

# Semi-supervised Constrained Hidden Markov Model Using Multiple Sensors for Remaining Useful Life Prediction and Optimal Predictive Maintenance

Xinyu Zhao<sup>1</sup>, Yunyi Kang<sup>2</sup>, Hao Yan<sup>3</sup>, Feng Ju<sup>4</sup>

<sup>1,2,3,4</sup> *School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, 85281, USA*

*xzhao119@asu.edu*

*ykang37@asu.edu*

*haoyan@asu.edu*

*fengju@asu.edu*

## ABSTRACT

Remaining Useful Life (RUL) estimation is critical in many engineering systems where proper predictive maintenance is needed to increase a unit's effectiveness and reduce time and cost of repairing. Typically for such systems, multiple sensors are normally used to monitor performance, which create difficulties for system state identification. In this paper, we propose a semi-supervised left-to-right constrained Hidden Markov Model (HMM) model, which is the first in the literature to simultaneously address the challenges of semi-supervised setting, left-to-right constraint, and monotonicity constraint in a multiple-sensor setting. This proposed method is also effective in estimating the RUL while capturing the jumps among states in condition dynamics. In addition, based on the HMM model learned from multiple sensors, we build a Partial Observable Markov Decision Process (POMDP) to demonstrate how such RUL estimation can be effectively used for optimal preventative maintenance decision making. We apply this technique to the NASA Engine degradation data and demonstrate the effectiveness of the proposed method.

## 1. INTRODUCTION

Machine and equipment monitoring data is increasingly collected and widely used for reliability assessment, lifetime prediction, and operational decision making. Through gathering and utilizing these information, i.e., the degradation signals from multiple sensors, the essential goal of such studies is to estimate the health status and make inference about the remaining useful life (RUL) of a unit accurately (Nelson, 2009). With additional information on system operations,

the conditional-based maintenance strategy (Mobley, 2002; Kang & Ju, 2019) can be generated and applied, which can effectively prevent unexpected failure, significantly improve process safety and ultimately reduce the overall operation cost (Caesarendra, Widodo, Thom, Yang, & Setiawan, 2011; Tobon-Mejia, Medjaher, Zerhouni, & Tripot, 2012).

In literature, many research works have focused on degradation modeling based on the continuous system state, such as state space model (Sun, Zuo, Wang, & Pecht, 2014), degradation path model (Lu & Meeker, 1993), and health-index-based approach (Yan, Liu, Zhang, & Shi, 2016). However, for many degradation systems, the hidden dynamics is not continuous and present many jumps between states. Therefore, in order to characterize the system dynamics of those systems more efficiently, the discrete system state representation is normally used. Hidden Markov model (HMM) is a discrete state transition model, which assumes that the transition follows the Markovian property. In literature, inspired from the introductory work by (Rabiner, 1989), the research and application of HMM on RUL estimation has received heavy attention in the area of machinery diagnostics and prognostics, i.e., predicting the fault time of machines, tools, rolling bearings, etc (Soualhi, Clerc, Razik, Guillet, et al., 2016; Boutros & Liang, 2011; T. Liu, Chen, & Dong, 2014). HMM models enable capturing the condition dynamics using a finite number of the latent states, which are trained by the ordered and timed observations, and show great explanation power onto the processes with complicated state transitions. For many of the HMM models, the structure is simple, typically with a single layer (Athanasopoulou, Li, & Hadjicostis, 2010; Geramifard, Xu, Zhou, & Li, 2014). Furthermore, to improve the flexibility of HMM, improvements have also been proposed, such as the two-layer model (Camci & Chinnam, 2010), factorial hidden Markov model (Le, Chatelain, & Bérenguer, 2016), hidden semi-Markov model (Q. Liu, Dong, Lv, Geng, & Li,

---

Xinyu Zhao et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2015), etc. The variety of the extensions enriches the applicability and prediction accuracy in complicated scenarios.

Furthermore, to improve the system performance, methods to generate maintenance policies are discussed, focusing on different benchmarks, such as average operating time (Klutke & Yang, 2002), machine cycle time (Liao, Elsayed, & Chan, 2006), operating costs (Liyanage & Kumar, 2003), and energy consumption (Luo, Yan, Hu, Zhou, & Pang, 2015). Among the different maintenance practice, preventative maintenance (PM) is commonly discussed, which aims at retaining a system into certain conditions when it is still operating (Alaswad & Xiang, 2017; F. Wang, Lu, & Ju, 2018). Following earlier works such as (Barlow & Hunter, 1960), many models are developed to supervise the decision-making procedures, such as determining proper maintenance intervals, optimizing consumption on the spare parts, and minimizing related operating costs (W. Wang, 2012). For example, paper (Douer & Yechiali, 1994) studies the PM actions for a system with single machines. Paper (Keizer, Flapper, & Teunter, 2017) focuses on the system with multiple machines and complex structures considering the machine condition variations. Furthermore, paper (de Jonge, Teunter, & Tinga, 2017) takes the survey and feedback from different industries on the benefit of conditional-based PM techniques and applications. Extended reviews for the models generating replacement, repair, and inspection policies are discussed in (Jardine, Lin, & Banjevic, 2006) and (H. Wang, 2002).

Despite these research efforts, there are still several challenges to be addressed. 1) Due to the vast advances in sensing and computing technology, *multiple sensors* have been widely used to simultaneously monitor the health status of a unit. These sensors are typically strongly correlated and each only contains partial information of the degrading unit. 2) The system state is generally *not fully observable*. Some states such as the initial state and the failure state are typically observable but not some of the intermediate states. 3) There are typically certain constraints in the system dynamics, for example, the system degradation is *one-directional* and the degradation signal is *monotone* without the maintenance (e.g., see Fig. 5 for an example), which implies that the system state transition is typically one-directional and follows a particular order. 4) There is no unified framework to use the multi-sensor degradation signal directly with system maintenance policy.

To address the aforementioned challenges, we propose a semi-supervised left-to-right constrained HMM for RUL prediction using multiple sensors. The use of semi-supervised learning framework ensures that the proposed method can efficiently combine information from the observable and non-observable states. Imposing both the left-to-right constraint and monotonicity constraint is important to degradation systems and can also dramatically stabilize the training for multi-

sensor systems. Even though the left-to-right constraint has been addressed in the literature, however, an HMM model combining with the monotonicity constraint in a multiple sensor setting has been not studied before. We also derive the RUL distribution under the proposed framework. These research efforts provide fundamental support to future machine performance improvement and operation control, such as working mode control and preventative maintenance.

The rest of the paper is organized as follows: Section II reviews the literature. Section III develops the proposed framework and details the estimation and decision algorithms. Section IV demonstrates the effectiveness of the proposed methodology based on the degradation dataset for aircraft gas turbine engines in (Saxena & Goebel, 2008) that contains 21 sensor signals. Finally, Section V provides a conclusion and discussion of future directions.

## 2. PROBLEM FORMULATION

In this work, we propose a decision-analytic framework from the degradation data. The proposed methodology contains two steps: 1) we are able to learn the system dynamics and then estimate the remaining useful life of the system based on a semi-supervised constrained Hidden Markov Model. 2) The learned system dynamics enable us to maintain the system appropriately to increase the lifetime of the system based on the estimated hidden states in the POMDP framework.

### 2.1. Hidden Markov Model

HMM is a probabilistic model which describes the transitions of a finite number of states over time. These states characterize two stochastic processes: the hidden state transition in discrete time and an observed process. In addition, three sets of probability distributions are utilized to characterize the system dynamics: the initial probabilities for all the hidden states; the transition probabilities between two hidden states and the emission probabilities of an observation from a hidden state. The elements of an HMM are defined as follows:

- Let  $\mathbf{Q} = \{1, \dots, K\}$  denote the set of hidden states, with the total number of states to be  $K$ . We denote  $S_{i,t} \in \mathbf{Q}$  as the system state at time  $t$  for unit  $i \in \{1, \dots, n\}$ . Furthermore, we denote  $S_{i,t}^k$  as a binary variable according to whether the system for unit  $i$  at state  $k$  in time  $t$ .
- A transition probability matrix is defined as  $\mathbf{A} = \{p_{kk'}\}$ , in which  $p_{kk'}$  is the transition probability that the system state is changed from state  $k$  to state  $k'$ ,  $k, k' \in \{1, \dots, K\}$  at time  $t$ ,

$$p_{kk'} = P(S_{i,t+1} = k | S_{i,t} = k'), 1 \leq k, k' \leq K. \quad (1)$$

- An emission probability is defined on multi-variate response  $\mathbf{O}_{i,t}$  by two sets of matrix  $\mathbf{B} = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ , in which  $\boldsymbol{\mu}_k$  is the mean of multiple sensors in state  $k$ ,

and  $\Sigma_k$  denote the covariance matrix of multiple sensing measurement. In another word,

$$(\mathbf{O}_{i,t} | S_{i,t} = k) \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad (2)$$

$$i = 1, \dots, n, k = 1, \dots, K.$$

- An initial state vector  $\boldsymbol{\Pi}_i = \{\pi_{i,k}\}$  is defined to express the distribution of the system states for unit  $i$  at state  $k$  at  $t = 1$ .

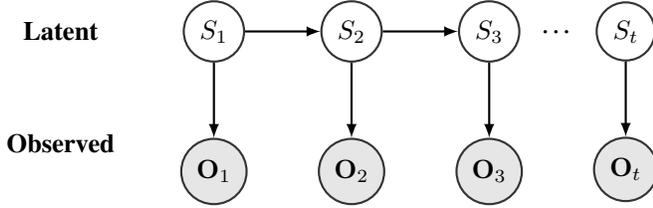


Figure 1. Illustrative picture of HMM

With the descriptions shown above, for the simplicity and clarification, the full HMM model is given by  $\Lambda = \{\mathbf{Q}, \mathbf{A}, \mathbf{B}, \boldsymbol{\Pi}\}$ . Figure 1 shows the logical dependencies among different random variables of an HMM. Furthermore, three major problems can be solved by the HMM (Rabiner, 1989):

- *Evaluation problem*- Given an HMM  $\Lambda$  and an observation sequence  $\{\mathbf{O}_{i,t}\}_{t=1}^T = (\mathbf{O}_{i,1}, \mathbf{O}_{i,2}, \dots, \mathbf{O}_{i,T})$ , find the probability that the sequence  $\{\mathbf{O}_{i,t}\}_{t=1}^T$  is accrued, or  $P(\{\mathbf{O}_{i,t}\}_{t=1}^T | \Lambda)$ , using the modeling. Such a problem is also called the evaluation problem. Specifically in fault prognostic, this problem is considered as failure detection.
- *Decoding problem*- Given an HMM  $\Lambda$  and an observation sequence  $\{\mathbf{O}_{i,t}\}_{t=1}^T$ , identify the system state at each time  $t$  or estimate  $P(S_{i,t} | \{\mathbf{O}_{i,t}\}_{t=1}^T)$ .
- *Learning problem*- Find the parameters in the model  $\Lambda$  (the  $\mathbf{A}, \mathbf{B}, \boldsymbol{\Pi}$ ) that maximize the probability  $P(\{\mathbf{O}_{i,t}\}_{t=1}^T | \mathbf{A}, \mathbf{B}, \boldsymbol{\Pi})$ .

## 2.2. Semi-supervised Left-to-right constrained Hidden Markov Model for Prognostics modeling

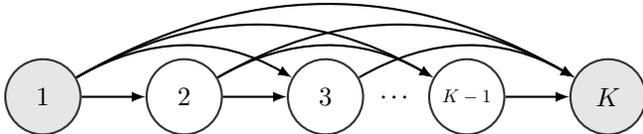


Figure 2. One directional Semi-supervised State Transition, Initial 1 and failure state  $K$  are typically observable

To model the system degradation, we assume that the system can only degrade to worse state over time, which means that

the transition is only one-directional. Therefore, we assume that the transition matrix is left-to-right only. Finally, we assume that the failure state is the only absorbing state in the system.

Commonly, EM algorithms are used for finding the maximum likelihood estimate of the parameters of a hidden Markov model given a set of observed feature vectors. However, the traditional EM algorithm is not able to provide the transition matrix with left-to-right constraint. Furthermore, the naive implementation of constraint such as mask out the transition matrix does not lead to robust estimation of the transition matrix and emission parameters. Here we propose to modify the EM algorithm of the HMM learning based on the left-to-right constraint and the monotonicity constraint in the multiple-sensor setting, which has not been addressed before. Due to the fact that the signal is mostly monotone, in each iteration, we rank the posterior mean  $\boldsymbol{\mu}_1 < \dots < \boldsymbol{\mu}_k$ , where  $k$  is the number of states.

## 2.3. Estimation of the constrained Hidden Markov Model

In this section, we will introduce the notation and framework of the HMM. The random variable  $\mathbf{O}_{i,t}$  denotes the  $i$ th signal observation at time  $t$  and  $S_{i,t}$  is the hidden state which denotes the health condition for the systems under consideration. The following is the joint likelihood of the problem.

$$P(\{\mathbf{O}_{i,t}\}_{i,t}) = \sum_{i=1}^N \sum_{s_1, \dots, s_T} P(\{\mathbf{O}_{i,t}\}_{t=1}^T, \{S_{i,t}\}_{t=1}^T)$$

$$= \sum_{i=1}^N \sum_{s_1, \dots, s_T} P(S_{i,1}) \prod_{t=2}^T P(S_{i,t} | S_{i,t-1}) \prod_{t=1}^T P(\mathbf{O}_{i,t} | S_{i,t}) \quad (3)$$

However, the joint likelihood is not tractable. The inference can be broken down into the forward-backward algorithm and the Baum-Welch Algorithm.

The goal of the forward-backward algorithm is to decode the system state given the degradation signals in (4). The step of Forward-backward algorithm of the proposed method is the same with ordinary HMM. For more details, please see (Rabiner, 1989).

$$\gamma_t^k = P(S_t^k = 1 | \{\mathbf{O}_t\}_{t=1}^T) \quad (4)$$

However, the original Baum-Welch algorithm cannot be used for the semi-supervised setting with the monotonicity constraint. Therefore, we proposed to modify the Baum-Welch algorithm to address the semi-supervised challenge, multiple sensors challenge, and the monotonicity state constraint as follows:

1) Addressing the challenges of *semi-supervised state* and *multiple sensors*. The starting state and failure state are typically observable. For these states, we can directly estimate the mean and covariance matrix from the multiple sensor measurements. The mean  $\boldsymbol{\mu}_k$  and covariance matrix  $\boldsymbol{\Sigma}_k$  of observed state  $k$  can be estimated as:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^n \sum_{t=1}^T S_{i,t}^k \mathbf{O}_{i,t}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^n \sum_{t=1}^T S_{i,t}^k (\mathbf{O}_{i,t} - \boldsymbol{\mu}_k)(\mathbf{O}_{i,t} - \boldsymbol{\mu}_k)^T$$

$$\text{where, } N_k = \sum_{i=1}^n \sum_{t=1}^T S_{i,t}^k. \quad (5)$$

For the unobserved system states, we first use the forward-backward algorithm to estimate the posterior probability of unit  $i$  in state  $k$  and time  $t$  as  $\gamma_{i,k}(t) = P(S_{i,t}^k = 1 | \{\mathbf{O}_t\}_{t=1}^T)$ . The mean and covariance matrix for each state  $k$  can be estimated as follows:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^n \sum_{t=1}^T \gamma_{i,k}(t) \mathbf{O}_{i,t}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^n \sum_{t=1}^T \gamma_{i,k}(t) (\mathbf{O}_{i,t} - \boldsymbol{\mu}_k)(\mathbf{O}_{i,t} - \boldsymbol{\mu}_k)^T$$

$$\text{where, } N_k = \sum_{i=1}^n \sum_{t=1}^T \gamma_{i,k}(t). \quad (6)$$

2) Addressing the challenges of *left-to-right state transition* and *monoton degradation signals*. We have the prior knowledge that  $\mu_{k'j} > \mu_{kj}$  should be larger for system state  $k' > k$  for some sensor  $j$ , and  $\mu_{k'j} < \mu_{kj}$  should be smaller for system state  $k' > k$  for the other sensor  $j$ . We propose to sort the  $(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k) = \text{sort}(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k)$  according to the majority voting procedure.

Finally, the initial probability is  $\pi_{ik} = \gamma_{ik}(1)$  and the state transition matrix can be estimated in the following:

$$\tilde{p}_{kk'} = \frac{\sum_{t=1}^{T-1} \xi_{kk'}(t)}{\sum_{t=1}^{T-1} \gamma_k(t)} \quad (7)$$

$$\text{where, } \xi_{kk'}(t) = P(S_{t-1} = k, S_t = k' | \mathbf{O}; \theta)$$

However, the estimated  $p_{kk'}$  are without any constraint. To impose the left-to-right constraint, we propose to perform the

following update step.

$$p_{kk'} = \begin{cases} 0 & k < k' \\ \sum_{k=k'}^K \tilde{p}_{kk'} & k \geq k' \end{cases} \quad (8)$$

$$(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k) = \text{sort}(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k) \quad (9)$$

Through our simulation study, the combined sorting (9) and projection (8) to lower triangular matrix could dramatically increase the robust of the training procedure than using the projection (8) alone.

The overall algorithm is summarized as follows:

---

**Algorithm 1:** Constrained HMM Fitting

---

**Input:** Number of state  $K$ , Multi-sensor observation  $\mathbf{O}_{i,t}$

**Output:** Transition  $P_{ij}$ , emission  $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$   
initialization;

Forward-backward algorithm and update state posterior distribution.

Estimate observed state emission using (5)

**while** *Not converged* **do**

    Estimate unobserved state emission using (6)

    Estimate transition (7)

    Projection and state sorting (4)

**end**

---

#### 2.4. Deriving the Remaining Useful Life Distribution

To estimate the RUL, we first need to find out the RUL for each state  $k$  as  $T_k$ . To achieve that we propose to simulate the Markov Chain starting from state  $k$  according to the estimated transition matrix  $p_{ij}$ . Finally, based on the Monte Carlo simulation results, we are able to find the empirical distribution for  $T_k$ . And then we can estimate the RUL with the posterior distribution. More specifically, the RUL for unit  $i$  at time  $t$  is denoted as  $RUL_i(t)$ . To estimate the RUL, we first need to estimate the posterior distribution of system state, denoted as  $\gamma_{i,k}(t)$  from the forward-backward algorithm. Therefore, the RUL is a mixture distribution of  $T_k$  given the system state distribution as

$$RUL_i(t) = \sum \gamma_{i,k}(t) T_k. \quad (10)$$

### 3. PREVENTATIVE MAINTENANCE DECISION USING POMDP

Given the RUL prediction, the next problem is to find the optimal PM policies to maximize the long-term engine performance, considering the dynamics of the engine conditions. To solve the problem, we develop a Partial Observable Markov Decision Process (POMDP) model based on the learned HMM from the previous section. This section details the model and the solution approach.

First of all, we make the following assumptions to describe the maintenance and repair operations:

- For any operating state, the engine can operate normally in each cycle. The transition rules among the machine states are as follows (shown in Figure 3). If no maintenance is performed (as the solid arrow shows), for a machine working at state  $i$ ,  $i = 1, 2, \dots, K - 1$ , it can remain at the same state  $i$ , transfer to the next worse state  $i + 1$  during the next cycle. The probabilities can be found from the state transition probabilities obtained in the HMM, from matrix  $\mathbf{A}$  as introduced in Section 2.1.

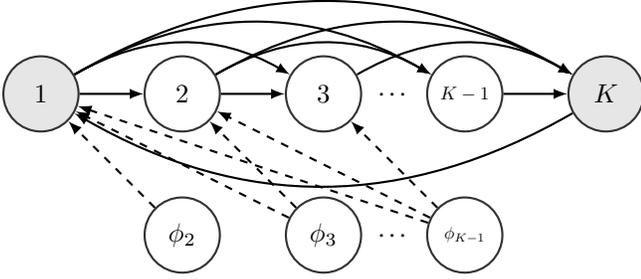


Figure 3. Engine state transition diagram with PM actions

- When the engine fails from state  $K$ , the failure is considered critical and full maintenance (i.e., repair) should be performed, the time of which follows a geometric distribution with mean  $\mu^{-1}$ . At this time, the machine will be fully repaired and restored to the best operating state (state 1, or the “brand new” state). We further expand the matrix  $\mathbf{A}$  to  $\mathbf{A}^*$  by adding the repairing process into the matrix.

$$A^*(i, j) = \begin{cases} A(i, j) & i, j \in \{1, \dots, K - 1\}, \\ 1 - \mu & i = j = K, \\ \mu & i = K, j = 1, \\ 0, & \text{Otherwise.} \end{cases} \quad (11)$$

- When the engine is in any imperfect operating state ( $2, \dots, K$ ) at the beginning of a time slot, the decision maker can choose whether to perform PM or not (as the dashed lines suggest in Figure 3). We define  $\phi_i$  to represent the PM states that the engine enters from imperfect operating state  $i$ ,  $i \in \{2, \dots, K\}$ .
- If PM is performed, then imperfect PM is allowed to recover the machine state to any better operating states. The transition probabilities for PM are state-dependent, and we further define the PM recovery matrix  $\Psi = \{\Psi(i, j)\}$  to represent the probabilities that the engine is recovered to state  $j$  when it is under PM state at state  $\phi_i$ .

A discrete-time Markov model can be built based on the problem descriptions. The state space of the system contains the operating states, the failure state, and the PM states, and therefore can be expressed as  $\mathcal{S} = \{1, \dots, K - 1, K, \phi_2, \dots, \phi_{K-1}\}$ .

The action space  $\pi$  is a collection of all possible PM actions given the engine state  $S \in \mathcal{S}$ . Let  $a_{ij}$  denote that PM action is taken to recover the engine state from state  $i$  to  $j$ ,  $i \in \{2, \dots, K - 1\}, j \in \{1, \dots, i - 1\}$ . Then the set can be expressed as  $\pi = \{a_{ij}\}$  with finite elements to represent the collections of all possible maintenance actions.

To determine the transition of the system state, the transition dynamics for both the robotic arm states and the product quantities should be considered. Denote  $S_t = i$  and  $S_{t+1} = j \in \mathcal{S}$ , given  $a_{u,v} \in \pi$  which represent the state of the system at time  $t$  and  $t + 1$ , and the control policy respectively. Let  $\mathbf{X} = \{X(S_t, a_t, S_{t+1})\}$  represent the state transition probability matrix for the system, then the values for the matrix can be obtained, using the state transition matrix  $\mathbf{A}^*$  and the PM recovery matrix  $\Psi$ , with the equations shown as follows:

$$X(i, a_{u,v}, j) = \begin{cases} A^*(i, j) & i \neq u \text{ and } i, j \in \{1, \dots, K\}, \\ 1, & i = u \text{ and } j = \phi_i, \\ \Psi(u, v) & i = \phi_u, \text{ and } j = v, \\ 1 - \Psi(u, v) & i = \phi_u, \text{ and } j = \phi_u, \\ 0, & \text{Otherwise.} \end{cases} \quad (12)$$

The reward of the system is counted towards the total number of cycle time under normal operations. Let  $r(S, a)$  denote the reward of taking action  $a$  when the system is in state  $S$ , then the system reward can be expressed using the equation as follows:

$$r(S, a) = \begin{cases} 1, & \text{if } S \in \{1, \dots, K\}, \\ 0, & \text{otherwise,} \end{cases} \quad (13)$$

Therefore, when the engine is in an operating state, it accumulates one cycle time. On the other hand, if the engine is down or under maintenance, then no reward will be received.

Since all the latent states are non-observable, we use a belief distribution over states, or the a belief state denoted as  $b = b(S) \in \mathcal{B}$ , to represent the likelihood that the system is in latent state  $S$  under a set of belief  $\mathcal{B}$ . First, define

$$Z(S, a, O) = Pr(O_{t+1} = O | S_{t+1} = S', a) \quad (14)$$

as the distribution that an observation will be captured when the engine is in state  $S'$  and action  $a_t$  was conducted. Notice that this is the decoding problem (as discussed in Section 2.1), which can be solved by the HMM using the historical observations. Then the transformation of the belief state to  $S'$  given action  $a$  and observation  $O$  can be further expressed

in the transition expression as follows:

$$b_o^a(S') = \frac{Z(S', a, O) \sum_{S \in \mathcal{S}} X(S, a, S') b(S)}{\sum_{S' \in \mathcal{S}} \left\{ Z(S', a, O) \sum_{S \in \mathcal{S}} X(S, a, S') b(S) \right\}} \quad (15)$$

Let  $J_\pi(b_0)$  denote the expected total discounted cost given  $b_0$  under policy  $\pi \in \Pi$ ; that is

$$\text{Minimize}_{\pi} J_\pi(b) = \mathbf{E} \left[ \sum_{t=1}^{\infty} \eta^t R(b, \pi) | b_0 \right], \quad (16)$$

where

$$R(b, a) = \sum_{S \in \mathcal{S}} b(S) r(S, a),$$

$0 < \eta < 1$  is a discount factor and  $\pi$  includes all the decisions to determine the PM policies.

Let  $V_t(b)$  denote the expected total achievable reward for the period  $t$  to the infinite horizon given our belief at time  $t$ , with  $V_0(\cdot) = 0$ , then

$$V_t(b) = \min_{a \in \Pi} \left\{ R(b, a) + \eta \sum_{b'} Pr(O|a, b) V_{t+1}(b_o^a) \right\}, \quad (17)$$

in which

$$Pr(O|a, b) = \sum_{S' \in \mathcal{S}} \left\{ Z(S', a, O) \sum_{S \in \mathcal{S}} X(S, a, S') b(S) \right\}.$$

On the right-hand side of Equation (17), it explicitly describes the two components of the reward: the reward for the current state at time  $t$ , and the discounted reward of reaching all the possible system states starting from the next time period. Then the optimal policies starting from time zero can be calculated by using the value iteration approach (Puterman, 2005).

## 4. CASE STUDY

### 4.1. Introduction to NASA Engine dataset

In this case study, we use the degradation dataset the turbofan engine provided in (Saxena & Goebel, 2008) to implement and evaluate our proposed HMM approach. The dataset is generated from Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) and has been widely applied to simulate the machine degradation process due to the reasons such as engine wear and tear (Y. Liu, Frederick, DeCastro, Litt, & Chan, 2012). Furthermore, the dataset considers the usage pattern under various operational conditions obtained from multiple sensors. The schematic diagram of the commercial aircraft gas turbine engine is shown in Fig.4.

In the dataset, there are in total  $n = 100$  units of run-to-failure experimental data, with 20631 records of observations (i.e.,

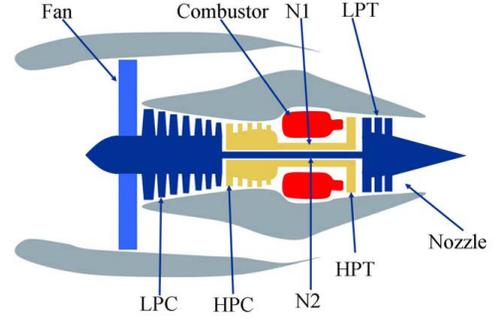


Figure 4. Illustrative picture for the engine systems

$\sum_{i=1}^n T_i = 20631$ ). For each record, there are in total 21 features/ measurements collected by the sensors, which monitor the operational conditions at different locations of the engine. The feature information of all the sensors are summarized in Table 1 and addition information can be found in research by (Saxena, Goebel, Simon, & Eklund, 2008). Fig.5 represents one instance of the run-to-failure sensor data where x-axis shows the time and y-axis represents the magnitude of the sensor signal. And the illustration of each sensor can be found in Table 1

Table 1. Feature descriptions of the dataset

Symbol	Description	Units
T2	Total temperature at fan inlet	°R
T24	Total temperature at LPC outlet	°R
T30	Total temperature at HPC outlet	°R
T50	Total temperature at LPT outlet	°R
P2	Pressure at fan inlet	psia
P15	Total pressure in bypass-duct	psia
P30	Total pressure at HPC outlet	psia
Nf	Physical fan speed	r/min
Nc	Physical core speed	r/min
epr	Engine pressure ratio (P50/P2)	—
Ps30	Static pressure at HPC outlet	psia
phi	Ratio of fuel flow to Ps30	pps/psi
NRf	Corrected fan speed	r/min
NRc	Corrected core speed	r/min
BPR	Bypass Ratio	—
farB	Burner fuel-air ratio	—
htBleed	Bleed Enthalpy	—
Nf_dmd	Demanded fan speed	r/min
PCNfR_dmd	Demanded corrected fan speed	r/min
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s

To conduct the analysis, we divide the dataset, with 80% of the units (i.e.,  $100 \times 80\% = 80$ ) to be the training units and the remaining 20% of the units ( $100 \times 20\% = 20$ ) to be the testing units. For the training units, the degrading conditions of an engine are assumed to be available. All the observations will be used to train the data fusion model using the proposed method. For the testing units, the signals are assumed to be non-observable until they reach some levels before the failure of the engine is detected. Besides, since for all the testing

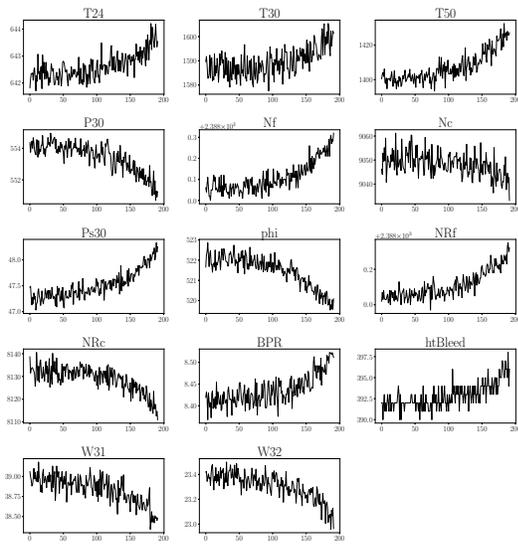


Figure 5. Illustration of the run-to-failure sensor signal

units, the mechanism and model formulations of the simulation model are not provided, it is not applicable to infer the health conditions of the engine using analytical approach (i.e., through parameters such as initial wear of units, parameters of the degradation model, and failure threshold). Therefore, the study is conducted with the underlying premise that only the available dataset provided as above is granted to the practitioners for the health status inference and the remaining lifetime predictions for each testing unit.

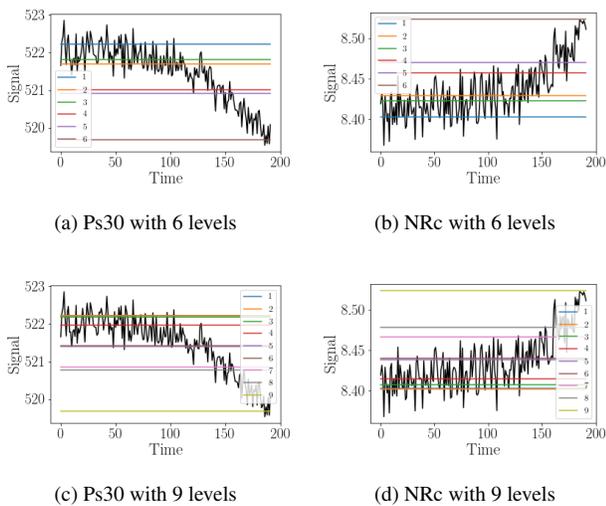
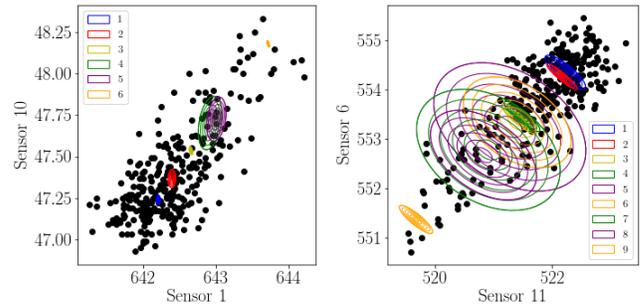


Figure 6. Sensor Levels with 6 and 9 system states

## 4.2. Testing Results

### 4.2.1. System State Estimation and Decoding

We first show the estimated system state for different sensors. Figure 6 shows the identified states for the selected sensors. The X-axis is time and the Y-axis is the magnitude of the signal. We present the results of 6 and 9 system states and the horizontal line shows the mean value of each state. With left-to-right constraint on the HMM, we force the state can only move along one direction. Furthermore, the first and last states will represent the initial condition and failure condition of the system. And the estimated states within those two will represent the health condition of the system. From Figure 6(a) we can see that, as signal decreasing over time, the mean values of our states also decrease. Figure 6(d) shows that the mean values of states increase as signal increase. This is due to the degradation of the system states may demonstrate different trends for different sensors. We can conclude that the proposed method is able to identify correctly the progression of the system states and how it would affect the system dynamics. Furthermore, with more levels, the proposed method is able to better discretize the levels with further details.



(a) State identification for sensor 11 (b) State identification for sensor 10 and sensor 6

Figure 7. States visualization

Furthermore, we would like to show the posterior  $P(S_t = k | \{O_t\})$ . Ideally, we should be able to see that the dynamics of how the system is changing the state gradually.

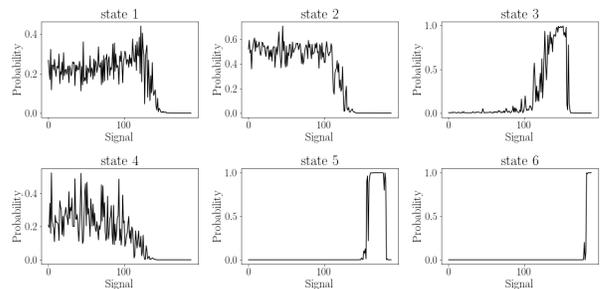


Figure 8. Illustrative picture for the engine systems

From Figure 8, we can conclude that clearly the system is changing states gradually. We can also see that with only 6 states, the states 1-5 are pretty distinct. However, state 0 and state 1 are pretty close to each other, which can also be seen in Figure 6 (a) and (b). With 9 system states, the posterior distribution of which the system is currently at is much noisy. The reason is that the states become more and more similar to each other. It is typically very hard to distinguish these states. However, this would not affect the RUL estimation accuracy since the similar states also share similar RUL, which can be observed in the next subsection.

### 4.3. RUL Prediction

Figure 9 shows the actual residual life with the red line and predicted residual life with the black line. From the result, we can observe that, the prediction error is large at the beginning. But when the system is close to the failure state, the error becomes smaller. Furthermore, with more system states in the system, the RUL follows the true RUL better.

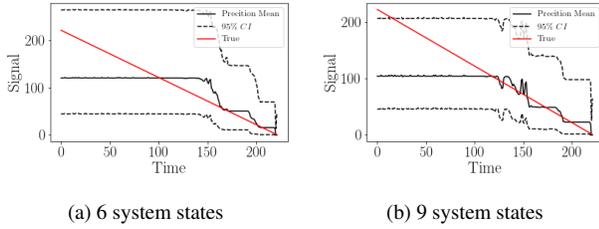


Figure 9. RUL predictions with 6 and 9 system states

#### 4.3.1. Prediction Accuracy

For each validation unit, we calculate the relative percentage error in predicting the actual failure time of the engine, with the performance measurement expressed as follows:

$$\epsilon = \frac{(t_{ob} + t_{pred}) - t_{true}}{t_{true}} \times 100\%, \quad (18)$$

which  $t_{ob}$  is the observed operating time,  $t_{pred}$  is the predicted remaining life time using our model, and  $t_{true}$  is the total operating time of the engine. Note that each life percentile is equal to the current number of cycles observed up to that life percentile plus the predicted number of cycles until failure, which is estimated using our proposed model.

From Figure 10, we can conclude that when observing more data, the prediction accuracy of both systems with 6 and 9 states would drop accordingly. Finally, with only 60% measurement data, the accuracy of RUL prediction is dropped below 0.1, which shows a great prediction power. Finally, the results with 9 states perform a bit better than 6 states due to better state identification. Furthermore, even with 9 states, the proposed algorithm is able to learn each state accurately,

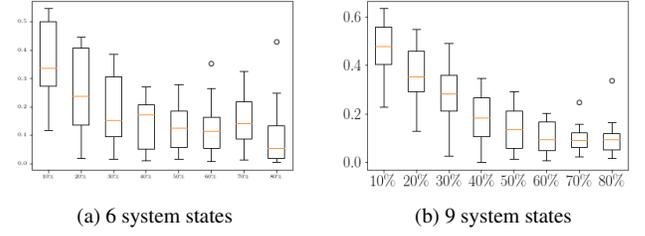


Figure 10. Boxplot of prediction accuracy with 6 and 9 system states when observing 10%-80% of data

which shows how the added constraint in the HMM help to learn a better and more stable state representation.

### 4.4. Preventative Maintenance Policy

Based on the results of the HMM with six latent states, we can generate the PM policies for the engine. The state transition matrix can be expressed as follows:

$$\mathbf{A}^* = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{bmatrix} 0.094 & 0.838 & 0.048 & 0.018 & 0.004 & 0 \\ 0 & 0.957 & 0.040 & 0.003 & 0 & 0 \\ 0 & 0 & 0.980 & 0.020 & 0 & 0 \\ 0 & 0 & 0 & 0.974 & 0.012 & 0.014 \\ 0 & 0 & 0 & 0 & 0.958 & 0.042 \\ 0.025 & 0 & 0 & 0 & 0 & 0.975 \end{bmatrix} \end{matrix}$$

For the transition from state 6 to state 1, we assume that the time for the repairing process follows a geometric distribution with parameter 0.025. The failure of the engine is treated as a severe event, which usually takes long time to repair.

Furthermore, we assume that the PM recovery matrix has the following format:

$$\Psi = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.112 & 0 & 0 & 0 & 0 \\ 0.084 & 0.126 & 0 & 0 & 0 \\ 0.070 & 0.126 & 0.136 & 0 & 0 \\ 0.030 & 0.034 & 0.040 & 0.054 & 0 \end{bmatrix} \end{matrix}$$

Note that the time and the conditions of the engines after maintenance, in practice, is determined by the hours of labors and the level of tooling under the specific applications (Ja, Kulkarni, Mitra, & Patankar, 2001). For example, for the vibration severity of the engines, the ISO IS2372 standard classifies different levels with different criteria and practice guidelines (Kiangala & Wang, 2018).

Then we implement the proposed POMDP approach, with  $\eta = 0.99$ , to derive the optimal control policy for the production performance. The optimal policy indicates that when the engine is in state 5 or 4, PM actions should be performed, which will recover the engine state to state 3. For the other

operating states, no PM will be scheduled. By conducting such PM performance, simulation experiments can show that the long-term engine efficiency, which has the format of:

$$\text{Efficiency} = \frac{\text{Total operating time}}{\text{Total time including maintenance}}, \quad (19)$$

will raise from 0.746 to 0.874, which is an over 15% increase on long term engine performance.

In practice, to apply the PM polices onto the engine, alarms can be set up on the threshold values of different features under the latent states in which PM are needed. For example, in Figure 11, when the signal level of sensor 10 reaches above 47.75, or the value for sensor 1 is above 643, an alarm will pop-up to schedule the PM. Then through continuous repairing and testing, the signal level will be recovered to lower levels, i.e., the value for sensor 10 will drop to 47.50 after PM.

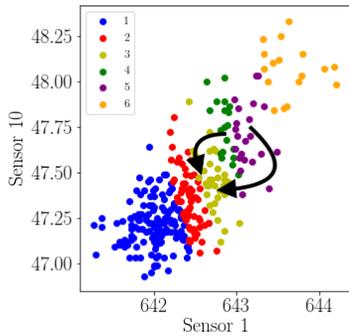


Figure 11. Maintenance strategy illustration using sensor 1 and 10

## 5. CONCLUSION

In this paper, we develop a semi-supervised left-to-right constrained HMM model to estimate the RUL of degrading unit using multiple sensors. The information from multiple sensors is utilized to identify the system state. Then the RUL is estimated as the earliest time of reaching to the failure state given a new observation. Furthermore, we propose a POMDP method based on the developed HMM to determine the optimal preventative maintenance strategy for the system under consideration. In the case study, We utilize the NASA Engine data set to demonstrate the effectiveness of the proposed method. The results show that the proposed method is effective in identifying the progression of the system states and capture the system dynamics with details. The predicted RUL is close to the ground truth values in the testing dataset. In addition, the generated preventative maintenance policy is effective in optimizing engine performance in the long term.

In the future, this work can be extended to more complicated systems with multiple degrading units, the behaviors of which

are tightly coupled. One typical example is a manufacturing system with multiple machines where each machine exhibits similar degrading dynamics. Analyzing such complicated systems will expose great challenges on system modeling and computational efficiency. Second, the structure of the models can be further extended, i.e., the HMM model can be extended to Hidden Semi-Markov Model (HSMM) with multiple layers. Such an extension enables better estimation on different time and observation frequency of the degradation process. Besides, new system improvement and control methods will be needed to incorporate the additional complexity these modeling extensions will bring.

## REFERENCES

- Alaswad, S., & Xiang, Y. (2017). A review on condition-based maintenance optimization models for stochastically deteriorating system. *Reliability Engineering & System Safety*, 157, 54–63.
- Athanasopoulou, E., Li, L., & Hadjicostis, C. N. (2010). Maximum likelihood failure diagnosis in finite state machines under unreliable observations. *IEEE Transactions on Automatic Control*, 55(3), 579–593.
- Barlow, R., & Hunter, L. (1960). Optimum preventive maintenance policies. *Operations research*, 8(1), 90–100.
- Boutros, T., & Liang, M. (2011). Detection and diagnosis of bearing and cutting tool faults using hidden markov models. *Mechanical Systems and Signal Processing*, 25(6), 2102–2124.
- Caesarendra, W., Widodo, A., Thom, P. H., Yang, B.-S., & Setiawan, J. D. (2011). Combined probability approach and indirect data-driven method for bearing degradation prognostics. *IEEE Transactions on Reliability*, 60(1), 14–20.
- 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.
- de Jonge, B., Teunter, R., & Tinga, T. (2017). The influence of practical factors on the benefits of condition-based maintenance over time-based maintenance. *Reliability engineering & system safety*, 158, 21–30.
- Douer, N., & Yechiali, U. (1994). Optimal repair and replacement in markovian systems. *Stochastic Models*, 10(1), 253–270.
- Geramifard, O., Xu, J.-X., Zhou, J.-H., & Li, X. (2014). Multimodal hidden markov model-based approach for tool wear monitoring. *IEEE Transactions on Industrial Electronics*, 61(6), 2900–2911.
- Ja, S.-S., Kulkarni, V. G., Mitra, A., & Patankar, J. G. (2001). A nonrenewable minimal-repair warranty policy with time-dependent costs. *IEEE Transactions on Reliability*, 50(4), 346–352.
- Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on

- machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, 20(7), 1483–1510.
- Kang, Y., & Ju, F. (2019). Flexible preventative maintenance for serial production lines with multi-stage degrading machines and finite buffers. *IISE Transactions*(just-accepted), 1–28.
- Keizer, M. C. O., Flapper, S. D. P., & Teunter, R. H. (2017). Condition-based maintenance policies for systems with multiple dependent components: A review. *European Journal of Operational Research*.
- Kiangala, K. S., & Wang, Z. (2018). Initiating predictive maintenance for a conveyor motor in a bottling plant using industry 4.0 concepts. *The International Journal of Advanced Manufacturing Technology*, 97(9-12), 3251–3271.
- Klutke, G.-A., & Yang, Y. (2002). The availability of inspected systems subject to shocks and graceful degradation. *IEEE Transactions on Reliability*, 51(3), 371–374.
- Le, T. T., Chatelain, F., & Bérenguer, C. (2016). Competing deterioration processes modeling and rul estimation based on factorial hidden markov models. In *9th ima international conference on modelling in industrial maintenance and reliability (mimar)* (pp. 108–113).
- Liao, H., Elsayed, E. A., & Chan, L.-Y. (2006). Maintenance of continuously monitored degrading systems. *European Journal of Operational Research*, 175(2), 821–835.
- 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., Chen, J., & Dong, G. (2014). Zero crossing and coupled hidden markov model for a rolling bearing performance degradation assessment. *Journal of Vibration and Control*, 20(16), 2487–2500.
- Liu, Y., Frederick, D. K., DeCastro, J. A., Litt, J. S., & Chan, W. W. (2012). User's guide for the commercial modular aero-propulsion system simulation (c-mapss): Version 2.
- Liyanage, J. P., & Kumar, U. (2003). Towards a value-based view on operations and maintenance performance management. *Journal of Quality in Maintenance Engineering*, 9(4), 333–350.
- Lu, C. J., & Meeker, W. O. (1993). Using degradation measures to estimate a time-to-failure distribution. *Technometrics*, 35(2), 161–174.
- Luo, M., Yan, H.-C., Hu, B., Zhou, J.-H., & Pang, C. K. (2015). A data-driven two-stage maintenance framework for degradation prediction in semiconductor manufacturing industries. *Computers & Industrial Engineering*, 85, 414–422.
- Mobley, R. K. (2002). *An introduction to predictive maintenance*. Elsevier.
- Nelson, W. B. (2009). Accelerated testing: statistical models, test plans, and data analysis.
- Puterman, M. L. (2005). Markov decision processes: Discrete stochastic dynamic programming (wiley series in probability and statistics).
- Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257–286.
- Saxena, A., & Goebel, K. (2008). *Turbofan engine degradation simulation data set*. [NASA Ames Prognostics Data Repository]. Retrieved 2019-05-30, from <http://ti.arc.nasa.gov/project/prognostic-data-repository>.
- 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).
- Soualhi, A., Clerc, G., Razik, H., Guillet, F., et al. (2016). Hidden markov models for the prediction of impending faults. *IEEE Transactions on Industrial Electronics*, 63(5), 3271–3281.
- Sun, J., Zuo, H., Wang, W., & Pecht, M. G. (2014). Prognostics uncertainty reduction by fusing on-line monitoring data based on a state-space-based degradation model. *Mechanical Systems and Signal Processing*, 45(2), 396–407.
- 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, F., Lu, Y., & Ju, F. (2018). Condition-based real-time production control for smart manufacturing systems. In *Ieee 14th international conference on automation science and engineering* (pp. 1052–1057).
- Wang, H. (2002). A survey of maintenance policies of deteriorating systems. *European journal of operational research*, 139(3), 469–489.
- Wang, W. (2012). An overview of the recent advances in delay-time-based maintenance modelling. *Reliability Engineering & System Safety*, 106, 165–178.
- Yan, H., Liu, K., Zhang, X., & Shi, J. (2016). Multiple sensor data fusion for degradation modeling and prognostics under multiple operational conditions. *IEEE Transactions on Reliability*, 65(3), 1416–1426.

## BIOGRAPHIES

**Xinyu Zhao** received the B.S. degree from the Department of Mechanical Engineering, the Jilin University, Jilin, China, and the M.S. degree from the Department of Mechanical Engineering, University of Washington, Seattle, WA, USA, in 2016 and 2018 respectively. He is now working towards the Ph.D. degree at the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University,

Tempe, AZ, USA. His research interests are in developing scalable statistical learning algorithms for large-scale high-dimensional data.

**Yunyi Kang** received the B.S. degree from the Department of Industrial Systems and Engineering, the Hong Kong Polytechnic University, Kowloon, Hong Kong, China, and the M.S. degree from the Department of Industrial Systems and Engineering, Rutgers University, New Brunswick, NJ, USA, in 2013 and 2015 respectively. He is now working towards the Ph.D. degree at the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA. He was a recipient of the best student paper finalist in IEEE CASE. His research interests are in modeling, analysis and real-time control of production systems.

**Hao Yan** received the B.S. degree in Physics from the Peking University, Beijing, China, in 2011. He also received the M.S. degree in Statistics, the M.S. degree in Computational Science and Engineering, and the Ph.D. degree in Industrial Engineering from Georgia Institute of Technology, Atlanta, in 2015, 2016, 2017, respectively. Currently, he is an assistant professor in the School of Computing, Informatics, &

Decision Systems Engineering at Arizona State University. His research interests focus on developing scalable statistical learning algorithms for large-scale high-dimensional data with complex heterogeneous structures to extract useful information for the purpose of system performance assessment, anomaly detection, intelligent sampling and decision making. He is a member of INFORMS and IISE.

**Feng Ju** received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2010, and the M.S. degree in electrical and computer engineering and Ph.D. degree in industrial and systems engineering from the University of Wisconsin, Madison, WI, USA, in 2011 and 2015, respectively. He is an Assistant Professor with the School of Computing, Informatics & Decision Systems Engineering, Arizona State University, Tempe, AZ, USA. His current research interests include modeling, analysis, continuous improvement, and optimization of manufacturing systems. He is a member of IEEE, INFORMS, and IISE. He has received multiple paper awards, including the best paper award in IFAC MIM and best student paper finalist in IEEE CASE and IFAC INCOM.