Check for updates

Received: 12 November 2019 | Revised: 11 June 2020 | Accepted: 12 August 2020 | Published on: 30 September 2020

DOI: 10.1002/nav.21947

RESEARCH ARTICLE

WILEY

Health assessment and prognostics based on higher-order hidden semi-Markov models

Ying Liao[1] | **Yisha Xiang**[1] | **Min Wang**[2]

[1]Department of Industrial, Manufacturing & Systems Engineering, Texas Tech University, Lubbock, Texas

[2]Department of Management Science and Statistics, University of Texas at San Antonio, San Antonio, Texas

Correspondence
Yisha Xiang, Department of Industrial, Manufacturing & Systems Engineering, Texas Tech University, 2500 Broadway, Lubbock, TX 79409.
Email: yisha.xiang@ttu.edu

Funding information
U.S. National Science Foundation, Grant/Award Number: 1943985.

Abstract

This paper presents a new and flexible prognostics framework based on a higher-order hidden semi-Markov model (HOHSMM) for systems or components with unobservable health states and complex transition dynamics. The HOHSMM extends the basic hidden Markov model (HMM) by allowing the hidden state to depend on its more distant history and assuming generally distributed state duration. An effective Gibbs sampling algorithm is designed for statistical inference of the HOHSMM. We conduct a simulation study to evaluate the performance of the proposed HOHSMM sampler and examine the impacts of the distant-history dependency. We design a decoding algorithm to estimate the hidden health states using the learned model. Remaining useful life is predicted using a simulation approach given the decoded hidden states. The practical utility of the proposed prognostics framework is demonstrated by a case study on National Aeronautics and Space Administration (NASA) turbofan engines. We further compare the RUL prediction performance between the proposed HOHSMM and a benchmark mixture of Gaussians HMM prognostics method. The results show that the HOHSMM-based prognostics framework provides good hidden health-state assessment and RUL estimation for complex systems.

KEYWORDS

Gibbs sampling algorithm, higher-order hidden semi-Markov model, prognostics, remaining useful life

1 | INTRODUCTION

In the past decade, prognostics has emerged as one of the key enablers for industrial systems to become more reliable, operationally available, and economically maintained (Sun, Zeng, Kang, & Pecht, 2012). Prognostics technologies aim to monitor the performance of a system (or a component), assess the health status, and predict the remaining useful life (RUL). Based on the predicted future performance, informed asset management strategies can be better planned to reduce operational risks and costs. Prognostics has been used for various engineering systems, such as engines (Peel, 2008; Wang & Zhang, 2005; Zaidan, Mills, Harrison, & Fleming, 2016), batteries (Saha, Goebel, Poll, & Christophersen, 2008; Zhang & Lee, 2011), electronics (Pecht, 2009; Rouet, Minault, Diancourt, & Foucher, 2007), and bearings (Huang et al., 2007; Li et al., 1999; Li, Kurfess, & Liang, 2000; Qiu, Seth, Liang, & Zhang, 2002). However, prognostics for complex systems still remains a challenging problem. Many complex engineering systems, such as turbofan engines, heavy machinery equipment (Xiao, Fang, Liu, & Zhou, 2018), and wind turbines (Kandukuri, Klausen, Karimi, & Robbersmyr, 2016), have complex failure mechanisms, for example, the health-state transition is dependent on more distant history states. Moreover, inspection of the actual health conditions of these systems often requires disassembling which is costly and can even induce failures. Thus, condition monitoring data (eg, sensor measurements) are widely used to assess the health states that are not directly observable. Advanced techniques that can model such complex transition behaviors and assess the hidden condition are of great practical importance. In this paper, we propose a new and flexible prognostics framework based on a higher-order hidden semi-Markov model (HOHSMM) to assess the health state and estimate the RUL for complex systems.

 © 2020 Wiley Periodicals LLC

Prognostics approaches can generally be classified into two categories: model-based approach and data-driven approach. There are also hybrid models (Acuña & Orchard, 2017; Bai & Wang, 2016; Di Maio, Tsui, & Zio, 2012; Liu, Wang, Ma, Yang, & Yang, 2012) that attempt to combine the strengths of model-based and data-driven approaches by fusing these two approaches. The model-based approaches require a good understanding of system physics-of-failure mechanisms. Most of model-based approaches deal with crack, wearing, and corrosion phenomena. For example, Paris-Erdogan equation is used to model fatigue crack growth in Myötyri, Pulkkinen, and Simola (2006) and Lei et al. (2016). Daigle and Goebel (2012) develop a model-based prognostics framework for a centrifugal pump and characterize the damage processes with physics-based models (eg, erosive wear equation, friction coefficient equation). More model-based prognostics methods can be found in Chiachío, Chiachío, Sankararaman, Saxena, and Goebel (2015), Haile, Riddick, and Assefa (2016), Liao (2013), and Qian, Yan, and Gao (2017). Model-based approaches are built on the knowledge of the processes and failure mechanisms occurring in the system of concern, and the approaches provide RUL estimation based on the developed physical model. However, it is a difficult task to understand the physics of damage occurring in complex systems.

With the rapid development of sensor technologies, it has become much easier and less costly to obtain condition monitoring data, including operational and environmental loads as well as performance conditions of the monitored system (eg, temperature, vibration, pressure, voltage, current) (Cheng, Azarian, & Pecht, 2010). Advancements in modern sensor instruments have greatly facilitated data-driven prognostics. Data-driven approaches use several tools, most of which originate from artificial intelligence (AI) and statistical domains. Neural networks (Guo, Li, Jia, Lei, & Lin, 2017; Li, Ding, & Sun, 2018; Malhi, Yan, & Gao, 2011; Tian, 2012; Zheng, Ristovski, Farahat, & Gupta, 2017), neuro-fuzzy systems (Chen, Vachtsevanos, & Orchard, 2012; Wang, 2007; Wang, Golnaraghi, & Ismail, 2004), and support vector machine/regression (Benkedjouh, Medjaher, Zerhouni, & Rechak, 2015; S. Dong & Luo, 2013; Khelif et al., 2016; Liu, Vitelli, Zio, & Seraoui, 2015; Widodo & Yang, 2011) have been widely used for engineering system prognostics. However, these AI techniques are always referred to "black boxes" (Lei et al., 2018) since it is difficult to have physical explanations of the constructed network structure (eg, the number of hidden layers and the number of nodes used in each layer) and the networks' outputs. The hidden Markov model (HMM), which characterizes doubly stochastic processes, is commonly used to infer the hidden health state directly from the observed data and predict the RUL (Baruah & Chinnam, 2005; Bunks, McCarthy, & Al-Ani, 2000; Camci & Chinnam, 2010; Giantomassi et al., 2011; Tobon-Mejia, Medjaher, Zerhouni, & Tripot, 2012). Bunks et al. (2000) illustrate the applications of HMMs by using the Westland helicopter gearbox data set and show that HMMs can provide a natural framework for both health diagnostics and prognostics. Baruah and Chinnam (2005) employ HMMs to identify the health state of metal cutting tools from sensor signals and predict the RUL. Tobon-Mejia et al. (2012) develop a mixture of Gaussians HMM (MoG-HMM) to assess the current condition of a bearing and estimate its RUL. HMMs have well-constructed theoretical basis and thus are allowed for a wide range of practical applications. An added benefit of employing HMMs is the ease of model interpretation in comparison with pure "black-box" modeling methods. However, standard HMMs have two inherent limitations. One is the assumption of first-order Markovian dynamics of the hidden-state process. The other is that the state duration (ie, sojourn time) implicitly follows a geometric distribution. The first-order assumption can be restrictive as the health state of complex systems usually evolves depending on its more distant history, not just the current state. Moreover, the duration time in one state does not always follow a geometric distribution.

To provide a more adequate representation of temporal structure, the hidden semi-Markov model (HSMM) extends the basic HMM by assuming that the state duration is generally distributed. The explicit duration HSMM is widely used in health monitoring of engineering systems, which assigns an explicit distribution for duration of each state and the duration distribution is only determined by corresponding state (Dong & He, 2007; Liu, Dong, Lv, Geng, & Li, 2015; Yu, 2010). Dong and He (2007) propose an HSMM-based diagnostics and prognostics framework by adding an explicit temporary structure into HMM. Through the estimated hidden-state duration distribution and the proposed backward recursive equations, the RUL of the equipment can be predicted. There also exist other special forms of HSMM used in prognostics, which make different assumptions regarding the dependence between state transition and duration. For example, Wang, Sun, Cai, Zhang, and Saygin (2014) present a duration-dependent HSMM for prognostics, which assumes that the state transition probability depends on the previous state and its respective duration time. Liu, Zhu, and Zeng (2018) propose a new HSMM that considers the dependency between durations of adjacent degradation states to assess the health state and predict the RUL. However, in the aforementioned HSMMs, the history states' dependency, which commonly exists in complex systems, has not been taken into account when modeling state transition probability.

In this paper, we propose a new prognostics framework based on HOHSMMs for systems with unobservable health-state and complex transition dynamics. In the HOHSMM-based framework, the important features extracted from the monitoring data are used as observations and the underlying health status of the concerned system is represented in the form of hidden states, which evolve depending not only on the current state but also on its more distant history. The sojourn time in each state is generally

distributed and is assumed to follow an explicit distribution. We design an effective Gibbs sampling algorithm for model inference and conduct a simulation study to evaluate the performance of the proposed HOHSMM sampler. The impacts of the distant-history dependency are also examined in the simulation study. The learned HOHSMM is then used to assess the current health state of a functioning system in operation and predict its RUL. Decoding algorithm is developed for health-state assessment using the learned model. The RUL is estimated using a simulation approach by generating paths from the current health state to the failure state. Furthermore, we demonstrate the practical utility of the proposed prognostics framework by conducting a case study on NASA turbofan engines and comparing the RUL prediction performance of the proposed method with that of a benchmark MoG-HMM prognostics method (Tobon-Mejia et al., 2012). The main contributions of this paper are twofold.

1. Develop a new and advanced HOHSMM-based prognostics framework to assess hidden health state and predict the RUL for complex systems. The proposed HOHSMM includes the HMM and HSMM as two special cases.
2. Design effective algorithms for HOHSMM inference, hidden-state decoding, and RUL prediction. A Gibbs sampling algorithm is developed for HOHSMM inference and the simulation study shows that the designed HOHSMM sampler is effective for learning model parameters from observations. Based on the learned model, a decoding algorithm is developed for hidden health-state assessment and an RUL estimation algorithm is developed for prognostics. The case study on NASA turbofan engines shows that the HOHSMM-based prognostics framework provides satisfactory hidden health-state assessment and RUL estimation for complex systems.

The remainder of this paper is organized as follows. Section 2 provides preliminaries on the higher-order HMM (HOHMM). In Section 3, we develop an HOHSMM and design an effective sampling algorithm for statistical inference. Section 4 presents the hidden-state decoding procedure using the learned model. The RUL is predicted using a simulation approach in Section 5. We conduct a simulation study to evaluate the performance of the proposed HOHSMM sampler in Section 6. A case study on NASA turbofan engines is demonstrated in Section 7. Section 8 discusses the concluding remarks and future work.

2 | PRELIMINARIES ON HOHMM

This section provides a brief overview of the HOHMM by summarizing the main results of Sarkar and Dunson (2018) and Yang and Dunson (2016). Based on the HOHMM in Sarkar and Dunson (2018), we develop the HOHSMM.

An HOHMM consists of two processes: a hidden process $\{c_t\}$, which evolves according to a higher-order Markov chain with discrete state space, and a potentially multivariate observation process $\{y_t\}$ observed sequentially over a set of discrete time points $t = 1, 2, \ldots, T$. HOHMMs extend the idea of basic HMMs by allowing the hidden-state sequence $\{c_t\}$ to depend on its more distant past. An HOHMM of maximal order q makes the following set of conditional independence assumptions:

$$p(c_t|c_1, \ldots, c_{t-1}) = p(c_t|\boldsymbol{c}_{(t-q):(t-1)}), \tag{1}$$

$$p(y_t|c_1, \ldots, c_t, y_1, \ldots, y_{t-1}) = p(y_t|c_t). \tag{2}$$

Note that an HOHMM is said to be of maximal order q if the distribution of c_t only depends on a subset of $\{c_{t-1}, \ldots, c_{t-q}\}$. If the distribution of c_t actually varies with the values at all the previous q time points, the HOHMM is considered to be of full order q.

While the HOHMM relaxes the restrictive first-order assumption of the basic HMM, it also brings significant dimensionality challenges. For known state space $\mathscr{C} = \{1, \ldots, C\}$, the transition probabilities are now indexed by C^q different possible values of the lags $\boldsymbol{c}_{(t-q):(t-1)}$ and involve a total number of $(C-1)C^q$ parameters, which increases exponentially with the order q. To address this issue, latent allocation variables $z_{j,t}$ for $j = 1, \ldots, q$ and $t = q+1, \ldots, T$ are introduced to shrink the total number of parameters. The allocation variable $z_{j,t}$, taking values from $\{1, \ldots, k_j\}$, is the respective latent class that a particular state of c_{t-j} is allocated into. The total number of the latent classes k_j $(1 \le k_j \le C)$ then determines the inclusion of the jth lag c_{t-j}. If $k_j = 1$, it means that c_{t-j} is not an important lag for c_t. If $k_j > 1$ for all $j = 1, \ldots, q$, the HOHMM is of full order q. Based on the allocation variable $z_{j,t}$, the hidden states $\{c_t\}$ are conditionally independent as shown in Figure 1.

We denote the probability that the jth lag c_{t-j} is allocated into latent class h_j by $\pi^{(j)}_{h_j}(c_{t-j})$, that is, $\pi^{(j)}_{h_j}(c_{t-j}) = p(z_{j,t} = h_j|c_{t-j})$. Given the combination of q allocated latent classes $(h_1, \ldots, h_q)$, the state transition probability is denoted by $\lambda_{h_1,\ldots,h_q}(c_t)$ for $c_t = 1, \ldots, C$,

$$\lambda_{h_1,\ldots,h_q}(c_t) = p(c_t|z_{1,t} = h_1, \ldots, z_{q,t} = h_q). \tag{3}$$

Then the transition probability can be structured through the following hierarchical formulation

$$(c_t|z_{j,t} = h_j, j = 1, \ldots, q) \sim \text{Mult}(\{1, \ldots, C\}, \lambda_{h_1,\ldots,h_q}(1), \ldots, \lambda_{h_1,\ldots,h_q}(C)), \tag{4}$$

$$(z_{j,t}|c_{t-j}) \sim \text{Mult}(\{1, \ldots, k_j\}, \pi^{(j)}_1(c_{t-j}), \ldots, \pi^{(j)}_{k_j}(c_{t-j})). \tag{5}$$

The parameters $\lambda_{h_1,\ldots,h_q}(c_t)$ and $\pi^{(j)}_{h_j}(c_{t-j})$ are nonnegative and satisfy the constraints:

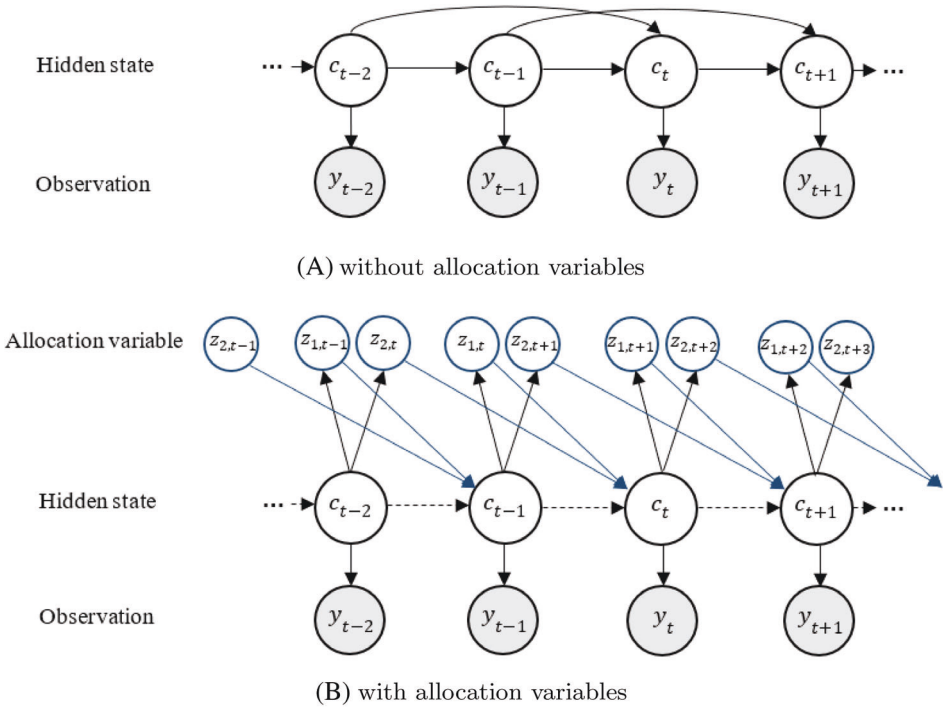

FIGURE 1 Dependence structure of a second-order hidden Markov model [Colour figure can be viewed at wileyonlinelibrary.com]

1. $\sum_{c_t=1}^{C} \lambda_{h_1,\ldots,h_q}(c_t) = 1$, for each combination $(h_1, \ldots, h_q)$;
2. $\sum_{h_j=1}^{k_j} \pi_{h_j}^{(j)}(c_{t-j}) = 1$, for each pair (j, c_{t-j}).

In such a factorization, the number of parameters is reduced to $(C-1)\prod_{j=1}^{q} k_j + C\sum_{j=1}^{q}(k_j - 1)$, which is much smaller than $(C-1)C^q$ if $\prod_{j=1}^{q} k_j \ll C^q$.

Marginalizing out the latent-class indicators $z_{j,t}$, the transition probability $p(c_t|\boldsymbol{c}_{(t-q):(t-1)})$ has an equivalent form as

$$p(c_t|\boldsymbol{c}_{(t-q):(t-1)}) = \sum_{h_1=1}^{k_1} \cdots \sum_{h_q=1}^{k_q} \lambda_{h_1,\ldots,h_q}(c_t) \prod_{j=1}^{q} \pi_{h_j}^{(j)}(c_{t-j}), \quad (6)$$

where $1 \le k_j \le C$ for all j. The generic form of the emission distribution is expressed as $p(y_t|c_t, \boldsymbol{\theta}) = f(y_t|\boldsymbol{\theta}_{c_t})$, where $\boldsymbol{\theta} = \{\boldsymbol{\theta}_c : c = 1, \ldots, C\}$ represents parameters indexed by the hidden states. The joint distribution of $\boldsymbol{y} = \{y_t : t = 1, \ldots, T\}$, $\boldsymbol{c} = \{c_t : t = q+1, \ldots, T\}$ and $\boldsymbol{z} = \{z_{j,t} : t = q+1, \ldots, T, j = 1, \ldots, q\}$ admits the following factorization:

$$\begin{aligned} &p(\boldsymbol{y}, \boldsymbol{c}, \boldsymbol{z} \mid \boldsymbol{\lambda}_k, \boldsymbol{\pi}_k, \boldsymbol{k}, \boldsymbol{\theta}) \\ &= \prod_{t=q+1}^{T} \{p(c_t|\boldsymbol{\lambda}_{z_t}) \prod_{j=1}^{q} p\left(z_{j,t}|w_{j,t}, \boldsymbol{\pi}_{k_j}^{(j)}, k_j\right)\} \prod_{t=1}^{T} f(y_t|\boldsymbol{\theta}_{c_t}) \\ &= p(\boldsymbol{y}|\boldsymbol{c}, \boldsymbol{\theta}) p(\boldsymbol{c}|\boldsymbol{z}, \boldsymbol{\lambda}_k, \boldsymbol{k}) p(\boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\pi}_k, \boldsymbol{k}), \end{aligned} \quad (7)$$

where $w_{j,t} = c_{t-j}$, representing the history state of c_t. The conditional independence relationships encoded in the factorization are used in deriving Markov chain Monte Carlo (MCMC) algorithms to draw samples from the posteriors. Detailed sampling algorithms for HOHMMs are referred to Sarkar & Dunson (2018).

3 | HIGHER-ORDER HIDDEN SEMI-MARKOV MODEL

In this paper, we extend an HOHMM to an HOHSMM, where the hidden-state sequence is governed by a semi-Markov chain. The HOHSMM is more flexible since it incorporates additional temporal structure by allowing the state duration to be generally distributed, rather than implicitly following a geometric distribution as in an HOHMM.

3.1 | Model development

We first give a brief description of the HSMM and then develop the HOHSMM. As discussed in the literature, there exist several specific models of the HSMM in prognostics by making different assumptions regarding the dependence between state transition and duration. For model simplicity and tractability, the explicit duration setting is widely used in health monitoring of engineering systems (Dong & He, 2007; Liu, Dong, et al., 2015), and we also use it to model the temporal structure in the proposed HOHSMM. Both HSMMs and HOHSMMs with explicit duration exclude state self-transitions because the duration distribution cannot fully capture a state's possible duration time if self-transitions are allowed.

An HSMM with explicit duration assumes that the underlying stochastic process is governed by a semi-Markov chain. Each state has a variable duration that follows an explicit state-specific distribution and a number of corresponding observations are produced while in the state (illustrated in Figure 2). The observation sequence

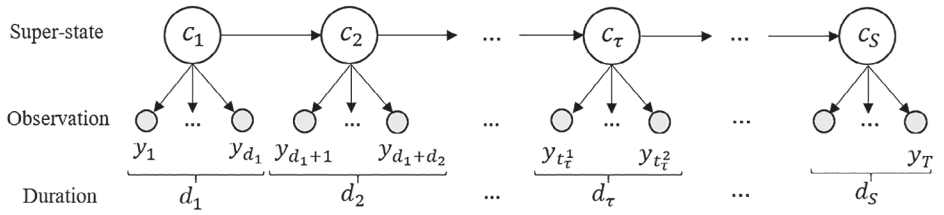

FIGURE 2 An hidden semi-Markov model (HSMM) with explicit duration

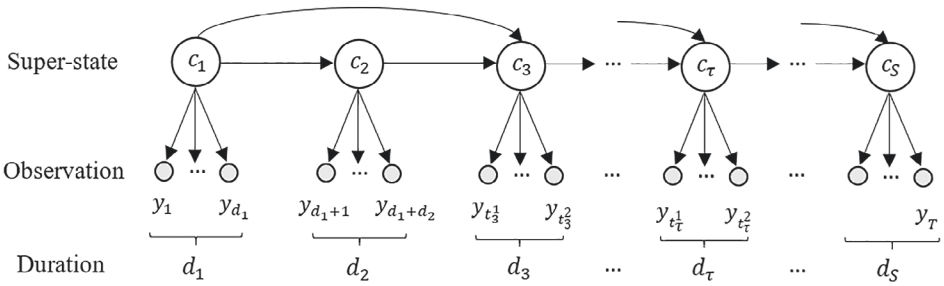

FIGURE 3 A second-order hidden semi-Markov model (HSMM)

$\{y_t : t = 1, \ldots, T\}$ is produced segmentally from the emission distribution $f(y|\theta_{c_\tau})$ indexed by the hidden super-state sequence $\{c_\tau : \tau = 1, \ldots, S\}$, where S is the number of segments. Observations are assumed to be collected discretely by a unit time, and therefore the number of observations produced in each super-state represents the state duration. For the τth segment, the state duration is denoted by d_τ and $\boldsymbol{y}_{t_\tau^1:t_\tau^2}$ denotes the produced observations, where $t_\tau^1 = \sum_{\psi<\tau} d_\psi + 1$, $t_\tau^2 = \sum_{\psi\le\tau} d_\psi$. In the last segment, the observations may be truncated, and we have $t_S^2 = \min\left\{\sum_{\psi\le S} d_\psi, T\right\}$.

In the proposed HOHSMM with the explicit duration setting, the hidden super-state sequence $\{c_\tau\}$ is assumed to be governed by a higher-order Markov chain and the state duration follows an explicit distribution, denoted by $g(\cdot|\xi_{c_\tau})$ with the parameters indexed by the specific hidden super-state c_τ. In general, the state duration can be modeled by Poisson, Gaussian, and gamma distributions, all of which belong to the exponential family (Yu, 2010). In real-world applications, the choice of distribution can be determined by finding the one that better fits the original histogram of state durations. For the problem of our interest, the condition monitoring data are typically collected on some discrete-time schedule (eg, hourly, daily, or weekly), and it is thus reasonable for us to employ the Poisson distribution to model the state duration.

An explicit-duration HOHSMM of maximal order q is constructed as follows:

$$p(c_\tau|c_1, \ldots, c_{\tau-1}) = p(c_\tau|c_{\tau-q}, \ldots, c_{\tau-1}), \quad \tau = q+1, \ldots, S,$$

$$d_\tau \sim g(d|\xi_{c_\tau}), \quad \tau = 1, \ldots, S,$$

$$\boldsymbol{y}_{t_\tau^1:t_\tau^2} \overset{\text{iid}}{\sim} f(y|\theta_{c_\tau}), \quad t_\tau^1 = \sum_{\psi<\tau} d_\psi + 1, \quad t_\tau^2 = \sum_{\psi\le\tau} d_\psi.$$

Figure 3 illustrates a second-order HSMM. In this example, the distribution of the hidden super-state c_τ depends on its previous two states $c_{\tau-1}$ and $c_{\tau-2}$, and the duration time in each super-state is generally distributed, following an explicit state-specific distribution.

To design an efficient MCMC sampling algorithm for HOHSMM inference, we first assign the prior distributions to model parameters. In order to exclude self-transitions in the super-state sequence for an HOHSMM, a modified hierarchical Dirichlet prior on the transition probability tensor is assigned as (Johnson & Willsky, 2013),

$$\begin{aligned}\boldsymbol{\lambda}_{i,h_2,\ldots,h_q} &= \{\lambda_{i,h_2,\ldots,h_q}(1), \ldots, \lambda_{i,h_2,\ldots,h_q}(C)\} \\ &\sim \text{Dir}\{\alpha\lambda_0(1), \ldots, \alpha\lambda_0(C)\}, \quad \forall(i, h_2, \ldots, h_q), \end{aligned} \tag{8}$$

$$\boldsymbol{\lambda}_0 = \{\lambda_0(1), \ldots, \lambda_0(C)\} \sim \text{Dir}(\alpha_0/C, \ldots, \alpha_0/C), \tag{9}$$

$$\begin{aligned}\overline{\lambda}_{i,h_2,\ldots,h_q}(i') &:= \frac{\lambda_{i,h_2,\ldots,h_q}(i')}{1-\lambda_{i,h_2,\ldots,h_q}(i)}(1-\delta_{ii'}), \\ \delta_{ii'} &= \begin{cases} 1 & \text{if } i = i', \\ 0 & \text{otherwise.} \end{cases}\end{aligned} \tag{10}$$

Equation (10) ensures that the self-transition probabilities are zeros. Note that i is the latent class the hidden super-state $c_{\tau-1}$ (the immediate precedent super-state of c_τ) is allocated into and i' is the state of c_τ (ie, $c_\tau = i'$). Therefore, to have a valid comparison between i and i' and exclude self-transitions, each possible state of $c_{\tau-1}$ must be allocated to a distinct latent class. In other words, each state of $c_{\tau-1}$ is a distinct latent class. To do so, we let $k_1 = C$ and $\boldsymbol{\pi}_C^{(1)}(c_{\tau-1}) = \{\pi_1^{(1)}(c_{\tau-1}), \ldots, \pi_C^{(1)}(c_{\tau-1})\}$, where $\pi_i^{(1)}(c_{\tau-1}) = \delta_{i,c_{\tau-1}}, \forall i = 1, \ldots, C$ and $\tau = q+1, \ldots, S$. For the remaining lags, the independent priors on the allocation distribution $\boldsymbol{\pi}_k$ are assigned as

$$\begin{aligned}\boldsymbol{\pi}_{k_j}^{(j)}(c_{\tau-j}) &= \{\pi_1^{(j)}(c_{\tau-j}), \ldots, \pi_{k_j}^{(j)}(c_{\tau-j})\} \sim \text{Dir}(\gamma_j, \ldots, \gamma_j), \\ &\forall(j, c_{\tau-j}), \quad j = 2, \ldots, q. \end{aligned} \tag{11}$$

By introducing latent allocation variables $z_{j,\tau}$ for $j = 1, \ldots, q$ and $\tau = q+1, \ldots, S$ with $z_{1,\tau} = c_{\tau-1}$, the hidden super-states $\{c_\tau\}$ are conditionally independent and the model can be represented through the following hierarchical formulation:

$$(c_\tau | z_{1,\tau} = i, z_{j,\tau} = h_j, j = 2, \ldots, q) \sim \text{Mult}(\{1, \ldots, C\}, \overline{\lambda}_{i,h_2,\ldots,h_q}(1), \ldots, \overline{\lambda}_{i,h_2,\ldots,h_q}(C)), \tag{12}$$

$$(z_{j,\tau} | c_{\tau-j}) \sim \text{Mult}(\{1, \ldots, k_j\}, \pi_1^{(j)}(c_{\tau-j}), \ldots, \pi_{k_j}^{(j)}(c_{\tau-j})), \quad \forall j = 1, \ldots, q. \tag{13}$$

The transition probability is then modeled as

$$p(c_\tau | \boldsymbol{c}_{(\tau-q):(\tau-1)}) = \sum_{i=1}^{C} \sum_{h_2=1}^{k_2} \cdots \sum_{h_q=1}^{k_q} \overline{\lambda}_{i,h_2,\ldots,h_q}(c_\tau) \times \pi_i^{(1)}(c_{\tau-1}) \prod_{j=2}^{q} \pi_{h_j}^{(j)}(c_{\tau-j}). \tag{14}$$

For an HOHSMM of order q with transition probability $p(c_\tau | \boldsymbol{c}_{(\tau-q):(\tau-1)})$ and emission distributions $\{f(y|\boldsymbol{\theta}_c) : c \in \mathscr{C}\}$, the r-step ahead predictive density is given by

$$f_{pred,S+r}(y|\boldsymbol{y}_{1:S}) = E_{(\boldsymbol{c},\boldsymbol{\zeta})} \left[\sum_{c_{S+r}} \cdots \sum_{c_{S+1}} f(y|c_{S+r}, \boldsymbol{\zeta}) \times p\left(c_{S+r} | \boldsymbol{c}_{(S+r-q):(S+r-1)}, \boldsymbol{\zeta}\right) \cdots \times p\left(c_{S+1} | \boldsymbol{c}_{(S+1-q):S}, \boldsymbol{\zeta}\right) \right], \tag{15}$$

where S is the number of segments for the observation sequence $\boldsymbol{y}$ and $\boldsymbol{\zeta} = (\boldsymbol{k}, \overline{\boldsymbol{\lambda}}_{\boldsymbol{k}}, \boldsymbol{\pi}_k, \boldsymbol{\theta})$.

Finally, the following independent priors are assigned on k_j's

$$p_{0,j}(k) \propto \exp(-\varphi j k), \quad j = 2, \ldots, q, \quad k = 1, \ldots, C, \tag{16}$$

where $\varphi > 0$. The prior $p_{0,j}$ assigns increasing probabilities to smaller values of k_j as the lag j becomes more distant, reflecting the natural belief that increasing lags have diminishing influence on the distribution of c_τ.

The joint distribution of $\boldsymbol{y} = \{y_t : t = 1, \ldots, T\}$, $\boldsymbol{c} = \{c_\tau : \tau = q+1, \ldots, S\}$, and $\boldsymbol{z} = \{z_{j,\tau} : \tau = q+1, \ldots, S, j = 1, \cdots, q\}$ can be presented as

$$p(\boldsymbol{y}, \boldsymbol{c}, \boldsymbol{z} \mid \overline{\boldsymbol{\lambda}}_k, \boldsymbol{\pi}_k, \boldsymbol{k}, \boldsymbol{d}, \boldsymbol{\theta}) = \prod_{\tau=q+1}^{S} \left\{ p(c_\tau | \overline{\boldsymbol{\lambda}}_{z_\tau}) \prod_{j=1}^{q} p\left(z_{j,\tau} | w_{j,\tau}, \boldsymbol{\pi}_{k_j}^{(j)}, k_j\right) \right\} \prod_{\tau=1}^{S} f(\boldsymbol{y}_{t_\tau^1:t_\tau^2} | \boldsymbol{\theta}_{c_\tau}), \tag{17}$$

where $w_{j,\tau} = c_{\tau-j}$, and $f(\boldsymbol{y}_{t_\tau^1:t_\tau^2} | \boldsymbol{\theta}_{c_\tau}) = \prod_{i=t_\tau^1}^{t_\tau^2} f(y_i | \boldsymbol{\theta}_{c_\tau})$, $t_\tau^1 = \sum_{\psi<\tau} d_\psi + 1$, $t_\tau^2 = \sum_{\psi \le \tau} d_\psi$. The conditional independence relationships encoded in the joint distribution are used in deriving MCMC algorithms for the HOHSMM.

3.2 | Model inference

We use the MCMC sampling method for explicit-duration HOHSMM inference. The sampler is designed based on the two-stage Gibbs sampling algorithms for HOHMM (Sarkar & Dunson, 2018). There are additional challenges in HOHSMM inference due to explicit temporal structure, excluding self-transitions, and multiple observed trajectories in the training data.

The first challenge is brought by incorporating explicit temporal structure (ie, duration distribution), which requires additional sampling to determine the number of segments (ie, the number of hidden super-states) and the duration time in each state. Existing sampling inference methods for HSMMs often use a message-backwards and sample-forwards technique to address this problem (Johnson & Willsky, 2013). We cannot directly apply these methods for an HOHSMM since the backwards messages are extremely difficult to define and compute when higher-order transitions present. The reversible jump MCMC provides a statistical inference strategy for Bayesian model determination, where the dimensionality of the parameter vector is typically not fixed (eg, the multiple change-point problem for Poisson processes) (Green, 1995). However, it cannot be used to sample change points of a sequence in an HOHSMM since there is no appropriate mechanism to update the hidden super-states affected by the moves of change points (eg, birth of a change point, death of a change point). The second challenge is brought by excluding self-transitions. A Dirichlet distribution is assigned as the conjugate prior for transition probability parameters in the HOHMM, but the conjugacy does not exist after setting self-transition probabilities to zeros. A mechanism to recover the conjugacy for updating transition probability parameters is needed. In addition, in many real-world applications, several identical units are typically monitored at the same time to collect sensor data. How to leverage all information provided by multiple observed trajectories (ie, observation sequences) instead of using just one sequence is the third challenge. We address these difficulties in the following two sections.

3.2.1 | Update segmentation

We denote P run-to-failure observation sequences by $\boldsymbol{y}^{(1:P)}$ and the pth observation sequence by $\boldsymbol{y}^{(p)} = \{y_t^{(p)} : t = 1, \ldots, T_p\}$, where T_p is the observed length and $p = 1, \ldots, P$. These sequences are assumed to be independent. To address the first challenge, we introduce a jump size threshold (α) to identify change points. For each observation sequence, if the Euclidean distance between a point and its immediate previous point is greater than α, this point is identified as a change point. The prior of α is assigned to be a uniform distribution with support $(\alpha_{\min}, \alpha_{\max})$, where $\alpha_{\min}$ and $\alpha_{\max}$ are the 5th and 95th percentile values obtained from the distances between two adjacent observed data points in all observation sequences, respectively. We then update the segmentation of the observation sequences and initialize the

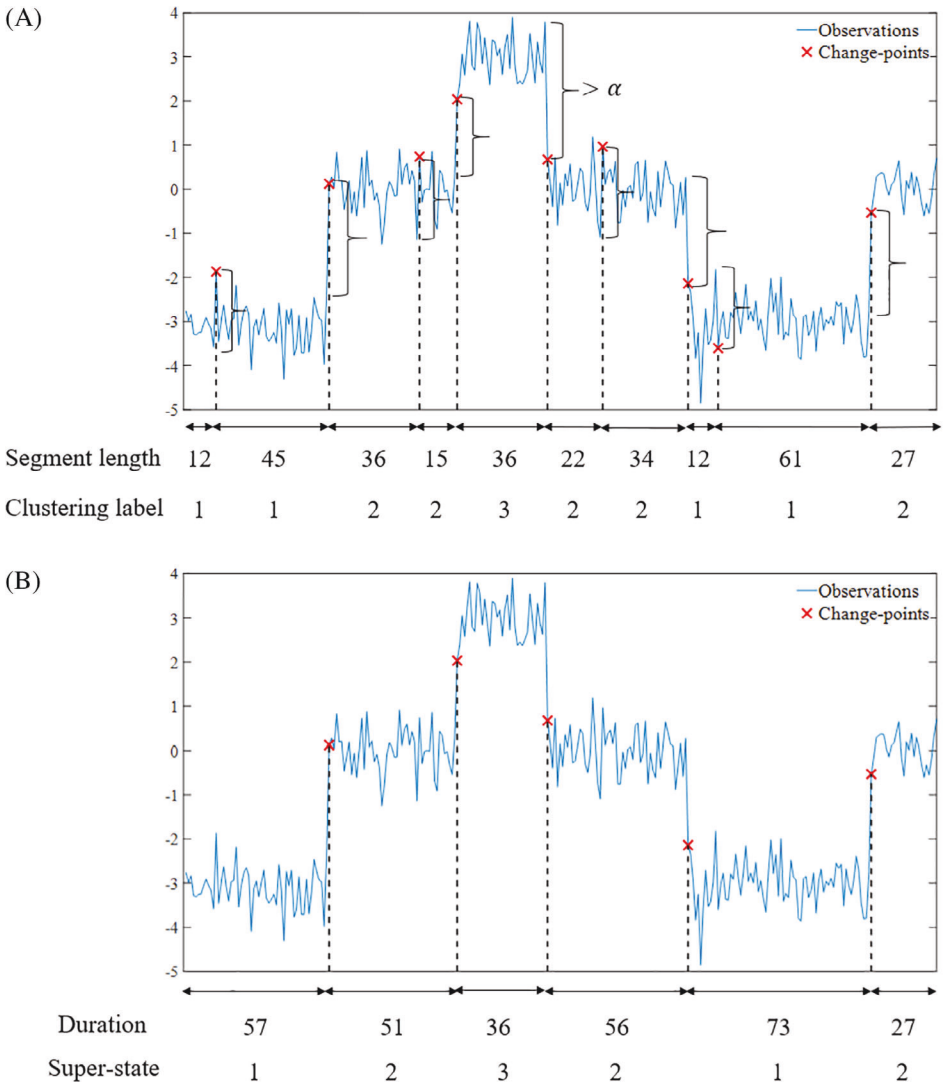

FIGURE 4 Illustration for updating segmentation and initializing hidden super-states given jump size threshold α (using one-dimensional observation as an example). (A), Identify change points given jump size threshold α. A red cross indicates a change point that is detected if the difference (absolute value) between an observation and its previous one is larger than α. Nine change points are identified and the observation sequence is segmented accordingly as presented by vertical dashed black lines. Clustering labels are derived by clustering the mean values of observations in these 10 segments. (B), Segmentation and hidden super-states initialization after clustering and merging processes. If two adjacent segments have the same clustering label, merge these two segments and use the clustering label as the initialized hidden super-state [Colour figure can be viewed at wileyonlinelibrary.com]

hidden super-state sequences iteratively by sampling the jump size threshold α.

In each iteration of the HOHSMM sampler, we propose a new threshold α from $U(\alpha_{\min}, \alpha_{\max})$. For each observation sequence, we mark change points based on the computed distances (illustrated in Figure 4(A)) and the sequence is segmented accordingly. After the initial segmentation, we compute the center of the observed data points for each segment and label the segments by clustering the centers. The hidden super-states are initialized by using the clustering labels. To exclude self-transitions, if two adjacent segments have the same clustering label, we merge these two segments. For example, the first two segments in Figure 4(A) share the same label 1, and these two segments are merged into one. After the clustering and merging processes, we obtain the final segmentation and the initialized hidden super-states of an observation sequence for a given jump size threshold (illustrated in Figure 4(B)). Based on the segmentation results, we also obtain the number of segments and the state duration for each observation sequence, denoted by S_p and $\boldsymbol{d}^{(p)}$, respectively.

The hidden super-state sequence $\boldsymbol{c}^{(p)}$, latent allocation variables $\boldsymbol{z}^{(p)}$, and other parameters $\boldsymbol{k}$, $\boldsymbol{\pi}_k$, λ_k, $\overline{\lambda}_k$, λ_0, $\boldsymbol{\theta}$ are updated using the two-stage Gibbs sampling algorithm for the proposed HOHSMM. The first stage is to identify the important lags by sampling $\boldsymbol{k}$ from the posterior. Given the determined $\boldsymbol{k}$, we collect the samples of other parameters in the second stage. The obtained samples will be used to compute the acceptance probability for updating jump size threshold α. In general MCMC sampling, the acceptance probability can be computed as

$$\min\{1, (\text{likelihood ratio}) \times (\text{prior ratio}) \times (\text{proposal ratio})\}. \tag{18}$$

Since the prior of α is a uniform distribution and α is also proposed from the uniform distribution, it is obvious that the prior ratio and the proposal ratio are equal to 1. The posterior mean of the likelihood can be approximated using the obtained samples (Ando, 2010), which is provided as

$$L_\alpha = \frac{1}{N}\sum_{j=1}^{N} f\left(\boldsymbol{y}^{(1:P)} \mid \boldsymbol{c}^{(1:P),j}, \boldsymbol{\theta}^j, \alpha\right) = \frac{1}{N}\sum_{j=1}^{N}\left[\prod_{p=1}^{P}\prod_{\tau=1}^{S_p} f\left(\boldsymbol{y}^{(p)}_{t_\tau^1:t_\tau^2} | \boldsymbol{\theta}^j_{c_\tau^{(p),j}}\right)\right], \tag{19}$$

where $\{\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^N, \boldsymbol{c}^{(1:P),1}, \ldots, \boldsymbol{c}^{(1:P),N}\}$ is a set of posterior samples generated from their posterior distributions.

Given all the collected samples of α, the most likely jump size threshold α^* is determined by computing the average value of the samples after burn-in. We then use α^* to update segmentation and repeat the two-stage Gibbs sampling process to obtain the final segmentation and samples for all parameters. Given an explicit distribution $g(\cdot|\xi_c)$ for each super-state's duration, the MLEs $\{\widehat{\xi}_c\}$ for parameters $\{\xi_c : c = 1, \ldots, C\}$ can be easily obtained using the final segmentation result.

3.2.2 | The two-stage Gibbs sampling algorithm for HOHSMMs

Given the segmentation, we modify the two-stage Gibbs sampling algorithm in Sarkar and Dunson (2018) to draw samples of $\boldsymbol{k}$, $\boldsymbol{\pi_k}$, λ_k, $\overline{\lambda}_{\boldsymbol{k}}$, λ_0, $\boldsymbol{\theta}$, $\boldsymbol{z}$, and $\boldsymbol{c}$ from the posteriors in the HOHSMM. The first stage is to determine the values of $\boldsymbol{k} = \{k_1, \ldots, k_q\}$, which is the important lag indicator. Given determined values of $\boldsymbol{k}$, we update other model parameters $\boldsymbol{\pi_k}$, λ_k, $\overline{\lambda}_{\boldsymbol{k}}$, λ_0, $\boldsymbol{\theta}$, latent allocation variables $\boldsymbol{z}$, and hidden super-state sequence $\boldsymbol{c}$ in the second stage. In this proposed sampling algorithm, the order of super-state transition dynamics does not need to be prespecified, rather it is determined through the important lag indicators $\boldsymbol{k}$. If only the first lag is identified to be important (ie, $k_1 > 1$ and $k_j = 1$ for $j = 2, \ldots, q$), then the hidden super-state transition depends only on the immediately previous super-state, indicating that the hidden super-state sequence is governed by a Markov chain. Therefore, the proposed two-stage sampling algorithm allows to learn the history dependency from the observed data by identifying the important lags. To address the third challenge of multiple trajectories, we use the joint distribution of all observation sequences. Based on the assumption that all sequences are independent, the joint distribution can be obtained as follows:

$$p\left(\boldsymbol{y}^{(1:P)}, \boldsymbol{c}^{(1:P)}, \boldsymbol{z}^{(1:P)} | \overline{\lambda}_k, \boldsymbol{\pi}_k, \boldsymbol{k}, \boldsymbol{d}^{(1:P)}, \boldsymbol{\theta}\right) = \prod_{p=1}^{P} p\left(\boldsymbol{y}^{(p)}, \boldsymbol{c}^{(p)}, \boldsymbol{z}^{(p)} | \overline{\lambda}_k, \boldsymbol{\pi}_k, \boldsymbol{k}, \boldsymbol{d}^{(p)}, \boldsymbol{\theta}\right). \tag{20}$$

The conditional independence relationships encoded in the joint distribution are used in deriving the two-stage Gibbs sampling algorithm for HOHSMMs.

Specifically, in the first stage, we identify important lags and the corresponding number of latent classes by sampling $\boldsymbol{k}$. In this stage, we use an approximated model which forces hard allocation of $z_{j,\tau}$'s instead of soft allocation. Hard allocation means that, partition the state space into k_j clusters for the jth lag, then each cluster corresponds to its own latent class. In other words, each state is allocated into one class with probability 1. For example, partition the states {1,2,3,4,5,6} into $k_j = 2$ clusters with $\mathscr{C}_{j,1} = \{1,2,3\}$ and $\mathscr{C}_{j,2} = \{4,5,6\}$ for the jth lag, hard allocation means that $c_{\tau-j} = 1$, 2, and 3 will be allocated to the first latent class and $c_{\tau-j} = 4$, 5, and 6 will be allocated to the second one with probability 1. In soft allocation, one state can be allocated into several possible classes with specific probabilities. The mixture probabilities in the approximated model are denoted by $\widetilde{\boldsymbol{\pi}}_{\boldsymbol{k}}$, indicating hard clustering while $\boldsymbol{\pi_k}$ indicates soft allocation.

Based on the approximated model, samples of the parameters are drawn from their respective conditional posteriors following the prespecified order. We first examine the posteriors of the transition distributions λ_k and λ_0. There exist computational machineries of sampling from the posteriors in hierarchical Dirichlet process (HDP) models (Teh, Jordan, Beal, & Blei, 2006). In the HOHMM, the Dirichlet distribution is the conjugate prior of transition probability parameter, so it is straightforward to update the parameters $\lambda_{\boldsymbol{k}}$. However, in our HOHSMM, the method used to exclude self-transitions makes the model not fully conjugate. Specifically, let $n_{i,h_2,\ldots,h_q}(c) = \sum_p \sum_\tau 1\{z^{(p)}_{1,\tau} = i, z^{(p)}_{2,\tau} = h_2, \ldots, z^{(p)}_{q,\tau} = h_q, c^{(p)}_\tau = c\}$, which counts the number of transitions from the latent allocation classes $(i, h_2, \ldots, h_q)$ to state c among all observation sequences where $i = 1, \ldots, C$ and $h_j = 1, \ldots, k_j$ for $j = 2, \ldots, q$. Because of no self-transitions, we have $n_{i,h_2,\ldots,h_q}(i) = 0$. We consider the posterior distribution of $\lambda_{1,h_2,\ldots,h_q} = \{\lambda_{1,h_2,\ldots,h_q}(1), \lambda_{1,h_2,\ldots,h_q}(2), \ldots, \lambda_{1,h_2,\ldots,h_q}(C)\}$,

$$\begin{aligned} p(\lambda_{1,h_2,\ldots,h_q} \mid \lambda_0, \boldsymbol{c}, \boldsymbol{z}) \propto\ & [\lambda_{1,h_2,\ldots,h_q}(1)]^{\alpha\lambda_0(1)-1} \\ & \times [\lambda_{1,h_2,\ldots,h_q}(2)]^{\alpha\lambda_0(2)-1} \cdots [\lambda_{1,h_2,\ldots,h_q}(C)]^{\alpha\lambda_0(C)-1} \\ & \times \left(\frac{\lambda_{1,h_2,\ldots,h_q}(2)}{1-\lambda_{1,h_2,\ldots,h_q}(1)}\right)^{n_{1,h_2,\ldots,h_q}(2)} \\ & \cdots \left(\frac{\lambda_{1,h_2,\ldots,h_q}(C)}{1-\lambda_{1,h_2,\ldots,h_q}(1)}\right)^{n_{1,h_2,\ldots,h_q}(C)}. \end{aligned} \tag{21}$$

Because of the extra $\frac{1}{1-\lambda_{1,h_2,\ldots,h_q}(1)}$ terms from the likelihood by excluding self-transitions, we cannot reduce this expression to the Dirichlet form over the components of $\lambda_{1,h_2,\ldots,h_q}$. Therefore, the model is not fully conjugate and new posteriors need to be derived. To recover conjugacy, we introduce auxiliary variables $\{\rho_s\}_{s=1}^n$, where $n = \sum_{c=1}^C n_{i,h_2,\ldots,h_q}(c)$. Each ρ_s is independently drawn from a geometric distribution with specific success parameter $1 - \lambda_{i,h_2,\ldots,h_q}(i)$ (Johnson & Willsky, 2013). We adjust the sampling algorithm by updating transition parameters $\lambda_{i,h_2,\ldots,h_q} = \{\lambda_{i,h_2,\ldots,h_q}(1), \ldots, \lambda_{i,h_2,\ldots,h_q}(i), \ldots, \lambda_{i,h_2,\ldots,h_q}(C)\}$ from the

Algorithm 1. Explicit-duration HOHSMM Sampler

Input: Observation sequences $\{\boldsymbol{y}^{(p)} : p = 1, \dots, P\}$ and sample size l.

Initialization:
  Compute distances between two adjacent data points in all sequences $\boldsymbol{y}^{(p)}$ and use the 5th and 9th percentile values as the lower bound and upper bound of the support $(\alpha_{\min}, \alpha_{\max})$.
  Set initial likelihood value: $L_0 \leftarrow e^{-10^{10}}$.
for $v = 1$ to l **do**
  Sample $\alpha_v \sim U(\alpha_{\min}, \alpha_{\max})$.
  Update segmentation:
    For each $\boldsymbol{y}^{(p)}$, identify change points given α_v. Compute the center of the observed data points for each segment and initialize hidden super-state sequence $\{c_\tau^{(p)}\}$ by clustering the centers. Merge adjacent segments that have the same label and derive the number of segment S_p and duration times $\boldsymbol{d}^{(p)}$, where $\tau = 1, \dots, S_p$, $p = 1, \dots, P$.
  Stage 1 (Determine $\boldsymbol{k}$):
    Update $\boldsymbol{\lambda_k}$:
      Let $n_{i,h_2,\dots,h_q}(c) = \sum_p \sum_\tau 1\{z_{1,\tau}^{(p)} = i, z_{2,\tau}^{(p)} = h_2, \dots, z_{q,\tau}^{(p)} = h_q, c_\tau^{(p)} = c\}$ and $n = \sum_{c=1}^{C} n_{i,h_2,\dots,h_q}(c)$, where $i = 1, \dots, C$ and $h_j = 1, \dots, k_j$ for $j = 2, \dots, q$.
      Independently sample $\rho_s \sim \text{Geo}(1 - \lambda_{i,h_2,\dots,h_q}(i)), s = 1, \dots, n$.
      Sample $\lambda_{i,h_2,\dots,h_q} = \{\lambda_{i,h_2,\dots,h_q}(1), \dots, \lambda_{i,h_2,\dots,h_q}(i), \dots, \lambda_{i,h_2,\dots,h_q}(C) \sim \text{Dir}\{\alpha\lambda_0(1) + n_{i,h_2,\dots,h_q}(1), \dots, \alpha\lambda_0(i) + \sum_{s=1}^{n} \rho_s, \dots, \alpha\lambda_0(C) + n_{i,h_2,\dots,h_q}(C)\}$.
    Update $\bar{\boldsymbol{\lambda}}_{\boldsymbol{k}}$: Compute $\bar{\boldsymbol{\lambda}}_{\boldsymbol{k}}$ by Equation (10).
    Update λ_0:
      For $r = 1, \dots, n_{i,h_2,\dots,h_q}(c)$, sample $x_r \sim \text{Bernoulli}\left\{\frac{\alpha\lambda_0(c)}{r-1+\alpha\lambda_0(c)}\right\}$.
      Let $m_{i,h_2,\dots,h_q}(c) = \sum_r x_r$ and $m_0(c) = \sum_{(i,h_2,\dots,h_q)} m_{i,h_2,\dots,h_q}(c)$.
      Sample $\lambda_0 = \{\lambda_0(1), \dots, \lambda_0(C)\} \sim \text{Dir}\{\alpha_0/C + m_0(1), \dots, \alpha_0/C + m_0(C)\}$.
    Update$\{c_\tau^{(p)} : \tau = 1, \dots, S_p\}$, $\{z_{j,\tau}^{(p)} : \tau = q+1, \dots, S_p, j = 1, \dots, q\}$, $\boldsymbol{\theta}, \boldsymbol{k}, \tilde{\boldsymbol{\pi}}_{\boldsymbol{k}}$ as in Sarkar & Dunson (2018).
  Stage 2 (Sample with determined $\bar{\boldsymbol{k}}$):
    Update $\boldsymbol{\pi_k}$:
      Let $n_{j,w_j}(h_j) = \sum_p \sum_\tau 1\{w_{j,\tau}^{(p)} = w_j, z_{j,\tau}^{(p)} = h_j\}$, where $w_{j,\tau}^{(p)} = c_{\tau-j}^{(p)}$.
      Sample $\boldsymbol{\pi}_{k_j}^{(j)}(w_j) = \{\pi_1^{(j)}(w_j), \dots, \pi_{k_j}^{(j)}(w_j)\} \sim \text{Dir}\{\gamma_j + n_{j,w_j}(1), \dots, \gamma_j + n_{j,w_j}(k_j)\}$.
    Update $\boldsymbol{\lambda_k}$, $\bar{\boldsymbol{\lambda}}_{\boldsymbol{k}}$, λ_0 as in **Stage 1**.
    Update $\boldsymbol{z}^{(p)}$: Sample from
      $$p(z_{j,\tau} = h | z_{l,\tau} = h_l, l \neq j, \bar{\boldsymbol{\lambda}}_{\boldsymbol{k}}, \boldsymbol{\pi_k}, \boldsymbol{c}) \propto \bar{\lambda}_{h_1,\dots,h_{j-1},h,h_{j+1},\dots,h_q}(c_\tau)\pi_h^{(j)}(c_{\tau-j}).$$
    Update $\boldsymbol{c}^{(p)}$: Sample from $p(c_\tau | \bar{\boldsymbol{\lambda}}_{\boldsymbol{k}}, \boldsymbol{\pi_k}, \boldsymbol{\theta}, \boldsymbol{z}) \propto \bar{\lambda}_{z_{1,\tau},z_{2,\tau},\dots,z_{q,\tau}}(c_\tau) f(\boldsymbol{y}_{t_\tau^1:t_\tau^2} | \boldsymbol{\theta}_{c_\tau}) \prod_{j=1}^{q} \pi_{z_{j,\tau+j}}^{(j)}(c_\tau)$.
    Update $\boldsymbol{\theta}$ as in **Stage 1**.
  Update α: Compute likelihood value L_{α_v} from Equation (19).
    if $\min\left\{\frac{L_{\alpha_v}}{L_0}, 1\right\} >$ rand **then**$\alpha(v) \leftarrow \alpha_v$ and $L_0 \leftarrow L_{\alpha_v}$.
    else $\alpha(v) \leftarrow \alpha(v-1)$.
    end if
end for
Determine α^*:
  Use the average value of sampled α after burn-in as the most likely jump size threshold α^*.
  Given α^*, repeat **Update segmentation**, **Stage 1** and **Stage 2** and collect final samples.
  Compute the MLEs $\{\hat{\xi}_c\}$ for duration distribution using samples of $\boldsymbol{c}^{(p)}$ and $\boldsymbol{d}^{(p)}$ for all p.

Output: α^*, $\boldsymbol{k}$, S_p^*, $\{\hat{\xi}_c\}$, and samples of $\bar{\boldsymbol{\lambda}}_{\boldsymbol{k}}$, $\boldsymbol{\pi_k}$, $\boldsymbol{\theta}$.

posterior distribution

$$\text{Dir}\left\{\alpha\lambda_0(1) + n_{i,h_2,\dots,h_q}(1), \dots, \alpha\lambda_0(i) + \sum_{s=1}^{n}\rho_s, \dots, \alpha\lambda_0(C) + n_{i,h_2,\dots,h_q}(C)\right\}.$$

Then we compute $\bar{\boldsymbol{\lambda}}_{\boldsymbol{k}}$ from Equation (10) and update $\boldsymbol{\lambda}_0$.

Since the observation sequences are independent, $\boldsymbol{c}^{(p)}$ and $\boldsymbol{z}^{(p)}$ are updated sequence by sequence. For each sequence, $\boldsymbol{c}^{(p)}$ and $\boldsymbol{z}^{(p)}$ are sampled by applying a Metropolis-Hastings step and using simulated annealing to facilitate the convergence. The full conditionals of $\boldsymbol{\theta}$ will depend on the choice of the emission distribution. Finally, a stochastic search variable

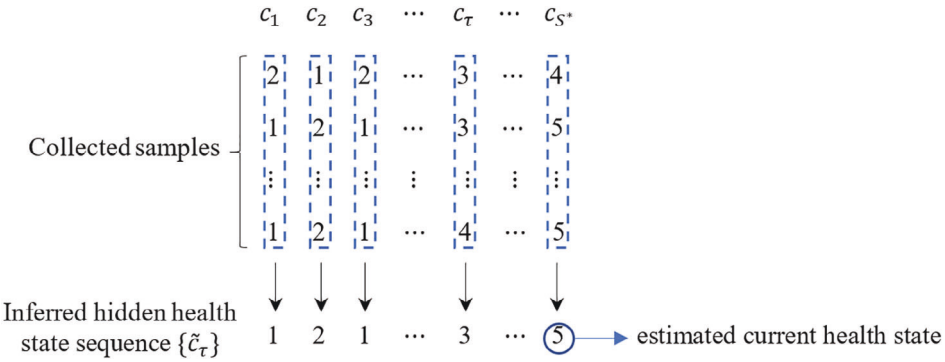

FIGURE 5 Illustration for determining the hidden health states [Colour figure can be viewed at wileyonlinelibrary.com]

selection method (George & McCulloch, 1997) is used to sample $\boldsymbol{k}$ from their posteriors and $\widetilde{\boldsymbol{\pi}}_{\boldsymbol{k}}$ are updated by the latent allocation cluster mapping. In the first stage, important lags can be determined and the number of latent classes for each important lag can be derived based the samples of $\boldsymbol{k}$.

The second stage, given the important lag inclusion result, is to sample parameters $\boldsymbol{\pi_k}$, λ_k, $\overline{\boldsymbol{\lambda}}_{\boldsymbol{k}}$, λ_0, $\boldsymbol{\theta}$, $\boldsymbol{z}^{(p)}$, and $\boldsymbol{c}^{(p)}$ iteratively. Given the segmentation, the elements of $\boldsymbol{c}^{(p)}$, $\boldsymbol{z}^{(p)}$, and $\boldsymbol{\pi}_k$ have either multinomial or Dirichlet full conditionals and can be straightforwardly updated. Sampling of λ_k, $\overline{\boldsymbol{\lambda}}_{\boldsymbol{k}}$, λ_0, and emission parameters $\boldsymbol{\theta}$ is the same as described in first stage. Details of the HOHSMM inference method are summarized in Algorithm 1.

4 | HEALTH-STATE DECODING

The ultimate purpose of a prognostics framework is to assess the current condition of a system (or component) and to make inferences regarding the remaining useful life (RUL). In this section, we first present how to use the HOHSMM-based prognostics framework to decode the hidden super-states. For an operating system with observation sequence $\boldsymbol{y}$, Equation (17) provides the joint distribution of $\boldsymbol{y}$, $\boldsymbol{c}$, and $\boldsymbol{z}$ conditioned on the learned model parameters $\boldsymbol{k}$, $\overline{\boldsymbol{\lambda}}_{\boldsymbol{k}}$, $\boldsymbol{\pi}_k$, $\boldsymbol{\theta}$, and duration times $\boldsymbol{d}$. We can directly use it for decoding the hidden super-states by sampling $\boldsymbol{c}$ and $\boldsymbol{z}$ from the posteriors. We need to first segment the observation sequence based on the learned α^* to determine state duration times $\boldsymbol{d}$ and then initialize the hidden super-state for each segment. Therefore, we use the same procedure described in Algorithm 1 by identifying change points given α^*. Next, we initialize the allocation variables $\boldsymbol{z}$ based on the initialized $\boldsymbol{c}$ and the learned allocation distribution $\boldsymbol{\pi}_k$. Given the values of $\boldsymbol{k}$ and historical samples of $\overline{\boldsymbol{\lambda}}_{\boldsymbol{k}}$, $\boldsymbol{\theta}$ from the learned model, updating $\boldsymbol{c}$ and $\boldsymbol{z}$ is the same as in Algorithm 1. The collected samples of $\boldsymbol{c}$ are used to determine the hidden health states of this specific system by using the most persistent sample (ie, the mode) for each segment. Figure 5 provides an illustrative example. In Figure 5, we can see that state 1 appears most in the posterior samples for the first super-state c_1, and is therefore used as the estimated super-state for the first segment. The same selection criterion is used to determine the hidden super-states for all segments. Details of the decoding procedure are summarized in Algorithm 2.

5 | RUL ESTIMATION

In this section, we estimate the RUL given the decoded hidden health states. For notational convenience, we omit the superscript of the number of segments S^* in the following analysis. It is impossible to analytically compute the RUL in an HOHSMM due to higher-order state transitions. Therefore, we use a simulation approach to predict the RUL, which is the expected time from the current health state to the failure state. Before presenting the RUL estimation method, we first show how to identify the failure state. The HOHSMM is trained using multiple run-to-failure independent observation sequences and historical samples of hidden super-states can be used to identify the failure state. For each sequence, from the decoded hidden super-states $\{c_\tau^{(p)}\}$, we identify the failure state $c_F^{(p)}$ by choosing the most persistent state in the last f states (Tobon-Mejia et al., 2012),

$$\begin{aligned}&\text{Super-state sequence} = (c_1^{(p)}, c_2^{(p)}, \ldots, c_{S_p}^{(p)}),\\ &\text{Last } f \text{ states} = (c_{S_p-f+1}^{(p)}, \ldots, c_{S_p-2}^{(p)}, c_{S_p-1}^{(p)}, c_{S_p}^{(p)}).\end{aligned} \tag{22}$$

The value of f can be chosen based on experience. Then, the final failure state c_F is given as the most persistent state in all $c_F^{(p)}$, $p = 1, \ldots, P$. Figure 6 illustrates the procedure to select the final failure state. For illustrative purpose, we arbitrarily use the last five super-states to identify the failure state for each sequence by choosing the most persistent state appeared in the last five states. We can see that state 5 appears most in the last five states for sequence 1 and 2, and state 4 is the most persistent state in the last five states for sequence P. We then choose the most persistent state in the identified failure states

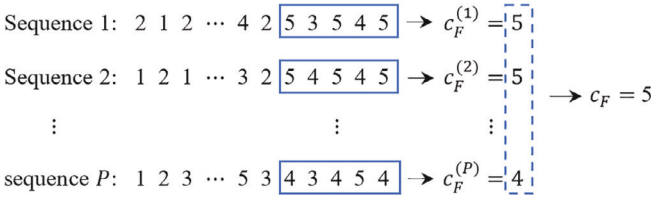

FIGURE 6 Illustration for identifying the failure state c_F with $f = 5$ [Colour figure can be viewed at wileyonlinelibrary.com]

Algorithm 2. Decoding for the HOHSMM

Input: Model parameters α^*, $\boldsymbol{k}$, $\bar{\boldsymbol{\lambda}}_{\boldsymbol{k}}$, $\boldsymbol{\pi}_{\boldsymbol{k}}$, $\boldsymbol{\theta}$ and observation sequence $\boldsymbol{y}$.

1: **Determine segmentation**:
Compute distances between two adjacent data points in $\boldsymbol{y}$ and identify change points given threshold α^*. Compute the center of the observed data points for each segment and initialize the hidden super-state c_τ by determining the clustering label based on the learned clustering rules. Merge adjacent segments that have the same label and derive the number of segments S^* and duration times $\boldsymbol{d}$, where $\tau = 1, \ldots, S^*$.

2: **Decode**:

3: **Initialize** $\boldsymbol{z}$:

4: Sample $(z_{j,\tau}|c_{\tau-j}) \sim \text{Mult}(\{1, \ldots, k_j\}, \pi_1^{(j)}(c_{\tau-j}), \ldots, \pi_{k_j}^{(j)}(c_{\tau-j}))$, where $j = 1, \ldots, q$.

5: **Update** $\boldsymbol{c}$:

6: Sample c_τ from $p(c_\tau|\bar{\boldsymbol{\lambda}}_{\boldsymbol{k}}, \boldsymbol{\pi}_{\boldsymbol{k}}, \boldsymbol{\theta}, \boldsymbol{z}) \propto \bar{\lambda}_{z_{1,\tau}, z_{2,\tau}, \ldots, z_{q,\tau}}(c_\tau) f(\boldsymbol{y}_{t_\tau^1 : t_\tau^2}|\boldsymbol{\theta}_{c_\tau}) \prod_{j=1}^{q} \pi_{z_{j,\tau+j}}^{(j)}(c_\tau)$.

7: **Update** $\boldsymbol{z}$:

8: Sample $z_{j,\tau}$ from
$$p(z_{j,\tau} = h|z_{l,\tau} = h_l, l \neq j, \bar{\boldsymbol{\lambda}}_{\boldsymbol{k}}, \boldsymbol{\pi}_{\boldsymbol{k}}, \boldsymbol{c}) \propto \bar{\lambda}_{h_1, \ldots, h_{j-1}, h, h_{j+1}, \ldots, h_q}(c_\tau) \pi_h^{(j)}(c_{\tau-j}).$$

9: **Determine** hidden health states $\{\tilde{c}_\tau : \tau = 1, \ldots, S^*\}$ by using the most persistent samples.

Output: $\{\tilde{c}_\tau : \tau = 1, \ldots, S^*\}$.

TABLE 1 Parameter setting of the simulation experiments

True dynamics		Third, second, first
Emission distribution	Mean (μ_1, μ_2, μ_3)	(−3,0,3), (−4.5,0,4.5), (−6,0,6)
	Variance σ_c^2	0.5^2, 1^2, 1.5^2
Duration distribution (ξ_1, ξ_2, ξ_3)		(22,18,14)

of all sequences as the final failure state, which is state 5 in this example.

Given the decoded hidden health states $\{\widetilde{c}_\tau : \tau = 1, \ldots, S\}$ and the identified failure state c_F, we use a simulation method to estimate the RUL. We simulate M hidden super-state paths. Each path starts from states $(\widetilde{c}_{S-q+1}, \ldots, \widetilde{c}_S)$ and the next state c_{S+1} is generated by drawing a sample from the multinomial distribution with probabilities $(p(c_{S+1} = 1|\widetilde{\boldsymbol{c}}_{(S-q+1):S}), \ldots, p(c_{S+1} = C|\widetilde{\boldsymbol{c}}_{(S-q+1):S}))$, which are computed using Equation (14) given the learned parameters $\bar{\boldsymbol{\lambda}}_{\boldsymbol{k}}$ and $\boldsymbol{\pi}_k$. Repeat this procedure by considering c_{S+1} as the current health state until the failure state c_F is first reached. Denote the ith paths by $\{c_{S+1}, \ldots, c_{S+N_i}\}$, where N_i is the total number of super-states generated in the ith paths and $c_{S+N_i} = c_F$ for all $i = 1, \ldots, M$. For each path, we sample the duration time for each super-state and use the sum of all sampled duration times as the RUL. We then estimate the mean RUL and the respective interval estimates based on the simulated RULs. The computation procedure is summarized as follows:

$$\widehat{\text{RUL}}_{\text{mean}} = \frac{1}{M} \sum_{i=1}^{M} \text{RUL}^i, \tag{23}$$

$$\widehat{\text{RUL}}_{\text{lower}} = \widehat{\text{RUL}}_{\text{mean}} - t_{M-1,1-\alpha/2} \frac{\widehat{s}}{\sqrt{M}}, \tag{24}$$

$$\widehat{\text{RUL}}_{\text{upper}} = \widehat{\text{RUL}}_{\text{mean}} + t_{M-1,1-\alpha/2} \frac{\widehat{s}}{\sqrt{M}}, \tag{25}$$

where $\widehat{s}$ is the sample standard deviation of the simulated RUL and $t_{M-1,1-\alpha/2}$ is the upper $(1-\alpha/2)$ critical point for the t distribution with $M-1$ degrees of freedom. Details of estimating RUL are summarized in Algorithm 3.

6 | SIMULATION STUDY

In this simulation study, we design the following experiments to evaluate the performance of our proposed sampling method for the HOHSMM with consideration of multiple independent observation sequences.

6.1 | Model setting

Suppose the state space is $\mathscr{C} = \{1,2,3\}$, the emission distribution is Gaussian, that is, $f(y|c_\tau = c) = \text{Normal}(y|\mu_c, \sigma_c^2)$, and the sojourn time at each super-state follows a Poisson distribution $g(d|\xi_c)$. Table 1 summarizes the parameter setting considered in the sensitivity analysis. The total number of experiments is 27.

For each experiment, we independently generate three observation sequences with sample size 2500. The true transition probability tensors λ_{h_1,h_2,h_3} in the third-order cases are generated as follows (Sarkar & Dunson, 2018):

$$\lambda_{h_1,h_2,h_3}(1) = \frac{u_1^2}{u_1^2 + (1-u_1)^2}, \quad u_1 \sim U(0,1),$$

$$\lambda_{h_1,h_2,h_3}(2) = \frac{u_2^2}{u_2^2 + (1-u_2)^2}[1 - \lambda_{h_1,h_2,h_3}(1)], \quad u_2 \sim U(0,1),$$

$$\lambda_{h_1,h_2,h_3}(3) = 1 - \lambda_{h_1,h_2,h_3}(1) - \lambda_{h_1,h_2,h_3}(2),$$

Algorithm 3. RUL estimation

Input: Model parameters $\bar{\lambda}_k$, π_k, $\{\hat{\xi}_c\}$, decoded hidden super-state sequence $\{\tilde{c}_\tau : \tau = 1, \ldots, S\}$, failure state c_F, and the number of simulation paths M.

1: **Compute transition probability matrix**: For each possible combination of $\boldsymbol{c}_{(\tau-q):(\tau-1)}$,compute $p(c_\tau = c|\boldsymbol{c}_{(\tau-q):(\tau-1)})$ using Equation (14), $c = 1, \ldots, C$.
2: **for** $i = 1$ to M **do**
3: **Initialize**: $\boldsymbol{c}_{\text{now}} \leftarrow (\tilde{c}_{S-q+1}, \ldots, \tilde{c}_S)$, $RUL^i \leftarrow 0$.
4: **while** $\boldsymbol{c}_{\text{now}}(q) \neq c_F$ **do**
5: Sample $c \sim \text{Mult}(\{1, \ldots, C\}, p(c = 1|\boldsymbol{c}_{\text{now}}), \ldots, p(c = C|\boldsymbol{c}_{\text{now}}))$.
6: Sample $d \sim g(d|\hat{\xi}_c)$.
7: $RUL^i \leftarrow RUL^i + d$.
8: $\boldsymbol{c}_{\text{now}}(1 : (q-1)) \leftarrow \boldsymbol{c}_{\text{now}}(2 : q)$ and $\boldsymbol{c}_{\text{now}}(q) \leftarrow c$.
9: **end while**
10: **end for**
11: **Compute mean RUL**: $\widehat{\text{RUL}}_{\text{mean}} = \frac{1}{M}\sum_{i=1}^{M} \text{RUL}^i$.
12: **Compute confidence interval** $(\widehat{\text{RUL}}_{\text{lower}}, \widehat{\text{RUL}}_{\text{upper}})$ using Equations (25) and (24).

Output: $\widehat{\text{RUL}}_{\text{mean}}$, $\widehat{\text{RUL}}_{\text{lower}}$, $\widehat{\text{RUL}}_{\text{upper}}$.

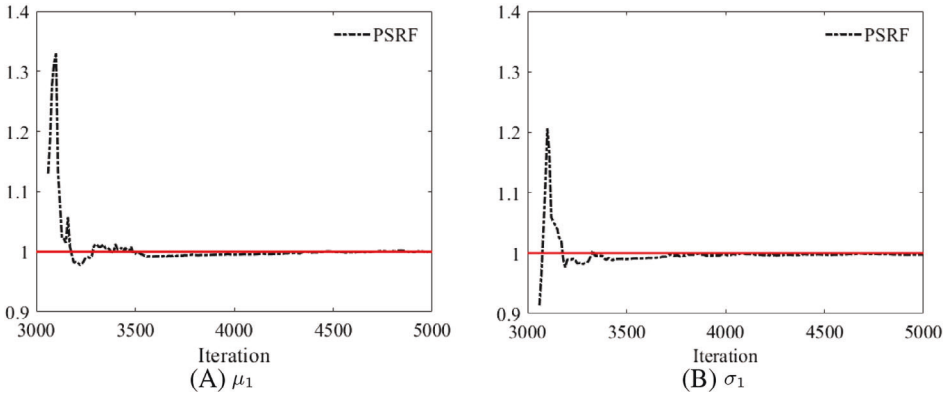

FIGURE 7 Convergence diagnostics: potential scale reduction factor (PSRF) plots [Colour figure can be viewed at wileyonlinelibrary.com]

where $h_1, h_2, h_3 \in \{1,2,3\}$. By excluding self-transitions, we obtain

$$\bar{\lambda}_{i,h_2,h_3}(j) = \frac{\lambda_{i,h_2,h_3}(j)(1-\delta_{ij})}{1-\lambda_{i,h_2,h_3}(i)}, \quad \delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

The true transition probability tensors of the first and second orders are generated similarly. The hyper parameters in the priors are set as $\alpha_0 = 1$ and $\gamma_j = 1/C = 1/3$ for all j.

6.2 | Results

In each experiment, we run 5000 MCMC iterations and discard the first 3000 iterations as burn-in. The remaining samples are thinned by retaining every 10th sample after burn-in to reduce autocorrelation. We compute the potential scale reduction factor (PSRF) (Gelman & Rubin, 1992) to diagnose the convergence, which is obtained based on normal theory approximations to exact Bayesian posterior inference. Figure 7 presents the convergence diagnostics results by providing two PSRF plots for the emission distribution parameters μ_1 and σ_1. The PSRF plots show good mixing behavior by achieving the statistic's asymptotic value 1. We have checked other model parameters and obtained the same diagnostics results. Therefore, the posterior samples generated by the proposed sampling algorithm have good mixing behavior and have produced stable estimates of the parameters of interest.

We evaluate the performance of our proposed HOHSMM sampling algorithm in terms of important lags inclusion result and hidden-state decoding performance. Table 2 summarizes the important lags inclusion results under different data variances. We observe from Table 2 that all experiments correctly identify the true order when variance is small ($\sigma_c^2 = 0.5^2$). However, more than 50% experiments with large variances fail to identify the important lags. This is because large variances in general cause more overlapping of the observations at different hidden states. The decoding performance is quantified using the Hamming distance between the true and the decoded hidden-state sequences, which is the total number of states that are decoded incorrectly. Table 3 shows the hidden-state decoding results using the normalized Hamming distance. We can see that the hidden-state decoding is satisfactory, especially for observations with small variances. The

TABLE 2 Important lags inclusion results: percentage of experiments that correctly identify the true order

$\sigma_c^2 = 0.5^2$	$\sigma_c^2 = 1^2$	$\sigma_c^2 = 1.5^2$
100%	77.78%	44.44%

TABLE 3 Hidden-state decoding results

True dynamics	Data variance	Normalized hamming distance (%)
Third	$\sigma_c^2 = 0.5^2$	0.10
	$\sigma_c^2 = 1^2$	3.69
	$\sigma_c^2 = 1.5^2$	11.52
Second	$\sigma_c^2 = 0.5^2$	0.11
	$\sigma_c^2 = 1^2$	4.17
	$\sigma_c^2 = 1.5^2$	12.03
First	$\sigma_c^2 = 0.5^2$	0.11
	$\sigma_c^2 = 1^2$	4.17
	$\sigma_c^2 = 1.5^2$	11.76

proposed sampling method is effective for HOHSMM inference in the simulation study, including the first order as a special case.

To examine the impacts of the distant-history dependency, we evaluate the performance of our proposed HOHSMM and the HSMM, which makes the strict first-order assumption for the hidden super-state transitions. Specifically, we compare the estimated one-, two-, and three-step ahead predictive densities of the two methods. We arbitrarily select some experiment settings in the sensivity analysis for illustration. We generate observation sequences from Gaussian distributions with $(\mu_1, \mu_2, \mu_3) = (-4.5,0,4.5)$ and consider three variances $\sigma_c^2 = 0.5^2$, $\sigma_c^2 = 1^2$, and $\sigma_c^2 = 1.5^2$ for $c = 1,2,3$. The state duration follows a Poisson distribution with $\xi_c = 22,18,14$ for $c = 1,2,3$, respectively. Three dependencies are considered: third, second, and first order.

For an HOHSMM of order q, the r-step ahead predictive density $f_{\text{pred},S+r}(y|\boldsymbol{y}_{1:S})$ is given by Equation (15). Based on M samples $\{(\boldsymbol{c}^{(m)}, \boldsymbol{\zeta}^{(m)})\}_{m=1}^{M}$ drawn from the posterior, $f_{\text{pred},S+r}(y|\boldsymbol{y}_{1:S})$ can be estimated as

$$\widehat{f}_{\text{pred},S+r}(y|\boldsymbol{y}_{1:S}) = \frac{1}{M}\sum_{m=1}^{M}\sum_{c_{S+r}}\cdots\sum_{c_{S+1}} f(y|c_{S+r}, \boldsymbol{\zeta}^{(m)}) \, p(c_{S+r}|\boldsymbol{c}^{(m)}_{(S+r-q):(S+r-1)}, \boldsymbol{\zeta}^{(m)})\cdots p(c_{S+1}|\boldsymbol{c}^{(m)}_{(S+1-q):S}, \boldsymbol{\zeta}^{(m)}). \tag{26}$$

The corresponding true density, denoted by $f^0_{\text{pred},S+r}(y)$, is obtained with true transition and emission distributions and true hidden-state sequence. The integrated squared error (ISE) (Sarkar & Dunson, 2018) is used to evaluated the density prediction performance, which is estimated by

$$\sum_{i=1}^{N}[f^0_{\text{pred},S+r}(y_i^{\Delta}) - \widehat{f}_{\text{pred},S+r}(y_i^{\Delta}|\boldsymbol{y}_{1:S})]^2\Delta_i, \tag{27}$$

where $\{y_i^{\Delta}\}_{i=1}^{N}$ are a set of grid points on the range of y and $\Delta_i = y_i^{\Delta} - y_{i-1}^{\Delta}$ for all i. For the first-order HSMM, the ISE is estimated similarly by setting $q = 1$. Table 4 summarizes the density prediction results of the proposed HOHSMM and the HSMM given different data variances. From Table 4, we can see that ignoring the distant-history dependency generally leads to larger average ISEs in estimating one-, two-, and three-step ahead predictive densities when higher-order transition dynamics present (ie, third- and second-order dependency), which shows the necessity of taking the higher-order dependency into consideration. We do not observe much differences in ISEs estimated from the HOHSMM and the HSMM when larger data variance presents. This is similarly

TABLE 4 Average ISEs in estimating one-, two-, and three-step ahead predictive densities for the HOHSMM and the HSMM given different data variances

	Average ISE ×100					
	HSMM			HOHSMM		
True dynamics	One	Two	Three	One	Two	Three
$\sigma_c^2 = 0.5^2$						
Third	4.36	7.05	5.60	3.98	4.09	3.14
Second	9.92	5.10	7.97	3.73	4.72	3.55
First	3.62	3.36	3.25	3.59	3.36	3.18
$\sigma_c^2 = 1^2$						
Third	5.63	4.60	6.47	3.58	3.47	5.42
Second	9.73	9.44	9.40	6.64	7.21	7.01
First	10.83	3.34	8.20	9.55	3.24	7.75
$\sigma_c^2 = 1.5^2$						
Third	9.91	5.58	6.81	10.03	5.15	6.30
Second	9.92	10.48	11.23	10.18	10.43	10.88
First	17.85	7.23	10.82	18.10	5.52	9.67

Abbreviations: HSMM, hidden semi-Markov model; HOHSMM, higher-order hidden semi-Markov model; ISEs, integrated squared errors.

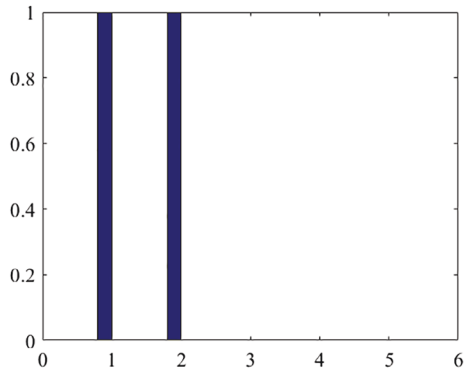

FIGURE 8 The higher-order hidden semi-Markov model (HOHSMM): The inclusion probability given α^* [Colour figure can be viewed at wileyonlinelibrary.com]

because large variances pose challenges in identifying the true order.

7 | CASE STUDY: TURBOFAN ENGINES PROGNOSTICS ANALYSIS

To further demonstrate the practical utility of the proposed HOHSMM on diagnostics and prognostics, we conduct a case study on turbofan engines from the NASA Prognostic Data Repository. The Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) data set is used in this paper, which is generated using a model-based simulation program developed by NASA (Saxena, Goebel, Simon, & Eklund, 2008).

For illustrative purpose, only the training set in data set FD001 is used in this paper, which contains 100 engines' run-to-failure trajectories. All trajectories in this training set are simulated under the same operational condition and have only one fault mode caused by high-pressure compressor degradation (Frederick, DeCastro, & Litt, 2007). Each trajectory is recorded in a given operational cycle, consisting of three values for operational settings and 21 values for engine performance sensor measurements. We randomly choose 10 trajectories to train the HOHSMM and randomly choose another four trajectories for testing.

Multiple sensor measurements bring dimensionality challenge for data analysis. To keep effective discriminant information and eliminate the redundant one, feature fusion process is used to transfer a set of sensors to a single health indicator. To obtain the health indicator, we use principle component analysis (PCA), which is an efficient technique in compressing information and eliminating the correlations between variables. The first principle component (FPC), accounting for the largest variability in data, is used as the health condition indicator (Moghaddass & Zuo, 2014). In the HOHSMM, we assume that the health indicator (ie, FPC) follows a state-specific normal distribution. We assume there are seven health states since it has been shown that the hidden health conditions are well represented by seven states (Moghaddass & Zuo, 2014).

From the important lags inclusion result (shown in Figure 8), we can see that the hidden health-state sequence is governed by a second-order Markov chain, implying that the health-state transition of turbofan engines depends on its past two history states. The performance of hidden-state decoding on training data is illustrated in Figure 9 using the first training trajectory as an example. We compare the state-specific posterior means of FPC and the true FPC computed from raw sensor data. We can see the decoding performance is very good since the estimated emission distributions and the decoded hidden super-state sequence describe the computed FPC well.

Next, we use the learned HOHSMM to predict the RULs for testing units using the simulation method presented in Section 5. Since the degradation in a system is generally not noticeable after the unit has been operated for some period of time, it is reasonable to estimate the RUL using a piece-wise linear function (Heimes, 2008), which limits the maximum value of the RUL. Thus, a piece-wise RUL plot is used to represent the true RUL, which serves as the benchmark for the predicted RUL. First, we compute the FPC for the four testing units based on the PCA results obtained from the training

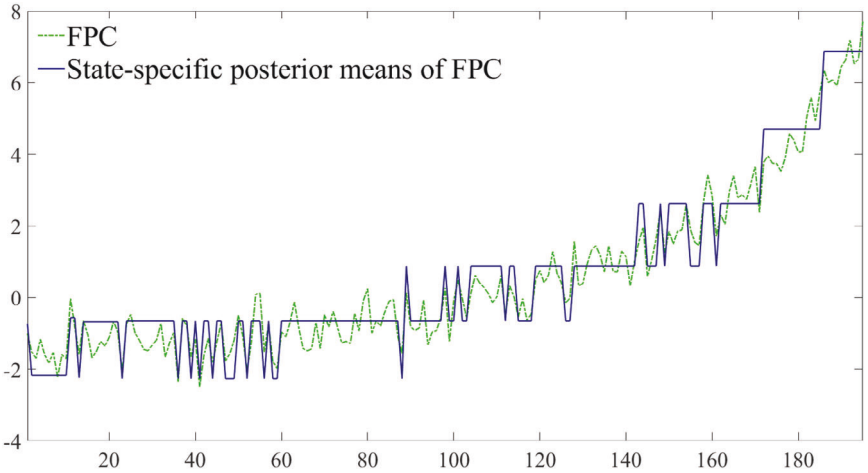

FIGURE 9 The higher-order hidden semi-Markov model (HOHSMM): The state-specific posterior means of first principle component (FPC; blue solid line) super-imposed over the FPC sequence (green dashed line) for training unit 1 [Colour figure can be viewed at wileyonlinelibrary.com]

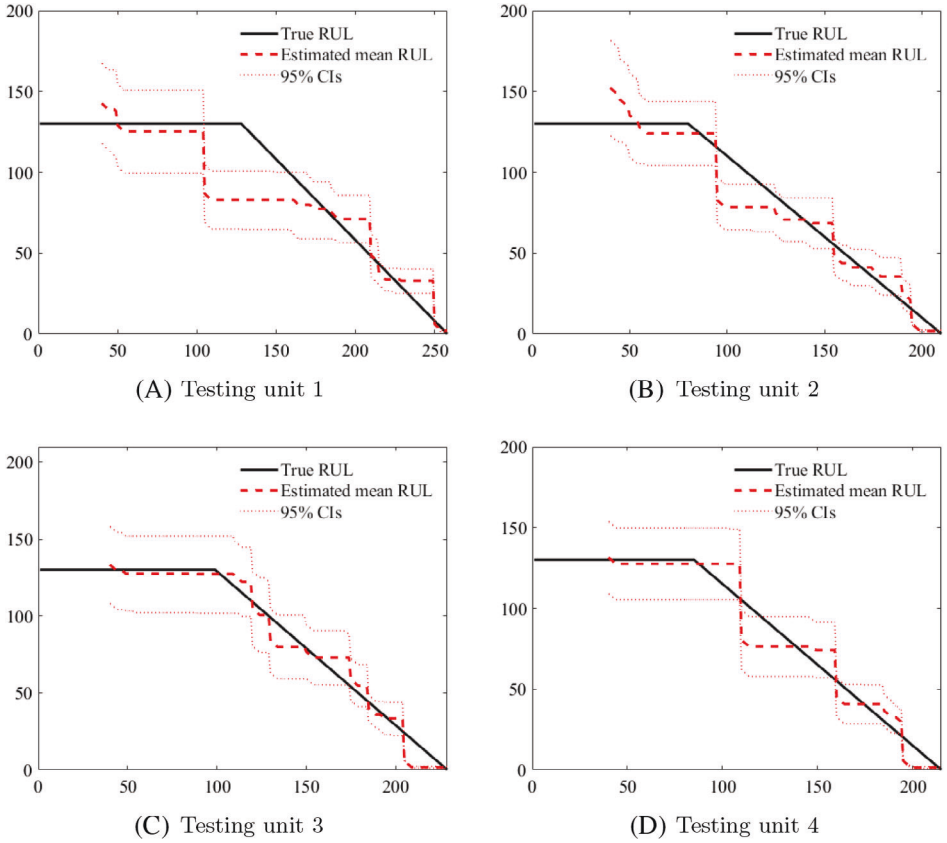

FIGURE 10 The higher-order hidden semi-Markov model (HOHSMM): mean remaining useful life (RUL) prediction for testing units [Colour figure can be viewed at wileyonlinelibrary.com]

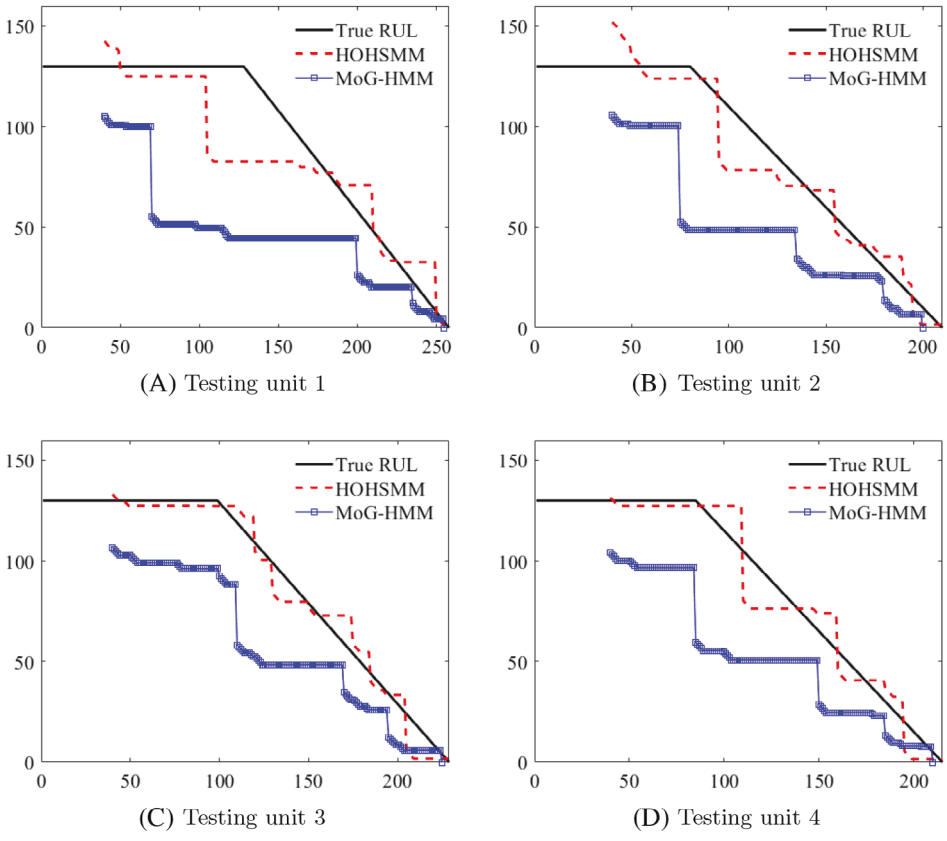

FIGURE 11 Comparison of mean RUL estimations between the HOHSMM and the MoG-HMM. HOHSMM, higher-order hidden semi-Markov model; MoG-HMM, mixture of Gaussians hidden Markov model; RUL, remaining useful life [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 5 Absolute mean RUL estimation errors of the HOHSMM and the MoG-HMM

Model	Unit 1	Unit 2	Unit 3	Unit 4
HOHSMM	7.73	5.47	9.02	14.84
MoG-HMM	31.28	28.77	34.62	45.46

Abbreviations: HOHSMM, higher-order hidden semi-Markov model; MoG-HMM, mixture of Gaussians hidden Markov model; RUL, remaining useful life.

data. The observations (ie, the computed FPC) of each testing unit are continuously fed into the learned HOHSMM, which are used for decoding the current health state and predicting the RUL. By generating 100 paths, we obtain the estimated mean RUL and the respective 95% confidence interval (CI) for each time point, shown in Figure 10. The mean RUL estimation results are very good for the testing units since the true RUL is close to the estimated mean RUL and is within the estimated 95% CIs at the majority of time points.

To further assess the performance of the proposed HOHSMM on the RUL prediction, we compare the performance of the proposed model with that of the mixture of Gaussians HMM (MoG-HMM) in Tobon-Mejia et al. (2012), which has been shown to be efficient in engineering system prognostics. The Baum-Welch algorithm is used to estimate the MoG-HMM parameters and the Viterbi algorithm is used to assess the current health state of the system. The RUL is predicted using simulation approach by generating hidden-state sequences from the current state to the failure state based on the estimated transition probabilities. We use the same training units to train the MoG-HMM and estimate the mean RUL on the same testing units. Figure 11 compares the estimated mean RUL based on the proposed HOHSMM and the MoG-HMM. We can see that the proposed HOHSMM gives better prediction on all four testing units. We further compute the absolute mean estimation errors of the testing units for both models, which are summarized in Table 5. From Table 5, we can see that the absolute mean RUL estimation errors of HOHSMM are smaller than the MoG-HMM for all testing units, indicating that our proposed method is effective in real-world applications.

8 | CONCLUSIONS

In this paper, we consider the problem of decoding the hidden health states and predicting the RUL for systems with unobservable health conditions and complex transition dynamics based on observations. We develop a flexible prognostics framework based on an HOHSMM. Our framework is flexible in that the HOHSMM allows the hidden state to depend on its more distant history instead of only depending on the current state and assumes generally distributed state duration. The proposed HOHSMM includes the HMM and HSMM as two special cases. A Gibbs sampling algorithm is designed for HOHSMM inference and is evaluated by conducting a simulation study. The results show that the proposed HOHSMM sampler is effective for learning model parameters from the observed data and it is necessary to consider distant-history dependency when higher-order transition dynamics present. Given the learned model, a decoding algorithm is developed to assess the current hidden health state of a functioning system in operation. The RUL is then predicted using a simulation approach by generating hidden-state sequences from the current state to the failure state. The NASA turbofan engine data set (ie, C-MAPSS data set) is used to demonstrate the practical utility of the proposed prognostics framework. Our case study shows that the HOHSMM-based prognostics framework provides satisfactory hidden health-state assessment and RUL estimation for complex systems. Furthermore, the comparison on RUL prediction between the proposed HOHSMM and the benchmark MoG-HMM shows that our proposed prognostics framework is effective in real-world applications.

The framework presented in this paper has raised a few important questions that require further study. First, the state space is generally unknown and the true number of states also need to be learned from the observed data. The existing HDP-HMM provides a powerful framework for inferring arbitrarily large state complexity from data (Teh et al., 2006). Moreover, the HDP-HSMM allows for both Bayesian nonparametric inference of state complexity as well as general duration distributions (Johnson & Willsky, 2013). A promising direction for future research is to consider a more general model, the hierarchical Dirichlet process HOHSMM, to address the unknown state space issue in our proposed HOHSMM-based prognostics framework. Second, there generally exists heterogeneity among different operating systems (or components), even in the same environmental conditions. It is also necessary to extend our prognostics framework to account for the unit-to-unit differences in the future work.

ACKNOWLEDGMENT

This work is supported in part by the U.S. National Science Foundation under award 1943985.

ORCID

Yisha Xiang https://orcid.org/0000-0003-0696-2924
Min Wang https://orcid.org/0000-0002-9233-7844

REFERENCES

Acuña, D. E., & Orchard, M. E. (2017). Particle-filtering-based failure prognosis via sigma-points: Application to lithium-ion battery state-of-charge monitoring. *Mechanical Systems and Signal Processing*, *85*, 827–848.

Ando, T. (2010). Bayesian model selection and statistical modeling. Boca Raton, FL: Chapman and Hall/CRC.

Bai, G., & Wang, P. (2016). Prognostics using an adaptive self-cognizant dynamic system approach. *IEEE Transactions on Reliability*, *65*(3), 1427–1437.

Baruah, P., & Chinnam, R. B. (2005). Hmms for diagnostics and prognostics in machining processes. *International Journal of Production Research*, *43*(6), 1275–1293.

Benkedjouh, T., Medjaher, K., Zerhouni, N., & Rechak, S. (2015). Health assessment and life prediction of cutting tools based on support vector regression. *Journal of Intelligent Manufacturing*, *26*(2), 213–223.

Bunks, C., McCarthy, D., & Al-Ani, T. (2000). Condition-based maintenance of machines using hidden Markov models. *Mechanical Systems and Signal Processing*, *14*(4), 597–612.

Camci, F., & Chinnam, R. B. (2010). Health-state estimation and prognostics in machining processes. *IEEE Transactions on Automation Science and Engineering*, *7*(3), 581–597.

Chen, C., Vachtsevanos, G., & Orchard, M. E. (2012). Machine remaining useful life prediction: An integrated adaptive neuro-fuzzy and high-order particle filtering approach. *Mechanical Systems and Signal Processing*, *28*, 597–607.

Cheng, S., Azarian, M. H., & Pecht, M. G. (2010). Sensor systems for prognostics and health management. *Sensors*, *10*(6), 5774–5797.

Chiachío, J., Chiachío, M., Sankararaman, S., Saxena, A., & Goebel, K. (2015). Condition-based prediction of time-dependent reliability in composites. *Reliability Engineering & System Safety*, *142*, 134–147.

Daigle, M. J., & Goebel, K. (2012). Model-based prognostics with concurrent damage progression processes. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *43*(3), 535–546.

Di Maio, F., Tsui, K. L., & Zio, E. (2012). Combining relevance vector machines and exponential regression for bearing residual life estimation. *Mechanical Systems and Signal Processing*, *31*, 405–427.

Dong, M., & He, D. (2007). A segmental hidden semi-Markov model (HSMM)-based diagnostics and prognostics framework and methodology. *Mechanical Systems and Signal Processing*, *21*(5), 2248–2266.

Dong, S., & Luo, T. (2013). Bearing degradation process prediction based on the PCA and optimized LS-SVM model. *Measurement*, *46*(9), 3143–3152.

D. K. Frederick, J. A. DeCastro, and J. S. Litt, "User's guide for the commercial modular aero-propulsion system simulation (C-MAPSS)," 2007.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, *7*(4), 457–472.

George, E. I., & McCulloch, R. E. (1997). Approaches for Bayesian variable selection. *Statistica Sinica*, *7*, 339–373.

Giantomassi, A., Ferracuti, F., Benini, A., Ippoliti, G., Longhi, S., & Petrucci, A. (2011). *Hidden Markov model for health estimation and prognosis of turbofan engines*. In ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (pp. 681–689). Washington, DC: American Society of Mechanical Engineers Digital Collection.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, *82*(4), 711–732.

Guo, L., Li, N., Jia, F., Lei, Y., & Lin, J. (2017). A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing*, *240*, 98–109.

Haile, M. A., Riddick, J. C., & Assefa, A. H. (2016). Robust particle filters for fatigue crack growth estimation in rotorcraft structures. *IEEE Transactions on Reliability*, *65*(3), 1438–1448.

Heimes, F. O. (2008). *Recurrent neural networks for remaining useful life estimation*. In 2008 International Conference on Prognostics and Health Management (pp. 1–6). Denver, CO: IEEE.

Huang, R., Xi, L., Li, X., Liu, C. R., Qiu, H., & Lee, J. (2007). Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods. *Mechanical Systems and Signal Processing*, *21*(1), 193–207.

Johnson, M. J., & Willsky, A. S. (2013). Bayesian nonparametric hidden semi-Markov models. *Journal of Machine Learning Research*, *14*, 673–701.

Kandukuri, S. T., Klausen, A., Karimi, H. R., & Robbersmyr, K. G. (2016). A review of diagnostics and prognostics of low-speed machinery towards wind turbine farm-level health management. *Renewable and Sustainable Energy Reviews*, *53*, 697–708.

Khelif, R., Chebel-Morello, B., Malinowski, S., Laajili, E., Fnaiech, F., & Zerhouni, N. (2016). Direct remaining useful life estimation based on support vector regression. *IEEE Transactions on Industrial Electronics*, *64*(3), 2276–2285.

Lei, Y., Li, N., Gontarz, S., Lin, J., Radkowski, S., & Dybala, J. (2016). A model-based method for remaining useful life prediction of machinery. *IEEE Transactions on Reliability*, *65*(3), 1314–1326.

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, *104*, 799–834.

Li, X., Ding, Q., & Sun, J.-Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, *172*, 1–11.

Li, Y., Billington, S., Zhang, C., Kurfess, T., Danyluk, S., & Liang, S. (1999). Adaptive prognostics for rolling element bearing condition. *Mechanical Systems and Signal Processing*, *13*(1), 103–113.

Li, Y., Kurfess, T., & Liang, S. (2000). Stochastic prognostics for rolling element bearings. *Mechanical Systems and Signal Processing*, *14*(5), 747–762.

Liao, L. (2013). Discovering prognostic features using genetic programming in remaining useful life prediction. *IEEE Transactions on Industrial Electronics*, *61*(5), 2464–2472.

Liu, J., Vitelli, V., Zio, E., & Seraoui, R. (2015). A novel dynamic-weighted probabilistic support vector regression-based ensemble for prognostics of time series data. *IEEE Transactions on Reliability*, *64*(4), 1203–1213.

Liu, J., Wang, W., Ma, F., Yang, Y., & Yang, C. (2012). A data-model-fusion prognostic framework for dynamic system state forecasting. *Engineering Applications of Artificial Intelligence*, *25*(4), 814–823.

Liu, Q., Dong, M., Lv, W., Geng, X., & Li, Y. (2015). A novel method using adaptive hidden semi-Markov model for multi-sensor monitoring equipment health prognosis. *Mechanical Systems and Signal Processing*, *64*, 217–232.

Liu, T., Zhu, K., & Zeng, L. (2018). Diagnosis and prognosis of degradation process via hidden semi-Markov model. *IEEE/ASME Transactions on Mechatronics*, *23*(3), 1456–1466.

Malhi, A., Yan, R., & Gao, R. X. (2011). Prognosis of defect propagation based on recurrent neural networks. *IEEE Transactions on Instrumentation and Measurement*, *60*(3), 703–711.

Moghaddass, R., & Zuo, M. J. (2014). An integrated framework for online diagnostic and prognostic health monitoring using a multistate deterioration process. *Reliability Engineering & System Safety*, *124*, 92–104.

Myötyri, E., Pulkkinen, U., & Simola, K. (2006). Application of stochastic filtering for lifetime prediction. *Reliability Engineering & System Safety*, *91*(2), 200–208.

Pecht, M. (2008). Prognostics and Health Management of Electronics. New Jersey: John Wiley.

Peel, L. (2008). *Data driven prognostics using a kalman filter ensemble of neural network models*. In 2008 International Conference on Prognostics and Health Management (pp. 1–6). Denver, CO: IEEE.

Qian, Y., Yan, R., & Gao, R. X. (2017). A multi-time scale approach to remaining useful life prediction in rolling bearing. *Mechanical Systems and Signal Processing*, *83*, 549–567.

Qiu, J., Seth, B. B., Liang, S. Y., & Zhang, C. (2002). Damage mechanics approach for bearing lifetime prognostics. *Mechanical Systems and Signal Processing*, *16*(5), 817–829.

Rouet, V., Minault, F., Diancourt, G., & Foucher, B. (2007). Concept of smart integrated life consumption monitoring system for electronics. *Microelectronics Reliability*, *47*(12), 1921–1927.

Saha, B., Goebel, K., Poll, S., & Christophersen, J. (2008). Prognostics methods for battery health monitoring using a Bayesian framework. *IEEE Transactions on Instrumentation and Measurement*, *58*(2), 291–296.

Sarkar, A., & Dunson, D. B. (2018). Bayesian nonparametric higher order hidden Markov models. *arXiv Preprint*, arXiv:1805.12201.

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). *Damage propagation modeling for aircraft engine run-to-failure simulation*. In 2008 International Conference on Prognostics and Health Management (pp. 1–9). Denver, CO: IEEE.

Sun, B., Zeng, S., Kang, R., & Pecht, M. G. (2012). Benefits and challenges of system prognostics. *IEEE Transactions on Reliability*, *61*(2), 323–335.

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, *101*(476), 1566–1581.

Tian, Z. (2012). An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring. *Journal of Intelligent Manufacturing*, *23*(2), 227–237.

Tobon-Mejia, D. A., Medjaher, K., Zerhouni, N., & Tripot, G. (2012). A data-driven failure prognostics method based on mixture of Gaussians hidden Markov models. *IEEE Transactions on Reliability*, *61*(2), 491–503.

Wang, N., Sun, S.-D., Cai, Z.-Q., Zhang, S., & Saygin, C. (2014). A hidden semi-Markov model with duration-dependent state transition probabilities for prognostics. *Mathematical Problems in Engineering*, *2014*, 632702.

Wang, W. (2007). An adaptive predictor for dynamic system forecasting. *Mechanical Systems and Signal Processing*, *21*(2), 809–823.

Wang, W., & Zhang, W. (2005). A model to predict the residual life of aircraft engines based upon oil analysis data. *Naval Research Logistics (NRL)*, *52*(3), 276–284.

Wang, W. Q., Golnaraghi, M. F., & Ismail, F. (2004). Prognosis of machine health condition using neuro-fuzzy systems. *Mechanical Systems and Signal Processing*, *18*(4), 813–831.

Widodo, A., & Yang, B.-S. (2011). Machine health prognostics using survival probability and support vector machine. *Expert Systems with Applications*, *38*(7), 8430–8437.

Xiao, Q., Fang, Y., Liu, Q., & Zhou, S. (2018). Online machine health prognostics based on modified duration-dependent hidden semi-Markov model and high-order particle filtering. *The International Journal of Advanced Manufacturing Technology*, *94*(1–4), 1283–1297.

Yang, Y., & Dunson, D. B. (2016). Bayesian conditional tensor factorizations for high-dimensional classification. *Journal of the American Statistical Association*, *111*(514), 656–669.

Yu, S.-Z. (2010). Hidden semi-Markov models. *Artificial Intelligence*, *174*(2), 215–243.

Zaidan, M. A., Mills, A. R., Harrison, R. F., & Fleming, P. J. (2016). Gas turbine engine prognostics using Bayesian hierarchical models: A variational approach. *Mechanical Systems and Signal Processing*, *70*, 120–140.

Zhang, J., & Lee, J. (2011). A review on prognostics and health monitoring of Li-ion battery. *Journal of Power Sources*, *196*(15), 6007–6014.

Zheng, S., Ristovski, K., Farahat, A., & Gupta, C. (2017). *Long short-term memory network for remaining useful life estimation*. In 2017 IEEE International Conference on Prognostics and Health Management (ICPHM) (pp. 88–95). Dallas, TX: IEEE.

How to cite this article: Liao Y, Xiang Y, Wang M. Health assessment and prognostics based on higher-order hidden semi-Markov models. *Naval Research Logistics* 2021;68:259–276. https://doi.org/10.1002/nav.21947