Degradation Modeling and Monitoring of Machines Using Operation-Specific Hidden Markov Models

Michael E. Cholette
Queensland University of Technology
2 George St. GPO Box 2434
Brisbane, Australia, 4001
michael.cholette@qut.edu.au

Dragan Djurdjanovic
University of Texas at Austin
1 University Station, C2200
Austin, Texas, 78712
dragand@me.utexas.edu

Abstract

In this paper, a novel data-driven approach to monitoring of systems operating under variable operating conditions is described. The method is based on characterizing the degradation process via a set of operation-specific hidden Markov models (HMMs), whose hidden states represent the unobservable degradation states of the monitored system while its observable symbols represent the sensor readings. Using the HMM framework, modeling, identification and monitoring methods are detailed that allow one to identify a HMM of degradation for each operation from mixed-operation data and perform operation-specific monitoring of the system. Using a large data set provided by a major manufacturer, the new methods are applied to a semiconductor manufacturing process running multiple operations in a production environment.

1 Introduction

In 1981, maintenance costs in the United States were estimated at 600 billion dollars, a figure doubled in the subsequent 20 years, with an estimated one-third of these costs wasted through ineffective maintenance [1]. Such staggering losses illustrate the importance of research in the maintenance of today's complex machinery and have motivated the exploration of condition-based maintenance (CBM), where the machine's condition information is inferred from the sensor readings that are considered to be indicative of machine health (vibrations, forces, voltage signals, etc.). Based on such condition information, cost-effective maintenance actions can be planned [2].

One of the main challenges in CBM is the quantitative description of the relationship between the equipment condition and the sensor readings. In a great majority of the relevant literature, the condition of the monitored system is related to sensor readings, such as accelerations, acoustic signals, forces and torques, via various signal processing and feature extraction methods [1]. Implicit in such work is an assumption that the sensor features themselves are *direct* indications of the state of health.

However, in highly complex engineering systems, such as semiconductor manufacturing equipment, automotive engines or other systems of interacting dynamic subsystems, it is not possible to firmly relate the machine condition with the available sensor readings. In such cases, either there are not enough sensor readings to establish a firm (deterministic) connection between the two, or the number of phenomena that influence the system condition is too large, making the relation between the sensor readings and the condition of the monitored system intractable. In such cases, one must acknowledge that the machine condition is an abstract entity that cannot be directly observed, but instead needs to be inferred from the available sensor readings. This inference must be made based on a stochastic relation that exists between the sensor readings the machine condition.

In the last two decades, hidden Markov models (HMMs) have been an important tool for modeling in CBM. Typically, HMMs are used in condition monitoring applications as non-stationary time series models of degradation indicators. Monitoring of the machine then consists of computing the likelihoods of new signals given these signal models. These likelihoods can then be employed to diagnose historical wear patterns directly [3, 4] or to detect deviations from the good-as-new tool [5]. Other examples of this classification-like approach can be found in Li et al. [6] and Purushotham et al.[7]. In contrast to such approaches, other HMM-based condition monitoring applications interpret the hidden state as an abstraction of the tool wear, taking the final state as the unacceptable "worn" state. Analysis of the hidden state can then be used for diagnosis and prediction of tool health [8, 9, 10]. Applications of HMM-based condition monitoring are most plentiful in rotating machinery [3, 4, 11], but have also found their way into monitoring of more complex systems, such as gear boxes [12], hydraulic pumps [13], antenna arrays [14] and even detection of terrorist networks [15].

Despite these successes, significant challenges remain for HMM-based modeling of degradation processes. One such challenge is accounting for operating regime dependent degradation dynamics [1]. Machines often operate in different regimes that result from variations in control

signals or environment. The dynamics with which equipment condition degrades can also vary with these operating regimes. Often there is a physical reason to expect different degradation dynamics and examples are plentiful. Varying loads on a bearing, different working materials for a drilling operation and highway vs. city driving for an automobile are but a few examples. Utilizing a degradation model that does not account for the effects of different usage patterns can result in overly-conservative diagnosis, missed detections and poor degradation prediction. The modeling of such regime-dependent degradation is one of the major challenges in CBM today [1]

In the HMM framework, operating condition variations are often handled by training an "average" model using data from various operating regimes of the monitored system, as was done in Ocak and Loparo [3]. Another approach was proposed by Bunks et al. [12], where the HMM states are combinations of the condition and the current operating regime. The authors showed how the vibrational signatures of a helicopter gear box varied under different torques, but that these changes could be distinguished from a set of faults. This classification was done using only the output densities and methods for estimating the state dynamics were not considered.

In this paper, a new scheme is proposed that tackles the problem of describing equipment condition and modeling of the dependency of degradation dynamics on the operating regimes of the monitored system. The new method models the degradation process through a collection of *operation-specific* hidden Markov models (HMMs), where the equipment conditions are hidden states. The framework extends HMM-based condition monitoring to machines with variable operating conditions and details identification methods for operation-specific Markov degradation models from operational data. Such models have been employed recently to conduct maintenance and product planning, but identification of such models was left open [16, 17, 18]. Furthermore, this paper details the application of the new identification and condition monitoring methodologies to a semiconductor manufacturing process running in a real fab. Such an application represents a significant increase in complexity from typical HMM applications in rotating machinery.

The remainder of this paper is organized as follows. In Section 2, the HMM-based modeling framework will be discussed and the identification methodology will be introduced. Section 3 will discuss the monitoring methodologies based on operation-specific HMMs. In Section 4 the operation-specific identification and monitoring methodologies are applied to a semicon-

ductor manufacturing process using a large production data set. Finally, Section 5 will detail conclusions of this work and avenues for future research.

2 Hidden Markov Model Based Degradation Modeling

The operation-specific HMM modeling paradigm is illustrated in Fig. 1. Instead of employing a single large degradation model (as proposed by Bunks et al. [12]), a multiple model framework is employed, where the degradation of the monitored system is assumed to depend on the particular operation conducted on it. Each operation that the system conducts emits a set of observations

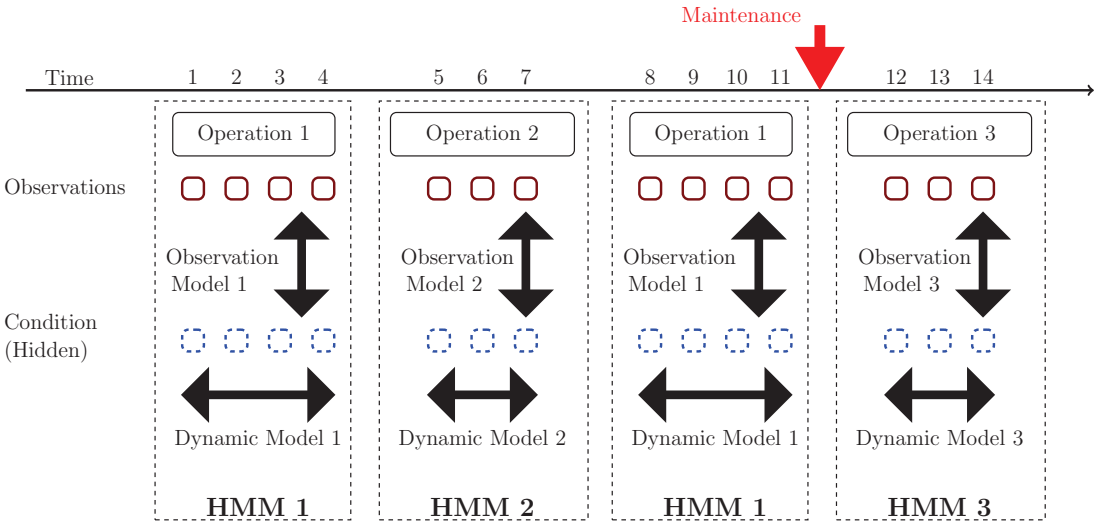

Figure 1: Generic modeling of degradation in an operation-dependent framework employed in this paper. The tool condition is thought of as an unobservable dynamic process and the sensors are modeled as observable emissions stochastically related to the unobservable process.

that are probabilistically related to the unobservable degradation state. After a number of operations, a maintenance action may be taken, which moves the system to a fresher state. Such an abstraction of the machine condition lends itself nicely to the HMM framework, with operation-dependent dynamics and emissions probabilities (i.e. a distinct HMM for degradation modeling in each operation).

In this paper, it is assumed that one has perfect knowledge of the operation regime at each instant in time. This is often appropriate for manufacturing applications, where the operation executed by the machine is known and available during training of the degradation models. Facilitating the paradigm illustrated in Fig. 1 for applications in which operating regimes are

not directly observable and have to be inferred from the available sensor readings is outside of the scope of this paper and will be considered in the future.

While the paradigm of Fig. 1 is conceptually elegant, the problem of estimation of the model parameters is complicated beyond what current HMM applications consider. The identification of the HMM must now be done with multiple HMM dynamics, which means that the number of parameters that must be estimated could be greatly increased. One also has the added constraint of continuous degradation, meaning that the end state of one operation-specific HMM needs to be the beginning state of the subsequent operation-specific HMM. This must be accounted for in the parameter identification procedure when mixed-operation data is used, precluding the application of well-known single HMM parameter identification methods.

In this section, the identification of operation-specific degradation HMMs will be addressed. First, we will briefly review the basic mathematical concepts related to the HMMs, after which new parameter identification algorithms will be discussed.

2.1 Hidden Markov Models

To enable the discussion of the work conducted in this paper, the HMM concept and some well-known computational solutions related to it are briefly discussed here. Much of the discussion in this section follows the seminal tutorial paper by Rabiner [19].

A HMM is a doubly-embedded, stochastic process that models the observation sequence as a consequence of an underlying Markov Chain. The states of the Markov Chain emit observations that can either be discrete or continuous in nature. For this paper, we limit ourselves to the discrete observation case. For an N state, M observation HMM, the observation sequence up to time T will be denoted as $\mathcal{O}_T = (o_1, o_2, \ldots, o_T)$ with $o_t \in \{\nu_1, \nu_2, \ldots, \nu_M\}$, and the corresponding state sequence will be denoted as $Q_T = (q_1, q_2, \ldots, q_T)$ where $q_t \in \{s_1, s_2, \ldots, s_N\}$, $t = 1, 2, 3, \ldots, T$. The underlying state sequence is assumed to evolve according to a Markov process and the transition probabilities are represented in matrix form

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix}$$

where $a_{ij} = P(q_t = s_j | q_{t-1} = s_i)$. For a HMM, these states are unobservable. Instead, observations are emitted from each state according to the conditional probability distribution,

$b_i(\nu_j) = P(o_t = \nu_j | q_t = s_i)$, which is represented as a matrix as well

$$B = \begin{bmatrix} b_1(\nu_1) & \dots & b_1(\nu_M) \\ \vdots & \ddots & \vdots \\ b_N(\nu_1) & \dots & b_N(\nu_M) \end{bmatrix}. \tag{1}$$

Clearly, The elements of A and B satisfy the constraints

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{2}$$

$$\sum_{j=1}^{M} b_i(\nu_j) = 1. \tag{3}$$

Together with the initial state distribution, $\pi \in \mathbb{R}^n$ where $\pi_i = P(q_1 = s_i)$ the HMM is usually denoted as the triplet

$$\lambda = (A, B, \pi).$$

The utilization of HMMs usually incorporates three common tasks:

1. **Evaluation** is the determination of the probability of an observation sequence given a HMM, denoted $P(\mathcal{O}|\lambda)$. This problem is common in classification-type tasks, such as fault diagnosis [3, 4], as well as speech and gesture recognition [20, 21]. An efficient procedure for solving this problem exists and is based on partial sequence probabilities called the forward variable

   $$\alpha_i(t) = P(o_1, o_2, \dots, o_t, q_t = s_i|\lambda) \tag{4}$$

   and backward variable

   $$\beta_i(t) = P(o_{t+1}, o_{t+2}, \dots, o_T|q_t = s_i, \lambda) \tag{5}$$

   which can be used to efficiently compute the probability of the sequence $\mathcal{O}_T$ using a recursive algorithm.

2. **Decoding** is the determination of the most likely state sequence given the observation sequence and a HMM. While there are many possible metrics for the most likely state sequence, the most common is the probability of the entire sequence, $P(Q_T|\mathcal{O}_T, \lambda)$. The maximization of this metric results in the well-known Viterbi algorithm which is commonly

utilized to compute the most likely hidden state sequence [22].

3. **Learning** is the determination of the HMM parameters that best explain a set of data. This is commonly done by maximizing the probability of the sequence of observations i.e.

$$\lambda = \arg\max_{\hat{\lambda}} P(\mathcal{O}|\hat{\lambda}).$$

This is by far the most difficult task, with the risk of many local optima and over-fitting. The algorithm used most often is the well-known Baum-Welch (BW) re-estimation, a version of which appears in Rabiner [19]. However, a number of other solutions exist, such as those based on maximum *a posteriori* estimation [23] or metaheuristics [24, 25, 26].

The aforementioned problems are well studied and documented in literature [19]. However, in the framework proposed in this paper, the observation sequence is emitted by a set of HMMs and the solutions to each of the above problems needs to be modified to account for the presence of multiple HMM dynamics. These modifications are the subject of the next section.

2.2 Operating Regime-Specific HMMs

Let us first formally describe the framework of operating regime-specific HMMs that were conceptually described in Fig. 1. The degradation state at time $t \in \{0, 1, 2, \ldots\}$ will be denoted as

$$q_t \in \{s_1, s_2, \ldots, s_N\}$$

with an initial distribution given by

$$\pi(0) = [\Pr(q_0 = s_1), \Pr(q_0 = s_2), \ldots, \Pr(q_0 = s_N)]\,.$$

Let

$$r : \mathbb{N} \to \{1, 2, \ldots, R\}$$

be a known function that describes the operating regime, $r(t)$, at time t. Also, let $\nu_j^{(\ell)}$, $j \in \{1, 2, \ldots, M_\ell\}$ denote the possible observation symbols for operation $\ell \in \{1, 2, \ldots, R\}$. Thus, the observation o_t satisfies

$$o_t \in \left\{\nu_1^{(r(t))}, \nu_2^{(r(t))}, \ldots, \nu_{M_{r(t)}}^{(r(t))}\right\}.$$

In many monitoring situations, the set of observation symbols are the same for all operating regimes, in which case one can use the more compact notation $\nu_j^{(\ell)} = \nu_j$.

The regime-dependent HMM set will be defined as the structure

$$\Lambda = (\pi(0), A_1, B_1, A_2, B_2, \ldots, A_R, B_R)$$

such that at any given time, t, the probability distribution of the states satisfies

$$\begin{aligned} \pi(t) &= [\Pr(q_t = s_1), \Pr(q_t = s_2), \ldots, \Pr(q_t = s_N)] \\ &= \pi(0) \cdot \left[\prod_{i=0}^{t-1} A_{r(i)} \right]. \end{aligned} \tag{6}$$

The probabilistic behavior of observable symbols $\nu_j^{(\ell)}$ corresponding to operation $\ell \in \{1, 2, \ldots, R\}$ is described by the relevant emission probabilities

$$b_i\left(\nu_j^{(\ell)}, \ell\right) = \Pr\left(o_t = \nu_j^{\ell} | q_t = s_i\right). \tag{7}$$

which yields the emission matrix

$$B_\ell = \begin{bmatrix} b_1\left(\nu_1^{(\ell)}, \ell\right) & \ldots & b_1\left(\nu_M^{(\ell)}, \ell\right) \\ \vdots & \ddots & \vdots \\ b_N\left(\nu_1^{(\ell)}, \ell\right) & \ldots & b_N\left(\nu_M^{(\ell)}, \ell\right) \end{bmatrix}. \tag{8}$$

From Equation (6), which defines the dynamics of the probability distribution of hidden states, it is evident that within periods of constant operating regimes, the state sequence exhibits the Markov property. Furthermore, continuity of degradation is enforced since the probability distribution of the hidden states at the end of operation $\ell = r(t-1)$ is the initial distribution of hidden states for the subsequent operation, $\ell = r(t)$. Equation (7) defines the operating regime-specific emission matrices that describe how the unobservable states are probabilistically related to the observations.

2.2.1 Evaluation and Decoding for Multiple HMM Sequences

For the case where $\mathcal{O}_T$ is emitted by multiple HMMs, the forward procedure typically used to compute $P(\mathcal{O}_T|\lambda)$ needs to be slightly modified. Given a common state space for all of the

models and assumed knowledge of $r(t)$, the forward variable is initialized as

$$\alpha_j(1) = \pi_j(0) \cdot b_j(o_1, r(1)) \tag{9}$$

and recursively computed as

$$\alpha_j(t+1) = \left(\sum_{i=1}^{N} a_{ij}^{(r(t))} \right) \cdot b_j(o_{t+1}, r(t+1)) \tag{10}$$

which is similar to the corresponding single HMM computation in [19], with the additional index $r(t) \in \{1, 2, \ldots, R\}$. Thus, $P(\mathcal{O}|\Lambda) = \sum_{j=1}^{n} \alpha_j(T)$, as in the single HMM case.

The computation of the backward probability, β, and Viterbi variables are modified in a similar manner, by simply replacing each term a_{ij} and $b_j(o_t)$ with $a_{ij}^{(r(t))}$ and $b_j(o_t, r(t))$, respectively.

2.2.2 Modified Re-estimation Procedure for Multiple HMM Parameter Identification

For the learning problem, the usual re-estimation formulae can be modified to account for the multiple sub-HMMs in the training sequence as

$$\hat{a}_{ij}^{(\ell)} = \frac{\displaystyle\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k} \delta_\ell(t) \alpha_i^k(t) a_{ij}^{(\ell)} b_j(o_{t+1}^k, \ell) \beta_j^k(t+1)}{\displaystyle\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k} \delta_\ell(t) \alpha_i^k(t) \beta_i^k(t)} \tag{11}$$

where

$$\delta_\ell(t) = \begin{cases} 1 & \text{if } r(t) = \ell \\ 0 & \text{otherwise} \end{cases}$$

$\ell \in \{1, 2, \ldots, R\}$ is the sub-HMM index, K is the total number of training sequences, while T_k and $P_k = P(\mathcal{O}_T^{(k)}|\Lambda)$ are the end time and probability of the training sequence k, respectively. Clearly, this estimation simplifies to the standard re-estimation equation when $R = 1$.

The equation for updating of the output probabilities, B_ℓ, can be modified in a similar

manner

$$\hat{b}_j(\nu_m, \ell) = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k} \delta_{\ell,m}(t) \alpha_j^k(t) \beta_j^k(t)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k} \delta_{\ell}(t) \alpha_j^k(t) \beta_j^k(t)} \tag{12}$$

where

$$\delta_{\ell,m}(t) = \begin{cases} 1 & \text{if } r(t) = \ell \text{ and } o_t^k = \nu_m \\ 0 & \text{otherwise} \end{cases}$$

Equations (10), (11) and (12) will be referred to as the *modified* Baum-Welch procedure in this paper.

Using the above re-estimation formulae, the parameters of the HMM can be estimated iteratively from an initial parameter set. However, the initial parameter selection can have a large influence on the resulting model, since the Baum-Welch re-estimation can only find a local optimum [19, 24, 25]. Methods for initializing the parameters of regime dependent HMMs will be discussed in the subsequent section.

2.3 An Identification Procedure for Regime Specific HMMs Based on a Genetic Algorithm

Most of the applications of HMMs so far have been of the classification-type, such as those in speech recognition tasks, where they were first widely applied. In classification-type tasks, training of a model that is close to the actual likelihood of the observation sequence is the primary objective. In such cases, some *a priori* knowledge combined with a series of random initializations is sufficient to produce an acceptable model.

However, this closeness in likelihood terms has been found to be relatively insensitive to the state transition probabilities, a_{ij} [27], indicating that small changes in the log probability can be the result of large changes in the estimation of a_{ij}. This, in turn, implies that getting trapped in local minima can result in poor estimation of a_{ij}.

As was discussed in Section 1, it is anticipated that the resulting model could be utilized in a maintenance decision making framework. Thus, it is desirable to closely pursue the model parameters (particularly the state transition parameters) through avoiding local optima. Furthermore, this local optima avoidance can lead to a more likely model, enhancing condition monitoring by providing a model that is a more accurate representation of the real process.

While there is no guarantee of a global optimum, the use of a metaheuristic is a possible

avenue for avoiding poor local optima. A variety of metaheuristic optimization methods have been applied to HMM identification, including Genetic Algorithms (GAs) [24, 28], Tabu Search (TS) [29] or a hybrid of metaheuristics and the Baum-Welch Algorithm [30, 25, 31, 32, 33]. However, these previous applications were identifying the model parameters from single HMM training sequences, or were focused on speech recognition applications, where no active model information was available and the primary objective was classification.

Hence, to the authors' knowledge, there is no tool available for local optima avoidance in the presence of operation-specific HMMs of the type proposed here. In light of this, a novel metaheuristic procedure was developed for the data-driven identification of parameters of regime-specific HMMs described in Section 2.2.

In the following, GA-based identification method will be detailed. A GA evolves populations of candidate solutions by combining and modifying portions of so-called "chromosomes" representing candidate solutions in the previous generation of solutions. Solutions that are better at optimizing the objective function are assigned higher fitness and are more likely to be selected for crossover, which mimics the natural selection process seen in nature. Further details on the basics of GAs can be found in [34, 35].

For training of regime-specific HMMs, each candidate solution is represented by the corresponding initial state probability distribution and the set of state transition and emission matrices. Let $\Lambda_1^{(k)}, \Lambda_2^{(k)}, \ldots, \Lambda_P^{(k)}$ denote the k^{th} generation of P candidate solutions. The fitness of any candidate solution is taken to be the likelihood of the sequence, $P(\mathcal{O}_T|\Lambda_p^{(k)})$.

The initial population of state transition matrices was created to give equal probability of all allowable transitions. For the emissions matrices, a manual segmentation of the training set was utilized to generate the initial population of emissions matrices. The segmentation was performed as follows: Assuming that all N states are present in an observation sequence of length T, the observation time series is divided into $\frac{1}{N}T$ length segments. Assuming $q_t = s_i$ is constant for each segment, the observation probabilities for a given state can be estimated simply as their relative frequency in each segment. Random perturbations of these initial emission probabilities were then used to ensure a sufficiently diverse initial population.

From the current generation, k, the $P/2$ $\Lambda_p^{(k)}$ candidate solutions with the highest fitness values were passed to the $k+1$ generation directly. In addition, the selected members were used as parents for creation of the remaining $\frac{P}{2}$ members of the next generation via crossover. The crossover operation was conducted as follows. Let $\Lambda_{p_1}^{(k)}$ and $\Lambda_{p_2}^{(k)}$ be two parents selected

for crossover. The crossover operation is then

$$\pi_{new}(0) = \alpha\pi_{p_1}^{(k)}(0) + (1-\alpha)\,\pi_{p_2}^{(k)}(0) \tag{13}$$

$$A_{new,\ell} = \alpha A_{p_1,\ell}^{(k)} + (1-\alpha)\,A_{p_2,\ell}^{(k)} \tag{14}$$

$$B_{new,\ell} = \alpha B_{p_1,\ell}^{(k)} + (1-\alpha)\,B_{p_2,\ell}^{(k)} \tag{15}$$

with

$$\alpha = \frac{\log\left[P\left(\mathcal{O}|\Lambda_{p_2}\right)\right]}{\log\left[P\left(\mathcal{O}|\Lambda_{p_1}\right)\right] + \log\left[P\left(\mathcal{O}|\Lambda_{p_2}\right)\right]}, \tag{16}$$

which is similar to the crossover operation in [24]. This chromosome crossover emphasizes the genetic material of the parent with the higher fitness function (parent HMM with the higher corresponding likelihood of the observation sequence). In addition, this crossover mechanism ensures that the new $\pi_{new}(0)$, $A_{new,\ell}$ and $B_{new,\ell}$ satisfy the stochastic constraints.

After crossover, mutation was set to occur with a probability, p_{mutate}. When mutation occurs, the mutation operation was performed as follows

1. Randomly select a regime, ℓ, for which the parameters will be mutated.

2. Let $a_{ij}^{(\ell)}$ be the i^{th} row and the j^{th} column of the state transition matrix corresponding to regime ℓ. Randomly generate a row index, i and two column indices j_1 and j_2 from the non-zero elements in row i. Set $a_{ij_1}^{(\ell)}$ to $a_{ij_2}^{(\ell)}$ and $a_{ij_2}^{(\ell)}$ to $a_{ij_1}^{(\ell)}$.

3. Repeat (2) for the emissions matrix.

4. Let $\pi_j(0)$ be the j^{th} element of the initial state probability vector. Randomly generate two indices j_5 and j_6. Set $\pi_{j_5}(0)$ to $\pi_{j_6}(0)$ and $\pi_{j_6}(0)$ to $\pi_{j_5}(0)$.

This training procedure is summarized in Fig. 2.3, with the parameter choices enclosed in the dashed boxes. The computational cost of the algorithm can be understood in terms of the number of BW iterations required, which dominate the computational intensity. For the GA, the number of BW iterations is $N_B P + \frac{P}{2}(k-1)\,N_B$, where k is the number of generations at training termination. This will likely be significantly higher than BW training alone (though hopefully achieving a more likely parameter set). We will return to this issue when we present an example of the parameter identification algorithm in Section 4.1.

Finally, a note should be made on the selection of the parameters. In this work, no effort was made to optimize the selection of these parameters, though there is a significant body of literature devoted to the selection of GA parameters to enhance exploration capabilities and accelerate the convergence (see [36, 37] for example). Instead of optimizing the input parameters, we will investigate the sensitivity of the identification to the population size and number of BW iterations in Section 4.1.

3 Machine Monitoring using Multiple Hidden Markov Models

Once the degradation model is identified, a number of methods can be utilized to monitor the condition of the machine, including utilization of the probabilities of the newly arrived observation sequences or utilization of the probabilities of the most degraded state of the HMM (computed using the Viterbi procedure).

In literature, the most frequently utilized procedures for HMM-based monitoring employ the likelihoods of observation sequences. In the classification-type approaches, a variety of HMMs are trained on normal *and* faulty data and a test observation sequence is assigned to the class whose HMM has the highest likelihood of emitting that sequence.

In the case that one has identified only a HMM of normal machine operation, λ_{normal}, one can compute $P(\mathcal{O}|\lambda_{\text{normal}})$ of subsequent observation sequences to provide a measure of "closeness" to normal behavior. If the system follows the nominal HMM dynamics identified from the training data, the log likelihood of the test observation sequence will linearly decrease with the length of the sequence [38]. Brown et al. [38] use this property and report a HMM-based monitoring strategy that accounts for different sequence lengths by monitoring differential changes in the probabilities of the observations sequences. Effectively, this can be done by looking at the slope of the log-likelihood trace of the observation sequences and interpreting deviations from the nominal line as changes in the condition of the monitored machine. However, in [38] the authors realize monitoring using only one HMM (one operating regime), whereas our goal is monitoring in the presence of multiple HMMs (multiple operating regimes). In this case, the slope of the log-likelihood line will depend on the underlying HMM (operating regime) and each HMM slope must be examined independently.

To formulate a monitoring strategy, we will use knowledge of the current operation and normalization of the operating regime dependent slope to facilitate monitoring in the presence

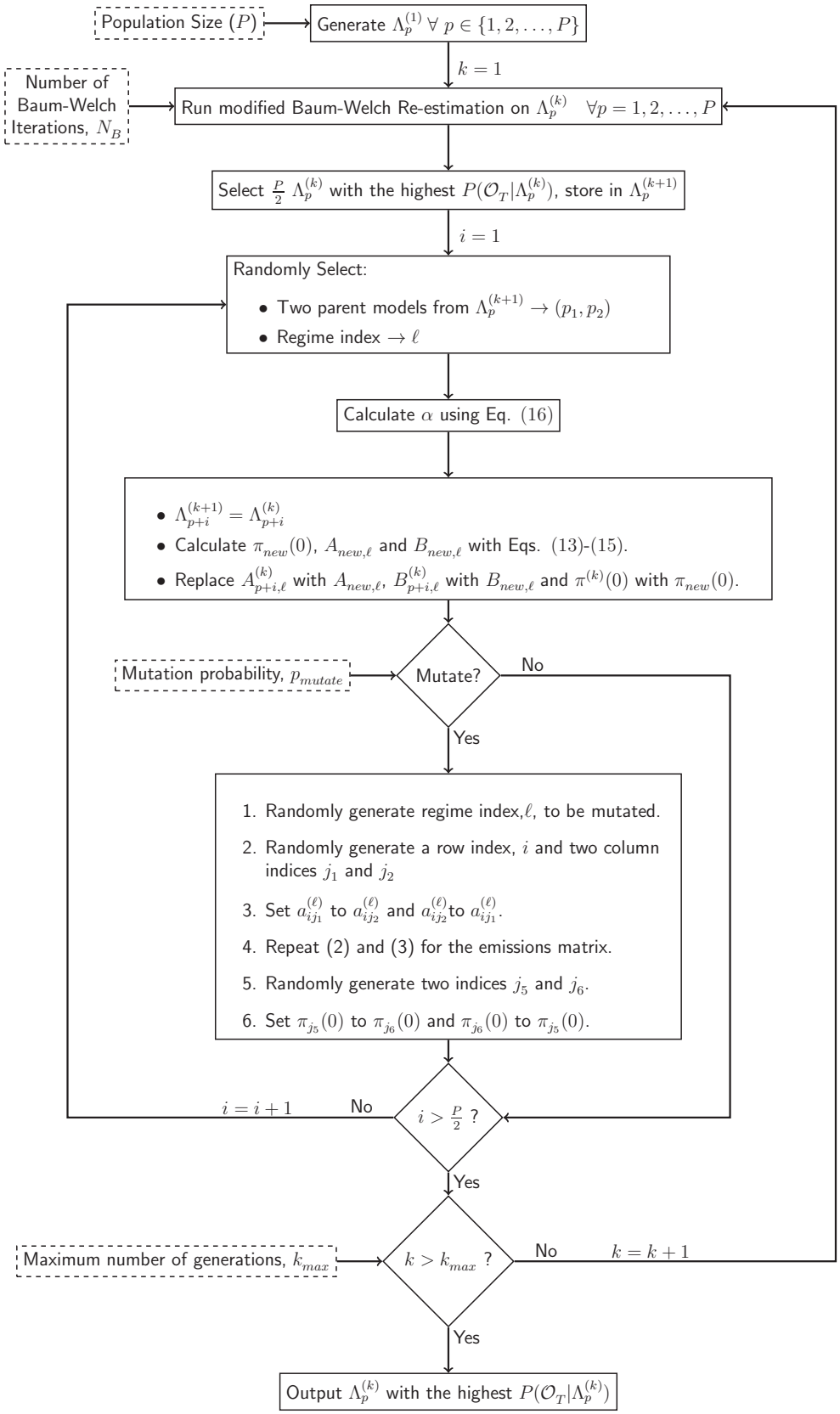

Figure 2: Flowchart for GA training from an observation sequence emitted by multiple HMMs.

of multiple operations. Normalization will be performed as follows. Let T_i denote the most recent termination time of a particular operation such that $r(t) = \ell \quad \forall t \in \left[T_{i-1}, T_i\right]$. The mean slope of the log-likelihoods for that operation in the period $\left[T_{i-1}, T_i\right]$ can be computed as

$$s_{T_i}^{(\ell)} = \frac{1}{T_i - T_{i-1}} \sum_{t=T_{i-1}}^{T_i} \left(\log\left[P(\mathcal{O}_{t+1}|\Lambda_\ell)\right] - \log\left[P(\mathcal{O}_t|\Lambda_\ell)\right]\right) \tag{17}$$

and the normalized slope during that period of time can be defined as

$$k_{T_i} \triangleq \frac{s_{T_i}^{(\ell)} - \mu_\ell}{\sigma_\ell} \tag{18}$$

where μ_ℓ and σ_ℓ are the mean and standard deviation of the slope for operation ℓ observed within the training data set[1]. Nominally, these normalized slopes should be centered around zero, with unit variance and their deviation from such behavior would signal an anomaly.

4 Results

In this section, the newly proposed methodology for the identification of the parameters of multiple HMMs will be applied to a synthetic problem and compared with BW training alone. In addition, the new HMM identification and monitoring algorithms will be applied to a data set emitted by a Plasma-Enhanced Chemical Vapor Deposition (PECVD) tool operating in a major semiconductor manufacturing facility.

4.1 A Case Study of the GA-based HMM Parameter Estimation Procedures Using Synthetic Observation Sequences

The new procedure for identifying operating regime-specific HMMs was tested on a synthetically generated set of 5000 sequences, each consisting of 10 individual observations. The synthetic model consisted of two, four state, left-right HMMs with state transition matrices

$$A^{(1)} = \begin{bmatrix} 0.8 & 0.2 & 0 & 0 \\ 0 & 0.7 & 0.3 & 0 \\ 0 & 0 & 0.81 & 0.19 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A^{(2)} = \begin{bmatrix} 0.90 & 0.10 & 0 & 0 \\ 0 & 0.76 & 0.24 & 0 \\ 0 & 0 & 0.51 & 0.49 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

[1]The training dataset is assumed to be representative of normal system behavior

Each of the 5000 sequences was generated from the initial state distribution $\pi(0) = [1, 0, 0, 0]$, which was assumed known (thus it was not estimated using the identification procedures). The number of observation symbols in the discrete observation alphabet, $o_t \in \{\nu_1, \nu_2, \ldots, \nu_M\}$, was varied from $M = 5$ to $M = 30$ to explore the effects of increasing the number of identified parameters. Obviously, a larger number of observation symbols requires identification of a larger number of emission probabilities. The emission probabilities for the synthetic models can be seen in the appendix.

In the following, the GA procedure is compared with results obtained using the BW re-estimation alone. Both the baseline BW procedure and the GA were initialized using the manual segmentation procedure described in Section 2.3, which is one of many possible initialization methods found in HMM literature [19, 39]. In the case of the GA, random perturbations from this initialization were used to generate the initial population of candidate solutions.

In this paper, no effort is made to optimize the input parameters of the GA method. Instead, two experiments were be conducted to explore the robustness of the optimization results to the user-selected parameters. In Experiment 1, the effect of the population size, P is explored by setting $N_B = 1$ and varying P from 10 to 50. In Experiment 2, the effect of changing N_B is explored by fixing the population at $P = 30$ and varying N_B from 1 to 5. In both experiments, $p_{mutate} = 0.2$ and $k_{max} = 1000$. For this example, an additional stopping criterion was used. When the log likelihoods of the best model did not change for 20 generations, training was terminated. Thus, k_{max} was not achieved since this condition was satisfied with less than 200 iterations in all instances.

Several metrics were used to evaluate the quality of identified model. One way of comparing the solution quality is through the log likelihood of the training sequence for the identified model. The final log-likelihoods were compared using the metric

$$\Delta \log[P(\mathcal{O}_T|\hat{\Lambda})] = \log[P(\mathcal{O}_T|\hat{\Lambda})] - \log[P(\mathcal{O}_T|\Lambda)] \tag{19}$$

where Λ is the actual (synthetic) model and $\hat{\Lambda}$ is estimated model identified using one of the new procedures. Clearly, if the identified model has *exactly* the same probability, one should see $\Delta \log[P(\mathcal{O}_T|\hat{\Lambda})] = 0$. Due to the presence of randomness in the GA training procedure, it was repeated 10 times, after which the median, maximum and minimum of $\Delta \log[P(\mathcal{O}_T|\hat{\Lambda})]$ are reported.

The quality of the parameters of the operation-specific HMMs was also evaluated by examining the norm of the difference between the corresponding actual (synthetic) and estimated matrices

$$\Delta_A = \sum_{r=1}^{R} \|A_{r,actual} - \hat{A}_r\| \tag{20}$$

$$\Delta_B = \sum_{r=1}^{R} \|B_{r,actual} - \hat{B}_r\|. \tag{21}$$

which provides a measure of the total estimation error over all R sub-HMMs. If the identification were perfect, both metrics would be identically zero.

The results for Experiment 1 can be seen in Figs. 3,4 and 5. Each show the median, maximum and minimum of the metrics from Eqs. (19), (20) and (21). The red line indicates the median of the 10 trials and the black "whiskers" indicate the extreme points. In all cases, the identified models with the value closer to zero are superior models. It can be clearly seen from Fig. 3 that the GA method achieves log likelihoods that are consistently closer to the actual model likelihood than the Baum-Welch only baseline. In Fig. 4 one can see that the GA consistently outperforms the baseline in identifying the state transition matrix. Finally, Fig. 5 shows that the estimates of the emission matrix are similar for both the GA and the baseline. Thus, the GA achieves more likely models, largely through superior estimation of the state transition matrix. Furthermore, the results are achieved over a range of settings of the parameter P.

The results for Experiment 2 are given in Figs. 6,7 and 8. It can be clearly seen from Fig. 6 that the GA method achieves log likelihoods that are consistently closer to the actual model likelihood than the Baum-Welch only baseline. From these three figures it is once again visible that the GA achieves more likely models, largely through superior estimation of the state transition matrix and that the results can be achieved for a range of settings of the parameter N_B.

An indication of the computational burden of each of the algorithms can be examined by looking at the number of BW iterations required. Both the GA and baseline training procedures were implemented in MATLAB on a Windows PC. The baseline algorithm required between 20 and 40 BW iterations while the GA method required between 300 and 2500 BW iterations. In terms of time elapsed, this translated to less than 10 minutes for the baseline while the TS and

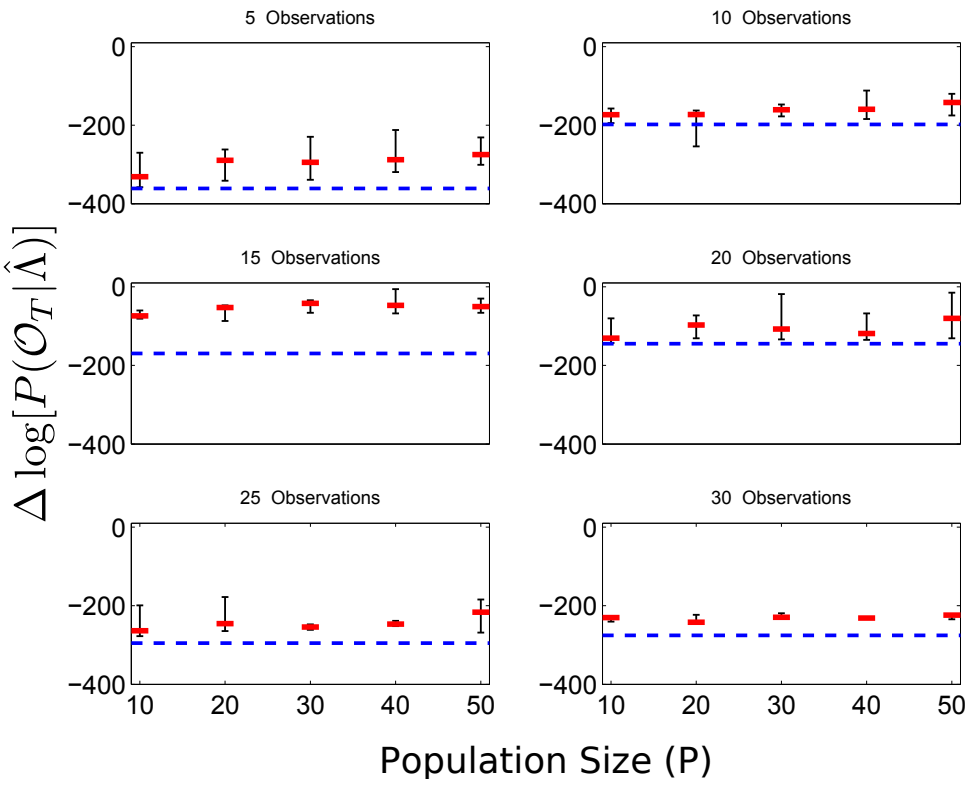

Figure 3: The difference in log-likelihood between the identified models and the actual model for different settings of the GA population size (Experiment 1). If the identification were perfect, the points would form a horizontal line at zero.

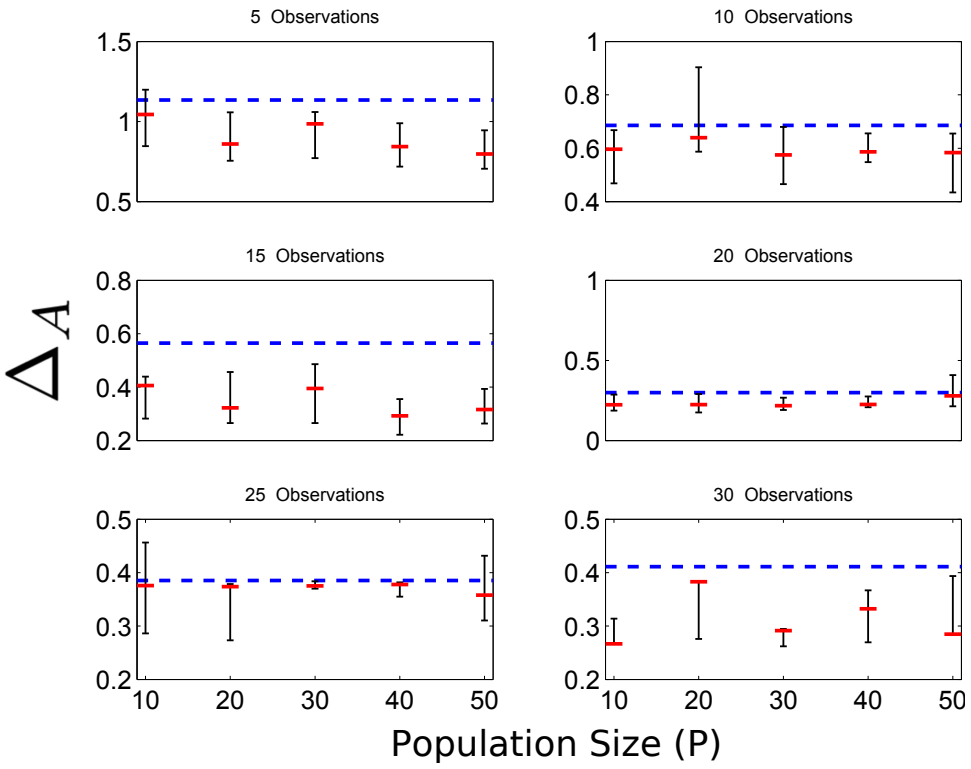

Figure 4: Sum of the norms of the differences between the estimated and actual state transition matrices for Experiment 1.

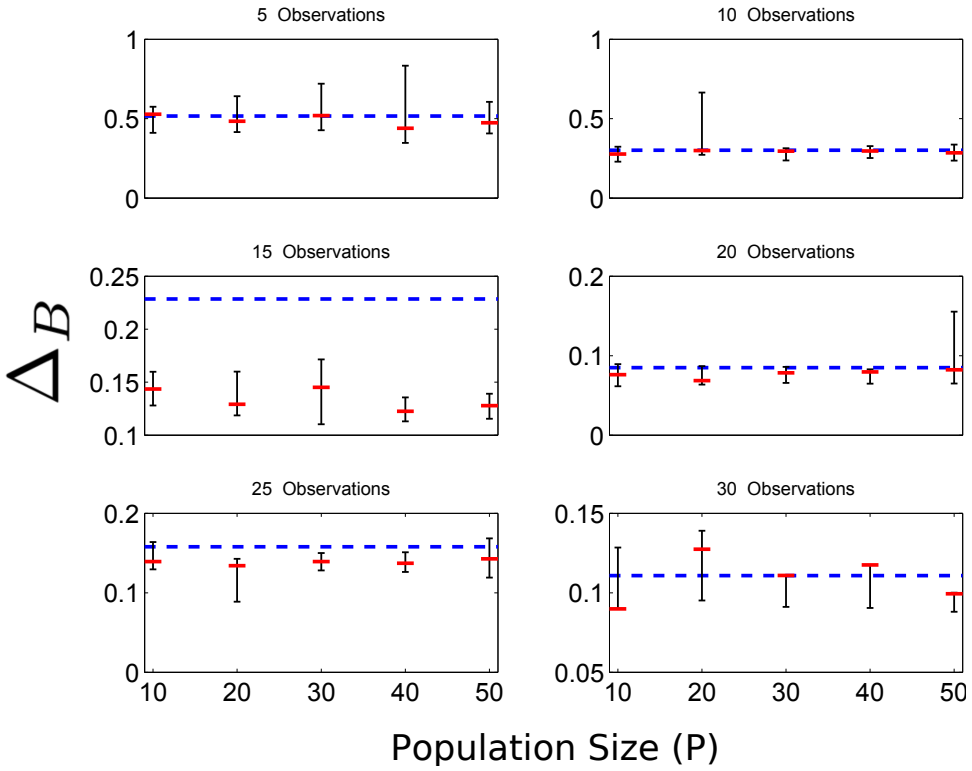

Figure 5: Sum of the norms of the difference between the estimated and actual emission matrices for Experiment 1.

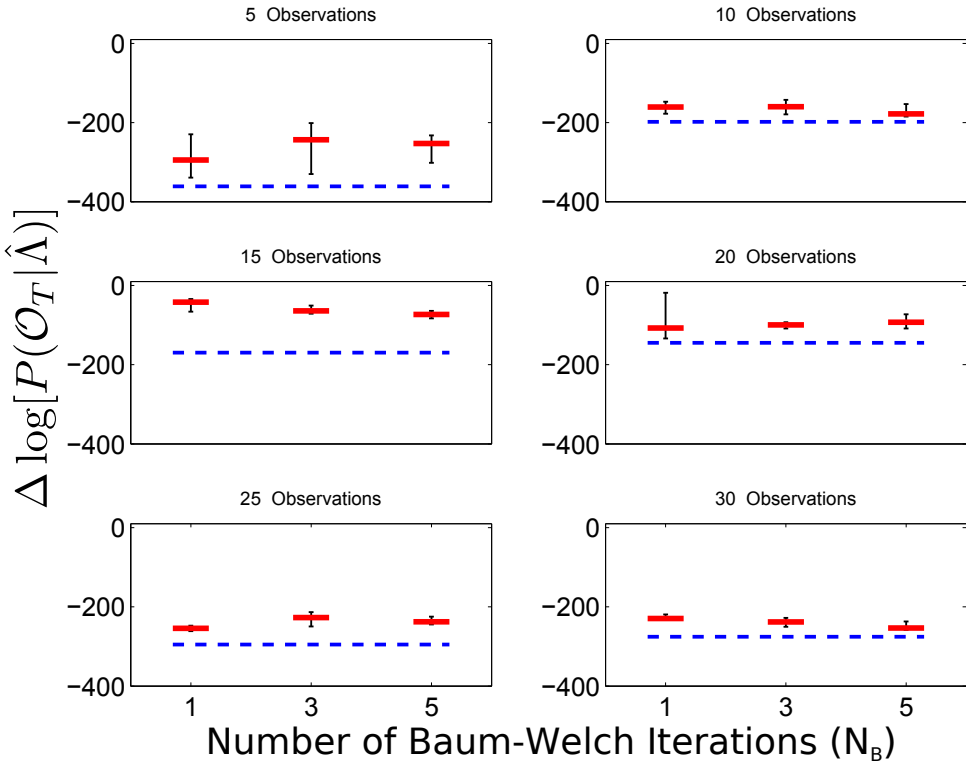

Figure 6: The difference in log-likelihood between the identified models and the actual model for different settings for the number of Baum-Welch iteration (Experiment 1). If the identification were perfect, the points would form a horizontal line at zero.

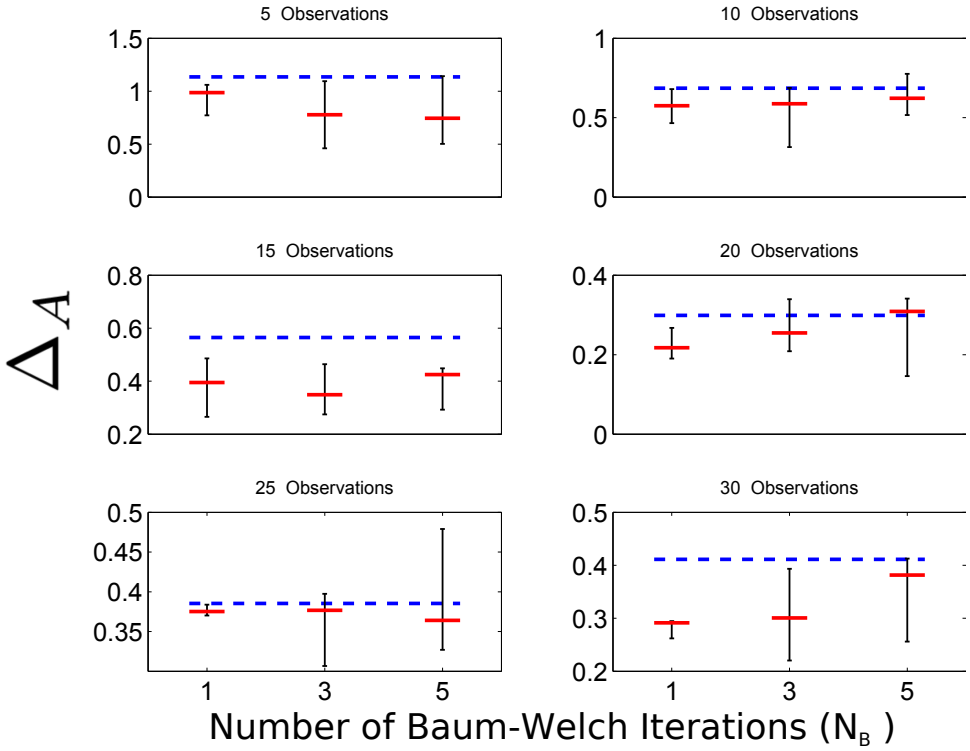

Figure 7: Sum of the norms of the differences between the estimated and actual state transition matrices for Experiment 2.

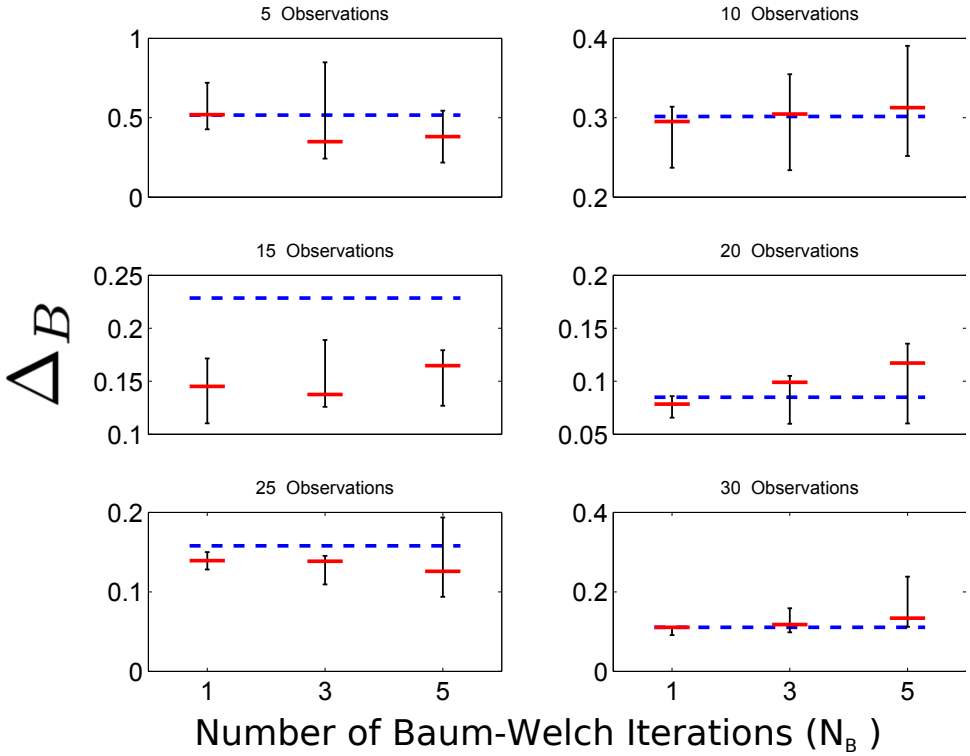

Figure 8: Sum of the norms of the difference between the estimated and actual emission matrices for Experiment 2.

GA methods each required between 20 minutes and 13 hours. Obviously this time is dependent on the size of the model (number of observations, in this case) and the GA population size. However, it is clear that the new identification methods provide superior solutions for a variety of parameter settings, at the cost of additional computational time.

4.2 Operating Regime-Specific Degradation Modeling and Monitoring of a Semiconductor Manufacturing Process

Plasma-Enhanced Chemical Vapor Deposition (PECVD) tools are used for depositing thin films onto silicon wafer substrates, which is one of the crucial steps in manufacturing. Inside a PECVD tool chamber, reactive gases pass over silicon wafers and react on their surfaces to form a thin layer. The gases are excited through radio frequency (RF) electrical power that creates plasma which allows the reaction to take place at lower temperatures. A single PECVD tool can run numerous operations. For instance, a variety of film thicknesses are often deposited using the same tool.

The operation of the PECVD tool in a typical semiconductor manufacturing facility consists of a number of consecutive depositions of films of different thicknesses. After a certain total amount of film deposition, an *in situ* clean is performed, where a set of reactive cleaning chemicals flow into the chamber to remove the deposition products from the chamber surfaces. Each deposition thickness is expected to degrade the tool differently, thus benefiting from the operation-specific modeling proposed in this paper.

In this section, the degradation modeling and monitoring methods developed in the paper will be applied to a PECVD tool operating in a major 300mm semiconductor manufacturing facility. The tool will be briefly described as will the selection of sensor features that will be used as the "observations" in the method proposed in this paper. Subsequently, the identification of the parameters of the HMM will be discussed along with some discussion of the resulting model. Finally, the operation-specific HMMs will be utilized in the monitoring scheme proposed in Section 3 for a period of real production that was not used for training the model. Along the way, the alarms that result from the monitoring method will be compared to the maintenance logs and quality measurements (metrology).

4.2.1 Data Collection and Feature Selection

A PECVD tool is composed of a reaction chamber, RF plasma generation system, gas delivery system, wafer load locks, and a robotic arm to carry wafers to and from the tool. The RF matching network that generates plasma by sending high-frequency energy through two matching capacitors (load and tune capacitors) which control the power delivered to the chamber. By varying their capacitances, the capacitors try to match the impedance of the circuit to the impedance of the chamber and thus deliver maximum RF power to the deposition gases. The RF energy excites the flowing gas into the plasma state necessary for lower temperature depositions. The gas delivery system consists of mass flow controllers for each gas used in various depositions and the film thickness is controlled by modifying the deposition time. A control valve controls the chamber pressure and evacuates deposition gases from the chamber. Temperature controlled top and lower chamber plates enclose the chamber and the walls are heated to minimize on- wall deposition and speed up the reaction during the *in situ* cleaning process.

From the various systems of the PECVD tool, a variety of standard sensors are available that record the capacitor positions, RF power, pressures, temperatures, gas flows, etc. These sensor signals were collected over a period of more than 6 months (due to the proprietary nature of the data, we cannot be more specific). The signals were collected at a 10Hz sampling rate and the total number of wafers in the data set was over 110,000.

A set of dynamic features, such as signal rise times, overshoots, settling times and amplitudes of various events during the deposition process were extracted from the sensor signals. A subset of these features were selected using a linear discriminant analysis (LDA) procedure [40] that yielded features that were statistically altered the most by the *in situ* cleans[2]. The argument for feature selection based on the *in situ* clean sensitivity was that such sensitive features would have a stronger connection with the hidden degradation state of the tool [41]. A stronger connection between the observables and hidden states can ease the process of identifying the operation-specific degradation models. The space of features selected by through LDA was then discretized by a self-organizing map (SOM) [42] to generate a discrete set of observable symbols (please note that a host of other discretization techniques could have been used to discretize the feature space).

In the conceptual illustration of Fig. 1, the "observations" will be a set of sensor features, the "operations" refer to deposition thicknesses and the "maintenance operations" correspond

[2]For further detail on the tool operation, feature extraction and feature selection, we refer the reader to [41]

to the *in situ* cleans.

During the period of signal collection, the tool was depositing films on standard 300mm silicon wafers of the same chemistry, but of significantly different thicknesses. The four most common deposition thicknesses were included in the modeling (the thickest recipe was approximately 20 times thicker than the thinnest recipe).

4.2.2 Identification of Operation-Specific HMMs

Using the GA method, the HMMs for the four thicknesses were identified from the data emitted during tool operation that was deemed acceptable by the tool operators. This data corresponded to several weeks of tool operation during which no unacceptable behavior or metrology excursion events were observed. The parameters for training were $P = 60$, $N_B = 2$, $p_{mutate} = 0.2$ and $k_{max} = 100$. These parameters were selected on an *ad-hoc* basis since the GA method was experimentally found to be relatively robust to GA parameter selection (see Section 4.1). The GA training procedure was repeated 10 times and the trained model with the highest log-likelihood was selected as the basis for monitoring. Given these parameters and the training set, training each of the 10 HMMs required about 9,000 BW iterations. This required about 2 hours of runtime for the MATLAB running on a Windows PC.

It is informative to examine the identified model. As the first point of examination, the log-likelihoods of the observation sequences evolution throughout the tool operation were observed. A plot of a portion of the log-likelihoods can be seen in Fig. 9. As expected, the log-likelihoods have relatively consistent decreasing linear trends[3], with clear operating regime-specific slopes, lending evidence to the benefits of operation-specific modeling.

The dynamics of the evolution of the hidden states and its dependence on the operating regime offer another point of examination. Figure 10 shows the probability of the most degraded state, $P(s_t = s_N|\lambda_r)$, for each operation-specific HMM, under the assumption of a single operating regime. The final state probability $P(s_t = s_N|\lambda_r)$ increases more quickly for thicker films, which is consistent with intuition, since the deposition of thicker films requires the flow of the deposition gases over a longer period of time leading to more pronounced degradation of the tool. Both of these observations provide evidence that the degradation process between *in situ* cleans is modeled well using the framework proposed in this paper.

[3]Examining other traces between *in situ* cleans reveals similar behavior

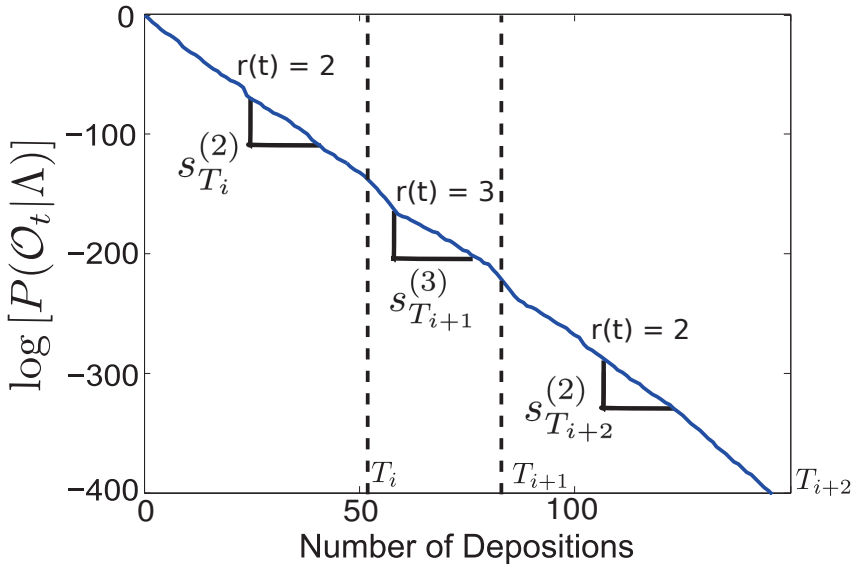

Figure 9: Log-likelihood trace for the PECVD tool. In this sequence, operations 2 and 3 are run on the tool and the different log likelihood slopes corresponding to the two different operations are visible.

4.2.3 Monitoring of the PECVD Tool

The subsequent production time period was monitored using Exponentially Weighted Moving Average (EWMA) control charts of the normalized log-likelihood slopes that are defined by Eq. (18), with the EWMA memory factor set to 0.2 and were correlated to the to the maintenance logs and metrology data that was recorded over the monitoring time frame. The EWMA chart of the normalized log-likelihood slopes with the 4σ control limits can be seen in Fig. 11. Only the lower control limit is shown, since the upper limit is never exceeded. The dashed line labeled "Training" indicates the end of the data used to identify the nominal degradation model. Six events that occurred during the monitoring period can be seen in Fig. 11. In the following, each event will be discussed in conjunction with the available maintenance and metrology data.

Event 1: The first event is a cluster of out-of-control points before the first long-term maintenance event. During this period, there are no maintenance log entries and the metrology is within limits. After discussion with those familiar with the PECVD tool, it is believed that the out-of-control points are related to particle formations, whose presence was confirmed some days later. Essentially, it is believed that deposition compounds accumulate on the chamber wall and periodically flake off, causing a sudden change in the electrical impedance of the chamber. This new impedance results in changes in the capacitors used in the RF power system matching network. Linear discriminant analysis revealed that, indeed, the RF network parameters changed

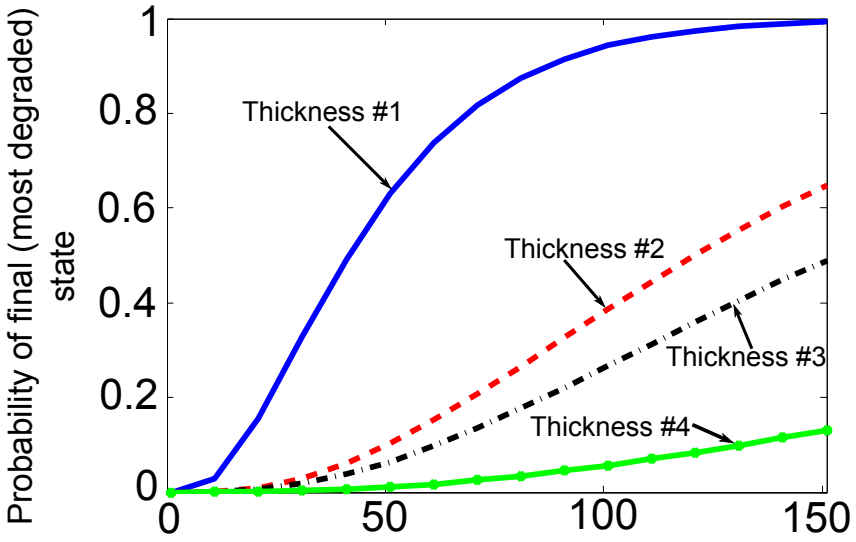

Figure 10: $P(s_t = s_N|\lambda_r)$ assuming a single-operation is run. Note that the thicker films approach s_N (the most degraded state) more quickly.

the most during this period of time. Gradually, through more depositions, the gap formed by the flaking event is filled, which causes the chamber impedance and, consequently, the matching network capacitors to return to their pre-flake value. Such a scenario is supported by the analysis of particle monitoring wafers (PMONs), used for monitoring particle contamination in the chamber. In the PMON logs, it is evident the PMON wafers that were passed through the tool near that time had several times more defects than PMONs corresponding to time intervals that appear in control in Fig 11.

Event 2: The out-of-control points in the box labeled "particle failures" in Fig. 11 correspond well with several weeks of tool downtime caused by defects seen on PMONs. The presence of particles in the chamber was later confirmed via laser scattering based particle monitoring. There is a noticeable shift downward in the EWMA chart immediately after the maintenance event and numerous particle failures appear in the maintenance logs during that time.

Event 3: An out-of-control point appears soon after Event 2. While there is no maintenance log entry for this time, the PMON close to this time has approximately 1.5 times the maximum particle count seen during the time intervals that appear to be in control. Nonetheless, these counts were still within the tolerance limits and did not stop production.

Event 4: A refractive index failure is reported in the maintenance logs on a film thickness that was not included in the monitoring procedure. Process control adjustments dealt with it

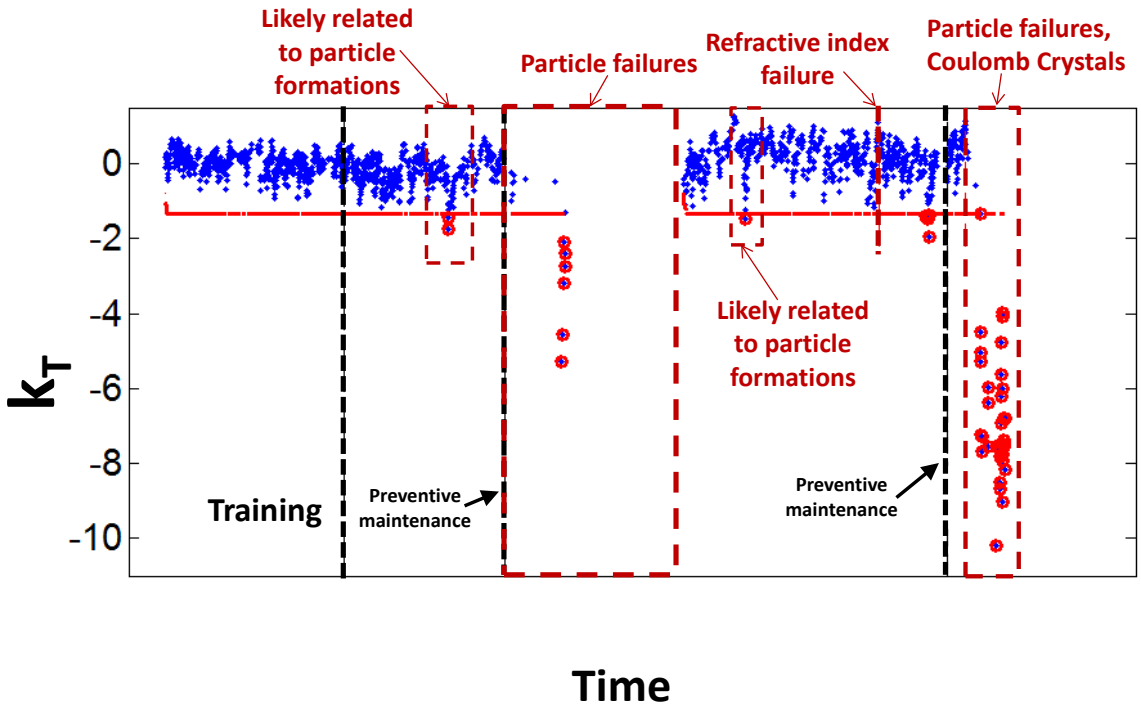

Figure 11: EWMA control chart of k_T with time period labels. Each period will be discussed in detail.

successfully and consequently there was no prolonged downtime of the tool.

Event 5: A cluster of out-of-control points appears just prior to the second preventive maintenance with no maintenance log entry. Once again, the PMON near this time shows counts that are elevated, but within tolerance.

Event 6: After another preventive maintenance event, the maintenance logs note a number of particle failures and "plasma formations." There is a clear downward trend in the EWMA chart which begins very close to the time when the particle failures on PMON wafers occur in the logs. A clear out-of-control cluster of points is evident during much of the period when the plasma formations are noted in the logs. Furthermore, PMONs near this time have order-of-magnitude elevated counts as well. After consultation with two experts, the so-called plasma formations were found to be the result of improper evaporation of the deposition product (in this case, Tetraethyl Orthosilicate, commonly referred to as "TEOS"). This improper evaporation resulted in liquid phase TEOS at the showerhead which caused a phenomenon known as Coulomb Crystals [43].

5 Conclusion

This paper presented a novel, context-dependent degradation modeling and monitoring methodology based on the concept of Hidden Markov Models (HMMs). The methodology utilizes information about known operational conditions to train a series of operation-specific HMMs, defined on the same set of hidden states representing degradation states of the monitored system. The HMMs of the degradation model were identified via a newly proposed procedure that uses a metaheuristic evolution of the initializations of a Baum-Welch algorithm, which was modified to enable parameter identification of interconnected, operating regime specific, degradation HMMs.

Using the aforementioned degradation model, a context-dependent monitoring methodology was presented based on the slopes of the log-probability traces of observations emitted by the monitored system. The monitoring method was demonstrated on a semiconductor manufacturing tool, operating in a production environment over a period of several months. Out-of-control events in the HMM monitoring were shown to be consistent with the maintenance and metrology events.

Several potential areas of future work exist. Extension of the newly introduced method for identifying degradation models to account for continuous observation densities could yield more accurate models by avoiding the discretization errors. Another potential improvement would be a description of the confidence in the identified HMM parameters. Understanding uncertainties in the observation and state transition probabilities yielded by the training algorithm are of high importance to the subsequent maintenance decision-making based on the diagnostic and predictive information obtained from such models.

6 Acknowledgements

This work was supported in part by the International SEMATECH Manufacturing Initiative. The authors would also like to thank Andy Filler, Alan Fritz, and Toysha Walker of Micron Technology, as well as Dr. Edward Augustiniak of Lam Research and John Rasberry for their help throughout the research that led to this paper.

References

[1] A. Heng, S. Zhang, A.C.C. Tan, and J. Mathew. Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing*, 23(3):724–739, April 2009.

[2] A.K.S. Jardine, D. Lin, and D. Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7):1483–1510, October 2006.

[3] H. Ocak and K.A. Loparo. HMM-Based Fault Detection and Diagnosis Scheme for Rolling Element Bearings. *Journal of Vibration and Acoustics*, 127(4):299–306, August 2005.

[4] Litao Wang, Mostafa G. Mehrabi, and Elijah Kannatey-Asibu. Hidden Markov Model-based Tool Wear Monitoring in Turning. *Journal of Manufacturing Science and Engineering*, 124(3):651, 2002.

[5] H.M. Ertunc and C. Oysu. Drill wear monitoring using cutting force signals. *Mechatronics*, 14(5):533–548, June 2004.

[6] Z. Li, Z. Wu, Y. He, and C. Fulei. Hidden Markov model-based fault diagnostics method in speed-up and speed-down process for rotating machinery. *Mechanical Systems and Signal Processing*, 19(2):329–339, March 2005.

[7] V. Purushotham, S. Narayanan, and S.A.N. Prasad. Multi-fault diagnosis of rolling bearing elements using wavelet analysis and hidden Markov model based fault recognition. *NDT & E International*, 38(8):654–664, December 2005.

[8] L. P. Heck and J. H. McClellan. Mechanical system monitoring using hidden Markov models. *ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing*, 3:1697–1700, 1991.

[9] L. Atlas, M. Ostendorf, and G.D. Bernard. Hidden Markov models for monitoring machining tool-wear. In *Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing.*, volume 6, pages 3887–3890, Istanbul, Turkey, 2000. IEEE.

[10] X. Zhang, R. Xu, C. Kwan, S. Y. Liang, Q. Xie, and L. Haynes. An integrated approach to bearing fault diagnostics and prognostics. *Proceedings of the 2005 American Control Conference*, pages 2750–2755, 2005.

[11] A.G. Rehorn, J. Jiang, and P.E. Orban. State-of-the-art methods and results in tool condition monitoring: a review. *The International Journal of Advanced Manufacturing Technology*, 26(7):693–710, 2005.

[12] C. Bunks, D. McCarthy, and T. Al-Ani. Condition-based Maintenance of Machines Using Hidden Markov Models. *Mechanical Systems and Signal Processing*, 14(4):597–612, July 2000.

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

[14] P. Smyth. Hidden Markov models for fault detection in dynamic systems. *Pattern Recognition*, 27(1):149–164, January 1994.

[15] J. Allanach, H. Tu, and S. Singh. Detecting, tracking, and counteracting terrorist networks via hidden Markov models. *2004 IEEE Aerospace Conference Proceedings*, 5:3246–3257, 2004.

[16] T.W. Sloan and J.G. Shanthikumar. Combined Production and Maintenance Scheduling for a Multiple-Product, Single- Machine Production System. *Production and Operations Management*, 9(4):379–399, January 2000.

[17] M. Celen and D. Djurdjanovic. Operation-dependent maintenance scheduling in flexible manufacturing systems. *CIRP Journal of Manufacturing Science and Technology*, 5(4):296–308, January 2012.

[18] S. Lee and J. Ni. Joint decision making for maintenance and production scheduling of production systems. *The International Journal of Advanced Manufacturing Technology*, 66(5-8):1135–1146, July 2012.

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

[20] M. Gales and S. Young. *The Application of Hidden Markov Models in Speech Recognition.* Now Publishers Inc, 2008.

[21] H.K. Lee and J.H. Kim. An HMM-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):961–973, 2002.

[22] G. D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[23] B. Keller and R. Lutz. Improved learning for hidden Markov models using penalized training. In *Artificial Intelligence and Cognitive Science*, pages 53–60. Springer, Berlin Heidelberg, 2002.

[24] M. Slimane, G. Venturini, J. Asselin de Beauville, T. Brouard, and A. Brandeau. Optimizing hidden Markov models with a genetic algorithm. In *Artificial Evolution*, volume 1063 of *Lecture Notes in Computer Science*, pages 384–396. Springer, 1996.

[25] F. Yang, C. Zhang, and G. Bai. A Novel Genetic Algorithm Based on Tabu Search for HMM Optimization. In *2008 Fourth International Conference on Natural Computation*, pages 57–61. IEEE, 2008.

[26] J.S. Lee and C.H. Park. Training hidden Markov models by hybrid simulated annealing for visual speech recognition. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 198–202. IEEE, October 2006.

[27] L. R. Rabiner, B. H. Juang, S. E. Levinson, and M. M. Sondhi. Some properties of continuous hidden Markov model representations. *AT&T Technical Journal*, 64(6):1251–1270, 1985.

[28] S. Kwong, C. W. Chau, K. F. Man, and K. S. Tang. Optimisation of HMM topology and its model parameters by genetic algorithms. *Pattern Recognition*, 34(2):509–522, February 2001.

[29] T.Y. Chen, X.D. Mei, J.S. Pan, and S.H. Sun. Optimization of HMM by the tabu search algorithm. *Journal of Information Science and Engineering*, 20(5):949–957, 2004.

[30] N. Thatphithakkul, K. Supphanat, and S. Kanokphara. HMM parameter optimization using tabu search. In *International Symposium on Communication and Information Technology*, volume 2004, pages 904–908. IEEE, 2004.

[31] D. Paul. Training of HMM recognizers by simulated annealing. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 10, pages 13–16, 1985.

[32] S. Huda, J. Yearwood, and R. Togneri. A stochastic version of Expectation Maximization algorithm for better estimation of Hidden Markov Model. *Pattern Recognition Letters*, 30(14):1301–1309, October 2009.

[33] S. Huda and J. Yearwood. A constraint-based evolutionary learning approach to the expectation maximization for optimal estimation of the hidden Markov model for speech signal modeling. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, 39(1):182–97, February 2009.

[34] M. Mitchell. *An introduction to genetic algorithms*. MIT Press, 1998.

[35] M. Srinivas and L.M. Patnaik. Genetic algorithms: a survey. *Computer*, 27(6):17–26, June 1994.

[36] JJ Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 16(1):122–128, 1986.

[37] I. Rojas, J. Gonzalez, H. Pomares, J.J. Merelo, P.A. Castillo, and G. Romero. Statistical analysis of the main parameters involved in the design of a genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 32(1):31–37, February 2002.

[38] A.J. Brown, V.M. Catterson, M. Fox, D. Long, and S.D.J. McArthur. Learning models of plant behavior for anomaly detection and condition monitoring. In *International Conference on Intelligent Systems Applications to Power Systems 2007*, pages 1–6, November 2007.

[39] A.V. Nefian and M.H. Hayes III. Hidden Markov models for face recognition. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 2721–2724, 1998.

[40] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, 2nd edition, 2001.

[41] A. Bleakie and D. Djurdjanovic. Dynamic Feature Monitoring Technique Applied to Thin Film Deposition Processes in an Industrial PECVD Tool. In *Proceedings of the ASME 2011 International Manufacturing Science and Engineering Conference*, pages 215–224, Corvallis, Oregon, 2011.

[42] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin Heidelberg, 1995.

[43] J.H. Chu and I. Lin. Direct observation of Coulomb crystals and liquids in strongly coupled RF dusty plasmas. *Physical review letters*, 72(25):4009–4012, 1994.

Appendix A: Emission Matrices for Synthetic Problem

For $M = 5$:

$$B^{(1)} = \begin{bmatrix} 0.832 & 0.068 & 0.027 & 0.072 & 0.001 \\ 0.263 & 0.340 & 0.128 & 0.214 & 0.056 \\ 0.116 & 0.032 & 0.234 & 0.228 & 0.389 \\ 0.081 & 0.154 & 0.241 & 0.109 & 0.415 \end{bmatrix} \qquad B^{(2)} = \begin{bmatrix} 0.343 & 0.061 & 0.064 & 0.489 & 0.043 \\ 0.040 & 0.003 & 0.157 & 0.785 & 0.014 \\ 0.015 & 0.008 & 0.421 & 0.052 & 0.504 \\ 0.008 & 0.233 & 0.394 & 0.265 & 0.100 \end{bmatrix}$$

For $M = 10$:

$$B^{(1)} = \begin{bmatrix} 0.584 & 0.046 & 0.021 & 0.043 & 0.002 & 0.045 & 0.082 & 0.022 & 0.149 & 0.006 \\ 0.064 & 0.125 & 0.037 & 0.101 & 0.013 & 0.513 & 0.081 & 0.035 & 0.019 & 0.011 \\ 0.071 & 0.011 & 0.085 & 0.033 & 0.118 & 0.272 & 0.053 & 0.250 & 0.033 & 0.073 \\ 0.030 & 0.075 & 0.112 & 0.074 & 0.233 & 0.039 & 0.032 & 0.223 & 0.014 & 0.168 \end{bmatrix}$$

$$B^{(2)} = \begin{bmatrix} 0.150 & 0.049 & 0.020 & 0.262 & 0.033 & 0.099 & 0.030 & 0.003 & 0.119 & 0.234 \\ 0.025 & 0.010 & 0.088 & 0.405 & 0.015 & 0.002 & 0.050 & 0.111 & 0.005 & 0.288 \\ 0.009 & 0.000 & 0.276 & 0.043 & 0.341 & 0.012 & 0.008 & 0.254 & 0.006 & 0.051 \\ 0.016 & 0.140 & 0.219 & 0.152 & 0.063 & 0.014 & 0.014 & 0.169 & 0.021 & 0.191 \end{bmatrix}$$

For $M = 10$:

$$B^{(1)} = \begin{bmatrix} 0.284 & 0.014 & 0.000 & 0.015 & 0.000 & 0.020 & 0.041 & 0.000 & 0.201 & 0.002 \\ 0.008 & 0.026 & 0.000 & 0.014 & 0.000 & 0.153 & 0.018 & 0.011 & 0.164 & 0.002 \\ 0.021 & 0.000 & 0.051 & 0.015 & 0.061 & 0.169 & 0.001 & 0.149 & 0.140 & 0.016 \\ 0.008 & 0.033 & 0.063 & 0.029 & 0.142 & 0.015 & 0.004 & 0.138 & 0.088 & 0.096 \\ & & & & & 0.005 & 0.006 & 0.000 & 0.094 & 0.319 \\ & & & & & 0.013 & 0.079 & 0.000 & 0.227 & 0.284 \\ & & & & & 0.050 & 0.001 & 0.039 & 0.000 & 0.287 \\ & & & & & 0.069 & 0.029 & 0.013 & 0.068 & 0.206 \end{bmatrix}$$

$$B^{(2)} = \begin{bmatrix} 0.047 & 0.008 & 0.000 & 0.082 & 0.000 & 0.099 & 0.011 & 0.000 & 0.154 & 0.083 \\ 0.016 & 0.000 & 0.060 & 0.315 & 0.000 & 0.007 & 0.029 & 0.089 & 0.074 & 0.230 \\ 0.000 & 0.000 & 0.252 & 0.030 & 0.311 & 0.016 & 0.000 & 0.231 & 0.017 & 0.037 \\ 0.000 & 0.064 & 0.117 & 0.081 & 0.032 & 0.079 & 0.005 & 0.090 & 0.041 & 0.097 \\ & & & & & 0.024 & 0.280 & 0.016 & 0.162 & 0.035 \\ & & & & & 0.001 & 0.012 & 0.052 & 0.000 & 0.116 \\ & & & & & 0.022 & 0.000 & 0.011 & 0.000 & 0.073 \\ & & & & & 0.005 & 0.270 & 0.000 & 0.021 & 0.096 \end{bmatrix}$$

For $M = 20$:

$$B^{(1)} = \begin{bmatrix} 0.227 & 0.011 & 0.000 & 0.012 & 0.000 & 0.016 & 0.033 & 0.000 & 0.056 & 0.002 \\ 0.011 & 0.034 & 0.000 & 0.018 & 0.000 & 0.197 & 0.024 & 0.014 & 0.000 & 0.003 \\ 0.018 & 0.000 & 0.043 & 0.013 & 0.052 & 0.144 & 0.001 & 0.127 & 0.000 & 0.014 \\ 0.005 & 0.020 & 0.039 & 0.018 & 0.088 & 0.009 & 0.003 & 0.086 & 0.000 & 0.060 \end{bmatrix}$$

$$\begin{bmatrix} 0.004 & 0.005 & 0.030 & 0.075 & 0.229 & 0.254 & 0.021 & 0.004 & 0.000 & 0.023 \\ 0.017 & 0.102 & 0.018 & 0.293 & 0.149 & 0.073 & 0.027 & 0.020 & 0.000 & 0.000 \\ 0.042 & 0.001 & 0.192 & 0.000 & 0.000 & 0.000 & 0.000 & 0.184 & 0.030 & 0.137 \\ 0.043 & 0.018 & 0.184 & 0.042 & 0.011 & 0.013 & 0.043 & 0.083 & 0.062 & 0.174 \end{bmatrix}$$

$$B^{(2)} = \begin{bmatrix} 0.041 & 0.007 & 0.000 & 0.072 & 0.000 & 0.020 & 0.009 & 0.000 & 0.150 & 0.073 \\ 0.015 & 0.000 & 0.054 & 0.279 & 0.000 & 0.000 & 0.025 & 0.079 & 0.046 & 0.204 \\ 0.000 & 0.000 & 0.217 & 0.026 & 0.268 & 0.003 & 0.000 & 0.199 & 0.078 & 0.032 \\ 0.000 & 0.049 & 0.091 & 0.063 & 0.025 & 0.000 & 0.004 & 0.070 & 0.044 & 0.075 \end{bmatrix}$$

$$\begin{bmatrix} 0.021 & 0.246 & 0.014 & 0.142 & 0.040 & 0.045 & 0.047 & 0.014 & 0.000 & 0.058 \\ 0.000 & 0.011 & 0.046 & 0.000 & 0.048 & 0.034 & 0.000 & 0.000 & 0.046 & 0.114 \\ 0.019 & 0.000 & 0.010 & 0.000 & 0.030 & 0.002 & 0.000 & 0.039 & 0.020 & 0.057 \\ 0.004 & 0.209 & 0.000 & 0.016 & 0.028 & 0.005 & 0.270 & 0.008 & 0.000 & 0.038 \end{bmatrix}$$

For $M = 25$:

$$B^{(1)} = \begin{bmatrix} 0.197 & 0.012 & 0.009 & 0.013 & 0.008 & 0.022 & 0.036 & 0.004 & 0.048 & 0.008 \\ 0.019 & 0.020 & 0.006 & 0.013 & 0.008 & 0.105 & 0.017 & 0.010 & 0.006 & 0.002 \\ 0.010 & 0.002 & 0.018 & 0.015 & 0.026 & 0.054 & 0.013 & 0.056 & 0.006 & 0.018 \\ 0.008 & 0.018 & 0.035 & 0.014 & 0.064 & 0.008 & 0.008 & 0.061 & 0.002 & 0.042 \end{bmatrix}$$

$$\begin{bmatrix} 0.007 & 0.005 & 0.002 & 0.065 & 0.205 & 0.221 & 0.026 & 0.006 & 0.008 & 0.027 \\ 0.018 & 0.062 & 0.014 & 0.156 & 0.084 & 0.040 & 0.026 & 0.014 & 0.014 & 0.002 \\ 0.026 & 0.013 & 0.019 & 0.006 & 0.001 & 0.010 & 0.002 & 0.077 & 0.017 & 0.058 \\ 0.038 & 0.015 & 0.007 & 0.037 & 0.008 & 0.016 & 0.039 & 0.065 & 0.044 & 0.119 \end{bmatrix}$$

$$\begin{bmatrix} 0.008 & 0.011 & 0.018 & 0.003 & 0.031 \\ 0.112 & 0.168 & 0.010 & 0.060 & 0.014 \\ 0.365 & 0.028 & 0.049 & 0.102 & 0.010 \\ 0.018 & 0.012 & 0.118 & 0.014 & 0.189 \end{bmatrix}$$

$$B^{(2)} = \begin{bmatrix} 0.042 & 0.012 & 0.007 & 0.059 & 0.001 & 0.024 & 0.015 & 0.001 & 0.031 & 0.063 \\ 0.026 & 0.001 & 0.063 & 0.280 & 0.001 & 0.011 & 0.027 & 0.078 & 0.005 & 0.207 \\ 0.006 & 0.008 & 0.227 & 0.032 & 0.285 & 0.011 & 0.007 & 0.210 & 0.006 & 0.034 \\ 0.005 & 0.053 & 0.093 & 0.063 & 0.026 & 0.001 & 0.013 & 0.073 & 0.001 & 0.078 \end{bmatrix}$$

$$\begin{bmatrix} 0.022 & 0.202 & 0.013 & 0.121 & 0.007 & 0.014 & 0.040 & 0.018 & 0.002 & 0.006 \\ 0.007 & 0.021 & 0.053 & 0.004 & 0.003 & 0.004 & 0.004 & 0.002 & 0.054 & 0.003 \\ 0.028 & 0.000 & 0.011 & 0.006 & 0.003 & 0.005 & 0.006 & 0.043 & 0.023 & 0.008 \\ 0.012 & 0.200 & 0.005 & 0.020 & 0.004 & 0.004 & 0.255 & 0.015 & 0.005 & 0.000 \end{bmatrix}$$

$$\begin{bmatrix} 0.008 & 0.020 & 0.002 & 0.260 & 0.008 \\ 0.011 & 0.004 & 0.002 & 0.110 & 0.019 \\ 0.002 & 0.004 & 0.015 & 0.012 & 0.008 \\ 0.002 & 0.007 & 0.009 & 0.049 & 0.004 \end{bmatrix}$$

For $M = 30$:

$$B^{(1)} = \begin{bmatrix} 0.195 & 0.009 & 0.007 & 0.013 & 0.006 & 0.021 & 0.032 & 0.003 & 0.049 & 0.002 \\ 0.012 & 0.017 & 0.001 & 0.010 & 0.008 & 0.069 & 0.011 & 0.009 & 0.005 & 0.004 \\ 0.006 & 0.000 & 0.015 & 0.004 & 0.013 & 0.038 & 0.001 & 0.035 & 0.007 & 0.006 \\ 0.008 & 0.013 & 0.026 & 0.011 & 0.052 & 0.007 & 0.005 & 0.053 & 0.001 & 0.044 \end{bmatrix}$$

$$\begin{bmatrix} 0.011 & 0.006 & 0.008 & 0.069 & 0.197 & 0.218 & 0.019 & 0.005 & 0.001 & 0.022 \\ 0.009 & 0.041 & 0.000 & 0.095 & 0.052 & 0.024 & 0.014 & 0.013 & 0.001 & 0.003 \\ 0.017 & 0.007 & 0.015 & 0.004 & 0.006 & 0.000 & 0.007 & 0.053 & 0.011 & 0.040 \\ 0.031 & 0.014 & 0.008 & 0.028 & 0.015 & 0.015 & 0.030 & 0.053 & 0.039 & 0.110 \end{bmatrix}$$

$$\begin{bmatrix} 0.010 & 0.006 & 0.017 & 0.009 & 0.028 & 0.005 & 0.009 & 0.007 & 0.015 & 0.001 \\ 0.069 & 0.105 & 0.004 & 0.038 & 0.004 & 0.011 & 0.086 & 0.003 & 0.255 & 0.029 \\ 0.244 & 0.021 & 0.031 & 0.073 & 0.006 & 0.224 & 0.081 & 0.017 & 0.004 & 0.015 \\ 0.013 & 0.006 & 0.104 & 0.013 & 0.163 & 0.051 & 0.007 & 0.053 & 0.017 & 0.010 \end{bmatrix}$$

$$B^{(2)} = \begin{bmatrix} 0.037 & 0.010 & 0.004 & 0.053 & 0.005 & 0.016 & 0.010 & 0.004 & 0.024 & 0.061 \\ 0.017 & 0.001 & 0.040 & 0.200 & 0.000 & 0.005 & 0.019 & 0.060 & 0.002 & 0.152 \\ 0.008 & 0.005 & 0.218 & 0.028 & 0.268 & 0.003 & 0.002 & 0.193 & 0.005 & 0.033 \\ 0.008 & 0.044 & 0.077 & 0.056 & 0.029 & 0.007 & 0.006 & 0.059 & 0.006 & 0.066 \end{bmatrix}$$

$$\begin{bmatrix} 0.021 & 0.180 & 0.014 & 0.109 & 0.008 & 0.007 & 0.040 & 0.018 & 0.008 & 0.006 \\ 0.002 & 0.010 & 0.034 & 0.001 & 0.000 & 0.004 & 0.003 & 0.000 & 0.037 & 0.006 \\ 0.027 & 0.001 & 0.011 & 0.007 & 0.008 & 0.002 & 0.005 & 0.041 & 0.024 & 0.000 \\ 0.009 & 0.175 & 0.002 & 0.017 & 0.003 & 0.006 & 0.228 & 0.012 & 0.001 & 0.005 \end{bmatrix}$$

$$\begin{bmatrix} 0.000 & 0.011 & 0.000 & 0.232 & 0.004 & 0.003 & 0.042 & 0.006 & 0.011 & 0.057 \\ 0.002 & 0.009 & 0.008 & 0.080 & 0.012 & 0.007 & 0.060 & 0.021 & 0.019 & 0.188 \\ 0.003 & 0.005 & 0.015 & 0.013 & 0.009 & 0.009 & 0.005 & 0.038 & 0.007 & 0.006 \\ 0.006 & 0.008 & 0.007 & 0.044 & 0.007 & 0.004 & 0.008 & 0.010 & 0.006 & 0.083 \end{bmatrix}$$