Decentralized causal-based monitoring for large-scale systems: sensitivity and robustness assessment

Rodrigo Paredes [1], Wei-Ting Yang [2], Marco S. Reis [1]

[1] *University of Coimbra, CERES, Department of Chemical Engineering, 3030-790 Portugal (Tel: +351 239 798 700; e-mail: marco@eq.uc.pt).*

[2] *Department of Data Science and Analytics, BI Norwegian Business School, Oslo, Norway*

Abstract: Ensuring safety and efficiency in industrial systems requires effective fault detection and diagnosis, which becomes increasingly challenging in high-dimensional and complex environments. Traditional multivariate statistical process monitoring methods, such as those based on Principal Component Analysis and Partial Least Squares, often fall short in their ability to diagnose localized faults due to their lack of causal modeling. This paper introduces a Causal Network-based Decentralized Multivariate Statistical Process Control (CNd-MSPC) framework, which employs causal networks and community detection—specifically the Leiden algorithm—to segment large systems into functional communities and perform distributed monitoring. This structural partitioning preserves essential causal and topological information, enhancing the sensitivity for fault detection in high-dimensional systems by allowing focused analysis of specific sub-networks. Through extensive testing with a graph-based data simulator, we demonstrate that CNd-MSPC consistently outperforms centralized methods across various network sizes, achieving higher fault detection sensitivity for both process perturbations and sensor biases, especially in large networks. The decentralized approach retains high sensitivity, even when data from several communities are missing due to process disruptions.

Keywords: Process Monitoring; Fault Detection; Causality; Community Detection; High-dimensional.

1. INTRODUCTION

Fault detection and diagnosis in industrial and engineering systems are essential to ensure operational safety, efficiency, and quality consistency. In large-scale industrial processes, many interdependent variables must be simultaneously monitored to quickly detect faults that could affect the process (Isermann, 2006; Reis & Gins, 2017). Causal networks provide valuable insights into such cause-and-effect relationships (Pearl, 2000; Peters et al., 2017); however, most conventional statistical process monitoring (SPM) methods, such as PCA and PLS-based approaches, are non-causal by design and tacitly overlook causality. This limitation reduces their ability to detect localized faults and impairs fault diagnosis due to the well-known "smearing-out" effect of correlation-based methods (Kourti & MacGregor, 1996; Van den Kerkhof et al., 2013).

As the systems' dimensionality and complexity increases, both causal and non-causal SPM methods face challenges in maintaining accuracy and sensitivity to detect localized faults (Joe Qin, 2003; Rato & Reis, 2014). While Multivariate Statistical Process Control (MSPC) methods are effective for smaller systems, they often fail to detect localized faults in large-scale processes because the impact of these faults on global monitoring metrics is diluted (Chiang et al., 2000). To address this limitation, decentralized approaches, such as Distributed PCA (DPCA) and Distributed Canonical Correlation Analysis (DCCA), have been explored to reduce the monitoring dimensionality (Ge & Song, 2013; Peng et al., 2020). However, these methods overlook the physical topology and causal structure of the process.

In contrast, community detection algorithms—such as Louvain, Girvan-Newman, and Leiden— provide network partitions based on modularity and connectivity principles, that better align with the physical interdependencies of the system (Harenberg et al., 2014; Javed et al., 2018). This structural partitioning preserves critical topological information, increases sensitivity to local perturbations within specific subnetworks, and provides a more robust and interpretable monitoring framework in high-dimensional, complex systems. Furthermore, combined with the Markov Blanket concept, this approach enables a fully decentralized monitoring framework that is suitable for both software and hardware implementations (Paredes et al., 2022).

To address these challenges, this study assesses the impact of increased system dimensionality using a data simulator based on a graph generative framework, where both centralized and decentralized causal-based SPM methods are tested and evaluated for their fault detection sensitivity. The proposed approach, Causal Network-based Decentralized Multivariate Statistical Process Control (CNd-MSPC), begins by inferring the system's causal network and then partitioning it into functional communities using the Leiden algorithm. This division manages dimensionality by focusing monitoring efforts on specific sub-levels within the system. A likelihood metric is calculated for each community, and a global metric is composed from the community-level statistics.

In all tested scenarios, CNd-MSPC demonstrates better performance in sensitivity for fault detection while improving fault diagnosis through the incorporation of causal information. This paper is structured as follows. Section 2 presents the monitoring methods. Then, in Section 3 describes the data simulator and fault types used to evaluate high-dimensional scenarios. Section 4 presents the results obtained. Finally, Section 5 provides a discussion and summarizes the main conclusions.

2. LARGE-SCALE MONITORING METHODS

This section presents the two causal-based Statistical Process Monitoring (SPM) methods used in this work: CNc-MSPC, a causal method that monitors the complete causal network, and CNd-MSPC, the proposed method that builds communities for dimensionality reduction and performs monitoring on each sub-network. These approaches were applied to high-dimensional linear scenarios with Gaussian random root nodes and additive white noise. Fig. 1 shows the main stages of the statistical process monitoring procedures.

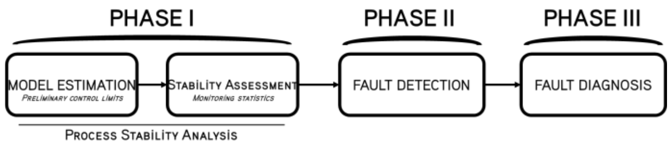

Fig. 1. Representative diagram of main stages to process monitoring.

2.1 CNc-MSPC

This causal methodology begins by inferring the causal network structure of the data using causal discovery methods. Here, we assume the causal network structure to be known, allowing us to focus on the effects of system dimensionality on fault detection performance.

In the initial stage, data are transformed using a structural causal model (SCM) that considers causal dependencies by regressing each variable X_i on its causal parents in the network, $\mathrm{Pa}(X_i)$. These yields $X_i = f(\mathrm{Pa}(X_i)) + e_i$, where the conditional distribution of $X_i|\mathrm{Pa}(X_i)$ is equivalent to the error term distribution e_i.

For fault detection, a local likelihood index L_i^L is calculated for each observation k. Assuming a normal distribution, the log-likelihood for an individual observation $L_i^L(k)$ is defined as:

$$L_i^L(k) = -\frac{\ln(2\pi)}{2} - \frac{\ln(\hat{\sigma}_{e_i}^2)}{2} - \frac{\left(e_i(k) - \hat{\mu}_{e_i}\right)^2}{2\hat{\sigma}_{e_i}^2} \quad (1)$$

where $e_i^{(k)}$ is the error term of variable X_i conditioned onto its parents, in the k-th observation, and $\hat{\mu}_{e_i}$ and $\hat{\sigma}_{e_i}$are the sample mean and standard deviation of $e_i$calculated from a reference dataset corresponding to normal operating conditions (NOC).

A global log-likelihood index L^G for the k-th observation is then computed to monitor the stability of the entire network by considering the joint distribution of the entire causal network. Leveraging the chain rule, the joint (or global) likelihood can be expressed as the product of individual likelihoods across the network's nodes. Therefore, applying logarithms, the global log-likelihood, $L^G(k)$, is defined in terms of the individual log-likelihoods, as:

$$L^G(k) = \sum_{i=1}^{n_{variables}} L_i^L(k) \quad (2)$$

Control limits for fault detection are established using the L^G obtained from a validation dataset. A kernel density estimation (KDE) was used to approximate the probability density function of these likelihood statistics (Silverman, 1986). The cumulative distribution function (CDF) of the fitted KDE is computed, and the Lower Control Limit (LCL) is identified by solving the following equation for a predefined Type I error rate, α:

$$\int_{-\infty}^{\mathrm{LCL}} \hat{f}_{KDE}(L^G)\ \mathrm{d}L^G = \alpha \quad (3)$$

where, $\hat{f}_{KDE}(L^G)$ is the estimated probability density function. Since high log-likelihood values indicate proximity to the reference distribution, only the Lower Control Limit (LCL) is needed for fault detection. This threshold is used for monitoring future observations, during Phase II (Fig. 1).

In Yang et al. (2022), this methodology was comparatively assessed against the classical PCA-MSPC approach using both a simulated and an industrial dataset from semiconductor manufacturing, showing similar performance in fault detection while demonstrating improved effectiveness in fault diagnosis.

2.2 CNd-MSPC

The proposed decentralized CNd-MSPC, also begins by inferring the causal network. Afterwards, the variables are divided into communities using a community detection algorithm that analyses the topology and density of the network. Such grouping into strongly connected communities effectively improves the fault detection sensitivity to localized faults. For this study, the Leiden algorithm was applied, including an extra refinement step not shared by the Louvain method. This step enhances the community detection quality by ensuring that communities are not only well-defined but also well-connected, addressing the limitations of the Louvain method in detecting disconnected communities (Traag et al., 2019).

The Leiden algorithm aims to maximize modularity Q, defined as:

$$Q = \frac{1}{2\mathrm{m}} \sum_{\mathrm{i,j}} \left(w_{ij} - \frac{\mathrm{k_i k_j}}{2\mathrm{m}} \right) \delta(\mathrm{c_i}, \mathrm{c_j}) \quad (4)$$

where w_{ij} is the edge weight between nodes i and j, k_i and k_j are their respective degrees, and $\delta(c_i, c_j)$ is a community indicator function (1 if i an j belongs to the same community, *i.e.*, $c_i = c_j$, and 0 otherwise). It consists of three main steps: (1) local moving of nodes to optimize modularity locally, (2) refinement to ensure that all communities are connected, and (3) aggregation of communities into super-nodes, with iterative repetition to achieve optimal partitioning. The resulting disjoint communities were evaluated using metrics of cohesiveness, separability, density, and *conductance* (a measure of the communication flow between a community and its neighbourhood) to assess the choice of parameters.

After the definition of the functional communities, a global log-likelihood metric (Equation 2) is built for each community (sub-level monitoring), denoted as L_c^G for $c = 1, 2, \ldots, n_{communities}$, where $n_{communities}$ is the number of communities. To aggregate the information from multiple communities we adopt an OR-gate approach (a global warning is issued, as soon as at least one community-level out-of-control signal is produced). In this study, CNd-MSPC monitors each L_c^G independently, with control limits set at the community level, with a significance level corrected to control the overall false alarm rate due to the use of multiple monitoring statistics. To control the false alarm rate due to multiple monitoring statistics, the Šidák correction is applied, setting the significance level for each community as $\alpha_c = 1 - (1 - \alpha)^{1/n_{communities}}$ (Šidák, 1967). The global alarm of the CNd-MSPC is then triggered if at least one community has a monitoring statistic below its control limit (OR-gate).

3. CASE STUDY

To evaluate both centralized and decentralized causal-based monitoring approaches, a data simulator based on scale-free directed acyclic graphs (DAGs) was constructed. This simulator can generate both normal and faulty data, including two types of faults analogous to those encountered in industrial contexts: sensor faults and process drift. Being static and linear, the simulator is well-suited for assessing simpler scenarios and investigating issues related to network complexity, dimensionality, and robustness. The performance of the methods was tested using networks with 50, 100, and 1000 nodes.

3.1. Network data simulator

The data simulator begins with the definition of a DAG $\mathcal{G}$, comprising $\mathcal{V}$ nodes representing the variables and $\mathcal{E}$ edges representing the relations between them. It should be noted that $\mathcal{V}$ can be divided into root nodes, if a node lacks parents, and non-root nodes, otherwise. The graph can be generated using various models, including potential-growth and forest fire models. However, to create a network structure that closely resembles real industrial scenarios, the Barabási-Albert model was used to generate scale-free networks (Albert & Barabási, 2002).

To examine the data sampling process, we consider Equations 5-9, from the algorithm presented in Table 1. The data simulator is built on several assumptions. The root nodes are assumed to follow a normal distribution, as indicated by Equation 5. Each link is assigned a coefficient obtained by sampling a Student's *t*-distribution with *DF* degrees of freedom (Equation 6). Additionally, an intercept is selected for each non-root node (Equation 7). The samples for each non-root node are generated using a causal approach, where each parent value of X_i is multiplied by the respective coefficient, plus the intercept (Equation 8). Data generation for the non-root nodes requires following a linear topological ordering of the graph nodes such that for every edge (v_x, v_y), from node v_x to v_y, v_x comes before v_y in the ordering, starting from the root nodes and proceeding along the DAG. Once the matrix **X** is constructed, measurement white noise is added to all variables using the Signal-to-Noise Ratio in decibels (Equation 9), enhancing the realism of the simulator.

$$X_i^{\text{root}} \sim \mathcal{N}(\mu_{\text{i}}, \sigma_{\text{i}}^2),\ i=1,\ldots,n_{variables} \tag{5}$$

$$\beta_{ij} \sim t_{DF},\ \text{i}=1,\ldots,\text{n}_{\text{variables}},\ \text{j}=1,\ldots,\text{n}_{\text{Pa(Xi)}} \tag{6}$$

$$\beta_{0_i}^{non-root} \in \mathbb{Z} \cup \mathbb{R} \tag{7}$$

$$X_i^{\text{non-root}} = \beta_{0_i}^{non-root} + \sum_{j \in \text{Pa}(X_i)} \beta_{ij} X_j \tag{8}$$

$$\text{SNR}_{\text{dB}} = 10 \log\left(\frac{\text{var}(X_i)}{\text{var}(e_{\text{i}})}\right) \tag{9}$$

Table 1. Algorithm for data simulation.

Algorithm 1. Data Simulator

INPUT: Graph ($\mathcal{G}$); Distribution parameters for root nodes ($\mu_{\text{i}}, \sigma_{\text{i}}^2$); Degrees of freedom for Student's *t*-distribution (*DF*); Intercept for non-root nodes ($intercept_i^{non\text{-}root}$); Coefficients ($\beta_{ij}$) Topological order of $\mathcal{G}$ (TopOrder($\mathcal{G}$)); Signal-to-Noise Ratio in decibel (SNR_{dB}); Number of observations ($n_{samples}$)

For X_i in TopOrder($\mathcal{G}$) **do**

  If X_i in root nodes **do**

- Pick $n_{samples}$ from the normal distribution

  Else

- Use the parent values and the intercept to generate the sample values

End

For X_i in $n_{variables}$ **do**

- Calculate the variance of the noise
- Pick $n_{samples}$ from the noise distribution $\sim\mathcal{N}(0, \sqrt{var(e_{\text{i}})})$
- Add the measurement noise to the sample values

End

OUTPUT: The generated dataset, $\mathbf{X}^{n_{samples} \times n_{variables}}$

3.2 Types of faults simulated

In this simulator, two types of faults were implemented to introduce changes in the network, simulating common industrial upsets: process perturbation (or drifting) and sensor bias. These faults can result from electrical or mechanical incidents, aging components, varying material suppliers, inconsistent operations, or many other possible root causes. We classify the faults into two main groups: process faults, which induce real changes in the process out of the NOC regions, and sensor faults, which lead to biased measurements due to faulty sensors. The faults were modelled with detailed mathematical formulations, as shown in Table 2.

Table 2. Type of faults and their implementation in the data simulator.

Type of fault	Meaning	Modified factor
Process perturbation	A step change in a root node, representing, e.g., excessive variation in raw materials or operational disturbances	$\mu_i = \mu_{NOC,i} + k\sigma_i$
Sensor bias	Abnormal sensor readings caused by faulty sensors.	$X_i = X_i + k\sigma_{e_i}$

The parameter k represents the fault magnitude, controlling its severity. Process perturbations affect key variables and propagate through the network, while sensor bias introduces errors in measurements. These faults provide a practical testbed for evaluating the robustness of monitoring methods.

3.3 Summary of the selected networks

Three network configurations were selected to evaluate the monitoring methods, where these generated networks vary in size and complexity, and cover a representative range of industrial environments. Table 3 summarizes key characteristics of each network, such as the number of nodes, edges, root nodes, and other structural properties.

Table 3. Characterization of the three networks under study.

Network Information	Network 1	Network 2	Network 3
Number of nodes	50	100	1000
Number of edges	131	212	3880
Number of root nodes	23	48	333
Graph density	0.054	0.021	0.004
Number of open triplets	967	1624	56118
Number of closed triplets	94	62	570
Number of communities	5	8	27

4. RESULTS

In order to evaluate the performance of the monitoring methods, the three networks detailed in Table 3 were considered. For each network, a comparison was made between centralised and decentralised causal-based monitoring under two fault types: process perturbation and sensor bias. While examining the process perturbation fault, we focused on analysing the behaviour of each root node in the network, applying 10 different fault magnitudes. The range of magnitudes was fixed for all tests conducted on the various networks. To account for variability in the results, each magnitude was simulated with 100 replicates. The simulations were run with 1000 observations for each replicate, and the signal-to-noise ratio was set to 10 dB.

Similarly, in the case of sensor bias, each fault magnitude was applied to every variable in the network, and the detection sensitivity was assessed in each case.

Both methodologies were evaluated at a false alarm rate (FAR) of α = 0.01.

To assess fault detection performance, we calculated the True Positive Rate (TPR) for each replicate, defined as the proportion of correctly detected faults (true positives) relative to the total number of faulty samples. This resulted in 100 TPR values per magnitude. To summarize all the results from the perturbations of all individual variables, and compare the two monitoring methods, we calculated the median TPR for each magnitude across all variables, building a single curve of median TPR versus magnitude; in other words, this curve summarizes the sensitivity for single faults in all the network nodes. The area under this curve (median TPR vs magnitude), henceforth designated as AUC, was then calculated for each faulty variable, as it provides a comprehensive measure of performance across different fault magnitudes. The AUC values were normalized using a max-min approach, where the maximum AUC corresponded to perfect detection (TPR = 100% for all magnitudes) and the minimum was equivalent to the false positive rate (FPR) when no faults were introduced.

4.1 Faul detection results

The results obtained are shown in Fig. 2 (process perturbation fault) and Fig. 3 (sensor bias fault), which illustrate the distribution of the median AUC for all network variables over different network sizes and compare the two methods. Fault diagnosis was also conducted, with all faulty variables being conclusively identified. However, due to space constraints, these results are not presented.

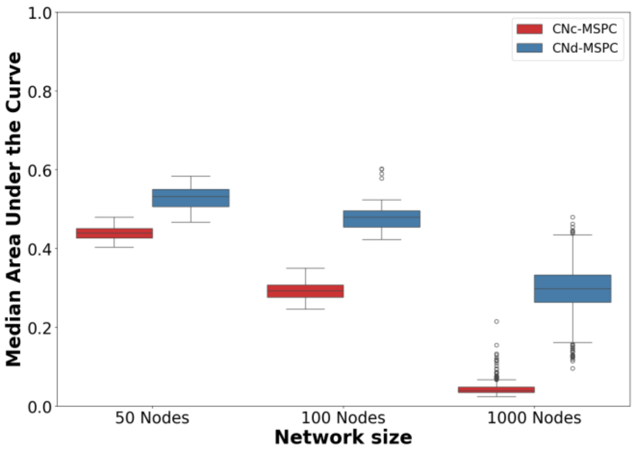

Fig. 2. The distribution of the median Area Under the Curve

(AUC) for CNc-MSPC and CNd-MSPC methods across varying network sizes in the process perturbation fault.

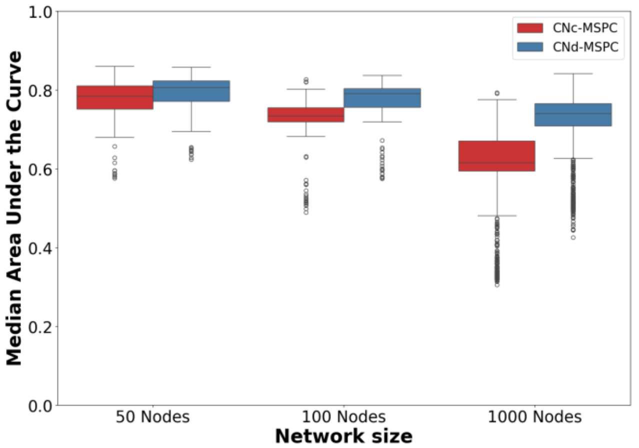

Fig. 3. The distribution of the median Area Under the Curve (AUC) for CNc-MSPC and CNd-MSPC methods across varying network sizes in the sensor bias fault.

4.2 Ablation test

One of the principal benefits of decentralised monitoring is its robustness in the event of information being lost due to abnormal situations or communication problems. To assess the influence of omitting information from multiple communities, we conducted an ablation test focusing on the 1000-node network, specifically analysing the variable X_5 belonging to community 4. A sensor bias fault was applied to assess the TPR across 10 different magnitudes, following the replication procedure outlined in the previous subsection. Initially, the test was performed using the complete dataset from all communities. Subsequently, we repeated the experiment while suppressing information from one to four communities that did not contain the faulty variable. The impact of these ablations was evaluated by comparing the TPR curves for each scenario (Fig. 4). The results indicate that the decentralised monitoring maintains robust sensitivity, with minimal loss observed across all tested cases, even when data from multiple communities were suppressed.

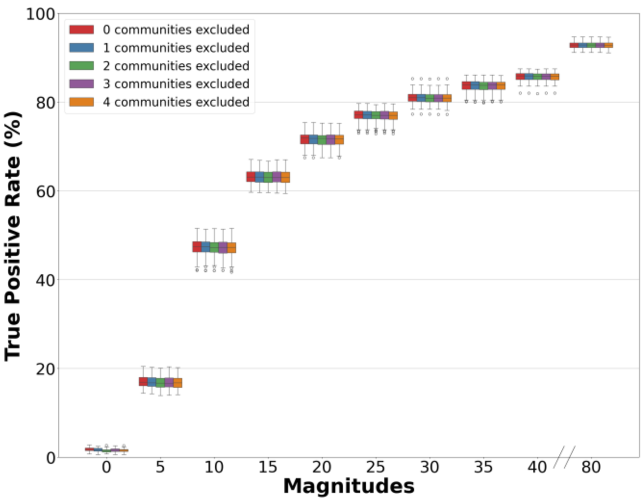

Fig. 4. The True Positive Rate (% TPR) as a function of fault magnitude for variable X_5 in a 1000-node network, evaluated under sensor bias fault. TPR values are based on 100 replicates, with results shown for scenarios excluding 0 to 4 communities' information.

5. DISCUSSION AND CONCLUSIONS

This study assesses the impact of increasing system dimensionality and complexity on fault detection sensitivity using a causal-based monitoring approach, which overcomes the limitations of traditional correlation-based methods, like PCA-MSPC, by explicitly modelling and incorporating cause-and-effect relationships. This is particularly advantageous in large industrial systems, enabling more accurate diagnosis.

The data were generated using a large-scale network static linear simulator, tailored for this study. Networks were configured as scale-free to mimic the characteristics of industrial processes, where certain nodes, known as hubs, exhibit a high degree of connectivity. Multiple replications for each fault scenario ensured the results' representativeness. Two fault types were simulated: process faults, by altering raw material setpoints, and sensor faults, introduced as localised bias errors. The propagation of process faults throughout the network contrasted with the localised nature of sensor faults, making direct comparison between these faults challenging due to their distinct propagation and detection characteristics. Nonetheless, process faults showed a steeper decline in detection sensitivity as the network size increased, whereas sensor fault detection degraded more gradually.

Fault detection sensitivity was assessed using the true positive rate (TPR) and the area under the curve (AUC), summarising each variable's performance across varying fault magnitudes. Results indicated that increasing the network size negatively impacted sensitivity, especially for process faults (Fig. 2 and Fig. 3).

A critical aspect of the study was the comparison between causal-based centralised and decentralised monitoring strategies. The Leiden community detection algorithm was employed to divide the network into communities, with monitoring then applied at the local level within each sub-network. The parameters of the Leiden algorithm were optimized based on the following criteria for each network: cohesiveness, separability, modularity, density, and conductance. Community detection leverages the natural structure of the network, ensuring that each sub-network remains cohesive and captures local dependencies, whereas traditional decentralized methods often disregard the topological structure. This more natural division improves fault detection sensitivity by isolating disturbances within more manageable communities.

The decentralised strategy proved particularly effective for large networks. For instance, in a network with 1,000 nodes subjected to process perturbations, the decentralised approach achieved AUC values comparable to those obtained using a centralised strategy in a smaller network of 100 nodes. This demonstrates how the community-based approach compensates for the sensitivity loss typically seen in centralised strategies as network size increases. By managing dimensionality through community-based statistics, the decentralised method improved fault detection while maintaining a false alarm rate near 1%. The reduced dimensionality not only enhances fault detection but also mitigates the computational burden associated with monitoring large-scale systems.

Moreover, ablation tests showed that excluding data from up to four communities did not affect the sensitivity of fault detection (Fig. 4). This underscores the robustness of the decentralised method, particularly in real-world industrial settings where data loss or communication failures are frequent challenges.

In conclusion, increasing the system dimensionality negatively impacts fault detection sensitivity. However, the decentralised approach effectively mitigates this issue. Future work will focus on refining the network division based on fault types and topological metrics, as well as comparing this approach with traditional decentralized monitoring methods in real-world scenarios, such as the Tennessee Eastman Process benchmark. By tailoring the monitoring strategy to the specific characteristics of each fault type, we aim to further enhance fault detection sensitivity in large-scale, complex networks.

ACKNOWLEDGEMENTS

Rodrigo Paredes acknowledges the support from the doctoral Grant (PRT/BD/154267/2022) financed by the Portuguese Foundation for Science and Technology (FCT), under MIT Portugal Program. Marco S. Reis acknowledges support from the framework of Agenda NEXUS: Innovation Pact Digital and Green Transition – Transports, Logistics and Mobility, Application nb. C645112083-00000059, Investment Project nb. 53, from the Incentive System "Mobilizing Agendas for Business Innovation", funded by the Recovery and Resilience Plan (PRR) and by European Funds NextGeneration EU. The authors also acknowledge support from FCT, I.P., through the projects with DOI:10.54499/UIDB/00102/2020 and DOI:10.54499/UIDP/00102/2020.

REFERENCES

Albert, R., & Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, *74*(1), 47–97. https://doi.org/10.1103/RevModPhys.74.47

Chiang, L. H., Russell, E. L., & Braatz, R. D. (2000). *Fault Detection and Diagnosis in Industrial Systems*. Springer Science & Business Media.

Ge, Z., & Song, Z. (2013). Distributed PCA Model for Plant-Wide Process Monitoring. *Industrial & Engineering Chemistry Research*, *52*(5), 1947–1957. https://doi.org/10.1021/ie301945s

Harenberg, S., Bello, G., Gjeltema, L., Ranshous, S., Harlalka, J., Seay, R., Padmanabhan, K., & Samatova, N. (2014). Community detection in large-scale networks: A survey and empirical evaluation. *WIREs Computational Statistics*, *6*(6), 426–439. https://doi.org/10.1002/wics.1319

Isermann, R. (2006). *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer Science & Business Media.

Javed, M. A., Younis, M. S., Latif, S., Qadir, J., & Baig, A. (2018). Community detection in networks. *Journal of Network and Computer Applications*, *108*(C), 87–111. https://doi.org/10.1016/j.jnca.2018.02.011

Joe Qin, S. (2003). Statistical process monitoring: Basics and beyond. *Journal of Chemometrics*, *17*(8–9), 480–502. https://doi.org/10.1002/cem.800

Kourti, T., & MacGregor, J. F. (1996). Multivariate SPC Methods for Process and Product Monitoring. *Journal of Quality Technology*. https://www.tandfonline.com/doi/abs/10.1080/00224065.1996.11979699

Paredes, R., Rato, T. J., & Reis, M. S. (2022). Causal Network Inference and Functional Decomposition for Decentralized Statistical Process Monitoring: Detection and Diagnosis. *Chemical Engineering Science*, 118338. https://doi.org/10.1016/j.ces.2022.118338

Pearl, J. (2000). *Causality: Models, Reasoning, and Inference: Models, Reasoning, and Interference* (2nd ed.). Cambridge University Press.

Peng, X., Ding, S. X., Du, W., Zhong, W., & Qian, F. (2020). Distributed process monitoring based on canonical correlation analysis with partly-connected topology. *Control Engineering Practice*, *101*, 104500. https://doi.org/10.1016/j.conengprac.2020.104500

Peters, J., Janzing, D., & Schölkopf, B. (2017). *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press. https://library.oapen.org/handle/20.500.12657/26040

Rato, T. J., & Reis, M. S. (2014). Sensitivity enhancing transformations for monitoring the process correlation structure. *Journal of Process Control*, *24*(6), 905–915. https://doi.org/10.1016/j.jprocont.2014.04.006

Reis, M., & Gins, G. (2017). Industrial Process Monitoring in the Big Data/Industry 4.0 Era: From Detection, to Diagnosis, to Prognosis. *Processes*, *5*(4), 35. https://doi.org/10.3390/pr5030035

Šidák, Z. (1967). Rectangular Confidence Regions for the Means of Multivariate Normal Distributions. *Journal of the American Statistical Association*, *62*(318), 626–633. https://doi.org/10.1080/01621459.1967.10482935

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. CRC Press.

Traag, V. A., Waltman, L., & van Eck, N. J. (2019). From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports*, *9*(1), 5233. https://doi.org/10.1038/s41598-019-41695-z

Van den Kerkhof, P., Vanlaer, J., Gins, G., & Van Impe, J. F. M. (2013). Analysis of smearing-out in contribution plot based fault isolation for Statistical Process Control. *Chemical Engineering Science*, *104*, 285–293. https://doi.org/10.1016/j.ces.2013.08.007

Yang, W.-T., Reis, M. S., Borodin, V., Juge, M., & Roussy, A. (2022). An interpretable unsupervised Bayesian network model for fault detection and diagnosis. *Control Engineering Practice*, *127*, 105304. https://doi.org/10.1016/j.conengprac.2022.105304