INTERNATIONAL JOURNAL OF COMPUTER INTEGRATED MANUFACTURING
2019, VOL. 32, NOS. 4–5, 466–481
https://doi.org/10.1080/0951192X.2019.1605201

Taylor & Francis
Taylor & Francis Group

ARTICLE

Check for updates

Model-free condition monitoring with confidence

M. Götzinger[a], N. TaheriNejad[b], H. A. Kholerdi[b], A. Jantsch[b], E. Willegger[b], T. Glatzl[c], A. M. Rahmani[b,d], T. Sauter[b,c] and P. Liljeberg[a]

[a]Department of Future Technologies, University of Turku, Turku, Finland; [b]Institute of Computer Technology, TU Wien, Vienna, Austria; [c]Department for Integrated Sensor Systems, Danube University Krems, Wiener Neustadt, Austria; [d]Department of Computer Science, University of California, Irvine, CA, USA

ABSTRACT

Computational Self-awareness can improve performance, robustness, and adaptivity of a system. As a key element of self-awareness, observation quality is critical to gain a correct and comprehensive understanding of the system, its own state, and the environment. This is of more importance in systems, where contextual information plays a crucial role in the functional operation of the system. In this paper, the authors introduce *confidence* as a quality metric of observation and leverage it to improve the correct identification of states of a system. To evaluate the impact of this factor on the context-aware monitoring system at hand and to show the generality of the approach, we conduct a series of tests with and without confidence for condition monitoring of an industrial AC motor and an experimental water pipe system. Our experiments show that *confidence* not only improves the quality of system performance but also simplifies the system architecture and enhances its robustness. These findings support the recent initiatives of paying more attention to observation as an important factor in self-awareness and, consequently, the performance of systems. The proposed system facilitates condition monitoring of various industrial systems and is easily deployable as it does not require a deep domain knowledge.

ARTICLE HISTORY
Received 2 June 2018
Accepted 13 March 2019

KEYWORDS
Industry 4.0; monitoring; model-free; context-awareness; self-awareness; confidence

1. Introduction

In industrial systems, manual maintenance and adjustment are often undesirable due to the high costs or the limited time available for adaptations. Increasingly, self-adaptiveness and self-awareness are desirable characteristics of many embedded and cyber-physical systems. They need to adapt to dynamic applications' phasic behaviour, changing environments and their own changing state due to phenomena such as ageing as well as the occurrence of faults. In many application domains, assessing the status of a system is commonly called *condition monitoring*, which serves diverse purposes such as early detection of faults, operation optimization, or planning of preventive maintenance.

In the early 2000's IBM formulated a vision of autonomic systems which called for self-adaptation and self-awareness (Kephart and Chess 2003). At the same time, Intel put forward a vision for proactive computing asking humans to get 'out of the loop' (Tennenhouse 2000), meaning that embedded systems should become more independent, autonomous and self-adaptive. A prerequisite for these ambitious goals is a detailed assessment of the state of a system and its environment. This assessment depends heavily on measuring relevant physical properties with sensors which is a part of Industry 4.0 (Alexopoulos et al. 2016). Many works on autonomous and adaptive systems assume the availability of correct measurements and their accurate interpretation. However, the quality of measurements and observations is not always easy to guarantee or assess. Hence, recently this challenge has received more attention (TaheriNejad, Jantsch, and Pollreisz 2016; Götzinger et al. 2017a; Anzanpour et al. 2017; Götzinger et al. 2016).

In this work, the authors focus on one of the observation aspects, namely *confidence*, and show how it can smoothen decision-making functions. Our concept of *confidence* is based on the work by TaheriNejad, Jantsch, and Pollreisz (2016) but differs from the *confidence levels assignment* used in other works, e.g. by Liu et al. (2008). To demonstrate this effect, we use this concept to enhance a black box monitor which we first introduced in (Götzinger et al. 2017b). This monitor distinguishes three different states (healthy, drifting, and broken) of the system it monitors. We demonstrate our approach by means of two case studies to illustrate its generality and application to different fields: an AC motor as in Götzinger et al. (2017b) and a water pipe system used as a model for condition monitoring in Heating, Ventilation and Air Conditioning (HVAC) systems (Glatzl et al. 2016). Extending our previous work (Götzinger et al. 2017b), we introduce the confidence metric and use fuzzy logic for its computation and for the decision-making process. We demonstrate that our proposed method improves the correct identification of motor states (i.e. increases robustness), reduces the dependency on system parameters such as threshold values (i.e. reduces sensitivity and increases robustness) and reduces the need for pre-processing of the sensory data (i.e. simplifies the architecture).

It is noteworthy that, with the growing prevalence of Industry 4.0 concepts and the increase in the number of cyber-physical systems and their associated sensors and actuators, monitoring has gained a growing importance while becoming more challenging. Model-free monitoring helps tackling this challenge from different angles. It can be applied to many

CONTACT M. Götzinger maxgot@utu.fi
© 2019 Informa UK Limited, trading as Taylor & Francis Group

different systems with substantial differences in their nature (e.g. as presented in this paper to condition monitoring of motors and water pipes) without requiring the deployment engineer to have a deep knowledge of the application field. Moreover, thanks to the fuzzy confidence evaluation proposed in this paper, it requires minimum or no adjustment of parameters, which saves a significant amount of time and resources otherwise necessary for implementing model-based monitors for a given application.

We introduce Confidence-based Context-Aware condition Monitoring (CCAM) with the following main contributions:

(1) We propose a fuzzy logic-based confidence metric for the quality assessment of systems under monitoring,
(2) we demonstrate the feasibility of the proposed monitor using two case studies: an industrial motor and a water pipe system,
(3) we demonstrate that CCAM gives equally good or better results than the similar system (Götzinger et al. 2017b), which does not use a confidence metric, and
(4) we demonstrate the robustness of the system by providing a sensitivity analysis of parameter settings and by showing that CCAM with confidence is better in identifying the correct system health status compared with a system without confidence. This is particularly notable when detecting drift situations in both the AC motor and the water pipe case studies.

After reviewing relevant related work in Section 2, we introduce confidence and describe our use of fuzzy logic in Section 3. In Section 4 we describe our proposed model-free condition monitoring approach that uses confidence to identify system states correctly. Section 5 shows and discusses results from our case studies, and Section 6 concludes the paper.

2. Background and related work

Measurements and processing of sensory data are relevant in a wide range of domains, like industrial processes, environmental monitoring, and medical applications. In all these areas the quality of measurements and observations are critical, however, less studied. Recently this aspect has come into the focus of research by characterizing various aspects (TaheriNejad, Jantsch, and Pollreisz 2016) and by demonstrating concrete benefits of using a nuanced and more realistic approach to observation (Götzinger et al. 2016; Anzanpour et al. 2017; TaheriNejad, Shami, and Manoj 2017). Götzinger et al. (2016) proposed data confidence to improve the reliability of patient monitoring in medical applications and Götzinger et al. (2017a) show that the usage of the plausibility of sensor values and value changes improves the robustness of the assessment of the health condition of a person.

AC motors are widely used in various industrial applications, and monitoring their health status is of interest to the industrial sector. A wear-out or other malfunctions in a motor may result in severe cracks or breaks in the rotor, stator or bearings, and can finally lead to reduced performance or – even worse – a failure (Ballal et al. 2007). Given the associated cost of these problems, non-invasive fault detection and preventative maintenance are important concerns in the industry (Gao, Cecati, and Ding 2015). It is known that during normal operation, the nominal range of outputs (e.g. current and speed) follows the nominal range of inputs (e.g. frequency or voltage). However, when the motor is free running in the presence of wearing out phenomena, some output signals deviate from the expected nominal values while the inputs remain unchanged. This can be used to detect wear-outs and certain other faults and failures. Various methods have been applied to the analysis of motor signals to detect faults (Kande et al. 2017). These monitoring systems mostly use methods such as current analysis (Féki, Clerc, and Velex 2013), temperature monitoring (Gao, Habetler, and Harley 2005), and vibration and noise analysis (Bellini et al. 2001); and they apply techniques such as hidden Markov modelling (Hatzipantelis and Penman 1993), thresholds (Mehala 2010), pattern recognition and neural networks (Bazan et al. 2017).

Condition monitoring in HVAC systems serves not only for fault detection, but also for optimizing control in order to reduce energy consumption (Massieh 2010). More generally, for the monitoring of water pipes, a wide variety of mostly distributed sensing principles is in use depending on the spatial extension of the network (Sadeghioon et al. 2018). For economic reasons, however, many monitoring systems rely on pressure and flow monitoring (Mounce et al. 2015), an approach we also apply in our case study in that we monitor the water flow.

With respect to data analysis, different methods have been proposed used for automated fault detection and diagnostic (FDD). These methods can be subdivided into quantitative model-based, qualitative model-based and process history-based methods (Katipamula and Brambley 2005). Process history methods are further differentiated in black-box and grey-box methods, and within the black-box methods, we distinguish between statistical, artificial neural network and other pattern recognition techniques. For example, Hyvrinen and Kärki (1996) present a fuzzy model for fault detection in HVAC systems. However, most of these methods require application-specific assumptions or models. Shun and Wen (2014) use Principal Component Analysis (PCA) and combine it with a Pattern Matching method to correctly identify the behaviour of the system without any additional knowledge of the system. While their work uses PCA's as well as distance similarity factors, we use confidence only based on fuzzy functions to detect the behaviour of the system correctly.

One generic method without detailed assumptions or a model of the system under observation is Context-aware Health Monitoring (CAH) (Götzinger et al. 2017b), which has been tested for monitoring industrial AC motors. CAH is a monitoring system that recognizes normal state changes as well as misbehaviour of the observed system. It accomplishes this task without a priori knowledge, and only through contextual information. However, two assumptions exist: first, the observed black box is in a steady state. Second, it is a *bijective function*, meaning that a unique input data set corresponds to one and only one output data set – and vice-versa. Thus, a change of the input inevitably is reflected in a change of the output and vice-versa. Otherwise, the observed system works incorrectly. We use CAH as the basis of our design and improve it using the concept of confidence to offset some of its major drawbacks, which are as follows: (i) A state-

full system, where current outputs depend on current inputs and the internal state of the system, cannot be reliably assessed; (ii) The signals of the observed system have to be smoothed through a filter; (iii) Despite filtering, the system performance is very sensitive to changes in signal values, e.g. due to normal transitions, noise, or instabilities; (iv) The thresholds used in CAH have to be set meticulously and accurately, otherwise, the system may not perform as expected, and hence, for each application tedious adjustment and tuning are necessary. Our proposed system addresses shortcomings (ii), (iii), and (iv), but shares the limitation (i) with CAH, as CCAM also can reliably assess only stateless systems.

3. Confidence

3.1. Definition of confidence

Confidence is a measure of the reliability of a system, a function, an analysis, or a process. Confidence can be defined as 'the extent to which a procedure may yield the same results on repeated trials' (TaheriNejad, Jantsch, and Pollreisz 2016). Hence, confidence can improve the self-awareness of a system regarding its subsystems and functions, and to what extent it can rely on the result each of them produces (Götzinger et al. 2016; Kholerdi, TaheriNejad, and Jantsch 2018).

To create a better and universal understanding of this concept, we formalize this definition in the following: Let us assume that f is an ideal function defined over $X = \langle x_0, \ldots, x_n \rangle$ and g is the unideal function at hand, also defined over X. We define the confidence of $g(x_i), x_i \in X$ as the inverse of a distance function $\Delta(g(x.), f(.))$, which captures the distance[1] between f and g based on some application specific distance metric:

$$c(g(x_i)) = \frac{1}{\Delta(f(x_i), g(x_i))}. \tag{1}$$

The overall confidence of g (i.e. confidence of the function/system in general, as opposed to its confidence at each point) is hence calculated as the average

$$C(g) = \frac{1}{n}\sum_{i=0}^{n} c(g(x_i)). \tag{2}$$

We note that $0 \leq c(g), C(g) \leq 1 = c(f) = C(f)$.

The implementation of a method to calculate c, however, is case dependent. In many cases, the ground truth (f) is not available and therefore, the distance cannot be calculated. In consequence, a function is often devised to estimate Δ. In the rest of this paper, we define and use heuristics as confidence functions without further reference to a distance metric.

3.2. Association assessment and confidence

As described in Section 2, the CAH system assumes to monitor a bijective function and hence, the relationship between the input and output data sets. Therefore, one of the main tasks is to assess the relationship of new data with previously observed data. For example, whether a new sample fits any of the recorded data in the history of the system or not. Thus, it is critical to make correct decisions as they affect the system performance, and it is important to know the confidence with which such decisions are made.

Let i be a variable of interest, for example, the voltage from a sensor or an abstracted observation after preprocessing. Further, let $v_{h_{i,j}}$ be the j^{th} value in the history of that variable h_i (i.e. j constitutes the position of the value in the history), and let $v_{i,new}$ be a new incoming value of this variable. Our task now is to assess whether the new sample belongs to the same group as the old sample(s) in the variable's history, and to provide a confidence value for our assessment. We use relative distance[2]

$$d_{i,j} = \left| \frac{v_{i,new} - v_{h_{i,j}}}{v_{i,new}} \right| \tag{3}$$

as a metric for this assessment and for computing the following confidence values.

Two different properties define how well a new value $v_{i,new}$ matches an existing data set: how many values of the existing data set are close to $v_{i,new}$, and how close they are. For the latter, inspired by fuzzy logic, we define four points and three intervals[3] for determining the membership of the new value (see Figure 1(a)). If the new value $v_{i,new}$ is in the interval $[d_b, d_c]$, it belongs to the same set of data as $v_{h_{i,j}}$ with certainty. If it falls in one of the intervals $[d_a, d_b]$ or $[d_c, d_d]$, it may or may not belong to that group; otherwise, it does not. Then, we define c_{sv} ('similar value') as the confidence of $v_{i,new}$ belonging to the same data set (group of values) as $v_{h_{i,j}}$, and we compute it as follows

$$c_{sv,i,j} = \begin{cases} \frac{d_a - d_{i,j}}{d_b - d_a} & \text{if } d_a < d_{i,j} < d_b \\ 1 & \text{if } d_b \leq d_{i,j} \leq d_c \\ \frac{d_d - d_{i,j}}{d_d - d_c} & \text{if } d_c < d_{i,j} < d_d \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

where the relative distance is calculated by Equation (3).

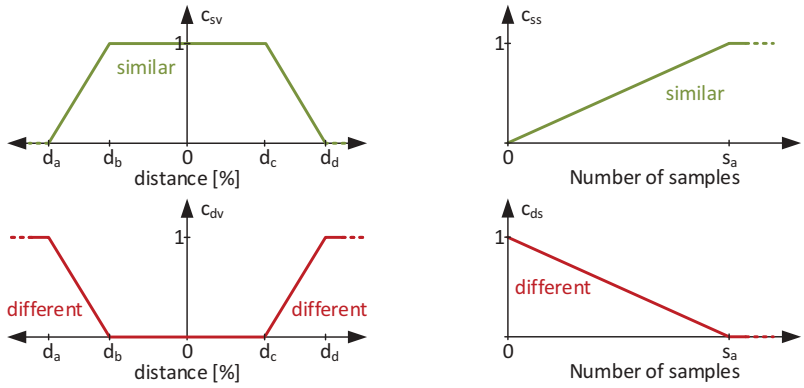
The counterpart of c_{sv} is c_{dv} ("different value"), the confidence of not belonging to the same data set, and we compute it as follows

$$c_{dv,i,j} = \begin{cases} \frac{d_b - d_{i,j}}{d_b - d_a} & \text{if } d_a < d_{i,j} < d_b \\ 0 & \text{if } d_b \leq d_{i,j} \leq d_c \\ \frac{d_{i,j} - d_c}{d_d - d_c} & \text{if } d_c < d_{i,j} < d_d \\ 1 & \text{otherwise.} \end{cases} \tag{5}$$

where the relative distance $d_{i,j}$ is also calculated by Equation (3).

If the intervals of the functions c_{sv} and c_{dv} are identical, the computation of c_{dv} can be simplified to $c_{dv,i,j} = 1 - c_{sv,i,j}$.

The other factor, which determines whether a new value $v_{i,new}$ matches an existing data set, is the number of values in the existing data set that are close to the new value. The more samples in close proximity, the likelier the new data fits the existing data set. Therefore, there are fuzzy functions such as the ones shown in Figure 1(b), in which, if a new sample $v_{i,new}$ does not match any of the existing data, the confidence $c_{ss,i}$ of $v_{i,new}$ being a member of the existing data set is 0, and the confidence $c_{ds,i}$ of $v_{i,new}$ not being a member of the existing data set is 1. If there are at least s_a samples in the vicinity of the new data, the new sample certainly fits the existing data set ($c_{ss} = 1$ and $c_{ds} = 0$). The function



(a) Confidences of a new sample being similar (c_{sv}) and being different (c_{dv}) to an existing datum, based on the of distance between them.

(b) Confidences of a new sample being similar (c_{ss}) and being different (c_{ds}) to an existing data set, based on the number of samples.

Figure 1.: Fuzzy functions showing the confidence of the system for considering a new sample fitting an already existing data set.

$$c_{ss,i,k} = \begin{cases} 1 & \text{if } k \geq s_a \\ \frac{k}{s_a} & \text{if } 0 \leq k < s_a \end{cases} \quad (6)$$

estimates the confidence for a new sample belonging to the existing data set, and

$$c_{ds,i,k} = \begin{cases} 0 & \text{if } k \geq s_a \\ \frac{s_a - k}{s_a} & \text{if } 0 \leq k < s_a \end{cases} \quad (7)$$

estimates with what confidence the new sample does not belong to the existing data set.

If the intervals of the functions $c_{ss,i,k}$ and $c_{ds,i,k}$ are identical, the computation of $c_{ds,i,k}$ can be simplified to $c_{ds,i,k} = 1 - c_{ss,i,k}$.

To make a final decision on whether the new value, $v_{i,new}$, belongs to the existing data set, the two factors (the confidences $c_{sv,i,j}$ and $c_{ss,i,k}$) above have to be combined, leading to the overall confidence $c_{b,i}$. Hence, $c_{b,i}$ (for $v_{i,new}$ belonging to an existing data set) is composed of the confidences $c_{sv,i,1} \ldots c_{sv,i,k}$ (Equation (4)) calculated by the comparisons of $v_{i,new}$ with k values of the existing data set, and the confidence $c_{ss,i,k}$ (Equation (6)). For this purpose, we propose to use the conjunction ($\wedge$) operator as follows

$$c_{b,i} = (c_{sv,i,1} \wedge \cdots \wedge c_{sv,i,k}) \wedge c_{ss,i,k} = (\bigwedge_{j=1}^{k} c_{sv,i,j}) \wedge c_{ss,i,k} \quad (8)$$

because the new sample $v_{i,new}$ belongs to the existing data set only if $v_{i,new}$ is sufficiently similar to all k values (and k should be a large enough number). A conjunction in fuzzy logic is equal to a minimum function (Ross 2009). Hence, this operation results in the minimum of all k calculated $c_{sv,i,j}$'s. Since the result of all k disjunctions is disjunctioned with $c_{ss,i,k}$, which is based on the number of comparisons, the confidence cannot be larger than $c_{ss,i,k}$.

The overall confidence of $v_{i,new}$ not belonging to an existing data set ($c_{n,i}$) is composed of all k confidences, $c_{dv,i,1} \ldots c_{dv,i,k}$, and $c_{ds,i}$. For this purpose, we propose to use the disjunction ($\vee$) operator as follows

$$c_{n,i} = (c_{dv,i,1} \vee \cdots \vee c_{dv,i,k}) \vee c_{ds,i,k} = (\bigvee_{j=1}^{k} c_{dv,i,j}) \vee c_{ds,i,k} \quad (9)$$

because the new sample $v_{i,new}$ does not belong to the existing data set if $v_{i,new}$ is different to at least one of the k values or the number of values k is too low. A disjunction in fuzzy logic is equal to a maximum function (Ross 2009). The result of this computation is equal to the maximum of all k $c_{dv,i,j}$, limited by $c_{ds,i,k}$ which is based on the number of comparisons.

In a last step, to determine the chances of the new sample belonging to the existing data set, we compare the calculated confidences. Only if c_b is larger than c_n, we declare the new value as belonging to the existing data set.[4]

4. Confidence-based context-aware condition monitoring

In this section, we present the details of the proposed CCAM system, in particular, the fuzzy operations. Thanks to the benefits of the fuzzy logic-based confidence concept, in CCAM there is no need for filtering, except for one case in the motor case study. In Section 5.4, we show this in detail. Note that the task of pre-processing is application specific and may differ from case to case. Some signals are not directly usable by the CCAM system because CCAM works only with steady states. Alternating signals, such as AC current and AC voltage will never be in a steady state. Hence, the pre-processing unit of the proposed system (shown in the green frame in Figure 2) abstracts the alternating signal into a non-alternating form. In the rest of this section, we first present a brief generic description of the state handler, and then move to part showing how different states and potential malfunctions are recognized. We end this section by presenting how the overall confidence of the system regarding its recognition is assessed.

4.1. State handler

The heart of the CCAM system is the State Handler (SH) which is shown in the blue frame in Figure 2. The SH detects when the System under Observation (SuO) i) is in a steady state, ii) changes its state, or iii) is not working correctly. For this reason, the SH saves information about these SuO states in C++ objects

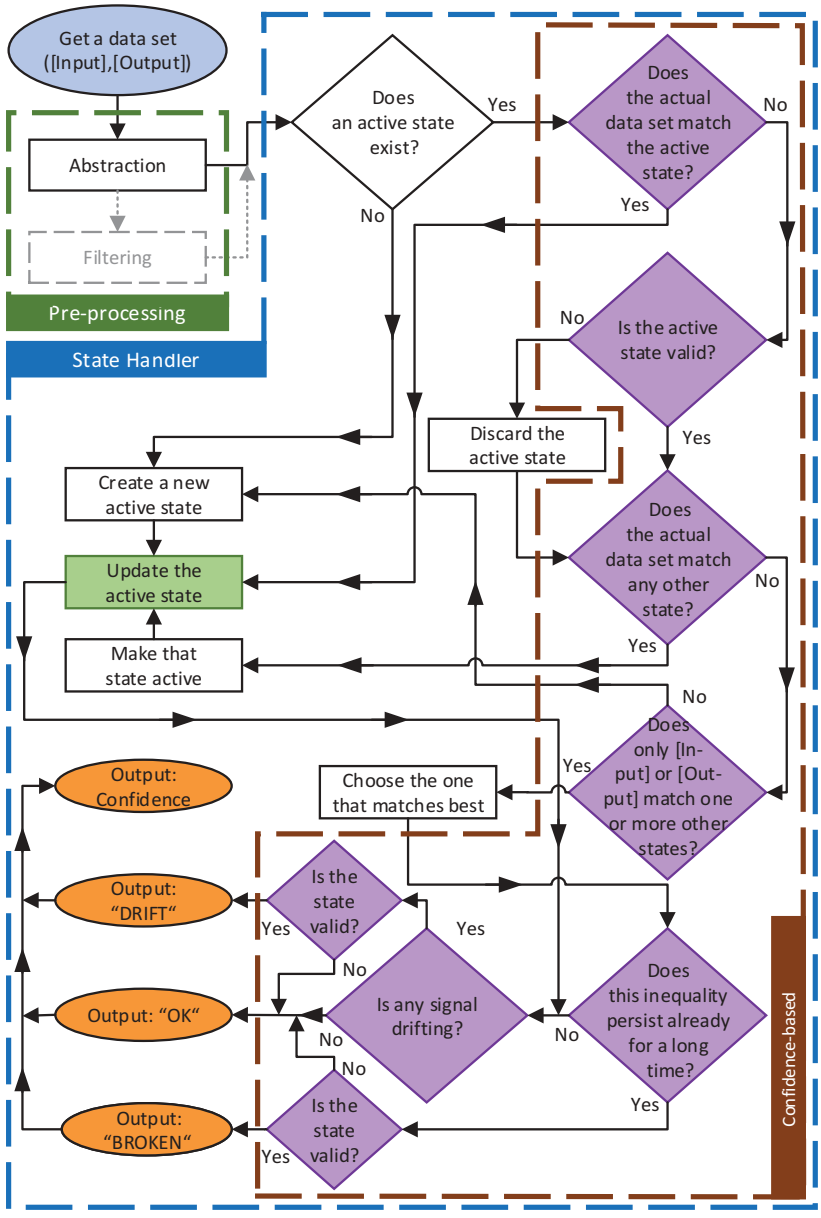


Figure 2. Flow chart of the CCAM system proposed here.

which are – for the sake of convenience – called states. These states contain a sliding history for each signal of the SuO. In the following, we describe the main tasks of this unit.

4.2. Recognizing states and detecting state changes

The SH (shown in the blue frame in Figure 2) saves information about every SuO state detected in a sliding history h. More precisely, one history exists for each variable i (input and output signals) of the SuO $\langle h_i \rangle$. Whether a state is valid or not is reflected by two confidence values: the confidence c_{val} which indicates whether a state is valid and the confidence c_{inv} when a state is invalid. Similar to the decision whether a new sample set fits a state (see Section 3.2), the less the deviations between the sample values saved in a state and the higher the number of sample sets in this state, the higher is the confidence of detecting a valid state. Conversely, large deviations between the samples or only a few samples inserted in the state indicate that the suspected state is most likely not a real state (e.g. a transient state). A state is considered as a valid state when c_{val} is higher than c_{inv}. In this case, the SH saves the respective state in the state vector. How these two confidence values are calculated is explained is detail in Section 4.4.

To observe regular state changes or an unwanted malfunction, the SH examines whether an incoming data set, consisting of the different signals $\langle v_{i,new} \rangle$ of the SuO, fits the actual state, termed active state (shaded in purple in Figure 2). Therefore, the SH calculates both, the confidence c_b (Equation (14)) for deciding that the new data set matches the active state and the confidence c_n (Equation (15)) which indicates that the new data set does not belong to the active state. Based on the confidence value, the SH decides whether the set of samples matches the active state.

How these two confidences are computed is shown in detail in Section 4.3). If c_b is higher than c_n, the data is inserted into the active state, i.e. the data is saved in the sliding window history of it. This observation is indicative of a well functioning system, unless the average value of any signal of the state is not drifting. How the SH observes a possible drifting of a signal is described in Section 4.6.

If the new data set mismatches the active state, a state change is identified. This can be either a normal state change or a malfunction. If both, input and output data sets mismatch the active state, a regular state change has happened. When the SH observes such a behaviour, the actual samples are compared to all states saved in the state vector. The state to which the samples are fitting is selected as the new active state. If the samples do not match any of the saved states, the SH creates a new active state. Regardless of whether the actual data set is compared with the active state or with any other state, the function of calculating the confidences c_b and c_n are the same. Afterwards, the actual samples are inserted into the new active state.

If exclusively one (either input or output) data set matches the actual state and the other one does not, a malfunction is detected, which is described further in Section 4.5.

4.3. Ascertaining a set of samples matches a state

To determine whether the new sample set matches a state, as a first step, a confidence value $c_{b,i}$ is computed for every new signal sample $v_{i,new}$ (e.g. voltage, current, etc.) by Equation (8). As described in Section 3.2, this confidence depends on both, the number of samples of the existing data in close proximity (see Equation (6)) of $v_{i,new}$ and how close they are (see Equation (4)). To compute the confidence whether the new sample set matches a state, the best fitting subset (size k) of history values is chosen. The new sample fits to the state if it is similar to a high number of history samples while at the same time the distance to all of the history samples is low. However, these two requirements can compete, particularly, in the case of history values with both lower and higher distances to the new sample. Furthermore, since belonging to a set is not determined by fixed thresholds, it is not trivial to decide whether the distance between the samples is close enough or how many samples have to be in each others' proximity to form a state. To find the best and – at the same time – the biggest matching subset of a variable's history, $\langle h_i \rangle$, all possible cases of comparisons are computed; from subset size 1 to n, whereas n is the size of the history. These computations, based on Equation (8), are computed as follows

$$\begin{aligned} \text{case } 1: \; c_{b_1,i} &= c_{sv,i,1} \wedge c_{ss,i,1} \\ \text{case } 2: \; c_{b_2,i} &= (c_{sv,i,1} \wedge c_{sv,i,2}) \wedge c_{ss,i,2} \\ &\vdots \\ \text{case n}: \; c_{b_n,i} &= (c_{sv,i,1} \wedge \cdots \wedge c_{sv,i,n}) \wedge c_{ss,i,n} \end{aligned} \tag{10}$$

where n denotes the size of the history (number of samples saved in the history), and $c_{sv,i,j} \geq c_{sv,i,k}; \forall j \leq k$.

Next, the SH chooses the best possible case with a subset of size k of Equation (10) (corresponding to Equation (8)) for $v_{i,new}$ belonging to the data set $\langle v_{h_i} \rangle$ as follows

$$c_{b,i}(v_{i,new} \in \langle v_{h_i} \rangle) = \bigvee_{j=1}^{n} c_{b_j,i}. \tag{11}$$

The confidence of not belonging ($c_{n,i}$) is also calculated in a similar fashion, but by using the disjunction operator for all n cases. That is,

$$\begin{aligned} \text{case } 1: \; c_{n_1,i} &= c_{dv,i,1} \vee c_{ds,i,1} \\ \text{case } 2: \; c_{n_2,i} &= (c_{dv,i,1} \vee c_{dv,i,2}) \vee c_{ds,i,2} \\ &\vdots \\ \text{case n}: \; c_{n_n,i} &= (c_{dv,i,1} \vee \cdots \vee c_{dv,i,n}) \vee c_{ds,i,n}. \end{aligned} \tag{12}$$

where n denotes the size of the history (number of samples saved in the history), and $c_{dv,i,j} \geq c_{dv,i,k}; \forall j \leq k$.

Finally, the lowest confidence with which we can consider $v_{i,new}$ not belonging to $\langle v_{h_i} \rangle$ is calculated using

$$c_{n,i}(v_{i,new} \ni \langle v_{h_i} \rangle) = \bigwedge_{j=1}^{n} c_{n_j,i}. \tag{13}$$

After the confidences $c_{b,i}$ and $c_{n,i}$ have been calculated for every variable i, the confidence of the whole sample set belonging to the considered state, c_b, is calculated by

$$c_b = \bigwedge_{i=1}^{m} c_{b,i} \tag{14}$$

because a new data set belongs to a given state only if the new samples of all variables match previous values of respective variables in that state. Its counterpart, the confidence of the whole sample set not belonging to the considered state, c_n, is calculated by

$$c_n = \bigvee_{i=1}^{m} c_{n,i} \tag{15}$$

because the data set does not belong to a state if one or more samples do not match the existing values of the variables of that state. In both equations m is the number of variables.[5]

We note that the confidence functions in Equations (6) and (7) are adaptable in a way that the boundary s_a is equal to the number of the already saved values in the state, bounded by the history length n. Without this adaptive property, due to Equations (11) and (13), the confidence c_n for each variable would be inevitably higher than c_b for any new state with few recorded values.

4.4. Validate a state

When an actual signal sample is added to a state, it is compared with all history values of this signal, saved in the state (Figure 3). From all these comparisons both, the confidences $c_{sl,i}$ (lowest $c_{sv,i,j}$) for being similar and $c_{dh,i}$ (highest $c_{dv,i,j}$) for being different are saved in the history in addition to the actual value. That is,

$$c_{sl,i} = \bigwedge_{j=1}^{n} c_{sv,i,j} \tag{16}$$

and

$$c_{dh,i} = \bigvee_{j=1}^{n} c_{dv,i,j} \tag{17}$$

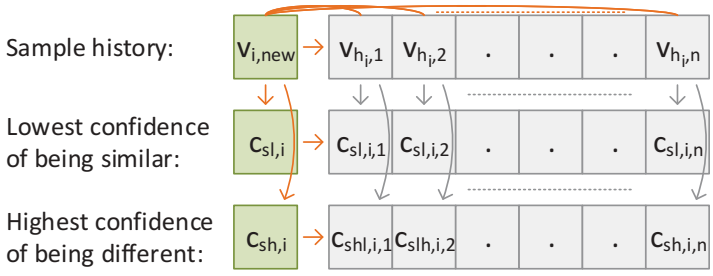


Figure 3. A graphic demonstration of a new value being compared to the values in the history. This figure shows also the storage of respective confidences of belonging and not belonging.

where n is the size of the history.

To validate the state, the confidences c_{val} of having found a valid state is calculated by

$$c_{val} = \left(\bigwedge_{i=1}^{m} c_{sl,i}\right) \wedge c_{ss,i,n} \tag{18}$$

because a state is valid if it contains many samples, and all of them are similar to each other. Its counterpart c_{inv} is calculated by

$$c_{inv} = \left(\bigvee_{i=1}^{m} c_{dh,i}\right) \vee c_{ds,i,n} \tag{19}$$

because a state is invalid if it contains only a few samples, or at least one of these samples is different. In both equations, m is the number of variables and n is the size of the history. The confidences $c_{ss,i}$ and $c_{ds,i}$ are calculated by Equation (6) and Equation (7). These confidences depend on the number of comparisons (i.e. length of the history), but it makes no difference which variable i is taken for $c_{ss,i,n}$ (Equation (18)) and $c_{ds,i,n}$ (Equation (19)) because the history of every signal has the same length. This algorithm is similar to evaluating whether a sample set belongs to a state (in Section 4.3) with the difference that the comparison is done with all of the history values and not only with a subset of them (i.e. k samples). A state is only valid if the confidence c_{val} is higher than its counterpart c_{inv}, whether both confidences are low or high.

We note that the historical information is not limited only the n values in the history. Some information of the previous values are saved in a more abstracted form. The reason being that the confidences of older history values contain information also about history values that are already out of the sliding window.

4.5. Recognizing a malfunction

Since the monitored system is treated as a bijective function, a change of only one data set (input or output exclusively) can indicate an anomaly. If the input and output data sets are or are not belonging to the active state is separately calculated through Equations (14) and (15). Since different systems show different delays to reflect a change in the output due to an input change, a small time gap between the two shall be allowed. In other words, if the other (unchanged) data set follows within a short time, the system still works correctly. Therefore, two other confidences c_{brk} and c_{ok} are calculated similarly using

$$c_{brk} = \begin{cases} 1 & \text{if } s_t \geq s_a \\ \frac{s_t}{s_a} & \text{if } 0 \leq s_t < s_a \end{cases} \tag{20}$$

and

$$c_{ok} = \begin{cases} 0 & \text{if } s_t \geq s_a \\ \frac{s_a - s_t}{s_a} & \text{if } 0 \leq s_t < s_a \end{cases} \tag{21}$$

where $s_t \in \mathbb{Z}_{\geq 0}$ is the time gap (in samples) between the change of the two data sets (input and output), thus, the time which the output needs to react on a change in the input. The more time elapses, the more likely the SuO is broken; reflected in c_{brk} is becoming higher and c_{ok} is becoming lower. In this case, CCAM signals that the SuO is broken but only if the actual state is valid. Otherwise, this discrepancy is most likely because the SuO is in a transient state. In this case, the SH just discards the active state, creates a new one and saves the actual samples in the new state.

We note that if the SuO changes back into an already known state, $c_{ss,i,n}$ of Equation (18) and $c_{ds,i,n}$ of Equation (19) denote the number of samples which were inserted into the state after re-entrance into it. This is necessary because the signals may still be unsteady after a change and cause a wrong recognition of the state. Therefore, the new active state should not be considered as valid directly after re-entrance. Once there have been enough number of samples similar to an already existing state, we can recognize that state as reactivated.

If the intervals of the functions c_{brk} and c_{ok} are identical, the computation of c_{ok} can be simplified to $c_{ok} = 1 - c_{brk}$ and the condition $c_{brk} > c_{ok}$ becomes equal to $c_{brk} > 0.5$.

4.6. Recognizing a signal drift

A system under observation can have another condition besides broken or healthy. When one or more signals are drifting (changes continuously but very slowly) characterizes another abnormal operation. In other words, a series of values of a signal belong to the same state, but the signal is gradually deviating outside its normal expected range. This very slow change is indistinguishable in the sliding history window of the state because the samples saved in the history change as the signal is drifting. Therefore, the task of inserting sample values to the active state is more complex than just saving them in the history (shaded in green in Figure 2). The SH additionally calculates Discrete Average Blocks (DABs) the average values of a certain number of sample values (DAB_{size}) inserted in the state (Figure 4). The more different the actual

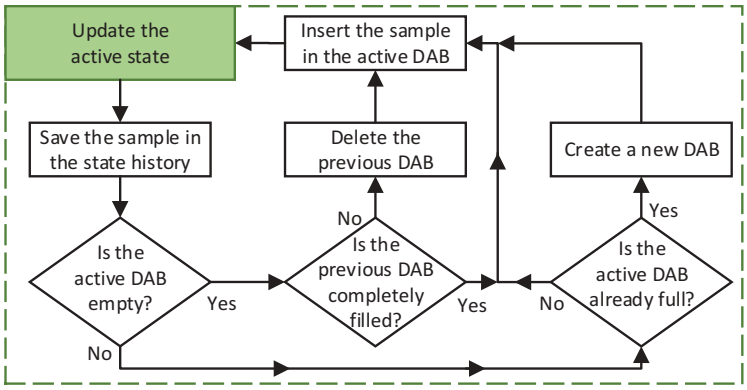


Figure 4. Block diagram of the state updating task of the proposed CCAM system. Discrete Average Blocks (DABs) are also created and kept in this procedure.

DAB to the first DAB, the more likely a signal is drifting. However, only completed DABs are compared with each other. An incomplete DAB could lead to a false result because of an outlier sample. The determination whether a signal is drifting happens after the new sample set was inserted into the active state. The confidences c_{dft} and c_{stb}, which indicates that the signal is drifting, respectively, stable, are calculated by

$$c_{dft} = \begin{cases} \frac{d_b - d_{dft}}{d_b - d_a} & \text{if } d_a < d_{dft} < d_b \\ 0 & \text{if } d_b \leq d_{i,j} \leq d_c \\ \frac{d_{dft} - d_c}{d_d - d_c} & \text{if } d_c < d_{dft} < d_d \\ 1 & \text{otherwise.} \end{cases} \tag{22}$$

and

$$c_{stb} = \begin{cases} \frac{d_{dft} - d_a}{d_b - d_a} & \text{if } d_a < d_{dft} < d_b \\ 1 & \text{if } d_b \leq d_{dft} \leq d_c \\ \frac{d_d - d_{dft}}{d_d - d_c} & \text{if } d_c < d_{dft} < d_d \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

where d_{dft} is the relative distance between the two compared DABs. That is,

$$d_{dft} = \left| \frac{v_{avg,DAB_{first}} - v_{avg,DAB_{new}}}{v_{avg,DAB_{first}}} \right| \tag{24}$$

where $v_{avg,DAB_{new}}$ is the average value of the latest (completed) DAB, and $v_{avg,DAB_{first}}$ is the average value of the state's first DAB.

If d_{dft} is higher or equal than d_{stb}, the SH of CCAM signals that the active state is drifting but only if the actual state is valid. Otherwise, this discrepancy is most likely because the SuO is in a transient state.

If the intervals of the functions c_{dft} and c_{stb} are identical, the computation of c_{stb} can be simplified to $c_{stb} = 1 - c_{dft}$ and the condition $c_{dft} \geq c_{stb}$ becomes equal to $c_{dft} \geq 0.5$.

4.7. Overall confidence of the CCAM system

The CCAM system outputs, besides the assessed health condition of the SuO, how confident it is about its assessment. This confidence c is case dependent and calculated by

$$c = \begin{cases} (c_{n,in} \vee c_{n,out}) \wedge c_{brk} \wedge c_{val} & \text{; if broken } (25) \\ (c_{b,in} \wedge c_{b,out}) \wedge c_{dft} \wedge c_{val} & \text{; if drifting } (26) \\ ((c_{b,in} \wedge c_{b,out}) \vee (c_{n,in} \wedge c_{n,out})) & \\ \quad \wedge c_{ok} \wedge c_{stb} \wedge c_{val} & \text{; if normal } (27) \end{cases}$$

where $c_{b,in}$, $c_{b,out}$, $c_{n,in}$, and $c_{n,out}$ are calculated by Equations (14) and (15) for input and output variables, respectively.

5. Evaluation

In this section, two case studies are investigated: an AC Motor and a water pipe system. First, we describe data and experimental setup used for testing of our proposed CCAM system. It is noteworthy that the configuration of CCAM is the same for both case studies. That is, a down sampling rate of 50, $[d_a, d_b, d_c, d_d] = [-14\%, -1\%, 1\%, 14\%]$ (Equation (4)), $s_a = 10$ (Equation (6)), $DAB_{size} = 10$, $[d_a, d_b, d_c, d_d] = [-30\%, -10\%, 10\%, 30\%]$ (Equation (22)), and $s_a = 20$ (Equation (20)). Next, we discuss the results and compare them with CAH (Götzinger et al. 2017b). The results of our sensitivity analysis, where we evaluate the effect of variations in configuration parameters on the performance of the system, is also presented.

5.1. Motor case study

We have used the simulation results from (Götzinger et al. 2017b) for comparison with our simulations of CCAM. Free running, change of speed and change of load behaviours have been simulated to cover normal states and state changes. For abnormal behaviour, the drift has been modelled applying an increase to the mechanical torque. The broken state, instead, has been provided by Bessous et al. (2018) from a real experiment on a motor with broken bearings. To evaluate the system more extensively, we added longer and more complex test scenarios simulated with the same simulation tools we used in Götzinger et al. (2017b). The motor is a single squirrel-cage, three-phase, 380 V, 50 Hz, 3 kW induction motor with four poles whose parameters are based on asynchronous machine model using the SI dialogue box in MATLAB®. The input and output signals are voltage, current, torque, and speed. For the broken state, the vibration signals have been collected in addition.

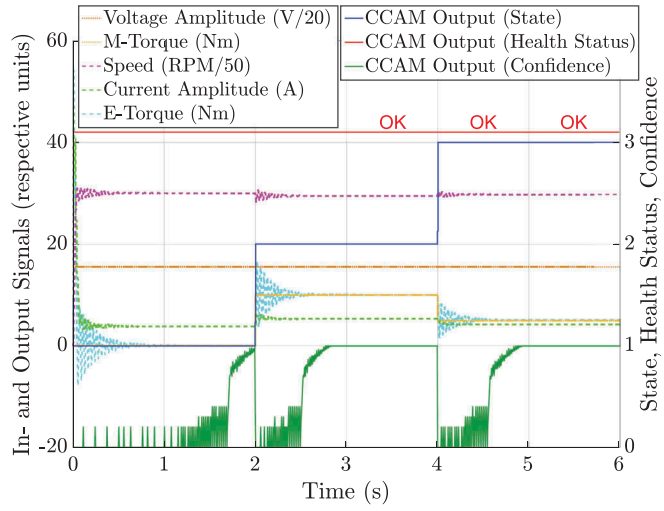
The CCAM system is modelled in C++ and fed with CSV-files containing data sets of the AC motor. The alternating signals such as AC voltage and AC current were abstracted beforehand to their amplitudes using MATLAB®. Table 1 outlines the various scenarios used in our AC motor experiments. In the following, we explain some example scenarios in more details and show how CCAM successfully classifies them.

5.1.1. Normal operation

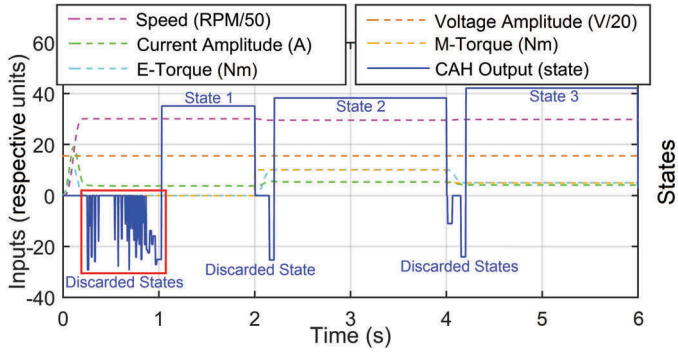
A normally working SuO can remain in a given state or change its state. We successfully tested the proposed CCAM system in various scenarios (Table 1). In the following, two of them will be discussed in detail. Figure 5(a) shows the scenario explained first (scenario 3 in Table 1). The motor was turned on, in the beginning, followed by two state changes caused by external load changes. The CCAM system continuously tries to identify the state of the system. When the motor is turned on, the output signals oscillate considerably for a relatively long time. Without detected state, it is not possible to determine whether only the input or the output data sets have changed. Because of this unsteady state, the CCAM system does not trust the system in the beginning of its first state, that is, the confidence is close to 0 ($c \approx 0$). Then, around 1.8 seconds the SuO settles to a steady state which is reflected in an increased confidence ($c \approx 1$). At 2*s*, the external load is changed from 0 to 10*Nm*. Therefore, all output signals start to change again; accompanied by an oscillating phase. Because of the relatively small change in the input signals, the period in which the SuO is unsteady is shorter. Around 0.5*s* after the external load has been changed, the CCAM system is confident about the recognition of the second state and the good health condition of the SuO. The same can be seen at 4*s* when the external load changes again. The scenario explained second (scenario 4 in Table 1) is similar but more complex. Figure 6 shows that the motor was turned on, followed by 21 times external load changes (with six different loads). CCAM detected all the six distinct states of the system and its 21 state changes.

Table 1. Various experiments (test scenarios) performed with an AC motor.

#	AC motor condition	Record length	Sampling frequency	Events in the scenario
1	OK	6 s	10 kHz	The motor was turned on in the beginning and remained without any state changes during the experiment.
2	OK	20 s	1 kHz	The motor was turned on in the beginning, followed by one state change caused by a change of input voltage and frequency.
3	OK	6 s	10 kHz	The motor was turned on in the beginning, followed by two state changes caused by external load changes. All three states differ from each other.
4	OK	60 s	1 kHz	The motor was turned on in the beginning, followed by 21 state changes caused by external load changes. This results in six different states for the motor.
5	OK	120 s	10 kHz	The motor was turned on in the beginning, followed by 51 state changes caused by external load changes. This results in seven different states for the motor.
6	OK	3600 s	1 kHz	The motor was turned on in the beginning, followed by 383 state changes caused by external load changes. This results in seven different states for the motor.
7	Drift	10 s	10 kHz	The motor was turned on in the beginning and remained without any state changes. Because of a wear-out of the motor signals are drifting very slowly.
8	Broken	10 s	12.5 kHz	The motor was abnormally running from the beginning without any state changes.



(a) Proposed CCAM system output during state changes. The legend shows the inputs of the SuO with dotted lines and the outputs with dashed lines.



(b) CAH system output during state changes (Götzinger et al. 2017b). The original plot does not reveal the numeric encoding of the state. Therefore, we labeled the states in the plot as states 1, 2, and 3. States shown as negative mean indicate that particular state was not saved during run-time.

Figure 5. Comparison of CCAM and CAH which does not consider confidence.

5.1.2. Wear-out

As shown in Figure 7(a), because of the oscillating signals (in scenario 7 of Table 1), in the beginning, it takes the CCAM system around 1.8*s* until it is confident that it has recognized a steady state. Shortly afterwards, the confidence of the CCAM is again falling because a drift is detected. Around 2*s*, the CCAM system changes status and raises the drift alarm. At around 3*s*, the system is highly confident about this decision. This circumstance lasts up to the end of the experiment.

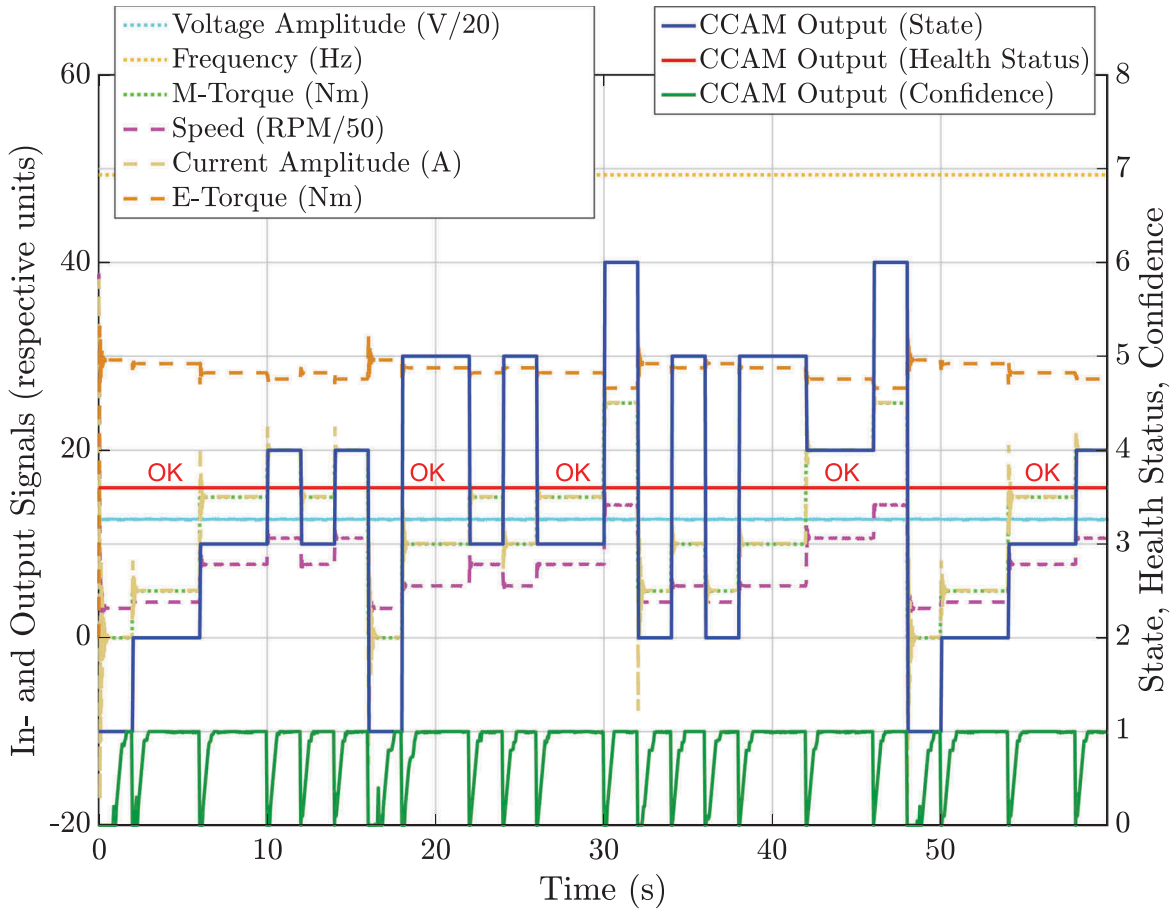
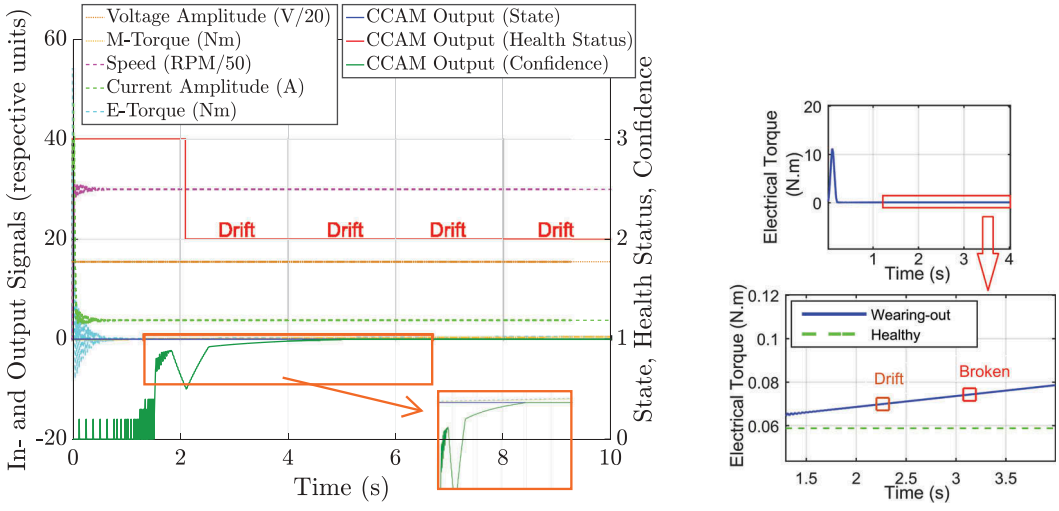


Figure 6. Proposed CCAM system output during many state changes of the AC motor. The legend shows the inputs of the SuO with dotted lines and the outputs with dashed lines.

5.1.3. Anomaly

In the case of a bearing defect, scenario 8 of Table 1 shown in Figure 8, the vibration and the current signals start to change significantly which leads to many peaks. These changes and peaks in the output signals of the SuO result in the CCAM system recognizing the SuO as broken. Due to the fact that the records of the broken motor lack data of the motor working well in the beginning, the CCAM system can find a state in this unsteady data



(a) Proposed CCAM system outputs during a drift. The legend shows the inputs of the SuO with dotted lines and the outputs with dashed lines.

(b) CAH system outputs during a drift (Götzinger et al. 2017b).

Figure 7. Monitoring drift in the AC motor case study. Comparison of CCAM and CAH.

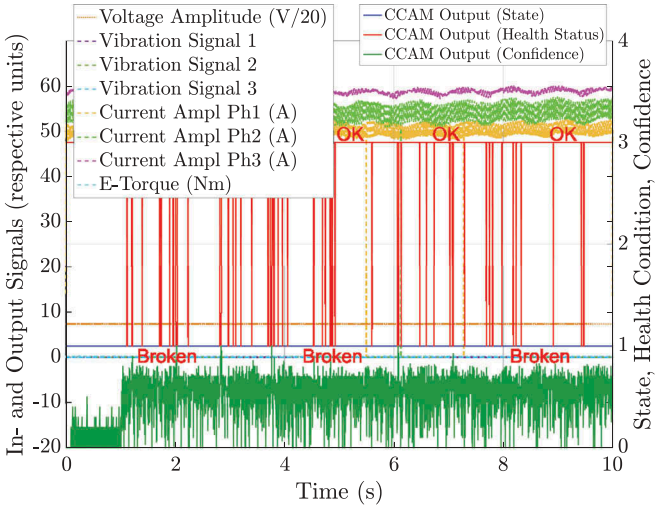


Figure 8. CCAM behaviour in the case of the bearing defect. Since the system is broken from the beginning, CCAM cannot identify a correct and stable state. Thus, it oscillates between broken and okay in its assessment and never converges. This oscillation represents the anomalous behaviour of the SuO. The legend shows the inputs of the SuO with dotted lines and the outputs with dashed lines.

(1*s*) and the mean of the confidence is rising. However, the output of CCAM changes between 'OK' and 'Broken' over and again, which signifies the broken condition. In future work, we will study if such unstable assessment could be automatically analyzed further and expressed explicitly and reliably as one broken alarm.

5.2. Water pipe system case study

In this case study, a pump is driving water through a pipe system. The system is controlled with a Raspberry Pi in combination with Arduino Uno. The desired velocities are set via a python script and all sensor values are logged. The input signal is a DAC output (normalized voltage) of the Raspberry Pi which ranges from 0.3 to 1.0, corresponding to the pump voltage range of 3 V to 10 V. The output signals are the water temperature as well as several volumetric flows. The temperature in the pipe system is measured with two temperature sensors (Pt100[6] at different positions) and the volumetric flow is measured with four flow sensors at different positions. The main part of the experimental setup is depicted in Figure 9. On the left side of this Figure, the sensors of Dynasonic and Riels are not displayed. Three different types of flow sensors are used to show that the system is able to work in a heterogeneous sensor system. The first type of sensors are in-situ ultrasonic sensors (Two Sharky FS 473 – SharkyS and SharkyB are their corresponding acronyms in the figures and in the rest of the paper). The second type is a clamp-on ultrasonic sensor (Dynasonics TFX Ultra), and the last one is another clamp-on ultrasonic sensor (Riels RIF600P). SharkyS and SharkyB are placed parallel to each other whereas the sensors Dyna and Riels are in series to each other and to the other two. The data of the Riels is not used for these experiments since the sensor was not set up correctly for this type of measurement. All measured values, actuator and sensor values are stored in CSV-files on the Raspberry Pi and later post-processed on another computer (because of the limited performance of the Raspberry Pi). Table 2 outlines several scenarios used in our experiments. In the following, we explain some of them in more details and show how CCAM classifies them correctly.

5.2.1. Normal operation

The CCAM system successfully detects different normal operation scenarios, with and without state changes. Two of these scenarios are discussed in detail in the following. Figure 10 shows scenario 3 of Table 2, where the voltage is increased two times after the water pump has been started. The temperature remains constant during the entire experiment. The valve before sensor SharkyS is closed, which means the sensors SharkyB and Dyna measure the entire water flow. The sensors SharkyS and Dyna need about 4 seconds of setup time until a steady state is reached allowing an accurate flow measurement. The CCAM system is able to detect all three states correctly. It needs 25–30 seconds to reach a high confidence in the state, and then remains high until the input is changed. As expected, the health status is also recognized as OK.

Figure 10(b) shows scenario 4 of Table 1 which is very similar to the previous scenario but the first and the last SuO states are the same. As we see in Figure 10(b) the voltage was increased first and then decreased again to its initial state. CCAM detected both state changes, recognized that the SuO

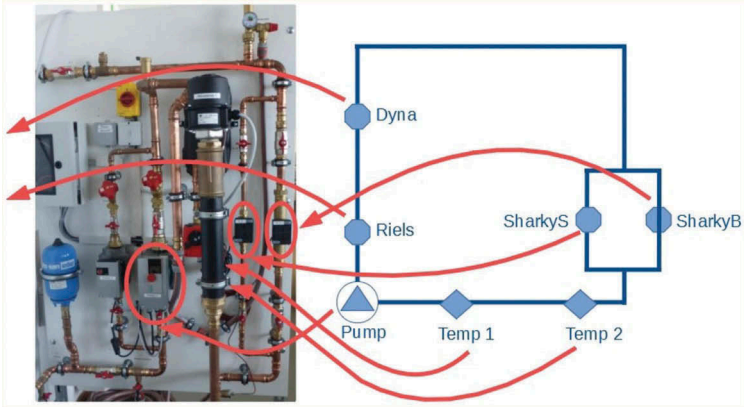


Figure 9. Parts of the water pipe system and schematics of the most important used elements. Dyna and Riels sensors are outside the picture (they would have been at the left side of the current frame of the picture) .

Table 2. Various experiments (test scenarios) using the water pipe system.

#	HVAC system condition	Record length	Sampling frequency	Events in the scenario
1	OK	649 s	30.5 Hz	The pump was started in the beginning and remained without any state changes during the experiment.
2	OK	625 s	30.5 Hz	The pump was started in the beginning, followed by one voltage increase, resulting in two different states during the experiment.
3	OK	627 s	30.5 Hz	The pump was started in the beginning, and afterwards, the voltage was increased two more times, resulting in three different states during the experiment.
4	OK	638 s	30.5 Hz	After the pump was started in the beginning, the voltage was increased once and changed back after some time. This resulted in two different states during the experiment.
5	Drift	629 s	30.5 Hz	The pump was started in the beginning and remained without any state changes during the experiment. However, signals start to drift in the middle of the experiment.
6	Broken	626 s	30.5 Hz	The pump was started in the beginning and remained without any input changes during the experiment. However, in the middle of the experiment a change happens (only) in the output, showing a broken system status.

changed again back to the first state, and classified the health status of the system correctly.

5.2.2. Wear-out

Figure 11(a) shows the input voltage of the pump is set to a constant value throughout the entire experiment. We note that the sensors SharkyS and Dyna deliver correct flow values only after about 4 seconds. After about 350 seconds the system starts to drift (mimicked by an opened valve that is not observed by CCAM) and both sensors detect a slowly rising water flow.

The CCAM system first detects correctly the stable state and then it detects the drifting behaviour of the system. The health status changes from OK to Drifting until the end of the experiment. The confidence in the state is rising again after the discovery of the deviation.

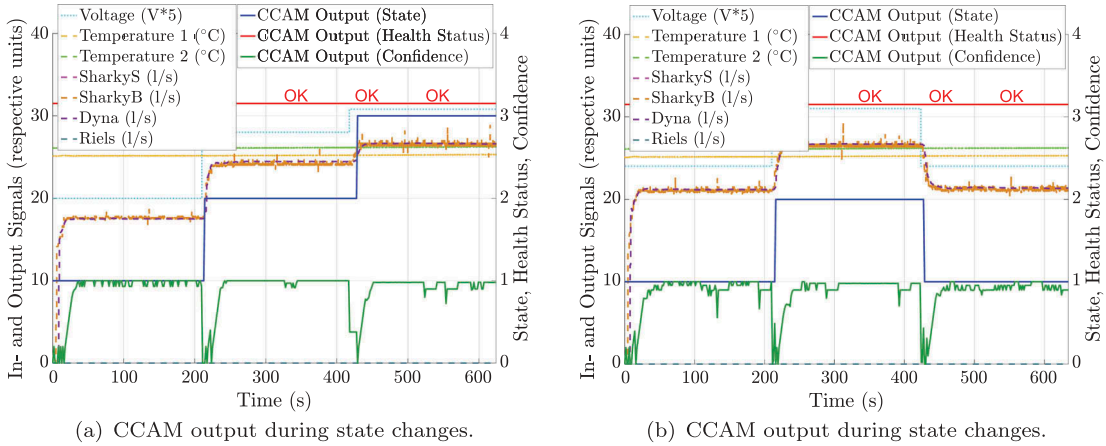
5.2.3. Anomaly

CCAM is also able to detect the break down of the system, and the results are shown in Figure 12. In this case, the pump voltage is set to a constant value for the entire experiment. At the beginning of the experiment, the entire water flow is in the pipe which is monitored from the sensor SharkyB but then the valve for this pipe is closed, and the valve of the pipe of sensor SharkyS is opened to simulate a hole in the other pipe.

In this case, the system detects the state, the confidence and the health status correctly at the beginning. After about 310 seconds it changes the health status to broken, which is the expected behaviour. The confidence drops significantly at the beginning of the broken state, and it needs about 30 seconds to return to a high confidence value.

5.3. Sensitivity analysis

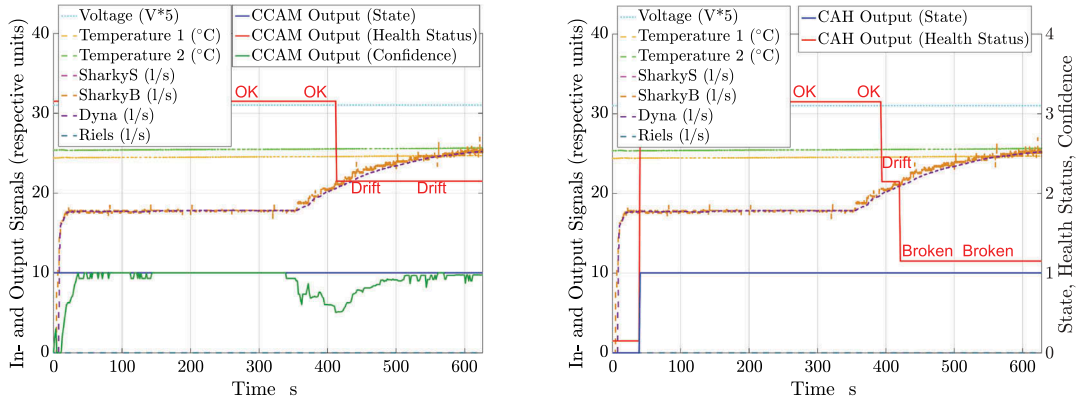
As mentioned, the proposed system benefits from an enhanced robustness of the recognition of states and a decreased dependence on parameter settings which reduces deployment time. To demonstrate these effects, we performed a sensitivity analysis where the impact of changing the value of certain parameters on the correct classification of the monitored system states is investigated. We studied the sensitivity of assessment results of CCAM on a set of data from the motor and water pipe system use cases. Tested parameters are the intervals of the fuzzy functions defined by Equation (4) and Equation (7), denoted by the points d_a, d_b, d_c, d_d, and s_a. In addition, various downsampling rates (DSR), DAB sizes and s_a from Equation (20) as well as d_a and d_c from Equation (22) are tested in the course of our sensitivity analysis. These parameters and their intervals are listed in Table 3. The high



(a) CCAM output during state changes.

(b) CCAM output during state changes.

Figure 10. CCAM output during state changes of the water pipe system. The legend shows the inputs of the SuO with dotted lines and the outputs with dashed lines.



(a) Proposed CCAM system outputs during a drift.

(b) CAH system outputs during a drift.

Figure 11. Monitoring of drifting phenomena by CCAM and CAH. The legend shows the inputs of the SuO with dotted lines and the outputs with dashed lines.

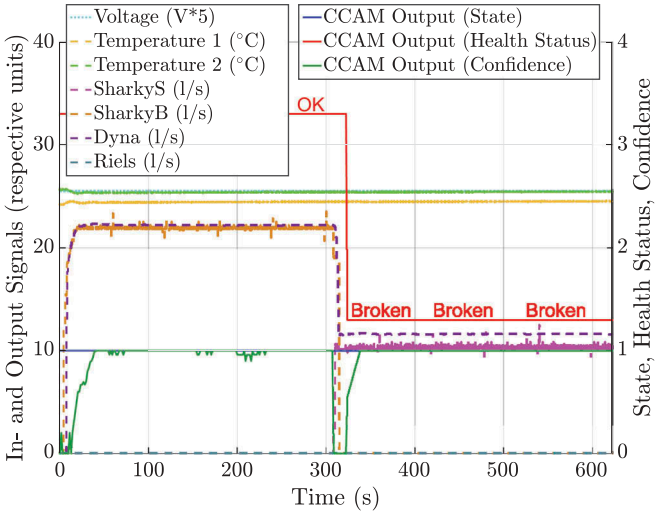


Figure 12. Proposed CCAM system outputs when observing the water pipe system showing anomalies. The legend shows the inputs of the SuO with dotted lines and the outputs with dashed lines.

number of tested values led to 2.2 million experiments for each recording which took about 20 days to run on two computers running in parallel on 32 and 24 cores.

Many configurations[7] led to results similar to those reported in the experiments above. More precisely, the classification of the well-being of SuO was correct, but the timing of the recognition and raising the alarms may differ to some extent. However, the focus was on the correct recognition rather than the time it takes before the states and their changes are detected.

Table 3. CCAM configuration parameter ranges used in the sensitivity analysis.

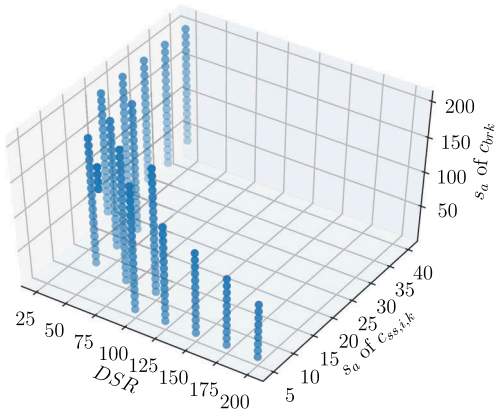
Parameter	Test Range	Test Steps	Working Range AC Motor	Working Range Water Pipe System
d_a of $c_{sv,i,j}$	[−20%, −2%]	1%	[−17%, −5%]	[−17%, −7%]
d_b of $c_{sv,i,j}$	[−10%, −1%]	1%	[−9%, −1%]	[−8%, −1%]
d_c of $c_{sv,i,j}$	[1%, 10%]	1%	[1%, 9%]	[1%, 8%]
d_d of $c_{sv,i,j}$	[2%, 20%]	1%	[5%, 17%]	[7%, 17%]
s_a of $c_{ss,i,k}$	[5, 40]	5	[5, 10]	[5, 40]
d_a of c_{dft}	[−50%, −20%]	10%	[−50%, −20%]	[−50%, −20%]
d_b of c_{dft}	10% const.	–	10%	10%
d_c of c_{dft}	10% const.	–	10%	10%
d_d of c_{dft}	[20%, 50%]	10%	[20%, 50%]	[20%, 50%]
s_a of c_{brk}	[10, 200]	10	[10, 200]	[10, 200]
DSR	[25, 200]	25	[50, 200]	[25, 200]
DAB_{size}	[5, 15]	5	[5, 15]	[5, 15]

It has to be noted that some of the parameters affect each other. Thus, not every theoretically possible combination of all these parameter values work. However, we found at least 1987 configurations which worked for all scenarios of both case studies. This includes our original configuration which was set – heuristically and with minimum effort – prior to this sensitivity analysis. In the following, we present and discuss the behavioural dependencies of various parameters based on the sensitivity analysis made for the water pipe system case study; that is, all scenarios of Table (2).

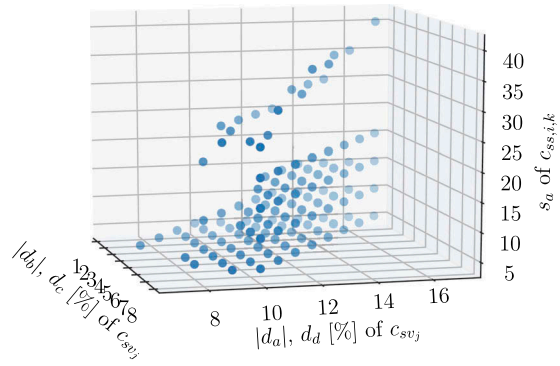
Figure 13 shows a small selection of the results of our sensitivity analysis. In these figures, each axis shows the range of the swept parameter and each dot shows the combination of those parameters which led to a correct classification (i.e. combination which did not function properly are not shown on the figure). In a closer look, in Figure 13(a) we observe that a lower Down-Sampling Rate (DSR) only works with higher s_a of $c_{ss,i,k}$, and a higher DSR needs the s_a of $c_{ss,i,k}$ to take a lower value. It can be also seen that s_a of $c_{ss,i,k}$ is rather independent of the DSR setup, whereas a higher DSR requires a lower s_a of c_{brk}.

Figure 13(b) shows that a lower s_a of $c_{ss,i,k}$ leads to a functional system with a larger set of values for the other two parameters, namely $|d_b|$, d_c of c_sv_j and $|d_a|$, d_d of c_{sv_j}. The relation between the latter two parameters is similar too; A lower $|d_b|$ and d_c of c_sv_j leads to a larger range of possible values for $|d_a|$ and d_d of c_{sv_j}. This is true the other way around too; lower $|d_a|$ and d_d of c_{sv_j} leads to a functional system with a larger set of configuration values for $|d_b|$ and d_c of c_sv_j.

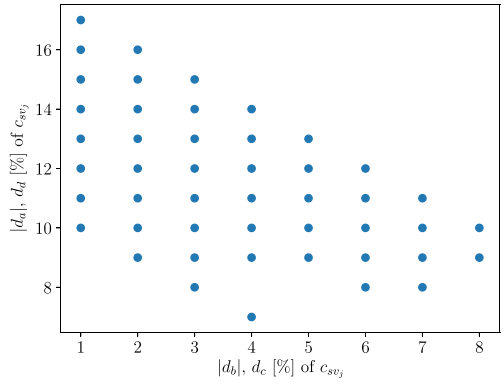
The most interesting finding, which clearly shows the advantage and importance of using fuzzy logic, can be seen in Figure 13(c). The lower $|d_b|$ and d_c of c_{sv_j}, the broader the range of acceptable $|d_a|$ and d_d of c_{sv_j}. Whereas, a higher value of $|d_b|$ and d_c of c_{sv_j} leads to the necessity of having a lower $|d_a|$ and d_d of c_{sv_j}. A low $|d_b|$ and d_c of c_{sv_j} in this context, constitutes a flat fuzzy membership function (see Figure 1(a)), whereas the opposite constitutes a steep fuzzy function. A steep fuzzy function



(a) Possible combinations of down sampling rate, s_a of $c_{ss,i,k}$, and s_a of c_{brk}, with those CCAM classifies the condition of SuO correctly.



(b) Possible combinations of $|d_a|$, $|d_b|$, d_c, and d_d of c_{sv_j}, s_a of $c_{ss,i,k}$, and s_a of c_{brk}, with those CCAM classifies the condition of SuO correctly.



(c) Possible combinations of $|d_a|$, $|d_b|$, d_c, and d_d of c_{sv_j}, with those CCAM classifies the condition of SuO correctly.

Figure 13. Sensitivity analysis for the different parameter combinations of CCAM. Each point on the charts represents a parameter combination which led to a correct classification of the SuO condition.

behaves very similar to a traditional threshold cut-off approach which we used in (Götzinger et al. 2017b). Therefore, the steeper is the fuzzy function (that is, more similar to a threshold function), the less flexible is the system (it functions with a smaller number of configuration values).

Based on the above analysis, we conclude that the system is in general very reliable, robust and not very sensitive to small changes of configurations with respect to those parameters studied above. In particular, the proposed method is more robust compared to CAH (Götzinger et al. 2017b) which does not use fuzzy logic or confidence.

5.4. Comparison with the context-aware health monitoring system

In this section, we compare our newly proposed CCAM system with CAH (Götzinger et al. 2017b) (which does not use confidence), under comparable conditions and using the same data.

Both, Figure 5(a) and 5(b) and 4 show the same scenarios (the same motor data) in which the external load changes. However, the curves of the motor signals appear different in the two figures. That is because Figure 5(a) shows unfiltered motor signals and Figure 5(b) shows filtered signals. This fact points to the first significant difference in the performance of the two systems. Whereas CAH needs the motor signals to be filtered in a preprocessing step, CCAM is less affected by signal instabilities and can handle unfiltered signals.[8] However, due to the unfiltered signal, CCAM requires more time than CAH to recognize the first state. While CAH recognizes the first valid state after 1*s* (Figure 5(b), CCAM needs around 1.8*s* for this. However, when CCAM analyses signals that are filtered, the first valid state is recognized after 1*s*.

The second difference is that CAH has a binary decision-making process based on fixed thresholds, whereas CCAM bases all decisions on confidences calculated with fuzzy functions. Therefore, the configuration of CCAM fuzzy parameters is less sensitive than the thresholds of the CAH system. In other words, the fuzzy functions increase the robustness. This partial independence of accurate adjustment also leads to a better recognition when the SuO drifts. We simulated 12 different wear-out scenarios with the deterioration rate of the speed signal from 0.0025 to 0.022 RPM. CAH detected the signal drift in 4 of these 12 cases (33% success) and switched after some time from drift- to broken alarm (as shown in Figure 7(b) too). In contrast, the CCAM system shows that the SuO is drifting in 100% of those cases and maintains its decision all

along (see Figure 7(a) as an example). The same unsteadiness of decision-making can be also seen when CAH monitors the water pipe system. Figure 11(b) shows that CAH detects a drift in the signals of SuO after $\approx$ 400 s. However, after again $\approx$ 25 s, CAH changes its decision and states that SuO is broken. In contrast to that, CCAM maintains its decision until the end.

The third major difference in performance is that whereas CCAM has been successfully tested for various down-sampling rates between 25 and 200, the CAH raises wrong alarms in some scenarios when the down-sampling factor is less than 50.

In summary, the results show that confidence, based on fuzzy functions, (i) simplifies the system, (ii) improves the quality of the system performance, and (iii) enhances its resilience.

6. Conclusions and future works

We present a Confidence-based Context-Aware condition Monitoring (CCAM) system which monitors the health condition of a bijective function black box system, without making further assumptions. In particular, CCAM is model-free and thus uses only contextual information to identify whether the observed system works correctly, is broken, or shows symptoms of wear-out. The advantages of such a system are the ease of deployment in a variety of applications, even those which are considerably different. The proposed system requires neither in-depth knowledge of the field, nor cumbersome effort to adjust the parameters to the given application. We are particularly pleased that CCAM – as we showed here – can monitor and assess proper working conditions of such diverse systems as a motor and water pipe system without model building or other customization.

The decision-making process of the proposed system uses confidence (which is a self-awareness property) computed using fuzzy logic. In our experiments, we ran a series of tests with and without confidence on an industrial AC motor as well as a water pipe system to show that the proposed system detects all of these behaviours correctly and more reliably while being simpler than a comparable monitor system without confidence. The proposed monitoring system is reliable as it does not depend on manually provided fixed threshold values; instead, it uses fuzzy set member functions that are robust against small variations of parameter settings.

In summary, in this paper,

(1) we introduced a fuzzy logic confidence metric for the condition monitoring of the state of a system,
(2) we demonstrated the feasibility of the proposed monitor using two case studies: an industrial motor and a water pipe system,
(3) we showed that the proposed system (CCAM) gives equally good or better results than the system without confidence (CAH) even though it does not pre-process the signals
(4) we demonstrated the robustness of our CCAM system by providing a sensitivity analysis showing that the system is robust against small variations of parameter settings.

6.1. Future work

In our experiments, we show that our proposed system is robust against small variations of parameter settings, and that it correctly detects/classifies state changes as well as the health status of the SuO. However, currently, CCAM uses a single fuzzy function for all signals of the SuO. This may lead to a wrong classification if the amplitude of the changes is significantly different among various input signals, output signals, or between input and output signals. For example, when a very small change in the input of the SuO causes a much larger change in the output. Therefore, in the future, we plan to have one fuzzy function setup for each signal of the SuO. However, that would lead to a larger number of parameters to tune. To circumvent a higher effort in setting up the parameters of CCAM, it is necessary that CCAM is able to learn from the signals themselves autonomously setup the fuzzy functions. This procedure will lead to an even easier and more effortless usage of CCAM.

In addition to that, we will study if an unstable assessment which we could see in the anomaly scenario of the AC motor could automatically be analyzed further and recognized explicitly and reliably as a single broken state. Detecting unsteady (transitory) states could be another important addition. Furthermore, CCAM should be able to detect patterns in case of a repeating sequence of state changes. A sudden change in a repeating sequence of SuO states could denote a malfunction too.

Notes

1. Distance means a difference which can be of any dimension such as performance or success rate difference, geometrical distance, time difference, difference in absolute values, and so forth.
2. We note that the relative distance, $d_{i,j}$, is not to be confused with Δ'. While $d_{i,j}$ is the distance of a sample to another sample, Δ' denotes the confidence that a membership assessment is correct.
3. There can be more than four points and three intervals. The fuzzy functions could be much more complex. However, the currently chosen shape has led to satisfactory results in our tests.
4. In fuzzy logic, the complement (negation) of a conjunction is not equal to a disjunction of complemented operands. Hence, in a generic case, $c_b \neq c_n$, although in some specific instances they may be equal.
5. We note that if input or output of the SuO is considered independently from each other, m constitutes the number of input and output variables, respectively.
6. 'Pt' stands for platinum and '100' shows the resistance at 0°C in ohms.
7. The smallest number of configurations which worked for all experiments of both case studies was 1987 sets.
8. We note that one scenario constitutes the exception: the anomaly (bearing defect). However, we believe that it would also work unfiltered if we had motor data which would show the motor working well in the beginning and before breaking. Unfortunately, we only have records of the broken motor, and therefore, the vibration signals are changing throughout the recording period. Thus, it is impossible for the SH to recognize a valid state, and the SH only raises an alarm if a valid state was found.

Acknowledgments

We gratefully acknowledge the financial support provided to us by the BMVIT/FFG under the program *Production of the Future* in the project SAVE (FFG 864883).

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the Bundesministerium für Verkehr, Innovation und Technologie [FFG 864883]; Österreichische Forschungsförderungsgesellschaft [FFG 864883].

References

Alexopoulos, K., S. Makris, V. Xanthakis, K. Sipsas, and G. Chryssolouris. 2016. "A Concept for Context-Aware Computing in Manufacturing: The White Goods Case." *International Journal of Computer Integrated Manufacturing* 29 (8): 839–849. doi:10.1080/0951192X.2015.1130257.

Anzanpour, A., I. Azimi, M. Götzinger, A. M. Rahmani, N. TaheriNejad, P. Liljeberg, A. Jantsch, and N. Dutt. 2017. "Self-Awareness in Remote Health Monitoring Systems Using Wearable Electronics." *Proceedings of Design and Test Europe Conference (DATE)*, Lausanne, March.

Ballal, M. S., Z. J. Khan, H. M. Suryawanshi, and R. L. Sonolikar. 2007. "Adaptive Neural Fuzzy Inference System for the Detection of Inter-Turn Insulation and Bearing Wear Faults in Induction Motor." *IEEE Transactions on Industrial Electronics* 54 (1): 250–258. doi:10.1109/TIE.2006.888789.

Bazan, G. H., P. R. Scalassara, W. Endo, A. Goedtel, W. Fontes Godoy, and R. H. C. Palácios. 2017. "Stator Fault Analysis of Three-Phase Induction Motors Using Information Measures and Artificial Neural Networks." *Electric Power Systems Research* 143: 347–356. doi:10.1016/j.epsr.2016.09.031.

Bellini, A., F. Filippetti, G. Franceschini, C. Tassoni, and G. B. Kliman. 2001. "Quantitative Evaluation of Induction Motor Broken Bars by Means of Electrical Signature Analysis." *IEEE Transactions on Industry Applications* 37 (5): 1248–1255. doi:10.1109/28.952499.

Bessous, N., S. E. Zouzou, W. Bentrah, S. Sbaa, and M. Sahraoui. 2018. "Diagnosis Of Bearing Defects in Induction Motors Using Discrete Wavelet Transform." *International Journal Of System Assurance Engineering and Management* 9 (2): 335–343. doi:10.1007/s13198-016-0459-6.

Féki, N., G. Clerc, and P. Velex. 2013. "Gear and Motor Fault Modeling and Detection Based on Motor Current Analysis." *Electric Power Systems Research* 95: 28–37. doi:10.1016/j.epsr.2012.08.002.

Gao, Z., C. Cecati, and S. X. Ding. 2015. "A Survey of Fault Diagnosis and Fault-Tolerant Techniques Part I: Fault Diagnosis with Model-Based and Signal-Based Approaches." *IEEE Transactions on Industrial Electronics* 62 (6): 3757–3767. doi:10.1109/TIE.2015.2417501.

Gao, Z., G. Habetler, and R. G. Harley. 2005. "An Online Adaptive Stator Winding Temperature Estimator Based on a Hybrid Thermal Model for Induction Machines." *IEEE International Conference on Electric Machines and Drives*, 754–761, San Antonio, TX, USA. doi:10.1109/IEMDC.2005.195807

Glatzl, T., H. Steiner, F. Kohl, T. Sauter, and F. Keplinger. 2016. "Development of an Air Flow Sensor for Heating, Ventilating, and Air Conditioning Systems Based on Printed Circuit Board Technology." *Sensors and Actuators A: Physical* 237: 1–8. doi:10.1016/j.sna.2015.11.016.

Götzinger, M., A. Azanpour, I. Azimi, N. Taherinejad, and A. M. Rahmani. 2017a. "Enhancing the Self-Aware Early Warning Score System through Fuzzified Data Reliability Assessment." In *International Conference on Wireless Mobile Communication and Healthcare*. Vienna, Austria: Springer. 10.1007/978-3-319-98551-0_1

Götzinger, M., A. M. Nima Taherinejad, P. L. Rahmani, A. Jantsch, and H. Tenhunen. 2016. "Enhancing the Early Warning Score System Using Data Confidence." *Proceedings of the 6th International Conference on Wireless Mobile Communication and Healthcare (Mobihealth), Milano*, Italy, November.

Götzinger, M., N. TaheriNejad, H. A. Kholerdi, and A. Jantsch. 2017b. "On the Design of Context-Aware Health Monitoring without a Priori Knowledge; an AC- Motor Case-Study." In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 1–5. Windsor, ON, Canada: IEEE. doi:10.1109/CCECE.2017.7946814

Hatzipantelis, E., and J. Penman. 1993. "The Use of Hidden Markov Models for Condition Monitoring Electrical Machines." In *Electrical Machines and Drives, 1993. Sixth International Conference On, September*, 91–96. Oxford, UK: IET.

Hyvrinen, J., and S. Kärki eds. 1996. "International Energy Agency Building Optimisation and Fault Diagnosis Source Book." *IEA ECBCS Annex 25l* .

Kande, M., A. J. Isaksson, R. Thottappillil, and N. Taylor. 2017. "Rotating Electrical Machine Condition Monitoring AutomationA Review." *Machines* 5 (4). doi:10.3390/machines5040024.

Katipamula, S., and M. Brambley. 2005. "Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems - A Review, Part I." *HVAC and R Research* 11 (1): 1–24.

Kephart, J. O., and D. M. Chess. 2003. "The Vision of Autonomic Computing." *Computer* 36 (1): 41–50. doi:10.1109/MC.2003.1160055.

Kholerdi, H. A., N. TaheriNejad, and A. Jantsch. 2018. "Enhancement of Classification of Small Data Sets Using Self-Awareness; an Iris Flower Case-Study." In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–5, Florence, Italy. doi:10.1109/ISCAS.2018.8350992

Liu, J., T. M. McKenna, A. Gribok, B. A. Beidleman, W. J. Tharion, and J. Reifman. 2008. "A Fuzzy Logic Algorithm to Assign Confidence Levels to Heart and Respiratory Rate Time Series." *Physiological Measurement* 29 (1): 81. doi:10.1088/0967-3334/29/1/006.

Massieh, N. 2010. "Fault Detection and Diagnosis in Building HVAC Systems." PhD diss., University of California, Berkeley United States of America.

Mehala, N. 2010. "Condition Monitoring and Fault Diagnosis of Induction Motor Using Motor Current Signature Analysis." PhD diss., National Institute of Technology Kurukshetra, India.

Mounce, S. R., C. Pedraza, T. Jackson, P. Linford, and J. B. Boxall. 2015. "Cloud Based Machine Learning Approaches for Leakage Assessment and Management in Smart Water Networks." *Procedia Engineering* 119: 43–52. Computing and Control for the Water Industry (CCWI2015) Sharing the best practice in water management. http://www.sciencedirect.com/science/article/pii/S1877705815025217

Ross, T. J. 2009. *Fuzzy Logic With Engineering Applications*. Hoboken, New Jersey / United States: Wiley / John Wiley & Sons, Inc.

Sadeghioon, A. M., N. Metje, D. Chapman, and C. Anthony. 2018. "Water Pipeline Failure Detection Using Distributed Relative Pressure and Temperature Measurements and Anomaly Detection Algorithms." *Urban Water Journal* 15 (4): 287–295. doi:10.1080/1573062X2018.1424213.

Shun, L., and J. Wen. 2014. "Application of Pattern Matching Method for Detecting Faults in Air Handling Unit System." *Automation in Construction* 43: 49–58. doi:10.1016/j.autcon.2014.03.002.

TaheriNejad, N., A. Jantsch, and D. Pollreisz. 2016. "Comprehensive Observation and Its Role in Self-Awareness - an Emotion Recognition System Example." In *Proceedings of the Federated Conference on Computer Science and Information Systems, Gdansk,* 117-124. September: Poland. doi:10.15439/2016F588

TaheriNejad, N., M. A. Shami, and S. P. D. Manoj. 2017. "Self-Aware Sensing and Attention- Based Data Collection in Multi-Processor System-on-Chips." In *2017 15th IEEE International New Circuits and Systems Conference (NEWCAS)*, 81–84, Strasbourg, France: IEEE.

Tennenhouse, D. 2000. "Proactive Computing." *Communications of the ACM* 43 (5): 43–50. doi:10.1145/332833.332837.