Reliability Engineering and System Safety 252 (2024) 110451

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

Remaining Useful Life prediction based on physics-informed data augmentation

Martin Hervé de Beaulieu [a], Mayank Shekhar Jha [a,*], Hugues Garnier [a], Farid Cerbah [b]

[a] CRAN, CNRS UMR 7039, Université de Lorraine, 2 rue Jean Lamour, Vandoeuvre-les-Nancy, 54506, France
[b] Dassault Aviation, Scientific Studies Department, 1 Rue du Val d'Or, Saint-Cloud, 92552, France

ARTICLE INFO

Keywords:
Prognostics
System degradation
Deep learning
Health index
Predictive maintenance
Remaining useful life
System identification

ABSTRACT

Current approaches for monitoring machine health (SOH) and effective prognostics depend on the extensive use of complete degradation data trajectories, implying the reliance on data generation techniques that involve functional degradation of the real system until the failure state is reached. These commonly adopted approaches that depend on labeled target data remain operationally and economically nonviable for most industries and safety critical systems. This paper presents novel approaches that alleviate the existing dependence of most prognostics procedures on Remaining Useful Life (RUL) labeled data for training. To this end, firstly, a hybrid data augmentation procedure is proposed that enables the integration of system knowledge available *a priori* as well as physics of failure, within the training data. Secondly, an unsupervised Health Index (HI) extraction approach is developed, followed by a long-term prediction of this same HI, that leads to an efficient prediction of RUL without labeled data. Finally, a reliability-based assessment is performed to validate the proposed approach. This comprehensive approach (i.e. integrating all the various stages involved in achieving a RUL prediction based on unlabeled data) is tested on a real industrial aircraft system demonstrating the effectiveness of the proposed approach in real industrial context.

1. Introduction

The purpose of maintenance in an industrial context is to ensure maximum availability of the equipment at the highest possible level of operation with reduced maintenance costs [1]. The integration of numerous and diversified sensors [2], as well as the exponential increase in data storage means in recent years, raises the possibility of transitioning from traditional maintenance using expertise-based diagnosis and fail-and-fix practices to data-based prediction and maintenance methodology optimized to intervene at the best time on the monitored system [3,4].

Prognostics is the process of assessing State of Health (SOH) of a system and predicting its EOL, which includes the detection of an incipient failure and the prediction of RUL [1]. Prognostics can target either component or sub-component levels, involving the prediction of a specific failure mode's progression from its onset to component failure [5,6]. The health index (HI) plays a key role in prognostics and health management (PHM) by reflecting the true State of Health (SOH) of the system and accurately describing the degradation process using the inherent information of the condition monitoring signals [3,7]. HI serves as an intermediate variable to predict the remaining useful life (RUL) [8]. RUL represents the time remaining between the current instant t and the End of Life (EOL) of the system [9]. Mathematically, RUL can be expressed as:

$$RUL(t_k) = t_{EOL} - t_k \tag{1}$$

where t_{EOL} is the EOL instant and t_k is the current time. Accurate estimation of RUL is crucial for proactive maintenance planning [10], accompanied by confidence intervals to account for the stochastic nature of degradation [11]. Moreover, uncertainty quantification is essential for risk-dependent maintenance decisions [12,13]. Prognostics approaches are usually divided into two main classes of approaches : physics-based approaches and data-driven approaches [3]. Physics model-based approaches attempt to describe degradation processes by constructing models based on existing physics laws and on the available knowledge or observation of the degradation mechanism [14], calling for availability of expert knowledge in the different scientific fields related to the degradation of the observed system (mechanics, thermodynamics, fluid physics, etc.) as well as an in-depth expertise about the system itself. On the other hand, AI-based data-driven approaches rely on highly complex computational models with coefficients that are adjusted based on large number of observations [9,15–19]. The advancement in the domain of AI opens extensive possibilities especially

* Corresponding author.
E-mail address: mayank-shekhar.jha@univ-lorraine.fr (M.S. Jha).

https://doi.org/10.1016/j.ress.2024.110451
Received 3 November 2023; Received in revised form 18 July 2024; Accepted 18 August 2024
Available online 21 August 2024

Acronyms

AI	Artificial Intelligence
CDF	Cumulative Distribution Function
CNN	Convolutional Neural Network
DCCN	Dilated Causal Convolutional Network
EOL	End of Life
FCL	Fully Connected Layer
GAN	Generative Adversarial Network
HI	Health Index
IV	Instrumental Variable
LSTM	Long Short-Term Memory
MSE	Mean Squared Error
NN	Neural Network
PEM	Prediction Error Method
PHM	Pronostic and Health Management
PI	Proportional Integral
PINNs	Physics-Informed Neural Networks
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
SOH	State of Health
SRIVC	Simplified Refined Instrumental Variable method for Continuous-time systems
TTF	Time to Failure

for complex industrial systems for which physics or traditional statistical models turn out to be limited [20]. However, AI-based approaches suffer from two major drawbacks:

1. *Requirement for RUL-labeled data in training phase:* Firstly, AI-based approaches are highly data-intensive. Complex neural models require huge amounts of data to be trained. Furthermore, most of the AI-based approaches leverage RUL-labeled data to perform direct mapping between sensor values and RUL in a supervised manner. Such approaches include Convolutional Neural Network (CNN) models [21,22], LSTM [23–26], attention mechanism [27,28], and Dilated Causal Convolutional Network (DCCN) [29–31], etc. However, in most of the real-life cases, such labeled data are not available [32]. Indeed, obtaining RUL-labeled data requires conducting experiments (accelerated degradation tests) until the EOL of the system is reached, which is time consuming, costly and often unrealistic in face of safety critical industrial systems. Therefore, as most of the AI-based existing approaches for prognostics rely heavily on the availability of labeled ground truth RUL data, the usefulness of such approaches remains limited in the absence of such a ground truth RUL in real cases.
2. *Poor integration of a priori knowledge and physics:* The second major limitation of AI-based methods lies in their inability to incorporate *a priori* knowledge and physics. This *a priori* knowledge could include information about the system architecture, partial physical understanding (e.g., degradation models or physical laws governing system components), observations of system behaviors, or feedback from maintenance actions. Additionally, these methods lack the ability to provide a physical interpretation of their predictions. Moreover, because they operate outside the traditional deterministic framework, similar to standard statistical approaches, assessing the reliability of these predictions becomes challenging. Consequently, such methods are often unsuitable for real industrial applications, where safety and security considerations necessitate the integration of available knowledge into the models.

To deal with the two challenges mentioned above, several deep learning approaches have emerged recently. Firstly, Physics-Informed Neural Networks (PINNs), which belong to a hybrid class of Neural Networks combining machine learning and physical laws can handle the issues of lack of integration of prior knowledge. In recent years, PINNs have gained significant attention due to their ability to learn from limited data while respecting the fundamental principles of physics [33]. Prior knowledge is incorporated within the data-driven model, enabling improvement of model performance in the absence of data, rendering model-interpretability. When the knowledge relies more on some expertise about the system than on physics law, such an approach is often referred to as "Knowledge-informed" or "Knowledge-guided" [34]. The integration of knowledge can be divided into three primary families [35,36]: within the training data, hypothesis set based, and within the learning process.

- Within the training data: Here, the approach focuses on integration of explicit knowledge into the training data through simulations [37]. These simulations, based on prior knowledge, are used to train a model that assimilates the knowledge contained within them. Such an approach can be referred to as physics-based data augmentation [38].
- Hypothesis test based: Here, the main focus remains on constraining the search space of the machine-learning model by delimiting it with a predefined hypothesis set based on conservative laws. This ensures that the results of the AI-based model will adhere to specific physical laws selected beforehand. These conservation laws can be embedded as linear equations in the Neural Network (NN) through weighted connections between the input layer and the first hidden layer, similar to a "pre-neural-network" concept [39,40].
- Within the learning process: In these approaches the focus remains on integration of knowledge within the learning process itself. The most common method is to add one or more external physics-based (or knowledge-based) losses to the traditional loss function of the learning algorithm [41]. The additional losses act as regularizing constraints during the training, guiding the network to a model that respects the desired knowledge/physics. Such a loss has notably been implemented in Pronostic and Health Management (PHM) applications using Weibull-law [42].

Such PINNs applied to PHM problems at system level can successfully handle RUL prediction and HI monitoring, however, integration of reliability metrics within such approaches is not a trivial task.

Another way to cope with large data requirements is Generative Adversarial Network (GAN). A GAN is a generative model that consists of two neural network models: a "generator" and a "discriminator" [43]. Such models are efficient in creating new realistic data samples on demand [44]. It can therefore be used to augment the amount of data available in order to help training a RUL prediction algorithm for instance. Special architectures of GAN, using notably Recurrent Neural Network (RNN) such as LSTM, have proven to be effective in generating realistic time series data [45]. Yet, such approaches are black boxes; i.e., the models that govern the generated dynamics are unknown and usually do not provide any guarantees regarding the physical consistency of the data generated.

Traditional reliability approaches are pivotal in validating AI-based methods by benchmarking them against established laws, bolstering their trustworthiness. Despite their black-box nature, AI-based methods can demonstrate conformity to well-established reliability laws, instilling confidence in their predictions. Reliability analysis involves quantifying failure rates and probabilities to gain a comprehensive understanding of the entire system fleet. Typically, estimating the probability of failure $F(t)$ or reliability $R(t) = 1-F(t)$ involves observing the end of life (EOL) of various systems within the population, often modeled using the Weibull distribution [46]. Widely applied in reliability engineering, the Weibull distribution accurately characterizes the

probability density of failure in diverse fields, facilitating maintenance planning, optimizing spare parts inventory, and resource allocation [47, 48]. The failure rate, which represents the instantaneous rate of failures at a given time t, serves as a critical metric indicating the probability of failure as time progresses. It can exhibit different patterns over time, such as constant, increasing (wear-out failures), and decreasing (early-life failures) rates, typically depicted in a bathtub curve [49]. However, traditional reliability approaches primarily provide a fleet-level failure rate λ and a failure probability $F(t)$, lacking pre-EOL health behavior estimates or detailed deterioration information for individual systems within the fleet (as accomplished with prognostics based approaches). Currently, there remains a wide gap between classical reliability-based approaches that assess historical databases (fleet level information) and prognostic-based maintenance approaches that focus on individual systems/degradation trajectories.

To bridge these aforementioned scientific gaps and alleviate various limitations, this work accomplishes the objective of generating RUL predictions by overcoming the two aforementioned challenges (lack of knowledge integration and requirements for large labeled data sets). To that end, three main contributions are proposed:

1. Non-implication of RUL-labeled data in training phase: A completely unsupervised strategy is developed for extracting the HI of the system, which avoids dependence on labeled data completely. In a second phase, long-term prediction of the HI makes it possible to obtain estimates of the RUL, again without using labeled data.
2. *Integration of a priori knowledge and physics*: A physics-informed data augmentation process (for both, healthy and degraded data) approach based on system identification techniques is proposed wherein injecting physical degradation models leads to generation of rich training data adhering to basic physics, leading further to greater transparency and interpretability of the results obtained.
3. *Reliability-based assessment:* A fleet-level reliability-based assessment is performed using the well-established Weibull distribution thereby validating the compliance of proposed AI based approach with the well-established laws of reliability.

As per the authors, there are currently no works that include, develop or employ system-identification based approaches for prognostics (via. data augmentation or otherwise). Moreover, for a closed loop system, this is a considerable challenge. This work develops and deploys well established system identification approach for efficient prognostics.

2. Proposed framework

The general overview of the proposed comprehensive framework is summarized in Fig. 1. It is comprehensive in that the work addresses all the essential steps involved in obtaining a RUL prediction from the raw data collected in the system, in the absence of labeled data. The proposed framework can be divided into three main steps which are described in the following subsections.

2.1. Step 1 - Hybrid data augmentation

The first step consists of augmenting and enriching the data using data-driven system identification [50–52] (Section 2.1). In this work, two different data augmentation strategies are carried out. The first one consists of augmenting the amount of nominal (i.e. healthy) data, and enriching it (by offering new possibilities of set-point values, operating conditions, etc.), in order to train a neural network structure for nominal signal reconstruction (Section 2.2). The second one consists of designing a new distribution of data, by injecting realistic physics-based degradation in the data-driven generated data previously obtained in Section 2.1.1. Such degraded data will then be used to train a LSTM neural structure that produces the future predictions of HI (Section 2.3).

2.1.1. Nominal data augmentation using data-driven system identification

The first stage of the proposed approach consists of increasing the amount of nominal data in order to address the problem of the lack of training data. To that end, system identification is used. System identification consists of building mathematical models that describe the behavior of a dynamic system based on observed data (including observed data of limited size, which is particularly relevant in the present case). There is a vast literature on system identification methods [51–54]. Although most industrial systems are non-linear, it is generally possible to assume the system to operate linearly around an operating point, in order to use linear model identification methods. Traditionally, an experiment adapted to the situation is designed in order to acquire, from the considered system, input/output signals that are sufficiently rich to be informative. The general data-driven system identification workflow in open loop can therefore be summarized as follows [52]:

1. Collect a sufficiently rich set of data from the system, removing trends and outliers.
2. Choose a class of models on which the identification will be carried out, in order to determine the model structure and the number of parameters to be estimated. For most of the industrial processes, linear low order model plus dead time is a sufficient approximation.
3. Estimate the parameters of the selected model on the basis of a chosen identification criterion. The standard approaches for parameter estimation in the literature are the Prediction Error Method (PEM) and the Instrumental Variable (IV) approach [50, 54,55].
4. Validate the model by assessing its performance on a validation dataset and, if necessary, repeating one or more of the previous stages to refine the result.

However, in most of the industrial scenarios, it is not possible to establish such experimental designs. Instead, the historical data from the operating processes must be leveraged. Moreover, most industrial systems run under closed-loop conditions (i.e. under feedback control) [56], calling for prognostics based investigation under closed loop conditions. Numerous closed-loop identification strategies have been developed [52,57–59]. In these approaches, it is generally assumed that the setpoint and/or the extra signal on the command will excite the system to a sufficient level [57]. However, if this is not the case, well-known identifiability issues may arise [60–62].

The standard block diagram of a process under feedback control is displayed in Fig. 2 where:

- G is the transfer function of the system to be identified
- C is the transfer function of the controller
- r is the reference (or setpoint)
- u is the command
- r_u is a possible extra-signal on the command u
- n is the output noise or sensor noise
- y is the output
- ε is the error ($\varepsilon = r - y$)

From the block diagram, the output can be expressed as:

$$y = Gu + n = G(C(r - y) + r_u) + n \tag{2}$$

hence

$$(1 + CG)y = G(Cr + r_u) + n \tag{3}$$

In a similar way, the command can be expressed as:

$$u = C(r - Gu - n) + r_u \tag{4}$$

hence

$$(1 + CG)u = C(r - n) + r_u \tag{5}$$

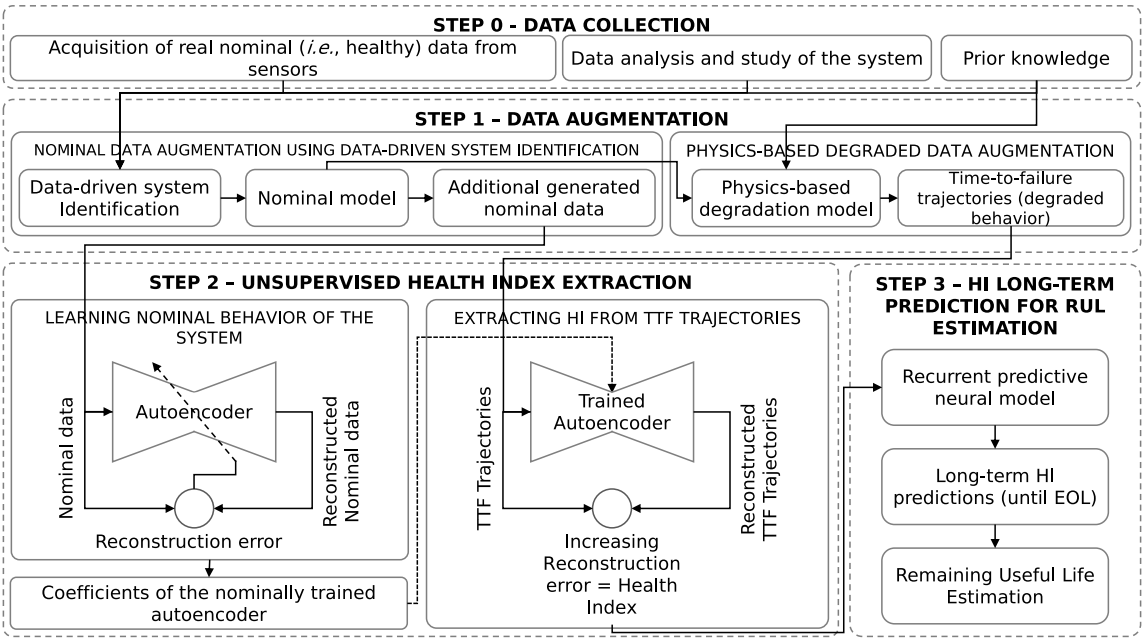


Fig. 1. The proposed framework for RUL prediction without using measured degradation data.

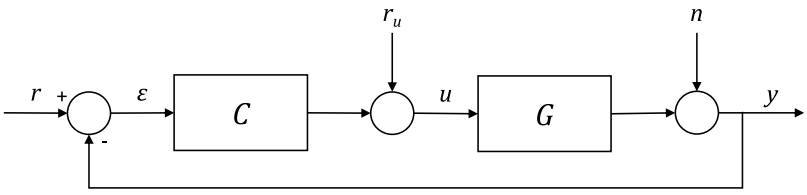


Fig. 2. Standard block diagram of a closed-loop system.

The ratio of the output signal over the command signal leads to:

$$\frac{y}{u} = \frac{CGr + Gr_u + n}{Cr + r_u - Cn} \tag{6}$$

External signals must be rich enough to excite the system and provide sufficient information [63]. Let us investigate the case where $r_u = 0$, i.e. when no extra signal is added to the command. Then:

$$u = C(r - y) \tag{7}$$

hence

$$y = \frac{Cr - u}{C} \tag{8}$$

Then, if the setpoint r remains constant (or null), the equation becomes:

$$y = -\frac{1}{C} \times u \tag{9}$$

As a result, without external excitation, any identification method will naturally estimate the inverse of the controller transfer function [62,64]. In the specific case of this study, it turns out that:

- The system operates in closed loop.
- Experiment design is not possible. Only a limited amount of data from historical flights is available.
- Only a few and not sufficiently informative changes in the setpoint occur in the historical data (i.e., $r_u = r = 0$ for the most part).
- The controller is not known and must be identified.

The data-driven system identification problem that is encountered here can then be formulated as follows:

From the limited historical data available, determine linear models of the controller C and the process G in order to approximate the dynamic behavior of the feedback control loop, with the goal of generating additional data and later incorporating additional degradation into the closed-loop components.

As a consequence, a custom approach is developed to handle the case study presented here. It is necessary to locate, in the available data, two very specific situations in order to be able to identify C and G models [65]:

- A "manual mode", where the system could be considered as operating in open loop, and the command signal u is varied enough to identify the model of the process G.
- An "automatic mode", where the system is controlled in closed loop, the setpoint r remaining constant. This is the case described by Eq. (9), where the controller model C can be deduced through the identification of its inverse (see Fig. 3).

The proposed methodology therefore aims at handling these two distinct cases in the following manner:

1. Locate historical data segments where the control is in "manual mode" in order to identify the process model G from "open-loop data".
2. Find historical data segments in which the control is in "automatic mode" in order to identify the inverse of the controller model of C from "closed-loop data".
3. Manually reproduce the effects of measurement noise.
4. Validate the closed-loop system model by verifying its ability to replicate the general dynamical behavior across several flights.

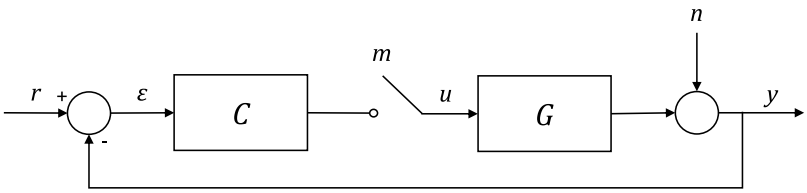


Fig. 3. Block diagram of the feedback control loop with an option to switch to "manual mode" (corresponding to an open-loop control).

In the end, once the parameters of the various linear models governing the dynamic behavior of the different subsystems have been determined, it becomes possible to perform nominal data augmentation. To this end, new data samples corresponding to nominal operation are obtained by simply specifying the evolution of the input set-point profile u, supplied to the closed-loop system model. This closed loop model consists of the combinations of estimated controller model $\hat{C}$, estimated process model $\hat{G}$, and the noise generator. In this way, data samples can be generated as required, simply by varying the duration and evolution profile of the set-point r.

2.1.2. Physics-based degraded data augmentation

Having ensured the nominal data augmentation, the other aspect involves enriching the data by integrating a degradation mechanism. To that end, an additional physics-based model of a degradation mechanism is injected into a component sub-model of the global system previously identified using data-driven methods.

Functional degradation mechanisms across various domains exhibit certain characteristics such as irreversibility and monotonicity. For instance, the Arrhenius model is traditionally used for various non- mechanical failure mechanisms, mostly depending on chemical reactions, diffusion processes or migration processes [66]. Paris' law and Coffin-Manson model are typically applied to mechanical failure, material fatigue or material deformation, especially crack growth [67]. The Eyring Model is used to describe changes in the rate of a chemical reaction as a function of temperature or stress [68]. The exponential behavior of the failure evolution is common to all these degradation models. Add to this the fact that, in practice, similar exponential degradation trends can be observed in most areas. Therefore, these conclusions motivate the choice of a generic exponential degradation model, as had already been done in the design of the C-MAPSS simulator [69].

By integrating the physics-based degradation model inside of the data-driven identified model of one selected component, a physics-informed data augmentation process is therefore obtained, allowing the generation of new unseen data samples incorporating a degradation based on a physics model.

2.2. Step 2 - Unsupervised health index extraction

With the initial nominal operating data set enriched by the data augmentation process, a neural network-based model is trained to extract HI of the system studied from the multisensor data.

2.2.1. Autoencoder structure

The autoencoder structure is trained to reconstruct nominal operating time-series, i.e., when the system is considered to be healthy. The autoencoder therefore learns a basic representation of the nominal healthy data distribution and its reconstruction with minimum error. Further, this nominal reconstruction model is used to reconstruct complete TTF time-series, including degraded operation of the system. The reconstruction error, i.e. the difference between real signals and reconstructed signals is then computed to obtain a HI estimation.

Autoencoders have proven to be very efficient in extracting features from raw sensors data [70,71]. In particular, the features obtained from autoencoders have monotonicity and clear trendability characteristic that are essential for HI estimation.

Autoencoder architecture is developed for encoding an input x into a compressed representation called "latent space", denoted as z, and then decoding it to try to reconstruct the original input [72]. The encoding function is denoted as $z = f_{\theta_e}(x)$ and the decoding function $\hat{x} = g_{\theta_d}(z)$ so that the overall learning is accomplished by the nested function (10).

$$\hat{x} = g_{\theta_d}(f_{\theta_e}(x)) \tag{10}$$

The training is realized by minimizing the reconstruction error, which is a function of x and $\hat{x}$, as defined in Eq. (11)

$$J_{AE}(\theta_e, \theta_d) = \sum L(x, \hat{x}) = \sum L(x, g_{\theta_d}(f_{\theta_e}(x))) \tag{11}$$

where L is a loss function such as the Mean Squared Error (MSE).

Encoding and decoding functions can be achieved by using multiple FCL, with layers that decrease progressively in size to force the computations flow through a bottleneck latent space. As the latent space has a limited size, the network prioritizes learning the most meaningful features that allow an accurate reconstruction of the input. Autoencoders are an unsupervised learning method, since they do not need any labeled data to train on. This is of key interest in the field of prognostic analysis where labeled data are very rarely available [73].

2.2.2. HI extraction using reconstruction error

A novel approach is developed in order to use the autoencoder reconstruction error as a HI. During the training phase, the autoencoder is trained with respect to nominal (i.e. healthy) data, thus learning the nominal operating behavior of the system. Both real data samples (if available) and simulated ones (obtained in the nominal data augmentation phase described in Section 2.1.1) can be used. Thus, augmented data are used as input data to train neural network models. The data augmentation part helps to construct a training dataset large enough to train complex deep learning models. Let us consider a nominal training domain $D_N = \{\mathbf{X}_N^i\}_{i=1}^{N_N}$ where N_N is the number of nominal training samples. Each sample $\mathbf{X}_N^i$ belongs to a *nominal feature space* $\mathcal{X}_N$. $\mathbf{X}_N^i$ denotes a multivariate sequential data sample of length T_i and with s features (for instance, $\mathbf{X}_N^i$ can be a flight of total duration T_i and with s collected signals, see Section 3).

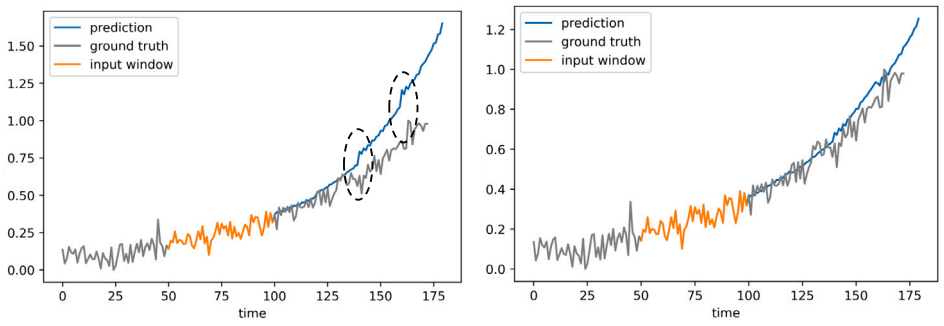
The nominal training data samples $\mathbf{x}_N^i$ are split into several windows:

$$\mathbf{X}_N^i(t_w) = \left\{ X_{t_k}^i \right\}_{k=w}^{w+\Delta} \tag{12}$$

where t_w is the start timestep of the window and Δ is its total duration. Parameters θ_e and θ_d of the encoder and decoder are learned on the basis of the nominal training data, i.e. with respect to the *nominal feature space* $\mathcal{X}_N$. To that end, the parameters are updated at each training iteration using the gradient of the reconstruction error (also called the loss):

$$E(t_w) = \left\| \mathbf{X}_N^i(t_w) - g_{\theta_d}(f_{\theta_e}(\mathbf{X}_N^i(t_w))) \right\| \tag{13}$$

Once the training has been accomplished, the parameters θ_e and θ_d of the encoder and decoder are frozen. Let us now consider a TTF trajectory that is being recorded on a running device (i.e. a real system



(a) Prediction without overlapping ($\Delta_{input} = 50, \Delta_{pred} = 20, \delta = 0$). Discontinuities between predicted windows are circled in dashed lines.

(b) Prediction with overlapping ($\Delta_{input} = 50, \Delta_{pred} = 30, \delta = 10$). The overall prediction obtained ensures better continuity between predicted windows

Fig. 4. Comparison of two chained sequence-to-sequence predictions made with a simple LSTM network, one with overlapping and the other without. In both cases, 4 chained predictions have been performed.

which, like all systems, is gradually deteriorating). In such a trajectory, the data can be said nominal (i.e. in $\mathcal{X}_N$) only at the very beginning of the life of the device. Then, as system life progresses, the actual feature space, including degradation, moves away from the *nominal feature space*.

Therefore, considering a system that is deteriorating, a TTF trajectory is denoted as $\mathbf{X}_D^i$ and belongs to a *degraded feature space* $\mathcal{X}_D$.

As the *nominal feature space* and the *degraded feature space* are different ($\mathcal{X}_N \neq \mathcal{X}_D$), the associated probability distribution is also different. More formally, the distribution of data samples belonging to the *degraded feature space* drifts increasingly from the distribution of nominal samples as the system approaches its failure (i.e its EOL) ($P(\mathbf{X}_N^i) \neq P(\mathbf{X}_D^i)$). For each window of a TTF trajectory, the following reconstruction error (scalar) is computed:

$$E_{total}(t_w) = \sum_{k=w}^{w+\Delta} \left\| \mathbf{X}_D^i(t_k) - g_{\theta_d^\star}(f_{\theta_e^\star}(\mathbf{X}_D^i(t_k))) \right\| \tag{14}$$

Due to the distribution shift in $P(\mathbf{X}_D^i)$, the reconstruction error $E_{total}(t_w)$ continues to grow until the system completely fails (that is, as w approaches T_i). This time-varying and monotonically increasing reconstruction error is used as an indicator of the SOH, namely as HI. Since any system degrades, inducing a deviation in all or part of the signals collected through the sensors arranged on the system, this HI based on the reconstruction error will inevitably increase in correlation with degradation.

2.3. Step 3 - HI long-term prediction for RUL estimation

Once a HI has been obtained for a portion of an aircraft's life, it must be forecast into the future to determine its evolution and thus deduce the RUL. As a reminder, one of the aims of the proposed approach is to avoid using RUL-labeled data. Consequently, the RUL estimate is deduced from the HI prediction. Therefore, it is proposed to perform long-term predictions of HI, until a predetermined threshold is reached. Such long-term HI predictions then lead to RUL prediction. To accomplish this task of time series prediction, RNN have been shown to be very efficient. It is a particular type of neural networks that uses both prior knowledge and current input to predict the output. An RNN can be thought of as multiple recurrent standard cells whose states are affected by both past states and current input. In addition, LSTM is a special RNN cell that has been widely used for RUL prediction [23,74]. Time series forecasts are generated using LSTM based sequence-to-sequence prediction framework.

2.3.1. Sequence-to-sequence prediction

Sequence-to-sequence prediction consists of predicting an output sequence given an input sequence [75].

$$\left[X_{t_{k=1}}, \ldots, X_{t_{k=\Delta_{input}}}\right] \mapsto \left[\hat{X}_{t_{k=\Delta_{input}+1}}, \ldots, \hat{X}_{t_{k=\Delta_{input}+\Delta_{pred}}}\right] \tag{15}$$

where Δ_{input} is the length of the input sequence and Δ_{pred} is the length of the predicted output sequence.

In order to make longer-term predictions, it is possible to extend the process, reusing the sequence previously predicted to generate future predictions [76,77].

$$\begin{gathered}\left[X_{t_{k=1}}, \ldots, X_{t_{k=\Delta_{input}}}\right] \mapsto \left[\hat{X}_{t_{k=\Delta_{input}+1}}, \ldots, \hat{X}_{t_{k=\Delta_{input}+\Delta_{pred}}}\right] \\ \left[\hat{X}_{t_{k=\Delta_{input}+1}}, \ldots, \hat{X}_{t_{k=\Delta_{input}+\Delta_{pred}}}\right] \mapsto \left[\hat{X}_{t_{k=\Delta_{input}+\Delta_{pred}+1}}, \ldots, \hat{X}_{t_{k=\Delta_{input}+2\Delta_{pred}}}\right] \\ \text{etc.}\end{gathered} \tag{16}$$

However, when such chain predictions are made, discontinuities between the windows are often observed experimentally. To prevent the latter, this work proposes to enhance the degree of correlation between the input sequence and the output sequence by applying an overlapping between them. To illustrate this, Fig. 4 shows two chain predictions, one without overlapping (i.e. $\delta = 0$) and the other with ($\delta = 10$). In the first case (Fig. 4(a)) discontinuities appear between the predicted windows, affecting the quality of the prediction. In the other case (Fig. 4(b)), the effective prediction length is the same as in the previous case, but overlapping is introduced. As a result, the discontinuities that had appeared in the first case have been significantly smoothed out, resulting in a better prediction. The choice of hyperparameters Δ_{input}, Δ_{pred} and δ has to be done on an empirical basis, according to the specific situation. A special study of the influence of these hyperparameters is conducted for the Dassault Aviation case study in Section 3.3.

2.3.2. Stacked-LSTM structure

For the HI sequence-to-sequence prediction task, a stacked-LSTM architecture is proposed [78]. It takes as input a sequence of HI values, and generates as output: a sequence of future HI values (prediction).

$h_{t_k}^l$ is the hidden state computed by the l-layer of a stacked-LSTM architecture at timestep t_k. The hidden states of the l-layer serve as input to the deeper $(l+1)$ layer.

The specific feature of the structure proposed here is that, at the output of the last LSTM layer of the network, the entire set of hidden states h_1^L to $h_{\Delta_{input}}^L$ are concatenated and vectorized, to be supplied to

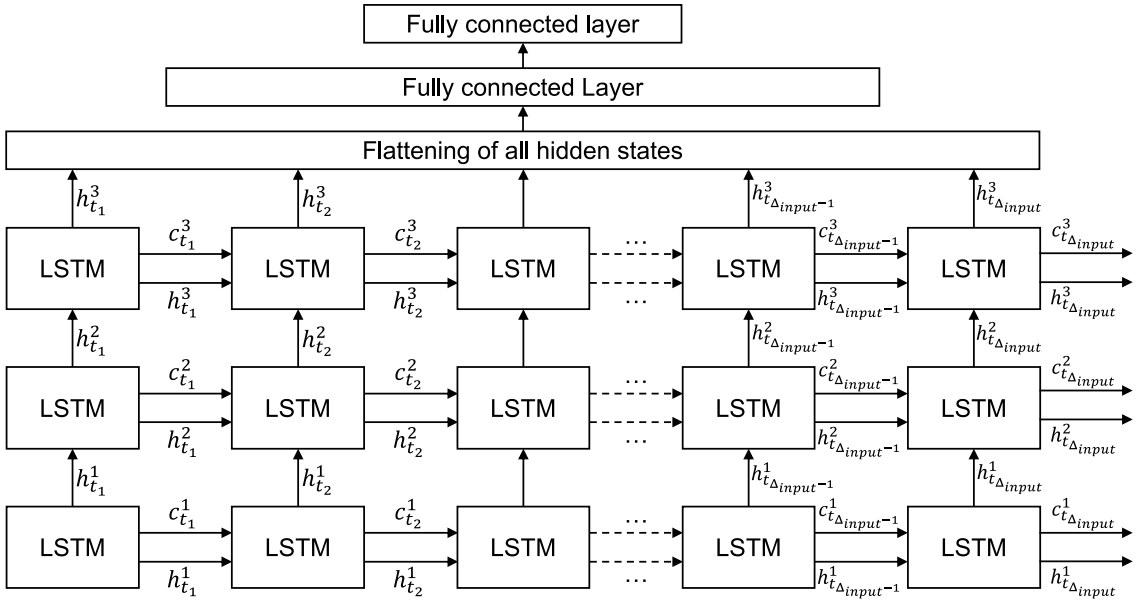


Fig. 5. Structure of a three-layered stacked-LSTM network. Here, $L = 3$ and $h^l_{t_k}$ (respectively $c^l_{t_k}$) refers to the hidden state (respectively the cell state) of layer l at timestep t_k.

a network of FCL that in turn, lead to a predicted time window of the desired size. This process ensures extensive use of all temporal information from the start to the end of the input sequence.

The training of this stacked-LSTM structure is done in a supervised manner. As such, a training dataset comprises complete HI time series (up to failure, that is, TTF trajectories), obtained from the degraded data augmentation process described in Section 2.1.2.

2.3.3. RUL prediction based on threshold overshooting

The final objective is to obtain an estimate of the RUL, but without using RUL-labeled data. To do this, a HI sequence-to-sequence prediction is performed using the stacked-LSTM structure presented in Section 2.3.2. The prediction is generated in sequential or chained fashion (i.e., by reusing previously predicted windows) until a threshold is reached [76], corresponding to the EOL. At this point, the RUL can be recursively deduced by counting the time-steps covered. This strategy is depicted in Fig. 6.

The detailed RUL estimation algorithm based on the prediction of long-term HI until reaching an EOL threshold can be found in Algorithm 1.

Algorithm 1: RUL prediction process using HI sequential prediction threshold overshooting strategy.

Input: A HI window of length Δ_{input} : $\mathbf{HI}(t_w) = \left\{ \mathrm{HI}_{t_k} \right\}_{k=w}^{w+\Delta_{input}}$.
Output: A RUL value (scalar).
$q = 0$;
$\mathbf{HI}(t_{w+\Delta_{input}-\delta}) \leftarrow h_\Theta(\mathbf{HI}(t_w))$ where h_Θ can be any NN-based approximation model parameterized by Θ (here, it is the stacked-LSTM network presented in Fig. 5);
while $max(\mathbf{HI}(t_{w+\Delta_{input}-\delta})) < EOL_{threshold}$ **do**
  $q \leftarrow q + 1$;
  $\mathbf{HI}(t_{w+\Delta_{input}-\delta+\Delta_{pred}}) \leftarrow h_\Theta(\mathbf{HI}(t_{w+\Delta_{input}-\delta}))$ (next predicted window from previous predicted window);
end
$\mathrm{RUL} = \Delta_{input} + (q-1) \times (\Delta_{pred} - \delta) + (\Delta_{EOL} - \delta)$;

2.3.4. Reliability-based assessment of the AI-based approach

Finally, a reliability-based evaluation procedure is proposed to provide an assessment of RUL predictions against the well established reliability laws. The objective is to show that despite the black-box nature of the approaches, they are consistent with established reliability laws. Considering a fleet of systems whose failure probability follows a two-parameter Weibull distribution, with the failure rate defined as [46]:

$$\lambda(t) = \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1} \tag{17}$$

with $\beta, \alpha, t > 0$.

The failure probability (Weibull Cumulative Distribution Function (CDF)) is expressed as [79]:

$$F(t) = 1 - e^{-\left(\frac{t}{\alpha}\right)^{\beta}} \tag{18}$$

β is called the shape parameter while α is the scale parameter, influencing both mean and spread of distribution.

To that end, a set of TTF trajectories are generated using the data-augmentation process (Step 1, presented in Section 2.1), with respect to a pre-selected Weibull distribution $F_{ref}(t)$. Then, these trajectories are tested through the autoencoder, already trained (see Step 2 described in Section 2.2), to obtain a HI trajectory. From the HI trajectories, sequence-to-sequence predictions of future HI values are performed (Step 3 described in Section 2.3) that leads to RUL predictions. The EOL instants can then be retrieved, thus allowing to identify the parameters of the Weibull distribution obtained from RUL predictions $\widehat{F}_{pred}(t)$.

The Kolmogorov–Smirnov test [80], which has been designed to compare a sample with a reference probability distribution, is applied to ensure that the probability of failure $\widehat{F}_{pred}(t)$ obtained from the RUL predictions follows the pre-selected distribution $F_{ref}(t)$.

3. Application results to a real-life industrial use case

The proposed approach has been tested in a real industrial setting in collaboration with Dassault Aviation, focusing on a commercial business jet's air distribution system within the cockpit. This system regulates cockpit temperature by collecting hot air from the engines (approximately 200 °C), cooling it, and directing it into the cockpit through a valve. Data for the study was collected during test flights conducted during the development phase, aimed at testing various on-board systems, rather than commercial flights.

The three stages outlined in Section 2 are applied to assess the effectiveness of the global approach.

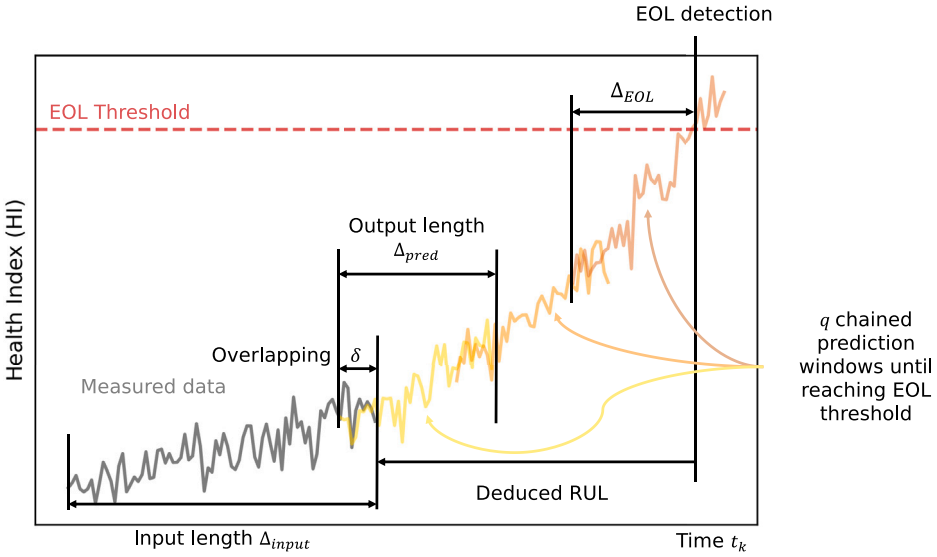


Fig. 6. Overall view of the threshold overshooting strategy. q windows of length Δ_{pred} are predicted, until the EOL threshold is reached. RUL value is then deduced by counting the timesteps covered between the start point and the EOL. Δ_{input} and Δ_{output} are the length of the input and output sequences, δ is the overlapping and Δ_{EOL} is the length of the partial last predicted window, which is cut off at the time-step where the EOL threshold is attained.

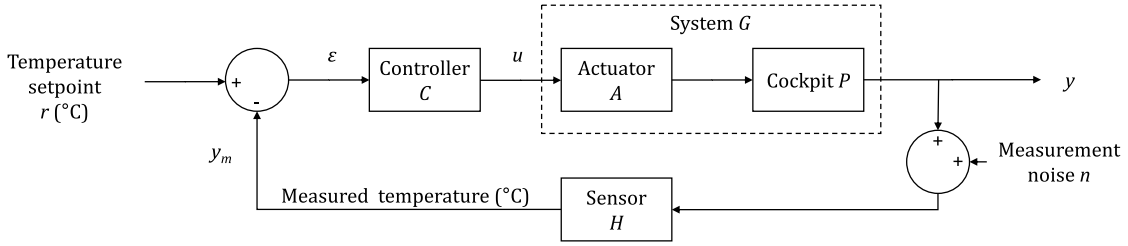


Fig. 7. Block diagram of the air distribution system in the cockpit. The controller C is driven by the error ε, obtained by subtracting the measured temperature y_m from the temperature setpoint r. The measured temperature y_m is the true system output y corrupted by noise n. The command u is given to the actuator A, which is a solenoid valve, allowing the cockpit P temperature to be adjusted by controlling the valve opening.

3.1. Step 1 - Hybrid data augmentation

3.1.1. Nominal data augmentation using data-driven system identification

Based on *a priori* knowledge of temperature control and preliminary data analysis, a block diagram of the closed-loop system has been constructed, as shown in Fig. 7.

Data selection. As presented previously (Section 2.1.1), it is first required to locate data segments corresponding to a "manual mode" (that is, open-loop control) and an "auto mode" (i.e., closed-loop control) in existing historical data. Among the ten exploitable flights, forced closure of the actuator (namely the valve) could be observed, corresponding to a "manual mode" (i.e. open loop control), generating possibility for identification the system model G, as proposed in Section 2.1.1. This particular flight is shown in Fig. 8.

On the other hand, among the ten flights available, there are several historical data segments corresponding to an "auto mode", with no setpoint variation (i.e. r remaining constant). that is the one that comes from the most recent flight (i.e. most up-to-date configuration of the system) and provides the largest number of points. This flight displayed in Fig. 9 is used for identification of the inverse of the controller.

Identification of the system model G. During a full valve closure command (see Fig. 8), the system operates in the "manual mode" (ie, as in open loop) so that the block diagram becomes the one shown in Fig. 10.

Firstly, the actuator A and the cockpit P are grouped into the lumped system model G. The sensor dynamics is much faster than the system dynamics G and therefore is considered negligible. The model is thus described by:

$$y_m = Gu + e \tag{19}$$

where G is modeled as a first-order transfer function taking into account the delay:

$$G(s) = \frac{Ke^{-\tau s}}{1 + Ts} \tag{20}$$

where τ is the delay, T the time-constant, K the steady-state gain and e is a white noise (this assumption corresponds to the use of the Simplified Refined Instrumental Variable method for Continuous-time systems (SRIVC) algorithm [54,55]). The parameters of the model are determined by SRIVC algorithm which is available in the CONtinuous-Time System IDentification (CONTSID) toolbox [81]. For $G(s)$, the following model is thus obtained:

$$\widehat{G}(s) = \frac{(0.3 \pm 0.002)e^{(-8\pm0.94)s}}{1 + (50.5 \pm 1.6)\ \mathrm{s}} \tag{21}$$

where the value after the $\pm$ sign represents the estimated standard deviation. This model $\widehat{G}(s)$ has been estimated by using the second data segment from the flight of October 5 (See Fig. 8). The model's ability to reproduce the temperature can be evaluated from the results depicted in Fig. 11. On both estimation and cross-validation data, the simulation (noise-free) obtained by the approximate low-order model $\widehat{G}(s)$ is able to capture the temperature response of the system.

Identification of the controller model C. The controller C is assumed to be a Proportional Integral (PI) controller. In the selected data segment (see Fig. 9), the system is in "auto mode" and the set point remains

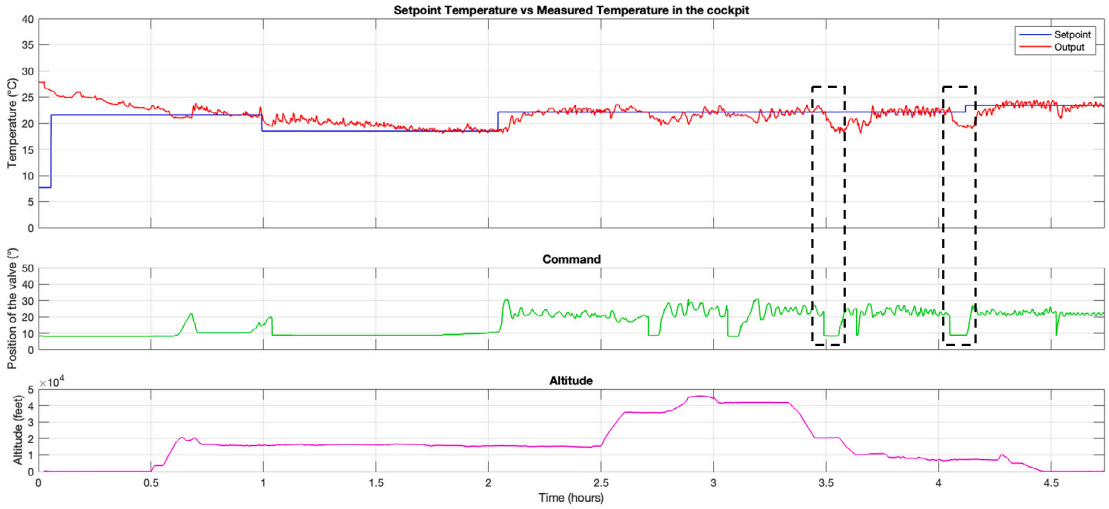


Fig. 8. Flight of October 5, 2021. The top plot shows the setpoint temperature r and the measured temperature y_m, the middle plot shows the command u and the bottom plot shows the altitude. The data segments corresponding to a "manual mode" selected for the estimation and the validation of the system model G are framed in dashed lines.

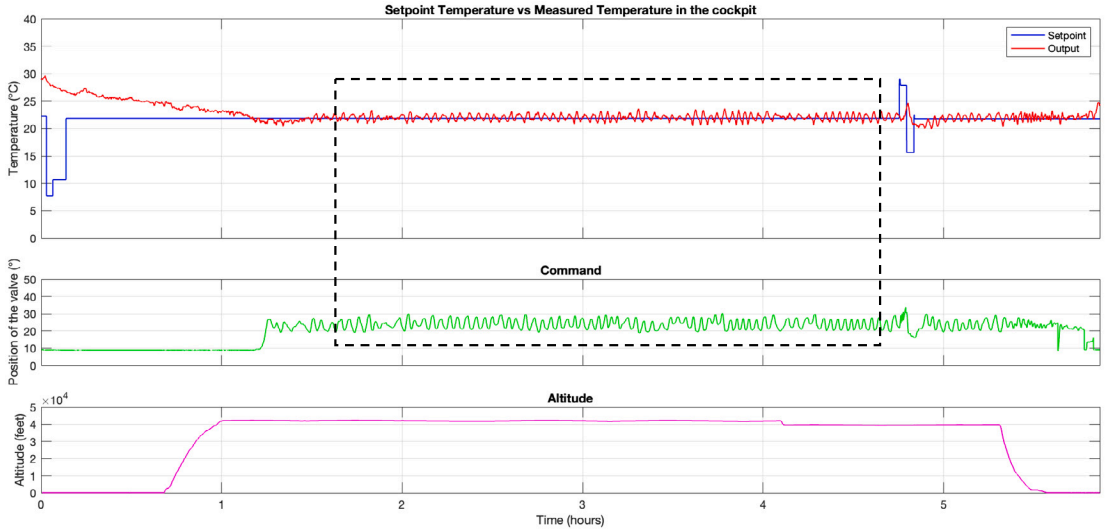


Fig. 9. Flight of September 29, 2021. The data segment selected to identify the inverse of the controller C is framed in dashed lines.

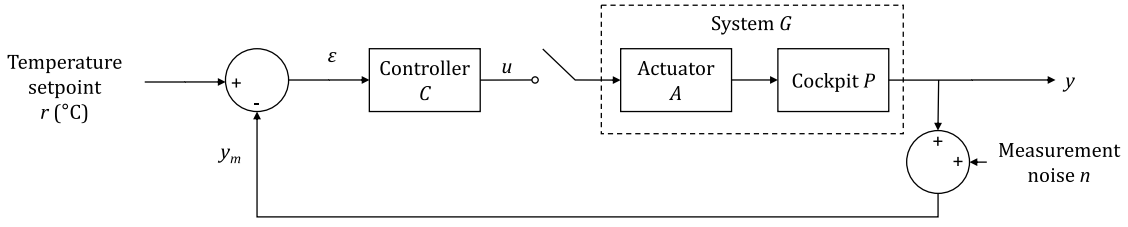


Fig. 10. Block diagram of the air distribution system operating in "manual mode" (i.e. open-loop) during a full closure command of the valve. In this particular case it becomes possible to identify the system model G.

constant. As stated in Section 2.1.1, the transfer function that is then identified (denoted as $L(s)$) results in:

$$\frac{y_m(s)}{u(s)} = L(s) = -\frac{1}{C(s)} \tag{22}$$

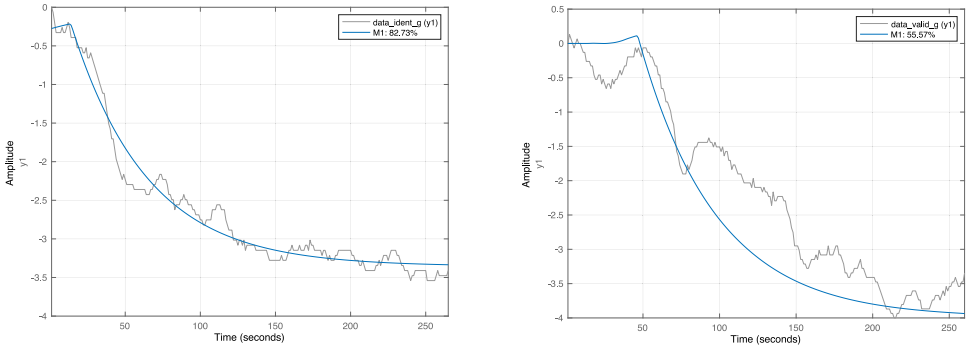
where

$$L(s) = \frac{b_1 s + b_0}{s + a_0} \tag{23}$$

The parameters of $\hat{L}(s)$ are first estimated by using the SRIVC algorithm from which the controller estimate $\hat{C}(s)$ is deduced as:

$$\hat{C}(s) = K_p\left(\frac{1 + T_i s}{T_i s}\right) = 1.6\left(\frac{1 + 8.6\ \text{s}}{8.6\ \text{s}}\right) \tag{24}$$

Validation and data augmentation. Given the difficulty of validating individually the controller model $\hat{C}$ in a closed-loop context, it will be validated in a complete closed-loop setting [82,83].



(a) Estimation data. (b) Cross-validation data.

Fig. 11. Comparison of the measured and simulated $\hat{G}$ model responses.

However, the measurement noise effects must be modeled first. To this end, an estimate $\hat{n}$ of the measurement noise has been generated from a low-pass AR(1) model [52,54]:

$$\hat{n}(t_k) = \frac{1 + d_1}{1 + d_1 q^{-1}} e(t_k) \tag{25}$$

where q^{-1} is the delay operator and $e(t_k)$ is a white noise process of variance σ_e^2. d_1 has been empirically set to 0.95 to low-pass filter the white noise while the variance of $e(t_k)$ has been empirically adjusted to reproduce the measurement noise levels observed in the historical data.

The model of the full closed-loop system is then validated by replicating historical flights in simulation, to assess whether the essential dynamics have been captured as depicted for a flight profile in Fig. 12. Validation by flight replication shows that the temperature control behavior is effectively reproduced.

From the identified process and controller models, it is then possible to generate completely new data (nominal data augmentation). This is done by specifying the desired flight time and setting the setpoint profile close to the operating point (i.e. around 22.5 °C) to ensure that the linearity assumption is met as closely as possible. The general model of the system for generating additional data is shown in Fig. 13.

It should be noted that there are two main advantages of using system identification to perform data augmentation:

- Firstly, augmented data (i.e. additional generated data) can be obtained with a very limited amount of estimation data and system knowledge. In comparison, it is generally necessary to use a large amount of data to train generative AI-based models such as GANs.
- Secondly, the use of low-order models provides guarantees on the behavior of the generated data. The use of highly complex deep learning models (such as GANs) is generally limited by the lack of consistency guarantees for the dynamics of the data generated (for example, the outliers can be generated unexpectedly).

3.1.2. Physics-based degraded data augmentation

The second aspect of data augmentation consists of injecting progressive degradation into one component of the air distribution system in order to generate TTF trajectories. In the signals obtained from test flights, minor oscillations are observed in the valve position. This phenomenon can stem from various sources, such as valve clogging [65], which may indicate a friction defect. Consequently, it is determined that the construction of a stiction model is necessary to simulate a degradation process in the actuator (the valve) by gradually intensifying the stiction phenomenon until complete failure.

One well-established stiction model is as follows [84]:

$$x_k = \begin{cases} x_{k-1} + (e_k - sign(e_k) f_D), & \text{if } |e_k| > f_s \\ x_{k-1}, & \text{if } |e_k| \leq f_s \end{cases} \tag{26}$$

with $e_k = u_k - x_{k-1}$. f_S and f_D are respectively the static and dynamic friction parameter. The graphical characteristic of a valve under such a stiction phenomenon is depicted in Fig. 14.

To simulate valve degradation, the parameters f_S and f_D are increased. Initially, a degradation mode is induced by solely increasing the parameter f_S. Practically, this translates to a rise in static friction resistance over the valve's lifespan, compelling the controller to increment the command value for the same movement. Consequently, this leads to progressively larger oscillations in the setpoint and a jerky valve behavior, as depicted in Fig. 15. The increase in f_S parameter follows a generic exponential degradation model, as elaborated in Section 2.1.2.

Initially, at the start of the system, the valve operates without friction ($f_S = 0$), maintaining a perfectly linear relationship between the controller command and the valve position. This behavior is illustrated in Fig. 16, where both variables align precisely. As the system ages, the valve begins to experience clogging, resulting in increased dry friction. Consequently, the valve's response to the controller command becomes imperfect, exhibiting intermittent operation. Fig. 17 illustrates this behavior for $f_S = 15$.

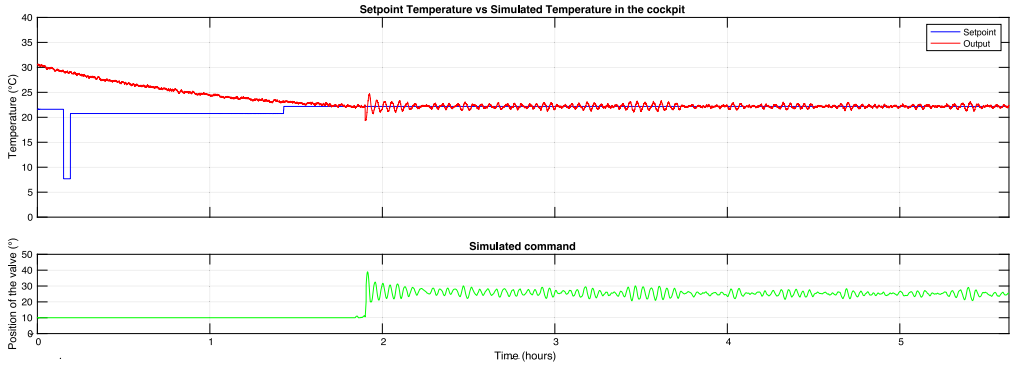
Fig. 18 illustrates an example of a generated Time-to-Failure (TTF) trajectory. At the system's End of Life (EOL), the valve becomes stuck in a position from which it cannot move. This occurs because the command range no longer permits a sufficiently high command to initiate valve movement, signifying complete failure and reaching the system's EOL. However, it is essential to note that in real-life scenarios, such abnormal valve behavior should be detected by the monitoring system well in advance of complete valve blockage.

3.2. Step 2 - Unsupervised health index extraction

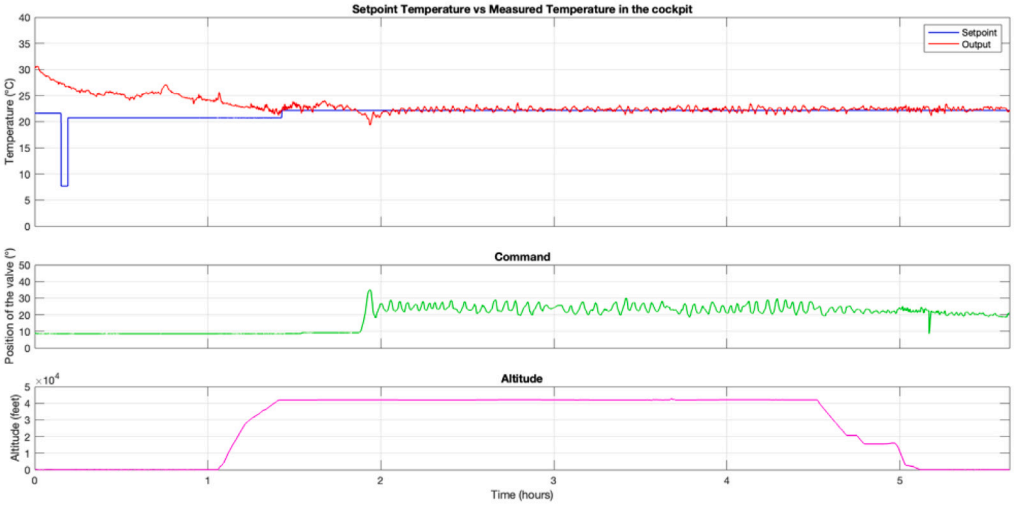
3.2.1. Experiment details

As proposed in Section 2.2, an autoencoder is trained to reproduce healthy data samples. Consider N_N nominal training flight index $1 \leq i \leq N_N$, $i \in \mathbb{N}$ for which data from S sensors $1 \leq s \leq S$, $s \in \mathbb{N}$ are collected. Namely, four signals are collected in this experiment (the temperature setpoint r, the command u, the measured temperature y_m and the valve position v), i.e. $S = 4$. Each training flight record i has a total duration denoted as $\mathcal{T}_i$, with index $t_1 \leq t_k \leq t_{\mathcal{T}_i}$, $k \in \mathbb{N}$.

Therefore, the nominal training set is the collection of objects: $\{\mathbf{X}_N^i\}_{i=1}^{N_N}$ with each object $\mathbf{X}_N^i \in \mathbb{R}^{\mathcal{T}_i \times S}$. The sth column $X^{s(i)}$ therefore corresponds to the vector of values of sensor s for the entire duration of the flight i (i.e. for all time-steps $t_1 \leq t_k \leq t_{\mathcal{T}_i}$) and the t_k-th row $X_{t_k}{}^{(i)}$ corresponds to the vector of values of all sensor $1 \leq s \leq S$ for the given timestep t_k. Finally, the scalar $X_{t_k}^{s\,(i)}$ is the single value recorded by



(a) Replicated flight



(b) Measured flight data

Fig. 12. Replication of flight of September 21, 2021 with the *system identification-enabled nominal data augmentation process*. Throughout the flight phase, when the control is active, the general dynamics is properly reproduced. However, towards the end of the recording (i.e., during landing and when the aircraft is on the tarmac), the behavior of the true system is not linear any more, which may be caused by unknown external disturbances (outside temperature, door opening, etc.).

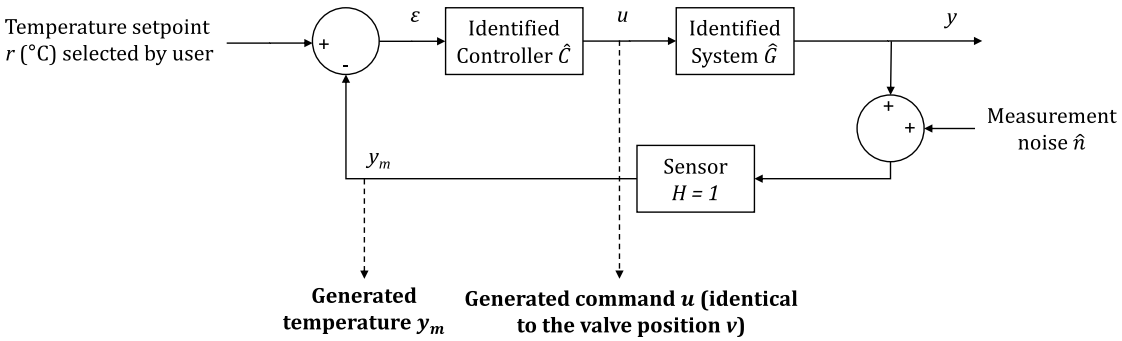


Fig. 13. Block diagram of the temperature feedback control system model used to generate additional nominal data (generated variables are in bold).

sensor s at time-step t_k on flight i. This can be summarized as follows:

$$\mathbf{X}_N^i = \begin{bmatrix} X_{t_1}^{1\,(i)} & X_{t_1}^{2\,(i)} & \cdots & X_{t_1}^{S\,(i)} \\ X_{t_2}^{1\,(i)} & X_{t_2}^{2\,(i)} & \cdots & X_{t_2}^{S\,(i)} \\ \cdots & \cdots & \ddots & \cdots \\ X_{t_{T_i}}^{1\,(i)} & X_{t_{T_i}}^{2\,(i)} & \cdots & X_{t_{T_i}}^{S\,(i)} \end{bmatrix} = \begin{bmatrix} X_{t_1}{}^{(i)} \\ X_{t_2}{}^{(i)} \\ \cdots \\ X_{t_{T_i}}{}^{(i)} \end{bmatrix} = \begin{bmatrix} X^{1\,(i)} & X^{2\,(i)} & \cdots & X^{S\,(i)} \end{bmatrix} \tag{27}$$

The nominal training data samples are then divided into several windows:

$$\mathbf{X}_N^i(t_w) = \left\{ X_{t_k}{}^{(i)} \right\}_{k=w}^{w+\Delta} \in \mathbb{R}^{\Delta \times S} \tag{28}$$

where t_w is the start timestep of the window and $\Delta = 60$ s is its total duration. Δ is a hyperparameter that depends on the system under study and its dynamics. In particular, $\Delta = 60$ s has **proven** to be an adequate range for capturing the dynamics of the control in operation.

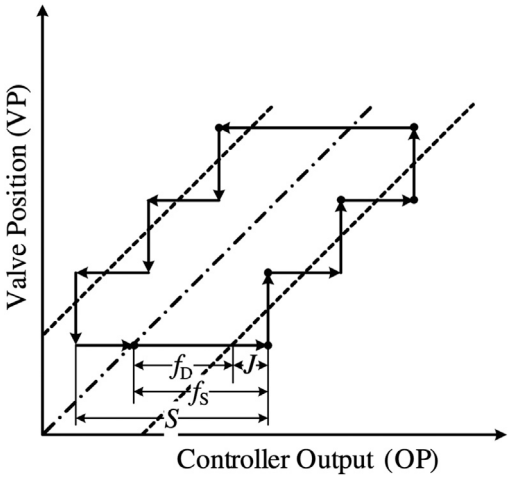


Fig. 14. Valve stiction modeling [85].

Respecting the sensor notation of the case study, this gives the following time window:

$$\mathbf{X}_N^i(t_w) = \begin{bmatrix} r_{t_w}^{(i)} & u_{t_w}^{(i)} & y_{m\,t_w}^{(i)} & v_{t_w}^{(i)} \\ r_{t_{w+1}}^{(i)} & u_{t_{w+1}}^{(i)} & y_{m\,t_{w+1}}^{(i)} & v_{t_{w+1}}^{(i)} \\ \cdots & \cdots & \cdots & \cdots \\ r_{t_{w+\Delta}}^{(i)} & u_{t_{w+\Delta}}^{(i)} & y_{m\,t_{w+\Delta}}^{(i)} & v_{t_{w+\Delta}}^{(i)} \end{bmatrix} \tag{29}$$

3.2.2. Data preprocessing

During the training phase, the autoencoder receives healthy data comprising ten flights generated via the healthy data augmentation process outlined in Section 2.1.1. To simulate real-world conditions accurately, these generated flights have random durations ranging between four and seven hours. Furthermore, the initial set-point is randomly set between 22.5 and 25 °C, and within each flight, a random amplitude and change in time set-point step is implemented. To ensure uniform feature weighting in the reconstruction training process, each sensor signal $X^{s(i)}$ is standardized as

$$X_{t_k\ scaled}^{s\ (i)} = \frac{X_{t_k}^{s\ (i)} - \mu(s)}{\sigma(s)} \tag{30}$$

with mean of sensor s:

$$\mu(s) = \frac{1}{N_N \times T_i} \sum_{i=1}^{N_N} \sum_{k=1}^{T_i} X_{t_k}^{s\ (i)} \tag{31}$$

and standard deviation of sensor s:

$$\sigma(s) = \sqrt{\frac{1}{N_N \times T_i} \sum_{i=1}^{N_N} \sum_{k=1}^{T_i} \left[X_{t_k}^{s\ (i)} - \mu(s) \right]^2} \tag{32}$$

An analysis of the evolution of the distribution of characteristics over time in ten TTF trajectories generated in the same way as in Fig. 18 can be found in Fig. 19. Taking the first quarter of life as a reference, the following observations can be formulated:

- In the second quarter of life, the medians for each of the sensors remain fairly unchanged, but the outliers have widened, as have the boundaries of the whiskers.
- By the third quarter of life, the medians have moved slightly away from their initial value and the interquartile range has increased significantly, as have the whiskers.
- In the last quarter of life (i.e., the fraction of life in which EOL is reached), the general distribution of each sensor is completely different from its initial distribution.

All this demonstrates that the feature space deviation hypothesis formulated in Section 2.2.2 ($\mathcal{X}_N \neq \mathcal{X}_D$) is evident.

Table 1
Set of hyperparameters for Dassault Aviation experiment.

Hyper-parameters	Values
Size of encoder layers	240, 120, 60, 30
Size of decoder layers	60, 120, 180
Window size Δ	60 s
Range of durations $\mathcal{T}_i$ of flight records for training	4 to 7 h
Iterations (also called Epochs)	200

3.2.3. Autoencoder structure and hyperparameters

The autoencoder structure used for sensor reconstruction consists of Fully Connected Layers (FCL). Each time window is vectorized to produce a vector ($\text{vec}(\mathbf{X}_N^i(t_w))$) of length $60 \times 4 = 240$, which serves as the input layer's size. The decoder mirrors the encoder's structure symmetrically, excluding the reconstruction of the set-point signal. The complete network structure is illustrated in Fig. 20, with details of the hyperparameters provided in Table 1.

The unsupervised HI extraction process can be visualized in Fig. 21, which distinguishes the training part (Fig. 21(a)) from the testing part (Fig. 21(b)).

More formally, the process of training the autoencoder to reconstruct multivariate sensor windows is described in Algorithm 2.

Algorithm 2: Autoencoder training for sensor reconstruction.

Input: A training set of N_N nominal (i.e. healthy) flight records $\{\mathbf{X}_N^i\}_{i=1}^{N_N}$, each of length $\mathcal{T}_i$.
Output: A trained autoencoder model with optimal weights $\theta_e^\star, \theta_d^\star$.

```
for iter = 1 to K do
    for i = 1 to N_N do
        Divide the flight record X_N^i into vectorized windows
         vec(X_N^i(t_w)) of duration Δ;
        for each vectorized window do
            vec(X̂_N^i(t_w)) ← g_{θ_d}(f_{θ_e}(vec(X_N^i(t_w))));
            E(t_w) = ‖vec(X_N^i(t_w)) − vec(X̂_N^i(t_w))‖;
            loss ← loss + E(t_w);
        end
        L_iter ← L_iter + loss;
    end
    Update θ_e, θ_d by computing gradient descend on iteration
     loss L_iter;
end
```

Once training is complete, the optimal weights $\theta_e^\star$ and $\theta_d^\star$ are frozen. The test process, i.e. the HI extraction from one TTF trajectory under degradation of duration $\mathcal{T}_i$, can be formulated as the total reconstruction of vector $vec(\mathbf{X}_D^i(t_w))$ in the trajectory, with t_w in t_1 to $t_{\mathcal{T}_i}$. This is summarized in Algorithm 3. Note that, for the present study, a trajectory obtained using the degraded data augmentation process developed in Section 3.1.2 is used as test data.

3.2.4. Experimental results

According to the process described in Fig. 21, the reconstructed signals are compared to the original ones, providing the evolution of the reconstruction error over time. The signals collected during the entire life of one example aircraft, together with the associated HI at each instant are displayed in Fig. 22.

It can be observed that the reconstruction error increases with the degradation of the system until the EOL is attained, just before 2000 min of life.

Finally, the HI trajectories are normalized between zero and one, taking into account the training trajectories (since the test trajectories

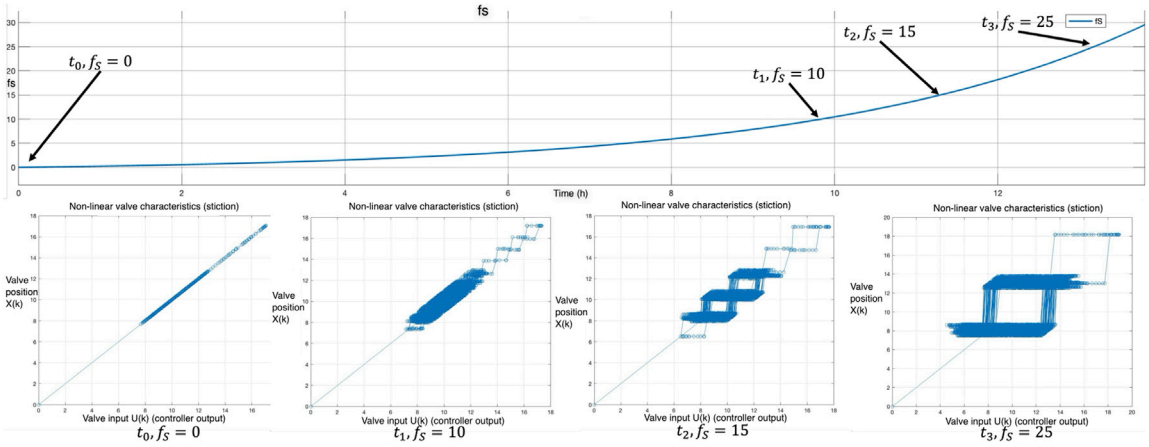
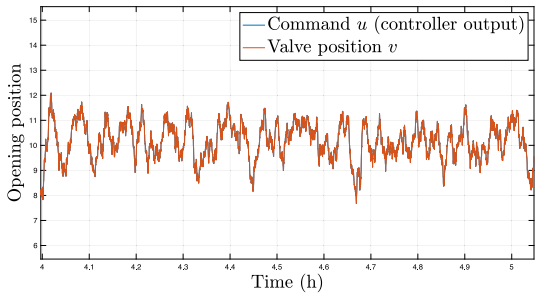
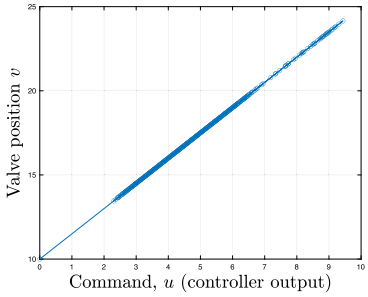


Fig. 15. The evolution of the valve characteristic as a function of the increase of the f_S coefficient.

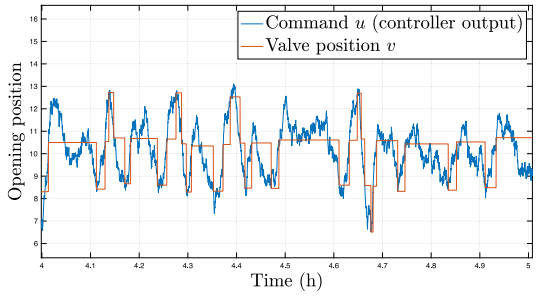


(a) Sample plot of the command sent by the controller and the valve position when $f_S = 0$. Both signals are identical in this situation.

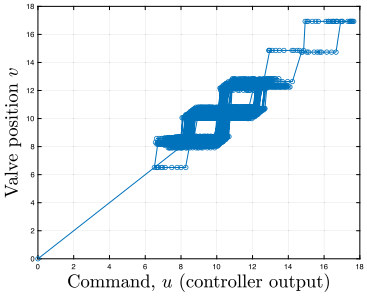


(b) The valve characteristic when $f_S = 0$.

Fig. 16. Valve behavior when $f_S = 0$ (perfectly linear).



(a) Sample plot of the command sent by the controller and the valve position when $f_S = 15$.



(b) The valve characteristic when $f_S = 15$.

Fig. 17. Valve behavior when $f_S = 15$ (i.e. half degradation).

Algorithm 3: HI extraction procedure using the reconstruction error for one test trajectory.

Input: One test trajectory $\mathbf{X}_D^i$ of duration $\mathcal{T}_i$.
Output: A HI trajectory $\mathbf{HI}_i$ of duration $\mathcal{T}_i$.
Divide the trajectory $\mathbf{X}_D^i$ into vectorized windows $vec(\mathbf{X}_D^i(t_w))$ of duration Δ;
for ***each*** *vectorized window* **do**
  $\widehat{vec(\mathbf{X}_D^i(t_w))} \leftarrow g_{\theta_d^\star}(f_{\theta_e^\star}(vec(\mathbf{X}_D^i(t_w))))$;
  $\mathrm{HI}_{t_w} \leftarrow \sum_{k=w}^{w+\Delta} \left\| vec(\mathbf{X}_D^i(t_k)) - \widehat{vec(\mathbf{X}_D^i(t_k))} \right\|$;
end

are not known) according to the following equation:

$$norm(\mathrm{HI}_{t_k}) = \frac{\mathrm{HI}_{t_k} - min(\mathbf{HI}_{train})}{max(\mathbf{HI}_{train}) - min(\mathbf{HI}_{train})} \tag{33}$$

where $\mathbf{HI}_{train} = \{\mathbf{HI}_i\}_{i=1}^N$ is the set of N HI trajectories obtained from the training set.

It should be pointed out that this last step of normalization is optional and allows the HI trajectories to be visualized between 0 and 1 (without any loss of information) in accordance with established practice in the PHM community. To conclude, the proposed approach is tested over a set of ten test TTF trajectories. The median HI trajectory, with standard deviation with respect to the fraction of total life passed

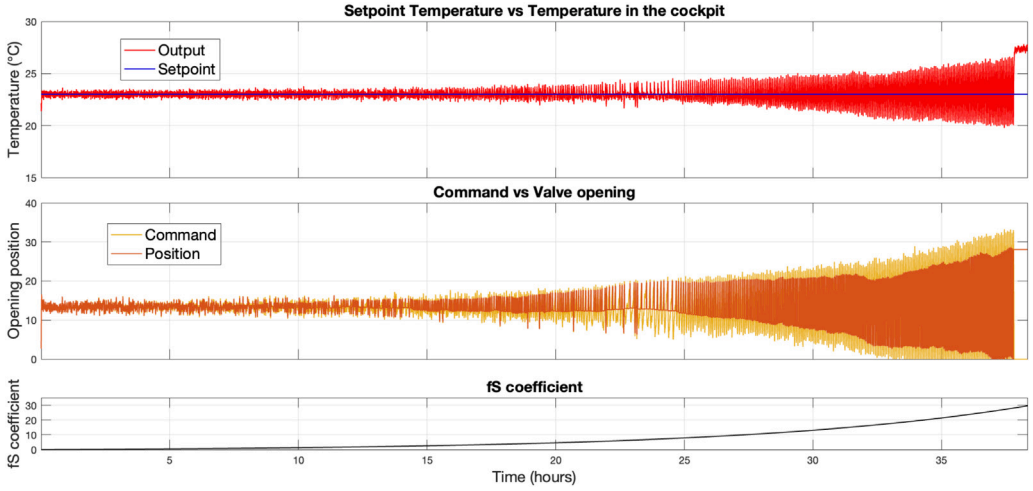


Fig. 18. Example of a TTF trajectory generated by the prognostics-oriented data augmentation process. The oscillation of the measured temperature around the set-point is increasing with the increase of the parameter f_S. At the end of life, the command sent by the controller saturates at 0, so the valve can no longer be opened and remains permanently blocked, ending all control. This corresponds to the EOL.

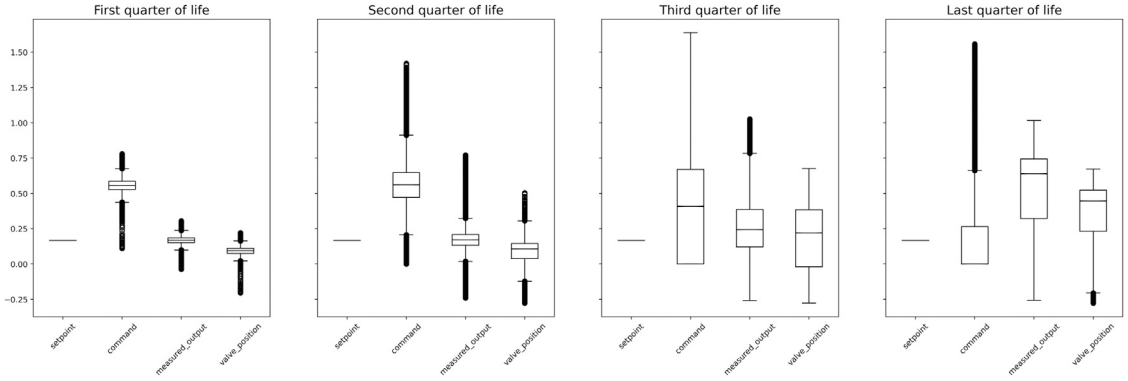


Fig. 19. The deviation of the *degraded feature space* $\mathcal{X}_D$ over time. TTF trajectories are divided into quarters of life, for each of which the distribution of features (i.e. sensors) is plotted in boxplots.

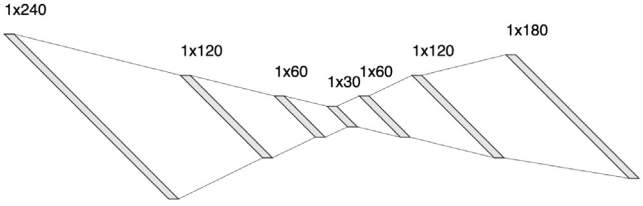


Fig. 20. The structure of the autoencoder for signal reconstruction, with different FCL sizes.

Table 2
Hyperparameters of the long-term HI prediction task.

Hyperparameter	Value
Number of stacked LSTM layers	3
Size of hidden state	240
Dropout	0.2
Size of FCL	$\lvert FCL_1 \rvert = (\lvert h_{t_k} \rvert \times \Delta_{pred})/4$ $\lvert FCL_2 \rvert = \lvert FCL_1 \rvert /4$ $\lvert FCL_3 \rvert = \Delta_{pred}$
Input window length Δ_{input}	
Input window length Δ_{input}	8 h
Output window length Δ_{output}	3 h
Overlapping δ	1.5 h

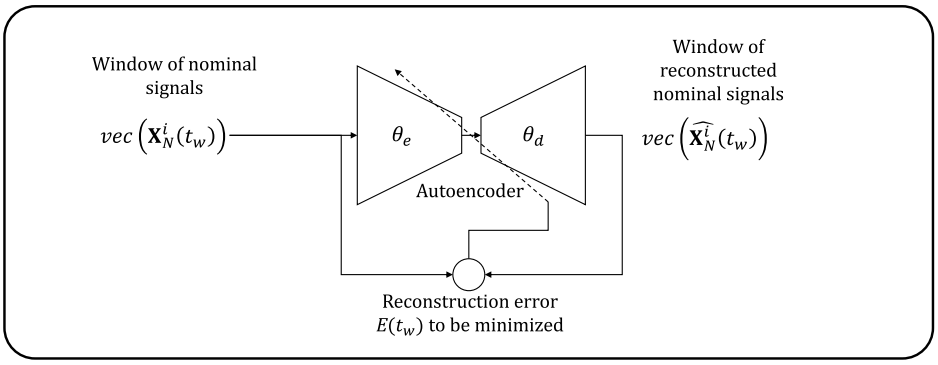
is depicted in Fig. 23 (without normalization), demonstrating the clear trend in the HI obtained by exploiting the reconstruction error.

3.3. Step 3 - HI long-term prediction for RUL estimation

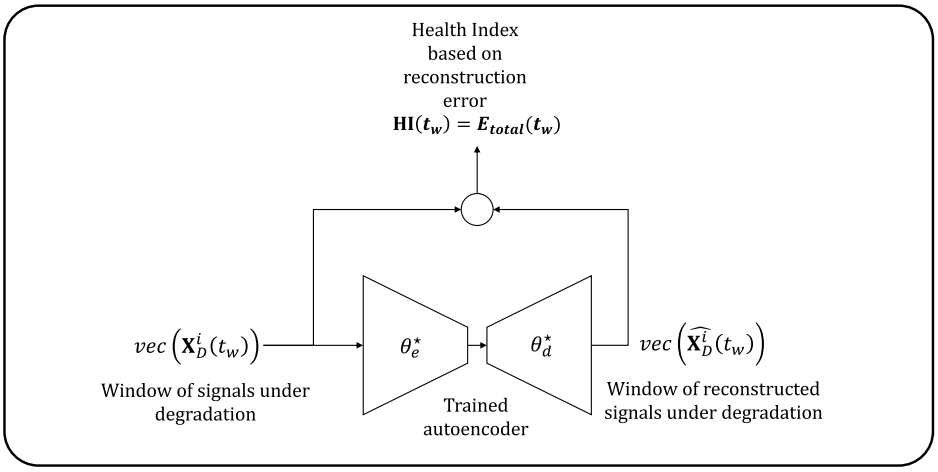
Long-term HI prediction. The set of 10 HI trajectories obtained in Section 3.2.4 serves as training data for the long-term sequence-to-sequence HI prediction model presented in Section 2.3.2. The hyperparameters of the LSTM prediction model, which are empirically chosen during the training phase, can be found in Table 2.

The LSTM prediction model is used to perform long-term chained predictions of sequences of HI, as described in Section 2.3.1, until the EOL threshold (equal to 1) is reached, according to the procedure described in Algorithm 1. Fig. 24 displays two HI chain predictions, for the same TTF test trajectory. In the first prediction (Fig. 24(a)), the prediction starts from $t_k = 480$ minutes, whereas in the second (Fig. 24(b)) the prediction starts at $t_k = 800$ minutes. It is observed that the earlier the prediction starts, the less accurate it is, as the total prediction duration is longer and more prediction errors accumulate.

RUL estimation algorithm. At each timestep of the life of the system, Algorithm 1 is performed, in order to obtain a RUL estimation. When executing it for the whole life of a test system, a complete RUL trajectory until the EOL is obtained. A test result of this RUL prediction process can be seen in Fig. 25, along with RUL ground truth. The estimation of the RUL becomes more and more precise as the EOL is neared (cf. Fig. 24(b)). It is also noted that early in the life of the



(a) Training phase. Learning of the parameters θ_e and θ_d of the encoder and decoder by minimizing reconstruction error E



(b) Testing phase. The optimal parameters $\theta_e^\star$ and $\theta_{d^\star}$ are frozen and the total reconstruction error E_{total} is used as HI.

Fig. 21. Overall process of the unsupervised extraction of HI based on the reconstruction error.

system, the evolution of the HI can lead to inaccurate predictions, as described in Fig. 24(a).

The RMSE for each individual TTF test trajectory can be seen in Fig. 26(a). Similarly, the RMSE computed on the first fifteen and last fifteen RUL predictions is depicted in Fig. 26(b). This confirms, at the level of each individual test trajectory, the significant improvement in prediction as the end of life of the system is approached.

Discussion about the choice of the hyperparameters. As introduced in Section 2.3.1, for sequence-to-sequence prediction, three main hyperparameters have a significant impact on the performance of the prediction, namely the input window length Δ_{input}, the output window length Δ_{pred} and the overlapping δ. In Table 3, various combinations of these three hyperparameters are tested, with mean RMSE and standard deviation obtained in each case.

The following observations can be formulated:

- In all cases, the use of overlapping δ improves the performance of RUL prediction, all other hyper-parameters being equal. This therefore confirms the statement made in Section 2.3.1.
- The more $\Delta_{input} > \Delta_{pred}$, the better the prediction.
- It is essential to avoid the case where $\Delta_{input} < \Delta_{pred}$, i.e. trying to predict a longer sequence than the one provided as input, which would be equivalent to extrapolating excessively on the basis of limited information.

However, it is important to note that the selection of these parameters is not solely based on optimizing prediction performance but is primarily dictated by the system's constraints. Specifically, the choice of Δ_{input} depends on the system's initial lifetime, where having no RUL prediction may be acceptable. Additionally, consideration should be given to the MTTF of the system when selecting Δ_{input} and Δ_{pred}. If the MTTF is small, shorter windows should be used, whereas for a large MTTF, input and output windows can be extended. Moreover, it is feasible to train multiple prediction models to enable predictions based on short input windows (Δ_{input} small) initially when few HI values are known, transitioning to a model utilizing longer input windows (Δ_{input} larger) as more data becomes available.

Some other hyperparameters that concern the model itself. For the stacked-LSTM model that is used here (see Fig. 5), the key parameters are the number of LSTM layers (denoted as L) and the size of the

Table 3
Impact assessment of hyper-parameters Δ_{input}, Δ_{output} and δ on RUL prediction (all other hyper-parameters being equal to the values indicated in Table 2).

Input window length Δ_{input}	Output window length Δ_{output}	Overlapping δ	Mean RMSE	STD
8 h	3 h	0	14.89	2.4
8 h	**3 h**	**1.5 h**	**8.44**	**1.65**
8 h	8 h	0	36.46	5.04
8 h	8 h	4 h	17.18	2.97
3 h	3 h	0	26.2	13.13
3 h	3 h	1.5 h	19.02	9.66
3 h	8 h	0	41.2	12.3
3 h	8 h	1.5 h	35.68	12.06

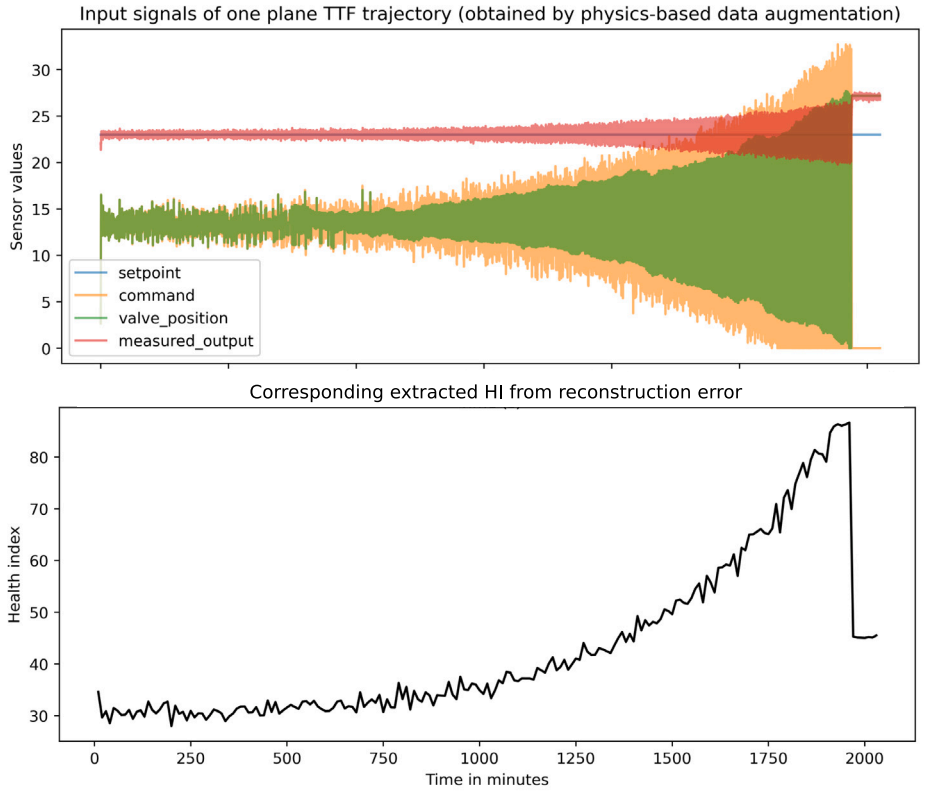


Fig. 22. HI trajectory obtained from the reconstruction error of the signals collected on one example TTF trajectory. Input multivariate sensor data is plotted on the top graph, corresponding HI trajectory is plotted on the bottom graph. Once the EOL moment has passed, the system no longer operates and, as a result, the reconstruction error (i.e. the HI) falls back.

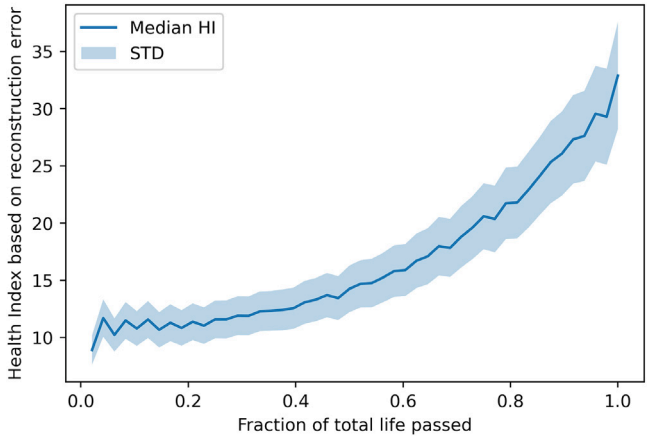


Fig. 23. Median HI trajectory along with standard deviation based on the reconstruction error approach for a test set of ten TTF trajectories, w.r.t. fraction of total life passed.

hidden state (denoted as $\left|h_{t_k}\right|$). Choosing these hyperparameters is formally equivalent to the question of model selection among a family of model [86]. Hyperparameter tuning can be achieved by manual grid searching, by selecting hyper-parameters quoted in research papers [75] or by implementing optimization algorithms [87]. The results of a manual grid tuning of the number of layers and the size of the hidden state are shown in Table 4.

3.3.1. Reliability-based assessment of the AI approach

The RUL prediction AI-based approaches are assessed against traditional reliability approach, as described in Section 2.3.4.

Table 4
Manual grid optimization of the number of layers and hidden state size for the stacked-LSTM model used for RUL prediction (all other hyper-parameters being equal to the values indicated in Table 2). Best combination performance is in bold.

Number of layers L	Hidden state size $\left\|h_{t_k}\right\|$	Mean RMSE	Standard deviation
	60	10.46	3.40
1	120	10.15	3.11
	240	9.31	2.73
	60	8.92	2.02
3	120	9.50	2.25
	240	**8.44**	**1.65**
	60	10.83	2.87
6	120	9.30	2.37
	240	10.63	2.70

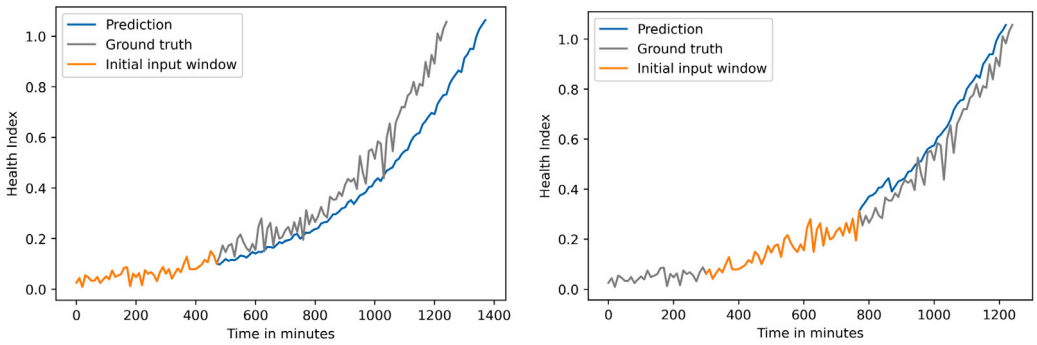
Overview of the proposed assessment approach. To that end, a set of $N = 20$ TTF trajectories is generated using the degraded data augmentation method described in Section 3.1.2, with respect to a predefined Weibull failure probability distribution $F_{ref}(t)$ (cf. Eq. (18)).

Through this comprehensive process, RUL predictions are derived from raw sensor data. Subsequently, a list of N EOL instants (i.e., scalars) can be reconstructed and utilized to identify the parameters of the corresponding approximated Weibull distribution. This enables comparison between the reference distribution $F_{ref}(t)$ and the approximated distribution obtained post-prediction of the RUL, $\widehat{F_{pred}}(t)$. The entire reliability-based assessment is outlined in Algorithm 4.

The failure distribution $\widehat{F_{pred}}(t)$ is estimated using the median rank estimation method [79].

$$\widehat{F_{HI}}(t_i) = \frac{n_i - 0.3}{N + 0.4} \tag{34}$$

with n_i as the cumulative number of faulty systems, N as the total number of systems (here $N = 10$) and t_i as the EOL time. Then, applying



(a) HI prediction or one TTF test trajectory, starting from $t_{k=480}$ minutes.

(b) HI prediction or one TTF test trajectory, starting from $t_{k=800}$ minutes

Fig. 24. Two HI predictions starting at different initial timesteps.

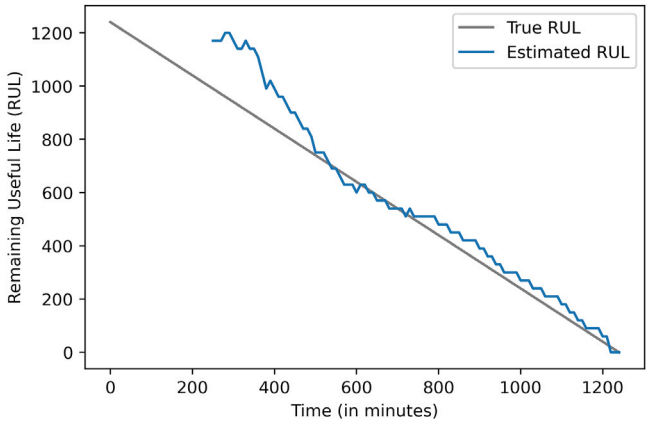


Fig. 25. Complete RUL estimation trajectory for one test flight.

Algorithm 4: Overall reliability-based assessment process.

Input: A set $\mathbf{X}$ of N multi-dimensional time series $\mathbf{X}_i = \left\{ X_{t_k} \right\}_{k=1}^{T_i}$ of duration T_i following a Weibull distribution F_{ref}, a selected distance d.

Output: A set of N EOL moments $\widehat{T}_i(t_{k=EOL-d})$ predicted starting from timestep $t_{k=EOL-d}$ (i.e. at a distance d of the true EOL) following a distribution $\widehat{F_{RUL}}(t_{k=EOL-d})$

for $i = 1$ *to* N **do**

  $\mathbf{HI}_i \leftarrow m_{\Theta_m}(\mathbf{X}_i)$ where m_{Θ_m} represents the HI extraction model parameterized by Θ_m;

  $\mathbf{RUL}_i \leftarrow n_{\Theta_n}(\mathbf{HI}_i)$ where n_{Θ_n} represents the RUL prediction model parameterized by Θ_n;

  $\widehat{T}_i(t_{k=EOL-d}) \leftarrow \mathrm{RUL}_i(t_{k=EOL-d}) + k$ where $\mathrm{RUL}_i(t_k)$ is the k-th RUL value selected from trajectory $\mathbf{RUL}_i$;

end

$\widehat{F_{RUL}}(t_{k=EOL-d}) \leftarrow$ empirical CDF from all $\widehat{T}_i(t_{k=EOL-d})$;

a double log transformation, a linear model is obtained. Then, a least squares based linear regression is applied to obtain the estimate of parameters α and β of the Weibull distribution.

$$\log(-\log(1-\widehat{F}(t))) = \widehat{\beta}\log(t) - \widehat{\beta}\log(\widehat{\alpha}) \tag{35}$$

Then, a least squares based linear regression can be applied to obtain estimated parameters $\widehat{\alpha}$ and $\widehat{\beta}$ of the corresponding Weibull distribution.

Kolmogorov–Smirnov test. The Kolmogorov–Smirnov test [80] compares the observed distribution of a statistical sample with a theoretical distribution. The hypothesis tested is:
"The CDF of the random variable T, denoted as F, is equal to the reference CDF F_{ref}" with a risk of error α.

The aim is thus to obtain an estimate of the CDF from the observed sample, denoted as $\widehat{F}$, and then compare it with the CDF of the theoretical distribution, denoted as F_{ref}. If the tested hypothesis is true, then the empirical CDF $\widehat{F}$ of the sample must be close to the CDF F_{ref}, because both mean and variance of the empirical CDF are unbiased estimators.

The empirical CDF is defined by:

$$\widehat{F}(t) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{T_i<t} \tag{36}$$

The fit of the CDF $\widehat{F}$ to F_{ref} is quantified using a specific distance known as the Kolmogorov–Smirnov distance, which is intuitively the maximum absolute difference between $\widehat{F}$ and F_{ref} for each T_i according to the formula [88]:

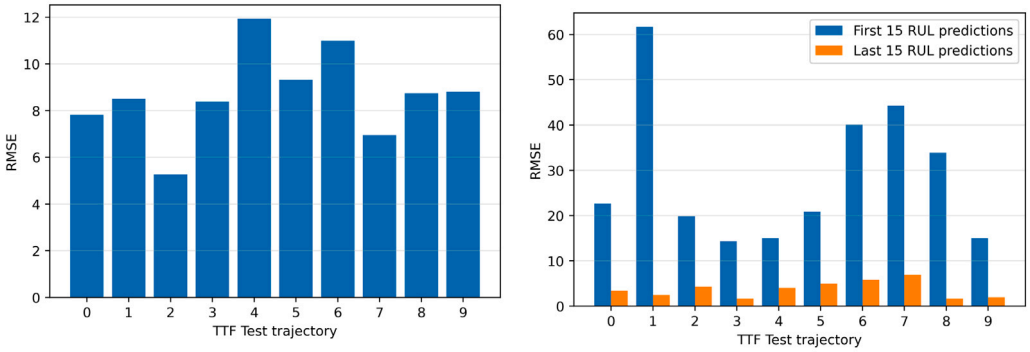
$$D_{KS}(F_{ref}, \widehat{F}) = \max_{i=1,\ldots,N} \left\{ \left| F(T_i) - \frac{i}{N} \right|, \left| F(T_i) - \frac{i-1}{N} \right| \right\} \tag{37}$$

Finally, the distance $D_{KS}(F_{ref}, \widehat{F})$ is compared with a critical value $D_{\alpha,N}$ provided by the Kolmogorov–Smirnov table [89]. The hypothesis is accepted with an α risk of error if $D_{KS}(F_{ref}, \widehat{F}) < D_{\alpha,n}$.

Experimental results. From the set 20 TTF trajectories generated using the degraded data augmentation method with respect to the predefined Weibull failure probability distribution $F_{ref}(t)$ ($\alpha = 27$ and $\beta = 3.5$), 10 are used to train the RUL prediction model and 10 are available as a test set.

As detailed in Section 3.3, the accuracy of RUL prediction improves as the EOL approaches and is less reliable at the system's initial stages. Consequently, assessments are conducted at various distances from the true EOL (d in Algorithm 4). Specifically, from all available RUL trajectories, EOL values T_i are retrieved from the RUL trajectories at $t_{k=EOL-15h}$, $t_{k=EOL-10h}$, $t_{k=EOL-5h}$, and $t_{k=EOL-3h}$, representing different points in time for RUL assessment. Fig. 27 illustrates the four time points when RUL values are retrieved for testing from a complete RUL trajectory. It is important to note that the number N of available T_i in each observed sample may vary because not all trajectories have the same duration, and thus, some may be too short to achieve the desired prediction distance d from the EOL. The results of the four tests conducted are presented in Table 5.

In each case, the hypothesis is accepted, indicating that the predictions are consistent with the reference distribution. Furthermore, it is observed that the distance $D_{KS}(\widehat{F}_{pred}, F_{ref})$ decreases as the EOL



(a) RMSE for each complete TTF test trajectory.

(b) RMSE for each complete TTF test trajectory, computed on the first and last 15 predictions, demonstrating the clear improvement in the RUL prediction when approaching the EOL.

Fig. 26. Evaluation of RUL prediction performance using RMSE.

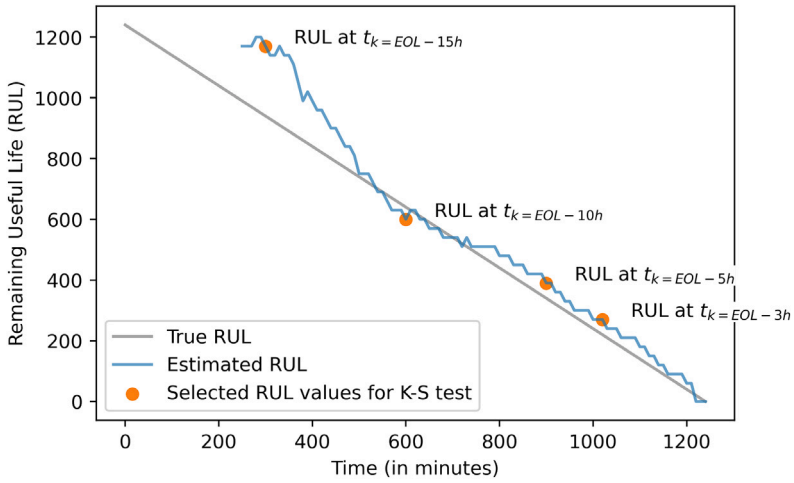


Fig. 27. Collecting RUL values at four different time moments to perform several Kolmogorov–Smirnov tests.

Table 5
Parameters of the Weibull distribution(Reference vs. estimated from predicted RUL values) at different moments, along with Kolmogorov–Smirnov test result.

	N TTF available	$\hat{\eta}$	$\hat{\beta}$	K-S Distance $D_{KS}(\hat{F}_{pred}, F_{ref})$	Critical limit $D_{0.05,N}$	Hypothesis accepted? (i.e. $D_{KS}(\hat{F}_{pred}, F_{ref}) < D_{0.05,N}$)
Reference Weibull distribution for data generation	20	27	3.5	–	–	–
Distribution $\hat{F}_{pred}$ at $t_{k=EOL-3h}$	10	24.8	5	0.125	0.410	Yes
Distribution $\hat{F}_{pred}$ at $t_{k=EOL-5h}$	8	22.8	3.9	0.24	0.457	Yes
Distribution $\hat{F}_{pred}$ at $t_{k=EOL-10h}$	5	23.1	3.9	0.4	0.565	Yes
Distribution $\hat{F}_{pred}$ at $t_{k=EOL-15h}$	4	22.1	3.4	0.5	0.624	Yes

approaches, indicating an increase in the prediction robustness. This reaffirms the findings observed with RMSE in Section 3.3. This trend is critical because as a system approaches its EOL, monitoring its SOH becomes increasingly essential for optimal maintenance. Consequently, the proposed reliability-based assessment strategy strengthens the AI-based approach. Despite performing prognostics at the system level using historical data from individual equipment and employing black-box approaches, it is demonstrated that by aggregating all individual predictions, established reliability laws are adhered to, enhancing the credibility of AI-based models.

4. Conclusion

In this paper, a comprehensive framework is proposed for RUL prediction in realistic industrial scenarios, eliminating the need for prior degradation data. A novel physics-informed data augmentation approach addresses this limitation, ensuring interpretability and consistency with general physics based laws. To that end, firstly, a hybrid data augmentation procedure is proposed, allowing integration of system knowledge (available *a priori*) and physics of failure into the training data. Secondly, an unsupervised Health Index (HI) extraction approach

is developed, followed by long-term HI prediction, enabling efficient prediction of RUL without labeled data. Finally, a reliability-based assessment is conducted to validate the proposed approach. The proposed approach is tested on an aeronautical system in real industrial context but remains applicable in a broad sense to various dynamical systems vulnerable to functional degradation(s). The unsupervised HI extraction using autoencoder reconstruction error does not require labeled data, offering excellent generalization capabilities solely based on nominal system behavior. Moreover, reliability-based assessment demonstrates that although autoencoder and LSTM networks are black-box models, the HI and RUL trajectories closely align with real failure probability distributions, boosting confidence in AI-based approaches. Future research will explore integrating uncertainty quantification in variational framework as well as development of advanced deep neural network structures for superior performance in the unsupervised context.

CRediT authorship contribution statement

Martin Hervé de Beaulieu: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation. **Mayank Shekhar Jha:** Writing – review & editing, Supervision, Methodology, Investigation. **Hugues Garnier:** Writing – review & editing, Supervision, Project administration. **Farid Cerbah:** Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

This research was carried out under the research collaboration between research lab CRAN (UMR CNRS 7039) and Dassault Aviation, France, with the support of the DGAC.

References

[1] Lee J, Wu F, Zhao W, Ghaffari M, Liao L, Siegel D. Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications. Mech Syst Signal Process 2014;42(1–2):314–34.
[2] Zhuang L, Xu A, Wang X-L. A prognostic driven predictive maintenance framework based on Bayesian deep learning. Reliab Eng Syst Saf 2023;234:109181.
[3] Lei Y, Li N, Guo L, Li N, Yan T, Lin J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. Mech Syst Signal Process 2018;104:799–834.
[4] Li Y, Wang Z, Li F, Li Y, Zhang X, Shi H, Dong L, Ren W. An ensembled remaining useful life prediction method with data fusion and stage division. Reliab Eng Syst Saf 2024;242:109804.
[5] Sikorska JZ, Hodkiewicz M, Ma L. Prognostic modelling options for remaining useful life estimation by industry. Mech Syst Signal Process 2011;25(5):1803–36.
[6] Jha MS. Particle filter based hybrid prognostics for health monitoring of uncertain systems in bond graph framework. Mech Syst Signal Process 2016;75:301–29.
[7] Jha MS, Bressel M, Ould-Bouamama B, Dauphin-Tanguy G. Particle filter based hybrid prognostics of proton exchange membrane fuel cell in bond graph framework. Comput Chem Eng 2016;95:216–30.
[8] Lei Y. Intelligent fault diagnosis and remaining useful life prediction of rotating machinery. Butterworth-Heinemann; 2016.
[9] Si X-S, Wang W, Hu C-H, Zhou D-H. Remaining useful life estimation–a review on the statistical data driven approaches. European J Oper Res 2011;213(1):1–14.
[10] Atamuradov V, Medjaher K, Dersin P, Lamoureux B, Zerhouni N. Prognostics and health management for maintenance practitioners-review, implementation and tools evaluation. Int J Progn Health Manag 2017;8(3):1–31.
[11] Hu C, Xing Y, Du D, Si X, Zhang J. Remaining useful life estimation for two-phase nonlinear degradation processes. Reliab Eng Syst Saf 2023;230:108945.
[12] Fallahi F, Bakir I, Yildirim M, Ye Z. A chance-constrained optimization framework for wind farms to manage fleet-level availability in condition based maintenance and operations. Renew Sustain Energy Rev 2022;168:112789.
[13] Mitici M, de Pater I, Barros A, Zeng Z. Dynamic predictive maintenance for multiple components using data-driven probabilistic RUL prognostics: The case of turbofan engines. Reliab Eng Syst Saf 2023;234:109199.
[14] Cubillo A, Perinpanayagam S, Esperon-Miguez M. A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery. Adv Mech Eng 2016;8(8):1687814016664660.
[15] Li X, Shao H, Lu S, Xiang J, Cai B. Highly efficient fault diagnosis of rotating machinery under time-varying speeds using LSISMM and small infrared thermal images. IEEE Trans Syst Man Cybern: Syst 2022;52(12):7328–40.
[16] Zhu X, Zhao X, Yao J, Deng W, Shao H, Liu Z. Adaptive multiscale convolution manifold embedding networks for intelligent fault diagnosis of servo motor-cylindrical rolling bearing under variable working conditions. IEEE/ASME Trans Mechatronics 2023.
[17] Zhao X, Zhu X, Liu J, Hu Y, Gao T, Zhao L, Yao J, Liu Z. Model-assisted multi-source fusion hypergraph convolutional neural networks for intelligent few-shot fault diagnosis to electro-hydrostatic actuator. Inf Fusion 2024;104:102186.
[18] Zhao X, Yao J, Deng W, Jia M, Liu Z. Normalized conditional variational auto-encoder with adaptive focal loss for imbalanced fault diagnosis of bearing-rotor system. Mech Syst Signal Process 2022;170:108826.
[19] Zhao X, Yao J, Deng W, Ding P, Ding Y, Jia M, Liu Z. Intelligent fault diagnosis of gearbox under variable working conditions with adaptive intraclass and interclass convolutional neural network. IEEE Trans Neural Netw Learn Syst 2022;34(9):6339–53.
[20] Fink O, Wang Q, Svensen M, Dersin P, Lee W-J, Ducoffe M. Potential, challenges and future directions for deep learning in prognostics and health management applications. Eng Appl Artif Intell 2020;92:103678.
[21] Li X, Ding Q, Sun J-Q. Remaining useful life estimation in prognostics using deep convolution neural networks. Reliab Eng Syst Saf 2018;172:1–11.
[22] Xiong J, Zhou J, Ma Y, Zhang F, Lin C. Adaptive deep learning-based remaining useful life prediction framework for systems with multiple failure patterns. Reliab Eng Syst Saf 2023;235:109244.
[23] Wu Y, Yuan M, Dong S, Lin L, Liu Y. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. Neurocomputing 2018;275:167–79.
[24] Wang J, Wen G, Yang S, Liu Y. Remaining useful life estimation in prognostics using deep bidirectional LSTM neural network. In: 2018 prognostics and system health management conference. Chongqing, China: IEEE; 2018, p. 1037–42.
[25] Yuan M, Wu Y, Lin L. Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network. In: 2016 IEEE international conference on aircraft utility systems. AUS, Beijing, China: IEEE; 2016, p. 135–40.
[26] Zheng S, Ristovski K, Farahat A, Gupta C. Long short-term memory network for remaining useful life estimation. In: 2017 IEEE international conference on prognostics and health management. ICPHM, Dallas, Texas, USA: IEEE; 2017, p. 88–95.
[27] Xu Z, Bashir M, Liu Q, Miao Z, Wang X, Wang J, Ekere N. A novel health indicator for intelligent prediction of rolling bearing remaining useful life based on unsupervised learning model. Comput Ind Eng 2023;176:108999.
[28] Dong S, Xiao J, Hu X, Fang N, Liu L, Yao J. Deep transfer learning based on Bi-LSTM and attention for remaining useful life prediction of rolling bearing. Reliab Eng Syst Saf 2023;230:108914.
[29] Wenqiang J, Jian C, Yi C. Remaining useful life prediction for mechanical equipment based on temporal convolutional network. In: 2019 14th IEEE international conference on electronic measurement & instruments. ICEMI, Changsha, China: IEEE; 2019, p. 1192–9.
[30] Liu C, Zhang L, Wu C. Direct remaining useful life prediction for rolling bearing using temporal convolutional networks. In: 2019 IEEE symposium series on computational intelligence. SSCI, Xiamen, China: IEEE; 2019, p. 2965–71.
[31] Zhou D, Li Z, Zhu J, Zhang H, Hou L. State of health monitoring and remaining useful life prediction of lithium-ion batteries based on temporal convolutional network. IEEE Access 2020;8:53307–20.
[32] Michau G, Fink O. Domain adaptation for one-class classification: monitoring the health of critical systems under limited information. 2019, arXiv preprint arXiv:1907.09204.
[33] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707.
[34] Li Y, Chen Y, Hu Z, Zhang H. Remaining useful life prediction of aero-engine enabled by fusing knowledge and deep learning models. Reliab Eng Syst Saf 2023;229:108869.
[35] Xu Y, Kohtz S, Boakye J, Gardoni P, Wang P. Physics-informed machine learning for reliability and systems safety applications: State of the art and challenges. Reliab Eng Syst Saf 2022;108900.
[36] Von Rueden L, Mayer S, Beckh K, Georgiev B, Giesselbach S, Heese R, Kirsch B, Pfrommer J, Pick A, Ramamurthy R, et al. Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. IEEE Trans Knowl Data Eng 2021;35(1):614–33.

[37] Sobie C, Freitas C, Nicolai M. Simulation-driven machine learning: Bearing fault classification. Mech Syst Signal Process 2018;99:403–19.
[38] Pan Y, Jing Y, Wu T, Kong X. Knowledge-based data augmentation of small samples for oil condition prediction. Reliab Eng Syst Saf 2022;217:108114.
[39] Lu Y, Rajora M, Zou P, Liang SY. Physics-embedded machine learning: case study with electrochemical micro-machining. Machines 2017;5(1):4.
[40] Ma Z, Liao H, Gao J, Nie S, Geng Y. Physics-informed machine learning for degradation modeling of an electro-hydrostatic actuator system. Reliab Eng Syst Saf 2023;229:108898.
[41] Guo R, Huang T, Li M, Zhang H, Eldar YC. Physics-embedded machine learning for electromagnetic data imaging: Examining three types of data-driven imaging methods. IEEE Signal Process Mag 2023;40(2):18–31.
[42] Von Hahn T, Mechefske CK. Knowledge informed machine learning using a Weibull-based loss function. 2022, arXiv preprint arXiv:2201.01769.
[43] Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA. Generative adversarial networks: An overview. IEEE Signal Process Mag 2018;35(1):53–65.
[44] Wang K, Gou C, Duan Y, Lin Y, Zheng X, Wang F-Y. Generative adversarial networks: introduction and outlook. IEEE/CAA J Autom Sin 2017;4(4):588–98.
[45] Yoon J, Jarrett D, Van der Schaar M. Time-series generative adversarial networks. Adv Neural Inf Process Syst 2019;32.
[46] McCool JI. Using the Weibull distribution: reliability, modeling, and inference. Vol. 950, John Wiley & Sons; 2012.
[47] Ma M, Chen X, Wang S, Liu Y, Li W. Bearing degradation assessment based on weibull distribution and deep belief network. In: 2016 international symposium on flexible automation. ISFA, IEEE; 2016, p. 382–5.
[48] Mathur N, Asirvadam VS, Aziz AA, Ibrahim R. Control valve life cycle prediction and effect of valve stiction in reliability analysis. In: 2020 16th IEEE international colloquium on signal processing & its applications. CSPA, IEEE; 2020, p. 46–51.
[49] Finkelstein M. Failure rate modelling for reliability and risk. Springer Science & Business Media; 2008.
[50] Ljung L. System identification. Springer; 1998.
[51] Garnier H, Wang L. Identification of continuous-time models from sampled data.. Identification of continuous-time models from sampled data. London: Springer-Verlag; 2008.
[52] Ljung L. System identification: Theory for the user, 2nd edition. Prentice Hall, Upper Saddle River; 1999.
[53] Pintelon R, Schoukens J. System identification: a frequency domain approach. Piscataway (USA): IEEE Press; 2001.
[54] Young PC. Recursive estimation and time-series analysis. An introduction for the student and practitioner. Berlin: Springer-Verlag; 2011.
[55] Garnier H. Direct continuous-time approaches to system identification. Overview and benefits for practical applications. Eur J Control 2015;24:50–62.
[56] Thuillier J, Jha MS, Le Martelot S, Theilliol D. Prognostics aware control design for extended remaining useful life: Application to liquid propellant reusable rocket engine. Int J Progn Health Manag 2024;15(1).
[57] Van den Hof P. Closed-loop issues in system identification. Annu Rev Control 1998;22:173–86.
[58] Forssell U, Ljung L. Closed-loop identification revisited. Automatica 1999;35(7):1215–41.
[59] Gilson M, Van den Hof P. Instrumental variable methods for closed-loop system identification. Automatica 2005;41(2):241–9.
[60] Box GE, MacGregor JF. Parameter estimation with closed-loop operating data. Technometrics 1976;18(4):371–80.
[61] Söderström T, Ljung L, Gustavsson I. A theoretical analysis of recursive identification methods. Automatica 1978;14(3):231–44.
[62] MacGregor J, Fogal D. Closed-loop identification: the role of the noise model and prefilters. J Process Control 1995;5(3):163–71.
[63] Eykhoff P. Identification theory: Practical implications and limitations. Measurement 1984;2(2):75–85.
[64] Huang B, Kadali R. Dynamic modeling, predictive control and performance monitoring: a data-driven subspace approach. Springer; 2008.
[65] Horch A. Condition monitoring of control loops (Ph.D. thesis), Signaler, sensorer och system; 2000.
[66] Arrhenius S. On the reaction velocity of the inversion of cane sugar by acids. In: Selected readings in chemical kinetics. Elsevier; 1967, p. 31–5.
[67] Pugno N, Ciavarella M, Cornetti P, Carpinteri A. A generalized Paris' law for fatigue crack growth. J Mech Phys Solids 2006;54(7):1333–49.
[68] Heckert NA, Filliben JJ, Croarkin CM, Hembree B, Guthrie WF, Tobias P, Prinz J. In: Heckert NA, Filliben JJ, Croarkin CM, Hembree B, William F,, editors. Handbook 151: NIST/SEMATECH e-handbook of statistical methods. 2002.
[69] Saxena A, Goebel K, Simon D, Eklund N. Damage propagation modeling for aircraft engine run-to-failure simulation. In: International conference on prognostics and health management. Denver, Colorado, USA: IEEE; 2008, p. 1–9.
[70] Hu Y, Palmé T, Fink O. Deep health indicator extraction: A method based on auto-encoders and extreme learning machines. In: PHM 2016, denver, USA. 2016, p. 446–52.
[71] Gensler A, Henze J, Sick B, Raabe N. Deep learning for solar power forecasting—An approach using AutoEncoder and LSTM neural networks. In: 2016 IEEE international conference on systems, man, and cybernetics. SMC, IEEE; 2016, p. 002858–65.
[72] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE Trans Pattern Anal Mach Intell 2013;35(8):1798–828.
[73] Bank D, Koenigstein N, Giryes R. Autoencoders. 2020, arXiv preprint arXiv:2003.05991.
[74] Zhang H, Zhang Q, Shao S, Niu T, Yang X. Attention-based LSTM network for rotatory machine remaining useful life prediction. IEEE Access 2020;8:132188–99.
[75] Brownlee J. Long short-term memory networks with python: develop sequence prediction models with deep learning. Machine Learning Mastery; 2017.
[76] Herve de Beaulieu M, Jha M, Garnier H, Cerbah F. Long range health index estimation based unsupervised RUL prediction using encoder-decoders. In: 11th IFAC symposium on fault detection, supervision and safety for technical processes. Vol. 55, Pafos, Cyprus: IFAC; 2022, p. 718–23.
[77] Hervé de Beaulieu M, Jha MS, Garnier H, Cerbah F. Unsupervised prognostics based on deep virtual health index prediction. In: PHM society European conference. Vol. 7, 2022, p. 193–9.
[78] Graves A, Fernández S, Schmidhuber J. Multi-dimensional recurrent neural networks. In: International conference on artificial neural networks. Springer; 2007, p. 549–58.
[79] Rinne H. The Weibull distribution: a handbook. CRC Press; 2008.
[80] Stephens MA. Introduction to Kolmogorov (1933) on the empirical determination of a distribution. In: Breakthroughs in statistics: methodology and distribution. Springer; 1992, p. 93–105.
[81] Garnier H, Gilson M, Muller H, Chen F. A new graphical user interface for the CONTSID toolbox for matlab. IFAC-PapersOnLine 2021;54(7):397–402.
[82] Bohlin TP. Interactive system identification: Prospects and pitfalls. Springer Berlin, Heidelberg; 1991.
[83] Ljung L, Glad T. Modeling and identification of dynamic systems. Studentlitteratur; 2016.
[84] Siraskar R. Reinforcement learning for control of valves. Mach Learn Appl 2021;4:100030.
[85] He Q, Wang J. Valve stiction modeling: First-principles vs data-drive approaches. In: Proceedings of the 2010 American control conference. IEEE; 2010, p. 3777–82.
[86] Bengio Y. Practical recommendations for gradient-based training of deep architectures. In: Neural networks: tricks of the trade: second edition. Springer; 2012, p. 437–78.
[87] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016, http://www.deeplearningbook.org.
[88] Massey Jr FJ. The Kolmogorov-Smirnov test for goodness of fit. J Am Stat Assoc 1951;46(253):68–78.
[89] Berger VW, Zhou Y. Kolmogorov–Smirnov test: Overview. In: Wiley statsref: Statistics reference online. Wiley Online Library; 2014.