% of the training data) are selected out, which means an extremely sparse model.

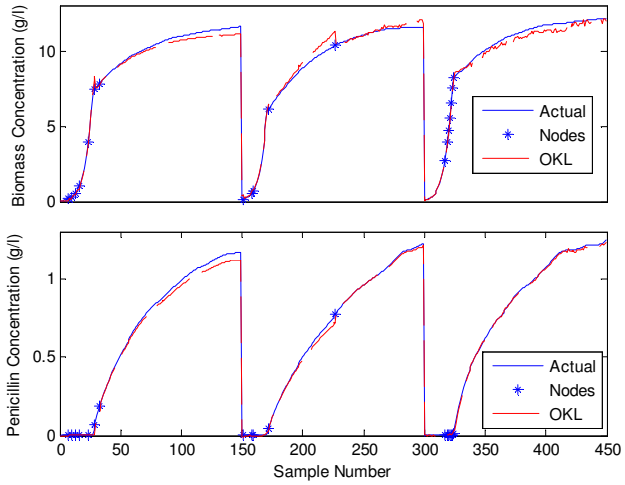

Fig. 2. Biomass and penicillin production concentration prediction (1st batch to 3rd batch) with large PEB

To simulate the industrial environment, the corresponding magnitude (2%) of the super-imposed Gaussian noise is added into the process in addition. The parameters adopted here is γ=10^{-3} and σ^2=50. Compared with the former cases, γ is set larger here due to the noise. PEB should be also set a relatively larger one to get rid of the effect caused by noise, and in this scenario $[\delta_{X1}, \delta_{X2}, \delta_{P1}, \delta_{P2}] = [0.2, 0.5, 0, 0.05]$. The result of biomass estimation is shown in Fig. 3, where only 29 nodes are selected out, about 6.4% of the total samples, clearly demonstrating a very concise model; further, the estimation is precise under this noise environment.

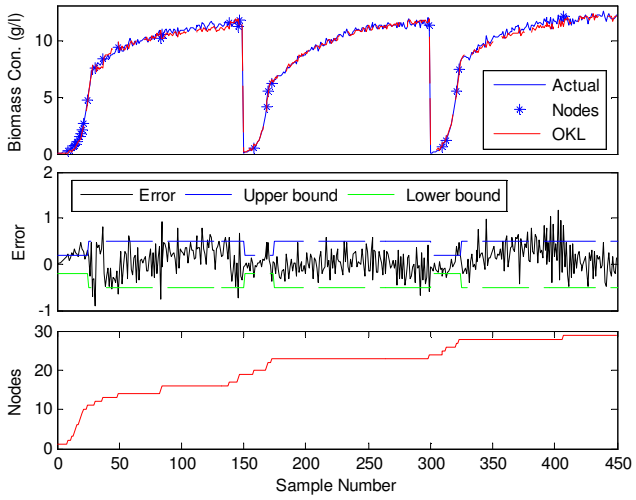

Fig. 3. Biomass prediction in noisy environment

With the batch increasing (e.g., 10 batches), the nodes selected out are larger than the node length N (Suppose

N=100.). Thus the pruning stage is performed in this scenario. The PEB is set normal and the parameters $\gamma=10^{-5}$ and $\sigma^2=50$ are the same with the first case.

Detailed performance indices of different pruning approaches are provided in Tab. 2. Compared with no pruning (Only consider the forward learning.) and another method that deletes the first node, the pruning strategy proposed here has the smallest RMSE. Pruning deletes the old information to capture the time-varying characteristic, thus the modeling performance can be generally improved.

Table 2. Different pruning methods and the corresponding modeling performance

Pruning Method	RMSE		MAE		Nodes
	X	*P*	*X*	*P*	
Proposed	**0.0840**	**0.0042**	0.6222	**0.0360**	100
Delete 1st node	0.0932	0.0044	0.6309	0.0360	100
No pruning	0.0853	0.0043	**0.6205**	0.0360	161

The biomass estimation (10 batches) is shown in Fig. 4. The nodes only marked with asterisk are the one pruned, noting as forward nodes; and the nodes marked with asterisk and square are the actual nodes after modeling of 10 batches. From Fig. 4, we conclude that the OKL based soft sensor model can capture the nonlinear and time-varying characteristics despite of the batch-to-batch variation in operating conditions of the fermentation process.

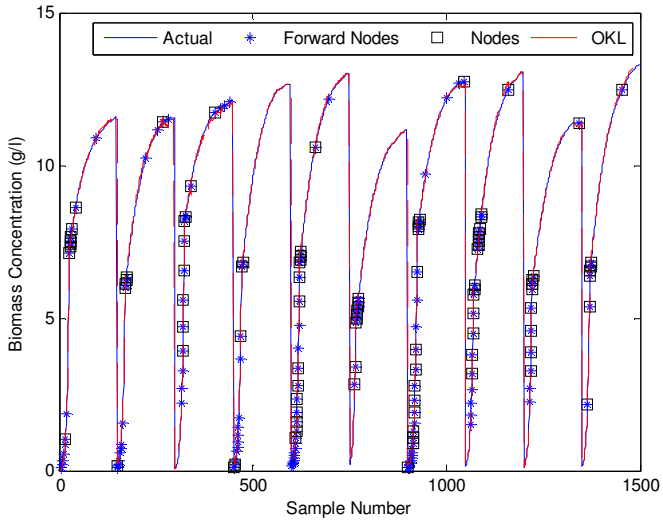

Fig. 4. Biomass prediction (10 batches) with pruning

4. CONCLUSIONS

A novel online modeling method for nonlinear MIMO processes is proposed and applied to soft sensor development for fermentation processes. The OKL based soft sensor model can adaptively control its complexity, which means that the learning machine will be sparse. Furthermore, it can learn forward when a new node is introduced and prune an old one, both with recursively updating forms. The ability of OKL to learn process characteristics with small sample set is a desirable trait that eases its implementation and heightens its modelling potential. Further, only few parameters of OKL are to be determined. The OKL based soft sensor model can perform well when presented with new "unseen" samples. The ability of integrating with prior knowledge of the process makes further OKL appealing to modeling the fermentation processes.

REFERENCES

Alford J.S. (2006). Bioprocess control: Advances and challenges. *Comput. Chem. Eng.*, **30**(10): 1464-1475.

Birol G., Undey C., Cinar A. (2002). A modular simulation package for fed-batch fermentation: penicillin production. *Comput. Chem. Eng.*, **26**(11): 1553-1565.

Cauwenberghs G. and Poggio T. (2001). Incremental and decremental support vector machine learning. *Advances Neural Information Processing Systems (NIPS 2000)*. Cambridge, MA: MIT Press.

Desai K., Badhe Y., Tambe S.S. et al. (2006). Soft-sensor development for fed-batch bioreactors using support vector regression. *Biochem. Eng. J.*, **27**(33): 225-239.

Golub G.H. and Van Loan C.F. (1996). *Matrix Computations*. (3rd ed.), Baltimore: The John Hopkins Univ. Press.

Li Y.F. and Yuan J.Q. (2006). Prediction of Key State Variables using Support Vector Machines in Bioprocesses. *Chem. Eng. Technol.*, **29**(3): 313-319.

Massimo C.D., Montague G.A., Willis M.J. et al. (1992). Towards improved penicillin fermentation via artificial neural networks. *Comput. Chem. Eng.*, **16**(4): 283-291.

Narendra K.S. and Parthasarathy K. (1990). Identification and Control of Dynamical Systems Using Neural Networks. *IEEE Trans. Neural Netw.*, **1**(1): 4-27.

Schölkopf B. and Smola A.J. (2002). *Learning with Kernels*. Cambridge, MA: MIT Press.

Suykens J.A.K., Van Gestel T., De Brabanter J. et al. (2002). *Least Squares Support Vector Machines*. Singapore: World Scientific.

Tham M.T., Morris A.J., Montague G.A. (1991). Soft-sensors for process estimation and inferential control. *J. Process Control*, **1**(1): 3-14.

Thibault J., Breusegem V.V., Cheruy A. (1990). On-line prediction of fermentation variables using neural networks. *Biotechnol. Bioeng.*, **36**(12): 1041-1048.

Wang H.Q., Li P., Gao F.R. et al. (2006). Kernel Classifier with Adaptive Structure and Fixed Memory for Process Diagnosis. *AIChE J.*, **52**(10): 3515-3531.

Yan W.W., Shao H.H., Wang X.F. (2004). Soft sensing modeling based on support vector machine and Bayesian model selection. *Comput. Chem. Eng.*, **28**(8): 1489-1498.