



RUL prediction for automatic machines: a mixed edge-cloud solution based on model-of-signals and particle filtering techniques

Matteo Barbieri¹ · Khan T. P. Nguyen² · Roberto Diversi¹ · Kamal Medjaher² · Andrea Tilli¹

Received: 12 May 2020 / Accepted: 14 October 2020 / Published online: 4 November 2020
© The Author(s) 2020

Abstract

This work aims to provide useful insights into the course of action and the challenges faced by machine manufacturers when dealing with the actual application of Prognostics and Health Management procedures in industrial environments. Taking into account the computing capabilities and connectivity of the hardware available for smart manufacturing, we propose a particular solution that allows meeting one of the essential requirements of intelligent production processes, i.e., autonomous health management. Indeed, efficient and fast algorithms, that does not require a high computational cost and can be appropriately performed on machine controllers, i.e., on edge, are combined with others, which can handle large amounts of data and calculations, executed on remote powerful supervisory platforms, i.e., on the cloud. In detail, new condition monitoring algorithms based on Model-of-Signals techniques are developed and implemented on local controllers to process the raw sensor readings and extract meaningful and compact features, according to System Identification rules and guidelines. These results are then transmitted to remote supervisors, where Particle Filters are exploited to model components degradation and predict their Remaining Useful Life. Practitioners can use this information to optimise production planning and maintenance policies. The proposed architecture allows keeping the communication traffic between edge and cloud in the nowadays affordable “Big data” range, preventing the unmanageable “Huge data” scenario that would follow from the transmission of raw sensor data. Furthermore, the robustness and effectiveness of the proposed method are tested considering a meaningful benchmark, the PRONOSTIA dataset, allowing reproducibility and comparison with other approaches.

Keywords Condition monitoring · Automatic machines · Edge-computing · System identification · Prognostics and health management · Particle filtering

Introduction

The introduction of emerging technologies, increasing computer computational capabilities and storage, e.g., Big-Data analysis tools and Artificial Intelligence, into the industrial world is enabling new models, new forms, and new methodologies to transform the traditional manufacturing system into a smart system. Among those, Prognostics and Health Management (PHM) of machines has risen, in recent years, as one of the main topics within Industry 4.0 and a relevant factor in firms adopting the main concepts of Smart Factory and Intelligent Manufacturing. Systems with the possibility to autonomously perform the diagnostics and prognostics of faulty conditions are now becoming a source of value and competitive advantage for machine builders and an essential requirement for their customers.

With a focus on the industrial automation and manufacturing domain, the involved research field has proposed a wide

✉ Matteo Barbieri
matteo.barbieri15@unibo.it

Khan T. P. Nguyen
tnguyen@enit.fr

Roberto Diversi
roberto.diversi@unibo.it

Kamal Medjaher
kamal.medjaher@enit.fr

Andrea Tilli
andrea.tilli@unibo.it

¹ Department of Electrical, Electronic and Information Engineering “Guglielmo Marconi” (DEI), University of Bologna, Viale del Risorgimento 2, Bologna, Italy

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

variety of tools and solutions to apply PHM on machinery (Gouriveau et al. 2016; Atamuradov et al. 2017; Vogl et al. 2019). These propositions are typically accurately defined from a methodological perspective, however, lacking in terms of actual design and implementation in the industrial environment. On the other hand, Industry 4.0 is pushing industrial automation suppliers to develop capable hardware, suitable software tools, and ensure reliable inter-system communication to support intelligent manufacturing. The main issue, in this case, is in the separation between the methodological aspect and the technological one in the industrial application of PHM. Hence, these two aspects should be simultaneously considered and developed.

In practice, the increase in computing capacity of Programmable Logic Controllers (PLCs), particularly in the case of PC-based ones, allows the integration of hardware and software solutions to perform PHM on traditional machinery. The intrinsic real-time features of PLCs already permit to handle, through Fieldbus, a large number of sensor measurements to capture information coming from working machine components. In addition, the increased computational power enables controllers to possibly house part of the processing required for PHM, working as refining edge-computing units. Besides, the increased connectivity allows the outsourcing of further elaborations to remote computing units, which can be either within the machine supervising network or through cloud-computing.

Given those considerations, PHM methodologies have to be studied and structured to take into account the possibility of adopting PLCs as edge-computing units running feasible information extraction/refinement algorithms. Then, more complex calculations can be remotely performed to interpret the useful health indicators for maintenance decision-making. Lastly, it is necessary to ensure that the interconnection of the systems involved is sustainable by the automation pyramid underlying network. The data flow, therefore, is feasible for the already in place networking solutions, such as Fieldbuses and supervising LANs. Also, manufacturers have the advantage of developing methodologies on known industrial platforms, based on their expertise.

For the purposes mentioned above, we firstly present a brief review of the commonly used PHM methodologies in automatic machines and secondly propose a suitable solution tailored to the available technologies mentioned above, taking into account their potential and limitations. We will develop a particular PHM course of actions in which data regarding the machinery degradation is collected and refined locally on board the controller. Then, this information is remotely transmitted for further elaborations and presented for maintenance purposes.

In detail, Model-of-Signals (MoS) technique, which it is suitable for PLC implementation (Barbieri 2017; Barbieri et al. 2018), is firstly implemented to obtain useful infor-

mation from sensor data. This technique is a data-driven method based on system identification, granting useful theoretical foundations for signal information extraction. In this scenario, the ability to process data locally enables the compression of measurement streams (i.e., Huge-Data) into compact pieces of information (i.e., Big-Data), which are easier to handle and transmit over the network. Secondly, following PLC-Supervisor/Cloud data transmission, Health Indicators (HIs) are computed from the obtained signal models. Finally, Particle Filtering (PF) is adopted to track the degradation evolution and to predict the system's Remaining Useful Life (RUL), which is one of the main indicators to optimise maintenance.

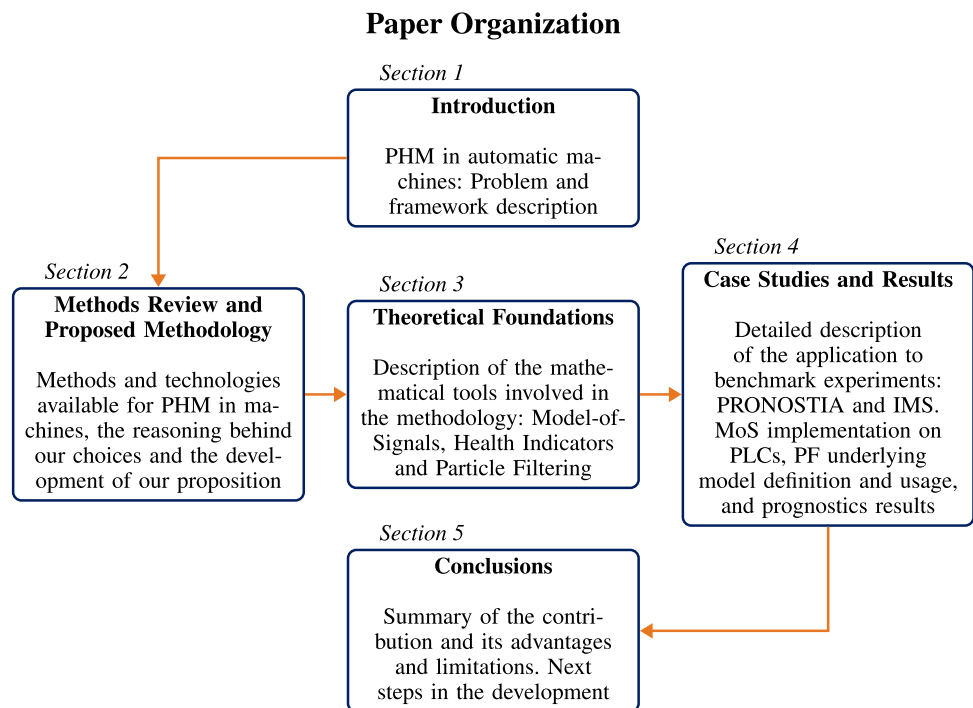
Furthermore, the applicability of the proposed methodology is validated using PRONOSTIA and IMS benchmarking data sets, taken from the NASA repository. PLC-Supervisor architecture is simulated to execute the various task involved. Even if it is not a real industrial application, we expect that making use of an open-source database to develop our discussion may be useful for practitioners willing to add PHM to their machinery with a repeatable example. Besides, the algorithm Prognostic Horizon (PH) performance metric is computed, allowing us to compare our results with other methodologies.

The remainder of the paper is organized as follows: in “Methods review and proposed methodology” section we give a brief review of the available PHM techniques and maintenance policies to highlight the proposed methodology. Then, in “Theoretical foundations” section, the theoretical background used to develop our proposition, from Model-of-Signals to Particle Filtering, is introduced. “Case studies and results” section aims to show a detailed application of the methodology to the PRONOSTIA and IMS open-source datasets and discusses the obtained results in terms of performance. Finally, “Conclusions” section draws the conclusions of the proposed work. A scheme of the paper organisation is shown in Fig. 1.

Methods review and proposed methodology

Tracking equipment condition during operations, on-board the system, is known as Condition Monitoring (CM). This technique allows the implementation of autonomous PHM solutions that provide useful information to perform suitable servicing decisions. The servicing strategies, based on CM of machinery parts, are referred to as Condition-Based Maintenance (CBM) (Jardine et al. 2006) and Predictive Maintenance (PM). The former is usually triggered when a device/component reaches a certain level of degradation, while the latter depends on the component/device's predicted level of degradation. This paper is dedicated to proposing a new prognostics solution that provides useful information

Fig. 1 Schematic of paper organisation



for optimisation of PM policies. In general, an autonomous health management procedure consists of the following three main steps:

1. data acquisition;
2. data processing, modelling, and analysis;
3. maintenance decision-making.

The application of Autonomous Health Management (AHM) procedures on machinery components require significant sensor measurements, suitable data processing algorithms, and appropriate servicing choices, either automated or with human intervention. In the literature, the majority of papers cover fault diagnostics and prognostics algorithms of machine's critical components, such as bearings, gears, drive mechanical parts, and electrical equipment (Lee et al. 2014). However, most of the existing works does not take into account systems available to manufacturers to perform the task, neither their implementation and deployment in industrial automation platforms. Hereinafter, the three previously mentioned AHM steps, and their main tools, are discussed to select suitable solutions that respond to the industrial automation framework's requirements.

Data acquisition

Machinery can typically provide measurement data from on-board sensors, such as temperature, current, sound and vibration, and supervisory data, such as information about products, production, and faults history. The latter is so far

the most used to define the reliability of systems and their servicing (Lee et al. 2014). Besides, the use of on-board sensors to gather data characterizing the machinery health state is the foundation of CM-based PHM procedures (Jardine et al. 2006). This data collection procedure is traditionally performed on external equipment (e.g., National Instruments DAQs). However, newly available PLCs I/O modules, such as Beckhoff EL3632 and B&R X20CM4800X, can also handle the same operation. In this fashion, to develop an AHM solution on-board, we propose the machinery controller as the edge-computing unit to process the sensor measurements for further elaborations without additional external devices.

Data processing, modelling and analysis

Data processing, modelling, and analysis, which aims to extract useful information from sensor signals, has a key role in the definition of what PHM can provide for the decision-making stage. The methods employed to achieve this task are usually classified into three groups, depending on how much they exploit (mathematically) the physical knowledge related to the monitored system:

- Model-Based methods.
- Data-Driven methods.
- Hybrid methods.

Model-Based methods (Isermann 2005) rely on physical modelling to build mathematical approximations, of increasing degree of complexity, to characterise the system

input/output behaviour, together with expressions characterising the degradation evolution of systems. These methods are particularly effective in terms of diagnostics and prognostics of faults. However, its complexity may result prohibitive for edge-computing in the automatic machines field. Machinery controllers' common programming languages have no libraries nor tools to run physical models in parallel with the system. On the other hand, Data-Driven methods (Cerrada et al. 2018) exploit signals measured on-board to perform CM, mainly through signal processing and machine learning techniques. The implementation of such strategies is simpler and requires, in general, less time and resources. Finally, hybrid methods combine the previously mentioned ones, a promising solution for the development of autonomous health management systems.

In this paper, the Model-of-Signals method (Isermann 2006), based on black-box system identification theory (Söderström and Stoica 1989), is used to track the signal changes, and we experimented it in the context of CBM (Barbieri et al. 2018, 2019a, b, 2020). This method can be considered as a hybrid one because it relies on mathematical expression to model the signal trend and also use the monitoring data to estimate the model parameters. In practice, it allows facilitating the integration of PHM modules into the on-board controller thanks to the following main properties:

- the proposed method allows compressing signal information into models, that are easier to handle, and facilitates distributed computing to increase the computational capacities in real-time.
- its developed mathematical models can carry inherent information about system physical characteristics, that is useful information to improve the diagnostics and prognostics results.
- the availability of its recursive algorithms permit its direct implementation on the PLC, hence exploiting its edge-computing capabilities.

Maintenance decision-making

This step relies on the processed data to drive maintenance decisions depending on the systems state. In this context, decision-making can be done either by human operators or with automated solutions. The latter nowadays becomes a fascinating subject that receives increasing attention in the literature. For autonomous maintenance decision making, the refined data from the previous steps are used to predict system faults, described with failure probabilities or Remaining Useful Life (RUL) indications. Then, defining a threshold on those quantities, it is possible to trigger an automated alarm and to launch an optimisation process for maintenance decision-making.

The key element of the autonomous health management process, RUL prediction, can be performed based on various data-driven techniques (Si et al. 2011), from statistics to machine learning. Among these techniques, Particle Filtering is an efficient method that is widely used to accomplish the RUL prediction task (Tulsyan et al. 2016; Wang et al. 2019). This method allows building the degradation model of the monitored system and then, based on this result, forecasting its RUL.

Proposed methodology

Smart manufacturing sites typically use PLCs to handle the production process and PCs to supervise their work. The technological development of those devices enables the integration of autonomous health management solutions alongside logic control tasks. Given these considerations, together with the previously mentioned brief review, we propose an efficient and practical methodology that allows integrating automated PHM modules in machine controllers, using standard industrial platforms and architectures. The proposed solution, presented in Fig. 2, is based on a widely used architecture by industrial manufacturers: PC-supervised PLCs connected via LAN. The PLCs ensure the edge-computing unit role, providing the monitoring equipment and the first refinement of the collected data. The PC supervisor is the remote computing unit providing the final data processing to extract reliable prognostics information for maintenance-decision making purposes.

Besides the logic control task, the PLC is capable of handling the measured signals (e.g., currents, temperatures, and vibrations), coming from the on-board sensing devices, and process them (see the left side of Fig. 2). In detail, the Model-of-Signals technique, implemented on PLCs, allows extracting useful information from sensor measurements. It compresses huge-data streams into models that retain the majority of the inherent signal characteristics. Its implementation can be easily performed based on main coding languages used to program machinery controllers, such as Structured Text (ST) (Barbieri 2017).

Then, the estimated model's parameters are used to calculate Health Indicators (HIs), giving an indication of the health state of monitored parts on top of which the signals are measured. Their elaboration can be done locally or remotely, depending on the complexity of the computations and related loads. In this work, we propose the construction of both a local HI and a remote HI (see "Anomaly detection and health indicator construction" section for the details). The first one serves as a backup for the worst-case scenario of a LAN disconnection. It allows continuous monitoring of the system's health state when it is impossible to receive information from the supervisor. The second one is the primary indicator used for health state prediction and maintenance decision-making.

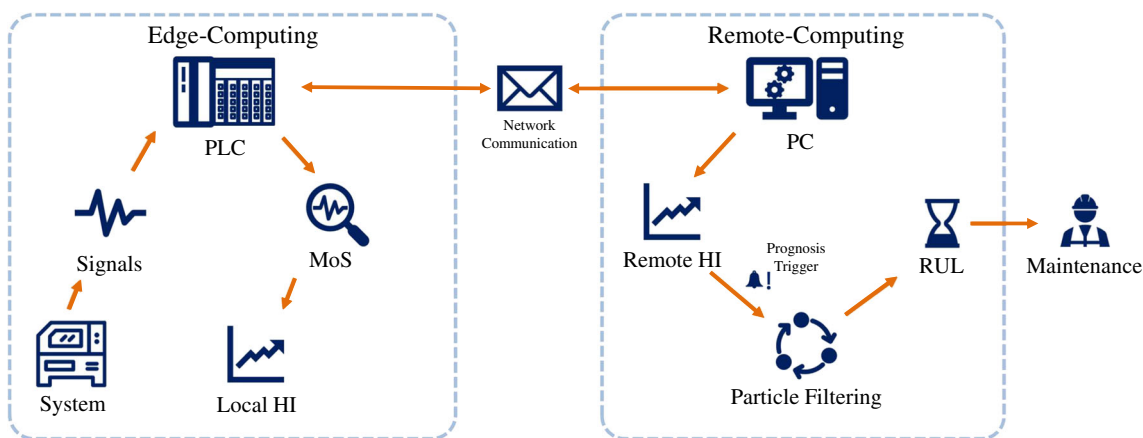


Fig. 2 Schematic of the overall methodology

The edge-computed model, built based on the Model-of-Signals technique, is sent over LAN to the remote computing unit, see Fig. 2 on the right side. This remote computing unit enables complex calculations that require higher computational resources for a reliable RUL prediction. To do that, the PC computes the related HI and monitors its evolution in time. Machines typically work under nominal conditions for a long time before an anomaly occurs in their components. During those healthy conditions, it is not necessary to perform prognostics. Hence, it is required the identification of a HI threshold $T_{h_{prog}}$ for anomaly detection, which is related to the moment when the degradation of the monitored systems starts. This event triggers the execution of the prognostics task performed by Particle Filtering. The PF is used to learn the degradation model and, consequently, predict its evolution toward the failure threshold. The definition of this second threshold is essential to evaluate the monitored system’s Remaining Useful Life (RUL). Finally, this information provides valuable indications for maintenance optimisation and decision-making.

The next section intends to describe in detail the theoretical foundations of the proposed methodology and also present the mathematical tools used to accomplish it.

Theoretical foundations

Model-of-signals

Model-of-Signals (MoS) technique uses mathematical expressions, whose parameters are estimated from the available data, to model the measured signals (Isermann 2006). Those mathematical expressions are, typically, constructed based on black-box system identification theory (Ljung 1999; Söderström and Stoica 1989), which provides essential rules and guidelines to derive the appropriate signal model. This

model and its variation in time can then be used as the primary source of information for fault diagnostics and prognostics methods.

Figure 3 shows a summary of the main steps to execute MoS. It starts with the sampling of the available signal for which the appropriate underlying model parameter structure and the proper model order are selected. Then, the estimation algorithm is defined depending on the picked model structure. In the end, the model’s parameters are obtained by injecting the sampled signal into the chosen estimation algorithm. Hereinafter, we present in detail how to apply this procedure for the case of bearings, which is the case study we are going to analyse later.

Vibration signals are used to track bearing degradation and modelled as AutoRegressive (AR) processes. Their identification is efficiently performed by means of the Recursive Least Squares (RLS) algorithm (Ljung 1999; Söderström and Stoica 1989). As shown in (Barbieri 2017; Barbieri et al. 2018, 2019a), the RLS implementation on board the PLC is feasible because of its recursive nature and requires the definition of a library containing the involved matrix operations. In this paper, we apply the Overdetermined Recursive Instrumental Variable (ORIV) (Friedlander 1984), which is an enhanced version of RLS that uses more equations to increase model estimation robustness.

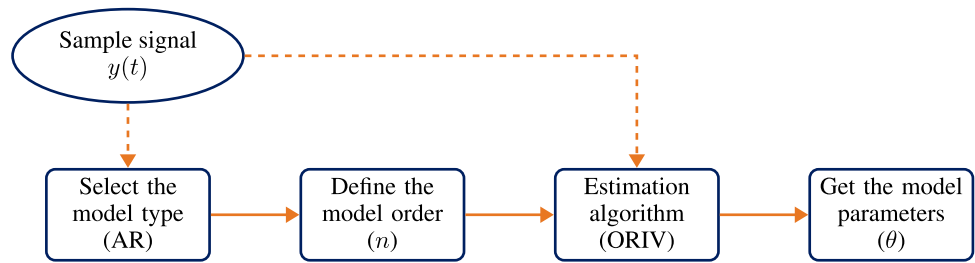
To introduce the algorithm, we assume the evolution of the vibrational signal $y(t)$ as modelled by the following AR process of order n :

$$y(t) + a_1 y(t-1) + a_2 y(t-2) + \dots + a_n y(t-n) = e(t) \quad (1)$$

or in its discrete-time transfer function form:

$$y(t) = \frac{1}{A(z^{-1})} e(t) \quad (2)$$

Fig. 3 Summary of model-of-signals



where the driving noise $e(t)$ is a zero-mean white process with variance σ_e^2 , n is the model order and $A(z^{-1})$ is a polynomial in the backward shift operator z^{-1} ($z^{-1}y(t) = y(t-1)$).

Equation (1) can be rewritten in the linear regression form

$$y(t) = \varphi_y^T(t)\theta + e(t) \tag{3}$$

with

$$\varphi_y(t) = [-y(t-1) \dots -y(t-n)]^T \tag{4}$$

$$\theta = [a_1 \dots a_n]^T. \tag{5}$$

Define also the following vector of instruments:

$$\bar{\varphi}_y(t) = [-y(t-1) \dots -y(t-n-q)]^T, \tag{6}$$

with $q \geq 0$. Starting from the set of available samples $y(1), y(2), \dots, y(N)$, it is possible to compute the following extended instrumental variable (IV) estimate (Söderström and Stoica 1989)

$$\hat{\theta} = \hat{R}^+ \hat{\rho}, \tag{7}$$

where

$$\hat{R} = \sum_{\tau=\tau_0}^N \bar{\varphi}_y(\tau)\varphi_y^T(\tau), \quad \hat{\rho} = \sum_{\tau=\tau_0}^N \bar{\varphi}_y(\tau)y(\tau), \tag{8}$$

$\tau_0 = n + q + 1$, and \hat{R}^+ denotes the pseudoinverse of the matrix \hat{R} . Note that, if $q = 0$ we have $\bar{\varphi}_y(\tau) = \varphi_y(\tau)$ so that the estimate (7) becomes the classic Least Squares (LS) estimate $\hat{\theta} = \hat{R}^{-1} \hat{\rho}$.

Given those premises, it is possible to introduce the recursive version of the extended IV algorithm, as proposed in (Friedlander 1984) where it is called Overdetermined Recursive Instrumental Variable (ORIV). The ORIV coding is attainable in PLC software environment. However, ORIV requires some slight modifications from the RLS implementation that we proposed in (Barbieri 2017). Equation (7) is rewritten with \hat{R}^+ expanded as $\hat{P}(t)\hat{R}^T(t)$:

$$\hat{\theta}(t) = \hat{P}(t)\hat{R}^T(t)\hat{\rho}(t) \tag{9}$$

where

$$\hat{\rho}(t) = \sum_{\tau=\tau_0}^t \bar{\varphi}_y(\tau)y(\tau) \tag{10}$$

$$\hat{R}(t) = \sum_{\tau=\tau_0}^t \bar{\varphi}_y(\tau)\varphi_y^T(\tau) \tag{11}$$

$$\hat{P}(t) = [\hat{R}^T(t)\hat{R}(t)]^{-1}. \tag{12}$$

Then, following the computations done in (Friedlander 1984), the recursive version of the algorithm is given below.

Algorithm 1 (ORIV)

1. $\eta(t) = R^T(t-1)\bar{\varphi}_y(t)$
2. $\phi(t) = (\eta(t) \quad \varphi_y(t))$
3. $\Lambda(t) = \begin{pmatrix} -\bar{\varphi}_y^T(t)\bar{\varphi}_y(t) & 1 \\ 1 & 0 \end{pmatrix}$
4. $v(t) = \begin{pmatrix} \bar{\varphi}_y^T(t)\rho(t-1) \\ y(t) \end{pmatrix}$
5. $K(t) = P(t-1)\phi(t) [\Lambda(t) + \phi^T(t)P(t-1)\phi(t)]^{-1}$
6. $\hat{\theta}(t) = \hat{\theta}(t-1) + K(t) [v(t) - \phi^T(t)\hat{\theta}(t-1)]$
7. $R(t) = R(t-1) + \bar{\varphi}_y(t)\varphi_y^T(t)$
8. $\rho(t) = \rho(t-1) + \bar{\varphi}_y(t)y(t)$
9. $P(t) = P(t-1) - K(t)\phi^T(t)P(t-1)$

The initial step may be defined in the following way

$$\begin{aligned} \hat{\theta}(0) &= 0, & P(0) &= \psi I_n, \\ \rho(0) &= 0, & R(0) &= 0, \end{aligned} \tag{13}$$

with ψ any large positive number.

Note that the algorithm requires no inversion of an $n \times n$ matrix, which is highly computationally demanding for a PLC, whereas the 2×2 matrix inversion at step (5) has fixed dimensions and well-known implementation.

Model order selection

The use of AR processes to represent the measured signals requires to define a proper model order n , see equation (1).

Two criteria that are often used for model order selection are the Final Prediction Error (FPE) and the Minimum Description Length (MDL) (Söderström and Stoica 1989). They are typically based on the statistical properties of the residuals of the LS identification. However, in this case we will make use of the ORIV identification residuals. Considering an AR model of order n and the associated parameter vector $\hat{\theta}_n$ identified by applying the ORIV method to a set of N measurements: $y(1), y(2), \dots, y(N)$, FPE and MDL are criteria with complexity terms that consist in selecting the order n leading to the minimum of the following loss functions (Söderström and Stoica 1989; Ljung 1999):

$$FPE(n) = \frac{N+n}{N-n} J(\hat{\theta}_n) \tag{14}$$

$$MDL(n) = N \log(J(\hat{\theta}_n)) + n \log N, \tag{15}$$

where

$$J(\hat{\theta}_n) = \frac{1}{N} \sum_{t=1}^N \varepsilon^2(t, \hat{\theta}_n), \tag{16}$$

and $\varepsilon(t, \hat{\theta}_n) = y(t) - \varphi^T(t) \hat{\theta}_n$ is the residual (prediction error) obtained after the ORIV identification. In this work, the choice of the order n is performed by combining the FPE and MDL criteria applied to the residual sequence $\varepsilon(1, \hat{\theta}_n), \dots, \varepsilon(N, \hat{\theta}_n)$.

Anomaly detection and health indicator construction

Health Indicators (HI) are obtained by computing model distance metrics. They permit to evaluate how far the current estimated model $\hat{\theta}(t)$, at some time point in time t , is from the reference one $\hat{\theta}_{nom}$, identified during nominal healthy conditions. HI time evolution is used either for fault detection or prediction depending on its behaviour.

Figure 4 summarizes the HI construction procedures proposed in this paper. We make use of two HIs: the first is directly evaluated on the PLC while the second is calculated on the PC supervisor. In detail, the Normalized Root Mean Square Error (NRMSE), which is effectively implementable on PLCs, is chosen to evaluate the distance between reference and current model:

$$HI_{NRMSE} = \sqrt{\frac{\|\hat{\theta} - \hat{\theta}_{nom}\|}{\|\hat{\theta}_{nom}\|}}, \tag{17}$$

This proposed HI is useful to prevent the system from reaching a non-return failure point. Especially, when the controller and the supervisor are not able to communicate it acts as a

last resort solution used to avoid unwanted damages. Indeed, it is a positive index that allows fault detection when exceeding a failure threshold. However, its high non-linear and non-smooth behaviours can lead to non-reliable prognostics results. Hence, it is necessary to simultaneously develop another HI on the PC supervisor to derive precise and robust prognostics results. This health indicator is calculated based on a symmetric version of the Itakura-Saito spectral distance (Itakura 1968; Magnant et al. 2014):

$$HI_{I-S} = \frac{1}{2} \left(D_{I-S}(\theta|\theta_{nom}) + D_{I-S}(\theta_{nom}|\theta) \right) \tag{18}$$

with

$$D_{I-S}(\theta^1|\theta^0) = \frac{\text{spec}(\theta^1)}{\text{spec}(\theta^0)} - \log \left(\frac{\text{spec}(\theta^1)}{\text{spec}(\theta^0)} \right) - 1, \tag{19}$$

where $\text{spec}(\theta^i)$ represents the Power Spectral Density (PSD) computed from the parameters θ^i of the AR model.

This HI, that typically can not be evaluated on PLCs because of its computational complexity, is used for RUL prediction purposes because it is well correlated over time with the degradation process of the system and can be easily modelled (i.e. its evolution fits simple exponential functions, see “Remote-computing on PC” section). In this framework, prognostics is triggered when this indicator exceeds the anomaly threshold, pointing out the starting of the degradation. Then, the RUL is computed by means of a upper limit failure threshold on the HI_{I-S} evolution prediction. In the next section we describe the tool used to perform this prediction: Particle Filtering (PF).

Particle filtering

Particle Filtering (PF) is based on the recursive Bayesian estimation framework (Gordon et al. 1993). It starts by considering the following state space model:

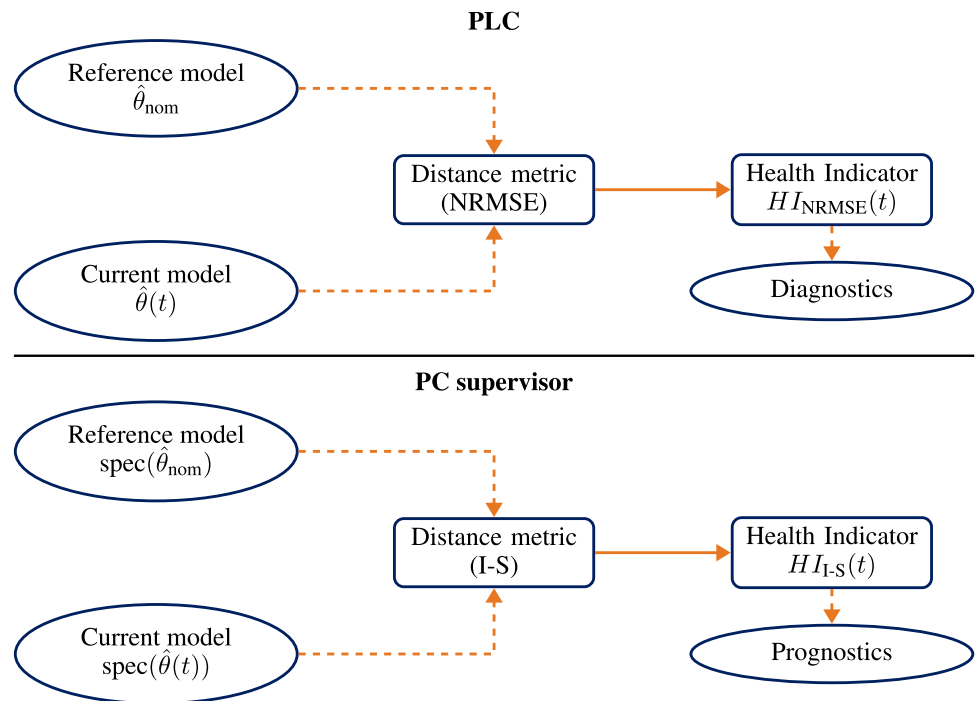
$$x_k = f(x_{k-1}, \omega_{k-1}) \tag{20}$$

$$y_k = h(x_k, v_k), \tag{21}$$

where x_k is the state vector; $f(\cdot)$ is the possibly non-linear state transition function; $h(\cdot)$ is the state to output mapping; ω_k and v_k are the independent identically distributed process and measurement noise respectively with known statistics; and $k \in \mathbb{N}$ index refers to the time step.

The goal is to reconstruct the Probability Density Function (PDF) of the current state x_k , defined as $p(x_k)$, exploiting the information collected through the observations sequence y_1, y_2, \dots, y_k , denoted $y_{1:k}$ compactly, given the conditional PDF $p(x_k|y_{1:k})$. This posterior probability can be obtained by means of a recursive procedure employing two main

Fig. 4 HI construction procedure implemented on PLC and PC supervisor



stages: prediction and update. Given the initial state PDF $p(x_0) = p(x_0|y_0)$, we start the description of the aforementioned stages at time $k - 1$, given the conditional PDF $p(x_{k-1}|y_{k-1})$.

The prediction stage makes use of Eq. (20) to obtain the prior PDF of the state vector at step k . This is used within the Chapman-Kolmogorov equation:

$$p(x_k | y_{1:k-1}) = \int p(x_k|x_{k-1}, y_{1:k-1}) p(x_{k-1}|y_{1:k-1}) dx_{k-1} \tag{22}$$

where if we assume the underlying process to be a first-order Markov one:

$$p(x_k|x_{k-1}, y_{1:k-1}) = p(x_k|x_{k-1}), \tag{23}$$

we get the following equation:

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1}) p(x_{k-1}|y_{1:k-1}) dx_{k-1}. \tag{24}$$

Then, once the new observation y_k is available it is possible to perform the update stage by using the Bayes' Rule:

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k) p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} \tag{25}$$

to get the posterior PDF of the state. The normalizing constant

$$p(y_k|y_{1:k-1}) = \int p(y_k|x_k) p(x_k|y_{1:k-1}) dx_k \tag{26}$$

depends on the likelihood function $p(y_k|x_k)$ which is based on (21). The recursion of Eqs. (22) and (25) pave the way to the optimal solution of this Bayesian framework.

Actually, the Bayesian solution is just a conceptual one, since, due to its complexity, it is possible to attain it analytically only under certain assumptions or conditions, such as Kalman Filters where all posterior PDFs are assumed to be Gaussian. Because of this reason, sub-optimal filters are required in order to at least approximate the solution.

One of the most used sub-optimal filters is the Particle Filter (PF), which is based on the Sequential Monte Carlo (SMC) method. The main idea of this filter is to exploit recursively MC simulations to represent the required posterior density function by a set of random samples with associated weights and to compute estimates based on these samples and weights. A suitably large number of samples, called particles in this case, is able to characterize an equivalent representation of the posterior PDFs involved in the framework. A thorough explanation of such methods is presented in Arulampalam et al. (2002).

The particle filtering approach proposed in this paper is Sampling Importance Resampling (SIR), which is used widely in the prognostic field (Tulsyan et al. 2016; An et al. 2013; Saha and Goebel 2011; Skima et al. 2016), and makes

use of N_p weighted particles:

$$\{x_k^i, w_k^i\}; i = 1, \dots, N_p\}, \text{ compactly } \{x_k^i, w_k^i\}_{i=1}^{N_p}, \quad (27)$$

to approximate $p(x_k|y_{1:k})$, where w_k^i is the weight of particle i at time k .

Starting from the initial distribution $p(x_0)$, approximated by the sequence $\{x_0^i\}_{i=1}^{N_p}$ with uniform weights $\{w_0^i = \frac{1}{N_p}\}_{i=1}^{N_p}$, we summarize the steps to accomplish SIR Particle Filtering within the recursive Bayesian filtering framework in the following:

Algorithm 2 (SIR PF)

1. **Prediction:** compute the new a priori PDF of the state x_k by propagating through model described by Eq. (20) the particle set from $\{x_{k-1}^i\}_{i=1}^{N_p}$ to $\{x_k^i\}_{i=1}^{N_p}$. In this way, the approximation of $p(x_k|x_{k-1})$, which is the state transition PDF, is obtained.
2. **Update:** once the new measurement y_k is available, the likelihood function $p(y_k|x_k)$ is exploited to compute importance weights according to:

$$\left\{ w_k^i = \frac{p(y_k|x_k^i)}{\sum_{i=1}^{N_p} p(y_k|x_k^i)} \right\}_{i=1}^{N_p} \quad (28)$$

3. **State estimation:** the estimate is given by the particle mean value:

$$\hat{x}_k = \frac{1}{N_p} \sum_{i=1}^{N_p} x_k^i w_k^i \quad (29)$$

4. **Re-sample:** Re-sample from $\{x_k^i, w_k^i\}_{i=1}^{N_p}$, proportionally to the new computed weights, the new sequence of particles $\{x_k^i\}_{i=1}^{N_p}$ is obtained. This sequence distribution is recursively approximated by $p(x_k|y_k)$.

Fault prognostics

In the previous section we described how to employ SIR Particle Filtering to estimate the system state x_k and its unknown parameters based on the collected observation y_k at time step k (instant t_k in time units, from now on). This means that, if we apply it to the degradation model of the studied system, we can learn, through its HI observations, the underlying system failure evolution. The left column of Fig. 5 refers to this course of actions: the learning stage.

Then, using this knowledge, we are able to propagate the posterior PDF to forecast the degradation state in the future. The simplest way to perform prognostics in this framework is, at a given moment in time t_k , to reiterate the prediction

step of Algorithm 2 by propagating the particles $\{x_k^i\}_{i=1}^{N_p}$ until x_k^i (or also y_k^i , obtained through Eq. (21)) reaches the failure threshold T_{hEoL} at the End-of-Life (EoL) time t_{EoL}^i . Then, the Remaining Useful Life PDF is obtained from the distribution of those failing times t_{EoL}^i with respect to t_k , i.e. $p(t_{EoL}^i - t_k)$. This is summarized in the right column of Fig. 5: the prediction stage.

In this paper, the HI_{I-S} evaluated on the PC supervisor is used for system health state estimation and RUL prediction. PF learning phase will start as soon as the HI exceeds the prognostics trigger threshold, while the RUL prediction phase will start when the HI reaches the EoL limit.

Case studies and results

This section aims to verify the efficiency and the robustness of the proposed method for machinery prognostics and in particular bearings. For this purpose, two benchmark datasets: PRONOSTIA (“Case studies and results” subsection) and IMS (“Case study: IMS dataset” section) in the NASA repository are investigated.

Case study: PRONOSTIA dataset

The first case study relies on the bearing prognostics dataset PRONOSTIA (Nectoux et al. 2012), obtained from the NASA prognostics Data Center (NASA 2019). It provides run-to-failure vibrational signals obtained from 17 ball bearings of the same typology under 3 different operating conditions, causing accelerated degradation on the component under test.

PRONOSTIA test bench description

The test bench setup, shown in Fig. 6, consist of an AC motor driving with timing belts. The shaft is connected to the inner ring of the bearing under test. The NI sensing system provides logging files of the data collected from two accelerometers positioned at 90° from each other: one measuring vibration in the direction perpendicular to the base, while another is setup parallel to the base. They are sampled with a frequency of $F_S = 25,600$ Hz for 0.1 s of acquired signal every 10 s. Accelerated degradation is simulated by means of a pneumatic actuator pushing radially on the bearing frame/outer ring. Shaft rotating speed and actuator force determine the load conditions under which the test is performed and are defined as follows:

1. Speed: 1800 rpm–Force: 4000 N;
2. Speed: 1650 rpm–Force: 4200 N;
3. Speed: 1500 rpm–Force: 5000 N.

Fig. 5 Application of particle filtering for prognostics purpose

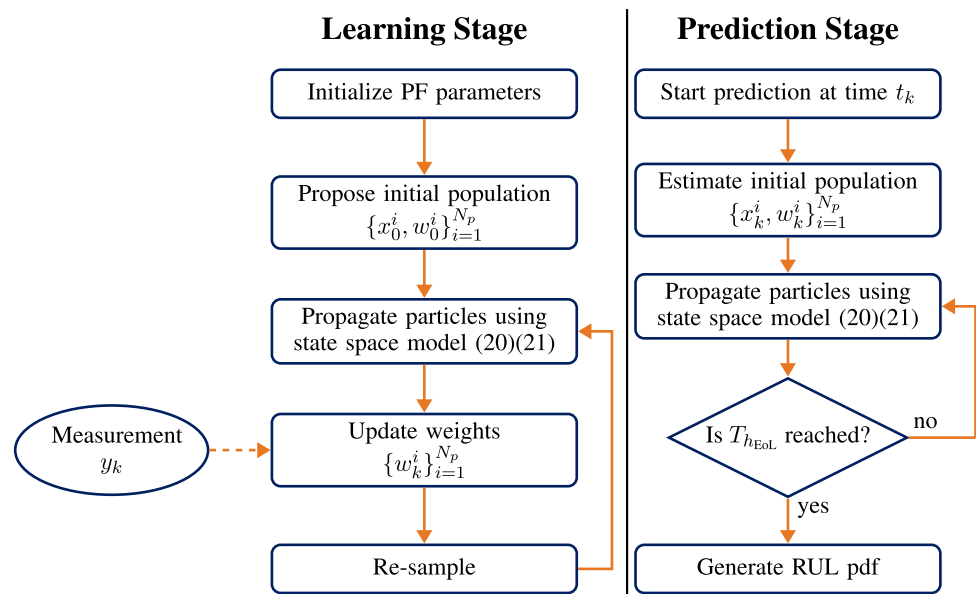
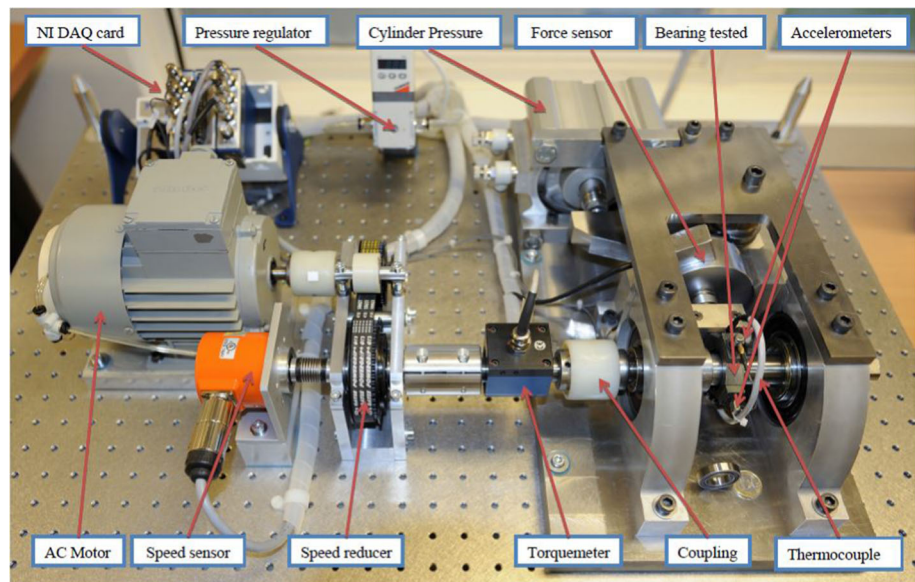


Fig. 6 Experimental ball bearing diagnostics and prognostics setup: PRONOSTIA (Nectoux et al. 2012)



The dataset contains run-to-failure measurements of 7 bearings under condition (1), 7 ones under condition (2) and 3 ones under condition (3). Vibration signal logs are organized in folders, one folder for one tested bearing, and divided in time-sorted .csv files, containing 2560 samples per file, for both accelerometers. Originally, the dataset was distributed for the *PHM Challenge* in 2012 and divided into two parts: training and validation. The former consists of 2 bearings per condition and the latter contain the remaining ones with truncated logs. Nowadays, the full data set is available for research purposes. An example of vibration signal from the database is shown in Fig. 7.

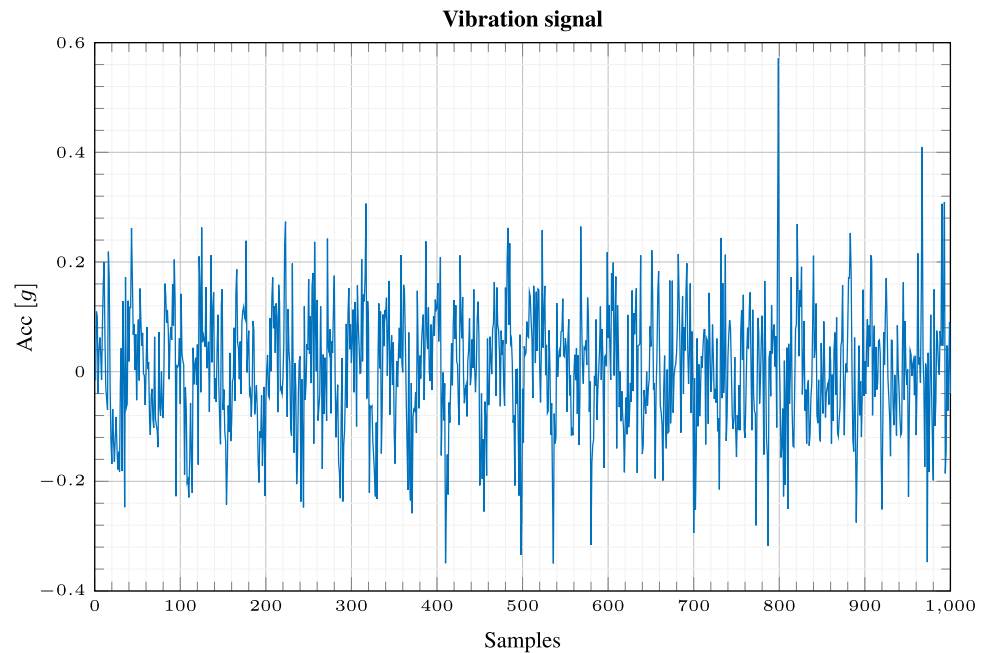
The proposed methodology (“Proposed methodology” section) is applied on the described PRONOSTIA open source database to serve as reference for practitioners

in industrial environments with PLC controlled automatic machines. Its algorithms are coded with Matlab, and can be easily implemented on the proposed PLC-Supervisor architecture. To do this, we provide practical guidelines, starting from the arrangement of measurements for MoS estimation, through the communication procedure between logic controller and supervisor, and ending with the use of Particle Filtering for RUL prediction.

Edge-computing on PLCs

Machinery controllers are based on real-time operating systems. They are able to respond to events within precise timing constraints, and they are reliable in accurately handling sensor measurements within their fieldbus network. For instance,

Fig. 7 Vibration signal from the database



Beckhoff EL3632 and B&R X20CM4800X I/O modules have sampling rates of up to 25 kHz. On the other hand, bearing failing modes are a function of the driving shaft speed and typically in the range of 1–500 Hz. In PRONOSTIA (Soualhi et al. 2014) those modes are around 200 Hz. Higher sampling rates require a lot of available bandwidth on the Fieldbus and also PC-based PLCs powerful enough to handle them, together with the control task. On the other hand, a sampling of 1 kHz is enough in this situation to catch the faulty behavior of the component. Thus, reducing the sampling rate allows the methodology to be feasible also in less powerful controllers without hindering the logic control task and without any loss of prognostics performance, reducing also the costs of its implementation.

Given those considerations, we downsampled PRONOSTIA bearing vibration signals to 2560 Hz (i.e. by a factor of 10), so a log file now carries 256 usable samples per accelerometer. Since MoS identification algorithm requires a higher number of data to feasibly perform its computations, we concatenated windows of 12 files together into a circular buffer: $N = 3072$ samples over a 2 min span in PRONOSTIA time steps. This solution measures vibrations for a brief interval ($T_{meas} = 0.1$ s) every larger interval ($T_{acq} = 10$ s) and concatenates them. It facilitates the implementation of such buffering modalities in non-powerful logic controllers thanks to its low requirements in terms of computational and memory resources. Moreover, the use of a circular buffer, that updates with a FIFO policy every time a new (batch) measurement is available, allows providing the sequence $y(1), \dots, y(N)$ to the MoS program every T_{acq} .

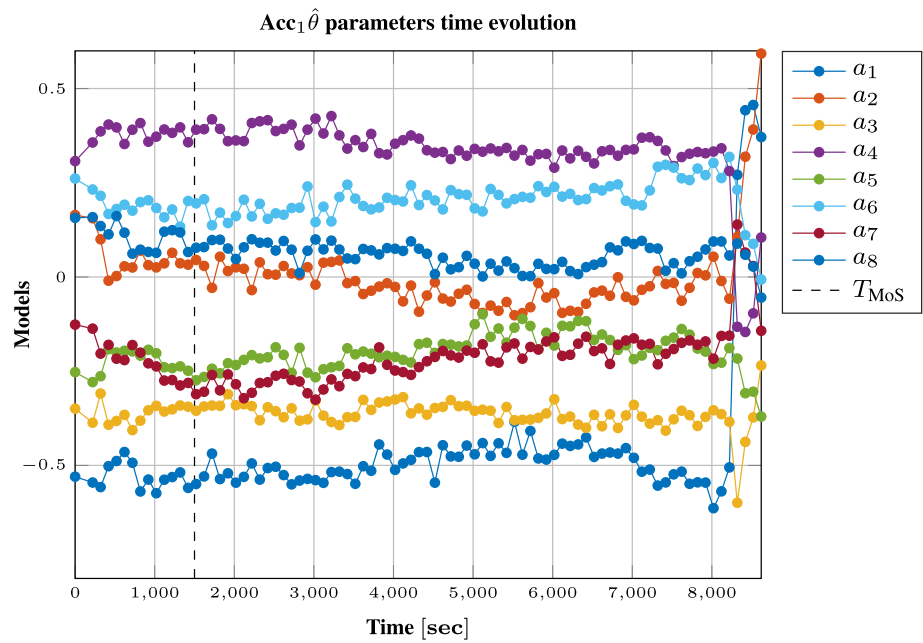
This means, in our case, the model can be elaborated every 10 s.

Once the buffer is ready, the array of data is fed to the ORIV Algorithm (1). It has been configured with model order $n = 8$, which was chosen by means of the *MDL* criterion, and the hyper-parameter $q = n = 8$, see “Model order selection” section. In the end, the algorithm is initialised with $\psi = 10$ as stated in (13). The model parameters obtained from Bearing 2’s vibration signals under test condition 1 are shown in Fig. 8, where it is possible to appreciate their evolution during the component life span. We highlighted the moment at 1500 s by a dashed vertical line because that is roughly the moment at which the bearing lubricant grease reaches its temperature working point (in the range of 90–110 °C). This instant marks the starting of model transmission to the supervisor, and the reference model θ_{nom} is obtained by computing the mean value of the first 10 collected ones.

The computations of the local HI start after the reference model is defined. When there is a disconnection between PLC and PC supervisor, the HI_{NRMSE} is used as a backup solution for fault detection, with the failure threshold defined in the range [1, 1.1]. A sample of the HI_{NRMSE} evolution is shown in Fig. 9 where the application of threshold $T_{h_{NRMSE}} = 1$ is applied to avoid severe failing modes.

At this point, the PLC has to handle the communication of the computed quantities to the supervisor PC. The proposed architecture does not require high payloads in data transmission because only a piece of information, i.e., model parameters, is remotely sent from the PLC to the supervisor PC every 10s. Various communication protocols are avail-

Fig. 8 MoS evolution in time of bearing 2 of testing condition 1



able to perform this task and can be easily integrated into the majority of PLCs. Various communication protocols are available together with their functions library, for instance, *OPC-UA* and *MQTT* are commonly used. In addition, logging of the computed θ s and of HI_{NRMSE} can be performed locally, if feasible for the system resources.

Given the previous considerations, we can elaborate more on the “Huge-Data to Big-Data” claim. Suppose we want to continuously stream the vibration signal from the PLC to the PC via LAN. The controller collects each vibration measurement into a *REAL* type variable, which is worth 4B (bytes), at a frequency of 2560 Hz, meaning 10 KB/s of raw data flow not counting overheads. As the PLC simultaneously performs the logic control task, this transmission may be difficult to handle. On the other hand, in our depicted solution where only model parameters (*REAL* variables) are sent, the transmission flow is reduced and equals to 3.2 B/s. Compared to the signal streaming, this is 3000-times smaller, and much less demanding in terms of network and storage.

Figure 10 presents the three functions previously described, i.e. measurement pre-processing, MoS generation and network and storage, that are implemented within the PLC programs. The I/O acquisition modules have higher sampling rates with respect to PLC cycling times. This is typically handled by the Fieldbus so that a set of samples is collected and provided at each cycle. Thus, the measurement pre-processing program, that fills the data buffer for MoS, should be attached to the main priority task with the lower period (usually 1 ms). This allows managing the data flow with feasible modalities, suitable to follow the acquisition schedule previously depicted. MoS generation is the most demanding in terms of computational load on the CPU of the PLC. It

should be appended to a lower priority task (e.g. a program with 10–100 ms cyclic period) than the main control program. Network and storage handling programs are usually already available on machines, for production supervision and quality control. The quantities required for transmission by the methodology should be added effortlessly, as discussed previously.

Remote-computing on PC

On the supervisor side, the remote computing unit, i.e., a PC, collects the data sent from the PLC and can possibly log and store them. Then, its next task is to compute the HI_{1-S} health indicator, presented in Eq. (18), on which the prognostics procedure is based. Firstly, since this index is quite noisy, a simple filtering method using the mean value of 12 samples is applied to smooth it. The obtained result is noted as \bar{HI}_{1-S} . Then, the index is monitored until it reaches the anomaly threshold $T_{h_{prog}} = 0.1$ where the prognostics procedure starts.

Prognostics is handled by exploiting the properties of Particle Filtering to learn the parameters of the degradation model and then to forecast its evolution. Considering Fig. 11, showing only the main cases for the sake of compactness and readability, \bar{HI}_{1-S} is an increasing function whose behaviour may seem different from bearing to bearing. By a closer analysis, it is possible to point out that such quantity follows a piece-wise evolution starting from an exponential growth, whose parameters are to be learnt, and ending with a polynomial tail. This means that the first stage of the degradation process can be represented by an exponential function, while the second stage, that starts at a random time, T_{change} , may

Fig. 9 HI_{NRMSE} evolution over time of bearing 2 under testing condition 1

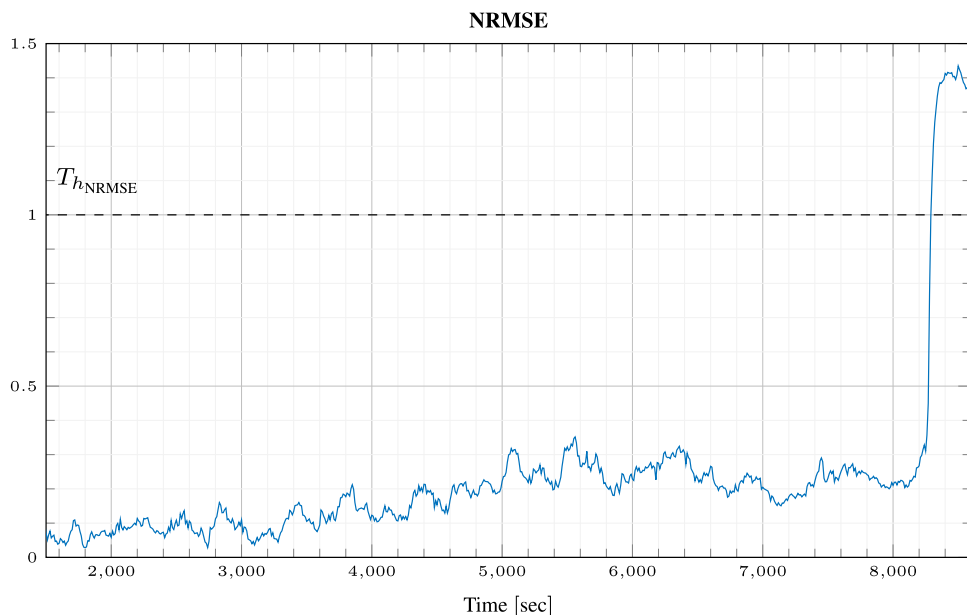
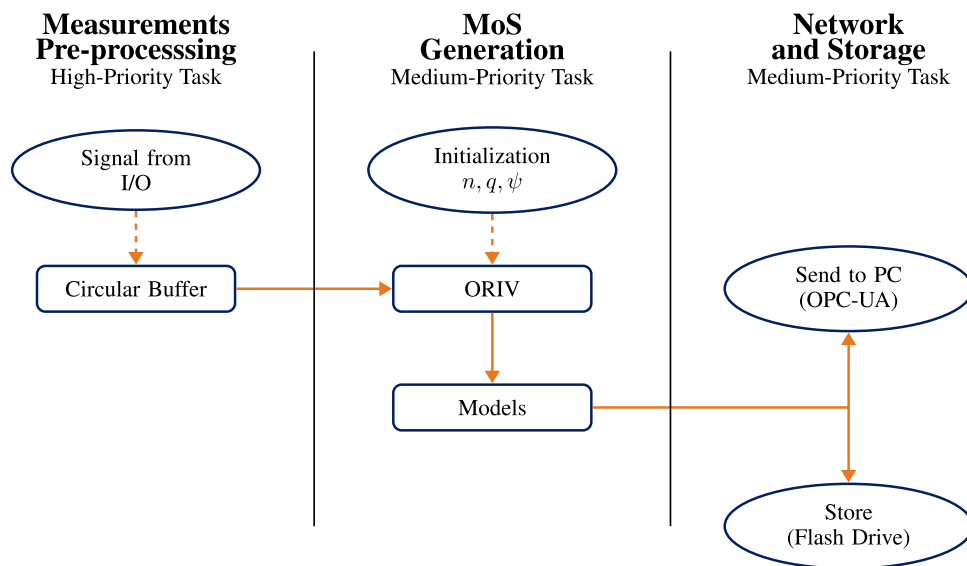


Fig. 10 PLC programs and internal information flow of the procedure on the edge side



be characterised by a polynomial function. To capture this changing point, after the aforementioned analysis, we define a threshold for the first order difference of the health indicator, namely $T_{h_{diff}}$. In detail, the changing point is identified by the moment, at which the difference of $\bar{H}I_{I-S}$ at two consecutive times is superior than $T_{h_{diff}}$. Hence, the $\bar{H}I_{I-S}$ evolution is re-written as follows:

$$\bar{H}I_{I-S}(t_k) = \begin{cases} ae^{bt_k} & \Delta\bar{H}I_{I-S}(t_k) \leq T_{h_{diff}} \\ c(t_k - T_{change})^2 + d & \Delta\bar{H}I_{I-S}(t_k) > T_{h_{diff}} \end{cases} \quad (30)$$

where

$$\Delta\bar{H}I_{I-S}(t_k) = \bar{H}I_{I-S}(t_k) - \bar{H}I_{I-S}(t_{k-1}) \quad (31)$$

and $T_{h_{diff}}$ is determined in this case as 0.2. Then, the coefficients a and d correspond to $T_{h_{prog}}$ and $\bar{H}I_{I-S}(T_{change})$, while b and c are positive parameters modelled as conditioned random walks:

$$b(t_k) = b(t_{k-1}) + \lambda(t_k) \quad (32)$$

$$c(t_k) = c(t_{k-1}) + \gamma(t_k) \quad (33)$$

with $\lambda \sim \mathcal{N}(0, \sigma_\lambda^2 = 10^{-5})$ and $\gamma \sim \mathcal{N}(0, \sigma_\gamma^2 = 10^{-5})$. The learning phase of the Particle Filter in this case is employed to estimate firstly b and secondly c . Then, the predictions are done at every time step and RUL is estimated based on the end of life threshold $T_{h_{EoL}} = 10$, which is a safe one, to stop production and to start maintenance activities before it is too late.

Fig. 11 Evolution of $\tilde{H}_{I_{LS}}$ of bearing 1 and 2 under testing condition 1 during their relevant run-to-failure tests

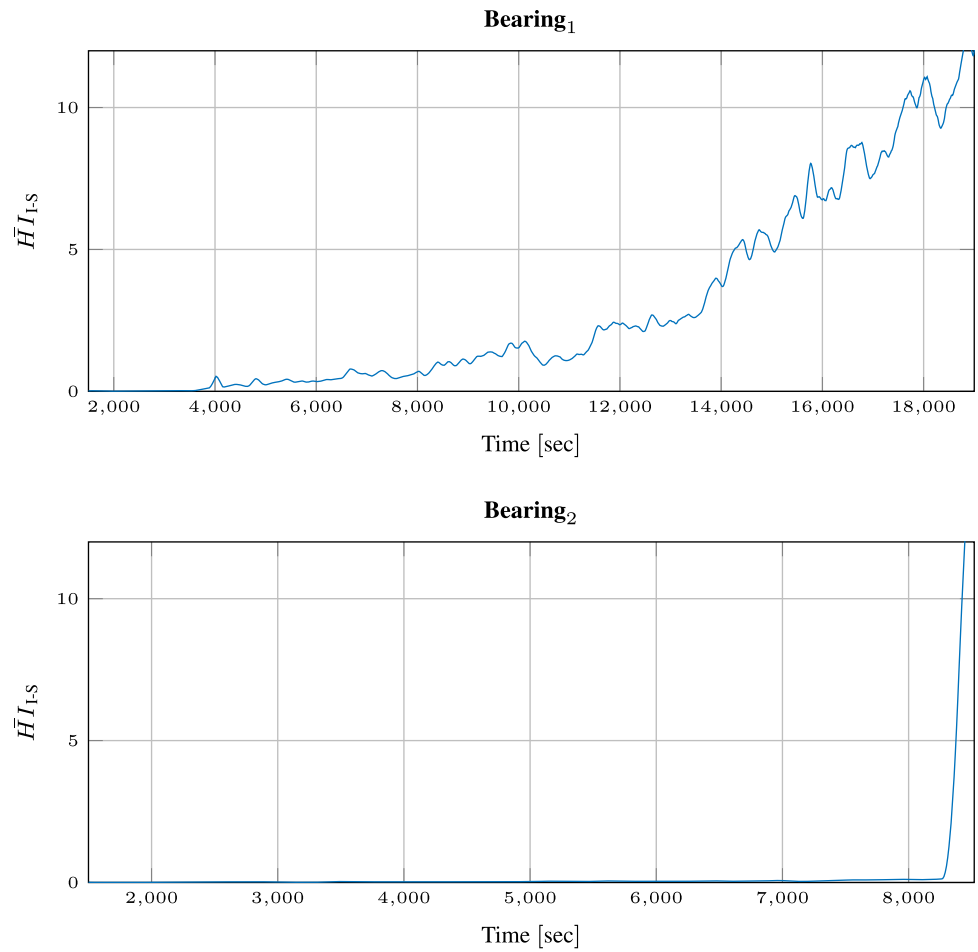


Fig. 12 Evolution of $\tilde{H}_{I_{LS}}$ of bearing 3 under test condition 1: Filtering, state estimation, and RUL prediction

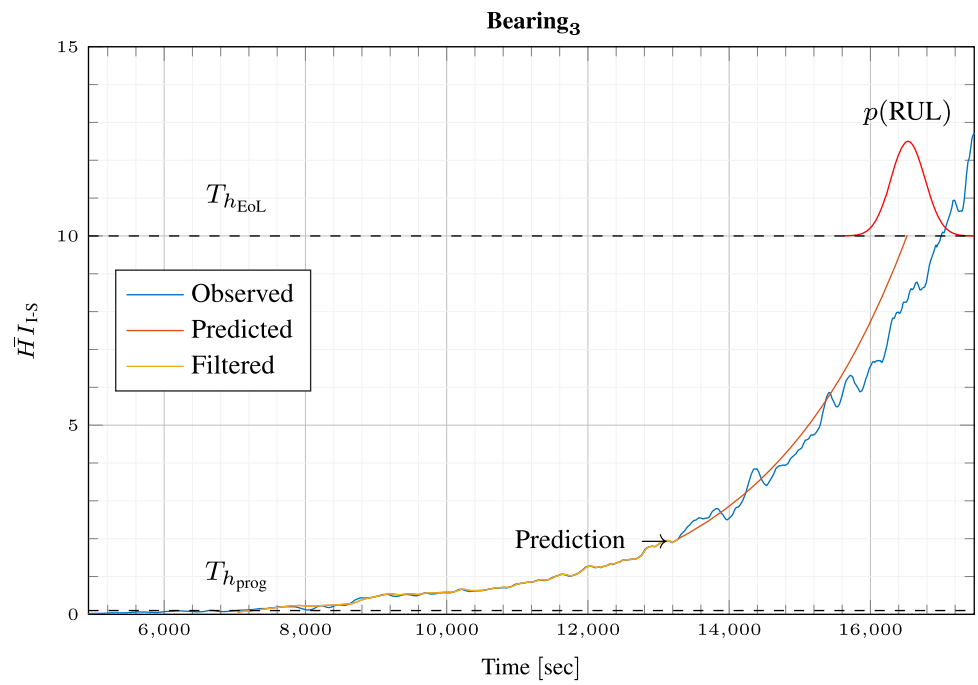
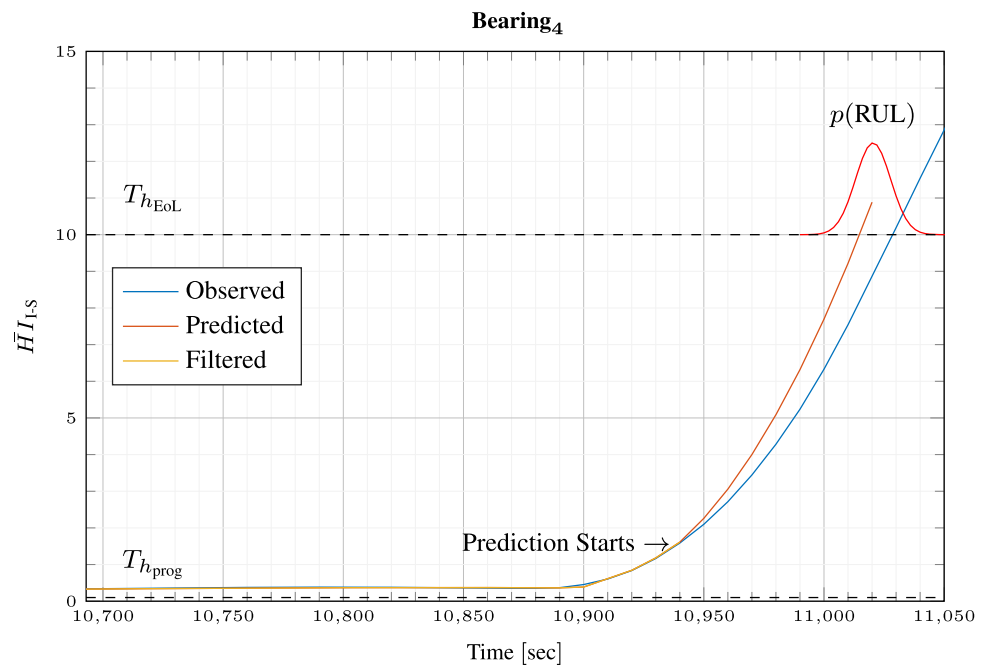


Fig. 13 Evolution of $\tilde{H}I_{1,S}$ of bearing 4 under test condition 1: Filtering, state estimation, and RUL prediction



Figures 12 and 13 illustrate how the Particle Filter performs when the degradation evolves only through the exponential function (e.g. Bearing 3) or when there exist the model changing point (e.g. Bearing 4). These figures describe what happens at a particular time instant t_k where the PF learning ends and the prediction starts. $\tilde{H}I_{1,S}$'s actual evolution is shown in blue, while the filtered PF mean state computed until t_k is shown in yellow and the state prediction from that point on is in orange. The introduced model allows state filtering close to the observed one and its propagation fits $\tilde{H}I_{1,S}$ evolution. Moreover, the final RUL probability density is shown in red together with threshold levels $T_{h_{prog}}$ and $T_{h_{EoL}}$ presented in dashed black lines. $p(\text{RUL})$ depends on how long the prediction is propagated to reach $T_{h_{EoL}}$. The longer the time it takes the greater its variance is. Moreover, in Fig. 12, RUL prediction helps in maintenance decision making, since, given the accelerated degradation, there is plenty of time to plan the servicing (we have 50 min to play with). Besides, Fig. 13 shows that RUL prediction can be used to avoid severe bearing degradation which may result in a critical failure.

A broader view of the RUL prediction is shown in Figs. 14 and 15. These figures represent, by the blue-dotted line, the mean value of the computed RULs of the various particles throughout the test, from the starting of prognostics to the component end of life. The orange-dashed line represents the actual RUL. Figure 14 shows feasible RUL prediction when there is no change in the underlying PF model structure, after an initial drift, the more the filter “learns” the more it gets close to the actual value. On the other hand, the moment when the model structure change occurs is uncertain.

From the available data, it turns out that it is not predictable, however, our procedure is capable of catching it as soon as it occurs, offering essential times for intervention. Figure 15 shows how the predictor treats the incoming observation with the exponential model until it recognises the change in the evolution and gets back in track with the actual RUL value. For these reasons, the communication between PLC and PC should be in both directions. In an automated maintenance perspective, the supervisor transmits the predicted RULs to the machinery to trigger fail-safe policies in the worst-case scenario.

Performance results

Finally, to assess the performance of the algorithm on the overall dataset we make use of the Prognostic Horizon (PH) metric (Saxena et al. 2010), a standard indicator that provides information on how the algorithm is able to suitably predict the system RUL. PH is defined as the difference between the time index t when the predictions first meet the specified performance criteria (based on data accumulated until time index t) and the time index for EoL, t_{EoL} . The performance requirement is specified in terms of an allowable error bound (α) around the true EoL where the choice of α depends on the estimate of time required to take a corrective action. A formal definition is given in the following:

$$PH = t_{EoL} - t_{k_\alpha} \quad (34)$$

Fig. 14 RUL over time of bearing 3 under test condition 1 using the proposed prognostics PF method

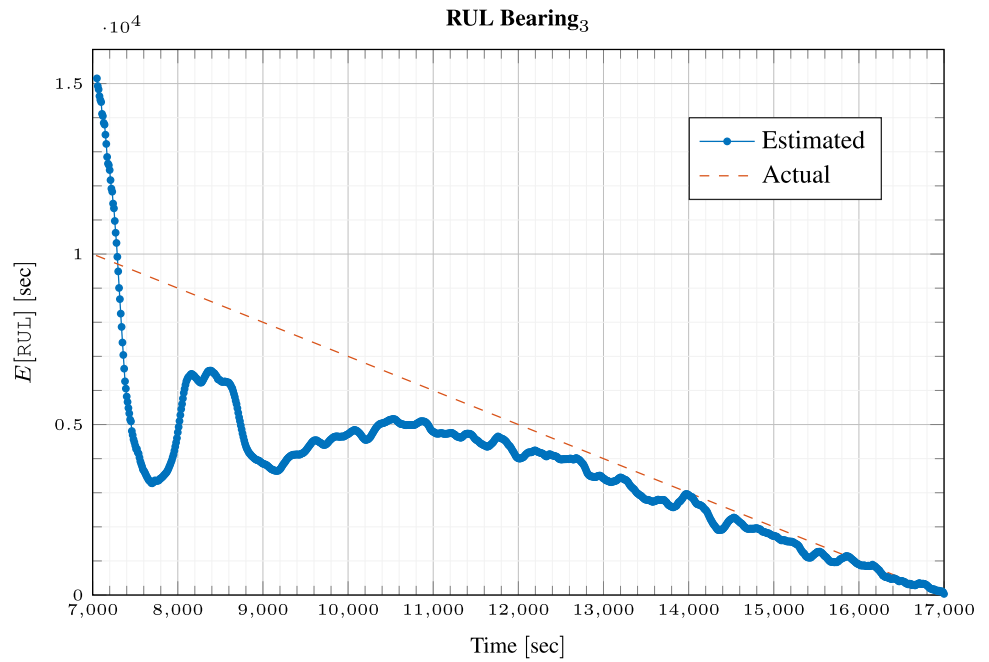
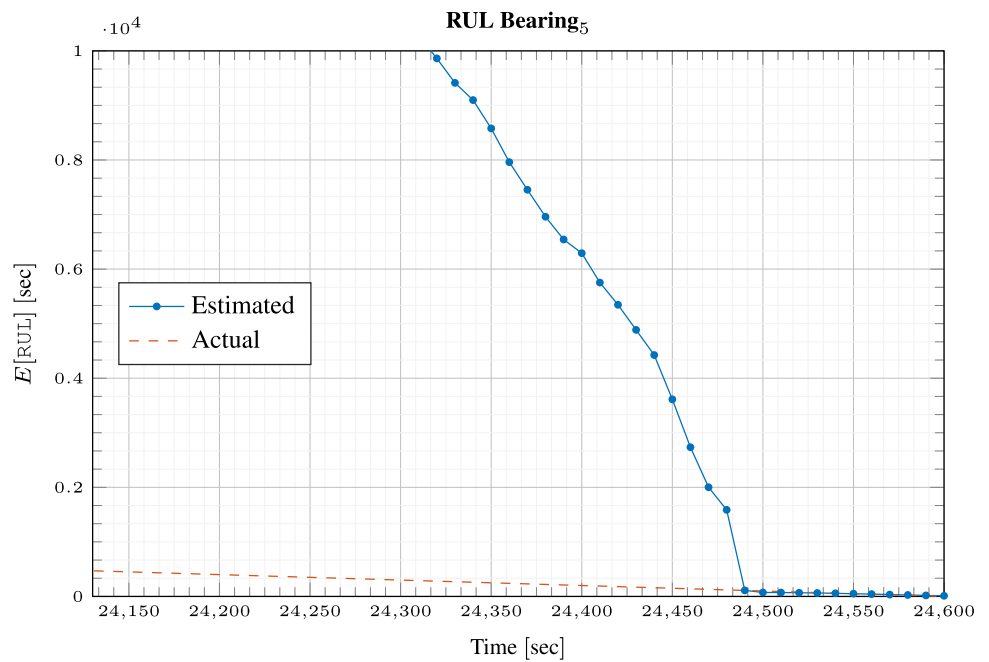


Fig. 15 RUL over time of bearing 5 under test condition 1 using the proposed prognostics PF method



with

$$k_\alpha = \min \{k \mid (k \in \mathbf{P}) \wedge ((r_{*k} - \alpha \cdot t_{\text{EoL}}) \leq \hat{r}(t_k) \leq (r_{*k} + \alpha \cdot t_{\text{EoL}})) \forall k \geq k_\alpha \} \quad (35)$$

where \mathbf{P} is the set of all time indices for which a prediction is made, r_{*k} is the true RUL at time t_k and $\hat{r}(t_k)$ is the predicted RUL at time t_k (i.e., its mean value in our particular case). Moreover, to stress the methodology robustness, we choose

the value of PH having also the predictions obtained after t_{k_α} within the bounds. Table 1 shows the obtained results with $\alpha = 0.2$. The considerations drawn before about the commuting degradation model still hold and the results show that when the model function changes to polynomial there is a minute span to handle the possible fault. On the other hand, when the degradation keeps its exponential trend, at least 20min are available to program preventive actions. Notice

Table 1 Prognostic Horizon results on PRONOSTIA dataset

PH (min)	Bearing						
	1	2	3	4	5	6	7
Cond. 1	136.5	2	125.17	1.67	2.5	3.17	4.3
Cond. 2	3.5	2.17	–	2	–	1.17	1
Cond. 3	1.17	20	3.67				

Data obtained with $\alpha = 0.2$

Table 2 Prognostic horizon results on PRONOSTIA dataset

PH (min)	Bearing		
	1	2	3
Soualhi et al. (2014)	6.8	2	1.2
Our results	136.5	2	125.17

Comparison with Soualhi et al. (2014)

that no results are available for bearing 3 and 5 of condition 2 because their run-to-failure data is incomplete.

To highlight the performance of the proposed approach, its results are compared to the outcomes obtained in (Soualhi et al. 2014) on the first three bearings under condition 1. As shown in Table 2, our procedure has better results for bearing 1 and 3, and the same result for bearing 2.

Case study: IMS dataset

To further validate our proposition, we port it on another bearing prognostics dataset within the NASA prognostics Data Center (NASA 2019), the IMS (Lee et al. 2007) one. It contains three run-to-failure experiments with four bearings each, constantly driven at 2000 rpm and under a radial load of 6000 lbs (2721.554 kg). Vibrational data of each bearing were collected for $T_{meas} = 1$ s at a 20 kHz every $T_{acq} = 10$ min and logged into .txt files until bearings EoL. We applied the proposed methodology with the same course of action previously described throughout “Case studies and results” section. To avoid the redundancy, for this case study, we briefly present in this section the modalities of data pre-processing to suit MoS estimation and PF structure, and also the definition of the method hyperparameters.

Given the data set organisation, the signal set-up for MoS estimation is similar to the previous one. The data are downsampled to 2500 Hz for the same parsimony and cost effectiveness reasons described for PRONOSTIA. Each logged file is regarded as the data buffer (with a window size of 2500 samples) on which a model is identified. In this case, the first two bearings of the first test were used to obtain the method’s hyperparameters in advance. The resulting model order was $n = 8$ and $q = 8$ more equations were used to increase the robustness of the ORIV estimation. Besides, the healthy references for the computations

Table 3 Prognostic horizon results on IMS dataset

PH (min)	Bearing			
	1	2	3	4
Test 1	90	100	10	10
Test 2	1010	1250	1300	810
Test 3	1440	1100	350	1450

Data obtained with $\alpha = 0.2$

of the HIs are computed by averaging the models obtained in the first 2 operational hours of each bearing. Then, the threshold to start prognostics on the $\bar{H}I_{I,S}$ indicator, which is filtered in the same way as in PRONOSTIA, has been set to $T_{h_{prog}} = 0.1$. The underlying particle filtering prognostics model is unchanged from (30)–(33) except for the commutation threshold, which has been set to $T_{h_{diff}} = 0.11$. Finally, for this data set the End-of-Life threshold, on which RUL predictions are computed, is $T_{h_{EoL}} = 2$. A snapshot of the prognostic task using the particle filtering model of Eq. (30) with the selected thresholds is shown in Fig. 16 for bearing 3 of the second run-to-failure test. The overall evolution of the mean value of the estimated RUL of that bearing is provided in Fig. 17.

Table 3 shows the results obtained by the methodology in prognostic performance using the PH metric. In this situation, the methodology still holds its performance and is able to grant at least a 10 min scope for prognosis when a model change occurs, while at least almost 6 hours when the model keeps its exponential evolution.

Conclusions

In this work, we proposed an autonomous health management solution for smart manufacturing. It focuses on exploiting the increased computational power of machinery controllers and their interconnection with the supervising PCs of the automation pyramid. In detail, the logic controller is the edge-computing unit that performs the condition monitoring task. It employs the Model-of-Signals technique to refine the information measured from onboard sensors into compact and meaningful features. The supervising PC acts as the remote-computing unit collecting those computed models to produce health indicators for the prognostics task. Then, it makes use of Particle Filtering to model the degradation of the components to forecast their Remaining Useful Life.

Manufacturers can take advantage of this methodology to integrate autonomous maintenance policies as features in their machines, keeping their expertise with standard automation platforms. The procedure exploits the structure of the automation pyramid as well as the commercial equipment already in place, avoiding the addition of non-standard equip-

Fig. 16 Evolution of $\bar{H}_{I,S}$ of bearing 3 of test 2: filtering, state estimation, and RUL prediction

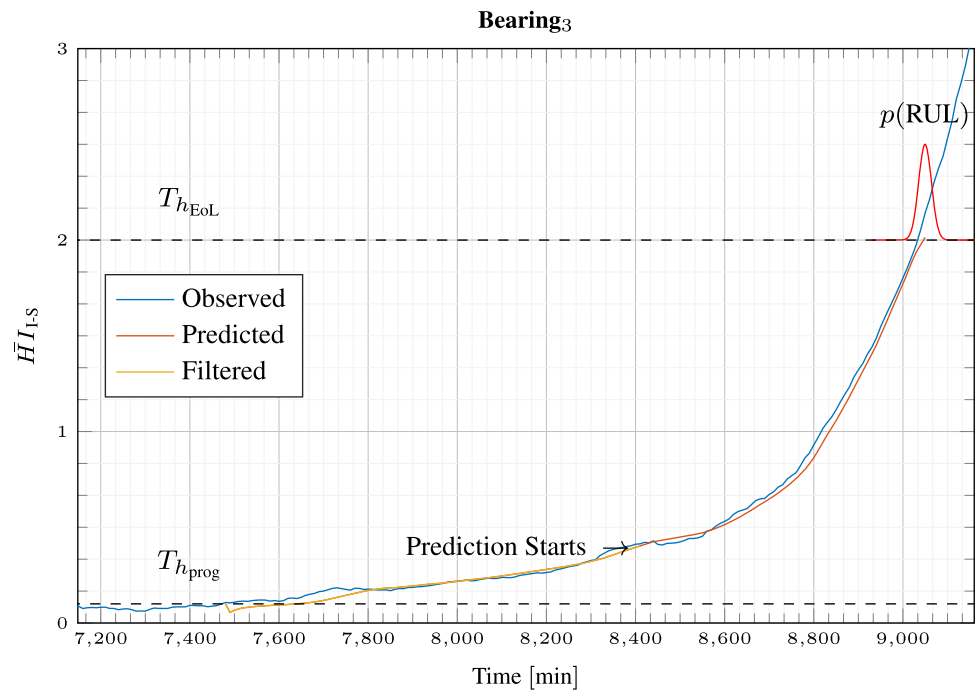
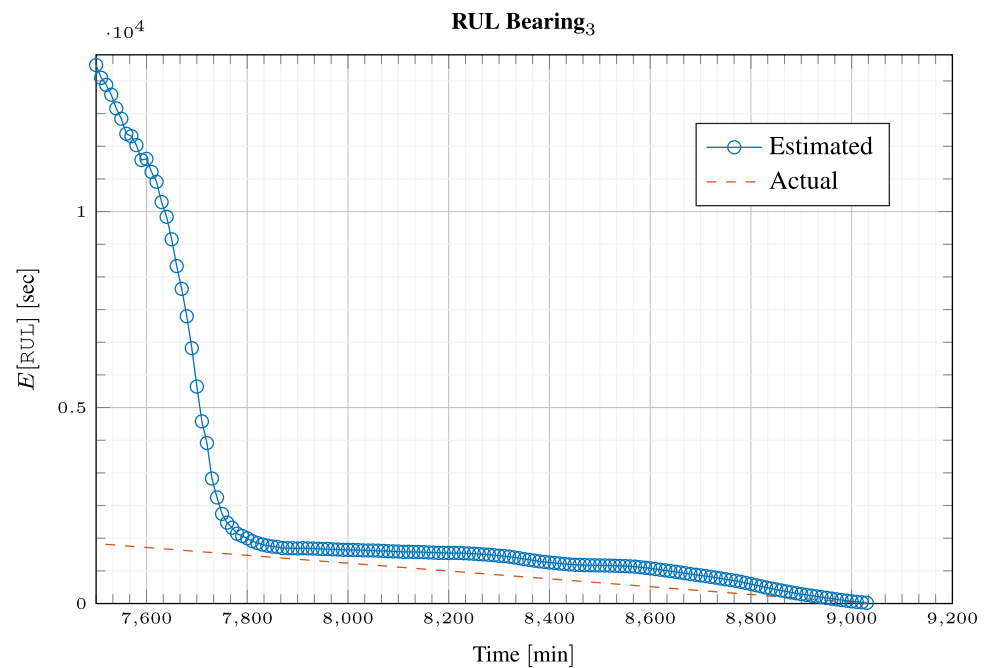


Fig. 17 RUL over time of bearing 3 of test 2 using the proposed prognostics PF method



ment to perform the task. Local sensor data refinement allows reducing the impact of PHM information transmission over the PLC-PC network and to distribute the computational load. This converts the “Huge-Data” problem of streaming raw sensor signals to remote computing units into a manageable one. This is possible because of the light weight nature of the recursive algorithm used to produce MoS that is able to feasibly run alongside the logic control task.

Two case studies are employed to test and validate the method. The PRONOSTIA dataset is investigated to guide

the industrial practitioners on how to apply the proposed methodology and lay the foundations for autonomous health management functionalities on PLCs. The procedure is described in detail from the implementation of condition monitoring through MoS on controllers to the definition of the PF degradation model, based on the HI evolution, and RUL forecasting on the supervising PC. In addition, the IMS dataset has been used to further evaluate the methodology under similar PHM conditions. Then, performance indicators

have been computed for both the datasets using the prognostic horizon metric.

Even though the focus was on the development of an “industrial technology aware” methodology, we made use of bearings datasets to develop our discussion on its implementation, allowing us to cover the main technological and architectural aspects of the method application. Consequently, this choice bonded our proposition with the mechanical domain and in particular with the use of vibrational signals as a starting point and the degradation model of the component. The selection of the adequate signals upon which apply MoS and the related recursive estimation algorithm is crucial to accomplish the first stage of the method. On the other hand, the proposed degradation model is not general and inevitably considers the nature of component under test and the indicator to which we link its degradation. Future studies should also take into account those aspects, with more in-depth analysis on other machinery provided measurements and components.

Given those considerations, the aim of this work is to build a first bridge between the industrial and research field. The obtained results are promising, laying the foundations for the deployment on industrial equipment of the method to “unlock” machinery smart potential, a step toward intelligent manufacturing.

Funding Open access funding provided by Alma Mater Studiorum-Università di Bologna within the CRUI-CARE Agreement.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- An, D., Choi, J. H., & Kim, N. H. (2013). Prognostics 101: A tutorial for particle filter-based prognostics algorithm using matlab. *Reliability Engineering and System Safety*, *115*, 161–169.
- Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174–188.
- Atamuradov, V., Medjaher, K., Dersin, P., Lamoureux, B., & Zerhouni, N. (2017). Prognostics and health management for maintenance practitioners—review, implementation and tools evaluation. *International Journal of Prognostics and Health Management*, *8*(060), 1–31.

- Barbieri, M. (2017). *Seamless infrastructure for “big-data” collection and transportation and distributed elaboration oriented to predictive maintenance of automatic machines*. Master’s thesis, University of Bologna.
- Barbieri, M., Bosso, A., Conficoni, C., Diversi, R., Sartini, M., & Tilli, A. (2018). An onboard model-of-signals approach for condition monitoring in automatic machines. In *Enterprise Interoperability: Smart Services and Business Impact of Enterprise Interoperability* (pp 263–269). Wiley – ISTE.
- Barbieri, M., Diversi, R., & Tilli, A. (2019a). Condition monitoring of ball bearings using estimated AR models as logistic regression features. In *18th European control conference (ECC 2019)* (pp 3904–3909).
- Barbieri, M., Mambelli, F., Diversi, R., Tilli, A., & Sartini, M. (2019b). Condition monitoring by model-of-signals: Application to gearbox lubrication. In IFAC (ed) *15th European workshop on advanced control and diagnosis, ACD 2019*.
- Barbieri, M., Mambelli, F., Lucchi, J., Diversi, R., Tilli, A., & Sartini, M. (2020). Condition monitoring of a paper feeding mechanism using model-of-signals as machine learning features. *PHM Society European Conference*, *5*, 10–10.
- Cerrada, M., Sánchez, R. V., Li, C., Pacheco, F., Cabrera, D., de Oliveira, J. V., et al. (2018). A review on data-driven fault severity assessment in rolling bearings. *Mechanical Systems and Signal Processing*, *99*, 169–196.
- Friedlander, B. (1984). The overdetermined recursive instrumental variable method. *IEEE Transactions on Automatic Control*, *29*(4), 353–356.
- Gordon, N. J., Salmund, D. J., & Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE proceedings F (radar and signal processing)* (Vol. 140, pp. 107–113). IET.
- Gouriveau, R., Medjaher, K., & Zerhouni, N. (2016). *From prognostics and health systems management to predictive maintenance I: Monitoring and prognostics*. New York: Wiley.
- Isermann, R. (2005). Model-based fault-detection and diagnosis-status and applications. *Annual Reviews in control*, *29*(1), 71–85.
- Isermann, R. (2006). *Fault-diagnosis systems: An introduction from fault detection to fault tolerance*. Berlin: Springer.
- Itakura, F. (1968). Analysis synthesis telephony based on the maximum likelihood method. In *The 6th international congress on acoustics* (pp. 280–292).
- Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483–1510.
- Lee, J., Qiu, H., Yu, G., & Lin, J. (2007). Bearing data set, NASA ames prognostics data repository.
- Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., & Siegel, D. (2014). Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications. *Mechanical Systems and Signal Processing*, *42*(1–2), 314–334.
- Ljung, L. (1999). *System identification: Theory for the user*. Upper Saddle River: Prentice-hall.
- Magnant, C., Giremus, A., & Grivel, E. (2014). On computing Jeffrey’s divergence between time-varying autoregressive models. *IEEE Signal Processing Letters*, *22*(7), 915–919.
- NASA (2019) Nasa data repository for prognostic. <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>.
- Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Chebel-Morello, B., Zerhouni, N., & Varnier, C. (2012). Pronostia: An experimental platform for bearings accelerated degradation tests. In *IEEE international conference on prognostics and health management, PHM’12., IEEE Catalog Number: CPF12PHM-CDR* (pp. 1–8).

- Saha, B., & Goebel, K. (2011). Model adaptation for prognostics in a particle filtering framework. *International Journal of Prognostics and Health Management*, 2, 61.
- Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management*, 1(1), 4–23.
- Si, X. S., Wang, W., Hu, C. H., & Zhou, D. H. (2011). Remaining useful life estimation—A review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1), 1–14.
- Skima, H., Medjaher, K., Varnier, C., Dedu, E., Bourgeois, J., & Zerhouni, N. (2016). Fault prognostics of micro-electro-mechanical systems using particle filtering. *IFAC-PapersOnLine*, 49(28), 226–231.
- Söderström, T., & Stoica, P. (1989). *System identification*. Upper Saddle River: Prentice Hall.
- Soualhi, A., Medjaher, K., & Zerhouni, N. (2014). Bearing health monitoring based on Hilbert–Huang transform, support vector machine, and regression. *IEEE Transactions on Instrumentation and Measurement*, 64(1), 52–62.
- Tulsyan, A., Gopaluni, R. B., & Khare, S. R. (2016). Particle filtering without tears: A primer for beginners. *Computers and Chemical Engineering*, 95, 130–145.
- Vogl, G. W., Weiss, B. A., & Helu, M. (2019). A review of diagnostic and prognostic capabilities and best practices for manufacturing. *Journal of Intelligent Manufacturing*, 30(1), 79–95.
- Wang, J., Gao, R. X., Yuan, Z., Fan, Z., & Zhang, L. (2019). A joint particle filter and expectation maximization approach to machine condition prognosis. *Journal of Intelligent Manufacturing*, 30(2), 605–621.

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