Multi-Output Selective Ensemble Identification of Nonlinear and Nonstationary Industrial Processes

Tong Liu, Sheng Chen, *Fellow, IEEE*, Shan Liang, *Member, IEEE*, Shaojun Gan, and Chris J. Harris

***Abstract*— A key characteristic of biological systems is the ability to update the memory by learning new knowledge and removing out-of-date knowledge so that intelligent decision can be made based on the relevant knowledge acquired in the memory. Inspired by this fundamental biological principle, this article proposes a multi-output selective ensemble regression (SER) for online identification of multi-output nonlinear time-varying industrial processes. Specifically, an adaptive local learning approach is developed to automatically identify and encode a newly emerging process state by fitting a local multi-output linear model based on the multi-output hypothesis testing. This growth strategy ensures a highly diverse and independent local model set. The online modeling is constructed as a multi-output SER predictor by optimizing the combining weights of the selected local multi-output models based on a probability metric. An effective pruning strategy is also developed to remove the unwanted out-of-date local multi-output linear models in order to achieve low online computational complexity without scarifying the prediction accuracy. A simulated two-output process and two real-world identification problems are used to demonstrate the effectiveness of the proposed multi-output SER over a range of benchmark schemes for real-time identification of multi-output nonlinear and nonstationary processes, in terms of both online identification accuracy and computational complexity.**

***Index Terms*— Adaptive local learning, multi-output nonlinear time-varying industrial processes, multivariate statistic hypothesis testing, pruning, selective ensemble.**

I. Introduction

GENERALLY, a biological system has two types of evolution or adaptation. The first type of evolution or long-term adaptation is over many generations of a biological system, based on the famous survival of the fittest principle. Many artificial learning algorithms mimic this long-term evolution principle, including genetic algorithm [1] and differential evolutionary algorithm [2]. A biological system also experiences the second type of adaptation or short-term evolving during its “daily life.” In its existence, a biological system must evolve or adapt to fast-changing environment and, therefore, it constantly updates its memory or “brain” by learning new knowledge and removing out-of-date knowledge in order to make an intelligent decision based on the relevant knowledge stored in its memory. Inspired by this fundamental short-term biological adaptation principle, in this work, we design a multi-output selective ensemble regression (SER) for online identification of multi-output nonlinear time-varying industrial processes.

Many real-life processes exhibit inherently nonlinear and nonstationary dynamic behaviors [3]–[9], where the underlying data-generating mechanisms are fast-changing over time. Under such circumstances, a predictive model must have the flexibility of adapting its structure and parameters from the fast-arriving nonstationary data stream in order to maintain its performance in the changing environment.

Construction of neural network models can be interpreted as learning and encoding the nonlinear dynamic information of the system presented in the training data in their hidden-layer nodes. For example, each hidden node of the radial basis function (RBF) network encodes a process state. When the system is nonstationary, it is necessary for a model to acquire the newly emerging process state by updating its structure. A classical approach for estimating and tracking the temporal variations of a nonlinear process is to employ adaptive algorithms, such as recursive least square (RLS) [10], [11] and its more recent variant multi-innovation RLS (MRLS) [12]. In particular, if the process’s input space is known a prior, by covering the input space with sufficiently dense nodes, the online sequential extreme learning machine (OS-ELM) only needs to update model weights online using the RLS algorithm [13]–[15]. Because the size of OS-ELM has to be very large to cover the training data space, online adaptation of the model weights is computationally costly and, moreover, there is no guarantee that fixed hidden nodes, no matter how dense they are, will also cover the changing nonstationary

Manuscript received February 20, 2020; revised July 22, 2020 and September 25, 2020; accepted September 26, 2020. Date of publication October 14, 2020; date of current version May 3, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61771077, in part by the Key Research Program of Chongqing Science and Technology under Grant CSTC2017jcyjBX0025, and in part by the Graduate Scientific Research and Innovation Foundation of Chongqing, China, under Grant CYB19072. The work of Tong Liu was supported by the Chinese Scholarship Council for his research at the School of Electronics and Computer Science, University of Southampton, U.K. *(Corresponding authors: Tong Liu; Shan Liang.)*

Tong Liu and Shan Liang are with the Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing 400044, China, and also with the School of Automation, Chongqing University, Chongqing 400044, China (e-mail: liutong42@cqu.edu.cn; lightsun@cqu.edu.cn).

Sheng Chen is with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K., and also with King Abdulaziz University, Jeddah 21589, Saudi Arabia (e-mail: sqc@ecs.soton.ac.uk).

Shaojun Gan is with the Beijing Key Laboratory of Traffic Engineering, College of Metropolitan Transportation, Beijing University of Technology, Beijing 100124, China (e-mail: s.gan@bjut.edu.cn).

Chris J. Harris is with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: chrisharris57@msn.com).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.3027701

data space well. Hence, the OS-ELM is unsuitable for fast time-varying data.

An alternative to single-global nonlinear modeling approach is the ensemble learning that employs multiple models to separately model the data space. The online ensemble learning (OEL) has attracted considerable attention recently, but most of the researches focus on online classification [16]–[23] and the regression problem is rarely discussed. The development of OEL for nonstationary systems is a challenging task, as the ensemble learner must not only maintain highly diverse models to cover a wide range of the observed data subspaces but also learn the new concept as fast as possible to timely capture the changing dynamics. The ensemble of OS-ELM (EOS-ELM) [24], however, does not have this capability, as all the base ELM models are trained on the same data set, and they are not updated online.

To design better OEL models, we take the inspiration from the aforementioned fundamental learning principle of biological systems, namely, the ability to update the memory by learning new knowledge and removing out-of-date knowledge so that intelligent decision can be made based on the most relevant knowledge acquired in the memory. For single-output nonlinear and nonstationary processes, the recently proposed selective ensemble-based multiple local model (SEMLM) learning [25] enables automatically identifying newly emerging process states online and combining the most up-to-date local linear models to make an accurate SER-based prediction. The SEMLM, however, does not have the ability to remove out-of-date process states that are no longer needed. Hence, a potential problem associated with the SEMLM is that for fast time-varying systems, over a long period of online adaptation, the local model set may grow to be very large, which may impose high online computational complexity. To overcome this drawback and in particular to fully consider the aforementioned short-term evolving principle of the biological learning system, the work [26] further proposed a growing and pruning SER (GAP-SER) for nonlinear and nonstationary data modeling. The GAP-SER can not only learn the newly emerging concept but also forget the past accumulated old concepts that are no longer relevant, hence balancing well the so-called stability-plasticity dilemma of a learning system. The experimental results of [26] show that the GAP-SER typically outperforms the SEMLM, in terms of both online prediction accuracy and computational complexity.

Although the GAP-SER achieves great success in online modeling of nonstationary data, it is suitable only for single-output nonlinear and nonstationary systems. This is because the key components of the GAP-SER, including its local model growing and pruning strategies and selective ensemble prediction, are restricted to single-output modeling, and they are not applicable to multi-output modeling. Practical systems and processes often contain multiple manipulated variables and controlled variables that exhibit strong coupling. Although we may apply the GAP-SER to identify the multiple single-output models for a multi-output system, this will increase the modeling effort considerably and more importantly, and it will lead to the degradation in achievable online prediction accuracy. The latter is owing to the fact that the multiple single-output models ignore the correlation or coupling effects of the different output variables.

Therefore, it is necessary to investigate efficient identification techniques for multi-output nonlinear and nonstationary systems. The online multi-output regression problem remains largely understudied. To the best of our knowledge, most of the existing online modeling approaches are restricted to single-output systems. For example, like the GAP-SER, the methods of [27]–[29] are restricted to single-output systems. Although a few studies have developed predictive models for multi-output nonlinear systems, such as the support vector regression (SVR) [30], the ELM [31] and the orthogonal LSs (OLSs) method [32]–[35], these models are designed specifically for stationary systems, and they cannot directly be applied to highly nonstationary systems. In the online soft sensor design, developing multi-output soft sensor to predict multiple primary variables has been demonstrated to be vital to achieve better performance than building multiple single-output soft sensors [36]. This motivates our current work.

In this article, we propose a multi-output SER-based evolving model for online identification of multi-output nonlinear and time-varying processes, which is inspired by the fundamental evolving principle of biological systems for coping with a fast-changing environment. As aforementioned, the short-term evolving of a biological system involves three levels of adaptation: adaptively acquiring new knowledge, adaptively making an intelligent decision based on the most relevant knowledge stored in the memory, and adaptively removing the out-of-date knowledge which are no longer valid from memory. Our multi-output SER-based evolving system also involves these three levels of adaptation. At the level of acquiring new knowledge, the newly emerging process state is automatically identified, and a multi-output local linear model is fitted to it. Note that the growth strategy of [26] cannot be applied to the multi-output system as it is based on the single-output statistical testing. We develop a new adaptive local learning approach with the appropriate statistics for multi-output hypothesis testing in order to construct a highly diverse and independent multi-output local model set. At the level of making an intelligent decision or online prediction, we construct a multi-output SER predictor based on a probability metric by optimizing the corresponding combining weights of the selective ensemble of the local multi-output linear models. At the level of removing out-of-date knowledge, a pruning strategy is developed to remove reliably the old local multi-output linear models that are no longer needed in order to further reduce the online computational complexity without degrading the prediction accuracy.

The main contribution of this work, therefore, is to develop a highly efficient and accurate online evolving model for multi-output selective ensemble identification of nonlinear and time-varying processes, with the completed design of online adaptive growing and pruning strategies and SER based adaptive prediction modeling. This new multi-output SER evolving model fully complies with the aforementioned biological system learning principle. Three case studies, a simulated two-output nonlinear system, a nonisothermal continuous stirred-tank reactor (CSTR) process [37], [38] and a real-world microwave heating system [39], are used to demonstrate the superior performance of our proposed multi-output SER over a range of benchmark schemes, in terms of both real-time prediction accuracy and online computational complexity.

II. Proposed Method

To achieve high online prediction accuracy while imposing low computational complexity, our proposed evolving model mimics the fundamental learning principle from biological systems in coping with a fast-changing environment, namely, the ability to update memory by learning new knowledge, to make an intelligent decision based on the most relevant knowledge in the memory, and to remove out-of-date knowledge from memory. Consequently, it involves three levels of adaptation, namely, adaptive local learning that grows the multi-output local linear model set by identifying the newly emerged process state, adaptive online prediction by a selective ensemble of the most relevant local models from memory, and adaptive removal of the most out-of-date local linear models in the memory to free space for acquiring new knowledge. We now detail these levels of adaptation or the three components of our proposed evolving model.

A. Adaptive Local Learning via Multivariate Statistic

At the first level of adaptation, the evolving model absorbs new information from the fast-arriving data stream, and this is achieved by an adaptive local learning strategy which automatically encodes the newly emerging process states. Consider a nonlinear time-varying process with m-dimensional input $\boldsymbol{x}(t) \in \mathbb{R}^m$ and p-dimensional output $\boldsymbol{y}(t) \in \mathbb{R}^p$. The task of local learning is to establish the local experts $\{\boldsymbol{f}_l\}_{l=1}^L$ that accurately model the process's L local states or local regions, represented by the subdata sets $\{\boldsymbol{X}_l, \boldsymbol{Y}_l\}_{l=1}^L$, where $\boldsymbol{f}_l$ are linear models. The local linear model set $\{\boldsymbol{f}_l\}_{l=1}^L$ correspond to the process states observed over time and, therefore, they represent the knowledge of the process acquired and stored in the evolving model's memory.

The basic idea of adaptive local learning is as follows. Let a local window $\mathcal{W}_{\text{ini}} = \{\boldsymbol{X}_{\text{ini}} \in \mathbb{R}^{W_G \times m}, \boldsymbol{Y}_{\text{ini}} \in \mathbb{R}^{W_G \times p}\}$ with W_G consecutive samples $\{\boldsymbol{x}(t), \boldsymbol{y}(t)\}_{t=t_{\text{ini}}}^{t_{\text{ini}}+W_G}$ be initially set. A multi-output local linear model $\boldsymbol{f}_{\text{ini}}$ is built on it as

$$\widehat{\boldsymbol{Y}}_{\text{ini}} = \boldsymbol{f}_{\text{ini}}(\boldsymbol{X}_{\text{ini}}) = \boldsymbol{\Phi}\boldsymbol{\Omega} \tag{1}$$

where $\boldsymbol{\Phi} = [\boldsymbol{1}_{W_G} \;\; \boldsymbol{X}_{\text{ini}}] \in \mathbb{R}^{W_G \times (1+m)}$, $\boldsymbol{1}_{W_G}$ is the W_G-dimensional vector whose elements are all one, and the model parameter matrix $\boldsymbol{\Omega} \in \mathbb{R}^{(1+m)\times p}$ is given by the LSs estimate as

$$\boldsymbol{\Omega} = \left(\boldsymbol{\Phi}^{\text{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\text{T}}\boldsymbol{Y}_{\text{ini}}. \tag{2}$$

In the challenging situation of the singular input observation matrix $\boldsymbol{\Phi}$ and multicollinearity among the inputs, we may change the LS estimate to the regularized LS estimate

$$\boldsymbol{\Omega} = \left(\boldsymbol{\Phi}^{\text{T}}\boldsymbol{\Phi} + \lambda \boldsymbol{I}_{m+1}\right)^{-1}\boldsymbol{\Phi}^{\text{T}}\boldsymbol{Y}_{\text{ini}}$$

where the regularization parameter λ is a very small positive number, e.g., $\lambda = 10^{-6}$ and $\boldsymbol{I}_n$ is the $n \times n$ identity matrix. The predicted error or residual matrix of this local model is

$$\boldsymbol{\Upsilon}_{\text{ini}} = \boldsymbol{Y}_{\text{ini}} - \boldsymbol{f}_{\text{ini}}(\boldsymbol{X}_{\text{ini}}) \in \mathbb{R}^{W_G \times p}. \tag{3}$$

By shifting the data window one sample ahead, a new window $\mathcal{W}_{\text{sft}} = \{\boldsymbol{X}_{\text{sft}}, \boldsymbol{Y}_{\text{sft}}\}$ is obtained, which contains the samples $\{\boldsymbol{x}(t), \boldsymbol{y}(t)\}_{t=t_{\text{ini}}+1}^{t_{\text{ini}}+1+W_G}$. If the two local data regions $\mathcal{W}_{\text{ini}}$ and $\mathcal{W}_{\text{sft}}$ are not significantly different, it can be considered that the data within $\mathcal{W}_{\text{sft}}$ follow the same distribution as in $\mathcal{W}_{\text{ini}}$ and the window continues to be shifted forward. Otherwise, $\mathcal{W}_{\text{sft}}$ is considered to represent a new process state different from the one for $\mathcal{W}_{\text{ini}}$, and a new local linear model $\boldsymbol{f}_{\text{new}}$ should be developed based on $\mathcal{W}_{\text{sft}}$. Let the estimation error matrix produced by $\boldsymbol{f}_{\text{ini}}$ on $\mathcal{W}_{\text{sft}}$ be denoted as

$$\boldsymbol{\Upsilon}_{\text{sft}} = \boldsymbol{Y}_{\text{sft}} - \boldsymbol{f}_{\text{ini}}(\boldsymbol{X}_{\text{sft}}) \in \mathbb{R}^{W_G \times p}. \tag{4}$$

Whether the two local data regions $\mathcal{W}_{\text{ini}}$ and $\mathcal{W}_{\text{sft}}$ are similar or not can then be turned into the equivalent testing that tests whether $\boldsymbol{\Upsilon}_{\text{ini}}$ and $\boldsymbol{\Upsilon}_{\text{sft}}$ are significantly different or not. Since $\boldsymbol{f}_{\text{ini}}$ is a p-output linear model, $\boldsymbol{\Upsilon}_{\text{ini}}$ and $\boldsymbol{\Upsilon}_{\text{sft}}$ are considered not significantly different when both their mean vectors and covariance matrices are the same. Accordingly, two null hypotheses $H_0^{\boldsymbol{\mu}}$ and $H_0^{\boldsymbol{\Sigma}}$ are set as

$$H_0^{\boldsymbol{\mu}}: \; \boldsymbol{\mu}_{\text{ini}} = \boldsymbol{\mu}_{\text{sft}} \tag{5}$$

$$H_0^{\boldsymbol{\Sigma}}: \; \boldsymbol{\Sigma}_{\text{ini}} = \boldsymbol{\Sigma}_{\text{sft}} \tag{6}$$

where the mean vectors of $\boldsymbol{\mu}_{\text{ini}}$ and $\boldsymbol{\mu}_{\text{sft}}$, and the covariance matrices of $\boldsymbol{\Sigma}_{\text{ini}}$ and $\boldsymbol{\Sigma}_{\text{sft}}$ come from the populations of $\boldsymbol{\Upsilon}_{\text{ini}}$ and $\boldsymbol{\Upsilon}_{\text{sft}}$, respectively. Since $\boldsymbol{f}_{\text{ini}}$ is an unbiased estimator, $\boldsymbol{\mu}_{\text{ini}} = \boldsymbol{0}_p$ and $\boldsymbol{\Sigma}_{\text{ini}} = (1/W_G)\boldsymbol{\Upsilon}_{\text{ini}}^{\text{T}}\boldsymbol{\Upsilon}_{\text{ini}}$, where $\boldsymbol{0}_p$ denotes the p-dimensional zero vector. In order to determine whether or not to accept $H_0^{\boldsymbol{\mu}}$ and $H_0^{\boldsymbol{\Sigma}}$, two statistics $F_{\boldsymbol{\mu}}$ and $F_{\boldsymbol{\Sigma}}$ are constructed as [40]

$$F_{\boldsymbol{\mu}} = \frac{(W_G - p)W_G}{(W_G - 1)p}(\widehat{\boldsymbol{\mu}}_{\text{sft}} - \boldsymbol{\mu}_{\text{ini}})\widehat{\boldsymbol{\Sigma}}_{\text{sft}}^{-1}(\widehat{\boldsymbol{\mu}}_{\text{sft}} - \boldsymbol{\mu}_{\text{ini}})^{\text{T}} \tag{7}$$

$$F_{\boldsymbol{\Sigma}} = (W_G - 1)\Big(\ln|\boldsymbol{\Sigma}_{\text{ini}}| - p - \ln\big|\widehat{\boldsymbol{\Sigma}}_{\text{sft}}\big| + \text{tr}\big(\widehat{\boldsymbol{\Sigma}}_{\text{sft}}\boldsymbol{\Sigma}_{\text{ini}}^{-1}\big)\Big) \tag{8}$$

where $\widehat{\boldsymbol{\mu}}_{\text{sft}}$ and $\widehat{\boldsymbol{\Sigma}}_{\text{sft}}$ are the mean vector and covariance matrix of $\boldsymbol{\Upsilon}_{\text{sft}}$, respectively, estimated with $\widehat{\boldsymbol{\mu}}_{\text{sft}} = (1/W_G)\sum_{i=1}^{W_G}\boldsymbol{\Upsilon}_{\text{sft}}(i,:)$ and $\widehat{\boldsymbol{\Sigma}}_{\text{sft}} = (1/W_G - 1)\sum_{i=1}^{W_G}\big(\boldsymbol{\Upsilon}_{\text{sft}}(i,:) - \widehat{\boldsymbol{\mu}}_{\text{sft}}\big)^{\text{T}}\big(\boldsymbol{\Upsilon}_{\text{sft}}(i,:) - \widehat{\boldsymbol{\mu}}_{\text{sft}}\big)$, in which $\boldsymbol{\Upsilon}_{\text{sft}}(i,:)$ denotes the ith row of $\boldsymbol{\Upsilon}_{\text{sft}}$, while $|\cdot|$ and $\text{tr}(\cdot)$ are the determinant and trace operators, respectively.

Assuming that both $\boldsymbol{\Upsilon}_{\text{ini}}$ and $\boldsymbol{\Upsilon}_{\text{sft}}$ follow the multivariate normal distributions, then $F_{\boldsymbol{\mu}}$ follows the F distribution with the degrees of freedom p and $W_G - p$, denoted as $F_{\boldsymbol{\mu}} \sim \mathcal{F}(p, W_G - p)$, when $H_0^{\boldsymbol{\mu}}$ holds, while $F_{\boldsymbol{\Sigma}}$ follows the F distribution with the degrees of freedom f_1 and f_2, denoted as $F_{\boldsymbol{\Sigma}} \sim b\mathcal{F}(f_1, f_2)$, when $H_0^{\boldsymbol{\Sigma}}$ holds, where $b = (f_1/1 - D_1 - f_1/f_2)$, $f_1 = (p(p+1)/2)$, $f_2 = (f_1 + 2/D_2 - D_1^2)$, $D_1 = (2p+1-2/(p+1)/6(W_G-1))$, and $D_2 = ((p-1)(p+2)/)6(W_G - 1)^2$. See for example [40]. Therefore, the condition of accepting both $H_0^{\boldsymbol{\mu}}$ and $H_0^{\boldsymbol{\Sigma}}$ are

$$F_{\boldsymbol{\mu}} < \lambda_{\boldsymbol{\mu}} \;\&\; F_{\boldsymbol{\Sigma}} < \lambda_{\boldsymbol{\Sigma}} \tag{9}$$

where $\lambda_{\boldsymbol{\mu}}$ is the threshold value given the significance level $\alpha_{\boldsymbol{\mu}}$ which satisfies $\Pr\{F_{\boldsymbol{\mu}} < \lambda_{\boldsymbol{\mu}}\} = 1-\alpha_{\boldsymbol{\mu}}$, while $\lambda_{\boldsymbol{\Sigma}}$ is the threshold value given the significance level $\alpha_{\boldsymbol{\Sigma}}$ which satisfies $\Pr\{F_{\boldsymbol{\Sigma}} < \lambda_{\boldsymbol{\Sigma}}\} = 1-\alpha_{\boldsymbol{\Sigma}}$.

Let the local model set contains $L > 1$ independent local linear models $\{\boldsymbol{f}_l\}_{l=1}^{L}$, and $\boldsymbol{f}_{\text{ini}} = \boldsymbol{f}_L$. When condition (9) is violated, $\mathcal{W}_{\text{ini}}$ and $\mathcal{W}_{\text{sft}}$ are significantly different, and the new local linear model $\boldsymbol{f}_{\text{new}} = \boldsymbol{f}_{\text{sft}}$ is different from $\boldsymbol{f}_L$. We still need to test whether $\boldsymbol{f}_{\text{new}}$ differs from $\{\boldsymbol{f}_l\}_{l=1}^{L-1}$. This task is also fulfilled based on the hypothesis testing. Let the prediction error matrices of $\mathcal{W}_{\text{new}} = \{\boldsymbol{X}_{\text{sft}} \in \mathbb{R}^{W_G\times m}, \boldsymbol{Y}_{\text{sft}} \in \mathbb{R}^{W_G\times p}\}$ based on $\boldsymbol{f}_{\text{new}}$ and $\boldsymbol{f}_l$ be defined, respectively, by

$$\boldsymbol{\Upsilon}_{\text{new}} = \boldsymbol{Y}_{\text{sft}} - \boldsymbol{f}_{\text{new}}(\boldsymbol{X}_{\text{sft}}) \in \mathbb{R}^{W_G\times p} \quad (10)$$

$$\boldsymbol{\Upsilon}_l = \boldsymbol{Y}_{\text{sft}} - \boldsymbol{f}_l(\boldsymbol{X}_{\text{sft}}) \in \mathbb{R}^{W_G\times p}, \quad 1 \le l \le L-1. \quad (11)$$

To test whether $\boldsymbol{\Upsilon}_{\text{new}}$ and $\boldsymbol{\Upsilon}_l$ are significantly different or not, two null hypotheses $H_0^{\boldsymbol{\mu}_l}$ and $H_0^{\boldsymbol{\Sigma}_l}$ are set as

$$H_0^{\boldsymbol{\mu}_l}: \ \boldsymbol{\mu}_l = \boldsymbol{\mu}_{\text{new}} \quad (12)$$

$$H_0^{\boldsymbol{\Sigma}_l}: \ \boldsymbol{\Sigma}_l = \boldsymbol{\Sigma}_{\text{new}} \quad (13)$$

where the mean vectors $\boldsymbol{\mu}_l$ and $\boldsymbol{\mu}_{\text{new}}$, and the covariance matrices $\boldsymbol{\Sigma}_l$ and $\boldsymbol{\Sigma}_{\text{new}}$ come from the populations of $\boldsymbol{\Upsilon}_l$ and $\boldsymbol{\Upsilon}_{\text{new}}$, respectively. Here, $\boldsymbol{\mu}_{\text{new}}$ and $\boldsymbol{\Sigma}_{\text{new}}$ are estimated based on $\boldsymbol{\Upsilon}_{\text{new}}$ as $\boldsymbol{\mu}_{\text{new}} = \boldsymbol{0}_p$ and $\boldsymbol{\Sigma}_{\text{new}} = (1/W_G)\boldsymbol{\Upsilon}_{\text{new}}^{\text{T}}\boldsymbol{\Upsilon}_{\text{new}}$, respectively.

Again, to determine whether or not to accept $H_0^{\boldsymbol{\mu}_l}$ and $H_0^{\boldsymbol{\Sigma}_l}$, two statistics are constructed as

$$F_{\boldsymbol{\mu}}^{(l)} = \frac{(W_G-p)W_G}{(W_G-1)p}\left(\widehat{\boldsymbol{\mu}}_l - \boldsymbol{\mu}_{\text{new}}\right)\widehat{\boldsymbol{\Sigma}}_l^{-1}\left(\widehat{\boldsymbol{\mu}}_l - \boldsymbol{\mu}_{\text{new}}\right)^{\text{T}} \quad (14)$$

$$F_{\boldsymbol{\Sigma}}^{(l)} = (W_G-1)\left(\ln|\boldsymbol{\Sigma}_{\text{new}}| - p - \ln\left|\widehat{\boldsymbol{\Sigma}}_l\right| + \text{tr}\left(\widehat{\boldsymbol{\Sigma}}_l\boldsymbol{\Sigma}_{\text{new}}^{-1}\right)\right) \quad (15)$$

where $\widehat{\boldsymbol{\mu}}_l$ and $\widehat{\boldsymbol{\Sigma}}_l$ are the mean vector and covariance matrix of $\boldsymbol{\Upsilon}_l$, respectively, estimated with $\widehat{\boldsymbol{\mu}}_l = (1/W_G)\sum_{i=1}^{W_G}\boldsymbol{\Upsilon}_l(i,:)$ and $\widehat{\boldsymbol{\Sigma}}_l = (1/W_G-1)\sum_{i=1}^{W_G}\left(\boldsymbol{\Upsilon}_1(i,:) - \widehat{\boldsymbol{\mu}}_l\right)^{\text{T}}\left(\boldsymbol{\Upsilon}_l(i,:) - \widehat{\boldsymbol{\mu}}_l\right)$. Under the assumption that $\boldsymbol{\Upsilon}_l$ and $\boldsymbol{\Upsilon}_{\text{new}}$ follow the multivariate normal distributions, $F_{\boldsymbol{\mu}}^{(l)} \sim \mathcal{F}(p, W_G-p)$ when $H_0^{\boldsymbol{\mu}_l}$ holds, and $F_{\boldsymbol{\Sigma}}^{(l)} \sim b\mathcal{F}(f_1, f_2)$ when $H_0^{\boldsymbol{\Sigma}_l}$ holds. Hence, $\boldsymbol{f}_l$ and $\boldsymbol{f}_{\text{new}}$ are regarded to be identical if the following condition is met

$$F_{\boldsymbol{\mu}}^{(l)} < \lambda_{\boldsymbol{\mu}} \ \& \ F_{\boldsymbol{\Sigma}}^{(l)} < \lambda_{\boldsymbol{\Sigma}}. \quad (16)$$

Under this circumstance, either $\boldsymbol{f}_l$ or $\boldsymbol{f}_{\text{new}}$ is redundant and one of them should be removed. Since $\boldsymbol{f}_l$ is "older" than the $\boldsymbol{f}_{\text{new}}$, $\boldsymbol{f}_{\text{new}}$ is kept and $\boldsymbol{f}_l$ is removed. On the other hand, if condition (16) is violated $\forall l \in \{1, 2, \ldots, L–1\}$, $\boldsymbol{f}_{\text{new}}$ is different from $\boldsymbol{f}_l$ for $1 \le l \le L$. Thus, we have identified a new process state, and we add $\boldsymbol{f}_{\text{new}}$ to the local model set by setting $L = L+1$ and $\boldsymbol{f}_L = \boldsymbol{f}_{\text{new}}$.

The significance level $\alpha_{\boldsymbol{\mu}}$ and $\alpha_{\boldsymbol{\Sigma}}$ are usually set to small values, e.g., 0.05, 0.01, and they can be different according to the process data characteristics. Similar to our previous work [25], [26], the selection of window size W_G is a tradeoff between the adaptive ability to capture the local characteristics and the accuracy of the local model. The proposed local learning procedure is summarized in Algorithm 1.

Algorithm 1 Adaptive Local Learning

```
Initialization
Collect $\mathcal{W}_{\text{ini}}$ with $W_G$ consecutive samples from historical data, and construct multi-output LS linear model $\boldsymbol{f}_{\text{ini}}$ on $\mathcal{W}_{\text{ini}}$.
Calculate $\boldsymbol{\Upsilon}_{\text{ini}}$, and estimate $\boldsymbol{\mu}_{\text{ini}}$ and $\boldsymbol{\Sigma}_{\text{ini}}$.
Set $L = 1$, $\{\mathcal{W}_L, \boldsymbol{f}_L\} = \{\mathcal{W}_{\text{ini}}, \boldsymbol{f}_{\text{ini}}\}$ and $\mathcal{W}_{\text{sft}} = \mathcal{W}_L$.
Step 1: New local model detection
When a new data sample is available, shift $\mathcal{W}_{\text{sft}}$ one sample ahead.
Calculate $\boldsymbol{\Upsilon}_{\text{sft}}$, and estimate $\widehat{\boldsymbol{\mu}}_{\text{sft}}$ and $\widehat{\boldsymbol{\Sigma}}_{\text{sft}}$.
Construct $F_{\boldsymbol{\mu}}$ and $F_{\boldsymbol{\Sigma}}$ statistics using (7) and (8).
If condition (9) is satisfied
    Go to Step 1.
End if
Construct multi-output LS linear model $\boldsymbol{f}_{\text{sft}}$ on $\mathcal{W}_{\text{sft}}$.
Set $\mathcal{W}_{\text{new}} = \mathcal{W}_{\text{sft}}$ and $\boldsymbol{f}_{\text{new}} = \boldsymbol{f}_{\text{sft}}$.
Calculate $\boldsymbol{\Upsilon}_{\text{new}}$, and estimate $\boldsymbol{\mu}_{\text{new}}$ and $\boldsymbol{\Sigma}_{\text{new}}$.
Step 2: Redundant local model deletion
For $l = 1, 2, \ldots, L-1$
    Compute $\boldsymbol{\Upsilon}_l$, and estimate $\widehat{\boldsymbol{\mu}}_l$ and $\widehat{\boldsymbol{\Sigma}}_l$.
    Construct $F_{\boldsymbol{\mu}}^{(l)}$ and $F_{\boldsymbol{\Sigma}}^{(l)}$ statistics using (14) and (15).
    If condition (16) is satisfied
        Delete $\boldsymbol{f}_l$, set $\boldsymbol{f}_i = \boldsymbol{f}_{i+1}$ for $i = l, l+1, \ldots, L-1$, set $L = L-1$, then go to Step 3.
    End if
End for
Step 3: Add new local model
Set $L = L+1$, $\mathcal{W}_L = \mathcal{W}_{\text{new}}$ and $\boldsymbol{f}_L = \boldsymbol{f}_{\text{new}}$.
Return to Step 1.
```

Remark 1: This local learning procedure can operate both offline and online. During online operation, when the newest data sample $\{\boldsymbol{x}(t_{\text{next}}), \boldsymbol{y}(t_{\text{next}})\}$ is available, the data window shift one sample ahead, and the corresponding learning procedure can then be carried out. Unlike our previous work for single-output modeling [26], we consider multi-output modeling via multivariate statistics. This local learning procedure automatically encodes a newly emerging process state in the memory as a new local linear model. The local models in the memory are independent and represent different states of the process. Hence, our proposed learner is capable of achieving the desired maximum diversity, and it is capable of acquiring all the different process states that have appeared.

B. Multi-Output Selective Ensemble-Based Online Prediction

To mimic the adaptive intelligent decision making procedure of biological systems, at the second level of adaptation, our evolving model constructs a selective ensemble of the most relevant subset local linear models to produce an accurate prediction at each sample. Specifically, after the online operations at sample t, the local linear model set $\{\boldsymbol{f}_l\}_{l=1}^{L}$ has been produced by Algorithm 1, which represents the knowledge that the evolving system has acquired and stored in its memory. At the next sample of $t_{\text{next}} = t+1$, the task of online modeling is to produce the model prediction $\widehat{\boldsymbol{y}}(t_{\text{next}})$ for the process's true output $\boldsymbol{y}(t_{\text{next}})$, given the process input $\boldsymbol{x}(t_{\text{next}})$ and the available local model set $\{\boldsymbol{f}_l\}_{l=1}^{L}$. Our evolving model adopts an ensemble of the selected M local linear models from the model library $\{\boldsymbol{f}_l\}_{l=1}^{L}$ based on the q latest labeled data $\{\boldsymbol{x}(t-i), \boldsymbol{y}(t-i)\}_{i=0}^{q-1}$. Let the modeling error matrix of the lth multi-output local linear model $\boldsymbol{f}_l$ over the prediction window $\{\boldsymbol{x}(t-i), \boldsymbol{y}(t-i)\}_{i=0}^{q-1}$ be defined as

$$\boldsymbol{\Upsilon}_l(t) = \left[\boldsymbol{e}_l(t)\ \boldsymbol{e}_l(t-1)\cdots\boldsymbol{e}_l(t-q+1)\right]^{\mathrm{T}} \in \mathbb{R}^{q\times p} \tag{17}$$

where for $0 \le i \le q-1$

$$\begin{aligned}\boldsymbol{e}_l(t-i) &= \boldsymbol{y}(t-i) - \boldsymbol{f}_l(\boldsymbol{x}(t-i))\\ &= \left[e_{l,1}(t-i)\ e_{l,2}(t-i)\cdots e_{l,p}(t-i)\right]^{\mathrm{T}}.\end{aligned} \tag{18}$$

The performance metric of the lth local model is defined by

$$J_l(t) = \mathrm{tr}\big(\boldsymbol{\Upsilon}_l^{\mathrm{T}}(t)\boldsymbol{\Upsilon}_l(t)\big). \tag{19}$$

The sum of squared errors $J_l(t)$ is further transformed into a probability metric. Specifically, $J_l(t)$ is first converted to a similarity measure [41] ranging from 0 to 1 as follows:

$$\mathrm{Sm}_l(t) = \tfrac{1}{1+J_l(t)}. \tag{20}$$

The probability metric $\mathrm{Pr}_l(t)$ of the lth model is computed as the normalized similarity measure according to

$$\mathrm{Pr}_l(t) = \tfrac{\mathrm{Sm}_l(t)}{\sum_{i=1}^{L}\mathrm{Sm}_i(t)}. \tag{21}$$

$\mathrm{Pr}_l(t)$ quantifies the contribution of the lth local model to the ensemble, since a large $\mathrm{Pr}_l(t)$ indicates that the lth model is a good identifier for the ensemble model and vice versus.

Arrange all the L local models according to their probability values in descending order as

$$\mathrm{Pr}_{l_1}(t) \ge \cdots \ge \mathrm{Pr}_{l_M}(t) \ge \mathrm{Pr}_{l_{M+1}}(t) \ge \cdots \ge \mathrm{Pr}_{l_L}(t). \tag{22}$$

We select the first M best local models for constructing the ensemble model when the termination criterion

$$1 - \sum_{m=1}^{M}\mathrm{Pr}_{l_m}(t) < \xi \tag{23}$$

is met, where $0<\xi<1$ is the desired tolerance. The selected models yield the M model outputs

$$\widehat{\boldsymbol{y}}_{l_m}(t-i) = \boldsymbol{f}_{l_m}(\boldsymbol{x}(t-i)), \quad 1 \le m \le M \tag{24}$$

for $0 \le i \le q-1$. Denote $\boldsymbol{y}(t-i) = [y_1(t-i)\cdots y_p(t-i)]^{\mathrm{T}}$ and $\widehat{\boldsymbol{y}}_{l_m}(t) = \left[\widehat{y}_{l_m,1}(t-i)\cdots\widehat{y}_{l_m,p}(t-i)\right]^{\mathrm{T}}$. The estimate $\widehat{y}_j(t-i)$ of the jth system output $y_j(t-i)$ is given as the weighted sum of the M selected subset models, which is computed by

$$\widehat{y}_j(t-i) = \sum_{m=1}^{M}\theta_{m,j}(t)\widehat{y}_{l_m,j}(t-i) \quad 0 \le i \le q-1 \tag{25}$$

for $1 \le j \le p$, where the nonnegative $\theta_{m,j}(t)$ is the combining coefficient for the mth selected local model of the jth output, and the combining coefficients must satisfy the constraint

$$\sum_{m=1}^{M}\theta_{m,j}(t) = 1, \quad 1 \le j \le p. \tag{26}$$

The jth estimation errors for $1 \le j \le p$

$$\varepsilon_j(t-i) = y_j(t-i) - \widehat{y}_j(t-i), \quad 0 \le i \le q-1 \tag{27}$$

are utilized to determine the combining coefficients of jth output. Specifically, the optimal combining coefficients can be obtained by minimizing the following cost function

$$V(t) = \frac{1}{2}\sum_{j=1}^{p}V_j(t) = \sum_{j=1}^{p}\sum_{i=0}^{q-1}\varepsilon_j^2(t-i). \tag{28}$$

Minimizing $V(t)$ is equivalent to minimizing each $V_j(t)$, $1 \le j \le p$, separately. Because of the constrain $\sum_{m=1}^{M}\theta_{m,j}(t) = 1$

$$\begin{aligned}V_j(t) &= \frac{1}{2}\sum_{i=0}^{q-1}\bigg(y_j(t-i) - \sum_{m=1}^{M}\theta_{m,j}(t)\widehat{y}_{l_m,j}(t-i)\bigg)^2\\ &= \frac{1}{2}\sum_{i=0}^{q-1}\bigg(\sum_{m=1}^{M}\theta_{m,j}(t)y_j(t-i) - \sum_{m=1}^{M}\theta_{m,j}(t)\widehat{y}_{l_m,j}(t-i)\bigg)^2\\ &= \frac{1}{2}\sum_{i=0}^{q-1}\bigg(\sum_{m=1}^{M}\theta_{m,j}(t)e_{l_m,j}(t-i)\bigg)^2 = \frac{1}{2}\boldsymbol{\theta}_j^{\mathrm{T}}(t)\bar{\boldsymbol{E}}_j(t)\boldsymbol{\theta}_j(t)\end{aligned} \tag{29}$$

for $1 \le j \le p$, where $\boldsymbol{\theta}_j(t) = \left[\theta_{1,j}(t)\cdots\theta_{M,j}(t)\right]^{\mathrm{T}}$ and $\bar{\boldsymbol{E}}_j(t)$ is the estimated error covariance matrix of the jth output, which is given by

$$\bar{\boldsymbol{E}}_j(t) = \sum_{i=0}^{q-1}\begin{bmatrix} e_{l_1,j}^2(t-i) & \cdots e_{l_1,j}(t-i)e_{l_M,j}(t-i)\\ \vdots & \ddots & \vdots\\ e_{l_1,j}(t-i)e_{l_M,j}(t-i)\cdots & e_{l_M,j}^2(t-i)\end{bmatrix}. \tag{30}$$

Hence, we can form the following optimization problem to determine the optimal $\boldsymbol{\theta}_j(t)$ for the jth output:

$$\begin{aligned}&\min_{\boldsymbol{\theta}_j}\ \frac{1}{2}\boldsymbol{\theta}_j^{\mathrm{T}}(t)\bar{\boldsymbol{E}}_j(t)\boldsymbol{\theta}_j(t),\\ &\text{s.t.}\ \sum_{m=1}^{M}\theta_{m,j}(t) = 1\end{aligned} \tag{31}$$

where $1 \le j \le p$. The Lagrangian function for the single-output optimization (31) is given by

$$L\big(\boldsymbol{\theta}_j(t);\gamma\big) = \frac{1}{2}\boldsymbol{\theta}_j^{\mathrm{T}}(t)\bar{\boldsymbol{E}}_j(t)\boldsymbol{\theta}_j(t) + \gamma\left(\boldsymbol{1}_M^{\mathrm{T}}\boldsymbol{\theta}_j(t) - 1\right) \tag{32}$$

where $\gamma > 0$ is a Lagrange multiplier. Letting $(\partial L/\partial\boldsymbol{\theta}_j(t)) = \boldsymbol{0}_M$ yields

$$\bar{\boldsymbol{E}}_j(t)\boldsymbol{\theta}_j(t) + \gamma\boldsymbol{1}_M = \boldsymbol{0}_M, \quad 1 \le j \le p. \tag{33}$$

This suggests that the optimal combining vector $\widehat{\boldsymbol{\theta}}_j$ can be obtained as follows. First, calculate

$$\widetilde{\boldsymbol{\theta}}_j(t) = \left[\widetilde{\theta}_{1,j}(t)\cdots\widetilde{\theta}_{M,j}(t)\right]^{\mathrm{T}} = \bar{\boldsymbol{E}}_j^{-1}(t)\boldsymbol{1}_M \tag{34}$$

which is followed by the normalization:

$$\widehat{\theta}_{m,j}(t) = \frac{1}{\sum_{i=1}^{M}\widetilde{\theta}_{i,j}(t)}\widetilde{\theta}_{m,j}(t), \quad 1 \le m \le M. \tag{35}$$

The jth prediction $\widehat{y}_j(t_{\mathrm{next}})$ for the jth system's true output $y_j(t_{\mathrm{next}})$ is produced as the selected ensemble

$$\widehat{y}_j(t_{\mathrm{next}}) = \sum_{m=1}^{M}\widehat{\theta}_{m,j}(t)f_{l_m,j}\big(\boldsymbol{x}(t_{\mathrm{next}})\big), \quad 1 \le j \le p \tag{36}$$

where $f_{l_m,j}(\boldsymbol{x}(t))$ denotes the jth element of $\boldsymbol{f}_{l_m}(\boldsymbol{x}(t))$.

Algorithm 2 Multi-Output Selective Ensemble Prediction

```
1: **Initialization**
2: Give $W_G$, $q$ and $\xi$.
3: At beginning of online operation with sampling instance $t$, local model set $\{\boldsymbol{f}_l\}_{l=1}^{L}$ has been constructed.
4: **Step 1: Online prediction**
5: Give input $\boldsymbol{x}(t_{\text{next}})$ at new sample time $t_{\text{next}} = t + 1$.
6: Calculate probability $\Pr_l(t)$ of each local model using (21) for $1 \le l \le L$.
7: Select $M$ subset models with termination criterion (23).
8: Calculate error covariance matrix $\bar{\boldsymbol{E}}_j(t)$, $1 \le j \le p$, for each output using (30).
9: Calculate optimal combining coefficients $\widehat{\boldsymbol{\theta}}_j(t)$, $1 \le j \le p$, using (34) and (35).
10: Predict true system outputs $y_j(t_{\text{next}})$, $1 \le j \le p$, with selective ensemble prediction (36).
11: **Step 2: Model adaptation**
12: When $\boldsymbol{y}(t_{\text{next}})$ is available, add $\{\boldsymbol{x}(t_{\text{next}}), \boldsymbol{y}(t_{\text{next}})\}$ to dataset with $t = t + 1$.
13: Carry out relevant operations to adapt local model set.
14: Set $t_{\text{next}} = t_{\text{next}} + 1$, and go to **Step 1**.
```

Algorithm 3 Multi-Output Local Model Pruning

```
1: **Initialization**
2: Give $W_P$, set counters of all local models $\text{count}_l = 0$ for $1 \le l \le L$, set $t = t_{\text{ini}}$ and $\text{index} = 0$.
3: **Step 1: Pruning in pruning model window**
4: Perform multi-output selective ensemble prediction.
5: **If** $(t - t_{\text{ini}} \le W_P)$
6:   **For** $l = 1, 2, \ldots, L$
7:     **If** $f_l$ is not selected at current sample $t$
8:       $\text{count}_l = \text{count}_l + 1$.
9:     **End if**
10:   **End for**
11:   Set $t = t + 1$ and go to **Step 1**.
12: **Else**
13:   **For** $l = 1, 2, \ldots, L$
14:     **If** $\text{count}_l = W_G$
15:       Add $l$ to pruning model index set $\Gamma$, and set $\text{index} = \text{index} + 1$.
16:     **End if**
17:   **End for**
18:   Delete $f_l$ for all $l \in \Gamma$, and set $L = L - \text{index}$.
19: **End if**
20: **Step 2: Pruning model window update**
21: Clear counters for all local models, set $t_{\text{ini}} = t$ and $\text{index} = 0$, and go to **Step 1**.
```

Algorithm 2 summarizes the multi-output selective ensemble-based online prediction. In line 13 of Algorithm 2, the relevant operations may include lines 6–24 of adaptive local learning in Algorithm 1.

Remark 2: The two algorithmic parameters for selective ensemble prediction are the desired tolerance of ξ and the number of latest labeled data q. A large q benefits the accuracy of the online SER prediction but imposes high computational complexity. A small q, on the other hand, offers high adaptability and is suitable for highly time-varying data. The threshold ξ trades off the accuracy of the SER prediction and the computational complexity. From all the L acquired independent local models that include the newest process state information, this SER procedure selects the most relevant subset of M local models to form an accurate prediction of the current process output. Obviously, the size of the selected ensemble M is different for different prediction samples. A very small positive regularization term λ_E can be added to the diagonal elements of $\bar{\boldsymbol{E}}_j(t)$ for $1 \le j \le p$ to ensure invertibility.

C. Local Model Set Pruning

In a fast-changing environment, a biological system must be capable of removing out-of-date knowledge that is no longer relevant from its memory to free up memory space for fast acquiring new knowledge. This characteristic is also vital for our multi-output SER evolving model, because for highly nonstationary processes, the base local model set is likely to become very large over a long period of online adaptation, and this imposes high online computational complexity in constructing the SER predictor. Therefore, at the third level of adaptation, a pruning strategy is adopted to remove out-of-date local models from memory, and hence, to alleviate the online computational burden. The essence of local model pruning is to remove those local base models that are far from the current data dynamics since these local models are not needed in modeling the current process dynamics.

Recalling (17)–(23), M local linear models are selected to produce the ensemble prediction, which also suggests a way of the local model set pruning. For example, the l_{M+1}th to l_Lth models are not selected for the current prediction, and thus, they may be removed. However, pruning a model based on its "one-sample" prediction horizon may not be sufficiently reliable, as these unselected models at the current sample may be important at $t_{\text{next}}+1$. To make the pruning more reliable, the work [26] introduced the concept of "memory depth" for an ensemble learner. In the local learning procedure, a local linear model is constructed based on a data window with the window size W_G. Within this data window, the process's dynamics are assumed to be stationary. Similarly, we introduce a data window for model pruning with the window size W_P. Specifically, if a local model is never selected for the consecutive W_P samples of the window, then it can be removed with high confidence.

The multi-output local model pruning strategy is listed in Algorithm 3. Line 4 of Algorithm 3 corresponds to the SER prediction operations of lines 6–10 in Algorithm 2. The rest pruning operations in Algorithm 3 occur at line 13 of Algorithm 2. Since we have a window size W_G for adaptive local learning, we can conveniently set $W_P = W_G$.

Remark 3: Since the newest local model f_L represents the newly emerged process state, it is critical to keep it in the current prediction horizon. Thus, we always set count_L to zero. To maintain the local model set diversity, and hence, the identification accuracy, a minimal number of local model $L_{\min}$ should be guaranteed. If the number of those unselected local models in Γ exceeds $L-L_{\min}$, only the oldest $L-L_{\min}$ local models can be removed. Let L_{ini} be the number of local linear models obtained in the initial training. We can set $L_{\min}=L_{\text{ini}}$. This pruning strategy provides good plasticity to our multi-output SER learner, and it can dramatically reduce the online computational complexity in adaptive SER prediction.

III. Case Studies

The performance of the proposed multi-output SER learning method is evaluated using three case studies, which are: a simulated two-output nonlinear system, a nonisothermal CSTR process [37], [38], and an industrial microwave heating system [39]. The testing prediction error covariance $\text{Cov}(\boldsymbol{E})$ is used to evaluate the multi-output online modeling performance, where $\text{Cov}(\boldsymbol{E})=(1/N-1)\sum_{i=1}^{N}(\boldsymbol{E}_i-\overline{\boldsymbol{E}})(\boldsymbol{E}_i-\overline{\boldsymbol{E}})^{\text{T}}$, N is the number of test samples and $\overline{\boldsymbol{E}}$ is the sample average of $\boldsymbol{E}$. The online computational complexity is quantified by its averaged computation time per sample (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 performance of our proposed method is compared with existing typical online learning approaches, including the multi-output OLS (MOLS) [32]–[35], the OS-ELM [14], the EOS-ELM [24], and the recently proposed single-output GAP-SER [26], which is denoted as the S-SER. For the MOLS, nonlinear modeling is achieved by using the Gaussian RBF kernel and the initial training is carried out by constructing a compact multi-output RBF model using the OLS learning algorithm. During online operation, the weight adaptation is performed by the RLS algorithm with the fixed model structure. For the OS-ELM, an RBF network is initialized during training by randomly selecting a large number of input data points as its RBF centers, and the online adaptation of the OS-ELM involves the weight updating using the RLS algorithm. Similarly, for the EOS-ELM, a number of OS-ELM base models are trained on the same data set, and each base model's centers are randomly selected from the training data points as in OS-ELM. The forgetting factor of the RLS algorithm is set to 0.98. For the S-SER, the multiple single-output SER models are constructed using the single-output GAP-SER of [26], one for modeling output of the system.

A. Simulated Two-Output Nonlinear System

The simulated two-output nonlinear system of [33] and [34] is considered. The data set contained 1000 noisy observations generated using the model

$$\begin{aligned} y_1(t) &= 0.5y_1(t-1)+u(t-1)+0.4\tanh(u(t-2)) \\ &\quad +0.1\sin(\pi y_1(t-2))y_2(t-1)+\epsilon_1(t), \\ y_2(t) &= 0.3y_2(t-1)+0.1y_2(t-2)y_1(t-1) \\ &\quad +0.4\exp(-u^2(t-1))y_1(t-2)+\epsilon_2(t), \end{aligned} \tag{37}$$

where the system input $u(t)$ is uniformly distributed in $[-0.5,\ 0.5]$, and the zero-mean Gaussian noise $\boldsymbol{\epsilon}(t)=[\epsilon_1(t)\ \epsilon_2(t)]^{\text{T}}$ has a covariance matrix $0.04\boldsymbol{I}_2$. Initial conditions are set as $y_1(0)=y_2(0)=y_1(-1)=y_2(-1)=0$ and $u(0)=u(-1)=0$. The first 500 data generated from (37) are used for training and the rest 500 samples for testing. Note that our proposed approach and the MOLS and the S-SER do not really need such a large number of training samples but the ELM-based models need a large number of training samples, as an ELM model must contain a large number of hidden nodes. In modeling this system, we use the model input vector as

$$\boldsymbol{x}(t)=[y_1(t-1)y_1(t-2)y_2(t-1)y_2(t-2)u(t-1)u(t-2)]^{\text{T}}\in\mathbb{R}^6. \tag{38}$$

In the simulation, 100 independent realizations are generated. The performance of each method are presented by its means and standard deviations (STDs) of the test $\log(\det(\text{Cov}(\boldsymbol{E})))$ and ACTpS, respectively, over the 100 realizations.

For our proposed model, the window size, innovation length, and decision threshold are empirically chosen to be $W_G=30$, $q=10$ and $\xi=0.9$, respectively. For the S-SER, three algorithmic parameters are chosen to be $W_G=20$, $q=10$ and $\xi=0.1$ to achieve the best prediction performance. Note that we do not attempt to optimize the algorithmic parameters in training, as such an optimal algorithmic setting is only meaningful when the underlying system is stationary. For online identification of a nonstationary system, the optimal algorithmic setting is fast time varying, and it is prohibitive to online learn this fast time-varying optimal algorithmic setting. Since each algorithmic parameter has a clear physical interpretation, the appropriate value can be set empirically. The detailed analysis can be found in our previous work [25], [26].

The test results of all the five methods, including the model size, the ACTpS, the error covariance $\log(\det(\text{Cov}(\boldsymbol{E})))$, and the average ensemble size, are summarized in Table I. This system is not seriously time varying, and all the methods are expected to perform well. This is confirmed by Table I. More specifically, the MOLS with 10 RBF nodes outperforms the OS-ELM with 100 nodes and the EOS-ELM of 5 base models with each base model having 100 nodes, in terms of both online modeling accuracy and computational complexity, and its performance is even slightly better than the S-SER. Among all the models, our proposed method attains the lowest $\log(\det(\text{Cov}(\boldsymbol{E})))$ and its average SER size is much smaller than that of the multiple ($p=2$) S-SER models. The comparison of average error covariance learning curves for various models is shown in Fig. 1.

B. Nonisothermal CSTR Process

The nonisothermal CSTR with an irreversible reaction (A→B) is widely used as a benchmark for nonlinear and time-varying process modeling and identification [37], [38]. Based on the underlying physical-chemical laws, it is well known that this process is governed by the following two nonlinear ordinary differential equations [42]

$$\frac{\mathrm{d}C_A}{\mathrm{d}t}=\frac{q_f}{V}(C_{Af}-C_A)-K_0\ C_A\exp\left(-\frac{E}{RT}\right)\phi_c(t)$$

TABLE I

SIMULATED TWO-OUTPUT NONLINEAR SYSTEM: COMPARISON OF ONLINE PREDICTION AND ADAPTIVE MODELING PERFORMANCE (AVERAGE±STD) FOR THE OS-ELM, EOS-ELM, MOLS, S-SER, AND THE PROPOSED METHOD

Algorithm	$\log(\det(\mathrm{Cov}(\boldsymbol{E})))$	ACTpS (ms)	Models/Nodes		Average ensemble size
			Initial	Final	
OS-ELM	-2.31±0.07	0.27±0.01	100	100	-
	-1.36±0.07	6.63±0.11	500	500	-
EOS-ELM	-2.32±0.04	2.30±0.05	100	100	5
	-1.46±0.16	58.71±2.37	500	500	5
MOLS	-2.60±0.04	0.04±0.01	10	10	-
	-2.59±0.05	0.06±0.01	20	20	-
	-2.54±0.05	0.08±0.01	30	30	-
S-SER	**-2.46±0.11**	SO1 0.15±0.02	2.13±1.01	3.02±1.39	2.71±1.19
		SO2 0.15±0.02	2.01±1.01	2.88±1.20	2.52±1.04
		MO **0.30±0.04**	**4.14±1.35**	**5.90±1.87**	**5.23±1.61**
Proposed	**-2.72±0.06**	**0.47±0.04**	**4.99±1.57**	**5.12±1.57**	**1±0**

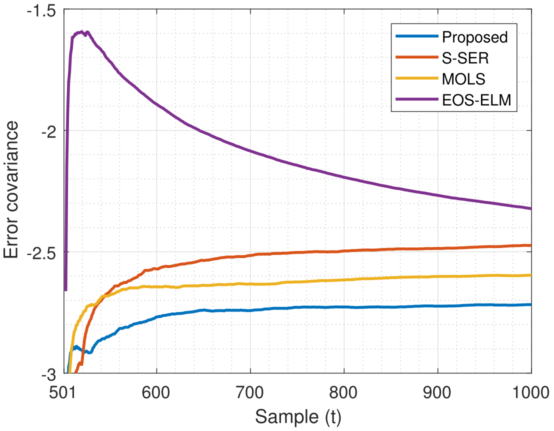


Fig. 1. Comparison of average $\log(\det(\mathrm{Cov}(\boldsymbol{E})))$ learning curves of various models for the simulated nonlinear system. The EOS-ELM has total of 500 hidden nodes, and the MOLS has 10 hidden nodes. The OS-ELM has a very close learning curve with the EOS-ELM and is omitted here.

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{q_f}{V}(T_f - T) + \frac{(-\Delta H)K_0 C_A}{\rho C_p}\exp\left(-\frac{E}{RT}\right)\phi_c(t) + \frac{\rho_c C_{pc}}{\rho C_p V}q_c\left(1-\exp\left(-\frac{hA}{q_c\rho C_{pc}}\phi_h(t)\right)\right)(T_{cf}-T) \tag{39}$$

where $\phi_h(t)$ is the fouling coefficient, $\phi_c(t)$ is the deactivation coefficient, C_A is the effluent concentration (controlled variable), T is the output temperature (controlled variable), q_c is the coolant flow rate (manipulated variable), q_f is the feed flow rate (manipulated variable), C_{AF} is the feed concentration, T_f is the feed temperature, and T_{cf} is the coolant inlet temperature. The other process parameters, together with the operating conditions, are given in Table II.

Accurate simulator of the nonisothermal CSTR is built by first-order differencing the ordinary differential equation (39) [37], [38]. To obtain the multiple operating conditions, ±5% step changes are added to both coolant (q_c) and feed (q_f) flow rates. The step size is set to be 0.1. We follow the same approach to generate the process input–output data. The process inputs are q_f and q_c, and its outputs are C_A and T.

TABLE II

NOMINAL CSTR OPERATING CONDITION

$h_A = 7\times10^5\mathrm{cal.min}^{-1}\mathrm{K}^{-1}$	$T = 440.2\mathrm{K}$
$K_0 = 7.2\times10^{10}\mathrm{min}^{-1}$	$T_f = 350\mathrm{K}$
$E/R = 9.95\times10^3\mathrm{K}$	$T_{cf} = 350\mathrm{K}$
$-\Delta H = 2\times10^5\mathrm{cal.mol}^{-1}$	$C_{Af} = 1\mathrm{moll}^{-1}$
$C_A = 8.36\times10^{-2}\mathrm{moll}^{-1}$	$V = 100\mathrm{l}$
$C_p,\ C_{pc} = 1\mathrm{calg}^{-1}\mathrm{K}^{-1}$	$\rho_c,\ \rho = 1000\mathrm{gl}^{-1}$
$q_c = 103.41\mathrm{l.min}^{-1}$	$q_f = 100\mathrm{l.min}^{-1}$

For this process, the system input vector is chosen as

$$\boldsymbol{x}(t) = \left[C_A(t-1)\ T(t-1)\ q_f(t-1)\ q_c(t-1)\right]^{\mathrm{T}} \in \mathbb{R}^4 \tag{40}$$

and the system output vector is

$$\boldsymbol{y}(t) = \left[C_A(t), T(t)\right]^{\mathrm{T}} \in \mathbb{R}^2. \tag{41}$$

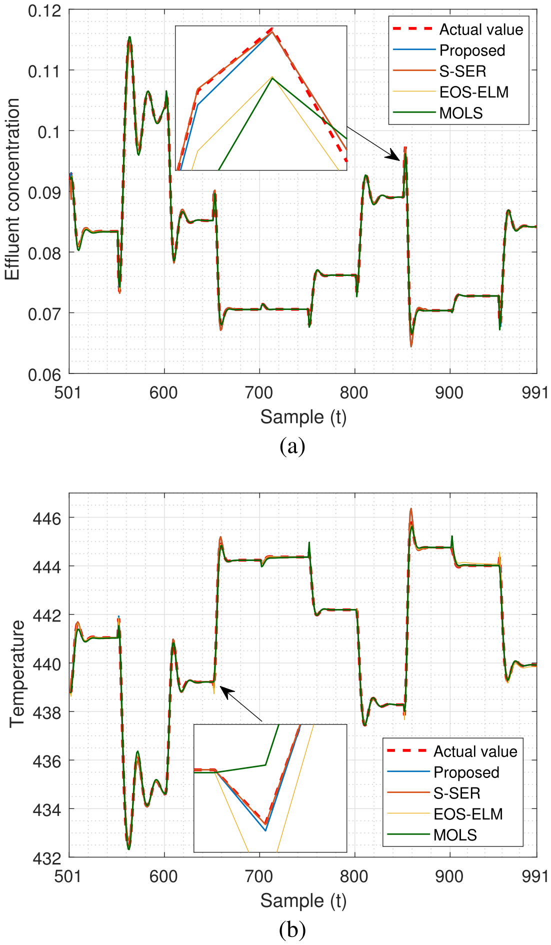
1) Constant Deactivation and Fouling Coefficients: We first consider the case that the deactivation coefficient $\phi_c(t)$ and the fouling coefficient $\phi_h(t)$ are both constant, which are set to 1. Then, 991 samples are generated with the first 500 samples for training and the remaining 491 data samples for testing. For our proposed method, the three algorithmic parameters are empirically chosen to be $W_G = 100$, $q = 20$ and $\xi = 0.9$, respectively. For each single-output model of the S-SER, the three algorithmic parameters are chosen to be $W_G = 10$, $q = 20$ and $\xi = 0.9$ to achieve its best modeling accuracy.

The test performance of the five models is compared in Table II. The OS-ELM with 500 nodes and the EOS-ELM with 5 base models each having 500 nodes attain a similar performance of $\log(\det(\mathrm{Cov}(\boldsymbol{E}))) = -10.07$, but they impose very high ACTpS of 6.58 and 58.54 ms, respectively. The MOLS with 20 nodes has a slightly worst accuracy but imposes the lowest ACTpS of 0.09 ms. The S-SER approach attains a better accuracy than the OS-ELM and EOS-ELM while imposing a significantly lower ACTpS of 1.04 ms. Our proposed method achieves the most accurate prediction with an ACTpS of 2.73 ms. The online prediction values

TABLE III

NONISOTHERMAL CSTR PROCESS WITH TWO CONSTANT PROCESS PARAMETERS: COMPARISON OF ONLINE PREDICTION AND ADAPTIVE MODELING PERFORMANCE FOR THE OS-ELM, EOS-ELM, MOLS, S-SER, AND THE PROPOSED METHOD

Algorithm	$\log(\det(\mathrm{Cov}(\boldsymbol{E})))$	ACTpS (ms)		Models/Nodes		Average ensemble size
				Initial	Final	
OS-ELM	-5.3252	0.3239		100	100	-
	-10.0731	6.5782		500	500	-
EOS-ELM	-5.2789	2.4132		100	100	5
	-10.0732	58.5402		500	500	5
MOLS	-7.3414	0.0716		10	10	-
	-9.1829	0.0942		20	20	-
	-7.1830	0.1069		30	30	-
S-SER	**-10.4252**	SO1	0.5562	65	65	5.66
		SO2	0.4801	62	62	6.01
		MO	**1.0372**	**127**	**127**	**11.67**
Proposed	**-12.5867**	**2.7334**		**45**	**53**	**5.47**



(a)

(b)

Fig. 2. Online identification of CSTR process with two constant process parameters: (a) Effluent concentration and (b) Output temperature.

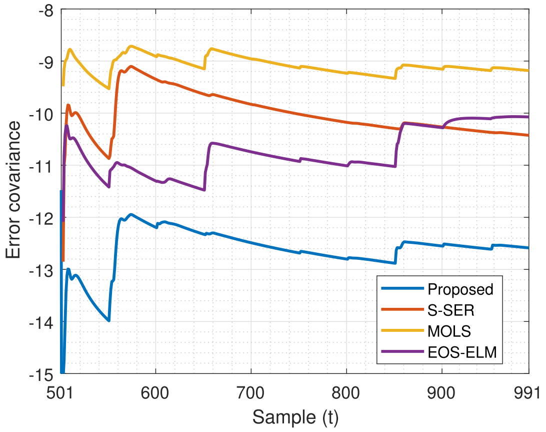


Fig. 3. Comparison of $\log(\det(\mathrm{Cov}(\boldsymbol{E})))$ learning curves of various models for the CSTR process with two constant process parameters. The EOS-ELM has total of 2500 hidden nodes, and the MOLS has 20 hidden nodes.

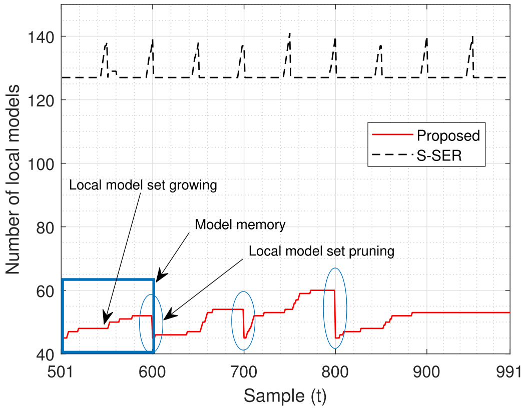


Fig. 4. Comparison of online local model set learning curves of the two SER methods for the CSTR process with two constant process parameters.

of the MOLS with 20 nodes, the EOS-ELM with a total of 2500 nodes, the S-SER and our proposed method are compared with the actual process output observations in Fig. 2. Fig. 3 shows the error covariance learning curves of various models.

From Table II, it can be seen that the online modeling efforts of our proposed multi-output SER method are lower than those of the S-SER method. Specifically, it imposes a smaller model set size and smaller average ensemble, compared with the multiple S-SER models. This is confirmed by the local online model set learning curves of the two SER methods shown in Fig. 4. Observe the red curve in Fig. 4 that our

TABLE IV

NONISOTHERMAL CSTR PROCESS WITH TWO TIME-VARYING PARAMETERS: COMPARISON OF ONLINE PREDICTION AND ADAPTIVE MODELING PERFORMANCE FOR THE OS-ELM, EOS-ELM, MOLS, S-SER, AND THE PROPOSED METHOD

Algorithm	$\log(\det(\text{Cov}(\boldsymbol{E})))$	ACTpS (ms)		Models/Nodes		Average ensemble size
				Initial	Final	
OS-ELM	-6.9808	0.3445		100	100	-
	-9.8185	6.5982		500	500	-
EOS-ELM	-6.9808	2.5011		100	100	5
	-9.9095	58.5402		500	500	5
MOLS	-8.0568	0.0796		10	10	-
	-8.4489	0.1036		20	20	-
	-8.4299	0.1088		30	30	-
S-SER	**-14.2614**	SO1	0.7722	40	40	14.83
		SO2	0.3797	27	27	10.45
		MO	**1.1519**	**67**	**67**	**25.29**
Proposed	**-21.0996**	**3.5615**		**47**	**57**	**5.56**

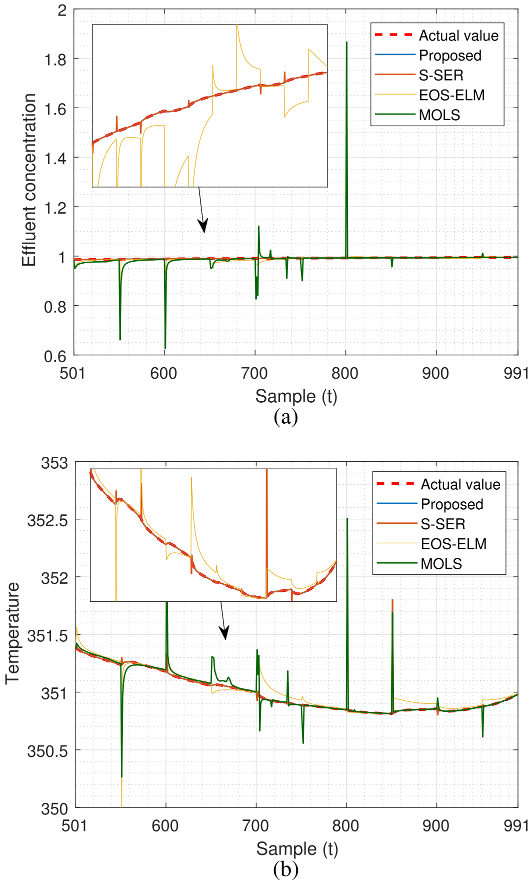
proposed growing and pruning strategies can effectively add new local linear models and remove out-of-date ones, respectively, within the model memory $W_G = 100$ (blue rectangle). However, the recorded ACTpS of the proposed method is higher than that of the S-SER. This is owing to MATLAB software implementation. The high-dimensional matrix calculation in MATLAB is complicated and time consuming. In real-time operation with other programming platforms, we expect that the online ACTpS of the proposed method will be significantly lower than that of the S-SER.

2) *Time-Varying Deactivation and Fouling Coefficients:* We next consider time-varying $\phi_c(t)$ and $\phi_h(t)$. Fouling and catalyst deactivation phenomena present two main sources to produce nonstationary characteristics in the CSTR process during its normal operation. These time-varying dynamic characteristics are governed by the following equations:

$$\phi_h(t) = 1 - 0.01t, \quad \phi_c(t) = \exp\left(-0.00067\frac{E}{RT}t\right) \tag{42}$$

where the fouling effect is caused by deposing material on the heat transfer surface, and catalyst is deactivated due to poisoning. Again, 991 samples are generated with the first 500 data for training and the rest 491 data for testing.

Table IV compares the performance of different models. For this seriously nonstationary process, the S-SER significantly outperforms the OS-ELM and EOS-ELM, in terms of both online prediction accuracy and computational complexity. Furthermore, our proposed method considerably outperforms the S-SER in online prediction accuracy. The MOLS has the poorest accuracy but imposes the lowest ACTpS. Online prediction values by the various models are compared with the actual system output observations in Fig. 5, while Fig. 6 shows the error covariance learning curves of various models. Again although the recorded ACTpS of the proposed method is higher than that of the SER, it actually requires smaller model set size and much smaller average ensemble than the multiple S-SER models, as confirmed by the local online model set learning curves of the two SER methods given in Fig. 7.



(a)

(b)

Fig. 5. Online identification of CSTR process with two time-varying process parameters: (a) effluent concentration and (b) output temperature.

C. Microwave Heating Process

Microwave heating technology has found wide-ranging applications due to its many advantages over conventional heating methods, including selective and volumetric heating, rapid heat transfer, and pollution-free environment. However, a major drawback of microwave heating is the temperature runaway, caused by properties of the material and the inner electromagnetic field distribution, which may lead to unwanted

TABLE V

MHP: COMPARISON OF ONLINE PREDICTION AND ADAPTIVE MODELING PERFORMANCE FOR THE OS-ELM, EOS-ELM, MOLS, S-SER, AND THE PROPOSED METHOD

Algorithm	$\log(\det(\mathrm{Cov}(\boldsymbol{E})))$	ACTpS (ms)		Models/Nodes		Average ensemble size
				Initial	Final	
OS-ELM	-10.4544	0.3261		100	100	-
	-13.3419	6.5582		500	500	-
EOS-ELM	-10.4561	2.2521		100	100	5
	-13.3418	57.5702		500	500	5
MOLS	-10.3060	0.0499		5	5	-
	-12.3751	0.0507		10	10	-
	-3.8084	0.0571		15	15	-
S-SER	**-12.7403**	SO1	0.2629	11	11	5.69
		SO2	0.2414	10	10	5.20
		SO3	0.3807	23	23	11.56
		MO	**0.8850**	**44**	**44**	**22.46**
Proposed	**-14.1326**	**1.4401**		**25**	**28**	**2**

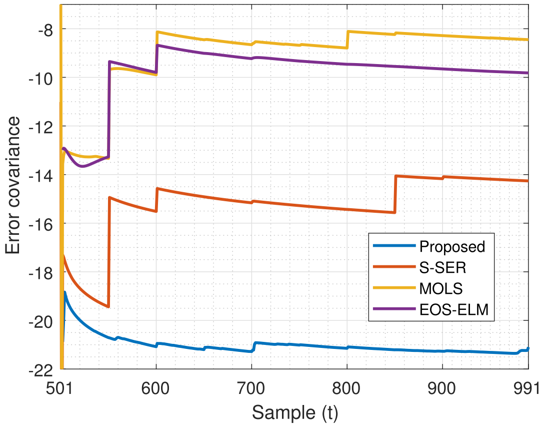


Fig. 6. Comparison of $\log(\det(\mathrm{Cov}(\boldsymbol{E})))$ learning curves of various models for the CSTR process with two time-varying process parameters. The EOS-ELM has total of 2500 hidden nodes and the MOLS has 20 hidden nodes.

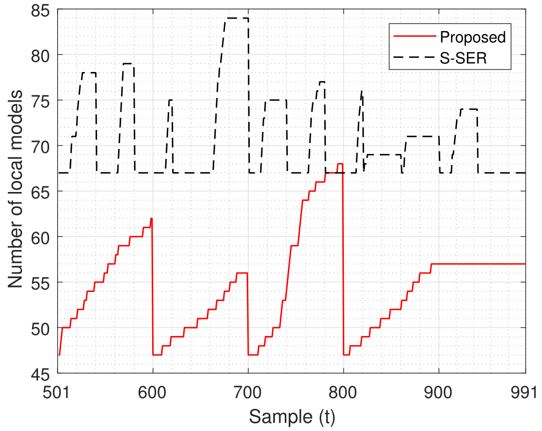


Fig. 7. Comparison of online local model set learning curves of the two SER methods for the CSTR process with two time-varying process parameters.

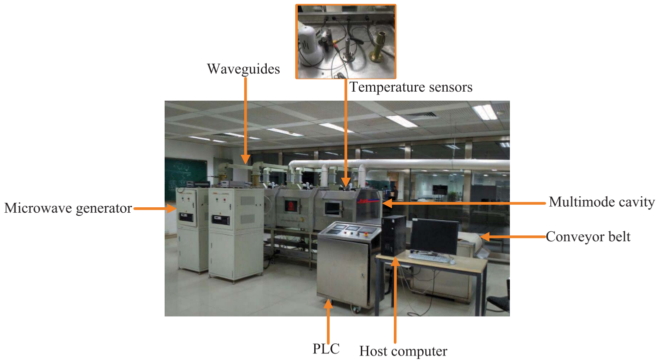


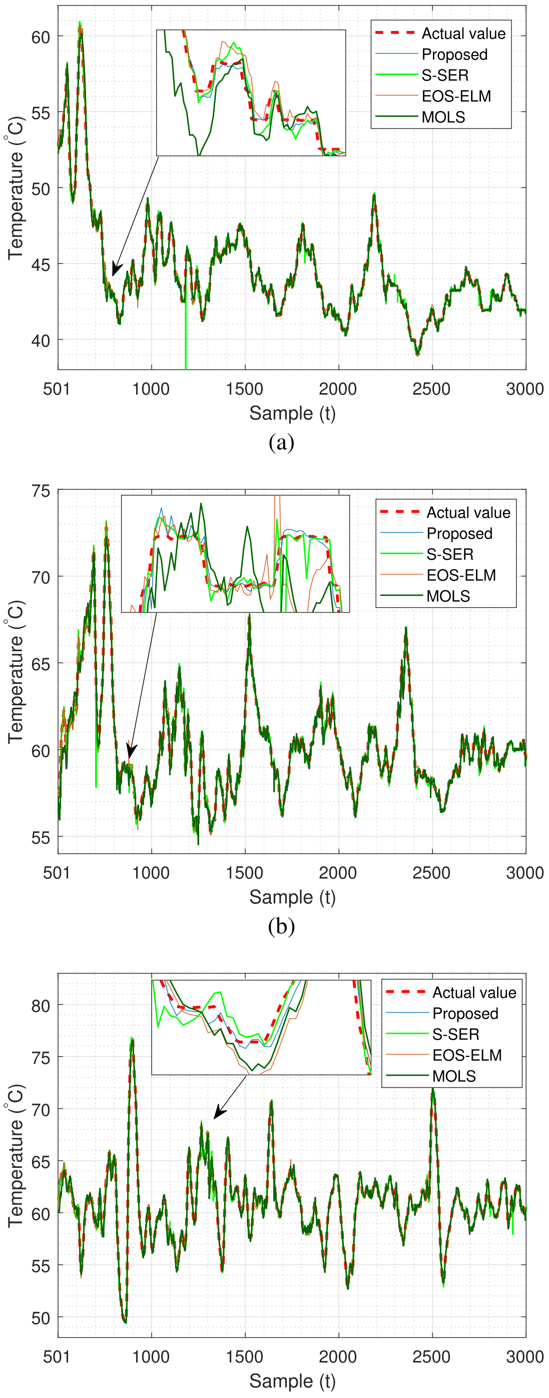
Fig. 8. Industrial microwave heating system.

combustion and destruction in industrial processes [39]. Therefore, in the control of the microwave heating process (MHP), material temperature is important feedback information. However, establishing a temperature model from the first principle is challenging since the MHP involves multiphysical fields coupling. Hence, assumptions have to be made and, consequently, unmodeled dynamics always exist in a model derived from the first principle [43], [44]. This motivates us to investigate data-driven predictive model [45]–[47].

A distributed microwave heating system [48], [49] is shown in Fig. 8. It consists of five microwave generators and waveguides. Microwave generated by each microwave generator is transmitted through the corresponding waveguide, fed into the cavity, and absorbed by the heated material. The material is continuously transported through cavity by the conveyor belt, whose speed can be adjusted by a motor driver. Three fiber optical sensors (FOSs), denoted as FOS1 to FOS3, are placed at three different locations to online record multiple-points of temperature. During the real-time operation of this MHP, the control center receives the measured temperature values from the FOSs and sends control commends, including the five microwave powers $u_{p_i}(t)$, $1 \le i \le 5$, for the five microwave generators and the conveyor speed $v(t)$ to the cavity. Thus, the control inputs to this MHP are given by

$$\boldsymbol{u}(t) = \left[u_{p_1}(t)\ u_{p_2}(t)\ u_{p_3}(t)\ u_{p_4}(t)\ u_{p_5}(t)\ v(t)\right]^{\mathrm{T}}. \quad (43)$$

Each FOS measures the temperature at its location, which is the MHP's output $y_i(t)$, $1 \le i \le 3$. Because of near



(a)

(b)

(c)

Fig. 9. Online identification of MHP by various models: (a) FOS1, (b) FOS2, and (c) FOS3.

instantaneous response of MHP, the process's temperatures $\boldsymbol{y}(t) = \left[y_1(t)\ \ y_2(t)\ \ y_3(t)\right]^{\mathrm{T}}$ can be adequately modeled as

$$\boldsymbol{y}(t) = \boldsymbol{f}_{\mathrm{nn}}(\boldsymbol{x}(t); t) \tag{44}$$

where $\boldsymbol{f}_{\mathrm{nn}}(\cdot; t)$ represents the unknown nonlinear time-varying system mapping with the input vector given by

$$\boldsymbol{x}(t) = \left[\boldsymbol{y}^{\mathrm{T}}(t-1)\ \boldsymbol{u}^{\mathrm{T}}(t-1)\right]^{\mathrm{T}} \in \mathbb{R}^9. \tag{45}$$

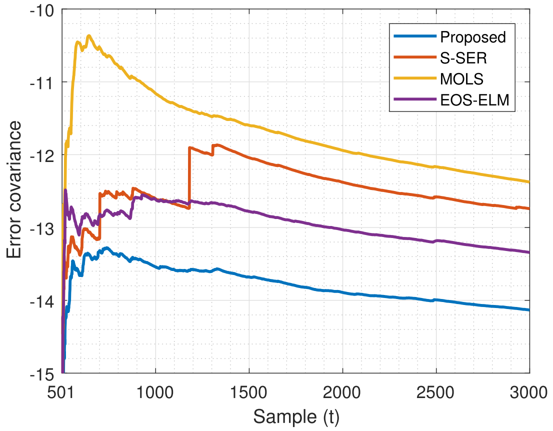


Fig. 10. Comparison of $\log(\det(\mathrm{Cov}(\boldsymbol{E})))$ learning curves of various models for MHP identification. The EOS-ELM has total of 2500 hidden nodes, while the MOLS has 10 hidden nodes.

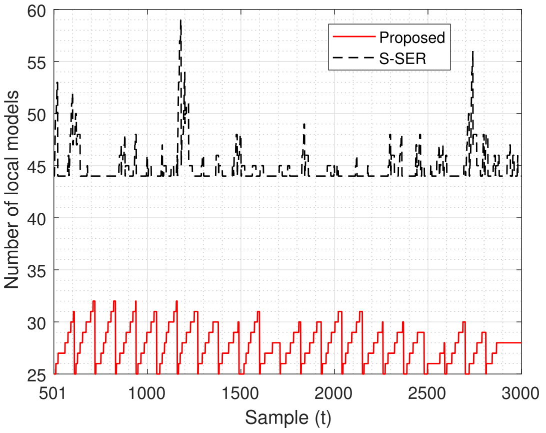


Fig. 11. Comparison of online local model learning curves for MHP identification.

From this distributed microwave heating system, 3000 process data have been collected. We first normalize the microwave power inputs and the temperature measurements according to

$$\bar{u}_{p_i}(t) = \frac{u_{p_i}(t)}{1000}, \quad 1 \le i \le 5, \tag{46}$$

$$\bar{y}_i(t) = \frac{y_i(t) - y_{i_{\min}}}{y_{i_{\max}} - y_{i_{\min}}}, \quad 1 \le i \le 3 \tag{47}$$

where $y_{i_{\min}}$ and $y_{i_{\max}}$ are the minimum and maximum temperatures recorded by the ith FOS, respectively. We use the first 500 samples for training, and the last 2500 samples for online prediction and adaptive modeling. For our proposed method, the three algorithmic parameters are empirically chosen to be $W_G = 110$, $q = 25$ and $\xi = 0.95$, respectively. For each S-SER model, the three algorithmic parameters are chosen to be $W_G = 20$, $q = 25$, and $\xi = 0.5$.

Table V compares the performance of different predictive models. The predictor outputs versus the actual process outputs by various models are shown in Fig. 9, where the EOS-ELM has a total of 2500 hidden nodes and the MOLS has 10 hidden nodes. Furthermore, the comparison of error covariance learning curves is given in Fig. 10. In this case, the OS-ELM with 500 nodes and the EOS-ELM with a total of 2500 nodes outperform both the S-SER and MOLS, but they impose extremely higher online computational complexity. In particular, the EOS-ELM with a total of 2500 nodes requires the highest ACTpS of 57.57 ms. By contrast, our proposed multi-output SER achieves the best prediction performance with a reasonable online computation time of ACTpS = 1.44 ms. The comparison of the local online model set learning curves of the two SER methods is given in Fig. 11, which also confirms that our proposed method needs a smaller model set and smaller average ensemble, compared with the three S-SER models.

IV. Conclusion

In this article, a novel multi-output SER evolving model has been developed for multi-output nonlinear and nonstationary process identification. Our proposed online adaptive learner has been inspired by the fundamental biological system learning principle, namely, the ability to acquire new knowledge to the memory and to remove out-of-date knowledge from the memory so that intelligent decision can be made based on the newest and relevant knowledge in the memory. Hence, our contribution has been threefold. First, our adaptive local learning enables automatically identifying every newly emerging process state and constructing a matching local multi-output linear model via multivariate statistic hypothesis testing. Second, the optimal online prediction of the system's multi-outputs is obtained by a selective ensemble of the most relevant subset of local linear models. Third, an effective pruning strategy removes the most out-of-date local linear models that are no longer needed in modeling the process in order to free up the memory for fast acquiring the new process knowledge. This pruning strategy also significantly reduces online computational complexity without scarifying the prediction accuracy. Extensive studies have been conducted, including a simulated nonlinear system, a simulator-based CSTR process and a real-world microwave heating system identification. The results obtained have demonstrated that our proposed multi-output selective ensemble identification technique attains the best online modeling accuracy, compared with a range of the state-of-art methods for online identification of nonlinear and nonstationary multi-output processes, while imposing a reasonably low online computational complexity which meets the real-time operation constraint. This study, therefore, has provided a reliable and accurate online system model for designing efficient real-time control strategy for multi-output nonlinear and nonstationary processes.

References

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.

[2] K. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Germany: Springer-Verlag, 2005.

[3] Z.-H. Zhou, N. V. Chawla, Y. Jin, and G. J. Williams, "Big data opportunities and challenges: Discussions from data analytics perspectives [discussion forum]," *IEEE Comput. Intell. Mag.*, vol. 9, no. 4, pp. 62–74, Nov. 2014.

[4] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: A survey," *IEEE Comput. Intell. Mag.*, vol. 10, no. 4, pp. 12–25, Nov. 2015.

[5] L. Rutkowski, "Generalized regression neural networks in time-varying environment," *IEEE Trans. Neural Netw.*, vol. 15, no. 3, pp. 576–596, May 2004.

[6] J. Liu and D.-S. Chen, "Nonstationary fault detection and diagnosis for multimode processes," *AIChE J.*, vol. 56, no. 1, pp. 207–219, Jan. 2010.

[7] H. Ning, G. Qing, T. Tian, and 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.*, vol. 30, no. 2, pp. 389–404, Feb. 2019.

[8] J. Shan, H. Zhang, W. Liu, and Q. Liu, "Online active learning ensemble framework for drifted data streams," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 486–498, Feb. 2019.

[9] J. L. Lobo, I. Laña, J. Del Ser, M. N. Bilbao, and N. Kasabov, "Evolving spiking neural networks for online learning over drifting data streams," *Neural Netw.*, vol. 108, pp. 1–19, Dec. 2018.

[10] S. Chen and S. A. Billings, "Recursive prediction error parameter estimator for non-linear models," *Int. J. Control*, vol. 49, no. 2, pp. 569–594, Feb. 1989.

[11] S. Chen, "Nonlinear time series modelling and prediction using Gaussian RBF networks with enhanced clustering and RLS learning," *Electron. Lett.*, vol. 31, no. 2, pp. 117–118, Jan. 1995.

[12] F. Ding, P. X. Liu, and G. Liu, "Multiinnovation least-squares identification for system modeling," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 3, pp. 767–778, Jun. 2010.

[13] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[14] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1411–1423, Nov. 2006.

[15] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, nos. 16–18, pp. 3460–3468, Oct. 2008.

[16] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Inf. Fusion*, vol. 37, pp. 132–156, Sep. 2017.

[17] L. L. Minku, A. P. White, and X. Yao, "The impact of diversity on online ensemble learning in the presence of concept drift," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 5, pp. 730–742, May 2010.

[18] S. Wang, L. L. Minku, and X. Yao, "A systematic study of online class imbalance learning with concept drift," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4802–4821, Oct. 2018.

[19] Y. Sun, K. Tang, L. L. Minku, S. Wang, and X. Yao, "Online ensemble learning of data streams with gradually evolved classes," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1532–1545, Jun. 2016.

[20] S. Wang, L. L. Minku, and X. Yao, "Resampling-based ensemble methods for online class imbalance learning," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1356–1368, May 2015.

[21] Y. Sun, K. Tang, Z. Zhu, and X. Yao, "Concept drift adaptation by exploiting historical knowledge," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4822–4832, Oct. 2018.

[22] D. Brzezinski and J. Stefanowski, "Reacting to different types of concept drift: The accuracy updated ensemble algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 1, pp. 81–94, Jan. 2014.

[23] G. Song, Y. Ye, H. Zhang, X. Xu, R. Y. K. Lau, and F. Liu, "Dynamic clustering forest: An ensemble framework to efficiently classify textual data stream with concept drift," *Inf. Sci.*, vol. 357, pp. 125–143, Aug. 2016.

[24] Y. Lan, Y. C. Soh, and G.-B. Huang, "Ensemble of online sequential extreme learning machine," *Neurocomputing*, vol. 72, nos. 13–15, pp. 3391–3395, Aug. 2009.

[25] T. Liu, S. Chen, S. Liang, and C. J. Harris, "Selective ensemble of multiple local model learning for nonlinear and nonstationary systems," *Neurocomputing*, vol. 378, pp. 98–111, Feb. 2020.

[26] T. Liu, S. Chen, S. Liang, and C. J. Harris, "Growing and pruning selective ensemble regression for nonlinear and nonstationary systems," *IEEE Access*, vol. 8, pp. 73278–73292, Apr. 2020.

[27] H. Chen, Y. Gong, and X. Hong, "A new adaptive multiple modelling approach for non-linear and non-stationary systems," *Int. J. Syst. Sci.*, vol. 47, no. 9, pp. 2100–2110, Jul. 2016.

[28] H. Chen, Y. Gong, X. Hong, and S. Chen, "A fast adaptive tunable RBF network for nonstationary systems," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2683–2692, Dec. 2016.

[29] H. Chen, Y. Gong, and X. Hong, "Online modeling with tunable RBF network," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 935–947, Jun. 2013.

[30] T. Xiong, Y. Bao, and Z. Hu, “Multiple-output support vector regression with a firefly algorithm for interval-valued stock price index forecasting,” *Knowl.-Based Syst.*, vol. 55, pp. 87–100, Jan. 2014.

[31] D. Du, K. Li, X. Li, M. Fei, and H. Wang, “A multi-output two-stage locally regularized model construction method using the extreme learning machine,” *Neurocomputing*, vol. 128, pp. 104–112, Mar. 2014.

[32] S. Chen, P. M. Grant, and C. F. N. Cowan, “Orthogonal least squares algorithm for training multi-output radial basis function networks,” *IEE Proc. F, Radar Signal Process.*, vol. 139, no. 6, pp. 378–384, Dec. 1992.

[33] S. Chen, “Multi-output regression using a locally regularised orthogonal least-squares algorithm,” *IEE Proc.-Vis., Image Signal Process.*, vol. 149, no. 4, pp. 185–195, Aug. 2002.

[34] S. Chen, C. J. Harris, and X. Hong, “Sparse multioutput radial basis function network construction using combined locally regularised orthogonal least square and D-optimality experimental design,” *IEE Proc.-Control Theory Appl.*, vol. 150, no. 2, pp. 139–146, Mar. 2003.

[35] S. A. Billings and S. Chen, “The determination of multivariable nonlinear models for dynamic systems,” in *Neural Network Systems Techniques and Applications* (Control and Dynamic Systems), vol. 7, C. L. Leondes, Ed. San Diego, CA, USA: Academic, 1998, pp. 231–278.

[36] W. Shao, S. Chen, and C. J. Harris, “Adaptive soft sensor development for multi-output industrial processes based on selective ensemble learning,” *IEEE Access*, vol. 6, pp. 55628–55642, Oct. 2018.

[37] C. Cheng and M. S. Chiu, “A new data-based methodology for nonlinear process modeling,” *Chem. Eng. Sci.*, vol. 59, no. 13, pp. 2801–2810, Jul. 2004.

[38] M. Nikravesh, A. E. Farell, and T. G. Stanford, “Control of nonisothermal CSTR with time varying parameters via dynamic neural network control (DNNC),” *Chem. Eng. J.*, vol. 76, no. 1, pp. 1–16, Jan. 2000.

[39] C. A. Vriezinga, S. Sánchez-Pedreño, and J. Grasman, “Thermal runaway in microwave heating: A mathematical analysis,” *Appl. Math. Model.*, vol. 26, no. 11, pp. 1029–1038, Nov. 2002.

[40] B. P. Korin, “On the distribution of a statistic used for testing a covariance matrix,” *Biometrika*, vol. 55, no. 1, pp. 171–178, Mar. 1968.

[41] S.-H. Jung, B.-C. Moon, and D. Han, “Unsupervised learning for crowdsourced indoor localization in wireless networks,” *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2892–2906, Nov. 2016.

[42] Y. You and M. Nikolaou, “Dynamic process modeling with recurrent neural networks,” *AIChE J.*, vol. 39, no. 10, pp. 1654–1667, Oct. 1993.

[43] J. Zhong, S. Liang, Y. Yuan, and Q. Xiong, “Coupled electromagnetic and heat transfer ODE model for microwave heating with temperature-dependent permittivity,” *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 8, pp. 2467–2477, Aug. 2016.

[44] J. Zhong, S. Liang, and Q. Xiong, “Improved receding horizon H_∞ temperature spectrum tracking control for debye media in microwave heating process,” *J. Process Control*, vol. 71, pp. 14–24, Nov. 2018.

[45] T. Liu, S. Liang, Q. Xiong, and K. Wang, “Two-stage method for diagonal recurrent neural network identification of a high-power continuous microwave heating system,” *Neural Process. Lett.*, pp. 1–22, Feb. 2019.

[46] T. Liu, S. Liang, Q. Xiong, and K. Wang, “Data-based online optimal temperature tracking control in continuous microwave heating system by adaptive dynamic programming,” *Neural Process. Lett.*, vol. 51, no. 1, pp. 167–191, Feb. 2020.

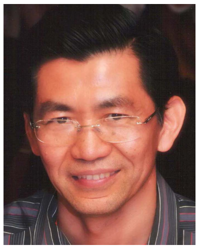
[47] T. Liu, S. Liang, Q. Xiong, and K. Wang, “Integrated CS optimization and OLS for recurrent neural network in modeling microwave thermal process,” *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12267–12280, Aug. 2020.

[48] T. Liu, S. Liang, Q. Xiong, and K. Wang, “Adaptive critic based optimal neurocontrol of a distributed microwave heating system using diagonal recurrent network,” *IEEE Access*, vol. 6, pp. 68839–68840, Dec. 2018.

[49] T. Liu, S. Liang, and J. Hu, “Expert control system based hierarchical control strategy for tunnel microwave rice drying,” in *Proc. ECC*, Naples, Italy, Jun. 2019, pp. 3619–3624.

Tong Liu received the B.Sc. degree in automation from the College of Automation, Chongqing University, Chongqing, China, in 2016, where he is currently pursuing the Ph.D. degree in control theory and control engineering.

From 2018 to 2019, he was a Visiting Ph.D. Student with the School of Electronics and Computer Science, University of Southampton, Southampton, U.K. His current research interests include online learning, system identification, neural networks, machine learning, and intelligent control system design.



Sheng Chen (Fellow, IEEE) received the B.Eng. degree in control engineering from the East China Petroleum Institute, Dongying, China, in 1982, and the Ph.D. degree in control engineering from the City, University of London, London, U.K., in 1986, and the D.Sc. degree from the University of Southampton, Southampton, U.K., in 2005.

From 1986 to 1999, he held research and academic appointments with The University of Sheffield, Sheffield, U.K., The University of Edinburgh, Edinburgh, U.K., and the University of Portsmouth, Portsmouth, U.K. Since 1999, he has been with the School of Electronics and Computer Science, University of Southampton, U.K., where he holds the post of Professor in intelligent systems and signal processing. He is currently a Distinguished Adjunct Professor with King Abdulaziz University, Jeddah, Saudi Arabia. His research interests include neural network and machine learning, wireless communications, and adaptive signal processing. He has published over 700 research articles.

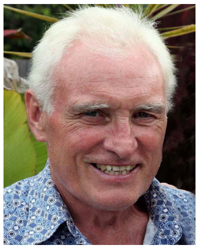
Dr. Chen is a Fellow of the Royal Academy of Engineering, U.K., and IET. He is an original ISI highly cited researcher in engineering in 2004. He has more than 15 200 Web of Science citations with an h-index 54 and more than 30 700 Google Scholar citations with an h-index 75.

Shan Liang (Member, IEEE) received the M.Sc. degree in control science and engineering from the College of Automation, Chongqing University, Chongqing, China, in 1995, and the Ph.D. degree from the Department of Mechanical Systems Engineering, Kumamoto University, Kumamoto, Japan, in 2004.

His current research interests include nonlinear system modeling and adaptive control, with the applications to microwave heating processes and intelligent transportation systems.

Shaojun Gan received the B.Eng. degree in automation from Huainan Normal University, Huainan, China, in 2010, and the Ph.D. degree in control engineering from the School of Automation, Chongqing University, Chongqing, China, in 2016.

From 2014 to 2016, he was a visiting Ph.D. Student with the School of Electronics, Electrical Engineering and Computer Science, Queen’s University Belfast, Belfast, U.K. From 2017 to 2019, he was a Research Fellow with the School of Electronic and Electrical Engineering, University of Leeds, Leeds, U.K. He is currently an Assistant Professor with the College of Metropolitan Transportation, Beijing University of Technology, Beijing, China. His main research interests include system modeling and machine learning methods, with the applications to manufacturing energy systems and intelligent transportation systems.



Chris J. Harris received the B.Sc. degree from the University of Leicester, Leicester, U.K., in 1967, the M.A. degree from the University of Oxford, Oxford, U.K., in 1976, and the Ph.D. and D.Sc. degrees from the University of Southampton, Southampton, U.K., in 1972 and 2001, respectively.

He held senior academic appointments with the Imperial College London, London, U.K., the University of Oxford, and The University of Manchester, Manchester, U.K. He was a Deputy Chief Scientist with the U.K. Government. He is currently an Emeritus Research Professor with the University of Southampton. He has coauthored over 500 scientific research articles during a 50-year research career.

Prof. Harris was elected to the U.K. Royal Academy of Engineering in 1996. He was a recipient of the IEEE Senior Achievement Medal for data fusion research and the IET Faraday Medal for distinguished international research in machine learning.