

# Applicability of Context-Aware Health Monitoring to Hydraulic Circuits

M. Götzinger\*, E. Willegger†, N. TaheriNejad†, A. Jantsch†, T. Sauter†‡, T. Glatzl‡, and P. Liljeberg\*

\* University of Turku, Turku, Finland

E-mail: {maxgot, pasi.liljeberg}@utu.fi

† TU Wien, Vienna, Austria

E-mail: {edwin.willegger, nima.taherinejad, axel.jantsch}@tuwien.ac.at

‡ Danube University Krems, Wiener Neustadt, Austria

E-mail: {thilo.sauter, thomas.glatzl}@donau-uni.ac.at

**Abstract**—Monitoring is an important aspect of operation and maintenance in virtually every industrial system. However, the extent and methods of monitoring vastly vary in different systems, from fully automated to fully manual. One of the challenges of automated monitoring is the tediousness of, and the extent of engineering time and effort required to develop necessary models or machine learning algorithms for the units to be monitored. Model-free monitoring, on the other hand, can save resources and efforts substantially. However, more often than not they have a very limited scope and application. Such a system is needed, for example, to monitor entire Heating, Ventilation and Air Conditioning (HVAC) systems, consisting of different types of sensors such as temperature, pressure, humidity or flow sensors. Recently, we proposed the Context-Aware Health Monitoring (CAH) system for model-free monitoring of any injective-function black-box, and it was tested successfully on an AC motor. In this paper, we evaluate the CAH system for an entirely different industrial use-case, that is, a hydraulic circuit. The results show the potential for considerable benefits in monitoring HVAC systems. Moreover, in the light of applying CAH to different use-cases which may potentially need a different setup of parameters, we performed a sensitivity analysis on the values of different parameters in the system. The results show the robustness of CAH with regard to the values of these parameters.

**Index Terms**—Monitoring, Hydraulic Circuits, Context-Awareness, Model-Free, Injective Function

## I. INTRODUCTION

The worldwide energy consumption is still increasing and, according to the Organization for Economic Cooperation and Development (OECD), industrial and residential sectors are two main consumers of electric energy [1]. It is possible to save energy significantly by improving the Heating, Ventilation and Air Conditioning (HVAC) systems [2]. In Europe, the energy efficiency directive of the European Union (EU) is one action of several efforts to improve the consumption of energy in Europe [3]. One possible measure put forward there is to improve the efficiency of HVAC systems as they are among the major energy consumers. For instance, in Austria HVAC systems were responsible for 27% to 30% of the total energy use within the years 2006 and 2016 [4]. It is possible to reduce 5 to 15 percent of the energy consumption of HVAC systems by fixing faults or with optimizing the building control

system [5]. Based on that, we estimate a potential saving of 15-45 petajoules per year for Austria alone. However, that requires continuous monitoring and better maintenance of these systems, which in turn imposes certain challenges.

Manual maintenance promotes long durability, but it is costly. The goal is to perform maintenance operations as seldom as possible but as often as necessary. However, automated monitoring of a device increases the engineering effort because the rules and patterns for normal behavior and anomalies have to be derived and built into the system as trigger conditions for maintenance actions. In order to reduce design time effort and cost, generic methods are desirable that can be applied to monitor a wide variety of devices. Various methods exist for black-box monitoring; however, these methods require significant resources and computational power and are thus only applicable to large-scale systems.

Therefore, a lightweight black-box health monitoring system for inexpensive devices with limited resources would facilitate the automated monitoring of many cyber-physical and IoT devices. In an earlier work [6], we proposed the Context-Aware Health Monitoring (CAH) system, a novel method, for monitoring a black-box without prior knowledge about it. In other words, the monitoring system has no model of the monitored system which is called Device under Monitoring (DuM) in the further course of this work. CAH utilizes context-awareness for monitoring the DuM. That is, looking at the contexts under which sensory data and their changes

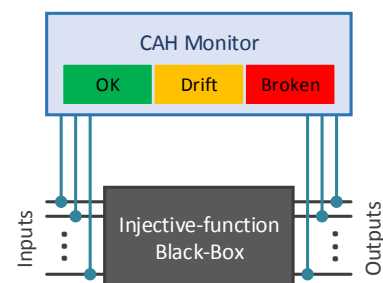


Fig. 1. Block diagram of the Context-Aware Health Monitoring (CAH) system [6].

are observed, it discovers system states, state changes, normal behavior, and abnormalities of the DuM.

Two assumptions are made: (1) the DuM is in steady state, meaning that CAH ignores transient states; (2) the implying that one unique input dataset corresponds to exactly one output dataset and vice versa. Consequently, if only the input or the output data is changing, the DuM shows per definition an abnormal behavior. Furthermore, CAH does not only detect anomalies that signify a broken device but also slight deviations such as a drift. Drifts can happen during a wear-out, and identifying them enables maintenance of the DuM before the device condition deteriorates further. This model-free approach based on context-awareness has the advantage that no retraining, recalibration or other design modifications are required when the system structure changes or sensors are replaced.

While we demonstrated the applicability of CAH to AC motors earlier [6]; in this paper, we extend its scope to the monitoring of a hydraulic system, as it is common in HVAC facilities. We show that CAH detects normal operations, state changes, drifts as well as abnormal functions. As the characteristics of these two applications are clearly distinct, the presented cases study significantly extends the scope of CAH, importantly with no modifications of the underlying algorithms. Furthermore, we conducted a sensitivity analysis of the parameters regarding their set values. These analyses show that CAH benefits from a high degree of robustness against the exact value of these parameters and their changes.

The rest of this paper is organized as follows: In Section II, we review the literature on the HVAC systems, analyze their requirements and specification. These facts explain why the usage of CAH is beneficial for this application. We briefly present the architecture of CAH in Section III, and show and discuss the experiment results in Section IV. Finally, Section V concludes the paper.

## II. BACKGROUND AND RELATED WORK

During the last few years, research on Automated Fault Detection and Diagnostic (AFDD) methods for building systems have been increasing steadily [7]. AFDD methods are subdivided into three different approaches based on qualitative models, quantitative models, and models based on process history [8]. 62% of the implemented AFDD methods are process history based models [7]. This kind of models are further subdivided into black-box and gray box approaches, and within the black-box approaches, there are three different methods possible: statistical, Artificial Neural Networks (ANNs) based, and other pattern recognition techniques [8]. In the following, we describe only the advantages and disadvantages of ANN and Fuzzy logic, and two approaches are discussed in more details as they are most relevant for this work.

Among the advantages of ANN [9] is that they can model non-linear systems without detailed knowledge of the DuM. Disadvantages are that vast amounts of training data is required to model complex systems, it is challenging to gain any physical insight from the ANN, and if some relevant input is

not part of the training data, the ANN may produce erroneous output. The advantages of Fuzzy logic are [9] that it can model non-linear behavior, the commission of Fuzzy Fault Detection and Diagnostic (FDD) schemes is easier, expert knowledge and knowledge learned from measured data are easy to combine, and the software implementation is not computationally intensive. Disadvantages are that the results are less precise compared to other approaches, and the used rule-based descriptions are fairly long. Several researchers have used ANN-based approaches [10]–[13], and pattern recognition techniques [14]–[17] as black-box methods. Fuzzy logic has also been used by FDD [18]. In the following, given their relevance, the approaches by Fan et al. [12] and Dexter et al. [18] are discussed in more details.

Fan et al. [12] combine Back-propagation Neural Network (BPNN) models with wavelet analysis and Elman neural networks. Their approach consists of two parts. In the first part, BPNN fault detection models are generated based on historical data under normal operating conditions. One model is based on variable correlation in the control loop and the other of sensitivity analysis. In the second part, the fault diagnosis model is created by a combination of wavelet analysis and an Elman neural network. The task of the fault diagnosis model is to determine the reason for the fault in the control loop. The diagnosis flow consists of the following five steps:

- 1) Both BPNN models are created.
- 2) New input data is analyzed by the BPNN fault detection model. If a fault is detected, step 3 is executed otherwise the FDD finishes.
- 3) Approximation coefficients from the historical data (including faulty and normal data) are extracted with the wavelet analysis. These coefficients are used for the Elman neural network to diagnose the sensor fault.
- 4) New data is entered, and if the BPNN fault detection model detects a new fault, the approximation coefficients are extracted and clustered.
- 5) If the input data is already known as fault data, then the Elman neural network tries to identify the fault type. If it is unknown, return to step 3 and add this data to the historical data to train the Elman neural network with it.

With his hybrid FDD strategy Fan et al. were able to detect new unknown faults in HVAC systems.

Dexter et al. [18] use a multi-step, Fuzzy model-based approach. They divide the process into two phases, the fault elimination and fault classification phase. The fault elimination phase is shown in Figure 2 and the other phase in Figure 3. Within the fault elimination phase confidence, similarity, and ambiguity of the similarities, and the strength of evidence of the possible state are calculated. Then the current evidence is combined with the evidence from previous operating points, and the values for belief in fault-free operation as well as faulty behavior are recalculated. The result of this step is:

- unambiguous belief in fault-free operation, or
- unambiguous belief in the presence of a fault, or
- it is not clear if a fault is present.

Only if two or more faults are present, the fault classification phase will be executed. The particular fault is isolated with the re-evaluated test data, but only the faults with unambiguous belief from the fault elimination phase are used. The result of this phase could be the unambiguous belief in one fault or a list of all possible faults. This approach was able to recognize a fault-free operation and detect faults in HVAC systems, such as sensor bias, a leaky valve, a fouled coil, and when the valve stuck at fully closed, midway or fully open position. Drawbacks of this solutions were that it was not able to detect drift, and Fuzzy reference models were created from the design specifications. Our proposed solution overcomes these drawbacks.

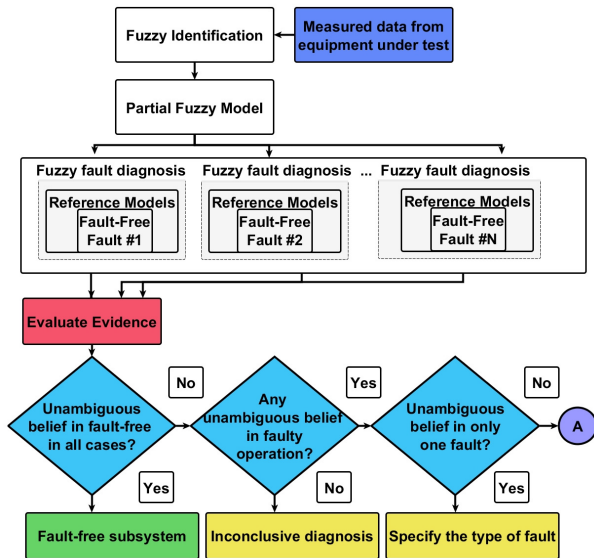


Fig. 2. The fault elimination phase. Adapted [reprinted] from [18].

### III. CONTEXT-AWARE HEALTH MONITOR

The CAH system consists of three function blocks (Fig. 4). Since CAH assumes that the DuM is in steady state, only stable signals are of interest. Periods in that signals are unstable (e.g., the transition during a state change), are not considered and have to be ignored. The first two functional blocks accomplish this task.

#### A. Pre-Processing

The Pre-processing block (shown in the red frame of Fig. 4) contains two different tasks which are case-specific: abstraction and filtering. Because datasets used in this study, namely water flow measurement data, do not need any abstraction, this unit is not used here. The second task of the pre-processing block is to filter all signals to reduce noise as well as oscillations during and shortly after a state change. For filtering, we used a MATLAB® Equiripple filter<sup>1</sup>

<sup>1</sup>The MATLAB® function *fdesign.lowpass* was used with following parameters:  $F_p = 0.005$ ,  $F_{st} = 0.1$ ,  $A_p = 0.15$  and  $A_{st} = 0.999$ . We note that a low-pass filter implementation is outside the scope of this work for which there are several light-weight methods of implementation on hardware and software.

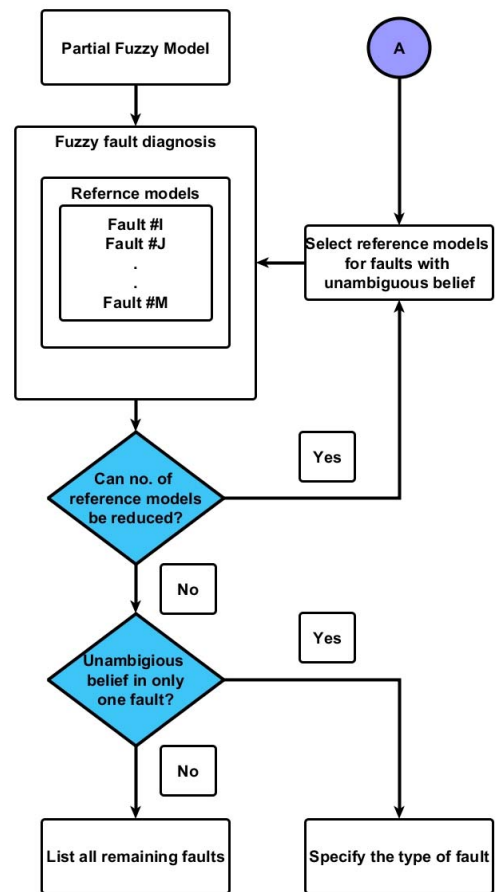


Fig. 3. The fault classification phase. Adapted [reprinted] from [18].

#### B. Stability Controller

Since CAH only considers steady states of the DuM, a Stability Controller (SC) is needed (shown in the green frame of Fig. 4). Whether the signals are filtered or not, this functional block is essential to discard the samples during a transition. Filtered signals may be more stable, but filtering does not replace the SC. The DuM is stable if all signals are stable. Therefore, SC provides a sliding history window for each signal. The SC saves the last samples of a signal in the history assigned to this signal, and a signal is stable if the value of the new sample is in the proximity to a certain number of the saved signal samples. In this regard, the new sample ( $s_{new}$ ) is in proximity to a saved sample ( $s_s$ ) when the relative distance between both samples is lower than a certain threshold (called *Threshold to be Different to Samples in the History*). In this regard, the relative distance  $d$  is given by

$$d = \frac{|s_{new} - s_s|}{|s_{new}|}. \quad (1)$$

We note that the size of the sliding window of this history, the number of samples which have to be similar (which are not different), and the threshold are adjustable. An adjustment may be necessary if the data collection sampling rate is changed.

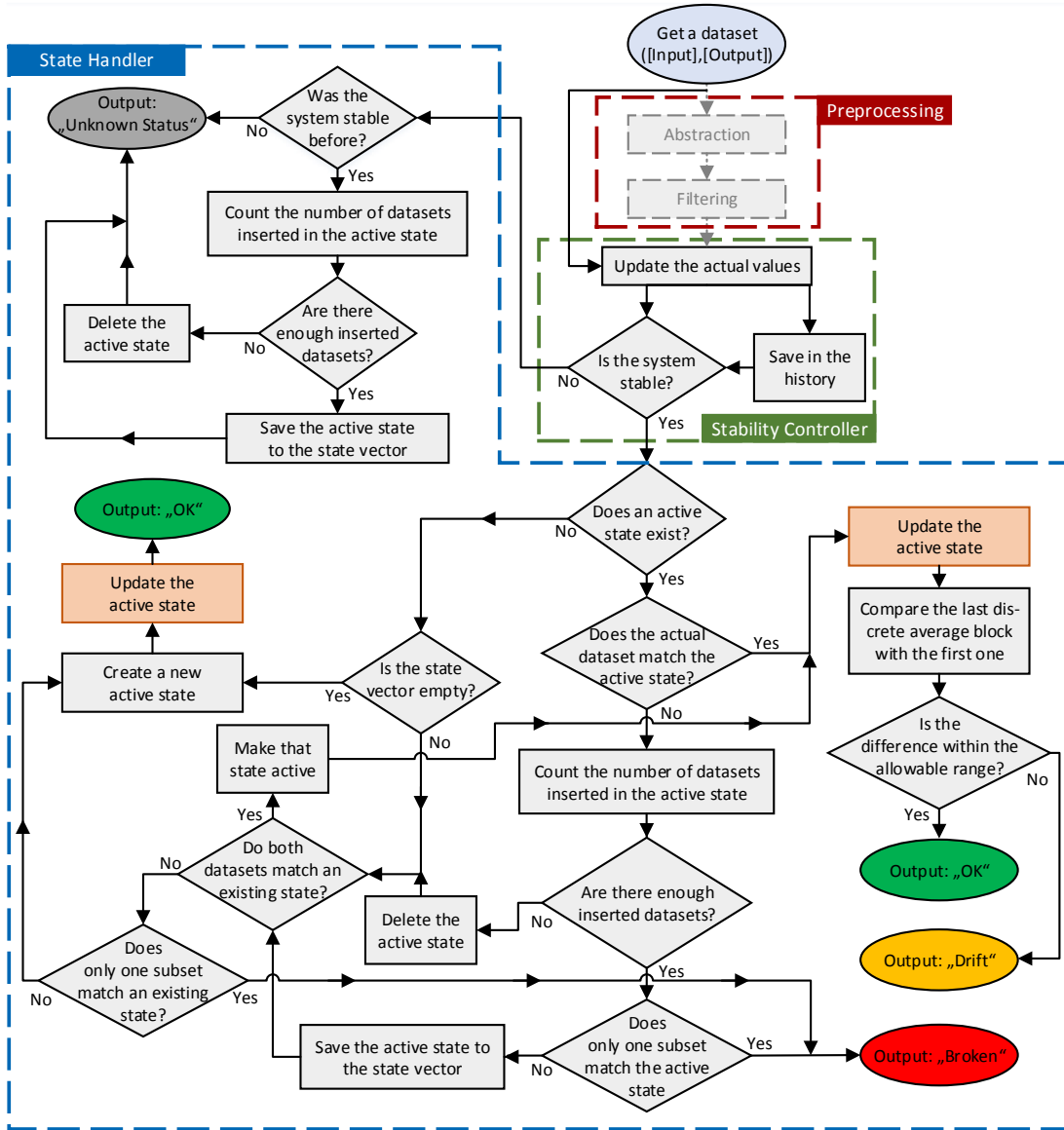


Fig. 4. Flowchart of the Context-Aware Health Monitoring system.

### C. State Observation

The State Handler (SH) is the centerpiece of the CAH. It detects states, state changes, malfunctions, and drifting signals of the DuM.

1) *Discovering a State*: When the SH detects a state of the DuM (a DuM state), it records the data in a C++ object. For the sake of convenience, these objects are called states, and all of these states are saved in the state vector of the SH. The information in such a state is saved as an average value, in particular, an average value for each signal of the DuM. Saving this information in an average value constitutes a very abstracted form of a history. When a new sample that belongs to the same state appears, it gets inserted in the state and influences the average value. A new sample belongs to a state if its value is in proximity to the average value. In this regard, the new sample ( $s_{new}$ ) is in proximity to the average

value ( $v_{avg}$ ) when the relative distance between them is lower than a certain threshold (called *Different-to-State Threshold*). In this regard, the relative distance  $d$  is given by

$$d = \frac{|v_{avg} - s_{new}|}{|v_{avg}|}. \quad (2)$$

In further consequence, a new sample set belongs to a state if all samples are similar (not different) to their respective average values.

2) *Discovering a State Change*: In the SH, the equivalent of the actual state of the DuM is called active state. If a new sample set matches the active state, the sample values are inserted into it, and it remains active. When the sample set, meaning, in particular, one or more samples of this set, are different to the active state, a state change is indicated. Such a change can be either normal or can constitute a malfunction of

the DuM. If both, input and output subsets are different from the active state, a normal DuM state change is happening. In this case, the SH compares the actual sample set with each state saved in the state vector. If one of them matches, that state is activated, and the SH inserts the new sample set into it. If no saved state matches the actual dataset, a new active state is created, and the new sample set is inserted into it.

A state is saved to the state vector if it is valid. A high number of inserted samples characterizes a valid state. If only a few samples were inserted in the active state before a state change happens, the SH discards the active state. This procedure is necessary because in some cases, a particular combination of the sampling rate and the signal curves can make the SC consider the DuM as stable even though it is not.

In contrast to a regular DuM state change, a malfunction is identified when only one subset (input or output sample set) is different from the active state. In this case, the DuM is classified as broken, but only if the active state is valid. Otherwise, the change of only one subset has been observed most likely because the DuM was not steady.

3) *Discovering a Drift*: A drift of a signal is another misbehavior of the DuM. Because a drift is a very slow change in a signal, it also changes the signal’s average value bit by bit. Thus, a drift cannot be discovered by comparing a new sample value with the average value. Therefore, the task of updating the active state with a new sample set (shaded in peach in Fig. 4) is a bit more complicated than just calculate a new average. The SH also saves so-called Discrete Average Blocks (DABs). A DAB is an average value of a certain number of samples. In other words, it is not a sliding window calculation over all samples inserted into a state but only over a certain number of samples. After this specific number of samples is inserted in a DAB, the average is calculated, and a new DAB is created. A drift can be detected when comparing the average values of two different DABs. If a new DAB ( $DAB_{new}$ ) is different to the very first DAB ( $DAB_1$ ) calculated for a state, it is clear that the signal has shifted over time, even if the signal has drifted only slightly. A signal is considered as drifting if the difference between these two DABs is higher than a certain threshold (called *Drift Threshold*). In this regard, the relative distance  $d$  is given by

$$d = \frac{|DAB_1 - DAB_{new}|}{|DAB_1|}. \quad (3)$$

#### IV. EVALUATION

##### A. Experimental Setup

The entire setup is illustrated in Fig. 5. It consists of a copper pipe system with four ultrasonic flow sensors (two Sharkey FS, one Dynasonics TFX Ultra and one RIELS RIF600P), two temperature sensors (Pt100), two water pumps (of which only one was used for the actual experiments) and an electric heater. A Raspberry Pi in combination with an Arduino Uno controls the entire system while logging the data of each

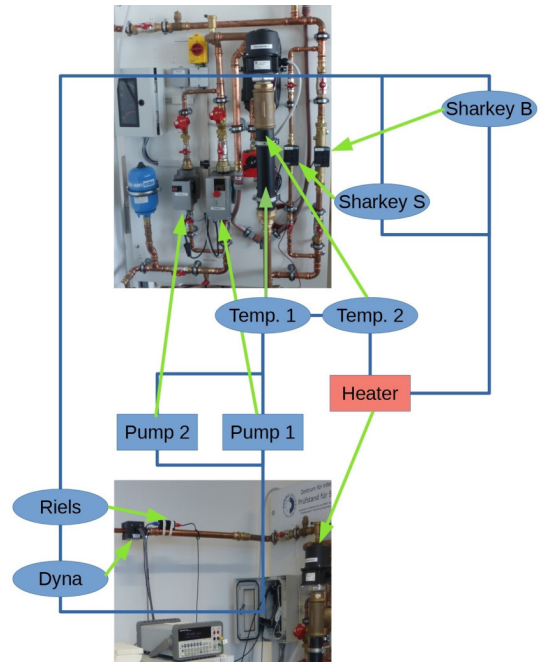


Fig. 5. Experimental setup.

sensor and actuator, realized via a Python software program. The water pump operates in a range between 3.0 to 10.0V, which corresponds to the normalized values of 0.3 to 1.0 in the gathered data. The input signal is the voltage applied to the Pump 1, and the output values are temperature and flow. The sampling frequency was 30.5Hz in average, and the gathered data are stored in CSV-files for post-processing purpose.

Table I lists the various scenarios applied to the HVAC system to validate the correctness of CAH. In the “normal” state scenarios only the voltage of the water pump was changed. The scenarios “anomaly” and “drift” were simulated through manual changes of the degree of the opening of some of the valves. For anomaly, Scenario 5, the valve before sensor SharkyB was opened completely, and the valve before SharkyS was closed at the beginning. Afterward, the SharkyB valve was closed, and the SharkyS valve was abruptly opened. This action simulated a burst of the pipe containing SharkyB. Drift, Scenario 6, was simulated with SharkyB valve half-opened initially and then it was slightly opened further every 15 seconds. This procedure reduces the flow in SharkyS which simulates the increase in sediment and gradual clogging of SharkyS.

##### B. Results

The output of the CAH system shows that it recognizes normal and abnormal behavior when monitoring a hydraulic system. CAH reads in the data with a down-sampling factor of 50 and classifies all of the scenarios correctly. During the experiments, it turned out that the CAH can also classify the hydraulic system correctly when its input and output signals are unfiltered. Skipping this task saves computational power and shows that CAH is robust enough to distinguish between

TABLE I

DIFFERENT TESTED SCENARIOS AND EVENTS DURING EACH OF THEM.

#	Condition of the HVAC system	Events in the scenario
1	OK	No state change during the experiment.
2	OK	One state change, resulting in two different states during the experiment.
3	OK	Two state changes, resulting in three different states during the experiment.
4	OK	Two state changes, whereas the second change leads again to state one.
5	Broken	No change in the input, but a change in the output in the middle of the experiment.
6	Drift	No state change, but a drifting signal starting in the middle of the experiment.

noise and real signal changes correctly. Some results are discussed in more detail in the following.

1) *Normal Mode*: All other normal operation scenarios (Table I) were tested, which led to a correct identification of the normal operation states and their changes.

As an example, Fig. 6 shows Scenario 4. Here, the pump starts operating at the beginning of the experiment, and all signals have settled approximately 40s later. At this time, the SC decides that all signals are stable, showed by the green graph raising. Further, the SH finds the first state (blue graph) and decides that the DuM is working correctly (red graph). This circumstance lasts until 210s, when a changed input voltage leads to a state change. After a ~35s long unstable period, the SC recognizes that all signals have settled again, and the SH discovers the second state. Since input and output datasets have both changed, the DuM is still considered as working normally. At ~425s the DuM changes back to its previous (initial) state. This change is again accompanied by a ~35s long unstable period. After that, CAH concludes, that the DuM is again in state 1 and working correctly. This condition continues until the end of the experiment.

2) *Anomaly*: Fig. 7 shows experimental results for Scenario 5, where significant changes in the output signals are observed. However, as the input voltage stays unchanged, the DuM is per definition broken. The records start with the start-up of the pump, which leads to a settled DuM at ~35s when the SC notifies the SH that it is now in a steady state, at which point the latter recognizes the first state. At 310s, the DuM suddenly changes its outputs, which is not triggered by an input change. After this change has started, it takes around 40s until the SC verifies the DuM as stable. In this 40s, the SH is just waiting to be triggered, and after that time, it recognizes that the DuM is still in the same state but shows symptoms of malfunctioning. Therefore, considering the context of these changes, the DuM is classified as broken.

3) *Drift*: Fig. 8 shows the results for a drifting system (Scenario 6). Once more, the pump is starting up at the beginning. After ~40s the SH recognizes the first state and the CAH concludes that the DuM works correctly. However, this changes at around 400s, when the SH recognizes a drift. After ~45s, the output signal value has drifted far enough from

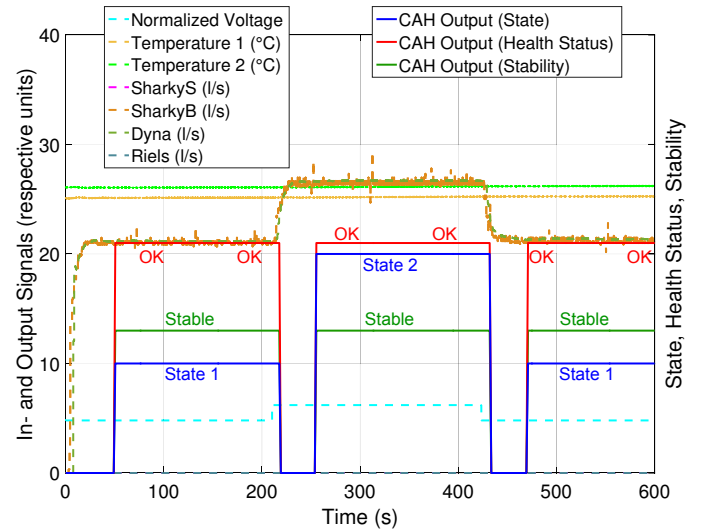


Fig. 6. Input and output of the DuM during two state changes, as well as the output of the CAH system.

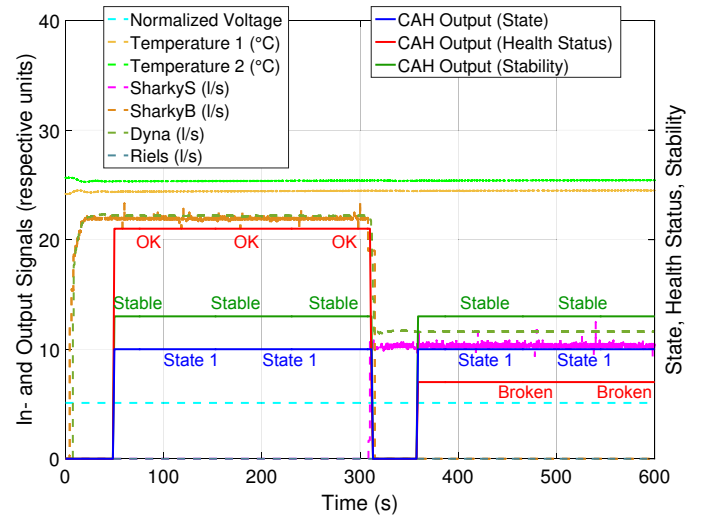


Fig. 7. Input and output of the DuM during an abnormal change leading the CAH system to classify it as broken.

the original state to signify a state change of the output signal. Since the input remains unchanged, the SH classifies the DuM as broken and replaces the drift alarm with a broken one. In the case of a slower drift, the drift alarm would last longer.

### C. Sensitivity Analysis

The robustness of CAH was investigated with a sensitivity analysis in which the CAH system processed all datasets with several parameter combinations. Table II lists the ranges of the values of the parameters analyzed. After running the algorithm with these values, all CAH outputs were analyzed to see whether the health condition of the DuM is accurately classified and its states are correctly detected.

Fig. 9 shows for which set of parameters CAH classifies all scenarios correctly. In other words, those combinations of

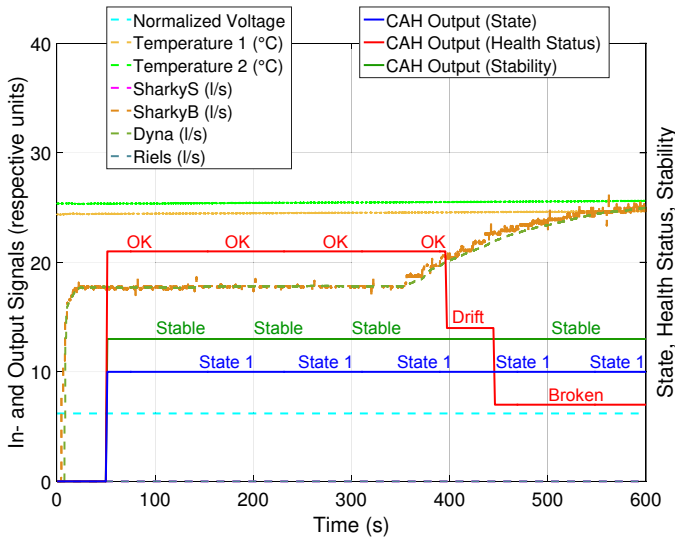


Fig. 8. The outputs of the CAH system while the DuM is drifting.

TABLE II  
TEST RANGES OF THE PARAMETERS OF CAH.

Parameter	Functional Block	Difference Calculation	Test Range	Test Steps
Sliding History Window Size	Stability Controller	NA	3-10	1
Number of Similar Samples in the History	Stability Controller	NA	0-10	1
Threshold to be Different to Samples in the History	Stability Controller	Eq. 1	1%-8%	1%
Number of Samples inserted into a State to be Valid	State Handler	NA	10	NA
Different-To-State Threshold	State Handler	Eq. 2	1%-40%	1%
Drift Threshold	State Handler	Eq. 3	1%-40%	1%
Downsampling Rate		NA	50	NA

parameter values which led to an incorrect classification are not drawn in the figure.

A few points should be noted:

- 1) The window size has to be higher than the number of similar samples required for classification, Number of Similar Samples in History  $\leq$  Sliding History Window Size.
- 2) The history window should contain at least 3 samples, Sliding History Window Size  $\geq$  3.
- 3) It is advisable to set Number of Similar Samples in History  $\geq$  3, as the classification is then more robust for a broader range of other parameter values.
- 4) The two parameters *Number of Similar Samples in History* and *Threshold to be Different to Samples in the History* should grow in tandem; if one is set higher, the other is to be set higher as well. E.g., if Number of Similar Samples in the History  $\geq$  7, then the *Threshold to be Different to Samples in the History*  $\geq$  3. This rule comes most likely from the fact that with a high

threshold, the SC can classify the DuM in some cases as stable although it is not yet. A higher number of Similar Samples in the History counteracts this phenomenon because more time elapses until this number is reached.

- 5) For the tested datasets, we can infer that the *Drift Threshold* and the *Different-to-State Threshold* have to be between 11% and 19%.

The fact that there are no gaps between the points shows that CAH is quite robust when the parameters are in these ranges.

Fig. 10, focuses on a point in the middle of the scatter diagram of Fig. 9 (specifically, *Sliding Window History Size* of 5, *# of Similar Samples in History* of 5, and *Threshold to be Different* of 4%) and analyzes with which threshold parameters the SH classifies the states of the DuM correctly. Each point marks a correct working classification for a set of two given threshold parameters. The points missing from the figure (mostly in the upper left half) represent the combinations which did not lead to a correct classification. Therefore, for the given datasets, we can infer from Fig. 10 that in this instance, choosing larger “Different-to-State Threshold” and lower “Drift Threshold” values leads to a higher probability of correct classification.

## V. CONCLUSION

CAH is a model-free, context-aware monitoring system that identifies states, normal behavior, and various anomalies, based on the context under which the Device under Monitoring (DuM) operates. Its main assumptions are (1) that the DuM resembles an injective function where input and output signals can only change together, and (2) that the DuM, as it operates, evolves from one stable state to the next. CAH has been introduced earlier with an AC industrial motor as guiding application case [6]. In this paper, we have extended the scope of CAH to the HVAC applications with good results. Considering that these two applications are clearly distinct concerning reaction times, the periodicity and regularity of signals, it is significant that the CAH setup had to be adapted only slightly. While the CAH algorithm was unchanged, the only necessary adaption concerned some thresholds due to different amplitudes of signals. Moreover, AC motors react to input changes an order of magnitude faster than hydraulic circuits. Because of these differences, at the discretion of experts conducting the measurements, the sampling rate of the measurements was also different whereas the sampling rate inside the algorithm remained the same. Apart from this adaption, CAH has shown remarkable robustness in identifying correct behavior, anomalies and drift when monitoring so diverse applications as AC motors and hydraulic circuits. We support this claim with a sensitivity analysis, which shows the wide range of values which the parameters can be set to, and have the CAH system classify every scenario correctly nevertheless. Thus, we expect that CAH is applicable to an even larger class of applications. That possibly includes medical and environmental monitoring devices, among others, which will be subject to future studies.

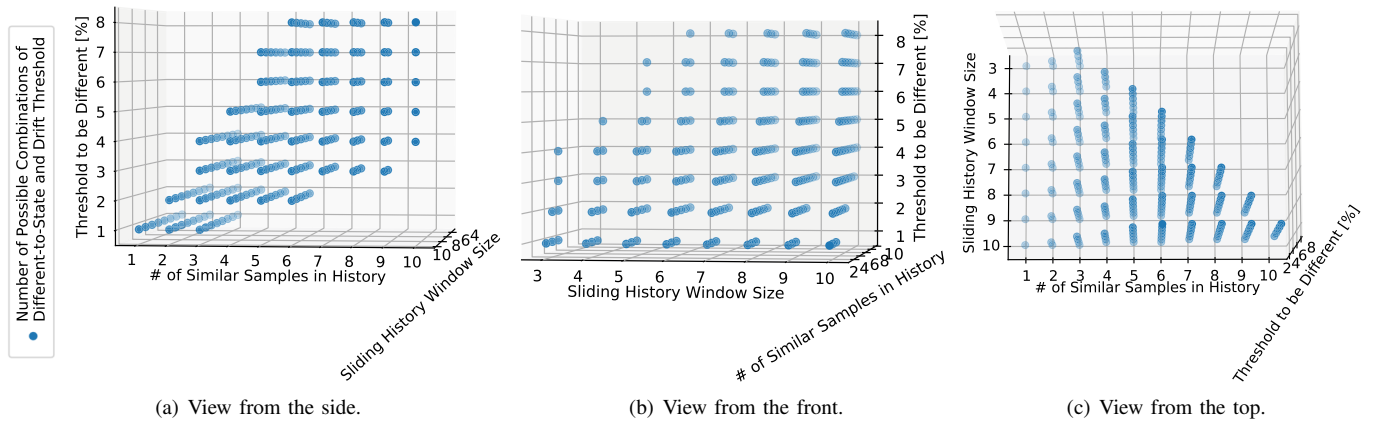


Fig. 9. The 3D scatter plot shown from three different angles represents the sensitivity analysis for the three different parameters of the SC. Each point on the chart represents a combination which led to a correct classification of the DuM state.

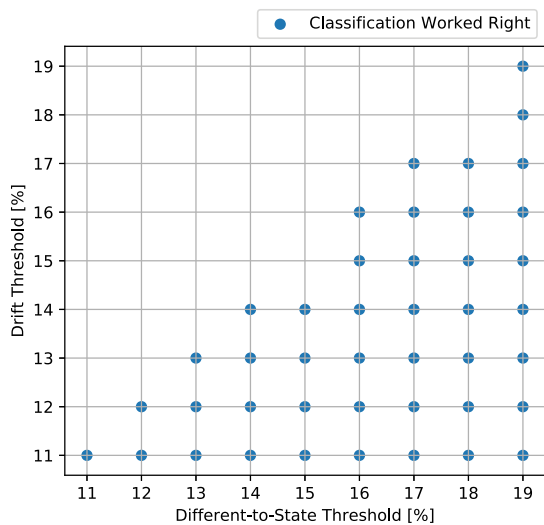


Fig. 10. Possible combinations of SH parameters with which the SH classifies the DuM correctly.

#### ACKNOWLEDGEMENT

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).

#### REFERENCES

- [1] "Electricity information: Overview," International Energy Agency, Note, 2017.
- [2] X. Cao, X. Dai, and J. Liu, "Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade," *Energy and Buildings*, vol. 128, pp. 198–213, Sep. 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378778816305783>
- [3] E. Commission, "Energy efficiency directive," Website, 2012, online at <https://ec.europa.eu/energy/en/topics/energy-efficiency/energy-efficiency-directive>; 06. June 2018.
- [4] S. Austria, "Energiestatistik: Energiebilanzen österreich 1970 bis 2016," Website, 2017, online at [https://www.statistik.at/web\\_de/statistiken/energie\\_umwelt\\_innovation\\_mobilitaet/energie\\_und\\_umwelt/energie/nutzenergieanalyse/index.html](https://www.statistik.at/web_de/statistiken/energie_umwelt_innovation_mobilitaet/energie_und_umwelt/energie/nutzenergieanalyse/index.html); 06. June 2018.

- [5] N. Massieh, "Fault detection and diagnosis in building hvac systems," Ph.D. dissertation, University of California, Berkeley United States of America, 2010.
- [6] M. Götzinger, N. TaheriNejad, H. A. Kholerdi, and A. Jantsch, "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)*. IEEE, 2017, pp. 1–5.
- [7] K. Woohyun and K. Srinivas, "A review of fault detection and diagnostics methods for building systems," *Science and Technology for the Built Environment*, vol. 24, pp. 3–21, 2018.
- [8] S. Katipamula and M. Brambley, "Methods for fault detection, diagnostics, and prognostics for building systems - a review, part i," *HVAC&R Research*, vol. 11, no. 1, pp. 1–24, 2005.
- [9] A. Dexter and J. Pakanen, "Demonstrating automated fault detection and diagnosis methods in real buildings," 2001.
- [10] Z. Du, B. Fan, X. Jin, and J. Chi, "Fault detection and diagnosis for buildings and hvac systems using combined neural networks and subtractive clustering analysis," *Building and Environment*, vol. 73, pp. 1–11, 2014.
- [11] H. He, D. Menicucci, T. Caudell, and A. Mammoli, "Real-time fault detection for solar hot water systems using adaptive resonance theory neural networks," in *Energy Sustainability, 2011. Fifth International Conference on*. ASME, August 2011, pp. 1059–1065.
- [12] B. Fan, Z. Du, X. Jin, X. Yang, and Y. Guo, "A hybrid fdd strategy for local system of ahu based on artificial neural network and wavelet analysis," *Building and Environment*, vol. 45, pp. 2698–2708, 2010.
- [13] Z. Hou, Z. Lian, Y. Yao, and X. Yang, "Data mining based sensor fault diagnosis and validation for building air conditioning system," *Energy Conversion and Management*, vol. 47, pp. 2479–2490, 2006.
- [14] Y. Guo, J. Wall, J. Li, and S. West, "Intelligent model based fault detection and diagnosis for hvac system using statistical machine learning methods," in *ASHRAE, 2013. Conference January Dallas*. ASHRAE, January 2013.
- [15] A. Srivastav, A. Tewari, and B. Dong, "Baseline building energy modeling and localized uncertainty quantification using gaussian mixture models," *Energy and Buildings*, vol. 65, pp. 438–447, 2013.
- [16] R. Sharifi and D. B., "Fault detection in lighting systems - first phase results," Philips Research North-America, Technical Note, 2011.
- [17] M. Najafi, D. M. Auslander, P. Haves, and M. D. Sohn, "A statistical pattern analysis framework for rooftop unit diagnostics," *HVAC&R Research*, pp. 406–416, 2012.
- [18] A. L. Dexter and D. Ngo, "Fault diagnosis in air-conditioning systems: A multi-step fuzzy model-based approach," *HAC&R Research*, vol. 7, pp. 83–102, 2001.