



Process Control

Adaptive Nonlinear Model Predictive Control Using an On-line Support Vector Regression Updating Strategy[☆]Ping Wang^{1,2}, Chaohe Yang¹, Xuemin Tian^{2,*}, Dexian Huang³¹ State Key Laboratory of Heavy Oil Processing, China University of Petroleum, Qingdao 266580, China² College of Information and Control Engineering, China University of Petroleum, Qingdao 266580, China³ Department of Automation, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:

Received 24 June 2013

Received in revised form 14 October 2013

Accepted 16 November 2013

Available online 20 June 2014

Keywords:

Adaptive control

Support vector regression

Updating strategy

Model predictive control

ABSTRACT

The performance of data-driven models relies heavily on the amount and quality of training samples, so it might deteriorate significantly in the regions where samples are scarce. The objective of this paper is to develop an on-line SVR model updating strategy to track the change in the process characteristics efficiently with affordable computational burden. This is achieved by adding a new sample that violates the Karush–Kuhn–Tucker conditions of the existing SVR model and by deleting the old sample that has the maximum distance with respect to the newly added sample in feature space. The benefits offered by such an updating strategy are exploited to develop an adaptive model-based control scheme, where model updating and control task perform alternately. The effectiveness of the adaptive controller is demonstrated by simulation study on a continuous stirred tank reactor. The results reveal that the adaptive MPC scheme outperforms its non-adaptive counterpart for large-magnitude set point changes and variations in process parameters.

© 2014 Chemical Industry and Engineering Society of China, and Chemical Industry Press. All rights reserved.

1. Introduction

Model predictive control (MPC) has been widely accepted as an advanced control strategy in process industry owing to its ability to handle complex control problems with constraints [1,2]. MPC uses a process model to predict future process behavior and future control actions are computed by minimizing a pre-specified cost function, so the effectiveness of MPC relies heavily on the availability of a reasonably accurate process model. Until recently, industrial applications of MPC have mainly based on linear models due to their inherent simplicity from conceptual and implementation points of view [1]. However, as many chemical plants exhibit highly nonlinear behavior when operated over a wide range, linear MPC often results in poor control performance, which motivates its extension to nonlinear MPC, with a more accurate nonlinear model used for prediction and optimization [3].

Among various nonlinear modeling methods, the support vector regression (SVR) method has been widely applied in data-driven modeling [4], since it not only shares many of its features with neural networks but also possesses some additional desirable characteristics.

The advantage of SVR is that [4], for a given modeling problem with a finite set of samples, it can automatically derive the optimal network structure with respect to generalization error. Furthermore, current experience shows that SVRs work as well as, and in some cases, better than classical statistical approaches on noisy or imprecise data. Because of these advantages, SVR has been found increasing applications in chemical processes [5], especially when training data are insufficient or the process has strong nonlinearity [6–8]. Nevertheless, a basic limitation of all data-driven models is their inability to extrapolate accurately once the information is outside the range of data used to generate the model [9]. As the training samples available only describes a period of process historical behavior and might not represent complete characteristics of true dynamics, the performance of the model may deteriorate substantially with time as a consequence of changes in the dynamics of process [10–14]. Although the model can be retrained from scratch when the training set is modified, it is cumbersome and computationally inefficient. In this case, the accurate online SVR (AOSVR) technique [15] seems to be a better alternative to SVR because it uses an incremental algorithm, which updates SVR model efficiently and accurately when a new sample is added to the training set without retraining from scratch. Several model-based control schemes based on AOSVR have been developed in recent years. Iplikci has proposed an adaptive generalized predictive control (GPC) method by combining SVR-based GPC approach with AOSVR [16]. An adaptive inverse control algorithm, where SVR model is used to construct the inverse model of

[☆] Supported by the National Basic Research Program of China (2012CB720500), Postdoctoral Science Foundation of China (2013M541964) and Fundamental Research Funds for the Central Universities (13CX05021A).

* Corresponding author.

E-mail address: tianxm@upc.edu.cn (X. Tian).

the process to be controlled online, has been studied [17]. More recently, AOSVR is employed to capture the abrupt and incremental faults in the framework of SVR model based fault tolerance predictive control scheme [18]. However, in these applications the AOSVR technique is used without taking into account the differences among the newly incoming samples and lacking mechanisms to prune redundant samples efficiently [10]. This may result in unreliable predictions and heavy computation burden as the number of training data increases, limiting the applications of recursive SVR algorithm to a long-term online modeling and control task.

In our recent work, an efficient on-line model updating strategy based on AOSVR has been developed and shown good performance for predicting the melt index of an industrial polypropylene process [19]. This strategy allows us to improve the estimation performance with affordable computation burden by updating the existing SVR model based on the novelty of new samples that arrive sequentially. In this paper, the benefits offered by such an updating strategy are further exploited to design an adaptive MPC controller. Several enhancements are also developed to make the model updating strategy capture the current behavior of process more effectively. Specifically, only those samples that violate the Karush–Kuhn–Tucker (KKT) conditions of existing SVR model are added to improve the estimation accuracy for new operating region where samples may not be scarce or nonexistent. As far as the accumulation of obsolete data is concerned, the old training sample with the maximum distance to the newly added sample in feature space will be recognized as the redundant sample, which will be removed from training database subsequently to enhance the estimation accuracy for the current behavior of process. The effectiveness of the adaptive MPC controller is illustrated by a simulation study on a benchmark continuous stirred tank reactor (CSTR) [20]. The adaptive MPC controller is able to achieve a smooth transition for large magnitude set point changes and maintain the process at an unstable operating point in the presence of unmeasured disturbances and random noise.

2. On-line Support Vector Regression

Given a training set $T = \{(\mathbf{x}_i, y_i), i = 1, 2, \dots, N\}$, where N is the total number of training samples, $\mathbf{x}_i \in R^n$ is the n -dimension input vector, and $y_i \in R$ is the output variable. In the feature space F , SVR builds a linear regression function in the following form [4]

$$\hat{y}(k) = \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b \quad (1)$$

where \mathbf{w} is a vector in F , $\Phi(\cdot)$ is a mapping from the input space to the feature space, b is the bias term, and $\langle \cdot, \cdot \rangle$ stands for the inner product operation in F . The SVR algorithms regard the regression problem as an optimization problem in dual space with the model given by

$$\hat{y}(k) = \sum_{j=1}^N a_j K_{ij} + b \quad (2)$$

where a_j is the coefficient of each sample and K_{ij} denotes the kernel function. Sample \mathbf{x}_j corresponding to a non-zero a_j value is referred to as the support vector (SV). By using Vapnik's ε -insensitive loss function, the dual form of the optimization problem becomes a quadratic programming (QP) problem

$$\min_{a, a^*} D_\varepsilon = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N K_{ij} (a_i - a_i^*) (a_j - a_j^*) + \varepsilon \sum_{i=1}^N (a_i + a_i^*) - \sum_{i=1}^N y_i (a_i - a_i^*) \quad (3)$$

subject to constraints

$$0 \leq a_i, a_i^* \leq C, \sum_{i=1}^N (a_i - a_i^*) = 0, i = 1, \dots, N \quad (4)$$

where ε is the maximum value of tolerable error and C is a regularization parameter that represents a trade-off between model complexity and effect of tolerance to the error larger than ε . The Lagrange formulation of the QP problem can be further represented as

$$L_D = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N K_{ij} (a_i - a_i^*) (a_j - a_j^*) + \varepsilon \sum_{i=1}^N (a_i + a_i^*) - \sum_{i=1}^N y_i (a_i - a_i^*) - \sum_{i=1}^N (\delta_i a_i + \delta_i^* a_i^*) + \varsigma \sum_{i=1}^N (a_i - a_i^*) + \sum_{i=1}^N [\xi_i (a_i - C) + \xi_i^* (a_i^* - C)] \quad (5)$$

where $\delta_i, \delta_i^*, \varsigma, \xi_i, \xi_i^*$ are the Lagrange multipliers. The KKT conditions of Eq. (5) are

$$\begin{aligned} \frac{\partial L_D}{\partial a_i} &= \sum_{j=1}^N K_{ij} (a_j - a_j^*) + \varepsilon - y_i - \delta_i + \varsigma + \xi_i = 0 \\ \frac{\partial L_D}{\partial a_i^*} &= -\sum_{j=1}^N K_{ij} (a_j - a_j^*) + \varepsilon + y_i - \delta_i^* - \varsigma + \xi_i^* = 0 \\ \delta_i, \delta_i^* &\geq 0, \delta_i a_i = 0, \delta_i^* a_i^* = 0 \\ \xi_i, \xi_i^* &\geq 0, \xi_i (a_i - C) = 0, \xi_i^* (a_i^* - C) = 0. \end{aligned} \quad (6)$$

According to Eq. (6), at most one of a_i and a_i^* will be nonzero and both are nonnegative. We define coefficient θ_i and margin function $h(\mathbf{x}_i)$ as

$$\theta_i = a_i - a_i^* \quad (7)$$

$$h(\mathbf{x}_i) \equiv f(\mathbf{x}_i) - y_i = \sum_{j=1}^N K_{ij} \theta_j + b - y_i. \quad (8)$$

Combining Eqs. (6)–(8), the KKT conditions can be rewritten as

$$\begin{cases} h(\mathbf{x}_i) \geq \varepsilon, & \theta_i = -C \\ h(\mathbf{x}_i) = \varepsilon, & -C < \theta_i < 0 \\ -\varepsilon \leq h(\mathbf{x}_i) \leq \varepsilon, & \theta_i = 0 \\ h(\mathbf{x}_i) = -\varepsilon, & 0 < \theta_i < C \\ h(\mathbf{x}_i) \leq -\varepsilon, & \theta_i = C. \end{cases} \quad (9)$$

Based on Eq. (9), the training samples in T can be divided into three subsets as follows [15]. Support set:

$$S = \left\{ i | (\theta_i \in (-C, 0) \wedge h(\mathbf{x}_i) = \varepsilon) \vee (\theta_i \in (0, C) \wedge h(\mathbf{x}_i) = -\varepsilon) \right\}.$$

Error set:

$$E = \{ i | (\theta_i = -C \wedge h(\mathbf{x}_i) \geq \varepsilon) \vee (\theta_i = C \wedge h(\mathbf{x}_i) \leq -\varepsilon) \}.$$

Remaining set:

$$R = \{ i | \theta_i = 0 \wedge |h(\mathbf{x}_i)| \leq \varepsilon \}.$$

The AOSVR algorithm consists of incremental algorithm and decremental algorithm [15]. The basic idea of incremental algorithm is to find a way to add a new sample to one of the three sets maintaining KKT conditions consistent. When a new sample \mathbf{x}_c is received, its corresponding θ_c value is initially set to zero and then gradually changes under the KKT conditions. The relationship between $h(\mathbf{x}_i)$, $\Delta\theta_i$ and Δb is given by

$$\Delta h(\mathbf{x}_i) = \sum_{j=1}^N K_{ij} \Delta\theta_j + K_{ic} \Delta\theta_c + \Delta b. \quad (10)$$

From Eq. (6), the relation between θ_c and θ_i is

$$\theta_c + \sum_{i=1}^N \theta_i = 0. \quad (11)$$

Among the three sets, only the support set samples can change θ_i , then the relation among $\Delta\theta_c$, $\Delta\theta_j$ and Δb can be formulated as [15]

$$\sum_{j \in S} K_{ij} \Delta\theta_j + \Delta b = -K_{ic} \Delta\theta_c, \text{ where } i \in S \quad (12)$$

$$\sum_{j \in S} \Delta\theta_j = -\Delta\theta_c. \quad (13)$$

Defining the support set as a list of elements $S = \{s_1, s_2, \dots, s_{|S|}\}$, Eqs. (12) and (13) can be rewritten in an equivalent matrix form

$$\begin{bmatrix} \Delta b \\ \Delta\theta_{s_1} \\ \vdots \\ \Delta\theta_{s_{|S|}} \end{bmatrix} = \beta \Delta\theta_c, \beta = \begin{bmatrix} \beta_b \\ \beta_{s_1} \\ \vdots \\ \beta_{s_{|S|}} \end{bmatrix} = -R \begin{bmatrix} 1 \\ Q_{s_1 c} \\ \vdots \\ Q_{s_{|S|} c} \end{bmatrix} \quad (14)$$

$$\text{with } R = \begin{bmatrix} 0 & 1 & \dots & 1 \\ 1 & Q_{s_1 s_1} & \dots & Q_{s_1 s_{|S|}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & Q_{s_{|S|} s_1} & \dots & Q_{s_{|S|} s_{|S|}} \end{bmatrix}^{-1}.$$

Updating θ_i values, the support set samples will be consistent. The $h(\mathbf{x}_i)$ values for support samples do not need update, because they are always 1. For error and remaining samples, the situation is opposite. They do not change θ_i value, but change $h(\mathbf{x}_i)$. Defining $N = E \cup R = \{n_1, n_2, \dots, n_{|N|}\}$ and rewriting the variation of $h(\mathbf{x}_i)$ in a matrix notation

$$\begin{bmatrix} \Delta h(\mathbf{x}_{n_1}) \\ \vdots \\ \Delta h(\mathbf{x}_{n_{|N|}}) \end{bmatrix} = \gamma \Delta\theta_c, \gamma = \begin{bmatrix} Q_{n_1 c} \\ \vdots \\ Q_{n_{|N|} c} \end{bmatrix} + \begin{bmatrix} 1 & Q_{n_1 s_1} & \dots & Q_{n_1 s_{|S|}} \\ 1 & Q_{n_2 s_1} & \dots & Q_{n_2 s_{|S|}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & Q_{n_{|N|} s_1} & \dots & Q_{n_{|N|} s_{|S|}} \end{bmatrix} \beta. \quad (15)$$

$\Delta\theta_i$, Δb and $h(\mathbf{x}_i)$ can be updated by computing β and γ . Eqs. (14) and (15) hold only when the samples in the support set do not change membership. Therefore, $\Delta\theta_c$ is chosen to be the largest value that can either maintain the support set unchanged or terminate the incremental algorithm. In this process, some samples may transfer from a set to another, and some may not. The process is repeated until the KKT conditions are satisfied for all data points including the most recent one. Instead of solving a complex QP problem each time, the structure of SVR can adjust dynamically to meet the KKT conditions, making the AOSVR algorithm efficient for online model updating [15–18].

3. Model Updating Strategy

3.1. Adding new samples based on KKT conditions

From a practical point of view, the idea of updating the existing model whenever a new sample is available is inappropriate, because frequently updating a model will result in high computation requirements and stability problems [11]. More importantly, the generalization ability of the existing model may deteriorate significantly [10,19]. An important prerequisite for successfully updating the existing model is that the newly added training samples contain enough new information. In this paper, the following proposition is used to distinguish the novelty of the training sample [19,21].

Proposition 1. Let $f(\mathbf{x})$ be the regression function of SV and $\{\mathbf{x}_c, \mathbf{y}_c\}$ be a new sample. If $\{\mathbf{x}_c, \mathbf{y}_c\}$ satisfies the KKT conditions, the SV set will not

change, otherwise it will change. The samples that violate KKT conditions are those satisfying the following condition

$$|h(\mathbf{x}_c)| = |f(\mathbf{x}_c) - \mathbf{y}_c| > \varepsilon. \quad (16)$$

Proposition 1 reveals the influence of a new sample on the SV set of the existing SVR model [21]. If the new sample violates the KKT conditions, the SV set of SVR model will change after the incremental learning of the new sample. Since the training samples can be completely represented by the SV set, the KKT conditions seem to be more reasonable than regression functions in classifying the new samples and the SVR model should be updated according to the novelty of training data. When a new sample is added to the training set, the geometrical interpretation of the shift in the regression curve is illustrated in Fig. 1. After adding new sample $\{\mathbf{x}_c, \mathbf{y}_c\}$, the updated regression curve, denoted by $\tilde{f}(x)$, is dragged towards the newly added sample in the high dimension feature space. In this way, the updated SVR model is able to absorb the information contained in the newly added sample. By introducing new samples selectively, the frequent update of the existing SVR model is avoided. Accordingly, the generalization ability of the model is well maintained, while the computation burden is reduced.

In applications, the outputs of process are always corrupted by random noise, leading to frequent model updating. To cope with this problem, a simple linear filter is introduced to smooth the training samples as it can be easily implemented online. The process output is smoothed by a weighted sum of previous measurements in a window of finite length

$$\bar{y}(k) = \frac{1}{N_0} \sum_{i=1}^{N_0} a_i y(k-i+1) \text{ with } \sum_{i=1}^{N_0} a_i = 1 \quad (17)$$

where N_0 is the filter length, which can be adjusted according to the level of process noise, a_i are filter coefficients with values between 0 and 1, and $\bar{y}(k)$ is the de-noised output.

3.2. Pruning old samples based on distance

The dimension of matrixes in Eqs. (14) and (15) increases with the accumulation of training samples, leading to excessive computation for model updating, especially when the model is used for control purpose [11,22]. In addition, the updated model may fail to track changes in the process timely as the training set contains too many out-of-data samples. Considering all these factors, it is necessary to delete those samples as long as they become redundant. One of the simplest pruning strategies is to remove the oldest sample from the memory whenever a new one is added, assuming that the oldest sample is the one with the least contribution to the construction of existing model. However, in

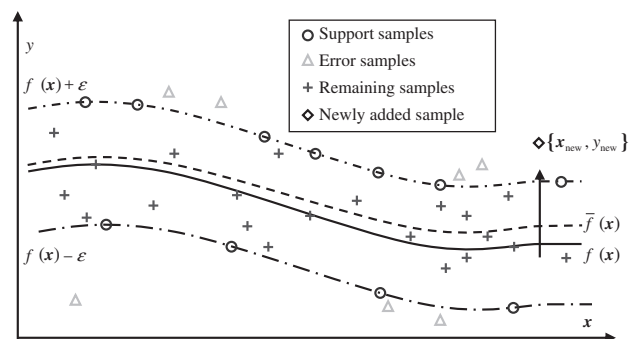


Fig. 1. Geometrical interpretation of the shift in regression curve with a new sample added.

many applications there is no guarantee that the oldest sample no longer represents the current behavior of process. Another common approach is to delete those samples that are not belonging to SVs, because SVR can be completely represented by the SVs. However, as mentioned earlier, samples may transfer from a non-SV to a SV during the implementation of AOSVR. Other pruning strategies, such as Bayesian and leave-one-out cross-validation, are mainly used in an offline manner because of their complexity [10].

Discarding a training sample from the model might inevitably degrade its performance. Pruning rules aim at finding samples whose removal is expected to have negligible effect on the quality of the model. For a new sample $\{\mathbf{x}_c, \mathbf{y}_c\}$ added to the training set and the correspondingly updated SVR model $\bar{f}(\mathbf{x})$, the condition to confirm if some old training samples need to be deleted is

$$e_c = |y_c - \bar{f}(\mathbf{x}_c)| > e_1 \quad (18)$$

where e_c is the approximation error of the updated SVR model with respect to the newly added sample and e_1 is a given threshold associated with model accuracy. If condition (18) is satisfied, the newly added sample is not yet adequately represented by the updated SVR model, and some old training samples should be deleted to enhance the estimation accuracy of the SVR model for the current behavior of process. Instead of pruning the oldest training sample, sample will be deleted according to its distance to the newly added sample in feature space. In feature space, the distance between sample \mathbf{x}_i and \mathbf{x}_c is defined as

$$\begin{aligned} d(\mathbf{x}_i, \mathbf{x}_c) &= \sqrt{\|\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_c)\|^2} = \sqrt{\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_i) + \Phi(\mathbf{x}_c) \cdot \Phi(\mathbf{x}_c) - 2\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_c)} \\ &= \sqrt{K(\mathbf{x}_i, \mathbf{x}_i) + K(\mathbf{x}_c, \mathbf{x}_c) - 2K(\mathbf{x}_i, \mathbf{x}_c)}. \end{aligned} \quad (19)$$

In the calculation of the distance in feature space, it does not need to know the explicit mapping function since it can be written solely in terms of the kernel function by the “kernel trick”. If condition (18) is satisfied or the number of training samples exceeds a given maximum limit, the sample with the maximum distance to the newly added sample will be removed from the training database in AOSVR.

When an old sample $\{\mathbf{x}_d, \mathbf{y}_d\}$ is deleted, the geometrical interpretation of the shift in the updated SVR regression curve is illustrated in Fig. 2. The regression curve of the updated model, denoted by $\bar{f}(\mathbf{x})$, tends to be dragged towards the location of the newly added sample $\{\mathbf{x}_c, \mathbf{y}_c\}$ after deleting the old sample. Considering that the kernel function of SVR is local in nature [23], it is anticipated that the estimation accuracy of the updated SVR model for the current behavior of process is improved. The relationship between the distribution of training samples and the robustness of SVR to outliers has been analyzed in detail [23]. It is found that the influence of outliers on SVR depends on the distance

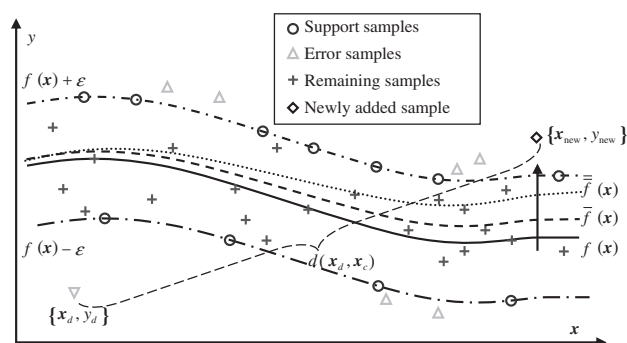


Fig. 2. Geometrical interpretation of the shift in the regression curve with an old sample deleted.

between outliers and the SVs nearest to them. Therefore, even if an outlier enters the training set, it may not have significant impact on the model performance, since the outlier generally locates far away from those normal training samples. As long as the new correct samples are constantly fed to the training set, the outlier will be removed from the training set in time and, accordingly, the model may recover from its damage quickly. The robustness to outliers can be improved, in such a way, to avoid data in abnormal conditions distorting the model. The computation burden of the pruning strategy depends on the calculation of Eq. (19), which is relatively small and suitable for online application.

The flowchart of the proposed model updating strategy is shown in Fig. 3. When a sample is available, the KKT conditions will be checked according to Eq. (16). If this condition is satisfied, the existing model is considered accurate enough and does not need update. Otherwise, the new sample should be added to update the SVR model. After that, the accuracy of the updated model with respect to the newly added sample will be evaluated under the condition of Eq. (18). If it is satisfied or the sample number reaches the given limit, the distance between the newly added sample and the old training samples is computed according to Eq. (19). The old sample with the largest distance is removed from the training set using the decremental algorithm. As a result, the updated model adjusts its structure to capture the changes in the process adaptively, while its size is maintained within a reasonable range with limited computation cost.

4. Nonlinear Model Predictive Control

We assume that the process to be controlled can be described by the SVR model with the following structure [22,24]

$$y(k) = f(y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)) + e(k) \quad (20)$$

where $y(k)$ and $u(k)$ are the process output and input variables with lags n_y and n_u , respectively, $e(t)$ is a white noise, and $f(\cdot)$ denotes the SVR model. At sampling instant k , the MPC problem is defined as

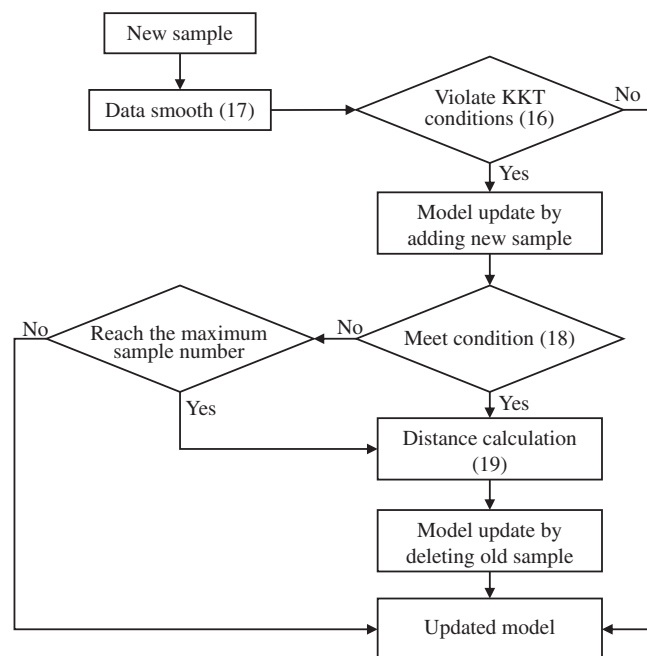


Fig. 3. Flowchart of the model updating strategy.

a constrained optimization problem, where the future manipulated input moves are determined by minimizing the objective function

$$\min_{u(k|k), \dots, u(k+M-1|k)} \sum_{i=1}^P (y_{sp} - y_p(k+i|k))^2 + \lambda \sum_{j=0}^{M-1} (\Delta u(k+j|k))^2 \quad (21)$$

subject to following constraints

$$\begin{aligned} \Delta u_{\min} &\leq \Delta u(k+j|k) \leq \Delta u_{\max}, \quad 0 \leq j \leq M-1 \\ u_{\min} &\leq u(k+j|k) \leq u_{\max}, \quad 0 \leq j \leq M-1 \\ y_{\min} &\leq y_p(k+i|k) \leq y_{\max}, \quad 1 \leq i \leq P \end{aligned} \quad (22)$$

where y_{sp} is the set point of controlled variable, $y_p(k+i|k)$ denotes the model prediction for process output $y(k+i)$ based on available measurements at sampling instant k , $\lambda \geq 0$ is a weighting factor, P and M denote the prediction horizon and control horizon, respectively, $\Delta u(k+j|k)$ is the future control increment, while $u(k+j|k)$ denotes the control action at the future sampling instant $k+j$ based on available measurements at instant k . The resulting constrained nonlinear optimization problem can be solved using standard nonlinear programming algorithms. MPC is implemented in a moving horizon manner, so after solving the optimization problem, only the first control action is applied to the plant and the optimization problem is reformulated at the next sampling instant $k+1$, based on the available information from the process.

The performance of MPC relies heavily on the quality of process model. However, even if an accurate model for the process is initially available, its performance may deteriorate substantially with time due to the shift at the operating point, changes in the disturbance characteristics, etc. Although the feedback mechanism of MPC tolerates some model mismatches, the presence of large plant-model mismatch may deteriorate the control performance significantly. To cope with this problem, an adaptive nonlinear MPC algorithm based on the SVR model updating strategy is presented. Specifically, at sampling instant k , the accuracy of existing SVR model, denoted as $f_k(\cdot)$, with respect to the new sample will be evaluated before it is used to provide predictions at sampling instant $k+1$. If it is not accurate enough to represent the process dynamics at the current operating point, it will be updated with the newly incoming sample, and control actions based on the current model will be applied to the process. Thus the adaptive control algorithm consists of two sub-stages, the model updating stage, where the model is updated to trace changes in the process and account for the modeling errors explicitly, and the control stage, where a nonlinear optimization problem is solved to yield control sequence and new samples for model updating.

5. Illustrative Example

An exothermic CSTR with irreversible reaction $A \rightarrow B$ is adopted to illustrate the method developed. The dynamic behavior of the CSTR process is described in a dimensionless form [20]

$$\begin{cases} \dot{x}_1 = -x_1 + Da(1-x_1) \exp\left(\frac{x_2}{1+x_2/\phi}\right) \\ \dot{x}_2 = -x_2 + B_h Da(1-x_1) \exp\left(\frac{x_2}{1+x_2/\phi}\right) + \beta_h(u-x_2) \end{cases} \quad (23)$$

where x_1 and x_2 are the reactant concentration and temperature, respectively, u is the input of cooling jacket with constraints $|u| \leq 2$ and $|u| \leq 0.25$, Da is the Damkohler number, ϕ is the activation energy, B_h is the heat of reaction, and β_h is the heat transfer coefficient. For $Da = 0.072$, $\phi = 20.0$, $B_h = 8.0$ and $\beta_h = 0.3$, the process exhibits output multiplicity and is open-loop unstable in the desired operating region [25]. Three operating points are selected as the set points for simulations: A ($x_1 = 0.144$, $x_2 = 0.8860$, $u = 0.0$) and C ($x_1 = 0.7646$, $x_2 = 4.7505$, $u = 0.0$) are stable operating points, and B ($x_1 = 0.4472$, $x_2 =$

2.7517, $u = 0.0$) is an unstable operating point. In this study, the output is $y = x_1$ and the dimensionless sample period is set to 0.5.

Initially, the SVR model is identified using the input–output data obtained from the closed-loop operation of the CSTR with a PID controller. The input and output lags are both set to 2. The parameters of the SVR model with the Gaussian kernel function are chosen as $\varepsilon = 0.01$, $C = 100$ and $\sigma = 0.1$. A set of 100 training data are used to train an initial SVR model, resulting in a model with 24 support vectors. For MPC, the prediction and control horizon are set to 7 and 2 samples, respectively, using different values and comparing control performance. The weighting factor λ is set to 0.4. The measurements of the process output, x_1 , are corrupted by the zero mean Gaussian white noise signals with the standard deviation of 0.05. It should be noted that identical parameters are chosen for all the MPC controllers in this section. Consequently, the conclusions drawn below do not depend on a particular choice of parameters. As far as the on-line model updating stage is concerned, the linear filter length N_0 is set to 6 and the threshold e_1 is set to 0.01.

The performance of MPC controllers with adaptive (denoted as MPC1) and non-adaptive SVR models (denoted as MPC2) for set point tracking is first investigated. The simulation results are shown in Fig. 4. The performance of the non-adaptive MPC controller is relatively good at the beginning, but deteriorates substantially with time as a consequence of shifts in the operating region. Specifically, when the set point changes from B to A at sample time 50 and changes from A back to B at sample time 100, the response of controller results in obvious steady state offset error and exhibits large overshoot behavior. The poor performance can be attributed to the lack of training data over the open-loop unstable region for the off-line trained model. In addition, it is noted that the output response for transition A to B shows an obvious spike, while the response for transition B to A is relatively smooth. This phenomenon is mainly caused by that B is an open-loop unstable operating point, which is more difficult to be controlled, compared with the stable operating point A. In contrast, the controller with adaptive model performs well consistently over the operating range of interest, though the transitions among different operating points could excite inherent nonlinearity of the process and pose significant challenges for process control. It indicates that the inclusion of the model updating stage is able to cope with the problem of plant-model mismatches in different operating regions.

Fig. 5 gives a better insight into the effectiveness of the model updating strategy, where the change in the number of SVs is depicted. The SVR model is not updated frequently even the process is subjected to changes in the operating region and random noise. Specifically, only 13 new samples are added and 12 old samples are deleted, about 3% of all of the samples (400). This is different from the conventional AOSVR algorithm based adaptive controller, where all of the samples are used to update the SVR model, resulting in frequent changes in the model formulation and, thus, a large computation load for online application.

The sum of squared error (SSE) between the set point and the process output over the sampling interval $[N_1, N_2]$ is used to assess the control performance, defined as

$$SSE = \sum_{k=N_1}^{N_2} (y_{sp} - y(k))^2 \quad (24)$$

The SSE values of the two controllers for set point tracking are listed in Table 1. The average CPU seconds needed by the two controllers at each step is also summarized in Table 1. The adaptive MPC controller outperforms the MPC based on the offline trained model, particularly when the process goes through the open-loop unstable region. It is noted that two controllers have nearly identical computation burden, though the adaptive controller needs to update the model whenever required. Compared with the computation time needed for solving the

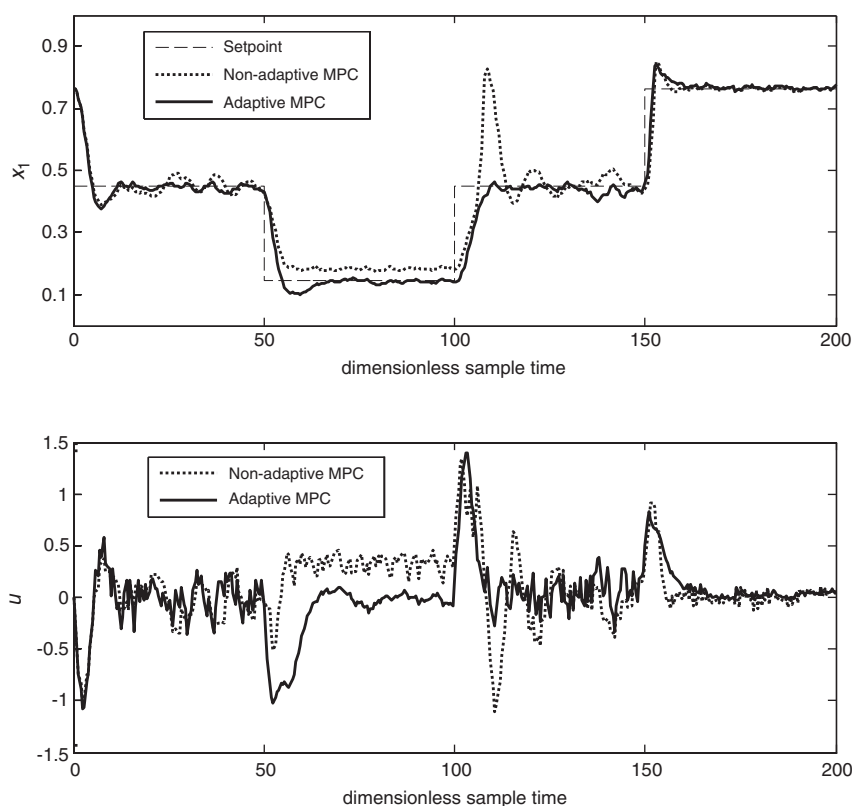


Fig. 4. Performance of MPCs with adaptive and non-adaptive models for set point tracking.

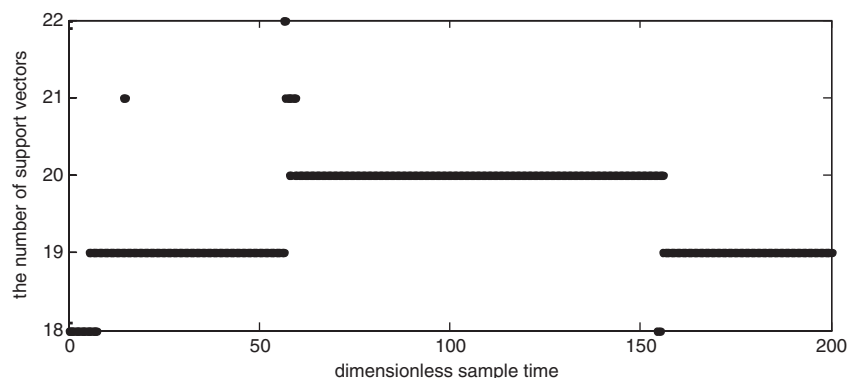


Fig. 5. Changes in the number of support vectors of the updated SVR model.

nonlinear optimization problem, the time required by the model updating stage is rather limited.

Starting from the unstable operating point B, the activation energy ϕ increases to 110% of its nominal value at sample time 40 and then the heat transfer coefficient β_h is subjected to a ramp type change with

decline rate of 5×10^{-3} per sample time at sample time 100 until it reaches 88% of its nominal value at sample time 125. In this case, the task of controller is to maintain the process output at the unstable operating point B despite disturbances. Responses of the two controllers for disturbance rejection are depicted in Fig. 6. The adaptive controller, which is able to recover from disturbances relatively quickly and maintain the process around the unstable operating point, showing better performance compared to its non-adaptive counterpart.

In Table 2, the SSE values and computation time are presented for the two MPC controllers. The results confirm that the adaptive controller presents better performance than the non-adaptive one, with limited computation time in the model updating stage. For this application, we conclude that the proposed model updating strategy is able to provide necessary process representation for developing an efficient adaptive MPC scheme, for both set point tracking and disturbance rejection.

Table 1

Performance of the two controllers for the set point tracking problem

Interval	SSE (MPC1)	SSE (MPC2)	Time/s (MPC1)	Time/s (MPC2)
[1, 50]	0.5823	0.6118	0.1291	0.1273
[51, 100]	0.3593	0.5550	0.1020	0.0984
[101, 150]	0.7362	1.3928	0.0752	0.0717
[151, 200]	0.2606	0.3563	0.1179	0.1154

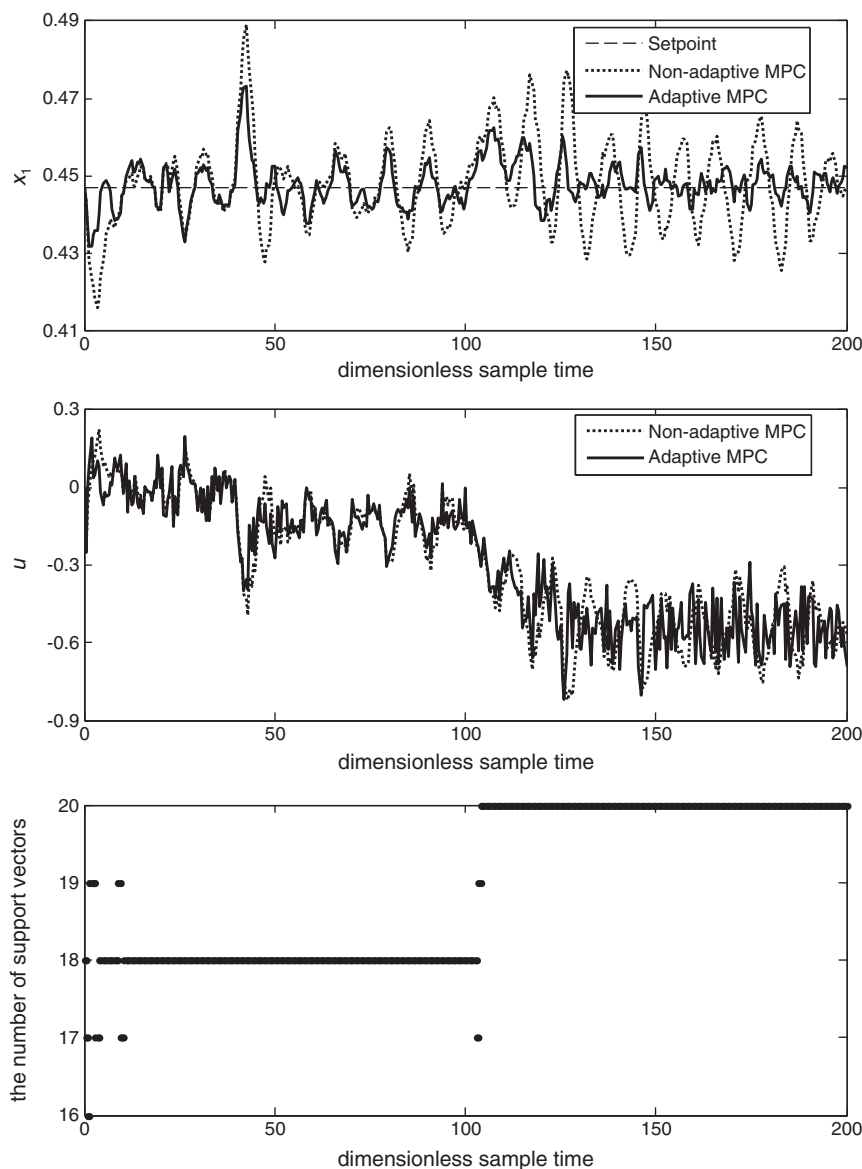


Fig. 6. Performance of MPCs with adaptive and non-adaptive models for disturbance rejection.

6. Conclusions

A methodology for efficiently updating SVR models based on the AOSVR algorithm was proposed. This is achieved by adding new sample that violates the KKT conditions of the existing SVR model and deleting old samples that has the maximum distance with respect to the newly added sample in feature space. The advantage of the proposed method is that it offers a scheme to update the existing SVR model in an online fashion with emphasizing the estimation accuracy for the current operating region. This provides a useful tool not only for predicting

the behavior of time varying processes, but also for developing adaptive model-based control schemes. The effectiveness of the adaptive control scheme is illustrated in the simulation study for control of a CSTR process. The simulation results confirm that the adaptive controller could provide better control performance than the non-adaptive one, while the computation time required by the model updating stage is rather limited. Further work is in progress to incorporate some persistent excitation conditions into the adaptive MPC scheme, according to process identification needs of the closed loop.

References

- [1] S.J. Qin, T.A. Badgwell, A survey of industrial model predictive control technology, *Control. Eng. Pract.* 11 (7) (2003) 733–764.
- [2] S. Engell, Feedback control for optimal process operation, *J. Process Control* 17 (3) (2007) 203–219.
- [3] L. Magni, M.D. Raimondo, F. Allgower, Nonlinear model predictive control: towards new challenging applications, *Lecture Notes in Control and Information Sciences*, vol. 384, Springer-Verlag, Berlin, 2009.
- [4] J.A.K. Suykens, Support vector machines and kernel-based learning for dynamical systems modelling, *Proceedings of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, 2009, pp. 1029–1037.
- [5] S.B. Chitrakha, S.L. Shah, Application of support vector regression for developing soft sensors for nonlinear processes, *Can. J. Chem. Eng.* 88 (5) (2010) 696–709.

Table 2

Performance of the two MPC controllers for disturbance rejection

Interval	SSE (MPC1)	SSE (MPC2)	Time/s (MPC1)	Time/s (MPC2)
[1, 50]	0.0060	0.0178	0.0976	0.0935
[51, 100]	0.0021	0.0060	0.0872	0.0854
[101, 150]	0.0042	0.0196	0.0945	0.0928
[151, 200]	0.0009	0.0104	0.0860	0.0856

- [6] J.L. Wang, T. Yu, C.Y. Jin, On-line estimation of biomass in fermentation process using support vector machine, *Chin. J. Chem. Eng.* 14 (3) (2006) 383–388.
- [7] S.N. Zhang, F.L. Wang, D.K. He, R.D. Jia, Real-time product quality control for batch processes based on stacked least-squares support vector regression models, *Comput. Chem. Eng.* 36 (10) (2012) 217–226.
- [8] J.A. Yu, Bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses, *Comput. Chem. Eng.* 41 (11) (2012) 134–144.
- [9] P. Kadlec, R. Grbic, B. Gabrys, Review of adaptation mechanisms for data-driven soft sensors, *Comput. Chem. Eng.* 35 (1) (2011) 1–24.
- [10] J.L. Liu, Development of self-validating soft sensors using fast moving window partial least squares, *Ind. Eng. Chem. Res.* 49 (22) (2010) 11530–11546.
- [11] Y. Liu, H.Q. Wang, J. Yu, Selective recursive kernel learning for online identification of nonlinear systems with NARX form, *J. Process Control* 20 (2) (2010) 181–194.
- [12] L.J. Li, H.Y. Su, J. Chu, Modeling of isomerization of C8 aromatics by online least squares support vector machine, *Chin. J. Chem. Eng.* 17 (3) (2009) 437–444.
- [13] K. Chen, J. Ji, H.Q. Wang, Adaptive local kernel-based learning for soft sensor modeling of nonlinear processes, *Chem. Eng. Res. Des.* 89 (10) (2011) 2117–2124.
- [14] D. Li, H. Li, T. Si, On-line robust modeling of nonlinear systems using support vector regression, Proceedings of 2nd International Conference on Advanced Computer Control (ICACC), Shenyang, China, 2010, pp. 204–208.
- [15] J. Ma, J. Theiler, S. Perkins, Accurate online support vector regression, *Neural Comput.* 15 (11) (2003) 2683–2704.
- [16] S. Iplikci, Online trained support vector machines-based generalized predictive control of non-linear systems, *Int. J. Adapt. Control Sig. Process* 20 (10) (2006) 599–621.
- [17] H. Wang, D.Y. Pi, Y.X. Su, Online SVM regression algorithm-based adaptive inverse control, *Neurocomputing* 70 (4–6) (2007) 952–959.
- [18] S.R. Sergio, M.J. Fuente, Fault tolerance in the framework of support vector machines based model predictive control, *Eng. Appl. Artif. Intell.* 23 (7) (2010) 1127–1139.
- [19] P. Wang, H.G. Tian, X.M. Tian, D.X. Huang, A new approach for online adaptive modeling using incremental support vector regression, *CIESC J.* 61 (8) (2010) 2040–2045.
- [20] J. Zhan, M. Ishida, The multi-step predictive control of nonlinear SISO processes with a neural model predictive control (NMPC) method, *Comput. Chem. Eng.* 21 (2) (1997) 201–210.
- [21] W.D. Zhou, L. Zhang, L.C. Jiao, An analysis of SVMs generalization performance, *Acta Electron. Sin.* 29 (5) (2001) 590–594.
- [22] Y. Liu, Y.C. Gao, Z.L. Gao, H.Q. Wang, P. Li, Simple nonlinear predictive control strategy for chemical processes using sparse kernel learning with polynomial form, *Ind. Eng. Chem. Res.* 49 (17) (2010) 8209–8218.
- [23] E. Wael, K. Mehmed, Local properties of RBF-SVM during training for incremental learning, Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, 2009, pp. 779–786.
- [24] W.M. Zhang, G.L. He, D.Y. Pi, Y.X. Su, SVM with polynomial kernel function based nonlinear model one-step-ahead predictive control, *Chin. J. Chem. Eng.* 13 (3) (2005) 373–379.
- [25] E. Hernandez, Y. Arkun, Study of the control-relevant properties of back propagation neural network models of nonlinear dynamical systems, *Comput. Chem. Eng.* 16 (4) (1992) 227–240.