

Computational framework for real-time diagnostics and prognostics of aircraft actuation systems

Original

Computational framework for real-time diagnostics and prognostics of aircraft actuation systems / Berri, Pier Carlo; Dalla Vedova, Matteo D. L.; Mainini, Laura. - In: COMPUTERS IN INDUSTRY. - ISSN 0166-3615. - ELETTRONICO. - 132:(2021), p. 103523. [10.1016/j.compind.2021.103523]

Availability:

This version is available at: 11583/2915214 since: 2021-07-27T10:09:06Z

Publisher:

Elsevier

Published

DOI:10.1016/j.compind.2021.103523

Terms of use:

openAccess

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

Computational framework for real-time diagnostics and prognostics of aircraft actuation systems

Pier Carlo Berri^{a,1,*}, Matteo D.L. Dalla Vedova^{a,2}, Laura Mainini^{a,b,3}

^a*Politecnico di Torino, c.so Duca degli Abruzzi 24, 10129 Turin (IT)*

^b*Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

Abstract

Prognostics and Health Management (PHM) are emerging approaches to product life cycle that will maintain system safety and improve reliability, while reducing operating and maintenance costs. This is particularly relevant for aerospace systems, where high levels of integrity and high performances are required at the same time. We propose a novel strategy for the nearly real-time Fault Detection and Identification (FDI) of a dynamical assembly, and for the estimation of Remaining Useful Life (RUL) of the system. The availability of a timely estimate of the health status of the system will allow for an informed adaptive planning of maintenance and a dynamical reconfiguration of the mission profile, reducing operating costs and improving reliability. This work addresses the three phases of the prognostic flow – namely (1) signal acquisition, (2) Fault Detection and Identification, and (3) Remaining Useful Life estimation – and introduces a computationally efficient procedure suitable for real-time, on-board execution. To achieve this goal, we propose to combine information from physical models of different fidelity with machine learning techniques to obtain efficient representations (surrogate models) suitable for nearly real-time applications. Additionally, we propose an importance sampling strategy and a novel approach to model damage propagation for dynamical systems. The methodology is assessed for the FDI and RUL estimation of an aircraft electromechanical actuator (EMA) for secondary flight controls. The results show that the proposed method allows for a high precision in the evaluation of the system RUL, while outperforming common model-based techniques in terms of computational time.

Keywords: Estimation of Remaining Useful Life (RUL), Prognostics and Health

*Corresponding author

Email address: pier.berri@polito.it (Pier Carlo Berri)

¹PhD Student, Department of Mechanical and Aerospace Engineering (DIMEAS), Politecnico di Torino

²Assistant Professor, Department of Mechanical and Aerospace Engineering (DIMEAS), Politecnico di Torino

³Visiting Professor, Department of Mechanical and Aerospace Engineering (DIMEAS), Politecnico di Torino. Research Affiliate, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

1. Introduction

The steadily increasing complexity of aircraft systems results in large amount of heterogeneous components to integrate. Each component is characterized by its own set of failure modes, which can interact with those of the other components, increasing the overall system failure rate and making the fault identification and isolation process difficult and time expensive. This can eventually lead to worsen the reliability and availability characteristics of the vehicle. The traditional approach to system life-cycle management is based on scheduling maintenance interventions *a priori*: components are replaced at the end of their design life, regardless their actual health status [1, 2, 3]. This strategy leads to high maintenance costs and cannot guarantee that no failure will occur before the predicted end of life, for example as the result of an undetected manufacturing defect; to reduce risk on safety-related equipment, critical components are redounded [4, 5], increasing weight and further reducing basic reliability. Conversely, latest approaches like Condition Based Maintenance (CBM) [6, 7, 8] and Integrated Vehicle Health Management (IVHM) [9, 10, 11] aim to account for advances in Prognostics and Health Management (PHM) disciplines, in order to better manage the maintenance schedule, reducing costs and increasing mission reliability [12, 13, 14, 15, 16]. PHM relies on continuous monitoring of the actual health status of components, to adaptively estimate the system Remaining Useful Life (RUL) [17, 18, 19, 20]. The benefits promised by CBM and IVHM motivate the great interest in enabling next generation systems and vehicles to autonomously detect damages and faults at their early stage, and predict the associated RUL during operations. This capability would allow to replace components only when really needed, avoid disposing systems that are still healthy, and even recalibrate systems operational envelope to guarantee a longer and safer system life.

Common approaches to PHM leverage either model-based strategies (i.e. relying on physics-based representations of the monitored system [21, 22, 23, 24, 25]) or data-driven methods [26, 27, 28, 29]. A review of model-based condition monitoring strategies to enable system prognostics is provided by Tinga and Loendersloot [30]. In [31] a structured residual between the system response and a digital twin is compared to a threshold in order to detect faults of industrial equipment. In [32] faults are detected online with a data-driven algorithm, and later identified offline employing a model-based strategy. Henry et al. [33] propose to compare attitude command and measurement of the inertial

platform to determine failures in the attitude control system of a spacecraft. Huang et al. [34] and Zhao et al. [35] provide reviews of data-driven approaches to prognostics leveraging statistical methods and deep learning. In addition, Rengasamy et al. [36] discusses the use of deep learning schemes for planning of aircraft maintenance and overhaul. In [37] an Extreme Learning Machine (ELM) is employed for fault detection of wind turbines, while in [38] feedforward networks are used for similarity-based prognostics. Autoregressive integrated moving average (ARIMA) is applied to the RUL prediction of milling machine cutting tools in [39]. Model-based techniques usually require large computational resources, and cannot be executed in real-time by on-board hardware. Data-driven methods, conversely, need large datasets for training, which are usually not available from field. Although latest generation aircraft include widespread sensor networks that monitor and record large amount of data, a suitable statistical characterization of rare events is still difficult to obtain. As an example, few field data is available regarding the system-level effects of critical failure modes that have by design a low probability to occur. In addition, thoroughly testing data-driven algorithms for safety-critical applications can be a demanding task, since interpretability, explainability and generalization of machine-learning results are still mostly open challenges.

This paper proposes a computational framework for a nearly real-time estimation of the Remaining Useful Life for dynamical assemblies from measurements available from installed feedback or diagnostic sensors. Those can be of heterogeneous nature: for example, current flowing inside an electric circuit, position and speed of an actuator, pressure and temperature of hydraulic fluid at given locations of the system. A preliminary version of this framework was presented in [40]; starting from that work, we revised the implementation of the algorithm, refined the framework, and extended the discussion of results and performances of the PHM strategy. The implementation is optimized to better leverage the offline-online structure of the framework and achieve fast real-time performances, while tracking the required computational effort. Additional extensive parametric studies have been conducted to investigate the performances and support the algorithmic choices proposed in this paper. The methodology combines an optimal signal compression strategy with reduced order modeling and machine learning techniques; this allows to obtain a computationally efficient map from the measured signals to the RUL, and to reduce the storage and processing power required for on-board, time and resource-constrained computations. Our strategy learns surrogate models of the system offline: online, these surrogate models are employed to speed up the computational burden associated with the determination of the current system health condition

and with the estimation of the RUL. Additionally, offline we determine the location of a set of informative components of the monitored signals to store and process; those are
70 employed online to reduce the dimensionality of the problem.

As an application of our methodology, we consider the case of actuators for aircraft flight control systems (FCSs). FCSs are critical aircraft systems because a failure can lead to the impossibility to control the vehicle, with catastrophic consequences. Hence, health monitoring for FCSs has great potential to bring significant improvements in
75 terms of mission reliability, operating costs, aircraft performance, and eventually relax requirements on system redundancies. The problem is inherently challenging: the models of FCS equipment need to combine different disciplines, as mechanical, aerodynamic, structural, hydraulic, and electrical/electronic subsystems operate together to achieve the required performances. The number of possible failure modes is high, and so the
80 dimensionality of the FDI problem. Additionally, different faults may result in similar effects on the system behavior, or particular operating conditions may be misidentified as faults. All these aspects make this application an interesting demonstration case for the proposed strategy, as they highlight the shortcomings of current approaches.

In this manuscript, Section 2 introduces the general formulation of the problem,
85 Section 3 details the methodology we propose for the prognostic analysis, Section 4 presents the demonstration problem discussed in this paper and the associated physical models, and Section 5 presents the results of our investigations.

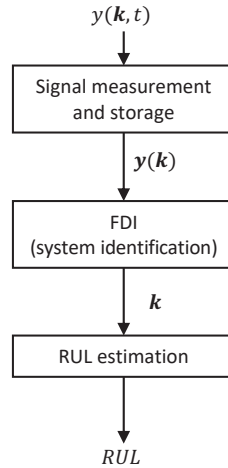


Figure 1: Schematic representation of the ideal RUL estimation flow

2. Prognostics and Health Management (PHM): problem formulation

The common prognostics flow includes three steps, namely signal measurement and storage, Fault Detection and Identification (FDI) and estimation of Remaining Useful Life (RUL), as depicted in Figure 1. In the signal measurement and storage phase, an output signal is measured from the system with a standard acquisition frequency. The signal, sensitive to the system condition, is an indicator of the health status of the components, and can be used to inform the subsequent FDI phase. In the FDI phase, the system output is processed to identify the early signs of damage and wear. Eventually, in the RUL estimation phase, the identified health condition is used to inform an estimate of the actual remaining useful life of the system.

In a traditional PHM process, the signal measurement and storage phase is the only one performed in real-time. It consists in acquiring data $y(\mathbf{k}, t)$ from the available sensors installed on the system. The acquisition frequency is usually fixed, and results from a compromise between hardware capabilities and retained information. The result of the acquisition phase is a vector $y(\mathbf{k})$, dependent on the health condition of the system \mathbf{k} . Large amounts of data can be easily produced in this phase, which can be cumbersome to store and to deal with in the following phases of the PHM process: this motivates why the FDI and RUL estimation tasks are usually performed offline. We address this issue aiming to compress the useful information in order to reduce the dimensionality of the FDI problem.

The subsequent FDI phase estimates the current health condition of the monitored system by processing the signals acquired and compressed in the previous step. Common approaches to FDI rely on the use of models, reliable representations of the physical systems and emulators of their dynamic behavior: a system output signal, sensitive to the damage condition, is measured and compared to the output signal computed with a numerical model. In [41], a physics-based model of aircraft flight dynamics is evaluated to compute the residual between the response of the physical system and its digital twin; then a statistical anomaly detection algorithm analyzes this residual to identify faults of the aileron actuation. A similar approach is proposed in [42] to determine anomalous behavior of the flight control actuator of a UAV; the strategy analyzes the effects of the failure at aircraft level: as a result, incipient faults are not detectable. In [43], a dynamical observer leveraging a Kalman Filter is employed for the model-based condition monitoring of wind turbines. Hence, the FDI problem is a system identification problem whose solution (the current fault condition \mathbf{k}^c) is the one that minimizes (ideally vanishes) the discrepancies between the measured signal \mathbf{y} and

the simulated one $\mathbf{y}^{\text{model}}(\mathbf{k})$:

$$\mathbf{k}^c = \arg \min_{\mathbf{k}} (err_y(\mathbf{y}, \mathbf{y}^{\text{model}}(\mathbf{k}))) \quad (1)$$

where, in the most general case, the error function $err_y(\mathbf{y}, \mathbf{y}^{\text{model}}(\mathbf{k}))$ is a monotonically increasing function of $\|\mathbf{y} - \mathbf{y}^{\text{model}}(\mathbf{k})\|$; the particular norm to be used may vary, and usually is chosen depending on the peculiar characteristics of the measured signals. If a purely model-based technique is employed, the computation of $\mathbf{y}^{\text{model}}(\mathbf{k})$ is usually expensive. The need to evaluate the error function iteratively within an optimization algorithm leads to computational times incompatible with real-time execution; additionally, the definition of a proper error function may be challenging. Conversely, data-driven strategies are faster, but require large datasets for training, as highlighted by Booyse et al. [44]. Such amounts of field data are often unavailable, especially during the design and validation of equipment, since their collection can only be carried out with several thousands of hours of operation of such equipment. The health condition \mathbf{k} determined with FDI is employed for the estimation of Remaining Useful Life.

The RUL of a system is the remaining time until the system will no more be able to meet its functional or performance requirements, that is, the time when the system will not be able to perform its function either at all or within the design performance parameters [45, 46]. This definition can be formalized as:

$$\begin{aligned} \text{RUL} &= \max(t) \\ \text{s.t. } \phi_a(\mathbf{k}(t)) &= \text{"healthy"} \end{aligned} \quad (2)$$

where $\phi_a(\mathbf{k})$ is an assessment function. $\phi_a(\mathbf{k})$ is a binary valued function assuming the possible values "healthy" or "faulty" that determines whether the fault vector \mathbf{k} corresponds to a healthy system or not (i.e. whether the system is still compliant to its functional and performance requirements).

With the traditional approach to life cycle management, the system useful life is computed *a priori* in the design phase, solely from the probabilistic combination of components failure rate. This strategy does not account for the real evolution of the components health status, and then produces estimates affected by a very large uncertainty interval [1, 2, 3, 4]. Popular approaches to RUL estimations aim at obtaining a more precise estimate of the system life either by extrapolating the current fault propagation rate [47], or by employing a model of damage growth until the damage condition reaches a threshold. In [48], a statistical approach combines a semi-markov model and the Maximum Likelihood Estimation (MLE) method to infer a degradation model for the equipment. Nascimento and Viana [49] discuss the use of recurrent neural networks

merging physics-informed and data-driven knowledge to model the time evolution of structural fatigue. Jacazio et al. [50, 51] propose to employ particle filtering to estimate the system RUL; in [52] particle filtering is combined with Canonical Variate Analysis (CVA) and Exponentially Weighted Moving Average (EWMA) in order to determine the RUL of rotating equipment. However, these methods often require a significant computational effort, or may be highly influenced by the effect of uncertainty in the estimation of the fault condition. Additionally, the definition of a proper critical failure threshold may be difficult: usually individual thresholds are set for each considered failure mode, not accounting for the combined effect of multiple faults. These can affect the system performance in a different way than the linear superposition of the effects of individual faults. As a result, a more general and comprehensive definition of a critical failure level may be needed.

Active research in PHM aims to enable early estimate the RUL, much in advance to the actual failure event, in order to allocate time for the optimal planning of maintenance strategies and for the logistics of fleet management. This motivates the interest for advanced FDI procedures to detect incipient faults at their early stages, before the system-level performances of the equipment start becoming significantly and adversely affected. To capture incipient faults, we specifically developed an importance sampling strategy (Section 3.1) for the computation of the dataset needed for training the machine learning tools. The proposed technique is intended to get denser sampling for small faults, where most useful information is expected. Nevertheless, the choice of an adequate sampling procedure is problem dependent, and other sampling strategies are used in different prognostic applications, such as estimating the probability of rare failures [53].

Both FDI and RUL estimation tasks imply the execution of a system emulator: usually this model is associated with an expensive computational effort. Therefore, most existing model-based [21, 22, 23] and data-driven [26, 54, 27] strategies are not suitable for real-time execution. Specifically, we wish to perform the FDI and RUL estimation tasks on-board, which requires to meet the hardware resources limitations to achieve a nearly real-time process. Therefore specific strategies are needed to achieve such computational efficiency and to meet these constraints.

3. Methodology

Our methodology proposes specific combinations of physics based knowledge and machine learning techniques into an original computational framework. Our hybrid

framework merges physics and data, and comprises efficient computational procedures to cover all the three phases of the flow described in Figure 1 in real-time. Specifically,
190 offline we merge physics and data to compute surrogate models, which are then employed to speed up the online computations. Figure 2 provides schematic representation of our hybrid framework for the entire prognostics flow.

For the first phase of signal acquisition and storage, we aim to reduce the data required to store a system output signal $y(\mathbf{k}, t)$. A uniform standard acquisition sampling
195 with a suitable frequency produces a vector $\mathbf{y}(\mathbf{k})$ whose size is impractical for the storage and subsequent processing. For example, monitoring a single electromechanical actuator may imply the acquisition of currents and voltages with frequencies in the order of tenths of kilohertz, resulting in a datarate up to several MB/s. To address this issue, offline we define an optimal signal compression in two steps: projection based model
200 reduction (Proper Orthogonal Decomposition) and unsupervised machine learning (Self-Organizing Maps) are combined to determine a set of informative components of $\mathbf{y}(\mathbf{k})$ to store and process. Online, the compressed output $\hat{\mathbf{y}}(\mathbf{k})$ is a vector containing only the selected informative components of $\mathbf{y}(\mathbf{k})$. To improve robustness against measurement uncertainty, $\hat{\mathbf{y}}(\mathbf{k})$ is not fed directly to the subsequent phases. Online we adopt Gappy
205 POD to reconstruct POD coefficients from the compressed representation of the signal $\hat{\mathbf{y}}$; those are used as input for fault estimation.

The second phase is the Fault Detection and Identification (FDI): this step aims at identifying the health condition of the system, i.e. the specific fault vector \mathbf{k} , with limited computational resources. Offline, we use supervised machine learning (Multi-
210 Layer Perceptron) to compute a model for the fault condition \mathbf{k} as a function of the coefficients $\boldsymbol{\alpha}$ of the POD expansion. Online we use the surrogate model learned offline to estimate the fault vector \mathbf{k} from the reconstructed POD coefficients.

The third phase is the RUL estimation. Here, a simple model of damage propagation is evaluated to compute an estimate of the remaining life of the system. In this
215 phase, an estimator of the computationally expensive assessment function is needed as a stopping criterion for the damage propagation. To meet the constraints in terms of time and available processing power, we propose the use of a binary classifier, specifically a Support Vector Machine (SVM) to replace the complete assessment function. The SVM is trained offline on the reference dataset and employed online to speed up the RUL
220 estimation.

The strategy is schematically illustrated in Figure 2: Sections 3.2, 3.3 and 3.4 describe the two steps signal compression, the FDI phase and the RUL estimation procedure,

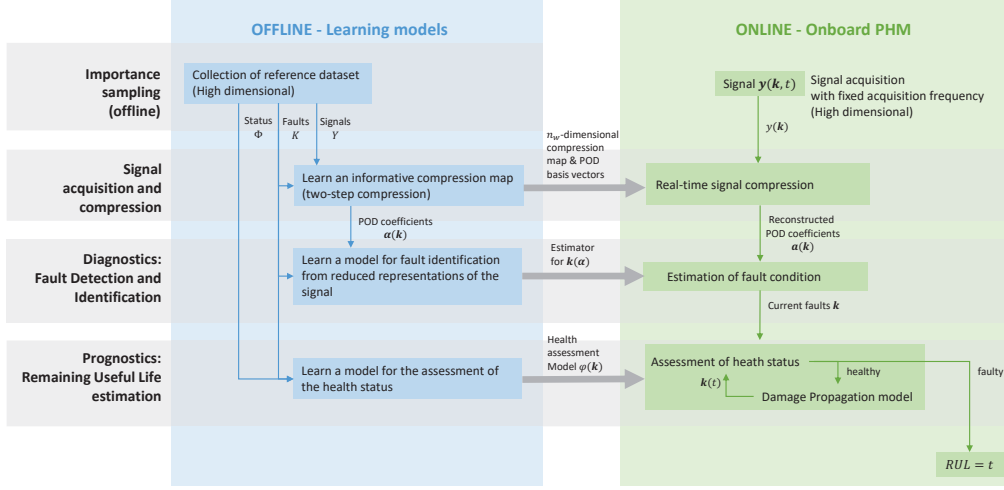


Figure 2: Schematic representation of our hybrid framework for the entire prognostics flow

respectively; Section 3.1 describes the collection of the reference dataset used to learn the models (offline).

225 3.1. Importance sampling via particular scaled latin hypercube strategy

Learning the surrogate models requires the collection of a training dataset representative of the system behavior under the expected operating conditions. It can be collected according to a variety of sampling strategies, depending on the specific problem at hand. For the application discussed in this paper we propose a particular importance sampling
230 method. The reference data can be collected from a variety of different sources including historical data, numerical simulations of the systems through evaluations of high fidelity models, or experimental measurements. In this paper we use data from high-fidelity, accurate models of the systems, considered as a ground-truth reference.

The data set is organized into the following quantities of interest, as also shown in
235 Figure 3:

- FAULT CONDITIONS MATRIX K : $K = [\mathbf{k}_1, \mathbf{k}_2, \dots, \mathbf{k}_{n_s}]^\top$ is a n_s -by- n_k matrix containing in its rows the n_s fault combinations \mathbf{k}_i collected in the dataset. \mathbf{k}_i are the n_k -dimensional fault vectors that carry the information about the system health condition. Each fault vector encodes in its elements a combination of progressive damages of the system. The elements of the fault vectors are, in general, related to physical quantities of different nature, such as the friction coefficient between two sliding surfaces of a mechanism, the mechanical play of a transmission, or the resistance of an electric circuit; to avoid the effect of different scales and inhomogeneous

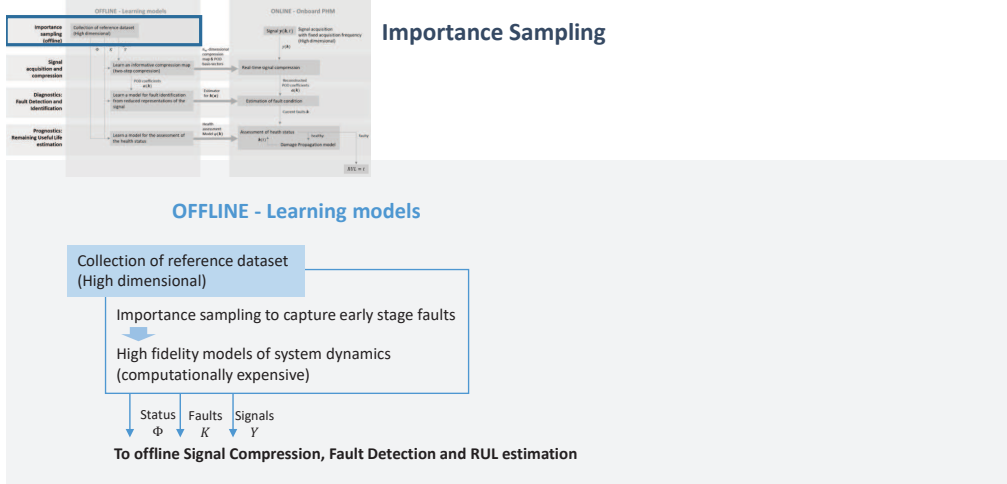


Figure 3: Importance sampling of high fidelity data

geneous measurement units, we chose to normalize those quantities, in order to bound the elements of \mathbf{k}_i between 0 and 1.

- MEASURED SIGNALS MATRIX Y : $Y = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{n_s}]$ is a n_e -by- n_s matrix containing in its columns the output vectors $\mathbf{y}(\mathbf{k}_i)$ of the system. $\mathbf{y}(\mathbf{k}_i)$ are the output signals of the system for each sampled fault combination \mathbf{k}_i , expressed in the form of n_e -dimensional vectors by capturing them at a fixed acquisition frequency.
- ASSESSMENT FUNCTION MATRIX Φ : $\Phi = [\phi_{a,1}, \phi_{a,2}, \dots, \phi_{a,n_s}]$ is a 1-by- n_s matrix containing the values of the assessment function corresponding to each fault combination. $\phi_{a,i} = \phi_a(\mathbf{k}_i)$ is the value of the assessment function for the fault vector \mathbf{k}_i .

The particular sampling strategy for collecting the dataset is problem dependent. The application presented in this paper requires to detect the early signs of a system damage with high accuracy in order to determine the Remaining Useful Life in advance enough to plan corrective actions [55]. For this reason, the training dataset shall be denser of health conditions close to the nominal one, that is, when either no faults are present or faults are small and do not have a significant effect on the system performances. In this context, we implemented a form of importance sampling strategy through a scaled latin hypercube. This technique is meant to increase the density of sampling points near the nominal condition; this allows to collect more informative samples to capture small and incipient faults.

For example, the assessment function is expected to assign “healthy” labels to fault
 265 conditions \mathbf{k} near the nominal condition \mathbf{k}_0 , and “faulty” labels to fault conditions far
 from the nominal one. Being the number of parameters large, a uniform distribution of
 points in the domain of \mathbf{k} would result in a small fraction of points near the nominal
 condition (i.e. “healthy” fault combinations), and the surrogate assessment function
 would be difficult to train. This sampling strategy is employed, in combination with
 270 the physics-based models described in Section 4.2, to obtain the reference data of the
 matrices \mathbf{K} , \mathbf{Y} and Φ .

3.2. Signal Acquisition and Compression

The first obstacle that results in a computationally expensive process is the high
 dimensionality of \mathbf{y} . Any output signal measured from the system (i.e. currents and
 275 voltages of an electrical machine, hydraulic pressures, accelerations etc.) with a uniform
 sampling is a vector composed by $n_e = \Delta t f_s$ elements, where f_s is the acquisition
 frequency and Δt is the observation time. In most applications, to capture the required
 amount of information, f_s needs to be in the order of tenths of kilohertz and n_e can
 easily be in the order of several thousands. This requires large storage capabilities
 280 and processing power for the subsequent Fault Detection and Identification phase. For
 instance, the use of common least squares methods for the subsequent FDI phase involves
 the QR factorization of an n_e -by- n_k matrix, where n_k is the number of fault parameters,
 which computational cost ($\mathcal{O}(n_e^3)$) makes the online/on-board execution impractical for
 the large n_e of common output signals \mathbf{y} .

285 To overcome this problem, we use a particular strategy for the optimal selection of
 a small number of the sampling points to retain, store and process. A first approach
 of this kind was introduced by Mainini and Willcox [56], where Proper Orthogonal
 Decomposition and Self-Organizing Maps are combined for the optimal placement of
 sensors for on-board assessment of structural capabilities. In this work a similar approach
 290 is adopted to reduce the computational burden associated to the FDI task by reducing
 the problem dimensionality to $n_w \ll n_e$.

The process is shown schematically in Figure 4. The selection of the signal points to
 process online is computed offline through a two-step procedure to learn a compression
 map: it combines low order representations of high dimensional data (projection based
 295 model reduction) and machine learning techniques (unsupervised machine learning) to
 identify the most informative instants of time of the measured signal to be stored for
 the subsequent online Fault Detection and Identification.

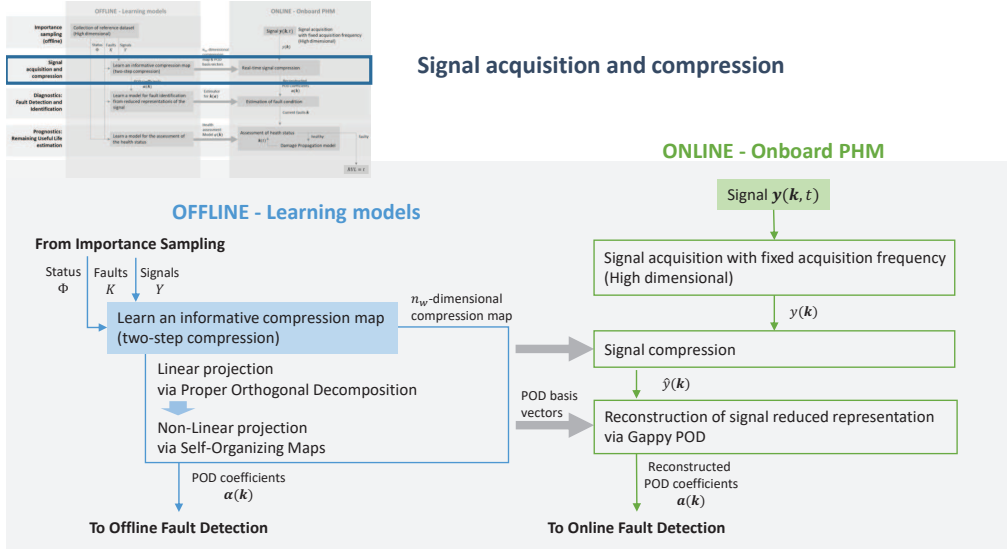


Figure 4: Flow chart of the signal compression strategy

3.2.1. Offline: Learn an informative compression map

We aim to determine an informative compression map for the signal $y(k, t)$. Only those points of the signal will be stored and processed online, reducing the required computational resources. The offline signal compression process takes as input a set of measurements from the system. For this purpose, we use the fault conditions in K and the associated output signals (snapshots) y assembled into the columns of the n_e -by- n_s measurement matrix Y . Through the proposed two-steps offline compression strategy, we determine the set of informative time-locations for the signal. The compression process is articulated into the two steps leveraging Proper Orthogonal Decomposition (POD) and Self-Organizing Maps (SOMs) respectively.

Linear projection via Proper Orthogonal Decomposition (POD). The first step of compression employs data gathered by simulations or experimental campaigns, with the purpose of obtaining a reduced order representation of the system. This reduced model is computed through Proper Orthogonal Decomposition (POD). POD [57, 58, 59, 60, 61] is a projection based reduced order modeling technique commonly employed to obtain low dimensional representations of high dimensional quantities, through the identification of underlying features (in the form of dominant modes). One of the most employed strategy is the method of snapshots [62]. Data points are represented in the n_e -dimensional space, and the dominant modes are the principal directions along which the points are dispersed. The eigenvalues associated with each mode encode the variance of the data set along that direction.

We apply POD to the measurements matrix Y , in order to extract the modes associated with the largest eigenvalues, that explain most of the variance of the dataset. The POD modes constitute an orthonormal basis for the measured signals collected in Y [63, 64]: it is optimal in the least squares sense and can be computed through Singular Value Decomposition (SVD) of matrix Y to obtain a representation of each training signal as:

$$\mathbf{y}(\mathbf{k}) = \mathbf{y}_0 + \sum_{i=1}^{n_s} \mathbf{v}_i \alpha_i(\mathbf{k}) \quad (3)$$

where \mathbf{y}_0 denotes a reference signal (in our application the system output in nominal conditions), n_s is the number of snapshots, equivalent to the total number of POD modes, \mathbf{v}_i are the POD modes and $\alpha_i(\mathbf{k})$ are the coefficients of the POD expansion. The eigenvalue λ_i associated to each mode \mathbf{v}_i is a measure of the dispersion of the high dimensional training data along the direction defined by the mode itself: by considering only the first n_m modes of the POD expansion (equation 3), the fraction of retained information is given by the cumulative sum of the eigenvalues $\sum_{i=1}^{n_m} \lambda_i / \sum_{i=1}^{n_s} \lambda_i$. The POD modes are ordered according to their associated eigenvalue, so we can truncate the expansion to retain only the first $n_m \ll n_s$ modes and to get a low dimensional representation of the signal. If the cumulative sum of the retained eigenvalues is close to 100%, the information lost in the compression is accordingly small; additionally, if the training set is statistically representative of the actual system behavior, the same compression can be applied to signals not belonging to the training set.

Through POD we obtain a set of basis vectors \mathbf{v}_i and the associated coefficients α_i for each column of the training set Y . Bases and coefficients of the POD expansion are employed both offline and online in the following steps of our procedure.

Non-linear projection via Self-Organizing Map (SOM). In the second step of signal compression we use the first n_m POD basis vectors to find a compressed representation of the basis vectors themselves through a Self-Organizing Map (SOM). This compressed representation is identified in the form of a set of $n_w \ll n_e$ highly informative time-locations for storing and processing the signal.

A Self-Organizing Map is a single layer neural network that can be used to identify subsets of similar data through unsupervised competitive learning [65, 66, 67]. The n_w neurons of the SOM have representations in the input space as weight vectors whose values are updated during the training. In this case the input space is the $n_m + 1$ dimensional parameter space given by the time coordinate t and the n_m modes of the POD (see Section 3.2.1). The training set for the SOM is given by the first n_m modes

of the POD and the corresponding time coordinate \mathbf{t} , arranged in an n_e -by- $(n_m + 1)$ array \mathbf{T} :

$$\mathbf{T} = [\mathbf{t}, \mathbf{v}_1, \dots, \mathbf{v}_{n_m}] \quad (4)$$

where \mathbf{t} and \mathbf{v}_i are column vectors of n_e elements. During training, all the points of the training set are presented to the network multiple times (*epochs*) in a different order, to avoid a training bias. For each training point $\boldsymbol{\tau}_i$ (the i -th row of \mathbf{T}), a winner neuron l is the one whose weight vector \mathbf{w}_l is the closest to the input point:

$$l = \arg \min_j (\|\boldsymbol{\tau}_i - \mathbf{w}_j\|) \quad (5)$$

where $\|\cdot\|$ denotes the L_2 norm adopted as similarity metric for the study discussed in this work. The neighbor neurons are activated according to a neighborhood function, defined in the space of the topological representation of neurons, usually decreasing with the distance from the winner neuron and symmetric about the winner neuron [65, 67]. One of the key characteristics of SOMs is that during training, the weight vectors of the neurons are updated to represent a non-linear projection of the high dimensional training data (the first n_m POD modes) onto a lower dimensional manifold, where prototype vectors encode representative points of the POD modes [56, 68]. From this perspective, self-organizing maps can be seen as a nonlinear generalization of POD, where the weight vectors of the neurons correspond to the POD eigenvectors, while the radius of each cluster of data is related to the associated eigenvalue [69]. As a result, once training is complete, the first components of all the n_w weight vectors encode the most informative time-locations $\hat{\mathbf{t}}$ for the signal y .

3.2.2. Online

Online, these specific values in the time coordinate are used to store and process the measured signals. Those signals are acquired in real-time by sensors installed on the monitored equipment, with a constant frequency high enough to capture the information related to the considered progressive failures. This results in a continuous data stream from the sensor to the acquisition electronics, with a rate than can reach the order of megabytes per second for a single sensor. Data measured during the observation time Δt could be stored in an n_e dimensional vector \mathbf{y} . However, leveraging the two steps compression computed offline, we can store only the n_w informative components. The resulting compressed signal is a n_w -by-1 vector $\hat{\mathbf{y}}(\mathbf{k})$, with $n_w \ll n_e$, that preserves the useful information regarding the faults affecting the system. As such, $\hat{\mathbf{y}}(\mathbf{k})$ is used as informative input for the subsequent phase.

In principle, the compressed signal $\hat{\mathbf{y}}(\mathbf{k})$ can be directly processed for the detections and identification of the associated fault condition; however, $\hat{\mathbf{y}}(\mathbf{k})$ carries measurement noise and a random error on the signal would directly affect the identification of faults (FDI). To mitigate the effect of measurement noise, we propose to compute the POD coefficients $\boldsymbol{\alpha}(\mathbf{k})$ from the compressed signal $\hat{\mathbf{y}}(\mathbf{k})$ via Gappy POD and move the FDI task onto the reduced space identified by the POD in Equation 3.

Gappy POD is a procedure derived from Proper Orthogonal Decomposition and is commonly used for the recovery of incomplete data [70, 71, 72, 73]. Given $\hat{\mathbf{y}}$, the reconstructed signal can be obtained as a linear superposition of the first n_m POD modes computed offline (Equation (3), Section 3.2.1). The expansion coefficients $\alpha_j(\mathbf{k})$ are computed to minimize the squared error between the known points of the compressed signal $\hat{\mathbf{y}}$ and its reconstruction in the n_w informative elements.

An approximation of the uncompressed signal could be recovered as the linear combination of the first n_m modes weighted by the coefficients $\boldsymbol{\alpha}$, as per Equation (3). However, for the purpose of this work, we are not interested in the reconstruction of the original signal, but exclusively in recovering the POD coefficients $\boldsymbol{\alpha}(\mathbf{k})$. These are employed in the next step to identify an estimate of the fault condition \mathbf{k} , as they encode the physical information of the system health in a compressed representation, easier to be dealt with by the FDI process.

3.3. Fault Detection and Identification (FDI)

The Fault Detection and Identification (FDI) phase of our strategy aims at identifying the health condition of the system (the specific fault vector \mathbf{k}) from the information of the compressed signal $\hat{\mathbf{y}}$. This task is a parameter identification problem and is formulated as an optimization problem (Equation 1). However, the use of common gradient-based or meta-heuristic optimization algorithms for parameter identification requires the iterative evaluation of system emulators that are frequently expensive. The accuracy level required for a reliable identification of the fault condition demands for the evaluation of models of the dynamical system that are usually too computationally expensive to evaluate online [74, 75].

To meet the efficiency requirements of time constrained online evaluations, we employ feedforward Neural Networks to estimate the fault vector \mathbf{k} from the compressed representation provided by the reconstructed POD coefficients $\boldsymbol{\alpha}$. Specifically, a Multi-Layer Perceptron (MLP) maps the POD coefficients to the fault vector. As shown in Figure 5, The FDI task is split into an offline phase, in which the Neural Network model

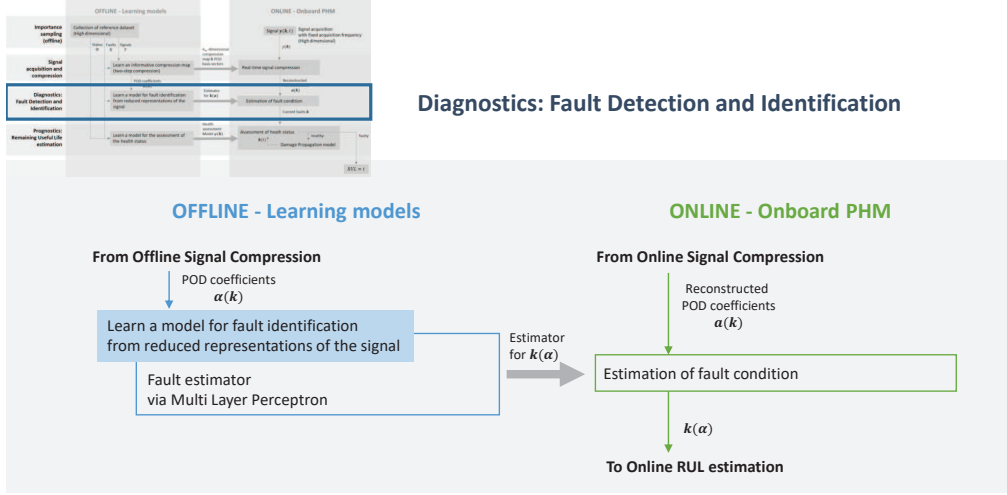


Figure 5: Flow chart of the Fault Detection and Identification process

is trained, and an online phase, in which it is evaluated to estimate \mathbf{k} .

3.3.1. Offline: Learn a model for fault identification from reduced representations of the signal

420 In order to estimate the fault vector \mathbf{k} within a computational time suitable for real-time evaluation, we train offline a neural network to obtain a surrogate model for the fault condition \mathbf{k} from the low dimensional representations of the measurement provided by $\alpha(\mathbf{k})$.

The specific implementation of Multi-Layer Perceptron (MLP) adopted in this paper
425 is characterized by a standard feedforward architecture, with a single hidden layer; more complex machine learning strategies may be tested in future works. The network receives in input the POD coefficients $\alpha(\mathbf{k})$ and returns the fault vector \mathbf{k} . The hidden layer has n_h neurons with sigmoid activation function, while the output layer has n_k neurons with a linear saturated activation function. The specific choices for the activation functions
430 reflect the physical characteristics of the input and output variables of the problem.

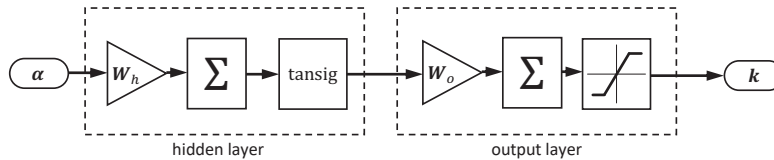


Figure 6: Architecture of the two layer perceptron employed for the FDI task. W_h are the weights of the hidden layer neurons, W_o are those of the output layer

The saturation is introduced to account for the bounds of the output fault vector whose components are bounded between 0 and 1, as defined for our application (Section 4.2). Figure 6 shows the complete network architecture. The weights and biases of each neuron are determined during training, in order to tune the network to approximate the expected output for a training data set. The training dataset is given by the input-output pairs (α_i, \mathbf{k}_i) , including the fault vectors \mathbf{k} collected with the sampling strategy detailed in Section 3.1 and the associated POD coefficients computed as per procedure described in Section 3.2.1.

During training, a Levenberg-Marquardt backpropagation algorithm [76, 77] updates the weight and bias variables to minimize a performance function, defined as the mean squared error between the expected and actual output of the network for the training set. The training is stopped when either the maximum number of epochs is reached or the performance gradient decreases below a threshold. Once training is complete, we obtain a model to map from α to \mathbf{k} using the n_s training signals.

3.3.2. Online: Estimation of the fault condition

The input of this phase are the POD coefficients α estimated via Gappy POD, as per Section 3.2.2. Those are fed to the MLP model learned offline, in order to estimate the faults \mathbf{k} . This approach is preferred over the straight adoption of a neural network over the full dimensional dataset because the compression allows to significantly reduce the computational cost, both in training and in evaluation of the MLP [78]. The output of the FDI process is an estimate of the fault vector \mathbf{k} , to be employed in the subsequent RUL estimation.

3.4. Estimate the Remaining Useful Life (RUL)

The estimation of the Remaining Useful life is the last phase of the PHM process. We aim to complete it onboard, given the fault condition \mathbf{k} estimated through the FDI procedure discussed in Section 3.3.

In this paper we propose a strategy for RUL estimation relying on a damage tolerant approach to system design similar to that adopted for the estimation of fatigue life in aircraft structures. Leveraging the definition introduced by Equation (2), the health state \mathbf{k} detected at the measurement time $t_0 = 0$ is used as an initial value to compute the evolution of the health condition through a specific model for damage propagation. The damage propagation model is in the form of an Ordinary Differential Equation (ODE) whose evaluation provides the rate of damage growth as a function of the current system health and the operating and environmental conditions. An assessment function $\phi_a(\mathbf{k})$

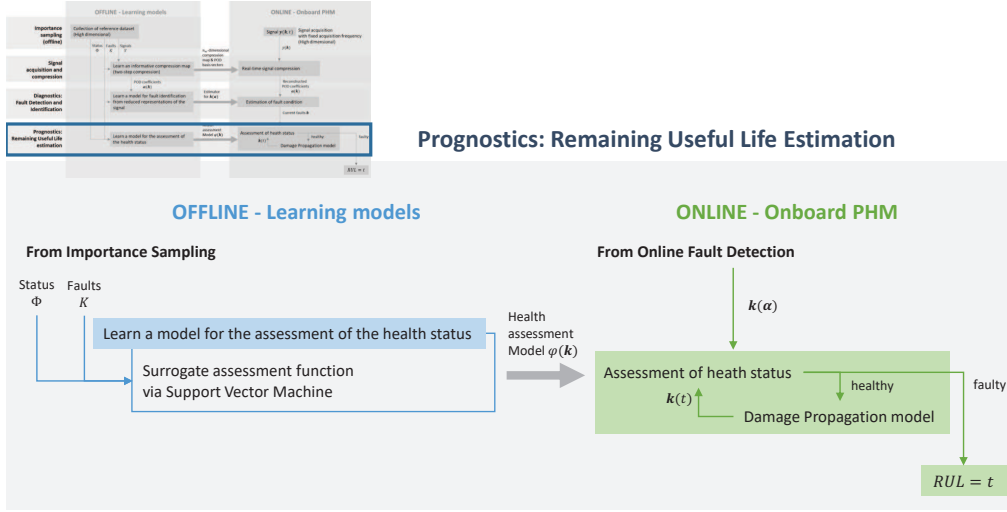


Figure 7: Flow chart of the RUL estimation process

is employed as a stopping criterion for the integration of the ODE model: according to the definition of Equation (2), it evaluates the system performances for each value assumed in time by the fault vector, to determine whether that specific fault vector is compatible with the system operation. When a faulty system is detected, the integration is stopped and the last time step is assumed as the system RUL.

The damage propagation model in the form of ODE may not be computationally expensive since a limited number of factors can be accounted for in the estimation of fault propagation rate (heat dissipation, vibrations, degradation of surfaces). Conversely, the evaluation of the assessment function $\phi_a(\mathbf{k})$ usually implies multiple executions of the models of system dynamics, which are expensive and unsuited for nearly real-time applications. To address this limitations we use supervised learning techniques (specifically Support Vector Machines, SVMs), which are trained offline on reference data to obtain surrogates of the assessment function to employ online. The online-offline structure of the proposed RUL prediction process is represented in Figure 7.

3.4.1. Offline: Learning a model for the assessment of the health status

The assessment function $\phi_a(\mathbf{k})$ is essentially a binary classifier: it analyzes the behavior of the system in presence of the fault combination \mathbf{k} and determines whether in this condition the functional and performance requirements are met. This process usually involves the resolution of a dynamical model of the system and requires a high computational effort. To meet the time constraints for on-board estimation, we adopt a standard implementation of a Support Vector Machine (SVM) as a binary classifier and

a surrogate for the assessment function $\phi_a(\mathbf{k})$ to run online.

A Support Vector Machine [79, 80, 81, 82] is a machine learning paradigm commonly used for data classification and regression. During its training process, the SVM identifies an optimal linear boundary to separate the two classes of the input data points. In our application, as the classes are not linearly separable, a polynomial kernel function is used to map the input points to a transformed predictor space where a linear boundary can be identified.

After training, new input points are classified according to the sign of a score function that quantifies the distance of the input from the boundary. The sign of the score function constitutes our surrogate for the assessment function $\phi_a(\mathbf{k})$, suitable to run online and used for the RUL estimation process.

3.4.2. Online

The SVM trained offline is employed as a surrogate assessment function to speed up the real-time computations involved in RUL estimation.

The fault propagation rate can usually be described by an Ordinary Differential Equations (ODE) model, accounting, in the most general case, for the current health condition of the system, the environmental and operating conditions, and the expected mission profile. The evolution of the system health status is computed through the numerical integration of this ODE model. The initial condition is set as the fault vector \mathbf{k} estimated in the previous FDI phase. At each integration time step t_i the surrogate assessment function $\tilde{\phi}_a(\mathbf{k})$ determines whether the current fault vector $\mathbf{k}(t_i)$ corresponds to a healthy system or not. Since this has to be evaluated iteratively, the use of the full model-based assessment function $\phi_a(\mathbf{k})$ would result in long computational times, not suitable for real-time evaluation. When a faulty condition is detected by the (surrogate) assessment function, the integration is stopped. At this point we can assume:

$$\text{RUL} = t \quad (6)$$

where t is the current integration time. That is, the Remaining Useful Life of the system is assumed to be equal to the timestep when the system transitioned from a healthy condition to a faulty one.

4. Diagnostics and Prognostics of Aircraft Actuation Systems

We develop and demonstrate our methodology for the real-time prediction of Remaining Useful Life for aircraft actuators. Actuation systems involve the interaction of

several, heterogeneous engineering disciplines, such as electronics and software, electrical machines, mechanical systems, hydraulics, structures, thermal dissipation problems, fluid dynamics, vibrations, and tribology. Additionally, damage propagation may be affected by operating conditions that are not completely predictable, as opposed to similar
520 actuation devices employed for static applications, such as for industrial automation. As a result, faults affecting such systems have effects on performances that are difficult to predict, and the associated detailed models are computationally expensive.

Computational methodologies intended to address the open challenges of diagnostic
525 and prognostics for aircraft actuators are of critical interest for mission reliability and cost effectiveness of the whole fleets. Failures in such subsystems can lead to increased down time of the vehicle and may require risk mitigation, since most of these devices are safety critical.

Similar actuation technologies are employed in different fields of engineering, sharing
530 the same open challenges regarding health monitoring and management. As an example, the failure of an actuator on a production line can require the shutdown of the whole production line for repair, with significant income losses. Given the multidisciplinary nature of the considered application, computational techniques developed to address the prognostic analysis of servo actuators can be extended to deal with health monitoring
535 of similar components not necessarily within the domain of actuation systems. Specifically, any dynamical system involving power electronics, electrical machines, sensors, or mechanical and hydraulic power transmission, can be a potential application of the proposed health monitoring strategy.

4.1. Problem Setup

The particular application addressed in this paper is the real-time estimation of
540 Remaining Useful Life for an Electromechanical Actuator (EMA) for aircraft Flight Control Systems from on-board measurement of the motor current. EMAs [83, 84, 85, 86] exploit an electric motor coupled to a mechanical transmission to convert power from the aircraft electric system into mechanical power to move the flight control system
545 aerodynamic surfaces. Those actuation systems are commonly employed in small scale UAVs and for secondary flight controls of larger manned aircraft.

The use of EMAs in aircraft systems is gradually spreading as they promise to achieve
significant reductions in operating costs, through weight and fuel savings, and simplification of maintenance interventions [87, 88, 89, 90]. On the other hand, the inherent
550 nature of some EMA failure modes (such as jamming of the mechanical transmission)

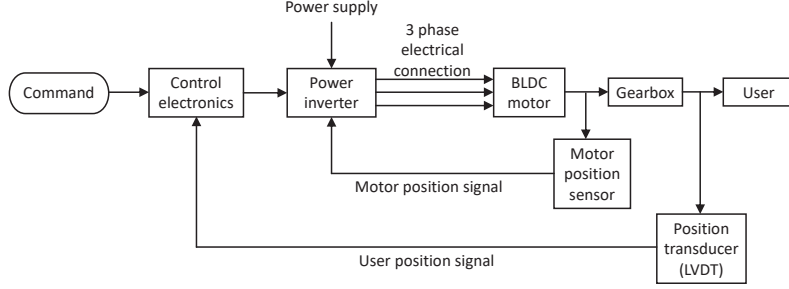


Figure 8: Architecture of the Electromechanical Actuator

holds them back from safety critical applications: in this context, PHM techniques would increase the safety of operations and permit to exploit the full potential of all-electric system architectures [91, 92, 93].

The block diagram of Figure 8 shows the architecture of the considered EMA. This includes a BrushLess Direct Current (BLDC) electric motor along with its power and control electronics, a transmission to increase the torque for the user, and a Linear Variable Differential Transformer (LVDT) sensor to close the position feedback loop.

For this study, we consider the effects of five different failure modes, chosen among the most common for EMAs [94, 95, 96] and characterized by a slow propagation rate, to allow an effective estimation of the Remaining Useful Life. Those are namely variations in dry friction (k_1) and backlash (k_2), partial short circuit of each of the three stator phases ($k_{3,4,5}$), rotor static eccentricity ($k_{6,7}$) and controller proportional gain drift (k_8); this results in a fault vector $\mathbf{k} = [k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8]$ of $n_k = 8$ elements. The complete definition of the fault vector \mathbf{k} is illustrated in Table 1. We chose the stator envelope current as the informative variable y to monitor for the prognostic analysis. The reason for this choice is twofold: it is highly sensitive to a number of fault modes and can be easily measured in a physical system, or is already measured with the purpose of closing a current feedback loop.

Table 1: Definition of the fault vector \mathbf{k}

fault parameter	fault mode	lower bound ($k_i = 0$)	upper bound ($k_i = 1$)
k_1	dry friction	nominal friction	300% of nominal friction
k_2	backlash	nominal backlash	100 times nominal backlash
k_3	phase A short circuit	no short circuit	full short circuit
k_4	phase B short circuit	no short circuit	full short circuit
k_5	phase C short circuit	no short circuit	full short circuit
k_6	rotor eccentricity	no eccentricity	eccentricity equal to air gap width
k_7	eccentricity phase	-180°	180°
k_8	proportional gain drift	50% of nominal gain	150% of nominal gain

4.2. Physical models of system dynamics

Two models of the actuator with different fidelity are employed: a High Fidelity (HF) model (Section 4.2.1) is used only offline as the source of reference data: for the high accuracy of the model this dataset can be used as a good emulator of ground truth reference data [97]. A Low Fidelity (LF) model (Section 4.2.2) is used within the assessment function to determine the frequency response of the actuator and compare it with its requirements. The accuracy of the LF model is considered suitable for the sake of this task; the computational cost of the HF model would be impractical for the iterative evaluation required by the assessment function, even for offline execution. Section 4.2.3 describes the model for the damage propagation rate, employed for the RUL estimation process.

4.2.1. High Fidelity (HF) model

The HF model is the accurate dynamical model of the EMA, simulating in detail the physical behavior of the actuator subsystems and components. The model accounts for the effects of Pulse Width Modulation (PWM) three-phase current control logic of the motor power electronics, and includes a complete lumped parameters model of the electromagnetic coupling between stator and rotor. This HF model is employed as a simulated test bench, to compute the reference data in replacement of a physical system.

The architecture of the model is shown in the block diagram of Figure 9. The Actuator Control Electronics (ACE) block compares the commanded and actual positions

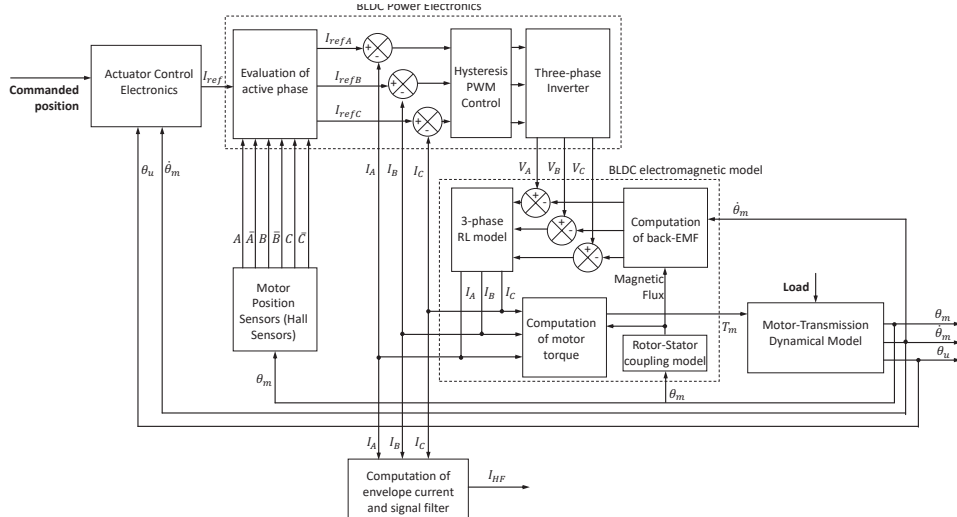


Figure 9: Detailed block diagram of the High Fidelity Model

to compute a reference current signal I_{ref} for the motor. The BLDC Power Electronics subsystem simulates the three-phase inverter used to power the BLDC motor. It applies the needed voltages (V_A , V_B and V_C for the three motor phases, respectively) on the motor windings to produce the currents I_A , I_B and I_C , managing the phase commutation according to the ACE and to the rotor position θ_m . The BLDC Electromagnetic model contains the rotor-stator magnetic coupling model, to compute the counter-electromotive force, and the motor torque T_m . The motor-transmission dynamical model is a nonlinear second-order model of the actuator mechanics; it computes the angular positions θ_m and θ_u of motor and user respectively, accounting for backlash, dry friction and mechanical endstops. The signal acquisition block computes the envelope I_{HF} of the three phase currents, low-pass filtered to suppress high frequency noise: the resulting output signal y is employed for the PHM analysis.

The HF model is implemented in the Matlab-Simulink simulation environment. Its accuracy comes at the expense of a relatively high computational effort. The simulation of a reference 0.5 seconds test signal takes about one minute on a common laptop PC, making this model unsuitable for real-time applications.

4.2.2. Low Fidelity (LF) model

The LF model is a simplified dynamical representation of the same EMA, with complex subsystems represented by simpler blocks. This model is used iteratively to compute the assessment function $\phi_a(\mathbf{k})$ (see Sections 3.4 and 4.2.3).

The block diagram of the model is shown in Figure 10. The most computationally expensive sections of the HF model are the three-phase inverter and the computation of the magnetic flux across the air gap. Those subsystems are differently handled in the LF model, replaced by a first order DC model whose governing equation directly relates the motor current I_m , voltage V_m and torque T_m through the back-Electromotive Force (back-EMF) coefficient κ_v :

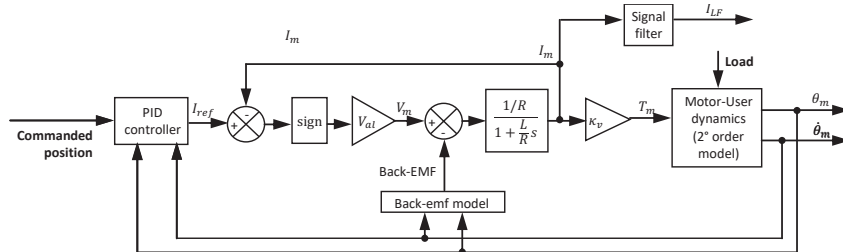


Figure 10: Block Diagram of the Low Fidelity Model

$$RI_m + L\dot{I}_m = V_m - \kappa_v\omega \quad (7)$$

$$T_m = \kappa_v I_m \quad (8)$$

615 where R and L are the stator resistance and inductance, and ω is the motor angular speed. This simplified model does not allow accounting directly for the electric faults of partial short circuit and rotor eccentricity. This paper follows the approach proposed by Berri et al. [97], evaluating the electrical faults through modulating functions applied to the back-EMF computation.

620 The computational time needed for the execution of the LF model is about two orders of magnitude lower than that of the HF model. The average root mean square discrepancy between the HF and LF model is in the order of 1%.

4.2.3. Damage propagation model

The model of damage propagation is an ODE based model assuming that each fault 625 grows linearly with the others. It is expressed as:

$$\dot{\mathbf{k}}(t) = \mathbf{F}\Delta\mathbf{k}(t) + \boldsymbol{\epsilon} \quad (9)$$

where $\dot{\mathbf{k}}$ denotes the fault growth rate, $\Delta\mathbf{k}(t) = \mathbf{k}(t) - \mathbf{k}_0$, \mathbf{k}_0 is the fault vector in nominal conditions, $\boldsymbol{\epsilon}$ is an independent identically distributed normal noise, and \mathbf{F} is a square matrix whose F_{ij} element expresses the influence of the j -th fault parameter on the growth rate of the i -th fault parameter. The matrix \mathbf{F} depends on the physics of the system, and can be identified from field data. The integration is stopped when the 630 assessment function $\phi_a(\mathbf{k})$ indicates that the fault condition \mathbf{k} achieves damage levels that jeopardize the system performance. The assessment function calls the LF model iteratively with a set of input frequencies to compute the system Bode diagram (it would be very impractical to use the HF model, even for offline computations). Then, the phase 635 margin, gain margin and cutoff frequency are compared to the thresholds imposed by performance requirements to determine whether the actuator is working correctly or not. The computational time of the assessment function on a common laptop PC is in the order of 10 seconds: despite the use of the LF model, the computational cost is not compatible with real-time RUL estimation; this motivates the need of developing a surrogate model for the assessment function to handle the task efficiently (Section 3.4.1).

5. Results and Discussion

This section discusses the results obtained with the application of the offline and online phases of the PHM methodology to the specific problem considered in this paper.

645 A training set is collected from data computed with the physics-based models of Section 4.2. Specifically, a set of $n_s = 10000$ fault combinations is computed with the importance sampling strategy described in Section 3.1, to obtain the matrix \mathbf{K} . Throughout the computational framework, 70% of this dataset is employed to train the algorithms, while 30% is used for testing and evaluation; the in-training test allows to mitigate the risk of overfitting. In our application, k_7 is usually excluded by the distribution rescaling process. This parameter encodes the phase of the rotor static eccentricity with respect to the stator windings; then, its probability distribution is necessarily uniform, and should not be modified. The HF model of the actuator (Section 4.2.1) is evaluated for each fault combination; a test command is employed, consisting in a linear chirp characterized by
650 a 0.5s duration, $5 \cdot 10^{-3}$ rad amplitude, 0Hz start frequency and 15Hz end frequency. The output signals $y(t, \mathbf{k})$ are acquired with a constant frequency of 20kHz (resulting in $n_e = 10001$), necessary to capture the information required by the FDI, and then assembled into the columns of the measurement matrix \mathbf{Y} . The assessment function described in Section 4.2.3 provided the “healthy” or “faulty” labels associated for each
660 fault combination, to be stored into the matrix Φ . Two additional validation sets, not used in the offline training of the algorithms, are assembled to assess the performance of the proposed strategy. A first validation set is computed to assess the offline training of surrogate models and the online compression and FDI (Section 5.1). This includes 500 fault combinations, sampled as per Section 3.1, and the associated signals and values
665 of the assessment function. A second validation set is employed to assess the online RUL estimation procedure (Section 5.2). This dataset includes 100 simulated run-to-failures, each with the record of fault combinations, the corresponding HF signals and assessment function values, and the RUL (with the associated uncertainty) computed through the damage propagation model described in Section 4.2.3. The validation of the
670 online procedure requires the initial faults to be small (i.e. near the nominal condition); otherwise, the system would be already faulty, its remaining Useful Life would be null, and the real time RUL estimation could not be properly tested. Therefore, this second test set is sampled with the procedure of Section 3.1 on a restricted domain in \mathbf{k} , to include mostly “healthy” conditions.

675 The following sections describe the application of our methodology to the problem of RUL estimation for aerospace EMAs, described in Section 4. In particular, Section

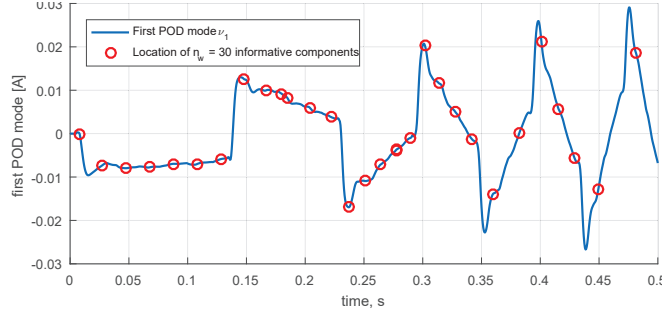


Figure 11: Placement of 30 sampling points over the first POD mode

5.1 shows the offline learning of surrogate representations of the models that describe the physics of the considered problem; Section 5.2 discusses the results of the online procedure. All computations were performed on a desktop PC with a i5 3330 quad-core
 680 processor @3.00GHz and 8GB of memory, running Windows 10 and Matlab R2016a

5.1. Offline: learning models for efficient surrogate representations

The following paragraphs discuss the application of the offline procedure to the specific problem addressed in this work: the results of the strategy for determining the optimal time coordinates to store and compress the measured signal, derived through
 685 the procedure described in Section 3.2; the performance of the model that maps the fault vector $\mathbf{k}(\boldsymbol{\alpha})$ as a function of the POD coefficients, derived according to Section 3.3.1; the outcome of the surrogate model for the assessment function $\phi_a(\mathbf{k})$, derived as per Section 3.4.1.

Two-steps data compression. The matrix \mathbf{Y} of of the training dataset is employed to
 690 learn the informative compression map described in Section 3.2. \mathbf{Y} is used to compute the POD expansion in the form of Equation (3) whose first n_m modes are retained and used for (i) optimal points selection for signal compression (determined offline) and (ii) POD coefficient reconstruction via gappy POD (to run online).

For the offline identification of the most informative signal points we use the two
 695 steps procedure of Section 3.2 that allows us to compute a set of $n_w = 30$ points. Those are optimally placed to capture the information of the first n_m POD modes. In this paper, the number of points is chosen to retain sufficient information for the FDI on the base of a previous investigation presented in [75]. Figure 11 shows an example of placement of the sampling points for $n_m = 1$: the points are not equally spaced in time,
 700 but rather tend to be placed by the algorithm in the most significant points to capture the shape of the mode.

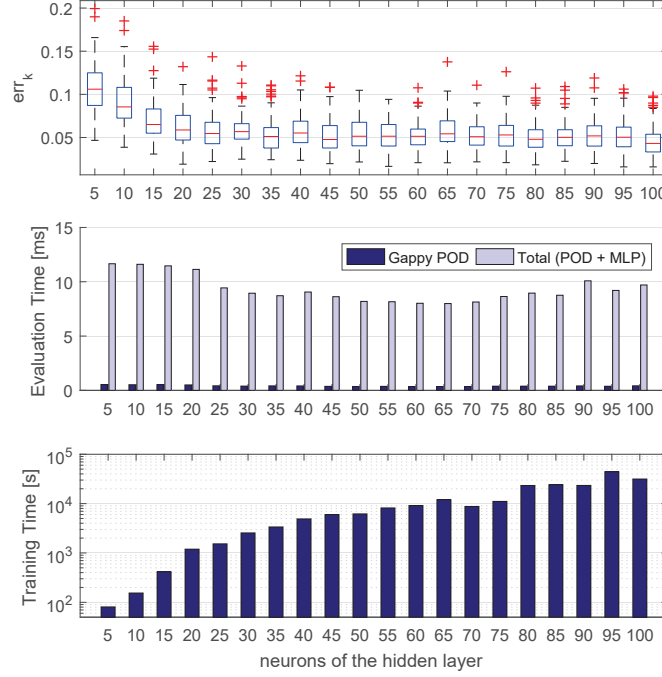


Figure 12: Error in fault identification (top), computational time in evaluation (middle) and in training (bottom) for variable number of neurons in the hidden layer.

Learning the model for $\mathbf{k}(\boldsymbol{\alpha})$. For the FDI step, a Multi-Layer Perceptron with one hidden layer is used to compute the estimated fault vector \mathbf{k} from the coefficients $\boldsymbol{\alpha}$ reconstructed via Gappy POD. The training set for the neural network model is composed by the POD coefficients computed in the previous step, and the matrix \mathbf{K} . The choice of a suitable number of neurons for the network emerges from a tradeoff between accuracy and computational time. A study was performed on the number of neurons, by varying the neurons of the hidden layer from 5 to 100, while the number of neurons of the output layer remains constant at $n_k = 8$. Figure 12 shows the computational time of the network in training and in evaluation, as well as the mean squared error in the fault identification plotted against the number of neurons in the hidden layer.

By increasing the number of neurons in the hidden layer, the accuracy initially increases and the mean squared error decreases down to 5% for 20 neurons. Adding more neurons does not produce significant benefits, neither on accuracy nor on computational time in evaluation; conversely, the increased complexity of the model reflects in longer computational times for training. Therefore, we consider the network with 20 neurons in the hidden layer as the most efficient candidate to perform the FDI for the addressed application. The computational time required by Gappy POD for the estimation of POD coefficients is at least one order of magnitude shorter than that required by the

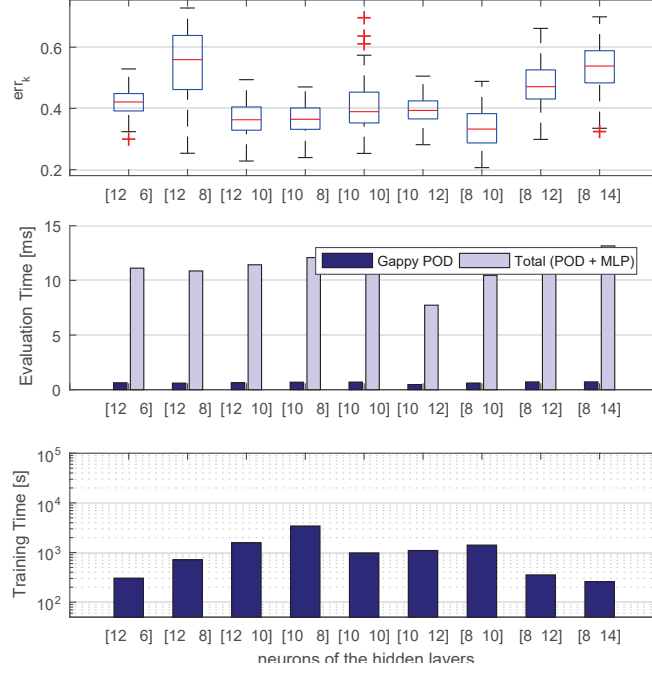


Figure 13: Performances of network architectures with two hidden layers

MLP in evaluation; so, the use of POD coefficients instead of the signal does not carry a significant penalty in computational time for real-time FDI. Figure 13 shows the performances of deeper learning architectures for the same FDI problem. MLPs with two hidden layers were considered, trained in the same conditions of the networks of Figure 11. It can be observed that the higher number of hidden layers makes the training more demanding and the networks are not able to converge on the considered problem.

The fault vector $k^{\text{estimated}}$ computed by the network is employed as initial condition for the RUL estimation.

Learning the model for the assessment function. For the surrogate modeling of the assessment function $\phi_a(\mathbf{k})$, we employ a polynomial kernel SVM, trained with the matrix \mathbf{K} as the input and Φ as the target. A parametric study to determine suitable settings for the SVM was discussed in [98]. The SVM is assessed over the 500 signals of the validation set. Figure 14 illustrates the ROC curve of the classifier; for the employed decision threshold, we observe a true positive rate of 96.75% and a false positive rate of 1.33%. It is possible to observe a success rate of 98.2% for the SVM emulating the assessment function $\phi_a(\mathbf{k})$, as pointed out in Table 2). The average computational time for the evaluation of the SVM classifier is in the order of 1ms, which allows to sensibly

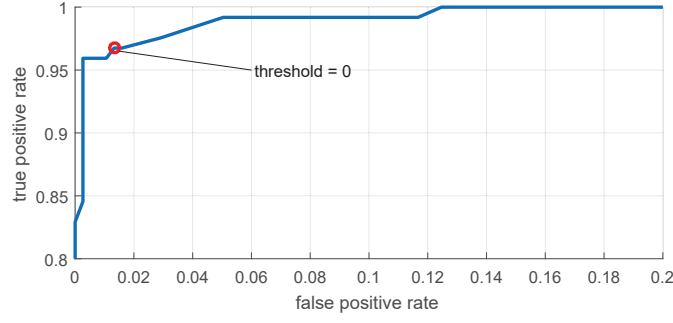


Figure 14: ROC curve of the SVM classifier

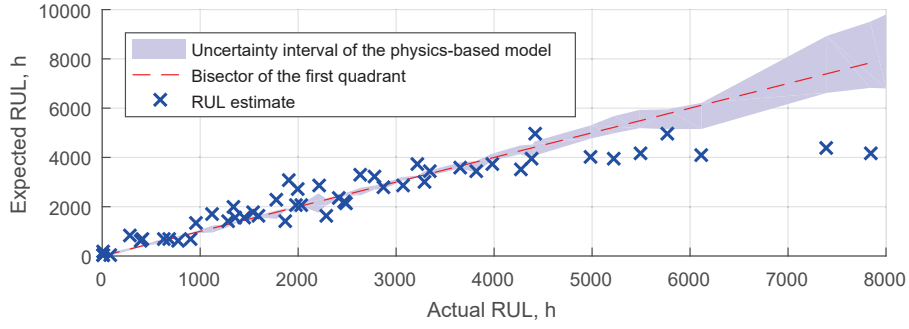


Figure 15: Predicted Remaining Useful Life with estimated fault condition and SVM surrogate assessment function

reduce the computational effort up to four orders of magnitude with respect to the evaluation time required by the complete assessment function.

740 5.2. Online: Results for the entire real-time PHM information flow

A final test assesses the whole real-time PHM flow, including the signal compression and FDI strategy in combination with the RUL estimation method. The SVM based

Table 2: Performance of the SVM; test set composed of 500 fault combinations, of which 123 corresponding to a healthy actuator and 377 corresponding to a system failure

correctly detected	119	96,75%
correctly undetected	372	98,67%
missed detections	4	3,25%
false positives	5	1,33%
total correct	491	98,20%
total wrong	9	1,80%
F1-Score	0.9636	

RUL estimation is computed with the initial condition $\mathbf{k}^{\text{estimated}}$ estimated through the FDI algorithm described in Section 3.3; the results, shown in Figure 15, are compared to the expected RUL (with the associated uncertainty interval) computed through the full damage propagation model. The global RUL estimate is affected by the errors introduced by both the RUL estimation itself and the FDI; then, the uncertainty is necessarily higher than that of the FDI algorithm alone (see Appendix 6), resulting in a slightly greater dispersion. RUL values larger than 4000 hours are commonly underestimated. In these cases, the initial fault parameters assume very small values, that are of the same order of magnitude of the uncertainty associated to the FDI process. The identified fault condition is commonly worse than the actual one, which results in estimating a faster fault propagation and a shorter RUL. Larger system faults can be detected with higher relative accuracy and the dispersion associated with the RUL estimate decreases. Therefore, our strategy achieves a higher accuracy when it is needed, that is when a failure is about to occur. When the Remaining Useful Life is long, there is no stringent need to know its value with high precision because there is long time ahead to plan the maintenance strategy at best, purchase spares and schedule replacements. Additionally, in the first part of the system operational life, an underestimation of the RUL is preferable to an overestimation, for obvious safety reasons, and does not trigger an unnecessary maintenance intervention, since the estimated time to failure is still long.

Figure 16 compares the uncertainty associated to the physics-based model for RUL with the relative error in RUL estimation err_{RUL} resulting from the RUL estimation alone (starting from $\mathbf{k}^{\text{actual}}$) and in combination with the FDI (starting from $\mathbf{k}^{\text{estimated}}$). We adopted a normalized absolute error as a performance metric; an asymmetric error definition would highlight the advantage of underestimating RUL, but would result in a

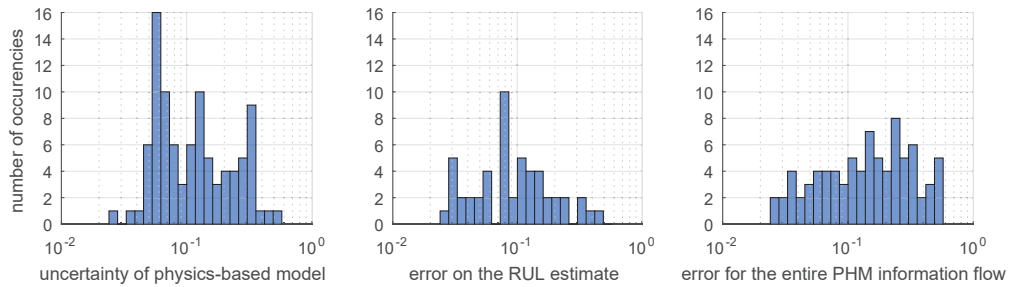


Figure 16: Comparison between the uncertainty of the physics-based model for RUL evaluation (left), the error of the real-time RUL estimation process (middle) and the error of the entire PHM online information flow (right).

less conservative evaluation:

$$err_{RUL} = \frac{|RUL_{estimated} - RUL_{actual}|}{RUL_{actual}} \quad (10)$$

and the uncertainty of the physics-based model is:

$$\Delta_{RUL} = \frac{|RUL_{95\%} - RUL_{5\%}|}{RUL_{actual}} \quad (11)$$

where $RUL_{95\%}$ and $RUL_{5\%}$ are the upper and lower bounds of the RUL with 90%
 770 confidence. The error of the RUL estimation is the same order of magnitude of the
 uncertainty interval of the physics based model. The error associated with the whole
 PHM process is still the same order of magnitude, but its distribution is more skewed to
 the right compared to the error of the RUL estimation alone. This reflects in a higher
 median error, confirming the underestimation of very long RULs that emerges from the
 775 comparison from Figures 20 and 15.

The absolute variance of the error is relatively high, being usually in the order of 20%
 of RUL, and in some cases almost comparable to the RUL itself. This behavior is partly
 due to the inherent uncertainty in the rate of damage propagation, which depends on
 a multitude of uncontrollable and unpredictable variables, such as environmental con-
 780 ditions and the particular time-history experienced during the system operational life.
 However, the error decreases as the damage grows, so the information about the system
 life is provided with higher accuracy when needed more (when a maintenance interven-
 tion needs to be planned). Additionally, this data driven strategy allows for a higher
 accuracy than the traditional *a priori* estimate [1, 2, 4, 3], and adds virtually no cost for
 785 the implementation, since no dedicated hardware is required. The computational time
 required online by the whole PHM process is in the order of milliseconds and allows the
 FDI and RUL estimation to be performed in real-time. In contrast, traditional model
 based approaches require computational times that range from minutes to tenths of
 minutes, which would be completely unsuited for on-board applications. These strate-
 790 gies [74, 99] imply solving numerically systems of ODEs with very small integration
 timesteps, and the identification is performed over the full dimensional dataset with
 empirical or semi-empirical optimization algorithms.

6. Concluding remarks

A comprehensive methodology for real-time fault detection and prognostics of dy-
 795 namical assemblies has been proposed. Our methodological framework leverages a com-
 bination of projection-based model reduction and machine learning strategies to achieve

reliable and timely estimates of the system useful life. The method has been developed and assessed for the overall Prognostics and Health Management (PHM) process applied to an electromechanical actuator for aircraft flight control systems. In addition, a simple model for estimating the fault propagation rate has been proposed, and a custom sampling technique has been employed to capture sufficient information from the system with a limited number of samples, which resulted particularly effective for our specific application.

The strategy permits to achieve an accuracy in FDI and RUL estimation that is comparable to computationally-intensive physics-based methods; at the same time, it requires few online data storage and processing resources, allowing for a fast and reliable on-board, real-time execution: the online computational time for fault detection is reduced by several orders of magnitude with respect to standard computationally expensive, physics-based methods. The availability of RUL estimate in real-time would permit to efficiently inform adaptive maintenance planning, allowing for significant cost reduction with respect to the standard periodical inspections and replacements, based on the analysis of the average failure rate of the components. The results show that our strategy for the real-time estimate of Remaining Useful Life allows to achieve high prediction accuracy when the monitored components are approaching the end of their operative life: this permits a dynamic and informed scheduling of maintenance interventions and an adaptive delivery of supplies and spares; additionally, the mission can be dynamically reconfigured to avoid overstressing faulty subsystems. When the system behaves nominally, our strategy tends to underestimate its useful life; however, this occurrence is safe and does not result in planning the unnecessary replacement of healthy components, since both the actual and estimated RUL are long.

Future developments include the assessment of alternative machine learning strategies for the FDI and RUL estimation, a more exhaustive study on uncertainty propagation, and the experimental validation of the models and algorithms.

Appendix A: Individual validation of the signal acquisition, diagnostics, and prognostics tasks

This section presents the application of the online process to the considered problem. Specifically, to assess the performance of the prognostic framework, first we test the signal compression, the FDI phase, and the RUL estimation phase separately; then, the RUL estimation is employed in combination with the FDI strategy to establish the overall accuracy of the method (Section 5.2).

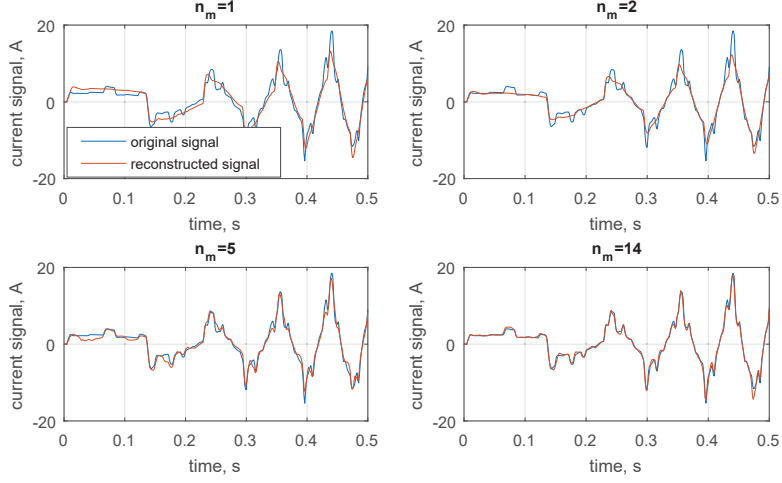


Figure 17: Signal reconstruction via Gappy POD with a variable number of retained modes n_m

Signal acquisition and compression. The signal is compressed online by storing and processing only the measures corresponding to the time coordinates determined offline through the procedure described in Section 3.2. Online, Gappy POD is employed to determine the coefficients $\alpha(\mathbf{k})$ from the stored signal. Figure 17 shows the reconstruction of a compressed signal from the test set via Gappy POD, with an increasing number of POD modes. To assess the accuracy of the signal estimate, we evaluate the normalized root mean squared error err_α of the POD coefficients α^{gappy} estimated with Gappy POD with respect to the coefficients α^{full} computed using the full signal:

$$err_\alpha = \frac{\sqrt{\frac{1}{n_m} \sum_{i=1}^{n_m} (\alpha_i^{\text{gappy}} - \alpha_i^{\text{full}})^2}}{\max(\alpha^{\text{full}}) - \min(\alpha^{\text{full}})} \quad (12)$$

Figure 18 shows the error computed for increasing size of the gappy matrix \mathbf{G} , that is, for increasing number of dominant components used for signal reconstruction through Gappy POD (according to Equation (3)).

For a number of modes larger than 10, it is already possible to identify the coefficients of the POD modes with an error in the order of 1%, comparable to the discrepancy yielding for the LF model with respect to the HF model (See Section 4.2). We choose to employ the first 22 POD modes for the following steps, corresponding to about 97% of the snapshot information; as shown in Figure 18, for our application this number of modes yields the minimum mean value of err_α . The use of a larger number of modes might increase the risk of including misleading information, affecting the accuracy of the signal reconstruction via Gappy POD. Additionally, with more than 30 modes (i.e. more modes than sampling points) the gappy matrix \mathbf{G} becomes ill conditioned; the signal

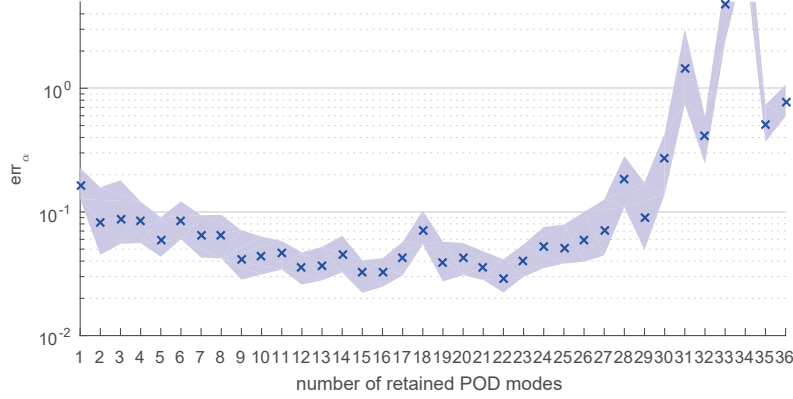


Figure 18: Offline preliminary assessment of Gappy POD coefficients identification, for varying number of modes retained. The blue crosses are the median error over the test set, while the amplitude of the tolerance band is set equal to the interquartile range. $n_m = 22$ modes result in the best accuracy, with a median error below 2%

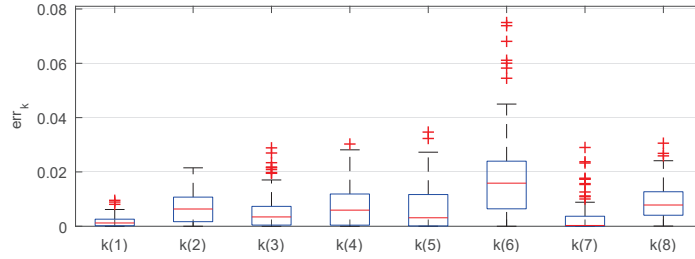


Figure 19: FDI error on each fault parameter

reconstruction is strongly affected by numerical noise with an increase of computational time and an error the same order of magnitude of the signal. A similar behavior has been already observed in literature [100].

Fault Detection and Identification. This section describes the assessment of the online performance of the two-layer perceptron for the FDI algorithm. Figure 19 shows the error on parameter identification highlighting the contribution of each component of the fault vector. The average error for each variable $|k_i^{\text{estimated}} - k_i^{\text{actual}}|$ is at most in the order of 1%, which is comparable to the discrepancy between the HF and LF model. Then, our data driven FDI technique performs comparably to a more traditional model-based technique, which commonly exploits an online optimization algorithm to match the HF and LF responses [74]. The FDI accuracy with respect to the individual fault parameters depends on the particular application and on the sensitivity of the monitored variable to the different fault modes. Figure 19 highlights how, for the considered application, the FDI performs better in the identification of the dry friction fault (k_1): the envelope current (i.e. the analyzed output signal y) is highly sensitive to this failure

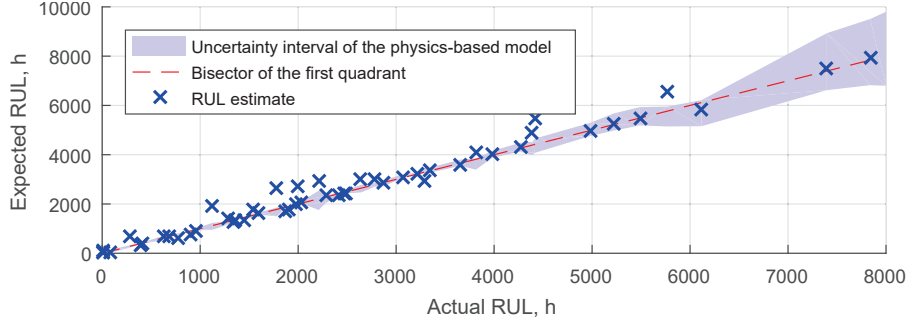


Figure 20: Expected Remaining Useful Life with physics-based assessment function and SVM, assuming the FDI error to be null (i.e. starting from the actual fault condition).

mode, resulting in better accuracy. On the other hand, the rotor eccentricity (k_6) is identified with a higher error and greater dispersion. This failure mode results in high frequency disturbances of the monitored signal, and the effect of information loss due to the truncation of the POD expansion is worse. The computational time is in the order
870 of a few milliseconds, allows computations to be run onboard, and enables real-time fault detection.

Estimate the Remaining Useful Life. To assess the accuracy of the online RUL estimation procedure alone, we test the performance of our online RUL step leaving out the error contribution associated with the FDI step. To do so, we test the SVM based
875 algorithm using the reference fault vector $\mathbf{k}^{\text{actual}}$ as initial condition, in place of the estimate computed in the FDI phase $\mathbf{k}^{\text{estimated}}$.

Figure 20 shows the upper and lower bound of the RUL (with 90% confidence) as a function of the expected RUL. On the horizontal axis is reported the RUL computed by the full model (i.e. the damage propagation model of Section 4.2.3 in combination
880 with the assessment function $\phi_a(\mathbf{k})$) at 50% probability, which we assume to be the actual value. The red dashed line is the bisector of the first quadrant, and represents an ideal RUL estimate (i.e. not affected by any error). The tolerance band represents the uncertainty interval of the full model: its lower and upper bounds correspond to the RUL computed by the full model respectively at 5% and 95% probability, respectively.
885 The blue crosses are the RUL values estimated with the SVM model from the actual initial condition; in most cases, the estimated values fall within the uncertainty interval associated to the reference physics-based model.

Acknowledgments

The work has been partly conducted under the Visiting Professor Program of Politecnico di Torino. Additional acknowledgments to the university's doctoral scholarship supporting Mr. Pier Carlo Berri. The authors thank Prof. Paolo Maggiore at Politecnico di Torino for his support.

References

- [1] Reliability data handbook. Universal model for reliability prediction of electronics components, PCBs and equipment, BSI British Standards, 2004.
- [2] J. C. Chesley, Handbook of Reliability Prediction Procedures for Mechanical Equipment, USA Navy, Naval Surface Warfare Center, NSWC Carderock Division, 2011.
- [3] R. Venkataraman, P. Seiler, M. Lukatsi, B. Vanek, Reliability assessment of actuator architectures for unmanned aircraft, Journal of Aircraft 54 (3) (2017) 955–966.
- [4] D. C. Garmendia, I. Chakraborty, D. N. Mavris, Method for evaluating electrically actuated hybrid wingbody control surface layouts, Journal of Aircraft 52 (6) (2015) 1780–1790.
- [5] X. Cui, S. Wang, Q. Li, J. Shi, Reliability assessment of aircraft power and actuation system based on three-axis control, in: 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Busan, South Korea, 7-11 July 2015, pp. 382–387.
- [6] K. Pipe, Practical prognostics for condition based maintenance, in: 2008 International Conference on Prognostics and Health Management, IEEE, Denver, CO, USA, 6-9 October 2008.
- [7] Z. Lv, J. Wang, G. Zhang, H. Jiayang, Prognostics health management of condition-based maintenance for aircraft engine systems, in: 2015 IEEE Conference on Prognostics and Health Management (PHM), IEEE, Austin, TX, USA, 22-25 June 2015.
- [8] N. Zerhouni, K. Medjaher, Health assessment, prognostics and remaining useful life - part a, in: From Prognostics and Health Systems Management to Predictive Maintenance 1, John Wiley & Sons, Inc., 2016, pp. 67–107.

- [9] M. D. Sudolsky, IVHM solutions using commercially-available aircraft condition monitoring systems, in: 2007 IEEE Aerospace Conference, IEEE, Big Sky, MT, USA, 3-10 March 2007.
- [10] J. Dunsdon, M. Harrington, The application of open system architecture for condition based maintenance to complete IVHM, in: 2008 IEEE Aerospace Conference, IEEE, Big Sky, MT, USA, 1-8 March 2008.
- [11] I. K. Jennions, Integrated Vehicle Health Management Perspectives on an Emerging Field, SAE International, 2011.
- [12] T. Sutharssan, S. Stoyanov, C. Bailey, C. Yin, Prognostic and health management for engineering systems: a review of the data-driven approach and algorithms, *The Journal of Engineering* 2015 (7) (2015) 215–222.
- [13] O. Benedettini, T. S. Baines, H. W. Lightfoot, R. M. Greenough, State-of-the-art in integrated vehicle health management, *Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 223 (2) (2009) 157–170.
- [14] Z. Williams, Benefits of IVHM: an analytical approach, in: 2006 IEEE Aerospace Conference, Big Sky, MT, USA, 4-11 March 2006, p. 9.
- [15] A. Ladj, F. B.-S. Tayeb, C. Varnier, An integrated prognostic based hybrid genetic-immune algorithm for scheduling jobs and predictive maintenance, in: 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, Vancouver, BC, Canada, 24-29 July 2016.
- [16] J. Yan, Maintenance planning and scheduling, in: *Machinery Prognostics and Prognosis Oriented Maintenance Management*, John Wiley & Sons Singapore Pte. Ltd, 2014, pp. 183–239.
- [17] M. Rigamonti, P. Baraldi, E. Zio, I. Roychoudhury, K. Goebel, S. Poll, Ensemble of optimized echo state networks for remaining useful life prediction, *Neurocomputing* 281 (2018) 121–138.
- [18] P. Polverino, C. Pianese, Model-based prognostic algorithm for online RUL estimation of PEMFCs, in: 3rd Conference on Control and Fault-Tolerant Systems (SysTol), IEEE, Barcelona, Spain, 7-9 September 2016.
- [19] R. Jihin, D. Söffker, N. Beganovic, Integrated prognostic model for RUL estimation using threshold optimization, in: *Structural Health Monitoring 2017: Real-*

- Time Material State Awareness and Data-Driven Safety Assurance, F-K Chang,
 950 F. Kopsaftopoulos eds., 2017.
- [20] X. Li, Q. Ding, J.-Q. Sun, Remaining useful life estimation in prognostics using
 deep convolution neural networks, Reliability Engineering & System Safety 172
 (2018) 1 – 11. doi:<https://doi.org/10.1016/j.ress.2017.11.021>.
 URL [http://www.sciencedirect.com/science/article/pii/
 955 S0951832017307779](http://www.sciencedirect.com/science/article/pii/S0951832017307779)
- [21] K. O. Kolcio, Model-based fault detection and isolation system for increased auton-
 omy, in: AIAA SPACE 2016, (AIAA 2016-5225), AIAA, Long Beach, California,
 13-16 September 2016.
- [22] M. Battipede, M. D. L. Dalla Vedova, P. Maggiore, S. Romeo, Model-based fault
 960 detection and isolation system for increased autonomy, in: AIAA Modeling and
 Simulation Technologies Conference, AIAA SciTech Forum, (AIAA 2015-2035),
 AIAA, Kissimmee, Florida, 5-9 January 2015.
- [23] C. Shi, S. Wang, X. Wang, J. Wang, Remaining useful life prediction based on
 particle filtering for high-speed pump in load sensing electro-hydrostatic actuator,
 965 in: CSAA/IET International Conference on Aircraft Utility Systems (AUS 2018),
 Guiyang, China, 19-22 June 2018, pp. 1–6.
- [24] W. Jia, L. Haimin, W. Xiao, Application and design of PHM in aircrafts integrated
 modular mission system, in: 2019 Prognostics and System Health Management
 Conference (PHM-Qingdao), 2019, pp. 1–6.
- [25] A. B. Rauzy, C. Haskins, Foundations for model-based systems engineering
 970 and model-based safety assessment, Systems Engineering 22 (2) (2019) 146–
 155. arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/sys.21469>,
 doi:10.1002/sys.21469.
 URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/sys.21469>
- [26] T. Sutharssan, S. Stoyanov, C. Bailey, C. Yin, Prognostic and health management
 975 for engineering systems: a review of the data-driven approach and algorithms, The
 Journal of Engineering 2015 (7) (2015) 215–222.
- [27] M. D. L. Dalla Vedova, D. De Fano, P. Maggiore, Neural network design for incip-
 980 ient failure detection on aircraft em actuator, International Journal of Mechanics
 and Control 17 (1) (2016) 77–83.

- [28] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, A. K. Nandi, Applications of machine learning to machine fault diagnosis: A review and roadmap, *Mechanical Systems and Signal Processing* 138 (2020) 106587. doi:<https://doi.org/10.1016/j.ymssp.2019.106587>.
- 985 [29] A. Listou Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, H. Zhang, Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture, *Reliability Engineering & System Safety* 183 (2019) 240 – 251. doi:<https://doi.org/10.1016/j.ress.2018.11.027>.
URL <http://www.sciencedirect.com/science/article/pii/S0951832018307506>
990
- [30] T. Tinga, R. Loendersloot, *Physical Model-Based Prognostics and Health Monitoring to Enable Predictive Maintenance*, Springer, 2019, pp. 313–353. doi:10.1007/978-3-030-05645-2_11.
- [31] T. Engelberth, D. Krawczyk, A. Verl, Model-based method for condition monitoring and diagnosis of compressors, *Procedia CIRP* 72 (2018) 1321 – 1326, 51st CIRP Conference on Manufacturing Systems. doi:<https://doi.org/10.1016/j.procir.2018.03.271>.
995
- [32] M. Cocconcelli, L. Capelli, J. Cavalaglio Camargo Molano, D. Borghi, Development of a methodology for condition-based maintenance in a large-scale application field, *Machines* 6 (2018) 17. doi:10.3390/machines6020017.
1000
- [33] D. Henry, J. Cieslak, J. Z. Torres, P. Colmenarejo, J. Branco, N. Santos, P. Serra, J. Telaar, H. Strauch, A. M. Giordano, M. De Stefano, C. Ott, M. Reiner, J. Jaworski, E. Papadopoulos, G. Visentin, F. Ankersen, J. G. Fernandez, Model-based fault diagnosis and tolerant control: the ESA’s e.deorbit mission, in: 2019 18th European Control Conference (ECC), 2019, pp. 4356–4361.
1005
- [34] B. Huang, Y. Di, C. Jin, J. Lee, Review of data-driven prognostics and health management techniques: Lessons learned from PHM data challenge competitions, 2017.
- [35] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, R. X. Gao, Deep learning and its applications to machine health monitoring, *Mechanical Systems and Signal Processing* 115 (2019) 213 – 237. doi:<https://doi.org/10.1016/j.ymssp.2018.05.050>.
1010

URL <http://www.sciencedirect.com/science/article/pii/S0888327018303108>

- 1015 [36] D. Rengasamy, H. P. Morvan, G. P. Figueredo, Deep learning approaches to aircraft maintenance, repair and overhaul: A review, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 150–156. doi:10.1109/ITSC.2018.8569502.
- 1020 [37] P. Qian, X. Ma, D. Zhang, J. Wang, Data-driven condition monitoring approaches to improving power output of wind turbines, IEEE Transactions on Industrial Electronics 66 (8) (2019) 6012–6020.
- [38] O. Bektas, J. A. Jones, S. Sankararaman, I. Roychoudhury, K. Goebel, A neural network framework for similarity-based prognostics, MethodsX 6 (2019) 383 – 390. doi:<https://doi.org/10.1016/j.mex.2019.02.015>.
- 1025 URL <http://www.sciencedirect.com/science/article/pii/S2215016119300263>
- [39] Y. Liu, Y. Chang, S. Liu, S. Chen, Data-driven prognostics of remaining useful life for milling machine cutting tools, in: 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), 2019, pp. 1–5.
- 1030 [40] P. C. Berri, M. D. L. Dalla Vedova, L. Mainini, Real-time fault detection and prognostics for aircraft actuation systems, in: 2019 AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, (AIAA Paper 2019-2210), AIAA, San Diego, California, 2019.
- [41] P. Freeman, R. Pandita, N. Srivastava, G. J. Balas, Model-based and data-driven fault detection performance for a small uav, IEEE/ASME Transactions on Mecha-
- 1035 tronics 18 (4) (2013) 1300–1309.
- [42] R. Venkataraman, P. J. Seiler, Model-based detection and isolation of rudder faults for a small uas, in: AIAA Guidance, Navigation, and Control Conference, AIAA SciTech Forum, (AIAA 2015-0857), AIAA, Kissimmee, Florida, 5-9 January 2015.
- 1040 [43] F. Meng, T. Meyer, P. Thomas, J. Wenske, Observer design and optimization for model-based condition monitoring of the wind turbine rotor blades using genetic algorithm, Journal of Physics: Conference Series 1037 (2018) 032027. doi:10.1088/1742-6596/1037/3/032027.
- URL <https://doi.org/10.1088/1742-6596/1037/3/032027>

- 1045 [44] W. Booyse, D. N. Wilke, S. Heyns, Deep digital twins for detection, diagnostics and prognostics, *Mechanical Systems and Signal Processing* 140 (2020) 106612. doi:<https://doi.org/10.1016/j.ymssp.2019.106612>.
- [45] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, B. Wu, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*, John Wiley & Sons, Inc., 2006.
- 1050 [46] R. Isermann, *Fault-Diagnosis Applications*, Springer Berlin Heidelberg, 2011.
- [47] L. Yongxiang, S. Jianming, W. Gong, L. Xiaodong, A data-driven prognostics approach for rul based on principle component and instance learning, in: 2016 IEEE International Conference on Prognostics and Health Management (ICPHM), Ottawa, ON, Canada, 20-22 June 2016, pp. 1–7.
- 1055 [48] S. Al-Dahidi, F. D. Maio, P. Baraldi, E. Zio, Remaining useful life estimation in heterogeneous fleets working under variable operating conditions, *Reliability Engineering & System Safety* 156 (2016) 109–124.
- [49] R. G. Nascimento, F. A. C. Viana, Fleet prognosis with physics-informed recurrent neural networks, in: *Structural Health Monitoring 2019*, DEStech Publications, Inc., 2019. doi:[10.12783/shm2019/32301](https://doi.org/10.12783/shm2019/32301).
1060 URL <https://doi.org/10.12783%2Fshm2019%2F32301>
- [50] A. Mornacchi, G. Vachtsevanos, G. Jacazio, Prognostics and health management of an electro-hydraulic servo actuator, in: *Annual Conference of the Prognostics and Health Management Society*, San Diego, CA, USA, 19-24 October 2015.
- 1065 [51] A. Nesci, A. D. Martin, G. Jacazio, M. Sorli, Detection and prognosis of propagating faults in flight control actuators for helicopters, *MDPI Aerospace* 7 (3) (2020) 20. doi:<https://doi.org/10.3390/aerospace7030020>.
- [52] X. Li, X. Yang, Y. Yang, I. Bennett, D. Mba, A novel diagnostic and prognostic framework for incipient fault detection and remaining service life prediction with application to industrial rotating machines, *Applied Soft Computing* 82 (2019) 105564. doi:<https://doi.org/10.1016/j.asoc.2019.105564>.
1070 URL <http://www.sciencedirect.com/science/article/pii/S1568494619303448>
- [53] F. Cadini, A. Gioietta, E. Zio, An Improvement of a Metamodel-Based Importance Sampling Algorithm for Estimating Small Failure Probabilities, 2014, pp. 2104–2114. doi:[10.1061/9780784413609.211](https://doi.org/10.1061/9780784413609.211).

- 1080 [54] E. Naderi, K. Khorasani, Data-driven fault detection, isolation and estimation of aircraft gas turbine engine actuator and sensors, in: 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, ON, Canada, 30 April-3 May 2017, pp. 1–6.
- [55] D. Rengasamy, M. Jafari, B. Rothwell, X. Chen, G. P. Figueredo, Deep learning with dynamically weighted loss function for sensor-based prognostics and health management, *Sensors* 20 (3). doi:10.3390/s20030723.
URL <https://www.mdpi.com/1424-8220/20/3/723>
- 1085 [56] L. Mainini, K. E. Willcox, Sensor placement strategy to inform decisions, in: 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, American Institute of Aeronautics and Astronautics, Denver, Colorado, 5-9 June 2017.
- [57] M. Hinze, S. Volkwein, Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control, in: Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2005, pp. 261–
1090 306.
- [58] J. L. Lumley, The structure of inhomogeneous turbulent flows, *Atmospheric Turbulence and Radio Wave Propagation* (1967) 166–178.
- [59] P. Holmes, J. L. Lumley, G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge Monographs on Mechanics, Cambridge University Press, 1996.
1095
- [60] K. Kunisch, S. Volkwein, Control of the burgers equation by a reduced-order approach using proper orthogonal decomposition, *Journal of Optimization Theory and Applications* 102 (2) (1999) 345–371.
- 1100 [61] E. J. Candès, J. K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, *Communications on Pure and Applied Mathematics* 59 (8) (2006) 1207–1223.
- [62] L. Sirovich, Turbulence and the dynamics of coherent structures. part 1: Coherent structures, *Quarterly of Applied Mathematics* 45 (3) (1987) 561–571.
- 1105 [63] V. Algazi, D. Sakrison, On the optimality of the karhunen-loève expansion (corresp.), *IEEE Trans. Information Theory* 15 (2) (2006) 319–321.

- [64] A. Dr, On the optimality of the discrete karhunen–loève expansion, SIAM Journal on Control and Optimization 36 (6) (1998) 1937–1939.
- 1110 [65] T. Kohonen, P. Somervuo, Self-organizing maps of symbol strings, Neurocomputing 21 (1-3) (1998) 19–30.
- [66] P. Somervuo, T. Kohonen, Self-organizing maps and learning vector quantization for feature sequences, Neural Processing Letters 10 (2) (1999) 151–159.
- [67] T. Kohonen (Ed.), Self-organizing Maps, 3rd Edition, Springer-Verlag, Berlin, Heidelberg, 2001.
- 1115 [68] L. Mainini, Structural assessment and sensor placement strategy for self-aware aerospace vehicles, in: Structural Health Monitoring 2017: Real-Time Material State Awareness and Data-Driven Safety Assurance, Vol. 1, F-K Chang, F. Kop-saftopoulos eds., 2017, pp. 1586–1594.
- 1120 [69] E. Oja, Simplified neuron model as a principal component analyzer, Journal of Mathematical Biology 15 (3) (1982) 267–273. doi:<https://doi.org/10.1007/BF00275687>.
- [70] R. Everson, L. Sirovich, Karhunen–loève procedure for gappy data, Journal of the Optical Society of America A 12 (8) (1995) 1657.
- 1125 [71] T. Bui-Thanh, M. Damodaran, K. E. Willcox, Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition, AIAA Journal 42 (8) (2004) 1505–1516.
- [72] K. Willcox, Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition, Computers & Fluids 35 (2) (2006) 208–226.
- 1130 [73] L. Mainini, K. Willcox, Surrogate modeling approach to support real-time structural assessment and decision making, AIAA Journal 53 (6) (2015) 1612–1626.
- [74] P. C. Berri, M. D. L. Dalla Vedova, Optimization techniques for prognostics of on-board electromechanical servomechanisms affected by progressive faults, International Review of Aerospace Engineering 12 (4) (2019) 160. doi:<https://doi.org/10.15866/irease.v12i4.17356>.
- 1135 [75] P. C. Berri, M. D. L. Dalla Vedova, L. Mainini, Diagnosis of actuation systems faults from dynamic data, in: 6th European Conference on Computational Mechanics (ECCM), Glasgow, UK, 11-15 June 2018.

- [76] D. W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, *Journal of the Society for Industrial and Applied Mathematics* 11 (2) (1963) 431–441.
- [77] M. Hagan, M. Menhaj, Training feedforward networks with the marquardt algorithm, *IEEE Transactions on Neural Networks* 5 (6) (1994) 989–993.
- [78] R. Swischuk, L. Mainini, B. Peherstorfer, K. Willcox, Projection-based model reduction:formulations for physics-based machine learning, *Computers & Fluids* doi:10.1016/j.compfluid.2018.07.021.
- [79] C. Cortes, V. Vapnik, Support-vector networks, *Machine Learning* 20 (3) (1995) 273–297.
- [80] T. V. Gestel, J. A. K. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. D. Moor, J. Vandewalle, Benchmarking least squares support vector machine classifiers, *Machine Learning* 54 (2004) 5–32.
- [81] A. Basudhar, S. Missoum, Adaptive explicit decision functions for probabilistic design and optimization using support vector machines, *Computers & Structures* 86 (19) (2008) 1904 – 1917. doi:<https://doi.org/10.1016/j.compstruc.2008.02.008>.
URL <http://www.sciencedirect.com/science/article/pii/S0045794908000515>
- [82] Q. Leng, S. Wang, D. Shen, Construction of binary tree classifier using linear SVM for large-scale classification, in: 2018 International Conference on Robots & Intelligent System (ICRIS), IEEE, Changsha, China, 26-27 May 2018.
- [83] Electromechanical sensors and actuators, *Sensor Review* 20 (1).
- [84] O. Bennouna, N. Langlois, Modeling and simulation of electromechanical actuators for aircraft nacelles, in: 9th International Symposium on Mechatronics and its Applications (ISMA), IEEE, Amman, Jordan, 9-11 April 2013.
- [85] C. T. Leondes, *Electromechanical Systems*, CRC Press, 2014.
- [86] J.-C. Maré, Electro-mechanical actuators, in: *Aerospace Actuators 2*, John Wiley & Sons, Inc., 2017, pp. 171–217.

- [87] R. Quigley, More electric aircraft, in: Proceedings Eighth Annual IEEE Applied Power Electronics Conference and Exposition, IEEE, San Diego, CA, 7-11 March 1993.
- 1170 [88] M. Howse, All-electric aircraft, *Power Engineer* 17 (4) (2003) 35.
- [89] A. G. Garriga, S. S. Ponnusamy, L. Mainini, A multi-fidelity framework to support the design of more-electric actuation, in: 2018 Multidisciplinary Analysis and Optimization Conference, AIAA Aviation Forum, (AIAA Paper 2018-3741), American Institute of Aeronautics and Astronautics, 25-29 June 2018.
- 1175 [90] M. J. Cronin, All-electric vs conventional aircraft - the production/operational aspects, *Journal of Aircraft* 20 (6) (1983) 481–486.
- [91] F. Annaz, Architecture philosophy in high integrity electromechanical actuators, in: 5th International Symposium on Mechatronics and Its Applications, IEEE, Amman, Jordan, 17-29 May 2008.
- 1180 [92] T. Missala, Electromechanical actuators – selected safety-related problems, in: Recent Advances in Automation, Robotics and Measuring Techniques, Springer International Publishing, 2014, pp. 175–186.
- [93] Y. Hussain, S. Burrow, P. Keogh, L. Henson, A review of techniques to mitigate jamming in electromechanical actuators for safety critical applications, *International Journal of Prognostics and Health Management* 9.
- 1185 [94] E. Balaban, P. Bansal, P. Stoelting, A. Saxena, K. F. Goebel, S. Curran, A diagnostic approach for electro-mechanical actuators in aerospace systems, in: 2009 IEEE Aerospace conference, IEEE, Big Sky, MT, USA, 7-14 March 2009.
- [95] E. Balaban, A. Saxena, K. Goebel, C. S. Byington, S. P. Bharadwaj, M. Smith, Experimental data collection and modeling for nominal and fault conditions on electro-mechanical actuators, in: Annual Conference of the Prognostics and Health Management Society, San Diego, CA, 27 September-1 October 2009.
- 1190 [96] E. Balaban, A. Saxena, S. Narasimhan, I. Roychoudhury, K. Goebel, M. Koopmans, Airborne electro-mechanical actuator test stand for development of prognostic health management systems, in: Annual Conference of the Prognostics and Health Management Society, Portland, OR, 10-16 October 2010.
- 1195

- [97] P. C. Berri, M. D. L. Dalla Vedova, P. Maggiore, A smart electromechanical actuator monitor for new model-based prognostic algorithms, International Journal of Mechanics and Control 17 (2) (2016) 19–25.
- 1200 [98] P. C. Berri, M. D. L. D. Vedova, G. Quattrocchi, P. Maggiore, Model-based strategy and surrogate function for health condition assessment of actuation devices, IOP Conference Series: Materials Science and Engineering 1024 (2021) 012101.
- 1205 [99] M. D. L. Dalla Vedova, P. C. Berri, S. Re, A comparison of bio-inspired meta-heuristic algorithms for aircraft actuator prognostics, in: Proc. of the 29th European Safety and Reliability Conference (ESREL 2019), Hannover, Germany, 22 - 26 September 2019.
- [100] L. Mainini, K. E. Willcox, Data to decisions: Real-time structural assessment from sparse measurements affected by uncertainty, Computers & Structures 182 (2017) 296–312.