DOI: 10.58895/ksp/1000174544-5 erschienen in:
Proceedings - 34. Workshop Computational Intelligence: Berlin, 21.-22. November 2024
DOI: 10.58895/ksp/1000174544 | https://www.ksp.kit.edu/site/books/e/10.58895/ksp/1000174544/

Evolving Fuzzy Model Predictive Control based on Optimization for Nonlinear Plate Heat Exchanger

Miha Ožbot, Igor Škrjanc

University of Ljubljana, Faculty of Electrical Engineering
Tržaška 25, 1000 Ljubljana, Slovenia
Email: {miha.ozbot, igor.skrjanc}@fe.uni-lj.si

Summary

In this paper, a novel evolving Fuzzy Model Predictive Control based on Optimization (eFMPC) is proposed. The controller adresses model adaptation in case of abrupt concept shifts in the system parameters. The model of the system is evolved during control based on online learning approaches with a self-monitoring approach to determine the quality of the local models. The control action is a result of optimization based on the Particle Swarm Optimization method. The control principle was tested on the real Plate Heat Exchanger pilot plant.

1 Introduction

Evolving Intelligent Controllers (EIC) can adjust their structure and parameters online in a recursive manner based on the latest available data, making them well-suited for real-time applications with changing system characteristics. This allows the system to condense data into nonlinear dynamic structures while maintaining transparency and interpretability [35], which is of great importance in an industrial setting where results must be reproducible and verifiable. Evolving systems are defined by their ability to *add new* rules when new data becomes available that is not adequately represented by the existing structure, especially in the case of concept drift [41]. Additionally, such systems must be able to *remove* redundant or faulty rules, *merge* clusters to simplify the structure when samples coalesce [17, 55], and perform *splitting* of rules when new concepts appear that conflict with the existing rulebase [40, 41]. Some of the most influential evolving fuzzy and fuzzy systems of the last decade include: eTS+ [7], FLEXFIS+ [37], AnYa [8], FBeM [34], PANFIS [47], eFuMo [19], GS-EFS [38], eGauss+ [51], and others. These have been applied to a variety of distinct problems, showcasing the adaptability of this framework, e.g., data clustering [6, 29, 51], classification [20, 27, 28], nonlinear system identification [13, 22, 39, 48], system control [1, 10], fault detection and diagnostics [3, 5, 12, 19, 24], design of experiments [45, 50], localization and mapping [33], data streaming [16,23,26,41,53], cybersecurity [30,54,56], image segmentation [43, 44], federated learning [46], time-series forecast [18, 21], among others.

Adaptive and predictive control are well established methodologies in the field of control systems engineering that address the limitations of traditional controllers by adapting to changes in the system or predicting future system behavior. These methods rely on an initial model that adapts its parameters over time. An adaptive Predictive Functional Controller (PFC) for hybrid continuous and discrete signals, based on the recursive least squares parameter identification method with exponential forgetting, is presented in [32]. This approach was examined on an exothermic batch reactor, a time-invariant multivariable process. The exponential forgetting factor can lead to an estimator windup phenomenon when persistent excitation is not available, serving as a safety mechanism. Here, the recursive least square gain denominator is used as a measure of proper excitation. A predictive controller with recursive parameter identification for a time-variant linear system is presented in [14]. This method is demonstrated on an unstable third-order system and decomposes the system into an affine linear part and a bias term, which can be quickly adapted when a parameter shift occurs. This approach requires an iterative method to estimate the optimal bias compensation at every time step. An adaptive fuzzy model predictive control (AFMPC) using ant-colony optimization (ACO) is presented in [11]. Fuzzy Takagi-Sugeno type models describe the nonlinear dynamical and static properties of the controlled system. In this approach, recursive least squares parameter identification is halted when the error between the actual and estimated output becomes smaller than a threshold. A model reference adaptive control (MRAC) and a controller output error method (COEM) were used in [58] to control a heating and cooling system with a fuzzy inverse plant model. Interval models are used in [31] to model uncertainties for tuning the parameters of a PID controller with particle swarm optimization (PSO). Closed-loop stability is ensured if the interval-based constraints are not violated. The method was examined on a three-tank hydraulic system and a batch reactor. An unknown nonlinear system was controlled with a data-driven Model-based Predictive Control (MPC) in [9], where future trajectories were computed based on a persistent excitation lemma [57].

However, model-based (indirect or direct) controllers are more challenging to implement because they require an initial model. Evolving systems can identify and adapt a model over time by changing the model's structure in addition to adapting its parameters. Several control algorithms based on evolving systems have been proposed in the literature, with a popular choice for the base model being the evolving Takagi-Sugeno type, as used in the adaptive controller with leakage in the control law [10], the two degrees-of-freedom control with feedforward and feedback components [59], self-tuning predictive control [60], and evolving PID control [15]. Most similar to our approach is the Robust Evolving Adaptive Controller (ReCCo) proposed in [1, 2], which is based on the AnYa [8] evolving system with a low number of a priori parameters and several safety mechanisms but lacks a splitting and merging mechanism for the evolving law. This controller starts from scratch without any initial fuzzy rules. However, the aforementioned approaches lack cluster merging and removal mechanisms, which are crucial for handling drifting data, as outdated rules can become inaccurate and detrimental to control performance. In addition, our method uses self-monitoring to detect anomalies in the system and remove erroneous clusters.

In this study, we examine an evolving fuzzy predictive functional control with fault detection of a nonlinear dynamical system in a changing environment. The proposed EIC methodology is illustrated in Fig. 1. The main contributions of the proposed eFMPC are:

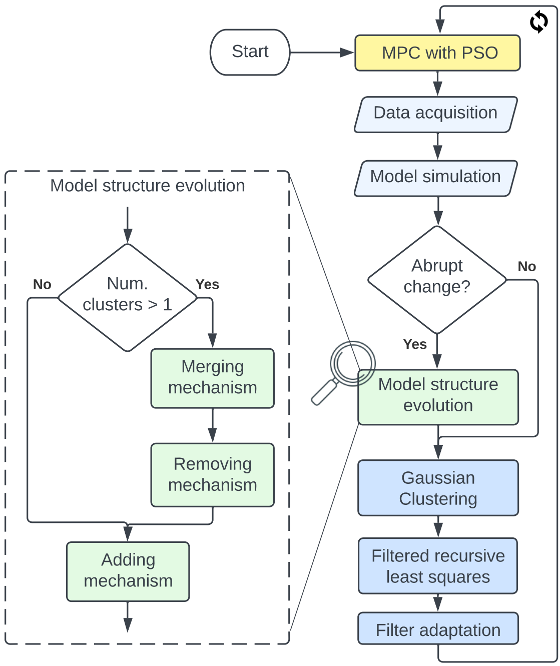

Figure 1: Software flow diagram of the proposed evolving intelligent control.

- An evolving fuzzy model identification method that can identify a model online during predictive functional control of a nonlinear dynamical system. The algorithm uses evolving mechanisms like rule addition, merging, and removal.
- Detection and compensation of abrupt system shifts with a self-monitoring approach, including recursive computation of the Fuzzy Mean Square Error (fMSE).

In this study, we assume that the controlled system is stable, the model operates in discrete-time and is nonlinear, and the disturbance is additive. The reference signal is limited to step changes, and the system experiences both slow drifting and abrupt shifts in operating conditions. The time horizon is small enough that the system can be approximated with a linear model. The evolving fuzzy system used in this study is based on [4, 45] with two notable additions: the Bhattacharyya overlapping measure and the self-monitoring rule removal mechanism. This method was chosen because it is specifically designed for online identification with single-step signal excitation.

2 Evolving fuzzy model identification

Evolving Neuro-Fuzzy Inference Systems (ENFIS) incorporate *fuzzy logic* within a network structure [55]. They consist of sub-models in the form of Takagi-Sugeno fuzzy rules $\mathcal{R}_i$, where $i = 1, 2, \cdots, c$. Each rule is characterized by a membership function defined by a multivariate Gaussian cluster in the antecedent and a local linear model (LLM) in the consequence. The multivariate Gaussian clusters are chosen to represent the input variables in the antecedent structure due to their demonstrated universal approximation capability and their ability to describe correlations between variables [42, 51]. The membership function $\mu_i(k) \in (\varepsilon, 1]$ for a clustering sample $\underline{z}(k) \in \mathbb{R}^{n_z}$ at time step $k = 0, 1, 2, \ldots$ is computed as

$$\mu_i(k) = \exp\Big(-\left(\underline{z}(k) - \underline{\nu}_i\right)^\top \underline{\Sigma}_i^{-1} \left(\underline{z}(k) - \underline{\nu}_i\right)\Big) + \varepsilon, \tag{1}$$

where $\underline{\nu}_i \in \mathbb{R}^{n_z}$ and $\underline{\Sigma}_i \in \mathbb{R}^{n_z \times n_z}$ represent the center and covariance matrix of the cluster, respectively, and $\varepsilon \in \mathbb{R}$ is a very small constant ensuring sufficient support throughout the input space.

The output of the evolving fuzzy model is calculated as a weighted sum of the outputs $\hat{y}_i(k) \in \mathbb{R}$ of all fuzzy rules $\mathcal{R}_i$ based on the normalized activation of each membership function $\Psi_i(k) \in (0, 1]$:

$$\hat{y}(k) = \sum_{i=1}^{c} \Psi_i(k) \hat{y}_i(k) = \frac{\sum_{i=1}^{c} \mu_i(k) \underline{\varphi}^\top(k) \underline{\theta}_i}{\sum_{i=1}^{c} \mu_i(k)}, \tag{2}$$

where $\hat{y}_i(k) = \underline{\varphi}^\top(k)\underline{\theta}_i$ is the consequence LLM of the rule $\mathcal{R}_i$, and $\underline{\varphi}(k)^\top = [u(k-1), \cdots, u(k-m), y(k-1), \cdots, y(k-n), 1] \in \mathbb{R}^{n+m+1}$ serves as the shared regressor.

The evolving fuzzy model identification uses a clustering method and a parameter identification method independently with different input signals. During the identification process, each training sample is used solely to identify the most recently added fuzzy rule, which enhances robustness and reduces computational complexity by eliminating the need to compute membership functions at every time step. The incremental multivariate Gaussian clustering is based on Welford's online algorithm for calculating variance [51]:

$$\underline{\nu}_i(n_i+1) = \underline{\nu}_i(n_i) + \frac{1}{n_i+1}\underline{e}_i(k), \tag{3}$$

$$\underline{S}_i(n_i+1) = \underline{S}_i(n_i) + \underline{e}_i(k)\left(\underline{z}(k) - \underline{\nu}_i(n_i+1)\right)^\top, \tag{4}$$

where the clustering error is computed as $\underline{e}_i(k) = \underline{z}(k) - \underline{\nu}_i(n_i)$, and the number of samples belonging to the i-th cluster is incremented as $n_i = n_i + 1$. The normalized covariance matrix is computed as $\underline{\Sigma}_i = \underline{S}_i / n_i$ only when it is required.

Evolving fuzzy systems commonly use the Fuzzily-Weighted Recursive Least Squares (wRLS) optimization method for the identification of the parameter of the consequence models [1, 36, 55]. However, the error formulation of this method results in the ARX model formulation, which is defined as $A(q)y(k) = B(q)u(k) + r$, while we would prefer an OE model $y(k) = \frac{B(q)}{A(q)}u(k) + r$. This is because the ARX model assumes a colored noise at the output of the system due to the auo-regression of the system output, while the OE model works in parallel to the system and assumes a more realistic white output noise. The OE model can be identified by filtering the regression signals with the denominator of the transfer function of the LLM as $\hat{A}(q)\underline{\varphi}_f(k) = \lim_{q\to 1}\hat{A}(q)\underline{\varphi}(k)$ [45]. The Filtered Recursive Least Squares method (fRLS) is then computed as [52]

$$\underline{\theta}_c(k) = \underline{\theta}_c(k-1) + \underline{\gamma}(k)\left(y_f(k) - \varphi_f^\top(k)\theta_c(k-1)\right), \tag{5}$$

$$\underline{\gamma}(k) = \frac{1}{\underline{\varphi}_f^\top(k)\underline{P}(k-1)\underline{\varphi}_f(k)+1}\underline{P}(k-1)\underline{\varphi}_f(k), \tag{6}$$

$$\underline{P}(k) = \left(\underline{\mathrm{I}} - \underline{\gamma}(k)\underline{\varphi}_f^\top(k)\right)\underline{P}(k-1), \tag{7}$$

where $\underline{P}(k)$ is an information matrix. The filter is updated online when enough data is collected, i.e. when the confidence interval of the identified model falls under a threshold value [45].

2.1 Rule addition mechanism

A new rule is added when the control error $e(k) = w(k) - y(k) \in \mathbb{R}$ changes abruptly $|e(k) - e(k-1)| > \kappa_e$. With this approach, a new rule is added when the reference signal changes, a large disturbance occurs, or the system experiences a shift in characteristics. The rule base is updated with $c = c + 1$ and initialized with a new antecedent multivariate Gaussian cluster:

$$n_c = 1, \quad \underline{\nu}_c = \underline{z}(k), \quad \underline{\Sigma}_c = \underline{0}, \tag{8}$$

and a new consequence OE-LLM as:

$$\underline{\theta}_c = \sum_{i=1}^{c-1} \Psi_i(k)\underline{\theta}_i, \quad \underline{P}(k) = \underline{P}_0, \tag{9}$$

where $\underline{P}_0 = \alpha_P \mathrm{I} \in \mathbb{R}^{n_\varphi \times n_\varphi}$ is the initial information matrix, with a large constant $\alpha_P \in [10^3, 10^7]$, used to ensure fast parameter convergence. The newly added rule is immediately used in the control law to enable a quick reaction to changes in the process. This requires setting the initial parameters of the rule's consequent linear model to the fuzzily weighted average of the existing rules at the time of creation.

2.2 Rule merging mechanism

Due to drifting data, a merging mechanism is required in cases of overlapping clusters, which can cause irregularities in rule activation and differences in the consequent OE-LLM. Overlapping clusters are detected based on the Bhattacharyya distance between two clusters. For multivariate Gaussian clusters, it is defined as [42]:

$$d_{pq}^{B} = \frac{1}{8}\left(\underline{v}_p - \underline{v}_q\right)^{\top} \underline{\Sigma}_{pq}^{-1}\left(\underline{v}_p - \underline{v}_q\right) + \frac{1}{2}\ln\left(\frac{\det(\underline{\Sigma}_{pq})}{\sqrt{\det(\underline{\Sigma}_p)\det(\underline{\Sigma}_q)}}\right), \tag{10}$$

where $\underline{\Sigma}_{pq} = \frac{1}{2}(\underline{\Sigma}_p + \underline{\Sigma}_q)$ is the average covariance matrix of the two clusters. The consequent OE-LLMs are compared based on the similarity of their transfer functions, which is simplified to the comparison of the steady-state variables [45]:

$$d_{pq}^{K} = \left|\frac{\hat{B}_p(1)}{\hat{A}_p(1)} - \frac{\hat{B}_q(1)}{\hat{A}_q(1)}\right|, \quad d_{pq}^{N} = \left|\frac{\hat{r}_p(1)}{\hat{A}_p(1)} - \frac{\hat{r}_q(1)}{\hat{A}_q(1)}\right|, \tag{11}$$

where d_{pq}^{K} and d_{pq}^{N} respectively represent the dissimilarity in steady-state gain and bias between the two compared systems.

The rules $\mathcal{R}_p$ and $\mathcal{R}_q$ are merged if the conditions for antecedent proximity and consequence similarity ($d_{pq}^{B} < \kappa_B$, $d_{pq}^{K} < \kappa_K$, and $d_{pq}^{N} < \kappa_r$) are satisfied. The antecedent clusters are merged based on the method proposed in [51], and the consequent parameters $\underline{\theta}_{pq}$ are merged as:

$$\underline{\theta}_{pq}(k) = \frac{n_p(k)\underline{\theta}_p(k) + n_q(k)\underline{\theta}_q(k)}{n_{pq}(k)}, \tag{12}$$

where $n_{pq}(k) = n_p(k) + n_q(k)$.

2.3 Rule removal mechanism

The proposed evolving system employs self-monitoring to determine the validity of the model online using the interleaved test-then-train or prequential approach [25], meaning that the samples used for model identification are first used to validate the existing model. Since the RLS method aims to minimize the MSE of the output, we use a fuzzy MSE (fMSE) to determine the quality of the local models online. It is computed recursively based on the fuzzy error proposed in [49] as:

$$\mathrm{MSE}_j(k) = \frac{M_j(k-1)\mathrm{MSE}_j(k-1) + \mathrm{e}_j^2(k)}{M_j(k)}, \tag{13}$$

$$M_j(k) = M_j(k-1) + \Psi_j^2(k), \tag{14}$$

where $\mathrm{e}_j(k) = \Psi_j(k)\big(y(k) - \hat{y}_j(k)\big)$ is the fuzzy error of the jth rule. The estimation is initialized with the creation of the rule as $\mathrm{MSE}_j = 0$ and $M_j = 1$. The fuzzy MSE is only computed if the rule membership activation is large enough, as this indicates that the sample should be represented by the rule: $\mu_i(k) > \kappa_\mu$. For samples that are far from every cluster, it is better not to attribute the error to any clusters, as we are not interested in samples that are distant from the rule base and they might even be detrimental to the model accuracy. Rules are removed if they have low accuracy ($\mathrm{MSE}_j > \kappa_{\mathrm{MSE}}$) or are overlapping ($d_{pq}^B > \kappa_B$) but p and q could not be merged, in which case the rule with the higher error is removed.

3 Model-based predictive control based on Predictive functional control concept

In general, a Model-based Predictive Controller (MPC) is a type of controller that uses a model of the controlled process ($y_m(k)$) to optimize a specified criterion function in order to obtain the finite-horizon control law. The criterion function J is generally given as:

$$J = \sum_{i=N_1}^{N_2} \Big(y_m(k+i) - y_r(k+i)\Big)^2 + \lambda \sum_{i=0}^{N_u} \Big(\Delta u(k+i)\Big)^2$$

where $y_m(k+i)$ represents the prediction of the process model output, $y_r(k+i)$ represents the prediction of the reference model, N_1 and N_2 are the lower and upper prediction horizons, $\Delta u(k+i)$ is the change in the control signal in the future, N_u is the control horizon (i.e., the number of future time instants where the control signal is taken into account), and λ is the weighting factor.

The solution in the form of a control law can be obtained analytically if the variables in the function are not subject to constraints. However, when dealing with constraints, the optimal solution should be obtained through optimization as follows:

$$\begin{aligned}
&\min_{\Delta u} J \\
&\text{subject to:} \\
&u_{\min} \le u(k+i) \le u_{\max}, \quad i = 1, \ldots, N_u \\
&\Delta u_{\min} \le \Delta u(k+i) \le \Delta u_{\max}, \quad i = 1, \ldots, N_u \\
&y_{\min} \le y_m(k+i) \le y_{\max}, \quad i = 1, \ldots, N_2
\end{aligned} \tag{15}$$

3.1 Predictive functional control concept

The concept of predictive functional control is based on the rule of equilibrium, given by the following equation:

$$y_r(k+h) - y_p(k) = y_m(k+h) - y_m(k) \tag{16}$$

where h stands for the coincidence horizon. This means that the change between the current value of the process output and the predicted reference model at the coincidence horizon should be equal to the difference between the current model output and the predicted model output at the coincidence horizon. The left side of the equation is called the process increment $\Delta_p(h)$, and the right side is called the model increment $\Delta_m(h)$. By comparing the change in the output of the identified model with the change in the output of the reference model, an integrating controller is achieved that can compensate for steady-state bias in the system model.

This means that at every time step k, a prediction of future model and reference model outputs ($y_m(k+i)$, $y_r(k+i)$, $i = 1,..,h$) is computed. This requires a dynamical model of the process, and some assumptions to easily calculate the predictions. We assume that the system is linear time-invariant during the prediction horizon, the reference signal is assumed to be constant $w(k)$, $i = 1, \cdots, h$, it is assumed to deal with mean level control law, which assumes a constant, the mean control signal, $u(k),\ i=1,\cdots,h$, and the output of the reference model at time step $y_r(k)$ is set to the measured output of the system $y_p(k)$. The output of a first order reference model at a time step $k+h$ is defined as [32]

$$y_r(k+h) = a_r^h y(k) + (1-a_r^h)w(k), \tag{17}$$

where $a_r,\ 0 < a_r < 1$ stands for the reference model pole in the discrete domain, which is selected by the control design. The reference model should have a faster time constants that the uncontrolled process and its order should be smaller or equal to the order of the system [1].

Taking into account that $y_p(k) = y_r(k)$, the control error is defiend as $e(k) = w(k) - y_r(k)$, and the prediction of control error at the coincidence horizon, with the assumption of constant reference $w(k)$ the following condition for the error is obtained

$$e(k+h) = a_r^h e(k). \tag{18}$$

which describes the exponentially decreasing control error. This implies that the process difference $\Delta_p(h)$, where $w(k) = w(k+h)$, and $y_r(k+h) = w(k) - a_r^h e(k)$, becomes equal

$$\Delta_p(h) = y_r(k+h) - y_p(k) = \left(1 - a_r^h\right) e(k) \tag{19}$$

From the basic rule of equilibrium given in Eq. 16 ($\Delta_m(h) = \Delta_p(h)$), the required value for the predicted model output is obtained. With this value of predicted model, the required dynamic of the whole closed-loop control system will be obtained. This is given as follows

$$y_m(k+h) = y_m(k) + \left(1 - a_r^h\right) e(k) \tag{20}$$

The dynamic of the controlled signal can be defined with the pole of the closed-loop system, and can be limited as $a_r \to 0$, to get the fastest possible response, i.e.

$$y_m(k+h) = y_m(k) + e(k) \tag{21}$$

3.2 Particle Swarm Optimization to obtain control value

The prediction of the model output at the coincidence horizon, $y_m(k+h)$, is now denoted as $y_{m,k+h}(u_k)$, indicating that it depends on the control signal applied at time instant k. The required value for the predicted model output at the coincidence horizon, given in Eq. 20, is now written as $y^{goal}_{m,k+h}$. The Particle Swarm Optimization (PSO) method is used to compute the control action, i.e., u_k or $u(k)$, to find the solution for $y_{m,k+h}(u_k)$ that is closest to $y^{goal}_{m,k+h}$. PSO is used here because it is easy to implement and flexible enough to add additional constraints to the control law if needed [11]. The idea is to evaluate a number of different values, i.e., particles u^i_k, as possible solutions, and to find the one that is optimal in the sense of the following simple criterion function:

$$\min_{u^i_k} J = \min_{u^i_k} \left(y^{goal}_{m,k+h} - y_{m,k+h}(u^i_k) \right)^2$$

If the process variables have constraints, i.e., $u_{min} \leq u_k \leq u_{max}$ and $\Delta u_{min} \leq \Delta u_k \leq \Delta u_{max}$, then the criterion function should be extended as follows:

$$\begin{aligned}\min_{u^i_k} J = \min_{u^i_k} & \left(y^{goal}_{m,k+h} - y_{m,k+h}(u^i_k) \right)^2 + \\ & + \lambda_{cons} \Big(1 + \mathrm{sign}(u^i_k - u_{max}) \Big) + \\ & + \lambda_{cons} \Big(1 + \mathrm{sign}(u_{min} - u^i_k) \Big) + \\ & + \lambda_{cons} \Big(1 + \mathrm{sign}(u^i_k - \Delta u) \Big)\end{aligned}$$

where $\lambda_{cons} \gg 1$, to heavily penalize solutions that violate the constraints.

The complete controller is presented in Algorithm 1. In order to adapt the model online the identification must be reliable and have a low number of problem-specific parameters.

Algorithm 1.: Evolving Neuro-Fuzzy Model-based Predictive Control with Particle Swarm Optimization

Input: $u_{\min}$, $u_{\max}$, $\underline{P}_0$, κ_μ, κ_B, κ_K, κ_N, κ_f, κ_e, κ_{MSE}
Initialize:
  $\underline{z} \leftarrow$ initial value, $c \leftarrow 1$, $\underline{\mu}_c \leftarrow z(0)$, $\underline{S}_c \leftarrow \underline{0}$, $n_c \leftarrow 1$, $P \leftarrow \underline{P}_0$, $\underline{\theta}_c \leftarrow \underline{0}$, $A_f(z) \leftarrow 1$

repeat
  $k \leftarrow k+1$
  Perform MPC with PSO
  Perform measurement and signal filtration
  Compute regression vector $\underline{\varphi}_f(k)$ and clustering vector $\underline{z}(k)$
  Compute model error (Eqs. (1) and (2))
  if $|e(k) - e(k-1)| > \kappa_e$ **then**
    repeat
      for $p = 1$ to c **do**
        Compute measure (Eqs. (10) and (11)) for $q \leftarrow \mathrm{argmin}(d^B_{pq})$
      Find $\mathrm{argmin}_{p,q}(d^B_{pq})$ subject to $(d^K_{pq} < \kappa_K) \wedge (d^N_{pq} < \kappa_N) \wedge (d^B_{pq} < \kappa_B)$
      if $(d^B_{pq} < \kappa_B)$ **then**
        Perform rule merging mechanism (Eq. (12))
    until no rules can be merged
    Perform rule removal mechanism (Eqs. (13) and (14))
    Perform rule addition mechanism (Eqs. (8) and (9))
  Perform incremental clustering (Eqs. (3) and (4))
  Perform recursive parameter identification (Eqs. (5), (6), and (7))
  Adapt filter $A_f(z)$
until end of control

4 Simulation study

In this study, we examined the proposed control algorithm on the control problem of a Plate Heat Exchanger (PHE). In the experiment, the control valve of the inlet cold water was abruptly closed, resulting in a shift in the parameters of the system. This allowed us to evaluate the capability of the evolving system to quickly adapt to a shift in parameters and assess the robustness of the proposed controller. The experiment on the PHE was conducted using a theoretical model of a plate heat exchanger as defined in [45], with the same parameters.

The examined theoretical system was sampled with a sampling time of $t_s = 4s$ and had a dead time of $t_d = 4t_s$. A heteroscedastic white Gaussian noise

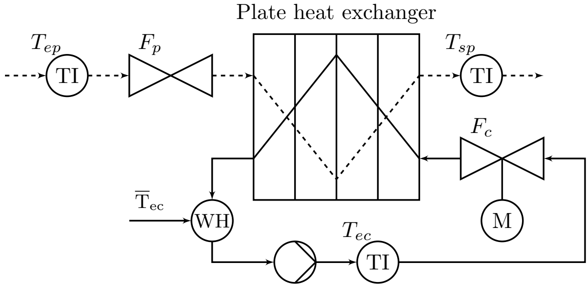

Figure 2: A schematic of the PHE pilot plant. The solid line represents the primary hot water flow circuit, while the dashed line represents the secondary cold water circuit.

$v \sim \mathcal{N}(0, 0.04K)$ was added as output error, where $K \in \mathbb{R}$, is the steady-state gain of the system in the observed operating point. The antecedent clustering vector was selected as $\underline{z}^\top(k) = [u(k), y(k)]$ and the consequence regression vector was selected as $\underline{\varphi}^\top(k) = [u(k-4-1), x(k-4-1), y(k-1), 1]$.

The experiment began with an initial model consisting of 5 rules that were identified using the proposed evolving methodology and a staircase excitation signal. The reference signal was changed every $k_h = 300$ samples to values in the range $w \in [8, 48]$, selected to ensure that the input signal does not reach saturation when the output reaches the reference value. The step height of 10 formed a staircase signal. At the beginning of the seventh step k=$6k_h$ the cold water valve F_p was partially closed from a value of 0.53 to 0.4 to simulate a shift in the environment, i.e., a fault of the inlet flow rate.

5 Real PHE Pilot plant study

The proposed eFMPC was used to control a nonlinear dynamical system in a changing environment. The plate heat exchanger pilot plant is subject to changes in the temperature of the inlet cold water and room air throughout the day [1]. The evolving model used for the PFC was built from scratch, with the system creating the first rule based on the first recorded sample without any prior knowledge, to examine the plug-and-play aspect of the proposed approach. However, note that even though the model identification starts from scratch with

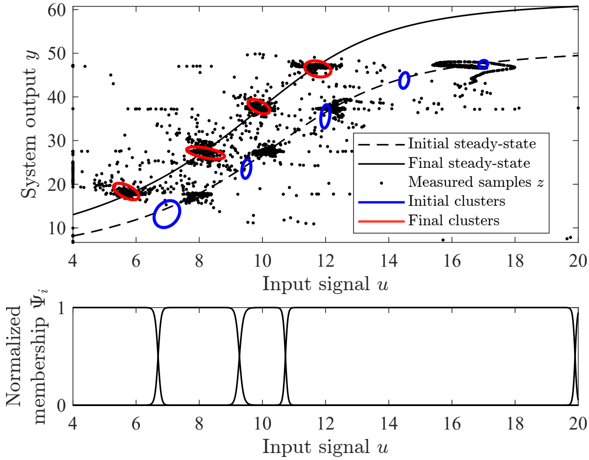

Figure 3: The top figure shows the antecedent multivariate Gaussian clusters at the start (blue ellipsoids) and at the end (red ellipsoids) of the simulation study after the disturbance, with the initial (dashed curve) and final (full curve) steady-state characteristic of the system. The bottom figure shows the final membership functions as a projection of the clusters to the input dimension.

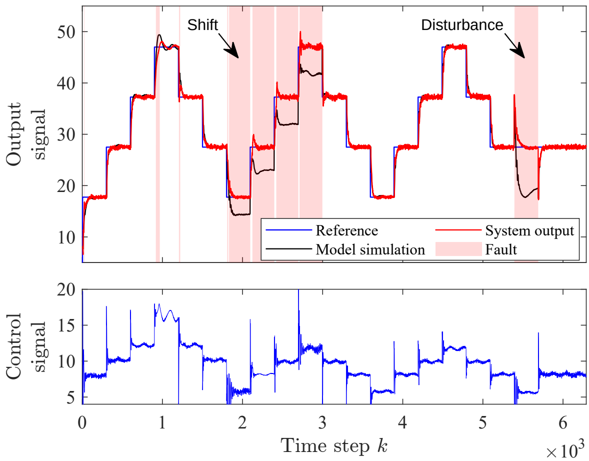

Figure 4: The output of the controlled system, the output of the model, and fault detection (top figure), while the control signal (bottom figure) with the proposed evolving predictive controller.

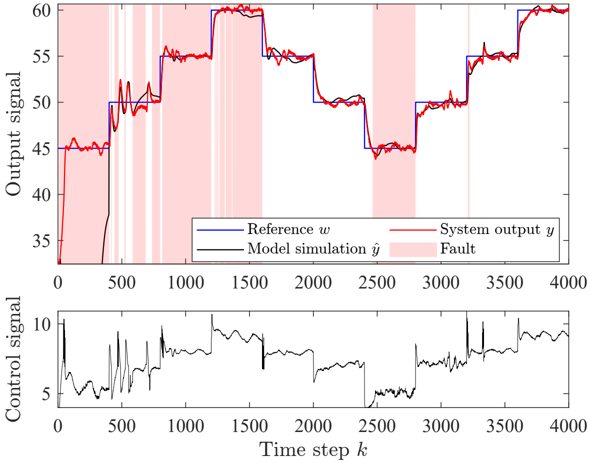

Figure 5: PHE pilot plant control.

a single cluster, the meta-parameters of the algorithm were still selected based on expert knowledge of the system.

6 Discussion

Self-monitoring during online learning present a dilemma in the sense that it is hard to determine whether the error is due to high parameter variance or actual bias. The proposed approach determines the local error of each rule, and only adapts the rule added last in order to avoid creating outlier rules due to large changes in the input signal and to maintain the number of rules as low as possible. This enables the adaptation of the evolving model while detecting low model accuracy, conceptual shifts and disturbances. The phenomena of drifting parameters is commonly addressed with a forgetting factor in the fRLS method, however we omitted it from this implementation as it requires persistent excitation [32] and relied rather on the self-monitoring approach as an alternative that does not suffer from this limitation. Another observation is that the Battacharyya distance works well for detecting overlapping clusters of similar size but can have some difficulties detecting overlapping clusters of different sizes. In general, this can result in small clusters becoming "trapped" inside larger clusters. However, this problem can easily be addressed with removal mechanism based on rule age. For practical application, additional robustness measure could be easily implemented as proposed in [1]. The PSO method is a reliable global optimization method that can find a solution in case of several local minima and it is very flexible to include additional constrains in the control law. However, if the number of particles of the PSO method is not large enough the global minima of the control action might not be found, which results in frequent changes of the input signal.

7 Conclusion

In this study, we proposed an Evolving Fuzzy Model Predictive Controller (eFMPC) that evolves the system model during control. The main benefit is that the model can be used for monitoring, detecting shifts in system parameters, and identifying disturbances as they occur, all while maintaining reference tracking and disturbance regulation. The model was validated through simulations and a real-world study on a plate heat exchanger. A downside of the proposed method is that the evolving identification procedure models every change in the system, including disturbances. Although these are eventually eliminated in the long term, they can affect the controller in the short term. Future work should examine the persistent excitation condition or an information criterion to selectively identify only informative samples.

References

[1] Goran Andonovski, Plamen Angelov, Sašo Blažič, and Igor Škrjanc. A practical implementation of Robust Evolving Cloud-based Controller with normalized data space for heat-exchanger plant. *Applied Soft Computing*, 48:29–38, 11 2016.

[2] Goran Andonovski, Plamen Angelov, Sašo Blažič, and Igor Škrjanc. Robust evolving cloud-based controller (recco). In Igor Škrjanc and Sašo

Blažič, editors, *2017 Evolving and Adaptive Intelligent Systems, EAIS 2017, Ljubljana, Slovenia, May 31 - June 2, 2017*, pages 1–6. IEEE, 2017.

[3] Goran Andonovski, Sašo Blažič, and Igor Škrjanc. Partial cloud-based evolving method for fault detection of HVAC system. In *2018 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2018, Rio de Janeiro, Brazil, July 8-13, 2018*, pages 1–6. IEEE, 2018.

[4] Goran Andonovski, Edwin Lughofer, and Igor Škrjanc. Evolving fuzzy model identification of nonlinear wiener-hammerstein processes. *IEEE Access*, 9:158470–158480, 11 2021.

[5] Goran Andonovski, Gasper Mušič, Sašo Blažič, and Igor Škrjanc. Evolving model identification for process monitoring and prediction of non-linear systems. *Eng. Appl. Artif. Intell.*, 68:214–221, 2018.

[6] Goran Andonovski and Igor Škrjanc. Evolving clustering algorithm based on average cluster distance - eCAD. In *IEEE International Conference on Evolving and Adaptive Intelligent System, EAIS 2022, Larnaca, Cyprus, May 25-26, 2022*, pages 1–5. IEEE, 2022.

[7] Plamen Angelov. Evolving Takagi-Sugeno Fuzzy Systems from Streaming Data (eTS+). *Evolving Intelligent Systems: Methodology and Applications*, pages 21–50, 4 2010.

[8] Plamen Angelov and Ronald Yager. A new type of simplified fuzzy rule-based system. *International Journal of General Systems*, 41:163–185, 2012.

[9] Julian Berberich, Johannes Kohler, Matthias A. Muller, and Frank Allgower. Linear Tracking MPC for Nonlinear Systems - Part II: The Data-Driven Case. *IEEE Transactions on Automatic Control*, 67:4406–4421, 9 2022.

[10] Saso Blažič, Igor Škrjanc, and Drago Matko. A robust fuzzy adaptive law for evolving control systems. *Evol. Syst.*, 5(1):3–10, 2014.

[11] S. Bououden, M. Chadli, and H. R. Karimi. An ant colony optimization-based fuzzy predictive control approach for nonlinear processes. *Information Sciences*, 299:143–158, 4 2015.

[12] Oscar Cartagena, Miha Ožbot, Doris Sáez, and Igor Škrjanc. Evolving fuzzy prediction interval for fault detection in a heat exchanger. *Appl. Soft Comput.*, 145:110625, 2023.

[13] Gregor Černe and Igor Škrjanc. Initial study on evolving state space neural networks (essnn). In Plamen Angelov, Yannis Manolopoulos, Edwin Lughofer, and Lazaros S. Iliadis, editors, *2018 IEEE Conference on Evolving and Adaptive Intelligent Systems, EAIS 2018, Rhodes, Greece, May 25-27, 2018*, pages 1–8. IEEE, 2018.

[14] Gregor Černe and Igor Škrjanc. Cascade affine constant recursive algorithm for model-based control. *IFAC-PapersOnLine*, 54:299–302, 1 2021.

[15] Bruno Sielly Jales Costa, Igor Škrjanc, Sašo Blažič, and Plamen Angelov. A practical implementation of self-evolving cloud-based control of a pilot plant. In *2013 IEEE International Conference on Cybernetics, CYBCO 2013, Lausanne, Switzerland, June 13-15, 2013*, pages 7–12. IEEE, 2013.

[16] Daniel F. Leite, Fernando A. C. Gomide, and Igor Škrjanc. Multiobjective Optimization of Fully Autonomous Evolving Fuzzy Granular Models. In *2019 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2019, New Orleans, LA, USA, June 23-26, 2019*, pages 1–7. IEEE, 2019.

[17] Daniel F. Leite, Igor Škrjanc, and Fernando A. C. Gomide. An overview on evolving systems and learning from stream data. *Evol. Syst.*, 11(2):181–198, 2020.

[18] Dejan Dovžan , Vito Logar, and Igor Škrjanc. Solving the sales prediction problem with fuzzy evolving methods. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2012, Brisbane, Australia, June 10-15, 2012*, pages 1–8. IEEE, 2012.

[19] Dejan Dovžan, Vito Logar, and Igor Škrjanc. Implementation of an Evolving Fuzzy Model (eFuMo) in a Monitoring System for a Waste-Water

Treatment Process. *IEEE Transactions on Fuzzy Systems*, 23:1761–1776, 10 2015.

[20] Dejan Dovžan and Igor 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*, pages 1–6. IEEE, 2015.

[21] Dejan Dovžan and Igor Škrjanc. Evolving fuzzy model for short-term prediction of energy consumption profiles. In *2016 IEEE Conference on Evolving and Adaptive Intelligent Systems, EAIS 2016, Natal, Brazil, May 23-25, 2016*, pages 98–102. IEEE, 2016.

[22] Edwin Lughofer and Igor Škrjanc. Evolving Error Feedback Fuzzy Model for Improved Robustness Under Measurement Noise. *IEEE Trans. Fuzzy Syst.*, 31(3):997–1008, 2023.

[23] Edwin Lughofer and Mahardhika Pratama and Igor Škrjanc. Online bagging of evolving fuzzy systems. *Inf. Sci.*, 570:16–33, 2021.

[24] Fabricio P. Lucas, Pyramo Costa, Rose M. S. Batalha, Daniel F. Leite, and Igor Škrjanc. Fault detection in smart grids with time-varying distributed generation using wavelet energy and evolving neural networks. *Evol. Syst.*, 11(2):165–180, 2020.

[25] Dongjiao Ge and Xiao Jun Zeng. Learning data streams online — An evolving fuzzy system approach with self-learning/adaptive thresholds. *Information Sciences*, 507:172–184, 1 2020.

[26] Goran Andonovski, Oscar Sipele, José Antonio Iglesias, Araceli Sanchis, Edwin Lughofer, and Igor Škrjanc. Detection of driver maneuvers using evolving fuzzy cloud-based system. In *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020, Canberra, Australia, December 1-4, 2020*, pages 700–706. IEEE, 2020.

[27] Igor Škrjanc, Goran Andonovski, José Antonio Iglesias, María Paz Sesmero, and Araceli Sanchis. Evolving Gaussian on-line clustering in social network analysis. *Expert Syst. Appl.*, 207:117881, 2022.

[28] Igor Škrjanc. eGauss+ evolving clustering in classification. In *15th IEEE International Symposium on Applied Computational Intelligence and Informatics, SACI 2021, Timisoara, Romania, May 19-21, 2021*, pages 533–538. IEEE, 2021.

[29] Igor Škrjanc. Evolving Cauchy Possibilistic Clustering for Data Stream Modeling and Classification. In *IEEE AFRICON 2023, Nairobi, Kenya, September 20-22, 2023*, pages 1–6. IEEE, 2023.

[30] Igor Škrjanc, Seiichi Ozawa, Tao Ban, and Dejan Dovžan. Large-scale cyber attacks monitoring using Evolving Cauchy Possibilistic Clustering. *Appl. Soft Comput.*, 62:592–601, 2018.

[31] Gorazd Karer and Igor Škrjanc. Interval-model-based global optimization framework for robust stability and performance of pid controllers. *Applied Soft Computing*, 40:526–543, 3 2016.

[32] Gorazd Karer, Igor Škrjanc, and Borut Zupančič. Self-adaptive predictive functional control of the temperature in an exothermic batch reactor. *Chemical Engineering and Processing*, 47:2379–2385, 2008.

[33] Gregor Klančar and Igor Škrjanc. Evolving principal component clustering with a low run-time complexity for LRF data mapping. *Applied Soft Computing*, 35:349–358, 10 2015.

[34] Daniel Leite, Rosangela Ballini, Pyramo Costa, and Fernando Gomide. Evolving fuzzy granular modeling from nonstationary fuzzy data streams. *Evolving Systems*, 3:65–79, 6 2012.

[35] Edwin Lughofer. On-line assurance of interpretability criteria in evolving fuzzy systems – Achievements, new concepts and open issues. *Information Sciences*, 251:22–46, 12 2013.

[36] Edwin Lughofer. Improving the robustness of recursive consequent parameters learning in evolving neuro-fuzzy systems. *Information Sciences*, 545:555–574, 2 2021.

[37] Edwin Lughofer, Jean Luc Bouchot, and Ammar Shaker. On-line elimination of local redundancies in evolving fuzzy systems. *Evolving Systems*, 2:165–187, 9 2011.

[38] Edwin Lughofer, Carlos Cernuda, Stefan Kindermann, and Mahardhika Pratama. Generalized smart evolving fuzzy systems. *Evolving Systems*, 6:269–292, 12 2015. Gen-Smart-EFS.

[39] Edwin Lughofer and Mahardhika Pratama. Online Active Learning in Data Stream Regression Using Uncertainty Sampling Based on Evolving Generalized Fuzzy Models. *IEEE Transactions on Fuzzy Systems*, 26:292–309, 2 2018.

[40] Edwin Lughofer, Mahardhika Pratama, and Igor Škrjanc. Incremental rule splitting in generalized evolving fuzzy regression models. In Igor Škrjanc and Sašo Blažič, editors, *2017 Evolving and Adaptive Intelligent Systems, EAIS 2017, Ljubljana, Slovenia, May 31 - June 2, 2017*, pages 1–8. IEEE, 2017.

[41] Edwin Lughofer, Mahardhika Pratama, and Igor Škrjanc. Incremental Rule Splitting in Generalized Evolving Fuzzy Systems for Autonomous Drift Compensation. *IEEE Trans. Fuzzy Syst.*, 26(4):1854–1865, 2018.

[42] Edwin Lughofer and Moamar Sayed-Mouchaweh. Autonomous data stream clustering implementing split-and-merge concepts – Towards a plug-and-play approach. *Information Sciences*, 304:54–79, 5 2015.

[43] Miloš Antić, Andrej Zdešar, and Igor Škrjanc. Depth-Image Segmentation Based on Evolving Principles for 3D Sensing of Structured Indoor Environments. *Sensors*, 21(13):4395, 2021.

[44] Miloš Antić, Andrej Zdešar, and Igor Škrjanc. Segmentation of Stereo-Camera Depth Image into Planar Regions based on Evolving Principal Component Clustering. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI 2021, Karlsruhe, Germany, September 23-25, 2021*, pages 1–5. IEEE, 2021.

[45] Miha Ožbot, Edwin Lughofer, and Igor Škrjanc. Evolving Neuro-Fuzzy Systems based Design of Experiments in Process Identification. *IEEE Transactions on Fuzzy Systems*, pages 1–11, 2022.

[46] Ožbot, Miha and Ozawa, Seiichi and Škrjanc, Igor. eFedGauss: A Federated Approach to Fuzzy Multivariate Gaussian Clustering. In *2024 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, page 1–8, June 2024.

[47] Mahardhika Pratama, Sreenatha G. Anavatti, Plamen P. Angelov, and Edwin Lughofer. PANFIS: A novel incremental learning machine. *IEEE Transactions on Neural Networks and Learning Systems*, 25:55–68, 1 2014.

[48] Sašo Blažič and Igor Škrjanc. Incremental Fuzzy C-Regression Clustering From Streaming Data for Local-Model-Network Identification. *IEEE Trans. Fuzzy Syst.*, 28(4):758–767, 2020.

[49] Igor Škrjanc. Fuzzy confidence interval for pH titration curve. *Applied Mathematical Modelling*, 35:4083–4090, 8 2011.

[50] Igor Škrjanc. Evolving Fuzzy-Model-Based Design of Experiments with Supervised Hierarchical Clustering. *IEEE Transactions on Fuzzy Systems*, 23:861–871, 8 2015.

[51] Igor Škrjanc. Cluster-Volume-Based Merging Approach for Incrementally Evolving Fuzzy Gaussian Clustering-eGAUSS+. *IEEE Transactions on Fuzzy Systems*, 28:2222–2231, 9 2020.

[52] Igor Škrjanc. An evolving concept in the identification of an interval fuzzy model of Wiener-Hammerstein nonlinear dynamic systems. *Information Sciences*, 581:73–87, 12 2021.

[53] Igor Škrjanc, Sašo Blažič, Edwin Lughofer, and Dejan Dovžan. Inner matrix norms in evolving Cauchy possibilistic clustering for classification and regression from data streams. *Inf. Sci.*, 478:540–563, 2019.

[54] Igor Škrjanc, Araceli Sanchis de Miguel, José Antonio Iglesias, Agapito Ledezma, and Dejan Dovžan . Evolving cauchy possibilistic clustering

based on cosine similarity for monitoring cyber systems. In Igor Škrjanc and Sašo Blažič, editors, *2017 Evolving and Adaptive Intelligent Systems, EAIS 2017, Ljubljana, Slovenia, May 31 - June 2, 2017*, pages 1–5. IEEE, 2017.

[55] Igor Škrjanc, Jose Iglesias, Araceli Sanchis, Daniel Leite, Edwin Lughofer, and Fernando Gomide. Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A Survey. *Information Sciences*, 490:344–368, 7 2019.

[56] Igor Škrjanc, Seiichi Ozawa, Dejan Dovžan , Tao Ban, Junji Nakazato, and Jumpei Shimamura. Evolving cauchy possibilistic clustering and its application to large-scale cyberattack monitoring. In *2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017, Honolulu, HI, USA, November 27 - Dec. 1, 2017*, pages 1–7. IEEE, 2017.

[57] Jan C. Willems, Paolo Rapisarda, Ivan Markovsky, and Bart L.M. De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54:325–329, 4 2005.

[58] Sofia Yousuf and Muhammad Bilal Kadri. Comparison of two different model free fuzzy control architectures based on inverse plant modeling. *2017 1st International Conference on Latest Trends in Electrical Engineering and Computing Technologies, INTELLECT 2017*, 2018-January:1–7, 1 2018.

[59] Andrej Zdešar, Dejan Dovžan, and Igor Škrjanc. Self-tuning of 2 DOF control based on evolving fuzzy model. *Appl. Soft Comput.*, 19:403–418, 2014.

[60] Žiga Stržinar and Igor Škrjanc. Self-tuned Model-Based Predictive Control Using Evolving Fuzzy Model of a Non-linear Dynamic Process. In *Explainable AI and Other Applications of Fuzzy Techniques - Proceedings of the 2021 Annual Conference of the North American Fuzzy Information Processing Society, NAFIPS 2021, Virtual Event / West Lafayette, IN, USA, June 7-9, 2021*, volume 258 of *Lecture Notes in Networks and Systems*, pages 406–421. Springer, 2021.