Journal of Process Control 93 (2020) 53–65

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Fast tunable gradient RBF networks for online modeling of nonlinear and nonstationary dynamic processes

Tong Liu[a,b], Sheng Chen[c,d,*], Shan Liang[a,b], Dajun Du[e], Chris J. Harris[c]

[a] Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing University, Chongqing 400044, China
[b] College of Automation, Chongqing University, Chongqing 400044, China
[c] School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK
[d] King Abdulaziz University, Jeddah 21589, Saudi Arabia
[e] School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, 200444, China

ARTICLE INFO

Article history:
Received 5 May 2020
Received in revised form 1 July 2020
Accepted 30 July 2020
Available online 12 August 2020

Keywords:
Nonlinear and nonstationary dynamic processes
Gradient radial basis function network
Online modeling
Adaptive tuning mechanism

ABSTRACT

Since most real-world processes exhibit both nonlinear and time-varying characteristics, there exists a need for accurate and efficient models that can adapt in nonstationary environments. Also for adaptive control purpose, it is vital that an adaptive model has a fixed small model size. In this paper, we propose an adaptive tunable gradient radial basis function (GRBF) network for online modeling of nonlinear dynamic processes, which meets these practical requirements. Specifically, a compact GRBF model is constructed by the orthogonal least squares algorithm in training, which is capable of modeling variations of local mean and trend in the data well. During online operation, the adaptive GRBF model tacks the time-varying process's dynamics by replacing a worst performing node with a new node which encodes the current new data. By exploiting the local predictor property of the GRBF node, the new node optimization can be done extremely efficiently. The proposed approach combining the advantages of both the GRBF network structure and fast tunable node mechanism is capable of tracking the time-varying nonlinear dynamics accurately and effectively. Extensive simulation results demonstrate that the proposed fast tunable GRBF network significantly outperforms the existing state-of-the-art methods, in terms of both adaptive modeling accuracy and online computational complexity.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Nonstationary processes or systems are commonly encountered in many areas of science and engineering [1–4]. Nonstationary characteristics of processes can dramatically degrade the performance of predictive models, specifically, prevent models from maintaining a long-term high-accuracy. This is recognized as the most critical problem encountered in industrial applications, such as soft sensor [5–10]. Hence, online modeling of nonlinear dynamic processes is an important and challenging problem, and it has been drawing significant attentions [11,12]. An online model learning method must not only be equipped with effective adaptation mechanism for tracking time-varying process dynamics but also be sufficiently efficient to meet online processing restrictions [13]. The latter issue is crucial, particularly in most real-time control systems, where all adaptation operations much be completed within a small sampling period. Existing methods in the literature for online modeling of nonstationary systems can be classified into two groups: multiple local model learning and global kernel learning.

In a multiple local model learning strategy, the model input space is partitioned into multiple subspaces with the aid of a shifting window of samples. In each of the subspaces, the system is considered to be a stationary process, which is covered or modeled by a local model [14–18]. Based on this principle, the recently proposed selective ensemble based multiple local model (SEMLM) learning enables automatically identifying newly emerged process states online and combining the most up-to-date local linear models to make an accurate selective ensemble regression (SER) based prediction [17]. To further reduce the online computation burden for computing SER prediction, the more recent growing and pruning SER (GAP-SER) can not only learn the newly emerged concept but also forget the past accumulated old local models that are no longer relevant [18]. Both methods achieve excellent online prediction performance for nonstationary data modeling, with the GAP-SER imposing much less averaged computation time per sample (ACTpS) than

[*] Corresponding author.
E-mail addresses: liutong42@cqu.edu.cn (T. Liu), sqc@ecs.soton.ac.uk (S. Chen), lightsun@cqu.edu.cn (S. Liang), ddj559@163.com (D. Du), chrisharris57@msn.com (C.J. Harris).

https://doi.org/10.1016/j.jprocont.2020.07.009
0959-1524/© 2020 Elsevier Ltd. All rights reserved.

the SEMLM. Obviously, the performance of these algorithms depends critically on the window size, and there is no general criterion for selecting a proper window size. Also the SER prediction model constructed by the SEMLM and GAP-SER changes from sample to sample. This makes it difficult to employ such a changeable-size adaptive prediction model for implementing an adaptive controller for online control purposes.

Kernel based learning, as effective means of globally modeling nonlinear characteristics from data, have been widely used for nonlinear process modeling and identification. One typical approach of such global kernel learning is to adopt radial basis function (RBF) neural network. With a set of nonlinear kernels imposed on the training input data, the orthogonal least squares (OLS) learning can readily be applied to construct a parsimonious RBF model [19–22]. During online operation, a RBF network can track the process variations by updating its weight vector using adaptive estimators [23,24]. Similarly, the online sequential extreme learning machine (OS-ELM) [25–28] randomly selects a large number of training data as the RBF centers to fix the RBF model structure during offline training, and updates the model weights online using the recursive least squares (RLS) algorithm. Because the size of an OS-ELM model has to be very large to cover the overall process dynamics, the online weight adaptation is computationally costly and, moreover, there is no guarantee that the fixed RBF nodes, no matter how dense they are in the training data space, will also cover the changing nonstationary data space well. Therefore, this type of adaptive kernel based learning performs reasonably well only when the process dynamics vary slowly with time, but may fail to track the system dynamics when the process changes rapidly or abruptly.

In order to track the time-varying process characteristics well, the model structure should also be updated online. A simple way of adapting the RBF model structure is to grow the RBF nodes with arriving input data based on their significance, and this is the strategy adopted by the recourse-allocating network (RAN) [29,30]. A potential drawback of the RAN is that its size may grow to be very large. Also as aforementioned, the growing size of the RAN may hinder online adaptive model-based controller implementation. From an online control perspective, it is far better to adopt a compact fixed-size RBF model with tunable nodes rather than a RBF model with growing size. Unlike the RAN, the fast tunable RBF method of [31] fixes a compact model size in advance, and adjusts the RBF nodes as well as weights online to adaptively modeling nonstationary data. Specifically, during online operation when the error for the current data becomes unacceptable, the fast tunable RBF adapts the model structure by replacing an 'insignificant' node with a new node which is optimized to fit the current data. In this way, the fast tunable RBF is capable of efficiently capturing the newly emerged process state and encoding it in the fixed-size model, while maintaining a low online computational complexity. Simulation results of [31] show that this tunable RBF outperforms the RAN and OS-ELM in terms of both adaptive modeling accuracy and online computational complexity for modeling nonstationary data.

For nonstationary time series involving variations of local mean and trend, the series can be made stationary by applying a difference operation on the raw data [32]. By incorporating a similar mechanism into the RBF model, the gradient RBF (GRBF) network was proposed for nonstationary time series prediction in [33]. The input to this GRBF model is generated by differencing the original data, and each of its hidden nodes is modified to be a perfect local one-step predictor for an input vector in training. The OLS learning can readily be applied to select a compact GRBF network during training. This GRBF network significantly outperforms the RBF model for nonstationary time series prediction [33]. However, this GRBF network is restricted to time series prediction, not for identification of dynamic processes. Moreover, nonstationary data generally exhibit other time-varying characteristics, not just variations of local mean and trend. The GRBF network of [33] is non-adaptive because its model structure, constructed from training data, is fixed. Therefore, it cannot track the time-varying underlying process dynamics that are unseen in training.

Motivated by the above background, this paper proposes a fast tunable GRBF network for online modeling of time-varying dynamic processes, which is capable of tracking the nonlinear and nonstationary data well with a fixed small GRBF model structure, while imposing very low online complexity. Our novel contributions are summarized as follows.

1. Our first contribution is to extend the original GRBF network of [33] to a general form which is suitable for dynamic system identification problems. During the initial training, a compact GRBF model can readily be constructed by the OLS algorithm, similar to other RBF type models, such as [31,33].
2. Our main contribution is to derive an adaptive mechanism to adapt the GRBF model online. During online operation, the worst performing node of the GRBF model is replaced by a new node when the current GRBF network does not fit the current data well. Unlike the tunable RBF network of [31,34], which involves a complicated high-complexity iterative procedure to determine the structure of the new replacing RBF node, our adaptive algorithm of determining the new replacing GRBF node is not only very simple and hence imposing much less online computational complexity but also is optimal in the sense that the replacing GRBF node is the local one-step predictor of the current new data. Specifically, owing to the fundamental property of GRBF network structure, we can place the center and the associated scalar of the new replacing node to the current data and the output gradient, respectively. The new node then becomes a perfect local one-step predictor that encodes the current new data.
3. Extensive simulation experiments involving a simulated nonlinear time-varying process and two real-world chemical processes are used to evaluate this fast tunable GRBF network. The results obtained demonstrate that our fast tunable GRBF network significantly outperforms the existing state-of-the-art online models, including the GAP-SER [18] and the fast tunable RBF network [31], in terms of both adaptive modeling accuracy and online computational complexity.

2. The system model

We consider the generic discrete-time nonlinear and nonstationary dynamic process that can be represented by the following system model [35,36]

$$y_t = F(\boldsymbol{x}_t; t) + \xi_t, \tag{1}$$

where t denotes the sampling index, $F(\cdot; t)$ is the unknown time-varying underlying system mapping, and ξ_t is the uncorrelated observation noise with zero mean, while

$$\boldsymbol{x}_t = [x_{1,t} \cdots x_{n,t}]^{\mathrm{T}} = [y_{t-1} \cdots y_{t-n_y}\ u_{t-1} \cdots u_{t-n_u}]^{\mathrm{T}}, \tag{2}$$

is the system input vector with dimension $n = n_y + n_u$. In this system representation, u_t and y_t are the process input and output at sample t, respectively. while n_u and n_y are the input and output lags, respectively. The task of online modeling is that given the observation $\{\boldsymbol{x}_t, y_t\}$, constructing an estimator to approximate the underlying dynamic $F(\cdot; t)$, at every sampling time t. It can be seen that this is essentially a one-step ahead predictor problem. Extension to the multiple-step ahead predictor is straightforward.

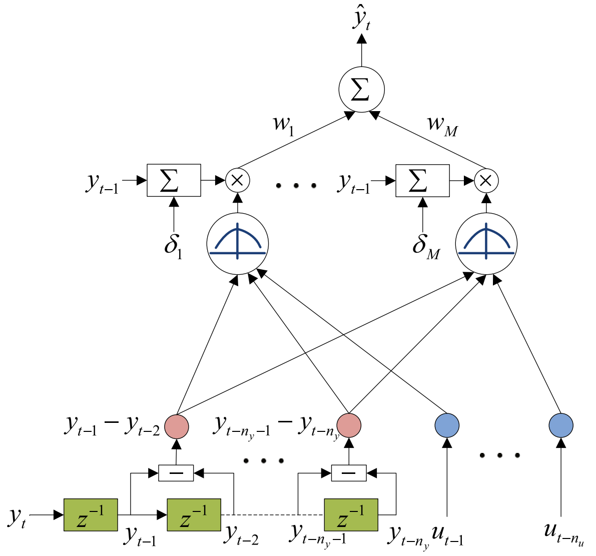


Fig. 1. The proposed GRBF network for dynamical system identification.

2.1. The proposed GRBF network

We extend the GRBF network of [33], originally designed for nonstationary time series prediction, to modeling the nonlinear time-varying dynamic system (1). The structure of our proposed GRBF network for this dynamical system identification is illustrated in Fig. 1. This identification task is to construct the GRBF network whose output for the given input (2) models the underlying system $F(\boldsymbol{x}_t;t)$. First, the input layer of the GRBF network generates the actual input vector to the hidden layer from $\boldsymbol{x}_t$ of (2) by differencing the system output sequence $\{y_{t-1},y_{t-2},\ldots,y_{t-n_y}\}$:

$$\boldsymbol{x}_t=\left[y_{t-1}-y_{t-2}\cdots y_{t-n_y}-y_{t-n_y-1}\; u_{t-1}\cdots u_{t-n_u}\right]^{\mathrm{T}}, \tag{3}$$

where for notational simplicity, we still denote this ‘differenced’ input vector as $\boldsymbol{x}_t$.

Let M be the number of the hidden nodes in this GRBF network. In a conventional RBF network, the Gaussian function is typically employed to serve as the hidden node's nonlinearity which compares the similarity of the input vector to the hidden node's center. This hidden node's function is kept in the GRBF network. The difference of the GRBF node and the classic RBF node is that the response of a GRBF node is further multiplied by an additional term $\left(y_{t-1}+\delta\right)$. Hence, the response of the jth GRBF hidden node to the input vector $\boldsymbol{x}_t$ is given by

$$\varphi_j(\boldsymbol{x}_t)=\exp\left(\frac{-\left\|\boldsymbol{x}_t-\boldsymbol{c}_j\right\|^2}{2\sigma^2}\right)\times\left(y_{t-1}+\delta_j\right), \tag{4}$$

where σ is the width of Gaussian kernel, $\boldsymbol{c}_j$ is the node center, and δ_j is a constant scalar associated with the hidden node. As the rule of thumb, the value of σ can be set as the maximum Euclidean distance among nodes [37]. The term $\left(y_{t-1}+\delta_j\right)$ an be interpreted as a local one-step prediction of y_t by the jth hidden node. From (4), if the input vector is very similar to the jth center, the value of the jth Gaussian function is close to 1 and the predictor $\left(y_{t-1}+\delta_j\right)$ becomes fully active.

The output of the GRBF network is then produced as the weighted sum of its hidden layer's response as

$$\hat{y}_t=\sum_{j=1}^{M}w_j\varphi_j(\boldsymbol{x}_t)=\boldsymbol{\phi}_{M,t}^{\mathrm{T}}\boldsymbol{w}_M, \tag{5}$$

which is an estimate of the system's output y_t, where $\boldsymbol{w}_M=[w_1\cdots w_M]^{\mathrm{T}}$ is the output layer's weight vector, and $\boldsymbol{\phi}_{M,t}=[\varphi_1(\boldsymbol{x}_t)\cdots\varphi_M(\boldsymbol{x}_t)]^{\mathrm{T}}$ is the hidden layer's response vector.

2.2. OLS construction of GRBF network in training

Given the initial training data set of $\left\{\boldsymbol{x}_t,d_t;y_t\right\}_{t=1}^{N}$, where

$$d_t=y_t-y_{t-1}, \tag{6}$$

a compact M-term GRBF network with $M\ll N$ can readily be constructed from the training data set $\left\{\boldsymbol{x}_t,d_t;y_t\right\}_{t=1}^{N}$ using the OLS algorithm [19–21]. Specifically, similar to [33], if $\boldsymbol{x}_t$ is selected as the jth center $\boldsymbol{c}_j$, we set $\delta_j=d_t$ to ensure that the jth hidden node is a perfect local predictor of y_t, which is illustrated in Fig. 2(a). By considering all the data points $\{\boldsymbol{x}_t,d_t\}_{t=1}^{N}$ as the GRBF hidden nodes $\{\boldsymbol{c}_j,\delta_j\}_{j=1}^{N}$, we obtain the full N-term GRBF network. This full N-term GRBF network over the training data set is expressed as

$$\boldsymbol{y}_N=\boldsymbol{\Psi}_N\boldsymbol{w}_N+\boldsymbol{\xi}_N, \tag{7}$$

where $\boldsymbol{y}_N=[y_1\cdots y_N]^{\mathrm{T}}$, $\boldsymbol{\xi}_N=[\xi_1\cdots\xi_N]^{\mathrm{T}}$ are the modeling errors, and

$$\boldsymbol{\Psi}_N=\left[\boldsymbol{\psi}_1\;\boldsymbol{\psi}_2\cdots\boldsymbol{\psi}_N\right]=\left[\boldsymbol{\phi}_{N,1}\;\boldsymbol{\phi}_{N,2}\cdots\boldsymbol{\phi}_{N,N}\right]^{\mathrm{T}}. \tag{8}$$

The OLS algorithm selects a subset of M significant model bases from the full set of N model bases defined in (8) [19–21]. Specifically, let the orthogonal decomposition of the regression matrix $\boldsymbol{\Psi}_N$ be $\boldsymbol{\Psi}_N=\boldsymbol{P}_N\boldsymbol{A}_N$, where $\boldsymbol{A}_N$ is the upper triangular matrix defined by

$$\boldsymbol{A}_N=\begin{bmatrix}1 & a_{1,2} & \cdots & a_{1,N}\\ 0 & 1 & \cdots & a_{2,N}\\ \vdots & \ddots & \ddots & \vdots\\ 0 & \cdots & 0 & 1\end{bmatrix}, \tag{9}$$

and $\boldsymbol{P}_N=\left[\boldsymbol{p}_1\;\boldsymbol{p}_2\cdots\boldsymbol{p}_N\right]\in\mathbb{R}^{N\times N}$ denotes the orthogonal regression matrix which satisfies $\boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{p}_j=0$ for $i\neq j$. The space spanned by the set of the orthogonal bases $\{\boldsymbol{p}_i\}$ is the same space spanned by the set of the original bases $\{\boldsymbol{\psi}_i\}$, and (7) can be rewritten as

$$\boldsymbol{y}_N=\boldsymbol{P}_N\boldsymbol{\theta}_N+\boldsymbol{\xi}_N, \tag{10}$$

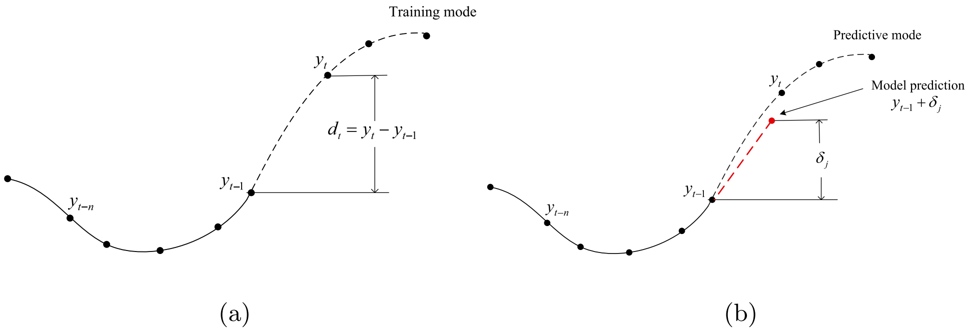
where the transformed weight vector $\boldsymbol{\theta}_N=\left[\theta_1\;\theta_2\cdots\theta_N\right]^{\mathrm{T}}=\boldsymbol{A}_N\boldsymbol{w}_N$, whose elements are readily calculated according to the least squares (LS) estimator as $\theta_i=\boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{y}_N/\boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{p}_i$ for $1\le i\le N$. The sum of squares or energy of $\boldsymbol{y}_N$ is given by

$$\boldsymbol{y}_N^{\mathrm{T}}\boldsymbol{y}_N=\sum_{j=1}^{N}\theta_j^2\boldsymbol{p}_j^{\mathrm{T}}\boldsymbol{p}_j+\boldsymbol{\xi}_N^{\mathrm{T}}\boldsymbol{\xi}_N. \tag{11}$$

The contribution of the jth node can be measured by its error reduction ratio $[\mathrm{err}]_j$ defined as

$$[\mathrm{err}]_j=\frac{\boldsymbol{p}_j^{\mathrm{T}}\boldsymbol{p}_j\theta_j^2}{\boldsymbol{y}_N^{\mathrm{T}}\boldsymbol{y}_N}. \tag{12}$$

This error reduction ratio offers an effective means of seeking the M significant model bases (centers and scalars $\{\boldsymbol{c}_i,\delta_i\}$) in a greedy-based forward selection procedure [19–21]. Specifically, at each selection step, the candidate with the maximum error reduction ratio among all the remaining candidates is selected as the new node. The regression procedure is terminated when $1-\sum_{j=1}^{M}[\mathrm{err}]_j$ is smaller than a pre-set threshold, and this yields a regression model with M hidden nodes $\{\boldsymbol{c}_j,\delta_j\}_{j=1}^{M}$, together with the selected subset orthogonal regression matrix $\boldsymbol{P}_M\in\mathbb{R}^{N\times M}$ and



(a) (b)

Fig. 2. The jth hidden node in (a) Training mode: if the training input vector $\boldsymbol{x}_t$ is selected as the center $\boldsymbol{c}_j$, the scalar δ_j is set to d_t, and the jth node becomes a perfect local predictor of y_t; and (b) Predictive mode: if the center $\boldsymbol{c}_j$ matches the new observation $\boldsymbol{x}_t$, $\left(y_{t-1}+\delta_j\right)$ becomes an accurate local estimator of y_t.

the associated weight vector $\boldsymbol{g}_M \in \mathbb{R}^M$ as well as the corresponding upper triangular matrix $\boldsymbol{A}_M \in \mathbb{R}^{M\times M}$. Then the weight vector $\boldsymbol{w}_M$ of the selected M-term subset GRBF network can readily be solved from $\boldsymbol{g}_M = \boldsymbol{A}_M\boldsymbol{w}_M$ by the backward substitution.

2.3. Prediction capability of GRBF network

After training, the resulting compact GRBF network is ready for predictive operation. Similar to the classic RBF network, each hidden node compares the current input $\boldsymbol{x}_t$ with its center $\boldsymbol{c}_j$. The Gaussian response of each hidden node indicates the degree of matching between $\boldsymbol{x}_t$ and $\boldsymbol{c}_j$. However, unlike the conventional RBF network whose prediction is the weighted Gaussian response of all the hidden nodes, where weights are fixed by training, in the case of the GRBF network, if the jth center $\boldsymbol{c}_j$ matches well the current input vector $\boldsymbol{x}_t$, $(y_{t-1}+\delta_j)$ is likely to be a very good prediction of y_t, which is illustrated in Fig. 2(b). Hence the jth hidden node's response is 'amplified' by its accurate local predictor. Consequently, the prediction of the GRBF network is weighted more heavily by the jth local predictor, leading to the more accurate prediction for y_t than the classic RBF network. This is the elegance of the GRBF network compared with the classic RBF network.

The above discussion indicates that the GRBF network has better prediction capability than the classic RBF network for stationary processes. Furthermore, since the GRBF network naturally models variations of local means and trend in nonstationary processes, the fixed or non-adaptive GRBF network obtained in training has better prediction capability than its RBF network counterpart for nonstationary processes.

For a highly time-varying process, however, the process dynamics can vary dramatically, and it exhibits the serious nonstationary characteristics other than variations of local means and trend, the fixed GRBF network constructed in training may become unable to track the process's changing dynamics. A main contribution of this paper is to propose a highly effective and very efficient adaptive learning strategy for adapting the GRBF network online.

3. Proposed online adaptive learning

A simple way of adapting the GRBF network online is to apply the RLS algorithm for online weight adaptation. However, this is unlikely to be sufficient for highly time-varying systems. Thus both the model structure and weight parameters need to be updated online. Clearly, optimizing all the hidden nodes every sample is impractical, because it cannot meet the real-time constraint. Furthermore, since the nodes of the GRBF network encode the past system dynamics learned, it may be unwise to change them all. In order to meet the tight real-time constraint of small sampling period, the fast tunable RBF method of [31] adopts the following online adaptive strategy. It carries the weight updating every sample using the RLS algorithm as usual. If the prediction performance of the RBF model is judged to be insufficient, it replaces the hidden node that contributes least with a new one. This involves an iterative optimization procedure based on gradient descend to determine the replacement node's center and width. This kind of adaptive strategy is attractive owing to its excellent online modeling performance, and we adopt a similar online adaptive learning strategy for the GRBF network.

Note that the complicated iterative optimization of [31] may impose relatively high online computational complexity that may violate the small sampling period constraint. By contrast, we do not need complicated iterative optimization to determine the replacement node's structure and, consequently, the adaptive learning for the GRBF network imposes much lower online computational complexity. Recalling from the structure of GRBF network, each hidden node basically encodes a local process state, i.e., an input vector as its center, and it acts as a perfect local predictor for the corresponding output. Hence, we only need to set the replacement node's center to the current input and its scalar to the current output gradient.

Specifically, there are two adaptive modes during online adaptation of GRBF network, namely, the *weight adaptation mode* and *node adaptation mode*. During online operation, the residual error of the GRBF network is monitored. If the GRBF network performs poorly, i.e., the residual error is large, the *node adaptation mode* take places to replace the worst node with a new one. Otherwise only the *weight adaptation mode* operates. Let $\boldsymbol{\phi}_{M,t}$ denote the hidden layer response vector for the given input $\boldsymbol{x}_t$ and $\boldsymbol{w}_{M,t-1}$ be the weight vector obtained at the previous sample. Then the residual for the prediction of y_t based on current model is given by

$$e_t = y_t - \boldsymbol{\phi}_{M,t}^{\mathrm{T}}\boldsymbol{w}_{M,t-1}. \tag{13}$$

The model performance can be measured by the cost

$$\widetilde{e}_t = \frac{e_t^2}{y_t^2}. \tag{14}$$

Then, we have the following criterion

$$\begin{cases} \text{IF } \widetilde{e}_t \geq \varepsilon & \textit{Node adaptation mode}, \\ \text{IF } \widetilde{e}_t < \varepsilon & \textit{Weight adaptation mode}, \end{cases} \tag{15}$$

where ε is a constant threshold which is set according to the performance requirement. In general, the smaller ε is, the better online modeling accuracy can achieve, but the more frequently the node replacement may occur which increases online computational complexity. This is because every time a node replacement takes place, the weight vector must be recalculated using the regularized LS estimator. See Section 3.1. The impact of ε on the achievable performance will be investigated further in Section 4.

3.1. Node adaptation mode

In this mode, the most insignificant node with least contribution to the model performance is replaced by a new node. Similar to [31], the significance of a node is defined by its weighted node output (WNO). Hence for the jth node

$$\mathrm{WNO}_j = w_{j,t-1}^2 \varphi_j^2(\boldsymbol{x}_t), \;\; 1 \le j \le M. \tag{16}$$

We select the node with the smallest WNO value from all the M nodes. Let

$$\mathrm{WNO}_K = \min_{1 \le j \le M} \mathrm{WNO}_j. \tag{17}$$

Then the node K is replaced by a new one.

Two parameters associated with the new replacement node K is its center $\boldsymbol{c}_K$ and scalar δ_K. Recall that in constructing the GRBF network during training, a hidden node's center and associated scalar are set to an input and the corresponding output gradient, respectively. Exploiting this physical property of GRBF hidden nodes, we can simply set $\boldsymbol{c}_K = \boldsymbol{x}_t$ and $\delta_K = y_t - y_{t-1}$. It can be seen that there is no need at all to use an iterative optimization procedure to determine the replacement node as in the case of the fast tunable RBF network [31], and the new replacement node is a perfect local predictor of the current output y_t. Thus, our tunable node mechanism enables automatically encoding the newly emerged process state and forgetting the most out-of-date data state, while imposing a very low complexity. Noted that since the set of the centers now contain a new one, the width of the Gaussian response σ should be recalculated based on the maximum Euclidean distance among the centers.

After the new hidden node is determined, the whole weight vector of the new GRBF network needs to be recalculated. We use the p latest data $\left\{\boldsymbol{x}_{t-i}, y_{t-i}\right\}_{i=0}^{p-1}$ to calculate the weight vector of the newly undated GRBF network based on the regularized LS estimator. Specifically, the estimate of $\boldsymbol{w}_{M,t}$ is obtained by minimizing the regularized LS cost function

$$J_t = \left\| \boldsymbol{y}_p - \boldsymbol{\Phi}_{p\times M} \boldsymbol{w}_{M,t} \right\|^2 + \lambda \| \boldsymbol{w}_{M,t} \|^2, \tag{18}$$

where $\lambda \ge 0$ is the regularization parameter, which is used to prevent overfitting when p is small, and $\boldsymbol{y}_p = [y_t \; y_{t-1} \cdots y_{t-p+1}]^{\mathrm{T}} \in \mathbb{R}^p$, while the regression matrix $\boldsymbol{\Phi}_{p\times M} \in \mathbb{R}^{p\times M}$ is given by

$$\boldsymbol{\Phi}_{p\times M} = \begin{bmatrix} \varphi_1(\boldsymbol{x}_t) & \varphi_2(\boldsymbol{x}_t) & \cdots & \varphi_M(\boldsymbol{x}_t) \\ \varphi_1(\boldsymbol{x}_{t-1}) & \varphi_2(\boldsymbol{x}_{t-1}) & \cdots & \varphi_M(\boldsymbol{x}_{t-1}) \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_1(\boldsymbol{x}_{t-p+1}) & \varphi_2(\boldsymbol{x}_{t-p+1}) & \cdots & \varphi_M(\boldsymbol{x}_{t-p+1}) \end{bmatrix}. \tag{19}$$

The regularized LS solution of (18) is given by

$$\boldsymbol{w}_{M,t} = \left(\boldsymbol{\Phi}_{p\times M}^{\mathrm{T}} \boldsymbol{\Phi}_{p\times M} + \lambda \boldsymbol{I}_M \right)^{-1} \boldsymbol{\Phi}_{p\times M}^{\mathrm{T}} \boldsymbol{y}_p, \tag{20}$$

where $\boldsymbol{I}_M$ denotes the $M \times M$ identity matrix.

In general, a large p offers accurate LS estimation but may degrade the tracking performance for fast time-varying systems, while imposing relatively high complexity. For highly time-varying processes with abrupt drift, a small p is preferred.

3.2. Weight adaptation mode

In this mode, the hidden layer of the GRBF network has not been changed, and we can simply apply the standard RLS algorithm to update the weight vector

$$\begin{cases} \boldsymbol{k}_t = \boldsymbol{\Gamma}_{t-1} \boldsymbol{\phi}_{M,t} \left(\gamma + \boldsymbol{\phi}_{M,t}^{\mathrm{T}} \boldsymbol{\Gamma}_{t-1} \boldsymbol{\phi}_{M,t} \right)^{-1}, \\ \boldsymbol{\Gamma}_t = \left(\boldsymbol{\Gamma}_{t-1} - \boldsymbol{k}_t \boldsymbol{\phi}_{M,t}^{\mathrm{T}} \boldsymbol{\Gamma}_{t-1} \right) \gamma^{-1}, \\ \boldsymbol{w}_{M,t} = \boldsymbol{w}_{M,t-1} + \boldsymbol{k}_t e_t, \end{cases} \tag{21}$$

Algorithm 1 Fast tunable GRBF network

```
1: Parameters: Node replacement threshold $\varepsilon$, bandwidth $p$,
   regularization parameter $\lambda$, forgetting factor $\gamma$.
2: Initialize: Construct initial $M$-term GRBF model based on
   training data set $\{\boldsymbol{x}_t, y_t\}_{t=1}^N$ with OLS algorithm, set $\boldsymbol{\Gamma}_N = \vartheta \boldsymbol{I}_M$,
   and denote weight vector as $\boldsymbol{w}_{M,N}$.
3: for sample instant $t = N+1, N+2, \ldots$ do
4:   Compute prediction $\widehat{y}_t = \boldsymbol{\phi}_{M,t}^{\mathrm{T}} \boldsymbol{w}_{M,t-1}$ for given $\boldsymbol{x}_t$.
5:   When $y_t$ is available, compute $\widetilde{e}_t$ with (13) and (14).
6:   if $\widetilde{e}_t < \varepsilon$
7:     Update weight vector to $\boldsymbol{w}_{M,t}$ with RLS (21).
8:   else if $\widetilde{e}_t \ge \varepsilon$
9:     Compute the WNO values for all $M$ nodes with (16).
10:    Find the minimum WNO node, node $K$, and replace it with
   a new node:
       set new center to $\boldsymbol{c}_K = \boldsymbol{x}_t$ and new scalar to $\delta_K = y_t - y_{t-1}$.

11:     Compute new Gaussian width $\sigma$ based on maximum
   Euclidean distance
       among nodes.
12:    Use $p$ latest data $\left\{\boldsymbol{x}_{t-i}, y_{t-i}\right\}_{i=0}^{p-1}$ to compute weight vector
   $\boldsymbol{w}_{M,t}$ with
       regularized LS estimation (20).
13:    Re-initialize $\boldsymbol{\Gamma}_t$ according to (22).
14:  end if
15: end for
```

where $\boldsymbol{k}_t \in \mathbb{R}^M$ is the Kalman gain vector, $0.9 \le \gamma < 1$ is the forgetting factor, and the inverse of covariance matrix $\boldsymbol{\Gamma}_t \in \mathbb{R}^{M\times M}$ is usually initialized to $\boldsymbol{\Gamma}_0 = \vartheta \boldsymbol{I}_M$ in which ϑ is a large positive constant.

It is worth noted that if a node adaptation takes place at sample t, the RLS updating (21) will not take place, and we need to initialize the inverse covariance matrix to

$$\boldsymbol{\Gamma}_t = \left(\boldsymbol{\Phi}_{p\times M}^{\mathrm{T}} \boldsymbol{\Phi}_{p\times M} + \lambda \boldsymbol{I}_M \right)^{-1} \tag{22}$$

after the regularized LS estimation (20), This will ensures a smooth transition from the *node adaptation mode* to the *weight adaptation mode* at the next sample.

3.3. Algorithm summary

Algorithm 1 summarizes the operations of our proposed fast tunable GRBF network, which consists of two learning phases: (1) off-line initialization, and (2) online prediction and adaptation. During the off-line initialization (line 2), a parsimonious M-term GRBF model is constructed based on the training data set with the OLS learning. The selected hidden nodes stores the most relevant knowledge or process characteristics from the training data. Since the process's dynamics are time-varying, during the online operation, in addition to provide the prediction, we need to remove the out-of-date knowledge and to encode the newly emerged process state as fast as possible. This is carried out by the online prediction and adaptation for each sample t (lines 4 to 14). More specifically, if the observed process variation is small, the GRBF network operates in the *weight adaptation mode*, where its weight vector is updated with the RLS algorithm to track the smooth and small data variation. When the abrupt changes occur in the process, the GRBF network switches to the *node adaptation mode*, where a most out-of-date node is replaced by a new node to encode the newly emerged process state.

The computational complexity of operating in the *node adaptation mode* is dominated by the regularized LS estimation (20), which is on the order of $\mathrm{O}(p^3)$, while the complexity of operating in the *weight adaptation node* is on the order of $\text{O}(M^2)$, which is determined by the RLS algorithm (21). Therefore, the online computational complexity per sample of Algorithm 1 is no more than $\max\{\text{O}(p^3), \text{O}(M^2)\}$. Since p and M are typically very small, the online computational complexity per sample of the proposed algorithm is clearly affordable and is much less than the fast tunable RBF, which involves an iterative optimization procedure to determine the replacement RBF node [31]. Very low adaptation complexity of our fast tunable GRBF network coupled with its small and fixed model size makes it particularly suitable for adaptive control applications.

4. Simulation results

The performance of the proposed fast tunable GRBF network is evaluated using three case studies, a synthetic nonlinear time-varying system and two real-world chemical processes, specifically, an industrial debutanizer column process [8,38,39] and a sulfur recovery unit (SRU) process [40]. The mean square error (MSE) metric

$$\text{MSE}_t = \frac{1}{t}\sum_{i=1}^{t}\left(y_i - \widehat{y}_i\right)^2, \tag{23}$$

is used to evaluate the online modeling accuracy, where $\widehat{y}_i$ is the model prediction for y_i. Following the common practice, the value of the MSE is expressed in dB, namely, in $10\log_{10}$ MSE. The online computational complexity is quantified by the ACTpS. The experiments are carried out on Matlab 2017a, running on a PC with i7-3770 3.40 GHz processor of 4 cores and 16 GB of RAM. The proposed fast tunable GRBF model is compared with various online modeling approaches, including the OS-ELM [25–28], the RBF network [19,20], the GRBF network presented in Section 2 of this paper, and the fast tunable RBF network [31] as well as our SEMLM [17] and GAP-SER [18].

For the OS-ELM with RBF hidden nodes, the centers are randomly chosen from the training data points, and online weight adaptation is performed by the RLS algorithm. Similar to the fast tunable GRBF network, the initial compact RBF, GRBF and fast tunable RBF networks are constructed from the training data set using the OLS algorithm. During online operation, the RBF network and the GRBF network only perform weight adaptation by the RLS algorithm, while the fast tunable RBF network adjust both weight and structure using an iterative scheme proposed in [31]. Like the proposed tunable GRBF network, the node replacement threshold ε and bandwidth p for the tunable RBF network need to be carefully chosen. For the tunable RBF, the gradient descent is used to optimize the new node, where the step size and the number of iterations are empirically set to 0.1 and 5, respectively, for all the three cases. Also the regularization parameter is $\lambda = 0.001$ for the regularized LS estimation and the forgetting factor of the RLS algorithm is $\gamma = 0.98$, for all the three cases.

The SEMLM and GAP-SER are quite different from the above fixed-size models. A set of local linear models are initialized on the training data set. During online operation, both SER prediction and local model adaptation are performed. The key algorithmic parameters for these two adaptive learning schemes are the window size W, bandwidth p and model selection threshold ε. The details of the algorithmic parameter sensitive analysis for these two methods can be found in [17,18].

4.1. Synthetic nonlinear time-varying system

This synthetic time-varying nonlinear system is described by the following difference equation [41]

$$y_t = 0.72y_{t-1} + 0.025y_{t-2}u_{t-1} + 0.01u_{t-2}^2 + 0.2u_{t-3}. \tag{24}$$

Table 1
Performance comparison of the OS-ELM, RBF, GRBF, SEMLM, GAP-SER, tunable RBF and proposed method for online identification of synthetic nonlinear time-varying system.

Model	Nodes/Models	MSE (dB)	ACTpS (ms)
OS-ELM	100 500	−17.7504 −22.3459	0.2635 4.0074
RBF	10 20	−22.0003 −22.1841	0.0171 0.0219
GRBF	10 20	−30.4484 −30.5935	0.0160 0.0227
SEMLM	2 to 231	−31.3484	0.7751
GAP-SER	2 to 2	**−31.2119**	**0.3398**
Tunable RBF	10	**−28.1882**	**0.1883**
Proposed	10	**−32.1531**	**0.1618**

The input signal u_t is generated as follows. For $t \le 500$, u_t is the independent and identically distributed (i.i.d.) uniform sequence over $[-1,\ 1]$, and

$$u_t = \begin{cases} \sin\left(\frac{\pi t}{45}\right), & 500 < t \le 1000, \\ \sin\left(\frac{\pi t}{25}\right), & 1000 < t \le 1250, \\ +1, & 1250 < t \le 1500, \\ -1, & 1500 < t \le 1750, \\ 0.3\sin\left(\frac{\pi t}{25}\right) + 0.1\sin\left(\frac{\pi t}{32}\right) & \\ \quad +0.6\sin\left(\frac{\pi t}{10}\right), & 1750 < t \le 2000. \end{cases} \tag{25}$$

The data set contains 2000 data samples $\{\boldsymbol{x}_t, y_t\}$ generated using (24) and (25) with the input vector $\boldsymbol{x}_t$ given by

$$\boldsymbol{x}_t = \left[u_{t-1}\ u_{t-2}\ u_{t-3}\ y_{t-1}\ y_{t-2}\right]^{\mathrm{T}}. \tag{26}$$

The first 1000 pairs of the data are used for training and the remaining 1000 pairs for online prediction and adaptation. Clearly, the system (24) with the input (25) is nonstationary, as the distribution of y_t is time varying.

In general, the model size of the OS-ELM network must be much larger than the RBF/GRBF networks constructed by OLS learning, as the OS-ELM randomly selects the input data points as its centers in training. For the SEMLM and GAP-SER, the window size W and bandwidth p are carefully tuned to be 30 and 5, while their model selection thresholds are empirically set to $\varepsilon = 0.01$ and 0.7, respectively. Note that the model selection criteria for the SEMLM and GAP-SER are different, and hence appropriate selection thresholds for them are different. For the tunable RBF and the proposed tunable GRBF, the node replacement thresholds are set to $\varepsilon = 0.1$ and 0.01, respectively, while their bandwidths are both $p = 1$. Note that choosing a very small p is appropriate for this highly time-varying process with abrupt dynamic changes.

Table 1 compares the performance of various online adaptive models. Observe that the performance of the OS-ELM is poor. To achieve a similar online prediction accuracy as a small RBF network, it needs 500 hidden nodes and imposes the highest ACTpS of 4 ms. The compact RBF and GRBF models attain the lowest ACTpS, as they only adapt a small weight vector using the RLS algorithm. It can also be seen that a small network size of 10 hidden nodes is sufficient for the RBF and GRBF networks. Observe that the GRBF network presented in Section 3 is 8 dB better in the online prediction MSE than the classic RBF model. For the SEMLM, the number of the local linear models grows from the initial 2 to 231. By contrast, the GAP-SER with its effective pruning strategy is able to adapt the set of linear local models while keeping its size small and consequently maintaining a lower ACTpS than the SEMLM. The online prediction accuracy of the tunable RBF model [31] is 6 dB better than that of the

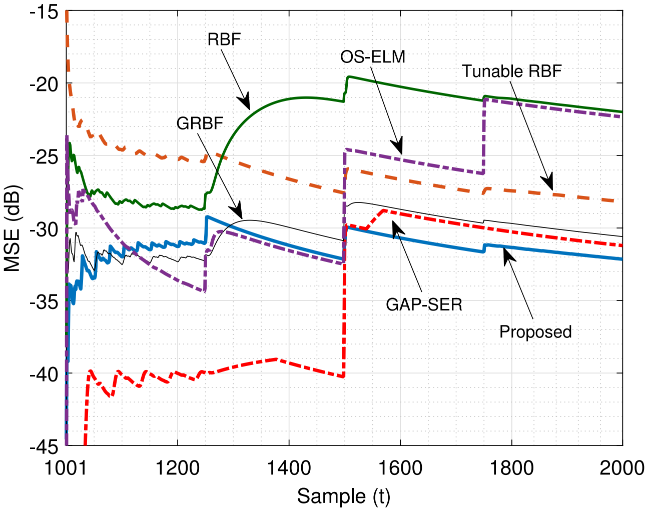


Fig. 3. MSE learning curves for synthetic nonlinear time-varying system identification by various adaptive models. Both the RBF and GRBF have 20 hidden nodes, while the OS-ELM has 500 nodes.

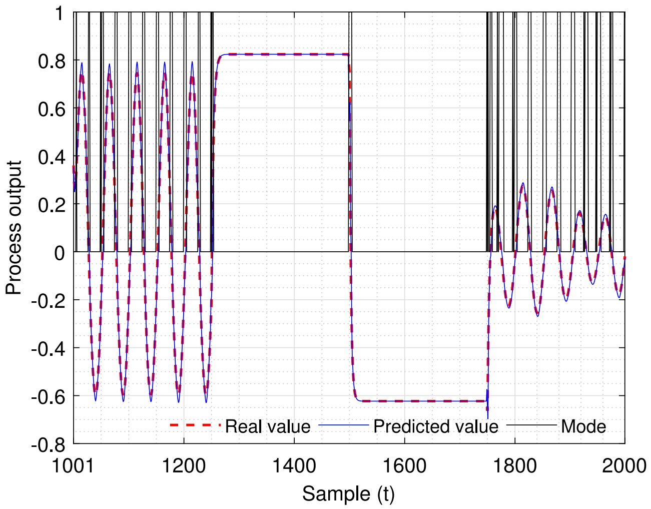


Fig. 4. Model prediction by the proposed method for online identification of synthetic nonlinear time-varying system. The black vertical line 'mode' indicates the sample where mode switching (either from *node adaptation mode* to *weight adaptation mode* or from *weight adaptation mode* to *node adaptation mode*) take places during online operation.

RBF network. The proposed tunable GRBF model achieves the best online prediction accuracy, while imposing a very low online complexity. Observe that the online prediction accuracy of the tunable GRBF is 2 dB better than the GRBF network.

The MSE learning curves of the OS-ELM with 500 nodes, the RBF and GRBF with 20 nodes, the GAP-SER, the tunable RBF and the proposed tunable GRBF are compared in Fig. 3. During the online operation, there are three major changes in the underlying process dynamics around $t = 1250$, 1500 and 1750, respectively, lined to the three switches in the input u_t as given in (25). The first and third changes are relatively smooth, as the related switches of u_t are relatively smooth with relatively small input magnitude changes. The second process dynamic change at $t = 1500$ is very sharp as u_t switches from $+1$ to -1 at $t = 1500$. The OS-ELM has the worst online adaptation ability, and it experiences sharp increases in MSE at $t = 1250$, 1500 and 1750. Also the GAP-SER experiences a sharp increase in MSE at $t = 1500$ when the process is undergoing a sharp dynamic change.

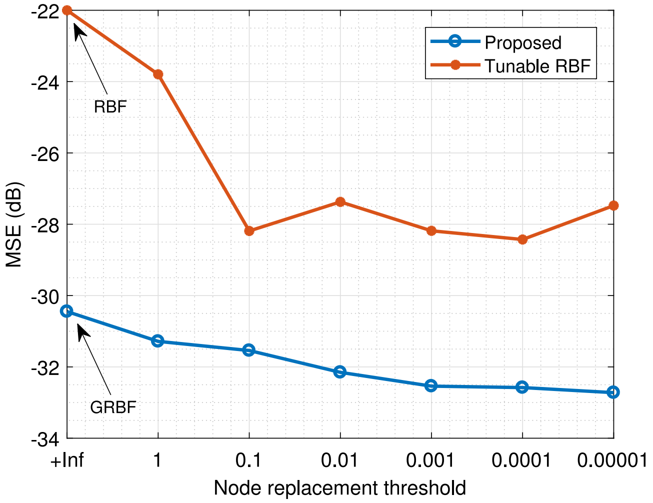


Fig. 5. The impact of node replacement threshold on the online modeling accuracy of synthetic nonlinear time-varying system for the tunable RBF and proposed tunable GRBF.

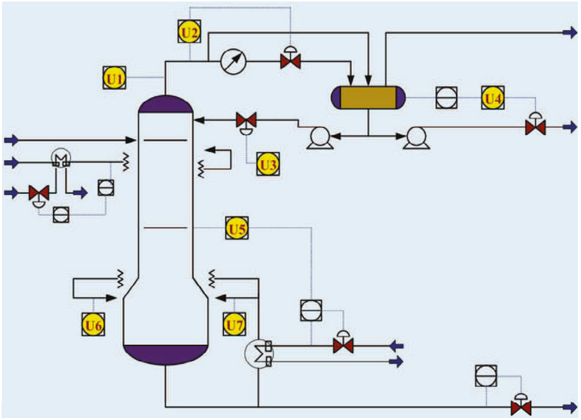


Fig. 6. Flowchart of the debutanizer column process [38].

By contrast, our method has an excellent online adaptation ability and can cope with sharp changes of the process dynamics well. Adaptive model prediction by the proposed tunable GRBF is shown in Fig. 4. Observe that the node replacement only happens when the process dynamics change; otherwise the weight adaptation is sufficient. Since the node replacement strategy of both the tunable RBF and tunable GRBF depends on the threshold, the impact of the threshold ε on online modeling accuracy is investigated in Fig. 5. It can be seen that the MSE first decreases as the threshold deceases. After reaching the minimum value, the MSE remains approximately the same as the threshold decreases further. Fig. 5 also confirms that setting the node replacement thresholds to $\varepsilon = 0.1$ and 0.01, respectively, are appropriate for the tunable RBF and our tunable GRBF.

4.2. Debutanizer column process

The debutanizer column [8,38,39] is an important part of real-world refinery process that is used for desulfuration and naphtha split. As illustrated in Fig. 6, the main task is to minimize butane content at the bottom of the column, and the butane content measurement is normally obtained by the gas chromatography with a large measurement delay. To deal with this long measurement delay and to improve the control quality of the column,

Table 2
Variable description in the debutanizer column process.

Input and output variables	Description
u_1	Top temperature
u_2	Top pressure
u_3	Reflux flow
u_4	Flow to next process
u_5	6th tray temperature
u_6	Bottom temperature A
u_7	Bottom temperature B
y	Concentration of butane

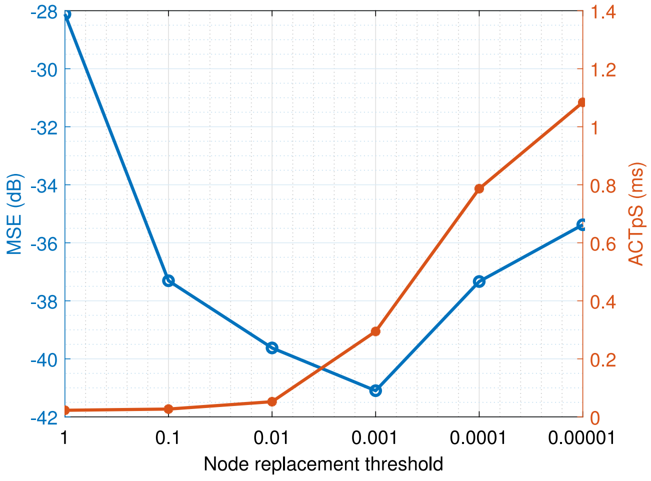


Fig. 7. The impact of node replacement threshold on the modeling accuracy and time complexity to the proposed tunable GRBF for debutanizer column process.

prediction model has to be adopted to timely estimate the concentration of butane in the bottom, which is regarded as the output y of this industrial process. Several hardware sensors, labeled as U1 to U7 in Fig. 6, are installed in the debutanizer column for obtaining the process inputs, which are denoted as u_1 to u_7. Table 2 lists the variables of this process.

Hence, at time sample t, the task of the prediction model is to make a K-step ahead prediction of y based on the available measurements of u_1 to u_7 and y at t, where $K \geq 1$. Guided by the expert knowledge and physical insight, the predictor structure can be expressed as follow [40]

$$\begin{aligned}\widehat{y}_{t+K} =& \widehat{F}\big(u_{1,t}, u_{2,t}, \ldots, u_{5,t}, (u_{6,t}+u_{7,t})/2, \\ & u_{5,t-1}, u_{5,t-2}, u_{5,t-3}, y_{t-4}, y_{t-5}, y_{t-6}; t\big), \end{aligned} \tag{27}$$

where $u_{i,t}$, $1 \leq i \leq 7$, denote the measurements of u_i at t, while the newest available measurement of y at t is y_{t-4}. A total of 2388 samples are collected from the process. The first 717 samples are used for training and the rest 1672 samples are for online prediction and model adaptation.

The window size W and bandwidth p for both the SEMLM and GAP-SER are empirically set to 35 and 5, respectively, while the threshold is set to 0.1 for the SEMLM and 0.6 for the GAP-SER. For the proposed tunable GRBF, the bandwidth is set to $p = 1$, and the node replacement threshold is a key parameter that needs to be carefully chosen. The impact of threshold on modeling accuracy and online time complexity for the debutanizer column process is shown in Fig. 7. Observe that the MSE decreases first and than increases after reaching its minimal value, while the ACTpS keeps stable and then increases dramatically when threshold $\varepsilon > 0.01$. Clearly, $\varepsilon = 0.01$ is the best choose which perfectly trades off the modeling accuracy and time complexity. For the tunable RBF, the bandwidth and threshold are set to 1 and 0.01, respectively.

Table 3
One-step prediction performance comparison of the OS-ELM, RBF, GRBF, SEMLM, GAP-SER, tunable RBF and proposed method for online modeling of debutanizer column process.

Model	Nodes/Models	MSE (dB)	ACTpS (ms)
OS-ELM	100	−23.4884	0.2892
	500	−22.4204	6.6783
RBF	10	−17.0616	0.0227
	20	−19.7042	0.0342
GRBF	10	−26.0051	0.0108
	20	−26.7762	0.0201
SEMLM	30 to 75	−33.2229	1.2101
GAP-SER	30 to 30	**−33.6665**	**0.3229**
Tunable RBF	10	**−34.2301**	**0.1804**
Proposed	10	**−39.6241**	**0.0655**

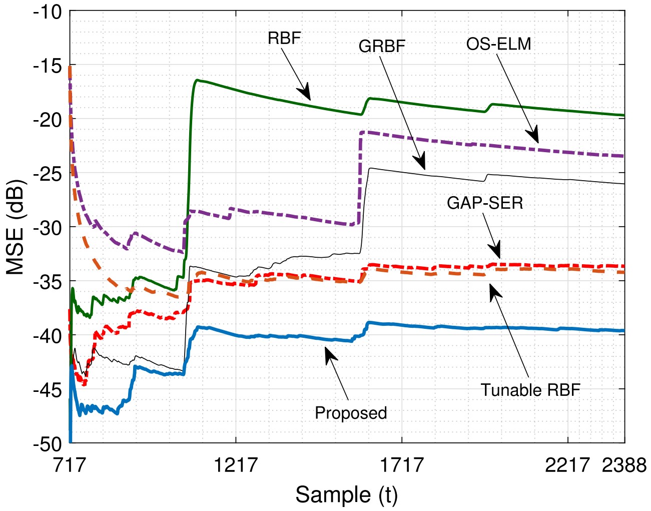
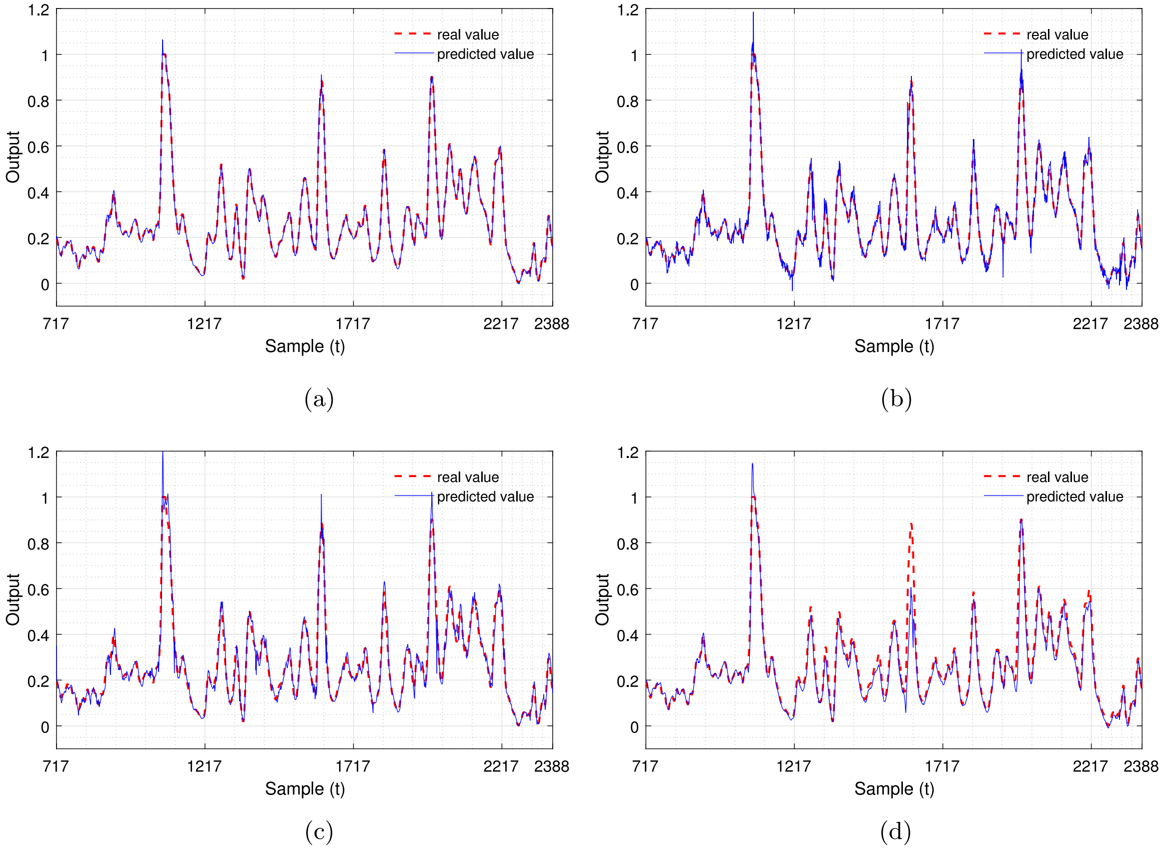


Fig. 8. MSE learning curves for online modeling of debutanizer column process by various one-step ahead prediction models. Both the RBF and GRBF have 20 hidden nodes, while the OS-ELM has 100 nodes).

The one-step ahead ($K = 1$) prediction performance comparison for various adaptive models are listed in Table 3. Similar conclusions to those observed for the synthetic nonlinear time-varying system can be drawn from the results of Table 3, which again confirm that the proposed fast tunable GRBF network attains the best online prediction accuracy, while imposing a very low online computational complexity. Only the RBF and GRBF models can achieve lower ACTpS, but their prediction accuracies are 20 dB and 13 dB worst than the proposed tunable GRBF model, respectively. Fig. 8 depicts the online MSE learning curves by various one-step prediction models. The online one-step ahead prediction $\widehat{y}_{t+1}$ values of the GRBF with 20 hidden nodes, GAP-SER, tunable RBF and our proposed tunable GRBF are compared with the actual process output observation y_{t+1} in Fig. 9. Contrast to the consistently accurate predictions produced by our fast tunable GRBF, the GRBF, GAP-SER and tunable RBF find it difficult to predict accurately when sharp and abrupt changes occur in the process output. Observe that during the online operation around $t = 1070$, the process output y_t changes sharply, possibly caused by raw materials, catalyst deactivation or external disturbances. The GRBF predictor experiences considerable increases in the prediction errors for the samples around $t = 1070$, as can be clearly seen from Fig. 9(d). This causes the first big jump in the MSE for the GRBF around $t = 1070$ shown in Fig. 8. When the second big change in y_t occurs around $t = 1210$, significant increases in the prediction errors for the samples around $t =$



(a)

(b)

(c)

(d)

Fig. 9. Online one-step modeling of debutanizer column process: (a) Proposed, (b) GAP-SER, (c) Tunable RBF, and (d) GRBF with 20 hidden nodes.

1210 are again observed in Fig. 9(d), and the second big jump in the MSE occurs in Fig. 8 for the GRBF predictor. When the third big change in y_t occurs, however, the predictor has seen it before, since the third big change in y_t is basically very similar to the second big change in y_t. Consequently, the response of the GRBF to the third big change in y_t is much better, as can be seen from Fig. 9(d), and the increase in the MSE is much smaller, as can be seen from Fig. 8.

We further investigate the impact of model size for the RBF, GRBF, tunable RBF and proposed tunable GRBF methods. It can be seen from Fig. 10 that the online prediction MSEs of the tunable RBF and GRBF decrease dramatically as the number of hidden nodes increases before reaching 6 hidden nodes. Then the prediction MSEs of the both online models remain approximately unchanged as the model size increases further. The performance of the RBF model with online weight updating only remains poor as the model size increases.

All the online prediction models considered can easily be extended to be multi-step-ahead predictors, and the K-step ahead prediction performance of the GAP-SER, the tunable RBF and the proposed tunable GRBF are compared in Fig. 11. The algorithmic parameters of these three online K-step predictors are the same as their one-step counterparts. It can be seen that the online prediction performance of the GAP-SER varies dramatically as the prediction step increases. By contrast the performance of the tunable RBF and proposed tunable GRBF remain consistent across the wide range of the prediction step. As expected, our proposed approach attains the best multi-step ahead online prediction accuracy.

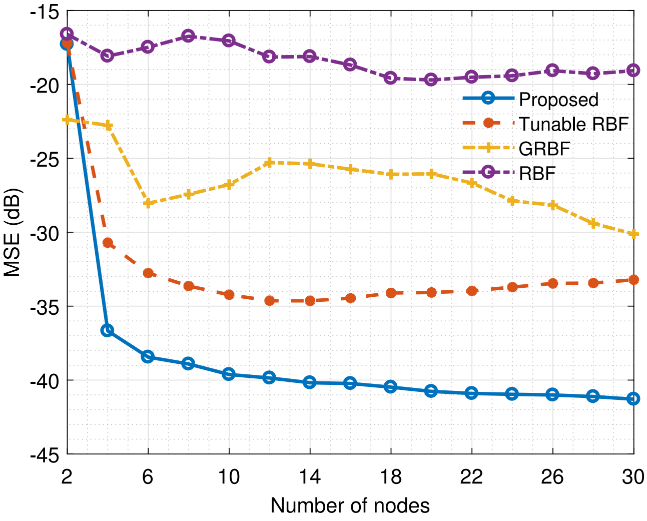


Fig. 10. Impact of model size on online modeling debutanizer column process by the RBF, GRBF, tunable RBF and proposed tunable GRBF networks.

4.3. Sulfur recovery unit

The SRU is used to remove environment pollutions from the acid gas streams [40]. A simplified block diagram of SRU is depicted in Fig. 12. Two main acid gases, the MEA gas that is rich in H_2S and the SWS gas that is rich in NH_3, enter the SRU. H_2S is transformed into pure sulfur by a partial oxidation with air. The nonconverted gas is fed to the Maxisulfur plant in the final conversion phase. The tail gas from the SRU contains residual H_2S,

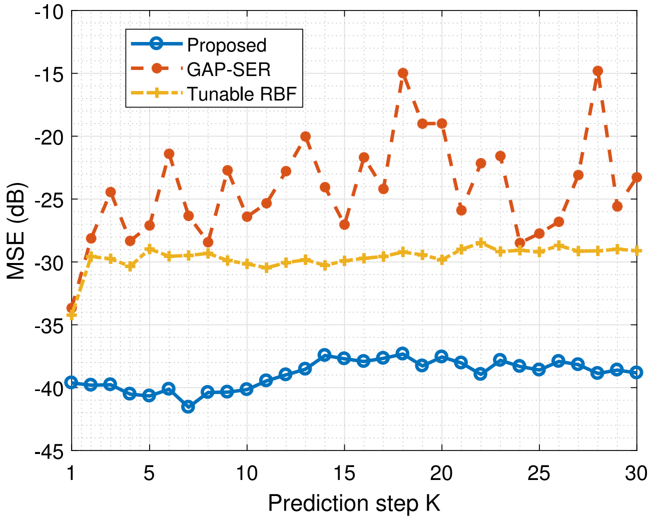


Fig. 11. Multi-step prediction performance of online modeling debutanizer column process by the tunable RBF, GAP-SER and proposed tunable GRBF.

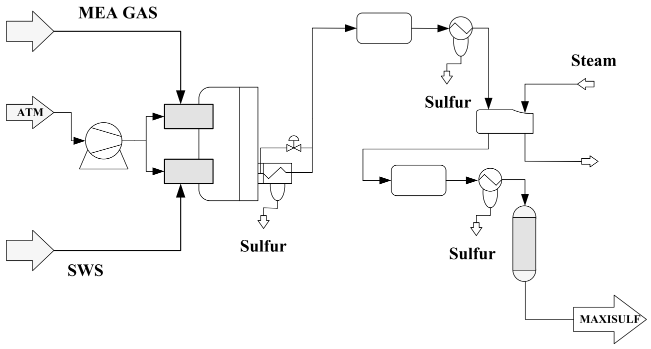


Fig. 12. The block diagram of SRU process.

Table 4
Variable description in the SRU process.

Input and output variables	Description
u_1	MEA gas flow
u_2	First air flow
u_3	Second air flow
u_4	Gas flow in SWS zone
u_5	Air flow in SWS zone
y	Concentration of H_2S

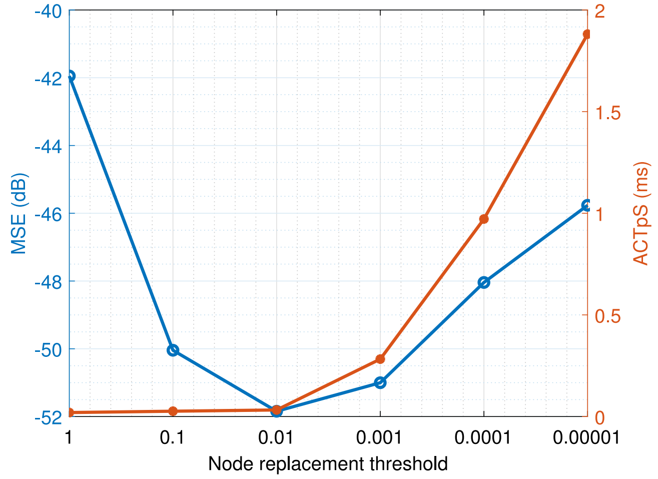


Fig. 13. The impact of node replacement threshold on the modeling accuracy and time complexity for sulfur recovery unit process given the prediction model framework of (28).

Table 5
Performance comparison of the OS-ELM, RBF, GRBF, SEMLM, GAP-SER, tunable RBF and proposed method for online modeling of sulfur recovery unit process given the prediction model framework of (28).

Model	Nodes/Models	MSE (dB)	ACTpS (ms)
OS-ELM	100 500	−23.5819 −25.9830	0.2968 6.6832
RBF	10 20	−27.8135 −26.8619	0.0277 0.0328
GRBF	10 20	−35.8132 **−40.2139**	0.0102 **0.0178**
SEMLM	33 to 44	−39.9077	0.5623
GAP-SER	51 to 51	−40.1138	0.3421
Tunable RBF	10	**−39.1152**	**0.0707**
Proposed	10	**−51.8437**	**0.0369**

which is harmful to the atmosphere and human health. Therefore, it is vital to monitor its concentration online and to make sure that the concentration is below the required value before it can be released into the atmosphere. Five process variables and concentration of H_2S tabulated in Table 4 are considered as the process inputs and output, respectively. Total 3000 samples are collected from the SRU data set, among which the first 1000 samples and the rest of the data samples are used for initial training and online prediction and model adaptation, respectively.

The measurement of acid gas H_2S has delay. Furthermore, acid gas H_2S often damages the sensors through corrosion, and hardware instruments are frequently removed for maintenance. Therefore, for online prediction of y, it is preferred not relying on the past measurements of y. A such online prediction model structure is given as [40]

$$\widehat{y}_{t+1} = \widehat{F}\big(u_{1,t}, u_{1,t-5}, u_{1,t-7}, u_{1,t-9}, u_{2,t}, u_{2,t-5}, u_{2,t-7}, u_{2,t-9}, \ldots, u_{5,t}, u_{5,t-5}, u_{5,t-7}, u_{5,t-9}; t\big), \qquad (28)$$

where $u_{i,t}$, $1 \le i \le 5$, are the measurements of the inputs u_i at sample t, and $\widehat{y}_{t+1}$ denotes the prediction of y at the next sample $t+1$. The window size W and bandwidth p for both the SEMLM and GAP-SER are empirically tuned to 40 and 2, respectively, while the threshold is set to 0.1 for the SEMLM and 0.8 for the GAP-SER. For the proposed tunable GRBF, the bandwidth is set to 1 and the impact of node replacement threshold is shown in Fig. 13. Similar tendency can be found with previous simulation, and we set threshold $\varepsilon = 0.01$. The bandwidth and threshold for tunable RBF are set to 1 and 0.01, respectively. Note that even with the model structure of (28), we still need the past output information (output gradients) in the training data set to construct hidden nodes of a GRBF network. By contrast, with the model structure of (28), a RBF model does not require any output information in making prediction.

The performance of various adaptive models under the prediction framework (28) are tabulated in Table 5. Our fast tunable GRBF model with 10 hidden nodes achieves the best prediction accuracy, which is almost 10 dB better than the second-best GRBF network with 20 hidden nodes, while imposing the third lowest ACTpS of 0.0369 ms. The GRBF network with only weight updating not only achieves the second best prediction accuracy but also

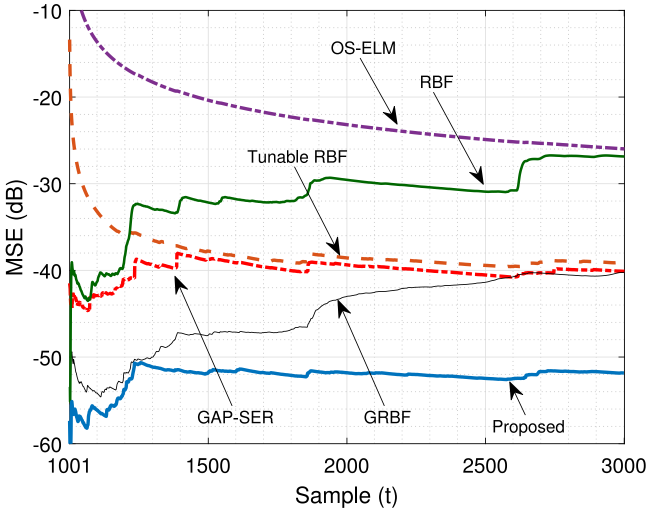


Fig. 14. MSE learning curves for online modeling of SRU by various adaptive models given the prediction model framework of (28). The RBF and GRBF have 10 and 20 hidden nodes, respectively, while the OS-ELM has 500 nodes.

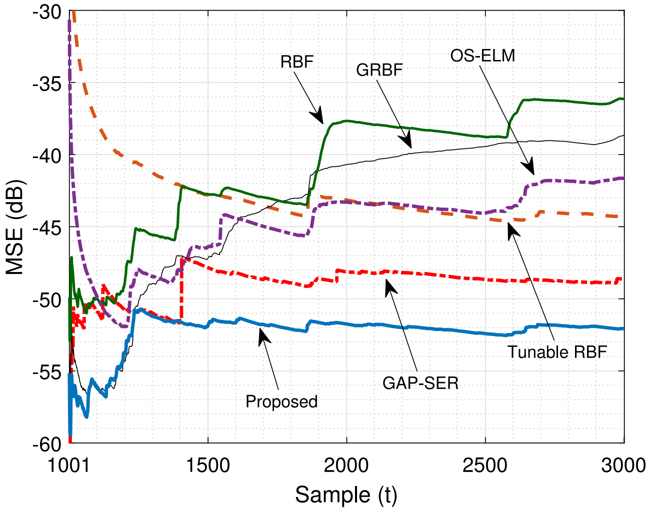


Fig. 15. MSE learning curves for online modeling of SRU by various adaptive models given the prediction model framework of (29). The RBF and GRBF have 20 20 hidden nodes, while the OS-ELM has 1000 nodes.

Table 6
Performance comparison of the OS-ELM, RBF, GRBF, SEMLM, GAP-SER, tunable RBF and proposed method for online modeling of sulfur recovery unit process given the prediction model framework of (29).

Model	Nodes/Models	MSE (dB)	ACTpS (ms)
OS-ELM	100 500	−41.6397 −39.5544	0.2752 6.6191
RBF	10 20	−32.3621 −36.1458	0.0216 0.0321
GRBF	10 20	−35.6760 −38.6540	0.0102 0.0174
SEMLM	51 to 93	−47.4914	1.5851
GAP-SER	51 to 51	**−48.5733**	**0.5231**
Tunable RBF	10	**−44.2916**	**0.0576**
Proposed	10	**−52.0765**	**0.0365**

has the lowest ACTpS. The online MSE leaning curves of the OS-ELM with 500 nodes, the RBF with 10 nodes, the GRBF with 20 nodes, the GAP-SER, the tunable RBF and the tunable GRBF are compared in Fig. 14. It can be seen that GRBF-type models significantly outperform their classic RBF-type counterparts. The reason is of course that a GRBF model is provided with more information, specifically, the past output information (output gradient), which a RBF type model does not have. Thus the comparison of Table 5 is unfair to RBF-type models.

If the past output measurements are available we may include them in the prediction model to improve the achievable performance. For a fairer comparison, for example, we may consider the prediction model structure of

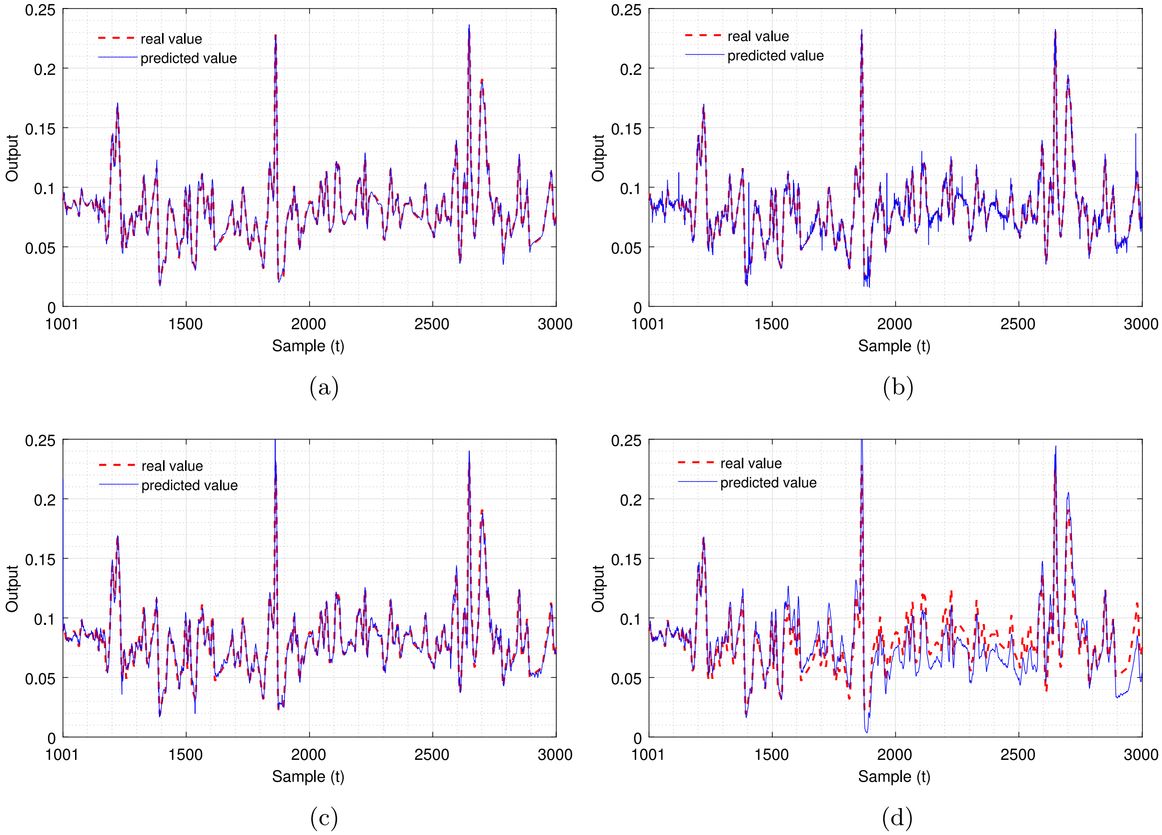
$$\widehat{y}_{t+1} = \widehat{F}\big(u_{1,t}, u_{1,t-5}, u_{1,t-7}, u_{1,t-9}, u_{2,t}, u_{2,t-5}, u_{2,t-7}, u_{2,t-9}, \ldots, u_{5,t}, u_{5,t-5}, u_{5,t-7}, u_{5,t-9}, y_{t-1}, y_{t-2}, y_{t-3}; t\big), \tag{29}$$

Table 6 compares the performance of various online adaptive models under the prediction model structure (29). Again, our tunable GRBF achieves the best online prediction accuracy while imposing the third lowest ACTpS. Observe however that the performance of the OS-ELM, RBF, SEMLM, GAP-SER and tunable RBF are significantly better than their respective counterparts based on the prediction structure of (28). This simply confirms the well-known fact that the past output information help predicting the future output. By contrast, the performance of the GRBF and tunable GRBF given in Table 6 are very similar to those of Table 5, because the information provided to a GRBF-type model under (29) is similar to that under (28). The online MSE learning curves of various adaptive models are compared in Fig. 15. The prediction outputs by the four adaptive models are compared with the real process output in Fig. 16. Observe that both the GAP-SER and tunable RBF have difficulty to track rate and gradient sign changes. By contrast, our proposed tunable GRBF is inherently immune to this difficulty.

5. Conclusions

In this paper, we have proposed an adaptive tunable GRBF network for online modeling and prediction of nonlinear time-varying processes. First, we have provided a novel GRBF network structure, where each hidden node encodes a local process state and has a geometric property of a local predictor of the process output. During initial training, a small GRBF model can readily be constructed using the OLS algorithm. Second, we have proposed a new adaptive mechanism to online update the GRBF network during real-time operation by replacing the least significant or most-out-of-date hidden node with a new hidden node that automatically encodes the newly emerged process state and acts a perfect local predictor of the current process output. Owing to the geometric property of GRBF hidden node, this optimal hidden node replacement is straightforward and imposes little online computational complexity. Extensive experiments involving a synthetic nonlinear time-varying system and two real-world chemical processes have demonstrated that our proposed fast tunable GRBF network significantly outperforms the existing state-of-art online modeling methods for real-time tracking changes in process dynamics. Notably, our method achieves excellent online prediction accuracy, while imposing very low real-time computational complexity.

In addition to accurate prediction accuracy and fast adaptation capability with very low online computational complexity, our proposed tunable GRBF method maintains a small fixed-size network structure. This makes it particularly attractive for implementation on embedded devices for adaptive control system applications.



(a)

(b)

(c)

(d)

Fig. 16. Online one-step modeling of SRU with prediction model framework (29): (a) Proposed, (b) GAP-SER, (c) Tunable RBF, and (d) GRBF with 20 hidden nodes.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

T. Liu would like to thank the sponsorship of Chinese Scholarship Council for funding his research at School of Electronics and Computer Science, University of Southampton, UK. This work was partly supported by the National Natural Science Foundation of China under grant 61771077, the Key Research Program of Chongqing Science & Technology Commission under grant CSTC2017jcyjBX0025, Graduate Scientific Research and Innovation Foundation of Chongqing, China under grant CYB19072.

References

[1] Z. Zhou, N.V. Chawla, Y. Jin, G.J. Williams, Big data ortunities and challenges: Discussions from data analytics perspectives, IEEE Comput. Intell. Mag. 9 (4) (2014) 62–74.

[2] G. Ditzler, M. Roveri, C. Ali, R. Polikar, Learning in nonstationary environments: A survey, IEEE Comput. Intell. Mag. 10 (4) (2015) 12–25.

[3] Z. Ge, Z. Song, S. Ding, B. Huang, Data mining and analytics in the process industry: The role of machine learning, IEEE Access 5 (2017) 20590–20616.

[4] M. Kano, M. Ogawa, The state of the art in chemical process control in Japan: Good practice and questionnaire survey, J. Process Control 20 (9) (2010) 969–982.

[5] W. Shao, S. Chen, C.J. Harris, Adaptive soft sensor development for multi-output industrial processes based on selective ensemble learning, IEEE Access 6 (2018) 55628–55642.

[6] W. Shao, Z. Ge, Z. Song, Soft-sensor development for processes with multiple operating modes based on semisupervised Gaussian mixture regression, IEEE Trans. Control Syst. Technol. 27 (5) (2019) 2169–2181.

[7] W. Shao, Z. Ge, Z. Song, J. Wang, Semi-supervised robust modeling of multimode industrial processes for quality variable prediction based on student's t mixture model, IEEE Trans. Ind. Inf. 16 (5) (2020) 2965–2976.

[8] X. Yuan, L. Li, Y. Wang, Nonlinear dynamic soft sensor modeling with supervised long short-term memory network, IEEE Trans. Ind. Inf. 16 (5) (2020) 3168–3175.

[9] H. Jin, et al., Dual learning-based online ensemble regression aoach for adaptive soft sensor modeling of nonlinear time-varying processes, Chemometr. Intell. Lab. Syst. 151 (2016) 228–244.

[10] J. Liu, D.-S. Chen, Nonstationary fault detection and diagnosis for multimode processes, AIChE J. 56 (1) (2010) 207–219.

[11] Y. Zhang, T. Chai, Z. Li, C. Yang, Modeling and monitoring of dynamic processes, IEEE Trans. Neural Netw. Learn. Syst. 23 (2) (2012) 277–284.

[12] H. Ning, G. Qing, T. Tian, X. Jing, Online identification of nonlinear stochastic spatiotemporal system with multiplicative noise by robust optimal control-based kernel learning method, IEEE Trans. Neural Netw. Learn. Syst. 30 (2) (2019) 389–404.

[13] J.L. Lobo, et al., Evolving spiking neural networks for online learning over drifting data streams, Neural Netw. 108 (2018) 1–19.

[14] H. Chen, Y. Gong, X. Hong, A new adaptive multiple modelling aoach for non-linear and non-stationary systems, Int. J. Syst. Sci. 47 (9) (2016) 2100–2110.

[15] Y. Li, H.-L. Wei, S.A. Billings, P.G. Sarrigiannis, Identification of nonlinear time-varying systems using an online sliding-window and common model structure selection (CMSS) aoach with aications to EEG, Int. J. Syst. Sci. 47 (11) (2016) 2671–2681.

[16] T. Liu, S. Liang, S. Chen, C.J. Harris, A multiple local model learning for nonlinear and time-varying microwave heating process, in: Proc. IJCNN 2019 (Budapest, Hungary), Jul. 14-19, 2019, pp. 1–8.

[17] T. Liu, S. Chen, S. Liang, C.J. Harris, Selective ensemble of multiple local model learning for nonlinear and nonstationary systems, Neurocomputing 378 (2019) 98–111.

[18] T. Liu, S. Chen, S. Liang, C.J. Harris, Growing and pruning selective ensemble regression for nonlinear and nonstationary systems, IEEE Access 8 (2020) 73278–73292.

[19] S. Chen, S.A. Billings, W. Luo, Orthogonal least squares methods and their aication to non-linear system identification, Internat. J. Control 50 (5) (1989) 1873–1896.

[20] S. Chen, C.F.N. Cowan, P.M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, IEEE Trans. Neural Netw. 2 (2) (1991) 302–309.

[21] S. Chen, X. Hong, C.J. Harris, P.M. Sharkey, Sparse modelling using orthogonal forward regression with PRESS statistic and regularization, IEEE Trans. Syst. Man Cybern. B 34 (2) (2004) 898–911.
[22] X. Hong, et al., Model selection aoaches for non-linear system identification: A review, Int. J. Syst. Sci. 39 (10) (2008) 925–946.
[23] S. Chen, Nonlinear time series modelling and prediction using Gaussian RBF networks with enhanced clustering and RLS learning, Electron. Lett. 31 (2) (1995) 117–118.
[24] S. Chen, S. Billings, Recursive prediction error parameter estimator for non-linear models, Internat. J. Control 49 (2) (1989) 569–594.
[25] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: Theory and aications, Neurocomputing 70 (1–3) (2006) 489–501.
[26] N. Liang, G.B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Netw. 17 (6) (2006) 1411–1423.
[27] X. Wang, M. Han, Online sequential extreme learning machine with kernels for nonstationary time series prediction, Neurocomputing 145 (2014) 90–97.
[28] X. Wang, M. Han, Improved extreme learning machine for multivariate time series online sequential prediction, Eng. Appl. Artif. Intell. 40 (2015) 28–36.
[29] J. Platt, A resource-allocating network for function interpolation, Neural Comput. 3 (2) (1991) 213–225.
[30] V. Kadirkamanathan, M. Niranjan, A function estimation aoach to sequential learning with neural networks, Neural Comput. 5 (6) (1993) 954–975.
[31] H. Chen, Y. Gong, X. Hong, S. Chen, A fast adaptive tunable RBF network for nonstationary systems, IEEE Trans. Cybern. 46 (12) (2016) 2683–2692.
[32] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, G.M. Ljung, Time Series Analysis: Forecasting and Control, fifth ed., Wiley, Hoboken, NJ, 2015.
[33] E.S. Chng, S. Chen, B. Mulgrew, Gradient radial basis function networks for nonlinear and nonstationary time series prediction, IEEE Trans. Neural Netw. 7 (1) (1996) 190–194.
[34] H. Chen, Y. Gong, X. Hong, Online modeling with tunable RBF network, IEEE Trans. Cybern. 43 (3) (2013) 935–947.
[35] S. Chen, S.A. Billings, Representation of non-linear systems: The NARMAX model, Internat. J. Control 49 (3) (1989) 1013–1032.
[36] S. Chen, S.A. Billings, C.F.N. Cowan, P.M. Grant, Practical identification of NARMAX models using radial basis functions, Internat. J. Control 52 (6) (1990) 1327–1350.
[37] D. Lowe, Adaptive radial basis function nonlinearities, and the problem of generalisation, in: Proc. 1st IEE Int. Conf. Artificial Neural Networks (London, UK), Oct. 16-18, 1989, pp. 171–175.
[38] W. Shao, X. Tian, P. Wang, X. Deng, S. Chen, Online soft sensor design using local partial least squares models with adaptive process state partition, Chemometr. Intell. Lab. Syst. 144 (15) (2015) 108–121.
[39] X. Yuan, et al., Deep learning-based feature representation and its aication for soft sensor modeling with variable-wise weighted SAE, IEEE Trans. Ind. Inf. 14 (7) (2018) 3235–3243.
[40] L. Fortuna, S. Graziani, A. Rizzo, M.G. Xibilia, Soft Sensors for Monitoring and Control of Industrial Processes, Springer-Verlag, London, 2007.
[41] H. Han, W. Lu, Y. Hou, J. Qiao, An adaptive-PSO-based self-organizing RBF neural network, IEEE Trans. Neural Netw. Learn. Syst. 29 (1) (2018) 104–117.