Downloaded from orbit.dtu.dk on: Apr 23, 2024

DTU Library

Combining operations documentation and data to diagnose procedure execution

Kirchhübel, Denis; Lind, Morten; Ravn, Ole

Published in:
Computers & Chemical Engineering

Link to article, DOI:
10.1016/j.compchemeng.2020.106940

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Kirchhübel, D., Lind, M., & Ravn, O. (2020). Combining operations documentation and data to diagnose procedure execution. *Computers & Chemical Engineering*, *140*, Article 106940. https://doi.org/10.1016/j.compchemeng.2020.106940

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Computers and Chemical Engineering 140 (2020) 106940

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Combining operations documentation and data to diagnose procedure execution

Denis Kirchhübel[a,*], Morten Lind[a], Ole Ravn[a]

Department of Electrical Engineering Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

ARTICLE INFO

Article history:
Received 10 October 2019
Revised 24 April 2020
Accepted 20 May 2020
Available online 29 May 2020

Keywords:
Alarm systems
Standard operating procedures
SCADA Systems
Fault diagnosis
Situation awareness

ABSTRACT

Established control room systems in processing industry are prone to overload operators during severe plant upsets. Automatic diagnostic assistants can assist operators in high load situations, but the system needs to be aware of configuration changes.

Standard Operating Procedures detail when and how the plant configuration is to be changed. Tracking the correct execution of a procedure is relevant to both, detecting errors during procedure execution and adapting diagnostic models. However, plant operators tend to deviate from the prescribed procedures. Therefore, the reference for correct procedure execution needs to be established from the procedure documentation but also capturing operator habits.

We describe methods for representing documented procedures and validating the procedures against log files of the control system. We propose a fast approach of detecting procedure executions and action sets associated with procedure steps. The presented methods are demonstrated in an industrial case study of a filtration system.

© 2020 The Authors. Published by Elsevier Ltd.
This is an open access article under the CC BY-NC-ND license.
(http://creativecommons.org/licenses/by-nc-nd/4.0/)

1. Introduction

Alarm systems are the primary interface for human operators in industrial plants in case of abnormal situations, where an alarm is intended to draw operators' attention to deviations requiring an immediate response (ISA International Society of Automation, 2009). And yet poorly managed alarms can cause severe operator overload and have been identified as the cause for a number of large accidents across different industry domains (Goel et al., 2017; Wang et al., 2016).

Developing operator support systems that provide less information with more meaningful context can alleviate the mental strain these situations have on operators. This work contributes to the effort of facilitating a comprehensive operator support system based on causal models (Kirchhübel et al., 2019), which requires that the model for diagnosis matches the current plant configuration. A configuration is a distinct physical setup and control structure in terms of active sub-systems and process streams in the plant. The majority of changes to the plant configuration are related to well documented procedures, such as start-up or maintenance operations (Quiñones-Grueiro et al., 2019), which could be used to track the correct plant configuration. Abdallah et al. (2018) demonstrated how known sequences of events can be used to select the model corresponding to the expected configuration. However, studies across industry domains have shown that operators may deviate from the documented procedures for a number of reasons such as convenience, or different judgement of the situation (Kanse et al., 2018). Because the way a procedure is executed may not always comply with the documented procedure, it is relevant to establish a procedure model for the normal procedure execution by additional means.

The supervisory control and data acquisition system (SCADA) provides comprehensive records of occurred alarms and actions taken on the plant (Bezerra et al., 2019). Many methods have been proposed to analyse this data, especially in the context of alarm management (Wang et al., 2016). Equally many approaches concern the representation and synthesis of operating procedures (Song et al., 2020; Chen and Chang, 2019; Abdallah et al., 2018; Viswanathan et al., 1998). Combining these two concepts has received far less attention though: For instance (Mo et al., 2007) suggest training neural networks based on simulated executions of computerised procedures and applying these networks for online evaluation of operator actions. However, outside the nuclear

* Corresponding author.
E-mail address: dekir@elektro.dtu.dk (D. Kirchhübel).

https://doi.org/10.1016/j.compchemeng.2020.106940
0098-1354/© 2020 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license. (http://creativecommons.org/licenses/by-nc-nd/4.0/)

industry computerised procedures, let alone exhaustive simulators are far from wide spread (Venkatasubramanian, 2019). Instead procedure documentation and operational reality often diverge. To integrate operating procedures into an intelligent operator support system these two domains need to be reconciled.

We propose a novel two step approach to identifying executions of a given operating procedure from SCADA data and subsequently determining the reference model of normal procedure execution from the combination of documented procedure steps and observed procedure executions. We present a formalism for representing operation procedures. By using data logged from SCADA, the representation can be refined and attributed with reference constraints for normal execution, enabling on-line diagnosis of the procedure execution. The presented industrial case study demonstrates the merit of validating the operating procedure against existing event logs.

In the following section a brief review of approaches to reducing operator overloading and tracking plant configurations is presented. Then the representation of operating procedures, related methods of mode-dependent alarms and sequence alignment, as well as the proposed methods for identifying and analysing procedure executions are outlined. The industrial case study is then introduced and its results are presented and discussed. Finally, the validity of the proposed method and the advantage for a rapid deployment of advanced support systems are summarised.

2. Background

In processing industry, control rooms rely on alarms to direct operators' attention to urgent situations in the plant, but can easily overload operators with a large number of concurrent alarms. In consequence, guidelines and indicators for successful alarm management have been developed by industry. Norms published by the ISA (International Society of Automation) (2009) and IEC (International Electrotechnical Comission) (2014) complemented by guidelines and best practice handbooks such as EEMUA (Engineering Equipment Materials Users' Association) (2013); Rothenberg (2009); Hollifield and Habibi (2006), highlight the industrial relevance of alarm management. Studies have shown the effectiveness of rigorously implemented alarm management (Soares et al., 2016).

The most crucial failures of alarm systems are alarm floods that overwhelm the operator with a large number of nearly simultaneous alarms. The multitude of information frequently leads the operators to start treating symptoms while missing out on the root causes (Hollnagel, 2002). A range of approaches to handle alarm floods have been proposed, with a dominant trend of employing data analysis methods. The common methods include sequence matching (Lai and Chen, 2017), analysis of temporal dependence (Folmer et al., 2014), and on-line classification as binary sequences (Lucke et al., 2019; Shang and Chen, 2019). In addition, plant connectivity and causality measures have been proposed as the basis for causal analysis of the plant (Abele et al., 2013; Hu et al., 2017b). Once a causal structure of the plant is established, consequential alarms can be suppressed to help operators focus on the alarms associated with root causes (Larsson et al., 2007; Wang et al., 2015). Given that a plant or systems experiences recurring abnormal situations and consequently a number of widely similar alarm floods, data based methods provide a good tool for examination and interpretation by process experts (Hu et al., 2018). Zhou et al. (2019) propose a method to facilitate the transfer of the knowledge embedded in this data between similar processes. However, new faults arising, due to the age of the plant or specific environmental conditions for instance, will not be captured by purely data-based solutions, while causal analysis can still provide insights in these situations.

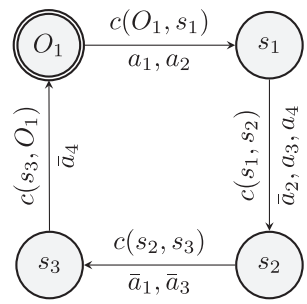

Fig. 1. Example automaton of an operation procedure. Procedures leading back to their starting state are typical for maintenance operations.

Since plants frequently operate in more than one configuration (Quiñones-Grueiro et al., 2019), these causal models need to adapt to account for different plant configurations in the same manner as Beebe et al. (2013) have identified the need to dynamically manage alarm configurations. Abdallah et al. (2018) have presented an approach for a diagnostic system where the underlying model is adjusted based on discrete control system events that can be tracked by an automaton describing the transitions between diagnosis models. At a design stage, the combination of automata of sub-systems can serve to synthesize operating procedures based on combined event sequences (Chen and Chang, 2019). In the context of fault detection and isolation, Quiñones-Grueiro et al. (2019) summarize a variety of approaches for tracking the current configuration of the system. Most of the presented approaches use one model per configuration and detect the current configuration as the best matching between observations and model (Quiñones-Grueiro et al., 2019). The alternative approach is to track the system transitions and determine the current configuration based on the combination of individual variable transitions (Srinivasan et al., 2005). At a plant wide level, transition information such as opening and closing of valves is readily available from SCADA. Logically, a procedure-tracking approach based on transitions at supervisory level can be based on available SCADA events.

3. Methods

This section details the relevant methods for representing operating procedures, determining triggering alarms, and identifying the execution sequence from data. Then we propose methods for detecting candidates for the execution of a specific procedure as well as a fast analysis method to validate the execution from available log files.

3.1. Representing operating procedures

Operating procedures are an essential part of plant operation, as they ensure a consistent execution and high quality and safety of operation (EPA, 2007). In emergency situations as well as during transients in operation, like start-up or shutdown, operation procedures define the sequence of actions and expected outcome in the system. These procedures are implemented in the automatic control system or executed by human operators. Operating procedures are thoroughly documented in terms of the specific steps and the timing or thresholds between each step to the extent of integrating computerised procedures with the operator interface. The steps of an operating procedure correspond to state transitions of the plant and can thus be represented as a finite state machine or automaton notation.

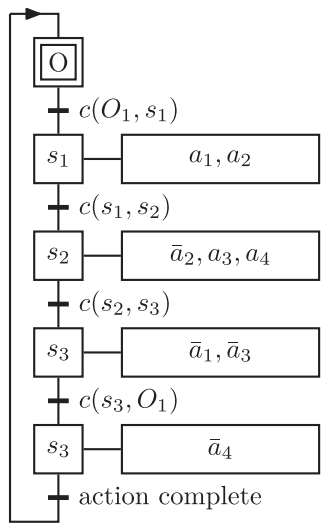

Fig. 2. Sequential Function Chart of the example operation procedure.

Time	Event	
1.0	a_1	
5.0	$\bar{a}_1$	
5.0	a_3	
5.0	$\bar{a}_4$	T_{s_3,O_1}
8.0	$\bar{a}_3$	
10.0	a_1	T_{O_1,s_1}
10.1	a_2	
14.5	a_5	
14.5	a_3	T_{s_1,s_2}
14.5	a_4	
14.6	$\bar{a}_2$	
20.1	$\bar{a}_1$	T_{s_2,s_3}
20.1	$\bar{a}_3$	
24.0	$\bar{a}_4$	T_{s_3,O_1}

Fig. 3. Example event log with occurrence candidates.

Time	a_1	a_2	$\bar{a}_2$	a_3	a_4	$\bar{a}_1$	$\bar{a}_3$	$\bar{a}_4$	a_5
1.0	1	0	0	0	0	0	0	0	0
5.0	0	0	0	1	0	1	0	1	0
8.0	0	0	0	0	0	0	1	0	0
10.0	1	0	0	0	0	0	0	0	0
10.1	0	1	0	0	0	0	0	0	0
14.5	0	0	0	1	1	0	0	0	1
14.6	0	0	1	0	0	0	0	0	0
20.1	0	0	0	0	0	1	1	0	0
24.0	0	0	0	0	0	0	0	1	0
	T_{O_1,s_1}		T_{s_1,s_2}			T_{s_2,s_3}		T_{s_3,O_1}	

Fig. 4. Matrix repersentation of the log with detected transition occurrences (gray boxes) and end of procedure indicator (dashed box) highlighted.

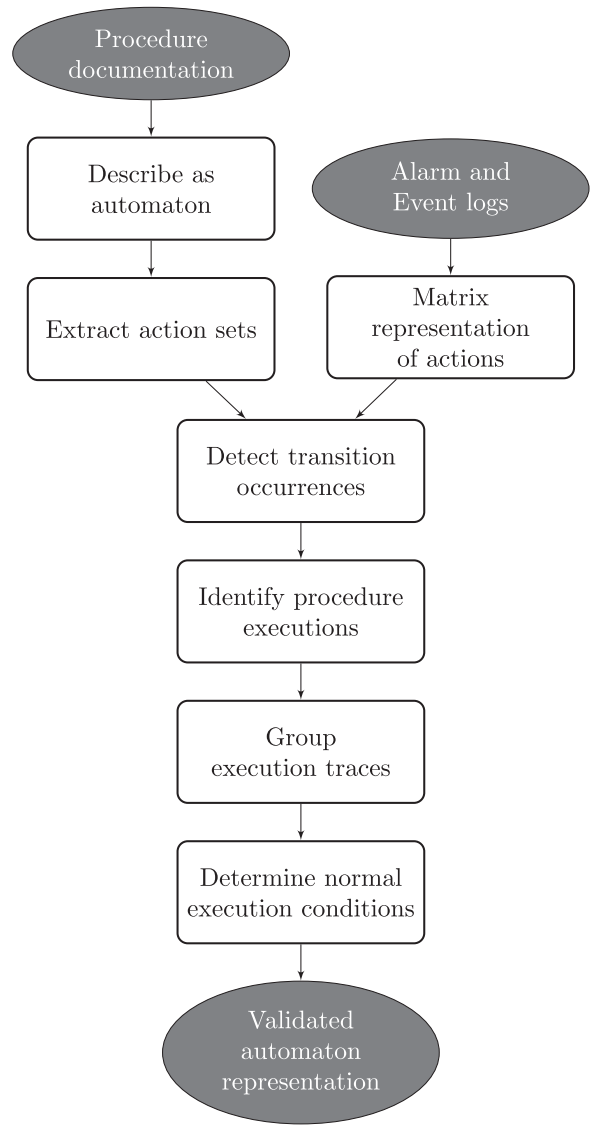

Fig. 5. Block diagram of the procedure representation and validation against alarm and event logs.

In an automaton notation each state corresponds to the current configuration and settings of the plant. Transitions are characterised by a condition and by the set of changes made to the plant. Each transition $T = \{c, \mathbb{A}\}$ describes the necessary condition c to execute the step and the set of actions $\mathbb{A}$ associated with the transition. One specific operation procedure to move the plant from operation point O_i to another operation point O_j is represented by a valid trace in the automaton (Lunze, 2004). The trace will normally contain a number of intermediate states $s_1, \ldots, s_n$. Like the operation points, these states are characterised by the configuration of valves and set points of the controllers throughout the plant. The condition to initiate the procedure by triggering transition T_{O_i,s_1} may be tied to a complex condition, such as a combination of multiple process variables exceeding pre-defined thresholds. In the interest of reliability in a stressful situation for the operator all following transitions $T_{s_k,s_{k+1}}$ are commonly conditioned on a time constraint or the behaviour of an individual process variable. Chen and Chang (2019) utilise the relation between automata that can be used for simulation, and Sequential Function Chart (SFC), which are applied for procedure representation in industrial

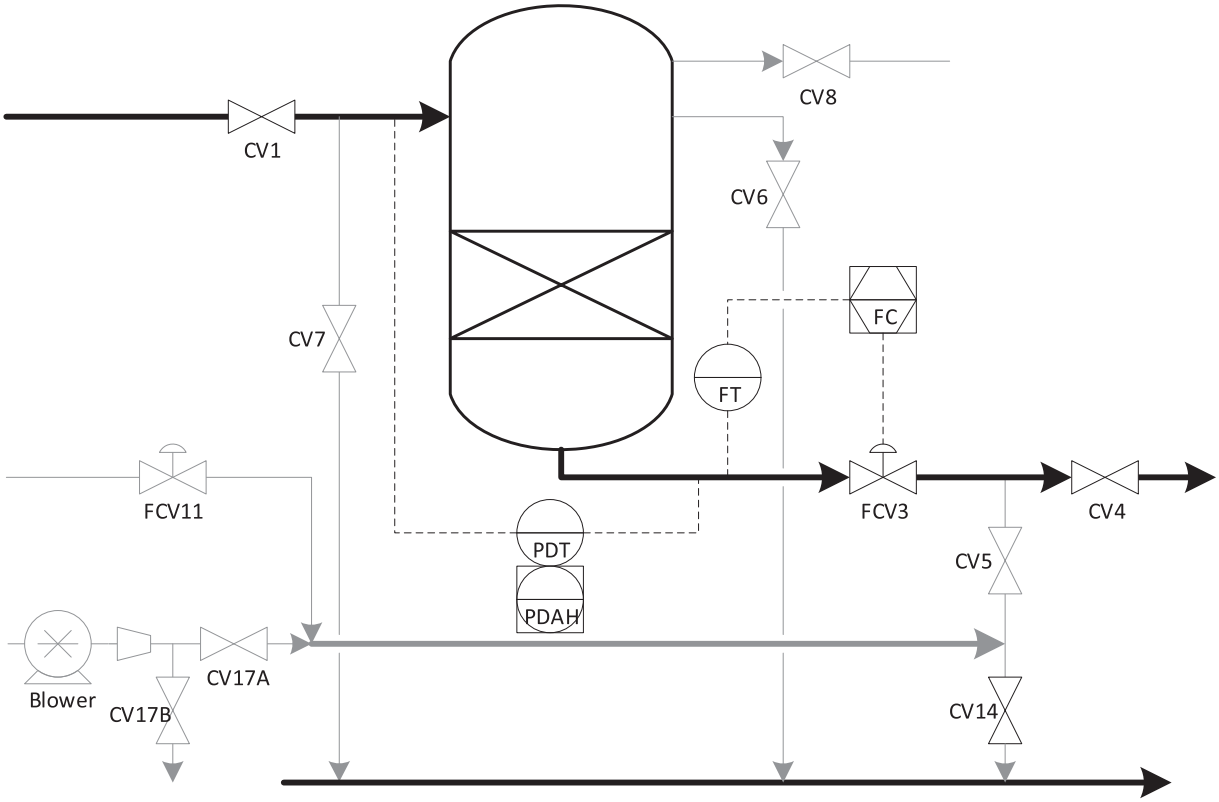

Fig. 6. Flow sheet of the studied filter unit. The direction of the main pipes (thick lines) refers to the normal configuration while in operation. The flow through FCV3 is reversed during back-washing. High differential pressure alarm PDAH provides an indication for filter clogging and the need to back-wash. PDAH is associated with a high δp measurement of PDT.

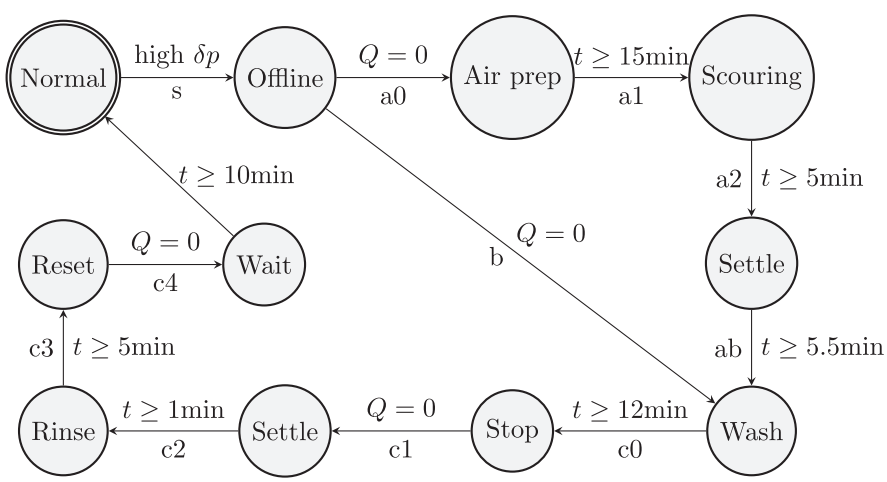

Fig. 7. Automaton of the filter unit. Edges are labelled with the conditions before the transition occurs (above/right) and the transition name (below/left). Conditions are either the time spent in the proceeding state or a specific measurement like the flow Q measured by the flow meter FT or the differential pressure δp measured by PDT.

automation systems. The automaton representation and equivalent SFC of a procedure are shown in Figs. 1 and 2.

3.2. Alarm dependent procedures

Standard operating procedures are defined for a specific plant situation, in the case of safety procedures a specific set of alarms could require the operator to shut down the affected part of the plant or start-up specific auxiliary systems. In a system with redundant installations, an alarm on one of the redundant systems could require to switch over to other parallel branches and perform maintenance on the affected system. Hence, alarms could serve as indicators for start of a procedure execution indicated by the transition T_{O_i,s_1}.

According to the association rule discovery proposed by Hu et al. (2018) a procedure can be considered to be alarm dependent based on two conditions: The procedure is executed frequently enough to be statistically relevant and the probability of the initial transition following the respective alarm is close to 1. The discovery method is applied to identify the set of alarms x_i that trigger a specific procedure execution y_i. The nomenclature of the conditions is adapted from Hu et al. (2018) as follows. The number of procedure executions ξ exceeds the user-defined threshold F_{th} and

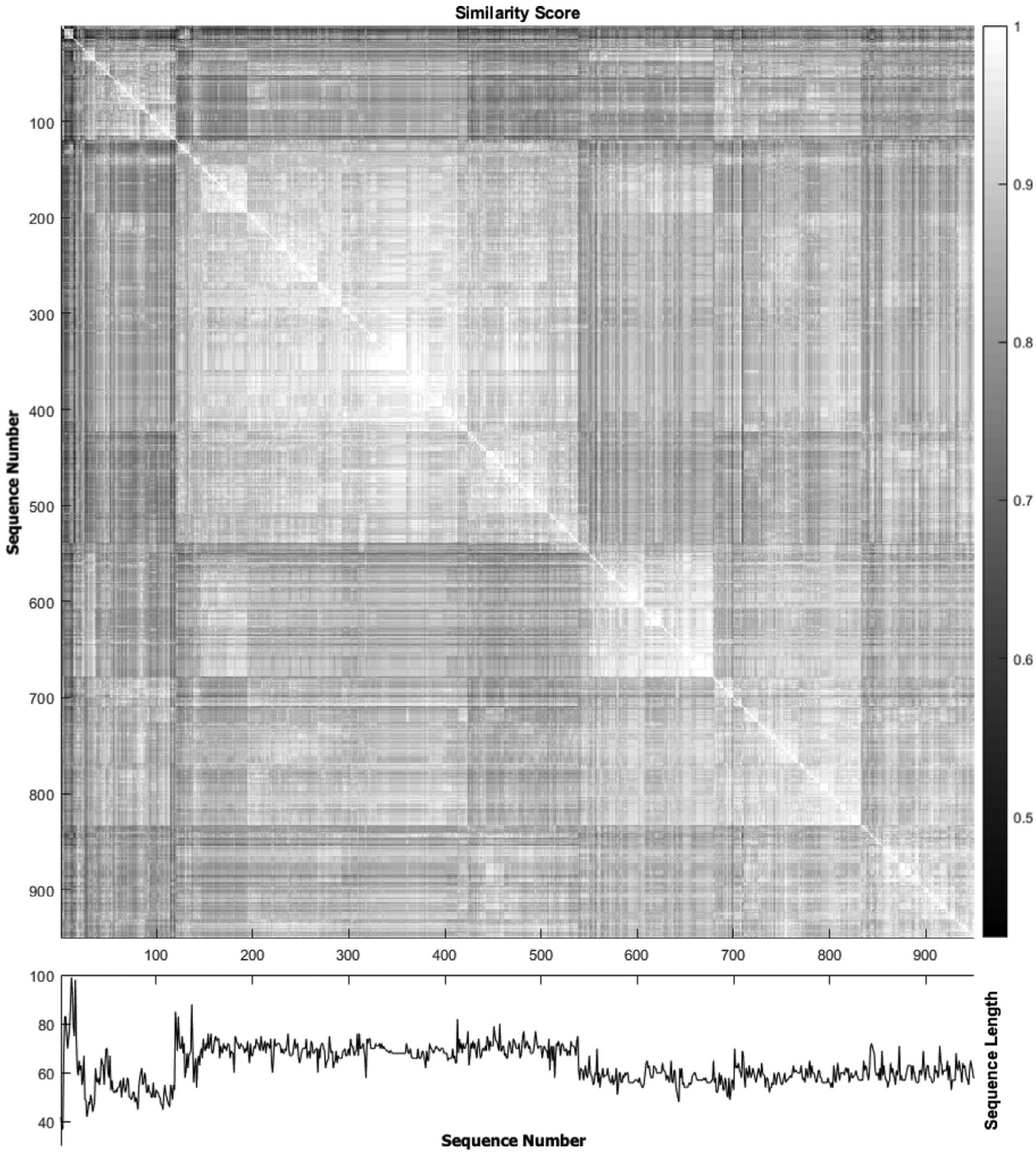

Fig. 8. Pairwise similarity scores and sequence length. The sequence length is the number of events recorded for the filtration system during the procedure execution. The sequences are clustered according to highest similarity scores determined by complete sequence alignment. The sequence length illustrates a correlation between high similarity scores and similar sequence length.

the approximated probability $\hat{p}(y_i|x_i)$ of the initial transition occurring after the alarm is 1 or close to 1.

$$\xi(x_i \rightarrow y_i) \geq F_{th} \tag{1}$$

$$\hat{p}(y_i|x_i) = \frac{\xi(x_i \rightarrow y_i)}{\xi(x_i)} \approxeq 1 \tag{2}$$

The time window W_{th} determining whether an occurrence of x_i is followed by y_i should be set sufficiently large to account for the operator's reaction time. Furthermore, for the alarm to be a reliable precursor for the specific procedure execution, the procedure should only be executed if the alarm occurred, introducing the additional constraint that, effectively, the number of occurrences of the alarm and the procedure execution match $\xi(x_i) \approx \xi(y_i)$.

3.3. Sequence alignment

Besides the start of a procedure, it is desirable to diagnose the intermediate execution steps of the procedure. To identify the normal execution the recorded data from the alarm and event logs are considered as reference. Considering each procedure execution as a sequence of events in the SCADA system, all sequences for the

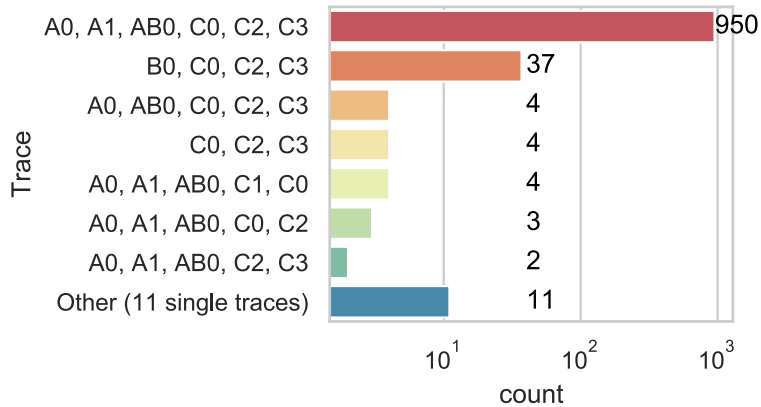

Fig. 9. Occurrences of transition sequences for back-washing during the case study period.

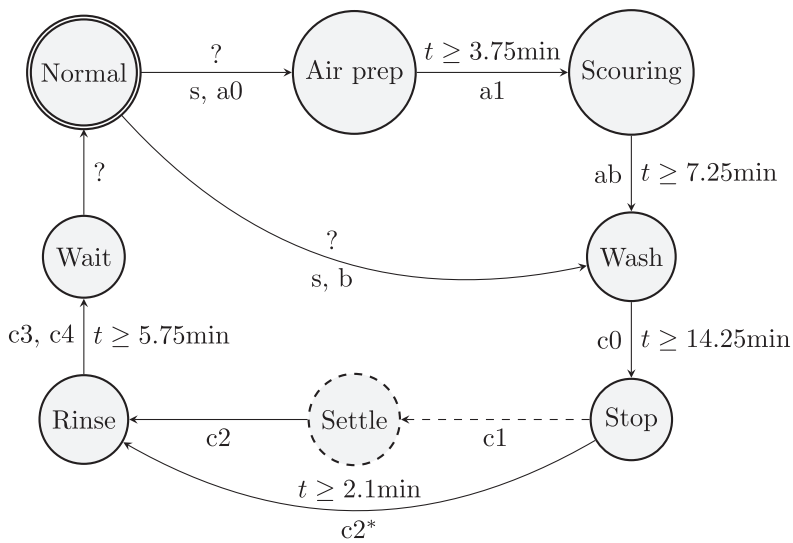

Fig. 10. Reduced automaton derived from the data analysis.

procedure can be aligned to provide a reference for the events and conditions related to the procedure.

Sequence alignment has been proposed to identify occurrences of similar alarm floods as a method for situation assessment during alarm flood incidents (Lai and Chen, 2017). Sequence alignment enables multiple aspects of the analysis: New sequences can be compared in similarity to known sequences as a basis for classification. Additionally, given a group of similar sequences, their common events and respective timings can be used as features for classification of new sequence observations. The method proposed by Lai and Chen (2017) is based on dynamic programming for optimal alignment of multiple sequences. For n sequences of similar length M the complexity is estimated as $\mathcal{O}(n \cdot M^2)$.

A sequence is considered as a set of tuples (e_k, t_i), where $k \in \{1.K\}$, $i \in \{1.M\}$, where K is the number of event types and M is the number of events occured in the sequence. First the sequence is represented as a $K \times M$ time distance matrix representing the time span d_{ik} between the event i assigned to the column and the event type k represented by each row. A weighing function is then applied to the time distance matrix:

$$w_{ik} = \exp\left(-\frac{d_{ik}^2}{2\sigma}\right)$$

where σ can be chosen to blur the order of the event by considering close-by matches in terms of timing as well (Lai and Chen, 2017).

While the similarity of sequences for an operating procedure should be considered a given, the common sequence alignment can be used to refine the reference for the sequence. Potentially the sequences reveal additional events whose correlation with the sequence is not obvious from plant documentations but can be used for diagnosis. A draw-back of a complete sequence alignment approach like this one is the analysis complexity of finding the total alignment of a large number of procedure executions. Guo et al. (2017) propose a simple matching based on a sliding window around the time projection of one sequence onto the other to determine the likelyhood of two sequences to match. However, to establish a reference pattern for a procedure the actual alignment is most relevant.

3.4. Execution detection

In order to identify the sequences representing a procedure, time windows with the known plant action can be detected in the log files based on the expected events, such as valve open-close signals. The events are analysed in a matrix representation where each row corresponds to a time instance and the columns represent the relevant logged events. The occurrence of an event at a specific time is marked by 1 while all other elements are 0. A transition log is compiled from the event sets associated with the respective transitions, corresponding to procedure steps. To distinguish the procedure steps from isolated manipulations a slid-

Time	Event	
00:13.880	CV4:Off	s, a0
00:14.880	CV1:Off	
00:16.880	CV6:On	
00:18.980	CV8:On	
00:38.840	CV3:Off	
03:23.960	CV3:On	a1
03:54.980	CV6:Off	
03:56.039	CV14:Off	
03:58.000	CV5:On	
04:00.999	CV17A:On	
04:03.000	CV17B:Off	
11:08.080	CV8:Off	ab
11:08.080	CV7:On	
11:11.080	CV17B:On	
11:11.879	CV17A:Off	
11:14.535	FC11:AUTO	
11:14.535	FC11:SP3	
11:49.536	CV11:On	
25:19.535	FC11:SP3	c0
25:19.535	FC11:SP1	
26:08.160	CV3:Off	
26:10.397	CV11:Off	
28:18.400	CV7:Off	c2*
28:20.400	CV1:On	
28:24.539	CV14:On	
28:26.540	FC3:SP3	
28:26.540	FC3:AUTO	
28:53.000	CV3:On	
32:28.420	FC3:SP3	c3, c4
32:28.420	FC3:AUTO	
32:53.020	CV3:Off	
33:18.360	CV5:Off	
33:20.360	CV4:On	
33:11.800	FC3:SP3	
33:23.360	FC3:SP1	
33:23.360	CV3:On	

Fig. 11. Total sequence alignment of a random sample of 50 sequences of back-washing with air scouring. Times in minutes, seconds, and milliseconds ("mm:ss.000") reflect the average relative to the start of the procedure execution.

ing window of size W_{th} needs to contain an occurrence of every event in the set. This is illustrated by the first event a_1 in Fig. 3. To additionally account for undocumented changes to the actual implementation of the procedure, this constraint is relaxed so that all instances where the window contains the maximum number of events associated with the transition are considered as occurrences.

Given this transition log, identification of the procedure executions is a matter of finding all sequences where the transitions occur in order, or at least the majority of the procedure execution matches. If all transitions of the procedure are associated with time conditions, the documented procedure can directly be interpreted as sequence and all potential executions are detectable by a sequence projection onto the transition log as used by Guo et al. (2017). More generally, the procedure is considered to be delimited by its first and last transition. Consequently, all time windows contained between a detection of the first and last transition are considered execution candidates. If the first or last transition consist of only one action, the sequence of the first or last two transitions respectively determines the limits of an execution candidate, to maintain the distinction from individual actions. Additionally, missing start or end identifiers can be substituted by an expected execution time to perform analysis of faulty executions. Alternatively, the candidates can be discarded as incomplete executions. Fig. 3 and Fig. 4 show a log file and the respective detection windows for transition candidates. Since T_{s_3,o_1} only contains one event in the considered automaton shown in Fig. 1, the last two transitions mark the end of an execution candidate to avoid confusion with isolated events.

Table 1
Configuration transitions of an individual filter unit. The actions correspond to valve signals reported by the SCADA system.

Transition	Action set
s	FCV3:Off
a0	CV1:Off, CV4:Off, CV6:On, CV8:On
a1	CV3:On, CV6:Off, CV14:Off, CV5:On, CV17A:On, CV17B:Off
a2	CV17A:Off, CV17B:On
ab	CV8:Off, CV7:On, CV11:On
b	CV1:Off, CV4:Off, CV14:Off, FCV3:On, CV5:On, CV7:On, CV11:On
c0	FCV3:Off, CV11:Off
c1	CV5:Off, CV7:Off
c2	CV1:On, CV5:On, CV14:On, FCV3:On
c3	FCV3:Off
c4	CV5:Off, CV4:On, CV3:On

3.5. Fast sequence analysis

Once the execution candidates are determined, the sequence of transitions during within each execution can be identified. This sequence of transitions is called a trace in the automaton and is represented by a directed path in the automaton graph. Assuming an accurate documentation, the trace occurring most frequently resembles the correct procedure execution. Other traces might be related to manual operations or could hint toward common mistakes if they appear in significant numbers. These other traces will not correspond to consistent paths in the automaton, and require the adaptation of the automaton representation if they are found to accurately reflect the procedure execution.

The complete analysis for representing a procedure as an automaton validated by logged data is outlined in Fig. 5. First the documented procedure is represented as an automaton with the expected sequence of transitions and the action sets corresponding to each transition. Based on the established transition sets, the event and alarm log from SCADA for a reference period is transformed into a transition log as shown in Fig. 4. Procedure execution candidates can then be identified by corresponding pairs of the start and end transitions as described in the previous section. For each execution candidate the trace is identified from the timed sequence of the occurring transitions and classes of identical traces are determined. Subsequently, the automaton structure and event sets per transition can be refined using the relevant classes of traces. Finally, the reference statistic for the transition executions can be determined from the occurrences within each class.

4. Industrial case study

As a case study we investigate the automatic back-washing modes of a sea-water fine filter unit, shown in Fig. 6. The filtration system consists of four parallel fine filter units, three of which are always in operation, while one may be back-washed. The sequence of valve operations is outlined in Table 1 and Fig. 7. These operations can be executed in two different procedures, leading either from normal operation to air scouring or directly to back-washing with water.

Table 2
Summary of Transition log, corresponding alarm and resulting sequence candidate.

Event	Occurrences
s	3447
a0	974
a1	963
a2	3841
ab	973
b	41
c0	1027
c1	10
c2	1014 *
c3	3447
c4	1009
DP alarm	1733 [1]
Start	1009 [2]

* CV5:Off never detected
[1] high differential pressure
[2] (s,a0) and (s,b) sequences

The operations manual states that air scouring only has to be used in rare cases, when the filter is severely clogged. Normally, the filter is back-washed with water when the differential pressure exceeds a preset threshold or at least once a day.

In the case study the proposed sequence identification is applied to detect sequence candidates. The sequences are then analysed in terms of alarm dependency using the presented pattern mining approach and the proposed fast sequence analysis is used to determine the reference automaton for normal execution which is evaluated for its completeness by multiple sequence alignment.

Identifying the documented transitions with window size $T_{th} = 30s$ from the alarm and event log generated by the SCADA system, the transition log and sequence candidates were established. Table 2 summarises the transition log. The two paths in the automaton are marked by the combinations (s, a0) and (s, b) as start of a sequence candidate, while any occurrence of c4 terminates a sequence candidate. Since the respective high differential pressure alarm occurred over 70% more often than the procedure was executed, the conditional probability can never approach 1. This discrepancy was likely caused by the additional logic restricting back-washing operation to only one out of four parallel units. Therefore, this sequence could not be reliably related with the alarm occurrence.

According to the operations manual the transitions conditioned with the flow measurement $Q = 0$ could be equivalent to waiting up to 3min in the proceeding state. However, the data analysis revealed these transitions to consistently occur within 20s after the preceding transition. This period is within the chosen detection interval for the action set of the same transition. Thus the combinations (s, a0), (s, b), and (c3, c4) were considered as complete transitions leading to the reduced automaton shown in Fig. 10. Furthermore, transition c1 is rarely detected and no complete occurrences of c2 were found in the analysed log files. Upon closer inspection, it was found that in fact the occurrences of c2* reliably contain the action set [CV7:Off, CV1:On, CV14:On, FCV3:On], effectively combining (c1, c2).

The identified transition traces in the execution candidates are visualised in Fig. 9, based on the reduced automaton. Some traces are incomplete since the succession of two start or end markers respectively was corrected by limiting the sequence to a 45min interval. The grouping reveals back-washing with air scouring as the most frequent procedure execution and back-washing with water as the second consistent trace. The time constraints shown in Fig. 10 were based on the two most frequent traces, which were considered the ground truth for the plant operation. To verify the identified traces the alignment of all sequences identified as back-washing with air scouring was determined. Fig. 8 shows the pairwise similarity scores and the respective sequence length, in terms of number of events during the execution period. All scores are relatively high at above 0.5 and the larger clusters show some correlation to the sequence length. The common alignment of a random sample of approximately 5% of sequences with the most frequent trace shown in Fig. 11 does not indicate any consistent deviations from the considered action sets. While the action sets are consistent, the order of the specific events varies which is reflected by the slight offset of the sequence start from time 0. The reported set point changes of flow controllers FC3 and FC11 appear as additional information, which could not be anticipated from the procedure documentation.

5. Conclusion

The transitions between operational modes as described in the procedures of the operations manual can be used to define diagnostic automata and identify relevant transitions in alarm and events logs. While the actual execution should follow the operations manual, our case study revealed some significant deviations allowing the consideration of a simplified automaton for the procedure diagnosis. The plant documentation provides a reliable basis for analysis, but due to operator habits or undocumented changes to the plant operation the ground truth represented in daily operation can differ. This highlights the complementary nature of engineering documentation and recorded operations data.

The presented method for execution detection and the action set based analysis of procedure executions enables the quick design of diagnostic automata validated by data from the specific plant. Incorporating well documented operating procedures into a diagnostic system allows more comprehensive support, for instance by tracking the proper execution or considering operating procedures in a prognosis for developing faults. The proposed validation process can be a valuable tool to revise operation culture if deviations from the designed procedures affect efficiency. In continuation of the presented work we are investigating the on-line adaptation of the causal presentation for accurate on-line diagnosis.

Declaration of interests

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

CRediT authorship contribution statement

Denis Kirchhübel: Conceptualization, Methodology, Investigation, Writing - original draft, Writing - review & editing. **Morten Lind:** Supervision, Writing - review & editing, Project administration. **Ole Ravn:** Project administration, Supervision.

Acknowledgements

This project is funded by the Danish Hydrocarbon Research and Technology Centre.

We would like to express our gratitude for the collaboration and discussion with Tongwen Chen, Shiqi Lai and Wenkai Hu at the Advanced Alarm Management and Design group of the University of Alberta. And finally, we thank our co-workers in the Automation and Control group at the Electrical Engineering Department of the Technical University of Denmark for proof reading and constant input to the project.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.compchemeng.2020.106940

References

Abdallah, I., Gehin, A.L., Ould Bouamama, B., 2018. On-line robust graphical diagnoser for hybrid dynamical systems. Eng. Appl. Artif. Intell. 69, 36–49. doi:10.1016/j.engappai.2017.12.002.

Abele, L., Anic, M., Gutmann, T., Folmer, J., Kleinsteuber, M., Vogel-Heuser, B., 2013. Combining knowledge modeling and machine learning for alarm root cause analysis. IFAC Proc. Vol. 46 (9), 1843–1848. doi:10.3182/20130619-3-ru-3018.00057.

Beebe, D., Ferrer, S., Logerot, D., 2013. The connection of peak alarm rates to plant incidents and what you can do to minimize. Process Saf. Prog. 32 (1), 72–77. doi:10.1002/prs.11539.

Bezerra, A., Silva, I., Guedes, L.A., Silva, D., Leitão, G., Saito, K., 2019. Extracting value from industrial alarms and events: a data-driven approach based on exploratory data analysis. Sensors 19 (12). doi:10.3390/s19122772.

Chen, T.-Y., Chang, C.-T., 2019. Design approach to synthesize, validate, and evaluate operating procedures based on untimed automata and dynamic simulation. Ind. Eng. Chem. Res. 58 (19), 8172–8183. doi:10.1021/acs.iecr.9b00015.

Zhou, B., Hu, W., Chen, T., 2019. Cross-process alarm flood similarity analysis based on abstracted alarm descriptors. 2019 IEEE 28th Int. Symp. Ind. Electron.. IEEE, pp. 1749–1754.

EPA (United States Environmental Protection Agency), 2007. Guidance for preparing standard operating procedures. EPA QA/G-6 (April).

EEMUA (Engineering Equipment Materials Users' Association), 2013. Alarm systems – a guide to design, management and procurement. EEMUA Publ. 191.

Folmer, J., Schuricht, F., Vogel-Heuser, B., 2014. Detection of temporal dependencies in alarm time series of industrial plants. IFAC Proc. Vol. 47 (3), 1802–1807. doi:10.3182/20140824-6-ZA-1003.01897.

Goel, P., Datta, A., Mannan, M.S., 2017. Industrial alarm systems: challenges and opportunities. J. Loss Prev. Process Ind. 50, 23–36. doi:10.1016/j.jlp.2017.09.001.

Guo, C., Hu, W., Lai, S., Yang, F., Chen, T., 2017. An accelerated alignment method for analyzing time sequences of industrial alarm floods. J. Process Control 57, 102–115. doi:10.1016/j.jprocont.2017.06.019.

Hollifield, B.R., Habibi, E., 2006. The alarm management handbook : A Comprehensive guide. PAS.

Hollnagel, E., 2002. Time and time again. Theor. Issues Ergon. Sci. 3 (2), 143–158. doi:10.1080/14639220210124111.

Hu, W, Chen, T, Shah, S. L., 2018. Discovering association rules of mode-dependent alarms from alarm and event logs. IEEE Trans. Control Syst. Technol. 26 (3), 971–983. doi:10.1109/tcst.2017.2695169.

Hu, W., Shah, S.L., Chen, T., 2018. Framework for a smart data analytics platform towards process monitoring and alarm management. Comput. Chem. Eng. 114, 225–244. doi:10.1016/j.compchemeng.2017.10.010.

Hu, W., Wang, J., Chen, T., Shah, S.L., 2017. Cause-effect analysis of industrial alarm variables using transfer entropies. Control Eng. Pract. 64, 205–214. doi:10.1016/j.conengprac.2017.04.012.

IEC (International Electrotechnical Comission), 2014. Management of alarms systems for the process industries. IEC 62682 Edition 1., 13.

ISA (International Society of Automation), 2009. Management of alarm systems for the process industries. ANSI/ISA 18.2–2009.

Kanse, L., Parkes, K., Hodkiewicz, M., Hu, X., Griffin, M., 2018. Are you sure you want me to follow this? a study of procedure management, user perceptions and compliance behaviour. Saf. Sci. 101, 19–32. doi:10.1016/j.ssci.2017.08.003.

Kirchhübel, D., Lind, M., Ravn, O., 2019. Toward comprehensive decision support using multilevel flow modeling. In: Proceedings of the 5th IFAC Conf. Intell. Control Autom. Sci.. IFAC-PapersOnLine, Belfast, UK, pp. 42–47.

Lai, S., Chen, T., 2017. A method for pattern mining in multiple alarm flood sequences. Chem. Eng. Res. Des. 117, 831–839. doi:10.1016/j.cherd.2015.06.019.

Larsson, J.E., Öhman, B., Calzada, A., 2007. Real-time root cause analysis for power wrids. In: Secur. Reliab. Electr. Power Syst.,Tallinn, Estonia, pp. 1–7.

Lucke, M., Chioua, M., Grimholt, C., Hollender, M., Thornhill, N.F.a., 2019. Advances in alarm data analysis with a practical application to online alarm flood classification. J. Process Control 79, 56–71. doi:10.1016/j.jprocont.2019.04.010.

Lunze, J., 2004. Complexity reduction in state observation of stochastic automata. IFAC Proc. Vol. 37 (18), 339–344. doi:10.1016/s147466701730769-3.

Mo, K., Lee, S.J., Seong, P.H., 2007. A neural network based operation guidance system for procedure presentation and operation validation in nuclear power plants. Ann. Nucl. Energy 34 (10), 813–823. doi:10.1016/J.ANUCENE.2007.04.008.

Quiñones-Grueiro, M., Prieto-Moreno, A., Verde, C., Llanes-Santiago, O., 2019. Data-driven monitoring of multimode continuous processes: a review. Chemom. Intell. Lab. Syst. 189, 56–71. doi:10.1016/j.chemolab.2019.03.012.

Rothenberg, D.H., 2009. Alarm Management for Process Control: A Best-practice Guide for Design. Momentum Press, New York.

Shang, J., Chen, T., 2019. Early classification of alarm floods via exponentially attenuated component analysis. IEEE Trans. Ind. Electron. 1. doi:10.1109/tie.2019.2949542, In press.

Soares, V.B., Pinto, J.C., de Souza, M.B., 2016. Alarm management practices in natural gas processing plants. Control Eng. Pract. 55, 185–196. doi:10.1016/j.conengprac.2016.07.004.

Song, M., Gofuku, A., Lind, M., 2020. Model-based and rule-based synthesis of operating procedures for planning severe accident management strategies. Prog. Nucl. Energy 123 (February), 103318. doi:10.1016/j.pnucene.2020.103318.

Srinivasan, R., Viswanathan, P., Vedam, H., Nochur, A., 2005. A framework for managing transitions in chemical plants. Comput. Chem. Eng. 29 (2), 305–322. doi:10.1016/j.compcheng.2004.09.024.

Venkatasubramanian, V., 2019. The promise of artificial intelligence in chemical engineering: is it here, finally? AIChE J. 65 (2), 466–478. doi:10.1002/aic.16489.

Viswanathan, S., Johnsson, C., Srinivasan, R., Venkatasubramanian, V., Ärzen, K.E., 1998. Automating operating procedure synthesis for batch processes: part i. knowledge representation and planning framework. Comput. Chem. Eng. 22 (11), 1673–1685. doi:10.1016/S0098-1354(98)00228-2.

Wang, J., Li, H., Huang, J., Su, C., 2015. A data similarity based analysis to consequential alarms of industrial processes. J. Loss Prev. Process Ind. 35, 29–34. doi:10.1016/j.jlp.2015.03.005.

Wang, J., Yang, F., Chen, T., Shah, S.L., 2016. An overview of industrial alarm systems: main causes for alarm overloading, research status, and open problems. IEEE Trans. Autom. Sci. Eng. 13 (2), 1045–1061. doi:10.1109/tase.2015.2464234.