2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Lifelong learning for monitoring and adaptation of data-based dynamical models: a statistical process control approach

Laura Boca de Giuli, Alessio La Bella, Giuseppe De Nicolao and Riccardo Scattolini

Abstract— **This paper addresses the monitoring and continual learning of data-based dynamical models. Throughout the lifespan of any process, many changes can occur. In an indirect control design framework, in order to maintain an effective control system, it is crucial to monitor the modelling performance and adapt the existing model to possible system variations while preserving previously acquired information. A comprehensive methodology is hence proposed to detect a system-model mismatch and its cause, and to update the model accordingly. The proposed idea consists in leveraging control charts constructed on operational data to spot an anomaly and to determine its cause (endogenous or exogenous). The procedure then provides an adaptation algorithm based on the type of change detected: if endogenous, the model is "partially" updated by means of a Moving Horizon Estimation (MHE) algorithm, if exogenous, the model is "incrementally" updated by means of a model uncertainty estimation algorithm. The proposed methodology is tested in simulation on a district heating system benchmark, showing promising results from the monitoring and continual learning perspective.**

Index Terms— **Lifelong learning, statistical process control, moving horizon estimation, model uncertainty estimation.**

I. INTRODUCTION

In recent years, data-based techniques have gained popularity in the control community thanks to their greater manageability with respect to large-scale physical models. In this context, two data-driven control design approaches can be adopted, i.e., direct methods, in which the control system is designed directly from data [1], and indirect methods, whose control law is synthesized based on a recovered data-driven model [2]. This last approach, especially when employed for designing Model Predictive Control (MPC) regulators [3], relies on the Certainty Equivalence Principle (CEP), which assumes that the underlying model resembles the real system [4]. Throughout the lifespan of any system, many changes may occur, resulting in an unreliable model and in deteriorated controller performances. As a consequence, two challenges arise: *i)* how to effectively monitor the model reliability and *ii)* how to adapt the existing model to changes without discarding previously acquired knowledge.

In order to monitor model goodness over process operation, some methods have been proposed in the literature, such as those based on open-loop modelling error triggering mechanisms [5], or those based on predictions-observations precision thresholds [6]. Even though thanks to these methods it is possible to detect a system-model mismatch, none of them distinguishes its cause. Being able of differentiating the type of anomaly is crucial to effectively adapt the existing model to the specific scenario encountered. In [7], a monitoring algorithm able to distinguish various system anomalies is shown, but no adaptation solution is proposed. By contrast, in [8], different adaptation algorithms based on the scenario encountered are proposed, but how to distinguish among various cases is not discussed. Being able to effectively adapt the system model to changes is of key importance to maintain high standard of identification and control performances. The ability to constantly learn over time by accommodating new knowledge while preserving previously learned experiences is commonly referred to as "lifelong learning". The latter is a well-established challenge for machine learning since the continuous acquisition of incrementally available information may interfere with previously learned knowledge, leading to the "catastrophic forgetting" [9]. Different algorithms have been proposed in the literature to update models over time, such as those based on Moving Horizon Estimators (MHE) [10], [11], on model uncertainty estimation [3], [12], or those retraining the existing model with online operational data [5]. None of these approaches, however, proposes an adaptation algorithm based on the type of anomaly.

In view of the above discussion, a comprehensive monitoring and lifelong learning algorithm for data-based dynamic models is here proposed. The main contributions of the work are summarized below. First, a monitoring strategy able to distinguish the anomaly cause is proposed. Two main categories of anomaly are identified: *i)* endogenous changes, e.g., due to structural plant modifications and *ii)* exogenous changes, e.g., due to operating conditions shifts. In order to differentiate such anomalies, the proposed procedure makes use of a well-known statistical process control tool, i.e., control charts [13]. In contrast to their traditional use [14], control charts are here leveraged to assess modelling reliability and to distinguish the source of the anomaly, if detected, by means of an operational data analysis. The scenario distinction enables to update the existing model according to the anomaly detected. If an endogenous anomaly is spotted, a "model partial update" based on MHE is performed. If an exogenous anomaly is detected, a "model incremental update" based on model uncertainty estimation is performed. In this work, the proposed approach is tested in simulation on a District Heating System (DHS) benchmark, i.e., the AROMA DHS [15], to monitor and adapt its data-based model, showing promising results.

Laura Boca de Giuli, Alessio La Bella and Riccardo Scattolini are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy (e-mails: *laura.bocadegiuli@polimi.it*, *alessio.labella@polimi.it*, and *riccardo.scattolini@polimi.it*). Giuseppe De Nicolao is with the Dipartimento di Ingegneria Industriale e dell'Informazione, Università degli Studi di Pavia, 27100 Pavia, Italy (e-mail: *giuseppe.denicolao@unipv.it*).

Notation

Given a vector $z \in \mathbb{R}^n$, its ith element is denoted as z_i. Considering a scalar $\alpha \in \mathbb{R}$, the inequality $z \leq \alpha$ is intended element-wise, i.e., $z_i \leq \alpha, \ \forall i \in \{1, \dots, n\}$. Moreover, given a vector variable $z \in \mathbb{R}^n$ and a sequence of m observations of z over time, i.e., $z(1), \dots, z(m)$, the matrix containing the m observations of z is denoted in bold, i.e., $\mathbf{z} = [z(1), \dots, z(m)] \in \mathbb{R}^{n \times m}$, and it can be compactly expressed as $\mathbf{z} = \{z(i)\}_{i=1}^{m}$.

II. Problem statement

Consider a discrete-time system $\mathcal{S}$:

$$\mathcal{S}: \begin{cases} x(k+1) = f(x(k), u(k), d(k)) \\ y(k) = g(x(k), u(k), d(k)) \end{cases}, \tag{1}$$

where f and g are non-linear functions, $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $y \in \mathbb{R}^{n_y}$, $d \in \mathbb{R}^{n_d}$ are the state, input, output and disturbance vectors, respectively, whereas k is the adopted discrete time index. In data-driven approaches, a dynamical model $\mathcal{M}$ of (1) can be identified from data for control design purposes, which generically reads as

$$\mathcal{M}: \begin{cases} \hat{x}(k+1) = \hat{f}(\hat{x}(k), u(k), d(k); \Theta) \\ \hat{y}(k) = \hat{g}(\hat{x}(k), u(k), d(k); \Theta) \end{cases}, \tag{2}$$

where $\hat{f}$ and $\hat{g}$ are non-linear functions, $\hat{x} \in \mathbb{R}^{\hat{n}_x}$ and $\hat{y} \in \mathbb{R}^{n_y}$ are the model state and output, respectively, whereas $\Theta \in \mathbb{R}^{n_\Theta}$ is the vector of model parameters that must be tuned during the training procedure. The latter is carried out by means of open-loop data collected by feeding the system with exciting inputs (e.g., multilevel pseudorandom binary sequences or MPRBS), exploring the entire operating range of interest, and with disturbances whose profiles derive from historical data. $\mathcal{M}$ can be implemented according to different model structures [16], such as Recurrent Neural Networks (RNNs), which are particularly suitable to approximate non-linear dynamical systems, thanks to their universal approximation capabilities [17].

According to indirect data-based control methods, after a model $\mathcal{M}$ is estimated from data, a control system $C_{\mathcal{M}}$ regulating $\mathcal{S}$ is designed to obtain desired closed-loop performances. The controller $C_{\mathcal{M}}$ may range from simple Proportional-Integral-Derivative (PID) to more complex MPC regulators. The correct functioning of the control scheme obviously depends on the accuracy of the identified model, based on which the controller is designed, typically relying on the CEP. Thus, the accuracy of model $\mathcal{M}$ is essential to be monitored in closed-loop online operation. This task can be addressed by supervising the identification error $e_{\hat{y}}(k) = y(k) - \hat{y}(k)$, which should be reasonably small if the model $\mathcal{M}$ accurately represents system $\mathcal{S}$ when fed with the same inputs and disturbances. However, the identification error $e_{\hat{y}}$ may unacceptably increase if:

- system $\mathcal{S}$ internally changes, e.g., in case of a structural plant modification (endogenous cause);
- system $\mathcal{S}$ operates in not known conditions, e.g., in case the disturbance d takes values not included in the training data used to learn $\mathcal{M}$ (exogenous cause). For simplicity, the scenario in which system $\mathcal{S}$ is fed with an input u outside the known operating range is not addressed here, since it is assumed that model $\mathcal{M}$ has been trained with exciting input signals covering the entire operating range of interest.

This work therefore aims first to supervise the integrity of model $\mathcal{M}$ in (2) during the closed-loop operation of system $\mathcal{S}$ regulated by $\mathcal{C}_{\mathcal{M}}$. Then, if an anomaly is detected, the second challenge consists in discerning its cause, i.e., if unacceptable modelling errors derive from internal system changes or from external unknown operating conditions. According to the anomalous scenario encountered, the third challenge lies in updating the existing model taking into account the new available information while not discarding previously acquired knowledge.

III. Proposed procedure

The proposed resolution procedure for the aforementioned problems is summarized by the flowchart reported in Figure 1. In a nutshell, it is here proposed to periodically assess the modelling reliability by means of control charts monitoring if model $\mathcal{M}$ operates as expected or if anomalies occur. A control chart is an online monitoring technique used in statistical process control to detect the occurrence of assignable causes of process changes so that corrective actions may be undertaken [13]. Hereafter, the different steps of the flowchart in Figure 1 are described, considering that Step 1 and Step 2, i.e., the identification of $\mathcal{M}$ and the design of $C_{\mathcal{M}}$, have been described in Section II.

Control charts characterization

In **Step 3** of the flowchart in Figure 1, closed-loop operational data are collected and control charts are constructed so as to properly characterize the scenario where model $\mathcal{M}$ accurately represents system $\mathcal{S}$. In particular, Step 3 is structured in the following operations:

- **Step 3.1)** A dataset $\mathcal{D} = \{y(i), \hat{y}(i), d(i), u(i)\}_{i=1}^{N}$ containing N observations, i.e., including all inputs and outputs of $\mathcal{S}$ and $\mathcal{M}$, is collected during the online operation of the control system. Disturbances d are assumed to be measurable or estimated through local observers. The benchmark dataset $\mathcal{D}$ is used to construct the modelling error control chart based on $\mathbf{e}_{\hat{\mathbf{y}}} = \{y(i) - \hat{y}(i)\}_{i=1}^{N}$, and the disturbance control chart based on $\mathbf{d} = \{d(i)\}_{i=1}^{N}$.
- **Step 3.2)** Considering that $\mathbf{e}_{\hat{\mathbf{y}}} \in \mathbb{R}^{n_y \times N}$ and $\mathbf{d} \in \mathbb{R}^{n_d \times N}$, multivariate control charts must be leveraged. The statistical (Mahalanobis) distance T^2 is therefore exploited, i.e., a popular statistical tool in multivariate process analysis [13]. Specifically, consider a reference dataset $\mathbf{z}$ containing N observations of a variable z, and a monitoring dataset $\widetilde{\mathbf{z}}$ containing $\widetilde{N}$ observations of the same variable. The Mahalanobis distance of $\widetilde{\mathbf{z}}$ with respect to $\mathbf{z}$ is computed as

$$T^2(\widetilde{\mathbf{z}}, \mathbf{z}) = \{(\widetilde{\mathbf{z}}(i) - \mu_{\mathbf{z}})'(\Sigma_{\mathbf{z}})^{-1}(\widetilde{\mathbf{z}}(i) - \mu_{\mathbf{z}})\}_{i=1}^{\widetilde{N}}, \tag{3}$$

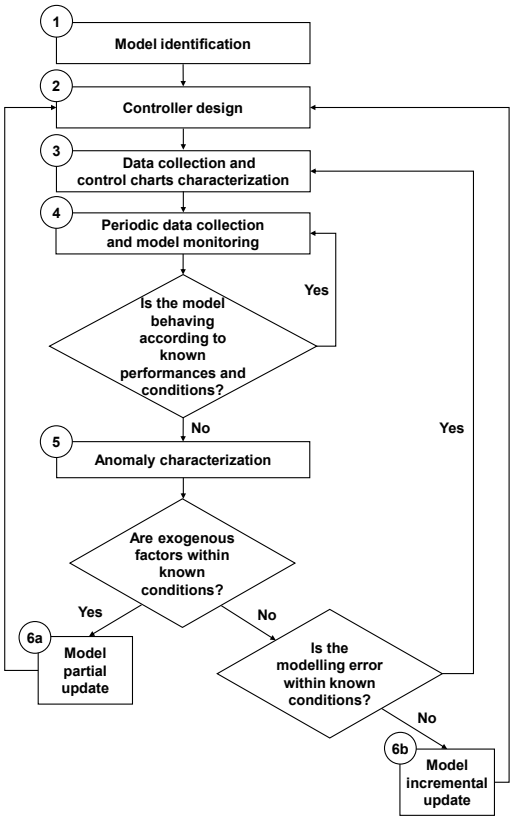

Fig. 1. Flowchart of the proposed procedure: the number of each step is reported on the corresponding top-left corner.

where

$$\mu_{\mathbf{z}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{z}(i), \tag{4}$$

$$\Sigma_{\mathbf{z}} = \frac{1}{N-1}\sum_{i=1}^{N} (\mathbf{z}(i)-\mu_{\mathbf{z}})(\mathbf{z}(i)-\mu_{\mathbf{z}})'. \tag{5}$$

In detail, $T^2(\widetilde{\mathbf{z}},\mathbf{z}) \in \mathbb{R}^{\widetilde{N}}$ contains a sequence of Mahalanobis distances, computed for each observation in $\widetilde{\mathbf{z}}$ with respect to the reference dataset $\mathbf{z}$. The first argument of $T^2(\widetilde{\mathbf{z}},\mathbf{z})$ corresponds to the monitoring dataset, whereas the second argument to the reference dataset used to compute the mean vector $\mu_{\mathbf{z}}$ and the covariance matrix $\Sigma_{\mathbf{z}}$, as evident from (4)-(5). Considering the problem here addressed, the Mahalanobis distances $T^2(\mathbf{e}_{\hat{\mathbf{y}}},\mathbf{e}_{\hat{\mathbf{y}}})$ and $T^2(\mathbf{d},\mathbf{d})$ are computed by using as monitoring dataset the reference one, in order to build the reference multivariate control charts of $\mathbf{e}_{\hat{\mathbf{y}}}$ and $\mathbf{d}$, as discussed next.

- **Step 3.3)** The Hotelling T^2 multivariate control charts are here developed, being suited to monitor vector variables through their Mahalanobis distances. More details on their design are available in [13]. The control charts development is commonly structured into two phases. First, it is evaluated if the available observations of monitored variables are "in control" from the statistical perspective, so that they can be used as reference. If this phase is successful, observations are exploited to develop benchmark control charts. This requires the definition, for each monitored variable, of an Upper Control Limit (UCL) and of a Lower Control Limit (LCL), so as to check if they are respected during the online monitoring process, i.e., LCL $\leq T^2(\widetilde{\mathbf{z}},\mathbf{z}) \leq$ UCL. If UCL and/or LCL are violated by future collected data, the process will no longer be "in control" and a fault will have occurred. As suggested in [13], being the Mahalanobis distance a positive quantity, LCL is set to 0. Since the assumption of data normal distribution may be not verified for $T^2(\mathbf{e}_{\hat{\mathbf{y}}},\mathbf{e}_{\hat{\mathbf{y}}})$ and $T^2(\mathbf{d},\mathbf{d})$, their UCLs, i.e., UCL_e and UCL_d, respectively, are here estimated from the observed data in $\mathcal{D}$ using percentiles. For a continuous random variable X, the k-th percentile is a value p_k such that $\mathbb{P}(X \leq p_k) = k/100$.

Hence, after having performed the operations in Step 3 for dataset $\mathcal{D}$, control charts characterized by their LCLs and UCLs are defined both for $\mathbf{e}_{\hat{\mathbf{y}}}$ and for $\mathbf{d}$. The latter are exploited in the subsequent steps to monitor if model $\mathcal{M}$ is properly representing $\mathcal{S}$ during online operation and, if not, to assess the nature of the anomaly.

Model monitoring

The model monitoring phase, represented by **Step 4** in Figure 1, is structured as follows:

- **Step 4.1)** At the end of each monitoring period, a new dataset $\widetilde{\mathcal{D}} = \{\widetilde{y}(i),\widetilde{\hat{y}}(i),\widetilde{d}(i),\widetilde{u}(i)\}_{i=1}^{\widetilde{N}}$ containing $\widetilde{N}$ observations is collected, then $\widetilde{\mathbf{e}}_{\hat{\mathbf{y}}} = \{\widetilde{y}(i)-\widetilde{\hat{y}}(i)\}_{i=1}^{\widetilde{N}}$ and $\widetilde{\mathbf{d}} = \{\widetilde{d}(i)\}_{i=1}^{\widetilde{N}}$ are defined.
- **Step 4.2)** The Mahalanobis distances of the new observed variables in $\widetilde{\mathcal{D}}$ with respect to the benchmark dataset $\mathcal{D}$ are computed using (3)-(5), i.e., $T^2(\widetilde{\mathbf{e}}_{\hat{\mathbf{y}}},\mathbf{e}_{\hat{\mathbf{y}}})$ and $T^2(\widetilde{\mathbf{d}},\mathbf{d})$.
- **Step 4.3)** Using the bounds defined in Step 3.3, in order to detect whether an anomaly occurred or not, the following condition is checked:

$$(\,T^2(\widetilde{\mathbf{e}}_{\hat{\mathbf{y}}},\mathbf{e}_{\hat{\mathbf{y}}}) \leq \text{UCL}_e\,) \wedge (\,T^2(\widetilde{\mathbf{d}},\mathbf{d}) \leq \text{UCL}_d\,). \tag{6}$$

If (6) is verified, it means that the identification errors and the disturbances are evolving according to the scenarios explored in $\mathcal{D}$, implying that model $\mathcal{M}$ is still able to correctly capture the system dynamics. Therefore, no further actions are required and Step 4 is restored, as evident from the flowchart in Figure 1. If (6) is not verified, an unknown model operation is detected and therefore Step 5 is required.

Anomaly characterization

Step 5 is essential to characterize the type of anomaly encountered in $\widetilde{\mathcal{D}}$ if condition (6) is violated. This step develops as follows:

- **Step 5.1)** As depicted in Figure 1, in order to distinguish the type of anomaly, it must be checked whether disturbances are operating within known conditions. Thus, the following condition is first checked:

$$T^2(\widetilde{\mathbf{d}},\mathbf{d}) \leq \text{UCL}_d. \tag{7}$$

If (7) is verified, the observed exogenous factors are statistically close to the original ones, and therefore (6) is violated due to an endogenous change in $\mathcal{S}$. Thus, Step 6a must be performed to adapt $\mathcal{M}$ to this new situation, as described later. On the other hand, if (7) is not verified, a new disturbance condition not represented in $\mathcal{D}$ is detected, and therefore Step 5.2 is performed.

- **Step 5.2)** When a new operating condition is detected, i.e., (7) is violated, the following condition is checked:

$$T^2(\widetilde{\mathbf{e}}_{\hat{\mathbf{y}}}, \mathbf{e}_{\hat{\mathbf{y}}}) \leq \mathrm{UCL}_e. \tag{8}$$

If (8) is verified and (7) is violated, it means that model $\mathcal{M}$ is able to accurately represent system $\mathcal{S}$, even though disturbances in $\widetilde{\mathcal{D}}$ are not represented in the original dataset $\mathcal{D}$. Thus, it is necessary to add a new control chart associated with $\widetilde{\mathcal{D}}$ (Step 3 is restored), without modifying the model $\mathcal{M}$ or the controller $C_{\mathcal{M}}$, as the modelling error is acceptable. If (7) and (8) are not verified, a new operating condition associated with unacceptable modelling errors is detected, and therefore Step 6b must be performed.

Ultimately, if a model anomaly is detected by control charts, the original model $\mathcal{M}$ must be adapted to the specific type of scenario encountered, as described in the following.

Model partial update for endogenous anomalies

According to the proposed procedure, an endogenous anomaly is detected if (7) is respected and (8) is violated. In this paper, an endogenous anomaly is intended as a change in a process internal factor which leads to a modification in the dynamics of $\mathcal{S}$, no more fully represented by model $\mathcal{M}$. In **Step 6a** of the flowchart in Figure 1, in order to adapt the model to such internal changes while still preserving the information contained in $\mathcal{M}$, i.e., avoiding catastrophic forgetting, an MHE inspired by [10] is exploited. The proposed algorithm, based on the underlying MHE optimization problem, aims at seeking the parameters update for model $\mathcal{M}$ in (2) that best describes the new collected data $\widetilde{\mathcal{D}}$, and it can be stated as

$$\Theta^{\mathrm{up}\,*} = \underset{\Theta^{\mathrm{up}}, \hat{x}(0)}{\arg\min} \sum_{k=1}^{\widetilde{N}} ||\widetilde{y}(k) - \hat{y}^{\mathrm{up}}(k)||^2 + \gamma ||\Theta - \Theta^{\mathrm{up}}||^2 \tag{9a}$$

subject to, $\forall\, k \in \{0, \ldots, \widetilde{N}\}$,

$$\hat{x}(0) = \bar{x}_0, \tag{9b}$$

$$\hat{x}(k+1) = \hat{f}(\hat{x}(k), \widetilde{u}(k), \widetilde{d}(k); \Theta^{\mathrm{up}}), \tag{9c}$$

$$\hat{y}^{\mathrm{up}}(k) = \hat{g}(\hat{x}(k), \widetilde{u}(k), \widetilde{d}(k); \Theta^{\mathrm{up}}), \tag{9d}$$

$$\phi\Theta^{\mathrm{up}} = \phi\Theta, \tag{9e}$$

where $\Theta \in \mathbb{R}^{n_\Theta}$ represents the vector of parameters of the original model $\mathcal{M}$ (2), $\Theta^{\mathrm{up}} \in \mathbb{R}^{n_\Theta}$ is the optimization variable, $\Theta^{\mathrm{up}\,*}$ is the optimal solution to (9), whereas $\phi \in \mathbb{R}^{n_\Theta \times n_\Theta}$ is a diagonal matrix used to constraint some components of Θ^{up} to be equal to the corresponding ones of Θ, as discussed later. The cost function (9a) penalizes the mismatch between the actual measured output $\widetilde{y}$ and that of the updated model $\hat{y}^{\mathrm{up}}$, and, moreover, discourages significant deviations from the previously computed optimal solution Θ. Hence, the coefficient γ defines the trade-off between the need to improve model performance (γ small) and the necessity not to forget the information previously acquired. The updated model $\mathcal{M}^{\mathrm{up}}$ is embedded in the MHE formulation in (9c)-(9d), with its state initialized in (9b) with $\bar{x}_0$, supposed to be measured or estimated by an appropriate state observer. To reduce computational complexity and avoid catastrophic forgetting, constraint (9e) is here added to update just some model parameters by means of the selective matrix ϕ. For instance, if RNNs are used, as visible in Figure 2(a), only the parameters related to the output layer could be updated, i.e., the weights and biases of the output equation $\hat{y}$ in (2). In this way, the information acquired through the original training procedure is embedded in the hidden layers parameters, whereas the information related to the current modified system is stored in the updated optimal solution $\Theta^{\mathrm{up}\,*}$. After the model partial update, the previously built control charts must be discarded, as the system has changed and the updated model $\mathcal{M}^{\mathrm{up}}$ is characterized by different modelling errors. Thus, the procedure restarts from Step 2, where $C_{\mathcal{M}^{\mathrm{up}}}$ is designed based on the updated model $\mathcal{M}^{\mathrm{up}}$.

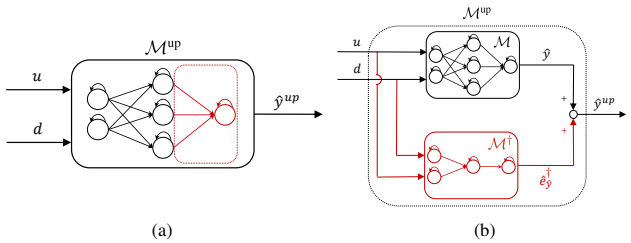

Fig. 2. (a) Model partial update in the RNN case: the partially updated model is highlighted in red. (b) Model incremental update in the RNN case: the incrementally updated model $\mathcal{M}^{\dagger}$ is highlighted in red.

Model incremental update for exogenous anomalies

According to the proposed procedure for anomaly characterization described in Step 5, an exogenous anomaly is detected if both (7) and (8) are not respected. In this paper, an exogenous anomaly is intended as a change in a process external factor which leads to a system dynamics modification that model $\mathcal{M}$ is not able to represent. Being the original model able to correctly identify the system dynamics in already explored external operating conditions, the model incremental update performed in **Step 6b** involves an enlargement of $\mathcal{M}$ so as to make it capable of identifying $\mathcal{S}$ even in presence of new observed exogenous signals. To this end, the proposed idea consists in exploiting an additive model that estimates the original model identification error (uncertainty) offline. Therefore, a new model $\mathcal{M}^{\dagger}$ is defined, operating in parallel with the original model $\mathcal{M}$ in (2), which reads as

$$\mathcal{M}^{\dagger}: \begin{cases} \hat{x}^{\dagger}(k+1) = \hat{f}^{\dagger}(\hat{x}^{\dagger}(k), u(k), d(k); \Theta^{\dagger}) \\ \hat{e}^{\dagger}_{\hat{y}}(k) = \hat{g}^{\dagger}(\hat{x}^{\dagger}(k), u(k), d(k); \Theta^{\dagger}) \end{cases}, \tag{10}$$

so that the overall updated output is

$$\hat{y}^{\text{up}}(k) = \hat{y}(k) + \hat{e}^{\dagger}_{\hat{y}}(k)\,, \tag{11}$$

with $\hat{y}$ deriving from the original model $\mathcal{M}$ in (2). For instance, considering $\mathcal{M}$ and $\mathcal{M}^{\dagger}$ identified by an RNN, the model incremental update scheme implemented is shown in Figure 2(b). In order to obtain an effective lifelong learning algorithm and thus to avoid catastrophic forgetting, $\hat{e}^{\dagger}_{\hat{y}}$ must not negatively impact on $\hat{y}$ when standard operating conditions are restored. To achieve this, the incremental model reported in (10) must be trained with a dataset $\mathcal{D}^{\dagger} = \mathcal{D}_o \cup \widetilde{\mathcal{D}}$, including a subset of the original operating conditions, i.e., $\mathcal{D}_o \subseteq \mathcal{D}$, and the dataset $\widetilde{\mathcal{D}}$ containing the new explored conditions. After the model incremental update, a new control chart both for errors and for disturbances must be constructed considering the observations in $\widetilde{\mathcal{D}}$. The original control charts, however, must not be discarded since they are still valid to monitor the model in the already known operating conditions. Thus, the procedure restarts from Step 2, where $C_{\mathcal{M}^{\text{up}}}$ is designed based on the updated model $\mathcal{M}^{\text{up}}$, defined by (2), (10), and (11).

IV. Numerical Results

The proposed lifelong learning algorithm is applied to a District Heating System (DHS), i.e., an efficient energy system crucial to reach the decarbonization objectives. A DHS is generally composed of a heating station and of an insulated water pipeline network transferring the generated heat to thermal loads [18]. The case study analysed in this work is the AROMA DHS, described in [15]. The input is the supply temperature at the heating station, i.e., $u = T_0^s$, whereas the external disturbances are the thermal loads demands, i.e., $d = [\{P_i^c\}'_{\forall i \in \mathcal{N}_c}]'$, where $\mathcal{N}_c$ is the thermal loads set. The output vector is $y = [T_0^r, q_0, \{T_i^s, T_i^c, q_i^c\}'_{\forall i \in \mathcal{N}_c}]'$, i.e., including the return temperature T_0^r and the water flow q_0 at the heating station, the load supply temperatures T_i^s, output temperatures T_i^c and water flows q_i^c. The AROMA DHS physical model described in [15] has been leveraged to develop a dynamic simulator, exploited to generate input-output data (see [19] for further details). To quantitatively assess the modelling performances of the developed data-based models, the FIT (%) index reported in [19] is employed.

According to Step 1 of the flowchart in Figure 1, the data-based original model $\mathcal{M}$ (2) of the AROMA DHS is based on a Physics-informed RNN, whose training procedure required 136 minutes, whereas Step 2 consists in the implementation of a Nonlinear MPC (NMPC) algorithm, exploiting the derived data-based model, as discussed in [19]. As explained in Section III, according to Step 3, it is first necessary to collect a benchmark dataset in order to characterize the control charts in a standard setting. The benchmark dataset $\mathcal{D}$ is collected through a monthly closed-loop simulation of the controlled system, where u is computed by the NMPC algorithm and d is characterized by typical thermal demand trends, as reported in [18]. After performing Step 3.1, 3.2 and 3.3, the benchmark control charts are built. The control limits UCL_e and UCL_d, computed as the empirical percentile $p_{99.73}$, will be used to monitor the system throughout its functioning. The monitoring phase described in Step 4 requires data collection every monitoring period (one day). Two datasets are here collected considering two scenarios:

- ***Case 1***: $\widetilde{\mathcal{D}}^{[1]}$, obtained by increasing by 5% the AROMA DHS pipes lengths and decreasing by 5% their diameters (endogenous change). The disturbances are similar to the ones explored in the original training set;
- ***Case 2***: $\widetilde{\mathcal{D}}^{[2]}$, obtained by using thermal demands significantly different from the ones explored in the original training set (exogenous change).

Case 1: endogenous change

Step 4.1 and 4.2 are performed on $\widetilde{\mathcal{D}}^{[1]}$, obtaining the error control chart shown in Figure 3(a) and the disturbance control chart shown in Figure 3(b). Being $T^2(\widetilde{\mathbf{e}}^{[\mathbf{1}]}_{\hat{\mathbf{y}}}, \mathbf{e}_{\hat{\mathbf{y}}}) > \text{UCL}_e$ and $T^2(\widetilde{\mathbf{d}}^{[\mathbf{1}]}, \mathbf{d}) \leq \text{UCL}_d$, according to Step 4.3 and 5.1, it is possible to conclude that the type of anomaly is endogenous (as expected) and therefore a model partial update is performed. After the model partial update based on the MHE algorithm described in Step 6a (optimization time of 16 minutes), the model with updated output layer parameters outperforms the old one in a testing simulation with the modified plant and operating conditions similar to the ones explored in the original training set. In Figure 3(e) it is possible to appreciate how the updated output variable (yellow) of the furthest load from the heating station captures its corresponding actual behaviour (blue), in contrast to the output variable identified by the old model (orange). Indeed, the average FIT of the updated model is 85.5%, against 53.1% of the original one. Finally, according to Step 6a, the original control charts are discarded, and the procedure restarts from Step 2, where $C_{\mathcal{M}^{\text{up}}}$ is designed based on $\mathcal{M}^{\text{up}}$.

Case 2: exogenous change

Step 4.1 and 4.2 are performed on $\widetilde{\mathcal{D}}^{[2]}$, obtaining the error control chart shown in Figure 3(c) and the disturbance control chart shown in Figure 3(d). Being $T^2(\widetilde{\mathbf{e}}^{[\mathbf{2}]}_{\hat{\mathbf{y}}}, \mathbf{e}_{\hat{\mathbf{y}}}) > \text{UCL}_e$ and $T^2(\widetilde{\mathbf{d}}^{[\mathbf{2}]}, \mathbf{d}) > \text{UCL}_d$, according to Step 4.3, 5.1 and 5.2, it is possible to conclude that the anomaly is exogenous (as expected) and therefore a model incremental update is performed. After the model incremental update based on the additive uncertainty estimation described in Step 6b, the updated model outperforms the old one in a testing simulation with the original plant and operating conditions significantly different from the ones explored in the original training set. In Figure 3(b) it is possible to appreciate how the updated output variable (yellow) of the furthest load from the heating station captures its corresponding actual behaviour (blue), in contrast to the output variable identified by the old model alone (orange). Indeed, the average FIT of the updated model is 82.7%, against 46.6% of the original one. When standard operating conditions are restored, the incrementally updated model $\mathcal{M}^{\text{up}}$ performs just as well (FIT of 84.8%) as the original one $\mathcal{M}$ (FIT of 84.9%), avoiding catastrophic forgetting. Finally, according to Step 6b, a new control chart

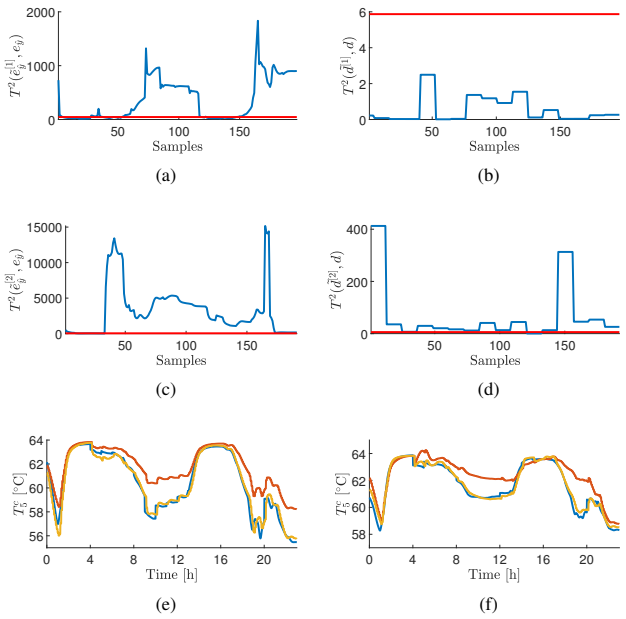

Fig. 3. (a) Case 1 error control chart: $T^2(\tilde{\mathbf{e}}_{\hat{\mathbf{y}}}^{[1]}, \mathbf{e}_{\hat{\mathbf{y}}})$ is depicted in blue, UCL_e in red. (b) Case 1 disturbance control chart: $T^2(\tilde{\mathbf{d}}^{[1]}, \mathbf{d})$ is depicted in blue, UCL_d in red. (c) Case 2 error control chart: $T^2(\tilde{\mathbf{e}}_{\hat{\mathbf{y}}}^{[2]}, \mathbf{e}_{\hat{\mathbf{y}}})$ is depicted in blue, UCL_e in red. (d) Case 2 disturbance control chart: $T^2(\tilde{\mathbf{d}}^{[2]}, \mathbf{d})$ is depicted in blue, UCL_d in red. (e) T_5^c after MHE update (yellow) compared to the measured value (blue) and the old model identification (orange). (f) T_5^c after incremental RNN update (yellow) compared to the measured value (blue) and the old model identification (orange).

both for errors and for disturbances is constructed based on the new observed data, and the original control charts are preserved. Thus, the procedure restarts from Step 2, where $C_{\mathcal{M}^{\text{up}}}$ is redesigned based on the updated model $\mathcal{M}^{\text{up}}$.

V. CONCLUSIONS

In this work, a novel monitoring and continual learning procedure for data-based dynamic models is proposed. The combined use of control charts of modelling errors and of disturbances allows to detect and distinguish the origin of an anomaly. The methodology then develops into two updating strategies, according to the anomalous scenario encountered. The developed procedure is tested on a District Heating System (DHS) benchmark referenced in the literature. The strategy allows first to distinguish between a plant modification and an operating condition shift. It is then shown how updating the customized model based on the encountered scenario allows for improved performance of the existing model in both cases, without the need to retrain the model from scratch. Future related works regard the development of an efficient methodology to monitor and adapt the controller itself, as its performance may be unrelated to the accuracy of the employed data-based model.

ACKNOWLEDGMENT

The work of Laura Boca de Giuli and Riccardo Scattolini was carried out within the MICS (Made in Italy - Circular and Sustainable) Extended Partnership and received funding from Next-Generation EU (Italian PNRR - M4 C2, Invest 1.3 - D.D. 1551.11-10-2022, PE00000004). CUP MICS D43C22003120001.

REFERENCES

[1] M. Tanaskovic, L. Fagiano, C. Novara, and M. Morari, "Data-driven control of nonlinear systems: An on-line direct approach," *Automatica*, vol. 75, pp. 1–10, 2017.

[2] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control: closed-loop guarantees and experimental results," *at-Automatisierungstechnik*, vol. 69, no. 7, pp. 608–618, 2021.

[3] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 269–296, 2020.

[4] M. R. James, "On the certainty equivalence principle and the optimal control of partially observed dynamic games," *IEEE Transactions on Automatic Control*, vol. 39, no. 11, pp. 2321–2324, 1994.

[5] Y. Zheng and Z. Wu, "Physics-informed online machine learning and predictive control of nonlinear processes with parameter uncertainty," *Industrial & Engineering Chemistry Research*, vol. 62, no. 6, pp. 2804–2818, 2023.

[6] P. Mitchell and J. Sheehy, "Comparison of predictions and observations to assess model performance: a method of empirical validation," in *Applications of Systems Approaches at the Field Level: Volume 2 Proceedings of the Second International Symposium on Systems Approaches for Agricultural Development, held at IRRI, Los Baños, Philippines, 6–8 December 1995*. Springer, 1997, pp. 437–451.

[7] Q. Li, J. Whiteley, and R. Rhinehart, "A relative performance monitor for process controllers," *International Journal of Adaptive Control and Signal Processing*, vol. 17, no. 7-9, pp. 685–708, 2003.

[8] G. M. van de Ven, T. Tuytelaars, and A. S. Tolias, "Three types of incremental learning," *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1185–1197, 2022.

[9] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural networks*, vol. 113, pp. 54–71, 2019.

[10] F. Bonassi, J. Xie, M. Farina, and R. Scattolini, "Towards lifelong learning of recurrent neural networks for control design," in *2022 European Control Conference (ECC)*. IEEE, 2022, pp. 2018–2023.

[11] K. F. Løwenstein, D. Bernardini, L. Fagiano, and A. Bemporad, "Physics-informed online learning of gray-box models by moving horizon estimation," *European Journal of Control*, p. 100861, 2023.

[12] A. Mesbah, K. P. Wabersich, A. P. Schoellig, M. N. Zeilinger, S. Lucia, T. A. Badgwell, and J. A. Paulson, "Fusion of machine learning and MPC under uncertainty: What advances are on the horizon?" in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 342–357.

[13] D. C. Montgomery, *Statistical quality control*. Wiley New York, 2009, vol. 7.

[14] A. Schirru, S. Pampuri, and G. De Nicolao, "Multilevel statistical process control of asynchronous multi-stream processes in semiconductor manufacturing," in *2010 IEEE International Conference on Automation Science and Engineering*. IEEE, 2010, pp. 57–62.

[15] R. Krug, V. Mehrmann, and M. Schmidt, "Nonlinear optimization of district heating networks," *Optimization and Engineering*, vol. 22, no. 2, pp. 783–819, 2021.

[16] L. Ljung, "System identification," in *Signal analysis and prediction*. Springer, 1998, pp. 163–173.

[17] F. Bonassi, M. Farina, J. Xie, and R. Scattolini, "On recurrent neural networks for learning-based control: recent results and ideas for future developments," *Journal of Process Control*, vol. 114, pp. 92–104, 2022.

[18] A. La Bella and A. Del Corno, "Optimal management and data-based predictive control of district heating systems: The Novate Milanese experimental case-study," *Control Engineering Practice*, vol. 132, p. 105429, 2023.

[19] L. Boca de Giuli, A. La Bella, and R. Scattolini, "Physics-informed neural network modeling and predictive control of district heating systems," *IEEE Transactions on Control Systems Technology*, 2024.