Engineering Applications of Artificial Intelligence 102 (2021) 104236

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A framework for smart control using machine-learning modeling for processes with closed-loop control in Industry 4.0

Gonen Singer [a,*], Yuval Cohen [b]

[a] Bar-Ilan University, Faculty of Engineering, Ramat-Gan, Israel
[b] Tel-Aviv Afeka Academic College of Engineering, School of Industrial Engineering and Management, Tel-Aviv, Israel

ARTICLE INFO

Keywords:
Industry 4.0
Intelligent manufacturing
Adaptive machine-learning models
Statistical process control
Closed-loop control

ABSTRACT

Anomaly detection for processes with closed-loop control has become a widespread need in Industry 4.0 shop floors. A major challenge in monitoring such processes arises from the unknown dependencies among monitored observations, as these dependencies may change dynamically and with high frequency. Motivated by these considerations, a novel framework is proposed for self-adaptive smart control using adaptive machine-learning models. On the one hand, data driven machine-learning algorithms can deal with patterns and dependencies within the data that were not necessarily known in advance. On the other hand, the recurrent self-adaptive mechanism triggers the need to switch to a new type of machine-learning model to capture and reflect new dependencies among monitored observations resulting from changes in the process. The proposed framework and the associated case study described in this paper could serve as a firm basis for implementing self-adaptive smart process control in Industry 4.0 shop-floor processes with closed-loop control.

1. Introduction

Recent improvements in sensor technology, data storage, and internet speeds, which have been integrated into the Industry 4.0 environment, together with the advent of AI, IoT, and other new technologies, have paved the way for the evolution of smart control (Anh et al., 2018; Xu et al., 2018; Liao et al., 2017; Fatorachian and Kazemi, 2018; Schroeder et al., 2019), including capabilities of self-configuration and self-adaptation (Vogel-Heuser et al., 2016; Dutt et al., 2016). These capabilities enable an embedded system to monitor its own state and behavior, and to apply changes intelligently and adaptively (Dutt et al., 2016).

In the literature, many studies have developed statistical process control (SPC) methods for monitoring and improving the quality of industrial processes. Such methods apply a control action each time there is an indication that the process is out of control. In general, the conventional SPC approach (a) assumes an independent process or an autocorrelated process that is based on a known underlying model reflecting the dependencies in the data and (b) is only useful for one-dimensional processes (e.g., Halim Lim et al., 2017; Peres and Fogliatto, 2018). Thus, these methods fail to monitor processes that can change dynamically with high frequency, and that can be generated from unknown distributions — circumstances that often arise in the Industry 4.0 environment. Recently, many studies have proposed methods for using machine-learning (ML) algorithms to identify when the process is out of control (Psarakis, 2011; Rato et al., 2016; De Ketelaere et al., 2015; Park et al., 2012; Bacher and Ben-Gal, 2017; Chou et al., 2020; Shao et al., 2017). The idea is that such methods would replace the conventional statistical process control approach. Compared to conventional SPC methods, machine-learning algorithms, which are data driven approaches, do not assume a known data distribution in advance and allow for all possible patterns and dependencies within the data. Furthermore, they can be useful for monitoring processes that have high dimensionality and can change dynamically. Such monitoring capabilities are increasingly important, not only due to advances in sensor technology but also as a result of applying a closed-loop control mechanism, which is very common in Industry 4.0 environments (Amini and Chang, 2018; Saif, 2019). In most studies, a specific machine-learning model is used, assuming a specific type of dependency between observations (such as a linear dependency of order 1), and once an 'out-of-control' event is identified, the model is updated.

Recently, in some manufacturing environments, a closed-loop control mechanism has been proposed for defect prevention (Radel et al., 2019; Liu et al., 2019), which actively compensates for process disturbances by performing adjustments. Several studies suggest combining a closed-loop control approach together with the SPC approach (Akram et al., 2012; Saif, 2019). An example of a process that might be amenable to combining these approaches is laser welding or laser cladding, where the feedback-control factors could be power, speed, or powder feed rate, and the monitoring state of the process could be

* Corresponding author.
E-mail address: Gonen.Singer@biu.ac.il (G. Singer).

https://doi.org/10.1016/j.engappai.2021.104236
Received 11 October 2020; Received in revised form 30 December 2020; Accepted 23 March 2021
Available online 5 April 2021

either the back-reflected laser radiation or the plasma plume emission (which can be measured optically); see Purtonen et al. (2014); Franciosa et al. (2019); and Mazumder (2015). A further example could be the integration of a feedback control loop during the printing of a plastic object using additive manufacturing. The printed object is made of several parts of different infill density values, which are the feedback-control factors in this problem. In order to achieve a desired objective stiffness (which is the state of the process), measurements are taken after each part is completed, and the infill density is adjusted accordingly, in a closed-loop framework, for the manufacture of future objects (Garanger et al., 2018, 2019; Hardin et al., 2019; Chua et al., 2017). However, when implementing closed-loop control for quality improvement, dependencies among monitored observations that cannot be assumed in advance are created (Ben-Gal and Singer, 2004; English et al., 2001), and thus the use of classical SPC methods is not recommended (Ben-Gal et al., 2003; Singer and Ben-Gal, 2007; Psarakis, 2011). Instead, machine-learning algorithms are required to identify the dependencies and patterns within the data (Guergachi and Boskovic, 2008). However, the autocorrelations and patterns within the monitored process can change dynamically with high frequency, which can significantly deteriorate the performance of the chosen machine-learning model type in detecting an anomaly (Kholerdi et al., 2018). This means that the chosen models can behave accurately during training, or for the monitoring period, as long as the data or patterns have not changed dramatically. In later monitoring periods, however, the test error or false-positive rate can increase due to over-fitting, and self-adaptive abilities are needed to identify when the model in use is no longer accurate, implying that an updated machine-learning algorithm should be implemented (Srivastava et al., 2014). In other words, since the type of dependency could change (for example, from a linear dependency to a non-linear dependency or from a dependency of order 1 to a dependency of higher order), updating the structure of the same model type could be insufficient in many cases. Thus, the challenges lie in developing a method to (a) track the appropriateness of the machine-learning SPC methods and (b) update the machine-learning algorithms to reliably reflect the recent and updated dependencies in the data.

The main contribution of this research is to propose a novel framework for smart anomaly detection for processes with closed-loop control that addresses the aforementioned challenges. The framework is able to identify when the process is 'out-of-control' or when the machine-learning model should be updated despite the fact that the process has not been identified as out of control. In addition, the proposed framework of using a self-adaptive smart control mechanism based on adaptive machine-learning models is illustrated by means of a simulated experiment in an Industry 4.0 environment with closed-loop control. The machine-learning models modify themselves when exposed to more data and can be updated dynamically without human intervention. Thus implementation of the framework improves both the interaction between the operator and the manufacturing environment (by decreasing overload) and the factory's production measures (Golan et al., 2019).

The remainder of the paper is organized as follows: Section 2 describes the adaptive smart process-control framework; Section 3 presents the simulation of the self-adaptive SPC; and Section 4 summarizes the contributions and draws conclusions.

2. Materials and methods

2.1. Adaptive smart process-control framework

This section describes the proposed adaptive smart process-control framework, its main components, and the relationships between these components (see Fig. 1). The proposed framework is based on the needs of monitoring complex manufacturing processes with a feedback control mechanism in an Industry 4.0 environment, as described in the Introduction. The framework is composed of three components, which work collaboratively with the aims of achieving a required quality of the process outputs and identifying failures. Components 1 and 2 in Fig. 1 are responsible for applying the closed-loop control by performing adjustments to prevent defects. Since the autocorrelation and patterns within the monitored process can change with high frequency, for example due to changes in the feedback control mechanism (as mentioned in the Introduction), component number 3 is responsible for monitoring the process outputs. This is termed the self-adaptive SPC component of the framework, and it aims to track the effects of changes in internal factors (such as a change in the controller parameters) or external factors (such as a change in humidity) on the process outputs. The three components are described in more detail in the following paragraphs.

In the *automatic process-control* (APC) *component*, feedback control is actively applied by performing adjustments to compensate automatically for process disturbances. Practically, the controller factors are adjusted according to the previously observed errors. The outputs of this component, the controller factor values, are used as inputs to the next process component, in which the observed error is calculated (see following paragraph).

The *process component*, having received the controller factor values from the APC component, assumes an expected process state and compares it with the real state generated by the process. The outputs of this component are transferred both to the APC component and to the self-adaptive SPC component. More precisely, the error between the expected state and the real state is transferred back to the APC component in order to estimate the adjustments required to compensate for the process disturbances. The quality of the generated state of the process is transferred to the next component, the self-adaptive SPC component, in order to allow for the possibility of identifying an 'out-of-control' state.

The *self-adaptive SPC component* tracks the effects of changes in the process' factors on the quality of the process outputs. In the case where an 'out-of-control' state is identified, the operator will investigate and will decide either to change the process' factors or to leave the process as it is, if the deterioration in quality is deemed to be non-significant. Regardless of the operator's decision, the machine-learning model type that is in current use is updated automatically after the 'out-of-control' indication (and after the changes to the process factors if the operator decides to interfere) to reflect the new dependencies in the observations. In the case just described, the fact that the current model type was able to identify an out-of-control state suggests that it still adequately reflects the behavior of the data. However, there may by other occasions where no out-of-control state is indicated, but the level of autocorrelation and the patterns within the monitored process have changed to such an extent that it would be preferable to switch to a different type of machine-learning model which can perform better at detecting anomalies. The idea is that, under such circumstances, a self-adaptive mechanism is triggered, which chooses from among a variety of different model types to reflect the new type of dependency that has arisen according to the most recent observations. This process happens automatically and no intervention from the operator is required. The selection of a new model type is necessary because otherwise, the performance of the machine-learning model in detecting anomalies and in identifying an out-of-control state would significantly deteriorate. It is important to appreciate that the third component works in parallel with the other components and its completion is not a condition for further progress of the process; therefore, it has no outputs to be used by other components.

Before describing in detail all the functions and flow of information within the framework, we first define the framework's parameters. Table 1 presents the list of notations.

The basic assumption of our framework is that the process to be executed has a specified target, $\mathcal{T}$. When the controller factor is calibrated, i.e., when the system sets $P_t = \overline{P}$, it is assumed that the expected state of the controlled process will result in the target, $f\left(\overline{P}\right) = S_{t,\overline{P}}^{expected} = \mathcal{T}$.

Table 1
A list of the notations used throughout this paper.

Notation	Meaning
$\mathcal{T}$	The process-specific target value for the process state
T	Total number of periods
P_t	Controller factor value at time t
$\overline{P}$	Expected (mean) controller factor value
$S^{expected}_{t,P_t}$	Expected state of the process at time t given P_t
S^{real}_{t,P_t}	Real state of the process at time t given P_t
$f(\cdot)$	Conversion of the controller factor value into the expected state of the process
k	Batch size of observations to transfer to SPC, $k \ll T$
n	Number of batches transferred to SPC during T
ξ_{t,P_t}	Error between the expected and real process states
ΔP_t	Change in the controller factor value at time t as a function of the previous errors
$h(\cdot)$	Conversion function of the state of the process into the quality of the process output o_t
v	Tolerance parameter of the conversion function defining the quality of the process output o_t
o_t	Quality of the process output relative to the target: P-positive, A-accurate, N-negative
d	Probability of deviations from the target due to temporary random changes
W_i	Weight of the error in computing ΔP_{t-1} observed at time i, where $\sum_{i=t-n}^{t-1} W_i = 1$

However, at each time t, the process state, $S^{real}_{t,\overline{P}}$, can be different, either as a result of permanent changes (whether in internal or external factors) or due to temporary random changes. To implement SPC, we transfer a batch of k observations from $N = 1$ to $\left\lfloor \frac{T}{k} \right\rfloor$, within the period T, assuming $k<<T$.

As mentioned, for any specified factor value, P_t, the expected value of the process state is $f\left(P_t\right) = S^{expected}_{t,P_t}$, while a different process state, S^{real}_{t,P_t}, may be observed in reality. The error ξ_{t,P_t}, at time t, between the real state of the process and the expected state of the process, derived from the controller factor value P_t, is defined by $\xi_{t,P_t} = S^{real}_{t,P_t} - S^{expected}_{t,P_t}$; this is transferred to the APC component. Thus, as a result of the observed error at time t, as well as previous errors, a change in the controller factor value occurs relative to the value at the calibration stage, $\Delta P_t = g(\xi_{t,P_t}, \xi_{t-1,P_{t-1}}, \ldots, \xi_{1,P_1})$, and the controller factor value at time t+1 will be $P_{t+1} = \overline{P} - \Delta P_t$.

2.2. Simulated experiment of printing a plastic object using additive manufacturing — generation of the desired objective stiffness (process state)

In this study we simulate a printing process of a plastic object using additive manufacturing with the integration of closed-loop control. The process-specific target value $\mathcal{T}$ of the process state is the desired objective stiffness. In order to achieve a desired objective stiffness, measurements of stiffness are taken after each object is produced and the infill density of the object, which is the feedback-control factor, is adjusted accordingly. Consider a non-intervened stochastic process whose state (objective stiffness) can be modeled by a probability distribution that is centered around the target, $\mathcal{T}$. The following simulation represents the procedure for generating the objective stiffness of the controlled process resulting from the automatic process control mechanism of adjusting the infill density (components 1 and 2 in Fig. 1). The observations of this simulated experiment, which are obtained by converting the state of the process into the quality of the process output, are transferred to the self-adaptive SPC (component 3 in Fig. 1), which is described in detail in Section 4.

Step 1: Generating a random sequence of data points as an input to the simulated experiment

The purpose of this step, which is an auxiliary process, is to generate random data points that will be introduced as components for calculating the measured sequence of process states (objective stiffnesses), $S^{real}_{t,P_t}, t = 1, 2, \ldots, T$ (see sub-component 2.2 in Fig. 1). Specifically, assuming calibration values for the infill density of the object (feedback-control factor) $\overline{P}$, we generate a string of T data points, $S^{real}_{t,\overline{P}}, t = 1, \ldots, T$, called the baseline process states, which represent the baseline accuracy of the objective stiffnesses compared to the target $\mathcal{T}$. The data are generated using an inverse transform method based on a discrete and symmetric distribution,

$$S^{real}_{t,\overline{P}} = \begin{cases} \mathcal{T} - 1 & \text{if } u_t \le d/2 \\ \mathcal{T} & \text{if } d/2 < u_t \le 1 - d/2 \\ \mathcal{T} + 1 & \text{if } u_t > 1 - d/2 \end{cases} \tag{1}$$

where $u_t \in [0, 1], t = 1, \ldots, T$ is a sequence of points from a uniform distribution. The parameter $0 \le d \le 1$ represents the probability of observing deviations from the target $\mathcal{T}$, as a result of temporary random changes, and $E\left(S^{real}_{t,\overline{P}}\right) = \mathcal{T}$.

Step 2: Implementing automatic process control — adjustment of the infill density of the object (sub-components 1.1 and 1.2 in Fig. 1)

In the automatic process control step, we adjust the infill density of the object (feedback-control factor) $P_t = \overline{P} - \Delta P_{t-1}, t = 1, \ldots T$, according to the n previous errors (distances of the actual object stiffness from the desired target) $\xi_{t-1,P_{t-1}}, \xi_{t-2,P_{t-2}}, \ldots, \xi_{1,P_1}$, using the following:

$$\Delta P_{t-1} = g\left(\xi_{t-1,P_{t-1}}, \xi_{t-2,P_{t-2}}, \ldots, \xi_{1,P_1}\right) = \begin{cases} 0, & t = 1, \ldots, n \\ \sum_{i=t-n}^{t-1} W_i \cdot \xi_{i,P_i}, & t = n+1, \ldots, T \end{cases} \tag{2}$$

where W_i is the weight of the error observed at time i, $\sum_{i=t-n}^{t-1} W_i = 1$.

Step 3: Calculation of the controlled process state — object stiffness values (sub-components 2.1–2.4 in Fig. 1)

Given the sequence of process states based on the calibration values of the controller factors in Step 1, $S^{real}_{t,\overline{P}}$, together with the automatic process control in step 2, the real controlled state is given by

$$S^{real}_{t,P_t} = S^{real}_{t,\overline{P}} - f\left(P_t\right). \tag{3}$$

The state of the process S^{real}_{t,P_t} is used to achieve two different goals: i) for error estimation (see sub-component 2.4 in Fig. 1), based on the difference relative to the expected value calculated in 2.1; and ii) for determination of the output symbol that indicates the quality of the process. In this simulation, we use the conversion function $h\left(S^{real}_{t,P_t}\right)$ (see sub-component 2.3 in Fig. 1) to discretize the process state values (actual object stiffness values) $S^{real}_{t,P_t}, t = 1, \ldots, T$, into the final symbol set, $o_t \in \{P, A, N\}, t = 1, \ldots, T$, representing the quality of the process. This set of symbols describes a positive deviation from the target (i.e., desired stiffness), an accurate hit on the target, or a negative deviation from the target, respectively:

$$o_t = \begin{cases} N & \text{if } S^{real}_{t,P_t} < \mathcal{T} - v \\ A & \text{if } \mathcal{T} - v \le S^{real}_{t,P_t} \le \mathcal{T} + v \\ P & \text{if } S^{real}_{t,P_t} > \mathcal{T} + v \end{cases} \tag{4}$$

where v indicates the specification of the accurate process state.

3. Results of the simulation of the self-adaptive SPC (sub-components 3.1–3.3 in Fig. 1)

The procedure proposed in the previous steps assumes a variety of factors that affect the measured real values of the process. These include internal factors, such as the APC parameters n and $W_i, i = t - n, \ldots, t-1$, and the parameter v for specification of the accurate process (which is defined by the operator), as well as external factors, such as d. In this simulation experiment, we apply changes to the factors to generate both 'in-control' and 'out-of-control' processes. A simple example of an 'in-control' process (which would be easy to identify without machine-learning algorithms) would be a process reflected by a high

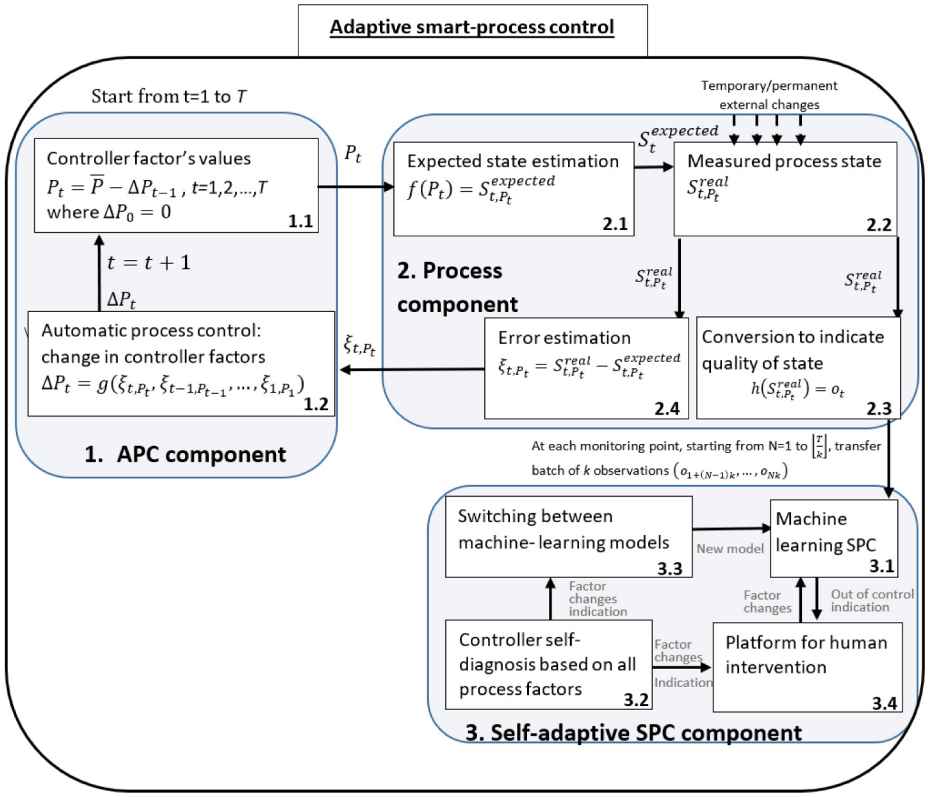

Fig. 1. The proposed adaptive smart process-control framework.

frequency of accurate 'hits' on the desired stiffness and a low frequency of deviations (randomly distributed between negative and positive) in a sequence of quality values of the process. Similarly, a simple example of an 'out-of-control' process would be a constant deviation of increased injection compared to the planned infill density, which would cause a high frequency of positive deviations from the desired stiffness. The operator receives an indication each time an out-of-control state is identified, together with information regarding the factor changes (if such information is available), which may cause him/her to decide to interfere with the process. The investigation procedure performed by the operator is beyond the scope of this paper. The machine-learning model of the current type is automatically updated each time an out-of-control indication is received. If the operator decides not to change the factors, the update is performed immediately after the out-of-control indication. Otherwise, the update occurs after the factor changes have been implemented. As explained in Section 2.1, a different type of action is taken (one that is entirely automatic) when no out-of-control indication is received but when particular factor changes are detected and the machine-learning model type that is in current use no longer performs well at detecting anomalies. A self-adaptive mechanism is proposed to switch from one model type to another model type that better reflects the new dependencies among monitored observations. The purpose of this experiment is to illustrate the capabilities of the SPC component to use self-adaptive mechanisms to either (a) identify out-of-control processes and update the existing model type or (b) transfer from one model type to another.

For the purposes of this simulation experiment, the change in the dependencies among monitored observations is effected by selecting different values for the factors. Specifically, we simulate two different scenarios in which the machine-learning based SPC should be updated based on an indication of (i) an out-of-control state and (ii) factor changes. In each of these scenarios, we generate a string of observations $o_t, t = 1, \ldots, 200,000$, along a period $T = 200,000$. After 100,000 observations, we change the process' factor values. In order to monitor the process, we transfer continuously batches of size $k = 5,000$ observations, as shown in Fig. 1, between the process component and the self-adaptive control component.

The proposed framework is designed to minimize the number of occasions on which all possible suggested model types are constructed from the data, so as to reduce the processing time and monitor the process in near-real time (which is not possible when rebuilding models with high frequency). Thus, we propose updating the models in just two different scenarios, as illustrated in the self-adaptive SPC component in Fig. 1 and as explained in Section 2.1. In the first scenario, every time an out-of-control indication is recorded, we update automatically (either analytically or numerically) the machine-learning model type that is in current use, to reflect the new dependencies in the monitored observations. The underlying assumption is that since the current model type was able to identify the out-of-control state, it adequately reflects the behavior of the data. Thus, in this scenario, we do not build new models for all types of machine-learning algorithm. In the second scenario, where there is no indication of an out-of-control state, but rather, an indication of factor changes (either a known manual change in internal factors or an indication of a change in external factors provided by a controller self-diagnosis module), all types of suggested model are updated. The machine-learning models are updated based on the data before the factor changes, and these models are then used to monitor the new process after change. In this scenario, we emphasize the importance of alerting for factor changes, as complementary information to the out-of-control indication, for triggering an update of the machine-learning model. The basic assumption of this mechanism is that the initial model that reflects the dependencies among monitored observations might not capture the new dependencies after the change, and thus new model types would need to be built.

In order to implement the APC component in our simulation, we used feedback rules similar to those presented in the *funnel experiment*, one of the popular examples presented by Deming (1986) to explain the idea of SPC. In this experiment, marbles fall through a funnel and are intended to hit a marked point on the floor (i.e., the marbles have

Table 2
Scenario 1: Modification of parameters to generate an 'out-of-control' process (sub-components 3.1 and 3.3 in Fig. 1).

Time periods	Parameter values
$o_t, t = 1, \ldots, 100,000$	Initial state of the process indicates 'in-control' process: $d, v = 0.5,\ W_{t-1} = 1,\ f\left(P_t\right) = P_t$
$o_t, t = 100,001, \ldots, 200,000$	$d, v = 0.5,\ f\left(P_t\right) = P_t$ *APC change*: $W_{t-1} = 0.5, W_{t-2} = 0.5$

Table 3
Scenario 2: Modification of parameters to create new dependencies among monitored observations without an 'out-of-control' indication (sub-components 3.1 and 3.2 in Fig. 1).

Time periods	Parameter values
$o_t, t = 1, \ldots, 100,000$	Initial state of the process indicates 'in-control' process: $d, v = 0.5,\ W_{t-1} = 0.5, W_{t-2} = 0.5, f\left(P_t\right) = P_t$
$o_t, t = 100,001, \ldots, 200,000$	$v = 0.5,\ f\left(P_t\right) = P_t$ *APC and baseline generation process change:* $d = 0.475$ $\left(W_{t-1}, W_{t-2}, W_{t-3}, W_{t-4}\right) = \begin{cases} (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}), \text{if } o_{t-1} = A \\ (\frac{1}{2}, \frac{1}{2}, 0, 0), otherwise \end{cases}$

a smaller diameter than that of the funnel opening). Before each fall of the marble, the funnel's position is adjusted following four feedback (compensation) rules, in order to correct for the distance between the previous marble fall and the marked point.

To generate data corresponding to the two aforementioned scenarios, we modified the process' factors as shown in Tables 2 and 3, respectively. The first row in Table 2 presents generation of the state process with a single weight, W_{t-1}, of the last error observed at time *t*-1, which reflects the implementation of Deming's (1986) second feedback rule. According to this rule, the position of the funnel is moved by an amount that is equal in magnitude but opposite in direction to the error observed at the last drop, indicating the initial 'in-control' process. The second row represents the second half of the process, after modification of the APC parameters. The feedback control rule reflected by the modified APC was proposed by Singer and Ben-Gal (2007) as an extension of the second feedback rule of the Deming funnel experiment, i.e., the feedback control rule uses the *two* previous error observations for compensation (reflected by two weights, W_{t-1} and W_{t-2}), instead of one. This type of modification would be identified as an out-of-control process. In the next scenario, this out-of-control process represents the in-control process in the first time period. In a similar manner, the first row in Table 3 indicates the 'in-control' process and the second row represents the second half of the process after modification of the APC parameters and the baseline generation process parameter *d*. A change in *d* usually reflects a change in external factors, which in turn alters the accuracy of the baseline process (e.g., a change in weather conditions, such as humidity or wind strength, can lead to a change in the likelihood of achieving the target value in the baseline process). Fig. 2 presents the observations of the stiffness (process state) and the changes in the infill density (feedback-control factor), for the scenarios described in Tables 2 and 3. Each graph presents observations before (observations 1–50) and after (observations 51–100) the modification of the parameters. It can be seen that, for both scenarios, the patterns within the monitored process have changed after modification of the parameters. Specifically, in scenario 1, the stiffness exhibits larger extreme values (both positive and negative) compared to the values before the change. In scenario 2, the stiffness exhibits larger negative extreme values and lower positive extreme values compared to the values before the change.

In our simulated experiment, in order to illustrate the implementation of the proposed framework, we used only two types of model: a Markov model of order 1, as suggested by Singer and Ben-Gal (2007) for SPC implementation, and a context-tree model, as suggested by Ben-Gal and Singer (2004) for production monitoring. We chose a Markov model because it is a simple statistical model that can be built analytically and can reflect the dependencies of the initial in-control process in a simple way. The context tree was chosen as a simple interpretable machine-learning model (very similar in its structure to other decision-tree models), which can numerically capture and reflect the dependencies within the data. Using these two model types allows us to highlight the phenomenon whereby one model may not be able to capture specific types of dependency, while another can, thus illustrating the importance of updating the model type. The proposed framework, of course, suggests taking into account several varieties of model (i.e., more than two) with different assumptions regarding dependency types. A description of the investigative procedure, and of the corresponding changes required to achieve a desired process state after an out-of-control identification, is beyond the scope of this paper. The following steps describe the implementation of the self-adaptive component in Fig. 1, for the scenarios in Tables 2 and 3.

Scenario 1 - Step 1. Estimation of the reference Markov model of the initial in-control process. The first generated string, consisting of 100,000 observations, represents the initial in-control process, with target $\mathcal{T} = 0$ and initial factor values as presented in the first row of Table 2. Since the procedure for generating the process is known, as are the applied feedback control mechanism and its parameters, we build a reference analytical Markov model, M_0, to reflect the in-control process, which might be more accurate for monitoring purposes than models that are constructed numerically from the data itself. Table 4 exemplifies the generation of 10 data points according to the procedure described above.

The applied feedback control (changes in the infill density) introduces dependencies between consecutive process states, and the resulting process can be modeled by a Markov chain. The Markov model is derived mathematically according to a three-stage procedure, as described in Appendix. Fig. 3 presents the Markov model obtained for the in-control process. The model parameters are the transition probabilities $P\left(o_{t+1}|o_t\right)$ between consecutive outputs $o_t, o_{t+1} \in \{N, A, P\}$, which are obtained by maximum-likelihood estimation methods.

Scenario 1 – Step 2. Monitor the process based on the reference analytical Markov model. In this step we monitor the process, similarly to Singer and Ben-Gal (2007), such that at each monitoring point, we construct numerically a Markov model $\hat{M}_N(o_t)$, with order 1, and try to identify out-of-control behavior based on a comparison with the reference analytical model, M_0. Specifically, we collect a sequence of $k = 5{,}000$ observations generated from the process at any monitoring point $N =$ 1 to 40. At each monitoring point, a Markov model of order 1 is constructed based on the maximum-likelihood estimation (MLE) of the relative transition probabilities, which are calculated from the observed frequencies of the output symbols. At each monitoring point, we use the chi-square statistic to compare the observed frequencies $\hat{n}(o_t|o_{t+1})$ from the Markov model $\hat{M}_N(o_t)$ with the expected frequencies $n(o_t|o_{t+1})$ from the reference analytical model M_0. Under the null hypothesis that the same underlying Markov model generated the data in the different time periods, the chi-square statistic is approximately chi-square distributed, i.e.,

$$0 \leq \sum_{o \in (N,A,P)} \frac{\left(\hat{n}\left(o_t|o_{t+1}\right) - n(o_t|o_{t+1})\right)^2}{n(o_t|o_{t+1})} \leq \chi^2_{b(b-1),\alpha} \tag{5}$$

where *b* represents the size of the symbol set (which is equal to 3) and α is the Type-I error value, which in our case is defined to be equal to 0.99. In our simulated experiment, we define the upper control limit (UCL) to be equal to $\chi^2_{6,0.99} = 16.81$. Figs. 4 and 5 present the SPC graphs with the 40 chi-square statistic values relative to the UCL. The 20 values in Fig. 4 represent the in-control process from the first time period, $o_t, t = 1, \ldots, 100,000$ in Table 2, while Fig. 5 presents the 20 values obtained from monitoring the process in the second time period,

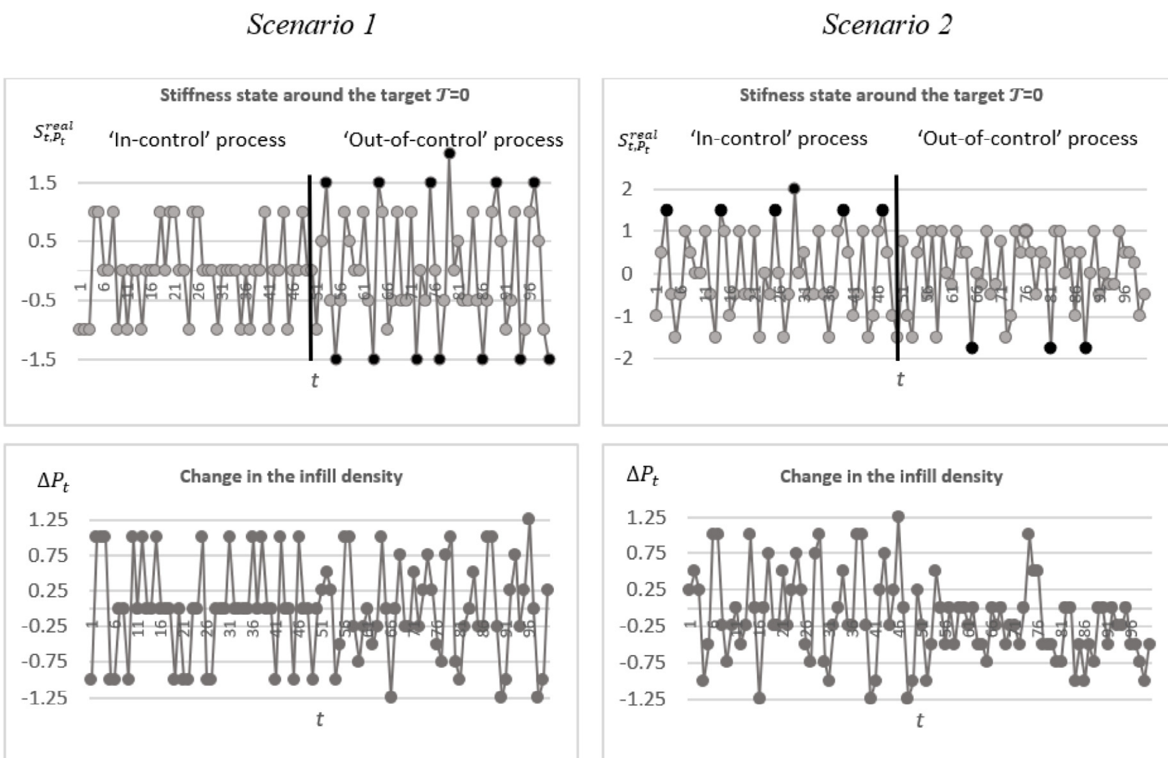

Fig. 2. Samples of observations of the stiffness state and the change in infill density before and after modification of the parameters for scenario 1 and scenario 2.

Table 4
Numerical generation of 10 'in-control' data points with automatic control.

Index	Uniformly random: $u_t \in [0,1]$	Inverse transform: $S^{real}_{t,\overline{P}}$ with $d = 0.5$	APC: $S^{real}_{t,P_t} = S^{real}_{t,\overline{P}}$-$\xi_{t-1,P_{t-1}}$	Discretization: x_t with $\upsilon = 0.5$
1	0.040	−1	−1	N
2	0.442	0	1	P
3	0.792	1	1	P
4	0.391	0	−1	N
5	0.933	1	1	P
6	0.264	0	−1	N
7	0.390	0	0	A
8	0.632	0	0	A
9	0.551	0	0	A
10	0.442	0	0	A

$o_t, t = 100,001,\ldots,200,000$ in Table 2, after the change in the input parameters.

As can be seen from the above figures, the Markov model identifies the out-of-control behavior resulting from the factor changes. Thus, we use the same model type, i.e., a Markov model of order 1, for the new process and feedback control parameters, to capture the new dependencies created in the observed data, as explained above. In the next scenario, this out-of-control process represents the in-control process in the first time period.

Scenario 2 – Step 1. Estimation of the reference Markov model of the initial in-control process. The first generated string of 100,000 observations in this scenario indicates the initial in-control process, with target $\mathcal{T} = 0$, similar to step 1 of scenario 1. The feedback control rule uses two observations for compensation instead of one. Similar to the first scenario, we aim to find a reference analytical Markov model M_0 to reflect the in-control process based on a similar procedure to that explained above, which yields the Markov model presented in Fig. 6.

Scenario 2 – Step 2. Monitor the process based on the reference analytical Markov model. We monitor the process in a similar way to that described in the second step of scenario 1. Since we use the same type of Markov model with order 1, the chi-square statistic and calculated UCL value are the same as in the previous scenario. Fig. 7 presents the SPC graph with the 40 chi-square statistic values, relative to the UCL, before and after the modification of the factor values, along the period of time $o_t, t = 1,\ldots,200,000$.

Scenario 2 - Step 3. Update other model types for monitoring the process. As can be seen from the previous step, since there was no out-of-control indication after modification of the factor values, according to the framework, we have to build models based on all other model types to verify that the process or dependencies between observations have not changed. For the purposes of this simulation experiment, we assume that one other model type, called context tree (see Ben-Gal and Singer, 2004), is available. We build a reference context tree numerically, based on the observed data before the change, to represent the in-control process. Fig. 8 shows the initial reference context tree (for explanations regarding tree growing and probability estimation, see Ben-Gal and Singer, 2004). The context tree is obtained by applying the context algorithm to the initial 5,000 observations generated from the in-control process. The structure of the tree in Fig. 8 represents the varying-length dependencies among the observations. As can be seen, the empirical conditional distributions of the symbols, represented by each node in the tree, are significantly different from each other. The tree has $S = 6$ optimal contexts (i.e., sequences of symbols that

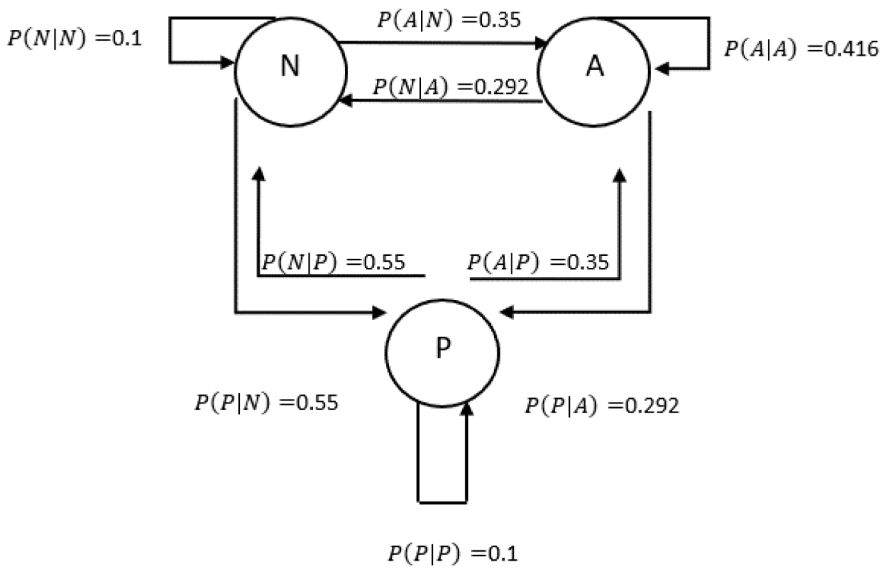

Fig. 3. The Markov chain representing the 'in-control' process of scenario 1.

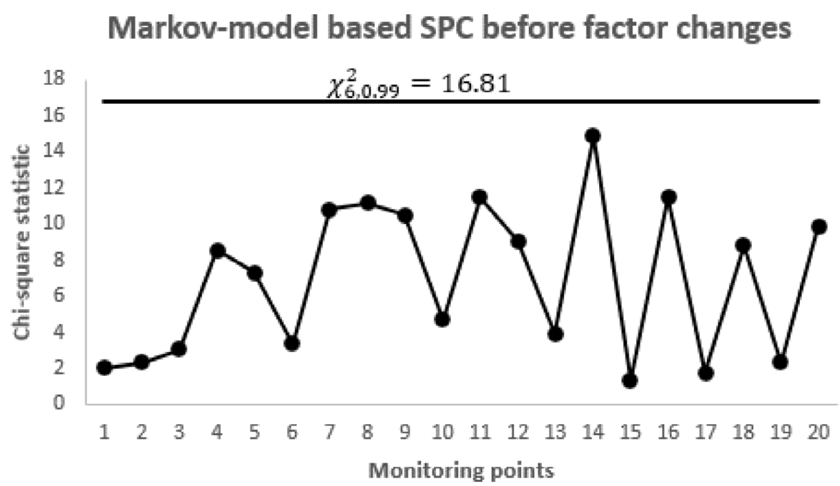

Fig. 4. SPC for the 'in-control' process based on the reference Markov model.

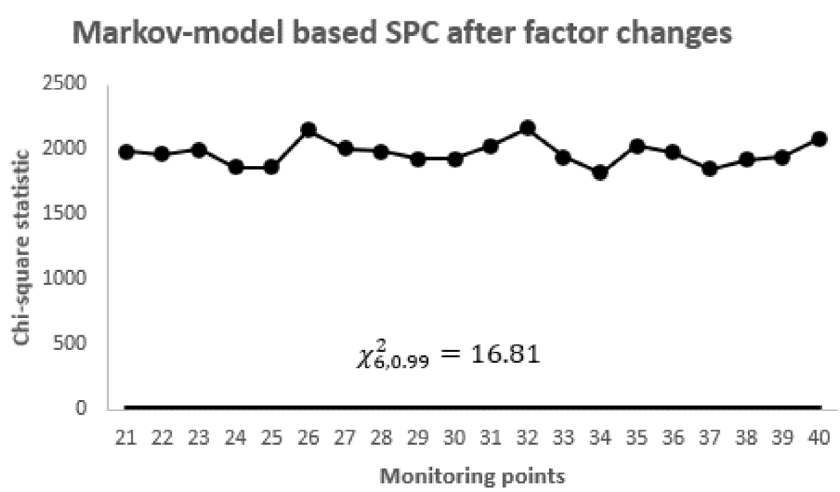

Fig. 5. SPC for the 'out-of-control' process based on the reference Markov model.

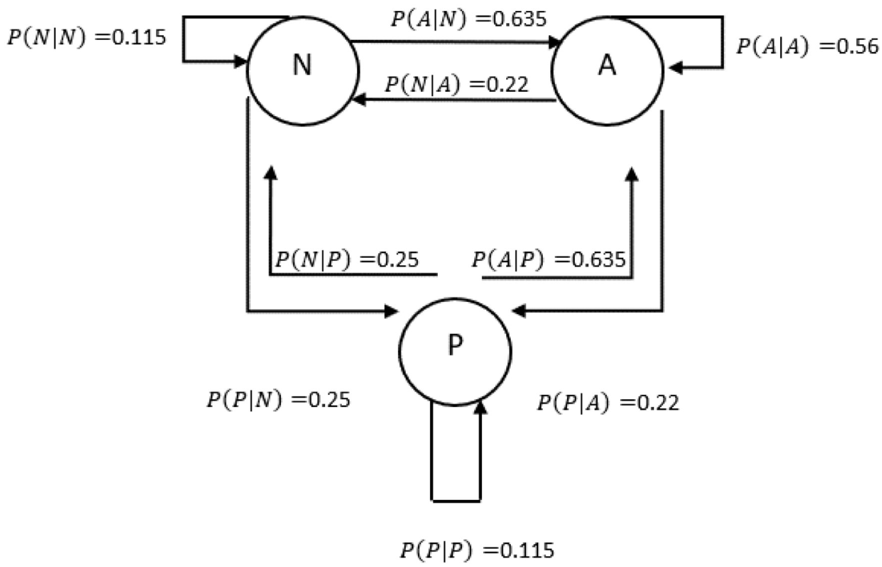

Fig. 6. The Markov chain representing the initial process of scenario 2.

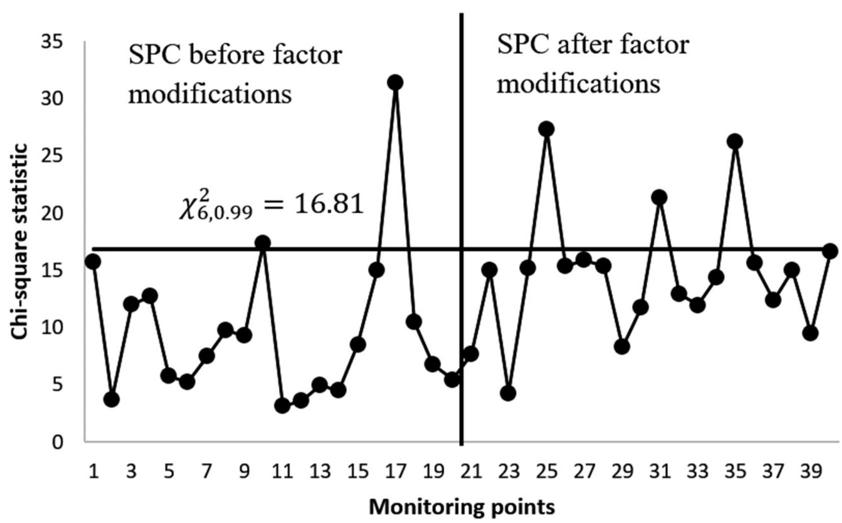

Fig. 7. Markov-based SPC before and after changes in the factor values.

influence the probability of each symbol), {N,A,P,AA,AAN,AAP}. Every node in the context tree (except for the root node) represents an optimal context and contains the empirical conditional distribution of the symbols given the optimal context (for further explanation of optimal contexts and context tree structure, see Ben-Gal and Singer, 2004). For example, the empirical conditional distribution of symbols given the optimal context AAN is $P_0(N, A, P|s = AAN)$ =(0.097,0.692,0.211). In comparison with the Markov model of order 1 explained above, the context tree captures more complex dependencies with different orders (up to order 3).

In order to monitor the process before and after the factor value modifications, at each monitoring point a context tree is constructed with the same structure as the reference context tree, based on the maximum-likelihood estimation (MLE) of the relative transition probabilities, which are calculated from the observed frequencies of the symbols. At each monitoring point, we use the Kullback Leibler (KL) statistic to compare the observed context-tree model $\hat{P}_N(o, S)$ with the reference context-tree model $P_0(o, S)$. Under the null hypothesis that the same underlying context tree generated the data during different periods, Ben-Gal and Singer (2004) showed that the KL statistic multiplied by $2k$ (where $k = 5{,}000$) is approximately chi-square distributed, i.e.,

$$0 \leq 2k \cdot KL\left(P_0(o, S), \hat{P}_N(o, S)\right) \leq \chi^2_{2(Sb-1),\alpha} \tag{6}$$

where b represents the size of the symbol set (which is equal to 3), S is the number of optimal contexts (which is equal to 6), and α is the Type-I error value, which in our case is defined to be 0.99. In our simulated experiment, we define the upper control limit (UCL) to be equal to $\chi^2_{34,0.99} = 56.06$. Fig. 9 presents the SPC graph with the $2k \cdot KL$ statistic values, relative to the UCL, before and after the factor value modifications, along the period of time $o_t, t = 1, \ldots, 200{,}000$. Note that all of the $2k \cdot KL$ statistic values in the first 20 samples (before the factor value modifications) fall below the upper control limit. The $2k \cdot KL$ statistic values after the factor value modifications are identified as generated from an out-of-control process.

In summary, note that the proposed smart adaptive control framework performs well with respect to both types of statistical error: 100% of the in-control monitoring points adhere to the control limit rule, while 85% of the out-of-control monitoring points are identified as having been generated by a modified process. The change point is clearly evident by a sharp jump in the $2k \cdot \mathrm{KL}$ values from $t = 100{,}001$ (monitoring point $N = 21$). Since the previous Markov model did not

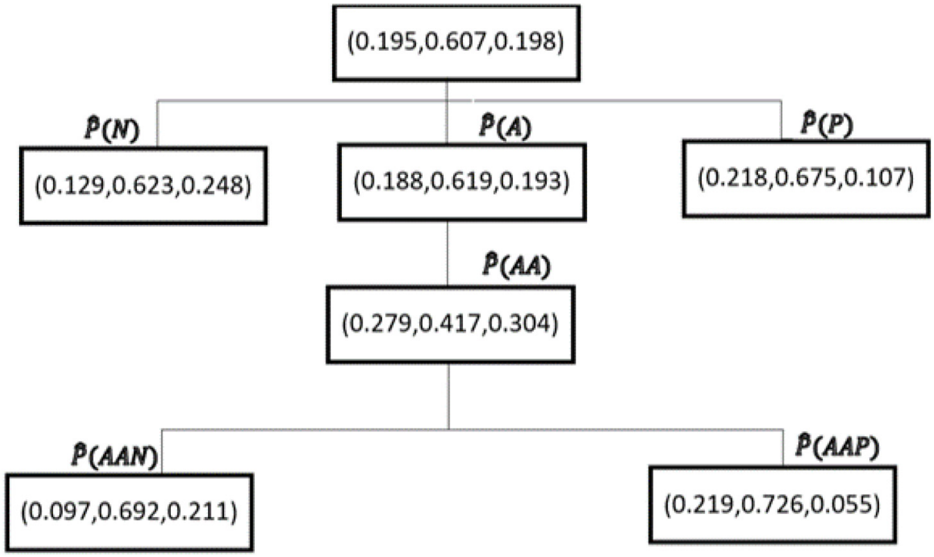

Fig. 8. The initial reference context tree.

capture dependencies of order higher than 1, it failed to identify the change and the out-of-control behavior, and thus a new model type was needed.

4. Discussion

This study proposes a framework for self-adaptive smart control based on adaptive machine-learning models that can cover all possible patterns and dependencies within data. These models are updated dynamically to capture the new dependencies created among monitored observations resulting from high-frequency changes in the process. The framework introduces a component of closed-loop control (called automatic process control), which actively compensates for process disturbances by performing adjustments. This could be an important tool in many Industry 4.0 environments, such as the process of laser welding, where the feedback-control factors could be the power, speed, or powder feed rate, or in the additive manufacturing of plastic objects, where the feedback-control factor could be the infill density. In this study, we used a simulated experiment to demonstrate how the proposed framework of a self-adaptive smart controller has the potential to address the challenges of monitoring Industry 4.0 manufacturing processes in which (a) data change dynamically and with high frequency and (b) the dependencies created among monitored observations cannot be assumed in advance.

Specifically, in the simulated experiment, by changing the parameters of the closed-loop control and other factors of the process, the dependencies among monitored observations were updated. We presented two different scenarios that triggered an update of the machine-learning algorithm. In the first scenario, identification of an out-of-control process led to a model update of the same model type, while in the second scenario, where there was a factor change indication without an out-of-control identification, all types of machine-learning model considered by the system were updated. These two scenarios demonstrate the capabilities of the self-adaptive mechanism in which the machine-learning based SPC method is tracked and an adaptive machine-learning mechanism is applied automatically to identify and reliably reflect 'new' dependencies within the data. This approach stands in contrast with classical SPC methods, which assume a particular data distribution in advance.

The main contributions of our research are as follows:

(1) Methodology. We proposed a novel framework for smart anomaly detection for processes with closed-loop control, which has become a widespread need in Industry 4.0 shop floors.
(2) Modeling. We developed a dynamic modeling approach, which is essential for capturing new dependencies created among monitored observations resulting from high-frequency changes in the process. The proposed approach consists of a self-adaptive smart control mechanism based on an adaptive machine-learning mechanism.
(3) Practice. We implemented a simulation experiment to illustrate the use of the self-adaptive smart control mechanism. These results of this experiment show promise that the proposed framework could serve as a firm basis for implementing smart process-control in an Industry 4.0 manufacturing environment with closed-loop control.

Future research could present an implementation of the framework on real data from manufacturing processes with a feedback-control mechanism. Other fruitful research directions could include the development of additional methods to enrich the self-adaptive smart control framework, such as introducing a diagnostic component that implements root-cause analysis based on machine-learning algorithms for identification of the factor changes and estimation of their effect.

CRediT authorship contribution statement

Gonen Singer: Conceptualization, Methodology, Data curation, Investigation, Writing - original draft, Writing - review & editing. **Yuval Cohen:** Conceptualization, Methodology, Data curation, Investigation, Writing - original draft, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

In this appendix, we present the three-stage procedure to construct the analytical Markov model obtained for the in-control process in scenario 1.

Stage 1: We obtain the stationary probabilities of all possible states assuming calibration factor values $\overline{P}$. These probabilities are derived from transition probabilities between tuples of two consecutive process states $\left(S^{real}_{t-1,\overline{P}}, S^{real}_{t,\overline{P}}\right) \rightarrow \left(S^{real}_{t,\overline{P}}, S^{real}_{t+1,\overline{P}}\right)$. Each pair of consecutive states

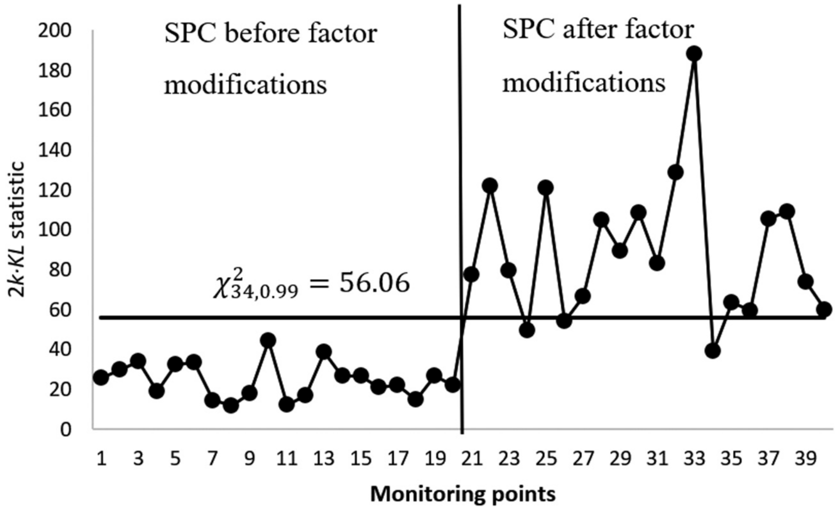

Fig. 9. Context-tree based SPC before and after modification of the factor values.

has $3^2 = 9$ different scenarios, which are denoted, respectively, by $s_i, i = 1, \dots, 9$, where $s_1 = (-1,-1)$, $s_2 = (-1,0)$, $s_3 = (-1,1)$, $s_4 = (0,-1)$, $s_5 = (0,0)$, $s_6 = (0,1)$, $s_7 = (1,-1)$, $s_8 = (1,0)$, and $s_9 = (1,1)$. The transition probability equations are as follows:

$$\begin{aligned} &0.25P(s_i) + 0.25P(s_{i+3}) + 0.25P(s_{i+6}) = P(s_{3i-2})\ i = 1,2,3 \\ &0.5P(s_i) + 0.5P(s_{i+3}) + 0.5P(s_{i+6}) = P(s_{3i-1})\ i = 1,2,3 \\ &0.5P(s_i) + 0.5P(s_{i+3}) + 0.5P(s_{i+6}) = P(s_{3i})\ i = 1,2,3 \end{aligned} \tag{A.1}$$

where $P(s_i)$ denotes the stationary probability of scenario s_i. The transition probabilities in (A.1) were calculated from the symmetric distribution in Eq. (1) and the parameters of the in-control process in scenario 1. For example, assuming $i = 1$, we can transfer from scenarios s_1, s_4, or s_7 to scenario s_1 with a transition probability of 0.25, as can be seen from the first row in (A.1). Solving the set of transition probabilities presented in (A.1) yields the following stationary probabilities: $P(s_1) = P(s_3) = P(s_7) = P(s_9) = \frac{1}{16}$; $P(s_4) = P(s_6) = P(s_2) = P(s_8) = \frac{1}{8}$; $P(s_5) = \frac{1}{4}$.

Stage 2: We calculate the steady-state probabilities of the discretized measure of the quality of the process (i.e., the deviation of the object stiffness value from the desired target), $P(o_t = P)$, $P(o_t = A)$ and $P(o_t = N)$, by summing up the appropriate probabilities as follows:

$$\begin{aligned} &P(A) = P(s_1) + P(s_5) + P(s_9) = 0.375 \\ &P(N) = P(s_4) + P(s_7) + P(s_8) = 0.3125 \\ &P(P) = P(s_2) + P(s_3) + P(s_6) = 0.3125 \end{aligned} \tag{A.2}$$

The first row in (A.2), for example, presents all scenarios (s_1, s_5, s_9) that reflect an 'accurate' output according to Eq. (4).

Stage 3: We calculate the transition probabilities between two consecutives outputs $o_t \to o_{t+1}$. For example, the probability of observing two sequential 'negative' outputs is calculated as follows:

$$\begin{aligned} &P(o_{t+1} = N | o_t = N) \\ &= \frac{P(s_4) \cdot P(N|s_4)}{P(N)} + \frac{P(s_7) \cdot P(N|s_7)}{P(N)} + \frac{P(s_8) \cdot P(N|s_8)}{P(N)} = \end{aligned} \tag{A.3}$$

$$P(o_{t+1} = N | o_t = N) = \frac{1/8 \cdot 0}{0.3125} + \frac{1/16 \cdot 0}{0.3125} + \frac{1/8 \cdot 0.25}{0.3125} = 0.1$$

The Markov model obtained for the in-control process in scenario 1 based on this procedure is presented in Fig. 3.

References

Akram, M.A., Saif, A.W.A., Rahim, M.A., 2012. Quality monitoring and process adjustment by integrating SPC and APC: a review. Int. J. Ind. Syst. Eng. 11 (4), 375–405.

Amini, M., Chang, S., 2018. A review of machine learning approaches for high dimensional process monitoring. In: Paper presented at the 2018 Industrial and Systems Engineering Research Conference, Orlando, FL, May. pp. 19–22.

Anh, D.T., Dąbrowski, K., Skrzypek, K., 2018. The predictive maintenance concept in the maintenance department of the 'Industry 4.0' production enterprise. Found. Manag. 10 (1), 283–292.

Bacher, M., Ben-Gal, I., 2017. Ensemble-Bayesian SPC: Multi-mode process monitoring for novelty detection. IISE Trans. 49 (11), 1014–1030.

Ben-Gal, I., Morag, G., Shmilovici, A., 2003. CSPC: A monitoring procedure for state dependent processes. Technometrics 45, 293–311.

Ben-Gal, I., Singer, G., 2004. Statistical process control via context modeling of finite-state processes: an application to production monitoring. IIE Trans. 36 (5), 401–415.

Chou, S.H., Chang, S., Tsai, T.R., Lin, D.K., Xia, Y., Lin, Y.S., 2020. Implementation of statistical process control framework with machine learning on waveform profiles with no gold standard reference. Comput. Ind. Eng. 106325.

Chua, Z.Y., Ahn, I.H., Moon, S.K., 2017. Process monitoring and inspection systems in metal additive manufacturing: Status and applications. Int. J. Precis. Eng. Manuf.-Green Technol. 4 (2), 235–245.

De Ketelaere, B., Hubert, M., Schmitt, E., 2015. Overview of PCA-based statistical process monitoring methods for time-dependent, high-dimensional data. J. Qual. Technol. 47, 318–335.

Deming, W.E., 1986. Out of the crisis. In: MIT Center for Advanced Engineering Study. Cambridge, MA, pp. 327–331.

Dutt, N., Jantsch, A., Sarma, S., 2016. Toward smart embedded systems: A self-aware system-on-chip (soc) perspective. ACM Trans. Embed. Comput. Syst. 15 (2), 1–22.

English, J.R., Martin, T., Yaz, E., Elsayed, E., 2001. Change point detection and control using statistical process control and automatic process control. In: IIE Annual Conference, Dallas, TX.

Fatorachian, H., Kazemi, H., 2018. A critical investigation of Industry 4.0 in manufacturing: theoretical operationalisation framework. Prod. Plan. Control 29 (8), 633–644.

Franciosa, P., Serino, A., Botros, R.A., Ceglarek, D., 2019. Closed-loop gap bridging control for remote laser welding of aluminum components based on first principle energy and mass balance. J. Laser Appl. 31 (2), 022416.

Garanger, K., Khamvilai, T., Feron, E., 2018. 3D printing of a leaf spring: A demonstration of closed-loop control in additive manufacturing. In: Paper presented at the 2018 IEEE Conference on Control Technology and Applications (CCTA), Copenhagen, Denmark, August 21–24.

Garanger, K., Khamvilai, T., Feron, E., 2019. Additive manufacturing for high precision structural properties via feedback control. arXiv preprint arXiv:1905.10411.

Golan, M., Cohen, Y., Singer, G., 2019. A framework for operator–workstation interaction in Industry 4.0. Int. J. Prod. Res. 1–12.

Guergachi, A., Boskovic, G., 2008. System models or learning machines? Appl. Math. Comput. 204, 553–567.

Halim Lim, S.A., Antony, J., Arshed, N., Albliwi, S., 2017. A systematic review of statistical process control implementation in the food manufacturing industry. Total Qual. Manag. Bus. Excell. 28 (1–2), 176–189.

Hardin, J.O., Grabowski, C.A., Lucas, M., Durstock, M.F., Berrigan, J.D., 2019. All-printed multilayer high voltage capacitors with integrated processing feedback. Addit. Manuf. 27, 327–333.

Kholerdi, H.A., TaheriNejad, N., Jantsch, A., 2018. Enhancement of classification of small data sets using self-adaptive - an iris flower case-study. In: Paper presented at the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, May 27–30.

Liao, Y., Deschamps, F., Loures, E.D.F.R., Ramos, L.F.P., 2017. Past, present and future of Industry 4.0-a systematic literature review and research agenda proposal. Int. J. Prod. Res. 55 (12), 3609–3629.

Liu, C., Law, A.C.C., Roberson, D., Kong, Z.J., 2019. Image analysis-based closed loop quality control for additive manufacturing with fused filament fabrication. J. Manuf. Syst. 51, 75–86.

Mazumder, J., 2015. Design for metallic additive manufacturing machine with capability for 'Certify as You Build'. Procedia CIRP 36, 187–192.

Park, M., Kim, J., Jeong, M.K., Hamouda, A.M.S., Al-Khalifa, K.N., Elsayed, E.A., 2012. Economic cost models of integrated APC controlled SPC charts. Int. J. Prod. Res. 50 (14), 3936–3955.

Peres, F.A.P., Fogliatto, F.S., 2018. Variable selection methods in multivariate statistical process control: A systematic literature review. Comput. Ind. Eng. 115, 603–619.

Psarakis, S., 2011. The use of neural networks in statistical process control charts. Qual. Reliab. Eng. Int. 27 (5), 641–650.

Purtonen, T., Kalliosaari, A., Salminen, A., 2014. Monitoring and adaptive control of laser processes. Physics Procedia 56, 1218–1231.

Radel, S., Diourte, A., Soulié, F., Company, O., Bordreuil, C., 2019. Skeleton arc additive manufacturing with closed loop control. Addit. Manuf. 26, 106–116.

Rato, T.J., Schmitt, E., Reis, M., De Ketelaere, B., Hubert, M., 2016. A systematic comparison of PCA-based statistical process monitoring methods for high-dimensional, time-dependent processes. AIChE J. 62 (5), 1478–1493.

Saif, A.W., 2019. A framework for the integration of statistical process control and engineering process control. In: 2019 Industrial & Systems Engineering Conference (ISEC). IEEE, pp. 1–4.

Schroeder, A., Ziaee Bigdeli, A., Galera Zarco, C., Baines, T., 2019. Capturing the benefits of industry 4.0: a business network perspective. Prod. Plan. Control 30 (16), 1305–1321.

Shao, Y.E., Chang, P.Y., Lu, C.J., 2017. Applying two-stage neural network based classifiers to the identification of mixture control chart patterns for an SPC-EPC process. Complexity.

Singer, G., Ben-Gal, I., 2007. The funnel experiment: The Markov-based SPC approach. Qual. Reliab. Eng. Int. 23 (8), 899–913.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.

Vogel-Heuser, B., Rösch, S., Fischer, J., Simon, T., Ulewicz, S., Folmer, J., 2016. Fault handling in PLC-based industry 4.0 automated production systems as a basis for restart and self-configuration and its evaluation. J. Softw. Eng. Appl. 9 (01), 1.

Xu, L.D., Xu, E.L., Li, L., 2018. Industry 4.0: state of the art and future trends. Int. J. Prod. Res. 56 (8), 2941–2962.