Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-023-02076-6

A framework for fault detection and diagnostics of articulated collaborative robots based on hybrid series modelling of Artificial Intelligence algorithms

Adalberto Polenghi[1] · Laura Cattaneo[2] · Marco Macchi[1]

Received: 7 September 2022 / Accepted: 2 January 2023

Abstract

Smart factories build on cyber-physical systems as one of the most promising technological concepts. Within smart factories, condition-based and predictive maintenance are key solutions to improve competitiveness by reducing downtimes and increasing the overall equipment effectiveness. Besides, the growing interest towards operation flexibility has pushed companies to introduce novel solutions on the shop floor, leading to install cobots for advanced human-machine collaboration. Despite their reliability, also cobots are subjected to degradation and functional failures may influence their operation, leading to anomalous trajectories. In this context, the literature shows gaps in what concerns a systematic adoption of condition-based and predictive maintenance to monitor and predict the health state of cobots to finally assure their expected performance. This work proposes an approach that leverages on a framework for fault detection and diagnostics of cobots inspired by the Prognostics and Health Management process as a guideline. The goal is to habilitate first-level maintenance, which aims at informing the operator about anomalous trajectories. The framework is enabled by a modular structure consisting of hybrid series modelling of unsupervised Artificial Intelligence algorithms, and it is assessed by inducing three functional failures in a 7-axis collaborative robot used for pick and place operations. The framework demonstrates the capability to accommodate and handle different trajectories while notifying the unhealthy state of cobots. Thanks to its structure, the framework is open to testing and comparing more algorithms in future research to identify the best-in-class in each of the proposed steps given the operational context on the shop floor.

Keywords Fault detection · Diagnostics · Collaborative robot · Cobot · Artificial intelligence

✉ Adalberto Polenghi
adalberto.polenghi@polimi.it

Laura Cattaneo
lcattaneo@liuc.it

Marco Macchi
marco.macchi@polimi.it

[1] Department of Management, Economics and Industrial Engineering, Politecnico di Milano, via Lambruschini 4/b, 20156 Milan, Italy

[2] School of Industrial Engineering, Università Carlo Cattaneo - LIUC, C.so Matteotti 22, 21053 Castellanza, Italy

Introduction

In the era of digitization, condition-based and predictive maintenance solutions are pervading smart factories (Bagheri et al., 2015; Osterrieder et al., 2020). Indeed, predictability, which in turn includes diagnosability, is core in building smart factories where Cyber-Physical System (CPS) is a key technological concept (Napoleone et al., 2020) to manage contingent situations including managerial errors bringing idle and down states due to failures (Mourtzis & Vlachou, 2018). The capability to act upon such unexpected events is fundamental to guarantee operational continuity and preserve product quality, leading towards improved firm competitiveness in a turbulent market (Morgan & O'Donnell, 2017).

Considering that smart factories are characterized by the introduction of new automation and manufacturing technologies on the shop floor, the heterogeneity of machines and systems that compose the production system is nowadays growing and it requires the capability to provide customized approaches to better deal with their specific requirements. Especially for condition-based maintenance (CBM) and predictive maintenance solutions, several challenges are to be faced, including both cultural and technical aspects (Bokrantz et al., 2020; Ingemarsdotter et al., 2021).

Specifically, robotic applications are increasing (Forcina et al., 2021) as they provide support to the operators, reducing the cognitive and physical efforts in a collaborative human-robot framework (Matheson et al., 2019). Therefore, the interaction with humans led to the concept of a “collaborative robot”, in short, a “cobot”, which combines the repetitive performance of robots with people’s skills and abilities. The spread of collaborative robots is due to the increased variety required in production systems for assembly and disassembly operations (Hashemi-Petroodi et al., 2020). Cobots are extremely flexible and could be trained on the field by humans (El Zaatari et al., 2019). Also, the need for almost no pre-training makes them adaptable to any situation they may face (Djuric et al., 2016).

The adaptability in the operations should have a mirror in maintenance: to enable the complete description of machines and systems in the CPS’s virtual world to support diagnosability and predictability, cobot-related models and algorithms should be included. With specific reference to CBM and predictive maintenance within CPS, optimizing the operations of the production system requires the knowledge of possible deviations from nominal requirements as well as the faults and related diagnoses incurring in different machines and systems, including cobots. Indeed, the development of CBM and predictive maintenance solutions for cobots is needed both for optimal decision-making in the production system, and the safe operations through the prevention of hazardous situations in which humans may be due to faults the cobot is experiencing.

The present work aims at contributing to cobot maintenance by supporting fault detection and diagnostics (FDD) at its initial stage when the deviation from nominal requirements is detected. In fact, considering a maintenance practice generally valid for industrial systems, cobot maintenance is assumed to act at two levels depending on the available data and the complexity of the solution (Márquez, 2007; Lee et al., 2011; Khan et al., 2020). In the first level, a deviation of the operation is identified to notify maintenance technicians to reactively act on the system; at this level, a first understanding can lead to describing the deviation from the baseline (healthy) behaviour by identifying the related functional failure. The second level includes information such as which fault is occurring, to isolate the faulty items determining the observed behaviour and, more accurately, the failure mode under evolution within those items. Finally, it provides a physical sense of the failure modes manifested in specific components (e.g. abrasive wear in the outer ring of a roller bearing). A diagnostic approach is then required to make the maintenance intervention.

Given these premises, this research aims to develop a data-driven solution for FDD of articulated collaborative robots that identifies deviation during cobot operation, providing a first understanding of the deviation, described in terms of the functional failure that is evolving; this is to be implemented, at the first level, autonomously by the operator who is advised to call technicians for further condition investigation. The diagnostics of faults, performed at the second level, will start after this call. The diagnosis at the second level is outside the scope of the present research.

The assumptions, defined based on a usual maintenance practice for industrial systems, were required because the review of the state of the art performed in this research work, enabled to provide only limited insights on cobot maintenance. The assumptions enable to align with a norm in industrial practice; within it, the innovation provided by this research affects the first level of intervention.

In particular, real-time data from the cobot are collected and elaborated via a hybrid modelling of Artificial Intelligence (AI) algorithms, as each algorithm accomplishes a specific task along the Prognostics and Health Management (PHM) process taken as a reference to develop the solution. Advancing from the current state of the art, a hybrid modelling approach is adopted, instead of a monolithic solution, to create a modular solution. The proposed solution is tested and validated using a 7-axis collaborative robot performing multiple trajectories in a laboratory environment. Thus, the framework, based on hybridised AI algorithms, demonstrates the capability to accommodate and handle different trajectories, also those trained by the operators on the shop floor; this finally allows for identifying those trajectories deviating from the nominal operation.

Overall, the perspective considered in the paper is one of applied research. Therefore, an application is focused, considering its manageability by an industrial engineer. Emphasis is then given to the structure of the proposed framework as well as the information enabling the operator to judge the need for maintenance intervention. The selected algorithms are used to show the effectiveness of the framework and its feasibility, whereas this research work does not claim these algorithms to be best-in-class for the application; instead, the framework is open to test and compare more algorithms to identify the best performers in each of the proposed steps.

The research paper is structured as follows: Sect. "Literature review on CBM and predictive maintenance solutions for cobots" reviews the extant scientific literature on CBM and predictive maintenance solutions for cobots to define the state of the art; Sect. "Framework and hybrid series modelling of AI algorithms for FDD of cobots" proposes the framework for the hybrid series modelling, and the related design choices; Sects. "Application of the framework" and "Artificial Intelligence algorithms tuning" show the experimental deployment and the assessment of the framework in the case of a 7-axis collaborative robot; eventually, Sect. "Results and discussion" elaborates over the obtained results, while conclusions and future research are drawn in Sect. "Conclusion".

Literature review on CBM and predictive maintenance solutions for cobots

The relevance of cobots in the current factories has gained momentum thanks to Industry 4.0 and smart factory concepts (Sherwani et al., 2020), while emphasis was given in the past on "hard automation", which compels the realisation of pre-defined tasks in an optimised way. Since collaborative robots are being more and more present at the shop floor level for various tasks jointly with humans, they need to be connected to the factory-level CPS, exchanging data and information, in order to make them part of the decision-making process for maintenance and production. Therefore, the health state of such equipment must be evaluated as well as for other equipment, like CNC machine tools, in order to provide the managers with all the information to judge prompt decisions of reconfiguration of the production (Tao et al., 2018). Hence, the goal of this literature review concerns the definition of the state of the art about maintenance solutions for collaborative robots, with particular attention towards CBM and predictive maintenance as cutting-edge technologies in various industrial fields.

Systematic literature review setup and application

The literature review aims at defining the state of the art of CBM and predictive maintenance solutions for cobots, by exploring which are the developed algorithms and models for FDD. Even though human-cobot collaboration is prominent in the current literature (Faccio et al., 2022), the scope of the review is centred only on the cobot as a technical system, in order to establish the background to develop effective solutions without presuming human intervention as a prerequisite.

To span the scientific knowledge about maintenance solutions for collaborative robots, a systematic literature review (SLR) has been established. The adopted methodology of the SLR is the one of (Brereton et al., 2007; Kitchenham et al., 2009), whose core characteristic is reproducibility. The steps are research protocol definition, search process, including screening, and document analysis, with descriptive statistics and content analysis. The research protocol, including keywords, databases, and eligibility criteria have been set as in Table 1.

Table 1 Research protocol for the systematic literature review

Research protocol	
Keywords	(*cobot** OR *"collaborative robot*"*) AND maintenance
Database	Scopus, Web of Science, IEEE Xplore
Eligibility criteria	Only English-written document

The results of the search process consist in a set of 70 documents from Scopus, 53 from Web of Science, and 4 in IEEE Xplore, for a total of 127 articles out of which 36 are duplicated and additional 8 documents do not have authors information as they are proceedings or collections. Therefore, 83 documents are considered eligible for the screening process. This set has been screened so as to include only documents compliant with the current research work in terms of scope, i.e., the development of maintenance solutions for collaborative robots. After title screening, 34 documents are kept, reduced to a set of 10 after the abstract screening. Finally, only 4 are eligible after full-text reading.

Content analysis and results

The 34 documents after title screening are published from 2017 onwards, anyhow highlighting that the research is relatively novel. However, after abstract screening and full-text reading, most of the selected works resulted not actually eligible, lacking a specific focus on CBM or predictive maintenance solutions for cobots. As a matter of fact, human-cobot collaboration is mainly tackled, as supported also by the results of the recent work by Aliev & Antonelli (2021). Indeed, these research works focused on the human-cobot interaction for various sectors (such as the meat production (Romanov et al., 2022) and civil engineering (Nagatani et al., 2021)) and operations (e.g., for productivity improvement (Mitrea & Tamas, 2018) and internal logistics (Donadio et al., 2018)). Most of the works focused on safety issues related to the use of collaborative robots in the same work environments as humans, by favouring the interaction (Bagheri et al., 2022), defining algorithms for trajectories and collisions prediction (Zhang et al., 2022). Some of them specifically target the use of collaborative robots in order to support maintenance intervention in harsh and hazardous environments, for example, to support ship maintenance, repair and conversion (Zacharaki et al., 2022), crack predictive maintenance in buildings (Kahouadji et al., 2021), and screwing operations (Koch et al., 2017).

Table 2 Summary of eligible research papers

Reference	Objective and description
(Wescoat et al., 2022)	The objective of the work is to compare three different machine learning algorithms to characterise the overall health state of a 6-axis collaborative robots. The authors compared Random Forest Regression, Support Vector Regression and a Deep Neural Network Regression, where the anomalous condition was the overload of the end-effector. Each algorithm has different performance with respect to the training, validation and test accuracy and a mixture of algorithms is suggested. The cobot under study was a Universal Robots UR10 and the work is an extension of the one below reported, namely (Wescoat et al., 2021).
(Aliev & Antonelli, 2021)	This research stems from the safety legislations needs for cobots to use real time and online data to predict failures and safe stops. The proposed solution relies on a regression model MLP and a classification model; the former is in charge of predicting quantitative variables, while the latter for safe stops. The cobot under study was a Universal Robots UR3.
(Wescoat et al., 2021)	The objective is to describe the health state of a collaborative robot by means of a Random Forest Regression with varying predictors. Failure data are simulated by adding weights (from 0.2 Kg to 1.0 Kg) to the end-effector. The five-fold cross-validation shows that the joint dynamics and current features result in good model performance. The cobot was a Universal Robots UR10.
(Park et al., 2021)	The authors faced the problem of applying FDD to non-fixed programmable motions. This was done through the development of a programmable motion-fault detection based on the analysis of motion residuals with respect to a representative data pattern determined by a target movement. In that way, also diagnostics is favoured, even though it is a future work. The collaborative robot used for experimentation was a 6-axis Niryo One cobot.

Among the findings, it is worth mentioning the work by (Xiao et al., 2021), whose goal is the reliability analysis and maintenance optimisation for systems of cobots. Even though it does not tackle directly FDD for the cobots, it proposes, by means of reliability engineering, the improvement of maintenance management for assembly lines where cobots are organised in cells. Aligned with this work, cobot fleet management also requires a huge effort from the architectural point of view to collect operational as well as production data, as remarked in (Ismail et al., 2020). Overall, out of the 34, 4 documents specifically face the issues of developing maintenance solutions to identify, diagnose and predict failures of collaborative robots. These are listed in Table 2.

As relevant remarks from the eligible documents, it is suggested by Wescoat et al., (2022) that the main limitation of AI-based approaches for FDD of cobots is the limited availability of failure data. Hence, laboratory applications with simulated failures (overloads and excessive friction) are required to develop solutions to be later transferred to real contexts. Also, some insights for the data analysis emerge from the selected papers. Firstly, it is worth remarking that, in each joint of a cobot, some variables may be correlated, especially the temperature and the load (Aliev & Antonelli, 2021). Afterwards, Wescoat et al., (2021) point out how the starting position of the cobot may influence the model performance. Overall, all the presented solutions adopt a one-shot approach, that is, trying to predict the health state of the collaborative robots through one algorithm only, fitted to the specific case under study.

A recent study by A recent study by Faccio et al. (2022) also confirms the shortage of CBM / predictive maintenance solutions framed within the context of human-cobot collaboration. In their study, the authors claim that consistency (designated as cobot capability to work reliably) has been not directly linked with any human factors as, instead, it is happening with the ergonomics or usability. It finally confirms the need to study cobot maintenance, now under-investigated, to guarantee reliable behaviour in cobot operations.

Concluding remarks from the literature review

From the eligible documents, it is possible to conclude that CBM and predictive maintenance solutions are still in the infant phase for collaborative robots. Moreover, CBM solutions face a challenging task, due to the intrinsic characteristics of the flexibility of cobots and the subsequent possibility to train them for new trajectories on field (Park et al., 2021). Overall, the approaches to cobot CBM and predictive maintenance are varied and there is no unique framework to follow to deploy FDD-purposed solutions.

As such, this research work aims at contributing to this field by proposing a framework, and related AI algorithms properly hybridized, for FDD, limited to health monitoring of the cobot in the light of possible deviations in the performed trajectories. It represents a first move towards the development of CBM and predictive maintenance of collaborative robots: given the increasing centrality in smart factories, cobots must be monitored and proactively maintained to preserve human safety and optimise production throughput. As the first move, the solution development addressed by this research is not considering human-cobot collaboration. This is noted as an extant gap to be addressed in future works aimed at integrating the human-in-the-loop of the CBM and predictive maintenance.

Framework and hybrid series modelling of AI algorithms for FDD of cobots

Given that there is no consolidated background on how to develop CBM and predictive maintenance solutions for cobots, the PHM process has been taken as a reference.

The PHM process is relevant for maintenance decision-making as it provides a general architecture over which multiple technology-independent solutions could be realised (Guillén et al., 2016). As described in ISO 13374-1, the steps, therein pointed out as "functional blocks", are: data acquisition, data manipulation, state detection, health assessment, prognostic assessment, and advisory generation. The PHM process allows it to go from raw data, usually automatically collected, to the notification of alerts and alarms to the decision-maker to set up suitable reactive and corrective actions, if required.

Considering the PHM process, this section is structured as follows in order to present the overall CBM solution for FDD of cobots: subsection "Proposed framework for FDD of collaborative robots" clears out the requirements for a CBM solution where cobots are the targeted machines and connects these requirements with the PHM process so to define the proposed framework; in subsection "Definition of the hybrid series modelling of AI algorithms supporting the framework", for each identified step of the framework, envisioned AI algorithms fitting the purpose are described.

Overall, the proposed framework aims to be as general as possible, making it customizable to different industrial applications of cobots and providing future users with the possibility to adapt the models/algorithms based on the specific problem setting. This aspect is a founding principle that represents the novelty with respect to the as-is state of the art.

Proposed framework for FDD of collaborative robots

The development of the framework for FDD of collaborative robots originates from some working hypotheses. The hypotheses are hereby listed:

1. the cobots are pre-trained with a set of trajectories that are known while developing the framework and related models/algorithms;
2. once defined, the nominal trajectory could be performed multiple times, but every time, there will be a degree of dissimilarity as the internal control does not find the same (sub)optimal solution in the hyperspace of possible trajectories from a point A to a point B; therefore, each cycle (a performed trajectory) could be different from another one;
3. also related to point 1, in industrial practice it may be that operators train cobots to perform trajectories not predefined, which are performed occasionally; therefore, there will be trajectories that are unknown;
4. the available data are only those coming from the control unit of the cobot, hence no additional sensors need to be installed.

The hypotheses lead to the definition of subsequent requirements:

1. availability of a known set of trajectories resulting from the cobot pre-training;
2. need to deal with a degree of dissimilarity (variability) between the known trajectories;
3. possibility to add new/unknown trajectories;
4. data streams from the control unit, including only sensors present for the cobot control.

The resulting three-step framework is drawn in Fig. 1, where the PHM process is also reported to highlight the adherence of the proposal with respect to ISO 13374-1. The grey-shaded hatching in the upper part identifies the novel and innovative part of the proposal.

The three envisioned steps are described in the reminder:

1. Trajectory clustering aims at grouping trajectories which are similar to each other.
2. Health state identification is performed only for known trajectories as it is possible to identify them as reference behaviour for the health state; in this step, anomaly detection methods are adopted so to identify if an abnormality arose or not (with respect to the known trajectories). In this step, the operator could be informed about the deviation from nominal requirements.
3. Functional failure clustering, whose goal is to group trajectories experiencing the same functional failure; this is applied only to those trajectories for which an anomaly arose, thus those that are unhealthy.

The envisioned solution is not a one-shot and monolithic as resulting from the state of the art. Indeed, it has been thought of as an evolving application that can be configured and learns time by time through proper algorithms. Moreover, if new known trajectories should be added, the framework is able to accommodate this new information and scale up to recognise them. These are potentialities of the solution that are kept in mind while developing the models/algorithms for each of the steps of the framework and impact on the selection of specific AI algorithms, as described in the next subsection.

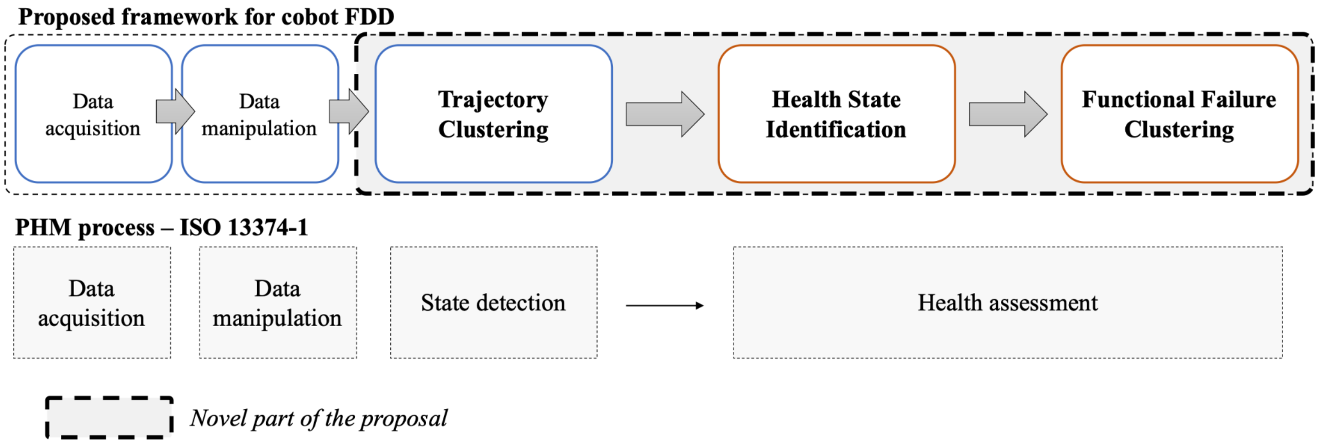

Fig. 1 Proposed framework for FDD for cobots and its adherence to the PHM process

Definition of the hybrid series modelling of AI algorithms supporting the framework

The decision to opt for a hybrid series modelling aims to favour the modularity of the solution. Therefore, each step has its own AI algorithm, whose output becomes the input for the next one. Also, to keep the framework as general as possible, the decision is to identify unsupervised AI algorithms that are not bounded by the presence of labelled data.

- For the trajectory clustering step, a clustering algorithm is exploited, which is an unsupervised machine learning technique allowing the discovery of grouping within data. In this way, it is possible to group together those trajectories that are similar and compare them with the nominal known trajectories.
- The health state identification step consists then in monitoring the asset condition to understand whether the trajectories can be associated with a healthy or unhealthy state. Just data belonging to known trajectories reaches this step because the healthy behaviour of data can be defined only if the trajectory performed by the cobot is known. In particular, anomaly detection is applied, which detects if something new, an anomaly, has happened, assuming that the healthy state of the asset under analysis has shifted from healthy to unhealthy (Pimentel et al., 2014).
- The functional failure clustering step is carried out by means of a clustering algorithm applied to the unhealthy trajectories only. Indeed, each unhealthy trajectory could stem from a specific functional failure and the clustering algorithm is in charge of grouping those trajectories whose underlining functional failure could be potentially the same.

The three-step framework is intentionally based on an unsupervised learning approach, to make it independent from the need to capture labels on the basis of field evidence and/or expert knowledge. In particular, it is worth pointing out that expert knowledge is not considered in the current functioning of the framework unless for its validation.

For the first and third steps, density-based clustering algorithms are envisioned as suitable ones. Indeed, density-based clustering models are defined as unsupervised learning methods which identify a cluster among the data points relying on the idea that a cluster is an area of high-density points separated from the other clusters by low-density points areas. The data points in the separating regions of high-density points are typically considered noise/outliers, and therefore they are not clustered. Instead, in the second step, given the need of comparing time series data, corresponding to the trajectory, the envisioned solution is the application of neural networks (NNs) and, specifically, of the autoencoder. The autoencoder is a reconstruction-based algorithm, namely a specific type of NNs, where the input is the same as the output. The input dimensions are reduced by means of successive compression steps (encoding phase), and then the output is reconstructed starting from the compressed input (decoding phase). For more details on autoencoder, please refer to Arul (2021).

The definition of such families of models defines the framework proposed in Fig. 1 and the related details are reported in Fig. 2. In this figure, each step is specified in terms of inputs, outputs and envisioned model.

The proposed three-step framework must be tested so to assess first if the decomposition in three steps is suitable and secondly if the foreseen AI algorithms are effective in identifying trajectories, health states, and failures. Therefore, the framework is applied to a collaborative robot working in a laboratory environment.

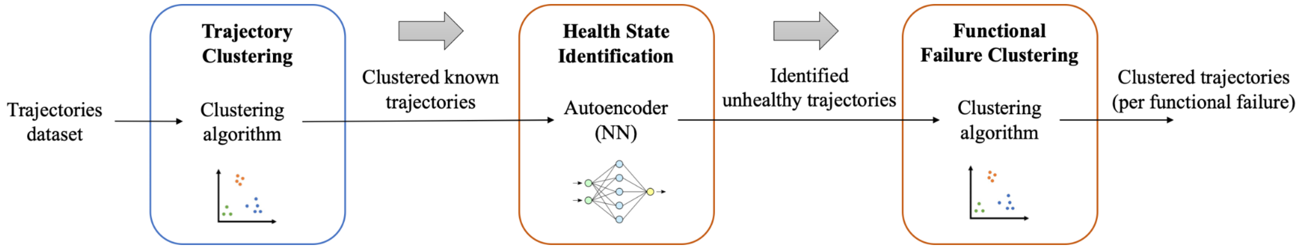

Fig. 2 Details of the proposed framework for FDD for cobots

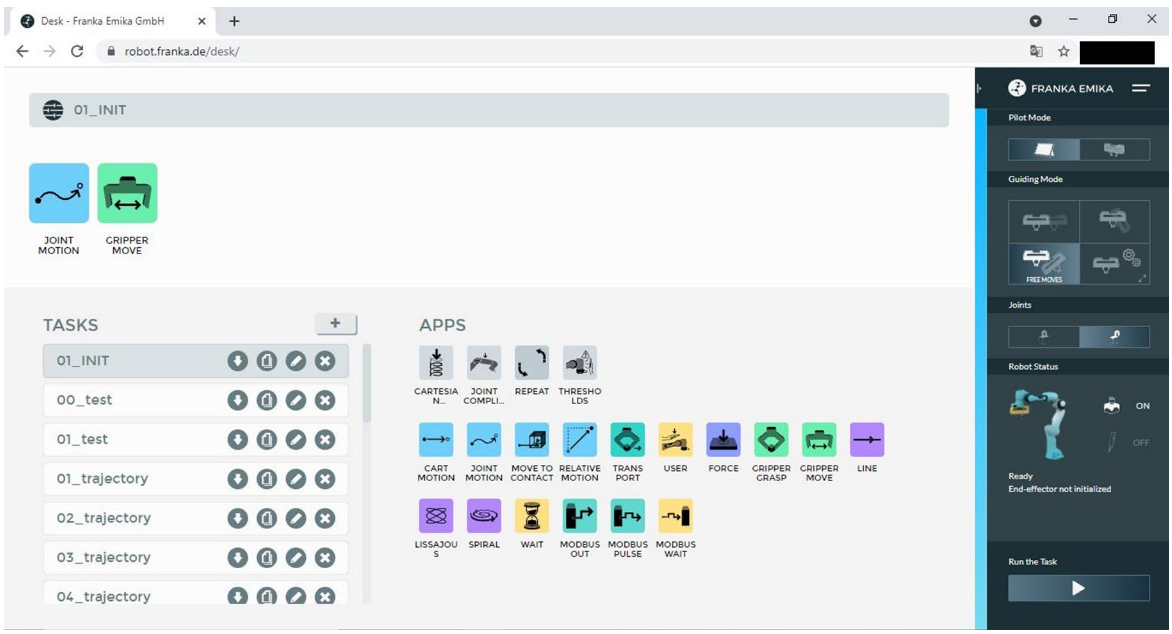

Fig. 3 Web application to define cobot trajectories

Application of the framework

A 7-axis Panda cobot manufactured by Franka Emika was taken into consideration to apply and assess the three-step framework for FDD so as to identify abnormal trajectories and cluster them according to different functional failures. Indeed, from an applied perspective, the framework aids the line operator to be aware of deviations given the high variety the cobot could perform so as to anticipate effects on the product quality, continuity and safety of operations. In this section, the experimental setup and the experimental campaign are described in subsections "Experimental setup" and "Experimental campaign", respectively. Then, the obtained datasets are used to feed the framework. This is described in Sect. "Artificial Intelligence algorithms tuning" given that, for each algorithm, an analysis of required features and hyperparameters definition is performed.

Experimental setup

Each joint of the cobot is equipped with a positional absolute encoder (14-bit resolution), a torque sensor (13-bit resolution) and a brushless DC motor, together with a 1 kHz communication bus.

For the data gathering, the robot device is equipped with Franka Emika's control hardware and software and any device with an internet connection can access the cobot network and cooperate with it via a direct user interface (Fig. 3). The interface allows defining of the trajectories and sets the velocity and force parameters.

Real time communication with the cobot relies on a software implementation platform called Robot Operating System (ROS). *frankaROS* comprehends ROS packages specifically developed for connecting cobots with the entire ROS ecosystem. It integrates *libfranka* into ROS Control and includes URDF (Unified Robot Description Format) models and detailed 3D meshes of both robots and end-effectors, which allows visualization (e.g. RViz) and kinematic simulations. Finally, ROS oversees accessing the

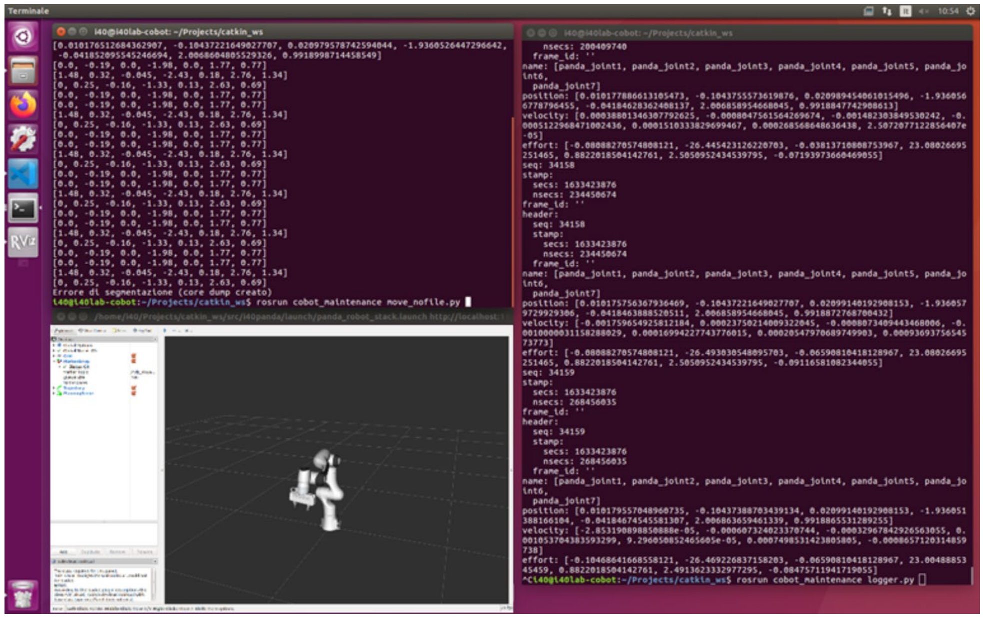

Fig. 4 Computer desktop: the upper-left corner shows the textual interface to plan cobot motion, the lower-left corner shows the cobot movement in real time via RViz, right-hand side shows the shell to collect data, namely position and torque

cobot sensors and actuators, processing their raw information and commanding the robot to execute tasks.

To run the experiment and collect data, the control unit, embedded in the server rack, is connected to the cobot via ethernet cable, while a PC (Personal Computer) is connected to the cobot network wireless. On the PC, the Ubuntu Linux operating system is running, with an i7, 2.9 GHz processor with 8GB RAM. Figure 4 shows the desktop through which the cobot is controlled and monitored in real-time via ROS interface.

Once the cobot and related data collection scripts were defined, the experimental campaign could be performed, as described in subsection "Experimental campaign".

Experimental campaign

Before running the experiments, it is important to clearly set the working assumptions:

- *One pose for each trajectory is assumed to be known*. As pointed out by the work of Wescoat et al., (2021), the precise identification of the starting point guarantees higher performance of models. As such, considering the industrial practice, it is possible to assume that there is a unique rest pose for all the possible trajectories that allows for an easy characterisation of the data series, i.e., the identification of all cycles.
- *There is not a slow degradation process, but a step change in the behaviour*. The FDD capabilities of the solution are tested by inducing sudden functional failures. Therefore, possible slow degradation due to, for example, outer race damage in a rolling bearing that increases through time inducing the highest level of vibration as time passes, is not considered.

In the concluding remarks of this work, the working assumptions will be discussed in their limits to inform future works.

Design of the experiments

Given these terms and assumptions, the experimental campaign could be properly designed. The main choice to be made is regarding the types of functional failure that lead to a change in the trajectory and how to induce them. From the literature analysis performed, an overload is widely used to realise experiments that cause deviations in the cobot trajectory. Additionally, to the overload, also excessive friction is introduced, which could affect the performed trajectory as it simulates a degradation of mechanical components, e.g., a bearing. Lastly, given the variability of the situation on the shop floor, the experiment does consider unpredictable factors that lead to stochastic changes in the trajectory.

Given these premises, the following experimental design choices are considered:

- 3 functional failures, related to the above considerations of deviation from nominal trajectories, namely:
  - *Overload*. A mass of 500 g was positioned on the end effector throughout a dumbbell fixed to the extremes of the cobot gripping hand by means of a couple of cable ties (Fig. 5.a).
  - *Excessive friction*. Two rubber bands were positioned in correspondence with the 2nd joint of the cobot (Fig. 5.b). To connect each of them to the cobot

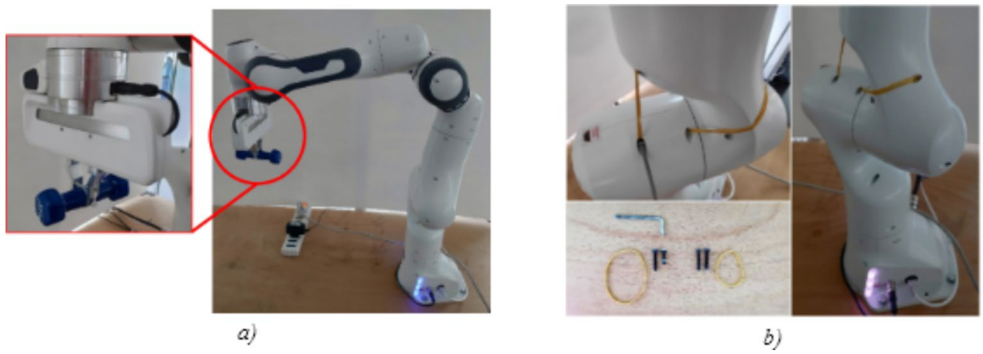

Fig. 5 Pictures of functional failures: overload on the end effector (a) and excessive friction of the second joint (b)

Table 3 Design of the experiments for the FDD framework

Health state		Trajectory			
		1	2	3	4
Healthy state	Training set	1.000	1.000	1.000	0 (unknown)
	Testing set	200	200	200	200
Unhealthy state	Overload	50	50	50	0
	Friction	0	0	50	0
	Ext. factors	50	0	50	0

body, two screws were screwed off then, the band was rolled onto the screws, and they were screwed back in the original position. This functional failure could represent a degradation of mechanical components (e.g., bearings) or an overheating of the joints (Olsson et al., 1998; Bittencourt & Gunnarsson, 2012).

  - *Unpredictable factors.* A random coefficient varying in the continuous range [− 0.02, + 0.02] was added for each joint to the target positions during the cobot motion to make the executed trajectory vary. This failure could represent problems with the robot controller or the positional encoders, defects in the robot-base securing system, or abnormal vibrations propagating from any faulty component along the robot's structure and frame.

- 4 different trajectories are reproduced to represent typical pick-and-place cobot operations. Also, 1 out of 4 trajectories is assumed to be unknown.

Correspondingly, the creation of the dataset passes through the identification of 13 work sequences, which are collections of consecutive cycles that all belong to the same trajectory and the same health state. The 13 work sequences are reported in Table 3. Each cell in Table 3 represents a work sequence (0 means no work sequence realised for that combination of trajectory and healthy/unhealthy state), and it is shown the number of consecutive cycles replicated for that sequence.

Hence, the first dataset refers to the healthy state and it is divided into training and testing sets, for which 1.000 and 200 replicas (i.e. consecutive cycles replicated for each work sequence) are performed, respectively. The fourth trajectory is assumed to be unknown, so there is no training, but only testing.

Then, the second dataset is realised for the unhealthy state, considering the induced functional failures. Specifically, for each of them, 50 replicas are performed, assuming that these conditions will be experienced sporadically by the cobot with respect to the healthy state. Each of the unhealthy states (50 replicas) is joined with the related dataset for the healthy state testing set. In that way, for each functional failure, the dataset will contain 200 cycles of the healthy state and 50 cycles of the unhealthy state. Finally, since the fourth trajectory is still the one unknown, no unhealthy data are generated.

The number of cycles has been chosen to ensure a trustworthy implementation of the proposed framework and a reliable estimation of the performances of the developed models/algorithms.

For each trajectory, three variables are collected, namely the angular position θ in radians, the angular speed ω in radians per second and the torque C in Newton. Indeed, the angular speed is not directly measured, but it is automatically calculated by means of the embedded control as expressed by Eq. 1.

$$\omega = \frac{\Delta\theta}{\Delta t} = 2\pi f \tag{1}$$

where:

- ω is the angular speed.
- θ is the angular position.
- t is the time.
- f is the frequency.

As a matter of example, Fig. 6 reports the gathered data for the overload functional failure, Fig. 7 for the excessive friction functional failure and, finally, Fig. 8 for the

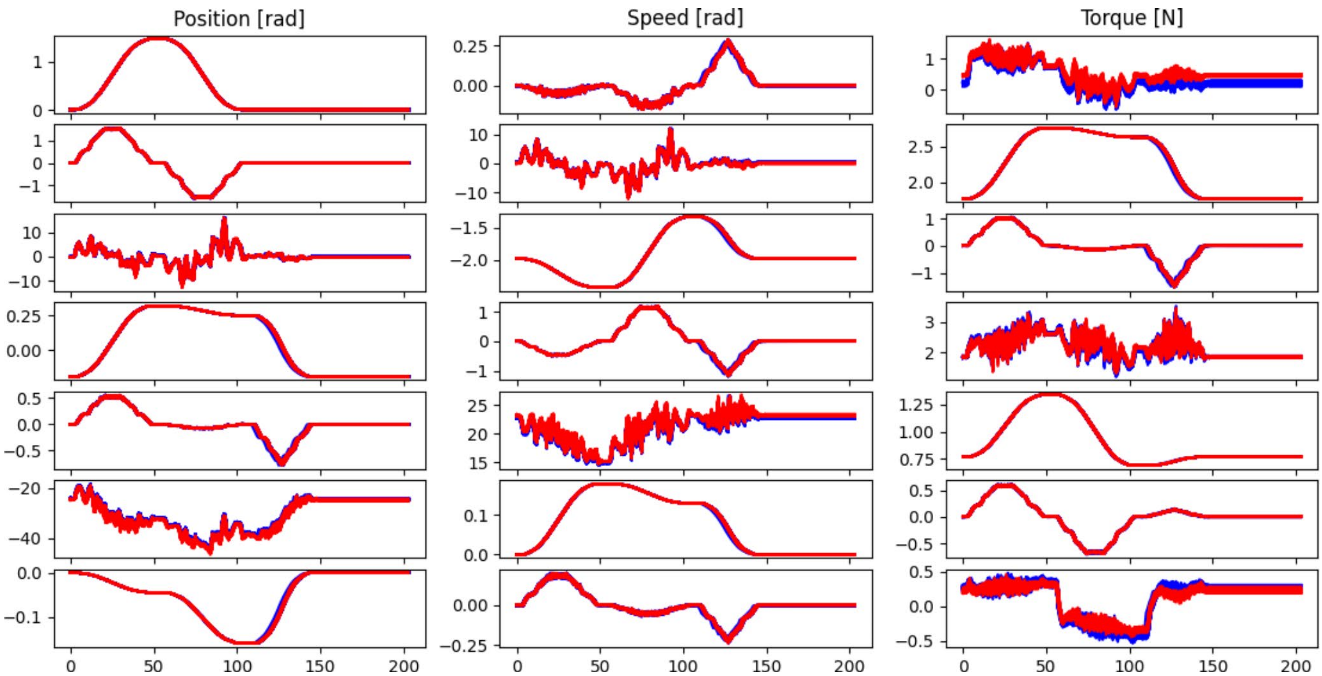

Fig. 6 Overload-related dataset

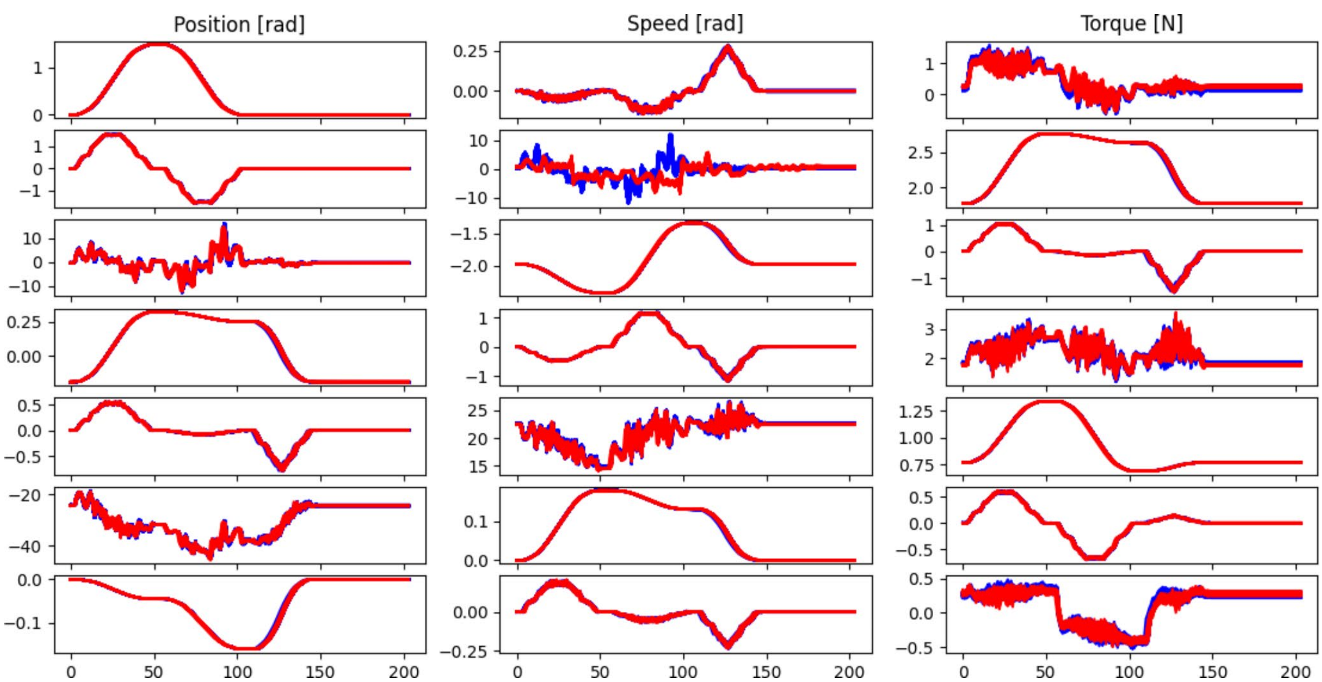

Fig. 7 Excessive friction-related dataset

unpredictable factors functional failure through a random coefficient, all for trajectory 3, which has all the possible combinations of healthy and unhealthy states. In all figures (from Figs. 6, 7 and 8), the 7 subplots, one per collected variable, represent each joint of the 7-axis cobot, for a total of 21 plots. The blue data points represent the healthy state, while the red points the unhealthy state. Indeed, hereinafter trajectory 3 is considered as a reference as it is the only one entailing all possible functional failures.

Once set up the experiment and all the datasets were available, the framework could be implemented, tested and assessed, as presented in Sect. "Artificial Intelligence algorithms tuning".

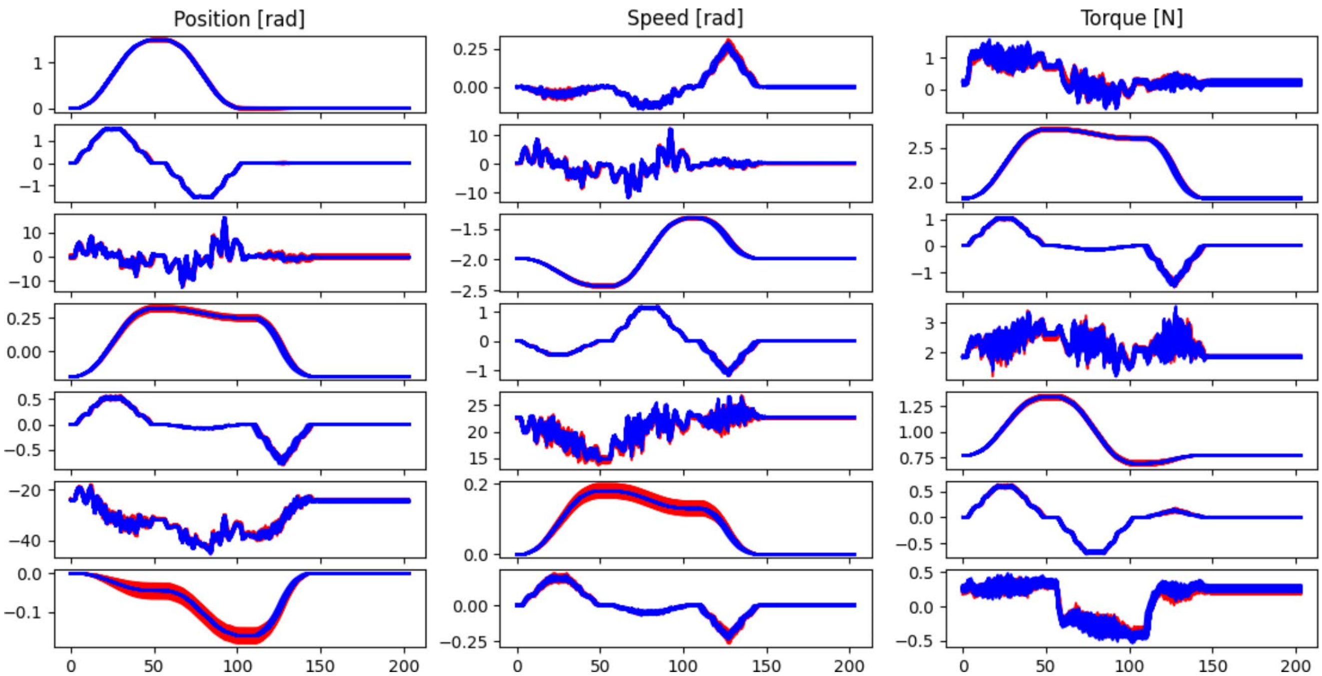

Fig. 8 Unpredictable factors-related dataset

Artificial Intelligence algorithms tuning

A preparatory step is necessary before the application of the AI algorithms, to characterise each cycle by means of the rest pose knowledge in terms of angular position, speed, and torque. Also, concerning the three AI algorithms defined in the framework, the dataset should be made homogeneous, especially for the autoencoders, whose columns in the input dataset must have the same length. Hence, the reference length of the cycle is set to be the longest one, thus, in the case of this experiment equal to 204 timestamps; then, the last observation carried forward procedure is adopted. By selecting the longest cycle, there is no risk to truncate previous observations before the completion of the entire set of trajectories. Also, given that cycles are separated by identifying the rest pose, carrying it forward (as the rest pose is always the last observation for each cycle) does not affect the capability to recognise the cycles themselves.

Once the dataset is prepared, the AI algorithms are tuned so to test the framework, as described in the remainder, starting with the trajectory clustering, then moving to the health state identification and, finally, ending with the functional failure clustering.

Trajectory clustering AI algorithm

The trajectory clustering step oversees grouping the trajectories the cobot performs. The selected AI algorithm is a density-based clustering, namely OPTICS (Ordering Points To Identify the Clustering Structure). This is an extension of DBSCAN that tries to overcome its main weakness, that is the difficulty of detecting clusters in datasets with varying density (Kanagala & Jaya Rama Krishnaiah, 2016). Also, OPTICS does require only the minimum points for each cluster to be defined, rather than the minimum points and minimum distance between points as for DBSCAN. Interested readers could refer to (Han et al., 2022) for insights into how these algorithms work and their differences.

The other challenge to be faced refers to the number and type of features to be extracted and how to manage the high computational power required if numerous trajectories are considered. For what concerns the features, only time-domain features are considered in order to identify the trajectories, namely: sum of the values (it is the sum of all the samples of the time series), median, mean, length (which is equal for each cycle as explained before), standard deviation, variance, Root Mean Square (RMS, which is the arithmetic mean of the squares of all the samples of the time series), maximum and minimum values. Regarding instead the possible high computational power required if the number of cycles is high, the adopted method is:

1. to launch OPTICS on the training dataset or, in general, on the dataset whose cycles are healthy;
2. to save an appropriate number of core points for each cluster resulting from the application of OPTICS;
3. to launch OPTICS on the testing dataset plus the core points.

Table 4 Dataset for testing the OPTICS algorithm for trajectory clustering

Health state		Trajectory			
		1	2	3	4
Healthy state	Training set	1.000	1.000	1.000	0 (unknown)
	Testing set	210	210	210	200
Unhealthy state	Overload	50	50	50	0

The core points are points laying in the high-density area of each cluster so that they can characterise the cluster itself, without the need to rerun OPTICS over the entire training dataset every time. In this way, the computational power is saved and limited to the clustering of only the testing dataset. The only parameter to be set is the number of core points to be saved. In this case, after some trials, 10 core points are considered and, as shown after, this allows to reach high-quality results.

The application of OPTICS shows that the algorithm can identify the known trajectories.

OPTICS algorithm applied to trajectory clustering

To test OPTICS, the gathered data are used. To see if there is any difference, in terms of performance, between testing different induced functional failures, OPTICS is applied one functional failure at a time. For the sake of brevity, in Table 4, the dataset used to test the OPTICS in case of overload is reported as it is the one induced for all three trajectories.

Firstly, the training dataset is used to define the 10 core reference points to be used to cluster the 3 trajectories. The results from which the core points are extracted is shown in Fig. 9.

Once the three known trajectories are characterised, it is possible to use the testing dataset joined with the induced overload functional failure dataset to test if OPTICS can correctly find the three original trajectories. The result reported in Fig. 10 shows that, for the case of the overload, the three known trajectories are correctly identified, plus one that is unknown.

Once the trajectories are correctly grouped, the dataset passes to the following health state identification step; to this end, it is worth mentioning that only the data related to the clustered known trajectories are passed.

Health state identification AI algorithm

According to the framework in Fig. 2, unknown trajectories will not pass to this second step, as they are grouped together by the clustering algorithm. Then, the health state identification step aims at identifying if there is any anomalous trajectory the cobot is performing.

An anomalous trajectory is a trajectory whose behaviour is significantly different from the reference one. Given that an autoencoder is selected as the model in charge of this step, firstly, in the training phase, it is required to define the bottleneck identity function. Then, in the testing phase, the cycles are passed to the encoder that, by applying the specified bottleneck identity function, tries to reconstruct the input. Given that in the testing also unhealthy cycles are passed, then the reconstruction error is higher. Therefore, by

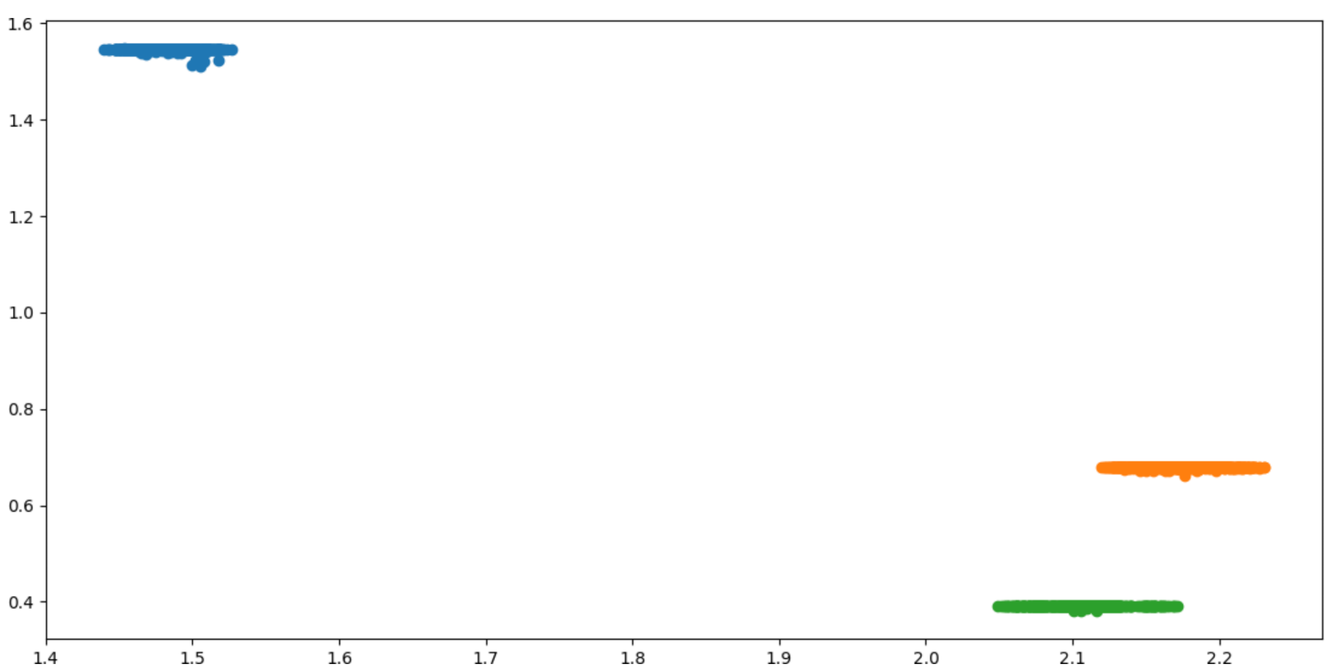

Fig. 9 OPTICS output: the three trajectories are correctly identified

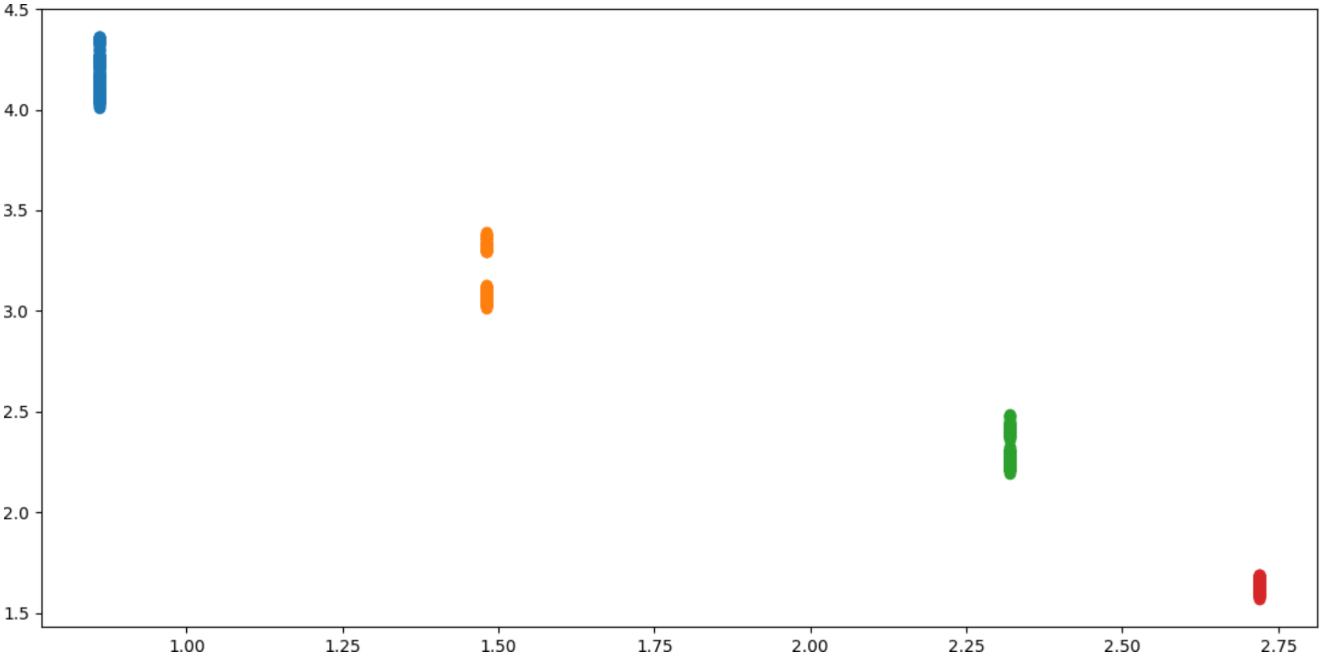

Fig. 10 OPTICS output for the overload case: the three trajectories are correctly grouped, plus one unknown trajectory (red one)

comparing the reconstruction error with a predefined limit (defined as a threshold), it is possible to label the cycle as healthy (below threshold) or unhealthy (above threshold).

The autoencoder options and hyperparameters are set up following a waterfall approach to the training dataset. Namely, the waterfall approach consists of the fine-tuning of one hyperparameter at a time. Firstly, one hyperparameter is optimised, keeping all the others fixed according to literature-based values. Then, once the optimal value for the first hyperparameter is found, a second one is selected and analysed, with all the others coming from literature, and so on, until the identification of optimal values for all hyperparameters.

First and foremost, the activation function has to be selected; amongst the possible options, the ReLU (Rectified Linear Unit) is the most commonly used and proven to outperform most of the alternatives, hence it is considered also in this work (Ding et al., 2018). Then, the loss function and optimiser have to be defined. Indeed, no common practice is established (Qian et al., 2022), and it is decided to adopt the MSE (Mean Square Error) as the loss function and the Adam optimiser.

The final tuning of the autoencoder hyperparameters is done by minimising the reconstruction error and leads to the following results:

- Zero padding method.
- Stride length equal to 2.
- Filters size ranges from 16 to 32 depending on the type of layer.

Table 5 Dataset for testing the autoencoder for health state identification.

Health state		Trajectory 1	2	3
Healthy state	Training set	1.000	1.000	1.000
	Testing set	200	200	200
Unhealthy state	Overload	50	50	50
	Friction	0	0	50
	Ext. factors	50	0	50

- Kernel size equal to 11×11.
- Dropout rate is equal to 0.1.
- Batch size is equal to 128 samples.
- Initial learning rate equal to 0.001 (Adam optimiser will automatically adjust it through the training process of the autoencoder).

Overall, the adopted algorithm is a convolutional reconstruction encoder with four layers, one encoding layer, one bottleneck layer and two decoding layers, plus the input and output layers. The dropout layers are added before the encoding layer and after the last decoding layer of the network. It is worth underlining that this type of autoencoder results from the specific dataset used in this research work.

Autoencoder applied to health state identification

Table 5 presents the dataset pertaining to the known trajectories, i.e., the first, second, and third ones.

Given that each autoencoder can model the behaviour of one variable at a time, 21 autoencoders for each trajectory

Table 6 Dataset for testing the autoencoder applied to health state identification

Induced functional failure	Autoencoder error	Trajectory		
		1	2	3
Overload	False alarms	3 (1.2%)	2 (0.8%)	1 (0.4%)
	Missed detections	0	0	0
Excessive friction	False alarms	3 (1.2%)	2 (0.8%)	1 (0.4%)
	Missed detections	0	0	0
Unpredictable factors	False alarms	3 (1.2%)	2 (0.8%)	1 (0.4%)
	Missed detections	0	0	1 (0.4%)

are required, as 21 features were extracted. Overall, 63 (21 autoencoders times 3 trajectories) autoencoders run to identify the health state of the cobot. If the reconstruction error of at least one autoencoder exceeds the threshold, then it is concluded that the cobot is in unhealthy state. To evaluate the threshold, the following method is adopted:

1. to evaluate the reconstruction error for each cycle of the training dataset;
2. to compute the MARE (Mean Absolute Reconstruction Error) for each trajectory;
3. to define the threshold between healthy and unhealthy as the maximum of the MARE values, so that those cycle whose reconstruction error is under the threshold could be labelled as healthy.

This method is needed as, even though the trajectories within the training dataset are performed with the cobot in the healthy state, the intrinsic trajectory optimisation of the cobot could lead to slightly different values of the variables in each cycle. Over 1.000 training trajectories, it is assumed that the identified threshold is properly defined.

The goodness of the models is measured in terms of false alarms and missed detections, as reported in Table 6, both in absolute and relative terms (over 250 cycles per experiment).

It is worth seeing that the results are relatively good for all three trajectories with no missed detection for the overload and the excessive friction failures, while the only missed detection is the one due to unpredictable factors. This may have been foreseen as the random coefficient applied to the trajectory acts in a stochastic way, while the overload and excessive friction act almost in the same way.

Functional failure clustering AI algorithm

Once the anomalous conditions were identified, the functional failure could be also grouped. In this phase, the input dataset is the one of only predicted anomalous cycles obtained from the health state identification step. The adopted clustering algorithm is again OPTICS due to its various advantages over DBSCAN as already pointed out in subsection "Trajectory clustering AI algorithm".

Table 7 Dataset and results for OPTICS applied to functional failure clustering

Induced functional failure		Trajectory		
		1	2	3
Input dataset	Overload	53	52	51
	Friction	3	2	51
	Ext. factors	53	2	50
Output dataset	Noise	6	0	17
	Overload	45	56	48
	Friction	58	0	45
	Ext. factors	0	0	42

Despite the algorithm being the same, the set of features in this third step of the framework is much larger. Specifically, 776 features were extracted, belonging to both time and frequency domains, such as mean, standard deviation, absolute energy, kurtosis, skewness, first location of maximum, distribution quantiles, autocorrelation at various lags, peaks, Fast Fourier Transform (FFT) coefficients, Fourier entropy and many others. This aims at providing a large set of information to empower the diagnostics capability of the operator.

OPTICS applied to the functional failure clustering

The input dataset is represented by the predicted anomalous cycles identified by the autoencoder in the second step of the framework; this is reported in the upper part of Table 7. The lower part of the same table shows the results of OPTICS. Indeed, it is worth noticing that OPTICS also identifies noisy data points; hence, in addition to the induced functional failure, also the "noise" class was added.

As a matter of example, Fig. 11 reports the graphical outputs for the third trajectory. Apart from some random noise identified by blue points, the three induced functional failures are correctly grouped: orange points represent the overload, green points the excessive friction, while the red points highlight the unpredictable factors.

In the next Sect. "Results and discussion", performance measures of the framework assessment are shown, alongside a discussion on the industrial relevance of the obtained results.

Results and discussion

Commonly known performance measures are now used to better describe the performance of the algorithms, namely: sensitivity (recall), specificity, precision, negative predicted

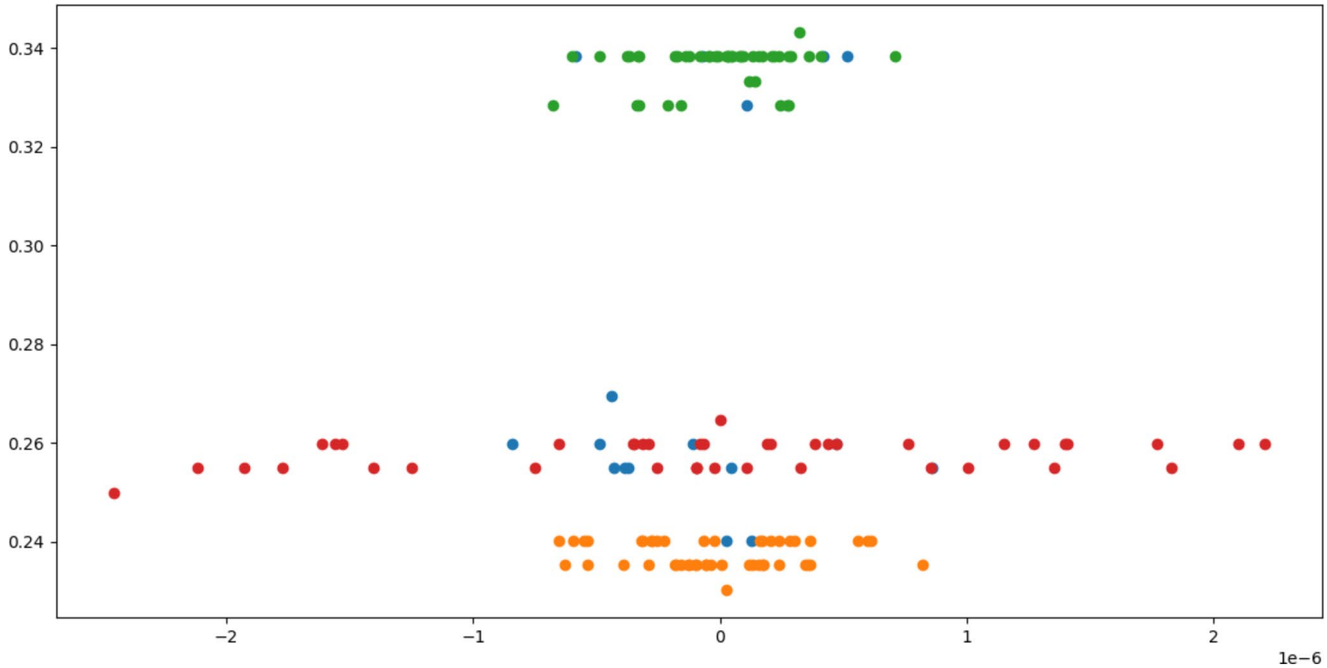

Fig. 11 OPTICS output for functional failure clustering for the third trajectory

value, accuracy and F1 score. These measures are derived from the confusion matrix (Hossin & Sulaiman, 2015) which allows for determining the performance of classifiers, showing the numbers of the true positive (TP), false negative (FN), false positive (FP), and true negative (TN). Specifically, in this work, it has been chosen to use classification performance metrics instead of the clustering ones as they could provide more evidence on the correct functioning of the framework. Given this choice, the human expert is involved in manually labelling the experimented trajectories with or without functional failure, and of which type, for the sole purpose of validation of the algorithms.

Noting that, in the proposed framework, the algorithms are organised in sequence, it is worth remarking that the performance of both the algorithms at the second and third steps is influenced by their predecessors since the input data of each algorithm is the output of the previous one. Also, all performance measures are influenced by the sample size, which reduces from the first step onwards: the data belonging to any unknown trajectory are discarded during the trajectory clustering step, and then the amount of data is further reduced by excluding healthy cycles after the health state identification step. This should be considered when looking at the performance of the entire three-step framework.

For what concerns the first step, i.e., trajectory clustering, the accuracy of the OPTICS performance for all the trajectories and the induced functional failures is 100%. This value is justified by the fact that, in this experimental campaign, the four trajectories are really different from each other.

Table 8 Autoencoder performance for the overload functional failure

		Predicted class	
		Unhealthy	Healthy
Health state	**Unhealthy**	*TP*	*FN*
		150	*0*
	Healthy	*FP*	*TN*
		6	*594*

Considering the second step, i.e., health state identification, the performance is summarized in Table 8, always for the overload.

The weighted average performance measures considering also the other functional failures are expressed in Table 9.

Finally, for the functional failure clustering step, the performance measures are reported in Table 10, by pointing out the number of trajectories correctly grouped together and not per functional failure. Those trajectories not correctly grouped are labelled as noise by the density-based clustering algorithm.

The first conclusion is that the OPTICS algorithm never classifies any kind of functional failure as another one, even though some trajectories were labelled as noise. It is worth

Table 9 Autoencoder performance measure for health state identification step

Sensitivity (recall)	Specificity	Precision	Negative predicted value	Accuracy	F1 score
99.67%	99%	94.32%	99.94%	99.10%	0.97

Table 10 OPTICS performance for the third trajectory per functional failure clustering

Induced functional failure	Correctly grouped trajectories	Missed grouped trajectories
Overload	48	3
Excessive friction	45	6
Unpredictable factors	42	8

saying that, amongst those trajectories labelled as noise, there are 3 trajectories that were false alarms as a result of applying the autoencoder. Hence, only 14 trajectories could be considered as real errors introduced by the clustering algorithm. Overall, also for the first and second trajectories, the results are satisfactory.

To conclude the performance assessment, it is also worth pointing out the computational performance of the framework. Overall, apart from the training of the three AI algorithms, it takes 189 s to run the three steps, considering all the above tests, on a PC equipped with an Intel Core i7, 2.9 GHz and 8 GB RAM.

Discussion

In light of the results shown above, it can be stated that the goal of this research is reached, that is, the development of a framework for the FDD of cobots. Indeed, the performance assessment of the proposed framework and the related algorithms shows an effective capability to cluster the trajectories, identify the relative healthiness, and, in case of anomalous cycles, the functional failure the cobot is experiencing.

Specifically, the framework was tested against three main functional failures, namely, an overload at the end effector, excessive friction in one joint and random variations due to unpredictable factors. This last one introduced some errors (see false alarms and missed detection in Table 6 and missed grouped trajectories in Table 10) given that the changes to trajectories are aleatory, whilst for the first two functional failures, the effect on trajectories is almost repetitive between cycles.

The capability to first recognise known from unknown trajectories, then to distinguish a healthy from an unhealthy cycle and finally discern between functional failures is enabled by three AI algorithms: two versions of the density-based OPTICS clustering applied first to trajectory clustering and then to functional failure clustering (first and third steps of the proposed framework), and an autoencoder applied to make the health state identification. All of them represent unsupervised machine learning as one of the goals the framework pursues is independency from an extant knowledge of cobot behaviour. As such, the framework works even when no labels are present in terms of trajectories and/or functional failure. Of course, it is evident that the availability of such kind of information is relevant since it makes it possible to precisely identify which task the cobot is performing or which functional failure is experiencing, further triggering a proactive maintenance action to guarantee operational continuity. This will be part of future work as discussed afterwards in the concluding section.

From an industrial application point of view, it is worth pointing out the core part of the proposed FDD framework, which is its second step, i.e. the health state identification. The capability of the autoencoder to identify deviation of the operation, corresponding to a possible shift between the healthy to an unhealthy state of the cobot, allows notifying the operator. Then, the maintenance intervention can be further triggered, in which detailed diagnostics activities are required from technicians, eventually informed by the clustering of functional failures gained in the third step of the framework.

Finally, it is worth discussing the expected benefits of the proposed solution in industrial practice:

- The proposed framework is adaptable to situations where collaborative robots are newly installed as well as for already working cobots as it only relies on a few thousand cycles.
- A second benefit of the framework is that it relies only on built-in sensors without the need to add additional monitoring systems; this could have a strong positive advantage as the ramp-up cost for this solution is relatively low.
- Thirdly, the framework is automatic and requires almost no updates: once the three AI algorithms have been trained, the real-time testing allows an understanding if the cobot is in a healthy state or not. In this regard, it is also relevant to note that the execution is fast, therefore it is possible to quickly analyse data and consequently plan maintenance interventions during the cobot runtime.
- Lastly, as an original choice, the proposed framework is designed based on modularity. This gives interested companies the possibility both (i) to decide to stop at the trajectory clustering step, or at the health state identification step depending on the purpose of the monitoring system, or embrace the entire three steps, and (ii) to configure a step with a different AI algorithm in case advantages are evident after experiments are run and the performance assessment is obtained. The modularity is also an enabling characteristic leading to the possibility to scale up the results: the AI algorithms could accommodate additional data at specific steps without the need to rebuild everything from scratch, in regard to new trajectories (passing from unknown to known trajectories,

in the first step of the framework) or new functional failures (in the third step of the framework).

Overall, the proposed framework, with the related AI algorithms, can be seen as the first solution aiming at providing CBM to collaborative robots. The modular design of the framework, in its three steps, may envision further improvements in the specific industrial application, and thus a potential for customization to given problem settings.

Conclusion

The current research proposes a framework for FDD of collaborative robots through hybrid series modelling of AI algorithms that aim at notifying anomalous cobot trajectories to the operator. The framework is tested on a 7-axis collaborative robot and three functional failures are induced, by means of an overload to the end effector, excessive friction in one joint and random variations due to unpredictable factors. The need for such a proposal stems from the identified gaps in the scientific literature and the growing adoption of cobots in the industry to improve operational flexibility. Currently, there are neither guidelines nor predefined solutions for cobot FDD in the published research works, while production systems are speedily evolving towards smart factories inclusive of new robotic applications as cobots; this requires a high-level integration and monitoring of all machine types at the shop floor for robust decision-making.

The proposed framework for cobot FDD consists of three consecutive steps: trajectory clustering, health state identification, and functional failure clustering. The decision to opt for a three-step approach with respect to one-shot and monolithic solutions is twofold: firstly, from each of the three mentioned steps it is possible to extract meaningful information and the data flowing throughout the framework is more controllable and manageable; then, since there are no guidelines to develop CBM solutions for cobots, the only reference is the PHM, whose process is organised in consecutive, yet separated steps. In addition, the solution is subsequently customizable according to company purposes, as not all the steps are required to be implemented, but only up to those deemed useful; moreover, a step could be configured with a different AI algorithm.

The framework is meant as a first move and is characterized by two major limits. Firstly, the framework habilitates the operator, working with the cobot, who is notified by anomalous cobot trajectories and thus can activate a call for a condition investigation at the next level. The next level should provide a deeper capability for diagnostics to move forward, i.e. from the isolation of the functional failure until the isolation of the faulty item/s and its/their failures, this requires further development. Secondly, another development should be to enable the human-in-the-loop for data analysis for collaborative robots, as domain knowledge by experts is not actually used in the functioning of the framework. Indeed, such expert knowledge and other field shreds of evidence could be used to label data beforehand, namely, trajectories and functional failures. In this way, AI algorithms would return more interpretable results that may trigger maintenance intervention, augmented by information about the occurring failures.

Other limitations this work suffers from, concern the experimental part, and are related to the abundance of tested trajectories and simulated functional failures. Namely, the number of trajectories may be higher, leading to a risk of less accuracy in the trajectory clustering step; given the consecutiveness of the steps of the framework, this may have an important impact on the final result. Also, the number of tested functional failures is relatively small; especially, the excessive friction could be experienced by other joints, even simultaneously.

Overall, the main future research has to tackle the main limitations the current proposal has, hence related to habilitate second-level diagnostic activity to identify the specific failures and the introduction of human support along the data analysis process for labelling activities to introduce expert knowledge. These will enable better-informed decision-making processes targeting erroneous trajectories as well as failures currently occurring. Envisioned additional future works related to the adopted algorithms: more effort should be put so to testing and comparing sets of algorithms to identify best-in-class able to deal with the gradual degradation of components for early identification.

Additional future works should extend the current proposal by (i) developing solutions to automatically identify a common pose between the trajectories to automate the characterization of cycles; (ii) analysing how many and which features are needed in order to characterise each trajectory, so as to optimize the feature engineering phase; (iii) inducing gradually evolving failures so to experience slow degradation processes and not only sudden failures as step changes; (iv) going forward in PHM process, looking for prognostic capabilities.

From an industrial application viewpoint, the scale-up of the solution may bring limits in the computational power. Hence, it is worth pointing out that the identified AI algorithms as well as their organisation in hybrid series modelling should be better tested and eventually improved according to the available computational resources and related computing architecture.

In the long run, it is advisable that the possibility to monitor, diagnose and predict the health state of collaborative robots, together with all other machine types on the shop floor, could empower the CPS on which smart factories are built. In this way, integrated decision-making based on thorough management of assets is fostered. Especially, predictability characteristics should be pushed forward. In this way, smart factories could be fully integrated, all their machines, cobots and other equipment connected and monitored, and equipped with the needed capabilities (i.e., diagnosability and predictability) to enable an optimised and sustainable performance, guaranteeing human safety, product quality and operational continuity.

Funding Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement.

Declarations The authors do not have to disclose any competing interests.

References

Aliev, K., & Antonelli, D. (2021). Proposal of a monitoring system for collaborative robots to predict outages and to assess reliability factors exploiting machine learning. *Applied Sciences*, *11*(4), 1621.

Arul, V. H. (2021). 5—Deep learning methods for data classification. In D. Binu & B. R. Rajakumar (Eds.), *Artificial Intelligence in Data Mining* (pp. 87–108). Academic Press. https://doi.org/10.1016/B978-0-12-820601-0.00001-X

Bagheri, B., Yang, S., Kao, H. A., & Lee, J. (2015). Cyber-physical Systems Architecture for Self-Aware Machines in Industry 4.0 Environment. *IFAC-PapersOnLine*, *48*(3), 1622–1627. https://doi.org/10.1016/J.IFACOL.2015.06.318.

Bagheri, E., De Winter, J., & Vanderborght, B. (2022). Transparent Interaction Based Learning for Human-Robot Collaboration.*Frontiers in Robotics and AI*, *9*.

Bittencourt, A. C., & Gunnarsson, S. (2012). Static friction in a robot joint—Modeling and identification of load and temperature effects.*Journal of Dynamic Systems, Measurement, and Control*, *134*(5).

Bokrantz, J., Skoogh, A., Berlin, C., Wuest, T., & Stahre, J. (2020). Smart maintenance: a research agenda for industrial maintenance management. *International Journal of Production Economics*, *224*, 107547. https://doi.org/10.1016/j.ijpe.2019.107547.

Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, *80*(4), 571–583. https://doi.org/10.1016/j.jss.2006.07.009.

Ding, B., Qian, H., & Zhou, J. (2018). Activation functions and their characteristics in deep neural networks. *2018 Chinese Control And Decision Conference (CCDC)*, 1836–1841. https://doi.org/10.1109/CCDC.2018.8407425

Djuric, A. M., Urbanic, R. J., & Rickli, J. L. (2016). A Framework for Collaborative Robot (CoBot) Integration in Advanced Manufacturing Systems. *SAE International Journal of Materials and Manufacturing*, *9*(2), 457–464.

Donadio, F., Frejaville, J., Larnier, S., & Vetault, S. (2018). Artificial intelligence and collaborative robot to improve airport operations. *Online engineering & internet of things* (pp. 973–986). Springer.

El Zaatari, S., Marei, M., Li, W., & Usman, Z. (2019). Cobot programming for collaborative industrial tasks: an overview. *Robotics and Autonomous Systems*, *116*, 162–180. https://doi.org/10.1016/j.robot.2019.03.003.

Faccio, M., Granata, I., Menini, A., Milanese, M., Rossato, C., Bottin, M., Minto, R., Pluchino, P., Gamberini, L., Boschetti, G., & Rosati, G. (2022). Human factors in cobot era: a review of modern production systems features. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-022-01953-w.

Forcina, A., Introna, V., & Silvestri, A. (2021). Enabling technology for maintenance in a smart factory: a literature review. *Procedia Computer Science*, *180*, 430–435. https://doi.org/10.1016/j.procs.2021.01.259.

Guillén, A. J., Crespo, A., Macchi, M., & Gómez, J. (2016). On the role of Prognostics and Health Management in advanced maintenance systems. *Production Planning and Control*, *27*(12), 991–1004. https://doi.org/10.1080/09537287.2016.1171920.

Han, J., Pei, J., & Tong, H. (2022). *Data mining: concepts and techniques*. Morgan kaufmann.

Hashemi-Petroodi, S. E., Thevenin, S., Kovalev, S., & Dolgui, A. (2020). Operations management issues in design and control of hybrid human-robot collaborative manufacturing systems: a survey. *Annual Reviews in Control*, *49*, 264–276. https://doi.org/10.1016/j.arcontrol.2020.04.009.

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), 1.

Ingemarsdotter, E., Kambanou, M. L., Jamsin, E., Sakao, T., & Balkenende, R. (2021). Challenges and solutions in condition-based maintenance implementation—A multiple case study. *Journal of Cleaner Production*, *296*, 126420. https://doi.org/10.1016/j.jclepro.2021.126420.

Ismail, B. I., Khalid, M. F., Kandan, R., Ahmad, H., Mydin, M. N. M., & Hoe, O. H. (2020). *Cobot Fleet Management System Using Cloud and Edge Computing*. 1–5.

Kahouadji, M., Lakhal, O., Yang, X., Belarouci, A., & Merzouki, R. (2021). *System of Robotic Systems for Crack Predictive Maintenance*. 197–202.

Kanagala, H. K., & Jaya Rama Krishnaiah, V. V. (2016). A comparative study of K-Means, DBSCAN and OPTICS. *2016 International Conference on Computer Communication and Informatics (ICCCI)*, 1–6. https://doi.org/10.1109/ICCCI.2016.7479923

Khan, S., Farnsworth, M., McWilliam, R., & Erkoyuncu, J. (2020). On the requirements of digital twin-driven autonomous maintenance. *Annual Reviews in Control*, *50*, 13–28. https://doi.org/10.1016/j.arcontrol.2020.08.003.

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering—A systematic literature review. *Information and Software Technology*, *51*(1), 7–15. https://doi.org/10.1016/j.infsof.2008.09.009.

Koch, P. J., van Amstel, M. K., Dębska, P., Thormann, M. A., Tetzlaff, A. J., Bøgh, S., & Chrysostomou, D. (2017). A skill-based robot co-worker for industrial maintenance tasks. *Procedia Manufacturing*, *11*, 83–90.

Lee, J., Ghaffari, M., & Elmeligy, S. (2011). Self-maintenance and engineering immune systems: towards smarter machines and manufacturing systems. *Annual Reviews in Control*, *35*(1), 111–122. https://doi.org/10.1016/j.arcontrol.2011.03.007.

Márquez, A. C. (2007). *The Maintenance Management Framework: Models and Methods for Complex Systems Maintenance* (1st ed.). Springer London. https://doi.org/10.1007/978-1-84628-821-0

Matheson, E., Minto, R., Zampieri, E. G. G., Faccio, M., & Rosati, G. (2019). Human–Robot collaboration in Manufacturing Applications: a review. *Robotics*, *8*(4), https://doi.org/10.3390/robotics8040100.

Mitrea, D., & Tamas, L. (2018). Manufacturing execution system specific data analysis-use case with a cobot. *Ieee Access : Practical Innovations, Open Solutions*, *6*, 50245–50259.

Morgan, J., & O'Donnell, G. E. (2017). Enabling a ubiquitous and cloud manufacturing foundation with field-level service-oriented architecture. *International Journal of Computer Integrated Manufacturing*, *30*(4–5), 442–458. https://doi.org/10.1080/0951192X.2015.1032355.

Mourtzis, D., & Vlachou, E. (2018). A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance. *Journal of Manufacturing Systems*, *47*, 179–198. https://doi.org/10.1016/j.jmsy.2018.05.008.

Nagatani, K., Abe, M., Osuka, K., Chun, P., Okatani, T., Nishio, M., Chikushi, S., Matsubara, T., Ikemoto, Y., & Asama, H. (2021). Innovative technologies for infrastructure construction and maintenance through collaborative robots based on an open design approach. *Advanced Robotics*, *35*(11), 715–722.

Napoleone, A., Macchi, M., & Pozzetti, A. (2020). A review on the characteristics of cyber-physical systems for the future smart factories. *Journal of Manufacturing Systems*, *54*(January 2020), 305–335. https://doi.org/10.1016/j.jmsy.2020.01.007

Olsson, H., Åström, K. J., De Wit, C. C., Gäfvert, M., & Lischinsky, P. (1998). Friction models and friction compensation. *Eur J Control*, *4*(3), 176–195.

Osterrieder, P., Budde, L., & Friedli, T. (2020). The smart factory as a key construct of industry 4.0: a systematic literature review. *International Journal of Production Economics*, *221*, 107476. https://doi.org/10.1016/j.ijpe.2019.08.011.

Park, Y. S., Yoo, D. Y., & Lee, J. W. (2021). Programmable Motion-Fault detection for a collaborative Robot. *Ieee Access : Practical Innovations, Open Solutions*, *9*, 133123–133142. https://doi.org/10.1109/ACCESS.2021.3114505.

Pimentel, M. A. F., Clifton, D. A., Clifton, L., & Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, *99*, 215–249. https://doi.org/10.1016/j.sigpro.2013.12.026.

Qian, J., Song, Z., Yao, Y., Zhu, Z., & Zhang, X. (2022). A review on autoencoder based representation learning for fault detection and diagnosis in industrial processes. *Chemometrics and Intelligent Laboratory Systems*, *231*, 104711. https://doi.org/10.1016/j.chemolab.2022.104711.

Romanov, D., Korostynska, O., Lekang, O. I., & Mason, A. (2022). Towards human-robot collaboration in meat processing: Challenges and possibilities.*Journal of Food Engineering*,111117.

Sherwani, F., Asad, M. M., & Ibrahim, B. S. K. K. (2020). Collaborative Robots and Industrial Revolution 4.0 (IR 4.0). *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, 1–5. https://doi.org/10.1109/ICETST49965.2020.9080724

Tao, F., Qi, Q., Liu, A., & Kusiak, A. (2018). Data-driven smart manufacturing. *Journal of Manufacturing Systems*, *48*(January), 157–169. https://doi.org/10.1016/j.jmsy.2018.01.006.

Wescoat, E., Krugh, M., & Mears, L. (2021). Random forest regression for predicting an anomalous condition on a UR10 cobot end-effector from purposeful failure data. *Procedia Manufacturing*, *53*, 644–655.

Wescoat, E., Kerner, S., & Mears, L. (2022). A comparative study of different algorithms using contrived failure data to detect robot anomalies. *Procedia Computer Science*, *200*, 669–678.

Xiao, H., Yi, K., Liu, H., & Kou, G. (2021). Reliability modeling and optimization of a two-dimensional sliding window system. *Reliability Engineering & System Safety*, *215*, 107870.

Zhang, S., Li, S., Li, X., Xiong, Y., & Xie, Z. (2022). A human-robot dynamic fusion safety algorithm for collaborative operations of cobots. *Journal of Intelligent & Robotic Systems*, *104*(1), 1–14.

Zacharaki, N., Dimitropoulos, N., & Makris, S. (2022). Challenges in human-robot collaborative assembly in shipbuilding and ship maintenance, repair and conversion (SMRC) industry. *Procedia CIRP*, *106*, 120–125.