



Intelligent monitoring of multi-axis robots for online diagnostics of unknown arm deviations

Moncef Soualhi^{1,2} · Khanh T. P. Nguyen² · Kamal Medjaher² · Denis Lebel³ · David Cazaban³

Received: 23 April 2021 / Accepted: 17 November 2021 / Published online: 10 January 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022, corrected publication 2022

Abstract

In the age of Industry 4.0, multi-axis robots are widely used in smart manufacturing thanks to their capacity of milling high complex forms and interacting with several systems in production lines. However, during manufacturing, the occurrence of small drifts in the robot arms may lead to critical failures and significant product quality damages and, therefore, high financial losses. Hence, this paper aims to develop an effective and practical methodology for online diagnostics of robot drifts based on information fusion of direct and indirect monitoring. The direct monitoring exploits the already installed encoders on each servomotor of the robot while the indirect monitoring uses heterogeneous sensors (current, vibration, force and torque) placed at the robot tool level. The sensor measurements of the robot tool are processed, in an offline phase, to build health indicators and fused to learn a classifier for drifts detection and diagnostics. Then, during the online phase and in the case of presence of new drift patterns, the encoder measurements are used to label these patterns and update the classifier learned previously to diagnose their origin. The efficiency and robustness of the proposed methodology are verified through a real industrial machining multi-axis robot that investigates different drift severities of its arms.

Keywords Prognostics and health management · Condition monitoring · Intelligent monitoring · Fault detection and diagnostics · Signal processing · Information fusion · Multi-axis robot · Industry 4.0 · Machine learning

List of symbols

HIs	Health indicators of the indirect monitoring	StD, σ	Standard Deviation
HIs'	Normalized HIs	VAR	Variance
\bar{HIs}'	Mean value of indirect monitoring HIs'	KUR	Kurtosis
RMS	Root Mean Square	y_h	Raw signal segment
		y'_h	Normalized raw signal segment
		L	Length of each signal segment y_h
		FFT	Fast Fourier Transform
		Y	Global raw signal
		Ne	Number of signal segments
		h	Ensemble of observations of each segment Ne
		Ω	Class of observations of indirect monitoring
		AE	Auto-Encoder
		f	Encoding function of the AE
		g	Decoding function of the AE
		w, b	Weights and bias of the encoding layer
		w', b'	Transposed w and b of the decoding layer
		O_1	HIs' input observations to the AE
		O_2	Fused HIs' through hidden layer encoding
		O_3	Reconstructed HIs' of AE output layer
		MSE	Mean Square Error of O_1 and O_3
		D_{O_2}	Euclidean distance of HIs' to class's centroid
		\bar{D}_{O_2}	Mean of Euclidean distance of D_{O_2}

¹ FEMTO-ST Institute, Université de Bourgogne Franche-Comté CNRS/UFC/ENSMM, 26 rue de l'Épitaphe, 25000 Besançon, France

² Production Engineering Laboratory, Toulouse University INPT-ENIT, 47 Av. d'Azereix, 65000 Tarbes, France

³ Technology Transfer Center, METALLICADOUR, 1 cours de l'Industrie, 64510 Assat, France

P_{thr}	Peripheral threshold of each HIs' class
C	Centroids of indirect monitoring
n	Number of encoder data observations
$error$	Position error of robot's axes
\bar{error}	Mean of position error of robot's axes
s	System health state
$Indi_i$	Direct monitoring indicator of the i th axis

Introduction

Nowadays, the rise of intelligent robots is one of the prominent enablers towards smart manufacturing facilities (Lee et al., 2015; Pivoto et al., 2021). The implementation of Artificial Intelligence (AI) algorithms in robotic systems improve their efficiency as well as their interaction with other systems in the production lines. In this direction, one can cite the machining robots that facilitate complex machining processes due to the flexibility of robot arms with high precision and high-performance (Gotlib et al., 2017). However, during the manufacturing process, the occurrence of machining tool defects or drifts in the robot arms impacts the accuracy of the robot Tool Center Position (TCP) and then leads to critical product quality damages. For example, the authors in Qiao and Weiss (2017) and Kuric et al. (2018) show that the errors produced in the robot positioning significantly affect the repeatability of the robot motions and consequently lead to significant product quality damages. In the same domain, the studies in Wang et al. (2021), Tobon-Mejia et al. (2012), and Terrazas et al. (2018) indicate that cutting tool defects are considered as the main factor of damages in milling processes. Therefore, it is necessary to deploy advanced monitoring techniques to maintain the system in good conditions and ensure adequate production quality.

In literature, the first issue concerning the tool wear damage is widely addressed by monitoring different types of measurements such as current, vibration, force and torque. The studies (Heinemann et al., 2007; Selvaraj et al., 2014; Shi et al., 2018; Ogedengbe et al., 2011; Madhusudana et al., 2016; Hsieh et al., 2012; Bhuiyan et al., 2014) addressed tool condition monitoring for fault detection, diagnostics and prognostics by using signal processing techniques and machine learning algorithms. In detail, the authors in Heinemann et al. (2007), Selvaraj et al. (2014), and Shi et al. (2018) used force signals to extract statistical features in the time domain (peak-to-peak, variance, mean, etc.) to characterize different faults and diagnose and prognose system failures (Tobon-Mejia et al., 2012). Besides, the authors in Bhuiyan et al. (2014), Hsieh et al. (2012), Ogedengbe et al. (2011), and Madhusudana et al. (2016) used frequency analysis of current and vibration signals to detect and localize tool abnormalities. In contrast, the presented works in Khajavi et al. (2016) and Ratava et al. (2017) used Artificial Neural Net-

work to learn different health states of machining tool using a pattern recognition technique. Nevertheless, all these studies are developed for Computer Numerical Controlled (CNC) machines and cannot be generalized to all systems and particularly to robots due to their dynamic behaviors.

Concerning the second issue related to robot behavior monitoring, it can be tackled in two ways, as shown in Fig. 1: (1) by focusing on the product quality or (2) by analyzing the robot axes positioning.

The first possibility remains an efficient way to detect anomalies in the robot. However, it is not possible to track back the origin of the defect, i.e. to identify the axis that deviates from its nominal trajectory. A major part of researches on monitoring the product quality proposes alternative solutions such as using additional sensors on the robot and then, based on their processing results, act on the system to correct the errors. In Segreto et al. (2015), the authors use an acoustic emission sensor placed at the robot tool level for pattern recognition of polished workpiece surface roughness. The presented work in Chen and Nof (2007) estimates that errors of the end-effector of a robot may highly impact the product quality. Regarding the second option, i.e. the monitoring of robot axes behavior, it remains unexplored. The first studies in this direction are conducted by the National Institute of Standards and Technology (NIST) center by implementing the prognostics and health management (PHM) concept for condition monitoring of multi-axis robots. Their published works (Qiao & Weiss, 2017, 2018) propose the utilization of a multi-dimensional laser tracker sensor to monitor the robot TCP in a multi-space dimension. However, this type of technology remains expensive for widespread use in real industrial processes. Therefore, the development of alternative techniques to monitor the behavior of the multi-axis robot is still an important challenge for researchers as well as for industrials.

Considering the above synthesis, one can notice that only a few works addressed the issue of condition monitoring of the robot axes behavior, despite its importance and the benefits it brings for the end-users. This is due to the difficulty of modeling the dynamic behavior of the robot axis motions in real applications while taking into account all possible combinations of axis drifts that can occur, such as drift occurrence in axis 1 and axis 2 or drift occurrence in axis 4 and axis 5 simultaneously and so forth. Moreover, given the complexity of the robot arm, it is challenging to identify the specific arm that impacts the robot tool center position. Therefore, in the remaining of this paper, a new methodology is proposed to deal with this issue.

Among different anomalies of a robot, the arm deviations from the nominal positions are one of the crucial degradations that must be cautiously tracked. Indeed, a small deviation in the robot positioning may significantly affect security and product quality. Hence, it is necessary to develop an efficient

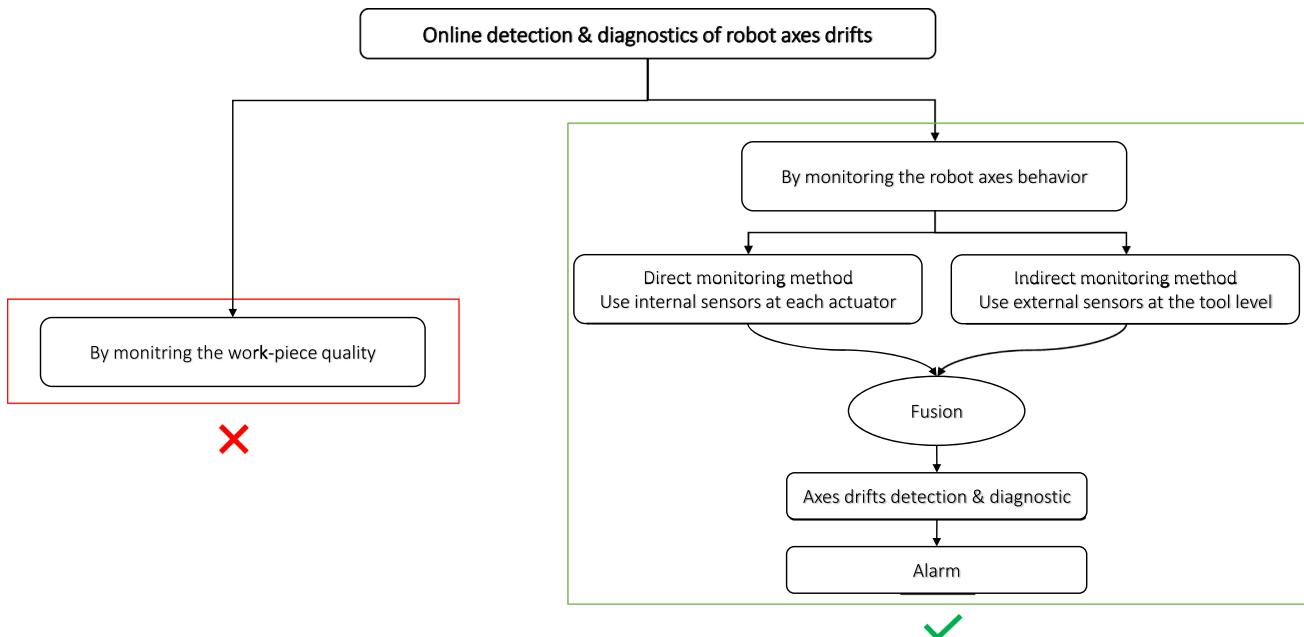


Fig. 1 Overview of the paper's positioning contribution

monitoring methodology to detect and diagnose the origin of the drifts. Thus, this paper presents a new methodology for online diagnostics of the origin of the robot axis drifts using an information fusion procedure of direct and indirect monitoring techniques. Its main contributions are summarized as follows:

- An efficient classification of robot axis drifts using indirect monitoring technique.
- Proposition of reliable health indicator constructed from the encoder measurements of each robot's axis to identify the origin of the drifts.
- Construction of a new fusion procedure between direct and indirect monitoring for efficient diagnostics of unknown robot axis drifts.

In detail, the direct monitoring exploits the already installed encoders on each servomotor of the robot while the indirect monitoring uses heterogeneous sensors (current, vibration, force and torque) placed at the robot tool level. First, in an offline phase, the heterogeneous sensor measurements of the indirect monitoring mode are used to build health indicators which are then fused through an auto-encoder (AE) network. The output of this latter model consists of new relevant health indicators that are projected on a multidimensional space to reveal the different health states of the robot. These states are then learned by a classifier for later detection and diagnostics. In practice, it happens that new drifts may occur during the operation of the robot. In this case, the previously learned classifier cannot recognize the new pattern and need to be updated. Therefore, the model is

updated in the online phase by exploiting the data provided by the direct monitoring which are used to localize the origins of the new pattern detected previously. The update consists of labeling the new detected drifts and using them to re-train the classifier. The advantage of using the indirect monitoring is that it can continuously assess the robot health state from the beginning to the end of the machining process and can early detect the abnormal behavior before failure. However, its performance strictly depends on the availability of the data related to all the possible drift scenarios that the robot may undergo to learn a representative classifier.

Concerning the direct monitoring, it allows us to precisely localize the origin of any drift in the robot axes, even if it is not learned. Nevertheless, this procedure is less useful to early detect the appearance of drifts in this case. This is because of an elapsed time needed to collect sufficient data from the encoder sensors for processing. Consequently, its practical implementation becomes questionable.

Hence, based on the advantage of indirect monitoring, that can early detect the drifts, and the advantage of the direct monitoring, to localize the origin of new drifts, the fusion of these two techniques can be a promising contribution. It allows practitioners to monitor their machining processes from the beginning, quickly detect and localize axes deviation origins, and proposing appropriate decisions for more availability, security and quality.

The remainder of this paper is structured as follows. Section 2 presents the global methodology for diagnostics of the robot-axes deviations. In Sect. 3, the performance of the proposed method is highlighted through experimental tests

carried out on a machining six-axis robot. Finally, the conclusion and perspectives of this work are given in Sect. 4.

Proposed methodology

This section presents a novel methodology based on the information fusion of direct and indirect monitoring approaches for multi-axis robot drifts detection and diagnostics. The proposed methodology allows inheriting the advantages of both monitoring and overcoming the drawbacks of each approach considered separately. Its pros and cons are summarized in Table 1.

In direct monitoring, the encoder sensors already placed on each robot actuator are used to track the errors between the nominal and the actual positions of the robot axes. Then, these errors are investigated to evaluate the health indicator of each robot axis, and thus localize the origin of the axes drifts. The direct use of the encoder measurements allows precisely and effectively identifying the drifts origin. However, this technique requires an elapsed time of few minutes to collect an essential amount of data for assessing the robot axis motions and therefore cannot be widely deployed in real-time applications for early anomaly detection.

In indirect monitoring, instead of directly tracking the robot motions, another part of the robot that is significantly affected by the robot axis deviations is investigated. In fact, the last robot arm (axis), which carries the end-effect tool, can represent the positioning of the previous articulations. When one of the arm motion deviates from its nominal position, it directly affects the accuracy of the tool center placement. Hence, monitoring the tool behavior can provide useful information about the deviation of the robot axes. Also, at the tool level, it is not difficult to install sensors for acquiring different signals such as vibration, force, and torque. Each of these signals brings a particular advantage to monitor the robot behavior, so the fusion of them can provide a promising alternative contribution to track the robot motions in real-time for early anomaly detection. However, as the sensors used in this

monitoring approach do not directly reflect the robot motions, it is necessary to map the relation between the acquired measurements and the axis drift types in the offline phase before deploying continuous monitoring online. In other words, the indirect monitoring approach, based on supervised learning techniques, requires many historical data to learn the fault patterns before diagnosing them.

Inspiring from the advantages of direct and indirect monitoring approaches, the proposed methodology, presented in Fig. 2, uses information of the former method to automatically label unknown anomalies and then update the classifier in the latter for early detection and diagnostics of these anomalies in the future. This methodology is principally composed of two phases, offline and online. The details of the offline phase, that exploits measurements from the indirect monitoring method to build health indicators, train a classifier model, and evaluate the peripheral threshold of the existing classes, are described in Sect. 2.1. In the online phase (Sect. 2.2), the monitoring system uses measurements from the tool-level sensors to indirectly track the robot motions, and thus early detect and diagnose the axis drifts, whose fault patterns have already been learned in the offline phase. For unknown anomaly observations from the tool-level sensors, the direct monitoring approach using the robot encoder measurements will be launched to localize the origin of the axis drifts. Using the information from the direct monitoring approach, the relevant observations from the indirect monitoring are labeled and thus, the classifier for online diagnostics of the robot axis drifts is updated. This update procedure will be presented in Sect. 2.2.1, while the principal steps of the direct monitoring mode will be detailed in Sect. 2.2.2.

Offline phase: learning failure patterns from indirect monitoring

This phase aims to process the recorded raw data of the sensors placed at the robot tool level to learn failure patterns

Table 1 Pros and cons of the proposed methodology

	Advantages	Disadvantages
Direct monitoring	No additional sensors	Require an elapsed time to start
	No additional acquisition system	Discrete inspection
	Localize the origin of drifts	Infeasible in case of faulty encoder
Indirect monitoring	Continuous monitoring	Require additional sensors
	Early detection and diagnostics of known robot axis drifts.	Require many historical scenarios to learn patterns.
Information fusion of direct and indirect monitoring	Continuous monitoring Automatic update of learned patterns Diagnostics of unknown axis drifts.	Infeasible in case of faulty encoder

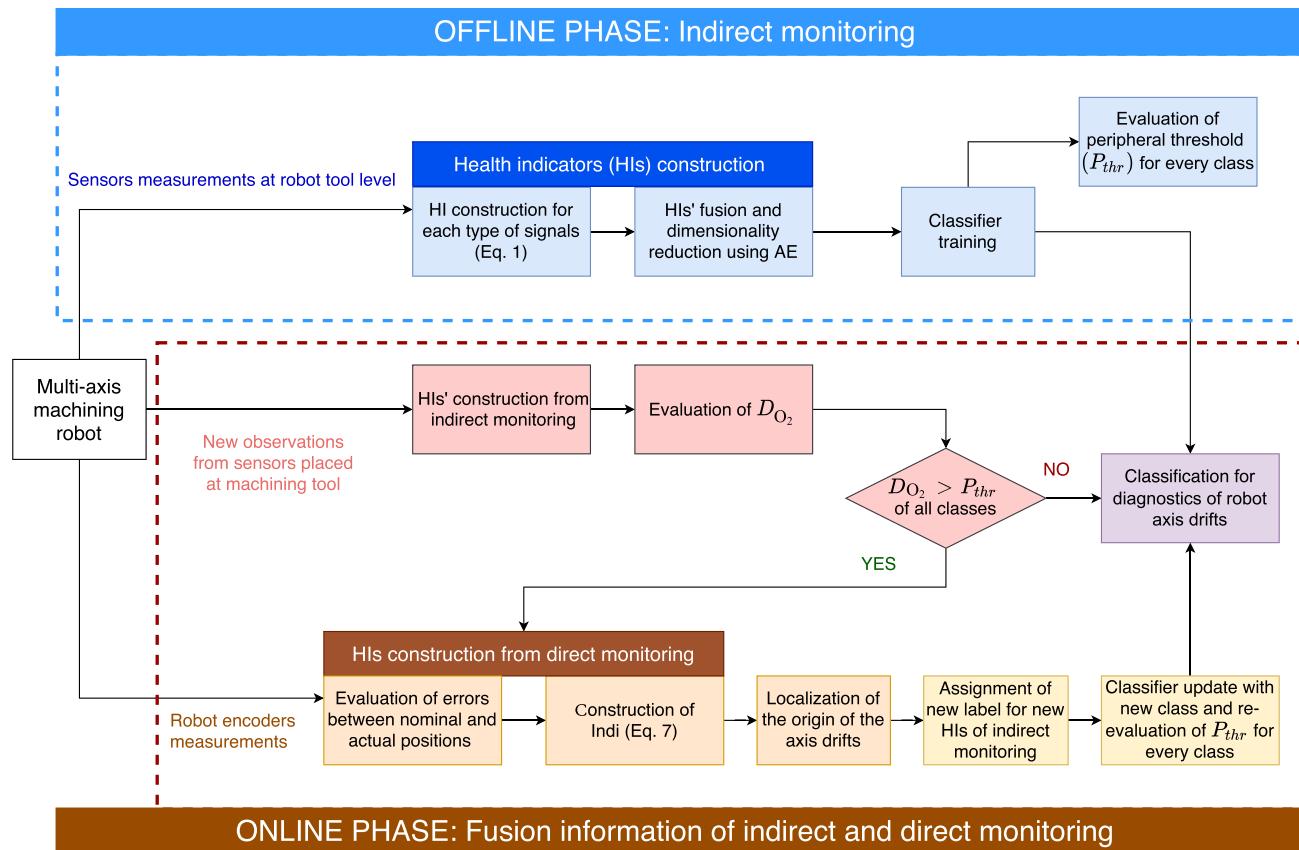


Fig. 2 Overview of the proposed methodology

from the indirect monitoring. For this purpose, the collected observations, corresponding to the common types of the axis drifts, are injected into a treatment procedure consisting of the following steps:

- Construction of health indicator for each signal type (current, vibration, force and torque)
- Fusion of these health indicators based on an auto-encoder network
- Classification of the robot axis drifts and evaluation of the peripheral threshold for every existing class

Health indicators construction

In this step, the data recorded, e.g. the current, the vibration, the force and the torque signals will be injected into a signal processing pipeline to extract relevant features and build health indicators. This pipeline, presented in Fig. 3, consists of three principal steps.

- **Raw data segment:** This step allows reducing the data size for signal processing and takes only the relevant features. It splits the raw signal Y into N_e segments denoted

y_h of length L , where h is the ensemble of observations of the segmented signal ($h \in [1 \dots N_e]$).

- **Condition normalization:** Each signal segment is divided by the max value of the segment spectrum, $\text{MAX}(FFT(y_h))$. It allows limiting features dispersion caused by the operation condition variations and then separating the classes that represent different levels of the operating modes.
- **Feature extraction and combination:** This step aims to extract statistical features from the normalized splitting signals to construct the health indicator (HI). Its expression is denoted by the following equation:

$$HI_h = KUR(y'_h) \times VAR(Y)^2 \quad (1)$$

where y'_h and Y are the normalized segment and the total raw data, respectively. The KUR and VAR are respectively the kurtosis and the variance feature values. The kurtosis measures the signal flattening, which indicates the impulsive property of the signal by the centered moment of order 4 of the normalized segmented signal y'_h . It reduces the signal's noises caused by the sensor measurements. The variance allows calculating

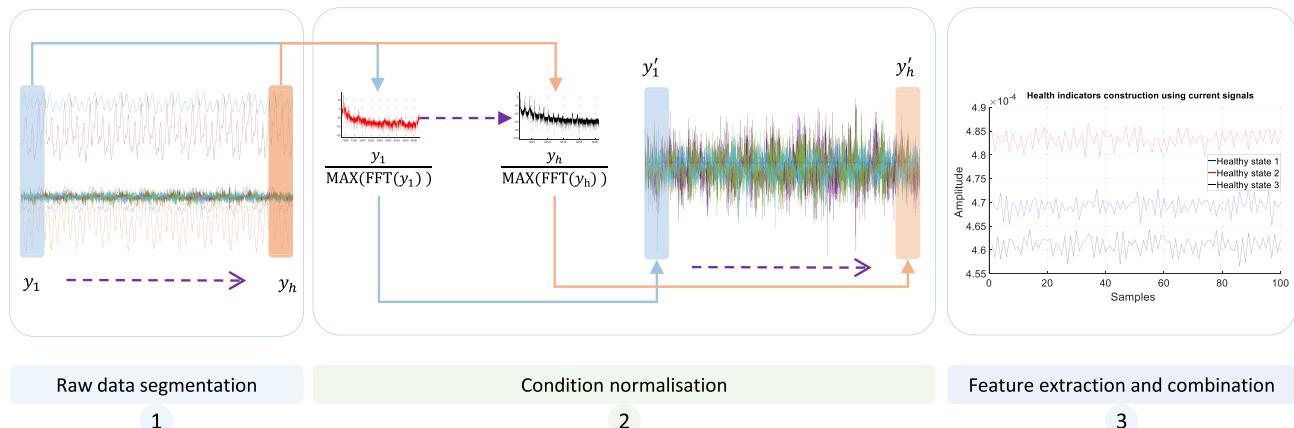


Fig. 3 Signal processing pipeline for health indicator construction (Soualhi et al., 2020)

the average energy of the total signal Y by measuring its dispersion.

In summary, the built health indicator is a combination of different features, that are extracted from both time and frequency domains. Its efficiency and robustness have been verified through different case studies taking into account numerous fault types of critical components such as bearings, gears, rotor-bars, unbalanced supply motors and cutting tools (Soualhi et al., 2020). In this paper, it is used to characterize different states of robot axis drifts. However, as the HI is extracted from the measurements of sensors placed at the tool level, it only indirectly reflects the deviation of the robot axes. Each HI extracted from each signal (current, vibration, force and torque) can bring a particular advantage to monitor the robot arms behavior. Hence, to reinforce the capacity to detect and diagnose multiple axis drifts, the fusion of these HIs will be discussed in the next subsection.

Health indicators fusion

This second step aims to combine the constructed health indicators from different signals to create a robust fused ones that detect well the origin of the robot deviations. In fact, as each measurement type has its own properties to reflect the system health states, considering simultaneously all the constructed health indicators from different measurements can be a promising solution to enhance the ability of system health assessment (Meng et al., 2020; Jimenez et al., 2020; Cai et al., 2020). However, processing these data separately may lead to confusion because, under the impact of the operating condition variations, some measurements do not properly reflect the actual status of the system. To remedy this situation, health indicators fusion is performed. This

process aims to fuse all the processed raw data into one representative pattern.

For this purpose, different techniques were proposed in the literature. One can cite the Principal Component Analysis (PCA) (Loutas et al., 2019), Isometric Feature Mapping (ISOMAP) (Ali & Saidi, 2018) and Auto-Encoder (AE) (Lin & Tao, 2019). Among these techniques, the AE can perform transformations with non-linear activation functions. Moreover, it provides a high level of fusion performance thanks to the learning task of the neural networks which do not require complex tuning parameters. Therefore, in this work, the AE network is proposed to fuse the health indicator observations extracted from each sensor. The general structure of the AE consists of three-layer types that are: an input layer, a hidden layer (encoding layer) and an output layer (decoding layer), as shown in Fig. 4.

As the health indicators are extracted from different sensors, it is necessary to normalize them into a nominal scale (Nguyen & Medjaher, 2019) before the fusion step with the AE network. Hence, the following Eq. (2) is used to normalize the health indicators, corresponding to the different sensors, to their dispersion.

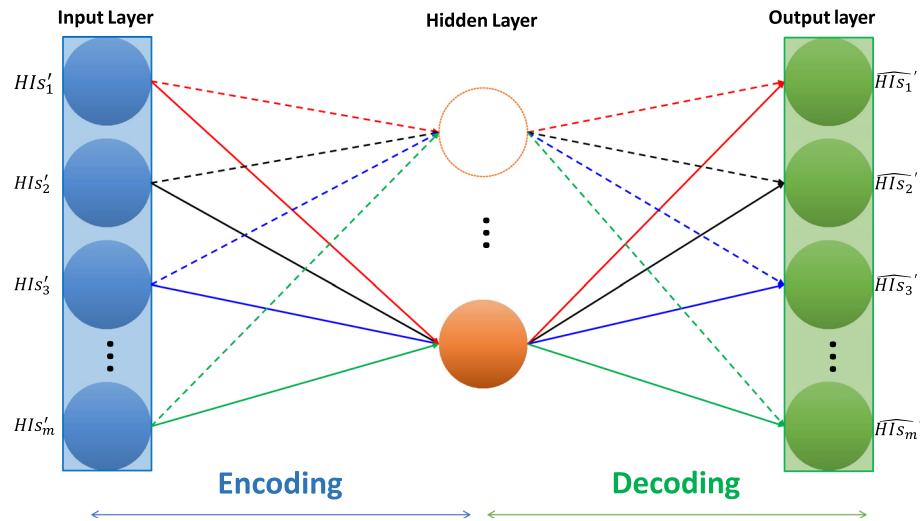
$$HIs' = \frac{(HIs - \bar{HIs})}{StD(HIs)} \quad (2)$$

HIs represents all the health indicators constructed from different sensors and corresponds to a matrix of $m \times x$ observations where m is the number of HIs samples and x is the number of sensors. The StD is the standard deviation of the global observations and is $1 \times x$ observations.

Next, the AE receives the normalized health indicators HIs' as the inputs for the encoding task as expressed by Eq. (3).

$$O_1 = [HIs'_{(1)}, HIs'_{(2)}, \dots, HIs'_{(m)}] \quad (3)$$

Fig. 4 Illustration of Auto-Encoder network structure



where O_1 is the input layer of the AE network and $HIS'_{(1)}, HIS'_{(2)}, \dots, HIS'_{(m)}$ are the observations of the normalized HIS .

Then, the encoding task (fusion) is performed through the activation function f .

$$O_2 = f(w_n \times O_1 + b) \quad (4)$$

where O_2 is the output of the hidden layer of the AE network and represents the fused HIS' , n is the number of hidden units and also the reduction size and f is the activation function, w and b are the weights and bias of the network, respectively. For an illustration, if $n = 3$, all the normalized HIS' will be reduced to 3 dimensions, as expressed by Eq. (4).

In the third layer, and by using Eq. (5), the reconstructed task is launched.

$$O_3 = g(w'_n \times O_2 + b') \quad (5)$$

O_3 is the output layer of the AE network which corresponds also the reconstructed input data and where w' and b' are the transposed weights and bias values used for decoding the data and g is the activation function which can be the same or different than the one used in the encoding phase.

The output of the decoding phase O_3 is then injected into a loss function that characterizes the difference between the AE output with the input data. Training an AE network aims to minimize this loss function by updating the weights and the bias of the encoding and decoding phases using the gradient-descent back-propagation algorithm. The performance of the AE is evaluated by the Mean Square Error (MSE) between the input and the output layers.

In this study, through the proposed AE, multiple HIS extracted from different signals are embedded into three-dimensional space to facilitate the graphical visualization of the fault patterns for further fault detection and diagnostics

tasks. This choice is based on the fact that in fault detection and diagnostics, one need to extract fault signatures to detect the healthy state from the faulty states in graphical interface before injecting all the observations in a classifier model for diagnostics. In general, one-dimensional space may cause confusion between the observations and lead to miss interpretation of the results. For this purpose, to have a clear visualization, the three-dimensional representation graphics allow to clearly show the separability between the patterns. The following Fig. 5, inspired from Soualhi et al. (2019a), illustrates the difference between 1D and 3D representations of fault patterns. One can see that there exists confusion between different classes in 1D space while 3D representation allows clearly visualizing them.

Classification of robot axis drifts and evaluation of their classes peripheral thresholds

The last step of the indirect monitoring aims to learn the failure patterns for online fault detection and diagnostics. For this purpose, the observations of the fused health indicators are taken as a training database (Ω_p). This database is used to learn the classifier model how to map each observation of (Ω_p) to its corresponding class s (e.g. $s = 0$ corresponds to the healthy class and $s = 1$ correspond to a faulty class 1). Note that this technique is a supervised learning which requires prior samples of the healthy and the faulty classes.

In literature, there exist several machine learning classifiers. One can cite the most popular ones such as Discriminant Analysis (DA) (Cho & Jiang, 2018), Support vector machine (SVM) (Fatima et al., 2015), K-Nearest Neighbor (K-NN) (Madeti & Singh, 2018), Decision Tree (DT) (Samantaray, 2009), Naive Bayes (NB) (Mukherjee & Sharma, 2012), and Adaptive Neuro-Fuzzy Inference System (ANFIS) (Soualhi et al., 2019b). All of these classifiers can perform good clas-

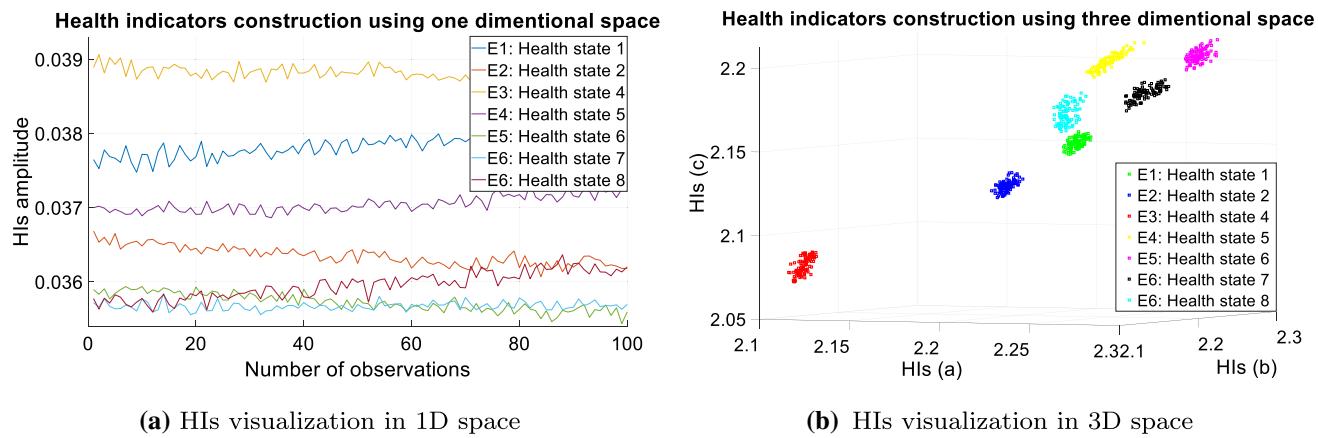


Fig. 5 Health indicators in one and three dimensional space representation

sification with a high level of accuracy results depending on the separability between the different health states to be trained. Thanks to the proposed processing methodology, the different health states of the system can be clearly separated with negligible dispersion between the HI observations and therefore it offers promising diagnostic results.

In this study, to highlight the performance and robustness of the proposed methodology, different classifiers, e.g. DA, SVM, K-NN, DT, NB and ANFIS, will be considered. Among them, the K-NN model is the most classic and popular one thanks to its simplicity of implementation and fast computing time.

After classification of the different types of the axis drifts, the peripheral threshold for each class (P_{thr}) is evaluated. This value is served to consider whether a new observation belongs to an identified class or not with a predetermined significance level. To calculate it, we firstly identify the centroid of each class, then evaluate the Euclidean distances, noted D_{O_2} , from each observation belonging to this class to its centroid. Then, inspired by the empirical 99.7% rule used in statistics, the peripheral threshold for each class is defined by:

$$P_{thr} = \bar{D}_{O_2} + 3 \times \sigma(D_{O_2}) \quad (6)$$

where \bar{D}_{O_2} and $\sigma(D_{O_2})$ are respectively the mean and the standard deviation values of the distances from each observation of the fused HIs' to its class's centroid. Comparing the distances of a new observation to the centroid of each existing class with its P_{thr} , the monitoring system can decide whether it is necessary to refer to the result of the direct monitoring mode to identify the label for this observation or not. This procedure will be detailed in the next subsection.

Online phase: detection and diagnostics of robot axes drifts

This phase of the proposed methodology performs an online continuous indirect monitoring of the robot motions. It uses the classifier model trained in the offline phase for early fault detection and diagnostics of the known types of the robot axis drifts. It also allows automatically labeling uncertain observations and updating the classifier with the information from the direct monitoring. The information fusion algorithm for an effective online detection and diagnostics of the robot axis drifts will be presented in Sect. 2.2.1. Then, the direct monitoring process will be detailed in Sect. 2.2.2 to clarify how to localize the origin of the axis drifts with the encoder measurements.

Information fusion for online detection and diagnostics of robot axis drifts

Algorithm 1 Online detection and diagnostics of robot axis drifts

At each observation instant:

- 1: Load the vectors C and P_{thr}
 - 2: Collect new data from indirect monitoring
 - 3: Construct the HI for each type of signals
 - 4: Fuse the HIs from different sensor sources
 - 5: **for** $j = 1$ to $\text{length}(P_{thr})$ **do**
 - 6: Calculate the distance ($D_{O_2}(j)$)
 - 7: **if** $D_{O_2}(j) \leq P_{thr}(j)$ **then**
 - 8: Use the trained classifier for this new HI observation
 - 9: Break **for loop**, go to **Step 1** of next observation instant
 - 10: **end if**
 - 11: **end for**
 - 12: Launch the direct monitoring
 - 13: Use the result of the direct monitoring to label the fused HIs' observations from indirect monitoring
 - 14: Retrain the classifier
 - 15: Update the vectors C and P_{thr}
-

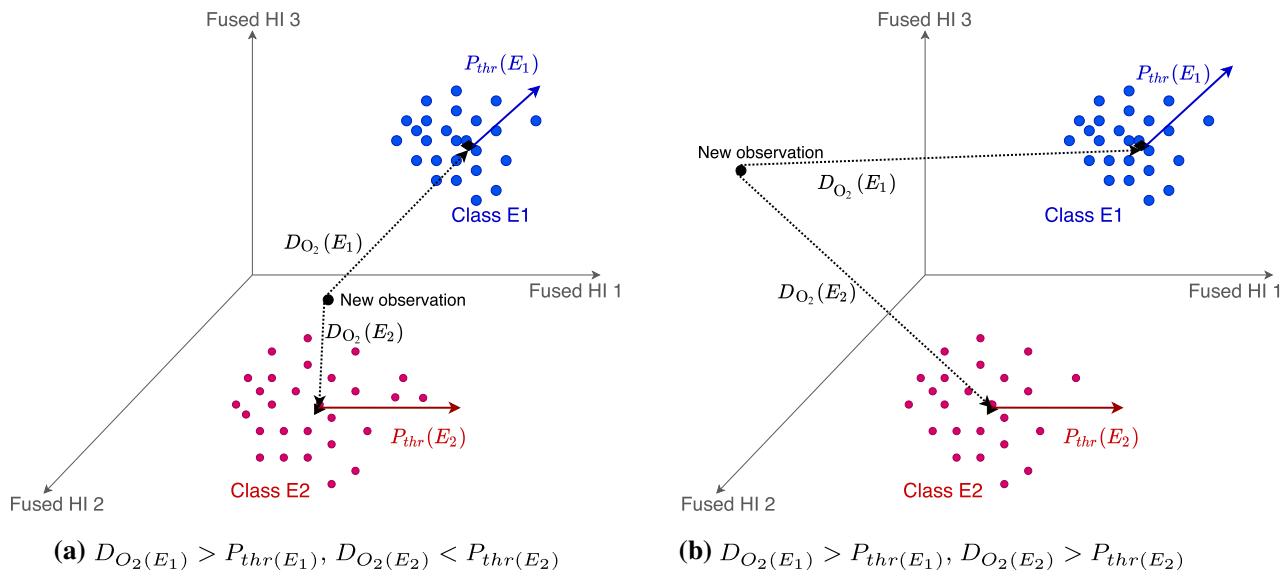


Fig. 6 Illustration of the comparison between D_{O_2} and P_{thr}

Algorithm 1 presents the procedure for reliable online detection and diagnostics of the robot axis drifts. From the beginning of a machining process (at each observation time) the proposed algorithm loads the actual vectors C and P_{thr} that consist of the centroids and the peripheral thresholds, respectively, of the existing classes characterizing all the current known types of robot axis drifts. In the same time, the data acquired from multiple sensor sources placed at the tool level are injected into the signal processing pipeline presented in Sect. 2.1.1 to construct the HIs' for each type of signal. These HIs are normalized by the previously calculated $MEAN$ and StD values (Eq. 2). They are fused using the AE model trained in the offline phase (2.1.2) to obtain new fused 3D indicators O_2 corresponding to the hidden layer encoding activity.

Next, the distance $D_{O_2(j)}$ from each observation coordinate to the centroid of each existing class j , is evaluated and compared with the peripheral threshold of this class, $P_{thr}(j)$ (Fig. 6). If at least a distance from the observation to the centroid of one existing class is lower than its peripheral threshold ($D_{O_2(j)}(E_2) < P_{thr}(E_2)$ in Fig. 6a), the trained classifier will be applied on this new observation to infer its class, i.e. diagnostics of the known axis drift. Otherwise, if $D_{O_2(j)}(E_2) > P_{thr}(j)$ for all j (Fig. 6b), the direct monitoring mode will be launched. After a necessary elapsed time needed to collect sufficient amount of encoder measurements, the direct monitoring can correctly localize the origin of the new axis drifts. This information is then used to label the observations obtained from indirect monitoring. Based on this, the classifier will be retrained and the vectors C and P_{thr} will be updated accordingly.

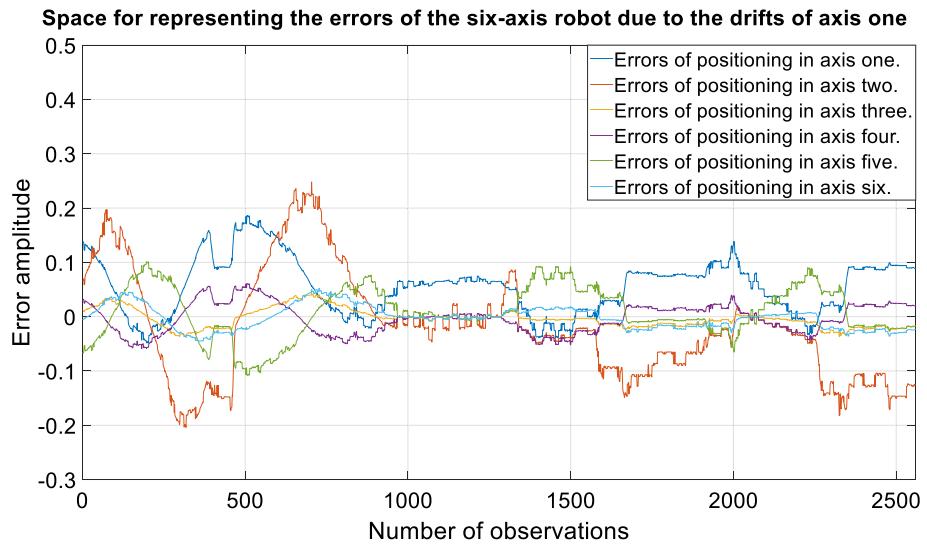
Direct monitoring using encoder measurements

As mentioned in Sect. 2, the online diagnostics of the robot axis drifts requires the combination of both the indirect and the direct monitoring methods. The indirect monitoring, presented in Sect. 2.1, uses measurements from sensors installed at the robot tool level while the direct monitoring aims to evaluate separately the positioning of each robot axis in order to localize the origin of the drift. In literature, only one study is proposed to ensure this latter task (Qiao et al., 2017). The authors use a laser tracker sensor to capture in space the axes motions but the method remains expensive and is not practical in a real process. Therefore, this subsection presents a new strategy to evaluate separately the robot axes behavior by the construction of a new health indicator. This indicator is built from the already installed encoder sensors in each servo-motor of the robot to track its motions. To do this, it is necessary to have access to the robot's control system to acquire the positioning values of each axis recognized by the robot's internal sensors. However, this task requires some elapsed time of few minutes to access and collect an essential amount of encoder measurements for the evaluation of the corresponding indicator $Indi$.

$$Indi_i = (RMS(error_i)/StD(error_i))^2 \quad (7)$$

where $error_i$ is the trajectory errors presented in the i -th axis during the elapsed time of the direct monitoring. For an illustration, Fig. 7 shows the trajectory errors of the six axes of the robot caused by the drifts of the first axis.

Fig. 7 Trajectory errors of all 6 axes due to the random deviation in the first axis



Once the errors are calculated, the *RMS* and *StD* expressed by Eqs. (8) and (9) can be combined to build an effective health indicator (Eq. 7) that directly detects the deviation origin of the robot axes.

$$RMS(error_i) = \sqrt{\frac{1}{n} \sum_{j=1}^n error_{ij}^2} \quad (8)$$

$$StD(error_i) = \sqrt{\frac{1}{n} \sum_{j=1}^n (error_{ij} - \overline{error}_i)^2} \quad (9)$$

where i and j represent the axis and observation number respectively, n is the total number of observations during the elapsed time of the direct monitoring, and \overline{error}_i is the mean value of the deviations in the i -th axis.

From Eqs. (8) and (9), one can see that the *StD* value is equal to *RMS* when the mean value of errors is 0. In this case, the $Indi_i$, expressed by Eq. (7), will be equal to 1. Hence, one can conclude that in the nominal case (no deviations) the proposed health indicator is close to 1 because the mean value of the errors tends to zero. In the presence of deviations, the health indicator will be significantly different from 1. From the statistical analysis of multiple empirical experiments, METALLICADOUR research and transfer center experts propose to fix the drift threshold to 2, i.e. if $Indi_i > 2$, then a drift in the i -th axis is detected. This threshold represents the minimum for which an impact on the quality of the final product is occurred. For an illustration, Table 2 presents the two observations of $Indi_i$ values corresponding to the i -th axis in the case where there exists drifts in the first axis. One can see that the direct observation of error trajectories, Fig. 7, does not allow detecting the drift origin, while the $Indi_i$ values presented in Table 2 clearly shows the origin

Table 2 Two observations of $Indi_i$ when there exists drifts in the first axis

$Indi_1$	$Indi_2$	$Indi_3$	$Indi_4$	$Indi_5$	$Indi_6$
5.82	1.00	1.08	1.05	1.06	1.05
6.5	1.3	1.03	1.2	1.05	1.3

of the robot axis drifts. Indeed, two $Indi$ observations of the first axis are greater than 2 (they are 5.82 and 6.5), while the $Indi$ values of other axes are, almost, close to 1.

Application and results

This section investigates the performance of the proposed methodology on a real case study realized at the METALLICADOUR research and transfer center in France. For this purpose, a multi-axis robot with six degrees of freedom is used to perform machining operations and to collect failure patterns corresponding to multiple single and combined drifts.

The details of this platform will be presented in Sect. 3.1. Then, Sect. 3.2 shows the performance of the direct monitoring used to localize the origin of the drifts and for which the failure patterns have not been learned by the classifier in indirect monitoring. Next, the performance of the indirect monitoring method for diagnostics of robot axis drifts, which failure patterns have already been learned, will be investigated in Sect. 3.3. Finally, Sect. 3.4 is dedicated to consider how the information fusion process performs online fault detection and diagnostics of known and unknown robot axis drifts.

Description of the machining platform

Figure 8 illustrates the global overview of the platform which consists of an ABB IRB 6660 six-axis robot known for its high-rigidity of machining metallic work-pieces with good quality. This robot is composed of six servomotors powered and controlled by an IRC5 control system. At the sixth axis of the robot, a High-Speed Machining (HSM) spindle is placed and equipped with a flat-end mill unit (of 10 mm diameter) for milling aluminum work-pieces. In this case study, the designed work-piece has a labyrinth form with a depth of 5 mm, as shown in Fig. 8. This shape allows taking into account all the robot arm motions leading to perform system-

level monitoring. Besides, the machining parameters are set as follows: spindle speed of 9000 rpm with 648 mm/mn of feed rate.

Table 3 summarizes the set-up parameters of two groups of experiments. The first group aims to process seven experiments that correspond to the nominal condition of the robot (E1), where no drift occurs, and six faulty states representing the single drifts of the six robot axis (E2, E3, ...E6), respectively. The second group of experiments is used to investigate the combined drifts of the robot arms under the same machining condition. For example, E1 & E4 is the case where there exist the combined drifts in the first and fourth axes. Note that the drifts are created by injecting errors in the axis rotation

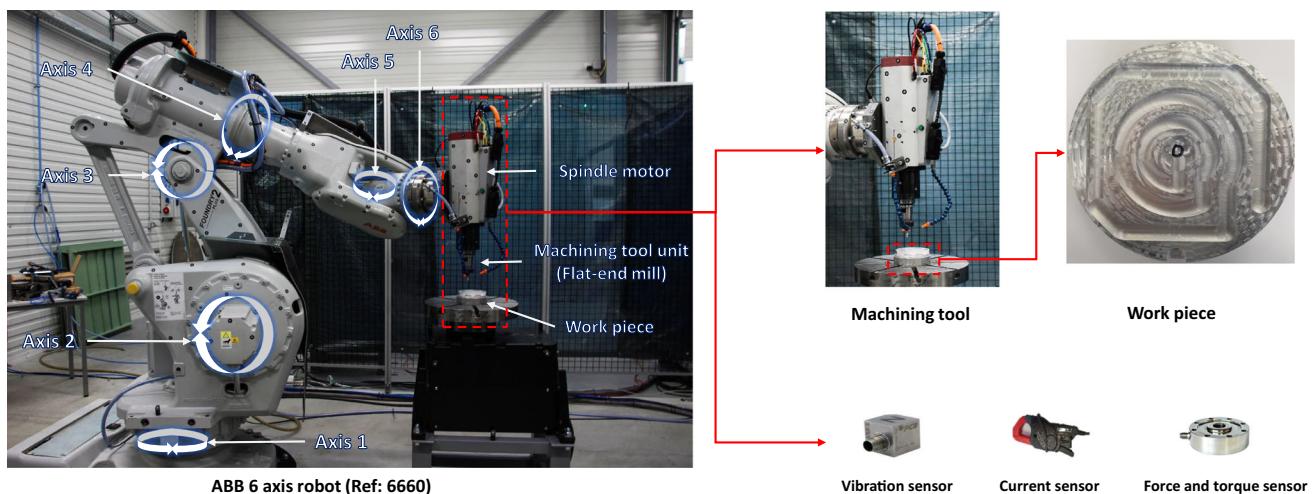


Fig. 8 Overall schema of the test bench

Table 3 Summary of experimental tests

Speed (rpm)	Feed rate (mm/mn)	Depth (mm)	Health state	Error degree (°)	Acquisition parameters
<i>First group of experiments</i>					
9000	648	5	E1: Healthy state E2: Drifts in axis 1 E3: Drifts in axis 2 E4: Drifts in axis 3 E5: Drifts in axis 4 E6: Drifts in axis 5 E7: Drifts in axis6	+0.065 +0.120 +0.080 +0.085 +0.120 +0.155	Hardware: IRC5/NI F_s (IRC5): 41.6 Hz F_s (NI): 25.6 KHz File format: (.csv) Duration: 60/file
<i>Second group of experiments</i>					
9000	648	5	E1: Healthy state E1 & E4: Drifts in axis 1 and 4 E2 & E4: Drifts in axis 2 and 4 E3 & E6: Drifts in axis 3 and 6 E4 & E5: Drifts in axis 4 and 5 E2 & E5: Drifts in axis 2 and 5 E4 & E6: Drifts in axis 4 and 6	-0.012/-0.04 +0.120/+0.04 -0.04/+0.155 +0.155/-0.04 -0.040/+0.04 -0.080/+0.18	Hardware: IRC5/NI F_s (IRC5): 41.6 Hz F_s (NI): 25.6 KHz File format: (.csv) Duration: 60/file

range during the process, i.e. $\pm 0.01^\circ$ of the actual position. Different error degrees from low to high can be performed while keeping tolerant values that should not be exceeded. These latter values can be determined according to standards used in the domain or given by experts. The development engineers at METALLICADOUR center recommend drifts in the range of $[\pm 0.06^\circ, \pm 0.18^\circ]$ from the nominal position. Also, note that the calculation of the drift degree is mainly related to the distance between the center of the workpiece and the center of the axis, which also needs an expert to evaluate the criterion threshold alarm.

Concerning the indirect monitoring process, the three-phase current sensors are placed at the IRC5 inverter output while the three-axis vibration, force and torque sensors are installed at the spindle tool level. The acquisition of these measurements is done by National Instrument (NI) devices with Labview acquisition software at a sampling frequency of 25.6 kHz. The recorded data are stored in (.csv) files with a duration of 5 s/file.

For the direct monitoring process, the servomotors of the robot are already equipped with encoders (one encoder for each servomotor). As the robot has six-axis, the total number of encoders equal to six. The measurements of each encoder are recorded by the IRC5 controller with a sampling frequency of 41.6 Hz and stored in (.xlsx) files.

Investigation on the performance of direct monitoring

As mentioned in the case study description, for direct monitoring, the encoders are installed on the rotating shaft of each servomotor and used to collect data during each experiment controlled by the IRC5 system. The encoder measurements represent the rotating position values of the robot arms. Following the method proposed in Sect. 2.2.2, these values are used to construct the indicators of the robot axes to directly assess the drift origins. Table 4 shows the obtained observations of $Indi_i$ through two groups of experiments. The first column indicates the origin of the drifts while the remaining six columns present the $Indi$ values of the 6 robot axes.

From Table 4, one can notice that the constructed health indicators allow clearly identifying which axis is the origin of the robot arm deviations. This identification is indicated when the health indicator values are greater than 2. For example, considering the first line in the first group of experiments, when the drift is injected into the first axis, the obtained values of the calculated health indicator from the second to the sixth axis are close to 1 (which means no drift) while the value of the first axis is equal to 2.80. This latter value indicates that the axis one deviated from its nominal trajectory during the process.

Moreover, even in the case of combined drifts, in the second group of experiments, the constructed health indi-

Table 4 HIs values of six robot axes when there exist single and combined drifts

Drift origin	<i>Indi</i> ₁	<i>Indi</i> ₂	<i>Indi</i> ₃	<i>Indi</i> ₄	<i>Indi</i> ₅	<i>Indi</i> ₆
<i>First group of experiments</i>						
Axis 1	2.80	1.00	1.00	1.05	1.06	1.05
Axis 2	1.06	2.57	1.00	1.06	1.06	1.00
Axis 3	1.04	1.03	9.00	1.02	1.05	1.00
Axis 4	1.06	1.03	1.02	13.7	1.03	1.00
Axis 5	1.00	1.02	1.03	1.25	4.02	0.50
Axis 6	1.13	1.01	1.01	1.14	1.12	121
<i>Second group of experiments</i>						
Axis 1 & axis 4	5.52	1.03	1.00	34.2	1.08	1.00
Axis 2 & axis 4	1.04	10.3	1.10	10.5	1.03	1.13
Axis 3 & axis 6	1.07	1.01	2.10	1.08	1.00	6.30
Axis 4 & axis 5	1.00	1.10	1.00	2.91	2.60	1.05
Axis 2 & axis 5	1.02	4.20	1.00	1.03	2.75	1.02
Axis 4 & axis 6	1.07	1.00	1.00	9.82	1.07	177

Bold indicates the robot axes present drifts

cators allow isolating the deviated axes during the process. For example, considering the first line in the second group of experiments, when the drifts are injected simultaneously into the first and fourth axes, the obtained values of the health indicators of these axes are respectively 5.52 and 34.2 while the ones of the remaining axes are close to 1.

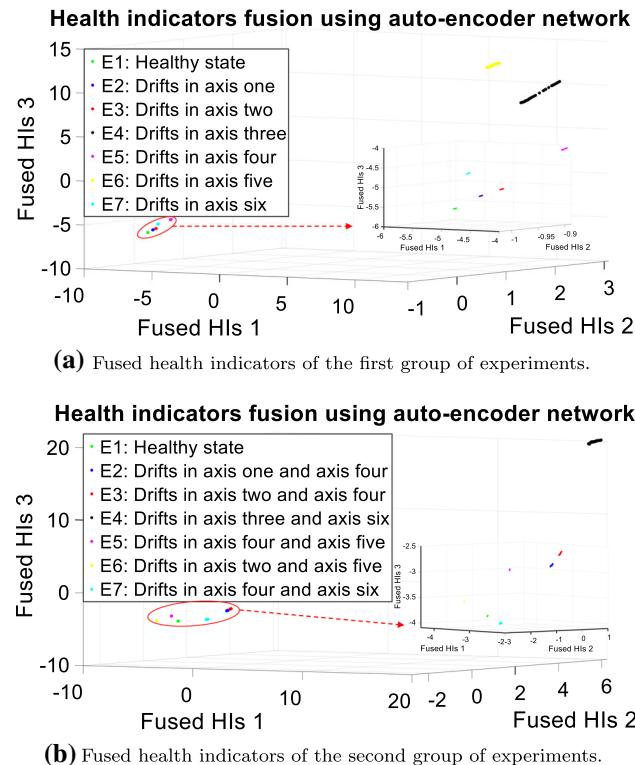
Investigation on the performance of indirect monitoring for the cases where the failure patterns have already been learned

For indirect monitoring, the recorded measurements of all sensors placed at the tool level (current, vibration, force and torque) are injected into the processing algorithm to extract features and build the fused health indicators. The tuning parameters of the AE network for *HIs'* fusion are summarized in Table 5. Figure 9 presents the distribution of the fused health indicators. From this figure, one can see that the proposed methodology allows clearly separating the different types of the robot axes deviations with a negligible dispersion between the observations. These positive results will allow significantly improving the performance of the classifier models for an automatic diagnostic of the origin of the robot trajectory deviations.

For an automatic detection and diagnostics of the origin of the robot drifts, the fused *HIs'* are fed into a classifier model. To highlight the performance of the fused *HIs'*, various common classifiers (SVM, LR, NB, DT, ANFIS and K-NN) are considered. Their accuracy, using the health indicators extracted separately from each measurement with the one handling the fused health indicators, are compared.

Table 5 Turning parameters of the AE network

Hidden unit	Epoch	Loss function	Encoder function	Decoder function
3	1000	MAE	Logsig	Purelin

**Fig. 9** Distribution of the fused *HIs'* under different machining conditions

There are in total ($100 \times 14 \times 4 \times 3$) observations of *HIs* from 4 sensors, that correspond to 100 samples for each type of signal in each experiment (14 experiments have been investigated). The fused health indicator observations will contain (1400×3) samples where 3 is the dimensional size of the fused *HIs'*.

To evaluate the robustness of the indirect monitoring method when the failure patterns of the robot axis drifts have been already learned, for each experiment the train and test set randomly take 50% of the total observations. The trained classifiers are then used to diagnose the origin of the robot drifts when considering the observations in the test set without prior knowledge about the operating condition and the fault type. The accuracy scores of these trained classifiers for diagnostic of the robot axis drifts are summarized in Table 6.

From Table 6, the performance of the fused health indicators obtained from the sensors placed at the tool level is highlighted with an accuracy score of 100% for all classifier models and all groups of experiments. In other words, when the failure patterns have been already learned, the proposed indirect monitoring method allows diagnosing exactly

the origin of the robot arm deviation. Considering the *HIs* extracted from only one sensor, one can notice that in the first group of experiments, where only the operating condition variation (speed) is investigated, the force and torque indicators can diagnose well the origin of the drifts with 100% accuracy while the current and vibration indicators cannot perform good results. Besides, in the second group of experiments, when injecting combined drifts on the robot axes, the current and the vibration indicators give better diagnostics (100% accuracy) instead of the force and torque indicators. These differences of results can be explained by the effects of the sensor positions on the quality of the indirect monitoring process. In fact, it is preferred to place the current sensor at the inverter or the supply controller of the robot to collect all useful information about current signals used and/or supplied at the system level. Also, the vibration sensors need to be placed as near as possible to the machining unit, while the force and torque sensors should be placed at the sixth axis level of the robot.

Besides, as the accuracy scores of all classifier models are 100% for all groups of experiments when using the fused *HIs'*, hereafter the authors propose to use the K-NN model, the most classic and simplest one, to learn the failure patterns in the offline phase and to diagnose the axes drifts in the online phase.

Investigation on the performance of the information fusion process

This section aims to investigate the performance of information fusion between direct and indirect monitoring for online detection and diagnostics of robot axes drifts. For this purpose, the first group of experiments is used to learn the pattern failures of single drifts in the offline phase, while the second one is used to illustrate how the information fusion process (Algorithm 1 presented in Sect. 2.2.1) works to detect and diagnose the unknown drifts (combined drifts) in the online phase.

Firstly, the data collected from the first experiment of the second group, corresponding to the healthy state (E1), are used to verify whether the proposed methodology can correctly group the E1 observations into the already existing classes (first group of experiments). Considering Table 7, its second and third columns present the centroid position and the peripheral threshold of the existing classes that have been already trained in the offline phase by using the data acquired from the first group of experiments. The last col-

Table 6 Accuracy score (%) of different classifiers for diagnostic of the robot axis drifts

Classifier model	Current <i>HIs</i>	Vibration <i>HIs</i>	Force <i>HIs</i>	Torque <i>HIs</i>	Fused <i>HIs'</i>
<i>First group of experiments</i>					
SVM	97.33	95.90	100	100	100
LR	98.67	97.57	100	100	100
NB	99.71	94.35	100	100	100
DT	96.79	94.86	100	100	100
ANFIS	95.15	94.11	100	100	100
KNN	99.74	95.64	100	100	100
<i>Second group of experiments</i>					
SVM	100	100	96.62	98.87	100
LR	100	100	97.51	97.79	100
NB	100	100	98.06	98.18	100
DT	100	100	97.70	96.29	100
ANFIS	100	100	98.80	96.50	100
KNN	100	100	98.20	97.94	100

Bold indicates the best results obtained

Table 7 Parameters of the existing classes and the distance from a new observation to the existing classes

Class	Centroid position	P_{thr}	D_{O_2}
Healthy state (E1)	(−5.65, −0.92, −5.65)	0.32	0.10
Drifts in axis one (E2)	(−5.31, −0.91, −5.31)	0.70	5.09
Drifts in axis two (E3)	(−5.16, −0.89, −5.16)	1.14	7.4
Drifts in axis three (E4)	(7.72, 3.03, 7.72)	111.13	211.63
Drifts in axis four (E5)	(−4.1, −0.89, −4.10)	0.50	23.82
Drifts in axis five (E6)	(13.72, 0.78, 13.72)	42.59	300.66
Drifts in axis six (E7)	(−4.54, −1.02, −4.54)	1.42	17.15

umn in Table 7 shows the distance from a new observation of the class E1 (second experiment group) to the centroid of each existing class (first experiment group). One can see that, as $D_{O_2}(E1) < P_{thr}(E1)$, the trained K-NN model is applied to infer the class for the new observation. It correctly identifies that this observation belongs to the healthy state (E1). Similarly, for other observations from the first experiment of the second group, the trained K-NN model can attain the best score with an accuracy of 100%. For an illustration, Fig. 10 shows the distribution of the existing classes and of the new observations (new class 1) from the first experiment of the second group. One can see that the new healthy group is close to the existing healthy state (E1) and far from the other faulty classes (E2, ..., E7) characterizing the drifts of each axis.

Secondly, the data collected from the other experiments in the second group, corresponding to the combined drifts and for which the failure patterns are unknown, are used to verify whether the proposed methodology can detect these anomaly points and then launch the direct monitoring. Considering Table 8, the second column shows the peripheral threshold, P_{thr} , of the existing classes (E1, E2, ..., E7) presented in the first column. The remaining columns indicate the distances from an observation of the unknown classes

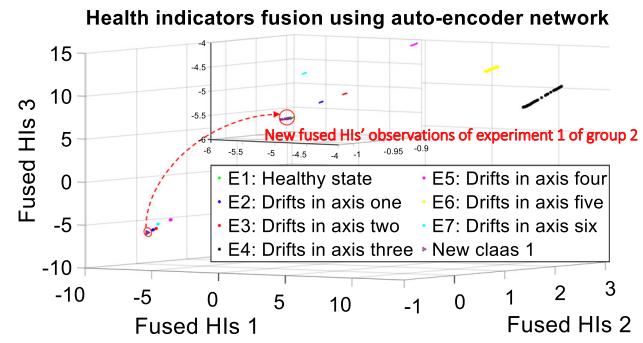


Fig. 10 Distribution of the existing classes and the new observations

(E1 & E4, E2 & E4, ..., E4 & E6) to the centroid of the existing classes. One can see that all of these distances are greater than the P_{thr} of the existing classes. Moreover, Fig. 11 illustrates the graphical distribution of the fused *HIs'* of the remaining experiments of the second group starting for the second one. Each new class does not belong to any of the peripheral threshold of the existing ones. Hence, the proposed methodology considers them as anomaly points and uses the information of the direct monitoring method to label them. The ability of the direct monitoring to correctly localize the drift origins has been highlighted in Sect. 3.2.

Table 8 Distances from new observations (of the combined drifts in the second experiment group) to the centroids of the existing classes (single drifts in the first experiment group)

	P_{thr}	E1 & E4	E2 & E4	E3 & E6	E4 & E5	E2 & E5	E4 & E6
E1	0.32	83.55	79.29	477.54	89.48	78.68	81.84
E2	0.7	50.37	74.97	472.38	82.59	73.04	75.69
E3	1.14	79.39	73.42	469.91	79.64	70.81	73.16
E4	111.13	291.51	277.78	296.81	233.47	260.56	251.17
E5	0.50	73.67	63.80	454.56	58.54	55.86	55.68
E6	42.59	404.76	382.86	345.83	353.60	378.04	369.50
E7	1.42	73.14	64.98	462.18	66.10	59.75	61.01

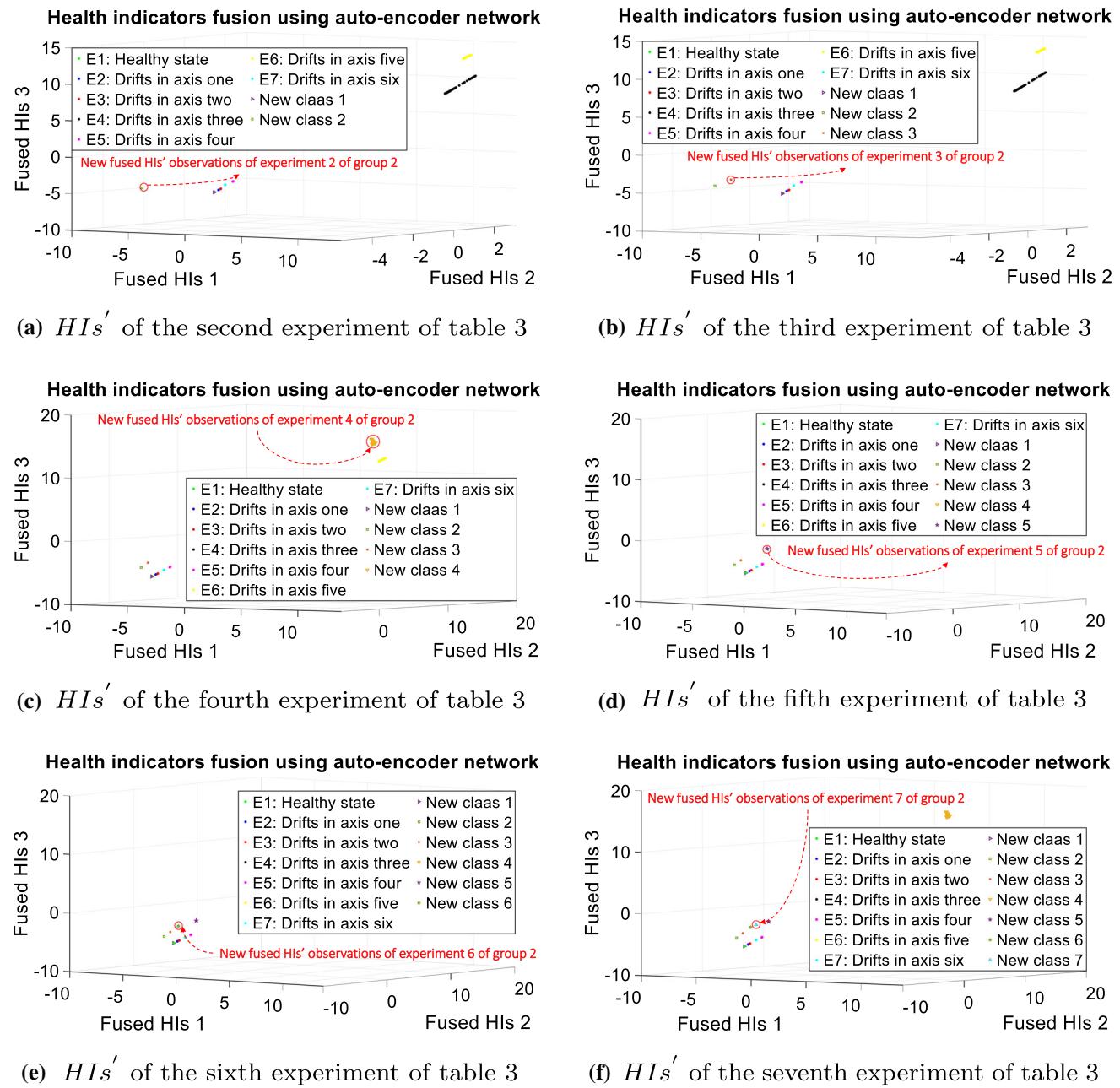


Fig. 11 Online detection of new robot axes drifts

To better understand the process of information fusion, let's consider a particular example consisting of the combined drift (E1 & E4) occurring for the first time. At this moment, the monitoring system is operating under the indirect mode. Hence, the measurements from the sensors placed at the tool level are collected and injected into the signal processing pipeline to create the fused *HIs'*. Then, the distances, D_{O_2} , from this fused *HIs'* observation to the centroids of the existing classes are evaluated to compare them with their peripheral thresholds P_{thr} . As $D_{O_2} > P_{thr}$ for all classes (comparing the third and the second columns of Table 8), the monitoring system switches to the direct mode (but still records the data from the indirect monitoring). After the elapsed time of 2 minutes to access the robot controller and to get the minimum amount of encoder measurements, the monitoring system can assess the health indicator for each robot axis: $Indi_1 = 5.52$, $Indi_2 = 1.03$, $Indi_3 = 1.00$, $Indi_4 = 34.2$, $Indi_5 = 1.08$, $Indi_6 = 1.00$. As $Indi_1$ and $Indi_4$ are greater than 2, the monitoring system can correctly identify the origin of this combined drifts, E1 & E4, and then launches an alarm. Besides, based on the direct monitoring result, the monitoring system labels the observations recorded from the indirect monitoring during the elapsed time of 2 minutes as "E1 & E4". Then, it retrains the classifier of the indirect monitoring with the new observations and new labels, and updates the P_{thr} of each class. The result shows that the P_{thr} of the old classes (E1, E2, ..., E7) do not change while the P_{thr} of the new class E1 & E4 is 0.3.

After updating the classifier and P_{thr} , the monitoring system returns to the indirect mode. Now, assuming that the combined drift (E1 & E4) occurs for the second time. Then, the distances from this new observation to the centroids of the existing classes (E1, E2, E3, E4, E5, E6, E7, and E1 & E4) are evaluated and respectively equal to 33.82, 32.53, 32.13, 118, 29.8, 163.88, 29.59, and 0.04. One can see that $D_{O_2}(E1 \& E4) = 0.04$ is lower than $P_{thr}(E1 \& E4) = 0.3$, hence the classifier is applied on this observation to infer its label. The result shows then that the monitoring system can correctly diagnose this combined drift without switching to the direct mode.

Conclusions and perspectives

An effective and practical methodology for online detection and diagnostics of robot axis drifts has been presented in this paper. It is based on the information fusion of direct and indirect monitoring methods. It allows a continuous monitoring of a machining robot motions using indirect measurements from sensors placed at the tool level to early detect and diagnose the axis drifts, whose failure patterns have been already learned. In addition, it has also the ability to automatically label the unknown drifts thanks to the information from direct

monitoring process using the robot encoder measurements. To illustrate the applicability of the proposed methodology, a real case study using an ABB IRB 6660 robot with six degrees of freedom was used.

The obtained results highlight the flexibility and the effectiveness of the monitoring system using both direct and indirect techniques. Indeed, thanks to the performance and the robustness of the fused HIs constructed from multiple tool-level-sensor signals, the classifier in indirect mode can diagnose the known types of drifts with the best accuracy score of 100%. For uncertain observations, that are outside of all peripheral thresholds of the existing classes, the monitoring system can automatically detect them and use the result of direct monitoring mode to label these observations. It also updates the indirect mode's classifier to quickly detect and correctly diagnose these types of faults when they occur again.

As a limitation of the proposed methodology, the indirect mode is a supervised method and therefore requires more data to learn faults. In addition, for unknown fault types, the performance of the proposed methodology strictly depends on the result of direct monitoring mode, which can be infeasible if there exists a failure of one of the encoders. These issues will be addressed in a future work.

Acknowledgements The project has been 65% cofinanced by the European Regional Development Fund (ERDF) through the Interreg V-A Spain France Andorra programme (POCTEFA 2014-2020). POCTEFA aims to reinforce the economic and social integration of the French-Spanish-Andorran border. Its support is focused on developing economic, social and environmental cross-border activities through joint strategies favouring sustainable territorial development.

References

- Ali, J. B., & Saidi, L. (2018). A new suitable feature selection and regression procedure for lithium-ion battery prognostics. *International Journal of Computer Applications in Technology*, 58(2), 102–115.
- Bhuiyan, M., Choudhury, I., & Dahari, M. (2014). Monitoring the tool wear, surface roughness and chip formation occurrences using multiple sensors in turning. *Journal of Manufacturing Systems*, 33(4), 476–487.
- Cai, W., Wang, J., Jiang, P., Cao, L., Mi, G., & Zhou, Q. (2020). Application of sensing techniques and artificial intelligence-based methods to laser welding real-time monitoring: A critical review of recent literature. *Journal of Manufacturing Systems*, 57, 1–18.
- Chen, X. W., & Nof, S. Y. (2007). Error detection and prediction algorithms: Application in robotics. *Journal of Intelligent and Robotic Systems*, 48(2), 225–252.
- Cho, S., & Jiang, J. (2018). Optimal fault classification using fisher discriminant analysis in the parity space for applications to NPPS. *IEEE Transactions on Nuclear Science*, 65(3), 856–865.
- Fatima, S., Mohanty, A., & Naikan, V. (2015). Multiple fault classification using support vector machine in a machinery fault simulator. In *Vibration engineering and technology of machinery* (pp. 1021–1031). Springer.
- Gotlih, J., Brezocnik, M., Balic, J., Karner, T., Razborsek, B., & Gotlih, K. (2017). Determination of accuracy contour and optimization of

- workpiece positioning for robot milling. *Advances in Production Engineering & Management*, 12(3), 233–244.
- Heinemann, R., Hinduja, S., & Barrow, G. (2007). Use of process signals for tool wear progression sensing in drilling small deep holes. *The International Journal of Advanced Manufacturing Technology*, 33(3–4), 243–250.
- Hsieh, W. H., Lu, M. C., & Chiou, S. J. (2012). Application of backpropagation neural network for spindle vibration-based tool wear monitoring in micro-milling. *The International Journal of Advanced Manufacturing Technology*, 61(1–4), 53–61.
- Jimenez, J. J. M., Schwartz, S., Vingerhoeds, R., Grabot, B., & Salaün, M. (2020). Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. *Journal of Manufacturing Systems*, 56, 539–557.
- Khajavi, M. N., Nasernia, E., & Rostaghi, M. (2016). Milling tool wear diagnosis by feed motor current signal using an artificial neural network. *Journal of Mechanical Science and Technology*, 30(11), 4869–4875.
- Kuric, I., Tlach, V., Ságová, Z., Císař, M., & Gritsuk, I. (2018). Measurement of industrial robot pose repeatability. In *MATEC web of conferences* (Vol. 244, pp. 1–9). EDP Sciences.
- Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3, 18–23.
- Lin, P., & Tao, J. (2019). A novel bearing health indicator construction method based on ensemble stacked autoencoder. In *2019 IEEE international conference on prognostics and health management (ICPHM)* (pp. 1–9). IEEE.
- Loutas, T., Eleftheroglou, N., Georgoulas, G., Loukopoulos, P., Mba, D., & Bennett, I. (2019). Valve failure prognostics in reciprocating compressors utilizing temperature measurements, PCA-based data fusion, and probabilistic algorithms. *IEEE Transactions on Industrial Electronics*, 67(6), 5022–5029.
- Madeti, S. R., & Singh, S. (2018). Modeling of PV system based on experimental data for fault detection using KNN method. *Solar Energy*, 173, 139–151.
- Madhusudana, C., Kumar, H., & Narendranath, S. (2016). Condition monitoring of face milling tool using k-star algorithm and histogram features of vibration signal. *Engineering Science and Technology, an International Journal*, 19(3), 1543–1551.
- Meng, T., Jing, X., Yan, Z., & Pedrycz, W. (2020). A survey on machine learning for data fusion. *Information Fusion*, 57, 115–129.
- Mukherjee, S., & Sharma, N. (2012). Intrusion detection using Naïve Bayes classifier with feature reduction. *Procedia Technology*, 4, 119–128.
- Nguyen, K. T., & Medjaher, K. (2019). A new dynamic predictive maintenance framework using deep learning for failure prognostics. *Reliability Engineering & System Safety*, 188, 251–262.
- Ogedengbe, T. I., Heinemann, R., Hinduja, S., et al. (2011). Feasibility of tool condition monitoring on micro-milling using current signals. *AU JT*, 14(3), 161–172.
- Pivoto, D. G., de Almeida, L. F., da Rosa, Righi R., Rodrigues, J. J., Lugli, A. B., & Alberti, A. M. (2021). Cyber-physical systems architectures for industrial internet of things applications in industry 4.0: A literature review. *Journal of Manufacturing Systems*, 58, 176–192.
- Qiao, G., & Weiss, B. A. (2017). Accuracy degradation analysis for industrial robot systems. In *Proceedings of ASME international manufacturing science and engineering conference* (pp. 1–9).
- Qiao, G., Schlenoff, C., & Weiss, B. A. (2017). Quick positional health assessment for industrial robot prognostics and health management (PHM). In *2017 IEEE international conference on robotics and automation (ICRA)* (pp. 1815–1820). IEEE.
- Qiao, G., & Weiss, B. A. (2018). Quick health assessment for industrial robot health degradation and the supporting advanced sensing development. *Journal of Manufacturing Systems*, 48, 51–59.
- Ratava, J., Lohtander, M., & Varis, J. (2017). Tool condition monitoring in interrupted cutting with acceleration sensors. *Robotics and Computer-Integrated Manufacturing*, 47, 70–75.
- Samantaray, S. (2009). Decision tree-based fault zone identification and fault classification in flexible AC transmissions-based transmission line. *IET Generation, Transmission & Distribution*, 3(5), 425–436.
- Segreto, T., Karam, S., Teti, R., & Ramsing, J. (2015). Feature extraction and pattern recognition in acoustic emission monitoring of robot assisted polishing. *Procedia CIRP*, 28, 22–27.
- Selvaraj, D. P., Chandramohan, P., & Mohanraj, M. (2014). Optimization of surface roughness, cutting force and tool wear of nitrogen alloyed duplex stainless steel in a dry turning process using Taguchi method. *Measurement*, 49, 205–215.
- Shi, X., Wang, X., Jiao, L., Wang, Z., Yan, P., & Gao, S. (2018). A real-time tool failure monitoring system based on cutting force analysis. *The International Journal of Advanced Manufacturing Technology*, 95(5–8), 2567–2583.
- Soualhi, M., Nguyen, K., Medjaher, K., Lebel, D., & Cazaban, D. (2019a). Health indicator construction for system health assessment in smart manufacturing. In *2019 prognostics and system health management conference (PHM-Paris)* (pp. 45–50). IEEE.
- Soualhi, M., Nguyen, K. T., & Medjaher, K. (2020). Pattern recognition method of fault diagnostics based on a new health indicator for smart manufacturing. *Mechanical Systems and Signal Processing*, 142, 1–20.
- Soualhi, M., Nguyen, K. T., Soualhi, A., Medjaher, K., & Hemsas, K. E. (2019b). Health monitoring of bearing and gear faults by using a new health indicator extracted from current signals. *Measurement*, 141, 37–51.
- Terrazas, G., Martínez-Arellano, G., Benardos, P., & Ratchev, S. (2018). Online tool wear classification during dry machining using real time cutting force measurements and a CNN approach. *Journal of Manufacturing and Materials Processing*, 2(4), 1–18.
- Tobon-Mejia, D., Medjaher, K., & Zerhouni, N. (2012). CNC machine tool's wear diagnostic and prognostic by using dynamic Bayesian networks. *Mechanical Systems and Signal Processing*, 28, 167–182.
- Wang, Y., Zheng, L., & Wang, Y. (2021). Event-driven tool condition monitoring methodology considering tool life prediction based on industrial internet. *Journal of Manufacturing Systems*, 58, 205–222.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.