Correntropy-Based Evolving Fuzzy Neural System

Rong-Jing Bao, Hai-Jun Rong, *Member, IEEE*, Plamen Angelov, *Fellow, IEEE*, Badong Chen, *Senior Member, IEEE*, and Pak Kin Wong

***Abstract*—In this paper, a correntropy-based evolving fuzzy neural system (correntropy-EFNS) is proposed for approximation of nonlinear systems. Different from the commonly used mean-square error criterion, correntropy has a strong outliers rejection ability through capturing the higher moments of the error distribution. Considering the merits of correntropy, this paper brings contributions to build EFNS based on the correntropy concept to achieve a more stable evolution of the rule base and update of the rule parameters instead of the commonly used mean-square error criterion. The correntropy-EFNS (CEFNS) begins with an empty rule base and all rules are evolved online based on the correntropy criterion. The consequent part parameters are tuned based on the maximum correntropy criterion where the correntropy is used as the cost function so as to improve the non-Gaussian noise rejection ability. The steady-state convergence performance of the CEFNS is studied through the calculation of the steady-state excess mean square error (EMSE) in two cases: i) Gaussian noise; and ii) non-Gaussian noise. Finally, the CEFNS is validated through a benchmark system identification problem, a Mackey-Glass time series prediction problem as well as five other real-world benchmark regression problems under both noise-free and noisy conditions. Compared with other evolving fuzzy neural systems, the simulation results show that the proposed CEFNS produces better approximation accuracy using the least number of rules and training time and also owns superior non-Gaussian noise handling capability.**

***Index Terms*—Evolving Fuzzy Neural System, Mean-Square Error, Correntropy, Steady-State Excess Mean Square Error, Nonlinear System**

I. Introduction

FUZZY neural systems (FNSs) are able to deal with uncertain information through IF-THEN rules and to model nonlinear systems by learning from input-output datasets or data streams [1], [2]. Extracting a set of suitable fuzzy rules from the available input-output datasets or data streams is a challenge, especially, if it is to be done automatically. In traditional approaches of designing the FNSs, the fuzzy rules are pre-trained and learnt from a given set of input-output data in an offline mode [3], [4]. This leads to a static rule base of the resulting system. However, in the real-world most processes are nonstationary and possess dynamical and changeable behavior over time. The FNSs with static structure can hardly handle nonstationary processes, in which the modeling performance is degraded. To address this issue, intensive research work have been concentrated on developing evolving FNSs (EFNSs) aiming to design FNSs with a higher level of flexibility and autonomy [2], [4]–[15]. EFNSs evolve their structure according to the information from the dynamic processes and thus are able to deal with data shift and drift due to changes in the operating environment over time.

Since the beginning of the century, various approaches of developing EFNSs have been proposed [4], [6]. Depending on the Euclidean distance of the fuzzy rules to the newest datum, some EFNSs, i.e., dynamic evolving neuro-fuzzy inference system (DENFIS) [16], self-constructing neuro-fuzzy inference network [17] are developed to recruit the rules. In addition to the distance criterion, the learning error at each instant is applied as a criterion for evolving the system structure in some EFNSs [7], [8]. But these EFNSs are prone to outliers. In parallel, based on the informative potential of the new datum representing accumulated spatial proximity information, an evolving Takagi-Sugeno (eTS) model [9] was proposed. In the eTS model, the sum distances to all data points is taken into account in the potential and thus outliers supported by few data points have no chance to become rule centers. A family of eTS models, such as simplified eTS model (Simp_eTS) [10], evolving extended TS model (exTS) [2], and evolving TS model from streaming data (eTS+) [11] have been developed. In a similar way, a meta-cognitive neuro-fuzzy inference system (McFIS) [12] is proposed to update the rule base based on the spherical potential of each learning datum that represents the novel knowledge contained in each datum.

Except the instantaneous information of each datum, the statistical contribution of the new datum defined as the significance criterion is proposed as an effective way to evolve the rule base in [13], [18]. The author has proposed a sequential adaptive fuzzy inference system (SAFIS) [14] where the determination of rules is based on the influence of a rule represented by its statistical contribution to the system output. But the calculation of rule influence requires that the input data are uniformly distributed. This limits its application. To improve the performance of SAFIS, the author has developed an extended SAFIS (ESAFIS) [15] based on the modified influence of a rule in which the uniform distribution of the input data is not necessary. In [19], a multivariable Gaussian evolving fuzzy modeling system (eMG) is developed using a recursive clustering algorithm inspired by the idea of participatory learning. A evolving fuzzy model (eFuMo) method [20] is proposed for a monitoring system based on the normalized Mahalanobis distance. To create a new cluster, certain previous samples must satisfy the distance adding criterion in order to reduce influence of the outliers. Using the Mahalanobis distance, an evolving possibilistic fuzzy modeling approach (ePFM) [21] is proposed to forecast realized volatility with jumps. The ePFM includes a utility measure to avoid unused

Rong-Jing Bao, Hai-Jun Rong are with State Key Laboratory for Strength and Vibration of Mechanical Structures, Shaanxi Key Laboratory of Environment and Control for Flight Vehicle, School of Aerospace, Xi'an Jiaotong University, Xi'an, P. R. China (e-mail: hjrong@mail.xjtu.edu.cn). Plamen Angelov is with Data Science Group, School of Computing and Communications, Lancaster University, Lancaster, UK, LA1 4WA. Badong Chen is with Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, P. R. China, 710049. Pak Kin Wong is with Department of Electromechanical Engineering, Faculty of Science and Technology, University of Macau, Macao.

clusters that persists during learning.

In addition to the identification of the fuzzy rules, the design of the EFNSs also requires mechanism for online adaptation of parameters of the rules. When analyzing the existing work, the parameter update is mainly based on the mean square error (MSE) criterion. To achieve the minimum of the MSE between the desired and the system outputs, some parameter optimization algorithms, i.e., recursive least squares (RLS) [22] or extended Kalman filter methods [23] are applied to adjust the parameters of rules. Actually, the minimum of MSE equivalently minimizes the variance of the modeling error distribution. Also, under the MSE criterion, only the second-order moment of the error distribution are considered. In this way, the utilization of MSE to update the parameters of EFNSs is optimal only when the modeling errors are distributed under the Gaussian law.

In general, the real-time dynamical and time-varying processes are highly nonlinear and, thus owns the complex non-Gaussian properties of the modeling errors. To tackle the non-Gaussian nature of the modeling errors, an effective way is to adopt the information content of the error distribution as a quality criterion [24]–[26]. By capturing the higher moments of the error distribution, the information contained in the higher moments are passed to the EFNSs instead of remaining in the error distribution. A measure of information content is correntropy that represents a nonlinear and local similarity measure between two arbitrary random variables. Correntropy is a special case of the cross entropy in information theoretic learning (ITL) [27]. In correntropy, the error space is mapped to a Reproducing Kernel Hilbert Space (RKHS). In the case, an L_2 distance is defined in the kernel space and a nonlinear distance measure in the original error space [28] is presented. In comparison with MSE, correntropy has been proven to have a strong outliers rejection ability in many applications, e.g., adaptive filtering [29] [30], pattern classification [31], dimension reduction [32] and feature selection [33].

Correntropy criterion has been applied to update the parameters of a neuro-fuzzy classifier by Principe, the creator of ITL theory. The resulting classifier is applied to classify the sleep state of the patient and better results are achieved compared with the MSE criterion. But in the work the rule base need to be pre-determined and can not be changed according to the learning process. Also some key issues still need to be solved, e.g., the convergence of parameter update under the correntropy criterion. Considering the merits of correntropy, this paper brings further contributions to build the EFNS based on the correntropy concept of ITL to achieve the evolution of the rules and update of the rule parameters instead of the absolute error and the MSE criteria widely used in the existing literature. Until know, not any publication has so far come to use online building of the EFNS based on correntropy. From the viewpoint of ITL, the proposed CENFS has several appealing advantages, which are given here.

1) The correntropy presents the probability of similarity between the desired and real outputs, thus it is directly related with the addition criterion in the proposed CEFNS. That is, the structure of fuzzy rules is evolved automatically based on the correntropy. This offers a robust mechanism to avoid the detrimental effect of outliers, and is intrinsically different from the use of the absolute error in existing techniques.

2) In the proposed CEFNS, online adaptation of parameters existing in the rules is achieved according to the maximum correntropy criterion (MCC) where the similarity between the desired and real outputs is maximized in the correntropy sense. Then the gradient ascent technique is adopted to train the consequent parameters of rules. Compared with MSE criterion, the MCC represents a local similarity criterion and is very effective to handle non-Gaussian dynamic environments. Also, the computational complexity of the proposed algorithm in this paper is lower than those of the existing works.

3) The steady-state excess mean square error (EMSE) of the proposed CEFNS under the MCC are studied in our work. This is totally different from the prior studies. Two cases of noise are considered. One is the Gaussian noise in which the exact value of the steady-state EMSE is gained through building a fixed-point equation. The other is the non-Gaussian noise where an approximated expression of the steady-state EMSE is analytically derived according to a Taylor expansion approach.

The rest of the paper is organized as follows. Section II presents the proposed CEFNS structure. The learning details of the proposed CEFNS is described in section III. The convergence analysis in terms of the EMSE for the proposed CEFNS is presented in section 5. In section V the proposed system is evaluated through simulation results. In section VI we give the conclusion.

II. CEFNS Structure

For the sequentially arrived input-output data streams $\{(\mathbf{x}_n, y_n), n = 1, 2, ...\}$ each of which contains the N_x-dimensional inputs and 1-dimensional output at time n, the construction of the proposed CEFNS aims to model an unknown function $\mathbf{F} : \Re^{N_x} \to \Re$ from the data stream. In the CEFNS, the fuzzy rules are built "from scratch" and gradually evolve from the given input-output data with an adaptive self-constructing rule generator. The rules make use of the form of the first-order Takagi-Sugeno (TS) fuzzy model, whose kth rule is defined to take the following form:

Rule k: if (x_1 is A_{1k}) $\cdots$ AND (x_i is A_{ik}) $\cdots$ AND (x_{N_x} is $A_{N_x k}$), then ($\hat{y}$ is β_k)
where $\beta_k = q_{k0} + q_{k1}x_1 + \cdots + q_{kN_x}x_{N_x} (k = 1, 2, ..., N_h)$ is the consequence, N_h is the number of fuzzy rules. The membership function A_{ik} corresponding to the ith input variable x_i in rule k is computed from a Gaussian function and represented as,

$$A_{ik}(x_i) = \exp\left(-\frac{(x_i - \mu_{ik})^2}{\sigma_k^2}\right) \tag{1}$$

where μ_{ik} is the Gaussian function center, σ_k is the Gaussian function width. According to the sum-product composition, the firing strength (if part) $R_k(\mathbf{x})$ is expressed as follows,

$$R_k(\mathbf{x}) = \prod_{i=1}^{N_x} A_{ik}(x_i) = \exp\left(-\frac{||\mathbf{x} - \boldsymbol{\mu}_k||^2}{\sigma_k^2}\right) \tag{2}$$

The TS-type consequence as a linear model with exogenous inputs is expressed as,

$$\beta_k = \mathbf{x}_e^T \mathbf{q}_k \tag{3}$$

where $\mathbf{x}_e$ is an extended input vector with 1, i.e., $[1, \mathbf{x}^T]^T$; $\mathbf{q}_k$ is the parameter matrix of rule k that equals as,

$$\mathbf{q}_k = [q_{k0}, \cdots, q_{kN_x}]]^T_{(N_x+1)\times 1} \tag{4}$$

With the weighted average defuzzification technique, the system output is calculated as,

$$\hat{y} = \frac{\sum_{k=1}^{N_h} R_k(\mathbf{x})\beta_k}{\sum_{k=1}^{N_h} R_k(\mathbf{x})} \tag{5}$$

We rewrite the network output (5) in a matrix form as follows:

$$\hat{y} = \mathbf{H}^T \mathbf{a} \tag{6}$$

where $\mathbf{H}$ is the normalized firing strength of rules and $\mathbf{a}$ is the parameter matrix of all the existing rules, which are given as,

$$\mathbf{a} = \left[\begin{array}{ccccccc} q_{10} & \dots & q_{1N_x} & \dots & q_{N_h 0} & \dots & q_{N_h N_x} \end{array}\right]^T_{N_h(N_x+1)\times 1} \tag{7}$$

and

$$\mathbf{H} = \left[\mathbf{x}_e^T \frac{R_1(\mathbf{x})}{\sum_{k=1}^{N_h} R_k(\mathbf{x})}, \cdots, \mathbf{x}_e^T \frac{R_{N_h}(\mathbf{x})}{\sum_{k=1}^{N_h} R_k(\mathbf{x})}\right]^T_{N_h(N_x+1)\times 1} \tag{8}$$

The design of the proposed CEFNS consists of two subtasks [9]. One is the structure identification with the determination of the antecedent part of the rules, such as the centers ($\boldsymbol{\mu}$) and widths ($\boldsymbol{\sigma}$) of the Gaussian membership functions and the number of rules (N_h). The other is the adjustment of the consequent parameters ($\mathbf{a}$) for the rules determined within the structure identification sub-task. In this paper, the two subtasks are realized based on the correntropy concept of the ITL and described below in details.

III. Learning Algorithm of CEFNS

In this section we present the learning algorithm of the CEFNS and describe how correntropy is used to evolve its structure and learn its parameters.

A. Correntropy

As a similarity measure between two random variables y and $\hat{y}$, the correntropy is generally defined as [27],

$$V(y, \hat{y}) = \mathbf{E}[\kappa(y - \hat{y})] \tag{9}$$

where $\mathbf{E}[\cdot]$ denotes mathematical expectation, $\kappa(\cdot)$ is a shift-invariant Mercer kernel. In reality, the joint distribution function of y and $\hat{y}$ is unknown. However for a finite number of samples $(y_n, \mathbf{y}_n)_{n=1}^N$, the estimator of correntropy is calculated as [27],

$$\hat{V}(y, \hat{y}) = \frac{1}{N} \sum_{n=1}^{N} \kappa(y_n - \hat{y}_n) \tag{10}$$

The Mercer kernel $\kappa(\cdot)$ in the correntropy transforms the data y from the original space to an infinite dimensional reproducing kernel Hilbert space (RKHS) $\mathbf{F}$ by means of a nonlinear mapping Φ. Thus, we obtain

$$\kappa(y_n - \hat{y}_n) = \langle \Phi(y_n), \Phi(\hat{y}_n) \rangle_{\mathbf{F}} \tag{11}$$

where $\langle \cdot, \cdot \rangle_{\mathbf{F}}$ is the inner product in $\mathbf{F}$. Two new vectors $\bar{y} = [\Phi(y_1), \Phi(y_2), \cdots, \Phi(y_N)]^T$ and $\bar{\hat{y}} = [\Phi(\hat{y}_1), \Phi(\hat{y}_2), \cdots, \Phi(\hat{y}_N)]^T$ are constructed in the Hilbert Space $\mathbf{F}$. Then their Euclidean distance ED$(\bar{y}, \bar{\hat{y}})$ can be represented as follows

$$\begin{aligned} & ED(\bar{y}, \bar{\hat{y}}) = (\langle (\bar{y} - \bar{\hat{y}}), (\bar{y} - \bar{\hat{y}}) \rangle)^{\frac{1}{2}} \\ & = \left(\langle \bar{y}, \bar{y} \rangle - 2 \langle \bar{y}, \bar{\hat{y}} \rangle + \langle \bar{\hat{y}}, \bar{\hat{y}} \rangle \right)^{\frac{1}{2}} \\ & = \left(\sum_{n=1}^{N} \kappa(y_n - y_n) - 2 \sum_{n=1}^{N} \kappa(y_n - \hat{y}_n) + \sum_{n=1}^{N} \kappa(\hat{y}_n - \hat{y}_n) \right)^{\frac{1}{2}} \\ & = \left\{ 2N \cdot [\kappa(0) - \hat{V}(y, \hat{y})] \right\}^{\frac{1}{2}} \end{aligned} \tag{12}$$

$ED(\bar{y}, \bar{\hat{y}})$ is an L2-norm in the Hilbert space. For translation invariant kernels like the Gaussian kernel, $V(y, \hat{y}) = \hat{V}(y - \hat{y})$. It can be shown that correntropy presents a nonlinear distance measure in the original error space. Also, Gaussian kernel has the smooth and strictly positive definite properties and has been widely used as the kernel function in correntropy. In this work, without mentioned otherwise, the kernel function in correntropy is Gaussian kernel, which is given as,

$$\kappa(y_n - \hat{y}_n) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(y_n - \hat{y}_n)^2}{2\sigma^2}) \tag{13}$$

where $\sigma > 0$ is the kernel width. Thus (10) becomes as,

$$\begin{aligned} J = \hat{V}(y, \hat{y}) &= \frac{1}{N} \sum_{n=1}^{N} \kappa(y_n - \hat{y}_n) \\ &= \frac{1}{\sqrt{2\pi}\sigma N} \sum_{n=1}^{N} \exp\left(-\frac{(y_n - \hat{y}_n)^2}{2\sigma^2}\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma N} \sum_{n=1}^{N} \exp\left(-\frac{e_n^2}{2\sigma^2}\right) \end{aligned} \tag{14}$$

where $e_n = y_n - \hat{y}_n$ is the error between the desired and the actual outputs. The modeling of the CEFNS is a continuous online learning process and thus the correntropy information at each time is necessary to build the structure of rule-based models and adapt the consequent parameters of rules. At the nth instant, the correntropy is written as

$$\begin{aligned} J_n =& \hat{V}(y_n, \hat{y}_n) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_n - \hat{y}_n)^2}{2\sigma^2}\right) \\ =& \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{e_n^2}{2\sigma^2}\right) \end{aligned} \tag{15}$$

Correntropy represents a nonlinear measure of similarity between the desired and actual outputs in kernel space. Taking Taylor series expansion of the exponential function yields,

$$\begin{aligned} V(y, \hat{y}) &= \frac{1}{\sqrt{2\pi}\sigma} \mathbf{E}\left[\exp\left(-\frac{(y - \hat{y})^2}{2\sigma^2}\right)\right] \\ &= \frac{1}{\sqrt{2\pi}\sigma} \sum_{n=0}^{\infty} \frac{(-1)^n}{2^n n!} \mathbf{E}\left[\frac{(y - \hat{y})^{2n}}{\sigma^{2n}}\right] \end{aligned}$$

As it can be seen, the correntropy is able to extract the higher (even) order statistics of the data. Hence, the correntropy can exhibit better performance than its linear counterparts, which has been verified in many applications and will be further confirmed in our study.

B. Rules Evolving

The evolution of rules includes adding new rules online to the fuzzy model structure. It improves the fuzzy model performance and adapts its structure. In some existing literature, the conditions of adding new rules are based on the model output error and the distance of the current sample to the existing rules [7], [8], and the information potential that depends on the current sample's spatial proximity to all other data points [2], [9]–[11]. In this paper, the correntropy in (15) together with the distance between the current sample to the existing rules is used as the criteria to determine whether a rule needs to be generated. Rule adding of the CEFNS is presented as follows. The CEFNS begins with an empty fuzzy rule base. When the first input-output data $(\mathbf{x}_1, y_1)$ sample comes in, the first new rule is generated with the mean and center of the new fuzzy set and its consequent parameters assigned by

$$\boldsymbol{\mu}_1 = \mathbf{x}_1, \ \sigma_1 = \varkappa\|\mathbf{x}_1\|, \ q_{10} = y_1, \ q_{1i} = 0, \ i = 1, 2, ..., N_x \tag{16}$$

For each subsequently incoming input-output $\mathbf{x}_n, y_n$ ($n > 1$ is the time index), fuzzy rules are added by judging

$$||\mathbf{x}_n - \boldsymbol{\mu}_{nr}|| > \theta_n \ \& \ \underline{e} < J_n = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{e_n^2}{2\sigma^2}) < \bar{e} \tag{17}$$

where $\underline{e}$ and $\bar{e}$ are the lower and upper thresholds of the rule addition chosen according to the desired accuracy of CEFNS. $\boldsymbol{\mu}_{nr}$ is the center of the fuzzy rule which is closest to $\mathbf{x}_n$. θ_n is the distance threshold which decays exponentially and is given as,

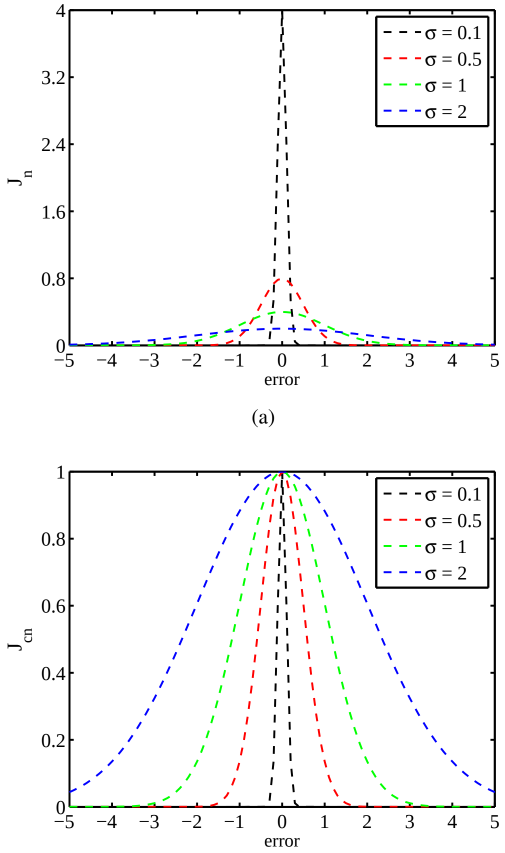
$$\theta_n = \max\{\theta_{\max} \times \gamma^n, \theta_{\min}\} \tag{18}$$

where $\theta_{\max}, \theta_{\min}$ are, respectively, the largest and smallest length of interest and γ is the decay constant. The exponential decay of θ_n initially brings the coarse learning with fewer fuzzy rules and at a later stage brings the fine learning with more fuzzy rules. Once the system generates a new fuzzy rule, the number of the fuzzy rules is set to $N_h = N_h + 1$. For the newly added rules, the antecedent parameters ($\boldsymbol{\mu}_{N_h+1}$, σ_{N_h+1}) and the initial consequent parameters are allocated as follows:

$$\begin{aligned} & q_{(N_h+1)0} = e_n, \ q_{(N_h+1)i} = 0, \ i = 1, 2, ..., N_x \\ & \boldsymbol{\mu}_{N_h+1} = \mathbf{x}_n, \ \sigma_{N_h+1} = \varkappa||\mathbf{x}_n - \boldsymbol{\mu}_{nr}|| \end{aligned} \tag{19}$$

(16) shows that the antecedent parameters are evolved from scratch in an evolvable way. For any new sample, a new rule is recruited when the criteria of adding rule are satisfied and then its antecedent parameters are determined with the newly loaded sample based on (19). In this way, the antecedent parameters are permanently updated whenever a new rule is generated.

In (17), the distance criterion ensures that the new rule is added at a sufficient distance from all existing rules while the correntropy guarantees the similarity between the desired and real outputs. Compared with the commonly used error criterion [7], [8], the correntropy criterion presents a local similarity within a small region that is determined by the kernel width σ. Fig. 1 depicts the curves of correntropy under different kernel widths, from which it can be seen that the behavior of correntropy is varied under different kernel width σ and also the correntropy is quite small when the error is large. Thus, the lower threshold of the correntropy is set to resist the effect of the noise in which the correntropy is small due to the large error. The upper threshold of the correntropy ensures a high similarity between the desired and real outputs. In this study, a new rule is deemed to be created when the similarity between the desired and real outputs is within the interval $[\underline{e}, \bar{e}]$ and the distance between the current sample and the existing rules is farther than the threshold θ_n.



(a)

(b)

Fig. 1. The correntropy J_n and J_{c_n} under different kernel widths σ.

From Fig. 1(a) one can see that the maximum value of correntropy J_n varies too much under different kernel width values, thus making the growing thresholds $(\underline{e}, \bar{e})$ vary a lot and are hard to choose. Defining $e_k = \sqrt{2\pi}\sigma\underline{e}$ and $e_g = \sqrt{2\pi}\sigma\bar{e}$, the correntropy criterion becomes $e_k < J_{c_n} = \exp(-\frac{\|\mathbf{e}_n\|^2}{2\sigma^2}) < e_g$. In this context, the maximum value of J_{c_n} lies around 1 with different kernel widths, as shown in Fig. 1(b). It is easy to choose the parameters (e_k, e_g) during a practical implementation. From Fig. 1(b), it can be found that the bigger kernel width yields bigger value of J_{c_n} under the same error value. When the kernel width chosen is too small, the value of J_{c_n} tends to zero for most errors. Consequently, we choose $\sigma > 0.5$ in our study for all the approximation problems.

C. Parameter Updating

In the CEFNS, the antecedent parameters of the rules are determined according to (19) in an evolvable way. When there is no addition of rules, the consequent parameters $\mathbf{a}$ from (6) are updated. In the existing work, estimation of the consequent parameters is transformed into a least square problem and updated recursively based on the MSE criterion. Here, the MCC is used as the performance function to adjust the consequent parameters ($\mathbf{a}$) of the CEFNS. In MCC, the system output is ensured to be close to the desired response through maximizing the error probability density at the origin. Also, the MCC adaptation can be applied in any noise environments where its maximum value is at the origin [28]. Thus, the MCC being a local criterion of similarity has the advantages over MSE and very effective to handle nonzero mean and non-Gaussian cases with large outliers.

Accordingly, for the given input-output dataset $(\mathbf{x}_n, y_n)$, $n = [1, N]$, the cost function based on the correntropy is expressed as,

$$J_n = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y_n - \mathbf{H}_n^T\mathbf{a}_n)^2}{2\sigma^2}\right) \tag{20}$$

where $\mathbf{H}_n = \left[\mathbf{x}_{en}^T\frac{R_1(\mathbf{x}_n)}{\sum_{k=1}^{N_h}R_k(\mathbf{x}_n)}, \cdots, \mathbf{x}_{en}^T\frac{R_{N_h}(\mathbf{x}_n)}{\sum_{k=1}^{N_h}R_k(\mathbf{x}_n)}\right]^T$, $\mathbf{x}_{en} = [1, \mathbf{x}_n]^T$. Under the MCC, the optimal consequent parameter $\mathbf{a}_n$ is recursively updated using the gradient ascent approach in order to maximize the cost function J_n. With a small step $\delta > 0$, the parameter is updated along the positive gradient of the performance function J_n in the parameter space and expressed as,

$$\mathbf{a}_{n+1} = \mathbf{a}_n + \delta\nabla J_n \tag{21}$$

where

$$\nabla J_n = \frac{\partial J_n}{\partial \mathbf{a}_n} = \left\{\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y_n - \mathbf{H}_n^T\mathbf{a}_n)^2}{2\sigma^2}\right)\right\}'$$

$$= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y_n - \mathbf{H}_n^T\mathbf{a}_n}{2\sigma^2}\right)\left(-\frac{(y_n - \mathbf{H}_n^T\mathbf{a}_n)^2}{2\sigma^2}\right)'$$
$$= \frac{1}{\sqrt{2\pi}\sigma^3}\exp\left(-\frac{(y_n - \mathbf{H}_n^T\mathbf{a}_n)^2}{2\sigma^2}\right)(y_n - \mathbf{H}_n^T\mathbf{a}_n)\mathbf{H}_n \tag{22}$$

According to (22), the updating equation of the consequent parameter at the next time instant $n+1$ equals to,

$$\mathbf{a}_{n+1} = \mathbf{a}_n + \frac{\delta}{\sqrt{2\pi}\sigma^3}\exp\left(-\frac{(y_n - \mathbf{H}_n\mathbf{a}_n)^2}{2\sigma^2}\right)(y_n - \mathbf{H}_n\mathbf{a}_n)\mathbf{H}_n \tag{23}$$

Denoting $\eta = \frac{\delta}{\sqrt{2\pi}\sigma^3 N} > 0$, the above parameter updating equation can be simplified as,

$$\mathbf{a}_{n+1} = \mathbf{a}_n + \eta\exp\left(-\frac{e_n^2}{2\sigma^2}\right)e_n\mathbf{H}_n = \mathbf{a}_n + \eta J_{c_n}e_n\mathbf{H}_n \tag{24}$$

where $e_n = y_n - \mathbf{H}_n\mathbf{a}_n$ is the estimation error, and η is the step size. (24) is called the MCC algorithm [34]. When $J_{c_n} = \exp\left(-\frac{e_n^2}{2\sigma^2}\right) = 1$ in (24), the MCC algorithm reduces to the least mean square (LMS) adaption algorithm which uses MSE criterion as its cost function,

$$\mathbf{a}_{n+1} = \mathbf{a}_n + \eta e_n\mathbf{H}_n \tag{25}$$

However, comparing (24) and (25), it can be observed that the MCC algorithm obtains an extra scaling factor that is also the correntropy criterion function J_{c_n}. The extra scaling factor $J_{c_n} = \exp(-\frac{e_n^2}{2\sigma^2})$ is an exponential function of the error e_n and depicts the outlier rejection property of the correntropy similarity measure. But for the learning algorithms like RLS and LMS based on the MSE criterion, the parameter update is related to the scalar factor e_n. It is obvious to see that $\eta(i) \leq \eta$ and thus the convergence speed of the MCC is reduced accordingly. However, when the kernel width is not too small, the effect of the convergence speed of the parameters in MCC is relatively weak. Fig. 2 depicts the curves of the two scalar factors existing in the MCC and MSE. As shown in Fig. 2 it can be seen that the scalar factor $J_{c_n}e_n$ in the MCC varies much less than the scalar factor e_n in the MSE criterion especially when the error e_n has a large variation. This means that the parameter adaption using the MSE is easy to be influenced seriously by a large error e_n that may be caused by outliers or impulsive noise, whereas parameter adaptation using MCC is more stable when the desired outputs have strong outliers or impulsive characteristics.

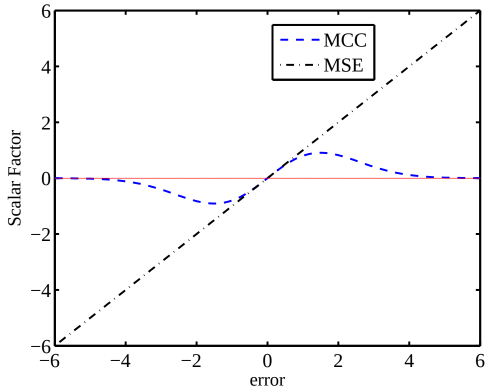


Fig. 2. The curves of two scalar factors $(J_{c_n}e_n, e_n)$ with different error values.

The learning algorithm of the proposed CEFNS is summarized in **Algorithm 1**.

IV. Steady-State Mean Square Convergence of CEFNS

The convergence performance is the key aspect of any learning algorithm. In this section, the steady-state convergence performance is evaluated for the proposed CEFNS algorithm. The evaluation is done by calculating its steady-state excess mean square error (EMSE).

A. EMSE of CEFNS

Suppose the desired output y_n with noise ν_n at time n can be represented as

$$y_n = \mathbf{H}_n^T\mathbf{a}_0 + \nu_n \tag{26}$$

Algorithm 1 CEFNS Learning Algorithm

```
Sample: $\mathbf{x}_n \in \mathbb{R}^{N_x}$, $y_n \in \mathbb{R}$
Initialize: setting $N_h = 0$ and giving the growing thresholds $\theta_n, e_k, e_g$
1. Calculate the system output and error
       $\hat{y}_n = \mathbf{H}_n^T \mathbf{a}_n,\ e_n = y_n - \hat{y}_n$
2. Compute the distance parameter
       $\theta_n = \max\{\theta_{\max} \times \gamma^n, \theta_{\min}\}, (0 < \gamma < 1)$
3. Apply the rule addition criteria
   if $||\mathbf{x}_n - \boldsymbol{\mu}_{nr}|| > \theta_n$ and $e_k < J_{c_n} < e_g$ then
       assign a new fuzzy rule $N_h + 1$ with
           $q_{(N_h+1)0,j} = e_{nj},\ q_{(N_h+1)i,j} = 0$
           $i = 1, 2, ..., N_x,\ j = 1, 2, ..., N_y$
           $\boldsymbol{\mu}_{N_h+1} = \mathbf{x}_n, \sigma_{N_h+1} = \varkappa ||\mathbf{x}_n - \boldsymbol{\mu}_{nr}||$
   else
       adjust the consequent parameters using the MCC algorithm $\mathbf{a}_{n+1} = \mathbf{a}_n + \eta \cdot J_{c_n} e_n \cdot \mathbf{H}_n$
   end if
```

where $\mathbf{a}_0$ is the desired unknown parameter and needs to be estimated, ν_n is the disturbance noise with variance σ_v^2. Substituting (26) into the estimation error $e_n = y_n - \mathbf{H}_n^T \mathbf{a}_n$ we have

$$e_n = \mathbf{H}_n^T \tilde{\mathbf{a}}_n + \nu_n = e_{an} + \nu_n \tag{27}$$

where $\mathbf{a}_n$ is the estimate of $\mathbf{a}_0$ at time n, $\tilde{\mathbf{a}}_n = \mathbf{a}_0 - \mathbf{a}_n$ is the parameter error vector at time n. $e_{an} = \mathbf{H}_n^T \tilde{\mathbf{a}}_n$ is the so-called a priori error, which is also called the error signal due to parameter mismatch. Its mean square of the priori error is $E[e_{an}^2]$, which is also named the EMSE. It is a popular performance evaluation measure for the learning algorithms.

Substituting $\tilde{\mathbf{a}}_{n+1} = \mathbf{a}_0 - \mathbf{a}_{n+1}$ and $\tilde{\mathbf{a}}_n = \mathbf{a}_0 - \mathbf{a}_n$ into (24) yields

$$\tilde{\mathbf{a}}_{n+1} = \tilde{\mathbf{a}}_n - \eta f(e_n) \mathbf{H}_n \tag{28}$$

where $f(e_n) = \exp\left(-\frac{e_n^2}{2\sigma^2}\right) e_n$ is the scalar function of the error e_n. Multiplying both sides of (28) by $\mathbf{H}_n^T$ yields

$$\mathbf{H}_n^T \tilde{\mathbf{a}}_{n+1} = \mathbf{H}_n^T \tilde{\mathbf{a}}_n - \mathbf{H}_n^T \eta f(e_n) \mathbf{H}_n \tag{29}$$

Define $e_{p_n} = \mathbf{H}_n^T \tilde{\mathbf{a}}_{n+1}$ as a posteriori error. Substituting $e_{an} = \mathbf{H}_n^T \tilde{\mathbf{a}}_n$ into (29) obtains

$$\eta f(e_n) = \frac{e_{p_n} - e_{an}}{\|\mathbf{H}_n\|^2} \tag{30}$$

According to (30), we get

$$\tilde{\mathbf{a}}_{n+1} = \tilde{\mathbf{a}}_n - \frac{e_{p_n} - e_{an}}{\|\mathbf{H}_n\|^2} \mathbf{H}_n \tag{31}$$

Squaring both sides of (31) is equal as,

$$\begin{aligned}\|\tilde{\mathbf{a}}_{n+1}\|^2 &= (\tilde{\mathbf{a}}_n - \frac{e_{p_n} - e_{an}}{\|\mathbf{H}_n\|^2} \cdot \mathbf{H}_n)^T (\tilde{\mathbf{a}}_n - \frac{e_{p_n} - e_{an}}{\|\mathbf{H}_n\|^2} \cdot \mathbf{H}_n) \\ &= \|\tilde{\mathbf{a}}_n\|^2 + \frac{|e_{p_n}|^2 - |e_{an}|^2}{\|\mathbf{H}_n\|^2} \end{aligned} \tag{32}$$

Taking the expected values of (32), one gets

$$\begin{aligned}&E\left[\|\tilde{\mathbf{a}}_{n+1}\|^2\right] \\ =&E\left[\|\tilde{\mathbf{a}}_n\|^2\right] + E\left[\frac{(e_{an} - \eta \cdot f(e_n) \cdot \|\mathbf{H}_n\|^2)^2 - |e_{an}|^2}{\|\mathbf{H}_n\|^2}\right] \\ =&E\left[\|\tilde{\mathbf{a}}_n\|^2\right] + \eta^2 E\left[f^2(e_n)\|\mathbf{H}_n\|^2\right] - 2\eta E\left[f(e_n) e_{an}\right] \end{aligned} \tag{33}$$

Next, we focus on deriving the theoretical value of the steady-state EMSE defined as $S = \lim_{n\to\infty} E[e_{an}^2]$. When the CEFNS algorithm arrives at the steady state, the parameter error satisfies

$$\lim_{n\to\infty} E\left[\|\tilde{\mathbf{a}}_{n+1}\|^2\right] = \lim_{n\to\infty} E\left[\|\tilde{\mathbf{a}}_n\|^2\right] \tag{34}$$

Then (33) becomes

$$2 \lim_{n\to\infty} E\left[f(e_n) e_{an}\right] - \eta Tr(R_{\mathbf{HH}}) \lim_{n\to\infty} E\left[f^2(e_n)\right] = 0 \tag{35}$$

where $\|\mathbf{H}_n\|^2$ is assumed to be asymptotically uncorrelated with $f^2(e_n)$ and $R_{\mathbf{HH}} = E\left[\mathbf{H}_n \mathbf{H}_n^T\right]$ equals the correlation matrix of the vector $\mathbf{H}_n$. $Tr(\cdot)$ denotes the trace operator.

In the following, two different kinds of noise cases are considered to derive the theoretical value of steady-state EMSE S.

1) Case of Gaussian Noise: The noise ν_n is assumed to be Gaussian with mean zero and variance σ_ν^2. Also the input-output datastreams is assumed to be long enough such that e_{an} is Gaussian. According to Price theorem [35] [36], one gets

$$\begin{aligned}&\lim_{n\to\infty} E\left[f(e_n) e_{an}\right] = \lim_{n\to\infty} E\left[{e_{an}}^2\right] E\left[f'(e_n)\right] \\ =&\frac{S}{\sqrt{2\pi}\sigma_e} \lim_{n\to\infty} \int_{-\infty}^{\infty} \exp\left(-\frac{e_n^2}{2\sigma^2}\right)\left(1 - \frac{e_n^2}{\sigma^2}\right) \\ &\cdot \exp\left(-\frac{e_n^2}{2\sigma_e^2}\right) de_n \\ =&\frac{\sigma^3 S}{(\sigma^2 + \sigma_\nu^2 + S)^{\frac{3}{2}}} \end{aligned} \tag{36}$$

where $\sigma_e = E\left[{e_{an}}^2\right] + \sigma_\nu^2$ is the variance of error e_n. Similarly,

$$\lim_{n\to\infty} E\left[f^2(e_{an} + \nu_n)\right] = \frac{\sigma^3(S + \sigma_\nu^2)}{(2\sigma_\nu^2 + \sigma^2 + 2S)^{\frac{3}{2}}} \tag{37}$$

Hence, the theoretical value of the steady-state EMSE S under the Gaussian noise is obtained according to the fixed-point equation given as

$$S = \frac{1}{2} \eta Tr(R_{\mathbf{HH}}) \frac{(S + \sigma_\nu^2)(\sigma^2 + \sigma_\nu^2 + S)^{\frac{3}{2}}}{(2\sigma_\nu^2 + \sigma^2 + 2S)^{\frac{3}{2}}} \tag{38}$$

The steady-state EMSE value of S can be achieved by solving the above fixed-point equation. (38) has only one fixed point whose value is real and positive. This has been verified in the following simulation studies.

2) Case of Non-Gaussian Noise: When the noise ν_n is non-Gaussian, the theoretical value of steady-state EMSE S is derived using the Taylor series expansion of the function $f(e_n)$ [37]. At the steady-state, the distributions of e_{an} and

e_n are unrelated to n and, thus, the time index is omitted for the purpose of brevity. (35) is rewritten as,

$$2E\left[f(e)e_a\right] - \eta Tr(R_{\mathbf{HH}})E\left[f^2(e)\right] = 0 \tag{39}$$

The Taylor expansion of $f(e_n)$ in relation to e_{an} around ν_n is given as,

$$f(e) = f(e_a+\nu) = f(\nu)+f'(\nu)e_a+\frac{1}{2}f''(\nu)e_a^2+o(e_a^2) \tag{40}$$

where $o(e_a^2)$ is the third and higher-order term, and

$$f'(\nu) = \exp\left(-\frac{\nu^2}{2\sigma^2}\right)\left(1-\frac{\nu^2}{\sigma^2}\right) \tag{41}$$

$$f''(\nu) = \exp\left(-\frac{\nu^2}{2\sigma^2}\right)\left(\frac{\nu^3}{\sigma^4}-\frac{3\nu}{\sigma^2}\right) \tag{42}$$

According to (40), one gets

$$E[e_af(e)] = E[e_af(\nu)+f'(\nu)e_a^2+o(e_a^2)] \approx E[f'(\nu)]S \tag{43}$$

and

$$E[f^2(e)] \approx E[f^2(\nu)] + E[f(\nu)f''(\nu)+|f'(\nu)|^2]S \tag{44}$$

Then, substituting (43) and (44) into (39), yields

$$S = \frac{\eta Tr(R_{\mathbf{HH}})\alpha}{2\beta - \eta Tr(R_{\mathbf{HH}})\gamma} \tag{45}$$

where $E[o(e_a^2)]$ is assumed to be small enough and omitted here.

$$\alpha = E\left[\exp(-\frac{\nu^2}{\sigma^2})\nu^2\right], \beta = E\left[\exp(-\frac{\nu^2}{\sigma^2})(1-\frac{\nu^2}{\sigma^2})\right]$$

$$\gamma = E\left[\exp(-\frac{\nu^2}{\sigma^2})(1+\frac{2\nu^4}{\sigma^4}-\frac{5\nu^2}{\sigma^2})\right]$$

From (45), it is clear that the EMSE S is related with the distribution of the noise v. When the distribution is given, the theoretical value of the steady-state EMSE in (45) can be obtained accordingly. But the steady-state EMSE of (45) is derived under the assumption that the steady-state *a priori* error e_a is small enough to ignore its third and higher-order terms. This can be ensured under the circumstances of small step size and noise power. Besides, from (38) and (45), one can note that the calculation of the steady-state EMSE S is directly related with $Tr(R_{\mathbf{HH}})$. In order to achieve this, the analytical expression of $Tr(R_{\mathbf{HH}})$ is shown in the following.

B. The evaluation of $Tr(R_{\mathbf{HH}})$

The vector $\mathbf{H}_n$ in (8) is further rewritten as

$$\begin{aligned}\mathbf{H}_n &= \frac{1}{\lambda}\left[\mathbf{x}_e^TR_1(\mathbf{x}_n), \cdots, \mathbf{x}_e^TR_{N_h}(\mathbf{x}_n)\right]^T \\ &= \frac{1}{\lambda}\left[\mathbf{x}_e^T\kappa(\mathbf{x}_n,\boldsymbol{\mu}_1), \cdots, \mathbf{x}_e^T\kappa(\mathbf{x}_n,\boldsymbol{\mu}_{N_h})\right]^T\end{aligned} \tag{46}$$

where $\lambda = \sum_{k=1}^{N_h} R_k(\mathbf{x}_n)$ is a scalar assumed as a constant here. For $1 \leq i,j \leq N_h$, the (i,j)-th entry of $R_{\mathbf{HH}}$ is expressed as

$$\begin{aligned}[R_{\mathbf{HH}}]_{ij} =& \frac{1}{\lambda^2}E[\mathbf{x}_e\mathbf{x}_e^T\kappa(\mathbf{x}_n,\boldsymbol{\mu}_i)\kappa(\mathbf{x}_n,\boldsymbol{\mu}_j)] \\ =& \frac{1}{\lambda^2}\left\{\mathrm{Cov}(\mathbf{x}_\mathrm{e}^\mathrm{T}\mathbf{x}_\mathrm{e}, \kappa(\mathbf{x}_\mathrm{n},\boldsymbol{\mu}_\mathrm{i})\kappa(\mathbf{x}_\mathrm{n},\boldsymbol{\mu}_\mathrm{j}))\right. \\ &\left. + E[\mathbf{x}_e^T\mathbf{x}_e]E[\kappa(\mathbf{x}_n,\boldsymbol{\mu}_i)\kappa(\mathbf{x}_n,\boldsymbol{\mu}_j)]\right\}\end{aligned} \tag{47}$$

where $\mathrm{Cov}(\mathbf{x}_\mathrm{e}^\mathrm{T}\mathbf{x}_\mathrm{e}, \kappa(\mathbf{x}_\mathrm{n},\boldsymbol{\mu}_\mathrm{i})\kappa(\mathbf{x}_\mathrm{n},\boldsymbol{\mu}_\mathrm{j}))$ is the covariance of $\mathbf{x}_e^T\mathbf{x}_e$ and $\kappa(\mathbf{x}_n,\boldsymbol{\mu}_i)\kappa(\mathbf{x}_n,\boldsymbol{\mu}_j)$. Here, it is assumed that $\mathbf{x}_e^T\mathbf{x}_e$ is nearly uncorrelated with $\kappa(\mathbf{x}_n,\boldsymbol{\mu}_i)\kappa(\mathbf{x}_n,\boldsymbol{\mu}_j)$. This assumption we introduced here has the same spirit as the independence assumption but it is weaker. And, it is restricted to the variance of $\mathbf{x}$ not being very large, which will be discussed at great length in the following subsection. Under this assumption, then (47) becomes

$$[R_{\mathbf{HH}}]_{ij} = \frac{1}{\lambda^2}R_{\mathbf{x}_e\mathbf{x}_e}[R_{\boldsymbol{\kappa\kappa}}]_{ij} \tag{48}$$

where $[R_{\boldsymbol{\kappa\kappa}}]_{ij} = E[\kappa(\mathbf{x}_n,\boldsymbol{\mu}_i)\kappa(\mathbf{x}_n,\boldsymbol{\mu}_j)]$ and $R_{\mathbf{x}_e\mathbf{x}_e} = E\left[\mathbf{x}_e\mathbf{x}_e^T\right]$.

Let $\boldsymbol{z} = [z_1, ..., z_n]^T$ be a Gaussian random vector with $E[\boldsymbol{z}] = 0$ and $E[\boldsymbol{z}\boldsymbol{z}^T] = R_{\boldsymbol{zz}}$. Let $Q = \boldsymbol{z}^T\boldsymbol{W}\boldsymbol{z} + \boldsymbol{b}^T\boldsymbol{z}$ be the quadratic form of $\boldsymbol{z}$ where $\boldsymbol{W}$ is an $n \times n$ matrix and $\boldsymbol{b}$ is an $n \times 1$ vector. The moment generating function of Q is given by [38]:

$$\begin{aligned}&\psi_Q(s) = E[e^{sQ}] \\ &= |\boldsymbol{I} - 2s\boldsymbol{W}R_{\boldsymbol{zz}}|^{-\frac{1}{2}}\exp\left(\frac{s^2}{2}\boldsymbol{b}^TR_{\boldsymbol{zz}}(\boldsymbol{I} - 2s\boldsymbol{W}R_{\boldsymbol{zz}})^{-1}\boldsymbol{b}\right)\end{aligned} \tag{49}$$

This result will be useful to compute the $R_{\boldsymbol{\kappa\kappa}}$ whose (i,j)-th entry is given as

$$\begin{aligned}&[R_{\boldsymbol{\kappa\kappa}}]_{ij} \\ =&E\left\{\exp\left(-\frac{1}{2\xi_i^2}||\mathbf{x}_n-\boldsymbol{\mu}_i||^2 - \frac{1}{2\xi_j^2}||\mathbf{x}_n-\boldsymbol{\mu}_j||^2)\right)\right\} \\ =&E\left\{\exp\left[-\frac{\xi_i^2+\xi_j^2}{2\xi_i^2\xi_j^2}\left(||\mathbf{x}_n||^2 - \frac{2(\xi_j^2\boldsymbol{\mu}_i^T+\xi_i^2\boldsymbol{\mu}_j^T)}{\xi_i^2+\xi_j^2}\mathbf{x}_n\right)\right]\right\} \\ &\cdot\exp\left(-\frac{||\boldsymbol{\mu}_i||^2}{2\xi_i^2} - \frac{||\boldsymbol{\mu}_j||^2}{2\xi_j^2}\right)\end{aligned} \tag{50}$$

Defining $\boldsymbol{b} = -\frac{2(\xi_j^2\boldsymbol{\mu}_i+\xi_i^2\boldsymbol{\mu}_j)}{\xi_i^2+\xi_j^2}, s = -\frac{\xi_i^2+\xi_j^2}{2\xi_i^2\xi_j^2}$ and according to (49), we can obtain (51).

The correlation matrix $R_{\mathbf{x}_e\mathbf{x}_e}$ of the extended input vector is calculated as

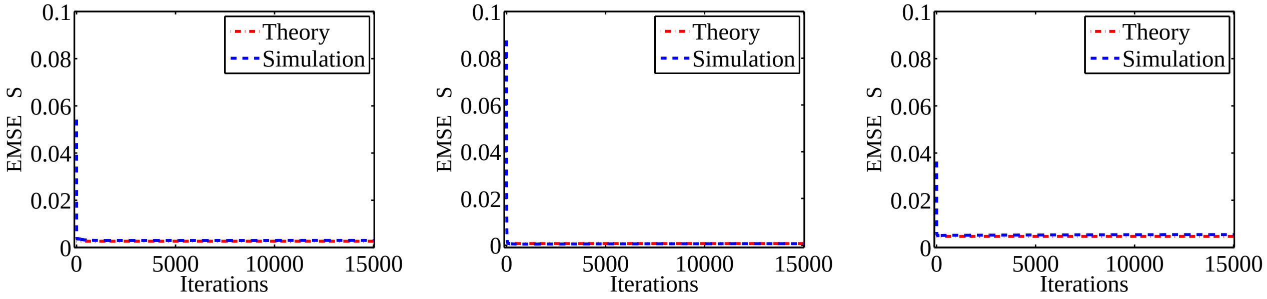
$$[R_{\mathbf{x}_e\mathbf{x}_e}] = E\begin{bmatrix} 1 & \mathbf{x}_n^T \\ \mathbf{x}_n & \mathbf{x}_n\mathbf{x}_n^T \end{bmatrix} = \begin{cases} 1, & i=j=1 \\ E[\mathbf{x}_n^T], & i=1, j\neq 1 \\ E[\mathbf{x}_n], & i\neq 1, j=1 \\ R_{\mathbf{xx}}, & i\neq 1, j\neq 1 \end{cases} \tag{52}$$

with $1 \leq i,j \leq N_x+1$.

According to (51) and (52), $Tr(R_{\mathbf{HH}})$ is obtained as

$$Tr(R_{\mathbf{HH}}) = \frac{1}{\lambda^2}\sum_{i=j=1}^{N_h\times(N_x+1)}[R_{\mathbf{HH}}]_{ij}$$

$$= \frac{1}{\lambda^2}\sum_{j=1}^{N_h}\sum_{i=1}^{N_x+1}[R_{\mathbf{x}_e\mathbf{x}_e}]_{ii}[R_{\boldsymbol{\kappa\kappa}}]_{jj}$$



(a) Gaussian noise

(b) Uniform noise

(c) Impulse noise

Fig. 3. Simulated and theoretical EMSE in different noise

$$= \frac{1}{\lambda^2}\sum_{j=1}^{N_h}\sum_{i=1}^{N_x}(1+[R_{\mathbf{xx}}]_{ii})|\boldsymbol{I}+\frac{2}{\xi_j^2}R_{\mathbf{xx}}|^{-\frac{1}{2}}$$
$$\cdot \exp\left(\frac{2}{\xi_j^4}\boldsymbol{\mu}_j^T R_{\mathbf{xx}}(\boldsymbol{I}+\frac{2}{\xi_j^2}R_{\mathbf{xx}})^{-1}\boldsymbol{\mu}_j - \frac{||\boldsymbol{\mu}_j||^2}{\xi_j^2}\right) \qquad (53)$$

C. An Illustration Example

In this subsection, an example is presented to verify the theoretical steady-state EMSE analysis of the proposed CEFNS algorithm by means of the comparisons between the simulated learning curves and analytical models under different kinds of noise. Here, a 'SinC' function is used as an example. It is given by

$$y(x) = \begin{cases} \sin(x)/x & x \neq 0 \\ 0 & x = 0 \end{cases} \qquad (54)$$

Here, the input x follows a zero-mean Gaussian random distribution with unit variance (2000 data are generated for the purpose of the training). Three kinds of noise are added to all the training samples. The first kind is the zero-mean Gaussian noise with variance 0.5. The other two kinds are non-Gaussian noise, that is the uniform noise within the interval $[-0.5, 0.5]$ and the unit-amplitude impulse noise that is imposed at an interval of 10 samples.

TABLE I
SIMULATED AND THEORETICAL EMSE IN DIFFERENT NOISE

Noise Distribution	Parameters Setting	Rules	Theoretical	Simulation
Gaussian	$\eta = 0.005, \sigma = 1.5$	14	0.0026	0.0030
Uniform	$\eta = 0.001, \sigma = 1.5$	16	0.00068	0.00066
Impulse	$\eta = 0.005, \sigma = 1$	14	0.0047	0.0055

In order to demonstrate arrival at the steady-state, 15000 iterations are run under each kind of noise. The steady-state EMSE values under the three different kinds of noise are depicted in Fig. 3, where the theory and simulated values are illustrated by dot and dash-dot lines, respectively. Table I shows the EMSE simulated values achieved at the final iteration and the theoretical values under the three kinds of noise. From the results, it can be observed that the simulated steady-state EMSE values concur the theoretically expected values calculated according to (38) under the Gaussian noise and (45) under the non-Gaussian noise very well. This further experimentally validates the steady-state mean square convergence of the proposed correntropy-EFNS.

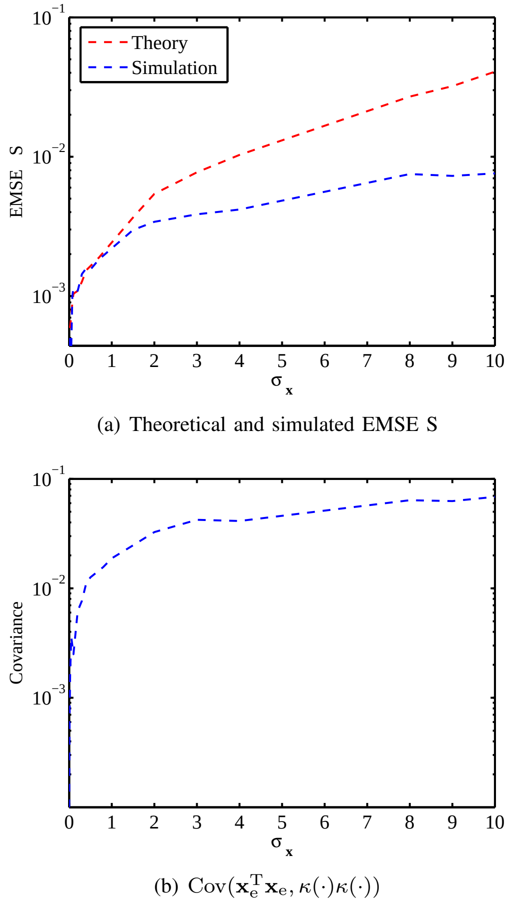
To demonstrate the reasonability of this uncorrelated assumption, we use the example and the same data set for simulation. Fig. 4(a) shows the simulated EMSE values and the theoretical ones in the Gaussian noise and different covariances of the input x. Also the covariances of $\mathbf{x}_e^T\mathbf{x}_e$ and $\kappa(\mathbf{x}_n,\boldsymbol{\mu}_i)\kappa(\mathbf{x}_n,\boldsymbol{\mu}_j)$ are presented in Fig. 4(b). From Fig.4(a), one can see that the simulated EMSE values calculated based on the assumption match the theoretical ones very well when the variance $\sigma_{\mathbf{x}}$ of the input is within 1. However, as $\sigma_{\mathbf{x}}$ grows, the error between the simulated and the theoretical EMSE accordingly increases. This is mainly because the covariance of $\mathbf{x}_e^T\mathbf{x}_e$ and $\kappa(\mathbf{x}_n,\boldsymbol{\mu}_i)\kappa(\mathbf{x}_n,\boldsymbol{\mu}_j)$ is getting bigger along with $\sigma_{\mathbf{x}}$ increasing. Table II shows these results in different covariances $\sigma_{\mathbf{x}}$. As shown in the table, $\mathrm{Cov}(\mathbf{x}_\mathrm{e}^\mathrm{T}\mathbf{x}_\mathrm{e}, \kappa(\mathbf{x}_\mathrm{n},\boldsymbol{\mu}_\mathrm{i})\kappa(\mathbf{x}_\mathrm{n},\boldsymbol{\mu}_\mathrm{j}))$ almost equals to zero when $\sigma_{\mathbf{x}}$ is very small. Based on these, it is assumed that $\mathbf{x}_e^T\mathbf{x}_e$ is nearly uncorrelated with $\kappa(\mathbf{x}_n,\boldsymbol{\mu}_i)\kappa(\mathbf{x}_n,\boldsymbol{\mu}_j)$. And, this assumption is restricted to the variance of $\mathbf{x}$ which should not be very large.

Besides, the example is further used to illustrate the difference of the convergence and accuracy between MCC, LMS and LSE parameter adaption methods. It is noteworthy that only the parameter adaptation methods are different while the rule evolution maintains same. Except 2000 training data, 2000 different data are generated for the purpose of the evaluation. Fig.5(a) shows the convergence curves in terms

$$[R_{\boldsymbol{\kappa\kappa}}]_{ij} = |\boldsymbol{I}+\frac{\xi_i^2+\xi_j^2}{\xi_i^2\xi_j^2}R_{\mathbf{xx}}|^{-\frac{1}{2}}\exp\left(\frac{1}{2\xi_i^4\xi_j^4}(\xi_j^2\boldsymbol{\mu}_i^T+\xi_i^2\boldsymbol{\mu}_j^T)R_{\mathbf{xx}}(\boldsymbol{I}+\frac{\xi_i^2+\xi_j^2}{\xi_i^2\xi_j^2}R_{\mathbf{xx}})^{-1}(\xi_j^2\mathbf{x}_i+\xi_i^2\mathbf{x}_j)\right)$$
$$\cdot \exp\left(-\frac{||\boldsymbol{\mu}_i||^2}{2\xi_i^2}-\frac{||\boldsymbol{\mu}_j||^2}{2\xi_j^2}\right) \qquad (51)$$

TABLE II
SIMULATED AND THEORETICAL EMSE VALUES AND COVARIANCES $\mathrm{Cov}(\mathbf{x}_\mathrm{e}^\mathrm{T}\mathbf{x}_\mathrm{e}, \kappa(\cdot)\kappa(\cdot))$ IN DIFFERENT $\sigma_\mathbf{x}$

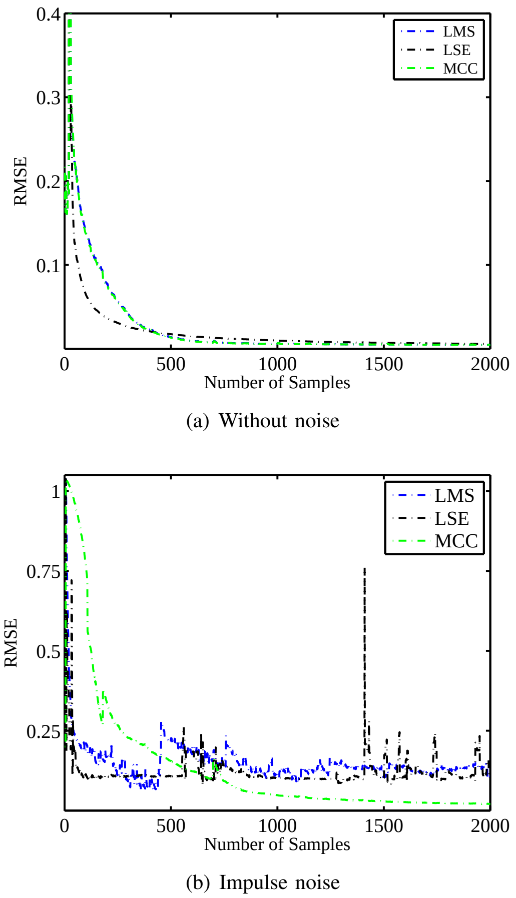
$\sigma_\mathbf{x}$	0.05	0.1	0.5	1	2	4	8
Theoretical S	0.0007	0.0010	0.0016	0.0026	0.0054	0.0103	0.0270
Simulated S	0.0006	0.0011	0.0015	0.0030	0.0034	0.0042	0.0075
$\mathrm{Cov}(\mathbf{x}_\mathrm{e}^\mathrm{T}\mathbf{x}_\mathrm{e}, \kappa(\cdot)\kappa(\cdot))$	0.0036	0.0025	0.0126	0.0188	0.0326	0.0412	0.0638



(a) Theoretical and simulated EMSE S

(b) $\mathrm{Cov}(\mathbf{x}_\mathrm{e}^\mathrm{T}\mathbf{x}_\mathrm{e}, \kappa(\cdot)\kappa(\cdot))$

Fig. 4. Simulated and theoretical EMSE values and covariances $\mathrm{Cov}(\mathbf{x}_\mathrm{e}^\mathrm{T}\mathbf{x}_\mathrm{e}, \kappa(\cdot)\kappa(\cdot))$ in different $\sigma_\mathbf{x}$



(a) Without noise

(b) Impulse noise

Fig. 5. Learning error evolution between LMS, LSE and MCC methods for Sinc example.

of the validation RMSE between the three methods without any noise. From the figure, one can find that LSE has a faster convergence speed than MCC and LMS. The MCC and LMS have very similar convergence speed. Besides, the three methods achieves similar accuracies when there are no noise. Fig.5(b) further illustrates the convergence curves under the impulse noise. From the figure, it is clear to see that the MCC has a more stable learning process and better testing accuracy than the LSE and LMS. This verifies the outlier rejection property of the MCC.

V. PERFORMANCE EVALUTION OF CEFNS

In this section, the performance of the proposed CEFNS is evaluated in details on one time-varying system identification problem, one Mackey-Glass time series prediction problem and three real-world benchmark regression problems. For these problems, CEFNS has been compared with other popular fuzzy neural system algorithms, such as SAFIS [14], ESAFIS [15], OS-Fuzzy-ELM [39], eTS [9] and Simpl_eTS [10]. All the algorithms go through the training data sequentially in a single pass and then performance comparison is made in terms of training time, training and testing accuracies and the number of rules. The root mean square error (RMSE) between the desired values without any noise and the approximated values is applied as the index of the training and testing accuracies. Also in the simulation studies, the dataset is separated into the training dataset and testing dataset. The structure and parameters of the fuzzy systems are built based on the training data. Once the training is finished, the structure and parameters are frozen and the testing performance of the algorithms is evaluated using the testing data without involving any training process. All the simulations are conducted in MATLAB 7.12.0 environment running on an ordinary PC with 3.30 GHz CPU and 4 GB RAM.

TABLE III
PERFORMANCE COMPARISON FOR TIME-VARYING NONLINEAR SYSTEM

Algorithms	Without Noise				With Impulse Noise			
	Training RMSE	Testing RMSE	Training Time	Rules	Training RMSE	Testing RMSE	Training Time	Rules
SAFIS	0.1588	0.1914	11.6689	39	0.2498	0.2710	48.5943	92
ESAFIS	**0.1478**	0.1799	21.5593	25	0.2740	0.2428	39.5775	34
OS-Fuzzy-ELM	0.1888	0.1787	5.9959	20	0.2295	0.2358	9.7953	35
eTS	0.1693	0.2104	35.5682	74	0.2516	0.3073	50.6379	106
Simpl_eTS	0.2597	0.2506	18.1429	34	0.3217	0.3335	20.6389	35
CEFNS	0.1569	**0.1736**	**1.6536**	**19**	**0.1650**	**0.1752**	**2.5272**	**34**

A. Guidelines for Parameter Selection

The performance of the proposed CEFNS depends on some parameters to be decided before learning. The predefined parameters contain the distance thresholds $(\theta_{\max}, \theta_{\min}, \gamma)$, the overlap factor (κ), the step size η, the kernel width (σ) and the growing thresholds (e_k, e_g) for a new rule. Following [14], a general selection procedure for these parameters is given as follows: $\theta_{\max}$ is set to around the upper bound of input variables, $\theta_{\min}$ is set to around 10% of $\theta_{\max}$, γ is set to around 0.999. The overlap factor κ is, generally, set to 1.0 which is used to initialize the width of the newly recruited rule. The growing threshold e_k is used to prevent small correntropy values caused by outliers or impulse noise to add rules. It is, generally, set to 0.5. In this study, the parameters described above are fixed as follows: $\gamma = 0.999, \theta_{\max} = 1.0, \theta_{\min} = 0.1, \kappa = 1.0, e_k = 0.5$. Another growing threshold e_g exhibits the similarity degree between the desired and real outputs and its selection is associated with the system performance, which is usually set around 0.9. The step size η determines the learning speed and the kernel width $\sigma > 0.5$ is considered when CEFNS is applied to different problems. They are chosen differently for different problems so as to achieve an optimal performance.

B. Time-Varying Nonlinear System Identification

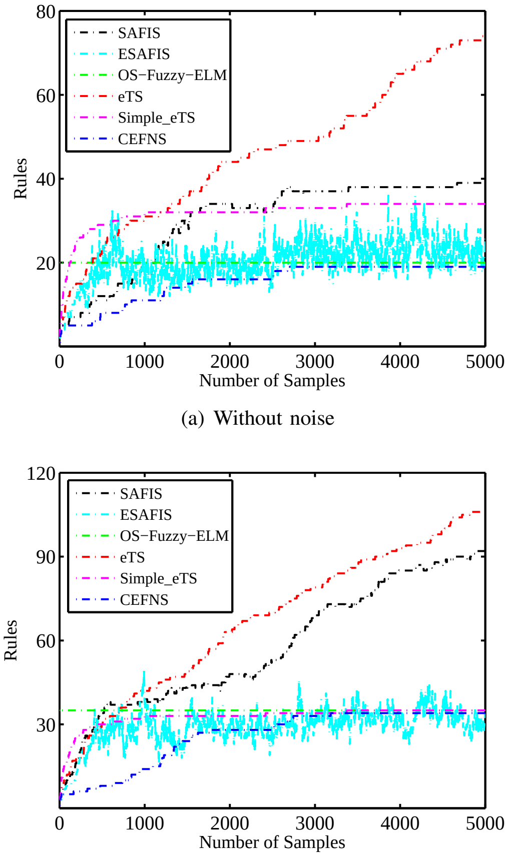
The time-varying nonlinear system considered in the study is described as

$$
\begin{aligned}
x(n) =& \frac{19\beta(n)}{40} \sin\left(\frac{16u(n-1) + 8x(n-1)}{\beta(n)(3 + 4u(n-1)^2 + 4x(n-1)^2)}\right) \\
&+ \frac{1}{5}u(n-1) + \frac{1}{5}x(n-1) \qquad (55)
\end{aligned}
$$

where $\beta(n)$ is a time-varying parameter and varied according to the following relationship:

$$
\beta(n) = \begin{cases} 1.0 & 0 \leq n \leq 1500 \\ 0.9 & 1500 < n \leq 2500 \\ 0.8 & 2500 < n \leq 5000 \end{cases}
$$

The training input $u(n)$ is uniformly selected in the range $[-1, 1]$ and the testing input is given by $u(n) = \sin(2\pi n/25)$. For the purposes of training and testing, 5000 training data and 200 testing data are produced, respectively. Also to further assess the non-Gaussian noise resistance performance of the CEFNS, the impulse noise with amplitude 2 at an interval of 10 samples are added in the input data.



(a) Without noise

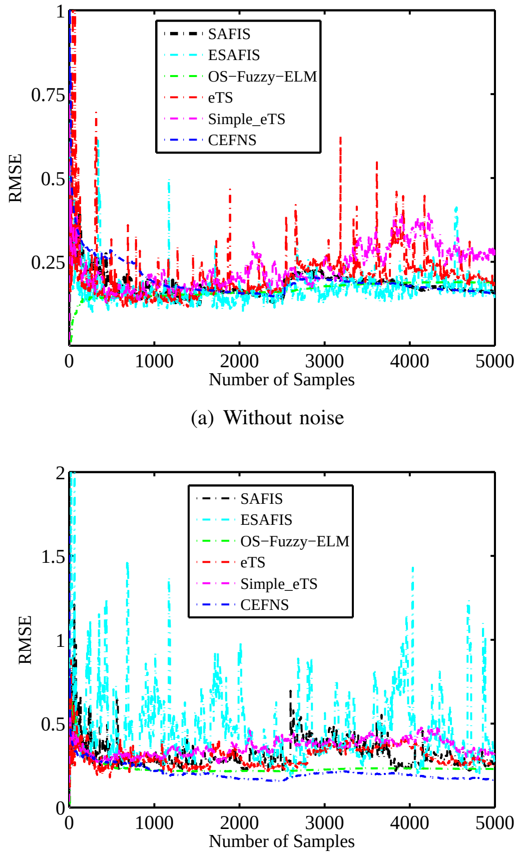
(b) Impulse noise

Fig. 6. Rules updating process between different algorithms for time-varying system.

Table III gives the performance comparison for the problem between the CEFNS and the other algorithms. From the table it can be seen that CEFNS achieves slightly better testing accuracy with smaller training time and number of rules in the noise-free condition. However, under the circumstance of impulse noise, the proposed CEFNS is very robust and can effectively suppress the impulse noise and achieve better performance. Similar to the existing fuzzy evolving algorithms, CEFNS can automatically determine the system structure and

TABLE IV
PERFORMANCE COMPARISON FOR MACKEY-GLASS TIME SERIES PREDICTION

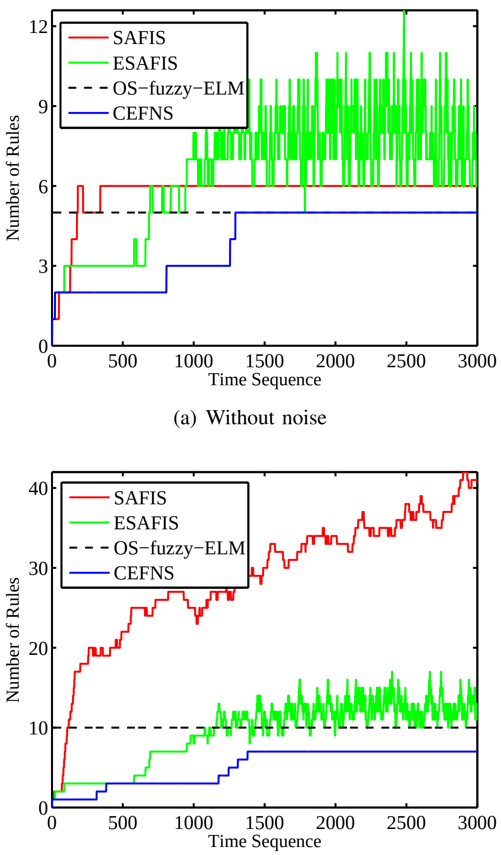
Algorithms	Without Noise				With Impulse Noise			
	Testing RMSE	Testing NDEI	Training Time	Rules	Testing RMSE	Testing NDEI	Training Time	Rules
SAFIS [14]	0.0939	0.3760	/	6	0.0983	0.3861	16.8169	41
ESAFIS	0.0752	0.2955	5.8344	6	0.0962	0.3780	14.8513	11
OS-Fuzzy-ELM	0.0761	0.2991	0.9253	5	0.1134	0.4455	2.0311	10
eTS [9]	0.0968	0.3805	/	9	0.1052	0.4134	68.2384	98
Simpl_eTS [10]	0.1003	0.3940	/	11	0.1364	0.5360	9.6877	34
CEFNS	**0.0679**	**0.2635**	**0.4368**	**5**	**0.0885**	**0.3478**	**1.1544**	**7**



(a) Without noise

(b) Impulse noise

Fig. 7. Learning error updating process between different algorithms for time-varying system.



(a) Without noise

(b) With impulse noise

Fig. 8. Rules updating process between different algorithms for Mackey-Glass prediction

update the parameters simultaneously. Here, the number of fuzzy rules required by the OS-Fuzzy-ELM algorithm is fixed and determined by a trial-and-error. The evolution of the fuzzy rules for different fuzzy algorithms is shown in Fig. 6 in the noise-free and noisy cases. From the figure one can see that the CEFNS uses the least number of rules compared with other algorithms. Also change of the rules together with β is observed in the figure. When the system dynamics changes due to the variation of β at step number 1500 and 2500, the rules are added to adapt for the variations. Fig. 7 shows the update process of the learning error RMSE index in the noise-free and noisy cases. It can be found from the figure that the error increases and decreases according to β. From the figure, one can find that the approximation error suddenly increases when the system dynamics changes at step number 1500 and 2500. But the proposed CEFNS can reject the large errors and generate the smooth learning process compared with other algorithms. Besides, in the impulse noise circumstances, the EFNS achieves a smooth learning process due to its outlier rejection property.

TABLE V
DETAILS AND PARAMETERS CHOSEN FOR REAL WORLD REGRESSION BENCHMARK PROBLEMS

Datasets	Input number	Output number	Training number	Testing number	Parameter η	Parameter σ	Parameter e_g
Autos	15	1	80	79	0.45	1.5	0.9
California Housing	8	1	10320	10320	0.3	1.6	0.9
Delta Ailerons	5	1	3000	4129	0.45	1.5	0.98

TABLE VI
PERFORMANCE COMPARISON FOR REGRESSION BENCHMARK PROBLEMS

Datasets	Algorithms	Without Noise				With Impulse Noise			
		Training RMSE	Testing RMSE	Training Time	Rules	Training RMSE	Testing RMSE	Training Time	Rules
Autos	SAFIS	0.0764	0.1184	0.4524	5	0.1089	0.1377	0.3120	9
	ESAFIS	0.0612	0.0604	0.2184	3	0.1256	0.1290	0.3276	5
	OS-Fuzzy-ELM	**0.0153**	0.0595	0.0296	2	0.1695	0.1612	0.0866	5
	eTS	0.0299	**0.0535**	0.2184	3	0.1497	0.1449	1.1856	19
	Simpl_eTS	0.0856	0.0689	0.5772	10	0.1195	0.1208	0.1716	6
	CEFNS	0.0644	0.0666	**0.0156**	**2**	**0.0834**	**0.0976**	**0.0312**	**5**
California Housing	SAFIS	0.0955	0.0988	21.9805	12	0.1189	0.1223	21.3565	14
	ESAFIS	0.0866	0.0892	30.2330	6	0.0932	0.0966	57.6892	9
	OS-Fuzzy-ELM	0.1302	0.1320	6.7252	5	0.1227	0.1253	6.2993	5
	eTS	**0.0744**	**0.0772**	7.6128	3	0.0920	0.0938	128.5604	41
	Simpl_eTS	0.0744	0.0773	5.5536	3	0.1181	0.1203	47.6115	17
	CEFNS	0.0847	0.0878	**1.5444**	**2**	**0.0900**	**0.0932**	**3.4788**	**4**
Delta Ailerons	SAFIS	0.0548	0.0549	6.8328	14	0.0961	0.0970	17.1167	37
	ESAFIS	**0.0493**	0.0506	12.3865	13	0.0729	0.0738	24.9064	24
	OS-Fuzzy-ELM	0.0497	0.0507	0.4602	3	0.1186	0.1222	5.9179	18
	eTS	0.0508	0.0513	2.1372	4	0.1166	0.1173	28.5638	51
	Simpl_eTS	0.0509	0.0512	2.0088	4	0.1253	0.1260	22.9789	34
	CEFNS	0.0519	**0.0502**	**0.3432**	**3**	**0.0704**	**0.0713**	**1.5288**	**16**

C. Mackey-Glass Time Series Prediction

The CEFNS is further evaluated by predicting the chaotic Mackey-Glass time series, which is generated from the following differential equation [14]:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{56}$$

In the study $\tau = 17$ and the initial state $x(0) = 1.2$. In the prediction process, the past three values and the current value consist of the input vector $[x(t-18), x(t-12), x(t-6), x(t)]$ and are used to predict the value $x(t+85)$. A total of 3000 training data is extracted from the interval between $t = 201$ and $t = 3200$. 500 testing data is extracted from the interval between $t = 5001$ and $t = 5500$. As in [9], [10] and [14], the input and output data are normalized in the range [0,1]. Likely, the impulse noise with unit amplitude at an interval of 10 samples are added in the input data to validate the non-Gaussian resistance performance of the CEFNS.

The prediction accuracy, the training time and the number of rules obtained by CEFNS, SAFIS, ESAFIS, OS-Fuzzy-ELM, eTS and Simpl_eTS algorithms are presented in Table IV. Besides the RMSE performance index, the Non-Dimensional Error Index (NDEI) which is defined as the RMSE divided by the standard deviation of the true output values is utilized as the performance index here. From table IV one can find that the CEFNS achieves the best prediction accuracy with the least training time and the smallest number of fuzzy rules in the noise-free and noisy cases. The evolution of fuzzy rules in CEFNS, SAFIS, ESAFIS and OS-Fuzzy-ELM is shown in Fig. 8. From Fig. 8 it can be seen that the CEFNS generates the least number of rules. Also CEFNS has a smooth rule evolution process while the rule evolution process of SAFIS and ESAFIS is clearly oscillatory in the noisy case.

D. Regression Problems

In this section, three real-world regression problems are further considered to evaluate the performance of CEFNS. Details of the problems are listed in Table V. The input and output attributes are normalized in the range $[0, 1]$. All the inputs are used in the simulation studies. The optimal parameters (e_g, σ, η) chosen for these problems are listed in Table V.

Table VI presents the performance comparison between SAFIS, ESAFIS, OS-Fuzyy-ELM, eTS and Simple_eTS. In order to illustrate the robustness of the CEFNS against the non-Gaussian noise, the unit-amplitude impulse noise have been added on the training samples at an interval of 10 samples while the testing data remain noise-free. The results in the noise-free and noisy experiments are both presented in table VI. From the table, it can been found that the proposed CEFNS produces similar testing accuracy with the least number of rules and training time compared with other algorithms in the noise-free condition. However, under impulse noise conditions, CEFNS obviously obtains better testing accuracy than other algorithms. Also, smaller training time is required and smaller number of rules are obtained compared with other algorithms.

VI. Conclusions

In this paper, a correntropy-based evolving fuzzy neural system (CEFNS) is proposed. Correntropy represents a nonlinear measure of similarity between the desired and the actual outputs in a kernel feature space and is able to extract the higher order statistical information of the error than the commonly used MSE criterion. Based on this, the correntropy together with the distance criterion is applied to evolve its rule base of the CEFNS, which is different from commonly used error criterion for adding rules. To prevent the data with outliers from becoming rules, the lower limit of the correntropy is set which ensures that the system obtains reasonable rules. The parameters of the CEFNS are adjusted according to the maximum correntropy criterion (MCC). Compared with the commonly used parameter adjusting techniques based on the MSE criterion, the MCC parameter adaption contains an extra scaling factor which is an exponential function of the error, which can reject the outliers in the training data. Besides, the steady-state convergence performance of the CEFNS in the case of Gaussian and non-Gaussian noise is analyzed in the study. The simulation results from one system identification problem, Mackey-Glass time series prediction and three real-world benchmark regression problems show that the proposed CEFNS obtains similar learning accuracy with smaller number of rules and less training time compared with other fuzzy algorithms in the noise-free circumstance. The results also demonstrate that the proposed CEFNS has the superior non-Gaussian noise handling capability to other fuzzy algorithms. In view of merits of the correntropy, it will be extended to different evolving fuzzy systems without user-specific parameters, such as eTS+ [11], and TEDA [40], in our future research work. In order to prevent the rule size from growing too large in case of real and long on-line data streams, it is necessary to merge or prune on-line similar or redundant fuzzy rules. This issue will be addressed in our future work.

Acknowledgment

The authors would like to thank the financial support from the Short-Term Visiting Scholar Scheme of Faculty of Science and Technology and Research grant (MYRG2014-00178-FST) of University of Macau. This work was also supported by the National Natural Science Foundation of China (91648208), the National Science Council of ShaanXi Province (2014JM8337) and the Fundamental Research Funds for the Central Universities.

References

[1] N. Kasabov, “Evolving fuzzy neural networks-algorithms, applications and biological motivation,” *Methodologies for the Conception, Design and Application of Soft Computing, World Scientific*, vol. 1, pp. 271–274, 1998.

[2] P. Angelov and X. Zhou, “Evolving fuzzy systems from data streams in real-time,” in *2006 International Symposium on Evolving Fuzzy Systems*, pp. 29–35, 2006.

[3] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*. Prentice Hall, 1997.

[4] P. P. Angelov and R. Buswell, “Automatic generation of fuzzy rule-based models from data by genetic algorithms,” *Information Sciences*, vol. 150, no. 1-2, pp. 17–31, 2003.

[5] E. Lughofer, *Evolving Fuzzy Systems – Methodologies, Advanced Concepts and Applications*. Springer, Berlin Heidelberg, 2011.

[6] P. Angelov and R. Buswell, “Evolving rule-based models: A tool for *intelligent* adaptation,” in *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, pp. 1062–1067, 2001.

[7] S. Wu and M. J. Er, “Dynamic fuzzy neural networks-a novel approach to function approximation,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 30, no. 2, pp. 358–364, 2000.

[8] G. Leng, T. M. McGinnity, and G. Prasad, “An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network,” *Fuzzy sets and systems*, vol. 150, no. 2, pp. 211–243, 2005.

[9] P. P. Angelov and D. P. Filev, “An approach to online identification of Takagi-Sugeno fuzzy models,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 1, pp. 484–498, 2004.

[10] P. Angelov and D. Filev, “Simpl_eTS: a simplified method for learning evolving takagi-sugeno fuzzy models,” in *The 14th IEEE International Conference on Fuzzy Systems*, pp. 1068–1073, 2005.

[11] P. Angelov, “Evolving Takagi-Sugeno fuzzy systems from data streams (eTS+),” in *Evolving intelligent systems: methodology and applications*, pp. 21–50, John Wiley and Sons, 2010.

[12] K. Subramanian and S. Suresh, “A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system,” *Applied Soft Computing*, vol. 12, pp. 3603–3614, 2012.

[13] M. Pratama, S. G. Anavatti, P. P. Angelov, and E. Lughofer, “PANFIS: A novel incremental learning machine,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 55–68, 2014.

[14] H.-J. Rong, N. Sundararajan, G.-B. Huang, and P. Saratchandran, “Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction,” *Fuzzy Sets and Systems*, vol. 157, no. 9, pp. 1260–1275, 2006.

[15] H.-J. Rong, N. Sundararajan, G.-B. Huang, and G.-S. Zhao, “Extended sequential adaptive fuzzy inference system for classification problems,” *Evolving Systems*, vol. 2, no. 2, pp. 71–82, 2011.

[16] N. Kasabov and Q. Song, “DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time series prediction,” *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 144–154, 2002.

[17] C.-F. Juang and C.-T. Lin, “An on-line self-constructing neural fuzzy inference network and its applications,” *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 144–154, 2002.

[18] M. Pratama, S. G. Anavatti, and E. Lughofer, “GENEFIS: Toward an effective localist network,” *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 3, pp. 547–562, 2014.

[19] A. Lemos, W. Caminhas, and F. Gomide, “Multivariable gaussian evolving fuzzy modeling system,” *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 1, pp. 91–104, 2011.

[20] D. Dovzan, V. Logar, and I. Skrjanc, “Implementation of an evolving fuzzy model (eFuMo) in a monitoring system for a waste water treatment process,” *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 5, 2014.

[21] L. Maciel, R. Ballini, and F. Gomide, “Evolving possibilistic fuzzy modeling for realized volatility forecasting with jumps,” *IEEE Transactions on Fuzzy Systems*, 2016.

[22] L. Ljung, *System Identification: Theory for the user*. Prentice Hall PTR, Upper Saddle River, New Jersey, 1999.

[23] A. Bifet and R. Gavalda, *Kalman filters and adaptive windows for learning in data streams*, vol. 4265. Springer, Berlin Heidelberg, 2006.

[24] B. Chen, X. Liu, H. Zhao, and J. C. Principe, “Maximum correntropy kalman filter,” *Automatica*, vol. 76, pp. 70–77, 2017.

[25] B. Chen, L. Xing, H. Zhao, N. Zheng, and J. C. Principe, “Generalized correntropy for robust adaptive filtering,” *IEEE Transactions on Signal Processing*, vol. 64, no. 13, pp. 3376–3387, 2016.

[26] B. Chen, J. Wang, H. Zhao, and N. Zheng, “Convergence of a fixed-point algorithm under maximum correntropy criterion,” *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1723–1727, 2015.

[27] A. Singh and J. C. Principe, “Information theoretic learning with adaptive kernels,” *Signal Processing*, vol. 91, no. 2, pp. 203–213, 2011.

[28] W. Liu, P. P. Pokharel, and J. C. Principe, “Correntropy: properties and applications in non-gaussian signal processing,” *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5286–5298, 2007.

[29] A. Singh and J. C. Principe, “Using correntropy as a cost function in linear adaptive filters,” in *Proceedings of the 2009 International Joint Conference on Neural Networks*, pp. 2950–2955, 2009.

[30] S. Zhao, B. Chen, and J. C. Principe, “Kernel adaptive filtering with maximum correntropy criterion,” in *The 2011 International Joint Conference on Neural Networks (IJCNN)*, pp. 2012–2017, 2011.

[31] A. Singh and J. C. Principe, “A loss function for classification based on a robust similarity metric,” in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, 2010.

[32] R. He, B.-G. Hu, W.-S. Zheng, and X.-W. Kong, “Robust principal component analysis based on maximum correntropy criterion,” *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1485–1494, 2011.

[33] X.-T. Yuan and B.-G. Hu, “Robust feature extraction via information theoretic learning,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1193–1200, 2009.

[34] B. Chen, L. Xing, J. Liang, N. Zheng, and J. C.Principe, “Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion,” *IEEE Signal Processing Letters*, vol. 21, no. 7, pp. 880–884, 2014.

[35] R. Price, “A useful theorem for nonlinear devices having gaussian inputs,” *IRE Transactions on Information Theory*, vol. 4, no. 2, pp. 69–72, 1958.

[36] T. Y. Al-Naffourl and A. H.Sayed, “Adaptive filters with error nonlinearities: mean-square analysis and optimum design,” *EURASIP Journal on Advances in Signal Processing*, vol. 2001, no. 1, pp. 192–205, 2001.

[37] B. Lin, R. He, X. Wang, and B. Wang, “The steady-state mean-square error analysis for least mean p-order algorithm,” *IEEE Signal Process Letters*, vol. 16, pp. 176–179, 2009.

[38] J. Omura and T. Kailath, “Useful probability distributions,” *Some Useful Probability Distributions*, 1965.

[39] H.-J. Rong, G.-B. Huang, N. Sundararajan, and P. Saratchandran, “Online sequential fuzzy extreme learning machine for function approximation and classification problems,” *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 39, no. 4, pp. 1067–1072, 2009.

[40] D. Kangin, P. Angelov, and J. A. Iglesias, “Autonomously evolving classifier TEDAClass,” *Information Sciences*, vol. 366, pp. 1–11, 2016.

Rong-Jing Bao received the B.Eng. from University of Science and Technology Beijing, Beijing, China, in 2010. He is currently working toward the Ph.D. degree in School of Aerospace, Xi’an Jiaotong University, Xi’an, China. His research interests include fuzzy systems, neural networks, pattern recognition and intelligent control.

Hai-Jun Rong received the B.Eng. from Xi’an Technological University, P. R. China, in 2000, the M.Eng. from Xi’an Jiaotong University, P. R. China, in 2003, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2008. From December 2006 to October 2008, she worked as a research associate and research fellow in Nanyang Technological University. Since then, she has been an associate professor in School of Aerospace, Xi’an Jiaotong University. She is an Associate Editor of the Evolving Systems journal (Springer). Her research interests include neural networks, fuzzy systems, pattern recognition and intelligent control.

Plamen P. Angelov (F’16, SM’04, M’99) holds a Personal Chair (full Professorship) in Intelligent Systems with the School of Computing and Communications, Lancaster University, UK. He obtained his Ph.D. (1993) and his D.Sc. (2015) from the Bulgarian Academy of Science. He is the Vice President of the International Neural Networks Society and a member of the Board of Governors of the Systems, Man and Cybernetics Society of the IEEE, a Distinguished Lecturer of IEEE. He is Editor-in-Chief of the Evolving Systems journal (Springer) and Associate Editor of IEEE Transactions on Fuzzy Systems as well as of IEEE Transactions on Cybernetics and several other journals. He received various awards and is internationally recognized pioneering results into on-line and evolving methodologies and algorithms for knowledge extraction in the form of human-intelligible fuzzy rule-based systems and autonomous machine learning. He holds a wide portfolio of research projects and leads the Data Science group at Lancaster.

Badong Chen (M’10, SM’13) received the B.S. and M.S. degrees in control theory and engineering from Chongqing University, Chongqing, China, in 1997 and 2003, respectively, and the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2008. He was a Postdoctoral Researcher with Tsinghua University from 2008 to 2010 and a Postdoctoral Associate with the University of Florida Computational NeuroEngineering Laboratory during the period October, 2010 to September, 2012. He visited the Nanyang Technological University as a Visiting Research Scientist during July to August 2015. He is currently a Professor at the Institute of Artificial Intelligence and Robotics, Xi’an Jiaotong University, Xi’an, China. He has published two books, three chapters, and more than 100 papers in various journals and conference proceedings. His research interests include signal processing, information theory, machine learning, and their applications in cognitive science and engineering. He is an Associate Editor of IEEE Transactions on Neural Networks and Learning Systems and Journal of The Franklin Institute, and has been on the editorial board of Entropy.

PakKin Wong received the Ph.D. degree in Mechanical Engineering from The Hong Kong Polytechnic University, Hong Kong, in 1997.He is currently a Professor in the Department of Electromechanical Engineering and the Associate Dean (Academic Affairs), Faculty of Science and Technology, University of Macau. His research interests include automotive engineering, fluid transmission and control, engineering applications of artificial intelligence, and mechanical vibration. He has published over 177 scientific papers in refereed journals, book chapters, and conference proceedings.