The University of Manchester

Statistical cluster analysis and visualisation for alarm management configuration

T D Butters, S Güttel, J L Shapiro and T J Sharpe

October 2014

MIMS EPrint: **2014.51**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://www.manchester.ac.uk/mims/eprints
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

Statistical cluster analysis and visualisation for alarm management configuration

***T D Butters*[*†], *S Güttel*[†], *J L Shapiro*[§], *T J Sharpe*[*]**

[*]*Sabisu, Arden Hall, Brooklands Road, Sale, M33 3SJ, UK, tim.sharpe@sabisu.co*
[†]*School of Mathematics, The University of Manchester, Manchester, M13 9PL, UK*
[§]*School of Computer Science, The University of Manchester, Manchester, M13 9PL, UK*

Keywords: Continuous Improvement, Preventative Action, Alarm Management, Asset Management.

Abstract

The effective performance of an alarm system is a key aspect of asset management for any industrial installation. However, it is not uncommon for alarm systems to be poorly configured, leading to large amounts of alarm noise and a potentially dangerous load on the operators. Here we present a novel method for the identification of redundant or bad actors in alarm systems through the application of statistical cluster analysis. This allows the system to be optimised to reduce the load on the operators through existing systems change management processes.

1 Introduction

Alarm systems play a vital safety role in alerting operators to unexpected behaviour, and in performing forced shut-downs if a severe danger is detected. It is well documented that failures in alarm systems can have catastrophic consequences with incidents such as the partial nuclear meltdown at Three Mile Island and the Texas City refinery explosion, both having root causes directly linked to suboptimal alarm management [1, 2]. It is therefore vital that alarm systems perform optimally as failure to do so poses a great risk to safety, asset viability, and profitability. The aim of this contribution is to provide a statistical analysis tool for gaining insight into the performance of alarm systems and guiding improvements to their configuration.

There are several metrics that can be used to quantify the performance of an alarm system. One of the most important is 'operator load', which is defined as the number of alarms each operator has to address in a given time interval. The global standard for alarm management configuration, EEMUA 191, allows for at most 1 alarm per operator every 10 minutes in the context of distributed control systems (DCS) [3].

Operator load gives a good indication of how well the alarm system is configured, and the viability of acknowledging and actioning each alarm in the appropriate manner. Alarm cascades, floods, or simply the proliferation of alarms relating to a single physical event are very problematic in practice. Complex manufacturing plants can see arrival rates of thousands of alarms per hour, producing an operator load so large it is impossible for the operators to properly deal with each fault. With such volumes it is impossible to manually prioritise the alarms and decipher the root cause of the problem [4]. It has been known for operators to switch off or discount 'noisy' alarms with serious consequences.

With sensors becoming cheaper it has become commonplace to install a large number across a plant, ensuring that every physical component is monitored. Although this may be necessary for some aspects of operation, it is usually unnecessary to configure an alarm for each single sensor, a practice that has also become common, particularly during initial configuration or where there is a lack of management continuity.

Often, sensors and their related alarms report on physically linked systems (*e.g.*, temperature conduction between components), causing several alarms for each physical 'event'. Identification of such redundancies would facilitate the optimisation of the alarm system, leading to a lower operator load and a fundamentally safer facility.

Although there are extant solutions available for the configuration of alarm management systems, many of them are complex in nature. They can require a computational model of the plant and permission to automatically intercept and suppress alarms in real-time [5, 6, 7]. These solutions are costly, require expensive and time consuming configuration, and cede control of a safety critical system to an algorithm.

Here we present an alternative approach to alarm management decision support which can be used to optimise the performance of alarm systems. This approach utilises a novel method of analysing alarm system performance which can identify clusters of alarms that frequently act together. This detects redundant alarms that provide little to no extra information, allowing these alarms to be analysed and following industry best practice change control processes, suppression rules enacted or alarms to be reconfigured. This reduces operator load and makes it easier for them to identify the physical cause of any alarm incident.

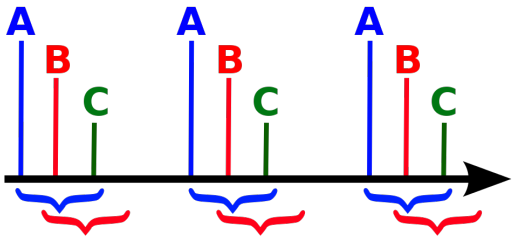


Fig. 1: Diagram showing basic cluster identification. Alarms **A**, **B** and **C** continually sound together. Each coloured bracket shows the search window for the corresponding alarm (the window for alarm **C** is not shown). Alarm **A** is the principle alarm for a cluster containing **B** and **C** as incident alarms, and alarm **B** is the principle alarm for a cluster containing alarm **C** as the incident alarm. There are no alarms within the search windows of **C**, therefore it is not the principle alarm for any cluster.

2 Method

The proposed method identifies clusters in historical alarm data which has been collected during the operation of a plant. This data-mining approach removes the need to construct complex models of the underlying processes, instead gaining the relevant information from the past behaviour of the facility. This requires very little configuration and ensures that the alarm management team retains complete control. To access, aggregate, and process the large datasets required the method was built in the Sabisu platform.

2.1 Cluster Identification

Alarms that act in clusters are likely to be linked in some physical way and therefore contribute a level of redundancy that is likely to increase the operators' alarm load without providing extra information.

To identify such clusters the alarm log for the system is analysed and alarms that occur often within a specified time window of one another are marked as a linked pair. Each linked pair consists of a principle alarm ($\mathbf{P}$) and an incident alarm ($\mathbf{I}$), with the principle alarm *implying* the incident alarm: $\mathbf{P} \rightarrow \mathbf{I}$. It may be found that one principle alarm has several corresponding incident alarms, with many of these forming their own linked sets as shown in Figure 1.

Statistical thresholds ensure that the system will only identify alarm clusters if the alarms forming them occur together a large percentage of the time. This suggests that it is rare to see the principle alarm without its corresponding incident alarm(s), and therefore the incident alarm is adding very little extra information to a physical event. It is, however, needlessly increasing the load on the operator. The statistical thresholds used for these calculations guarantee not only that each incident alarm is often seen with its principle alarm when compared to the number of occurrences of the principle alarm, but also that the linked pairs are seen in the alarm log relatively often. On the other hand, an alarm phenomenon seen relatively infrequently will not be included in the analysis as the statistical confidence would not be considered high enough.

2.1.1 Dependent Alarms

Dependent alarms are a slightly different concept from alarm clusters as the statistical threshold linking a principle to an incident alarm is not *necessarily* met, but the incident alarm is *never* seen by itself, only ever following its principle alarm. This could identify redundancy, as with the cluster analysis, or it could represent a distinct separate event. Dependent alarms are automatically identified, but should be treated differently when considering the best course of action; the alarms constituting a dependent set may provide extra information compared to the set's principle alarm alone, so simply removing the alarm may not be appropriate. However, it may be the case that a set of dependent alarms could be replaced with a single alarm to indicate this event.

2.2 Visualisation

Although reporting identified clusters as a simple text list is fast to render and non-ambiguous, it is harder to see the alarms in context and to compare the importance of clusters quickly. Visualisation is therefore useful, providing extra insight into the links between the alarms and giving a good overview of the whole system. For this application two visualisation methods were chosen; a dynamically sortable adjacency matrix and a force directed graph. Both of these are implemented using the D3 visualisation library (http://www.d3js.org) and the Sabisu platform.

2.2.1 Force Directed Graph

A force directed graph (FDG) is a natural way to visualise alarm cluster data. In this representation each alarm appears as a node on the graph, with links between the nodes shown with arrows. The directions of the arrows follow that of the implications, *i.e.*, if alarm $\mathbf{A}$ is the principle alarm of a linked pair $\mathbf{A}$ and $\mathbf{B}$ then there will be an arrow from node $\mathbf{A}$ to node $\mathbf{B}$ on the FDG.

The number and size of clusters present in the alarm system is immediately apparent, with the directed nature of the graph allowing quick identification of the principle alarms it is easy to estimate the impact of applying suppression rules to an individual alarm. An example FDG is shown in Figure 2. From this graph it can be estimated that the application of the identified suppression rules triggered on alarm $\mathbf{A}$ will disperse the left cluster, reducing the number of alarms from 4 to 1 for each occurrence of alarm $\mathbf{A}$.

2.2.2 Adjacency Matrix

An adjacency matrix is a mathematical object that can be used to represent a graph by specifying which vertices are adjacent

to one another [8]. For our purposes it is possible to ignore the link directions, so that the adjacency matrix $\mathbf{M}$ associated with the FDG is symmetric. More precisely, we set:

$$\mathbf{M}_{ij} = \mathbf{M}_{ji} = \begin{cases} 1, & \text{if alarms } i \text{ and } j \text{ are linked} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

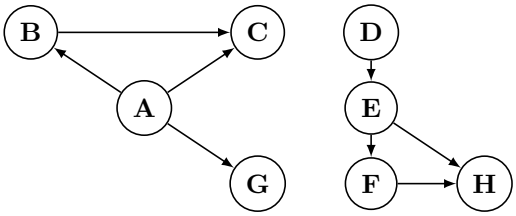
By representing the entries of the adjacency matrix $\mathbf{M}$ on a square grid (a so-called 'sparsity plot'), clusters of alarms can easily be identified. More precisely, given a link between alarm i and alarm j, the entry $\mathbf{M}_{ij}$ will be coloured. Different colours are used to group grid squares that are members of the same cluster, with the opacity of the square relating to the strength of the connection.

The adjacency matrix can be sorted in three ways, each sorting giving a different insight:

i. Ordering the columns and rows of $\mathbf{M}$ by alarm name makes it easy to locate an alarm of interest and quickly identify the key alarms it is linked with.

ii. Cluster ordering attempts to group each cluster together so that their relative sizes can be assessed.

iii. Ordering by frequency, which is a measure of the 'connectivity' of an alarm. For an individual alarm, the more alarms linked with it, and the more times it appears in the alarm log, the higher its frequency. This allows each alarm to be ranked by its effect on the alarm load, with the worst offending alarms in the top left of the matrix. This provides an alternative ranking method to cluster size, as although size is a good indicator it could be beneficial to deal with smaller clusters that are more active first, as they could have a larger overall effect on the alarm load. An example alarm graph is shown in Figure 2, with the symmetric adjacency matrix ignoring the link directions shown below.



	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**
A	0	1	1	0	0	0	1	0
B	1	0	1	0	0	0	0	0
C	1	1	0	0	0	0	0	0
D	0	0	0	0	1	0	0	0
E	0	0	0	1	0	1	0	1
F	0	0	0	0	1	0	0	1
G	1	0	0	0	0	0	0	0
H	0	0	0	0	1	1	0	0

Fig. 2: A simple force directed graph example showing the links between alarms (above). The adjacency matrix (below) does not take into account the link directions and is therefore symmetric.

2.3 Suppression Rules and Removal

The results of this analysis provide information about redundancy in the alarm system, which can be dealt with in different ways depending on the underlying cause.

2.3.1 True Redundancy

True redundancy occurs when an alarm provides no extra information to the operator under any circumstances; the redundancy is detected for all modes of operation and is entirely independent of the running conditions of the facility. If true redundancy is detected it would be recommended that the alarm is removed from the system entirely.

2.3.2 Mode Dependent Redundancy

Mode dependent redundancy indicates that during certain times the alarm may provide useful information, but becomes unimportant during certain running modes, *e.g.*, when a certain feedstock is being used, or the product slate is changed. Removal of the alarm would be undesirable, however, mode dependent suppression rules triggered during the redundant periods of the alarm should be implemented. This reduces the alarm load without compromising the safety of the facility by retaining the important information delivered by the alarm system.

The specific suppression rules used depend on the settings used to analyse the system, with incident alarms suppressed on detection of their principle alarm for the same length of time as the window used for the cluster identification. For example, if a 30 s search window was used, the incident alarms are suppressed for 30 s whenever the principle alarm is detected.

2.3.3 Suppression Safety

Any suppression rules are assessed as part of the alarm management change control process to ensure that it is safe to suppress the alarm for the specified period. For high priority alarms it may not be possible to apply the specific suggested rule, but a shorter suppression window may still be beneficial. To quantitatively determine this the cluster analysis could be performed with a shorter search window.

3 Case Study

A two week section of alarm log data from a large-scale industrial plant was made available for analysis. On-site alarm experts considered the system to perform in an acceptable manner, but were interested in finding ways in which its perfor-

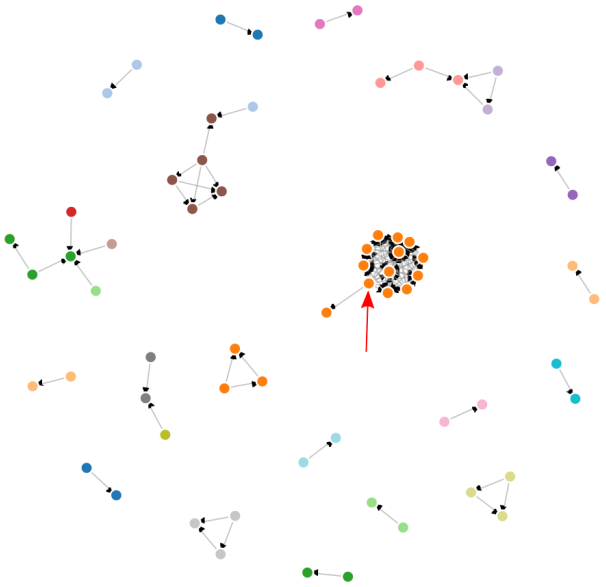


Fig. 3: A force directed graph (FDG) of the identified clusters from the industrial alarm data. The principle alarm for the largest cluster is marked with a red arrow.

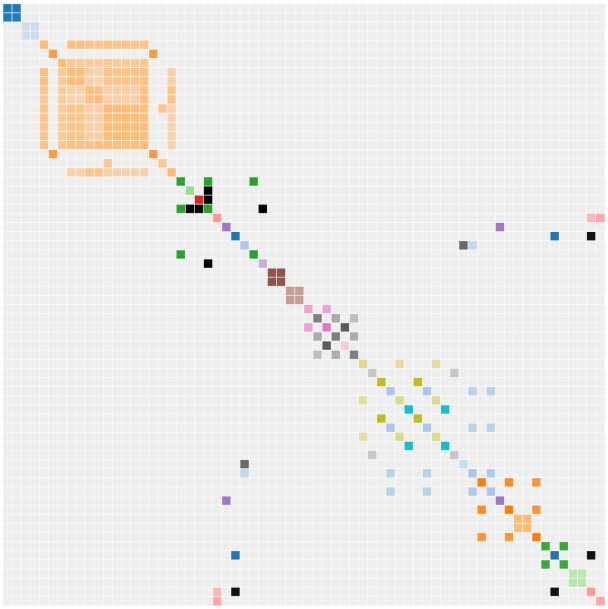


Fig. 4: Adjacency matrix of the industrial alarm data ordered by alarm name.

mance could be optimised further. This analysis was performed with a minimum link threshold of 70%, a minimum count threshold of 50 occurrences, and a search window width of 30 s.

3.1 Cluster Identification

As can be seen in Figure 3, several clusters of varying sizes were identified. The largest cluster contains 13 alarms, with a principle alarm directly identifiable from the FDG. There are also a relatively large number of linked alarms, which could also have a significant effect on the alarm load. Some of the clusters contain alarms that are also members of other clusters, meaning that the larger cluster can be separated into two or more sub-clusters that share one or more incident or principle alarms. These appear as connections between alarms of different colours on the FDG. If the sub-clusters share an incident alarm it is likely that two suppression rules would need to be implemented to completely disperse the cluster.

Figures 4–6 show the different adjacency matrix orderings for the data. The name ordering in Figure 4 is useful in finding specific alarms, but does not group clustered alarms together. However, it is clear that even with this sorting clusters are present, which indicates that alarms with similar names are acting together. If alarms are named using a hierarchical structure, (*e.g.*, `Pump3_Alarm-A`, `Pump3_Alarm-B`, *etc.*...) this would be predicted, as due to inherent redundancy several alarms from the same area are triggering together for single events.

Figure 5 shows the adjacency matrix ordered by cluster, pro-

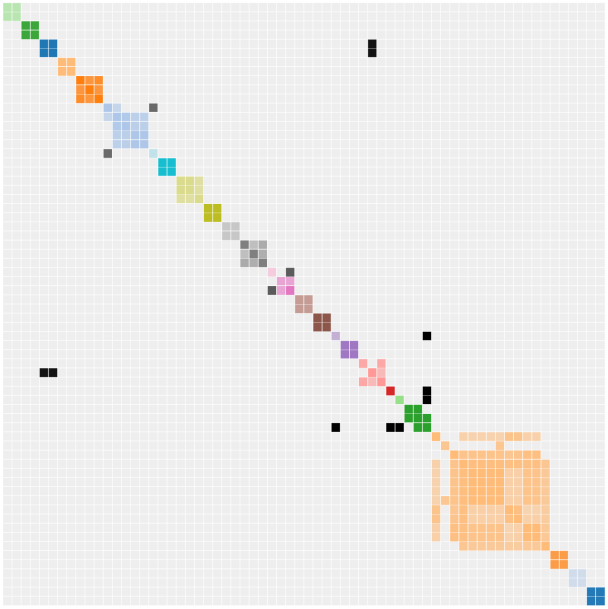


Fig. 5: Adjacency matrix of the industrial alarm data ordered by cluster. This attempts to group all of the alarms that are members of the same cluster into square blocks on the matrix.

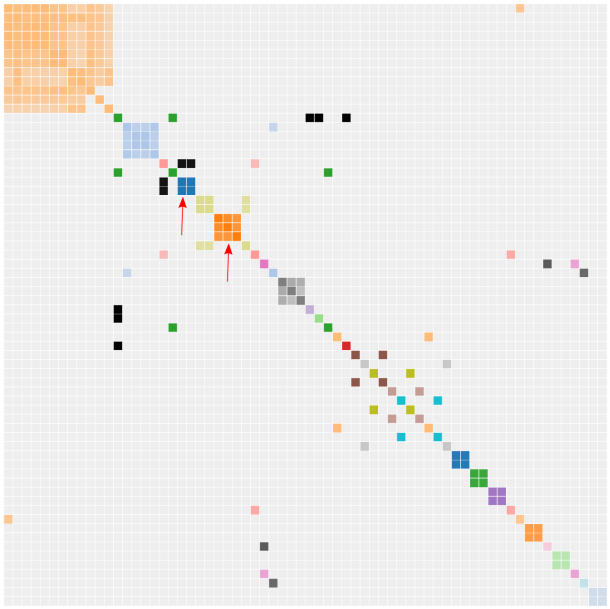


Fig. 6: Adjacency matrix of the industrial alarm data ordered by frequency. This orders each alarm by its connectivity, with the most active alarms in the top left of the matrix and the least active in the bottom right.

viding similar information to the FDG. The sizes of the clusters are easily assessed, with the number of diagonal elements in a group indicating the cluster size. From this view it is easy to see that there are a number of small clusters and one larger cluster. The elements that lie off the diagonal indicate alarms that are members of more than one cluster.

The frequency ordering in Figure 6 shows which alarms are most active in the system. This is an important view as it provides a method of intelligently ranking the clusters. The large cluster appears in the top left of the matrix, indicating that this would be the most beneficial cluster to target first, though given its size relative to the rest of the alarm clusters and prominence in other views this might be expected. However, without this ordering view it would be difficult to prioritise the smaller clusters; the 2-alarm blue cluster has a higher frequency than the larger 3-alarm orange cluster (both marked with red arrows in Figure 6), which shows that a greater effect on the alarm load would be seen by addressing this smaller cluster first. This ranking mechanism is important as the alarm system changes will be enacted through the site's change management system, and therefore will not be implemented immediately.

3.2 Alarm System Improvements

The initial performance level of the considered alarm system is good, with an acceptable operator load under normal conditions and relatively small number of clusters identified through this analysis. However, it is still possible to improve this system, and the application of the suggested suppression rules from this analysis reduces the average operator load over the two week period by more than 12%, eliminating 6233 alarms. The load was reduced by much larger amounts during highly active periods, significantly improving safety during these times by alleviating the pressure on the operators.

4 Discussion

The efficient operation of an alarm management system is a key part of effective asset management. Not only does a poorly performing alarm system have severe safety implications, but also the timely acknowledgement and amendment of faults leads to a reduction in unexpected downtime. Greater clarity in the control room also allows operators and shift managers to make better decisions, running the plant more efficiently and profitably.

As shown in the example case study the alarm management software developed using Sabisu provides a highly effective method for identifying redundancy in alarm systems. The suggested suppression rules also give important information on how to rectify this, with upper limits on suppression windows required to eliminate clusters of alarms.

The visualisation tools allow extensive analysis to be performed by the user, with both high-level overviews of the system as well as mechanisms to dissect the internal structure of the alarm interactions. Through these methods the importance of the identified clusters can be assessed based on their impact on operator load and the appropriate decision made.

The elimination of clusters has a positive effect on alarm system performance, even for relatively efficient systems such as the one shown in the case study. It is predicted that a greater impact would be seen when analysing a poorly configured system where clusters of redundant alarms are more frequent and larger. The analysis was performed quickly, with the results returned by the software in $\sim$3 s. This includes the generation of a text list identifying principle alarms and their clusters, as well as the necessary data to produce the FDG and adjacency matrix visualisations.

4.1 Dependent Alarms

The analysis from the given case study focuses entirely on the identified clusters and does not include the dependent alarm results. This is because dependent alarms potentially indicate groups of alarms that can be revised and replaced with a single alarm. This is especially true if the same group appears in the cluster analysis and the dependent results. However, to enact changes of this nature a longer time period would need to be analysed.

However, the initial dependent results suggested 10 clusters of alarms that should be investigated, several of which were identified by the on-site alarm management team as responses to a single physical event.

4.2 Journal Alarms

Although the emphasis in alarm management optimisation is often on the operator load, it is still beneficial to process 'journal alarms' that do not produce an audible alert and provide information about non-critical aspects of operation. Although these are often not a safety concern they could act as principle alarms for non-journal audible alarms, and so could be used to trigger appropriate suppression rules to disperse clusters. Logging large numbers of events is also undesirable due to the resulting strain on the network and database connection. As journal alarms also form clusters the proposed method is effective in alleviating this strain.

Journal alarms can also be used to elucidate the cause of some alarm clusters, and in doing so show where it may be useful to upgrade some key journal alarms that were acting as principle alarms in a cluster, while suppressing the incident alarms. This would alert the operator to the event at the earliest possible stage whilst still reducing their overall load.

4.3 Modes of Operation

Many industrial assets can run in different modes depending on the current conditions, *e.g.*, a different feedstock being used for a petrochemical manufacturing process, a change in the required product quality, or variations in environmental conditions. Therefore it cannot be assumed that an alarm system will behave in the same way indefinitely, though its behaviour is likely to be constant throughout each mode of operation. It is therefore beneficial to run cluster analysis for each operating mode to ensure the correct settings are in place. With an analysis time of $\sim$3 s it is possible to quickly prepare and repeat the analysis periodically to ensure any emerging clusters are identified.

4.4 Future Work

Work on extending the functionality of the alarm analysis software through the automatic estimation of the alarm system transition matrix is well advanced. This allows higher order links between alarms to be found, effectively highlighting any links between clusters. This provides another method of cluster ranking by identifying any clusters that in-turn may lead to other clusters being activated [8].

Within this work the transition matrix is also used for predictive purposes to estimate the future state of the alarm system [8]. This is done in a semi real-time way to predict short-term future behaviour, or alternatively to assess the characteristic behaviour of the alarm system under certain conditions. For example, it could be used to track the predicted progression of alarms given the activation of every alarm in a certain area of the facility, or to track the overall number of alarms given the activation of a large portion of the system, *i.e.*, given 80% activation of alarms how many are active within the next 4 time steps. This would provide information on whether the number of alarms tends to increase, decrease or remain constant within the system.

5 Conclusions

The alarm management optimisation software developed using Sabisu provides a novel method to enhance the performance of industrial alarm systems. The identification of redundancy allows a significant reduction in the operators' alarm load without compromising safety. The visualisation tools provide a high level of insight into the current state of the alarm system, and simple suppression rules are provided to disperse any detected clusters. The decision support nature of this method ensures that the alarm management team remains in complete control of the system, with all changes being incorporated through the site's existing change management structure.

Acknowledgements

This work was partially funded by the Technology Strategy Board Knowledge Transfer Partnership – KTP 9315.

References

[1] L. M. Toth. "The three mile island accident diagnosis and prognosis", *189th Meeting of the American Chemical Society.*, (American Chemical Society, 1986).

[2] D. Holmstrom, F. Altamirano, J. Banks, G. Joseph, M. Kaszniak, C. Mackenzie, R. Shroff, H. Cohen, S. Wallace. "CSB investigation of the explosions and fire at the BP Texas City refinery on March 23, 2005", *Process Safety Progress*, **25(4)**, pp. 345–349, (2006).

[3] EEMUA 191. *Alarm Systems - A guide to design, manufacturing and procurement*, 3rd edition, (The Engineering Equipment and Materials Users' Association, 2013).

[4] R. B. Whittingham. *The Blame Machine: Why Human Error Causes Accidents*, (Elsevier Butterworth-Heinemann, 2004).

[5] J. Liu, K. W. Lim, W. K. Ho, K. C. Tan, R. Srinivasan, A. Tay. "The intelligent alarm management system", *IEEE Software*, **20(2)**, pp. 66–71, (2003).

[6] Y. Xu, Q. Zhu. "Alarm optimization for process industry based on matter-element analysis", *Journal of Chemical Industry and Engineering (China)*, **7**, p. 2, (2008).

[7] J. Liu, J. Zhou, S. L. R. Li, K. Tan, W. K. Ho, R. Srinivasan, A. Tay. "Intelligent alarm management through suppressing nuisance alarms and providing operator advice", *Proceedings of the 4th IFAC Workshop On-Line Fault Detection and Supervision*, pp. 81–86, (2001).

[8] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*, (Society for Industrial and Applied Mathematics, 2000).