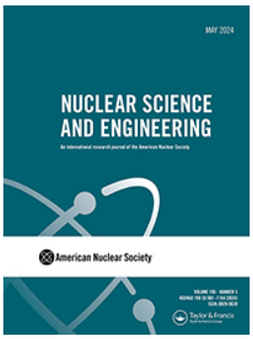


Nuclear Science and Engineering

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/unse20

Combining Similarity Measures and Left-Right Hidden Markov Models for Prognostics of Items Subjected to Perfect and Imperfect Maintenance

Mattia Zanotelli, J. Wesley Hines & Jamie B. Coble

To cite this article: Mattia Zanotelli, J. Wesley Hines & Jamie B. Coble (07 Feb 2024): Combining Similarity Measures and Left-Right Hidden Markov Models for Prognostics of Items Subjected to Perfect and Imperfect Maintenance, Nuclear Science and Engineering, DOI: 10.1080/00295639.2024.2303165

To link to this article: https://doi.org/10.1080/00295639.2024.2303165

Published online: 07 Feb 2024.

NUCLEAR SCIENCE AND ENGINEERING

DOI: https://doi.org/10.1080/00295639.2024.2303165

Combining Similarity Measures and Left-Right Hidden Markov Models for Prognostics of Items Subjected to Perfect and Imperfect Maintenance

Mattia Zanotelli,* J. Wesley Hines, and Jamie B. Coble

University of Tennessee, Department of Nuclear Engineering, 863 Neyland Drive, Knoxville, Tennessee 37916

Received September 8, 2023
Accepted for Publication December 25, 2023

Abstract — *In the nuclear industry, high system reliability requirements are essential since in-service failure can result in undesirable consequences in terms of costs or safety. However, the current approach to maintaining systems and components is costly and known to involve overly conservative periodic maintenance activities. It is, therefore, appropriate to develop monitoring, detection, and predictive tools to enable operators to create optimal maintenance strategies. These strategies can vary from the substitution of an item to its repair, intending to avoid unexpected consequences. The repair can restore the item's functionality to an as-good-as-new condition (perfect repair) or sometimes can keep some accumulated degradation and change the item's degradation rate (imperfect or partial repair). Current techniques and models that can perform prognostics with extraordinary accuracy are often designed on the assumption that following maintenance, the item is restored to an as-good-as-new condition. When these models are used to predict items that follow imperfect repairs, the predictions are likely to be inaccurate. Therefore, the present work focuses on the condition-based prognostics of items, considering and handling the criticalities that arise after the items undergo different kinds of repairs. The proposed solution involves a data-driven framework that employs Left-Right Gaussian Hidden Markov Models (LR-GHMMs). These models can intrinsically manage accumulated degradation. The idea is to train different LR-GHMMs, each specific to a degradation path, and then combine them to cover possible intermediate paths. The effectiveness of the approach is tested in two case studies. In the first one, we consider simple artificial sequences that are useful to explain the method's capabilities. In the second case study, we consider semi-simulated nuclear data describing the degradation transients of a condenser that undergoes fouling. The framework is trained with data collected from items that start without accumulated degradation. The test data represent either new items or items that undergo imperfect repairs. The results demonstrate an attractive elasticity of the framework in adapting to nonstandard degradation behaviors. In addition, the applications provide interpretable and highly accurate outputs.*

Keywords — *Condition-based prognostics, imperfect maintenance, Hidden Markov Models, nuclear power plants, condenser fouling.*

Note — *Some figures may be in color only in the electronic version.*

*E-mail: mzanotel@vols.utk.edu

I. INTRODUCTION

Prognostics and health management (PHM) is a process to assess, predict, and maintain the health condition of components, systems, and structures using monitoring data. Accurately predicting possible failures within an industrial system prevents corrective

maintenance, reduces system downtime, and helps transition the maintenance strategy from preventive (i.e., scheduled) maintenance to predictive (condition-based) maintenance.[1] This can directly improve operational efficiency and reduce life cycle costs. In view of the high impact and costs usually associated with nuclear power plant (NPP) failures, methods that can identify and predict such events have long been investigated.[2] Given the capability of these methods in supporting and sustaining improvements in the operation and maintenance of NPPs, an appropriate pipeline design that embeds optimal PHM strategies is critical to ensuring the competitiveness of the nuclear industry in the global energy market.

The key modules of PHM are detection, diagnostics, and prognostics. In the first module, data are collected from the system to detect abnormal deviations of signals from normal behavior. In the diagnostics phase, faults are isolated to one or more specific components of the system, and the causes of the faults are subsequently identified. Last, prognostics can predict the system's remaining useful life (RUL). All the modules provide helpful information for operation and maintenance.[3]

In this work, the attention is focused on the prognostic phase that aims at predicting the RUL of items and systems. Nowadays, we have at our disposal many sophisticated techniques and models that can perform prognostics with extraordinary accuracy.[4] However, most of them are designed on the assumption that following maintenance, the item's condition is restored to an as-good-as-new condition. This is true when a maintenance strategy involves the replacement (also known as perfect maintenance operation) of the failed item. Nevertheless, given the high costs associated with replacements, it is often convenient to repair an item instead, as a balance of economic performance and system reliability. A repair usually results in imperfect maintenance and implies that the system's condition after maintenance is somewhere between bad-as-old and good-as-new.[5] More specifically, a repair can remove only part of the degradation that accumulated before and potentially change the degradation evolution of the item. In general, we will use the term "degradation paths" to indicate distinct behaviors. Instead, when a degradation path depends on a constant degradation rate over time (e.g., it can be modeled through a linear or an exponential function), we will identify different behaviors using the term "degradation rates."

To clarify, Fig. 1 highlights the differences between perfect and imperfect maintenance operations, showing possible different degradation restorations and rates after an imperfect repair. Although repairs benefit plant economics, they also create difficulties when predicting the RUL. It has been proven that general models are often unsuitable for analyzing data from items that begin operating with some accumulated degradation just after a maintenance operation, resulting in inaccurate predictions.[3,6]

A previous study addressed the aforementioned criticalities using a gamma process to model the degradation and combined it with a Markov renewal process to simulate imperfect repairs and inform the predictions.[7] The authors of Ref. [8] investigated the impacts of imperfect maintenance action to predict items' deterioration correctly and to develop an adaptive maintenance policy. In Ref. [9], the Wiener process with jumps is employed to model the degradation path of a deteriorating system, where the jump parts are used to characterize the influence of maintenance activities on the system degradation. Another study addressed the aforementioned criticalities by developing a maintenance-dependent framework for detecting anomalies, decoupling faults, doing prognosis analysis on each fault, and reinitializing the critical model

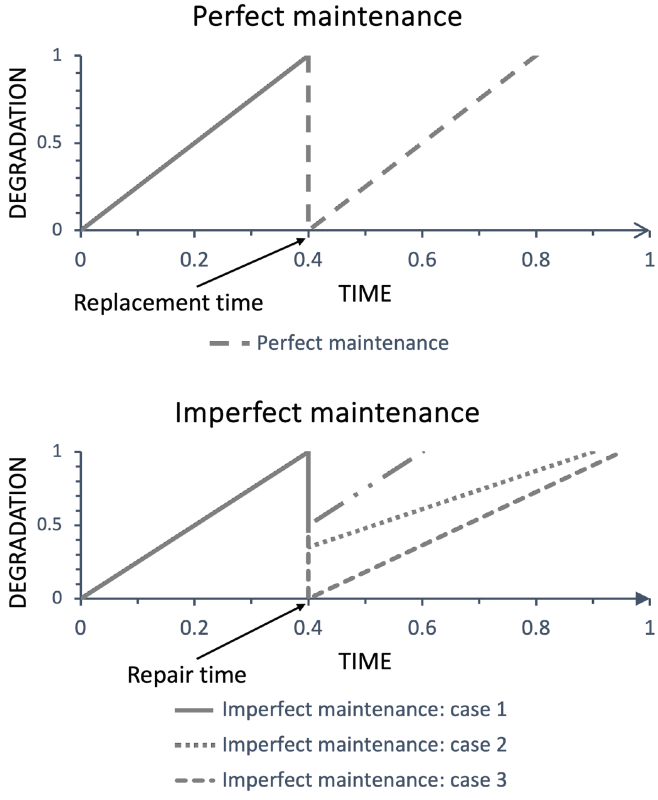


Fig. 1. Perfect and imperfect maintenance. The different kinds of dashed lines indicate the functional behavior of the degradation after the execution of different kinds of maintenance.

parameters depending on specified repair techniques.[10] In that work, the focus was on the prognostic phase for which the authors implemented a so-called Auxiliary Particle Filter Prognostic Model (APFPM). An alternative approach to handle the criticalities of the imperfect repairs can be the forage of a model or the design of a framework that (1) is trained only on degradation sequences of items that start with no accumulated degradation, (2) accurately predicts the RUL of items that follow a perfect maintenance operation, and (3) accurately predicts the RUL of items that start with accumulated degradation and possibly follow new degradation paths when it is informed about the kind of imperfect repair. The present study focuses on this last solution, developing a data-driven framework that employs Left-Right Gaussian Hidden Markov Models (LR-GHMMs),[11] which can intrinsically manage accumulated degradation. It is worth mentioning that the approach is designed to be trained on univariate prognostic parameters.[12] The overall idea is to train different models, each specific to a degradation path, and then combine them to have a coverage of possible intermediate paths. The combination is managed by proposing an aggregation method that finds an interpretable average of the single-model RUL estimates.

The LR-GHMM framework is demonstrated with two case studies. First, artificial sequences are generated to simulate the behavior of prognostic parameter evolutions of items that start without accumulated degradation. Each sequence has a noisy exponential behavior with a different degradation rate, representing distinct degradation paths. Then, the framework is tested on other artificial sequences simulating intermediate degradation paths and random initial accumulated degradation. The second case study considers semi-simulated NPP circulating water system (CWS) data. Condenser tube fouling is introduced into the CWS to simulate both evolutions of initially brand-new items and items that start with accumulated degradation and an unknown degradation path. Once the faults are detected, prognostic parameters are generated to predict failures. Results show that the framework provides a reliable way for NPP prognostics when perfect and imperfect repairs are involved.

II. METHODOLOGY

This section focuses on the development of the proposed data-driven framework. The theory behind LR-GHMMs is first explained, mentioning the algorithm to train a model on an observation sequence and focusing on the procedure to predict the RUL of a test sequence using the model. Then, the aggregation approach to average the predictions is contextualized and explained. A visual representation of the framework is depicted in Fig. 2.

II.A. The Left-Right Gaussian Hidden Markov Model

This section delineates the structure of a general Hidden Markov Model (HMM)[11] and mentions the general training procedure given an observation sequence. Then, the specifics and constraints to make the model Gaussian and Left-Right are briefly explained. The HMM is an extension of the Markov chain[13] in which the states of the process are not directly measurable and can be inferred only through the analysis of an observed signal. The relationship between the signal values and the hidden states can be modeled through conditioned probabilistic functions. In other words, the observed signal is a random variable whose probability density function (pdf) depends on the current state of the underlying Markov chain.

A standard HMM consists of a set of elements that are defined as follows:

1. $S = S_1, ..., S_N$ = a set of hidden states.
2. $A = a_{11}, ..., a_{ij}, ..., a_{NN}$ = a transition probability (in our case, defined per unit time) matrix from any state i to any state j.
3. $O = o_1, ..., o_t, ..., o_T$ = a sequence of T observations (e.g., a set of prognostic parameter values equally spaced in time).
4. $B = \{b_i(o_t)\}$ = a set of N observation likelihoods (or emission probability densities) that can be discrete or continuous.
5. $\pi = \pi_1, \pi_2, ..., \pi_N$ = an initial probability distribution of the hidden states.

The HMMs inherit the main property of the Markov chains, known as the Markov assumption. If the hidden state at time t is indicated as q_t, the assumption states that $P(q_t|q_1, ..., q_{t-1}) = P(q_t|q_{t-1})$; i.e., the probability of a particular state depends only on the state at the previous time step. Another assumption specific to HMMs is the output independence: $P(o_t|q_1, ..., q_t, o_1, ..., o_t) = P(o_t|q_t)$; i.e., the probability of an output observation o_t depends only on the state q_t that produced the observation and not on any other states or any other observation. Figure 3 depicts the structure of a hidden Markov process, showing the relationships between the hidden states and the observation variable at any time step and the transition probabilities between the states at a time instant and the

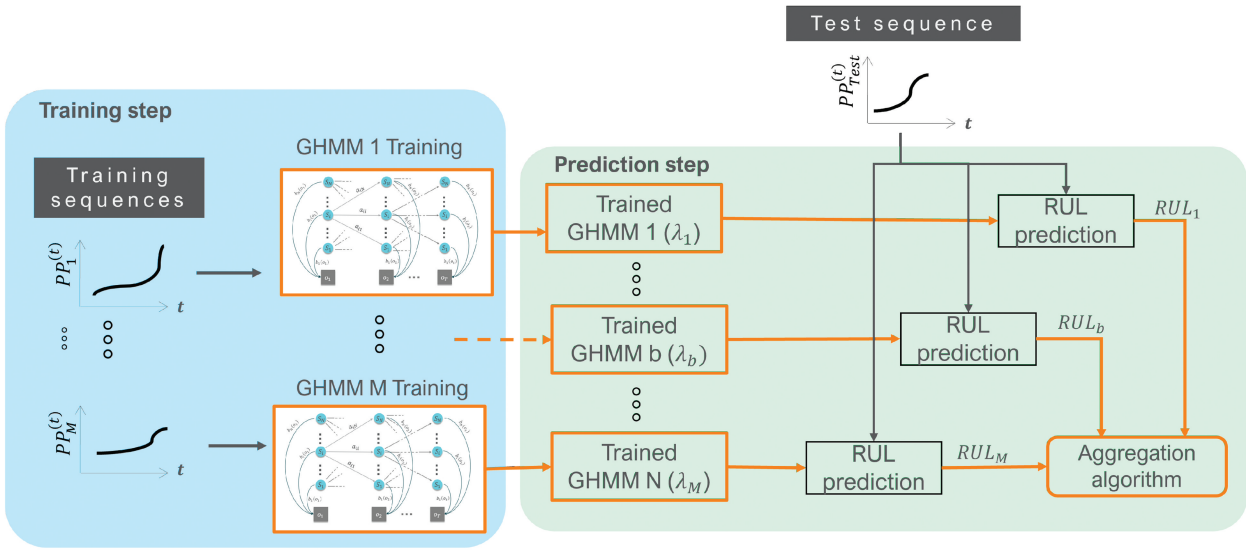


Fig. 2. LR-GHMM framework for prognostics. $PP_1, PP_2, \ldots$ are prognostic parameters extracted combining signals from time series of items used to train the models. PP_{Test} is the prognostic parameter extracted from a general test item.

next one. A training process for an HMM aims to set all the free parameters related to the elements shown in the figure to make the model as consistent as possible with the sequence of observations that we want to describe.

The training step is performed employing the Baum-Welch (B-W) algorithm,[11] which trains a single model λ on an observation sequence O_{Train}. The algorithm is a special case of the Expectation-Maximization algorithm[14] and requires the initialization of A, B, and π. Statistically combining model parameters and the observation sequence, the procedure obtains an initial estimate of A and B; then, these estimates are used to compute better estimates, and so on, iteratively. Once the iterations terminate, the B-W algorithm provides an estimate of the A and B that maximize the likelihood $P(O|\lambda)$. A limitation of this approach is that it finds a parameter

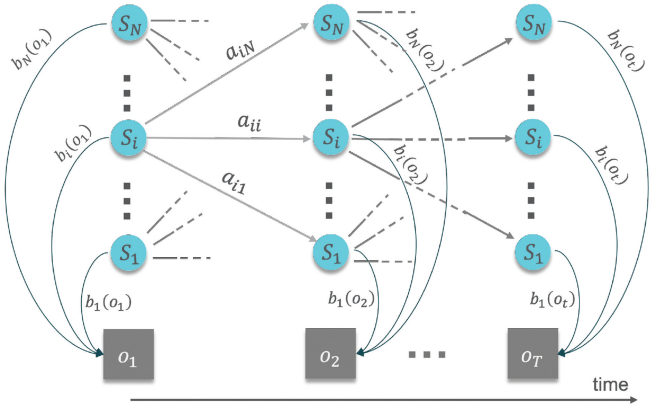


Fig. 3. General structure of a hidden Markov process.

set for λ that usually corresponds to a local likelihood maximum. It has been observed that the algorithm is sensitive to all the parameters' initialization[15]; thus, the global maximum is reached only if the initialization is adequately performed. The initialization technique used for this work will be discussed later.

It was mentioned that the selected model for the present application is a LR-GHMM. The choice of a Left-Right model is related to the nature of degradation processes where the components cannot recover from a bad health condition to a better one over time (unless a repair is performed). When describing the degradation states of an item through the hidden states of a Markov process, the mentioned physical behavior is translated setting $a_{ij} = 0$ for $j < i$. In other words, the index of states numerically proceeds from left to right. Furthermore, it was decided to impose additional constraints on the state transition coefficients to make sure that significant changes in state indices do not occur: $a_{ij} = 0$ for $j > i + 1$. That is, from one time step to the next, the system can remain at the same degradation state or progress one state.

Moreover, since the analyzed industrial prognostic signal belongs to a continuous domain, the observation likelihoods are chosen to be continuous and Gaussian. Although there are representations of likelihoods in the literature that are more comprehensive and accurate,[11] simple Gaussian pdf's are good enough for industrial prognostic applications involving a univariate parameter. The new reestimation formulas for the coefficients of the densities can be found in Ref. [14] while the reestimation formula for a_{ij} remains unchanged.

Figure 4 shows a simplified example of a prognostic parameter sequence. On the right side, a visual representation of a trained HMM is presented to show how we expect it to statistically model the sequence. The consecutive states are associated with observation likelihoods with increasing expected value (increasing degradation) as a direct consequence of the parameter's monotonicity and the HMM's left-right constraint. Each colored region indicates a subdomain of parameter values where the observational probability of being in a specific state is high. Thus, in this case, it is clear that the first observation tells that we are starting from state 1 while the second and the third suggest that we moved to state 2. The transition probability values should reflect this behavior.

II.B. RUL Prediction Using a LR-GHMM

A procedure that is specific to the LR-GHMMs is introduced here to predict the RUL of a system. This procedure relies only on the Left-Right properties and not the kind of likelihood function so that it can be potentially applied to any LR-HMM. Given a new observation sequence, $O_{test} = o_1, ..., o_t, ..., o_{T_{test}}$, and a trained LR-GHMM model λ, a discrete probability distribution for RUL(t) is calculated estimating the remaining number of time steps first to reach the failure state q_N from the current time t:

$$\begin{aligned} r_i^\tau &\triangleq P(\text{RUL}(t) = \tau | q_t = S_i, i < N) \\ &= P(q_{t+\tau} = S_N, q_{t+\tau-1} \neq S_N, ..., q_{t+1} \neq S_N | q_t \\ &= S_i, i < N) \quad . \end{aligned}$$

Given the model's constraints, the probability can be recursively calculated with the following procedure:

When $q_t = S_{N-1}$,

$$r_{N-1}^\tau = \begin{cases} a_{N-1,N} & \text{if } \tau = 1 \\ a_{N-1,N-1} \cdot r_{N-1}^{\tau-1} & \text{if } \tau \geq 2 \end{cases} . \tag{1}$$

When $q_t = S_i, i \leq N-2$,

$$r_i^\tau = \begin{cases} 0 & \text{if } \tau = 1 \\ a_{i,i} \cdot r_i^{\tau-1} + a_{i,i+1} \cdot r_{i+1}^{\tau-1} & \text{if } \tau \geq 2 \end{cases} . \tag{2}$$

We now define $\gamma_t(i) \triangleq P(q_t = S_i | o_1, .., o_t, \lambda)$ as the probability of being in the state S_i at time t, given the observation sequence and the model, which can be calculated with the well-known forward algorithm. We can then write

$$P(\text{RUL}(t) = \tau | o_1, .., o_T, \lambda) = \sum_{i=1}^{N-1} P(\text{RUL}(t) = \tau | q_t = S_i, i < N, o_1, .., o_T, \lambda) \cdot P(q_t = S_i | o_1, .., o_T, \lambda) = \sum_{i=1}^{N-1} r_i^\tau \cdot \gamma_t(i) \ . \tag{3}$$

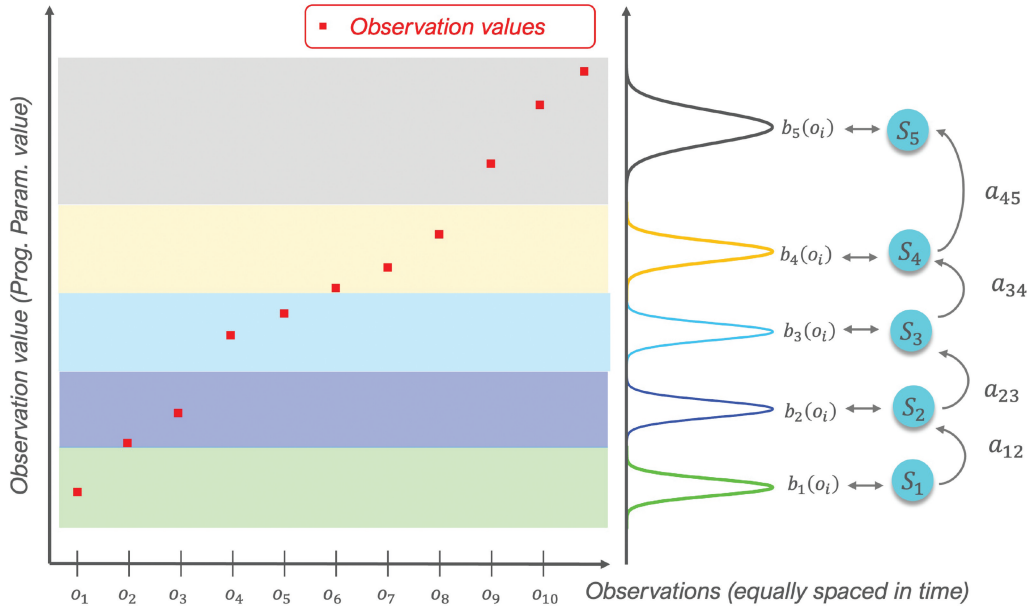


Fig. 4. Expected trained HMM form.

The point estimate for the RUL, $\overline{\mathrm{RUL}(t)}$, is then calculated by averaging the distribution.

II.C. Aggregation of LR-GHMM Predictions

It was noticed that a single LR-GHMM could not reproduce the intrinsic statistical behavior of items that degrade following different degradation rates. One model can represent a cluster of very similar items whose degradation evolves in time in a sufficiently similar way. Therefore, given a collection, $\mathbb{O} = [\boldsymbol{O_1}, ..., \boldsymbol{O_M}]$, of M training sequences representing different degradation modes, it was decided to train a separate model on each sequence. In other words, for each $\boldsymbol{O_b}$, with $b = 1, ..., M$, a LR-GHMM λ_b is trained. Notice that in the context of PHM, an appropriate observation sequence for training a prognostic model is a collection of prognostic parameter values equally spaced in time. It is reasonable to select the first value at the instant of time when the detection module detects abnormal behavior. Accordingly, we will imply that the first observation of a train or test sequence will be the one at the time of detection.

When a new test observation sequence, $\boldsymbol{O_{test}} = o_1, ..., o_t, ..., o_{T_{test}}$, becomes available, where T_{test} indicates the time at which we are interested in making a prediction, M estimates of the RUL are calculated employing the different models, $(\overline{\mathrm{RUL}(T_{test})}|\lambda_b)$ for $b = 1, ..., M$. Afterward, the estimates are combined through an aggregation method. In general, aggregation entails assigning a weight to each model's forecast and combining models' predictions using a weighted average:

$$\overline{\mathrm{RUL}} = \frac{\sum_{b=1}^{N_b} (\overline{\mathrm{RUL}(T_{test})}|\lambda_b) \cdot w_b}{\sum_{b=1}^{N_b} w_b} \quad . \tag{4}$$

Each weight w_b can be calculated by means of a membership measure between the test sequence and the model. In general, the most statistically meaningful measure for HMMs is the likelihood $P(\boldsymbol{O_{test}}|\lambda_b)$.[11] However, in this industrial application, likelihood is not a reliable membership index for test sequences since they pass through states in the same order as training sequences; the only difference lies in the transition rate, but this aspect alone does not affect the index consistently. Consequently, a new measure of membership is proposed. Given a training sequence, $\boldsymbol{O_b} = o_{b,1}, ..., o_{b,t}, ..., o_{b,T_b}$ and a test sequence, $\boldsymbol{O_{test}} = o_1, ..., o_t, ..., o_{T_{test}}$

$$w_b = \mu(\boldsymbol{O_b}, \boldsymbol{O_{test}}) = \frac{\sum_{t=1}^{T_L} \left(\frac{k}{T_{test}}\right) \cdot \ell(o_{b,t}, o_t)}{T_{test}} \quad , \tag{5}$$

where $\ell(o_{b,t}, o_t) = e^{-\frac{|o_{b,t} - o_t|^2}{2*s}}$ = Gaussian membership function with length scale s; $T_L = \min(T_b, T_{test})$. In this way, the measure gives more importance to the similarity as the time index approaches the end of the test sequence. However, it is penalized if the test sequence is longer than the training one.

Figure 5a shows how the distance $|o_{b,t} - o_t|$ is calculated at each time step in the case where the test sequence, $\boldsymbol{O_{test}} = o_1, ..., o_t, ..., o_7$, starts with observation values comparable to the training sequence's initial values. In this case, it makes sense to calculate the point-by-point similarity without translating one sequence with respect to the other. It is appropriate to use this approach when testing a brand-new item or an item that follows a perfect repair. In general, the test sequence might start with values that differ significantly from those at the beginning of the training sequence because of a less-than-perfect maintenance operation (accumulated degradation). When this happens, a translation of the time index of the training sequence is performed to find the best time window that maximizes the membership. Figure 5b shows the translation effect, where the index is shifted by three units.

The proposed similarity measure could provide unreliable outcomes when the training and the testing sequences are characterized by high noise. It was noticed that the performance is even worse when the test sequence is significantly shorter than the training one. A univariate smoothing spline technique[16] is then applied to denoise the compared two sequences. In practice, it was decided to use the interpolation spline module in the SciPy Python library. The smoothing parameter can be optimized by following the directions in the documentation.[17]

II.D. Initialization and Set of Model Parameters

The HMMs are characterized by many parameters that must be set or initialized before the training. Each operation must be separately done for each HMM of the framework.

In general, when a LR-GHMM is considered, it is a good practice to set the initial probabilities such that $\pi_1 = 1$ and $\pi_i = 0$ for $i > 1$ since the state sequence is trained to begin in state 1 (and end in state N). However, when a test sequence starts with significantly higher values than the means of the first hidden states of the model, the actual initial state is, with high probability, a state with an index higher than 1. It was tried to initialize π accordingly to this aspect, but it was noticed that a good initialization does not provide any additional

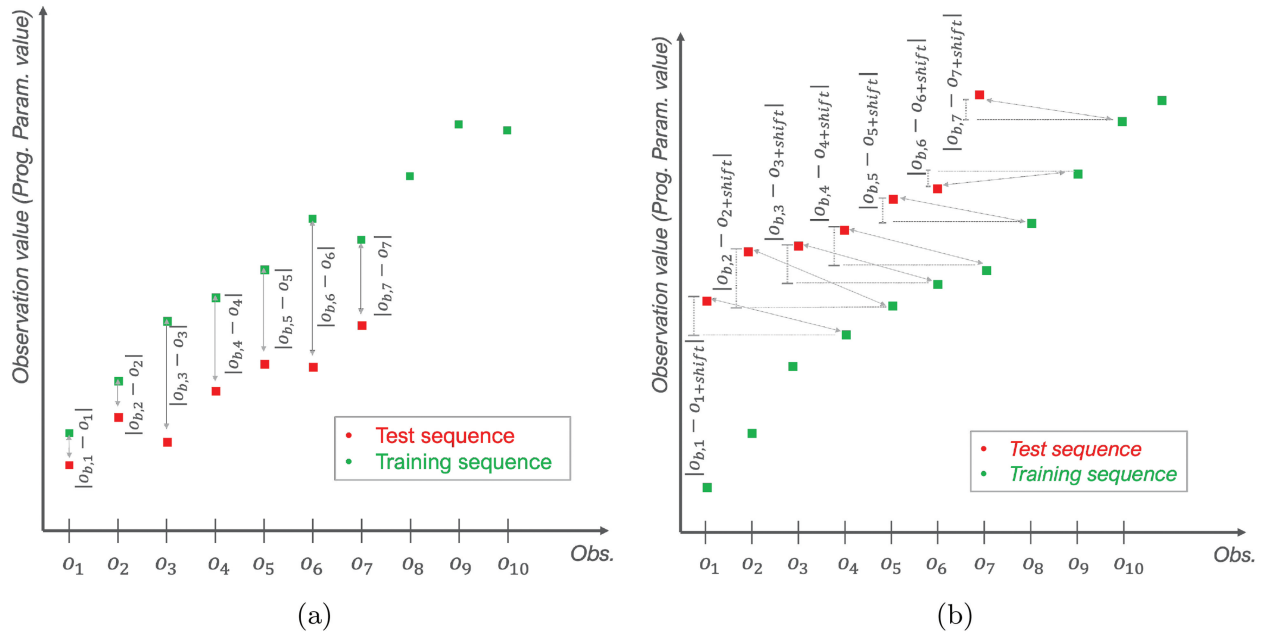


Fig. 5. (a) Calculation of point differences between two observation sequences with comparable initial values. (b) Calculation of point differences between two observation sequences using the time index translation technique.

benefit to the predictions. The nature of the HMM can explain this last phenomenon since even if the initialization of the a priori probabilities is wrong, the model needs just a few observations to "understand" the actual hidden state to update $\gamma_t(i)$, for $i = 1, ..., N$ accordingly.

The initialization of the observation likelihoods is another critical task and helps seek the global likelihood maximum with the training procedure.[11] Many sophisticated techniques have been proposed.[15] However, such sophistication is not necessary when HMMs are trained on prognostic parameters that are univariate and monotonic and have sufficiently regular functional behavior. It was then established to initialize the means and the variances of the likelihoods intuitively, as follows. The means are initialized as equally spaced values between the upper and lower bounds of the training sequence. The standard deviations were set equal to half of the distance between the means of two adjacent states.

The same argument for the transition probabilities applies to the observation likelihoods. Sophisticated methods are explained in the same articles, but in the present work, it was decided to initialize as follows:

$$a_{i,i+1} = \frac{N}{T} \quad \text{for } i = 1, ..., N-1 \quad , \tag{6}$$

where T is the number of observations of the training sequence. Provided that the number of transitions in the training sequence equals the number of hidden states, it is reasonable to initialize the transition probability as in Eq. (6).

The choice of the number of states in an HMM model is always challenging. We chose the Bayes information criterion (BIC)[18] to set a good number of states. The criterion is here adapted to the current notations as

$$\text{BIC} = k \cdot \ln(T_{train}) - 2 \cdot \ln(P(\boldsymbol{O}_{\boldsymbol{train}}|\lambda)) \quad , \tag{7}$$

where k is the number of parameters estimated by the model. In the case of a LR-GHMM, k is the sum of the number of means N, variances N, and transition probabilities $(N-1)$.[a] The criterion states that the BIC must be minimized, but in our case, the increase in the number of hidden states N continues to reduce the BIC value. It is then appropriate to stop increasing N when the rate of change is sufficiently low. The last parameter to be set before the entire framework can be used is the length scale s of the Gaussian membership function, $\ell(o_{b,t}, o_t) = e^{-\frac{|o_{b,t}-o_t|^2}{2*s}}$, employed in the procedure described in Sec. II.C. When the smoothing technique is not applied, setting $s^2 = 2 \cdot \sigma^2_{noise}$ is reasonable. When the smoothing technique is used, it was noticed that dividing the length scale $s = \sqrt{2} \cdot \sigma_{noise}$ by a factor ranging between 5 and 10 improves the membership evaluations.

[a] The number of means and variances is equal to the number of hidden states, which is N.

III. CASE STUDY I: ARTIFICIAL DATA SET

The framework was initially tested using artificial sequences that simulate the behavior of univariate prognostic parameter evolution (after the detection of a fault). It was decided to represent the degradation evolution by positive exponential functions and set the degradation path by changing the rate coefficient. Then, uncorrelated Gaussian noise is added to the function to simulate the measurement error. Since we are considering artificial data, for the sake of generality, time t and the prognostic parameter Y are considered to be dimensionless:

$$Y = e^{D*t} + \xi \quad , \tag{8}$$

where D = degradation rate; $\xi \sim N(0, \sigma_n^2)$. Afterward, the function is discretized to extract equally spaced values in time to generate an observation sequence. The training set is obtained by simulating three different sequences $[\boldsymbol{O_1}, \boldsymbol{O_2}, \boldsymbol{O_3}]$ with the following degradation rates: 0.7, 1, 1.5. The evolution of each is stopped when the exponential reaches a value of e. With this artificial choice, we want to simulate a simplified version of a realistic situation where a particular item can follow different degradation paths. The result is depicted in Fig. 6.

The test set is divided into two groups. In the first group, we collected sequences that start with no accumulated degradation, while in the second group, some sequences with initial degradation were considered. The first group is obtained by simulating five different sequences, $\boldsymbol{O}_{test1,1}, \boldsymbol{O}_{test1,2}, \ldots, \boldsymbol{O}_{test1,5}$, with the following degradation rates: 0.7, 0.9, 1, 1.3, and 1.5. Each sequence represents a single degradation path evolution considering an item whose onset of the degradation is detected at $t = 0$. Two sequences with intermediate degradation rates, 0.9 and 1.3, are used to test the coverage of the framework, although the training set does not include those rates. The sequences of the second group, $\boldsymbol{O}_{test2,1}, \boldsymbol{O}_{test2,2}, \ldots, \boldsymbol{O}_{test2,5}$, are instead simulated in the following manner. The degradation is randomly restored by picking random values between 30% and 50% of the maximum degradation, and the rates are set to have the following five distinct values: 0.7, 0.8, 1, 1.3, and 1.5.

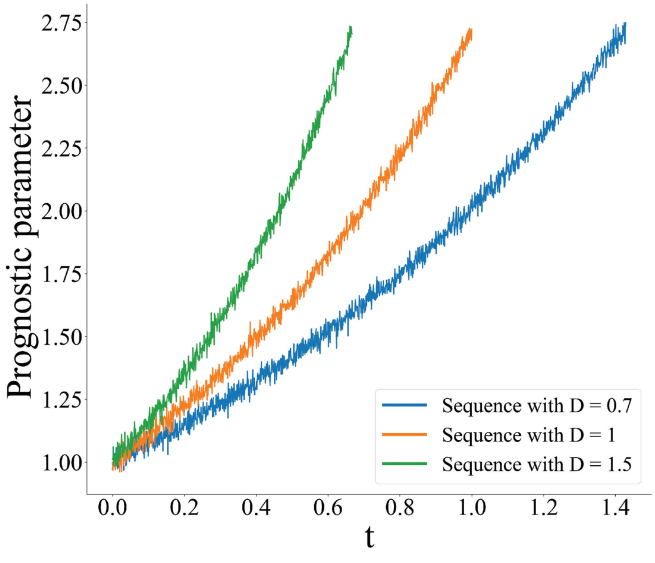


Fig. 6. Artificial dataset: training sequences.

III.A. Case Study I: Results

Having three well-distinguished training sequences $[\boldsymbol{O_1}, \boldsymbol{O_2}, \boldsymbol{O_3}]$, the framework trains three separate LR-GHMMs: $[\lambda_1, \lambda_2, \lambda_3]$. The observation likelihood of the first model is visually plotted in Fig. 7; the likelihoods of the other two models are very similar to the first one. According to the BIC criterion, the optimal number of hidden states is 18 for all the models. Figure 8 shows how the BIC values flatten for the number of states that are higher than 18. This result makes sense because of two aspects: (1) the training sequences are exponentially increasing functions that do not present actual stationary points and (2) the prognostic parameter values, not considering the noise, are defined in the same ranges. Another consequence of these aspects is the approximately constant standard deviation of the observation sequences, resulting in quite equally distributed pdf's along the domain spanned by the prognostic parameter.

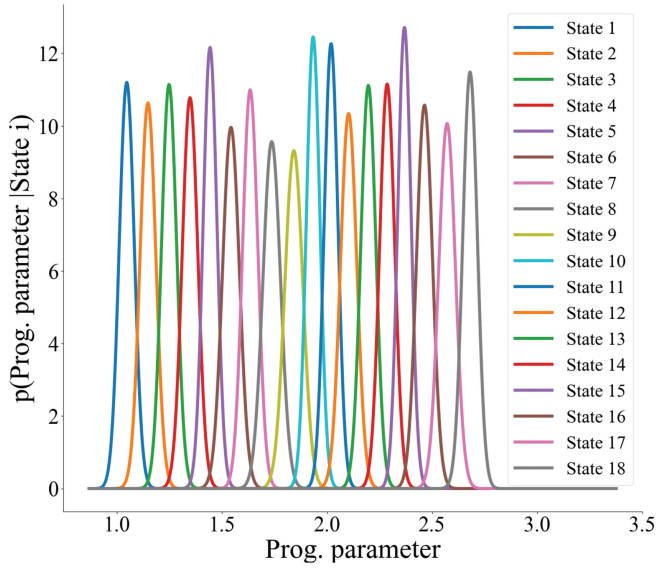


Fig. 7. Observation likelihoods of the model trained on the first training sequence ($D = 0.7$).

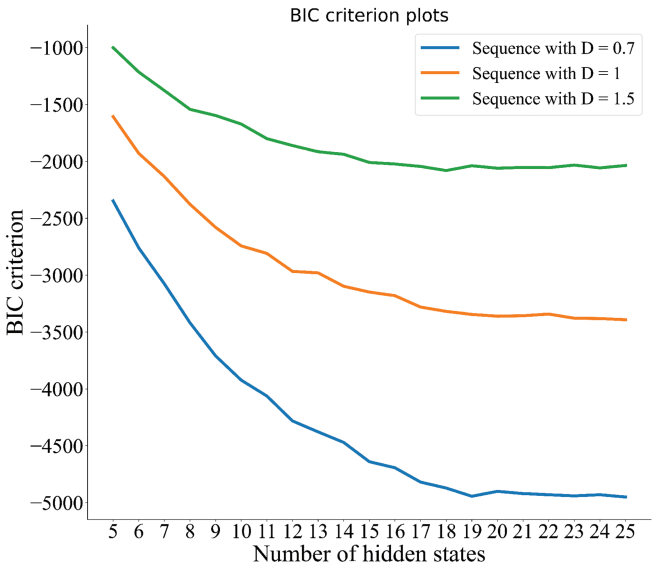


Fig. 8. Plot of the BIC criterion values for all the models.

The performance is visually inspected using the $\alpha - \Lambda$ metric.[19] Briefly, the metric evaluates whether the following condition is met:

$$(1-\alpha)\cdot \mathrm{RUL}_G(t_\Lambda) \le \overline{\mathrm{RUL}}(t_\Lambda) \le (1+\alpha)\cdot \mathrm{RUL}_G(t_\Lambda)\ , \qquad (9)$$

where α = accuracy modifier; $\Lambda \in [0, 100]$ = time index that states the progression percentage toward the end of life (EoL); $\mathrm{RUL}_G(t_\Lambda)$ = exact RUL.

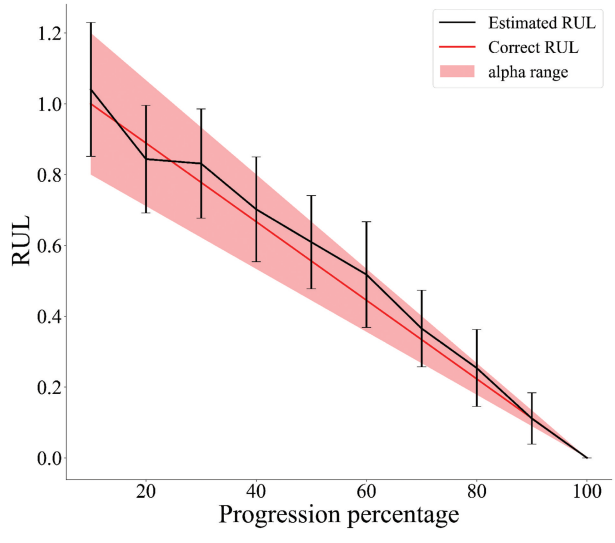
Figure 9 shows the plots of the metric evaluation on the most relevant test sequences representing the first cycle, with $\alpha = 0.2$. In these plots, each error bar's size equals the standard deviation of the respective RUL distribution. Figure 10 shows the plots of the metric evaluation on the most relevant test sequences representing the second cycles after maintenance.

To show the overall performance of the framework for most of the simulated sequences, cumulative relative accuracy (CRA), Eq. (10), is also evaluated:

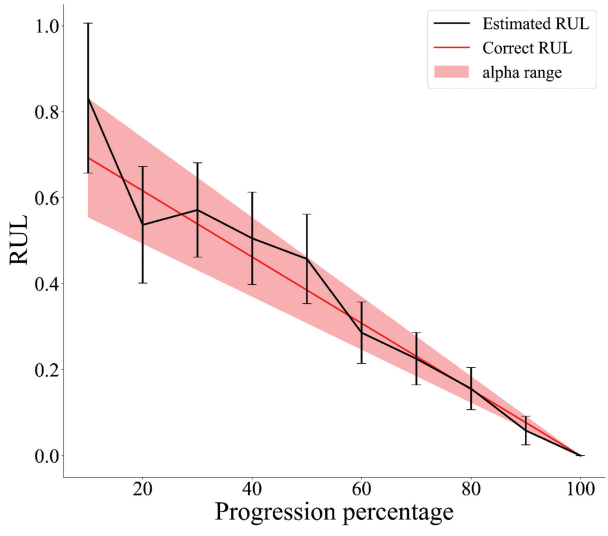
$$\mathrm{CRA} = \frac{1}{\sum_{\Lambda=10}^{100}\Lambda} \cdot \sum_{\Lambda=10}^{100}\left(1 - \frac{\overline{\mathrm{RUL}}(t_\Lambda) - \mathrm{RUL}_G(t_\Lambda)}{\mathrm{RUL}_G(t_\Lambda)}\right)\cdot \Lambda \ . \qquad (10)$$

When assessing the prognostic performance of a model, it is desirable to weigh the relative accuracies $\left(RA_\Lambda = 1 - \frac{\overline{\mathrm{RUL}}(t_\Lambda) - \mathrm{RUL}_G(t_\Lambda)}{\mathrm{RUL}_G(t_\Lambda)}\right)$ higher when closer to the EoL ($\Lambda = 100$). This is motivated by the maintenance action strategies that often rely on predictions that are closer to the EoL. Equation (10) considers this aspect by multiplying each RA_Λ by Λ and normalizing $\left(\frac{1}{\sum_{\Lambda=10}^{100}\Lambda}\right)$. The results are listed in Table I.

The visual and numerical results demonstrate the accuracy and robustness of the method, displaying good prediction capability and interpretability. Further discussion of the results can be found in Sec. V.



(a)



(b)

Fig. 9. Performance of two test sequences from the first group (no initial degradation): (a) $D = 0.9$ and (b) $D = 1.3$.

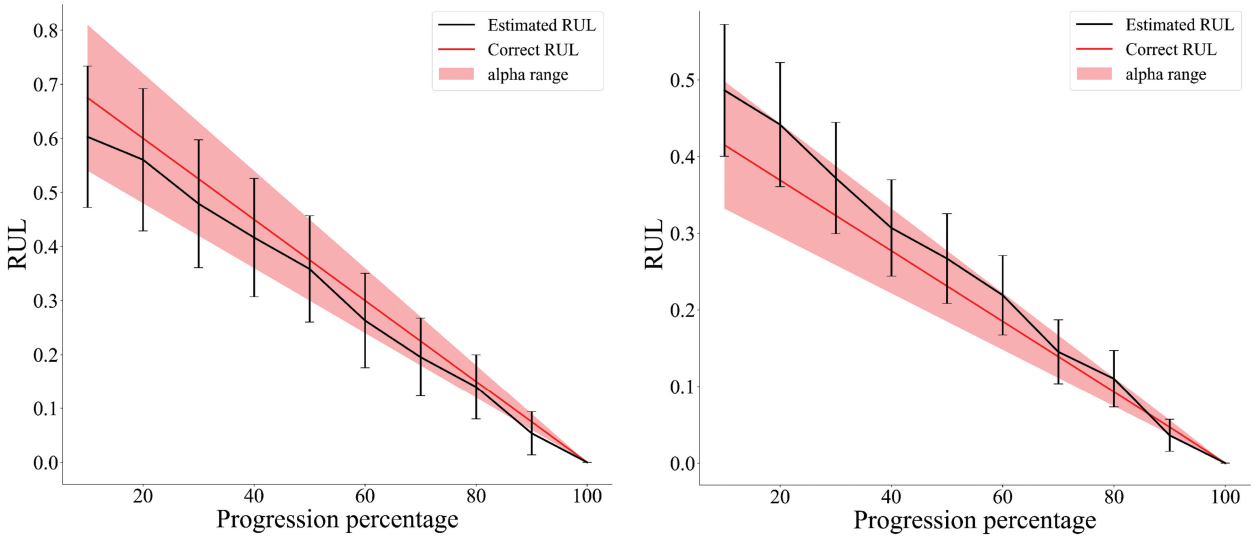


Fig. 10. Performance of two test sequences from the second group (with initial degradation): (a) $D = 0.8$ and (b) $D = 1.3$.

TABLE I

Artificial Dataset Results: Cumulative Relative Accuracy

Degradation Rate (Group 1)	CRA (Group 1)	Degradation Rate (Group 2)	CRA (Group 2)
0.7	0.94	0.7	0.93
0.9	0.90	0.8	0.89
1	0.93	1	0.93
1.3	0.91	1.3	0.88
1.5	0.97	1.5	0.90

III.B. Case Study I: A Closer Look to the Membership and the Importance of the Smoothing

The first test sequence from the second group, $\boldsymbol{O}_{\boldsymbol{test2,1}}$ with $D = 0.7$, is employed here to highlight how the membership values behave and how a RUL prediction is computed. The test sequence is truncated at 60% of this progression, and three RUL predictions are computed: $(\overline{\mathrm{RUL}(T_{test})}|\lambda_b)$ for $b = 1, 2, 3$. The memberships are then computed according to the procedure explained in Sec. II.C. The results are listed in Table II, and a visual representation of the operation is plotted in Fig. 11. The smoothed version of the test sequence is displayed three times with the optimal displacement selected by the aggregation algorithm to compare it with each training sequence. This visualization allows us to understand that the degradation path of the test is exactly the same as $\boldsymbol{O}_{\boldsymbol{1}}$ with $D = 0.7$. This aspect is coherent with the membership values, $w_1 >> w_2 >> w_3$.

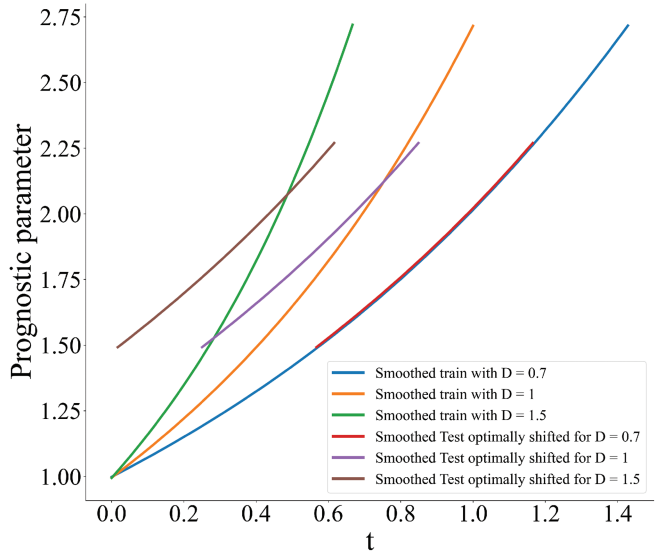


Fig. 11. Smoothed $\boldsymbol{O}_{\boldsymbol{test2,1}}$ displayed three times with the optimal displacement selected by the aggregation algorithm to compare it with each training sequence.

TABLE II

Memberships and RUL Predictions of $\boldsymbol{O}_{\boldsymbol{test2,1}}$ at 60% of Its Progression

Memberships and Single-Model Predictions		
With Smoothing	Without Smoothing	RUL Predictions
$w_1 = 0.477$ $w_2 = 0.021$ $w_3 = 0.008$	$w_1 = 0.364$ $w_2 = 0.183$ $w_3 = 0.080$	$(\overline{\mathrm{RUL}(T_{test})}\|\lambda_1) = 0.243$ $(\overline{\mathrm{RUL}(T_{test})}\|\lambda_2) = 0.158$ $(\overline{\mathrm{RUL}(T_{test})}\|\lambda_3) = 0.122$
Aggregated RUL Predictions		
With smoothing $\overline{\mathrm{RUL}(T_{test})} = 0.238$	Without smoothing $\overline{\mathrm{RUL}(T_{test})} = 0.194$	True RUL $\mathrm{RUL}_G(T_{test}) = 0.247$

A closer look at the results in Table II helps to demonstrate the importance of smoothing. Since it is clear that the test sequence degrades in exactly the same way as the first training sequence, a good membership function must assign all or most of the weight to the first model. When smoothing is not applied, the first model is still the most weighted, but there is no substantial difference with the weights of the other two as in the first case. This then leads to ascribing truthfulness to the second and third models and translating the aggregate RUL toward a lower value than the correct one (which is well estimated by the first model).

IV. CASE STUDY II: CONDENSER FOULING INSIDE A NPP

The framework is demonstrated with semi-simulated data from a NPP. A SIMULINK model Asherah[20] was tuned to approximate the Vogtle NPP and was used to provide the process data for this application. The power level and river temperature data came from a 2-year collection of Vogtle Unit 1 actual process data. Four sensors located in the CWS were selected through a correlation analysis to inspect the health state of the condenser:

1. x_1 = pump inlet temperature.
2. x_2 = reactor power.
3. x_3 = condenser temperature.
4. x_4 = condenser pressure.

The CWS module was then modified to simulate condenser tube fouling. Since tube fouling causes an increase in thermal resistivity and reduces the cross-sectional area of the flow rate, two degradation modules were accordingly inserted to simulate these behaviors in the CWS. The dynamic of the condenser fouling process is slow, with typical timescales that range between 6 and 24 months. Therefore, the simulations for collecting sensor data were performed as follows. For each degradation path, periodic steady-state responses of the Asherah model were simulated to generate a state sequence. The values of the degradation parameter for each state were set according to the elapsed time and realistic degradation functions. Patterns of sensor values were then collected from each state to generate time series. Then, the signals were analyzed through an autoassociative kernel regression that relied on data collected from a condenser in healthy conditions. The generated residuals were examined by a Sequential Probability Ratio Test for fault detection.[10] Once a fault is detected, a prognostic parameter can be computed by subtracting the residual of the pump inlet temperature from the condenser temperature residual. The parameter can then be fed into the prognostic module.

Since the condenser fouling consists of debris deposition, the degradation will likely follow a saturating trend since after a certain accumulation, the debris no longer sticks to the walls. Therefore, for this application, we select monotonic functions of the fouling factor[21] with saturating behavior.

After detecting a fault, the prognostic framework is applied to a set of observation sequences extracted from the prognostic parameter. Since the degradation rate is not constant over time, it is convenient to distinguish one sequence from the other by its length of life. The training set $[\boldsymbol{O_1}, \boldsymbol{O_2}, \boldsymbol{O_3}]$ is obtained by simulating three different sequences assigning three different lengths of life: 6 (months), 9 (months), and 12 (months). Notice that the degradation patterns were artificially generated but are consistent with what might actually happen

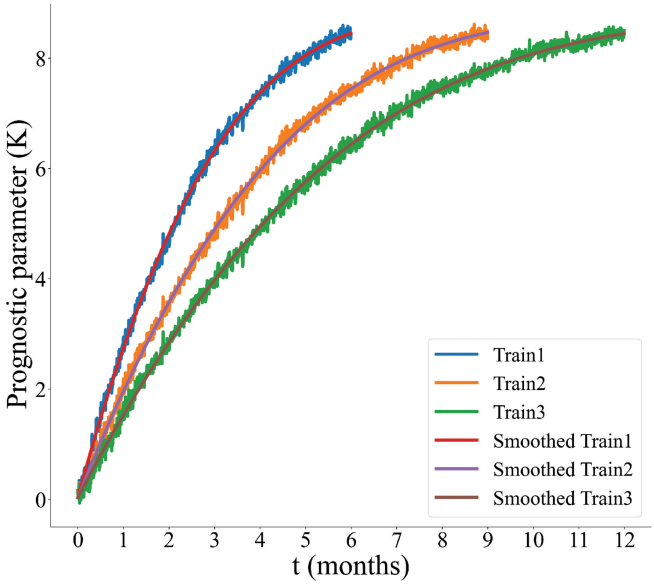


Fig. 12. Nuclear data set: training sequences.

experimentally. Condensers may have different lengths of life due to the stochasticity of the process, the operational and environmental conditions to which they are subjected, or repairs that have been made. Thus, with access to experimental data, it would not be difficult to select good sequences covering an adequately wide span of degradation paths. In particular, the requirement for a sequence to be good for the training is that no degradation accumulation is present.

In the current application, the evolution of each sequence is stopped when the fouling factor R_F[21] reaches a value of $4 \times 10^{-6}[(\mathrm{m}^2 \cdot \mathrm{K})/\mathrm{W}]$. The result is depicted in Fig. 12.

The framework is initially tested on the first cycles in which the simulated items start without degradation. The test set is obtained by simulating sequences with a length of life included in the span of the training. These sequences represent either brand-new or repaired items whose degradation has been completely restored. The second cycles are then simulated to represent imperfect maintenance operations that leave some accumulated degradation. The degradation is randomly restored by picking normally distributed values of fouling factors around 50% of the threshold value.

IV.A. RESULTS

An example of a pair of test sequences is depicted in Fig. 13a. Two examples of truncated sequences (at 10% of progression for the first cycle and 90% for the second cycle) are also plotted on the left. In contrast, the smoothed versions of these subsequences and the smoothed training sequences are plotted in Fig. 13b. The second cycle is displayed three times with the optimal displacement selected by the aggregation algorithm to compare it with each training sequence. This visualization allows us to understand that the degradation path of

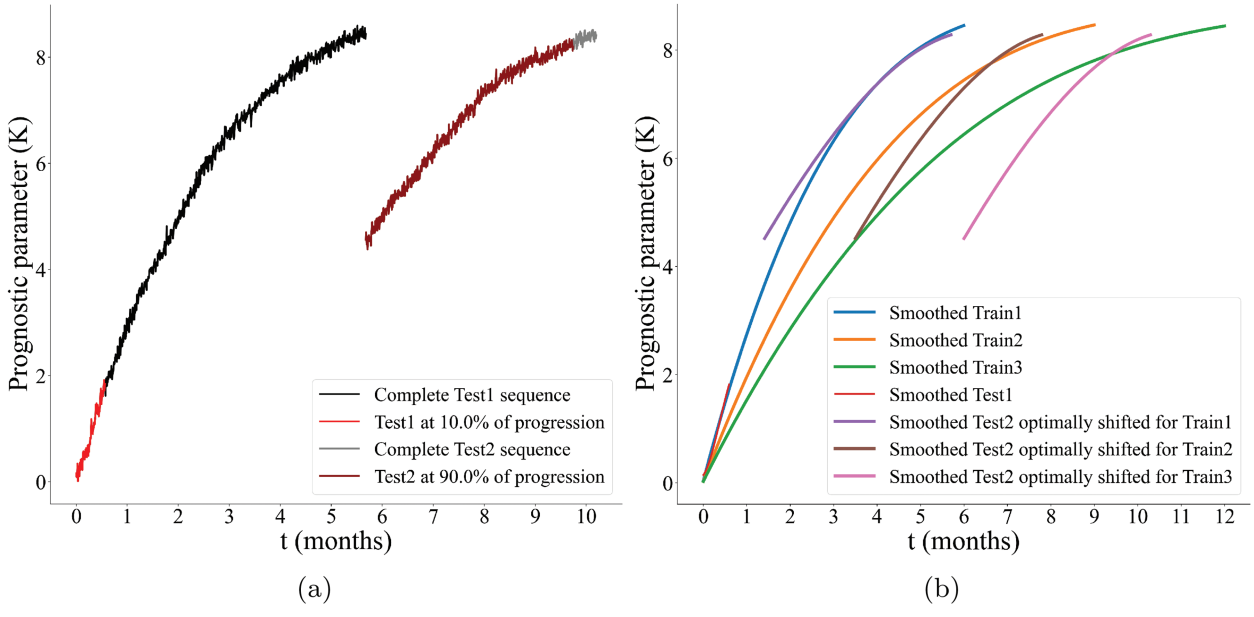


Fig. 13. (a) First (Test 1) and second (Test 2) cycle test sequences with lengths of life of 6 (months) and 4.5 (months), respectively. Two examples of sequence progressions (10% and 90%, respectively) are also plotted. (b) Smoothed versions of the training and test sequences.

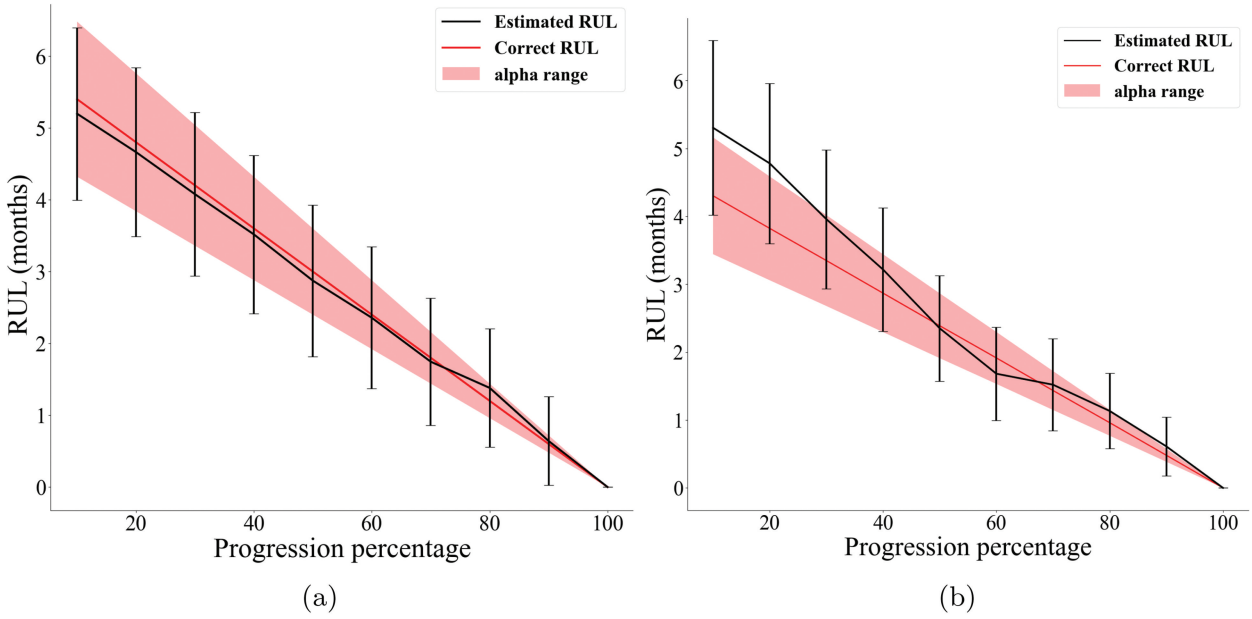


Fig. 14. (a) Performance of a first cycle test sequence [length of life 6 (months)]. (b) Performance of a second cycle test sequence [length of life 4.5 (months) with accumulated degradation around 50%].

TABLE III

Nuclear Data Set Prognostic Results: Cumulative Relative Accuracy

Case	Test Length (First Cycle)	CRA (First Cycle)	Test Length (Second Cycle)	CRA (Second Cycle)
1	6 months	0.94	4.5 months	0.87
2	9 months	0.95	5.7 months	0.90
3	7.2 months	0.93	4.8 months	0.91

the second cycle is somewhere between the first and the second training sequences' degradation paths.

The performance is visually inspected using the $\alpha - \Lambda$ metric, according to Eq. (9). Figure 14 shows the plots of the $\alpha - \Lambda$ metric of the performance of the framework on the two test sequences. Each error bar's size in these plots equals the standard deviation of the respective RUL distribution.

To show the overall performance of the framework for most of the simulated sequences, CRA, Eq. (10), is also evaluated. The results are listed in Table III.

V. DISCUSSION

The visual and numerical results of both case studies demonstrate the accuracy and robustness of the method, which handles the criticality of relevant accumulated degradation and adapts to the current prognostic parameters' functional behaviors. The framework denotes good prediction capability when tested on degradation paths for which the LR-GHMMs were trained. The performance is also good with intermediate paths but with a little delay because the membership measure must properly weigh the predictions of two models instead of one. When an item degrades according to an intermediate path and starts with accumulated degradation (second cycles), the framework intuitively needs even more time to compute an accurate outcome since the membership measure is fed with less data than the previous cases. This aspect causes a little drop in terms of CRA. However, by looking at the plots, we can state that when the progression percentage Λ is greater than 50, the prediction capability of the method is more than reliable, allowing enough time to develop an effective maintenance strategy.

It is worth noticing that all the test sequences were selected for the degradation paths (or rates) to be included in the training span. Intuitively, the current framework

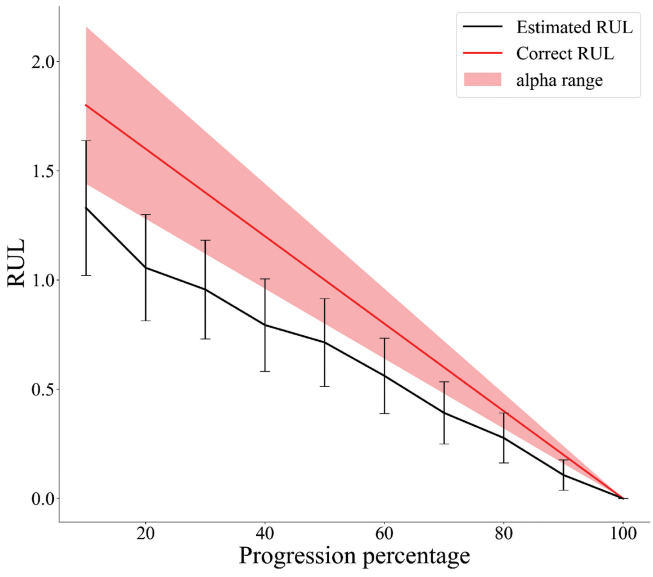


Fig. 15. Testing an artificial sequence with a degradation rate ($D = 0.5$) that is outside the training span of degradation ranges.

cannot produce accurate predictions when a test sequence degrades with a rate that is outside the span of the training. Figure 15 proves the mentioned limitation, showing the predictions of the first case study framework when an artificial sequence with $D = 0.5$ is tested. As expected, the framework underpredicts all the RULs since it was trained using items that degrade much faster. Future work should consider implementing a method that can detect when an item degrades according to a path that is not covered by training data.

VI. CONCLUSION

Maintenance-dependent processes are difficult to handle when partial repairs are considered. The eventuality of accumulated degradation and possible change of the degradation path can lead to unacceptable prediction inaccuracies when standard models are used. This study presents a strategy to deal with the critical issues that imperfect repairs involve. The strategy considers the aggregation of LR-GHMM, which handles the accumulated degradation after the repairs and guarantees high performance even when the repair changes the degradation rate of an item. In the current work, only the information about the kind of maintenance action was used to inform the predictions. Future work might consider leveraging other information stored in maintenance records to inform the aggregation algorithm through Bayesian techniques.

Applying the strategy to an artificial and realistic nuclear case study proved the approach is an effective and interpretable solution. The applications showed that LR-GHMMs can adapt to different degradation functional shapes over time, which is a great advantage over other methods. In fact, some previous works required selecting a mathematical function to model the degradation, which is not always a straightforward and reliable solution, especially when different degradation paths are involved. The main drawback of the methodology is the high standard deviation of the RUL prediction, which is difficult to reduce since it is intrinsic to the HMM method. Although previous works achieved much lower prediction standard deviations, the proposed frameworks depend on the choice of mathematical functions to model the degradation and rely on the availability of many maintenance records. These requirements can sometimes limit the applicability of these methods in some scenarios and could lead the user to prefer LR-GHMMs.

Acknowledgments

This work was supported by a Nuclear Energy University Programs grant sponsored by the U.S. Department of Energy, Office of Nuclear Energy, award number DE-NE0009278. The support is gratefully acknowledged.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by a Nuclear Energy University Programs grant sponsored by the U.S. Department of Energy, Office of Nuclear Energy, award number [DE-NE0009278].

ORCID

Mattia Zanotelli http://orcid.org/0000-0002-9484-3580

References

1. A. K. JARDINE, D. LIN, and D. BANJEVIC, "A Review on Machinery Diagnostics and Prognostics Implementing Condition-Based Maintenance," *Mech. Syst. Sig. Process.*, **20**, *7*, 1483 (2006); http://dx.doi.org/10.1016/j.ymssp.2005.09.012.
2. M. FARSI and E. ZIO, "Industry 4.0: Some Challenges and Opportunities for Reliability Engineering," *Int. J. Reliab. Risk Saf. Theory Appl.*, **2**, 23 (2019); http://dx.doi.org/10.30699/IJRRS.2.1.4.
3. J. B. COBLE et al., "A Review of Prognostics and Health Management Applications in Nuclear Power Plants," *Int. J. Progn. Health Manag.*, **6**, *3* (2015); http://dx.doi.org/10.36001/ijphm.2015.v6i3.2271.
4. X.-S. SI et al., "Remaining Useful Life Estimation—A Review on the Statistical Data Driven Approaches," *Eur. J. Oper. Res.*, **213**, *1*, 1 (2011); http://dx.doi.org/10.1016/j.ejor.2010.11.018.
5. M. BROWN and F. PROSCHAN, "Imperfect Repair," *J. Appl. Probab.*, **20**, *4*, 851 (1983); http://dx.doi.org/10.2307/3213596.
6. Z. WELZ et al., "Maintenance-Based Prognostics of Nuclear Plant Equipment for Long-Term Operation," *Nucl. Eng. Technol.*, **49**, *5*, 914 (2017); http://dx.doi.org/10.1016/j.net.2017.06.001.
7. I. CASTRO and S. MERCIER, "Performance Measures for a Deteriorating System Subject to Imperfect Maintenance and Delayed Repairs," *Proc. Inst. Mech. Eng. Part O*, **230** (2016); http://dx.doi.org/10.1177/1748006X16641789.
8. P. DO et al., "A Proactive Condition-Based Maintenance Strategy with Both Perfect and Imperfect Maintenance Actions," *Reliab. Eng. Syst. Saf.*, **133**, 22 (2015); http://dx.doi.org/10.1016/j.ress.2014.08.011.
9. Z. WANG et al., "A Simulation-Based Remaining Useful Life Prediction Method Considering the Influence of Maintenance Activities," *Proc. Prognostics and System Health Management Conf. (PHM-2014)*, Zhangjiajie, China, August 24–27, 2014; http://dx.doi.org/10.1109/PHM.2014.6988180.
10. H. XIAO et al., "Prognostics and Health Management for Maintenance-Dependent Processes," *Nucl. Technol.*, **209**, *3*, 419 (2023); http://dx.doi.org/10.1080/00295450.2022.2073949.
11. L. RABINER, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, **77**, *2*, 257 (1989); http://dx.doi.org/10.1109/5.18626.
12. J. COBLE and J. HINES, "Fusing Data Sources for Optimal Prognostic Parameter Selection," *Trans. Am. Nucl. Soc.*, **100**, 211 (2009).
13. A. FRIGESSI and B. HEIDERGOTT, *Markov Chains*, pp. 772–775, Springer Berlin Heidelberg, Berlin, Heidelberg.
14. J. A. BILMES, "A Gentle Tutorial of the EM Algorithm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," CTIT Technical Reports Series, International Computer Science Institute (1998).
15. T. LIU, J. LEMEIRE, and L. YANG, "Proper Initialization of Hidden Markov Models for Industrial Applications," *Proc. 2014 IEEE China Summit and Int. Conf. Signal and Information Processing (ChinaSIP)*, Xi'an, China, July 9–13, 2014, p. 490 (2014); http://dx.doi.org/10.1109/ChinaSIP.2014.6889291.
16. T. HASTIE, R. TIBSHIRANI, and J. FRIEDMAN, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer (2009).
17. P. VIRTANEN, R. GOMMERS, and T. OLIPHANT, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nat. Methods*, **17**, 261 (2020); http://dx.doi.org/10.1038/s41592-019-0686-2.
18. A. CHAKRABARTI and J. K. GHOSH, "AIC, BIC and Recent Advances in Model Selection," *Philosophy of Statistics; Handbook of the Philosophy of Science*, Vol. 7, p. 583, P. S. BANDYOPADHYAY and M. R. FORSTER, Eds., North-Holland, Amsterdam; http://dx.doi.org/10.1016/B978-0-444-51862-0.50018-6.
19. A. SAXENA et al., "On Applying the Prognostic Performance Metrics," *Proc. Annual Conf. Prognostics and Health Management Society*, San Diego, California, September 27–October 1, 2009, PHM Society (2009).
20. R. BUSQUIM E SILVA et al., "Advanced Method for Neutronics and System Code Coupling RELAP, PARCS, and MATLAB for Instrumentation and Control Assessment," *Ann. Nucl. Energy*, **140**, 107098 (2020); http://dx.doi.org/10.1016/j.anucene.2019.107098.
21. S. M. IBRAHIM and S. I. ATTIA, "The Influence of Condenser Cooling Seawater Fouling on the Thermal Performance of a Nuclear Power Plant," *Ann. Nucl. Energy*, **76**, 421 (2015); http://dx.doi.org/10.1016/j.anucene.2014.10.018.