OPERATIONAL ROBUSTNESS ASSESSMENT OF DISTRIBUTED MONITORING SYSTEMS APPLIED TO FREEZE-DRYING EQUIPMENT

Sergey Kravchenko[1], Natalia Kravchenko[1], Nikolajs Kulesovs[1], Kristine Carjova[1, 2], Daria Panova[1]
[1]SIA "Cryogenic and vacuum systems", Latvia; [2]Tallinn University of Technology, Estonia
info@cvsys.eu,

Abstract. Distributed monitoring systems are increasingly used to support the operation and maintenance of agricultural processing equipment deployed across geographically dispersed locations. In such systems, reliable data acquisition and communication robustness are critical to ensure continuous monitoring under real-world operating conditions, including network disturbances and partial system failures in distributed environments. This paper presents an evaluation of the operational robustness of a distributed monitoring system designed for agricultural processing equipment. The study focuses on system-level behaviour under adverse conditions such as intermittent network connectivity, delayed data transmission, and temporary node unavailability. The monitoring framework is based on distributed embedded units and centralized data processing services, enabling remote acquisition of operational data without direct interference with primary control functions. The approach is consistent with distributed industrial monitoring, Industry 4.0 and distributed IoT architectures [1-4]. The evaluation was conducted using a test environment that emulates multiple equipment units operating under controlled disturbance scenarios. Freeze-drying equipment was used as a representative application case. Experimental results demonstrate the system's ability to maintain monitoring continuity, recover from communication disruptions, and preserve data consistency during abnormal operating conditions. The findings highlight key design considerations for robust distributed monitoring solutions and confirm their suitability for supporting reliable operation of agricultural processing equipment in distributed and resource-constrained environments.

Keywords: distributed monitoring, robustness, IoT, agricultural equipment, freeze-drying.

Introduction

The adoption of distributed monitoring systems in agricultural and food-processing industries has increased significantly with the development of Industry 4.0 technologies. These systems enable remote supervision of equipment, data-driven decision-making, and integration with cloud-based or distributed services [1-3].

In distributed environments, monitoring systems must operate reliably under non-ideal conditions, including communication disturbances, network latency, and partial system failures. Communication reliability and data availability therefore become critical factors influencing system performance [2-5].

Freeze-drying equipment represents a complex and demanding application case due to its sensitivity to process parameters and multi-stage operation, requiring stable monitoring conditions [10]. Ensuring reliable monitoring of such systems requires robust system design capable of maintaining functionality under adverse conditions [6-8].

Recent developments in pharmaceutical freeze-drying emphasize improved monitoring, process control, and robustness under varying operating conditions [9].

Unlike diagnostic-focused approaches, which concentrate on fault detection, this study focuses on operational robustness and system-level behaviour under disturbance conditions.

The main contributions of this paper are:

- evaluation of system robustness under communication disturbances;
- analysis of monitoring continuity in distributed environments;
- assessment of system behaviour during partial failures;
- identification of design principles for robust monitoring systems.

The presented study contributes to the understanding of robustness in distributed monitoring systems by providing experimental validation under controlled disturbance conditions.

Materials and methods

System architecture. The proposed monitoring system is based on a distributed architecture consisting of multiple embedded edge devices, communication infrastructure, and centralized data processing services.

DOI: 10.22616/ERDev.2026.25.TF081

Each equipment unit is equipped with a local embedded controller responsible for data acquisition and preliminary processing. Data are transmitted via network communication protocols to a centralized server, where aggregation, storage, and visualization are performed. Sensor data and system state variables are collected locally at each node and transmitted via the network infrastructure for monitoring and analysis.

The system is designed to ensure continuous data acquisition and communication resilience, even under partial network failures. The overall layered architecture is illustrated in Fig. 1.

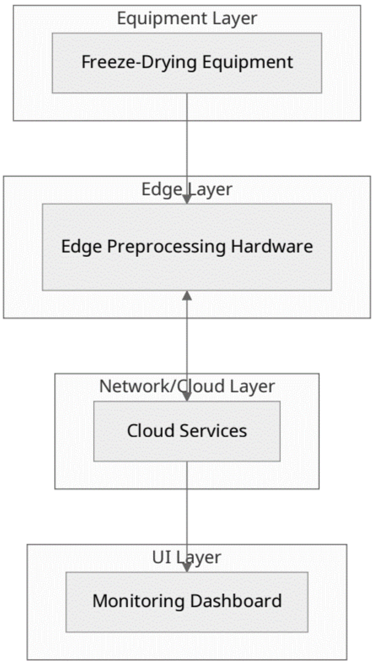

Fig. 1. **Layered architecture – Four-tier architecture of the distributed monitoring system**

The deployment concept supports both laboratory validation and real-world operation. Distributed architectures are widely used in industrial monitoring systems due to their scalability and resilience under communication variability [2; 4; 5]. The deployment model is shown in Fig. 2.

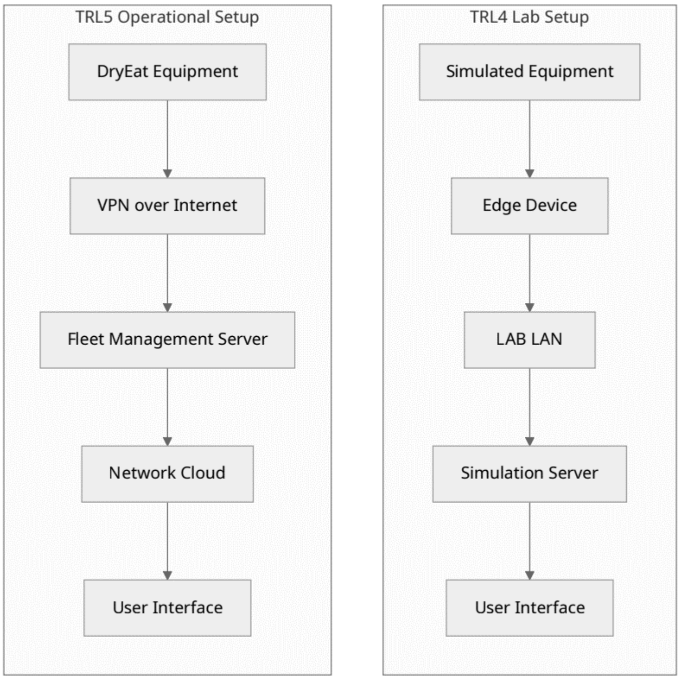

Fig. 2. **Deployment model: TRL4 laboratory setup and TRL5 operational environment**

Implementation details of the monitoring system. The distributed monitoring system consists of embedded nodes based on microcontroller platforms (including Arduino Mega 2560, STM-32 Nucleo-L053R8, Intel Celeron N4020, Raspberry Pi 3 and 4 class devices), representing typical implementations of AVR-, ARM-, and x86-based embedded architectures, interfaced with local sensors and industrial communication interfaces. Each node performs periodic data acquisition with a sampling interval in the range of 1–5 s, depending on the monitored parameter and process dynamics.

Acquired data are stored locally in a structured SQLite database and transmitted to a central server using a store-and-forward synchronization mechanism. Communication is implemented over standard TCP/IP protocols using HTTP and MQTT interfaces. In the event of communication loss, data buffering continues locally, ensuring no loss of information until connectivity is restored.

The central server is implemented in a Linux-based environment, providing REST API endpoints for data ingestion and system monitoring. Data visualization and analysis are performed via a web-based dashboard interface.

The experimental platform includes multiple distributed nodes connected via a managed Ethernet network, enabling controlled introduction of communication disturbances and reproducible evaluation of system behaviour under varying operational conditions.

Laboratory test environment. A laboratory-scale distributed test platform was developed to emulate multiple freeze-drying units operating simultaneously.

The platform consists of 6 independent simulator nodes, representing freeze-drying equipment units, interconnected via a TCP/IP network. The laboratory platform is shown in Fig. 3.

Fig. 3. **Distributed laboratory test platform with simulator nodes**

Each simulated unit represents an independent monitoring node connected via a network switch to a central server. The system allows controlled generation of disturbance scenarios, including communication interruptions and delayed data transmission.

The platform allows controlled emulation of network interruptions, delayed data transmission, node disconnections, and inconsistent data streams.

Such controlled testing environments are commonly used to evaluate robustness and reliability of industrial processes, including freeze-drying operations [6-8; 10].

Unlike diagnostic validation, the platform is used here to evaluate system robustness and communication behaviour.

Experimental methodology. The robustness evaluation focuses on system behaviour under the following disturbance conditions:

- intermittent network connectivity;
- communication latency;
- temporary node unavailability.

The following scenarios were implemented:

- Network interruption – temporary loss of communication between nodes and central system;
- Communication delay – introduction of latency in data transmission;
- Node unavailability – simulation of offline equipment units;
- Data inconsistency – incomplete or delayed data sequences.

Each scenario was executed independently across multiple nodes to evaluate system-level behaviour.

The monitoring workflow is shown below (Fig. 4).

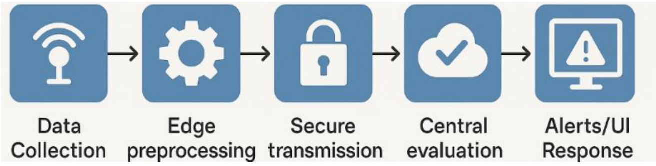

Fig. 4. **Monitoring and data processing workflow under disturbance conditions**

Performance was evaluated based on the following criteria:

- continuity of monitoring;
- data recovery capability after communication loss;
- system resilience;
- system response to disturbances;
- consistency of transmitted data.

These criteria correspond to key performance indicators for distributed monitoring systems in industrial environments [1; 3; 5].

Results

The field evaluation was conducted over a one-month observation period using a production fleet of seven freeze-drying machines with varying operational conditions. The fleet comprised four baseline nodes operating under stable network conditions, two connectivity-challenged nodes deployed at remote locations with LTE modems (where operators frequently disconnected the communication hardware), and one hardware-challenged node experiencing intermittent equipment faults that stress-tested the fault recovery mechanisms.

During the observation period, the fleet executed approximately 54 production runs. Table 1 summarizes the fleet composition and operational characteristics. The connectivity-challenged nodes experienced extended periods without network access, with the longest continuous connectivity gap lasting approximately one week due to deliberate modem removal by the equipment operator.

Despite these challenging conditions, the system achieved a 100% data recovery rate. Each production run was recorded locally as a self-contained archive comprising a SQLite database of sensor readings (sensors.sqlite) and a process log file (run.log). This architecture enabled complete post-mortem analysis even for nodes that remained offline for extended periods. When connectivity was restored, the buffered archives synchronized automatically with the central monitoring system [2; 4; 5]. The monitoring results directly informed iterative system development during the observation period. Based on field data analysis, five firmware releases and three host application releases were deployed to address identified issues and optimize system behaviour. This rapid iteration cycle was enabled by the comprehensive operational data captured by the monitoring infrastructure.

A significant finding was the system's capacity to detect hardware faults through continuous monitoring data analysis. Three distinct hardware issues were identified during the observation period:

1. a faulty vacuum sensor providing erratic readings;
2. broken shelf heater wiring causing inconsistent temperature distribution;
3. a reset button that was not properly wired in the control dialog, representing a software-hardware integration defect.

Fault identification methodology. Hardware faults were identified through analysis of temporal inconsistencies and cross-sensor correlations in the recorded operational data. The detection approach is based on comparison between expected process behaviour and observed system responses across multiple experimental cycles. For example, the faulty vacuum sensor was identified by non-physical pressure fluctuations that were inconsistent with expected process stage transitions. The heater wiring fault was detected through persistent temperature deviations between thermally coupled shelves operating under identical control inputs. The reset button malfunction was identified through missing or incomplete state transition sequences in event logs.

These anomalies were not detectable through isolated measurements but became evident through longitudinal log analysis and pattern repetition across multiple runs, confirming the effectiveness of log-based diagnostics for hardware-level fault detection. None of these faults had been detected through conventional maintenance procedures or operator observation. The monitoring system's ability to correlate sensor anomalies across multiple production runs enabled identification of these latent defects, demonstrating an unexpected diagnostic capability of the distributed monitoring approach [7; 10; 11].

The local recording architecture proved particularly valuable for nodes experiencing extended connectivity outages. The self-contained zip archives preserved complete process histories including high-resolution sensor data and timestamped event logs, enabling retrospective analysis equivalent to real-time monitoring [6; 11].

Table 1 presents the operational summary for the observation period, categorized by node type. The data confirm that architectural decisions prioritizing local data persistence over real-time transmission successfully addressed the connectivity challenges inherent in distributed agricultural processing environments.

Table 1

Fleet composition and operational summary

Node category	Count	Network conditions	Key characteristics
Baseline	4	Stable connectivity	Standard operation, continuous data transmission
Connectivity challenged	2	LTE modem, frequent disconnection	Remote locations, longest gap ~1 week
Hardware challenged	1	Stable connectivity	Intermittent equipment faults, stress-test recovery
Total	7	–	54 production runs, 100% data recovery

The hardware-challenged node served as an unintended stress test for the fault recovery mechanisms. Despite experiencing multiple hardware anomalies throughout the observation period, data integrity was maintained and all production runs were successfully captured for analysis. This resilience was attributed to the defensive programming practices and graceful degradation strategies implemented at the node level.

The connectivity-challenged nodes validated the store-and-forward synchronization mechanism. Even after week-long connectivity gaps, data synchronization completed successfully upon network restoration, with automatic conflict resolution preserving the temporal ordering of events. This behaviour confirmed that the system design effectively decouples data capture from data transmission.

The system behaviour across all node categories is illustrated in Fig. 5 and Fig. 6, which show representative monitoring interface screenshots demonstrating both normal operation and post-synchronization states following connectivity recovery.

Fig. 5. **Example of API response structure illustrating real-time acquisition of sensor data, including timestamped measurements and parameter grouping used for monitoring continuity and anomaly detection**

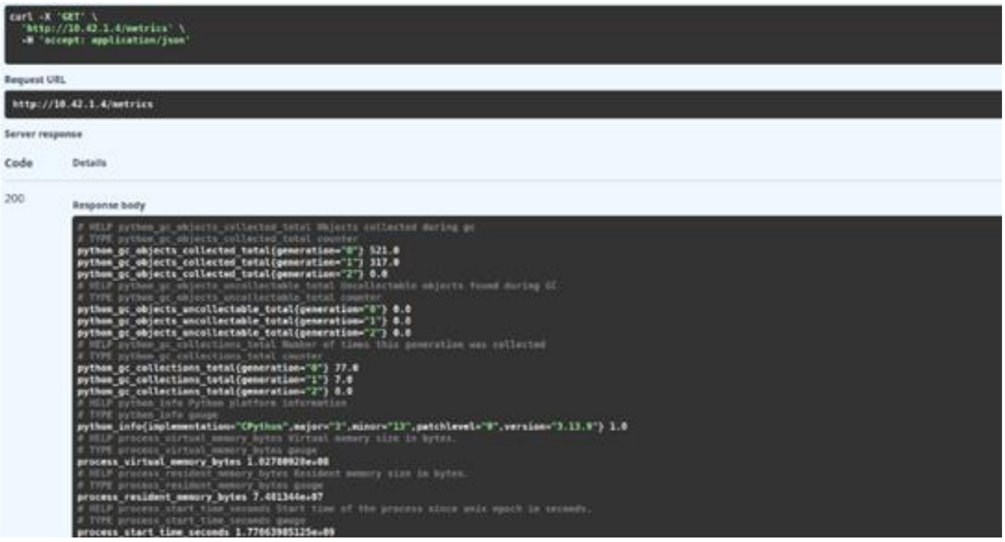

Fig. 6. **System metrics response showing aggregated process indicators, including temperature, pressure, and system status variables used for evaluating operational robustness and detecting abnormal system behaviour**

Discussion

The field evaluation yielded several findings that extend beyond the initial validation objectives. Most significantly, the monitoring system transitioned from a passive observation tool to an active enabler of iterative development, fundamentally changing the relationship between deployed systems and their ongoing improvement.

The 100% data recovery rate achieved despite connectivity gaps of up to one week validates the architectural decision to prioritize local data persistence over real-time transmission. This finding has significant implications for distributed monitoring in environments where network reliability cannot be guaranteed – a common characteristic of agricultural and food-processing facilities in rural areas [2; 5].

The discovery of three hardware faults through software monitoring represents an unexpected but valuable outcome. Traditional maintenance approaches, including visual inspection and periodic testing, had failed to identify these latent defects. The continuous monitoring data enabled pattern recognition across production runs that revealed anomalies invisible to point-in-time observations. This suggests that distributed monitoring systems may serve a diagnostic function beyond their primary role of process documentation [1; 3].

The iterative development cycle observed during the evaluation – five firmware releases and three host application releases within one month demonstrates how comprehensive field monitoring accelerates system maturation. Real operational data exposed edge cases and failure modes that laboratory testing had not anticipated, enabling rapid response to issues as they emerged in production environments.

Sensitivity to low-level anomalies. Detection of small or gradually developing anomalies represents a known limitation of rule-based log analysis approaches. In the present study, detection thresholds were implicitly defined through deviations from expected process patterns observed across repeated experimental cycles.

While this approach is effective for identifying significant deviations, detection of subtle anomalies requires more advanced analytical techniques. Future system development will incorporate statistical methods and machine learning-based anomaly detection to improve sensitivity and enable early identification of low-amplitude faults.

The contrast between laboratory simulation and real-world deployment merits particular attention. The connectivity-challenged nodes, with their week-long outages and operator-induced disconnections, presented conditions that would be difficult to replicate authentically in a controlled test environment. Similarly, the hardware-challenged node's intermittent faults provided organic stress testing of recovery mechanisms. These observations suggest that field evaluation with representative operational conditions is essential for validating distributed monitoring system robustness.

Several limitations constrain the generalizability of these findings. The sample size of seven machines, while diverse in operational characteristics, is modest. The one-month observation period, although sufficient to capture extended connectivity gaps and identify hardware faults, may not reveal seasonal or long-term degradation patterns. Additionally, all equipment was of the same type and manufacturer, which may mask compatibility issues that could arise in more heterogeneous deployments.

Future work should address these limitations through extended observation periods and larger, more diverse fleets. The demonstrated value of hardware fault detection through monitoring data suggests opportunities for developing automated anomaly detection algorithms. Integration with predictive maintenance frameworks may further enhance the practical utility of distributed monitoring systems in industrial freeze-drying applications [6; 9; 11].

System interoperability considerations. Although the experimental validation was conducted using equipment from a single manufacturer, the proposed monitoring architecture is based on modular design principles and standard communication protocols. This enables integration with heterogeneous equipment systems from different manufacturers, provided that access to sensor data or operational logs is available through standard interfaces.

The use of protocol abstraction and data normalization allows the system to operate independently of specific hardware implementations. Future work will focus on extending interoperability through unified data models and standardized communication layers to support large-scale deployment in mixed-equipment environments.

Conclusions

1. The distributed monitoring system demonstrated robust operation across 54 production runs on 7 machines over a 1-month observation period, including nodes with severely constrained connectivity (LTE, gaps up to 1 week).
2. The store-and-forward architecture achieved 100% data recovery rate, validating the design decision to prioritize local buffering over real-time transmission for distributed agricultural equipment.
3. Continuous monitoring enabled discovery of 3 hardware faults (faulty vacuum sensor, broken shelf heater wiring and reset button wiring) that would have been difficult to diagnose without structured logging.
4. Monitoring insights directly drove iterative product improvement: 5 firmware and 3 host application releases during the observation period.
5. The results confirm that operational robustness in distributed monitoring can be achieved through architectural simplicity (local ZIP archives, post-reconnect synchronisation) rather than complex real-time protocols.

The results demonstrate that robust distributed monitoring can be achieved through appropriate architectural design and data-handling strategies without increasing system complexity. The proposed approach provides a practical and scalable solution for improving reliability and maintenance support in distributed agricultural processing systems.

Acknowledgements

The authors are grateful for the support of the Latvian Food Competence Centre, The Central Finance and Contracting Agency of Latvia, and the European Regional Development Fund within the framework of project Nr. 2.2.1.3.i.0/1/24/A/CFLA/002 Research no. DP-13 “Development and establishment of an automated service system for the sublimation equipment park (acronym: ČetriS)”.

Author contributions

Conceptualization, S. Kravčenko, N. Kravčenko and N. Kuļešhovs; methodology, S. Kravčenko, N. Kuļešhovs and K Carjova; software, validation, S. Kravčenko and N. Kuļešovs; formal analysis, N. Kravčenko, N. Kuļešovs, K Carjova, S. Kravčenko and D. Panova; investigation, S. Kravčenko, N. Kuļešovs, N. Kravčenko, D. Panova and K Carjova; data curation, N. Kuļešovs, N. Kravčenko, K

Carjova and D. Panova; writing - original draft preparation, S. Kravčenko; writing - review and editing, S. Kravčenko, N. Kravčenko and N. Kuļešovs; visualization, N. Kravčenko, K Carjova and D. Panova.; project administration, N. Kravčenko.; funding acquisition, N. Kravčenko. All authors have read and agreed to the published version of the manuscript.

References

[1] Lee J., Bagheri B., Kao H.A. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. Manufacturing Letters, vol. 3, 2015, pp. 18-23.
[2] Xu L.D., He W., Li S. Internet of Things in industries: A survey. IEEE Transactions on Industrial Informatics, vol. 10(4), 2014, pp. 2233-2243.
[3] Tao F., Qi Q., Liu A., Kusiak A. Data-driven smart manufacturing. Journal of Manufacturing Systems, vol. 48, 2018, pp. 157-169.
[4] Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, vol. 29(7), 2013, pp.1645-1660.
[5] Borgia E. The Internet of Things vision: Key features, applications and open issues. Computer Communications, vol. 54, 2014, pp. 1-31.
[6] Juckers A., Knerr P., Harms F., Strube J. Emerging PAT for Freeze-Drying Processes for Advanced Process Control. Processes, vol. 10, 2022, pp. 2059.
[7] Vanbillemont B., Greiner A.-L., Ehrl V., Menzen T., Friess W., Hawe A. A model-based optimization strategy to achieve fast and robust freeze-drying cycles. International Journal of Pharmaceutics: X, vol. 5, 2023, pp. 100180.
[8] Bosca S., Fissore D., Demichela M. Reliability Assessment in a Freeze-Drying Process. Industrial & Engineering Chemistry Research, vol. 56(23), 2017, pp. 6685-6694.
[9] Rhieu S.Y., Korang-Yeboah M., Anderson D. D., Arigo J., O'Connor T., Shah R. Recent trends in pharmaceutical freeze-drying and process monitoring. AAPS Open, vol. 11, 2025, article number 27, pp. 1-11.
[10] Nail S.L. et al. Recommended Best Practices for Process Monitoring Instrumentation in Pharmaceutical Freeze Drying. AAPS PharmSciTech, vol. 18, 2017, pp. 2379-2393.
[11] Massei A., Falco N., Fissore D. NIR-Based Real-Time Monitoring of Freeze-Drying Processes: Application to Fault and Endpoint Detection. Processes, vol. 13, 2025, pp. 452.