Evaluation of Drift Detection Algorithms in the Condition Monitoring Domain

Alireza Estaji, Maximilian Götzinger, Benedikt Tutzer, Stefan Kollmann,
Thilo Sauter, *Fellow, IEEE*, and Axel Jantsch, *Fellow, IEEE*

***Abstract*—In condition monitoring, early detection of process signal drifts indicating, e.g., equipment degradation is crucial. exponentially weighted moving average (EWMA), cumulative sum (CUSUM), and discrete average block (DAB)-based drift detectors are statistical and commonly used methods. Each has benefits and limitations, suited to different data types. However, EWMA and CUSUM are fixed mean drift detectors, limiting their applicability and adaptability. This article explores adding dynamic behavior to drift detection methods. We use a wide range of synthetic data based on a real-world manufacturing process. The investigated parameter space includes standard deviation, drift rates, and outliers. Besides, each algorithm has some tuning parameters that define its behavior. Two metrics validate experiments against labeled data. Based on our observations, EWMA performs better for drift detection on average, but CUSUM is superior in detecting very small drifts. Furthermore, we derive guidelines for the choice and application of drift detection in practice.**

***Index Terms*—Algorithm tuning and evaluation, condition monitoring, cumulative sum (CUSUM), drift detection, exponentially weighted moving average (EWMA).**

I. Introduction

THE fourth industrial revolution is in full swing, and the digital transformation of industrial manufacturing needs to address the digitization of production, automation, and automated data exchange. In order to address those challenges, manufacturing systems can use advanced condition monitoring methods based on modern technologies, such as industrial Internet of Things, edge and cloud computing, or data analytics.

Modern machines usually integrate complex monitoring systems. However, many conventional types of machinery are not equipped with such facilities. For these types of machines, model-free machine health and condition monitoring systems are interesting as a real time, data-driven approach that enables predictive maintenance (PM). Although an integrated monitoring system is generally favorable because it has direct access to the process variables, it is usually associated with significantly increased engineering efforts and costs. The apparent advantage of a model-free machine health monitoring system is that no deep knowledge of the monitored system is required. A generic monitoring system detects system states and classifies a system's health condition based on observed parameters. Besides state detection and malfunction discovery, an essential constituent of process monitoring is the detection of drifts in the process signals [1].

In general, *drift* means that statistical properties of the observed data are changing slowly over time. This could be the mean value, the standard deviation, parameters of underlying distributions, characteristic frequencies, to name but a few. In the context of this article, drift specifically refers to a gradual change in the mean value or standard deviation of numerical input data obtained from an industrial process. A typical example is the gradual wear-out of a motor that leads to a slow speed deviation from its nominal value.

In a wider sense, condition monitoring and the associated methods are not only applicable to machines and industrial equipment, but also to monitoring the quality of a process and the goods produced [2], in the form of in-process monitoring or classical end-of-line tests, which enable real-time decision support. The requirements of smart manufacturing, automated process monitoring, and PM are motivations to find more general approaches to process and monitor a wide range of input data. At any rate, a focus point of such monitoring approaches is to detect anomalies and drifts.

There are multiple ways to implement drift detection, such as discrete average block (DAB)-based methods [1]. Two classical approaches with more than half a century of history are the cumulative sum (CUSUM) and exponentially weighted moving average (EWMA) algorithms [3]. Simplicity of implementation while maintaining acceptable results makes these two well-established algorithms still popular and in active use. EWMA

Received 18 October 2023; revised 25 April 2024; accepted 30 July 2024. This work was supported in part by the Klima- und Energiefonds and carried out under the Energy Research Program 2020, in part by the Klima- und Energiefonds, and in part by the FFG (Austrian Research Promotion Agency) for the KI4HVACs project within the programme "Energieforschung (e!MISSION)." Paper no. TII-23-4061. *(Corresponding author: Alireza Estaji.)*

Alireza Estaji and Stefan Kollmann are with the Department for Integrated Sensor Systems, Danube University Krems, 2700 Wiener Neustadt, Austria (e-mail: alireza.estaji@donau-uni.ac.at; stefan.kollmann@donau-uni.ac.at).

Maximilian Götzinger, Benedikt Tutzer, and Axel Jantsch are with the Institute of Computer Technology, Vienna University of Technology, 1040 Vienna, Austria (e-mail: maximilian.goetzinger@tuwien.ac.at; benedikt.tutzer@tuwien.ac.at; axel.jantsch@tuwien.ac.at).

Thilo Sauter is with the Department for Integrated Sensor Systems, Danube University Krems, 2700 Wiener Neustadt, Austria, and also with the Institute of Computer Technology, Vienna University of Technology, 1040 Vienna, Austria (e-mail: thilo.sauter@donau-uni.ac.at).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2024.3452208.

Digital Object Identifier 10.1109/TII.2024.3452208

has been found to perform better for small drift rates [4], which is interesting especially in the context of condition monitoring and the detection of stealthy process degradation. Similarly, CUSUM is effective in detecting small drifts. Moreover, both approaches are independent of window size, which is also a desired property.

The essential deficiency of the original algorithms is that they have no provisions for dynamically updating the mean value or so-called "target value." The primary assumption is that the nominal mean and standard deviation values are constant. EWMA was later extended to cases where the mean or standard deviation may not be constant. However, this still assumes that each random variable follows a normal distribution, which is a serious limitation that is not satisfied in many applications [5].

The purpose of this article is to revisit the classical drift detectors EWMA and CUSUM and enhance their drift detection capabilities. Specifically, we investigate ways to make them adaptive to changing mean values of the data series without imposing assumptions on statistical properties. This is also an essential prerequisite for being application-agnostic and model-free. Therefore, the investigated methods are purely data driven and work solely on measured numerical data.

Our experiment dataset covers a wide range of drifts, from almost undetectable to clearly detectable drifts alongside varying standard deviations and different levels of outliers.

The rest of this article is organized as follows. After a short review of the existing literature in Section II, selected approaches are explained in Section III. Experiment setups and obtained results are presented in Sections IV and V, and we summarize the applied steps and generalize them in a practicable guidelines in Section VI. Finally, Section VII concludes this article.

II. Background and Related Work

In the literature, many proposed drift detection methods are based on the statistical properties of the input streams [3]. Barros et al.[6] presented a drift detector comparison for many well-known methods. Eight different drift detectors with different artificial datasets, containing abrupt and gradual concept drift, with several drift rates and in the presence or absence of noise compared by Gonçalves Jr. et al. [7]. Jaramillo-Valbuena et al. [8] compared four different drift detection methods to determine their robustness for detecting drift from data streams in the presence of noise.

Yi et al. [9] suggested an adaptive CUSUM chart based on a generalized likelihood ratio statistic and estimation of a shift size. A detailed explanation and a "Head-to-Head" comparison of CUSUM and EWMA is presented by Hawkins et al. [10]. Götzinger et al. [1] developed the confidence-based context-aware condition monitoring (CCAM), a generic monitoring system that provides state detection and drift detection. Zwetsloot and Woodall [2] have reviewed the literature on multivariate time-between-events monitoring. They explained and used a version of the multivariate EWMA proposed by Xie et al. [11] to compare the results.

Knoth and Schmid [12] pointed out that in the case of autocorrelation in data, which is widespread in industrial datasets, neglecting the dependencies and applying standard control charts can cause more false signals and long delays in detection. Therefore, the classical control charts cannot be applied directly, and it is necessary to use the time-series structure for constructing new control charts. They concluded that no chart emerges as consistently superior for independent variables overall.

The previously mentioned works are based on statistical analyses. Clearly, other nonstatistical approaches have their own pro and cons. Zenisek et al. [13] presented a concept drift detection comparison based on machine learning (ML) methods for PM. Lu et al. [14] provided a high-quality instructive review of the drift methodologies, techniques, and learning under concept drift. Kollmann et al. [15] investigated different combinations of ML methods, preprocessing, and varying degrees of drifts.

Noor-ul-Amin et al. [16] introduced an adaptive EWMA algorithm in Bayesian theory and provided a list of related research for different combinations of EWMA and CUSUM with complementary approaches to optimize them in the case of both small and large shifts at the same time. Sparks et al. [17] presented a Poisson adaptive EWMA by dynamically changing a main configuration parameter (λ). Kollmann et al. [15] exemplified how to take advantage of CUSUM and EWMA to investigate the performance of different regression-based ML approaches on realistic industrial measurement in the PM domain. To exemplify the application of the mentioned drift detectors in their intended setup, Wendt et al. [18] presented a cognition-inspired architecture that utilizes CUSUM and EWMA as drift detectors.

Gan [19] explained and examined one-sided and two-sided EWMA charts and compares the results with CUSUM and Shewhart charts. To detect an increase in the mean value, Hu et al. [4] used an upper Poisson EWMA chart to reset the EWMA statistic. Also, they compare four different versions of EWMA and indicate that for different drift rates, no candidate can uniformly perform better than the others.

III. Drift Detection Algorithms

A. Drift Detection in Condition Monitoring

An application-agnostic condition monitoring system does not require knowledge about the system under observation (SuO), such as models. It treats the observed system as a black box and tries to determine whether it works correctly or malfunctions. As an example, Fig. 1 shows the block diagram of CCAM listening to the inputs and outputs of a SuO in order to make conclusions about its condition.[1] For drift detection, CCAM calculates and stores DABs. In this context, a DAB is the average value of a certain number of consecutive signal samples. Such DABs are shown in red in Fig. 2. A drift in the signal can be seen in the changing average over a set of DABs. Using DABs instead of just single signal samples for this comparison has two advantages: 1) the DABs effectively act as a low-pass filter on the input signal in order to reduce noise and outliers, and 2) it can save memory and computation time because fewer comparisons are performed.

[1] An open-source implementation is available at https://phabricator.ict.tuwien.ac.at/source/SoC_Rosa_repo.git

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

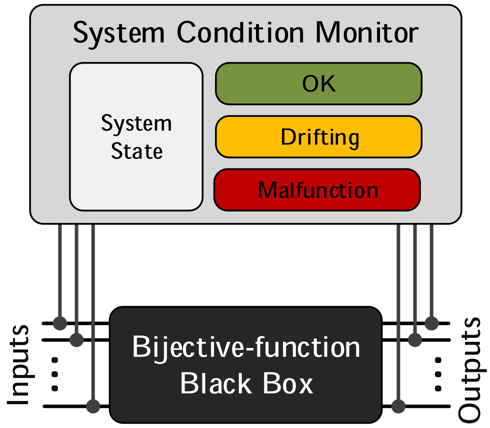

Fig. 1. Block diagram of the working principle of CCAM.

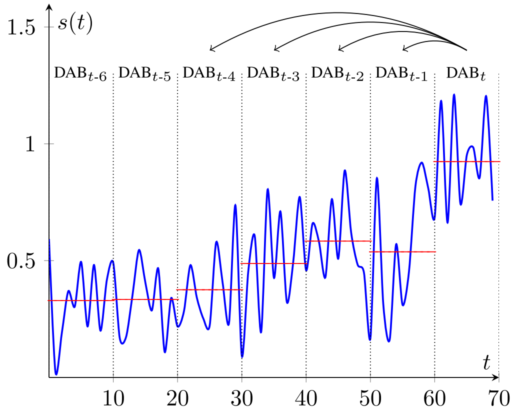

Fig. 2. Finding a drift in a signal by means of stored DABs.

Although the general principle of this rather simple filtering method proved to work [1], it has limitations in the case of more complex data (very noisy or prolonged drifts), which are the focus of this article.

B. Exponentially Weighted Moving Average

Roberts introduced EWMA in [3]. It is the right choice for detecting small drifts, is easy to set up and operate, and a two-sided version is defined as

$$Z_i = \lambda X_i + (1-\lambda)Z_{i-1} \tag{1}$$

where Z_i is the calculated value for data sample X_i. The last sample has a weight of λ, and the previous samples have a weight of $1-\lambda$. In this definition, λ is a constant $(0 < \lambda \leq 1)$ and an essential parameter for configuring the system. Z_0 initializes with the mean value of the input signal. For each incoming sample X_i, Z_i is calculated, and any boundary violation means drift. These boundaries are referred to as upper control limit (UCL) and lower control limit (LCL), which in steady state of classic EWMA, are defined as

$$\text{UCL} = \mu_0 + L\sigma\sqrt{\frac{\lambda}{(2-\lambda)}}$$

$$\text{LCL} = \mu_0 - L\sigma\sqrt{\frac{\lambda}{(2-\lambda)}} \tag{2}$$

where σ represents the standard deviation of input data, a primary assumption being that it is constant, and L is a fixed configuration parameter. On the other hand, the value of $\sqrt{(\lambda/(2-\lambda)}$ is constant and as mentioned, σ is also constant. Therefore, we decoupled the definition of UCL and LCL from λ, which simplifies the configuration.

By assuming

$$\Delta = L\sigma\sqrt{\lambda/(2-\lambda)} \tag{3}$$

the boundaries are updated to

$$\text{UCL} = \mu_0 + \Delta \quad \text{and} \quad \text{LCL} = \mu_0 - \Delta. \tag{4}$$

Now, Δ is the configurable parameter that defines the distance of limits from the mean value. This change slightly increased the average score in our experiments.

One explanation is that when a signal's mean value changes, the signal's standard deviation at that window frame also slightly increases. It means an undesired widening of the boundaries at drift time. The modification explained prevents changing the boundaries, which increases the score, as explained in Section IV-D2.

Another aspect of the decoupling parameters is that after finishing an experiment, we can effortlessly investigate the impact of changing LCL and UCL without rerunning the experiment. In fact, after running the experiment, we can determine which Δ maximizes the score. It is a remarkable help for training and (auto-)tuning of the algorithm.

A well-known issue of EWMA is resistance and delay in detecting drifts on the opposite side. Woodall et al. [20] recommend always using a Shewhart chart in conjunction with an EWMA as one way to counteract the signal resistance. In addition, during the drift phase, Z_i is increasing. When the drift is over and another drift in the opposite direction starts, Z_i is far from the center line. Thus, it will take time to return to the center and reach the other limit. Our solution to overcome this issue is to limit Z_i with the help of δ as follows:

$$\begin{aligned} &\text{if } \ Z_i > \text{UCL} + \delta, \ \text{set } Z_i = \text{UCL} + \delta \\ &\text{if } \ Z_i < \text{LCL} - \delta, \ \text{set } Z_i = \text{LCL} - \delta. \end{aligned} \tag{5}$$

The impacts of these correctors in long and steady drift phases are more recognizable. This concept is similar to the boundary limitation on one-sided EWMA. Gan [19] recommended using limits on one-sided EWMA charts to ensure that the EWMA stays within a certain distance of the chart limit regardless of the situation.

C. Cumulative Sum

CUSUM was introduced by E. S. Page in 1954 in the statistical quality control concept [3]. CUSUM aggregates the deviations from the mean value in the sequence of input data. As long as the cumulated value is in the predefined range, the system is stable. The tabular or algorithmic CUSUM works by accumulating deviations from the mean value in two separate variables, C^+

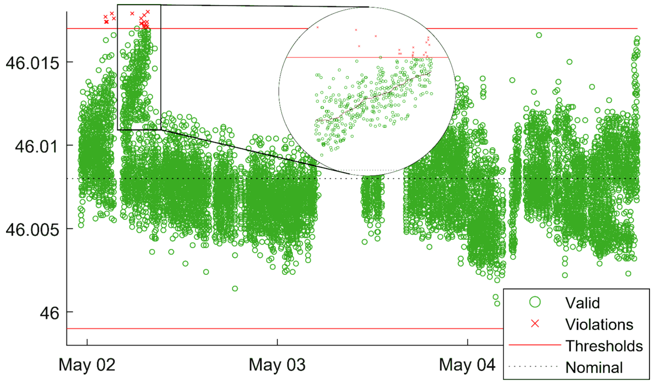

Fig. 3. Scatterplot of a short section of real-world data with a magnified drift region.

and C^-, which are also called one-sided upper and one-sided lower CUSUM, respectively. Let X_i be the ith input sample, and μ_0 be the mean value of the input data, then the one-sided upper and one-sided lower CUSUM are defined as

$$\begin{aligned} C_i^+ &= \max[0, X_i - (\mu_0 + K) + C_{i-1}^+] \\ C_i^- &= \max[0, (\mu_0 - K) - X_i + C_{i-1}^-] \\ \text{where } C_0^+ &= C_0^- = 0. \end{aligned} \tag{6}$$

In these equations, K is a configuration parameter also called the reference or slack value. Based on (6), deviations smaller than K are not accumulated. C^+ and C^- are updated for each new input data. If one of them is bigger than H, the signal drifts. The H parameter is the decision interval and defines the stable boundary around the mean value.

CUSUM algorithm shows a delay when detection after drift, especially for a drift following another drift of the opposite direction. To reduce this well-known delay, we limited the increment of C^+ and C^- and considered a margin of 10% of the H value.

By introducing a new parameter named $\hat{K}$, we control the impact of incoming samples by limiting them to the $\hat{K}$. In other words, deviations from the mean value smaller than K are ignored, and deviations bigger than $\hat{K}$ are truncated to $\hat{K}$. This concept can be implemented either as algorithm modification or separately as a preprocessing task (see the truncation span in Section IV-B1).

Another modification is that after any drift, the value of the one-sided upper/lower parameter for the opposite direction is reset to 0.5 times H. This modification helps the algorithm to be more agile in detecting drift in the opposite direction.

IV. Experiments

A. Experimental Data

Over six months of real-world measurements taken from a conrod production line at an automotive factory are available for our experiments. Fig. 3 shows a short section of it. There are several noticeable drifts in the data, with various directions and amplitudes that, in one case, lead to a region of increased violations of the quality tolerance band. The respective section of the scatter plot was magnified to show this drift in more detail. The additional line drawn within the inset represents the running mean of the region, calculated with a window size of 128 steps. The delta of this moving mean is 1.58×-10^{-5} on average and 7.03×10^{-5} maximum. Several sections, similar to the depicted, have been analyzed and provide the base for the sigma and delta parameterization of the synthetic data. Analysis of the whole dataset has shown that the moving sigma varies between 1.7×10^{-3} and 2.0×10^{-3}, depending on the actual window size.

TABLE I
Dataset Parameters

Parameter	Range
Mean	[0.0]
Standard deviation	[0.01, 0.02, ...0.10]
Delta	[2.0, 4.0, ..., 20.0]
Outlier chance	[0.1]
Outlier severity	[0.0, 1.0, 2.0, 3.0, 4.0]

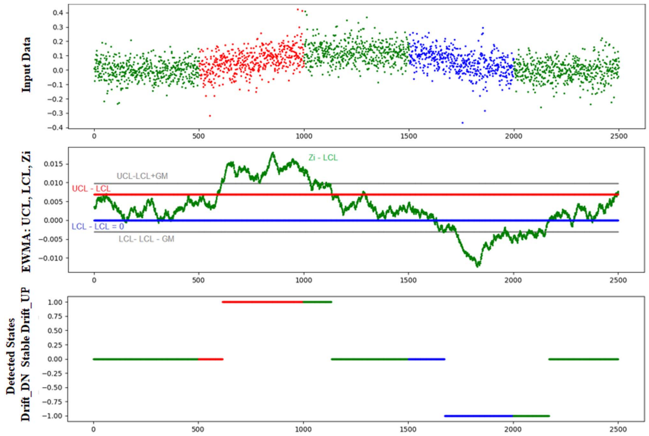

Fig. 4. Input data (Green: Stable, Red: Drift up, Blue: Drift down), statistical properties obtained from real-word data, e.g., Fig. 3. EWMA (UCL, LCL, and Zi, for better visual display, shifted by −LCL) and detected states.

1) Parameter Space: We extracted statistical properties as guidelines for defining the parameter space (see Table I). With this, we generated synthetic data with a uniform shape that reflected the significant statistical properties of the real measurements. All experiments in this dataset commenced with a stable phase, followed by a drift upward. After another stable phase, there was a drift downward, and finally, another stable phase. This approach provided us with extensive experimental data that was correctly labeled with the direction of drift for validation (see Section V). Fig. 4 illustrates the general form of the input signals.

The data generation is based on the parameters given in Table I and starts with a basic, clean signal. The signal models the drifts by slowly changing the mean up or down, according to delta. Then, we generate two different sets of noise, one using the standard deviation given in the parameter table, and a second, broader distribution where the standard deviation is multiplied by the outlier severity parameter. Finally, we apply the generated noise to the clean signal by adding the noise value from the smaller or larger distribution, with the probability of choosing the larger distribution given by the outlier chance parameter, to the clean signal.

2) Data Segmentation: Regarding the parameter space, presented in Table I, there are ten different levels for the standard deviations and drift rates, referred to as SD0.01 to SD0.10 and DR2.0 to DR20.0. The drift rate defines how fast or slow the mean value of input data changes over time. There are five levels for the outliers and five instances for each element. The dataset contains 2500 elements, and each element contains 2500 data samples. We use the *experiment* term to refer to a particular element with an algorithm configuration. Here, a subset of the dataset is called a data segment. The dataset is initially divided into 100 data segments according to standard deviation and drift rate. For instance, data segment (SD0.02, DR4.0) means all data elements with SD0.02 and DR4.0 properties, which means 25 experiments.

B. Data Processing

In this section, we initially describe a method for preprocessing, followed by two methods for postprocessing.

1) Truncation Span: In both algorithms, CUSUM and EWMA, the distance of the incoming sample from the mean value is measured to indicate the signal status. The outliers, by their nature, are mostly far from the mean. In fact, undesired outliers have more impact on the results. To minimize the impact of outliers, we define a truncation span parameter SPN used for each incoming data point, X_i, as

$$\begin{aligned}&\text{if } \ X_i > \mu_0 + \text{SPN} \ \text{ set } X_i = \mu_0 + \text{SPN}\\ &\text{if } \ X_i < \mu_0 - \text{SPN} \ \text{ set } X_i = \mu_0 - \text{SPN}\end{aligned} \tag{7}$$

where μ_0 represents the mean value of input data. We can either use this concept as data prefiltering or modify the algorithms. The proposed $\hat{K}$ in CUSUM, explained in Section III-C, is an algorithm modification to implement truncation span.

Another simple and effective method is the moving average. In addition, well-known generalizations of the moving average are also practical. Locally estimated scatterplot smoothing or the so-called Savitzky–Golay filter and locally weighted scatterplot smoothing are common approaches [21]. In our implementation, the preprocessing is separated from the main logic. This separation and modularity allow using other suitable preprocessing and noise-filtering approaches that match the input data.

2) Gray Margin (GM): Algorithms largely compare calculated values (e.g., Z_i or C^+/C^-) with a reference value or threshold to determine the signal status. If the comparing value shows some fluctuations, the result bounces between different levels. In the validity assessment of the experiments, frequent changes in detections are not allowed. By defining a boundary around the threshold, the algorithm switches the status when the comparing value is bigger than the threshold plus the GM. Similarly, the status changes when the comparing value is less than the threshold minus the GM. For the values between, the algorithm sticks to the last detection.

3) Switching Inertia (INR): In the INR method, the algorithm switches to a new state if, at least for the predefined value of the INR parameter, the algorithm has detected only the new condition. In practice, it ignores the detection sequences that persist less than INR samples.

TABLE II
SAMPLE-BASED EVALUATION WEIGHTS

		Signal status detected		
		STABLE	DRIFT UP	DRIFT DN
Data label	STABLE	1.0	−0.5	−0.5
	DRIFT UP	−0.5	1.0	−2.0
	DRIFT DN	−0.5	−2.0	1.0

C. Experimental Procedure

Each dataset is saved in a separate file with its key attributes, such as drift rate and standard deviation, decoded in the filename. A central application creates instances of the target algorithms with the desired configurations and sequentially inserts the input data into the algorithms. The output files the algorithms create for each experiment contain the algorithm's classification results for each data point, the input data, and the configuration parameters. Another script evaluates these results and determines whether the algorithm's classification results are right or wrong.

D. Evaluation Metrics

For performance evaluation, the main objective of the algorithms is to confidently detect the status of the input signal without switching back and forth between possible statuses. The second objective is the speed of the detection. The status of the input stream data is defined as stable when the mean value is not changing. On the other hand, the signal shows a drift downward or upward when its mean value decreases or increases, respectively.

1) Phase-Based Metric: Here, a *phase* is defined as a subset of sequential input data in which the signal's status does not change. The length of a phase is the number of data samples with the same status. A delay in detection (DD) refers to the number of samples before the first correctly detected status. This parameter corresponds to the average run length, and here, it refers to the exact value of one observation, not an average value. In order to avoid confusion the DD term is used here. An algorithm detects a phase correctly if it detects, after the DD, all samples of that phase correctly.

Fig. 4 presents an example in the top part containing five color-coded phases: stable, drift up, stable, drift down, and stable. The red and blue parts of the lines in the middle of the lower section represent the DD for the drift up and drift down phases.

2) Sample-Based Metric: Contrary to the phase-based metric, which looks for periods of stable signal status, the sample-based metric is focused on the individual data points. In this metric, for each data sample, depending on the detected status and input data labels, a predefined score is considered, the sum of all scores per sample shows the sample-based score.

Table II depicts the selected weights for a sample-based evaluation. With this definition, the maximum possible score is equal to the number of samples, and it is convenient to use it

TABLE III
TOP: PERCENTAGE OF VALID EXPERIMENTS (PHASE-BASED) BOTTOM: AVERAGE OF THE SAMPLE-BASED METRIC (DATA SEGMENT: SD0.01, DR2.0)

$\lambda\downarrow$ / $L\rightarrow$	0.6	0.7	0.8	0.9	1.1	1.3	1.5	1.7	1.9	2.1
0.005	4	8	16	20	24	52	64	72	84	88
0.01	52	56	48	68	84	100	100	96	100	100
0.02	0	8	4	24	32	72	60	76	96	100
0.035	0	0	0	0	8	16	32	40	32	52
0.05	0	0	0	0	4	8	12	24	20	8

$\lambda\downarrow$ / $L\rightarrow$	0.6	0.7	0.8	0.9	1.1	1.3	1.5	1.7	1.9	2.1
0.005	0.31	0.33	0.30	0.29	0.29	0.28	0.30	0.30	0.31	0.32
0.01	0.48	0.49	0.51	0.52	0.53	0.55	0.56	0.57	0.57	0.58
0.02	-	0.69	0.69	0.68	0.70	0.70	0.70	0.71	0.71	0.71
0.035	-	-	-	-	0.79	0.77	0.77	0.78	0.77	0.77
0.05	-	-	-	-	0.80	0.81	0.80	0.80	0.80	0.80

TABLE IV
CUSUM, MAX SAMPLE-BASED SCORE, AND BEST MATCHING SETTINGS

DR	K	H	GM	Score	K	H	GM	Score
2.0	**0.05**	0.06	0.0175	0.435	0.05	0.06	0.0175	0.435
4.0	**0.08**	0.04	0.0175	0.619	0.08	0.04	0.0200	0.611
6.0	**0.07**	0.04	0.0150	0.639	0.08	0.04	0.0200	0.628
8.0	**0.08**	0.04	0.0175	0.637	0.08	0.04	0.0200	0.631
10.0	**0.08**	0.04	0.0200	0.631	0.08	0.04	0.0200	0.631
12.0	**0.08**	0.04	0.0150	0.631	0.08	0.04	0.0200	0.620
14.0	**0.07**	0.04	0.0150	0.607	0.08	0.04	0.0200	0.605
16.0	**0.08**	0.05	0.0200	0.579	0.08	0.05	0.0200	0.579
18.0	0.08	0.05	0.0200	0.579	0.08	0.05	0.0200	0.579
20.0	0.08	0.04	0.0150	0.590	0.08	0.05	0.0200	0.550

relatively, which is equal to the aggregated score divided by the number of samples.

In the sample-based metric, the detection order does not matter, and shuffling the results inside a phase does not affect the score. Most systems are sensitive to the frequent switching between possible states, so the sample-based metric alone is not a good choice and complementary metrics are required.

3) Experimenting and Validity Test: Correctly detecting all drifts in all experiments of a particular data segment is a prerequisite. Even missing one drift in one experiment makes a disqualified configuration. After passing this strict filter, maximizing the sample-based score is the secondary goal. Regarding the slightest drift, finding an acceptable configuration is challenging. However, by increasing the drift rate, the total number of adequate configurations increases, and the focus shifts to finding the best sample-based score.

The upper section of Table III depicts the percentage of valid experiments in the selected segment according to the EWMA algorithm. Only λ and L are varying, and other parameters are fixed (WND:50, INR:10, SPN:0.01, GM: 0.0008). The lower section of Table III shows the average values of the sample-based metric's valdiation. Of the five combinations that passed the validity test, marked green, the best sample-based score is 0.71.

V. RESULTS

A. CCAM Results

As mentioned, CCAM is a condition monitoring system designed for autonomous state detection and does not utilize pre and postprocessors for drift detection. According to the experimenting dataset, especially for the small drifts (DR2.0 and DR4.0), it often fails to pass the validity tests. It requires smaller data segments, which contrasts with the generalization of drift detection to cover relatively big data segments. Therefore, we conclude that the CCAM's drift detection is unsuitable for the dataset described above.

B. CUSUM Results

Finding a CUSUM configuration that detects the slightest drift correctly is challenging. Fig. 7 depicts that the average sample-based score for DR2.0 is much lower than other drift rates. To increase the score, we can use smaller data segments, which reduces the algorithm's generalization.

The sample-based score slightly reduces for higher drift rates (DR8.0 to DR16.0). One explanation is that increasing the drift rate means the comparison references (C^+ and C^-) need more time to hit the threshold value. Therefore, the sample-based score is reduced. It is a good sign for detecting the change point of configuration and is useful in autoparametrization.

Regarding the smallest drift, although it is difficult to find a suitable configuration for CUSUM, the sample-based score is slightly higher compared to EWMA. Data segments with DR2.0 properties are the only data segments for which CUSUM has an advantage in sample-based scores. We can conclude that the DR2.0 is the minor detectable drift with the assumed validity test.

Table IV shows the combinations of parameters for multiple data segments (SD0.05, WND=50, INR=10), grouped by drift rate, which maximize the sample-based score. These experiments have passed the validity test. The parameters shown on the left side represent the best average sample-based scores. The right side of the table represents alternative combinations of parameters from the same data segment, which are closer to the neighbor's configurations and not necessarily the maximum value of the sample-based score but very close to it, which is demonstrated in Fig. 5. In this process, the average sample-based score was reduced only from 0.595 to 0.587, which is less than 2%.

The averages of sample-based scores for each data segment over standard deviation are almost the same (between 0.581 and 0.605), which means that the varying standard deviation is covered by CUSUM nicely.

C. EWMA Results

The drift rate defines how the mean value of input data changes over time. Experiments with the DR2.0 attribute contain the smallest and most challenging drifts. Only a few configurations can satisfy the requirements, and the sample-based score is relatively low. In experiments with data segment (SD0.01, DR2.0),

TABLE V
EWMA, AVERAGE SAMPLE-BASED SCORE (SD0.04, DR2.0)

$\lambda\downarrow$ / $L\rightarrow$	**0.6**	**0.7**	**0.8**	**0.9**	**1**	**1.1**	**1.3**	**1.5**	**1.7**	**1.9**	**2.1**	**2.3**	**2.5**	**2.7**	**2.9**
0.005	0.28	0.28	0.28	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.28	0.29	0.29
0.01	0.42	0.43	0.44	0.44	0.45	0.46	0.48	0.49	0.5	0.52	0.52	0.53	0.54	0.55	0.55
0.02	0.56	0.57	0.59	0.6	0.61	0.62	0.63	0.64	0.65	0.66	0.67	0.68	0.68	0.69	0.69
0.03	0.6	0.62	0.64	0.65	0.66	0.67	0.69	0.7	0.71	0.72	0.72	0.73	0.73	0.74	0.74
0.04	0.63	0.66	0.67	0.68	0.7	0.71	0.72	0.73	0.74	0.75	0.75	0.76	0.77	0.77	0.77
0.05	0.66	0.68	0.69	0.71	0.72	0.73	0.74	0.75	0.76	0.77	0.78	0.78	0.78	0.79	0.79

TABLE VI
EWMA, VALIDITY TEST (SD0.04, DR2.0)

$\lambda\downarrow$ / $L\rightarrow$	**0.6**	**0.7**	**0.8**	**0.9**	**1**	**1.1**	**1.3**	**1.5**	**1.7**	**1.9**	**2.1**	**2.3**	**2.5**	**2.7**	**2.9**
0.005	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0.01	x	x	x	x	x	Ok	x	x	Ok	Ok	Ok	Ok	Ok	Ok	Ok
0.02	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0.03	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0.04	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0.05	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

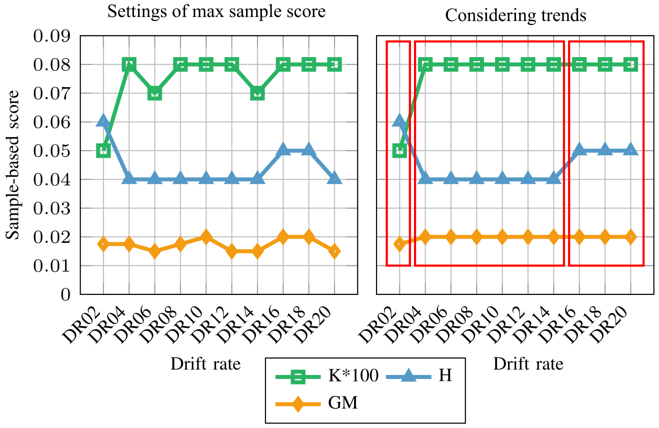

Fig. 5. CUSUM: Unifying settings (ordinate labeling is valid for both plots).

only three of 1080 combinations have passed the validity test, which means very limited flexibility for configuration.

In other data segments with different standard deviations but with the same drift rate (DR2.0), the same results are observable, and $\lambda = 0.005$ or $\lambda = 0.01$ is a strong constraint.

According to the sample-based metric, Table V depicts that by increasing λ and L, EWMA reacts faster to the changes, and the sample-based score increases in a limited range. The undesired side effect is increasing false detection and failing in the experiment's validity test. Table VI shows that only a few marked green combinations have passed the validity test.

Also, for the smallest drifts, finding an acceptable configuration is challenging. Still, for relatively bigger drifts, there are many options, and maximizing the sample-based score matters, and the possibility for merging the neighbor data segments increases.

Table VII shows selected combinations of parameters for multiple data segments (SD0.04), grouped by drift rate. These experiments have passed the validity test (100% correct phase detection in all experiments). The parameters shown on the left side represent the best average sample-based scores. The right side of the table represents another combination of parameters from the same data segment, which have passed the validity test and are most similar to the neighbor's configurations. Fig. 6 shows that in the new selections, parameters are nicely aligned and primarily unified. In this process, the average sample-based score was only 2% reduced. This figure also gives an idea of how a new data segmentation looks ([DR2.0], [DR4.0–DR6.0], [DR8.0 to DR20.0]).

TABLE VII
EWMA, MAX SAMPLE-BASED SCORE, AND BEST MATCHING SETTINGS

DR	λ	SPN	GM	L	Score	λ	SPN	GM	L	Score
2.0	0.005	0.06	0.0035	1.7	0.359	0.005	0.06	0.0035	1.7	0.358
4.0	0.030	0.06	0.0135	1.1	0.688	0.030	0.06	0.0135	1.1	0.688
6.0	0.040	0.05	0.0135	1.5	0.729	0.030	0.06	0.0135	1.1	0.695
8.0	0.050	0.06	0.0135	2.1	0.770	0.050	0.06	0.0135	2.1	0.770
10.0	0.050	0.05	0.0100	2.1	0.773	0.050	0.06	0.0135	2.3	0.766
12.0	0.050	0.04	0.0100	2.5	0.762	0.050	0.06	0.0135	2.3	0.760
14.0	0.050	0.06	0.0135	2.7	0.764	0.050	0.06	0.0135	2.7	0.764
16.0	0.050	0.06	0.0100	2.7	0.768	0.050	0.06	0.0135	2.3	0.755
18.0	0.050	0.04	0.0065	2.7	0.776	0.050	0.06	0.0135	2.5	0.755
20.0	0.050	0.05	0.0100	2.9	0.756	0.050	0.06	0.0135	2.3	0.740

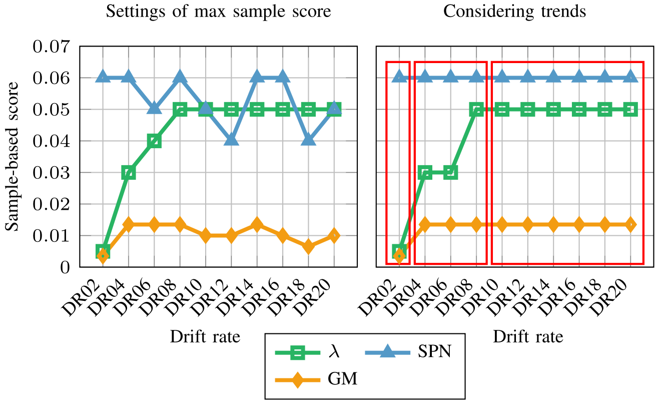

Fig. 6. EWMA: Unifying settings (ordinate labeling is valid for both plots).

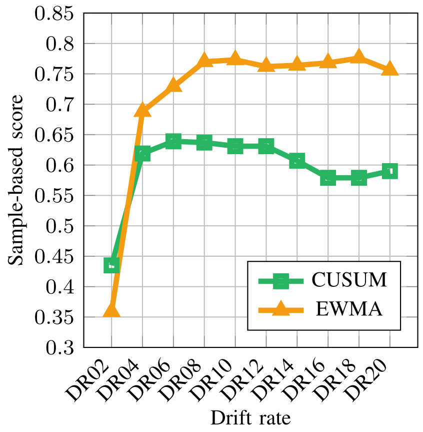

Fig. 7. EWMA versus CUSUM: Sample-based score over drift rate.

D. Comparison Result

The comparison of the classic version of EWMA and CUSUM, defined by Montgomery [3], with the modified version explained in Sections III-B and III-C, shows that for CUSUM in most regions, the modified version has better sample-based scores. However, for the EWMA, the sample-based scores are very close, and the superiority depends on the settings. For the segments with small drifts, it is impossible to pass the validity tests with classic versions of algorithms.

Fig. 7 depicts the sample-based score of both EWMA and CUSUM for the selected configurations that have passed the validity test. CUSUM has a better score of DR2.0, but EWMA has an obvious advantage over the rest.

In general, if a configuration is suitable for detecting small drifts, it can detect relatively bigger drifts, but slowly, the side effect is reducing the sample-based score.

VI. Guidelines for Practical Use

This section provides some generalized and practicable tips for setting up a new experiment.

1) First of all, be aware of the input data, examine the statistical properties, and control the validity of assumptions.
2) *Define the preprocessing methods:* It often changes the statistical properties of data and, consequently, the algorithm configurations. It is a double-edged sword: it can complicate the configuration or create an opportunity to unify the statistical properties of the input data to benefit from a pretuned algorithm.
3) *Adjust the algorithm and select an acceptable range of parameters:* For example, Table V demonstrates that increasing the range of L is a wise decision while the best value is just in the border.
4) *CUSUM:* An estimation for K is selecting halfway between the mean value and drift situation, and an advisable value for H, the decision interval, is around five times the standard deviation of input data.
5) Select or develop an appropriate strategy for periodically updating the mean value:
6) *EWMA:* A typical value for λ is between 0.02 and 0.2. Use smaller values for smaller drifts. For the standard version, consider L between 1.5 and 3.0, but for using the modified version, select Δ from (4) according to the allowed fluctuation of the mean value.
7) *EWMA:* In order to limit Z_i, a practical value for δ is 10% of Δ. When the INR parameter is big enough, selecting smaller values for Δ is possible.
8) An appropriate value for INR depends not only on preprocessing methods but also on the tradeoff between early drift detection and false detection. Increasing the INR reduces the false detections and also the sample-based score.
9) Using suitable postprocessing techniques like the INR parameter can improve the overall performance. Increasing this parameter will reduce false detection, but will also increase the DD.
10) Although it is possible to use GM and INR simultaneously, consider them as alternative solutions. The INR method is efficient when more extensive changes in the comparison variable exist. The GM performs better when there are numerous and more minor fluctuations.
11) The GM only considers the distance of samples from the reference value without considering the number of samples. In contrast, INR counts the number of samples over the threshold without considering the distance of samples from the reference value.
12) To estimate the range of the GM, run the algorithm over a stable phase. For EWMA, adjust the parameters so that the maximum Z_i is between UCL and UCL plus GM, and the minimum Z_i is between LCL and LCL minus GM. In the same way, it is assignable for CUSUM.
13) Moreover, evaluate the outputs and fine-tune the configuration iteratively.

VII. Conclusion

The primary drift detection of CCAM effectively detects medium and abrupt drifts. However, utilizing a modified version of EWMA or CUSUM is better for detecting tiny drifts. Standardizing the interfaces to the condition monitoring system allows for orchestrating a suitable set of preprocessing methods, drift detection algorithms, and postprocessors. It provides simplicity in configuration and experimenting.

Proposed changes in the algorithms, such as the dynamic adaptation of the mean value, decoupling the algorithm limits from the algorithm parameters, and using the corrector equations, make the EWMA an effective mean-adaptive algorithm and a well-tuned EWMA can cover a wide range of input data. Introducing $\hat{K}$ and resetting the one-sided upper/lower parameter after drift detection makes the CUSUM more agile. Due to the versatility of the approach in drift detection, the algorithm will be tested in the building automation environment to determine maintenance in HVAC systems using the mix of simulated data and on-site measurements.

The rigid validity test for detecting all the signal phases correctly and without switching between the possible statuses, alongside the wide range of drifts, creates a complicated situation. In many practical use cases, covering such a wide range of drifts is not necessary. Instead quicker detection and reaction may be of interest. Limiting the variable space reduces the complexity of the configuration and improves the efficiency of the specialized profiles for dealing with a specific task.

There are two basic approaches to processing the input data: first, adopting the input data with the algorithms. For example, modified data with desired properties are delivered to the well-tuned algorithms using preprocessors and normalizing input data. In the second approach, the algorithm is flexible to handle a broad range of input data, which requires some adaptation and autoparametrization. Our work mostly benefits from the first method, and the second approach is future work. Autoparametrization of drift detectors prevents the exhausting and time-consuming configuration process, and, more importantly, it is required for adopting and processing dynamic input streams. In this issue, autoparametrization and self-tuning of algorithms, besides a discrete definition of EWMA/CUSUM, are planned as future works.

References

[1] M. Götzinger et al., "Model-free condition monitoring with confidence," *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 4/5, pp. 466–481, 2019, doi: 10.1080/0951192X.2019.1605201.

[2] I. M. Zwetsloot, T. Mahmood, and W. H. Woodall, "Multivariate time-between-events monitoring: An overview and some overlooked underlying complexities," *Qual. Eng.*, vol. 33, no. 1, pp. 13–25, 2020.

[3] D. Montgomery, *Introduction to Stat. Qual. Control*. Hoboken, NJ, USA: Wiley, 2020. [Online]. Available: https://books.google.at/books?id=oh7zDwAAQBAJ

[4] J. Hu, L. Shu, P. W. Tse, and K.-L. Tsui, "A comparison of EWMA-type charts under linear drifts in Poisson means," in *Proc. IEEE Int. Conf. Qual. Rel.*, 2011, pp. 297–301.

[5] J. Zhang, Z. Li, B. Chen, and Z. Wang, "A new exponentially weighted moving average control chart for monitoring the coefficient of variation," *Comput. Ind. Eng.*, vol. 78, pp. 205–212, 2014.

[6] R. S. M. Barros and S. G. T. C. Santos, "A large-scale comparison of concept drift detectors," *Inf. Sci.*, vol. 451, pp. 348–370, 2018.

[7] P. M. Gonçalves Jr, S. G. de Carvalho Santos, R. S. Barros, and D. C. Vieira, "A comparative study on concept drift detectors," *Expert Syst. Appl.*, vol. 41, no. 18, pp. 8144–8156, 2014.

[8] S. Jaramillo-Valbuena, J. M. Londoño-Peláez, and S. A. Cardona, "Performance evaluation of concept drift detection techniques in the presence of noise," *Perform. Eval.*, vol. 38, no. 39, 2017, Art. no. 16.

[9] F. Yi and P. Qiu, "An adaptive CUSUM chart for drift detection," *Qual. Rel. Eng. Int.*, vol. 38, no. 2, pp. 887–894, 2022.

[10] D. M. Hawkins and Q. Wu, "The CUSUM and the EWMA head-to-head," *Qual. Eng.*, vol. 26, no. 2, pp. 215–222, 2014.

[11] Y. Xie, M. Xie, and T. N. Goh, "Two MEWMA charts for Gumbel's bivariate exponential distribution," *J. Qual. Technol.*, vol. 43, no. 1, pp. 50–65, 2011.

[12] S. Knoth and W. Schmid, "Control charts for time series: A review," *Front. Statist. Qual. Control*, vol. 7, pp. 210–236, 2004.

[13] J. Zenisek, F. Holzinger, and M. Affenzeller, "Machine learning based concept drift detection for predictive maintenance," *Comput. Ind. Eng.*, vol. 137, 2019, Art. no. 106031.

[14] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2346–2363, Dec. 2019.

[15] S. Kollmann, A. Estaji, A. Bratukhin, A. Wendt, and T. Sauter, "Comparison of preprocessors for machine learning in the predictive maintenance domain," in *Proc. IEEE 29th Int. Symp. Ind. Electron.*, 2020, pp. 49–54.

[16] M. Noor-ul Amin and S. Noor, "An adaptive EWMA control chart for monitoring the process mean in Bayesian theory under different loss functions," *Qual. Rel. Eng. Int.*, vol. 37, no. 2, pp. 804–819, 2021.

[17] R. S. Sparks, T. Keighley, and D. Muscatello, "Improving EWMA plans for detecting unusual increases in poisson counts," *Adv. Decis. Sci.*, vol. 2009, 2009, Art. no. 512356, doi: 10:1155/2009/512356.

[18] A. Wendt, S. Kollmann, A. Bratukhin, A. Estaji, T. Sauter, and A. Jantsch, "Cognitive architectures for process monitoring—an analysis," in *Proc. 18th IEEE Int. Conf. Ind. Informat.*, 2020, pp. 167–173.

[19] F. Gan, "Designs of one-and two-sided exponential EWMA charts," *J. Qual. Technol.*, vol. 30, no. 1, pp. 55–69, 1998.

[20] W. H. Woodall and M. A. Mahmoud, "The inertial properties of quality control charts," *Technometrics*, vol. 47, no. 4, pp. 425–436, 2005.

[21] R. W. Schafer, "What is a Savitzky-Golay filter? [Lecture notes]," *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 111–117, Jul. 2011.

Alireza Estaji received the B.Sc. degree in software engineering from the Ferdowsi University of Mashhad (FUM), Mashhad, Iran, in 1997, and the Master of Engineering degree in information technology from the Oulu University of Applied Sciences, Oulu, Finland, in 2011. He is currently working toward the Ph.D. degree in street lighting optimization with the Institute of Computer Technology, Vienna University of Technology, Vienna, Austria.

For more than ten years, he experienced and studied "industrial production and management."

Maximilian Götzinger received the master's degree in computer technology from the Vienna University of Technology (TU Wien), Vienna, Austria, and the Ph.D. degree from the Department of Computing, University of Turku, Turku, Finland, in 2021.

His activities with both universities can be quantified with 15 publications at international conferences and journals.

Dr. Götzinger was the recipient of the Best Paper Award, a Best Teacher Award, a Best Lecture Award for his activities with TU Wien and University of Turku, and various scholarships from notable institutions.

Benedikt Tutzer received the B.S. degree in computer engineering and the master's degree in embedded systems from TU Wien, Vienna, Austria, in 2018 and 2022, respectively.

In 2017 and 2018, he was with the Interactive Media Systems Group, TU Wien, researching the applications of virtual-reality headsets as a seeing aid for visually impaired patients at the Vienna General Hospital. Since 2018, he has been with the Institute of Computer Technology, TU Wien, focusing on electronic design automation.

Stefan Kollmann studied Media Informatics at the Vienna University of Technology, Vienna, Austria, where he has been actively engaged in scientific research since 2013. In addition to his work in data and information visualization, he contributed to several foundational Bionically Inspired Artificial General Intelligence research projects. In recent years, he focused on more traditional Machine Learning methods and their application in various fields, including building optimization and smart grids.

Thilo Sauter (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the Vienna University of Technology (TU Wien), Vienna, Austria, in 1999.

He is currently a Professor of Automation Technology with TU Wien and was the Founding Director of the center for integrated sensor systems, Danube University Krems, Wiener Neustadt, Austria.

Dr. Sauter is a Senior Administrative Committee Member of the IEEE Industrial Electronics Society.

Axel Jantsch (Fellow, IEEE) received the Dipl.Ing. and Ph.D. degrees in computer science from the Vienna University of Technology (TU Wien), Vienna, Austria, in 1987 and 1992, respectively.

From 1997 to 2014, he was an Associate Professor and since 2002, he has been a Full Professor in Electronic Systems Design with the KTH, the Royal Institute of Technology, Stockholm, Sweden. Since 2014, he has been a Professor of Systems on Chips with TU Wien.