Evolving fuzzy model identification of nonlinear Wiener-Hammerstein processes

GORAN ANDONOVSKI¹, EDWIN LUGHOFER², AND IGOR ŠKRJANC.³, (Member, IEEE)

¹Faculty of Electrical Engineering, University of Ljubljana, Slovenia, (e-mail: goran.andonovski@fe.uni-lj.si)
²Institute for Mathematical Methods in Medicine and Data Based Modeling, Johannes Kepler University, Linz, Austria (e-mail: edwin.lughofer@jku.at)
³Faculty of Electrical Engineering, University of Ljubljana, Slovenia, (e-mail: igor.skrjanc@fe.uni-lj.si)

Corresponding author: Goran Andonovski (e-mail: goran.andonovski@fe.uni-lj.si).

“The second and third authors acknowledge the support by the “LCM — K2 Center for Symbiotic Mechatronics” within the framework of the Austrian COMET-K2 program.”

ABSTRACT This paper presents a new approach to neuro-fuzzy model identification based on a filtered recursive least-squares method combined with an incrementally evolving Gaussian clustering method. The proposed identification algorithm generates the model on the fly and requires few user-defined parameters, which is one of the main advantages compared to other methods. The partitioning of the input-output data space depends on the chosen criteria and thresholds and depends only on the operating point of the model. As an example, the Wiener-Hammerstein type of a dynamic process was identified to show the potential of the proposed method in identifying nonlinear dynamic models. The Wiener-Hammerstein structure was used because a variety of processes can be modeled with this type of structure. Moreover, we tested the same identification concept on a real heat exchanger plant with strong nonlinear behavior. In addition, the limitations of the real sensors and actuators represent a serious challenge to the identification procedure. Both experiments, on a simulated Wiener-Hammerstein model and on a real plant, have shown that the proposed new neuro-fuzzy model identification with the new merging concept is very easy to implement, can perform all the necessary computation online, and can generate meaningful models.

INDEX TERMS Data Stream, Evolving Clustering, Filtered Recursive least-squares Identification, Neuro-fuzzy Models

I. INTRODUCTION

Data mining in the form of neuro-fuzzy models from data streams is an emerging and important area of research. In this concept, it is considered that the data is processed step by step, online in real time, updating the parameters, and evolving the structure of the identified model. The algorithms with these functionalities are generally called evolving algorithms, [1], [2], and they can be only incremental in nature [3], [4], [5], or they can simultaneously evolve the structure and the parameters of the model [6], [7]. These algorithms are often an extension of the classical neuro-fuzzy algorithms, i.e., an extension of the evolving Gustafson-Kessel clustering algorithm is proposed in [8], [9], and [10], or the evolving extension of possibilistic clustering in [11], [12], where a Cauchy data distribution is assumed, and an evolving algorithm based on the Gath-Geva method is presented in [13]. An extensive survey of evolving fuzzy and neuro-fuzzy approaches for clustering, regression, identification, and classification is demonstrated in [14].

The basic adaptive and evolving principles have also been adopted in the area of dynamic neuro-fuzzy model identification. Among the most successful were adaptive model identification methods (e.g., ANFIS [15], GAFIS [16], rFCM [17], rGK [18]), in which the models have a fixed structure, i.e., the number of clusters (the rules) is constant, and the algorithm adapts the model parameters and the membership functions. In the case of incremental methods (e.g., DENFIS [20], eTS [5], Gen-Smart-EFS with rule splitting functionality [22], PANFIS [23]), the algorithm adapts the parameters and the clusters, and they are able to add the cluster into the model structure, and the most advanced are evolving methods (e.g., SAFIS [24], [25], eTS+ [26], ENFM [27], FLEXFIS++ [28], SOFMLS [29], eFuMo [9]), which, besides an adding mechanism, also implement merging and splitting mechanisms.
These methods are mainly based on recursive least-squares method (RLS) for learning consequent parameters in incremental manner. The classical recursive least-squares (RLS) method used in the above methods results in unbiased model parameters only when the process output noise is generated as filtered white noise by the filter 1/A(q), where A(q) stands for the denominator of the process transfer function. This type of noise is very unrealistic in most practical situations. A much more common and realistic noise model is additive white noise at the process output. This structure is called an output error model (OE). In this case, the classical RLS leads to biased model parameters. RLS estimation of OE model parameters requires filtering the input-output regressor with an appropriate filter [31].

Iterative filtering of input-output variables is well known from batch least square methods to obtain different output error models and estimate the noise filter model. The principle of filtering has been successfully extended to the recursive paradigm, i.e., filtered recursive least-squares (FRLS). Recently, an input-output data filtering based recursive least-squares identification of an autoregressive model structure (AR) was proposed in [32]; a data filtering based recursive least-squares algorithm with separation principle was introduced in [33]; [34] describes two-stage parameter estimation algorithms for Box-Jenkins model (BJ); in [35] an iterative and recursive least-squares estimation algorithms for moving average models (MA) are described; in [36], filtered least-squares based recursive and iterative estimation for output error moving average systems is proposed; two gradient approaches based on data filtering for estimation of model parameters are given in [37] and [39].

In our approach, the concept of evolving systems and the filtered recursive least-squares (FRLS) has been integrated to identify the neuro-fuzzy model of nonlinear dynamic stable processes in the OE model structure. In this sense, the design of appropriate excitation signals to acquire the identification data is of great importance. In the case of nonlinear systems, when the output of a system depends nonlinearly on its inputs (Wiener-Hammerstein type of process), this is even much more demanding than in the case of linear systems, since the process dynamics changes due to the change of the operating point. The quality of the identification signals determines the upper bound of the model accuracy. Very often, a signal such as amplitude modulated pseudorandom binary signals (APRBS) is used because of its frequency and amplitude variance, which can cover both the frequency spectrum and the amplitude range of the process. In our case, the sequence of stairs is proposed to excite the process from the minimum to the maximum input value, with a specific holding time, i.e., the time of a step in which the system settles. The number of steps should be defined to have an appropriate amplitude variability. At each step, the FRLS define the local model parameters that form the consequent part of the new rule, and the premise of the rule is defined by an evolving clustering algorithm as proposed in [40]. At each new step, a new cluster is formed and the covariance matrix of FRLS is reset to the initial value. The last two local linear models and the corresponding clusters are compared, i.e., their transfer functions are compared and merged if they do not differ significantly. The same is done with the corresponding clusters.

The proposed approach was tested on a simulated Wiener-Hammerstein (WH) model and on a real plate heat exchanger (PHE) process. The WH processes have been widely studied in the literature, due to the fact that many real-world problems (as shown for the PHE process) can be approximated with this type of model structure. Different identification methods for Wiener-Hammerstein nonlinear systems are given in [41]; in [42], a filtering-based, recursive, least-squares algorithm for Wiener-Hammerstein FIR-MA systems is presented; an auxiliary model-based least-squares identification methods for Wiener-Hammerstein output-error systems is discussed in [43]; [44] described gradient-based and least-squares-based iterative algorithms for WH systems using the hierarchical identification principle; a least-squares-based gradient-based iterative identification for Wiener nonlinear systems is given in [45]; and a hierarchical least-squares estimation algorithm for Hammerstein-Wiener systems is proposed in [46].

This paper is organized as follows: At the beginning, the reasons and requirements for a simple neuro-fuzzy dynamic model identification technique are presented; in Section II, the identification of nonlinear dynamic processes with filtered recursive least-squares, the excitation signal and the reasons for filtering the regressor are given; in Section III, the evolving clustering in the input-output data domain is presented, where the mechanism of adding new clusters, the recursive parameter adaptation, and merging of clusters is given. Section IV explains the whole procedure in an algorithmic manner, and Section V presents the calculation of the prediction and simulation model. Two examples of evolving neuro-fuzzy dynamic model identification is given in Section VI, and the Conclusion is given in Section VII.

II. IDENTIFICATION OF NONLINEAR DYNAMIC PROCESSES WITH RECURSIVE LEAST-SQUARES (RLS)
To find the local linear models around different equilibrium points in the case of nonlinear dynamic processes using an autoregressive model with exogenous (ARX) input, the extended regressor vector should be used as follows

\[
\varphi^T(k) = [u(k-1) \cdots u(k-m) y(k-1) \cdots y(k-n)] \quad (1)
\]

where \(u(k-1), \ldots, u(k-m)\) and \(y(k-1), \ldots, y(k-n)\) stand for measured process input and output variables and its delayed values, and the constant value 1 stands to implicitly estimate the equilibrium. The vector of model parameters \(\theta\) then becomes as follows

\[
\theta = [b_1 \cdots b_m a_1 \cdots a_n r]. \quad (2)
\]

Therefore, the process transfer function is assumed as

\[
G(q) = \frac{B(q)}{A(q)} = \frac{b_1 q^{-1} + \cdots + b_m q^{-m}}{1 + a_1 q^{-1} + \cdots + a_n q^{-n}} \quad (3)
\]
where \( m \) and \( n \) stand for the number of delayed inputs and outputs, respectively and \( r \) in Eq. 2 incorporates the information about the equilibrium point \([u_0, y_0]\).

The input signal \( u(k) \) of the identified process is crucial to obtain an appropriate model, because it is the only signal that excites the process and provides information about its behavior. The real applications require a certain choice of input signals with a large number of restrictions according to a large number of constraints. One of these constraints is always the boundedness of the input signal. In any real process the input signal is bounded between a minimum \( u_{min} \) and maximum value \( u_{max} \). The next constraints are the limited measurement time and the shape of the input signal. The shape of the input signal defines the spectrum of excited frequencies in the process, which is extremely important to obtain a suitable model.

A. RECURSIVE LEAST-SQUARES IDENTIFICATION (RLS)

The idea of recursive least-squares identification is based on the adaptation of current estimated model parameter vector \( \hat{\theta} \) with the information obtained from a new regression vector. That is, the estimated parameter vector at time \( k \), defined as \( \hat{\theta}(k) \), is calculated as an adaptation of the previous parameter vector \( \hat{\theta}(k-1) \) with a correction vector. This correction vector depends on the regressor \( \varphi(k) \) and the measured process output \( y(k) \).

The basic equations of the recursive least-squares method are given as follows [31]:

\[
\hat{\theta}(k) = \hat{\theta}(k-1) + \gamma(k)e(k)
\]

\[
\gamma(k) = \frac{1}{\varphi^T(k)P(k-1)\varphi(k) + 1} \quad \text{P}(k-1)\varphi(k)
\]

\[
P(k) = (I - \gamma(k)\varphi^T(k))P(k-1)
\]

where \( P^{-1}(k) \) defines the data matrix \( P^{-1}(k) = \psi^T(k)\psi(k) \), the information about all past regressor vectors is comprised \( \psi^T(k) = [\varphi(1) \cdots \varphi(k)] \), \( e(k) \) is the prediction error given as

\[
e(k) = y(k) - \varphi^T(k)\hat{\theta}(k-1)
\]

which defines the error between the measured process output \( y(k) \) and one-step-ahead prediction of the model output \( \hat{y}(k|k-1) = \varphi^T(k)\hat{\theta}(k-1) \).

The initial model parameters \( \hat{\theta}(0) \) should be chosen to assure a stable model parameters and the initial value of \( P(0) \) is chosen as a diagonal matrix as follows: \( P(0) = \alpha J \), where \( \alpha \) is defined as \( 10^4 \) or \( 10^5 \). This leads to a major adaptation of the parameters at the beginning [19].

B. STEP SIGNAL AS AN EXCITATION

The spectrum of the process input signal determines the frequencies that are excited. The frequencies with stronger excitation are estimated with higher accuracy than those excited with lower excitation. This means that the design of the input signal is of great importance and also depends on the purpose of the model. In the case of identifying the model for control purposes [31], the step input signal is highly suitable. In addition to the excitation of higher frequencies, the low frequencies are emphasized; therefore, the static gain is estimated very accurately, which is very important for control problems.

The expected value of quadratic cost function of estimated parameters \( I(\theta) \) is, according to Parseval’s theorem, in the frequency domain related to the spectrum of the prediction error \( \Phi_e(\omega) \)

\[
E\{I(\theta)\} = \frac{1}{4\pi} \int_{-\pi}^{\pi} \left| G(e^{j\omega T}) - G(e^{j\omega T}, \hat{\theta}) \right|^2 \Phi_u(\omega)
\]

\[
+ \Phi_n(\omega) \left( \frac{1}{|H(e^{j\omega T}, \hat{\theta})|^2} \right) d\omega
\]

where the real process is described as \( y(k) = G(q)u(k) + H(q)v(k) \) with \( G \) as transfer function of real process and \( H \) the noise model. The \( G \) and \( H \) are estimated transfer functions of process model and the noise model, respectively. The spectra of the input and the output noise are denoted as \( \Phi_u \) and \( \Phi_n \), and \( T \) stands for the sampling time. The right-side of equation is in the frequency domain, and therefore \( G \), \( \hat{G} \), and \( H \) are written as functions of \( z = e^{j\omega T} \).

In Eq. 7 expression \( G(e^{j\omega T}) - G(e^{j\omega T}, \hat{\theta}) \) quantifies the error between the process and the model in the frequency domain \( \omega \), and the term \( \Phi_u(\omega)|/|H(e^{j\omega T}, \hat{\theta})|^2 \) acts as frequency dependent weighting factor, i.e., the power spectra of the input signal and the frequency response of the noise model shape the accuracy of the model in the frequency domain, i.e., the error \( G(e^{j\omega T}) - G(e^{j\omega T}, \hat{\theta}) \) becomes smaller at frequencies with higher weight, because it participate more in the cost function.

C. FILTERING OF REGRESSORS

When identifying nonlinear systems, excitation with the step input signal could cause problems. This is due to the constant values of the step signal and the constant value added to expand the regressor to implicitly estimate the operating point. These two linearly dependent variables in the matrix \( \psi(k) \) cause the singularity of the covariance matrix \( \psi^T(k)\psi(k) \), which is therefore not invertible. This means that \( P(k) \) cannot be computed directly, i.e., \( P(k) = (\psi^T(k)\psi(k))^{-1} \). To overcome this problem in the proposed solution, the input and output process variables are filtered before the calculation of RLS [42].

In the case of the least mean squares, the generalized predictive error is optimized to identify the parameters of the model [30], i.e., the model is in the form of ARX (AutoRegressive model with exogenous input) with predictive error

\[
e(k) = A(q)\left( y(k) - \hat{G}(q)u(k) \right).
\]

This means that in case of the process with output noise \( y(k) = G(q)u(k) + n(k) \) the prediction error will be a colored noise \( e(k) = A(q)n(k) \) and the model parameters do not converge to the right values, i.e., are biased.
Filtering of the process input and process output with filter \( F(q) \) means that the prediction error becomes as follows:

\[
e_{F}(k) = F(q) \hat{A}(q) \left( y(k) - \hat{G}(q)u(k) \right). \tag{9}
\]

If the filter is defined as

\[
F(z) = \frac{1}{\hat{A}(z)}. \tag{10}
\]

the prediction error in Eq. 9 becomes equal to the white noise. This means that the model parameters converge to the right values, i.e., the estimation is unbiased.

In the case of FRLS, the polynomial \( \hat{A}(z) \) (see Eq. 3) is not known in advance and therefore, the algorithm starts with initial (user-defined) stable polynomial \( A_f(z) \) which is used in Eq. 10 instead of \( \hat{A}(z) \). After some learning steps \((K_{min})\) and when the roots of the estimated polynomial \( \hat{A}(z) \) are stable (lie in the unity circle)

\[
\hat{A}(z_i) = 0, \quad |z_i| < 1, \quad \forall i
\]

then \( \hat{A}(z) \) is used instead of \( A_f(z) \) as filtering polynomial in Eq. 10.

The filtering procedure means that the regressor vector becomes as follows:

\[
\varphi^T(k) = [u_f(k-1) \cdots u_f(k-m) \; y_f(k-1) \cdots y_f(k-n)]^T \tag{12}
\]

where \( u_f(k-j), \; i = 1, \ldots, m \) and \( y_f(k-j), \; j = 1, \ldots, n \) stand for filtered input and output signals, i.e., \( A_f(q)u_f(k-m) = u(k-m) \) and \( \hat{A}(q)y_f(k-n) = y(k-n) \).

This means that the filtering of regressors solves the problem of bias in model parameters estimation in the case of the output error models, and also overcomes the problem of singularity of the matrix \( \psi^T(k)\psi(k) \) and with that connected problem of it’s inverse, i.e., calculation of the covariance matrix \( P(k) \) [47].

D. SEQUENCE OF STEPS IN WHOLE OPERATING DOMAIN

To obtain information about the behavior of the process throughout the whole operating range, the sequence of changing steps is used to excite the process. This sequence is generated between the minimum value \( u_{min} \) and the maximum value \( u_{max} \), where the step size \( \Delta u \) is given. The duration of one step, i.e., the hold time \( k_h \) should be defined to enable the process output to reach the steady-state. This means approximately 4 to 5 dominant time constants of the process. The approximate dominant time constant of the process is therefore a priori knowledge about the process. When the step is changed to a new value, the covariance matrix \( P(k) \) is initialized to \( P(0) \) and a new identification cycle is started to estimate the parameters of the current local linear model.

III. EVOLVING CLUSTERING IN INPUT-OUTPUT DATA DOMAIN

The recursive least-squares method in an extended form is used to estimate the local linear model parameters of each neuro-fuzzy rule, i.e., the consequent part. The premise of the rule is defined by an evolving clustering principle based on Gaussian probability distribution [40]. In this paper we are dealing with a processes where nonlinearity depends only of the operating point in the steady state. As will be shown in the next subsection, the adding and merging mechanisms are related to the input signal and the static gain of the process, respectively. Due to this, the clustering is realized in input-output domain, i.e., it is realized of the following data set:

\[
z(k) = [u(k) \; y(k)]^T, \; k = 1, \ldots, n \tag{13}
\]

where each cluster is defined by the center \( \mu \), the covariance matrix \( \Sigma \) and the number of samples \( n \). Therefore, the cluster is fully represented with the triplet \((\mu, \Sigma, n)\).

The center of the cluster is a two-dimensional vector, i.e., \( \mu = [\mu_u, \mu_y] \). The covariance matrix \( \Sigma \) is defined as follows (in the matrix form):

\[
\Sigma = \frac{1}{n-1}(Z - EM)(Z - EM)^T, \quad \tag{14}
\]

where \( Z \) represents the data dimension \( n \times 2 \) \(( Z^T = [z(1), \ldots, z(n)] )\), \( M \) is a diagonal matrix \( M = diag(\mu_u, \mu_y) \) and \( E \) the matrix of dimension \( n \times 2 \) with all elements equal to 1.

A. ADDING NEW CLUSTERS

In our approach, a new cluster is automatically added when the input signal \( u(k) \) changes, which is different from known methods where the clusters are added when a certain conditions are satisfied. That is when \(|u(k) - u(k-1)| > 0 \)
the number of clusters \( c \) is increased to \( c + 1 \), the number of elements in the cluster is initialized to \( n_c = 1 \), and the center and the covariance matrix of the cluster are initialized to \( \mu_c = z(k) \) and \( \Sigma_c = 0 \), respectively. To avoid possible unstable situations, the covariance matrix \( \Sigma_c \) could be initialized with a diagonal matrix with small epsilon entries (e.g. 0.05).

B. RECURSIVE ADAPTATION OF CLUSTER PARAMETERS

If the input signal \( u(k) \) does not change, i.e., \( u(k) - u(k-1) = 0 \), then a new measured input-output sample \( z(k) \) is used to adapt the center and covariance matrix of the cluster. This means that the number of samples in the cluster is increased, and the cluster mean and covariance matrix are adapted. Therefore, in our notation, the center of the \( j \)-th cluster with \( n_j \) samples is denoted as \( \mu_j^{n_j} \), and \( \Sigma_j^{n_j} \) denotes the corresponding covariance matrix.

The center and the covariance matrix are updated recursively. The update is performed in the following steps. First, the difference between the current sample and the current center is calculated:

\[
e_j(k) = z(k) - \mu_j^{n_j}. \tag{15}
\]

Next, the center is updated

\[
\mu_j^{n_j+1} = \mu_j^{n_j} + \frac{1}{n_j+1}e_j(k). \tag{16}
\]
After that, the states of the un-normalized covariance matrix are computed as:
\[ S_{j}^{n_{j}+1} = S_{j}^{n_{j}} + \epsilon_{j}(k) (z(k) - \mu_{j}^{n_{j}+1})^{T} \]  
(17)
and the covariance matrix is then calculated as:
\[ \Sigma_{j}^{n_{j}+1} = \frac{1}{n_{j}} S_{j}^{n_{j}+1}. \]  
(18)

C. MERGING CLUSTERS

The neuro-fuzzy model is now completely defined by a set of triplets \((\mu_{j}, \Sigma_{j}, G_{j}(z, \theta_{j}))\), \(j = 1, ..., c\), in which \(G_{j}(z, \theta_{j})\) stands for the transfer function of \(j\)-th linear model. The parameters of transfer function are given in \(\hat{\theta}_{j}\).

One of the goals of neuro-fuzzy modeling is a requirement for a parsimonious model. Therefore, a mechanism of similarity (or better dissimilarity) between neighboring (two consecutive) local linear models is necessary. The dissimilarity measure is calculated by comparing the frequency responses of these models. In the following equation, the dissimilarity measure is calculated by comparing the frequency responses of these models. In the following equation, the dissimilarity measure is calculated by comparing the frequency responses of these models.

The dissimilarity measure is calculated using Eqs. 21, 22, 23, 24, and 25. The proposed calculation enables the exact calculation of the joint cluster covariance matrix without storing the old data samples. The only information needed is the triplet of cluster parameters.

When the clusters are merged, the parameters of local linear models should also be merged. This is done by calculating the mean of both vectors of parameters as follows:
\[ \hat{\theta}_{pq} = \frac{\hat{\theta}_{p} + \hat{\theta}_{q}}{2}. \]  
(26)

The complete merging algorithm is presented in Alg. 1.

Algorithm 1 Merging of clusters.

1: Input: \(G_{p}(1)\) and \(G_{q}(1)\)
2: Initialization: \(\kappa_{merge}\)
3: Computation of frequency responses dissimilarity \(\kappa_{pq}\):
4: \(\kappa_{pq} = |\hat{G}_{p}(1) - \hat{G}_{q}(1)|\)
5: if \(\kappa_{pq} < \kappa_{merge}\)
6: Merge clusters \(p\) and \(q\): \(\mu_{pq}, \Sigma_{pq}, n_{pq}\)
7: Merge local model parameters: \(\hat{\theta}_{pq}\)
8: \(c \leftarrow c - 1\)
9: end

IV. EVOLVING IDENTIFICATION OF NEURO-FUZZY MODEL

The complete evolving identification algorithm of neuro-fuzzy model is described in Alg. 2 and the corresponding flowchart is presented in Fig. 1. The user-defined parameters in the proposed algorithm are the following: \(A_{f}(z)\) is denominator of initial filter transfer function to filter the regressors, \(k_{hf}\) is the hold time of constant step value, \(\Delta u\) is the step change in the sequence of input signal, and \(\kappa_{merge}\) stands for the threshold of dissimilarity to merge the clusters and the corresponding consequent local linear model parameters. The vector of parameters \(\theta\) can be always initialized to the zero vector.

V. SIMULATION OF NEURO-FUZZY MODELS IN PARALLEL

The parallel model or simulation model means that the predicted output should be calculated based on delayed model outputs as follows:
\[ \hat{y}(k) = f_{FM}(u(k-1), \ldots, u(k-m), \hat{y}(k-1), \ldots, \hat{y}(k-n)). \]  
(27)
where \(f_{FM}\) stands for the neuro-fuzzy model, and \(\hat{y}(k - i), i = 1, ..., n\) are delayed simulated model outputs.
Algorithm 2 Evolving identification of neuro-fuzzy model.

1. Input: $z^T(1) = [u(1)\ y(1)]$
2. User-defined parameters: $k_0, \Delta u, \kappa_{merge}, A_f(z), K_{min}$
3. Initialization: $c \leftarrow 1, \mu_c \leftarrow z(1), k \leftarrow 1$
4. repeat $k \leftarrow k + 1$
5. Generation of process input: $u(k)$
6. Acquisition of process output: $y(k)$.
7. Filtering: $1/A_f(z) : u(k) \rightarrow u_f(k)$ and $y(k) \rightarrow y_f(k)$
8. Regressor (RLS): $\varphi^T(k) = [u_f(k-1) \cdots y_f(k-n) \ 1]$
9. Regressor for clustering: $z^T(k) = [u(k)\ y(k)]$
10. if $|u(k) - u(k-1)| = 0$
11. $k_f \leftarrow k_f + 1$
12. Update of last cluster $c$ with new sample
13. $e_c(k) \leftarrow z(k) - \mu_c$
14. $\mu_c \leftarrow \mu_c + \frac{1}{n_c+1}e_c$
15. $S_c = S_c + e_c(z(k) - \mu_c)^T$
16. $n_c \leftarrow n_c + 1$
17. RLS: Recursive calculation of parameters
18. $e(k) = y_f(k) - \varphi^T(k)\hat{\theta}(k-1)$
19. $\gamma(k) = \varphi^*(k)\hat{\varphi}(k+1)^T\hat{\gamma}(k-1)\varphi(k)$
20. $P(k) = (I - \gamma(k)\varphi^T(k))\hat{P}(k-1)$
21. $\hat{\theta}(k) = \hat{\theta}(k-1) + \gamma(k)e(k)$
22. else
23. $k_f = 0$
24. if $c > 1$
25. Start algorithm: Merging clusters
26. end
27. Add and initialize new cluster
28. $c \leftarrow c + 1$
29. $n_c \leftarrow 1$
30. $\mu_c \leftarrow z(k)$
31. RLS: Reset of covariance matrix $P(k)$
32. $P(k) = P(0)$
33. end
34. Calculation of roots: $z_i, A(z_i) = 0, i = 1, \ldots, n$
35. if $\max_i |z_i| < 1$ and $k_f > K_{min}$
36. Adapt the filter $A_f(z) \leftarrow A(z)$
37. end
38. until end of data stream

As already mentioned, the neuro-fuzzy model consists of $c$ rules that are defined by a set of triplets $(\mu_j, \Sigma_j, G_f(z, \Theta_j)), j = 1, \ldots, c$. The information about the validity of certain local linear model is given by the cluster center and covariance matrix, $\mu_j$ and $\Sigma_j$.

For prediction and simulation use of the model, the clusters, given in input-output domain, should be projected into the input variable domain only. The input domain is obtained by singular value decomposition of covariance matrix $\Sigma_j$ as follows:

$$\Sigma_j = \begin{bmatrix} p_{1u} & p_{2u} \\ p_{1y} & p_{2y} \end{bmatrix} \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \begin{bmatrix} p_{1u} & p_{2u} \\ p_{1y} & p_{2y} \end{bmatrix}^T$$

where $p_{iu}, p_{iy}, i = 1, 2,$ represent the input and output components of the first, and second eigenvectors $(i = 1, 2)$, and $\sigma_1^2$ and $\sigma_2^2$ are corresponding eigenvalues. The projection of the $j$-th cluster to the input domain, defined as $\delta_j$, equals to the maximum of input components of both principal eigenvectors, and is defined as follows:

$$\delta_j = \max(p_{1u}\sigma_1, p_{2u}\sigma_2)$$

The membership value obtained from input projected clusters is defined as follows:

$$\gamma_j(k) = e^{-\frac{1}{\delta_j}((u(k) - \mu_{ij})^T(u(k) - \mu_{ij}))}$$
In the context of approximation, the neuro-fuzzy model in a certain operating point is obtained by a linear combination of individual local linear models. The weights of the linear combination are obtained from the membership degrees of the individual clusters. This leads to the following estimation of the model parameters:

$$\hat{\theta}(k) = \sum_{j=1}^{c} \gamma_j(k) \hat{\theta}_j$$

(31)

where $\hat{\theta}(k)$ represents the linear combination of model parameters at a certain moment, and $\hat{\theta}_j, j = 1, \ldots, c$ represents local linear model parameters.

The simulated output of the neuro-fuzzy model is then calculated as follows:

$$\hat{y}(k) = \hat{\varphi}(k)^T \hat{\theta}(k), \quad k = n + 1, \ldots, N$$

(32)

where $\hat{\varphi}$ represents the regressor, which consists of delayed estimated model output values as follows:

$$\hat{\varphi}(k)^T = [u(k-1), \ldots, u(k-m), -\hat{y}(k-1), \ldots, \hat{y}(k-n), 1]$$

(33)

and for $k = 1, \ldots, n$ the model is initialized with measured process output values $y(1), \ldots, y(n)$.

In the rule-based notation, the neuro-fuzzy model is given as follows

$$R_j : \text{if } u(k) \text{ is } Z^u_j \text{ then } \hat{y}(k) = \hat{\varphi}(k)^T \hat{\theta}_j$$

(34)

where $Z^u_j$ defines the projection of cluster to the input domain.

VI. EXAMPLES OF EVOLVING NEURO-FUZZY IDENTIFICATION

A. SIMULATION EXPERIMENT

In this subsection, the results of the algorithm are shown on identification of simple second order nonlinear dynamic process of a Hammerstein structure:

$$y(k) = 1.7y(k-1) - 0.85y(k-2) + 0.15 \arctan(u(k-1)).$$

(35)

The choice for the $\arctan(\cdot)$-nonlinearity is motivated by the observation that many real processes exhibit such a type of saturation behavior. The Hammerstein structure rises, for example, when the actuator of a plant introduces the dominant nonlinear effect to the overall system and possesses a saturation characteristics. The input $u$ will vary between $-4$ and $4$, and the process is excited by a sequence of steps.

The linearization in the operating point shows that the system express a strong nonlinearity. The gain of the process is calculated as $K = \frac{1}{1+u_0^2}$ and changes between $K = 1$ at $u_0 = 0$ and $K = \frac{1}{17}$ at $u_0 = \pm 4$. The output of the process in Eq. 35 is corrupted by noise with the noise model $H(q) = 1$, i.e. $y(k) = B(q)/A(q)u(k) + w(k)$, where $w(k)$ represents the white noise with varying statistical property $N(0, n_0/\{1 + u^2\})$ where $n_0 = 0.04$.

The results of identification in the noisy data stream environment were made for the input data space from $u_{min} = -4$ to $u_{max} = 4, k_H = 250, \Delta u = 0.5, K_{min} = 10, \kappa_{merge} = 0.30$, and the denominator of initial filter transfer function is chosen as $A_f(z) = 1 - 1.8z^{-1} + 0.81z^{-2}$. By our experience, the best practice when choosing the filter parameters, is to choose two equal stable poles (when dealing with second order system). That means that the poles lie into the unite circle ($|z| < 1$), e.g. $(1 - 0.9z^{-1})(1 - 0.9z^{-1}) = 1 - 1.8z^{-1} + 0.81z^{-2}$. The final cluster in input-output domain are shown in Fig. 2, the time courses of the estimated model parameters are shown in Fig. 3. In Fig. 4 the simulated model output is presented to show the quality of the obtained neuro-fuzzy model.

The example with merging threshold $\kappa_{merge} = 0.30$ results in modelling with only three rules and an adequate size of the modelling error, i.e. the merging threshold is quite high. The comparison was made with the example where
the merging threshold was chosen to be $\kappa_{\text{merge}} = 0.10$. The clusters in the input-output domain are shown in Fig. 5, where eight clusters are generated. It is shown that when choosing smaller merging parameter the obtained simulated neuro-fuzzy model is expected to be better.

The worst and the best frequency responses of the local linear models are compared with the frequency responses of the linearized process at the same operating point ($u_{\text{w}}$—worst, $u_{\text{b}}$—best) and plotted in Fig. 6, and it can be observed that both are very close to the theoretical values.

Another comparison is given where the regressors are filtered by a constant filter. The filter is equal to the original choice during the recursive computation of the parameters, i.e., $A_f(z) = 1 - 1.8z^{-1} + 0.81z^{-2}$. The merging threshold $\kappa_{\text{merge}} = 0.15$ was implemented, obtaining seven local linear models. The resulting worst and best frequency responses of the corresponding local linear models are shown in Fig. 7. It is obvious that the model parameters do not converge to the right values, i.e., they are biased. The main differences occur at higher frequencies.

**B. REAL PLANT EXPERIMENT**

In this subsection, the identification results are shown on a real system of a plate heat exchanger (PHE). The basic task of a PHE system is to transfer heat between two fluids, usually water or glycol. For a better understanding of the whole process of heat transfer, the schematic of the process is shown in Fig. 8. In our case, the PHE consists of two separate water circuits. The water in the first circuit is heated by an electric heater controlled by an on-off thermostat. Therefore, the temperature $T_{ec}(k)$ of the primary water circuit oscillates...
around the set point [38] (see second graph in Fig. 10). The water flow in the primary circuit is controlled by the motor-driven valve $V_1$ which represents input to the process $u(k)$. The secondary circuit has a constant water flow $F_p(k)$ and a constant inlet temperature $T_{sp}(k)$. The outlet temperature $T_{sp}(k)$ represents a controlled variable $y(k)$.

As mentioned above, the process input of the PHE system is the motor-driven valve $V_1$ and the process output is the temperature $T_{sp}(k)$. The statistic characteristics with nonlinear behavior of the PHE plant is shown in the first graph of Fig. 9. Moreover, the process gain also shows strong nonlinear behavior (second plot in Fig. 9). Actually, the value of merging parameter $\kappa_{merge} = 0.2$ (see algorithm 1) is chosen according to the process gain of the plant. The value of the second parameter, $K_{min} = 10$, is the same as in the simulation experiments.

The identification results of the PHE process are shown in Fig. 10, where the bottom graph shows the comparison between the model $\hat{y}(t)$ and the process output $y(t)$. Despite the negative and oscillatory influence of the reservoir temperature $T_{ec}(k)$ (middle plot in Fig. 10) we can conclude that the identification procedure efficiently deal with the nonlinearity of the process. Finally, Fig. 11 shows the number of identified clusters in the input-output space.

### VII. CONCLUSION

The proposed new approach for identifying neuro-fuzzy models, based on a filtered recursive least-squares method combined with incrementally evolving Gaussian clustering, was presented and shows great potential in constructing neuro-fuzzy models for nonlinear dynamic processes. The model is generated online, and the algorithm requires only four user-defined parameters, which are easy to understand and therefore easy to tune. The fuzzification of the data space can be realized by using different criteria and therefore can be used in many different applications. The algorithm was tested on simulated Hammerstein-type nonlinear dynamic process and on a real plate heat exchanger system. The Hammerstein-type identification procedure is a suitable structure for a wide range of real processes. The algorithm is transparent, very easy to implement, can be easily extended to deal with higher-dimensional data sets, and can generate parsimonious models.

### REFERENCES

[1] Angelov P.P., and X. Zhou, “Evolving fuzzy-rule-based classifiers from data streams”, *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1462 – 1475, 2008.

[2] Angelov P.P., E. Lughofer, and X. Zhou, “Evolving Fuzzy Classifiers using Different Model Architectures”, *Fuzzy Sets and Systems*, vol. 159, no. 23, pp. 3160 – 3182, 2008.

[3] Wang Y., L. Chen, and J.P. Mei, “Incremental Fuzzy Clustering With Multiple Medoids for Large Data ”, *IEEE Trans., on Fuzzy Systems*, vol.
FIGURE 11. Real plant. The clusters in input-output data space in the case of $\gamma = 0.20$.

22, no. 6, pp. 1557–1568, 2014.

[4] Havens T.C., J. Bezdek, C. Leckie, L. Hall, and M. Palaniwami, “Fuzzy c-means algorithms for very large data”, IEEE Trans. on Fuzzy Systems, vol. 20, no. 6, pp. 1130–1146, 2012.

[5] Angelov P., Lugohe E., “Data-driven evolving fuzzy systems using eTS and FLEXFIS: comparative analysis,” International Journal of General Systems, vol. 37, no. 1, Soft Computing Systems, pp. 45-67, 2008.

[6] Angelov P., “An approach for fuzzy rule-base adaptation using on-line clustering”, Int. J. Approx. Reasoning, vol. 35, no. 3, pp. 275–289, 2004.

[7] Angelov P. D. F. Ivel and N. Kasabov, “Evolving intelligent systems—methodology and applications”, Wiley, New York, 2010.

[8] Dovžan D. and I. Škrjanc, “Recursive clustering based on a Gustafson-Kessel algorithm”, Evolving Systems journal, vol. 2, pp. 15 – 24, 2011.

[9] Dovžan D., V. Logar and I. Škrjanc, “Implementation of an Evolving Fuzzy Model (eFuoMo) in a Monitoring System for a Waste Water Treatment Process”, IEEE Trans. on Fuzzy Systems, vol. 23, no. 5, pp. 1761–1776, 2015.

[10] Fıle D. and O. Georjeva, “An extended version of the Gustafson-Kessel algorithm for evolving data stream clustering”, in: Evolving Intelligent Systems: Methodology and Applications, Eds.: P. Angelov, D. File, A. Kasabov, John Willey and Sons, IEEE Press Series on Computational Intelligence, pp. 273–300, 2010.

[11] Škrjanc I., S. Oraza, T. Ban and D. Dovžan, “Large-Scale Cyber Attacks Monitoring using Evolving Cauchy Possibilistic Clustering”, Applied Soft Computing, vol. 62, pp. 2833–2839, 2017.

[12] Škrjanc I., S. Blažič, E. Lugohe and D. Dovžan, “Inner Matrix Norms in Evolving Cauchy Possibilistic Clustering for Classification and Regression from Data Streams”, Information Sciences, vol. 478, 2018, 2018.

[13] Soleimani-B H., C. Lucas, and B. N. Araabi, “System Identification: Theory for the User”, Prentice Hall PTR, Prentice Hall Inc., Upper Saddle River, New Jersey, 1999.

[14] Kasabov N.K. and Q. Song, “DENeSIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction,” IEEE Trans. Fuzzy Syst., vol. 10, no. 2, pp. 144–154, 2002.

[15] Lugohe E., “FLEXFIS: A Robust Incremental Learning Approach for Evolving Takagi–Sugeno Fuzzy Models”, IEEE Trans. on Fuzzy Systems, vol. 16, no. 6, pp. 1393–1410, 2008.

[16] Lugohe E., Pratama, M. and Škrjanc I., “Incremental Rule Splitting in Generalized Evolving Fuzzy Systems for Autonomous Drift Compensation”, IEEE Trans. on Fuzzy Systems, vol. 26(4), pp. 1854–1865, 2018.

[17] Pratama M., S. G. Anavatti, P. Angelov, and E. Lugohe, “PANFIS: A novel incremental learning machine,” IEEE Trans. Neural Netw. Learning Syst., vol. 25, no. 1, pp. 55–68, 2014.

[18] Wang D. and F. Ding, “Auxiliary model-based least-squares identification for Hammerstein output-error systems,” Systems and Control Letters, vol. 56, no. 5, pp. 373–380, 2007.
[44] Ding F., X. Liu, and J. Chu, “Gradient-based and least-squares-based iterative algorithms for Hammerstein systems using the hierarchical identification principle,” IET Control Theory and Applications, vol. 7, no. 2, pp. 176–184, 2013.

[45] Wang D. and F. Ding, “least-squares based and gradient based iterative identification for Wiener nonlinear systems,” Signal Processing, vol. 91, no. 5, pp. 1182–1189, 2011.

[46] Wang D. and F. Ding, “Hierarchical least-squares estimation algorithm for Hammerstein-Wiener systems,” IEEE Signal Processing Letters, vol. 19, no. 12, pp. 825–828, 2012.

[47] Bitmead R.R., “Persistence of Excitation Condition and the Convergence of Adaptive Systems,” IEEE Transaction on Information Theory, vol. 30, no. 2, pp. 183 – 192, 1984.

GORAN ANDONOVSKI received the B.Sc., and Ph.D. degrees in electrical engineering from the Faculty of Electrical and Computer Engineering, University of Ljubljana, Ljubljana, Slovenia, in 2012, and 2018, respectively. He is currently a Teaching assistant of Automatic Control at the Faculty of Electrical Engineering, University of Ljubljana, and a researcher at the Department for Control Systems and Cybernetics, His main research interests include evolving data-based model identification, control and fault detection and isolation. In 2017 he received the first price at Autonomous Learning Machines Competition (ALMA) organized by IEEE Systems, Man, and Cybernetics Society.

EDWIN LUGHOFER received his PhD-degree from the Johannes Kepler University Linz (JKU) in 2005. He is currently Key Researcher with the Fuzzy Logic Laboratorium Linz / Institute for Mathematical Methods in Medicine and Data Based Modeling (JKU) in the Softwarepark Hagenberg, He has participated in several basic applied research projects on European and national level, with a specific focus on topics of Industry 4.0 and Factories of the Future. He has published more than 220 publications in the fields of evolving fuzzy systems, machine learning and vision, data stream mining, chemometrics, active learning, classification and clustering, fault detection and diagnosis, quality control and predictive maintenance, including 100 journals papers in SCI-expanded impact journals, a monograph on ‘Evolving Fuzzy Systems’ (Springer, Heidelberg Berlin), an edited book on ‘Learning in Non-stationary Environments’ (Springer, New York) and an edited book on ‘Predictive Maintenance in Dynamic Systems’ (Springer, New York). His publications received 7100 references achieving an h-index of 44. He is member of the editorial boards and/or associate editor of the international journals Information Sciences, IEEE Transactions on Fuzzy Systems, Evolving Systems, Information Fusion, International Journal of Big Data and Analytics in Healthcare, Soft Computing and Complex and Intelligent Systems, the general chair of the IEEE Conference on EASIS 2014 in Linz, the publication chair of IEEE EASIS 2015, 2016, 2017, 2018, 2020 and 2022, the program co-chair of the International Conference on Machine Learning and Applications (ICMLA) 2018, the tutorial chair of IEEE SSCI Conference 2018, the publication chair of the 3rd INNS Conference on Big Data and Deep Learning 2018, and the Area chair of the FUZZ-IEEE 2015 conference in Istanbul. He co-organized around 12 special issues and more than 20 special sessions in international journals and conferences. In 2006 he received the best paper award at the International Symposium on Evolving Fuzzy Systems, and in 2013 the best paper award at the IFAC conference in Manufacturing Modeling, Management and Control (800 participants) and in 2016 the best paper award at the IEEE Intelligent Systems Conference.

IGOR ŠKRJANC received B.S., M.S. and Ph.D. degrees in electrical engineering, in 1988, 1991 and 1996, respectively, at the Faculty of Electrical and Computer Engineering, University of Ljubljana, Slovenia. He is currently a Full Professor with the same faculty and Head of Laboratory for Autonomous and Mobile Systems. He is lecturing the basic control theory at graduate and advanced intelligent control at postgraduate study. His main research areas are adaptive, predictive, neuro-fuzzy and fuzzy adaptive control systems. His current research interests include also the field of autonomous mobile systems in sense of localization, direct visual control and trajectory tracking control. He has published 126 papers with SCI factor and 31 other journal papers. He is co-author and co-author of 274 conference contributions, 37 lectures at foreign universities. He is also mentor at 9 Phd thesis, 4 MSc thesis and 36 diploma works. And co-mentor of 2 Phd thesis and 4 MSc thesis. He is author of 20 university books, 50 international and domestic projects and 5 patents and 2 patent applications. In 1988 he received the award for the best diploma work in the field of Automation, Bedjaniči award, in 2007 the award of Faculty of Electrical Engineering, University of Ljubljana, Slovenia. He is currently a Full Professor in the field of intelligent control, in 2012 the 1st place at the competition organized by IEEE Computational Society, Learning from the data in the frame of IEEE World Congress on Computational Intelligence 2012, Brisbane, Australia: Solving the sales prediction problem with fuzzy evolving methods, and in 2013 the best paper award at IEEE International Conference on Cybernetics in Lausanne, Switzerland. In 2017 he won the first prize on Autonomous Learning Machines Competition (ALMA) organized by IEEE SMC Society. In 2008 he received the most important Slovenian research award for his work in the area of computational intelligence in control – Zois award. In 2016 he received a Golden plaque of the University of Ljubljana for outstanding merits in the development of scientific and pedagogical work and for strengthening the reputation of the University in the world. In year 2009 he received a Humboldt research award for long term stay and research at University of Siegen. He is Humboldt research fellow, research fellow of JSPS and Chair of Excellence at University Carlos III of Madrid. He is also a member of IEEE CIS Standards Committee, IFAC TC 3.2 Computational Intelligence in Control Committee and Slovenian Modelling and Simulation society and Automation Society of Slovenia. He also serves as an Associated Editor for IEEE Transaction on Neural Networks and Learning System, IEEE Transaction on Fuzzy Systems, the Evolving Systems journal and International journal of artificial intelligence. In 2017 he organized an IEEE Conference on Evolving and Adaptive Intelligent Systems - EAIS 2017.