ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/321216179

Evolving model identification for process monitoring and prediction of non-linear systems

Article *in* Engineering Applications of Artificial Intelligence · November 2017

DOI: 10.1016/j.engappai.2017.10.020

CITATIONS
10

READS
160

4 authors, including:

Goran Andonovski
University of Ljubljana
17 PUBLICATIONS **79** CITATIONS
SEE PROFILE

Gašper Mušič
University of Ljubljana
63 PUBLICATIONS **543** CITATIONS
SEE PROFILE

Saso Blazic
University of Ljubljana
136 PUBLICATIONS **1,434** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Chair of Excellence, University Carlos III of Madrid, Spain View project

Project Develop a new control law for a mobile manipulator robot View project

All content following this page was uploaded by Goran Andonovski on 12 February 2019.

The user has requested enhancement of the downloaded file.

ARTICLE IN PRESS

Engineering Applications of Artificial Intelligence ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Evolving model identification for process monitoring and prediction of non-linear systems

Goran Andonovski *, Gašper Mušič, Sašo Blažič, Igor Škrjanc

Faculty of Electrical Engineering, University of Ljubljana, Slovenia

ARTICLE INFO

Keywords:
Cloud-based identification
Evolving fuzzy system
Process monitoring
Tennessee Eastman model
Water chiller plant

ABSTRACT

This paper tackles the problem of model identification for monitoring of non-linear processes using evolving fuzzy models. To ensure a high production quality and to match the economic requirements, industrial processes are becoming increasingly complicated in both their structure and their degree of automation. Therefore, evolving systems, because of their data-driven and adaptive nature, appear to be a useful tool for modeling such complex and non-linear processes. In this paper the identification of evolving cloud-based fuzzy models is treated for process monitoring purposes. Moreover, the evolving part of the algorithm was improved with the inclusion of some new cloud-management mechanisms. To evaluate the proposed method two different processes, but both complex and non-linear, were used. The first one is a simulated Tennessee Eastman benchmark process model, while the second one is a real water-chiller plant.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

In general, modern industrial processes are typical dynamic systems with complex structures and frequently operating under a variety of environmental conditions. For successful and optimal operation of any process, it is important to detect, or even better, to predict undesired events as early as possible. Due to this, model-based methods play a crucial rule in the field of process monitoring and fault detection (Isermann, 1984). These methods can be used for measurable signals prediction and for non-measurable parameters estimation. The method proposed in this paper is used for both, prediction of measured signals and for estimation of non-measurable parameters (performance production indicators, pPIs).

The methods of process monitoring and fault diagnosis can be classified into three general categories: methods based on mathematical/physical knowledge of the process; statistical data-driven methods; and data-driven model-based methods.

The methods based on the mathematical/physical knowledge of the process have been successfully applied in different industrial applications (Isermann, 2004; Gertler, 1998; Venkatasubramanian et al., 2003b; Campos-Delgado and Espinoza-Trejo, 2011; Huang et al., 2012; He et al., 2013). This type of methods use an *a priori* knowledge based on fundamental understanding of the physics of the process. Beside their wide-range usage (Isermann, 2011), the methods have several disadvantages. They are limited to linear models and in some cases to very specific nonlinear models (using linear approximation). Other problems are disturbance simplification, parameter drifts, *a priori* estimation of classification errors, adaptability to varying process' conditions, etc.

Due to the information revolution and data expansion new data-driven techniques have been investigated and developed. Data-based schemes for system monitoring mainly concentrate on the data collected from the processes. Statistical approaches (Qin, 2003) use this data to extract the knowledge and to detect the faults. Principle component analysis (PCA) (Li et al., 2000; Gertler and Cao, 2004; Chen et al., 2016) and partial least squares (PLS) (Li et al., 2010; Zhang et al., 2010; Chen et al., 2016) are two basic techniques. More recently, independent component analysis (ICA) (Zhang and Qin, 2008; Tsai et al., 2013) has received a lot of attention and has seen great success in practice and (Venkatasubramanian et al., 2003a; Yin et al., 2014) have provided a review of the basic statistical data-driven approaches for process monitoring. In general, statistical data-based schemes can effectively monitor only the industrial processes when they operate under stationary conditions (Yin et al., 2014). However, this type of methods are not suitable to handle the complex process dynamics under changing environmental conditions.

The applicability of statistical data-based methods can be improved by considering the system dynamics (Chapter 3 in Simani et al. (2003)). Fault detection methods which combine the data-driven with model-based approaches, have been presented by Precup et al. (2015). As

* Corresponding author.
E-mail address: goran.andonovski@fe.uni-lj.si (G. Andonovski).

https://doi.org/10.1016/j.engappai.2017.10.020
Received 12 December 2016; Received in revised form 12 September 2017; Accepted 27 October 2017
Available online xxxx
0952-1976/© 2017 Elsevier Ltd. All rights reserved.

Please cite this article in press as: Andonovski G., et al., Evolving model identification for process monitoring and prediction of non-linear systems. Engineering Applications of Artificial Intelligence (2017), https://doi.org/10.1016/j.engappai.2017.10.020.

dynamic models are required, the evolving-based identification methods play an important role (Lughofer, 2011, 2015). In this case the process model is not known *a priori* but is identified recursively from the data streams. Furthermore, the acquired model could be used for the prediction of the future system behavior.

The evolving methods can be divided into different categories according to their structure, ability of learning, level of autonomous adapting, etc. In the field of fault detection and process monitoring, most of the evolving methods use Takagi–Sugeno (Takagi and Sugeno, 1985) type of fuzzy structure to construct the process model (Lughofer and Guardiola, 2008; Chivala et al., 2010; Petković et al., 2012; Lemos et al., 2013; El-Koujok et al., 2014; Dovžan et al., 2015). In this methods the membership function (in a form of data clusters) usually follow the Gaussian distribution using a predefined distance measure.

By introducing the new simplest form of fuzzy system (Angelov and Yager, 2011), named AnYa, a new branch of evolving methods have been developed. AnYa fuzzy system uses a non-parametric (cloud-based) antecedent part which does not require any explicit definition of the membership function or even *a prior* assumption of its form. The membership functions and the evolving mechanisms are based on the relative density of the current data according to the existing clouds. The data clouds represent sets of previous data points with similar properties. Contrary to the clusters, clouds do not have any boundaries and they directly and exactly represent all previous data samples. More information about differences between the data clouds and the clusters can be found in (Angelov and Yager, 2011).

The AnYa based methods (Angelov et al., 2013; Costa et al., 2013; Rosa et al., 2014) use local and global density to evolve the structure, while the methods in (Škrjanc et al., 2014; Blažič et al., 2014) use just local density with simple threshold to evolve the structure. The proposed method in this paper represents an extension of the latter methods with introducing new evolving mechanisms.

Based on the AnYa fuzzy system a fault detection method was proposed by Costa et al. (2014a, b, 2015) which uses a recursive density estimation to detect novelty in a statistical manner (without including any dynamics of the process into the model).

For the purpose of process monitoring in this paper we propose an improved evolving fuzzy model based on AnYa fuzzy system. This method uses the ability of evolving the fuzzy structure to cope with changing environmental conditions. On the other hand an NARX model is used to deal with the process dynamics. This is the main advantage over the existing process monitoring methods based on AnYa fuzzy system (Costa et al., 2015; Precup et al., 2015). The new evolving mechanisms are able to protect from addition of new clouds (rules) based on outliers. Moreover, a new mechanism for removing the "less active" and the "less informative" clouds is introduced. The *activity* is a property of the cloud and it is defined as a relative number of data samples associated with a particular cloud from its creation. On the other hand, the second removing mechanism deletes the clouds that has obtained less information and are less active in comparison with the other clouds.

The proposed cloud-based model, as a tool for estimation of the non-measurable parameters, is tested on a simulated input/output data acquired from the Tennessee Eastman (TE) (Downs and Vogel, 1993) benchmark process model. The non-measured production objectives of the systems are defined through the production performance indicators (pPIs), namely, *Cost*, *Production* and *Quality* (Glavan et al., 2012). The models for these three pPIs are identified with the proposed fuzzy-cloud-based method and the results are compared with the eFuMo identification tool proposed by Dovžan et al. (2012, 2015) and with the NNSYSID neural network tool (Norgaard et al., 2000). The main goal is to monitor the process by detecting potential undesired future trends based on the estimated production performance indicators.

A practical example to test the usability of the proposed method is a real water-chiller plant (WCP) located in a local factory. The proposed method is used as a model identification tool for undesired events prediction. Using the real data, two variables are identified: the WCP's power production and the factory's power consumption. The goal of monitoring these indicators is to predict the future behavior of the system in order to prevent unnecessary short-time start-ups of the water chillers. This can improve the overall efficiency of the whole system.

The paper is organized as follows. In Section 2 the cloud-based identification method is presented, while in Section 3 an improved evolving mechanism for adding and removing clouds is presented. Section 4 introduces the experimental results for a Tennessee Eastman process, while in Section 5 the practical results for a water chiller plant are presented. Finally, in Section 6, the main ideas and results are summarized.

2. Cloud-based identification of a dynamic system

2.1. Fuzzy-rule-based model

Fuzzy systems are general approximation tools for the modeling of non-linear dynamic processes. In this paper we use a fuzzy-rule-based system with a non-parametric antecedent part presented by Angelov and Yager (2011). The main difference is the simplified antecedent part that relies on the data relative density. The rule-based form of the ith rule is defined as:

$$\mathcal{R}^i : \text{IF} \quad (\boldsymbol{x}_f \sim X^i) \quad \text{THEN} \quad y^i = f^i(\boldsymbol{x}_f) \tag{1}$$

where the data sample (regression vector) $\boldsymbol{x}_f(k) = [y(k-1), \ldots, y(k-n_a), u(k-1), \ldots, u(k-n_b)]$ includes the delayed system inputs and outputs . The operator $\sim$ is linguistically expressed as '*is associated with*', which means that the current data $\boldsymbol{x}_f$ is related to one of the existing clouds X^i according to the membership degree (the normalized relative density of the data). The input and output orders are denoted as n_a and n_b, respectively. Note that the input $u(k)$ does not have an immediate influence on the output $y(k)$. The partial NARX model of the ith rule is defined as:

$$f^i(k) = \boldsymbol{\theta}^{i\,T} \boldsymbol{\psi}(k) \tag{2}$$

where the vector $\boldsymbol{\psi}(k) = \left[\boldsymbol{x}_f,\ 1\right]^T$ consists of the regression vector $\boldsymbol{x}_f$ (used for partitioning the data space) to which we usually add a regressor 1. The vector of parameters for the ith cloud (rule) is denoted as $\boldsymbol{\theta}^i = \left[a_1^i, \ldots, a_{n_a}^i,\ b_1^i, \ldots, b_{n_b}^i,\ r^i\right]^T$. Once we have declared all the parameter vectors $\boldsymbol{\theta}^i$ for each cloud ($i = 1, \ldots, c$) we can define the output of the system in a compact matrix form:

$$y(k) = \sum_{j=1}^{c} \beta^j(\boldsymbol{x}_f)\boldsymbol{\theta}^{j\,T}\boldsymbol{\psi}(k) = \boldsymbol{\beta}^T(\boldsymbol{x}_f)\boldsymbol{\Theta}^T\boldsymbol{\Psi}(k) \tag{3}$$

where c is the number of existing clouds [1] (fuzzy rules), $\boldsymbol{\beta}^T(\boldsymbol{x}_f) = \left[\beta^1,\ \beta^2, \ldots, \beta^c\right]$ is the vector of normalized relative densities determined between the current data $\boldsymbol{x}_f$ and all the existing clouds, and $\boldsymbol{\beta}^T$ will be discussed in the next subsection. The matrix $\boldsymbol{\Theta} = \left[\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \ldots, \boldsymbol{\theta}^c\right] \in \mathbb{R}^{(1+n_a+n_b)\times c}$ contains the vectors of the parameters for all the existing clouds.

2.2. Identification of the antecedent part

In this subsection we will describe an identification method for the non-parametric antecedent part of the fuzzy-rule-based system AnYa (Angelov and Yager, 2011). The method starts with zero fuzzy rules (clouds) and the first cloud is initialized with the first data $\boldsymbol{x}_f$ received. For each of the following data the normalized relative densities β^i are calculated and then the current data is associated with one of the existing clouds (according to the maximum density β^i, where

[1] We use the term 'existing clouds' because this method is an evolving one and the number of clouds changes when some requirements are fulfilled.

Please cite this article in press as: Andonovski G., et al., Evolving model identification for process monitoring and prediction of non-linear systems. Engineering Applications of Artificial Intelligence (2017), https://doi.org/10.1016/j.engappai.2017.10.020.

$i = 1, \ldots, c$) or a new cloud is added (evolving mechanisms are explained in Section 3).

Before calculating the vector $\boldsymbol{\beta}$ we need to calculate the local relative density, which is defined by a suitable kernel over the distances between the current data $\boldsymbol{x}_f(k)$ and all the data previously associated with the cloud. The Euclidean distance ($d_j^i(k) = \|\boldsymbol{x}_f(k) - \boldsymbol{x}_f^i(j)\|$) was chosen in this case (also used by Angelov and Yager 2011, Angelov et al. 2013, Škrjanc et al. 2014), but any other distance could also be used, e.g., the Mahalanobis distance was used by Blažič et al. (2015).

The local density $\gamma^i(k)$ of the current data $\boldsymbol{x}_f(k)$ with the ith cloud is defined using the Cauchy kernel as follows:

$$\gamma^i(k) = \frac{1}{1 + \frac{\sum_{j=1}^{M^i}(d_j^i(k))^2}{M^i}} \tag{4}$$

where M^i is the number of data points that belong to the ith cloud. Eq. (4) should be rewritten in the recursive form for easier implementation, as follows:

$$\gamma^i(k) = \frac{1}{1 + \|\boldsymbol{x}_f(k) - \boldsymbol{\mu}^i(k)\|^2 + \sigma^i(k) - \|\boldsymbol{\mu}^i(k)\|^2} \qquad i = 1, \ldots, c \tag{5}$$

where $\boldsymbol{\mu}^i(k)$ and $\sigma^i(k)$ denote the mean value vector and the mean-square length of the data vector from the ith cloud, respectively. Both of them, $\boldsymbol{\mu}^i(k)$ and $\sigma^i(k)$, can therefore be recursively calculated as:

$$\boldsymbol{\mu}^i(k) = \frac{M^i - 1}{M^i}\boldsymbol{\mu}^i(k-1) + \frac{1}{M^i}\boldsymbol{x}_f(k) \tag{6}$$

$$\sigma^i(k) = \frac{M^i - 1}{M^i}\sigma^i(k-1) + \frac{1}{M^i}\|\boldsymbol{x}_f(k)\|^2 \tag{7}$$

The mean value and the mean-square length of the data vectors for each new cloud are initialized as $\boldsymbol{\mu}^i(0) = \boldsymbol{x}_f(k)$ and $\sigma^i(0) = \|\boldsymbol{x}_f(k)\|^2$, respectively. We have to note here that only the properties, $\boldsymbol{\mu}^i(k)$ and $\sigma^i(k)$, of the chosen cloud are updated by using (6) and (7), while the properties of all the other clouds are kept constant.

Once we have calculated the local densities $\gamma^i(k)$ where $i = 1, \ldots, c$ we can calculate the degree of membership of the current data $\boldsymbol{x}_f(k)$ with the cloud X^i. This is done according to the normalized relative local density, which is defined as follows:

$$\beta^i(\boldsymbol{x}_f(k)) = \frac{\gamma^i(k)}{\sum_{j=1}^{c}(\gamma^j(k))} \qquad i = 1, \ldots, c \tag{8}$$

where c is the current number of existing clouds.

2.3. Identification of the consequent part

In the previous subsection we updated the properties, $\boldsymbol{\mu}^i(k)$ and $\sigma^i(k)$, of the chosen cloud and we calculated the degree of membership $\beta^i(\boldsymbol{x}_f(k))$. Moreover, in Section 2.1 we mentioned that for the consequent part we use an NARX model, the parameters of which are updated (identified) using the Recursive Weighted Least Squares (rWLS) method (Chapter 2 in Lughofer, 2011):

$$\begin{aligned}
\boldsymbol{\psi}(k) &= \left[\boldsymbol{x}_f(k), \quad 1\right]^T \\
\boldsymbol{P}^i(k) &= \frac{1}{\lambda_r}\left(\boldsymbol{P}^i(k-1) - \frac{\beta^i(k)\boldsymbol{P}^i(k-1)\boldsymbol{\psi}(k)\boldsymbol{\psi}^T(k)\boldsymbol{P}^i(k-1)}{\lambda_r + \beta^i(k)\boldsymbol{\psi}^T(k)\boldsymbol{P}^i(k-1)\boldsymbol{\psi}(k)}\right) \\
\boldsymbol{\theta}^i(k) &= \boldsymbol{\theta}^i(k-1) + \boldsymbol{P}^i(k-1)\boldsymbol{\psi}(k)\beta^i(k)\left(y(k) - \boldsymbol{\psi}^T(k)\boldsymbol{\theta}^i(k-1)\right) \\
& \qquad i = 1, \ldots, c
\end{aligned} \tag{9}$$

where $\boldsymbol{\psi}(k)$ is the extended regression vector, while the second and the third lines in (9) represent the updating mechanisms of the covariance matrix $\boldsymbol{P}^i(k)$ and the model parameters $\boldsymbol{\theta}^i(k)$, respectively. The parameter $\lambda_r \leq 1$ stands for the exponential forgetting factor, where the data that appears p samples ago is weighted by $(\lambda_r)^p$ (Nelles, 2001). The parameter λ_r also prevents the covariance matrix $\boldsymbol{P}^i(k)$ from becoming too small. It is initialized with a large positive definite matrix, usually a diagonal one ($\boldsymbol{P}^i(0) = \alpha\boldsymbol{I}$, $\alpha \gg 1$, $\boldsymbol{I} \in \mathbb{R}^{(1+n_a+n_b)\times(1+n_a+n_b)}$). Once we have identified/updated the parameters of the consequent part in the kth step, we can calculate the estimated output as follows:

$$\hat{y}(k) = \sum_{j=1}^{c}\beta^j(\boldsymbol{x}_f(k))\boldsymbol{\theta}^{jT}\boldsymbol{\psi}(k) = \boldsymbol{\beta}^T(\boldsymbol{x}_f(k))\boldsymbol{\Theta}^T\boldsymbol{\Psi}(k) \tag{10}$$

3. Evolving mechanisms of the cloud-based method

In this section the evolving procedure of the antecedent part is described. The cloud-based method is capable of evolving the structure in an on-line manner using different mechanisms.

3.1. Adding new clouds (fuzzy rules)

As we mentioned earlier, this method starts with zero clouds and the first cloud is initialized with the first data sample. Moreover, when some requirements are fulfilled a new cloud (fuzzy rule) can be added, and usually there are certain criteria (detection of non-linearity in the system, preventing the addition of clouds based on outliers, etc.). In our case we have four criteria for adding new clouds:

(1) The first criterion (Škrjanc et al., 2014) is related to the *closeness* of the current data to all of the existing clouds. In the case that all the local densities $\gamma^i(k)$ are smaller than some threshold γ_{max} than this criteria is fulfilled and is mathematically expressed as:

$$C_1^{add} = (\gamma_{max} > \max_i(\gamma^i(k)));$$

where $\gamma_{max} \in [0,\ 1]$. The value of this parameter was determined experimentally to 0.85, but in practice any other value from the range could be used, depending on the problem nonlinearity. A larger value of the parameter means that more clouds are created, while a smaller value results in adding just a few clouds (fuzzy rules).

(2) The second criterion is new and complements the first one and prevents new clouds being created based on outliers (see Fig. 1). We wait for a certain number τ_{add} of consecutive samples that satisfies the first criterion C_1^{add}, and after that we add a new cloud:

$$C_2^{add} = (k > k_{C_1} + \tau_{add});$$

where k_{C_1} is the number of consecutive samples fulfilling criterion C_1^{add}. The value of the parameter τ_{add} is equal to the dimensionality of the regression vector ($\tau_{add} = n_a + n_b$). If the value of τ_{add} is too large we need to think how we will deal with the data points k_{C_1} that should be stored in a buffer and used for initializing the next data cloud.

(3) The third criterion (Škrjanc et al., 2014) is the number of data samples n_{add} that have to pass from the time stamp k_{add} of the last added cloud:

$$C_3^{add} = (k > k_{add} + n_{add});$$

where n_{add} is one of the design parameters. This criterion freezes the adding mechanisms for n_{add} samples. This allows the newly created cloud to accumulate some information from the data. This helps the covariance matrix of the rWLS algorithm to be calculated properly.

(4) The last criterion (Škrjanc et al., 2014) is related to the maximum number of clouds c_{max} that can be added:

$$C_4^{add} = (c_{max} > c);$$

where c is the number of already-existing clouds. Actually this parameter can be set to infinity because of the adding and removing mechanism included in the evolving procedure. But

Please cite this article in press as: Andonovski G., et al., Evolving model identification for process monitoring and prediction of non-linear systems. Engineering Applications of Artificial Intelligence (2017), https://doi.org/10.1016/j.engappai.2017.10.020.

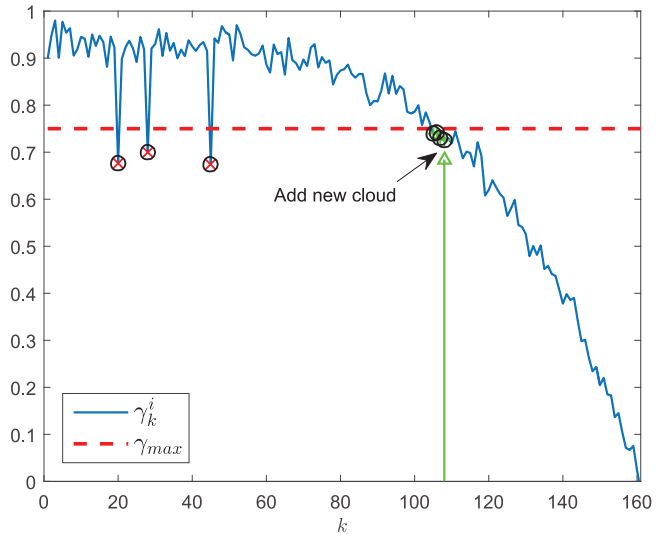

Fig. 1. The red dashed line represents the threshold γ_{max} and the blue line represents the value of maximal density $\max(\gamma^i(k))$ at each step k. The outliers fulfill criterion C_1^{add}, but new cloud is added after τ_{add} consecutive data points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

still, if the operator deals with known process structure (or at least he has a good assumption of it) this parameter can be set to an appropriate value.

Therefore, if all four criteria (C_1^{add} **and** C_2^{add} **and** C_3^{add} **and** C_4^{add}) are satisfied, a new cloud is added. For this adding mechanism we have to set three parameters γ_{max}, n_{add} and c_{max}.

3.2. Removing clouds (fuzzy rules)

Besides all of the criteria that try to prevent the addition of new clouds based on outliers, we defined two new properties, *activity* and *informativeness* to detect "less active" or "less informative" clouds (rules), respectively. The *activity* is defined as a relative number of data samples associated with the cloud, counted from the moment of the cloud's creation. The condition for removing clouds is expressed as:

$$C_1^{rem} = \left(\frac{M^i}{k - k^i} < \zeta \frac{1}{c} \right) \quad i = 1, \ldots, c;$$

where $\zeta \in [0, 1]$ is a constant parameter. If $\zeta = 0$ then removing mechanism is disabled, while if $\zeta = 1$ then with each new added cloud the previous one is removed. Choosing $\zeta = 1$ is not a reasonable solution, and therefore in practice the constant ζ should be within $[0, 1)$, usually 0.1 (Dovžan et al., 2015).

When removing clouds we should be very careful and conservative, because we do not want to remove a cloud containing useful information. Using just the first criterion C_1^{rem} could possibly lead to such a problem. Therefore, beside the criterion C_1^{rem} we propose an additional one for removing "less informative" clouds. The *non-informative* is a new property of the cloud and expresses the number of data samples that are most far away from a particular cloud. For each data point $\boldsymbol{x}_f(k)$ we calculate all the local densities and find the one with the minimum value ($\min_i(\gamma^i(k))$). The number of such points for each cloud is denoted as $\tilde{M}^i$. Finally, the second criterion is defined as:

$$C_2^{rem} = \left(1 - \frac{\tilde{M}^i}{k - k^i} < \frac{\zeta}{2} \right) \quad i = 1, \ldots, c;$$

where ζ is the same parameter as the one in C_1^{rem} and has the same value (0.1). The whole mechanism for removing a cloud is a logical combination (C_1^{rem} **and** C_2^{rem}).

The whole procedure for the fuzzy-cloud-based algorithm is explained through the flowchart presented in Fig. 2. This flowchart

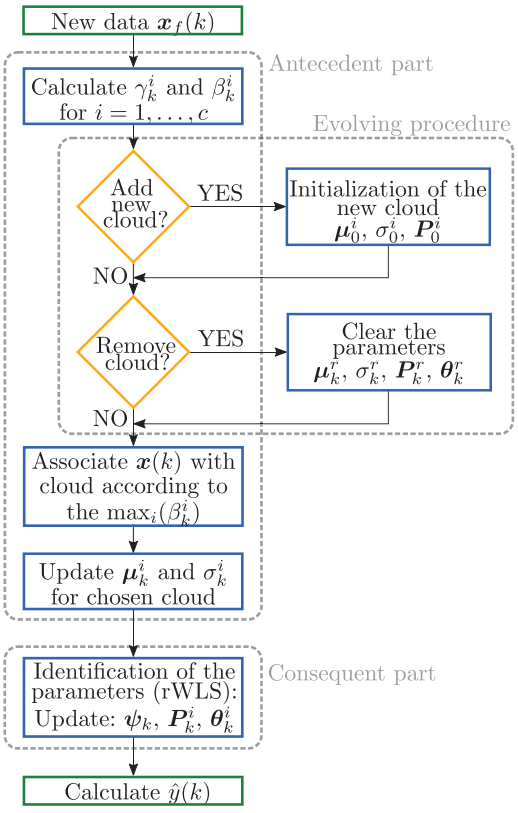

Fig. 2. Cloud-based identification procedure (flowchart for one data sample).

represents the procedure for one data sample, and the same steps are repeated for each data sample received. In Fig. 2 we can see three steps: the identifying of the antecedent part, the evolving mechanism (adding and removing clouds), and identifying the parameters of the consequent part.

4. Simulation example: Tennessee Eastman process

4.1. Problem description

The Tennessee Eastman (TE) process presented in (Downs and Vogel, 1993) is a well-known benchmark model of a real chemical production process.

TE is a complex non-linear, open-loop unstable process and it consists of 41 measured and 12 manipulative variables, which opens up a wide set of possible implementations. The process consists of five major unit operations: the reactor, the product condenser, a vapor–liquid separator, a recycle compressor and a product stripper. The five gaseous inputs, namely A, B, C, D and E, are fed to the reactor where they react to form liquid products. The products leave the reactor as vapors while the catalyst remains in the reactor. In the next phase using the cooler the products condense and they are separated in the vapor–liquid separator. The non-condensed components are returned to the reactor feed while the condensed move to the product stripper. At the end, the two products, G and H, exit the stripper. There are also the byproduct and the inert which are purged from the system in the vapor–liquid separator phase. The detailed information about the TE process could be found in (Downs and Vogel, 1993).

The production objectives for the TE process are defined through the *Cost* proposed by Downs and Vogel (1993), and *Production*, and *Quality* proposed by Glavan et al. (2013). Moreover, in (Glavan et al., 2013) the most relevant manipulative input variables were selected (see Table 1), which have the biggest impact on the selected production performance

Please cite this article in press as: Andonovski G., et al., Evolving model identification for process monitoring and prediction of non-linear systems. Engineering Applications of Artificial Intelligence (2017), https://doi.org/10.1016/j.engappai.2017.10.020.

Table 1
Process manipulative variables selected by Glavan et al. (2012).

Notation	Controlled variable setpoints
F_p	Production rate index
R_2	Striper level
R_3	Separator level
R_4	Reactor level
R_5	Reactor pressure
R_7	%C in purge
R_8	Recycle rate
R_9	Reactor temperature
r_2	D/E feed rates

indicators (pPIs). The regression vectors $\boldsymbol{x}_f(k)$ for each pPI (model) were selected as follows:

$$\begin{aligned} Cost &: \boldsymbol{x}_f(k) = [y(k-1), y(k-2), F_p(k-1), \\ &\quad F_p(k-2), F_p(k-4), R_4(k-1), R_7(k-1), \\ &\quad R_7(k-5), R_9(k-1), R_9(k-5), r_2(k-1), \\ &\quad r_2(k-4), r_2(k-5)]^T \\ Production &: \boldsymbol{x}_f(k) = [y(k-1), y(k-2), F_p(k-1), \\ &\quad F_p(k-5), R_4(k-1), R_4(k-4), R_7(k-1), \\ &\quad r_2(k-5)]^T \\ Quality &: \boldsymbol{x}_f(k) = [y(k-1), y(k-2), F_p(k-1), \\ &\quad R_4(k-1), r_2(k-5)]^T \end{aligned}$$

The main goal is to identify the defined pPIs using the evolving cloud-based method. The identification model is then used for on-line prediction of the production performance indicators.

4.2. Simulation experiment

To evaluate the proposed method three sets of input/output signals are acquired with a simulation of the TE process model. The first set of signals is called TRAIN and it is used to acquire the models for each output (learning phase). The other two sets (VAL1 and VAL2) are used for the validation (4-step prediction) of the obtained models. Furthermore, the estimated output $\hat{y}(k)$ is compared to the original output signal $y(k)$ and evaluated using the Mean Square Error (MSE), which is defined as follows:

$$MSE = \frac{1}{N}\sum_{k=1}^{N}(\hat{y}(k) - y(k))^2 \tag{11}$$

where N is the number of samples in the output signals.

As we mentioned above, the proposed cloud-based method is compared to the well-established fuzzy technique eFuMo, proposed by Dovžan et al. (2012) and with the artificial neural network (ANN) toolbox NNSYSID (Norgaard et al., 2000). The results for eFuMo and ANN are part of the technical report (Gradišar et al., 2013). We have to note here that the ANN models were optimized for each model separately. The parameters are tuned using the Levenberg–Marquardt optimization method (Nelles, 2001) and the Optimal Brain Surgery method (Hassibi and Stork, 1993) is used for ANN pruning. In contrast to that, the eFuMo and the proposed cloud-based method use the default algorithm parameters for all models. The parameters chosen for eFuMo by Gradišar et al. (2013) are presented in Table 2. The same parameters are used to identify all three models (*Cost*, *Production*, and *Quality*). For a more appropriate and fair comparison of the proposed fuzzy cloud-based method with eFuMo, the parameters that have the same meaning are chosen the same (i.e., $\gamma_c = \lambda_r = 0.9999$, $\tau = n_{add} = 80$ and $c_{max} = c_{max} = 20$). The evolving parameter of the proposed cloud-based method is defined as $\gamma_{max} = 0.85$.

To evaluate the efficiency of the newly proposed evolving mechanisms in Section 3 four different experiments are provided, as follows: OLD (old evolving mechanism, contains only C_1^{add}, C_3^{add}, and C_4^{add}),

Table 2
Chosen parameters of eFuMo method.

γ_c	γ_v	λ_r	τ	c_{max}	N_{out}	k_n
0.9999	0.9999	0.9999	80	20	50	3

ADD (OLD with the addition of C_2^{add}), REM (OLD with the addition of C_1^{rem} and C_2^{rem}) and NEW (new evolving mechanism includes all the evolving mechanisms from Section 3). The general idea is to see how each of this mechanisms influence the efficiency of the algorithm.

The validation phase consists of two different approaches, off-line and on-line. In the off-line mode the parameters $(\boldsymbol{\mu}^i, \sigma^i, \boldsymbol{P}^i, \boldsymbol{\theta}^i)$ of the algorithm are acquired just from the TRAIN data and during the validation phase they are frozen. In the on-line mode the parameters are updated during the validation phase. Therefore, we want to show that the algorithm is capable of learning and evolving the structure in an on-line manner.

The mean square error values (MSE) of the validation phase are shown in Tables 3–5, for each output *Cost*, *Production* and *Quality*, respectively. The models are acquired from the TRAIN data, while they are validated on VAL1 and VAL2 data. We can see that the on-line identification improves the performance of the cloud-based algorithm in comparison with the off-line mode. Moreover, the cloud-based method performs better than eFuMo for modeling the *Production* and the *Quality*, while in the case of *Cost*, eFuMo provides better results. Furthermore, we can compare the results between the different evolving mechanisms proposed in the paper. In some cases the novelties improve the performance of the algorithm, but unfortunately, in other cases they provide worse results. We can notice that the ANN model outperforms both, eFuMo and the proposed method, but this is due to the fact that the ANN is optimized and tuned for each individual set of data (*Cost*, *Production* and *Quality*). Moreover, ANN has fixed structure and cannot cope with the changes in nonlinear system dynamics. On the other hand, eFuMo and the proposed method use the default values of algorithm parameters and they are capable of evolving the structure according to changing conditions. In this experiment, no explicit changes in system dynamics takes part, but the evolving approach is used here to obtain the structure of the system automatically –the structure does not have to be chosen or optimized a priori as in the case of ANN.

5. Real process: Water-chiller plant

5.1. Problem description

An important part of a factory is the water-chiller plant (WCP). The main purpose of the WCP is to supply cold liquid (usually water or a glycol–water mixture) to the HVAC (heating, ventilation and air conditioning) systems and other technological processes in the factory. The WCP consists of water chillers and cooling towers as well as pumps for distributing the media to the consumers. A single WCP is mainly dedicated to an associated building, but it is also directly connected to the central cooling system (CCS) in a star network. The WCP can transfer the excess energy (cold water) to the CCS or it can receive energy from the CCS when required. In the literature, we can find several approaches to controlling, monitoring and optimizing a WCP (Albieri et al., 2007; Zhang and Zhang, 2010a, b; Mu et al., 2016).

The water-chiller plant in our case consists of four water chillers (WC01÷04) and four cooling towers (CT1÷4), one for each chiller (see Fig. 3). The WC removes heat from the liquid (water) via a vapor-compression cycle (blue lines in Fig. 3) and, as a necessary byproduct, creates waste heat that must be exhausted. Water-cooled chillers are cooled by a separate condenser water loop (the green lines in Fig. 3) that is connected to outdoor cooling towers (CTs). The latter are usually located on the roof of the building to expel heat to the atmosphere. Cooling the WCs is an important part of the process because it improves the thermodynamic and working effectiveness.

Please cite this article in press as: Andonovski G., et al., Evolving model identification for process monitoring and prediction of non-linear systems. Engineering Applications of Artificial Intelligence (2017), https://doi.org/10.1016/j.engappai.2017.10.020.

Table 3
MSE values (11) of validation based on the 4-step-ahead prediction for the first output ($Cost$). TRAIN signal is used for the training phase, while VAL1 and VAL2 are used separately for the validation phase.

TRAIN	VAL1				VAL2			
	OLD	OLD + ADD	OLD + REM	NEW	OLD	OLD + ADD	OLD + REM	NEW
Cloud based (on-line)	4.9654	4.8923	4.9654	4.8923	7.1088	7.0964	7.1088	7.0964
Cloud based (off-line)	5.1641	5.1478	5.1641	5.1478	7.4398	7.4436	7.4398	7.4436
eFuMo (off-line)		6.7610				3.7930		
Neural network		3.5050				5.8610		

Table 4
MSE values (11) of validation based on the 4-step-ahead prediction for the second output ($Production$). TRAIN signal is used for the training phase, while VAL1 and VAL2 are used separately for the validation phase .

TRAIN	VAL1				VAL2			
	OLD	OLD + ADD	OLD + REM	NEW	OLD	OLD + ADD	OLD + REM	NEW
Cloud based (on-line)	0.1761	0.1788	0.1761	0.1788	0.1642	0.1659	0.1642	0.1659
Cloud based (off-line)	0.1964	0.2006	0.1964	0.2006	0.1639	0.1664	0.1639	0.1664
eFuMo (off-line)		0.2654				0.3282		
Neural network		0.1393				0.1285		

Table 5
MSE values (11) of validation based on the 4-step-ahead prediction for the third output ($Quality$). TRAIN signal is used for the training phase, while VAL1 and VAL2 are used separately for the validation phase.

TRAIN	VAL1				VAL2			
	OLD	OLD + ADD	OLD + REM	NEW	OLD	OLD + ADD	OLD + REM	NEW
Cloud based (on-line)	0.1123	0.1387	0.1203	0.1466	0.0687	0.0687	0.0684	0.0682
Cloud based (off-line)	0.2310	0.2302	0.2853	0.2729	0.0683	0.0682	0.0694	0.0688
eFuMo (off-line)		0.0715				0.0881		
Neural network		0.0596				0.0599		

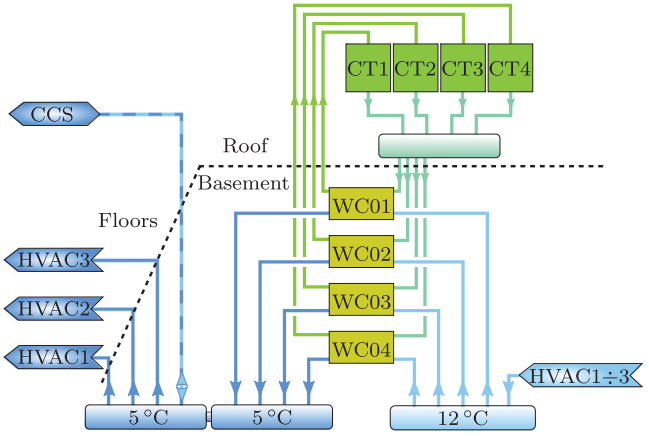

Fig. 3. Scheme of the water chiller plant CCS (Central Cooling System), WC (Water Chiller), CT (Cooling tower), HVAC (Heating, ventilation and air conditioning). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

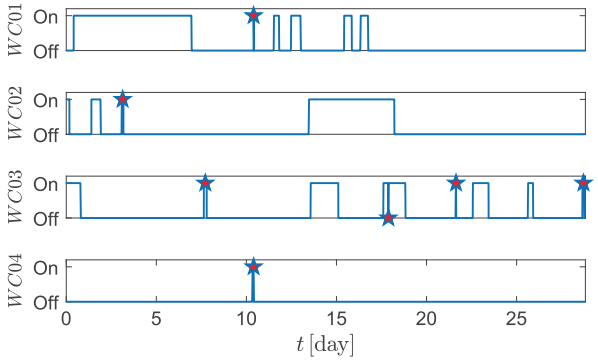

Fig. 4. Turning *on* and *off* periods for each water chiller. The stars denote undesired situations of short term start-ups and shut-downs of the chiller.

For this WCP, on the control level, the mechanism for turning *on* and *off* separate WC is implemented. There are several criteria that are taken into consideration for these procedures, such as the current power consumption (P_{HVAC}) of the facility (HVAC1 ÷ 3), the actual working capacity (P_{WCP}) and the operating time of each WC. Besides this, each WC has a working priority (from 1 to 4) that is circularly shifted once in a month. Thereby, an equal operating time for all the chillers is ensured. Also the number of compressor start-ups cannot be larger than a given limit value, which is another factor that influences the turning *on* and *off* mechanism. When appropriate criteria are satisfied (not all the parameters are measurable), an additional WC is turned *on* or the working WC is turned *off*. In Fig. 4 the *on* and *off* statuses of each chiller for a period of 28 days are presented. For these procedures the transient period takes approximately 15 min (or slightly more). Therefore, we want to avoid short-term start-ups of the chillers. It would be really useful if we could predict such scenarios and prepare our system to be on time for the demands of the consumers.

5.2. Real experiment

To solve the problem described in the previous subsection, we want to acquire models for the two key variables (P_{HVAC} and P_{WCP}) according to the environmental temperature (T_{env}). Here we want to note that P_{HVAC} and P_{WCP} are calculated on the control level and directly stored in a local database. Also, the environmental temperature is stored. The historical values for a duration of two months (see Fig. 5) are used for the identification and validation of the models for both variables. In Fig. 5 the left part of the signals (before the vertical dashed red line) is used for the identification, while the right part (after the vertical dashed red line) is used for the validation. The sampling time of the signals is $T_s = 0.25$ h. The regressor vectors for acquiring the models $\hat{P}_{HVAC}$ and $\hat{P}_{WCP}$ using the fuzzy-cloud-based method are chosen as

Please cite this article in press as: Andonovski G., et al., Evolving model identification for process monitoring and prediction of non-linear systems. Engineering Applications of Artificial Intelligence (2017), https://doi.org/10.1016/j.engappai.2017.10.020.

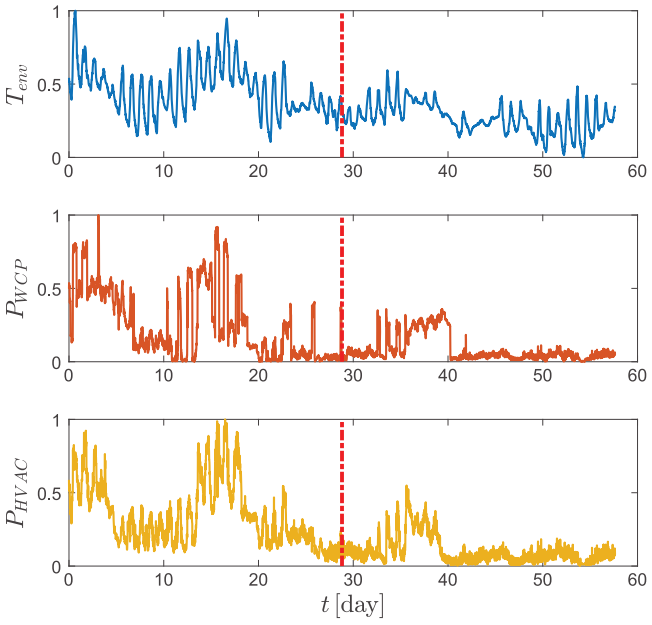

Fig. 5. Real historical values of the input variable output variables T_{env} in the top plot and output variables P_{HVAC} and P_{WCP} (middle and bottom plot). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

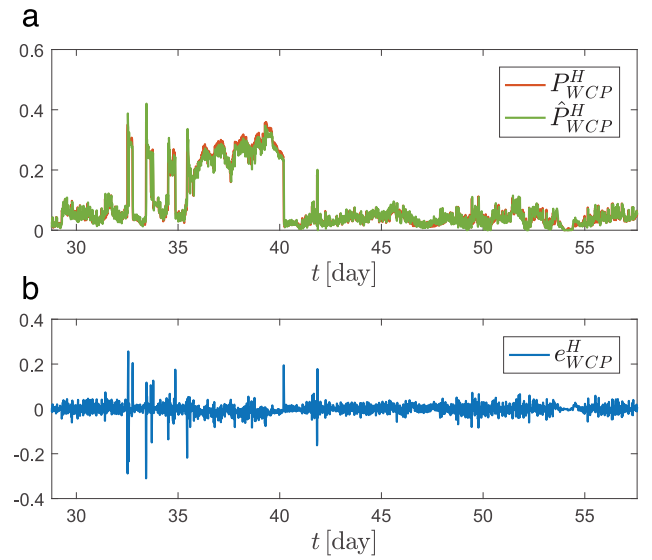

Fig. 6. Power production. (a) Comparison between the real output P^H_{WCP} and the predicted output $\hat{P}^H_{WCP}$ for 2-step ahead prediction. (b) Prediction error.

follows:

$$\hat{P}_{WCP} : \boldsymbol{x}_f(k) = [P_{WCP}(k-1), P_{WCP}(k-2), T_{env}(k-1), T_{env}(k-2), T_{env}(k-3)]^T$$

$$\hat{P}_{HVAC} : \boldsymbol{x}_f(k) = [P_{HVAC}(k-1), P_{HVAC}(k-2), T_{env}(k-1), T_{env}(k-2), T_{env}(k-3)]^T$$

where for both models the same structure is used, $n_a = 2$ and $n_b = 3$. The other design parameters were tuned the same as in the simulation example (see Section 4).

Acquiring relevant models is just a very first step when optimizing the overall efficiency of the WCP. These models serve for the prediction/estimation of the power consumption and the power production, which in turn translates to the on/off commands for the WCs. Using the predicted/estimated values of the output variables, $\hat{P}^H_{WCP}$ and $\hat{P}^H_{HVAC}$,

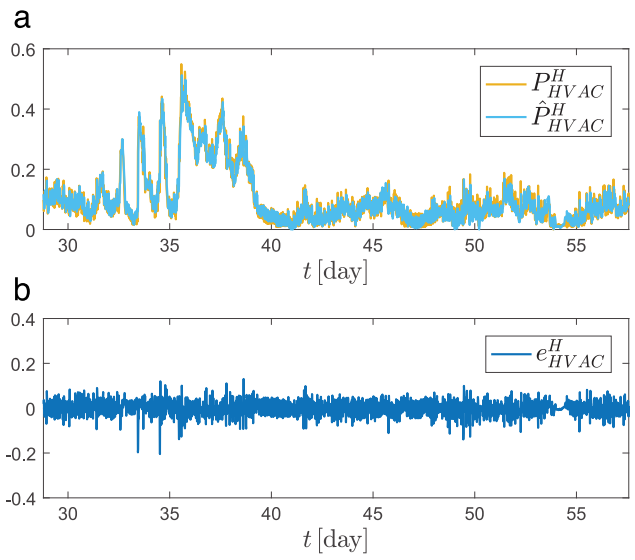

Fig. 7. Power consumption. (a) Comparison between the real output P^H_{HVAC} and the predicted output $\hat{P}^H_{HVAC}$ for 2-step ahead prediction. (b) Prediction error.

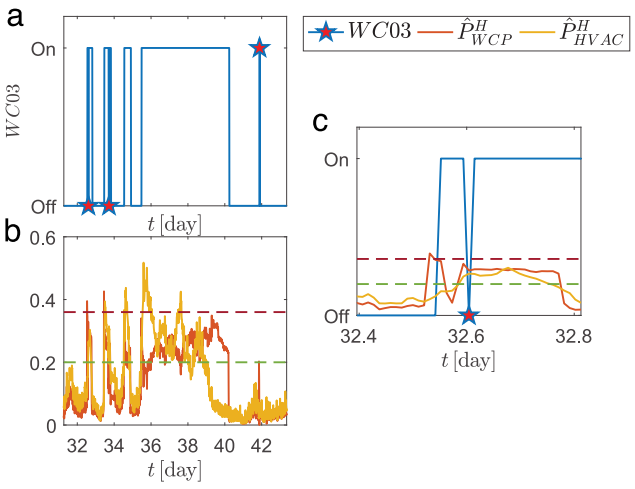

Fig. 8. Example of process monitoring of WCP. (a) Water chiller $WC03$ state (red stars are undesired start-ups and shut-downs). (b) 2-step (30 min) ahead predicted values of power production $\hat{P}^H_{WCP}$ and consumption $\hat{P}^H_{HVAC}$. (c) A detail view of (a) and (b). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

for a period of 30 min (2-step ahead prediction), this can be easily done. From Figs. 6 and 7 we can see that the predicted variables follow the trend of the real ones. The mean square errors (MSE) for $\hat{P}^H_{WCP}$ and $\hat{P}^H_{HVAC}$ are 0.64×10^{-3} and 0.95×10^{-3}, respectively. In Fig. 8 an example is shown how the acquired models could be used for the purposes of process monitoring. With 2-step ahead predicting of the values $\hat{P}^H_{WCP}$ and $\hat{P}^H_{HVAC}$ (Fig. 8b) we can prevent unnecessary turn-off of the third WC (Fig. 8c). At point 32.6 an undesired *turn-off* is detected while the predicted values show that the consumption and the production in the future are expected to be within the limits. When the values are above the upper limit a condition for turn-on an additional WC is fulfilled.

6. Conclusion

In this paper an evolving fuzzy model was applied for the monitoring of non-linear processes. We focused on implementing new evolving mechanisms to improve the performance of the cloud-based identification method. These mechanisms prevent the addition of new clouds

Please cite this article in press as: Andonovski G., et al., Evolving model identification for process monitoring and prediction of non-linear systems. Engineering Applications of Artificial Intelligence (2017), https://doi.org/10.1016/j.engappai.2017.10.020.

(rules) based on outliers and remove the existing ones that are either inactive or not informative enough. The proposed evolving procedure was tested on simulated sets of signals acquired from the Tennessee Eastman process model. Three different models were identified and evaluated using a 4-step-ahead prediction. Moreover, the results were compared with the well-established identification method eFuMo and with artificial neural network model. Beside this, a cloud-based method was also tested on real data acquired from the water-chiller plant (WCP). Two models have been identified, i.e., power production and power consumption, which were further used for the prediction (2-step-ahead) of these key indicators. The main goal was to optimize the overall efficiency of the WCP by reducing the necessary start-ups of each chiller. From the results we can conclude that the proposed evolving fuzzy system is an efficient tool for monitoring non-linear processes.

References

Albieri, M., Beghi, A., Bodo, C., Cecchinato, L., 2007. A simulation environment for the design of advanced chiller control systems. In: Proceedings of the 3rd Annual IEEE Conference on Automation Science and Engineering, Scottsdale, AZ, USA, pp. 962–967.

Angelov, P., Škrjanc, I., Blažič, S., 2013. Robust Evolving Cloud-Based Controller for a Hydraulic Plant. In: 2013 IEEE Conference on Evolving and Adaptive Intelligent Systems, EAIS, pp. 1–8.

Angelov, P., Yager, R., 2011. Simplified fuzzy rule-based systems using non-parametric antecedents and relative data density. In: Symposium Series on Computational Intelligence, IEEE SSCI 2011 – IEEE Workshop on Evolving and Adaptive Intelligent Systems, EAIS 2011 pp. 62–69.

Blažič, S., Angelov, P., Škrjanc, I., 2015. Comparison of Approaches for Identification of All-data Cloud-based Evolving Systems. In: 2nd IFAC Conference on Embedded Systems, Computer Intelligence and Telematics CESCIT 2015, Maribor, Slovenia, pp. 129–134.

Blažič, S., Dovžan, D., Škrjanc, I., 2014. Cloud-based Identification of an Evolving system With Supervisory Mechanisms. In: 2014 IEEE International Symposium on Intelligent Control, ISIC, Antibes, France, pp. 1906–1911.

Campos-Delgado, D.U., Espinoza-Trejo, D.R., 2011. An observer based diagnosis scheme for single and simultaneous open switch faults in induction motor drives. IEEE Trans. Ind. Electron. 58 (2), 671–679.

Chen, A., Zhou, H., An, Y., Sun, W., 2016. PCA and PLS Monitoring Approaches for Fault Detection of Wastewater Treatment Process. In: 2016 IEEE 25th International Symposium on Industrial Electronics, ISIE, pp. 1022–1027.

Chivala, D., Mendonca, L.F., Sousa, J.M.C., Sa da Costa, J.M.G., 2010. Application of evolving fuzzy modeling to fault tolerant control. Evol. Syst. 1 (4), 209–223.

Costa, B.S.J., Angelov, P.P., Guedes, L.A., 2014a. A new unsupervised approach to fault detection and identification. In: Proceedings of the International Joint Conference on Neural Networks, pp. 1557–1564.

Costa, B.S.J., Angelov, P.P., Guedes, L.A., 2014b. Real-time fault detection using recursive density estimation. J. Control Autom. Electr. Syst. 25 (4), 428–437.

Costa, B.S.J., Angelov, P.P., Guedes, L.A., 2015. Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier. Neurocomputing 150 (Part A), 289–303,. http://dx.doi.org/10.1016/j.neucom.2014.05.086.

Costa, B.S.J., Škrjanc, I., Blažič, S., Angelov, P., 2013. A practical implementation of self-evolving cloud-based control of a pilot plant. In: IEEE International Conference on Cybernetics, CYBCONF, pp. 7–12.

Dovžan, D., Logar, V., Škrjanc, I., 2012. Solving the sales prediction problem with fuzzy evolving methods. In: WCCI 2012 IEEE World Congress on Computational Intelligence, Brisbane, Australia, pp. 10–15.

Dovžan, D., Logar, V.,Škrjanc, I., 2015. Implementation of an evolving fuzzy model (eFuMo) in a monitoring system for a waste-water treatment process. IEEE Trans. Fuzzy Syst. 23 (5), 1761–1776.

Downs, J.J., Vogel, E.F., 1993. A plant-wide industrial process control problem. Comput. Chem. Eng. 17 (3), 245–255.

El-Koujok, M., Benammar, M., Meskin, N., Al-Naemi, M., Langari, R., 2014. Multiple sensor fault diagnosis by evolving data-driven approach. Inform. Sci. 259, 346–358.

Gertler, J., Cao, J., 2004. PCA-based fault diagnosis in the presence of control and dynamics. AIChE J. Process Syst. Eng. 50 (2), 388–402.

Gertler, J.J., 1998. Fault Detection and Diagnosis in Engineering Systems. CRC Press, New York, USA.

Glavan, M., Gradišar, D., Atanasijević-Kunc, M., Strmčnik, S., Mušič, G., 2013. Input variable selection for model-based production control and optimisation. Int. J. Adv. Manuf. Technol. 68 (9–12), 2743–2759.

Glavan, M., Gradišar, D., Strmčnik, S., Mušič, G., 2012. Production modelling for holistic production control. Simul. Model. Pract. Theory 30, 1–20.

Gradišar, D., Glavan, M., Hauptman, B., Mušič, G., 2013. Model tehnološke in stroškovne učinkovitosti proizvodnega procesa. Kompetenčni center za sodobne tehnologije vodenja, Ljubljana, pp. 1–85.

Hassibi, ., Stork, D., 1993. Second order derivaties for network prunning: Optimal brain surgeon. In: Advances in Neural Information Processing Systems 5. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 164–171.

He, N., Han, R.P., Wang, S.Q., 2013. A New Multivariate Statistical Process Monitoring Method Using Modified Fast ICA. In: 2013 Ninth International Conference on Natural Computation, ICNC, pp. 106–110.

Huang, S., Tan, K.K., Lee, T.H., 2012. Fault diagnosis and fault-tolerant control in linear drives using the Kalman filter. IEEE Trans. Ind. Electron. 59 (11), 4285–4292.

Isermann, R., 1984. Process fault detection based on modeling and estimation methods-A survey. Automatica 20 (4), 387–404.

Isermann, R., 2004. Model-based fault detection and diagnosis - status and applications. IFAC Autom. Control Aerosp. 29 (c), 71–85.

Isermann, R., 2011. Fault-Diagnosis Applications. Springer-Verlag, Berlin Heidelberg 2011, p. 354.

Lemos, A., Caminhas, W., Gomide, F., 2013. Adaptive fault detection and diagnosis using an evolving fuzzy classifier. Inform. Sci. 220, 64–85.

Li, G., Qin, S.J., Zhou, D., 2010. Geometric properties of partial least squares for process monitoring. Automatica 46 (1), 204–210.

Li, W., Yue, H., Valle-Cervantes, S., Qin, S.J., 2000. Recursive PCA for adaptive process monitoring. J. Process Control 10, 471–486.

Lughofer, E., 2011. Evolving fuzzy systems Methodologies, advanced oncepts and applications, first ed. Springer-Verlag, Berlin Heidelberg, p. 480.

Lughofer, E., 2015. Evolving Fuzzy Systems - Fundamentals, Reliability, Interpretability, Useability, Applications. In: Angelov, P.P. (Ed.), Handbook on Computational Intelligence. Springer Verlag, Berlin Heidelberg, pp. 67–136, chapter 3.

Lughofer, E., Guardiola, C., 2008. Applying Evolving Fuzzy Models with Adaptive Local Error Bars to On-Line Fault Detection. In: 2008 3rd International Workshop on Genetic and Evolving Systems, March, pp. 35–40.

Mu, B., Li, Y., Salsbury, T.I., House, J.M., 2016. Optimization and Sequencing of Chilled-water Plant Based on Extremum Seeking Control *. In: 2016 American Control Conference, ACC, Boston, MA, USA, pp. 2373–2378.

Nelles, O., 2001. Nonlinear System Identification, first ed. Springer-Verlag, Berlin Heidelberg.

Norgaard, M., Ravn, O., Poulsen, N., Hansen, L., 2000. Neural networks for modelling and control of dynamic systems: A Practitioner's Handbook, vol. 1. Springer-Verlag New York, Inc., Secaucus, NJ, USA, p. 246.

Petković, M., Rapaić, M.R., Jeličić, Z.D., Pisano, A., 2012. On-line adaptive clustering for process monitoring and fault detection. Expert Syst. Appl. 39 (11), 10226–10235.

Precup, R.-E., Angelov, P., Costa, B.S.J., Sayed-Mouchaweh, M., 2015. An overview on fault diagnosis and nature-inspired optimal control of industrial process applications. Comput. Ind. 74, 75–94.

Qin, S.J., 2003. Statistical process monitoring: Basics and beyond. J. Chemom. 17 (8–9), 480–502.

Rosa, R., Gomide, F., Dovžan, D., Škrjanc, I., 2014. Evolving neural network with extreme learning for system modeling. In: Evolving and Adaptive Intelligent Systems, EAIS, 2014 IEEE Conference on, pp. 1–7.

Simani, S., Fantuzzi, C., Patton, R.J., 2003. Model-based fault diagnosis in dynamic systems using identification techniques. Springer-Verlag London Ltd..

Škrjanc, I., Blažič, S., Angelov, P., 2014. Robust Evolving Cloud-based PID Control Adjusted by Gradient Learning Method. In: 2014 IEEE Conference on Evolving and Adaptive Intelligent Systems, EAIS, pp. 1–8.

Takagi, T., Sugeno, M., 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man Cybern. SMC-15 (1), 116–132.

Tsai, D.M., Wu, S.C., Chiu, W.Y., 2013. Defect detection in solar modules using ICA basis images. IEEE Trans. Ind. Inf. 9 (1), 122–131.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin, K., 2003a. A review of process fault detection and diagnosis Part III: Process history based methods. Comput. Chem. Eng. 27 (3), 327–346.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.N., 2003b. A review of process fault detection and diagnosis Part I: Quantitative model-based methods. Comput. Chem. Eng. 27 (3), 293–311.

Yin, S., Ding, S.X., Xie, X., Luo, H., 2014. A review on basic data-driven approaches for industrial process monitoring. IEEE Trans. Ind. Electron. 61 (11), 6414–6428.

Zhang, J., Zhang, K.Y., 2010a. Application of tabu search heuristic algorithms for the purpose of energy saving in optimal load distribution strategy for multiple chiller water units. In: IEEE International Conference on Progress in Informatics and Computing, PIC, vol. 1, pp. 303–307.

Zhang, J., Zhang, K.Y., 2010b. Optimal load distribution strategy for multiple chiller water units based on adaptive genetic algorithms. In: Proceedings of 2nd WRI Global Congress on Intelligent Systems, GCIS 2010, vol. 2, pp. 5–8.

Zhang, Y., Qin, S.J., 2008. Improved nonlinear fault detection technique and statistical analysis. AIChE J. Process Syst. Eng. 54 (12), 3207–3220.

Zhang, Y., Zhou, H., Qin, S.J., Chai, T., 2010. Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares. IEEE Trans. Ind. Inf. 6 (1), 3–10.

Please cite this article in press as: Andonovski G., et al., Evolving model identification for process monitoring and prediction of non-linear systems. Engineering Applications of Artificial Intelligence (2017), https://doi.org/10.1016/j.engappai.2017.10.020.