Journal of Industrial Information Integration 41 (2024) 100667

Contents lists available at ScienceDirect

Journal of Industrial Information Integration

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

Review article

Design of supervision solutions for industrial equipment: Schemes, tools and guidelines for the user

Mirko Mazzoleni

Università degli Studi di Bergamo, Department of Management, Information and Production Engineering, via Marconi 5, 24044, Dalmine (BG), Italy

ARTICLE INFO

Keywords:
Supervision
Condition-based maintenance

ABSTRACT

The advent of Industry 5.0 envisages production systems that are more resilient, embrace human–machine collaboration and promote sustainability driven by technological research. The development of supervision solutions for industrial equipment fills in this picture as a basis for more proactive Condition-Based Maintenance strategies. The goal of this paper is to provide a self-contained set of guidelines to design such supervision solutions. With respect to existing literature on the topic, we provide a design process with a strong focus on experimental data collection and failure reproduction activities. Moreover, the connections between the steps of the proposed process are clearly highlighted to guide the user. First, the paper provides a set of tools to select the critical items and the methodological approaches for supervision. Then, these tools are used and referenced in the proposed design process. Finally, the proposed process is exemplified on two industrial case studies to show its effectiveness. Considerations, hints, and a user guidelines are given at the end of most sections.

1. Introduction

Unplanned downtimes of industrial equipment cause huge production losses and high maintenance costs in manufacturing industries [1,2]. *Supervision solutions* are algorithms that target the diagnosis, monitoring or prognostics of specific failure modes of the equipment items, relying both on collected event data and continuous physical measurements from the equipment [2–4]. Based on the *supervision outputs* of such technical solutions, Condition-Based Maintenance (CBM) actions can be performed [5, Chapter 9.3], eventually supported by Prognostics and Health Management (PHM) tools [6]. The CBM/PHM combination is considered by several authors to be the basis for the development of more proactive e-maintenance strategies [3,7], able to deal with the increase in complexity and uncertainty of industrial systems making them resilient and engineering immune [6,8].

Manufacturers of industrial equipment are nowadays leveraging the opportunities offered by supervision solutions as technical pillar for CBM/PHM. This was somewhat predicted: in a survey conducted by the IFAC industry committee to their members in 2018 to determine the current and future impact of several control technologies, intelligent control and fault diagnosis placed in the two top positions, with an increment of +30% from "high current impact" to "high future impact" responses [9].

In addition to the fundamental economic, managerial and operational aspects involved in the development of a CBM/PHM strategy [3,6,10], the design and development of an algorithmic supervision solution is ultimately a technical problem. Still, each solution strongly depends on the equipment configuration and the nature of the specific failure modes [11]. While some literature envisage "general" methodological approaches for supervision, the authors of this work are firmly convinced that considering the specificities of each equipment is essential for a practical utilization of such solutions. Thus, diversity of the appliance and complexity of the industrial environment demand *a systematic and organized process* for the design of supervision solutions, a necessity already advocated by several authors [2,6,12,13].

In this context, the authors in [11] provide a recent review of proposed processes to design supervision solutions (i.e. data-processing algorithms) and overall implementing CBM/PHM strategies (i.e. supervision algorithms and maintenance decisions) [1,6,10,12,14,15]. Considering such contributions, the most relevant steps to address are here summarized within five macro-steps:

1. perform a cost–benefit analysis on employing a CBM/PHM strategy;
2. select the most critical items to be monitored, their failure modes and related parameters to be measured from sensors. Identify the possible causes and symptoms of the selected failure modes;
3. select the methodological approaches (i.e. algorithms, signal processing and data mining techniques) to address the supervision aims of diagnosis, monitoring and prognostics;

E-mail address: mirko.mazzoleni@unibg.it.

https://doi.org/10.1016/j.jii.2024.100667
Received 10 July 2023; Received in revised form 5 July 2024; Accepted 25 July 2024
Available online 31 July 2024

4. define methods and tools to support maintenance decision-making;
5. perform reviews on the choices made on the previous steps.

Among the aforementioned contributions, [1,14] specifically target the manufacturing industry. Tools and practical suggestions are provided for the above macro-step 2, where the use of a Reliability Centered Maintenance (RCM) rationale and tools like Failure Modes, Effects, and Criticality Analysis (FMECA) is suggested. Macro-steps 1, 4 and 5 are also briefly considered, and case studies are provided on industrial robots and bearing diagnosis, respectively. However, no overall guidelines are provided on macro-step 3. The work [15] also advocates RCM as item criticality selection tools in macro-step 2, and suggest a strategy based on features and decision fusion for supervision. While two case studies are reported on an electrical motor for elevator systems and a low methane compressor used in petrochemical plants, the proposed process remains too abstract for rapid practical usage. The authors of [10] describe a CBM development process considering the macro-steps 1, 2, 4 and 5, presenting three case studies on a power plant, a manufacturing and a paper mill company. However, since the proposed process is broad, few details are provided on how to implement each step of the process, and since the focus is on vibration analysis, little consideration is given to macro-step 3. The work [12] starts with listing the benefits of CBM/PHM in industry, and then presents a set of relevant aspects to be considered when implementing a PHM solution, also indicating the use of appropriate graphical tools for presenting the supervision results. The work lightly touches all five macro-steps in common to the reviewed development processes, and again RCM and FMECA are suggested tools for criticality analysis of the equipment items. Nonetheless, little guidelines are provided on the other steps of the process, and no experimental case studies are discussed. Authors in [6] focus on the prognostics of rotary machines. Regarding the macro-step 1, the work suggests the use of a four-quadrant chart to identify critical items, representing items fault frequency vs. the average downtime caused by those items. The macro-step 3 is tackled by listing common measurements and algorithms for the supervision of rotary items, with advantages and disadvantages. Also, Quality Function Deployment [16], commonly used in product design, is suggested as a tool to select between different methodological approaches. Graphical tools denoted as Degradation Chart, Performance Radar Chart, Problem Map and Risk Radar Chart are proposed to support decisions at macro-step 4. Three case studies are presented, related to an alternator, a chiller and a spindle bearing. While review [11] considers [6] to be the most relevant contribution as a development support for CBM/PHM, it also points out its specific focus on rotary items. Lastly, the standard ISO 13372 [17] establishes a comprehensive procedure in eight steps for implementing a condition monitoring program, that covers all five steps listed in the previous paragraph with some practical considerations. The standard provides, for ten types of machines, examples of failures and associated parameters to be monitored. ISO 13379-1 [18] suggests RCM and a modified FMECA sheet to list the symptoms associated to each failure mode. However, these standards are conceived for a "traditional" view of CBM (i.e. relying on vibration, thermographic, acoustic or ultrasound measurements) and should be adapted to consider more recent PHM tools [6,12]; along this line, no recommendations related to macro-step 3 and macro-step 4 are present. The work [19], not present in [11] since it is more recent, proposed a design process that is based on a structured definition of a set of requirements for such solutions, and a set of symptoms and diagnostic rules. After the definition of such elements, the author employs a Bayesian Network to select the most suitable set of diagnostic rules that satisfy the requirements. Although interesting for its schematic approach, the work does not provide enough details on the implementation of the process, and assumes that a set of diagnostic rules is present prior to their selection.

It is worth mentioning the work [20], that suggests how to represent the components of a supervision solution once is has been developed, and thus applies after its design process.

Considering the reviewed papers, [11] concludes that "more guidelines and tools to facilitate the application and enhance the effectiveness of each step" are necessary. Indeed, each CBM/PHM development process has its own strengths and weaknesses, since more importance is given to different aspects of the process. One of such aspects is related to *how to perform experiments for data collection*, so that such data will guide the development of the supervision solution. This important point is not considered in any of the reviewed processes.

Based on these premises, the main purpose of this paper is to provide an additional process for the design of supervision solutions for industrial equipment, pointing to the most recent advancements related to PHM and supervision tools and algorithms. Indeed, as far as the authors are aware, the most recent contributions to CBM/PHM development processes dates back to 2014 [6,12,14]. The process proposed in this paper comprises eleven steps covering the previously mentioned macro-steps 2 to 4, where emphasis is given to problem definition and to data collection, a phase neglected or poorly covered until now in existing processes and international standards. A comparison of the steps of the proposed process with respect to the reviewed ones is reported in Table A.12. The proposed process covers only the design steps, that is the actions necessary to obtain a prototype of the algorithmic solution. It will not cover the integration of such solution into the existing hardware/software of the equipment. Likewise, this paper does not cover methods for maintaining the functionality of the equipment in case of faults [21], or recent advancements in control theory integrating prognostics information into the control laws of machinery [22]. The point of view of the paper is from a control engineering perspective: however, we argue that related communities as reliability engineering [23] and statistical process monitoring engineering [24] may find value in the paper, although the terminology is slightly different between those communities on the control one.

Contributions. Since not all industrial companies have a clear understanding of what entails to develop a supervision solution, with this work we want to provide a bridge between theory and practice in the design of such solutions, thus increasing the awareness of both industries and newcomers to the field. For this reason, throughout the paper user guidelines provide practical tips for the implementation of the proposed process, and specific sections are devoted to cover important tools and concepts to support those guidelines.

The contribution of this paper with respect to the existing literature is to:

- *provide, and experimentally validate, a practical process to design supervision solutions algorithms for industrial equipment*. With respect to existing literature, the proposed process gives more focus to the design of experiments (data collection and failure reproduction), not limiting to list the steps of the process, but also emphasizing the relations between those steps.

Outline. The paper is organized so that, after a contextualization in Section 2, a discussion about technical tools and concepts for the design of supervision solutions is provided in Sections 3 and 4. Then, each step of the proposed design process is presented in Section 5. Each step is related to one or more tools or concepts described in the previous technical sections. Whenever a tool or concept is necessary for the implementation of a step, links to the related sections are provided. Technical readers can start by directly focusing on Section 5 and go back to technical sections when necessary. The structure of the paper is reported in Fig. 1.

The remainder of the paper is as follows. Section 2 briefly presents an introduction to the topic of the supervision of industrial equipment. Next, Section 3 reviews the Failure Modes and Effects Analysis (FMECA) procedure and introduces a modified Supervision-oriented FMECA worksheet to support the design of supervision solutions. This

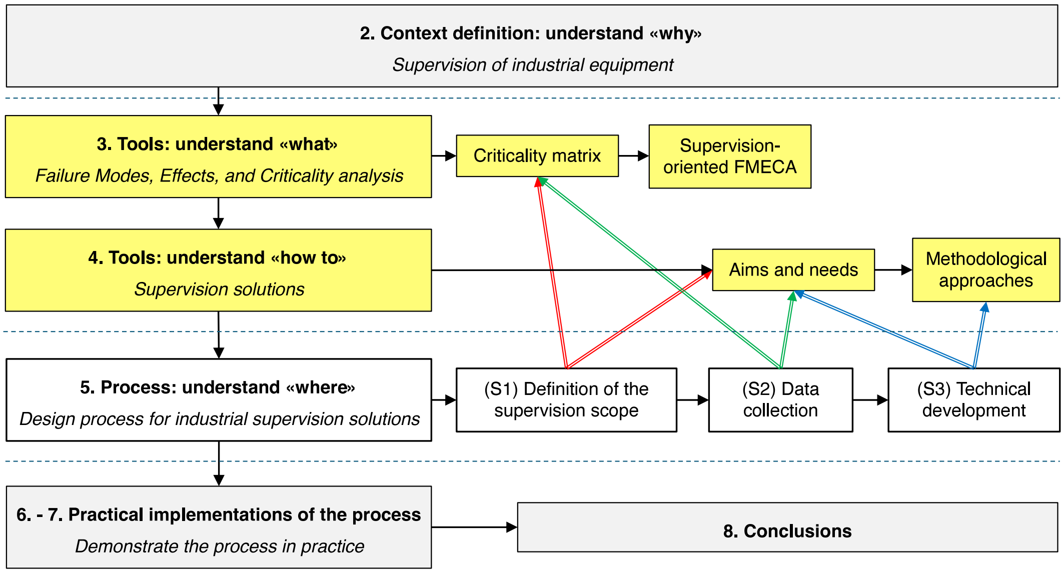

Fig. 1. Structure of the paper. The paper first defines in Section 2 the application context and the "why" of supervision solutions of industrial equipment. Then, Sections 3 and 4 (in yellow) are devoted to introduce the necessary tools to select "what" items are the focus of the analysis (using a proposed Supervision-oriented FMECA) and "how to" perform such analysis (pointing to a clear definition of the aims and methodological approaches). Section 5 aims to provide a "compass" for orientation and understand "where" the engineer is in the design of the supervision solution. This process relies on the tools defined previously in Sections 3 and 4. Finally, practical examples implementing the proposed process and conclusions are provided.

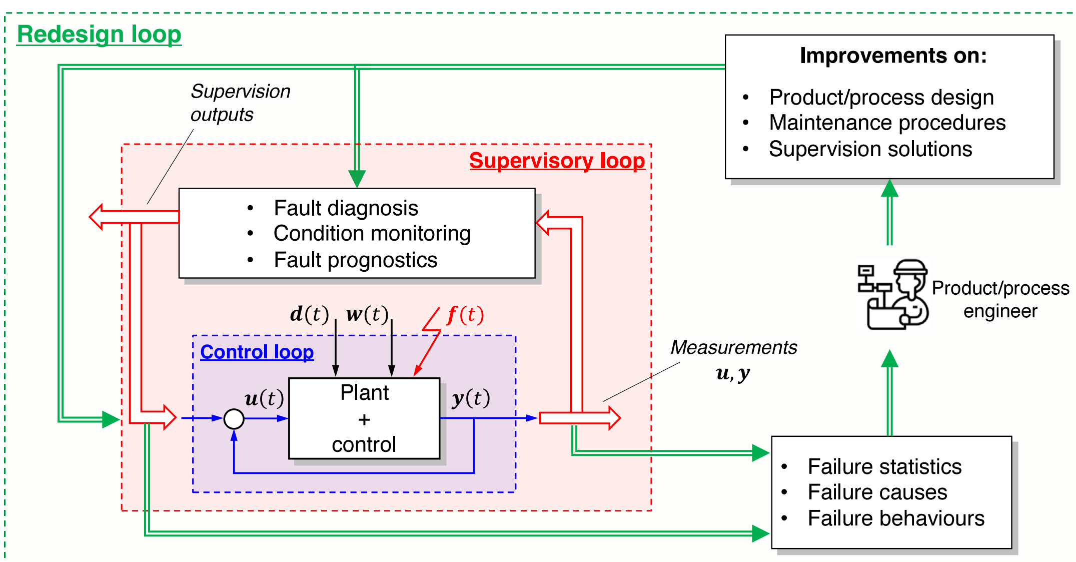

Fig. 2. Redesign loop schematic. Plant measurements and supervision outputs are integrated with failures behaviors, failures statistics and their causes to improve the product or process, maintenance procedures and tune supervision solutions.

is a suggested tool to select the critical components, their failure modes and symptoms. Section 4 reviews the main types of supervision solutions, guiding the selection of the most appropriate methodological approach. Thus, considering the five macro-steps common to all the reviewed processes, we cover the macro-steps 1, 2, and 3. Section 5 introduces the proposed process for designing a supervision solution targeted to industrial equipment. This is the core contribution of the paper. Sections 6 and 7 evaluate the proposed process on experimental settings. Section 8 concludes the paper.

2. Supervision of industrial equipment

Fig. 2 shows a general feedback scheme where supervision functions are integrated into an industrial equipment (a physical plant with closed-loop control instrumentation). The plant supervision can be performed by measuring its inputs $\boldsymbol{u}(t) \in \mathbb{R}^{m_u \times 1}$ and outputs $\boldsymbol{y}(t) \in \mathbb{R}^{p \times 1}$ (and also additional physical quantities if necessary, like vibrations or superficial temperature). The aim of supervision is to assess the presence of faults $\boldsymbol{f}(t) \in \mathbb{R}^{m_f \times 1}$ notwithstanding the presence of unmeasurable disturbances $\boldsymbol{d}(t) \in \mathbb{R}^{m_d \times 1}$ and noises $\boldsymbol{w}(t) \in \mathbb{R}^{m_w \times 1}$ that influence $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$. Here, t denotes a continuous time instant.

The most basic supervision strategy is *limit checking*, that consists in evaluating if the values of inputs and outputs signals are outside tolerance intervals. Considering the outputs, this mean to check the conditions $\boldsymbol{y}_{\min} < \boldsymbol{y}(t) < \boldsymbol{y}_{\max}$. Similarly, *trend checking* considers the first derivative of these signals, so that the check becomes $\dot{\boldsymbol{y}}_{\min} < \dot{\boldsymbol{y}}(t) < \dot{\boldsymbol{y}}_{\max}$, with the advantage of a possible earlier detection [25, Chapter 7]. Another rationale is given by *plausibility checks*, as a way to combine

Table 1
Severity levels of a failure mode, for a qualitative criticality analysis. The levels express a classification of a failure mode based on a judgmental evaluation of its undesired effects on the equipment and humans.

Severity level (s_L)	Failure mode effect	Interpretation
1	No effect	No effect on equipment/no injury on humans
2	Minor	Equipment partly down or minor failure/slight injury on humans
3	Severe	Items down or failure/injury on humans
4	Major	Equipment shut down or serious damage/serious injury on humans
5	Catastrophic	Equipment damage or destruction/death on humans

limit-checking of more measurements through simple rules in the form of IF–THEN statements describing the plant behaviors [26].

Limit checking and plausibility checks are fine if the plant works in a steady-state operating regime, does not present large signal transients, and faults are directly visible from raw signals data. When such conditions do not hold, supervision algorithms for fault diagnosis, condition monitoring and fault prognostics are needed to attain adequate detection and low false positive performance [25].

Detected faults must be immediately displayed to the operator and used for taking stopping or maintenance actions on the equipment.

In addition to the control and supervisory loops, a supervision solution supports the definition of a *redesign loop*, see again Fig. 2, where information about failures statistics, causes, and effects can be leveraged by product/process engineers to improve product/process design and quality, optimize maintenance procedures and refine the supervision solutions [12]. In this way, for the equipment producer, the supervision solutions become not only an “optional” to be sold along with the equipment, but also a strategic asset that can be improved the more the solution is used.

The next section shows how the FMECA procedure can be used to prioritize the items that benefit from a supervision solution, proposing also a *Supervision-oriented FMECA worksheet* to aid their design process.

2.1. User guidelines

- » *Measurements*: when considering dynamical systems, in addition to the plant outputs, measure also the plant inputs (control actions and reference signals if the system runs in a closed-loop configuration). When considering passive mechanical components (e.g. bearings, gears) carefully consider the best sensor location (also trying different installations) to monitor their behavior. This entails to understand the physical phenomenon that needs to be captured and how the sensor works.
- » *Limit and trend checking*: checking the tolerances of raw signals, their first derivative, or a combination of both by plausibility checks against fixed or adaptive thresholds might be enough to detect most faults. Adaptive thresholds depend on the magnitude of the input signals, signals trends [25, Chapter 7] and modeling uncertainties in a model-based approach [27, Chapter 2].
- » *Wear-in period*: the parameters of a tolerances check or diagnostic method should be refined after an initial wear-in (also known as burn-in) working period of the item under monitoring, to stabilize the behavior of its mechanical components. Periodic retuning of warning and alarm thresholds may be necessary to adapt the thresholds to changes in the equipment not related to degradation of its functions.

3. Failure modes, effects, and criticality analysis

The supervision solution should tackle (every or a subset of) the failures, that, after a design and testing phase, remain unavoidable and are dangerous from the reliability and safety point of view [28, Chapter 4.6]. The identification of those failures can be performed by a FMECA procedure defined in international standards [29,30] and books [5, Chapter 4.2] [23, Chapter 2.6]. The *criticality* term denotes a measure of the severity of a failure mode and its frequency of occurrence.

Table 2
Frequency of occurrence levels of a failure mode, for a qualitative criticality analysis. The levels express a classification of a failure mode based on a judgmental evaluation of its frequency of occurrence.

Frequency level (o_L)	Failure mode frequency	Interpretation
1	Very unlikely	$\geq$ 1 every 1000 years
2	Remote	1 every 100 years
3	Occasional	1 every 10 years
4	Probable	1 every 1 year
5	Frequent	$\leq$ 1 every 1 month

The FMECA procedure can be performed *qualitatively* or *quantitatively* [25, Chapter 4][5,30]. In this paper we propose a criticality table based on qualitative FMECA as a tool for prioritizing the items that should be targeted by the supervision solution. Specifically, the criticality analysis produces a measure of risk of the effect that a failure mode has on the successful operation and safety of the equipment. This use of FMECA for supervision solutions is supported by a proposed Supervision-oriented FMECA worksheet.

3.1. Qualitative criticality analysis

A qualitative criticality analysis consists into defining, for each failure mode, qualitative levels of severity and frequency of occurrence. Severity levels can be defined as in Table 1, ranging from 1 to 5 and denoting the no effects, minor, severe, major or catastrophic effects, respectively.[1]

The same rationale is applied to the occurrence frequency of a failure mode, ranging from 1 to 5 and denoting very unlikely, remote, occasional, probable and frequent failure modes, respectively, see Table 2.

A qualitative criticality analysis requires less effort than a quantitative one, and can be used when numerical failure rates of the items are not available. Failure rates for a quantitative analysis can be deduced from in-house data (experimental or judgmental) or from several data sources, as [31,32], [25, Chapter 3], [33, Appendixes 4–5], [5, Chapter 16].

3.2. Criticality matrix

The criticality matrix provides a graphical interpretation for classifying failure modes into several risk categories, based on their severity and frequency of occurrence. In this paper, it will be used for the selection and prioritization of the items that have to be monitored by a supervision solution, based on their most critical failure modes.

Fig. 3 presents a criticality matrix example. The abscissa axis can be a discrete frequency level (in case of a qualitative analysis) as in Table 2 or the continuous criticality number as suggested in [30] in case of a quantitative analysis. Fig. 3 defines heuristically three risk categories. The least risky failure modes, denoted by green cells, do not require any corrective design action (for instance, replacing the

[1] The reader should be aware that this is just a possible categorization: the severity categories and their meaning should be defined such that they are relevant for the specific practical application considered.

Table 3
Supervision-oriented FMECA worksheet aimed at the selection of failure modes, and relative items, that should be considered as the focus of a supervision solution algorithm. Each row of the table is a failure mode. Additional columns can be added, by reporting the name, function of the item or sub-item, notes or reliability information such as the Mean Time Between Failures of the item, or possible methods for supervising the failure mode.

Failure description					Failure effect		Failure criticality		Failure reproduction	
Mode	Causes	Detection method	Affected variables	Symptoms	On item	On equipment	Severity (1–5)	Freq.cy (1–5)	Fault injection	Fault degradation
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

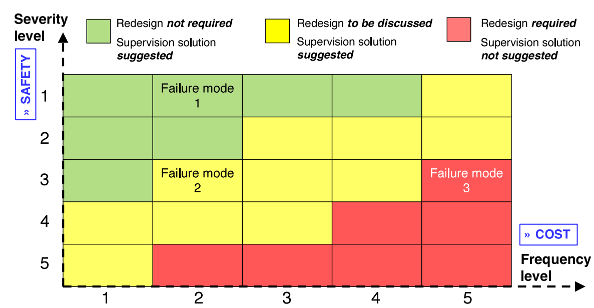

Fig. 3. Criticality matrix, representing a categorization of the equipment failure modes based on their qualitative severity level and their qualitative or quantitative frequency of occurrence. Given the classification of each failure mode in the matrix cells, proper actions can be prioritized.

item with one of higher quality or provide a physical redundancy) and can be tackled by supervision algorithms. The most risky failure modes, colored in red, require important corrective actions that no algorithm can provide. In-between failure modes, denoted in yellow, should be discussed singularly. However, the consideration of which failure modes to tackle should depend on whether safety or cost is the driving factor for the analysis. The proposed tool for selecting critical item has a similar rationale to the four-quadrant char provided in [6] and referenced in the introduction of this paper.

3.3. Supervision-oriented FMECA worksheet

Several examples of FMECA tables have been reported in specialized literature and standards [5,23,25,30,34], with various levels of complexity and detail. Table 3 proposes a qualitative Supervision-oriented FMECA
(SoFMECA) worksheet, with columns tailored for a FMECA oriented to the design of a supervision solution. A similar worksheet is proposed in ISO13379-1 with focus on symptoms [18]: however, the proposed one additionally focuses on failure reproduction procedures for data acquisition. The worksheet is structured as follows. Each row of the table is a failure mode, relative to a certain item. The columns are grouped into:

1. FAILURE DESCRIPTION: these columns provide a description of the failure. The *Mode* column reports the failure mode, that should be specified as a nonfulfillment of the functional requirements for that item.[2] The *Causes* column lists the circumstances that may produce or contribute to the failure mode. Column *Detection method* contains the various possibilities for the detection of the failure mode. These may involve diagnostic testing, alarms, proof testing, or human perception. Column *Affected variables* lists all the physical quantities (measurable or not) that might be influenced by the failure mode. Column *Symptoms*[3] describes the observable symptoms produced by a failure mode.
2. FAILURE EFFECT: devoted to describe the failure effects on the item (*On item*) and on the overall equipment (*On equipment*), respectively.
3. FAILURE CRITICALITY: devoted to qualitatively ascertain the criticality of the failure mode. *Severity (1–5)* and *Freq.cy (1–5)* columns refer to Table 2 and Table 1, respectively.
4. FAILURE REPRODUCTION: devoted to the description of various ways to introduce a state of fault in the item (that can lead to a failure mode), by directly replacing the healthy item with a failed one (*fault injection*) or by carrying out actions that progressively lead a healthy item to become failed (*fault degradation*). These actions are mandatory for acquiring experimental data in failed or degraded situation.

After the selection of relevant items, the aims, needs and the type of approach of the supervision solution must be chosen.

3.4. User guidelines

» *Criticality analysis*: start with a qualitative criticality analysis following Tables 1 and 2, tayloring meanings of the levels to the application. If data and time are available, refine the results with the quantitative approach.
» *Failure modes assessment*: perform a FMECA using the SoFMECA worksheet, especially considering the affected physical variables and failure reproduction possibilities.
» *Failure modes reproduction*: assess if a failure mode can be reproduced on the machine by fault injection or fault degradation, and the degree of approximation of the introduced fault or failure mode with respect to the nominal one. Note that some failure modes only allow one simulation modality, and some cannot be realistically reproduced at all.
» *Prioritization of items*: build a criticality matrix to categorize and prioritize the failure modes (and thus the related items) that will benefit the most from a supervision solution.
» *Experimental effort estimation*: based on chosen items and possibility to reproduce the failure modes, estimate how much experimental time is required to collect data in the failed condition, so to reserve machine and human personnel availability for experimental tests. If the machine is scarcely available, consider building an ad-hoc test bench for the items to be tested.
» *Define the circumstances of the symptoms*: establish the operating regimes where the symptoms appear, for each fault or failure mode. External factors and other conditions should be annotated.

[2] Sometimes the distinction between a failure mode and a fault is blurry. In the FMECA and criticality table, it may be of interest to consider not only the failure modes, but also some faults (which do not compromise the item functionality).

[3] Perception, made by means of human observations and measurements (descriptors), which may indicate the presence of one or more fault. [17]. Symptoms can be expressed by defining: (i) the time constant of the evolution of the descriptor; (ii) the type of evolution of the magnitude change; (iii) the descriptor used; (iv) the location where the symptom is observable on the equipment; (v) the circumstance. *Examples: Slow and regular increase of the magnitude of the first harmonic of vibration acceleration; Bearing temperature is 10°C above usual value in nominal condition.*

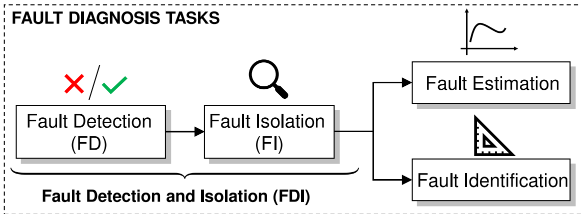

Fig. 4. Fault diagnosis includes fault detection, isolation, estimation and identification. Sometimes, the terms fault estimation and fault identification are used as synonyms, where one includes also the aims of the other.

4. Supervision solutions

In the specific view of the authors, the terms fault diagnosis, condition monitoring and fault prognostics denote three different supervision aims and needs. We distinguish such supervision aims based on:

1. the *type of the output* produced by the methods;
2. the *time instant* at which that output refers to.

Fault diagnosis methods produce (mostly) a *dichotomous* (healthy/failed) output indication at each *actual* time. The same applies for condition monitoring, except by producing a *continuous* output in the form of condition indicators. Fault prognostics builds on condition monitoring, by further providing values for the condition indicator (or indicators) that refers to *future* time instants.

Supervision solutions are characterized by their *aims and needs* and *methodological approaches* to reach those aims.

4.1. Aims and needs of supervision

The supervision aims considered in this article are fault diagnosis, condition monitoring and prognostics.

4.1.1. Fault diagnosis

Fault diagnosis is a reactive approach after the happening of faults and failures that entails the following tasks [35, Chapter 3.1], see Fig. 4:

1. *Fault detection*: to discover anomalous behaviors occurring in the items of the equipment. It consists in the *detection* (a 0/1 dichotomous logical output) of fault occurrence and the determination of the *time* at which the item switches to a failed state.
2. *Fault isolation*: (following fault detection) to *locate* a fault within the equipment, also among other faults. For instance, this may mean to recognize which item has failed.
3. *Fault analysis* or *identification*: (following fault isolation) to characterize the *type, size* (severity) and *nature* (cause or mechanism) of detected faults.
4. *Fault estimation*: (following fault isolation) to *reconstruct* the time-varying behavior (shape) of the fault signals. Fault estimation can serve as basis for control law reconfiguration or virtual sensor development [36, Chapter 14].

4.1.2. Condition monitoring

Condition Monitoring (CM) refers to the *continued oversight* of the progressive degradation of the monitored equipment or item. The main difference with respect to fault diagnosis lies in the output that CM provides when evaluating the presence of faults. In the fault diagnosis case, the main interest is in a binary answer, that is the presence or absence of a fault (and eventually where it is located and its type if fault isolation and identification are of concern). In CM, instead, the aim is for a *continuous evaluation* of the system health condition. Ideally, CM generates one or more condition indicators that monotonically evolve as equipment/item degradation progresses. Usually, this progression manifests with the lifetime increase, even if “infant mortality” behaviors can occur in some components.

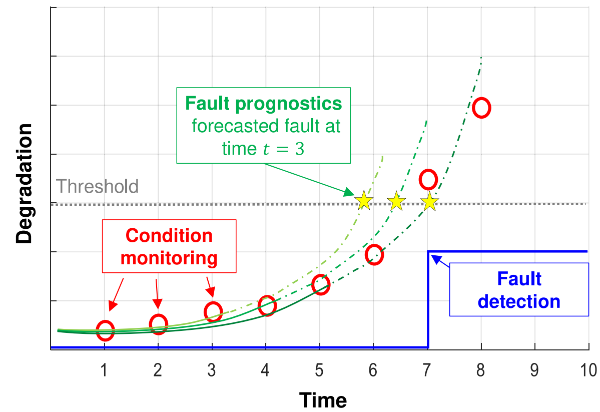

Fig. 5. Fault detection vs. condition monitoring vs. fault prognostics. Fault detection provides a binary output when the fault presence is detected. Condition monitoring produces continuous monitoring indicators of the item health state, that can be used for fault prognostics to forecast the health state at future times, given current monitoring information. The figure plots a monitoring indicator as a function of equipment/item lifetime. Blue dots represent the value given by the condition monitoring indicator, and the red dotted horizontal line is the fault threshold. Assume that a fault is present at time $t = 7$. At $t = 7$, the monitoring indicator exceeds the threshold, and the fault diagnosis (fault detection in this case) output goes from low (logical 0) to high (logical 1) state, indicating the detected presence of a fault. The prognostics module, at time $t = 3$, forecasts that the threshold will be exceeded at time $t = 6$. With more available data points, the prognostic model updates its predictions for the evolution of the monitoring indicator.

Fault estimation methods can be used to perform CM by directly estimating fault signals. In these cases, the condition indicator is the estimated fault signal.

4.1.3. Fault prognostics

Fault prognostics denotes procedures that employ a condition indicator, related to an item, to *forecast the indicator future evolution* [37]. The work [38] provide and excellent tutorial on how to design condition indicators and how to evaluate the performance of a prognostic solution. Whenever possible, condition indicators should be driven by the physics of the system [39].

The prognostics of a fault usually requires the development of a model for the evolution of the condition indicator, in order to extrapolate its future trend. This model can be based on physics laws, relying on a mathematical model of the failure mode (such as the rate of crack growth, or the state of charge of Li-ions batteries) or based on data, where a degradation model is fit on the indicators, and then an extrapolation from the model is performed [40, Chapter 6][41]. Physics-based models for bearing and gear faults propagation are given in [42, Chapter 19] relative to vibration parameters.

The extrapolated information from the model is used to estimate the Remaining Useful Life of the item at a certain actual time instant, that is interpreted as the period of time between: (i) the time instant when the item no longer meets its functional performance specifications (time to failure) and (ii) the actual time instant. Fault prognostics is an iterative procedure, since the forecasts have to be updated every time a new value of the condition indicator is provided by the CM function.

Fig. 5 provides a graphical representation the discussed supervision aims.

4.2. Methodological approaches for supervision

The most common methodological approaches for satisfying the supervision aims and needs can be categorized as [4,43–45]:

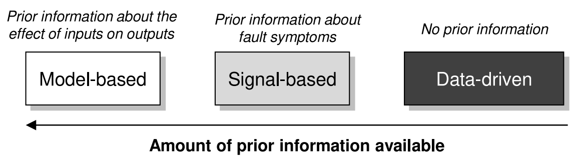

Fig. 6. Main methodological approaches for supervision. Model-based approaches exploit the highest amount of information about the equipment, since it models the inputs–outputs relations. Here, the term input refers to both measurable and controlled inputs, and also to not measurable ones such as disturbances, noises and faults. Signal-based approaches rely on a-priori known specific descriptors computed from specific signals. Data-driven approaches rely only on equipment data collected from sensors readings in specific operating regimes.

1. model-based;
2. signal-based;
3. data-driven.

The choice between a methodological approach and another depends on several factors, which are briefly described in the standard ISO 13379-1, along with suggestions on how to determine a confidence level for the diagnosis produced by such methods [18]. Model-based approaches are able to exploit prior information about the system dynamics and how faults and disturbances/noise enter the system. Signal-based approaches do not rely on a system model, but instead employ (or assume) certain signals behaviors to exploit specific diagnostic techniques for extracting fault indicators (*descriptors*[4]) from specific signals, using prior knowledge about the fault symptoms. Data-driven methods use the least amount of prior information about the faults, that has to be mined from data only, see Fig. 6.

We now briefly cover each of the three mentioned approaches.

4.2.1. Model-based approaches

Model-based approaches entail the *analytical redundancy* idea of comparing the actual input/output behavior of the monitored plant to the behavior simulated by a mathematical plant model. Plant measurements, inputs $\boldsymbol{u}(t) \in \mathbb{R}^{m_u \times 1}$ and outputs $\boldsymbol{y}(t) \in \mathbb{R}^{p \times 1}$, are checked for consistency with the mathematical model: the output of the consistency check are called *residuals signals* and denoted by $\boldsymbol{r}(t) \in \mathbb{R}^{q \times 1}$. Residuals are nominally zero, and become nonzero in the presence of: (i) faults $\boldsymbol{f}(t)$, (ii) disturbances $\boldsymbol{d}(t)$, (iii) noises $\boldsymbol{w}(t)$, and (iv) modeling errors. The analysis of the residuals allows producing supervision decisions. A model-based *residuals generator* is thus a dynamical system, with (m_u+p) inputs and q outputs, that given the plant measured inputs and outputs produces the residual signals. The residuals enter a *residuals evaluation* stage that outputs the diagnostic decisions. Fault detection is possible with a single residual ($q = 1$). Fault isolation almost always requires $q > 1$ residuals, and can be made possible by enhancing them with *structured* or *directional* properties [46, Chapter 7, Chapter 8].

The main model-based residuals generator implementations consist in the following approaches, and relationships have been found between them:

1. parity-space [35,46];
2. observers [27,36];
3. stable factorization (frequency-domain) [27,36,47];
4. parameters estimation [25,46].

Parity space and observers-based designs share many similarities, and are usually employed to deal with additive faults (i.e. when external unmeasurable signals enter the plant). One of the significant properties of parity-space residual generators, also widely viewed as their main advantage over the observer-based approaches, is that their design can be carried out in a straightforward manner. In fact, it only deals with solutions of linear equations or linear optimization problems, although numerical issues in the computation of the solution can be present. To solve this is issue, [35] proposed reliable computational methods to alleviate this issue [48]. Stable factorization approaches allow to define robust specification directly in the frequency domain, thus using robust control tools. Parameters estimation approaches are used to deal with multiplicative faults (i.e. changes in the plant physical parameters). A recent line of research consists in the *direct data-driven* design of residual generators, using subspace or other system identification approaches [49–51]: the idea is to directly estimate the residual generator from input/output data, without first estimating a model of the dynamical system. Another current research line is related to the use of the uncertainty of the identified model for increasing the robustness of the residual generator to modeling errors [52–55].

4.2.2. Signal-based approaches

Signal-based approaches entails the computation of descriptors from signal measurements, so that a supervision output is produced by comparing the actual descriptor values with *prior knowledge about the fault symptoms*. These methods lie on the assumption that certain signals carry information about the faults. Thus, the user has to know:

- *where* (which measurements) and,
- *what* (which descriptors computed from those signals),

to look for evaluating the health state of the item.

Supervision is achieved by comparing the actual value of the descriptors with their values in the healthy equipment state. Their employment is especially useful to supervise "passive" items such as mechanical or electrical components (bearings, gears, transformers, inverters, etc.) where it is not meaningful or not possible to develop a dynamical model of the item. The work [15] reviews, among commercial products for supervision, some common descriptors and processing techniques categorized into time, frequency and time–frequency domains, see also [43,44]:

1. *time domain*: root-mean-square value (RMS), kurtosis, crest factor, number of zero crossings, cycle counting (number of velocity reversals), on raw measurements or after preprocessing, for instance, but not limited to, correlation functions, envelope analysis, Dynamic Time Warping [56], cepstrum (when harmonics to be detected have small magnitude compared to the other ones) [57,58];
2. *frequency domain*: if the frequency content is time-invariant, descriptors are computed on frequency spectra obtained by Discrete Fourier Transform of the raw time domain measurements. Classical analysis of bearings and Motor Current Signal Analysis for electromechanical motors lie in this category [59,60]. If the frequency content is time-varying (e.g. it depends on the item rotation speed), order analysis can be used (given the availability of an additional tachometer signal) [61];
3. *time–frequency domain*: if the frequency content varies with time, methods as Short-Time Fourier Transform (STFT), wavelet transform and, for cyclostationary signals, Wigner-Ville Spectrum [62,63]. The work [64] provides a review of how Empirical Modes Decomposition can be used for supervision. It is however stated in [65] how such technique is less effective for bearings, and more fitting for gears.

The design of signal-based condition indicators for industrial equipment, given a probability of false alarms and starting from a deductive statistical point of view, is proposed in [66], while [67] proposed a unified framework for expressing common descriptors in equipment monitoring. Recent research is devoted to integrating signal-based approaches with data-driven (machine learning) ones, making the latter *physics-informed* [68–71].

[4] Data item derived from raw or processed parameters or external observations [17]. *Examples: amplitude of the first harmonic of the vibration acceleration, crest factor of the vibration acceleration, rotation speed.*

Vibrations analysis is a widespread instrument for monitoring equipment and their items, especially for *rotating* ones as bearings [40,57,72,73] or gears [74,75]. Acceptability levels for machine vibrations measured at *nonrotating* locations, given in terms of vibration velocity (mm/s), for different power classes of machines, is given in [76, Appendix A2] which is based on an earlier version of ISO 20816 [77].

4.2.3. Data-driven approaches

Data-driven approaches do not rely on a prior knowledge about system behaviors or symptoms, but they assume that this indication is present, but hidden, in the measured data and just need to be discovered. Data-driven approaches share a two-stages procedure:

1. a *training (offline) phase*, where supervision knowledge is represented using data in healthy and (possibly) failed condition;
2. an *evaluation (online) phase*, where the incoming data are compared with the information extracted from the training phase.

Many data-driven algorithms lie under the artificial intelligence (mainly machine learning) umbrella [78]. Several challenges are present in the adoption of artificial intelligence for supervision in industry [79,80], as: (i) imbalanced and unlabeled data sets; (ii) interpretability of the model decisions; (iii) integration of several heterogeneous data sources. To face those challenges, a recent research trend is the use of *transfer learning* methods, able to reuse the knowledge extracted from one problem domain to another (e.g. similar machines but different speeds, loads, etc.) [81,82].

Statistical Process Control (SPC) methods provide statistical indicators (most common of which are the T^2 and SPE statistics) for evaluating anomalous behaviors in the data, relying only on healthy training dataset. SPC methods are mainly based on Principal Component Analysis of the data matrix (where each row is an observation of a features/descriptors vector) [24,83]. Alternatively, Partial Least Squares methods can be used to correlate process variables with quality indicators [84,85]. Fault isolation can be performed by reconstruction plots [86], while prognostics with such plots is faced in [87]. The SPC indicators assume a static setting, i.e. they must consider data from the plant in a steady-state equilibrium, although some dynamical variants exist [88].

Other methods that follow under the category of data-driven approaches consist in classification and change-point detection algorithms [89–92]. As the name suggests, change-point detection is the task of finding changes in the underlying model of a signal or time series [93].

4.3. User guidelines

- *Definition of the supervision aims and needs*: prior to the development of the supervision solution, it is necessary to choose, for each item previously selected for supervision, which supervision aim should be pursued.
- *Choice of the methodological approach*: the following aspects should be considered:
    1. if the item being monitored is a passive one, use signal-based or data-driven approaches;
    2. if the item being monitored is an active dynamical one, and time and cost resources are available, use a model-based approach. This allows the residual to have some robustness against disturbances and noise. We suggest the approach of [35] as numerical aspects of implementation are taken into consideration. A software tool is available [48];
    3. several methodological approaches can also be combined, e.g. by evaluating model-based residuals with signal processing tools for further insights, like fault isolation [94].
- *Operating regimes and external factors*: especially for signal-based and data-driven approaches, the data collection during the *usage phase* of the supervision solution must be performed in conditions as similar as possible as during the *design phase* of the supervision solution.
- *Circumstances for detection*: the supervision solution is not necessarily required to be run during normal equipment operation. It is often possible to define *specific motions of the equipment with the only aim to collect data for supervision*. The supervision solution then will process these data, that can be periodically collected.

5. Design process for industrial supervision solutions

This section describes the proposed process for the design of supervision solutions for industrial equipment, making use of the concepts introduced in the previous sections. Three main steps are considered, each one articulated in several tasks as follows.

(S1) DEFINITION OF THE SUPERVISION SCOPE

- **T1.1** supervision aims and needs;
- **T1.2** technical specifications;
- **T1.3** critical items;
- **T1.4** supervision information;
- **T1.5** supervision approaches.

(S2) DATA COLLECTION

- **T2.1** experimental test plan;
- **T2.2** acquisition of measurements.

(S3) TECHNICAL DEVELOPMENT

- **T3.1** algorithmic technique;
- **T3.2** internal presentation of the results;
- **T3.3** deployment of the solution;
- **T3.4** human–machine interface.

Step **(S1)** is the starting point. It requires a clear definition of the aims and needs of the supervision solution. This step entails three tasks. Task **T1.1** specifies the supervision outputs that the supervision solution should provide. Task **T1.2** requires considering from the start the technical specifications of the hardware that should run the solution and the processing flow of the data. Task **T1.3** selects the items (and related failure modes) that will be the target, and will benefit the most, of the supervision solution. A study on how and under which circumstances (operating regimes + external factors) the symptoms of faults appear is essential. Each item (and relative fault/failure modes) may have a different supervision aim. Task **T1.4** analyzes the information that might help in performing the supervision on the items, as consulting maintenance reports, talk with maintenance operators and monitor physical quantities. Finally, Task **T1.5** requires the initial choice of the more adequate supervision approaches.

The second Step **(S2)** consists in the collection of experimental data in the form of equipment measurements. A test campaign shall be planned in Task **T2.1** to acquire data during defined operating regimes, in both healthy and (if possible) failed or degraded items states. Depending on the considered failure modes, specific physical quantities may have to be measured.

The third Step **(S3)** is devoted to the development of the supervision solution. Task **T3.1** selects the algorithmic techniques, based on the methodological approaches, to perform the supervision aims for each item. Task **T3.2** entails the internal presentation of the supervision solution results. Task **T3.3** focuses on the deployment of the solution onto the hardware platform running the supervision algorithms. Task **T3.4** considers the display of the supervision outputs to the end-user of the solution.

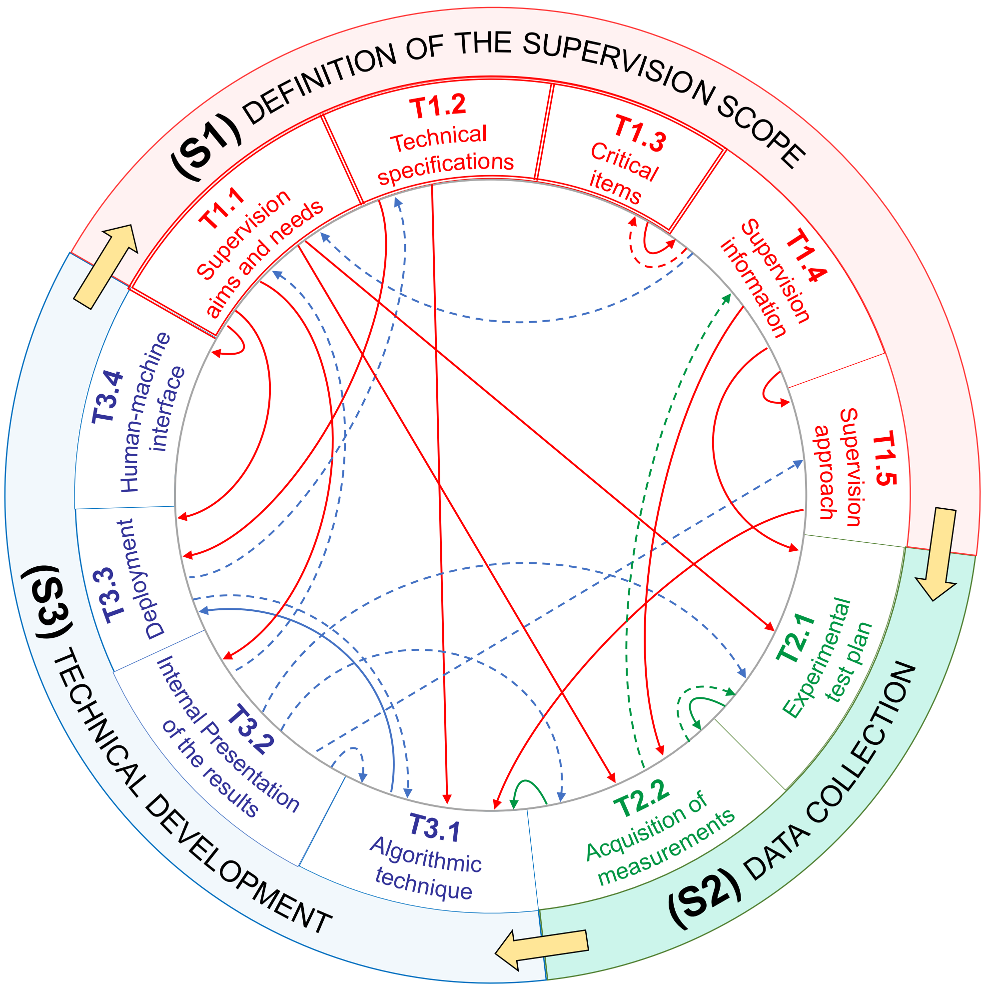

Fig. 7. Steps in the design of a supervision solution. The three main steps are at the top level, and each one is exploded into its tasks. The dependence between two tasks is shown by an arrow. The sink task (where the arrow points) depends on the source one (where the arrow starts). Continuous arrows denote a forward path in the task sequence, while dashed arrows represent a backward path. The starting tasks are highlighted by a double edge boundary.

Fig. 7, Table 4 and Figs. 8–19 show the most significant relations between the steps and their tasks. The proposed process for the design of a supervision solution is *iterative*. Often, a first prototype of the solution is developed, and then tested against requirements and specifications. If the tests are not successful, a retuning of the solution, or a change of the supervision aims, should be considered. The tasks are dependent: once a decision is taken on a particular task, it will influence several successive ones. Moreover, the execution of one task may shed light on previous decisions, so that the scope of the whole supervision solution can be refined.

The specific focus on the data collection tasks and on the relations between tasks is what differentiate the proposed process with respect to existing ones.

5.1. (S1) Definition of the supervision scope

The scope of the supervision solution entails:

1. the *supervision outputs* for each failure mode;
2. the *technical specifications* for its implementation;
3. the definition of the *items to be supervised*;
4. the gathering of all *supervision information*;
5. the choice of the *methodological approaches* for each failure mode.

The outputs of Step **(S1)** directly influence the tasks of Step **(S2)** and Step **(S3)**.

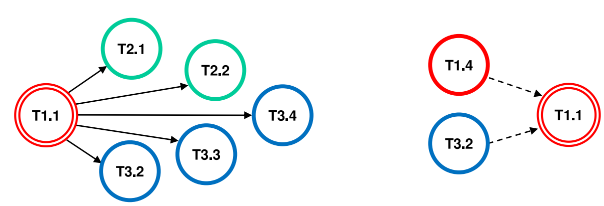

Fig. 8. Task T1.1 dependencies. An edge indicates that the source node influences the sink node. Dashed edges indicate that a subsequent task may have influence on a previous one.

5.1.1. T1.1 Supervision aims and needs

This task must answer the question:

Which kind of *outputs* the supervision solution shall provide?
Reference section: 4.1 "Aims and needs of supervision".

The clarification of what the supervision solution has to provide must be performed at the beginning of the design cycle. The definition of the supervision aims usually starts with a single aim that encompasses all items. After that all failure modes to be supervised are selected, it is possible to decide for a different supervision aim for each of them.

Table 4
Adjacency matrix of the relations between the tasks of the proposed design process. The (i, j)-th element denotes that the task at row i influences the task at column j. Filled squares (■) indicate that a task influences a subsequent one. Empty squares (□) indicate that a task influences a preceding one. The number of row-wise squares determines the overall influence of a task *over* other ones. The number of column-wise squares determines the overall dependence of a task *from* other ones.

	T1.1 aims	T1.2 specs.	T1.3 items	T1.4 info	T1.5 approach	T2.1 tests	T2.2 measures	T3.1 technique	T3.2 int. results	T3.3 deploy	T3.4 hmi	Row sum
T1.1 aims						■	■		■	■	■	5
T1.2 specs.								■		■		2
T1.3 items				■								1
T1.4 info	□		□		■	■	■					5
T1.5 approach								■				1
T2.1 tests							■					1
T2.2 measures				□		□		■				3
T3.1 technique										■		1
T3.2 int. results	□				□	□	□	□				5
T3.3 deploy		□						□				2
T3.4 hmi												0
Column sum	2	1	1	2	2	4	4	5	1	2	1	

The definition of which supervision aim is of interest influences the experimental tests (see Section 5.2.1 “T2.1 Define the test plan”), since:

- the *fault detection and isolation* aims will only require tests with healthy and (possibly) failed items. *Fault injection* has to be planned;
- the *condition monitoring* or *prognostics* aims will require (possibly multiple) endurance sessions to progressively degrade the item, leading to its failure. Especially in the prognostics case, this is important for building a statistical distribution of the times to failure. *Fault degradation* has to be planned.

The supervision aims and related outputs define how many resources (number of measurements, amount of data for each measurement, memory requirements) have to be managed, both in the development and usage stages of the solution (see Section 5.2.2 “T2.2 Acquire the measurements” and Section 5.3.3 “T3.3 Deploy the solution”). For instance, a fault detection aim may require to store only a batch of periodically replaced data, while a prognostics strategy, in addition, has to keep in memory also previous values of the monitoring indicator to update the prognostic model.

How to present the supervision outputs both internally and to the end-user depends on the aims (see Section 5.3.2 “T3.2 Present the experimental results internally” and Section 5.3.4 “T3.4 Present the supervision outputs to the end-user (human–machine-interface)”): if only fault detection is required, then a simple binary indicator, or sound, can be displayed. If condition monitoring or prognostics are of interest, a plot that shows the trends of the monitoring indicators, with warning and alarm thresholds, is appropriate. In both cases, however, caution should be paid to not show too erratic outputs behaviors: if so, the end-user will not get a good perception of the supervision outputs, and its trust in the provided solution will drop.

5.1.2. T1.2 Define the technical specifications

This task must answer the question:

> Which are the *technical characteristics* and *functionalities* of the hardware that will run the supervision solution?

Fig. 9. Task T1.2 dependencies.

Fig. 10. Task T1.3 dependencies.

The technical specifications should be considered in the early phases of the project, in particular regarding its limitations. These include computing power, memory requirements, Internet connection, supply power, programming language for the deployment. Their knowledge impacts the choices of algorithmic techniques, see Section 5.3.1 “T3.1 Choose the algorithmic technique” and Section 5.3.3 “T3.3 Deploy the solution”.

5.1.3. T1.3 Select the critical items

This task must answer the question:

> Which are the *items*, and relative *faults* or *failure modes*, that the supervision solution has to consider?
>
> **Reference section**: 3.2 “Criticality matrix”.

Most of the times, equipment producers desire a supervision solution for a machine composed by several items working together to fulfill an high-level functionality. The number of items can vary from few to many. In these latter cases, it is unfeasible to develop a supervision solution for all the items from scratch. Thus, a feasible rationale is to focus on the most critical items and their faults and failure modes.

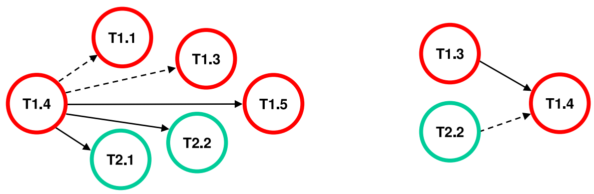

Fig. 11. Task T1.4 dependencies.

Fig. 12. Task T1.5 dependencies.

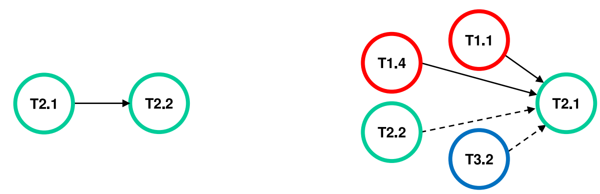

Fig. 13. Task T2.1 dependencies.

Furthermore, a maintenance strategy can be composed by a mixture of rationales: for some items, time-based, preventive, or even reactive maintenance might be the best solution. The selection of the items to be supervised requires interacting with the operation and maintenance personnel.

The critical items directly influence the supervision information that should be collected regarding them (see Section 5.1.4 "T1.4 Collect available supervision information").

5.1.4. T1.4 Collect available supervision information

This task must answer the questions:

1. Are there any *additional information* useful to perform the supervision aims?
2. Which are the *circumstances* that lead to a failure mode?
3. How it is possible to *reproduce* a failure mode?
4. Which *physical quantities* are mostly affected by the failure mode?

Reference section: 3.3 "Supervision-oriented FMECA worksheet".

Ancillary information that help in the supervision of the selected items are usually present, like historical records of fault occurrences, failure modes, failure rates, failure causes and mechanisms. Possible approaches to experimentally reproduce the failed conditions of the items can be discovered in this phase. Often, only a proxy the failed conditions can be introduced, as the failure/degradation process can be hardly reproducible. An analysis of the physical quantities that are most sensitive to faults and failure modes to be supervised should be also performed; such indications for typical machines and related failures is provided in [6,95]. As in the previous task, it is of uttermost importance to consult the operation and maintenance personnel.

The supervision information influence the supervision approaches that can be used, for instance based on the active/passive nature of the items and their faults or failure modes (see Section 5.1.5 "T1.5 Select the supervision approaches"), the type of experimental tests regarding the fault reproduction strategy (see Section 5.2.1 "T2.1 Define the test plan") and most suitable measurements for each fault (see Section 5.2.2 "T2.2 Acquire the measurements").

If the supervision information relative to an item are not enough accurate or informative supervise the item, its supervision aim or failure mode should be reconsidered (see Section 5.1.1 "T1.1 Supervision aims and needs" and Section 5.1.3 "T1.3 Select the critical items").

5.1.5. T1.5 Select the supervision approaches

This task must answer the question:

Which are the *supervision approaches* that are most suitable to supervise the selected failure modes?

Reference section: 4.2 "Methodological approaches for supervision".

A supervision solution can have different supervision approaches, one for each failure mode of the items; moreover, a failure mode can be supervised by more than one supervision approach. These choices influence the algorithmic technique employed (see Section 5.3.1 "T3.1 Choose the algorithmic technique").

5.2. (S2) Data collection

The data collection step entails:

1. the planning of the *experimental test plan* for each fault and failure mode;
2. the *acquisition of measurements* from each test run.

The outputs of Step **(S2)** directly influence the tasks of Step **(S3)**, but they can shed light also on the tasks of Step **(S1)**.

5.2.1. T2.1 Define the test plan

This task must answer the questions:

1. In which *experimental conditions* the tests should be run?
2. How much *time* the tests will require?
3. How many *experimental sessions* are required?

Reference section: 3.3 "Supervision-oriented FMECA worksheet".

The test plan definition should be carefully done prior to perform the experiments. The aim is to perform experimental tests in healthy condition and tests in a failed condition (that is, with a fault or degradation introduced in the item). This requires to have clear:

- what it meant by *healthy and failed states*;
- the fault *circumstances*, as they guide how the equipment will run during the tests. Tests should be performed also in typical (nominal) operating regimes, if there are any (e.g. a set of constant rotation speeds, nominal loads, etc.);
- if the equipment runs *with or without load*;
- the *motion profiles* of the item, including their duration, jerk and accelerations levels;
- the *set of measurements* that will be collected, along with the feasibility to install new sensors, both in the prototype and production supervision systems;

Fig. 14. Task T2.2 dependencies.

- the possible *degree of approximation* that the fault reproduction introduces. In a fault degradation test case, it might not be possible to degrade the component in a reasonable amount of time. In this case, it can be possible to *accelerate the degradation*, e.g. by removing lubricant or running the equipment at higher than nominal load conditions;
- the *repeatability* and *quantifiability* of the fault injection or degradation procedure. Also, the manufacturing variability of healthy items should be assessed, since a to high variability in the measurements can hidden the faults.

Usually, *one fault at a time* is evaluated for data collection, to avoid overlapping of fault and failure effects on the measurements. Fault injection of different faults requires time to dismount and remount the items. When the equipment returns to an healthy state, a test should be performed to check if the equipment performs as in the starting healthy condition (prior to any fault injection).

It is highly suggested to prepare a *spreadsheet test table*, where each row is a test, and the columns reports experimental information related to that test, as duration, load, motion profile, acquired measurements, notes and so on. This helps also in the quantification of the time needed for performing the tests.

Experimental tests on an item can be performed:

- with the item *directly mounted* on its equipment;
- with the item isolated in a dedicated *test bench*.

The definition of the experimental tests plan affects how measurements are collected (see Section 5.2.2 "T2.2 Acquire the measurements").

5.2.2. T2.2 Acquire the measurements

This task must answer the questions:

1. Are additional *sensors* required?
2. What is the *sampling frequency*?
3. Do measurements come from different sources and need to be *synchronized*?
4. How much computer *storage* is required?

Reference section: 3.3 "Supervision-oriented FMECA worksheet".

Physical variables can be acquired by the equipment built-in sensors and logging capabilities and by introducing additional sensors for prototyping the supervision solution. If the additional sensors are found necessary, they will have to be integrated in the equipment. It is important to consider the following aspects:

- if additional sensors are mounted, other than buying the acquisition hardware and set up the acquisition code (that requires time), it might be necessary to *synchronize* the measurements from these sensors and those already present in the machine;
- the amount of *computer storage* required;
- check the *repeatability* of the measurements by performing more experiments in the same settings;

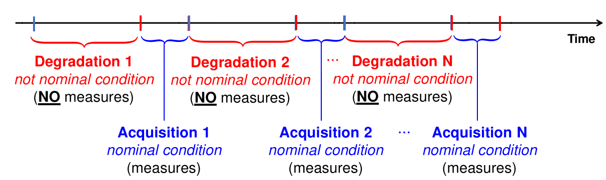

Fig. 15. Data collection strategy typical of fault degradation experiments. Degradation tests (usually in not nominal conditions to accelerate the degradation) are alternated with acquisition tests (in nominal conditions). In this way, a continuous acquisition is turned into a batch acquisition, with savings on computer storage.

- if fault degradation tests are considered, it might unfeasible to store the data continuously due to storage limits. A typical strategy is to alternate between *degradation tests* (where the item is degraded, usually in an accelerated manner, an no measurements are stored) and *acquisition tests* (where the item is run at nominal conditions and measurements are stored), see Fig. 15;
- prepare beforehand a computer script to *load and visualize* the data at regular intervals during their acquisition. If some sensor provides data that are clearly wrong or do not align with expectations, a promptly intervention can save an entire day of work.

The characteristics of the measurements affects the algorithmic technique that can be used (see Section 5.3.1 "T3.1 Choose the algorithmic technique"). Sometimes it is already known how measurements look like when there is a fault. In these cases, if measurements do not show the expected behavior (after a fast, also visual, data processing) it can be necessary to think of different ways to perform the experiments, for instance by changing operating regime or artificially inject an harsher fault condition on the item (see Section 5.2.1 "T2.1 Define the test plan"). If the measurements are not sensitive enough to the fault behavior, it may be necessary to rethink the information needed for the supervision (see Section 5.1.4 "T1.4 Collect available supervision information").

5.3. (S3) Technical development

The technical development step entails:

1. the choice and development of the *algorithmic technique* for each failure mode;
2. the *internal presentation* of the results obtained by applying the algorithmic technique to the experimental data;
3. the *deployment* of the solution;
4. the design of the *human–machine interface* for presenting the supervision outputs to the end-user.

The outputs of Step **(S3)** influence the entire Step **(S2)** and have repercussions also onto the tasks of Step **(S1)**.

5.3.1. T3.1 Choose the algorithmic technique

This task must answer the question:

Given a failure mode and a methodological approach, which *technique* is best to process the measurements for reaching the supervision aim?

Reference section: 4.2 "Methodological approaches for supervision".

The best supervision algorithm, for each considered failure mode and item, strongly depends on the chosen supervision approach, available measurements and the physics/functionality of the item. Model-based approaches may select a technique based on the modeling of

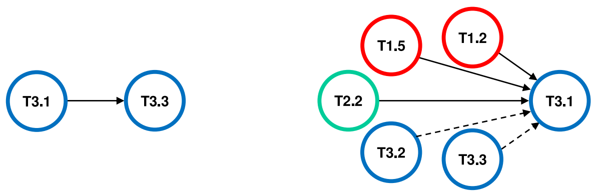

Fig. 16. Task T3.1 dependencies.

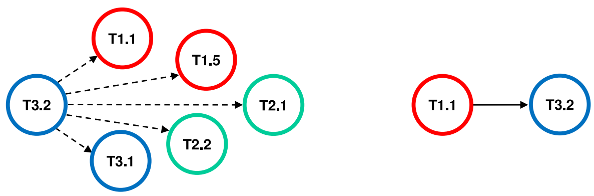

Fig. 17. Task T3.2 dependencies.

faults as additive or multiplicative, and robustness against noise and model uncertainties. Vibration measurements for rotating components benefit from a signal-based approach, with techniques rooted in envelope and frequency analysis. Techniques based on data-driven approaches can be distinguished based on the presence or lack of data in failed conditions. In the former case, classification methods can be used. In the latter, anomaly detection, statistical process monitoring or change detection techniques have to be used, that rely only on data characterizing the nominal healthy condition.

During the technical development, it is important to select the circumstances for the detection, that is a set of motion profiles, loads, etc. such that the collected signals are as clean as possible to be processed by the supervision algorithms.

The choice of the technique affects its deployment, due to its computational and memory requirements (see Section 5.3.3 "T3.3 Deploy the solution").

5.3.2. *T3.2 Present the experimental results internally*

This task must answer the question:

> How to *summarize the results* of the prototype supervision solution for internal discussion?
>
> **Reference section**: 4 "Supervision solutions".

A summary of the experimental results is where an overall evaluation of the supervision solutions is made. These results are usually produced by a model-in-the-loop paradigm (e.g. in MatLab, R, Julia or Python software environments). A quantitative evaluation of a supervision solution can consider the percentage of detection of a set of failures and their missed detections, along with the time required for their detection and robustness with respect switch-off of measured signals and symptoms delay [96]. A set of table templates for presenting qualitative results is proposed in Tables 5–7. The tables represent the supervision results of each item (failure mode) at different detail levels. Table 5 is a low-level representation where different algorithmic techniques and descriptors for the same items are compared, considering the measurements upon which they relied. The supervision aims and the need for additional sensors is also considered. The cells of the tables can contain a description of the symptoms and the circumstances where the data should be acquired for later processing. Table 6 is a mid-level representation, where the techniques and descriptors are considered on an aggregate level than Table 5, and gives more importance to needed measurements. Table 7 is a higher-level representation of the best possible set of sensors to reach the supervision aims. With decreasing detail level, only the best result is kept as indication.

Fig. 18. Task T3.3 dependencies.

If the solution does not provide the required performance, even if all necessary measurements have been collected, then it may be possible that the aims are not attainable. This may happen because the symptoms in the data are too weak with respect to the noise level, or because information that are wrong or unrelated to the failure mode are considered. In these cases, the problem definition and approaches should be revised (see Section 5.1.1 "T1.1 Supervision aims and needs", Section 5.1.5 "T1.5 Select the supervision approaches"). Another possibility for not satisfactorily results are experimental tests in not appropriate conditions or not informative measurements (see Section 5.2.1 "T2.1 Define the test plan" and Section 5.2.2 "T2.2 Acquire the measurements"). A modification of the algorithmic technique is possible (see Section 5.3.1 "T3.1 Choose the algorithmic technique").

5.3.3. *T3.3 Deploy the solution*

This task must answer the questions:

> *Where* the algorithms will run? *Which* information are computed? What is the *data processing flow* of information?

The integration and testing of the solution can be performed either in a software-in-the-loop or hardware-in-the-loop manner, and then directly evaluated on the equipment. For rapid control prototyping and hardware-in-the-loop, possible solutions are given by Speedgoat and DSpace hardware, both interacting via software with MatLab. For data acquisition, the combination of National Instruments CDAQ hardware and LabView software is a great choice. Moving closer to deployment, Siemens provides PLCs (with limitations on sampling frequency) that can be integrated with SM1281 accelerometers acquisition hardware (with much higher sampling frequency). In addition, computing capabilities can be extended with the edge device Siemens SIMATIC IPC427E. Here, applications can be developed using Python language.

The implementation for production should first consider the physical locations where the algorithms will run and which information is computed in each location. There are two main alternatives:

1. *on-premise*: the algorithms are implemented on a electronic control unit or an edge device, usually written in C or Python programming language;
2. *on-cloud*: the algorithms run on a web-server, on process aggregated data transmitted via an Internet connection.

In the first case, there are constraints related to memory and computational power that need to be allocated on the hardware that runs the supervision algorithms. In the second case, communication bandwidth and cost per data-packet transmission (especially in less developed countries) may pose a limit, so that only few aggregated monitoring indexes can be transmitted to the web-server for computation.

The initial deployment specifications are then updated and reconsidered as the development proceeds, as well as the algorithmic technique (see Section 5.1.2 "T1.2 Define the technical specifications" and Section 5.3.1 "T3.1 Choose the algorithmic technique").

Considering the deployment, there are not only technical choices, but also *commercial* ones. In fact, if the equipment producer is interested in selling the supervision solution as an optional service provided to

Table 5
Low-level detail results table, highlighting the algorithmic techniques. A green cell indicates that the technique was able to provide fault detection, and the results has been validated both theoretically and experimentally. A green cell with blue border indicates that a fault isolation result has been attained. A yellow cell indicates a fault detection result, validated only empirically without a strong theoretical support. A red cell indicates that the fault has not been detected. Blank cells indicate a not applicable or not considered case.

	Item	No additional sensors			With additional sensors	
		Position	Current		Vibration	
		Tracking error	*FFT*	*RMS*	*Envelope analysis*	*RMS*
Fault injection	Bearing	 + circumstances for detection and data collection + symptoms				
	Valve					
	Motor					
Fault degradation	Bearing					

Table 6
Mid-level detail results table. A green cell indicates that at least one algorithmic technique was able to provide fault detection, and the results has been validated both theoretically and experimentally. A green cell with blue border indicates that at least one fault isolation result has been attained. A yellow cell indicates at least a fault detection result, validated only empirically without a strong theoretical support. A red cell indicates that the fault has not been detected. Blank cells indicate a not applicable or not considered case.

	Item	No add. sensors		With add. sensors
		Position	Current	Vibration
Fault injection	Bearing			
	Valve			
	Motor			
Fault degradation	Bearing			

Table 7
High-level detail results table. The meaning of the cells colors is analogue to Table 6.

	Item	Nominal set of sensors + accelerometer
Fault injection	Bearing	
	Valve	
	Motor	
Fault degradation	Bearing	

their clients, they have to think about what is the most remunerative way to offer this functionality.

As suggested, the technological limitations of the computing hardware should be known beforehand. The software development team has to be included in the definition of the supervision algorithms, to share information about the hardware resources that will be available. For instance, the transcription of the code from a prototype implementation (e.g. MatLab) to a production one (e.g. C, Python) might undergo a performance degradation of the supervision, due to limited resources in the latter case that mandates for downgrading of the solution.

Standards have been proposed to describe the data processing flow of information in a supervision solution. In this context, ISO 13374 [97] proposes a sequence of steps to run a supervision solution. This is the basis for the Machinery Information Management Open Systems Alliance (MIMOSA) published a set of conceptual data representation schema to describe the communication of machinery information in computer systems resorting to an eXtendable Markup Language,

Fig. 19. Task T3.4 dependencies.

without proprietary protocols [98]. Additional standards are analyzed in [99].

5.3.4. T3.4 Present the supervision outputs to the end-user (human–machine-interface)

This task must answer the questions:

> *How* to present the supervision outputs to the end-user of the equipment in an informative way?

The display of the supervision outputs to the end-user should provide a mean to identify, confirm or understand an abnormal equipment state. As discussed in the introduction, [6] proposes four graphical representations to communicate the supervision results.

Although display formats should be customized for individual applications, for many users the display can be separated into five distinct areas that provide a summary of the equipment health state [97]:

1. *condition monitoring*: this area of the display presents information as trend data (e.g. a certain monitoring indicator versus operating time) with corresponding abnormality zones and warning/alarm thresholds;
2. *overall health assessment*: summarizes the results of condition monitoring and diagnosis. A health index on a scale from 0

Table 8
Simplified chronological view of the tasks of the proposed process.

Task	Description	Yes	No
Start: T1.1, T1.3	Have supervision aims and items been defined?	Go to T1.2	Go to T1.1, T1.3
T1.2	Has the technical hardware been defined?	Go to T1.4	Go to T1.2
T1.4	Has further information been collected?	Go to T1.5	Go to T1.1, T1.3, T1.4
T1.5	Have the supervision approaches been defined?	Go to T2.1	Go to T1.4, T1.5
T2.1	Has the test plan been defined?	Go to T2.2	Go to T1.4, T1.5, T2.1
T2.2	Has the data been acquired?	Go to T3.1	Go to T2.1, T2.2
T3.1	Have the algorithmic techniques been defined?	Go to T3.1	Go to T1.4, T1.5, T2.2, T3.1
T3.2	Have the results been presented?	Go to T3.3	Go to T3.1, T3.2
T3.3	Has the solution been deployed?	Go to T3.4	Go to T3.1, T3.3
T3.4	Has the HMI been tested?	Go to T1.1	Go to T3.3, T3.4
End	Is a review necessary?	Go to T1.1	Go to **End**

Table 9
Questions that should be answered during the design of a supervision solution within the proposed design process.

N.	Question
1	Which kind of *outputs* the supervision solution shall provide?
2	Which are the *technical characteristics* and *functionalities* of the hardware that will run the solution?
3	Which are the *items*, and relative *faults* or *failure modes*, that the supervision solution has to consider?
4	Are there any *additional information* useful to perform the supervision aims?
5	Which are the *circumstances* that lead to a failure mode?
6	How it is possible to *reproduce* a failure mode?
7	Which *physical quantities* are mostly affected by the failure mode?
8	Which are the *supervision approaches* that are most suitable to supervise the selected failure modes?
9	In which *experimental conditions* the tests should be run?
10	How much *time* the tests will require?
11	How many *experimental sessions* are required?
12	Are additional *sensors* required?
13	What is the *sampling frequency*?
14	Do measurements come from different sources and need to be *synchronized*?
15	How much computer *storage* is required?
16	Given a failure mode, which *technique* is best to process the measurements for reaching the supervision aim?
17	How to *summarize the results* of the prototype supervision solution for internal discussion?
18	*Where* the algorithms will run? *Which* information are computed? What is the *processing flow* of information?
19	*How* to present the supervision outputs to the end-user of the equipment in an informative way?

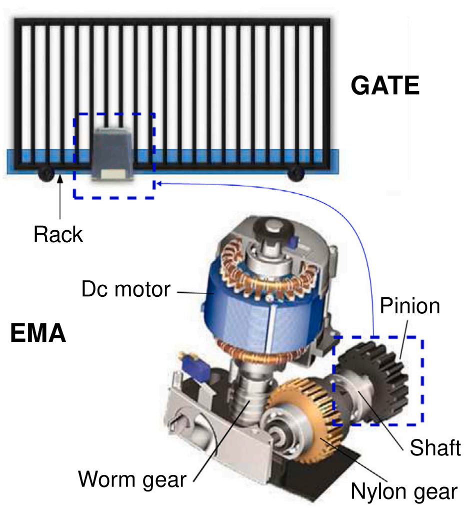

Fig. 20. Equipment items and their connections.

(complete failure) to 10 (as new) can be displayed, along with information about fault diagnosis (detection, isolation and identification);

3. *prognostics*: presents prognostics information as the estimated remaining useful life;
4. *recommended actions*: suggests recommended actions to be taken, as "replace or repair items", or "reduce the load";
5. *equipment information*: describes information about equipment number, item number, assessment date.

Table 8 illustrates a simplified chronological sequence of the tasks. Moreover, Table 9 summarizes the set of questions that should be answered during the tasks of the proposed design process. A practical use case is now presented to illustrate the proposed process, that considers the supervision of DC motor transmission items.

6. Illustrative example: supervision of transmission items from an electromechanical actuator

The following practical example illustrates the main steps of the proposed design process on fault detection and isolation of mechanical components from the transmission of an electromechanical actuator (EMA). The example is based on [94], reinterpreted in the context of this article.

The equipment consists of a EMA that actuates a sliding gate. The EMA contains a direct current motor with nominal voltage of $V_0 = 24\,\mathrm{V}$. The gate moves by means of steel wheels on a steel rail. The motor is connected to the gate through a transmission that converts the motor rotation to a linear movement. The transmission is composed of: (i) a worm gear, (ii) a nylon gear, (iii) a shaft, (iv) a pinion and (v) a rack that belongs to the gate.

Fig. 20 depicts how these items are connected. The worm gear is welded to the rotor of the motor, and it is coupled with the nylon gear. Since the rack is external to the EMA cover, a shaft connects nylon the gear to the steel pinion, which in turn it is paired with the gate rack. The EMA rotation is transformed into a linear motion by the pinon and the rack. An encoder measures the motor speed $\omega_M(t)$; this rotation speed can be converted to axial speed by the transmission ratio of the overall transmission.

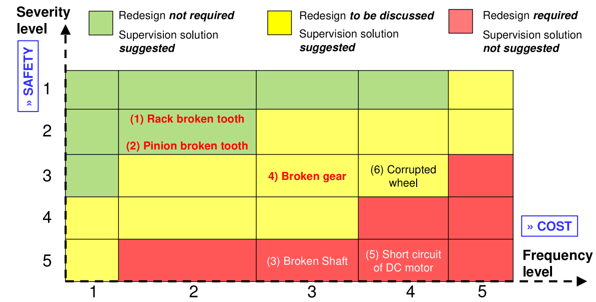

Fig. 21. Criticality matrix resulting from FMECA on the considered EMA equipment. Red bold entries are considered for investigation by the supervision solution.

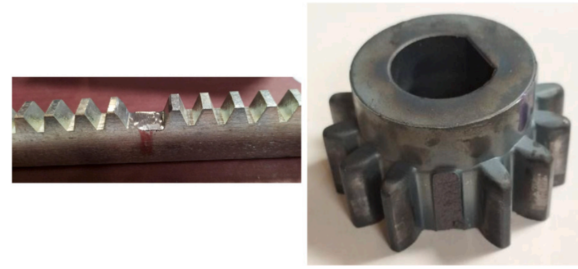

Fig. 22. Failed rack (left) and pinion (right) items.

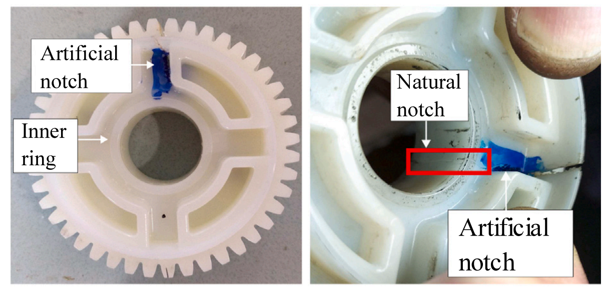

Fig. 23. Failed nylon gear without breaking the inner ring (left); natural notch that breaks the inner ring (right). The width of the natural notch is less than the width of the artificial one (highlighted in blue).

6.1. (S1) Definition of the supervision scope

6.1.1. T1.1 Supervision aims and needs

Design of a fault detection and isolation solution for a low-cost industrial EMA.

6.1.2. T1.2 Define the technical specifications

The supervision solution must be composed of algorithms that run on a low-cost consumer electronics, with a limited set of sensors (possibly, the ones already used for controlling the actuator).

6.1.3. T1.3 Select the critical items

The most occurring faults and failure modes, appearing in years of maintenance reports, are:

1. rack with broken tooth;
2. pinion with broken tooth;
3. broken shaft;
4. broken nylon gear;
5. short circuit of the motor;
6. corrupted wheel.

The resulting criticality matrix is depicted in Fig. 21. Based on this table and after considerations with the maintenance and operation personnel, the failure modes (and relative items) considered as focus of the supervision solution are:

- Rack with broken tooth;
- Pinion with broken tooth;
- Broken nylon gear.

6.1.4. T1.4 Collect the supervision information

Circumstances that lead to a fault or failure mode. The fault and failure circumstances are due to extended usage of the components and high gate loads.

Failure modes reproduction. Since the aim is fault detection and isolation, fault injection has to be performed. The rack and pinion faults were injected by removing a tooth using a vise.

To reproduce the break of the nylon gear, firstly one hundred gate movements (opening and closing) are performed to wear-in the component. After that, the fault is injected by carving perpendicularly the 80% of the total gear radius using a saw. The width of this notch is about 1 mm, see Fig. 23-(left). The fault is injected in the area where the gear is subject to the force applied by the shaft through its joint. In this notched condition, the inner ring of the gear, i.e. the part that delimits the shaft slot, is still not broken. Thus, to induce its breakage, the carved gear is mounted on the EMA and about 50 openings and 50 closings are performed. Fig. 23-(right) represents the condition of the gear after the 100 movements.

The nylon gear and pinion failure modes can be observed many times during the gate motion, but the failed portion of the rack is visible only one time per gate movement (see Fig. 22).

Useful measurements. The measurements that can be sensitive to the faults consist in the motor output speed, input voltage and current. No additional sensors can be added in production. However, for prototyping and validation purposes, a piezoelectric accelerometer is mounted on the motor housing.

6.1.5. T1.5 Select the supervision approaches

The items which are the focus of the supervision solution are passive components that rotates at constant speed for a considerable time, due to how the movement of the gate is performed. However, they are actuated by a motor, for which a model can be usually easily derived.

A model of a direct current motor, with voltage as input and axial speed as output is usually easily affordable. Since also isolation of faults is of interest, the combination of a model-based and signal-based approach is envisaged.

Since additional sensors on items cannot be considered for the final solution, motor measurements have to be used. For prototyping, the analysis of vibrations collected in constant speed conditions can be performed using standard spectrum and envelope analysis as pointed out in Section 4.2.

6.2. (S2) Data collection

6.2.1. T2.1 Define the test plan

The experimental protocol is composed of four different test plans: (i) tests with items in healthy condition; (ii) gear fault tests; (iii) pinion fault tests; (iv) rack fault tests.

Each test plan consists in opening and closing gate movements, interspersed with a break of 7 s, in order to not overheat the motor. The motor is commanded in open-loop with trapezoidal voltage profiles, that define acceleration, constant speed, and deceleration phases. The rise and fall times of the acceleration and deceleration phases have been set to 1 s (the minimum settable acceleration/deceleration time). This choice is motivated by two ideas: (i) perform movements that are stressful for the system (to enhance the fault detectability); (ii) practical use of the gates, where the fastest opening and closing movements (but within laws regulations) are usually desirable.

Experimental conditions. One fault at a time is considered. All experiments share the same gate, gate binary and environment.

Time required to perform the tests. The whole experimental protocol requires an entire day to be performed.

Number of experimental sessions. The whole experimental protocol is performed one time on two different EMAs, to evaluate the inner variability on the healthy items.

6.2.2. T2.2 Acquire the measurements

The following measurements are collected:

1. *motor speed* $\omega_M(t)$, measured by the motor encoder;
2. *motor working phase* $p(t)$, showing which working phase the motor is currently performing (acceleration phase, constant velocity phase, deceleration phase);
3. *motor current* $i(t)$, flowing in the DC motor coils;
4. *motor voltage* $V(t)$, powering the motor.

Additional sensors. An accelerometer is mounted on the motor housing.

Sampling frequency. The EMA hardware a custom electronic board allows the acquisition with a sampling frequency of $f_s = 5\,\text{kHz}$. The accelerometer is acquired by a National Instrument CDAQ with a IEPE module able to supply and acquire the sensor at 12.8 kHz.

Synchronization of measurements. Since the accelerometer data are collected from a different hardware than EMA measurements, a post-acquisition synchronization of the data is necessary. A synchronization signal is acquired by both hardware (custom board and CDAQ) to allow the synchronization of both vibration and EMA measurements.

6.3. (S3) Technical development

6.3.1. T3.1 Choose the algorithmic technique

The overall supervision solution consists in:

1. an output-error model-based approach for residual generation [25, Chapter 10], by comparing the motor axial speed with the one simulated by the model of the motor;
2. the FFT of the envelope of the residual signal, considering the constant speed phase of the EMA motion;
3. a Support Vector Machine (SVM) algorithm for the classification of two descriptors computed from the FFT spectrum derived in step 2.

Circumstances for the detection. The application of the envelope spectrum analysis necessitates of data collected during constant speed of the motor.

Residual generation. The motor model consists in a first-order dynamical model with a delay. The starting values of the model parameters are available from the datasheet of the motor. An identification phase is performed to align the model to experimental data.

The residual $r(t)$ is computed by an output-error based scheme given an input $V(t)$, as

$$r(t) = v(t) - \hat{v}(t), \tag{1}$$

where $v(t)$ is the measured motor axial speed (computed from the measure of $\omega_M(t)$), and $\hat{v}(t)$ is the axial speed simulated by the motor model.

Residual evaluation. Residual evaluation is performed to detect and isolate the faults into the considered fault categories. Envelope analysis is performed on the residual signal $r(t)$ in (1). The spectrum of the envelope of the residual is looked for the fault frequency f_{fault} that may appear on the shaft that connects the nylon gear to the steel pinion (which is then linked to the rack of the gate).

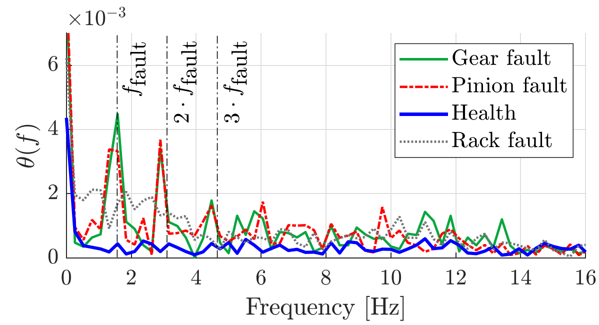

Fig. 24. FFT spectrum of the residual envelope signal.

Table 10
Confusion matrix of the classifier trained on all the data.

		Actual EMA state		
		Healthy	Pinon + gear faults	Rack fault
Estimated	Healthy	**130**	1	2
	Pinion + gear faults	0	**89**	4
	Rack fault	1	2	**31**

During the constant speed motion, the rotational motor speed is known and it is about $\omega_M(t) = 4100\,\text{rpm}$. The shaft rotational speed is $\frac{4100\,\text{rpm}}{44} \approx 93\,\text{rpm}$, where 44 is the ratio reduction from motor to shaft (i.e. the number of teeth of the nylon gear). Hence, the fault frequency of the components that are coupled to the shaft is about $f_{\text{fault}} = 1.55\,\text{Hz}$, see Fig. 24.

Decision logic. Fault isolation is performed by first computing two descriptors F_1 and F_2 from the FFT spectrum $\theta(t)$ of the envelope of the residual signal $r(t)$:

$$F_1 = \sum_{k=1}^{3} \theta(k \cdot f_{\text{fault}}); \quad F_2 = \sum_{k=1}^{3} \sum_{j=k\cdot f_{\text{fault}}\cdot 0.95}^{k\cdot f_{\text{fault}}\cdot 1.05} \theta(j). \tag{2}$$

The indicators in (2) extract the frequency amplitude at the first three harmonics of the fault frequency f_{fault} and in the area in their neighborhoods, respectively.

A linear SVM is then used to perform fault isolation. The SVM algorithm classifies the features in (2) into three classes: (i) health, (ii) pinion fault + nylon gear fault and (iii) rack fault. The nylon gear and pinion faults are difficult to isolate since these items are connected to the same shaft, so their rotation frequency is the same. Thus, a single "fault class" has been considered for their isolation.

6.3.2. T3.2 Present the experimental results internally

Fig. 25 depicts the 2D-plane composed by the F_1 and F_2 descriptors, computed from the first and the second tested EMA.

The classification boundaries show a good capability of isolating the various kind of failed conditions with an accuracy of 96.15%. The classifier performance is then evaluated by 10-fold cross-validation on all the data, resulting in an average cross-validation accuracy of 86.18% and a classification variance of 0.33%.

Table 10 presents the confusion matrix of this classifier, using all the data. Vibration data collected by the additional accelerometer did not provide useful information for the FDI aim, due to the high vibrations level produced by the gate.

Table 11 shows a summary of the supervision results.

7. Illustrative example: supervision of items from blow molding equipments in bottling plant

The equipment consists in a blow molding machine for bottling plant schematized in Fig. 26. The bottling process consists of eight

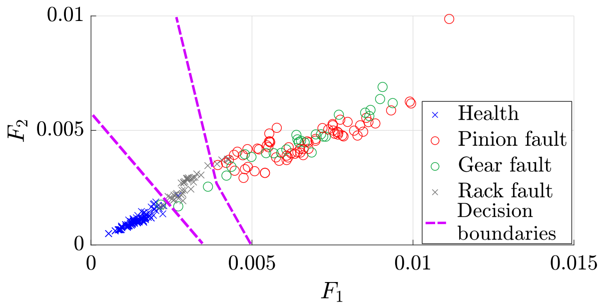

Fig. 25. Descriptors plane and faults classification boundaries. The pinion and gear faults overlaps and are not isolable using the developed solution. However, all faults are detectable and the rack one also isolable.

Table 11
Mid-level results table for the actuator equipment. The combined model-based and signal-based approach provides fault detection for all faults and isolation for the rack fault. The processing of vibration measurements, collected by an additional accelerometer placed at the motor housing, does not provide the detection of any fault.

	Item	No add. sensors	With add. sensors
		Voltage and Position	Vibration
Fault injection	Gear		
	Pinion		
	Rack		

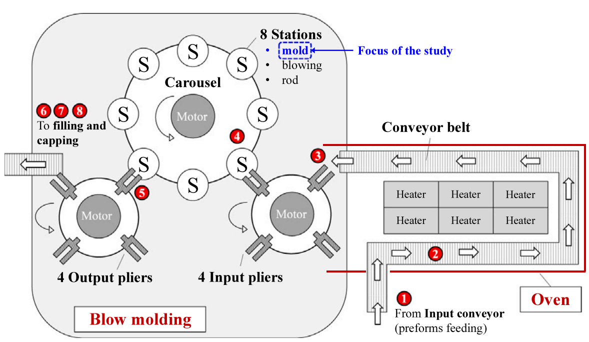

Fig. 26. Schematic of the blow molding bottling equipment. The steps of the production process, from (1) to (8), are highlighted with respect to the components responsible for each step. The focus of this example is on fault detection of the mold item.

steps [73]: (i) plastic preform feeding; (ii) heating of preforms; (iii) blowing of preforms; (iv) bottles filling and capping; (v) labeling; (vi) transportation along the production line; (vii) packaging and (viii) palletizing.

The equipment consists in the following components:

- the rotating *carousel*, that supports one or more blowing stations;
- the *stations*, the main component for blowing the preforms. Each station is made up of the following items:
  - a *mold*, that defines the form of the blown bottle
  - a *rod*, which is used to stretch the heated preform prior to its blowing;
  - the *blowing system*;
- the output and input *gripping pliers*. Their supervision is faced in [73].

In the following, we focus on the station component of the blow molding machine, responsible for shaping the plastic bottles when the heated plastic preforms are blew and thus formed. This covers the step 3 of the bottling process.

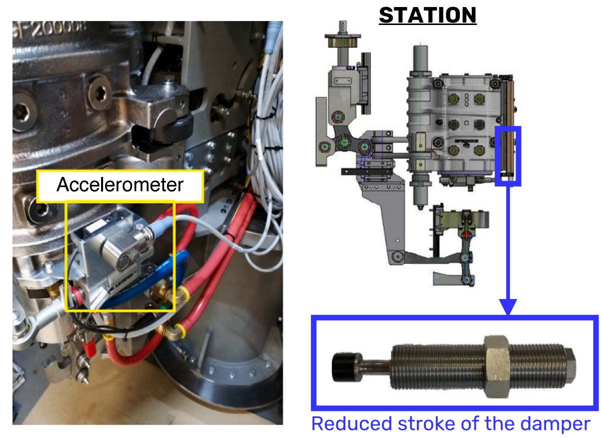

Fig. 27. Detail of the mold item. An accelerometer is added to characterize the mold movement. The fault injection regards the damper of the mold. In particular, we manually reduced the stroke of the damper.

7.1. (S1) Definition of the supervision scope

7.1.1. T1.1 Supervision aims and needs

Design of a fault detection solution for the components of the stations in the carousel of a blow molding machine.

7.1.2. T1.2 Define the technical specifications

The supervision solution can run on a dedicated electronic (e.g. an edge device). It is possible to use additional sensors.

7.1.3. T1.3 Select the critical items

Considering the station component, the most critical faults and failure modes are related to:

1. dust or iron chipboard on the rod;
2. loosened mold support;
3. loosened (reduced stroke) mold damper.

In the following, we focus on the mold damper item. The damper has the aim to attenuate the forces that are generated when the mold closes around the heated plastic preform, prior to preform blowing, see Fig. 27.

7.1.4. T1.4 Collect the supervision information

Circumstances that lead to a fault or failure mode. The circumstances of a loosened mold damper are due to extended usage of the components.

Failure modes reproduction. Since the aim is fault detection, fault injection has to be performed. The loosening of the damper has been performed by manually reducing its maximum allowable stroke through changing the nut position on the damper screw. The injected fault can be observed every time the mold closes.

Useful measurements. The measurements that can be sensitive to injected fault are the current of the mold motor and the mold piezoelectric accelerometer that we added as in Fig. 27.

7.1.5. T1.5 Select the supervision approaches

The mold item is passive components that is subject to compression forces when the mold closes. Thus, signal-based or data-driven approaches need to be used. Important variables might be the motor quadrature current (proportional to the motor torque) and the mold accelerometer.

7.2. (S2) Data collection

7.2.1. T2.1 Define the test plan

The experimental protocol consists in tests with a healthy mold damper and with mold damper with reduced stroke. We performed experiments at 1800 revolutions per hour of the carousel, where each test lasts about 3 min. We performed 10 tests in each healthy and failed condition. Each station presents a brushless motor that actuates each of its components (rod and mold). Motor-related measurements, as the voltage, currents and positions of such motors, are available for supervision purposes.

Experimental conditions. All experiments are performed at the same rotational speed and with the same type of fed plastic preforms.

Time required to perform the tests. The whole experimental protocol requires 4 h to be completed.

Number of experimental sessions. The whole experimental protocol is performed on two different stations (the experiments can be performed with two failed stations simultaneously).

7.2.2. T2.2 Acquire the measurements

The following measurements are collected by the electronic of the equipment: (i) current of the mold motor; (ii) position of the mold motor.

Additional sensors. An accelerometer is mounted on the side of the mold.

Sampling frequency. The equipment electronics allows the acquisition with a sampling frequency of $f_s = 1$ kHz. The accelerometer is acquired by a National Instrument CDAQ with a IEPE module able to supply and acquire the sensor at 12.8 kHz.

Synchronization of measurements. Since the accelerometer data are collected from a different hardware than the motor measurements, a post-acquisition synchronization of the data is necessary. A synchronization signal is acquired by both hardware (custom board and CDAQ) to allow the synchronization of both vibration and EMA measurements.

7.3. (S3) Technical development

7.3.1. T3.1 Choose the algorithmic technique

From the experiments, it has been noticed that the quadrature current that actuates the opening/closing of the mold is highly repeatable. Moreover, such current is sensitive to the loosening of the mold damper, as it influences the torque of the mold motor especially during closing and opening events. Thus, the supervision rationale is as follows:

1. using the current data in healthy tests, compute an "average signal" by averaging multiple current signals corresponding to an opening/closing cycle of the mold;
2. compute a residual signal that is the difference between the average and actual current signals;
3. compute the root-mean-square(RMS) value of the residual signal.

Fig. 28 depicts the torque of the mold motor (proportional to its current), during a single closing/opening cycle, in a healthy and failed condition. Differences in the magnitude of the residual signals are noticeable.

Circumstances for the detection. From the experiments, it has been noticed that the current of the mold motor is affected by the pressure generated by the air blew in the preform. To avoid such disturbance, the data for supervision should be collected without the blowing process. Moreover, it is important that the equipment has been run for several minutes, to wear-in its motors.

7.3.2. T3.2 Present the experimental results internally

Fig. 29 presents the experimental results, where a set of healthy tests and a set of failed tests are plot together. For different samples of the current (torque) profile, the RMS value of the residual signal is up to five times greater in the failed condition that in the healthy one.

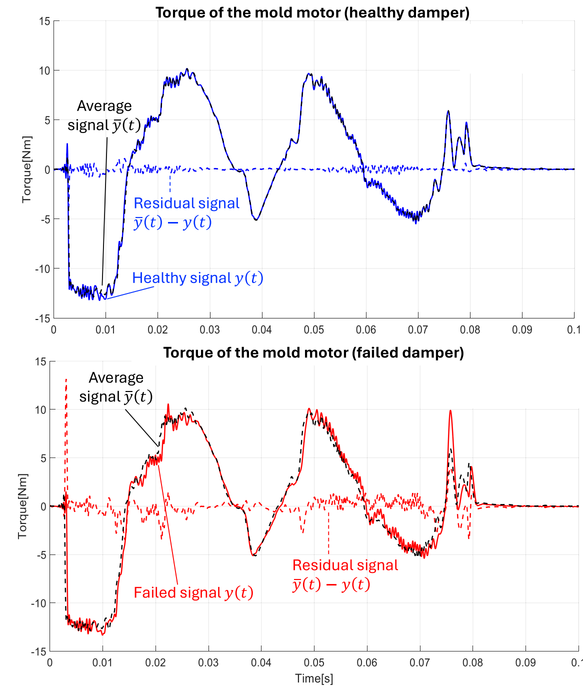

Fig. 28. Torque signals of the mold motor. (Top) mold in healthy conditions. (Bottom) mold in failed conditions. The residual signal, computed as the difference between the healthy and failed data, show increased magnitude in a failed condition.

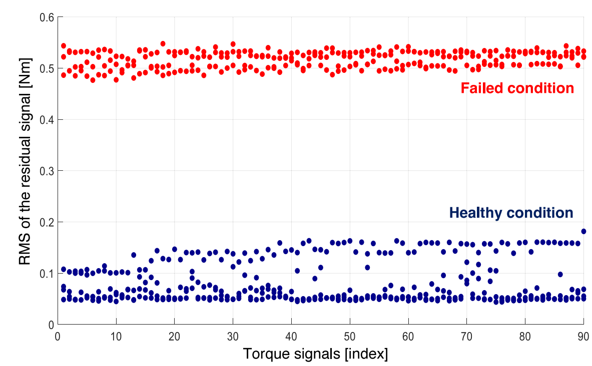

Fig. 29. Detection results for the loosened damper of the mold. Tests in failed conditions present higher values of the RMS of the residual signal.

8. Conclusions and outlook

The design of supervision solutions for industrial equipment requires a broad knowledge and wide set of "hard" competences, ranging from higher-level production and business skills to lower-level control and signal processing ones, and "soft" skills needed to understand and talk with the personnel. Thus, all these competences should be included in the team that designs the supervision solution.

This article proposed an iterative process for the design of such supervision solutions, comprising three main steps: "(S1) Definition of the supervision scope", "(S2) Data collection", and "(S3) Technical development". The most important points, which are the starting ones, are the clarification of the aims and needs for supervision; technical specifications for deployment and the definition of critical items. After an equipment audit, the items to be supervised, with their faults and failure modes, must be selected. This guides the types of experiments to be performed and measurements to be acquired. As a consequence, specific methodological approaches and algorithmic techniques can be evaluated on collected data.

Table A.12
Comparison of the steps in the design processes proposed in the literature.

Ref.	1. Cost–benefit analysis	2. Critical items Failure modes Causes and symptoms	3. Methodological approach	4. Decision-making 5. Review
ISO 13372 [17]	Cost–benefit analysis	Equipment audit Reliability and criticality audit	Maintenance strategy Monitoring method Data acquisition and analysis	Maintenance actions Review
Starr [1]		Criticality survey Maintenance audit Select units	Match technique to failure mode Routine monitoring	Assess technique and cost
Niu et al. [15]		Object identification Failure modes Failure effects	Maintenance tasks and interval Cost analysis Sensor module Signal processing Condition monitoring Health assessment Prognostics	Decision support Presentation
Al-Najjar [10]	Problem areas	Significant components Damage causes and development	Monitoring systems Technical justification Measuring system Data gathering Normal, warning and replacement levels Data analysis	Presentation of result Maintenance actions Human resources Technical and economic impacts Follow-up Maintenance roles
Rastegari et al. [14]	Concept study Define responsibilities	Assets selection	Techniques and technologies Installation, Data handling Training Measurement and setting baseline data Data analysis	Evaluation Improvement
López et al. [12]	Preparatory task and implementation plan	Asset hierarchy RCM analysis	Signals and detection methods for critical failure modes Algorithms to support decision-making	Transferring results Following the efficiency and effectiveness of maintenance
Lee et al. [6]		Critical components Sort, filtering, prioritize data	Features, Tools, Hardware Interface for DAQ Connectivity	Information visualization Management tools Value chain
This paper	Supervision aims and needs Technical specifications	Critical items Supervision information	Supervision approaches Experimental test plan Acquisition of measurements Algorithmic technique Presentation results Deployment, HMI	

The advance in autonomous supervision solutions, for instance using artificial intelligence methods, *must not disregard the understanding of the physics of the items being monitored, that is always important to improve the effectiveness of a supervision solution*. This includes the dynamic behavior of an item, its motion laws inside the equipment, and the loads it is subject to. To further support this statement, and in line with the industry 5.0 idea, the human operator can play a determinant role in the supervision of its equipment.

Computational and causal methods, able to take into account also human knowledge in the form of a causal graph, will be of utmost importance for rapid supervision of abnormal states. Future research challenges could be in this direction.

CRediT authorship contribution statement

Mirko Mazzoleni: Conceptualization, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

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.

Data availability

The authors do not have permission to share data.

Appendix. Comparison of design approaches

Table A.12 compares the steps of the design processes proposed in the literature for the design of supervision solutions. The grouping of the steps in five macro-steps has been performed to systematize the comparison.

References

[1] A. Starr, A structured approach to the selection of condition based maintenance, in: Fifth International Conference on Factory 2000 - the Technology Exploitation Process, 1997, pp. 131–138, http://dx.doi.org/10.1049/cp:19970134.

[2] A.K. Jardine, D. Lin, D. Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance, Mech. Syst. Signal Process. 20 (7) (2006) 1483–1510, http://dx.doi.org/10.1016/j.ymssp.2005.09.012.

[3] A.J. Guillén, A. Crespo, J.F. Gómez, M.D. Sanz, A framework for effective management of condition based maintenance programs in the context of industrial development of E-maintenance strategies, Comput. Ind. 82 (2016) 170–185, http://dx.doi.org/10.1016/j.compind.2016.07.003.

[4] M. Mazzoleni, G. Di Rito, F. Previdi, Electro-Mechanical Actuators for the More Electric Aircraft, Springer International Publishing, 2021, http://dx.doi.org/10.1007/978-3-030-61799-8.

[5] M. Rausand, A. Barros, A. Hoyland, System reliability theory: models and statistical methods, John Wiley & Sons, 2020.

[6] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, D. Siegel, Prognostics and health management design for rotary machinery systems - reviews, methodology and applications, Mech. Syst. Signal Process. 42 (1) (2014) 314–334, http://dx.doi.org/10.1016/j.ymssp.2013.06.004.

[7] A. Muller, A. Crespo Marquez, B. Iung, On the concept of e-maintenance: Review and current research, Reliab. Eng. Syst. Saf. 93 (8) (2008) 1165–1187, http://dx.doi.org/10.1016/j.ress.2007.08.006.

[8] J. Lee, M. Ghaffari, S. Elmeligy, Self-maintenance and engineering immune systems: Towards smarter machines and manufacturing systems, Annu. Rev. Control 35 (1) (2011) 111–122, http://dx.doi.org/10.1016/j.arcontrol.2011.03.007.

[9] T. Samad, B. M., S. Bortoff, S. Di Cairano, L. Fagiano, P. Odgaard, R. Sosseh, Industry engagement with control research: Perspective and messages, Annu. Rev. Control 49 (2020) 1–14, http://dx.doi.org/10.1016/j.arcontrol.2020.03.002.

[10] B. Al-Najjar, On establishing cost-effective condition-based maintenance: Exemplified for vibration-based maintenance in case companies, J. Quality Mainten. Eng. 18 (4) (2012) 401–416, http://dx.doi.org/10.1108/13552511211281561.

[11] H.N. Teixeira, I. Lopes, A.C. Braga, Condition-based maintenance implementation: a literature review, Procedia Manuf. 51 (2020) 228–235, http://dx.doi.org/10.1016/j.promfg.2020.10.033, 30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021).

[12] A.J.G. López, A.C. Márquez, J.F.G. Fernández, A.G. Bolaños, Towards the industrial application of PHM: challenges and methodological approach, in: PHM Society European Conference, Vol. 2, No. 1, 2014, http://dx.doi.org/10.36001/phme.2014.v2i1.1563.

[13] Y. Peng, M. Dong, M.J. Zuo, Current status of machine prognostics in condition-based maintenance: a review, Int. J. Adv. Manuf. Technol. 50 (2010) 297–313, http://dx.doi.org/10.1007/s00170-009-2482-0.

[14] A. Rastegari, M. Bengtsson, Implementation of condition based maintenance in manufacturing industry - a pilot case study, in: 2014 International Conference on Prognostics and Health Management, 2014, pp. 1–8, http://dx.doi.org/10.1109/ICPHM.2014.7036377.

[15] G. Niu, B.-S. Yang, M. Pecht, Development of an optimized condition-based maintenance system by data fusion and reliability-centered maintenance, Reliab. Eng. Syst. Saf. 95 (7) (2010) 786–796, http://dx.doi.org/10.1016/j.ress.2010.02.016.

[16] J.B. Revelle, J.W. Moran, C.A. Cox, The QFD handbook, John Wiley & Sons, 1998.

[17] ISO 13372:2012, Condition monitoring and diagnostics of machines - Vocabulary, Standard, International Organization for Standardization, 2012.

[18] ISO 17359-1:2012, Condition monitoring and diagnostics of machines - Data interpretation and diagnostics techniques - Part 1: General guidelines, Standard, International Organization for Standardization, 2012.

[19] M. Amarowicz, Requirements-based approach for the diagnostic systems design process, Expert Syst. 35 (6) (2018) http://dx.doi.org/10.1111/exsy.12300.

[20] A.J. Guillén, A. Crespo, M. Macchi, J. Gómez, On the role of prognostics and health management in advanced maintenance systems, Prod. Plan. Control 27 (12) (2016) 991–1004, http://dx.doi.org/10.1080/09537287.2016.1171920.

[21] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, J. Schröder, Diagnosis and Fault-Tolerant Control, third ed., Springer Berlin, Heidelberg, 2016, http://dx.doi.org/10.1007/978-3-662-47943-8.

[22] F.K. Pour, D. Theilliol, V. Puig, G. Cembrano, Health-aware control design based on remaining useful life estimation for autonomous racing vehicle, ISA Trans. 113 (2021) 196–209, http://dx.doi.org/10.1016/j.isatra.2020.03.032.

[23] A. Birolini, Reliability engineering: theory and practice, Springer-Verlag Berlin Heidelberg, 2017, http://dx.doi.org/10.1007/978-3-662-54209-5.

[24] S.J. Qin, Survey on data-driven industrial process monitoring and diagnosis, Annu. Rev. Control 36 (2) (2012) 220–234, http://dx.doi.org/10.1016/j.arcontrol.2012.09.004.

[25] R. Isermann, Fault-Diagnosis Systems - An Introduction from Fault Detection to Fault Tolerance, Springer-Verlag Berlin Heidelberg, 2006, http://dx.doi.org/10.1007/3-540-30368-5.

[26] M. Mazzoleni, K. Sarda, A. Acernese, L. Russo, L. Manfredi, L. Glielmo, C. Del Vecchio, A fuzzy logic-based approach for fault diagnosis and condition monitoring of industry 4.0 manufacturing processes, Eng. Appl. Artif. Intell. 115 (2022) 105317, http://dx.doi.org/10.1016/j.engappai.2022.105317.

[27] J. Chen, R.J. Patton, Robust Model-based Fault Diagnosis for Dynamic Systems, Springer US, 1999, http://dx.doi.org/10.1007/978-1-4615-5149-2.

[28] R. Isermann, P. Ballé, Trends in the application of model-based fault detection and diagnosis of technical processes, Control Eng. Pract. 5 (5) (1997) 709–719, http://dx.doi.org/10.1016/S0967-0661(97)00053-1.

[29] IEC 60812, Failure modes and effects analysis (FMEA and FMECA), Tech. rep., International Electronic Commission (IEC), 2018.

[30] MIL-STD-1629A, Procedures for performing a failure mode, effects and criticality analysis, Tech. rep., USA Department of defense, Washington DC, 1994.

[31] MIL-HDBK-217F, Reliability prediction of electronic equipment, Tech. rep., Reliability Information Analysis Center (RIAC), USA Department of defense, Washington DC, 1995.

[32] IEC 61709, Electric components - Reliability - Reference conditions for failure rates and stress models for conversion, Tech. rep., International Electronic Commission (IEC), 2017.

[33] D.J. Smith, Reliability, maintainability and risk: practical methods for engineers, Butterworth-Heinemann, 2021.

[34] FMD-2016, Failure Mode / Mechanism Distributions, Tech. rep., Reliability Information Analysis Center (RIAC), USA Department of defense, Washington DC, 2016.

[35] A. Varga, Solving Fault Diagnosis Problems, vol. 84, Springer International Publishing, 2017, http://dx.doi.org/10.1007/978-3-319-51559-5.

[36] S. Ding, Model-based fault diagnosis techniques: design schemes, algorithms, and tools, second ed., Springer Science & Business Media, 2013, http://dx.doi.org/10.1007/978-1-4471-4799-2.

[37] K. Javed, R. Gouriveau, N. Zerhouni, State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels, Mech. Syst. Signal Process. 94 (2017) 214–236, http://dx.doi.org/10.1016/j.ymssp.2017.01.050.

[38] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, J. Lin, Machinery health prognostics: A systematic review from data acquisition to RUL prediction, Mech. Syst. Signal Process. 104 (2018) 799–834, http://dx.doi.org/10.1016/j.ymssp.2017.11.016.

[39] L. Maurelli, M. Mazzoleni, F. Previdi, A. Camisani, A new physics-informed condition indicator for cost-effective direct current solenoid valves using significant points of the excitation current, J. Dyn. Syst. Meas. Control 146 (3) (2024) 031007, http://dx.doi.org/10.1115/1.4064602.

[40] R.B. Randall, Vibration-based condition monitoring: industrial, aerospace and automotive applications, second edition, John Wiley & Sons, 2021, http://dx.doi.org/10.1002/9781119477631.

[41] A. Heng, S. Zhang, A.C. Tan, J. Mathew, Rotating machinery prognostics: State of the art, challenges and opportunities, Mech. Syst. Signal Process. 23 (3) (2009) 724–739, http://dx.doi.org/10.1016/j.ymssp.2008.06.009.

[42] D.J. Inman, C.R. Farrar, V.L. Junior, V.S. Junior, Damage prognosis: for aerospace, civil and mechanical systems, John Wiley & Sons, 2005.

[43] Z. Gao, C. Cecati, S.X. Ding, A survey of fault diagnosis and fault-tolerant techniques - part I: Fault diagnosis with model-based and signal-based approaches, IEEE Trans. Ind. Electron. 62 (6) (2015) 3757–3767.

[44] Z. Gao, C. Cecati, S.X. Ding, A survey of fault diagnosis and fault-tolerant techniques - part II: Fault diagnosis with knowledge-based and hybrid/active approaches, IEEE Trans. Ind. Electron. 62 (6) (2015) 3768–3774, http://dx.doi.org/10.1109/TIE.2015.2419013.

[45] K. Tidriri, N. Chatti, S. Verron, T. Tiplica, Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges, Annu. Rev. Control 42 (2016) 63–81, http://dx.doi.org/10.1016/j.arcontrol.2016.09.008.

[46] J. Gertler, Fault Detection and Diagnosis in Engineering Systems, CRC Press, 1998.

[47] X. Ding, P.M. Frank, Fault detection via factorization approach, Systems Control Lett. 14 (5) (1990) 431–436, http://dx.doi.org/10.1016/0167-6911(90)90094-B.

[48] A. Varga, FDITOOLS - The Fault Detection and Isolation Tools for MATLAB, 2024-06-28, URL https://sites.google.com/view/andreasvarga/home/software/fditools.

[49] S. Ding, Data-driven design of monitoring and diagnosis systems for dynamic processes: A review of subspace technique based schemes and some recent results, J. Process Control 24 (2) (2014) 431–449, http://dx.doi.org/10.1016/j.jprocont.2013.08.011, ADCHEM 2012 Special Issue.

[50] S.X. Ding, Y. Yang, Y. Zhang, L. Li, Data-driven realizations of kernel and image representations and their application to fault detection and control system design, Automatica 50 (10) (2014) 2615–2623, http://dx.doi.org/10.1016/j.automatica.2014.08.022.

[51] J. Noom, O. Soloviev, M. Verhaegen, Proximal-based recursive implementation for model-free data-driven fault diagnosis, Automatica 165 (2024) 111656, http://dx.doi.org/10.1016/j.automatica.2024.111656.

[52] J. Dong, M. Verhaegen, F. Gustafsson, Robust fault detection with statistical uncertainty in identified parameters, IEEE Trans. Signal Process. 60 (10) (2012) 5064–5076, http://dx.doi.org/10.1109/TSP.2012.2208638.

[53] J. Dong, M. Verhaegen, F. Gustafsson, Robust fault isolation with statistical uncertainty in identified parameters, IEEE Trans. Signal Process. 60 (10) (2012) 5556–5561, http://dx.doi.org/10.1109/TSP.2012.2208639.

[54] P. Boni, M. Mazzoleni, F. Previdi, Robust data-driven design of a jamming detection filter for airborne electromechanical actuators, Eur. J. Control 75 (2024) 100926, http://dx.doi.org/10.1016/j.ejcon.2023.100926.

[55] M. Mazzoleni, N. Valceschini, F. Previdi, Model uncertainty-aware residual generators for SISO LTI systems based on kernel identification and randomized approaches, in: 2023 62nd IEEE Conference on Decision and Control, CDC, 2023, pp. 4849–4854, http://dx.doi.org/10.1109/CDC49753.2023.10383264.

[56] C. Shuai, Y. Sun, X. Zhang, F. Yang, X. Ouyang, Z. Chen, Intelligent diagnosis of abnormal charging for electric bicycles based on improved dynamic time warping, IEEE Trans. Ind. Electron. (2022) 1–10, http://dx.doi.org/10.1109/TIE.2022.3206702.

[57] R.B. Randall, A history of cepstrum analysis and its application to mechanical problems, Mech. Syst. Signal Process. 97 (2017) 3–19, http://dx.doi.org/10.1016/j.ymssp.2016.12.026, Special Issue on Surveillance.

[58] R. Randall, W. Smith, P. Borghesani, Z. Peng, A new angle-domain cepstral method for generalised gear diagnostics under constant and variable speed operation, Mech. Syst. Signal Process. 178 (2022) 109313, http://dx.doi.org/10.1016/j.ymssp.2022.109313.

[59] I. Culbert, J. Letal, Signature analysis for online motor diagnostics: Early detection of rotating machine problems prior to failure, IEEE Ind. Appl. Mag. 23 (4) (2017) 76–81, http://dx.doi.org/10.1109/MIAS.2016.2600684.

[60] J. Jung, Y. Park, S.B. Lee, C.-H. Cho, K. Kim, E.J. Wiedenbrug, M. Teska, Monitoring journal-bearing faults: Making use of motor current signature analysis for induction motors, IEEE Ind. Appl. Mag. 23 (4) (2017) 12–21, http://dx.doi.org/10.1109/MIAS.2016.2600725.

[61] D. Abboud, J. Antoni, Order-frequency analysis of machine signals, Mech. Syst. Signal Process. 87 (2017) 229–258, http://dx.doi.org/10.1016/j.ymssp.2016.10.024.

[62] J. Antoni, Cyclostationarity by examples, Mech. Syst. Signal Process. 23 (4) (2009) 987–1036, http://dx.doi.org/10.1016/j.ymssp.2008.10.010.

[63] J. Antoni, Cyclic spectral analysis in practice, Mech. Syst. Signal Process. 21 (2) (2007) 597–630, http://dx.doi.org/10.1016/j.ymssp.2006.08.007.

[64] Y. Lei, J. Lin, Z. He, M.J. Zuo, A review on empirical mode decomposition in fault diagnosis of rotating machinery, Mech. Syst. Signal Process. 35 (1) (2013) 108–126, http://dx.doi.org/10.1016/j.ymssp.2012.09.015.

[65] R.B. Randall, J. Antoni, Why EMD and similar decompositions are of little benefit for bearing diagnostics, Mech. Syst. Signal Process. 192 (2023) 110207, http://dx.doi.org/10.1016/j.ymssp.2023.110207.

[66] J. Antoni, P. Borghesani, A statistical methodology for the design of condition indicators, Mech. Syst. Signal Process. 114 (2019) 290–327, http://dx.doi.org/10.1016/j.ymssp.2018.05.012.

[67] D. Wang, Z. Peng, L. Xi, The sum of weighted normalized square envelope: A unified framework for kurtosis, negative entropy, gini index and smoothness index for machine health monitoring, Mech. Syst. Signal Process. 140 (2020) 106725, http://dx.doi.org/10.1016/j.ymssp.2020.106725.

[68] H. Zhou, X. Huang, G. Wen, Z. Lei, S. Dong, P. Zhang, X. Chen, Construction of health indicators for condition monitoring of rotating machinery: A review of the research, Expert Syst. Appl. 203 (2022) 117297, http://dx.doi.org/10.1016/j.eswa.2022.117297.

[69] D. Wang, Y. Chen, C. Shen, J. Zhong, Z. Peng, C. Li, Fully interpretable neural network for locating resonance frequency bands for machine condition monitoring, Mech. Syst. Signal Process. 168 (2022) 108673, http://dx.doi.org/10.1016/j.ymssp.2021.108673.

[70] B. Hou, D. Wang, J.-Z. Kong, J. Liu, Z. Peng, K.-L. Tsui, Understanding importance of positive and negative signs of optimized weights used in the sum of weighted normalized Fourier spectrum/envelope spectrum for machine condition monitoring, Mech. Syst. Signal Process. 174 (2022) 109094, http://dx.doi.org/10.1016/j.ymssp.2022.109094.

[71] B. Hou, D. Wang, Y. Chen, H. Wang, Z. Peng, K.-L. Tsui, Interpretable online updated weights: Optimized square envelope spectrum for machine condition monitoring and fault diagnosis, Mech. Syst. Signal Process. 169 (2022) 108779, http://dx.doi.org/10.1016/j.ymssp.2021.108779.

[72] R.B. Randall, J. Antoni, Rolling element bearing diagnostics - a tutorial, Mech. Syst. Signal Process. 25 (2) (2011) 485–520, http://dx.doi.org/10.1016/j.ymssp.2010.07.017.

[73] N. Valceschini, M. Mazzoleni, L. Pitturelli, F. Previdi, Experimental fault detection of input gripping pliers in bottling plants, IFAC-PapersOnLine 55 (6) (2022) 778–783, http://dx.doi.org/10.1016/j.ifacol.2022.07.221, 11th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2022.

[74] A. Brandt, Noise and vibration analysis: signal analysis and experimental procedures, John Wiley & Sons, 2011, http://dx.doi.org/10.1002/9780470978160.

[75] E. Bechhoefer, M. Kingsley, A review of time synchronous average algorithms, in: Annual Conference of the PHM Society, Vol. 1, No. 1, 2009.

[76] A.R. Mohanty, Machinery condition monitoring: Principles and practices, CRC Press, 2014.

[77] ISO 20816:2016, Mechanical vibration - Measurement and evaluation of machine vibration - Part 1: General guidelines, Standard, International Organization for Standardization, 2016.

[78] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, A.K. Nandi, Applications of machine learning to machine fault diagnosis: A review and roadmap, Mech. Syst. Signal Process. 138 (2020) 106587, http://dx.doi.org/10.1016/j.ymssp.2019.106587.

[79] H. Ding, R.X. Gao, A.J. Isaksson, R.G. Landers, T. Parisini, Y. Yuan, State of AI-based monitoring in smart manufacturing and introduction to focused section, IEEE/ASME Trans. Mechatronics 25 (5) (2020) 2143–2154, http://dx.doi.org/10.1109/TMECH.2020.3022983.

[80] N. Valceschini, M. Mazzoleni, F. Previdi, Inertial load classification of low-cost electro-mechanical systems under dataset shift with fast end of line testing, Eng. Appl. Artif. Intell. 105 (2021) 104446, http://dx.doi.org/10.1016/j.engappai.2021.104446.

[81] B. Yang, Y. Lei, F. Jia, S. Xing, An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings, Mech. Syst. Signal Process. 122 (2019) 692–706, http://dx.doi.org/10.1016/j.ymssp.2018.12.051.

[82] W. Li, R. Huang, J. Li, Y. Liao, Z. Chen, G. He, R. Yan, K. Gryllias, A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges, Mech. Syst. Signal Process. 167 (2022) 108487, http://dx.doi.org/10.1016/j.ymssp.2021.108487.

[83] M. Mazzoleni, F. Previdi, M. Scandella, G. Pispola, Experimental development of a health monitoring method for electro-mechanical actuators of flight control primary surfaces in more electric aircrafts, IEEE Access 7 (2019) 153618–153634, http://dx.doi.org/10.1109/ACCESS.2019.2948781.

[84] S.J. Qin, Y. Zheng, Concurrent projection to latent structures for output-relevant and input-relevant fault monitoring, in: 2012 IEEE 51st IEEE Conference on Decision and Control, CDC, 2012, pp. 7018–7023, http://dx.doi.org/10.1109/CDC.2012.6426571.

[85] Y.C. Pan, Y. Dong, S.J. Qin, Fault diagnosis using concurrent projection to latent structures, IFAC-PapersOnLine 48 (8) (2015) 1276–1281, http://dx.doi.org/10.1016/j.ifacol.2015.09.144, 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015.

[86] C.F. Alcala, S.J. Qin, Reconstruction-based contribution for process monitoring, Automatica 45 (7) (2009) 1593–1600, http://dx.doi.org/10.1016/j.automatica.2009.02.027.

[87] G. Li, S.J. Qin, Y. Ji, D. Zhou, Reconstruction based fault prognosis for continuous processes, Control Eng. Pract. 18 (10) (2010) 1211–1219, http://dx.doi.org/10.1016/j.conengprac.2010.05.012.

[88] E.L. Russell, L.H. Chiang, R.D. Braatz, Data-driven methods for fault detection and diagnosis in chemical processes, Springer Science & Business Media, 2012, http://dx.doi.org/10.1007/978-1-4471-0409-4.

[89] M. Mazzoleni, S. Formentin, F. Previdi, S.M. Savaresi, Fault detection via modified principal direction divisive partitioning and application to aerospace electro-mechanical actuators, in: Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on, IEEE, 2014, pp. 5770–5775, http://dx.doi.org/10.1109/CDC.2014.7040292.

[90] M. Mazzoleni, M. Scandella, Y. Maccarana, F. Previdi, G. Pispola, N. Porzi, Condition monitoring of electro-mechanical actuators for aerospace using batch change detection algorithms, in: 2018 IEEE Conference on Control Technology and Applications, CCTA, IEEE, 2018, pp. 1747–1752, http://dx.doi.org/10.1109/CCTA.2018.8511334.

[91] M. Mazzoleni, M. Scandella, Y. Maccarana, F. Previdi, G. Pispola, N. Porzi, Condition assessment of electro-mechanical actuators for aerospace using relative density-ratio estimation, in: 18th IFAC Symposium on System Identification (SYSID), Vol. 51, No. 15, 2018, pp. 957–962, http://dx.doi.org/10.1016/j.ifacol.2018.09.070.

[92] M. Mazzoleni, Y. Maccarana, F. Previdi, A comparison of data-driven fault detection methods with application to aerospace electro-mechanical actuators, in: 20th IFAC World Congress, Vol. 50, No. 1, 2017, pp. 12797–12802, http://dx.doi.org/10.1016/j.ifacol.2017.08.1837.

[93] C. Truong, L. Oudre, N. Vayatis, Selective review of offline change point detection methods, Signal Process. 167 (2020) 107299, http://dx.doi.org/10.1016/j.sigpro.2019.107299.

[94] N. Valceschini, M. Mazzoleni, F. Previdi, Model-based fault diagnosis of sliding gates electro-mechanical actuators transmission components with motor-side measurements, IFAC-PapersOnLine 55 (6) (2022) 784–789, http://dx.doi.org/10.1016/j.ifacol.2022.07.222, 11th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2022.
[95] ISO 17359:2011, Condition monitoring and diagnostics of machines - General guidelines, Standard, International Organization for Standardization, 2011.
[96] J.M. Koscielny, M. Bartys, The requirements for a new layer in the industrial safety systems, IFAC-PapersOnLine 48 (21) (2015) 1333–1338, http://dx.doi.org/10.1016/j.ifacol.2015.09.710, 9th IFAC Symposium on Fault Detection, Supervision andSafety for Technical Processes SAFEPROCESS 2015.
[97] ISO 13374:2019, Condition monitoring and diagnostics of machines - Data processing, communication and presentation - Part I: General guidelines, Standard, International Organization for Standardization, 2019.
[98] MIMOSA (Machinery Information Management Open Standards) Alliance, (Open systems architecture for condition based maintenance (OSACBM), v3.2.19, 2024, URL https://www.mimosa.org.
[99] G.W. Vogl, B.A. Weiss, M.A. Donmez, Standards for prognostics and health management (PHM) techniques within manufacturing operations, in: Annual Conference of the PHM Society, Vol. 6, No. 1, 2014, http://dx.doi.org/10.36001/phmconf.2014.v6i1.2503.