Hybrid System Identification by Incremental Fuzzy C-regression Clustering

Sašo Blažič
Faculty of Electrical Engineering
University of Ljubljana
Ljubljana, Slovenia
saso.blazic@fe.uni-lj.si

Igor Škrjanc
Faculty of Electrical Engineering
University of Ljubljana
Ljubljana, Slovenia
igor.skrjanc@fe.uni-lj.si

***Abstract*—In this paper, an approach to the identification of hybrid systems is discussed. It is based on the incremental fuzzy C-regression clustering. Based on the distance between the current measurement and the hyperplane of the local model, local models are updated. If necessary, a new local model is constructed. To increase the robustness and prevent false local models, the data are kept in the buffer temporarily. The approach produces good results as shown in two examples. The first example can be modelled as a piecewise affine dynamical system and the second one as a switched dynamical system.**

***Index Terms*—Incremental clustering, Fuzzy C-regression clustering, Hybrid systems, Local model, Stream data, Identification.**

I. Introduction

In the last two decades we have witnessed an enormous increase in the volume of the data generated. In the time of IOT, there are sources that generate huge amounts of data. The data come not only from industry, telemetry, traffic, financial trading and e-commerce but also from a variety of web based facilities, social networks, and an immense number of smart devices. Very often we speak about streaming data that has to be processed on the fly. Finding some patterns in the data can find possible applications not only in control, modelling and fault detection/diagnosis but also in non-technical area where data mining, artificial intelligence, security, and safety are key buzzwords.

Such data need to be processed sequentially and incrementally on a record-by-record basis or over sliding time windows to obtain the information about the behavior instantaneously. This has sparked an increased interest in the on-line identification of nonlinear models that combine fuzzy logic and neuro-fuzzy networks, as presented, for example, in [1]–[19]. Essentially, these methods are based on various fuzzy clustering algorithms, but they are modified and extended for processing the data streams [20]–[22]. The extension of the Gustafson-Kessel clustering algorithm for data stream clustering is presented in [23] and [24], in [25] the recursive method based on the Gath-Geva clustering algorithm is given, while an evolving clustering method (ECM) is proposed in [2]. Much attention has been given to an evolving algorithm that adapts the structure of Takagi-Sugeno fuzzy model (eTS) [1]. Similar approaches can be found in the evolving fuzzy control systems [26]–[31]. The identification of a fuzzy model in general requires the partitioning of the input-output space in the first phase and the estimation of local-linear parameters in the second phase. The partitioning can be done either using some a priori knowledge or, more efficiently, by implementing a clustering algorithm. Combining the latter with a changeable model structure results in an evolving paradigm, in which the rule base of the fuzzy model is updated when a new data sample from the stream is available; the estimation of the clusters parameters is calculated by the recursive clustering algorithm [32] in which subtractive clustering is used as presented in [33], [34]. The self-tuning of membership functions in which the parameters are adjusted automatically is presented in an extended Takagi-Sugeno model (exTS [35] and in eTS+ [36]). This enables the detection of clusters of various shapes. The algorithms differ from the algorithm proposed in [1] regarding the calculation of the fuzzy covariance matrix and in the adaptation of cluster centers.

Describing a nonlinear system is also possible by utilising a hybrid system model. In this context, we refer to a hybrid system in the sense of [37] where we deal with a continuous-time or a discrete-time system whose dynamics change abruptly switching between different modes of operation. Some authors also refer to such system as "switched systems" but technically the latter term describes a subclass of hybrid systems where switching or mode change is done arbitrarily (from an external source) while in some hybrid systems these changes are a consequence of the internal state. Hybrid systems enable the description of a nonlinear systems with arbitrary accuracy. Hybrid systems can be viewed as a special case of fuzzy systems where only one (local) model contributes to the output of the system. This detail is very important when discussing the interpretability of fuzzy systems – whenever the same variable is shared in the antecedent part and the consequent part of a fuzzy model, either the interpretability of the local model is problematic or the global model does not have a good fit [38]. This can be avoided using different approaches: excluding problematic variables from the consequent part [39], excluding them from the antecedent vector, by choosing very narrow (unnormalised) membership functions which results in

This work has been supported by Slovenian Research Agency (ARRS) with the Research Program P2-0219.

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

very flat normalised membership functions in a large area of input space, or by using the so called zones of isolation and interpolation zones within membership functions [40]. Hybrid systems do not suffer from these problems as they do not allow blending the local models, and therefore a particular (local) model is used in a certain time (or in a certain region of the state space) – there is no difference between the current local and the current global model. The interpretability can therefore not be an issue.

Identification of hybrid systems is not a new technique, some approaches (under different names) can be traced decades back [41]–[43]. In recent years the identification of hybrid systems has been an active research area [37], [44]–[51].

The approach adopted in this paper is based on the incremental fuzzy C-regression clustering [52]. A very simple identification approach is proposed that is evolving in its nature which means that based on a simple test a new local model can be constructed, the parameters of a local model updated, the measurement can be buffered or discarded. The hallmarks of the approach are simplicity of use, easy tuning of a low number of parameters, and high level of robustness to the change of most of design parameters.

II. Local model network vs. hybrid system

Many approaches that deal with the modelling of nonlinear systems (see e.g. the references given in the introduction) assume that the output of the system y can be modelled using a local model network

$$y = \sum_{j=1}^{m} \mu_j(\boldsymbol{u}_p) y_j(\boldsymbol{u}) \tag{1}$$

where $\boldsymbol{u}_p \in \mathbb{R}^q$, $\boldsymbol{u} \in \mathbb{R}^r$, $\mu_j \in [0,1]$ gives validity of the respective local model output y_j, i.e. $\mu_j(\boldsymbol{u}_p)$ defines the regions in the space of $\boldsymbol{u}_p$ where the local models are valid. The validity function μ_j is constructed so that the partition of unity is fulfilled in the convex set $\mathcal{C}$ that includes the whole region of interest of $\boldsymbol{u}_p$. In the context of Takagi-Sugeno fuzzy models, q-element vector $\boldsymbol{u}_p$ represents antecedent variables, r-element vector $\boldsymbol{u}$ represents consequent variables, $\mu_j(\cdot)$ define membership functions, and m is the number of rules in the rule base. Eq. (1) can be seen as a mapping from $\boldsymbol{u}_p$ and $\boldsymbol{u}$ to y. According to the nature of the system, the variables in $\boldsymbol{u}_p^T = [u_{p1}, \ldots, u_{pq}]$ that define the partitioning of the input-output space are not necessarily the inputs included in the regression vector $\boldsymbol{u}$.

The functions $y_j(\boldsymbol{u})$ can take an almost arbitrary form although linear or affine functions are often used for the sake of simplicity. In our case, affine functions will be utilized:

$$y_j(\boldsymbol{u}) = \theta_{j0} + \theta_{j1}u_1 + \theta_{j2}u_2 + \ldots + \theta_{j,r}u_r \quad j = 1, \ldots, m \tag{2}$$

where the r-element vector $\boldsymbol{u}^T = [u_1, u_2, \ldots, u_r]$ has been introduced. To simplify further derivations, we shall also use the augmented vector $\boldsymbol{u}_e^T = [1, u_1, u_2, \ldots, u_r]$ and the corresponding vector of parameters $\boldsymbol{\theta}_j^T = [\theta_{j0}, \theta_{j1}, \ldots, \theta_{j,r}]$. Combining Eqs. (1) and (2) we obtain

$$y = \sum_{j=1}^{m} \mu_j(\boldsymbol{u}_p)\boldsymbol{u}_e^T\boldsymbol{\theta}_j \tag{3}$$

The idea behind the hybrid systems is essentially the same. They also make use of a large number of local models. The difference between the hybrid systems and the aforementioned local model networks is that in a hybrid system only one local model is active at a particular time. To use the same notation as in (3) the output of the hybrid system can be given as:

$$y = \boldsymbol{u}_e^T\boldsymbol{\theta}_j \quad j = 1, \ldots, m \tag{4}$$

All the models are numbered by integers j, and when j changes we often speak about the sudden (abrupt) change of the mode of operation. The switching can be done in one of the following ways:

- The mode of operation is obtained from a particular partition of some "input" space. This is actually the space that corresponds to the space of $\boldsymbol{u}_p$ which means that the mapping from $\boldsymbol{u}_p$ to j is unique. This means we can still use Eq. (3) but with the limitation of binary validities (0 or 1).
- The mode of operation is governed by some (external) quantity(ies) or signal(s). This quantity (if known) can be included in the vector $\boldsymbol{u}_p$ and again Eq. (3) still holds.
- The mode of operation can change according to time (schedule). Technically, the system then becomes linear (or affine) time-varying and therefore no longer a hybrid system in the classical sense.

Hybrid systems can be seen as a subset of the nonlinear systems described by (3). In this paper we will tackle the problem of hybrid system identification where an approach originally developed for local model networks [52] will be adapted for hybrid systems.

III. Incremental C-regression for identification of hybrid systems

A. Fixed number of local models

Incremental C-regression [52] is an approach based for local model networks identification. A unique property of this technique is that a cluster prototype is not some finite compact set as assumed usually, but a hyperplane in the space of $\boldsymbol{u}_p$. But the method goes a step further and assumes that the local model used in the consequent of the fuzzy rules also play the role of the prototype. This makes the system much less complex, the number of parameters reduces enormously. The approach is based on the distances between the measurements and the hyperplanes of the individual local model prototypes. The difference from the original method in [52] is that we do not adopt here the principle of fuzzy (distributed) local model updates but only update the winning local model. The winner is obtained by finding the minimal distance:

$$d_{jk}^2 = \left(y_k - \boldsymbol{u}_{ek}^T\boldsymbol{\theta}_j\right)^2 \quad j = 1, \ldots, m \tag{5}$$

over all the m local models that exist at a particular time k. If this number m is fixed or known a priori, we denote the winning model by an integer w $(1 \leq w \leq m)$

$$w = \arg \min_{1 \leq j \leq m} d_{jk}^2 \tag{6}$$

The value of w obviously depends on time k but this dependence is omitted in the description.

In the next step, the parameters of the w-th local model have to be updated. A weighted recursive least square algorithm is used, which consists of the following steps [53]:

1) The weighted estimation error e_{wk} at time instant k, the error between the current output y_k and the weighted model output based on old parameter estimate $\boldsymbol{u}_{ek}^T \boldsymbol{\theta}_{w,k-1}$, is calculated as follows

$$e_{wk} = y_k - \boldsymbol{u}_{ek}^T \boldsymbol{\theta}_{j,k-1} \tag{7}$$

2) The innovation gain vector $\boldsymbol{K}_{wk}$ at time instant k is given in the following way

$$\boldsymbol{K}_{wk} = \boldsymbol{P}_{w,k-1} \boldsymbol{u}_{ek} \left(\gamma + \boldsymbol{u}_{ek}^T \boldsymbol{P}_{w,k-1} \boldsymbol{u}_{ek} \right)^{-1} \tag{8}$$

where $\boldsymbol{P}_{w,k-1}$ is the estimate-error covariance matrix at time instant $(k-1)$, and $0 < \gamma \leq 1$ stands for the forgetting factor, which is to be selected by the user (this value is usually between 0.95 and 1).

3) The estimate-error covariance matrix $\boldsymbol{P}_{wk}$ is calculated as

$$\boldsymbol{P}_{wk} = \frac{1}{\gamma} \left(\mathbf{I} - \boldsymbol{K}_{wk} \boldsymbol{u}_{ek}^T \right) \boldsymbol{P}_{w,k-1} \tag{9}$$

4) The current model parameters $\boldsymbol{\theta}_w$ are updated:

$$\boldsymbol{\theta}_{wk} = \boldsymbol{\theta}_{w,k-1} + \boldsymbol{K}_{wk} e_{wk} \tag{10}$$

A general approach given in this paper is not strictly connected to the online parameter-estimation algorithm presented above. Any other recursive-least-squares-based algorithm can be used as well.

If the number of local models is fixed and a wrong guess is taken, several problems can arise: if a too low number is assumed, the existing local models are blended; if a too high number is assumed, some phantom models can be constructed.

B. Adaptive number of local models

To overcome the problems just mentioned, the number of local models have to adapt to the problem treated. For that the incremental c-regression clustering method is used as described next.

The algorithms starts without any local models. Based on the data arriving from the stream, new local models can be added, their parameters updated, local models can also be deleted or merged. These are typical actions performed by evolving systems. In our algorithm, when a new measurement arrives a decision has to be taken:

- If the measurement lies "close" to an existing local model, it is decided that the measurement belongs to this local model. Consequently, the parameters of the corresponding local model are updated following the procedure given in Section III-A.
- If the measurement is "far" from the existing models but not "very far" (meaning that it is not regarded as an outlier), the data are put into the buffer. The contents of the buffer are analyzed after each update of the buffer in order to either use the data to construct a new local model or simply discard all these data if the informational content is not adequate.
- If the measurement lies "very far" from the existing local models, it is decided that it is probably an outlier, and the data are not stored anywhere.

The meaning of "close", "far", etc. will be quantified with respect to the existing local models, i.e. based on the distance d_{jk}^2 given by Eq. (5). When associating a measurement at time k with the winning (w-th) local model, we take into account the distance d_{wk}^2 between the measurement and the local model hyperplane at this particular time. The winning local model is denoted by w as given by (6). The mean value of the past distances $\overline{d_j^2}$ is kept for each local model. It is updated recursively upon the addition of a new measurement to the winning local model. At time k, only $\overline{d_j^2}$ of the winning local model is updated, as proposed in [36]:

$$\overline{d^2}_{wk} = \frac{k_w - 1}{k_w} \overline{d^2}_{w,k-1} + \frac{1}{k_w} d_{wk}^2 \tag{11}$$

where k_w stands for the current number of samples in the w-th local model (k_w is the number of samples in the winning local model), and index of $\overline{d_w^2}$ is the current time. Initialization at time $t = 0$: $\overline{d^2}_{j,0} = 0$ for all local models.

Probably the most crucial part of an incremental system is the decision regarding how and when a new local model is added. When a new measurement arrives, the distances $d_{jk}^2 (j = 1, 2, \ldots, m)$ from existing local models are calculated. If the distance of the winning model is considerably higher than the mean of the previous distances of the data associated with this model, i.e.,

$$d_{wk}^2 > \kappa_{min} \overline{d^2}_{w,k-1} \tag{12}$$

the decision is made that this particular piece of data will not be associated with any of the existing local models. Although being a relative (dimensionless) quantity by its definition, Eq. (12), κ_{min} still has to be tuned in order to achieve better results (good results are usually obtained if κ_{min} is in the range of 1 to 100). By tuning κ_{min} the trade-off between the achieved modelling error and the number of the local models is performed easily.

It is extremely important not to start a new local model immediately after deciding that the current measurement does not belong to any of the existing local models. Thus we prevent false local models (either as a result of outliers or poor information content of the data). Temporary data are kept in the buffer. When it is decided that the data in the buffer can form a new local model, a new model is initialized based on the data in the buffer. Several criteria can be implemented for this, normally it is very advisable to at least require that the data in the buffer span a unique hyperplane. For practical purposes, it is also not recommended to form a new model based on the nonconsecutive patches of data.

When a decision is made to construct a new local model, the number of local models m is increased by 1, and all the parameters defining the new local cluster have to be initialized. The parameter vector of the local affine model that defines the hyperplane can be obtained from the data in the buffer using the well-known least square method for affine systems. The initialization of other cluster parameters ($\boldsymbol{P}_m, k_m, \overline{d_m^2}$) is also straight-forward.

It has to be noted that the samples that fulfill condition given by Eq. (12) can lie very far from all the existing clusters. These samples may be outliers and, therefore, the associated data are not used or stored. The condition for classifying a measurement as an outlier follows the same idea as (12). If the following is satisfied:

$$d_{wk}^2 > \kappa_{max}\overline{d^2}_{w,k-1} \tag{13}$$

the current sample is not taken into account for adapting the current local model or adding a new local model, nor it is stored in the buffer ($\kappa_{max} > \kappa_{min}$ is a design parameter with the recommended value of $2\kappa_{min}$).

The local models can easily be merged or deleted if necessary although the details will not be given here. Since a local model is represented with the hyperplane, it is easy to merge two models if the parameters are close enough taking into account a certain metric. Also, a local model can be deleted if either it was based on a very low number of measurements or these measurements are old enough to assume they are outdated.

IV. Examples

The proposed algorithm was tested on some simulated examples to demonstrate the main features. The first example is a simple simulated system with a tank whose cross-section is constant until a certain level, above this level the tank has a higher but again constant cross-section. This example is a simple representative of a piecewise affine system. The second system is a control system. The idea here is to identify the (already) controlled system. The controller with multiple control modes is a typical example of a switched system.

As already stressed, the proposed approach has a very attractive property that the number of design parameters is low while the algorithm is quite insensitive to most of these parameters. The system is also very robust with regards to the forgetting factor, which was kept at $\gamma = 0.98$ during all tests. The only parameter that was changing was κ_{min}, while κ_{max} was always set to $2\kappa_{min}$.

A. Tank with piecewise-constant cross-section

The example treats the case of a tank with changing cross-section as depicted in Fig. 1. The area of the cross-section depends on the water level h:

$$A(h) = \begin{cases} A_1, & \text{if } h < h_1 \\ A_2, & \text{otherwise.} \end{cases} \tag{14}$$

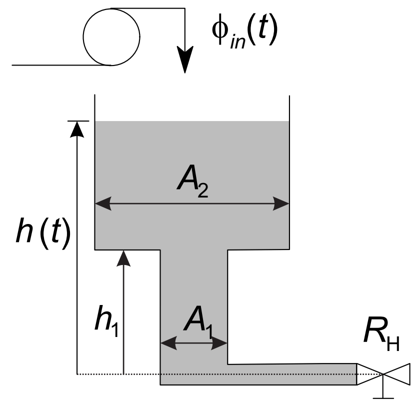

Fig. 1. The schematic representation of the tank used in Example in Section IV-A.

The outlet valve is assumed linear with the hydraulic resistance R_H. The model of the system is very simple and given by an ordinary differential equation:

$$A(h)\dot{h} = \phi_{in} - \frac{1}{R_H}h \tag{15}$$

It is straight-forward to show that this system is an example of a continuous-time hybrid system with two modes of operation and the mode changes when the level in the tank $h(t)$ crosses h_1. In our case discrete-time or sampled version of the system will be identified. Note that the system was simulated as a continuous-time one. Then the level was sampled with the sampling time of 10 s. The measurement noise was added to these samples (zero mean with the standard deviation of 0.001). This level of noise seems very low but some it causes measurements to lie quite far from the ideal hyperplane. Other system parameters (given without the units for brevity): $A_1 = 0.1$, $A_2 = 0.4$, $R_H = 400$, $h_1 = 0.5$.

The following form of the discrete-time local models will be used:

$$h_{k+1} = \theta_1\phi_{1k} + \theta_2 h_k \tag{16}$$

Although the model (16) is extremely simple, there are some aspects of this model that require a special attention. The underlying physical model is continuous-time, while the model (16) is discrete-time. This transformation is not defined completely unless a certain regime is assumed between sampling moments. This is a minor problem, however. A bigger problem is the fact that the change between the two operating modes of the original continuous system can occur in any moment between taking two samples. This means that none of the two discrete local models can describe the system if the change of mode occurs between samples k and $k + 1$. This happens even in the perfect disturbance-free case.

An empty tank was excited by the inflow changing sinusoidally between $0.5 \cdot 10^{-3}$ and $2.5 \cdot 10^{-3}$ with the period of 500 s for 2000 s. Fig. 2 shows the actual level in the tank (in black), and noisy measurements taken (blue crosses in the upper part of the tank, red pluses in the lower part). The proposed algorithm was run on these 201 measurements. The only parameter that was changing in all the examples presented

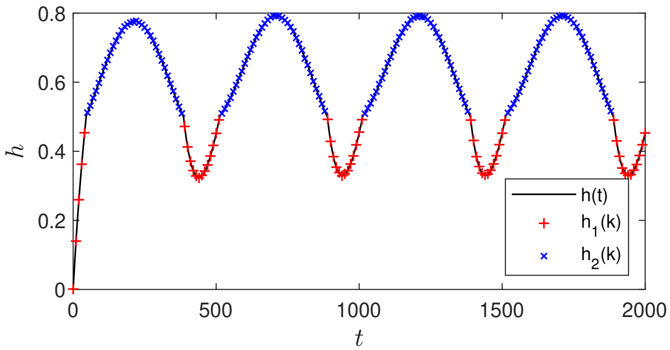

Fig. 2. The measurements used in Example in Section IV-A.

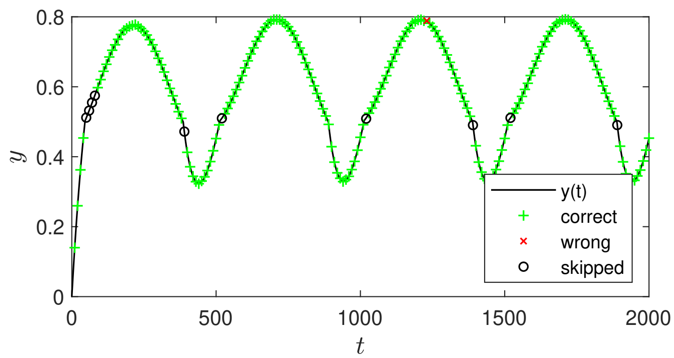

Fig. 3. Classification results for the Example in Section IV-A: green pluses depict measurements correctly assigned to a local model, red crosses show wrong classification, and black circles unassigned measurements that were skipped.

in this paper is κ_{min} which was set to 10. Fig. 3 shows the results of the classification (green pluses depict measurements correctly assigned to a local model, red crosses show wrong classification, and black circles unassigned measurements that were skipped).

Fig. 4 show the course of parameters as they are adapted with time. Red line shows the parameters of the first local model (lower part of the rank), and the blue one the parameters of the second local model. Dotted lines show the "true" parameters but we have to stress here that even the "true" parameters are not constant (they depend on the change of the input sinusoid between the two samples) but these oscillations of the true parameters can be neglected.

In conclusion we can observe that good identification results are obtained in this case although the system is not as benign as it seems from the first look.

B. Control system

The case treated in this example is a control system consisting of an on-off controller with a hysteresis. The model of the plant is:

$$G(s) = \frac{1}{(s+1)^2} \tag{17}$$

The reference signal is a unit step at $t = 0$. The controller outputs 5 in the on state, and 0 in the off state. The control algorithm uses the hysteresis that prevents switches inside

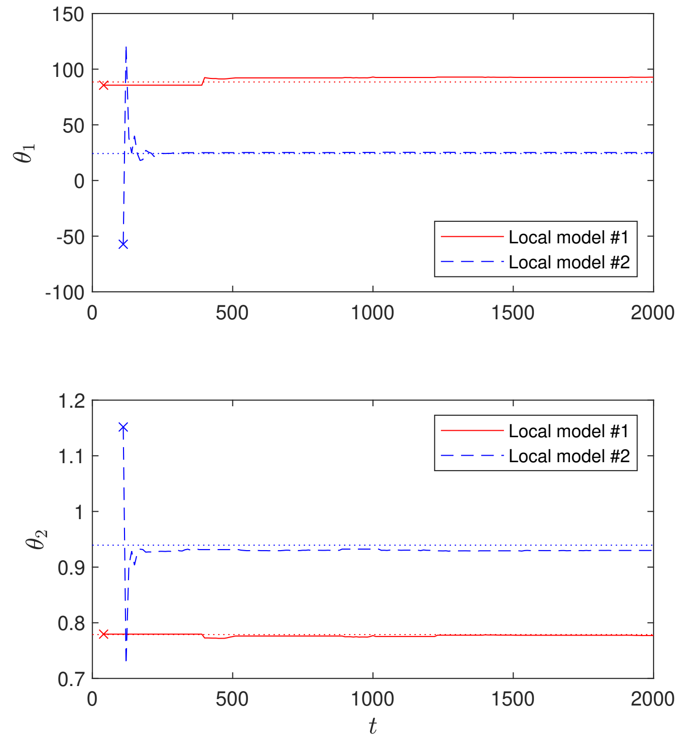

Fig. 4. Adaptation of parameters of local models for the Example in Section IV-A. Crosses show the points at which the local models are first initialized.

the ± 0.1 region around the reference. The controller is again implemented in the continuous-time fashion. This time some zero-mean normally distributed signal acts as a disturbance rather than being just a measurement noise. The sampling frequency is 20 Hz, and the experiment lasts for 10 s (again 201 samples are taken). Both initial conditions of the plant are 0.

The controlled system is treated as an autonomous system meaning that the closed-loop model is identified as a time series. The following local model is assumed:

$$y_{k+2} = \theta_1 + \theta_2 y_{k+1} + \theta_3 y_k \tag{18}$$

This local model has similar problems than the one used in Section IV-A: switches occur asynchronously from the sampling which causes problems (in the vicinity of switching). We expect to identify two modes: the mode when the on-off controller is in the on state (denoted as the on mode), and the mode when the on-off controller is in the off state (the off mode).

The design parameter κ_{min} was set to 4 in this case. Fig. 5 shows the output of the system (when there is no disturbance) with a green line, and the output when disturbance is active (black line). The large difference between the green and the black line is due to the disturbance. Grey area shows the hysteresis. Measurements are shown with blue crosses in the off mode, and red pluses in the on mode. Fig. 6 shows the results of the classification (green pluses depict measurements correctly assigned to a local model, red crosses show wrong

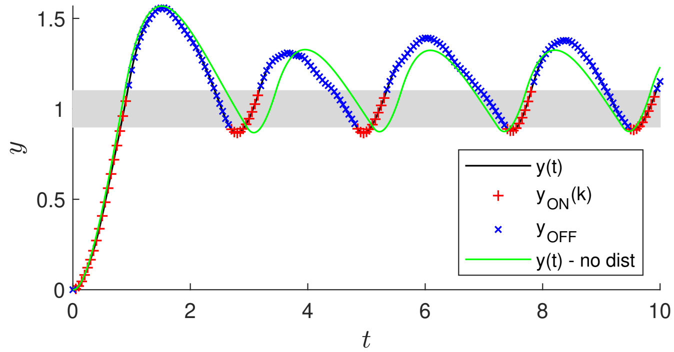

Fig. 5. The measurements used in Example in Section IV-B.

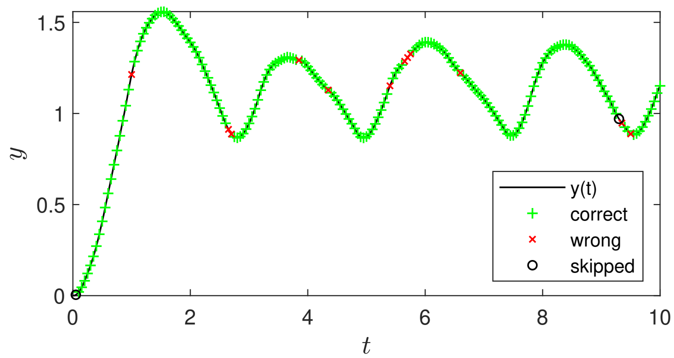

Fig. 6. Classification results for the Example in Section IV-B: green pluses depict measurements correctly assigned to a local model, red crosses show wrong classification, and black circles unassigned measurements that were skipped.

classification, and black circles unassigned measurements that were skipped).

Fig. 7 show the course of parameters as they are adapted with time. Red line shows the parameters of the first local model (on mode), and the blue one the parameters of the second local model (off mode). The second and the third parameter converge to the same true value while the first one defines the obvious difference between the two modes. The difference seems low which is the consequence of the fact that plant poles lie very close to 1.

Again, good results can be observed.

V. Conclusion

The proposed approach of using incremental fuzzy C-regression clustering in the on-line identification of hybrid systems shows a great potential. It is easy to implement, it only has a few design parameters, of which all except one were kept on the default values. Only one parameter (κ_{min}) is used for tuning. The approach was successfully tested on two simple simulated examples.

References

[1] P. P. Angelov and D. P. Filev, "An approach to online identification of takagi-sugeno fuzzy models," *IEEE Trans. Syst. Man Cyber. part B*, vol. 34, no. 1, pp. 484–497, February 2004.

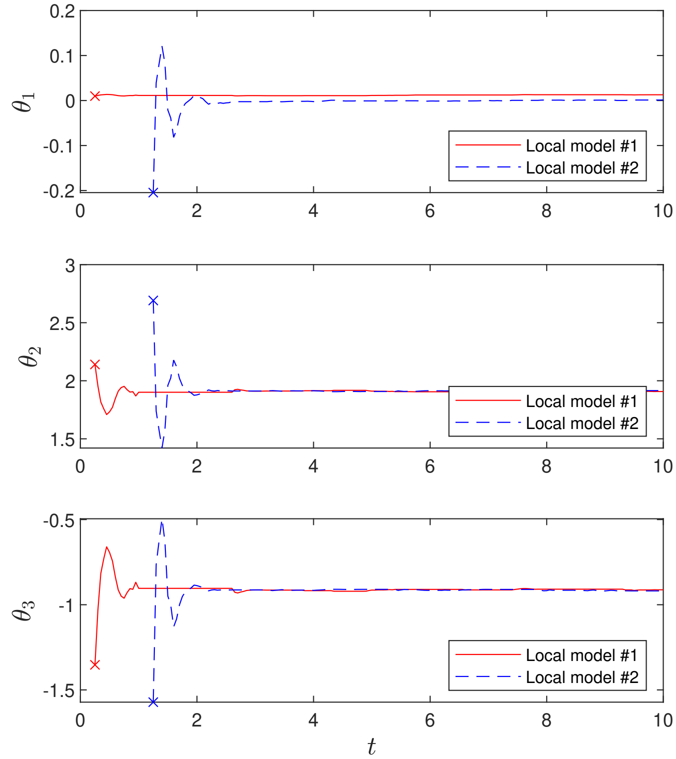

Fig. 7. Adaptation of parameters of local models for the Example in Section IV-B. Crosses show the points at which the local models are first initialized.

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

[3] N. Kasabov, "Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning," *IEEE Trans. Syst. Man Cyber. part B*, vol. 31, no. 6, pp. 902–918, December 2001.

[4] ——, "Evolving fuzzy neural networks – algorithms, applications and biological motivation," in *Proc. 4th Int. Conf. Soft Comput.*, 1998, pp. 271–274.

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

[6] F. J. Lin, C. H. Lin, and P. H. Shen, "Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive," *IEEE Trans. on Fuzzy Syst*, vol. 9, no. 5, pp. 751–759, October 2001.

[7] S. Wu and M. J. Er, "Dynamic fuzzy neural networks – a novel approach to function approximation," *IEEE Trans. Syst. Man Cyber. part B*, vol. 30, no. 2, pp. 358–364, April 2000.

[8] D. F. Leite, G. Andonovski, I. Škrjanc, and F. Gomide, "Optimal rule-based granular systems from data streams," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 3, pp. 583–596, 2020.

[9] D. Dovžan and I. Škrjanc, "Fuzzy space partitioning based on hyperplanes defined by eigenvectors for takagi-sugeno fuzzy model identification," *IEEE Trans. Industrial Electronics*, vol. 67, no. 6, pp. 5144–5153, 2020.

[10] I. Škrjanc, S. Blažič, E. Lughofer, and D. Dovžan, "Inner matrix norms in evolving cauchy possibilistic clustering for classification and regression from data streams," *Inf. Sci.*, vol. 478, pp. 540–563, 2019.

[11] S. Wu, M. J. Er, and Y. Gao, "A fast approach for automatic generation of fuzzy rules by generalized dynamicc fuzzy neural networks," *IEEE Trans. on Fuzzy Syst*, vol. 9, no. 4, pp. 578–594, August 2001.

[12] C. T. Lin, "A neural fuzzy control system with structure and parameter learning," *Fuzzy Sets and Systems*, vol. 70, pp. 183–212, 1995.

[13] C. Garcia, A. Esmin, D. F. Leite, and I. Škrjanc, "Evolvable fuzzy systems from data streams with missing values: With application to

temporal pattern recognition and cryptocurrency prediction," *Pattern Recognit. Lett.*, vol. 128, pp. 278–282, 2019.

[14] S. G. Tzafestas and K. C. Zikidis, "Neurofast: On-line neuro-fuzzy art-based structure and parameter learning tsk model," *IEEE Trans. Syst. Man Cyber. part B*, vol. 31, no. 5, pp. 797–802, October 2001.

[15] G. Andonovski, G. Mušič, S. Blažič, and I. Škrjanc, "Evolving model identification for process monitoring and prediction of non-linear systems," *Eng. Appl. Artif. Intell.*, vol. 68, pp. 214–221, 2018.

[16] G. Černe, D. Dovžan, and I. Škrjanc, "Short-term load forecasting by separating daily profiles and using a single fuzzy model across the entire domain," *IEEE Trans. Industrial Electronics*, vol. 65, no. 9, pp. 7406–7415, 2018.

[17] M. F. Azeem, M. Hanmandlu, and N. Ahmad, "Structure identification of generalized adaptive neuro-fuzzy inference systems," *IEEE Trans. on Fuzzy Syst*, vol. 11, no. 5, pp. 666–681, October 2003.

[18] I. Škrjanc, J. A. Iglesias, A. Sanchis, D. Leite, E. Lughofer, and F. Gomide, "Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey," *Information Sciences*, vol. 490, pp. 344 – 368, 2019.

[19] S. Blažič, D. Dovžan, and I. Škrjanc, "Cloud-based identification of an evolving system with supervisory mechanisms," in *2014 IEEE International Symposium on Intelligent Control, ISIC 2014, Juan Les Pins, France, October 8-10, 2014*. IEEE, 2014, pp. 1906–1911.

[20] D. Dovžan and I. Škrjanc, "Possible use of evolving c-regression clustering for energy consumption profiles classification," in *2015 IEEE International Conference on Evolving and Adaptive Intelligent Systems, EAIS 2015, Douai, France, December 1-3, 2015*. IEEE, 2015, pp. 1–6.

[21] I. Škrjanc, D. Dovžan, and F. A. C. Gomide, "Evolving fuzzy-madel-based on c-regression clustering," in *2014 IEEE Conference on Evolving and Adaptive Intelligent Systems, EAIS 2014, Linz, Austria, June 2-4, 2014*. IEEE, 2014, pp. 1–7.

[22] E. Lughofer, M. Pratama, and I. Škrjanc, "Incremental rule splitting in generalized evolving fuzzy regression models," in *2017 Evolving and Adaptive Intelligent Systems, EAIS 2017, Ljubljana, Slovenia, May 31 - June 2, 2017*, I. Škrjanc and S. Blažič, Eds. IEEE, 2017, pp. 1–8.

[23] D. Filev and O. Georgieva, *An Extended Version of the Gustafson-Kessel Algorithm for Evolving Data Stream Clustering*. John Wiley & Sons, Ltd, 2010, ch. 12, pp. 273–299.

[24] D. Dovžan and I. Škrjanc, "Recursive clustering based on a gustafson-kessel algorithm," *Evolving Systems*, vol. 2, pp. 15–24, 2011.

[25] H. Soleimani-B., C. Lucas, and B. N. Araabi, "Recursive gath–geva clustering as a basis for evolving neuro-fuzzy modeling," *Evolving Systems*, vol. 1, no. 1, pp. 59–71, Aug 2010.

[26] P. P. Angelov, I. Škrjanc, and S. Blažič, "Robust evolving cloud-based controller for a hydraulic plant," in *Proceedings of the 2013 IEEE Conference on Evolving and Adaptive Intelligent Systems, EAIS 2013, IEEE Symposium Series on Computational Intelligence (SSCI), 16-19 April 2013, Singapore*. IEEE, 2013, pp. 1–8.

[27] G. Andonovski, P. P. Angelov, S. Blažič, and I. Škrjanc, "Robust evolving cloud-based controller (recco)," in *2017 Evolving and Adaptive Intelligent Systems, EAIS 2017, Ljubljana, Slovenia, May 31 - June 2, 2017*, I. Škrjanc and S. Blažič, Eds. IEEE, 2017, pp. 1–6.

[28] G. Andonovski, B. S. J. Costa, S. Blažič, and I. Škrjanc, "Robust evolving controller for simulated surge tank and for real two-tank plant," *Automatisierungstechnik*, vol. 66, no. 9, pp. 725–734, 2018.

[29] R. Precup, M. Tomescu, S. Preitl, E. M. Petriu, J. C. Fodor, and C. Pozna, "Stability analysis and design of a class of MIMO fuzzy control systems," *J. Intell. Fuzzy Syst.*, vol. 25, no. 1, pp. 145–155, 2013.

[30] K. Mls, R. Cimler, J. Vascák, and M. Puheim, "Interactive evolutionary optimization of fuzzy cognitive maps," *Neurocomputing*, vol. 232, pp. 58–68, 2017.

[31] R. Precup and M. Tomescu, "Stable fuzzy logic control of a general class of chaotic systems," *Neural Computing and Applications*, vol. 26, no. 3, pp. 541–550, 2015.

[32] P. Angelov, "An approach for fuzzy rule-base adaptation using on-line clustering," *Integration of Methods and Hybrid Systems*, vol. 35, no. 3, pp. 275–289, 2004.

[33] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intellegent and Fuzzy Systems*, vol. 2, pp. 267–278, 1994.

[34] R. Yager and D. Filev, "Generation of fuzzy rules by mountain clustering," *J. of Intelligent and Fuzzy Systems*, vol. 2, pp. 209–219, 1994.

[35] P. Angelov and X. Zhou, "Evolving fuzzy systems from data streams in real-time," in *2006 International Symposium on Evolving Fuzzy Systems*, Lake District, UK, IEEE Press, pp.29-35, 2006, pp. 7–9.

[36] P. Angelov, *Evolving Takagi-Sugeno Fuzzy Systems from Streaming Data (eTS+)*. John Wiley & Sons, Ltd, 2010, ch. 2, pp. 21–50.

[37] F. Lauer and G. Bloch, *Hybrid System Identification*. Springer, 2019.

[38] J. Yen, L. Wang, and C. Gillespie, "Improving the interpretability of tsk fuzzy models by combining global learning and local learning," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 4, pp. 530–537, 1998.

[39] J. Abonyi, L. Nagy, and F. Szeifert, "Fuzzy model-based predictive control by instantaneous linearization," *Fuzzy Sets and Systems*, vol. 120, no. 1, pp. 109–122, 2001.

[40] A. Riid and E. Rüstern, "Identification of numerically accurate first-order takagi-sugeno systems with interpretable local models from data," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 9, no. 5, pp. 526–533, 2005.

[41] R. E. Quandt, "The estimation of the parameters of a linear regression system obeying two separate regimes," *J. Am. Stat. Assoc.*, vol. 53, no. 284, pp. 873–880, 1958.

[42] W. S. DeSarbo and W. L. Cron, "A maximum likelihood methodology for clusterwise linear regression," *J. Classif.*, vol. 5, no. 2, pp. 249–282, 1988.

[43] J. K. Tugnait, "Detection and estimation for abruptly changing systems," *Automatica*, vol. 18, no. 5, pp. 607–615, 1982.

[44] O. Nelles, *Nonlinear System Identification, From Classical Approaches to Neural Networks and Fuzzy Models*. Berlin: Springer, 2001.

[45] L. Bako, "Identification of switched linear systems via sparse optimization," *Automatica*, vol. 47, no. 4, pp. 668 – 677, 2011.

[46] P. Vo Tan, G. Millérioux, and J. Daafouz, "A contribution to the identification of switched dynamical systems over finite fields," in *49th IEEE Conference on Decision and Control (CDC)*, Dec 2010, pp. 4429–4434.

[47] G. Pillonetto, "A new kernel-based approach to hybrid system identification," *Automatica*, vol. 70, pp. 21 – 31, 2016.

[48] F. Bianchi, M. Prandini, and L. Piroddi, "A randomized two-stage iterative method for switched nonlinear systems identification," *Nonlinear Analysis: Hybrid Systems*, vol. 35, p. 100818, 2020.

[49] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, "A bounded-error approach to piecewise affine system identification," *IEEE Trans Autom. Control*, vol. 50, no. 10, pp. 1567–1580, 2005.

[50] N. Ozay, M. Sznaier, C. M. Lagoa, and O. I. Camps, "A sparsification approach to set membership identification of switched affine systems," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 634–648, 2012.

[51] H. Ohlsson and L. Ljung, "Identification of switched linear regression models using sum-of-norms regularization," *Automatica*, vol. 49, no. 4, pp. 1045–1050, 2013.

[52] S. Blažič and I. Škrjanc, "Incremental fuzzy c-regression clustering from streaming data for local-model-network identification," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 4, pp. 758–767, 2020.

[53] S. S. Haykin, *Adaptive filter theory, 4th ed.* Prentice Hall, Upper Saddle River, 2002.