Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Bootstrap Confidence Interval on IOHMM Parameters for System Health Diagnostic Under Multiple Operating Conditions

Kamrul Islam Shahin*, Christophe Simon, Philippe Weber*****

kamrul-islam.shahin@univ-lorraine.fr, christophe.simon@univ-lorraine.fr, philippe.weber@univ-lorraine.fr

Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France

Abstract: The operating conditions have an important impact on system degradation. This paper uses the Input-Output Hidden Markov Model to represent the system degradation having multiple operating conditions. In this paper the bootstrap method is applied to estimate the model parameters and to diagnostic the system health. Parameters of the model are computed with 95% confidence intervals. The uncertainty about multiple data sequences and degradation speed is handled according to the operating conditions. A numerical application is given to explain the methodologies used to estimate the model parameters and diagnostic the system health.

Keywords: System health, PHM, Input Output Hidden Markov Model, Condition based diagnostic, Degradation design, Condition monitoring

1. INTRODUCTION

Diagnostics play a very important role in prognostic and health managing (PHM) of systems. Health diagnosis ensures the level of system degradation. Based on the estimated diagnosis, remaining useful life (RUL) of the system is evaluated which is an essential part of the next generation of maintenance. Maintenance is very crucial for trouble-free production, as the system must not be shut down during the operation. It is necessary to predict the health states of the system during the operation leading to the system for repairing or replacing before the unplanned shutdown. In this case, it is very important that the system is correctly diagnosed, otherwise, effective RUL prediction is not possible. Therefore, this will be worth the wrong estimate, and the system will be failed before maintenance. Thus, the production speed will decrease, and the production cost will increase. To solve this problem, many scientific works are proposed on diagnostic applications based on degradation identification. Degradation is not easy to assess because it is a hidden and unknown process. It is impossible to know the evolution of degradation until the system is stopped and observed directly inside, which is completely contrary to our goal. However, as the system ages, it gets damaged. If the system operates at higher pressures, it decomposes quickly, otherwise, the process is slow for the lower pressures. This means that the lifetime of the system very much depends on the operating conditions of the system. This document takes into account the impact of operating conditions on the degradation and offers a model for identifying the system's degradation under several operating conditions and assessing the state of the system (diagnostics). There are many models available in PHM applications that can be classified into three approaches: model-based, data-based, and hybrid. The purpose of this paper requires a parametric and non-supervised model, that is why the data-driven model is ideal for the solution. The aim is to have a stochastic system in which states are unknown and their evolution is random. In this case, the Hidden Markov Model (HMM) can be an excellent choice. HMM was first introduced by Baum in the early 1970s [Baum 1966], and [Rabiner 1989] used it for the first time in an application for recognition of speech. It was later used in PHM challenges. For example [Kumar 2019, Tobon 2011, Baruah 2005] proposed HMM-based applications for diagnostics, but without regard to operating conditions. Some other researchers have used the "Hidden Semi-Markov models" [Dong 2007], "Mixture of Gaussian Hidden Markov Models" [Tobon 2012], "Hidden Markovian Hierarchical Models" [Camci 2006] in various diagnostic applications. They have tried different versions of HMMs to produce better results, but no one integrates the operating conditions as inputs. These models cannot be used to consider the operating condition because HMMs mentioned above do not allow any input. However, the Input-Output Hidden Markov Model (IOHMM) which is another advanced version of HMM, an interesting model in which operating conditions can be considered as input conditions. Similar work is being done in [Le 2015], where a Multi-Branch HSMM (MBHSMM) is proposed. This is a motivating work, but there are also some limitations. The author divided the dataset into different parts for different models that correspond to the operating conditions. They fixed one operating condition while applying the matching model. In reality, it is not the same because operating conditions may change at any time during the operation. On the other hand, IOHMM trains several models simultaneously without separating the data sequence and allows us to switch the models at given input sequences. Thus, IOHMM assesses a more realistic degradation of the system with the effect of operating conditions.

IOHMM was introduced in [Bengio 1995] for the first time. Then, it is used in diagnostic and prognostic applications [Shahin 2019a, Shahin 2019b]. First, a training approach to the model parameters has been proposed. Estimating model parameters from data through the algorithm proposed in [Shahin 2019b] does not give the confidence intervals in results. To solve this problem the bootstrap-IOHMM method is applied to the model and estimated the model parameters with a 95% confidence interval. A statistical exposition of the bootstrap is given by [Efron and Gong 1983]. The bootstrap Hidden Markov Model (HMM) is used in several applications, such as clustering time series [Oates 2000] and quality of synthetic speech [Kim 2002], etc. Inspired by this work we designed the bootstrap-IOHMM method to achieve the goal of this paper. The basic idea of the bootstrap involves the variability in an unknown distribution from which the data are drawn by resampling with replacement from the dataset [Felsenstein 1985]. It is most useful when a set of estimates has a distribution that approximates the distribution of the actual estimate. The bootstrap method is used to estimate model parameters with the confidence interval, average sample, standard error, lower limit, and upper limit. IOHMM uses three adapted algorithms: the Baum Welch algorithm which is a class of EM (Expectation-Maximization) algorithm based on the forward-backward algorithm in the training and adapted Viterbi algorithm in the diagnostic application. The adaptation of the algorithms for IOHMM is described in [Shahin 2019b].

This paper is organized as follows: Section 2 defines the basis of model structure, training, and diagnostic methodologies. Then, a brief discussion of bootstrap and confidence intervals are given in section 3. After that, section 4 shows a numerical application and corresponding results. Finally, a conclusion and perspectives are developed in section 5.

2. PROPOSED MODEL

IOHMM is a class of HMM. It is a stochastic model where the state probability depends on its previous state, observation, and the input condition. This model can be used to represent systems with multiple operating conditions and multiple outputs. An IOHMM is presented (Fig1) as in Dynamic Bayesian Network (DBN) [Salem 2007].

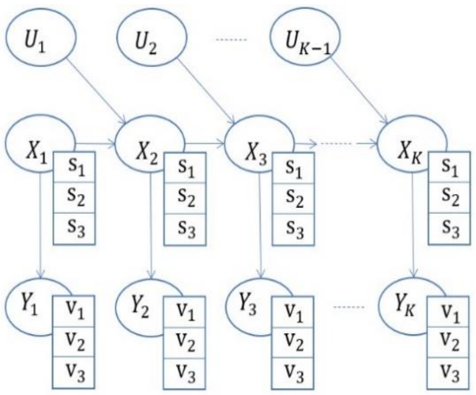

Fig. 1. DBN representation of 3-States IOHMM.

2.1 *Model Structure*

2.1.1 Transition Probability

IOHMM evolves in a sequence of the variable X which holds one of the hidden states $\{s_1, s_2, s_3, \cdots, s_N\}$ at each time instant k ($k \in \mathbb{N}$ is discrete time). $A = (a_{ij})$ represents the transition matrix where $a_{ij} = P(X_k = s_j | X_{k-1} = s_i)$ is a transition from state $X_{k-1} = s_i$ to state $X_k = s_j$ ($1 \le i, j \le N$). The summation of a_{ij} for each state j is 1. Multiple transition matrices are represented as $A^p = (a^p{}_{ij})$, where p is the operating conditions number.

2.1.2 Emission probability

The set of possible emitted symbols of the output variable Y_k^q are assumed as $V = \{v_1, v_2, v_3, \cdots, v_M\}$. $B = (b_{jl})$ denotes the state emission probability matrix, where $b_{jl} = P(Y_k = v_m | X_k = s_j)$ is the emission probabilities of state $x_k = s_j$. The summation of b_{jl} for each state j is 1. Multiple output-observations are represented as $B^q = (b^q{}_{jl})$, where q is the number of outputs of the system.

2.1.3 Input Condition

In this model, the input sequence is presented as $U_{1:k}$, and the transition probabilities are computed as follows $P(X_k | X_{k-1}, U_{k-1})$. All the sequences are not length sensitive and sequences could come with any length.
If the initial state probability distribution is $\pi = P(X_1 = s_i)$ and the model denoted by Λ, then the triplet $\Lambda = (A^p, B^q, \pi)$ completely defines the model structure.

2.2 Method: Model Training

The Baum Welch and the Forward-Backward algorithms are commonly used in HMM which does not allow us to consider the input condition nor the multiple output case. That is why the adapted algorithms are used which able to consider multiple inputs and outputs during the training session. The detail adaptation is discussed in [Shahin 2019b]. The forward-backward algorithm is given in (1) and (2).

- Forward algorithm:

Basis: $\alpha(X_1) = P(Y_1 | X_1) P(X_1)$
Recursion:

$$\alpha(X_k) = \sum_{X_{k-1}=s_1}^{s_N} \alpha(X_{k-1}) P(X_k | X_{k-1}, U_{k-1}) P(Y^q{}_k | X_k) \quad (1)$$

here $\alpha(X_k) = P(X_k, Y_{1:k})$, and Y^q presents multiple sequences where q is the number of outputs of the system.

- Backward algorithm:

Basis: $\beta(X_K = s_i) = \begin{pmatrix} 1 \\ 1 \\ .. \\ 1 \end{pmatrix}$

K is the length of the sequence.

Recursion:

$$\beta(X_k) = \sum_{X_{k+1}=s_1}^{s_N} \beta(X_{k+1}) P(X_{k+1}|X_k, U_k) P(Y^q{}_{k+1}|X_{k+1}) \quad (2)$$

here $\beta(X_k) = P(Y_{k+1:K}|X_k)$.

The Baum-Welch algorithm takes $\alpha(X_k)$ and $\beta(X_k)$, applies the Baum Welch algorithm to update the parameter in an iterative process.

- The Baum Welch algorithm:

The probability of being in state j at time k given the observed sequences $(Y^q_{1:k})$and the parameters of Λ is given below by (3):

$$\omega_k(j) = \frac{\alpha_i(X_k)\beta_j(X_k)}{P(Y^q_{1:k}|\Lambda)} \quad (3)$$

The probability of being in state i and j at time k and $k+1$ given the observed sequences of $(Y^q_{1:k})$ the input operating conditions U and the parameters of Λ is given by (4):

$$\varepsilon_k(i,j) = \frac{\alpha_i(X_k).a^p(U_k)_{ij}.b^q{}_{jk}.\beta_j(X_{k+1})}{P(Y^q_{1:k}|\Lambda)} \quad (4)$$

Parameters update:

- Initial state probability:

  $\pi_i = \varepsilon_1(i,j)$, where $1 \le i \le N$, where N is the number of hidden states. (5)

- Transition probabilities:

$$\hat{a}^p{}_{ij} = \frac{\sum_{k=1}^{K-1} \varepsilon_k(i,j).1_{X_k(U_k=p)}}{\sum_{k=1}^{K-1} \omega_k(j).1_{X_k(U_k=p)}} \quad (6)$$

here $1_{X_k(U_k=p)} = \begin{cases} 1 \text{ if } X_k(U_k=p) \\ 0 \text{ others} \end{cases}$

- Emission probabilities:

$$\hat{b}^q{}_{jk} = \frac{\sum_{k=1}^{K} \omega_k(j).1_{Y^q{}_k=v_m}}{\sum_{k=1}^{K} \omega_k(j)} \quad (7)$$

here $1_{Y^q{}_k=v_m} = \begin{cases} 1 \text{ if } Y^q{}_k = v_m \\ 0 \quad \text{otherwise} \end{cases}$

Now, these steps are repeated iteratively until the changes between two consecutive results fall into a given tolerance and fixed the parameters of the model as $\hat{\Lambda} = (A^p, B^q, \pi)$, which is used in diagnostic of the system.

2.3 Method: Diagnostic

The Viterbi algorithm is used to diagnostic the system using the trained model $\Lambda = (A^p, B^q, \pi)$. It is an interesting algorithm that estimates the maximum path using (8). This algorithm is also adapted to IOHMM [Shahin 2019b]. It computes the maximum likelihood state sequence considering operating conditions as input.

The Viterbi algorithm:
Basis: $\gamma(X_1) = P(X_1, Y^q{}_1)$
Recursion:

$$\gamma(X_k) = max_{(X_{k-1})} P(Y^q{}_k|X_k) P(X_k|X_{k-1}U_{k-1}) \gamma(X_{k-1}) \quad (8)$$

here $\gamma(X_k) = max_{(X_{1:k})} P(X_{1:k}, Y_{1:k})$.

Based on this set of algorithms the next section explains the confidence interval and the bootstrap method.

3. BOOTSTRAP-IOHMM

The bootstrap-IOHMM method is a sampling technique used to estimate IOHMM parameters statistic by sampling a dataset with replacement. It can be used to estimate measures of accuracy, such as confidence intervals, the sample mean, standard deviation, variance, etc. Resampling with replacement selects some random dataset from the original sample and put those back into the sample again for another selection. Resampling size should be equal to the sampling size which may have some repeated dataset. This technique maintains data structure but reshuffles values, extrapolating to the data population. This repeated process uses the new sample to generate the sampling distribution of the mean. Some important definitions are needed to understand for understanding the bootstrapping.

Confidence interval (CI): Confidence interval estimated from observed statistical data, which may contain an unknown population parameter. The CI communicates the accuracy of a probabilistic estimate. It expresses a range in which it is fairly certain that the population parameter is present. The range-width depends on the variation within the population of interest and the sample size. [Efron 1986]

Population variation: If all values in a large data population are almost the same, then the sample also has a small variation. It gives a small confidence interval. On the other hand, more varied data will lead to more varied samples, which makes less sure that the sample average is close to the population mean. That means the CI is large in this case. The greater variation of the data leads to a wider CI.

Sample size: The sample size also affects the width of a confidence interval. Small samples differ more from each other and have less information. There is more variation due to a sampling error. The CI may be larger. On the other hand, larger samples will be more similar. The effect of the sampling error is less, and the information is more. The confidence interval may be smaller in this case. [Efron 1986]

Calculating confidence intervals: The confidence interval calculation for a mean uses formula (9).

$$CI = \bar{X} \pm t\frac{s}{\sqrt{n}} \quad (9)$$

where $\bar{X}$ is the sample mean, t is the t-distribution which depends on the sample size and the chosen level of confidence, s is the sample standard deviation and n is the sample size.

Sample: A sample is a selection of observations from the population of interest. Different selection criterion is simple, random, convenient, systematic, clustered, layered, etc.

Sampling error: A sample is only a selection of objects from the population. It will never be a perfect representation of the population. Different samples of the same population will yield different results. This is called sampling error or sampling variation. There will always be a sampling error. [Efron 1986]

The sample means: Defined as the average of observations in the sample of the population. The sample mean is considered as the estimate of the population mean.

Sample standard deviation: It is the average distance of the sample data from the sample mean.

Useful:

- Bootstrapping is useful for modelling non-normal data
- It provides unknown statistic properties (e.g. PCA results)
- It has unknown statistic properties and shows the standard calculation such as a 95% confidence interval.
- Easy to measure result accuracy or compute standard error, sample mean, standard deviation, variance etc.

4. NUMERICAL APPLICATION

A system has represented in this application which has two outputs to be observed applying an operating condition with two different modes. The operating modes provide two different transition matrices. The degradation of the system assumed to have three hidden states (good, moderate, bad) for easy and simple computation. Each of the states emits two outputs with two probabilities which are represented by two emission matrices. There are three discrete symbols considered in the emitted observation sequences.

The proposed model is designed to use the sensor data as a unit. The system temperature and the system vibration are used as two observations, and the speed of the system is considered as an input where the speed has two modes (high and low). The goal is divided into two steps:

1. Synchronize the system with IOHMM and training the model using the available data sets.
    a. Use (1) to (7) for parameter estimation.
    b. Provide a confidence interval for each of the estimate parameters of the model using the bootstrap-IOHMM method (9).
2. Use the trained up IOHMM to estimate the system diagnostic according to a given data set.
    a. Use the Viterbi algorithm to compute the system health condition at any time given a new data set.
    b. Provide a comparison between the model trained with bootstrap method and without bootstrap method.

The goal is to show how the bootstrap method is useful when a small data-amount can be used to estimate parameters of a big data population.

4.1 Data Generating

To validate the procedure, we have generated the data sequences using a given model structure. Later, this (original) model is compared with the estimated model in the result section.

Model architecture:

- Data unit is discrete: continuous signal data could be converted into discrete unit.
- Model type is left-right: a system cannot comeback from the bad health to good health.
- Hidden states are three (could be more).
- Observation symbols are three for each output (could be more).
- Transition matrices are two according to two input modes. The matrices have zero on the lower parameters of the diagonal because of the left-right property. Switching model is illustrated in Fig. 2.

$$A^1 = \begin{pmatrix} 0.9788 & 0.0212 & 0 \\ 0 & 0.9516 & 0.0484 \\ 0 & 0 & 1 \end{pmatrix}$$

$$A^2 = \begin{pmatrix} 0.8443 & 0.1557 & 0 \\ 0 & 0.7899 & 0.2101 \\ 0 & 0 & 1 \end{pmatrix}$$

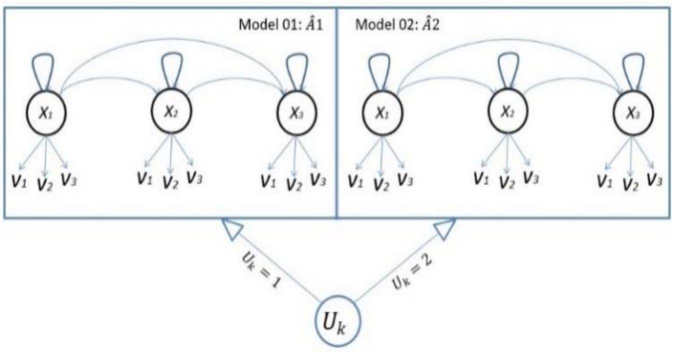

Fig. 2. Model switching by input U_k

- Emission matrices: two for two outputs

$$B^1 = \begin{pmatrix} 0.8980 & 0.0513 & 0.0507 \\ 0.0534 & 0.8980 & 0.0486 \\ 0.0499 & 0.0505 & 0.8996 \end{pmatrix}$$

$$B^2 = \begin{pmatrix} 0.7981 & 0.1520 & 0.0499 \\ 0.2017 & 0.7488 & 0.0495 \\ 0.1007 & 0.0495 & 0.8498 \end{pmatrix}$$

- Initial state distribution: (assumed as in good health)

$$\pi = (1 \quad 0 \quad 0)$$

A set of 1000 data is generated using a simulator following this given model architecture. This data sequences are used in IOHMM training to estimate the parameters with 95% confidence intervals. The bootstrap approach is applied where the 30 data sets are randomly selected with data replacement for multiple times. The iteration was about 1000 times.

4.3 Results

4.3.1 Training results

Fig. 3 shows the confidence interval for a total of 36 parameters of the transition and emission matrices. The X-axis of each rectangle represents the probabilities and the Y-axis represents the boot execution number.

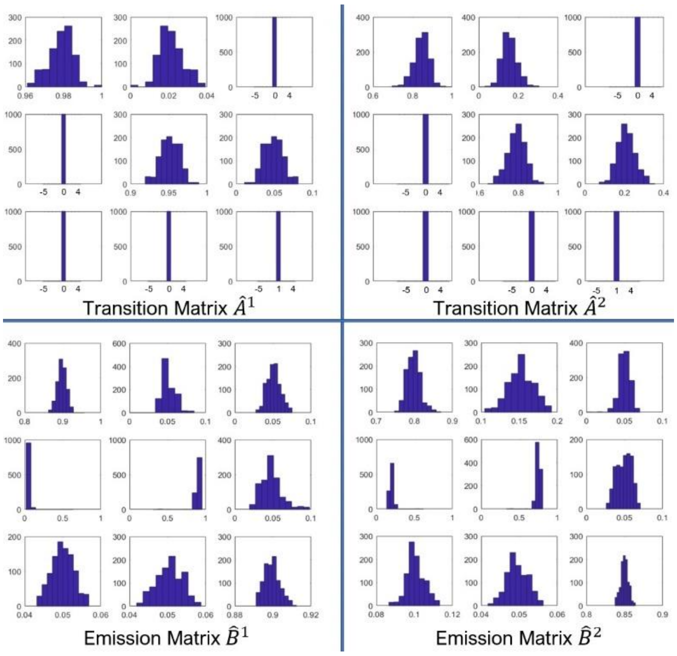

Fig. 3. Distribution of matrices parameters

Both the transition matrices are having some zeros on corresponding parameters. These parameters did not get any transition probability during the training following the nature of the system. The left-right model is used in data simulation as mentioned earlier. This is the reason the transition matrices have zeros on (2,1), (3,1), (3,2) position (highlight in *Fig. 4*). The green circle on the position (3,3) presents the absorbent state with a 100% probability. Besides these four, all the other parameters are estimated with a 95% confidence interval (see Table 1).

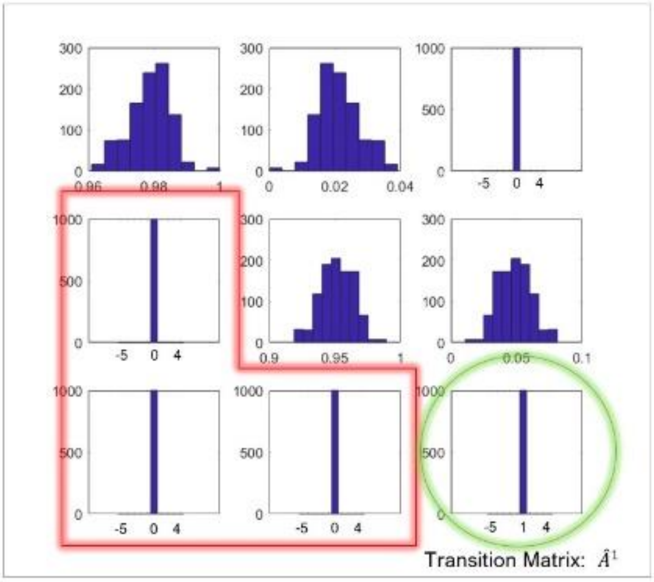

Fig. 4. Parameter distribution for the first transition matrix.

Table 1 shows 31 parameters of transition matrices, emission matrices, and the initial state distributions. Each row represents different information (lower bound, upper bound, mean, standard error) about a parameter. Parameters having zero value are ignored in the table.

Table 1. Bootstrap parameters

Para-meter	Lower bound	Higher bound	Mean value	Standard Error
Transition Matrix $\widehat{A}^1$				
$\hat{A}^1_{11}$	0.9783	0.9791	0.9787	1.96×10^{-4}
$\hat{A}^1_{12}$	0.0209	0.0217	0.0213	1.96×10^{-4}
$\hat{A}^1_{22}$	0.9508	0.9523	0.9515	3.91×10^{-4}
$\hat{A}^1_{23}$	0.0477	0.0492	0.0485	3.91×10^{-4}
$\hat{A}^1_{33}$	1	1	1	0
Transition Matrix $\widehat{A}^2$				
$\hat{A}^2_{11}$	0.8428	0.8477	0.8453	0.0012
$\hat{A}^2_{12}$	0.1523	0.1572	0.1547	0.0012
$\hat{A}^2_{22}$	0.7861	0.7916	0.7889	0.0014
$\hat{A}^2_{23}$	0.2084	0.2139	0.2111	0.0014
$\hat{A}^2_{33}$	1	1	1	0
Emission Matrix $\widehat{B}^1$				
$\hat{B}^1_{11}$	0.8970	0.8984	0.8977	3.70×10^{-4}
$\hat{B}^1_{12}$	0.0506	0.0517	0.0512	2.82×10^{-4}
$\hat{B}^1_{13}$	0.0506	0.0517	0.0511	2.80×10^{-4}
$\hat{B}^1_{21}$	0.0524	0.0591	0.0557	17×10^{-4}
$\hat{B}^1_{22}$	0.8929	0.8999	0.8964	18×10^{-4}
$\hat{B}^1_{23}$	0.0470	0.0487	0.0479	4.32×10^{-4}
$\hat{B}^1_{31}$	0.0498	0.0501	0.0500	0.894×10^{-4}
$\hat{B}^1_{32}$	0.0503	0.0507	0.0505	1.07×10^{-4}
$\hat{B}^1_{33}$	0.8993	0.8998	0.8995	1.29×10^{-4}

Emission Matrix $\hat{B}^2$				
$\hat{B}^2_{11}$	0.7966	0.7988	0.7977	5.57×10^{-4}
$\hat{B}^2_{12}$	0.1513	0.1533	0.1523	5.25×10^{-4}
$\hat{B}^2_{13}$	0.0496	0.0505	0.0500	2.33×10^{-4}
$\hat{B}^2_{21}$	0.2011	0.2065	0.2038	14×10^{-4}
$\hat{B}^2_{22}$	0.7444	0.7497	0.7470	14×10^{-4}
$\hat{B}^2_{23}$	0.0486	0.0498	0.0492	3.05×10^{-4}
$\hat{B}^2_{31}$	0.1003	0.1009	0.1006	1.47×10^{-4}
$\hat{B}^2_{32}$	0.0493	0.0496	0.0494	0.869×10^{-4}
$\hat{B}^2_{33}$	0.8096	0.8502	0.8499	1.60×10^{-4}
Initial state distribution				
π (1)	0.9761	0.9917	0.9839	0.0040
π (2)	0.0083	0.0239	0.0161	0.0040
π (3)	0	0	0	0

Total standard error in matrix $\hat{A}^1$ is 11.74×10^{-4}, matrix $\hat{A}^2$ is 52×10^{-4}, matrix $\hat{B}^2$ is 51.894×10^{-4}, matrix $\hat{B}^2$ is 48.139×10^{-4}, and initial state distribution is 80×10^{-4}. The matrix $\hat{A}^2$ comparably has a larger standard error than the matrix $\hat{A}^1$ because the amount of training data dedicated to each matrix. $\hat{A}^2$ is trained with about 20% data while 80% data are used to train matrix $\hat{A}^1$.

Now, if the matrices organized with the mean value then we find the estimated parameters of IOHMM as following:

- Estimated transition parameters:

$$\hat{A}^1 = \begin{pmatrix} 0.9787 & 0.0213 & 0 \\ 0 & 0.9515 & 0.0485 \\ 0 & 0 & 1 \end{pmatrix}$$

$\hat{A}^1$ is the low stressed (e.g. low speed) model transitions where the mean transition probability from the first state to the last state is $(0.0213 + 0.0485)/2 = 0.0349$.

$$\hat{A}^2 = \begin{pmatrix} 0.8453 & 0.1547 & 0 \\ 0 & 0.7889 & 0.2111 \\ 0 & 0 & 1 \end{pmatrix}$$

$\hat{A}^2$ is the high stressed (e.g. high speed) model transitions where the mean transition probability from the first state to the last state is $(0.1547 + 0.2111)/2 = 0.1829$.

- Estimated emission parameters:

$\hat{B}^1$ presents emission probabilities for the first output sequences (e.g. temperature) and $\hat{B}^2$ is for second output (e.g. vibration).

$$\hat{B}^1 = \begin{pmatrix} 0.8977 & 0.0512 & 0.0511 \\ 0.0557 & 0.8964 & 0.0479 \\ 0.0500 & 0.0505 & 0.8995 \end{pmatrix}$$

$$\hat{B}^2 = \begin{pmatrix} 0.7977 & 0.1523 & 0.0500 \\ 0.2038 & 0.7470 & 0.0492 \\ 0.1006 & 0.0494 & 0.8499 \end{pmatrix}$$

- Initial state distribution: (estimated as good health)

$$\pi = (0.9839 \quad 0.0161 \quad 0)$$

Fig. 5 presents the distance between the estimated parameters and the original parameters used in data simulation. The parameters estimated twice: the first one is with the bootstrap-IOHMM method where the model uses the data with replacement technique from the simulated for training and the second one is the classical method without bootstrap which uses all the data samples.

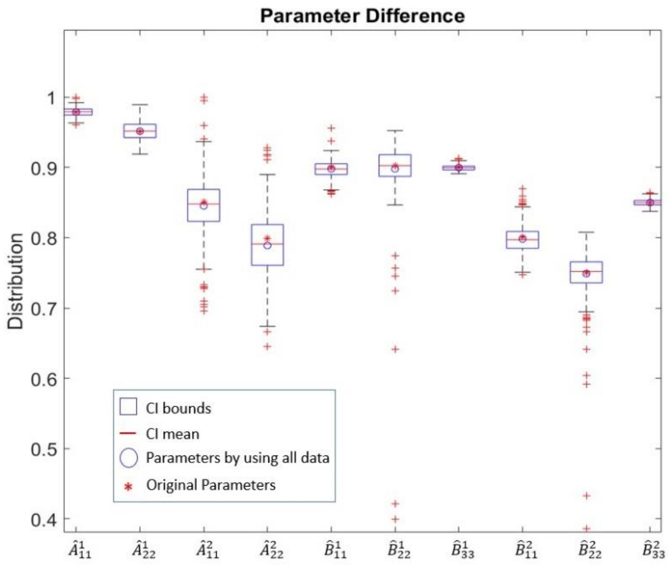

Fig. 5. Parameter distance between learned and original model parameters

The figure presents transition and emission parameter distances where only the diagonal probabilities except the absorbent states are compared. The blue box represents the CI bounds, the red line inside the box represents the CI mean and the star symbol represents the original parameters. The CI mean is the estimated parameters using the bootstrap-IOHMM method. There is another circle is inside each box which represents the second estimated parameters (without bootstrap). This time the IOHMM trained by all the simulated data (1000 sequences) while the bootstrap-IOHMM uses a selective approach. However, both the estimated parameters are very close to the original parameters. Therefore, we can say that the bootstrap method-IOHMM is very crucial and effective while a small data amount is available to estimate the parameters of a big data population.

The next section diagnostic the system health by using both the estimated parameters and explains the difference between them.

4.3.2 Diagnostic results

A new data set is given to the model to estimate the health state of the system. The diagnostic method uses the Viterbi algorithm and calculates the state distribution from the starting point to the breakdown point of the system. The method shows the estimated time about the system degradation stays in each state (Fig. 6).

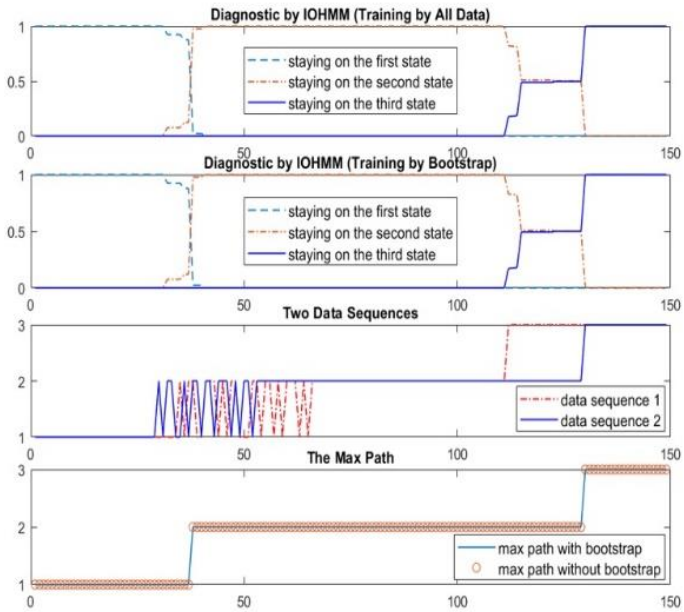

Fig. 6. Diagnostic over the time from start to end.

The first two parts of the figure are the estimated diagnostic using the models with and without bootstrap given two data sequences (in third part). Two diagnostic results show a similar transition from one state to another state, but distribution is not the same. Despite the fact, their distributions are not same they still give the same max path shown in fourth part of the figure. Degradation level of the system health can be found in this max path given by the Viterbi algorithm, which fulfils the objective of this paper.

This method could be very handy while the data sequence from health degradation is not easy to get, because degradation is a slow process that needs a long time to produce the data. So, the bootstrap-IOHMM can be used to estimate an effective diagnostic of system health.

5. CONCLUSIONS

This paper proposed a bootstrap-IOHMM method considering multiple operating conditions. 95% confidence intervals are given for each of the parameters of the IOHMM model. The estimated parameters used in the diagnostic application and show the results with and without applying the bootstrap method. The comparison between the results is explained to show the importance of the proposed method. This contribution could be used in the PHM domain to estimate different RUL considering the operating condition and control the system operation.

REFERENCES

Baum, L.E. and Petrie, T., 1966. Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics*, *37*(6), pp.1554-1563.

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

Bengio, Y. and Frasconi, P., 1995. An input output HMM architecture. In *Advances in neural information processing systems* (pp. 427-434).

Camci, F. and Chinnam, R.B., 2006, July. Hierarchical HMMs for autonomous diagnostics and prognostics. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings* (pp. 2445-2452). IEEE.

Dong, M. and He, D., 2007. A segmental hidden semi-Markov model (HSMM)-based diagnostics and prognostics framework and methodology. *Mechanical systems and signal processing*, *21*(5), pp.2248-2266.

Efron, B., AND G. Gong. 1983. A leisurely look at the bootstrap, the jackknife, and cross-validation. Amer. Statist. 37:36-48

Efron, B. and Tibshirani, R., 1986. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical science*, pp.54-75.

Felsenstein, J., 1985. Confidence limits on phylogenies: an approach using the bootstrap. *Evolution*, *39*(4), pp.783-791.

Kumar, A., Chinnam, R.B. and Tseng, F., 2019. An HMM and polynomial regression-based approach for remaining useful life and health state estimation of cutting tools. *Computers & Industrial Engineering*, *128*, pp.1008-1014.

Kim, Y.J. and Conkie, A., 2002. Automatic segmentation combining an HMM-based approach and spectral boundary correction. In *Seventh International Conference on Spoken Language Processing*.

Le, T.T., Berenguer, C. and Chatelain, F., 2015, January. Multi-branch hidden semi-markov modeling for RUL prognosis. In *2015 Annual Reliability and Maintainability Symposium (RAMS)* (pp. 1-6). IEEE.

Oates, T., Firoiu, L. and Cohen, P.R., 2000. Using dynamic time warping to bootstrap HMM-based clustering of time series. In *Sequence Learning* (pp. 35-52). Springer, Berlin, Heidelberg.

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), pp.257-286.

Salem, A.B., Muller, A. and Weber, P., 2007. Dynamic Bayesian Networks in system reliability analysis. In *Fault Detection, Supervision and Safety of Technical Processes 2006* (pp. 444-449). Elsevier Science Ltd.

Shahin, K.I., Mai, C. and Weber, P., 2019a, June. Input-output hidden Markov model for diagnosis of complex system. 13th *International Conference CIGI QUALITA 2019*, Canada.

Shahin, K.I., Simon, C. and Weber, P., 2019b, September. Estimating IOHMM parameters to compute remaining useful life of system. 29th European Safety and Reliability Conference, Germany.

Tobon-Mejia, D.A., Medjaher, K., Zerhouni, N. and Tripot, G., 2011, May. Hidden Markov models for failure diagnostic and prognostic. In 2011 Prognostics and System Health Managment Confernece (pp. 1-8). IEEE.

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