# Multiway Continuous Hidden Markov Model-Based Approach for Fault Detection and Diagnosis

**Deepthi Sen**
Dept. of Chemical Engineering, Indian Institute of Technology Madras, Chennai, Tamil Nadu 600036, India

**Dilshad Raihan A.V.**
Dept. of Mechanical Engineering, Indian Institute of Technology Madras, Chennai, Tamil Nadu 600036, India

**M. Chidambaram**
Dept. of Chemical Engineering, Indian Institute of Technology Madras, Chennai, Tamil Nadu 600036, India

*DOI 10.1002/aic.14386*
*Published online February 15, 2014 in Wiley Online Library (wileyonlinelibrary.com)*

*A fault detection and classification scheme that uses probabilistic inference based on multiway continuous hidden Markov models (MCHMM) which is capable of capturing complex system dynamics and uncertainty is proposed. A set of observations from normal and faulty runs of the system was collected and used to generate the training dataset. The training data is assumed to follow a finite Gaussian mixture model. The number of mixture components and associated parameters for the optimal Gaussian mixture fit of the observed data was computed subsequently by clustering using the Figueiredo–Jain algorithm for unsupervised learning. The segmental k-means algorithm was used to compute the HMM parameters. The applicability of the proposed scheme is investigated for the case of an inverted pendulum system and a fluidized catalytic cracker. The monitoring results for the above cases with the proposed scheme was found to be superior to the multiway discrete hidden Markov model (MDHMM) based scheme in terms of the accuracy of fault detection, especially in case of noisy observations.* © 2014 American Institute of Chemical Engineers *AIChE J*, 60: 2035–2047, 2014
*Keywords: fault detection and classification, hidden Markov models, finite Gaussian mixture models, Figueiredo–Jain algorithm, segmental k-means algorithm, fluidized catalytic cracker, inverted pendulum*

## Introduction

Detection and identification of abnormal modes of operation is central to ensuring safety and productivity in industrial systems and processes. Accurate and well-timed diagnosis of faults enables timely mitigative intervention and suitable corrective measures.[1] Fault detection and diagnosis (FDD) schemes are mainly classified as either model-based methods or data driven schemes. In model-based schemes, a model describing the dynamics of the process or system is used as basis for FDD.[2,3] Building a fitting model that can accurately describe the dynamics of a complex industrial system is often a difficult and time consuming process. Data driven methods conversely make use of the data obtained from the prior operation of the system.[4,5]

A description of various FDD schemes and a comparison between statistical methods and model-based methods was given by Venkatasubramanian et al.[6,7] Commonly used statistical methods include principal component analysis (PCA), partial least squares (PLS), Fisher discriminant analysis, and so forth.[8–13] The methods PCA and PLS assume that the operating data follows a unimodal Gaussian distribution. This condition may not often be valid in real systems and processes. A particular approach to enable PCA to handle multimodal operating data required enforcing constraints on the covariance structures.[14] Another approach to deal with multimodality of data is a mixture PCA method.[15] This work proposed the integration of heuristic smoothing clustering techniques with PCA. Real time FDD schemes that use dynamic PCA and dynamic PLS have been proposed, but these require a linear dynamic model structure.

A Gaussian mixture model (GMM) is capable of capturing the multimodality of the operating data. Choi et al. proposed categorizing the captured data into one of the Gaussian clusters which can then be subjected to an associated PCA scheme for detecting abnormal operation.[16] The Gaussian mixture parameters are computed using an expectation-maximization (EM) algorithm. The optimal number of clusters is chosen based on the targeted accuracy and the size of training dataset. An enhanced GMM-based FDD scheme was proposed by defining density of mixture modeling index.[17] The number of Gaussian clusters used is computed using a Bayesian information criterion. Yu and Qin proposed a process monitoring scheme that uses finite GMM (FGMM) and a Bayesian inference strategy.[18] The FDD scheme is trained using the Figueiredo–Jain (FJ) algorithm. The FJ algorithm is capable of determining the optimal number of Gaussian components required from the training data as opposed to EM algorithms which require the user to specify the number of clusters.[19,20]

Another class of FDD schemes are machine learning techniques such as support vector machines, neural networks,

Correspondence concerning this article should be addressed to M. Chidambaram at chidam@iitm.ac.in.

and hidden Markov models (HMM) that are capable of detecting and identifying abnormalities in processes and systems.[7,21] HMM have been used in speech recognition and hand-written character recognition problems. HMM-based methods have been proposed to detect anomalies for network security and to monitor tool wear in machinery.[22,23] A multiway discrete HMM (MDHMM)-based FDD scheme was proposed by Yu.[24] The model was trained using an iterative EM algorithm. The results obtained were demonstrated to be superior to those of MPCA and MDPCA.

In this article, we propose an HMM-based FDD scheme that relies on a GMM for the operating data. Hence, instead of assigning probabilities to discrete values of the process variables, the process data is clustered into multivariate Gaussian distributions. The optimum number of clusters and the respective weights, means, and covariance are computed using FJ algorithm. Each mode of operation is assigned a corresponding state in HMM against which the probability of occurrence of measured values are computed. Probabilities indicating temporal transition from one state to another will also be defined. The state sequence corresponding to a measurement sequence is then computed using a modified Viterbi algorithm that uses a conditional probability so that accuracy is not lost.

The remainder of this article is organized as follows. A basic discussion of HMMs along with the algorithms involved, namely, segmental k-means and Viterbi algorithms, is given under the section "A Brief Description of HMM." Descriptions of FGMMs and the FJ algorithm used in arriving at the optimal FGMM are provided under this section. The implementation of the proposed FDD scheme is explained under the section "Methodology." The proposed FDD scheme has been applied on an inverted pendulum system and a Model IV fluidized catalytic cracker (FCC), to study its effectiveness. Descriptions of these systems and the procedure followed in testing are given under the section "Application Examples." The results of FDD applied on the aforementioned systems and the analysis of the same are covered under the heading "Results and Discussion." The summary of the conducted study and key inferences are provided under "Conclusions."

## A Brief Description of HMM

A random process is called a Markov process if its state at any instant depends only on the previous state and not on any other prior state. A system that exhibits this property may be modeled as a Markov model. An HMM describes a system whose temporal state sequence is governed by the Markov property but is not directly observable to an external viewer. Instead, a set of measurable observations that depend on the state at each instant may be recorded. Let the system in question exist in $N$ distinct invisible states $x_1, x_2, \ldots, x_N$ where $x_i \in X$. These states make up the state sequence $S_1, S_2, \ldots, S_T$ such that a sequence of observation vectors $\boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_T}$ are emitted where $\boldsymbol{y_t} \in Y$ for $1 < t < T$. A canonical description of an HMM may be written as[25]

$$P(S_{t+1}=x_j|S_t=x_i, S_{t-1}=x_k, \ldots)=P(S_{t+1}=x_j|S_t=x_i)$$

$$P(\boldsymbol{y_{t+1}}=\boldsymbol{y}|S_t, S_{t-1}, \ldots, S_1, \boldsymbol{y_t}, \boldsymbol{y_{t-1}}, \ldots \boldsymbol{y_1})=P(\boldsymbol{y_{t+1}}=\boldsymbol{y}|S_t)$$

The complete description of an HMM requires the following parameters to be defined.

1. Number of states in the HMM (N)
2. Dimension of the observation vector $\boldsymbol{y_t}$ (D)
3. State transition probability distribution, $A=[a_{i,j}]_{N\times N}$

$$a_{i,j}=P(S_{t+1}=x_j|S_t=x_i)(1 \le i \le N, 1 \le j \le N)$$

4. Probability distribution in the states of measurements

$$b_i(y)=P(\boldsymbol{y_t}=\boldsymbol{y}|S_t=x_i)(1 \le i \le N)$$

5. Probability distribution of the initial state, ($\pi$)

$$\pi(x_i)=P(S_1=x_i)(1 \le i \le N)$$

In this work, each hidden state is representative of a mode of operation that we wish to detect. In addition to this, the measurement data emitted by each state is assumed to follow a multivariate Gaussian mixture distribution. Mathematically

$$b_i(\boldsymbol{y_t})=\sum_{k=1}^{M} \omega_{i,k}\mathcal{N}(\boldsymbol{y_t}, \mu_{i,k}, \Sigma_{i,k}) \quad (1)$$

where $\mu_{i,k}$ and $\Sigma_{i,k}$ are, respectively, the mean and covariance of the $k$th component out of $M$ components in the Gaussian mixture for the $i$th state.

To arrive at the concerned HMM from a large set of historical observations, we find the maximum likelihood (ML) estimates of the three probabilities mentioned above. This procedure, referred to as "training," involves computing $A, b(\boldsymbol{y})$, and $\pi$ that maximize some conditional probability of the state and measurement sequences. The most common methods adopted to train the HMM are the Baum–Welch algorithm and the segmental k-means estimation algorithm.

### *Segmental k-means estimation algorithm*

To compute the ML estimate for the HMM, the following probability density function of the observation sequence $\boldsymbol{Y}=\{\boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_T}\}$ is to be defined

$$P(\boldsymbol{Y}|A, b, \pi)=\sum_{S_{t-1}, S_t=x \forall x \in X} \pi_{S_0} \prod_{t=1}^{T} a_{S_{t-1}, S_t} b_{S_t}(\boldsymbol{y_t})$$

The set of parameters $A, b, \pi$ that maximizes the probability $P(\boldsymbol{Y}|A, b, \pi)$ over all such possible parameter sets is the ML estimate of the HMM. The Baum–Welch algorithm may be used to meet this objective. Although in theory, Baum–Welch assures convergence, it is known that the algorithm is slow and in certain cases, restarts may be required which further increases computation time.[26] A different optimization method for the HMM parameters is the segmental k-means algorithm. Unlike Baum–Welch algorithm, the segmental k-means algorithm does not compute the transition over all possible states. Only those transitions as predicted by the most likely path are considered in the optimization. This makes segmental k-means algorithm much faster and more robust compared to Baum–Welch even though it may be a little less accurate in many cases.[27]

The segmental k-means algorithm computes the HMM $\lambda=\lambda(A, b, \pi)$,where

$$\lambda(A, b, \pi)=\underset{\lambda}{\operatorname{argmax}} \{\max_{\boldsymbol{S}} f(\boldsymbol{S}, \boldsymbol{Y}|\lambda)\}$$

That is, given the sequence of states $\boldsymbol{S}$ based on the present estimate of HMM parameters which maximizes the probability of occurrence of the observation sequence, the segmental k-means algorithm re-estimates an HMM, $\lambda$, that

maximizes this state optimized likelihood. But for the purpose of training the HMM via segmental k-means, a continuous multivariate probability distribution function that accounts for the observed dataset corresponding to each state, has to be generated in order to compute the emission probability density function $b(\boldsymbol{y})$. As a unimodal Gaussian distribution may be insufficient to explain the observed data from complex dynamical systems, a finite mixture of Gaussian probability densities is used to model the same. The number of Gaussian components and a good estimate of the mean and covariance matrices for the components may be computed by clustering the training data. Re-estimation of HMM parameters with segmental k-means requires definitions of the following variables.

For $1 \le t \le T, 1 \le i \le N$

1. Forward Variable, $\alpha_t(i)$

$$\alpha_t(i)=P(\{\boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_t}\}, S_t=x_i|\lambda) \tag{2}$$

2. Backward Variable, $\beta_t(i)$

$$\beta_t(i)=P(\{\boldsymbol{y_{t+1}}, \boldsymbol{y_{t+2}}, \ldots, \boldsymbol{y_T}\}|S_t=x_i, \lambda) \tag{3}$$

3. An xi to xj transition variable, $\psi_t(i,j)$

$$\psi_t(i,j)=P(S_t=x_i, S_{t+1}=x_j|\boldsymbol{Y}, \lambda) \tag{4}$$

4. A variable representing the occurrence of $x_i$

$$\gamma_t(i)=P(S_t=x_i|\boldsymbol{Y}, \lambda) \tag{5}$$

Given a training dataset and the most likely sequence of hidden states, the observation sequence is labeled and assigned the corresponding state. Then, the transition variable may be updated as

$$\psi_t(i,j)=\begin{cases} 1 & \text{if } \boldsymbol{y_t} \text{ corresponds to } x_i \text{ and } \boldsymbol{y_{t+1}} \text{ to } x_j \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Once $\psi$ is computed, the variable $\gamma$ may be computed as follows

$$\gamma_t(i)=\sum_{j=1}^{N} \psi_t(i,j), \quad 1 < n < T-1 \tag{7a}$$

$$\gamma_T(i)=\sum_{j=1}^{N} \psi_t(j,i) \tag{7b}$$

Assume that $L$ is the total number of batches and $T_l$ is the length of the $l$th sequence. Define a probability that the observation $\boldsymbol{y_t}$ was generated by the $k$th component of the $i$th state

$$\gamma_t(i,k)=\gamma_t(i)\frac{\omega_{ik} b_{ik}(\boldsymbol{y_t})}{b_i(\boldsymbol{y_t})} \tag{7c}$$

Then, the HMM parameters may be estimated as

$$\pi_i=\frac{\sum_{l=1}^{L} \gamma_1^l(i)}{L} \tag{7d}$$

$$a_{i,j}=\frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l-1} \psi_t^l(i,j)}{\sum_{l=1}^{L}\sum_{t=1}^{T_l-1} \gamma_t^l(i)} \tag{7e}$$

$$\omega_{i,k}=\frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l} \gamma_t^l(i,k)}{\sum_{l=1}^{L}\sum_{t=1}^{T_l} \gamma_t^l(i)} \tag{7f}$$

$$\mu_{i,k}=\frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l} \gamma_t^l(i,k)\boldsymbol{y_t}}{\sum_{l=1}^{L}\sum_{t=1}^{T_l} \gamma_t^l(i,k)} \tag{7g}$$

$$\Sigma_{i,k}=\frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l} \gamma_t^l(i,k)(\boldsymbol{y_t}-\mu_{i,k})(\boldsymbol{y_t}-\mu_{i,k})^T}{\sum_{l=1}^{L}\sum_{t=1}^{T_l} \gamma_t^l(i,k)} \tag{7h}$$

Once all the aforementioned parameters are computed, the resulting HMM can be used to arrive at an updated most-likely state sequence estimate, using Viterbi algorithm. Based on the new state sequence, the HMM parameters can be re-estimated following the steps mentioned above. This process is continued till the model parameters are converged.

### *Finite Gaussian mixture models*

The parameters of the FGMM have to be arrived at iteratively using the EM algorithm which was originally developed for computing the ML estimates of model parameters for an incomplete observed dataset.[28] In addition to this, the algorithm may be used in a case where the ML function does not permit analytical optimization but the solution is considerably simplified on assuming the existence of "missing" parameters. A complete dataset $Z$ is assumed to exist such that $Z=(X,Y)$ where $X$ is a set of unknown parameters and $Y$ is the incomplete observation dataset. For a given $\mathbf{y} \in Y$, the EM algorithm computes the parameter $\Phi$ that maximizes the incomplete probability density function $p(\mathbf{y}|\theta)$ using another function $p(\mathbf{z}|\theta)$ where $z \in Z$.[29]

The algorithm consists of recursively carrying out the below steps:

*Expectation Step.* The expected value of the conditional log-likelihood of complete data is computed, given the incomplete observation dataset $Y$ and current parameter estimates

$$Q(\Theta, \Theta^{i-1})=E[\log p(X,Y|\Theta)|Y, \Theta^{i-1}]$$

On applying the E-step in the FGMM parameter estimation problem with an initial parameter estimate of $\Theta_0=(\omega_0, \mu_0, \Sigma_0)$, we derive the expression for the posterior probability $P(C_k|y_{j,i})$ of the $j$th observation vector in the $i$th state belonging in the $k$th Gaussian component in the $r$th iteration

$$P^{(r)}(C_{k,i}|y_{j,i})=\frac{\omega_{k,i}^{(r)} g(y_{j,i}|\mu_{k,i}^{(r)}, \Sigma_{k,i}^{(r)})}{\sum_{s=1}^{M} \omega_{s,i}^{(r)} g(y_{j,i}|\mu_{s,i}^{(r)}, \Sigma_{s,i}^{(r)})} \tag{8}$$

*Maximization Step.* A new parameter estimate is arrived at by maximizing the expectation computed in the first step

$$\Theta^i=\underset{\Theta}{\operatorname{argmax}}\ Q(\Theta, \Theta^{i-1})$$

The maximization step for FGMM parameter estimation is given below

$$\mu_{k,i}^{(s+1)}=\frac{\sum_{j=1}^{n_i} P^{(s)}(C_{k,i}|y_{j,i}) y_{j,i}}{\sum_{j=1}^{n_i} P^{(s)}(C_{k,i}|y_{j,i})} \tag{9a}$$

$$\Sigma_{k,i}^{(s+1)}=\frac{\sum_{j=1}^{n_i} P^{(s)}(C_{k,i}|y_{j,i})(y_{j,i}-\mu_{k,i}^{(s+1)})(y_{j,i}-\mu_{k,i}^{(s+1)})^T}{\sum_{j=1}^{n_i} P^{(s)}(C_{k,i}|y_{j,i})} \tag{9b}$$

$$\omega_{k,i}^{(s+1)} = \frac{\sum_{j=1}^{n_i} P^{(s)}(C_{k,i}|y_{j,i})}{n_i} \tag{9c}$$

Even though, in theory, the EM algorithm may be shown to iteratively converge to the ML estimate of the mixture model parameters, we come across a number of its limitations on putting it into practice[19]

1. Due to the presence of local maxima in the probability density distribution of FGMMs, the final ML estimate computed by the EM algorithm is strongly sensitive to the initial guess supplied.
2. In case of FGMMs with unbounded likelihood, the algorithm returns unreasonable values for ML estimates as it converges to the boundaries of the parameter space of the FGMM.
3. The adequacy of the FGMM in describing the observation dataset is dependent on the number of Gaussian components in the FGMM as too few components might fail to effectively capture system dynamics and too many might lead to overfitting.

These limitations are overcome by using the FJ algorithm which automatically optimizes the number of components in the FGMM and ensures the best fit FGMM for the given set of observations.

### *FJ algorithm*

This algorithm is initialized by considering an arbitrarily large number of Gaussian components to describe the model. The minimum message length criterion is used in specifying the objective function for optimization so that insignificant components are discarded iteratively. The objective function for $n$ samples of observation vectors, $\mathcal{Y}=\{\boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_n}\}$ is as given below[18,19]

$$\mathcal{L}(\mathcal{Y}, \Theta) = \frac{V}{2} \sum_{m:\omega_m>0} \log \frac{n\omega_m}{2} + \frac{k_{nz}}{2} \log \frac{n}{2} + \frac{k_{nz}(V+1)}{2} - \log P(\mathcal{Y}|\Theta)$$

Here, $V=\frac{1}{2}D^2+\frac{3}{2}D$ represents the number of independent scalar parameters required to specify a single $D$-variate Gaussian component and $k_{nz}$ stands for the number of significant Gaussian components with weights greater than zero.

### *Viterbi algorithm*

On having generated the parameters for the HMM describing a certain system, one has to consider the state estimation problem, where, given a sequence of observation vectors, an optimal estimate of the state sequence that gave rise to the observations has to be computed. Various methods that differ in the imposed optimality criterion are available for this purpose. One such way is by calculating the most probable state at each instant independent of previous instants. This maximizes the expected number of correctly identified states. But this procedure is not suitable for systems where one or more elements in the transition matrix are zero. Hence, we resort to another global optimality constraint where the single best state sequence is computed, given the entire observation sequence.[30] This is achieved using the dynamic programming-based Viterbi algorithm.

The Viterbi algorithm, originally developed as a means to decode convolutional codes, may be used to recursively track the optimal state sequence in a discrete state Markov process by solving the concerned maximum *a posteriori* probability estimation problem.[31] The implementation of this algorithm requires us to define a new variable $\delta_t(i)$ such that[24]

$$\delta_t(i) = \max_{S_1, S_2, \ldots, S_{t-1}} P(S_1, S_2, \ldots, S_t-1, S_t=x_i, \boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_t}|\lambda) \tag{10}$$

This may be expressed as an iterative expression as given below

$$\delta_{t+1}(j) = \max_i [\delta_t(i) a_{ij}] b_j(\boldsymbol{y_{t+1}}) \tag{11}$$

$\delta_t(i)$ is computed for $1 \le t \le N$ at each $t$ and the state $x_i$ that maximizes $\delta_t$ is stored. After the evaluation of $\delta_T$, backtracking offers the required optimal path.

## Methodology

A diagram describing the sequential procedure to be followed while implementing the MCHMM scheme is given in Figure 1.

### *Training*

A sufficiently large historical observation set is collected or generated such that it contains a sufficient number of runs (say $L$) of length $T$ consisting of $D$-dimensional observation vectors $\boldsymbol{y}$ generated by the normal and each of the faulty modes of operation that we set out to detect. In doing so, we obtain a three-dimensional matrix of dimension $(T \times D \times L)$, $\bar{\boldsymbol{Y}}=\{\boldsymbol{Y_1}, \boldsymbol{Y_2}, \ldots, \boldsymbol{Y_L}\}$, where $\boldsymbol{Y_l}, 1 \le l \le L$, stands for $\{\boldsymbol{y_{1,l}}, \boldsymbol{y_{2,l}}, \ldots, \boldsymbol{y_{T,l}}\}$. We assign a state $S_t \in X$ to the observation vector at each time instant $t$ in each run. Through this step, we incorporate our prior knowledge of fault characteristics into the HMM. The set of data vectors is segregated into $N$ groups where each group contains all observation vectors corresponding to a particular state. Here, we take care to ensure that each state contains an ample number of data vectors $n_i$ so as to form a viable FGMM. The FJ algorithm is executed on each group of data vectors until convergence is reached. The best fit model obtained from FJ algorithm $\{\bar{\omega}, \bar{\mu}, \bar{\Sigma}\}$ is fed into the formal re-estimation algorithm (segmental k-means) along with the original state sequence. As we have prior knowledge of the states in the training data, the most likely path required for segmental k-means estimation is held fixed at the original sequence and is not updated. The initial estimates of $\{\bar{\omega}, \bar{\mu}, \bar{\Sigma}\}$ are readjusted and the transition probability $A$ is computed iteratively to obtain the final trained HMM model $\lambda=\{A^\star, \omega^\star, \mu^\star, \Sigma^\star\}$.

### *Testing*

To use the trained HMM for the purpose of fault detection, the new test batch to be monitored, $\boldsymbol{Y}_{\text{test}}$, is generated. The optimal state sequence, given the HMM model $\lambda$, is generated on applying the Viterbi algorithm to $\boldsymbol{Y}_{\text{test}}$. It should be noted that $\delta_t(i)$ gets progressively smaller along the sequence from $t=1$ to $t=T$ and comparison of such small numbers may prove inaccurate due to computational constraints. To overcome this drawback, we normalize $\delta_t(i)$ for each $i$ by dividing this number by $\Delta_t=\sum_i \delta_t(i)$ before comparison.

## Application Examples

The applicability of the proposed scheme is investigated for the case of a controlled inverted pendulum system and a Model IV FCC reactor/regenerator system.

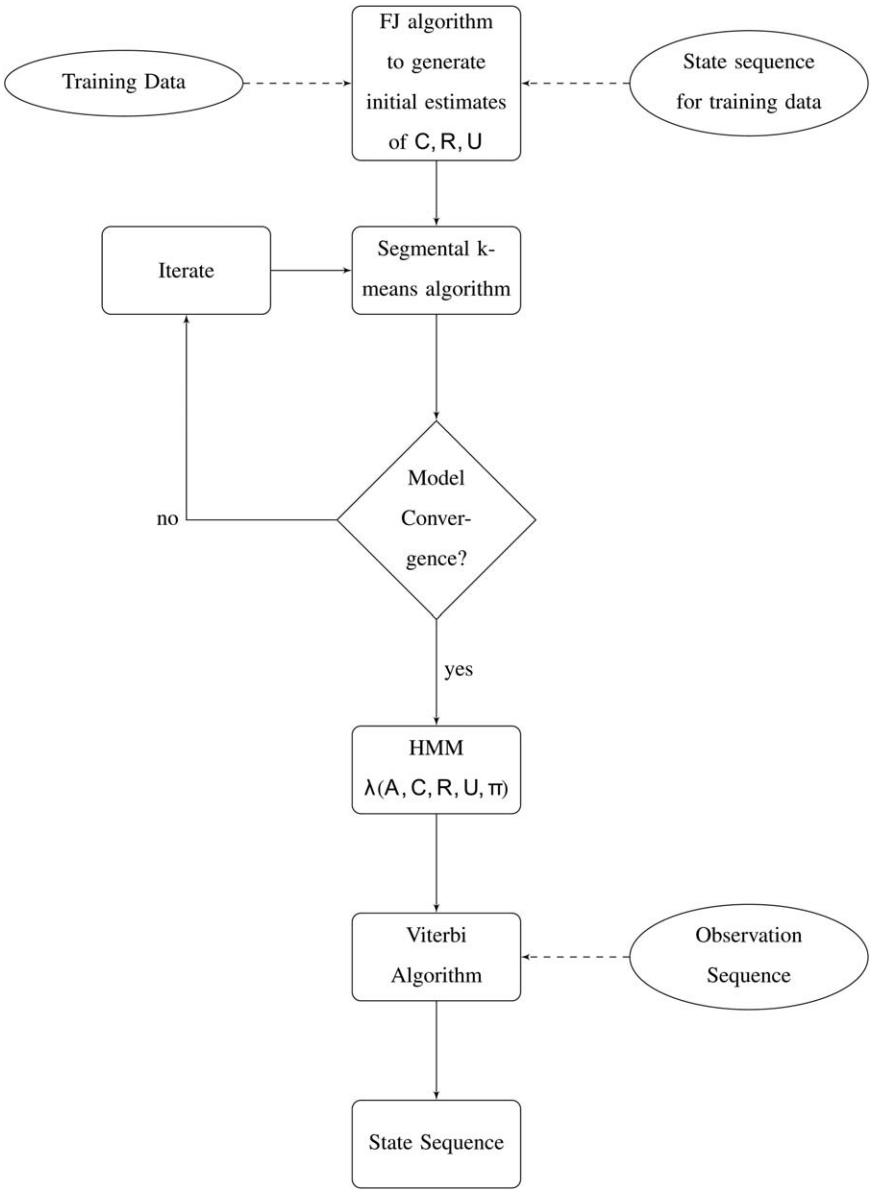


**Figure 1. Flow chart for implementing Fault diagnosis using MCHMM.**

### *Inverted pendulum*

An inverted pendulum, as the name indicates, is a regular pendulum kept in the inverted position, with its center of mass positioned above the point of support. The pendulum is stabilized in the vertical position by the application of a torque produced by the motion of a cart on which the pendulum system is mounted. A state feedback controller has been designed using pole placement method for the purpose of stabilizing the system. The state of the inverted pendulum system at any instant may be completely specified in terms the following variables.

1. Cart position $r$ as measured from the equilibrium position at $r = 0$
2. Cart velocity $\dot{r}$
3. Angular displacement of the pendulum $\theta$ about the vertical position
4. Angular velocity $\dot{\theta}$

The equations governing the dynamics of the pendulum can be found in Ref. 32.

To study the effectiveness of the proposed HMM scheme, the following additive faults are introduced in the inverted pendulum system:

*1. Frictional Force on the Actuator.* Frictional resistance to the motion of the cart may prevent the pendulum from equilibrating resulting in abnormal operation. In such a situation, the friction may be represented by the Stribeck model[33]

$$F_{\text{fric}} = g(r)\text{Sign}\,(r) \tag{12}$$

where

$$\text{Sign}\,(r) = \begin{cases} 1 & \text{if } r > 0 \\ [-1, 1] & r = 0 \\ -1 & r < 0 \end{cases} \tag{13}$$

$$g(r) = F_c + (F_c - F_s)\exp\left(-\left|\frac{r}{v_s}\right|\right)_s^{\delta} \tag{14}$$

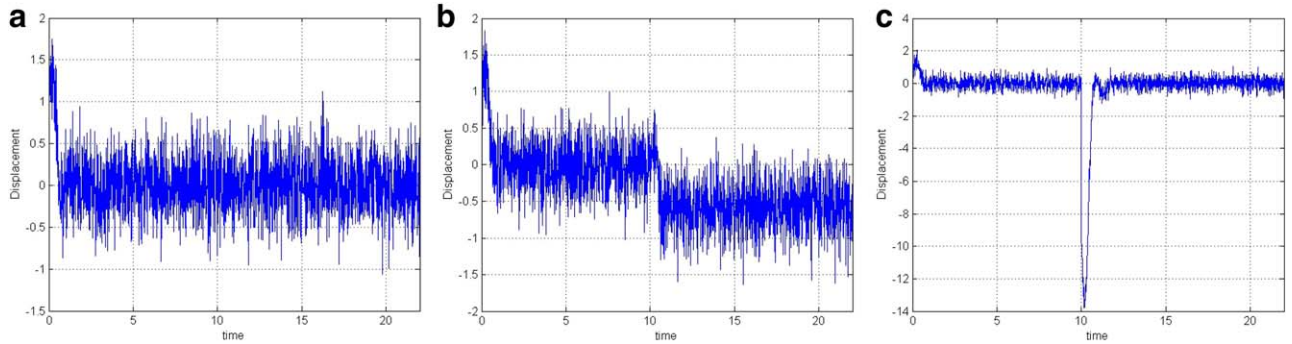


**Figure 2. Inverted pendulum—various modes of operation ($x_1$ vs. $t$), Noise power = 0.01.**

**(a) Normal operation. (b) Fault 1 at $t = 10$. (c) Fault 2 at $t = 10$. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]**

Here, $F_c$ and $F_s$ stand for Coulomb and static friction and $\delta_s$ and $v_s$ are the Stribeck velocity and shaping parameter.

*2. An Offset on the Displacement Sensor.* A constant offset is introduced in the displacement sensor.

As both these faults are additive, they do not alter the model parameters of the system, but only introduce additive terms in the state space equation. The linear displacement of the system is nondimensionalized and plotted against time for different modes of operation as given in Figure 2. It may be noted that the equilibrium point of the inverted pendulum system gets permanently shifted on introducing either of the two faults. But in the case of Fault 2, the constant offset in sensor is added to all measured values. Addition of this offset prevents the actual shift in equilibrium from being visible in the recorded measurements.

Consequently Fault 2 appears as only a temporary spike in measurements as seen in Figure 2c.

The normal operating point is taken as $X_{OP} = [1 \quad 0 \quad 0 \quad 0]^T$ about which the nonlinear system model is linearized. This point is described as

$$r=1, \dot{r}=0, \theta=0, \dot{\theta}=0$$

The system model, after nondimensionalization, is expressed as

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} n_1 r \\ n_2 \theta \\ n_3 \dot{r} \\ n_4 \dot{\theta} \end{bmatrix} \tag{15}$$

where $n_1, n_2, n_3, n_4$ are the nondimensionalizing constants. The linear state space model shall be represented as

$$\dot{x}=Ax+Bu \tag{16a}$$

$$y=Cx+v \tag{16b}$$

where ($A$, $B$, $C$) is the state-space realization of inverted pendulum system under normal operation and $v$ is the measurement noise. On introducing the faults, $u$ becomes $u'$ where

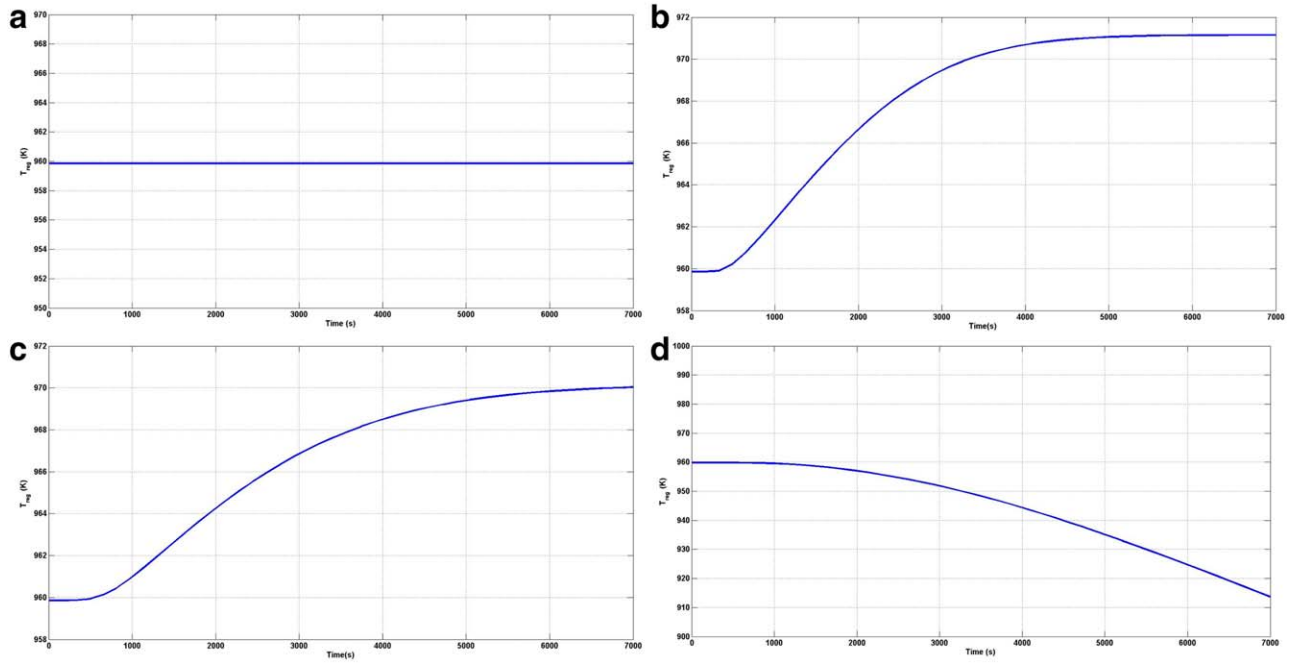


**Figure 3. FCC unit—various modes of operation (regenerator temperature vs. Time), noise-free case.**

**(a) Normal Operation. (b) Fault 1 at $t = 1000$. (c) Fault 2 at $t = 1000$. (d) Fault 3 at $t = 1000$. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]**

**Table 1. Inverted Pendulum Results**

| Case | Noise Power | FDD Scheme | $e_{train}$ % | $e_{fresh}$% |
|---|---|---|---|---|
| 1 | 0.0000001 | Proposed Method | 0.93 | 0.93 |
| | | MDHMM | 0.93 | 0.97 |
| 2 | 0.00001 | Proposed Method | 0.93 | 0.93 |
| | | MDHMM | 0.93 | 2.00 |
| 3 | 0.0001 | Proposed Method | 0.93 | 0.93 |
| | | MDHMM | 2.28 | 9.27 |
| 4 | 0.001 | Proposed Method | 0.93 | 1.01 |
| | | MDHMM | 25.77 | 27.07 |
| 5 | 0.005 | Proposed Method | 6.01 | 6.62 |
| | | MDHMM | 26.10 | 27.44 |
| 6 | 0.01 | Proposed Method | 8.13 | 11.83 |
| | | MDHMM | 26.34 | 31.17 |

$$u' = [\, u + f_A \quad f_{S,1} \quad 0 \quad 0 \quad 0 \,]^T$$

Here, $f_A$ represents the actuator fault and $f_{S,1}$ represents fault in the displacement sensor. Hence, $u'$ denotes the combined input to the system that comprises both the control signal and the faults under consideration. Substituting appropriate numerical values for the model parameters, the state space equations are rewritten as

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.1 & -2 & -0.08 \\ 0 & 21.47 & 26.31 & -0.13 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -6.13 & 0 & 0 & 0 & 0 \\ 84.30 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u+f_A \\ f_{S,1} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{17a}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u+f_A \\ f_{S,1} \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \tag{17b}$$

Here, the matrix $C$ is taken as the $4 \times 4$ identity matrix. In addition to this, the coefficient matrix of $u'$ in Eq. 17b is chosen such that its first column is a zero vector. This ensures that, in the absence of faults and noise, the value of output $y$ will be equal to the state vector $x$. The parameters $[\, v_1 \quad v_2 \quad v_3 \quad v_4 \,]^T$ are zero mean Gaussian white noises. A discussion on other parameters and their values can be found in Ref. 32.

### *Simulation*

The dynamics of the inverted pendulum is simulated in Simulink which uses a variable step size Runge–Kutta method (ODE45 Dormand Prince) to solve the state-space equations.

### *Model IV FCC Unit*

In this part of the article, we implement the proposed FDD scheme on the simulation of a controlled FCC maintained at a predefined set point. The FCC model, which simulates a Model IV FCC reactor/regenerator section for the purpose of control simulation, was developed combinedly by the Amoco Corporation and the Lehigh University.[35] It was originally implemented in advanced continuous simulation language and incorporates all major dynamics of such a system. A Simulink model for the same was later developed by Dr. Emad Ali of King Saud University. This model, which may be accessed online,[35] has been used in this study after introducing white measurement noise in each of the output variables to mimic actual data that is obtained from a plant.

The model consists of 20 input variables, 20 output variables, and 19 state variables. A complete description of all the variables and their normal operating values can be found in reference.[34] The faults considered in this work refer to abnormal changes in the model parameters that lead to marked deviations in process operation. A graphical representation of all four modes of operation is shown in Figure 3. Here, the regenerator temperature ($T_{\text{reg}}$) is plotted against time as the change in dynamics of the system due to the faults can be readily seen in the $T_{\text{reg}}$ against time plots.

*Description of Faults.*

1. Step change of 0.05 in coke formation factor which results in increased coke deposition in riser and spent catalyst $C_{\text{sc}}$.
2. A positive ramp of slope 0.0556 K/s in fresh feed preheat temperature for 300 s after which it stays constant.
3. A negative ramp of slope $-0.453 \times 10^{-3}\,\text{kg/s}^2$ in slurry recycle rate.

Just as in the case for inverted pendulum, a training dataset consisting of equal number of runs for each mode of operation is recorded by repeatedly running the simulation model. In this study, we have used 25 batches per mode of operation for this purpose.

## Results and Discussion

Based upon the training data $\bar{\boldsymbol{Y}}$ generated through the aforementioned models, the proposed algorithm was performed on both the training batches as well as a set of new batches to detect and classify instances of abnormal operating modes. The error in fault detection is measured in terms of $e\%$ which is defined as

$$e\% = \frac{s_{\text{error}}}{s_{\text{tot}}} \tag{17}$$

where $s_{\text{error}}$ is the number of instants at which the state has been wrongly estimated from all batches in the test data and $s_{\text{tot}}$ is the total number of instants in the test data. Furthermore, $e_{\text{training}}\%$ and $e_{\text{fresh}}\%$ refer to the error measurements when testing has been performed on the training batch and a fresh set of runs, respectively.

Another measure of performance of an FDD scheme is its false alarm rate. A false alarm or a false positive response is said to occur when the FDD scheme reports a fault in spite of the system being under normal operation. False positive rate is defined as the fraction of normal instants that are reported with a false positive response. A false negative response, conversely, is said to occur when the FDD scheme reports no faults in spite of the system being under an abnormal mode of operation. False negative rate refers to the fraction of instants of abnormal operation that are reported with a false negative response.

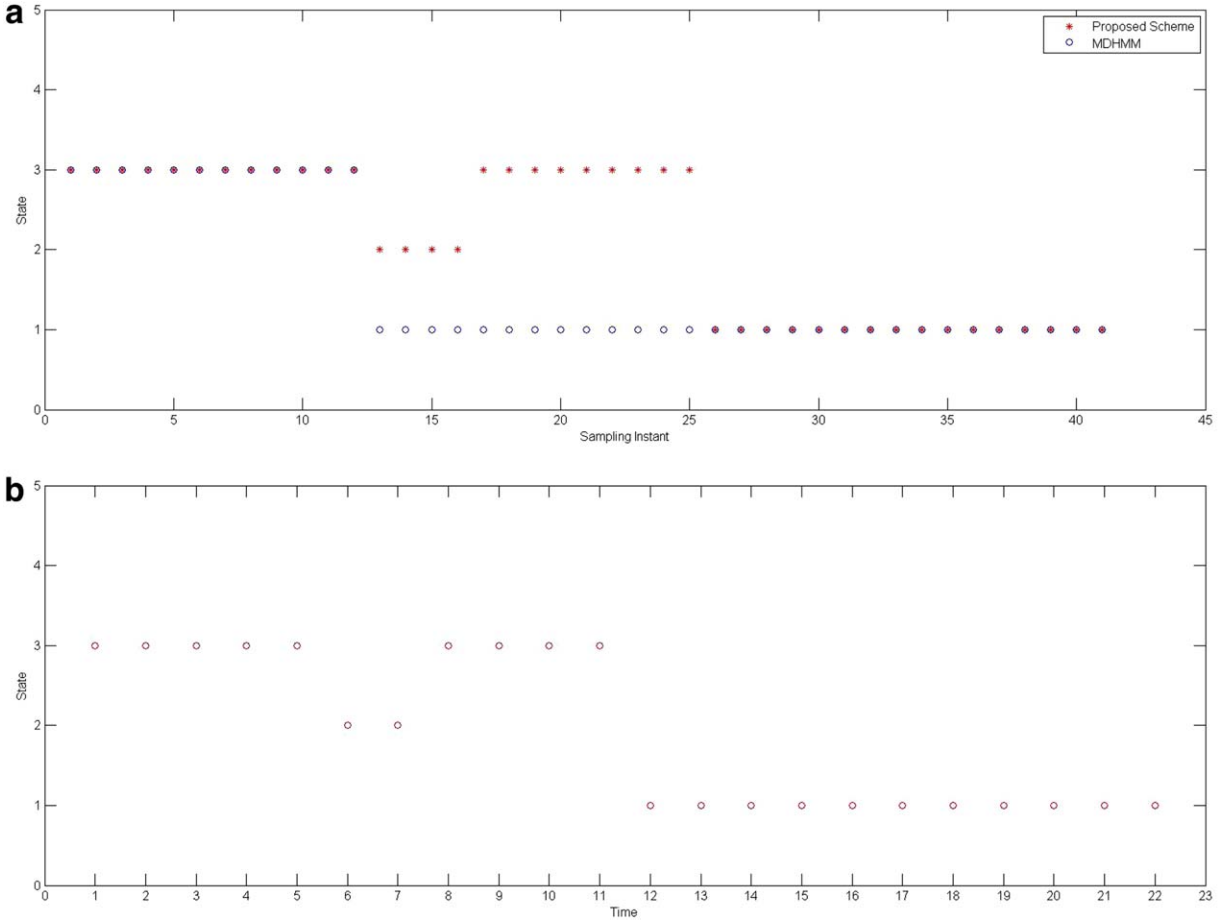


**Figure 4. Comparison of results between MDHMM, the proposed scheme and actual data for noise power = 0.01.**

**(a) Fault diagnosis results for MDHMM and the proposed scheme. (b) Actual modes of operation [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]**

The summary of the FDD results of the proposed scheme in comparison with those of the MDHMM-based fault detection and classification scheme[24] is discussed below:

### *Results—Inverted pendulum*

It may be seen from Table 1 that both MDHMM-based scheme and the proposed multiway continuous HMM (MCHMM)-based scheme perform accurate diagnosis for low noise training and test data with comparable error rates. On comparing the results predicted by the proposed FDD scheme with the actual states, it was found that the continuous observation model is capable of accurately arriving at the mode of operation of the system at each instant. The performance of the proposed scheme was studied in terms of the percentage error across different noise levels in the observation data. This was achieved by varying the noise power in the band limited white noise blocks from Simulink. The noise power indicates the amplitude of the power spectral density of the Gaussian white noise. A steady increase in fault detection error is seen in MDHMM with higher noise power whereas rise in error rate in the case of the proposed MCHMM based scheme is considerably less.

Under real conditions, an FDD scheme has to be applied to freshly available data which may be quite different from the training dataset. It is observed that at low noise powers, the proposed scheme gives accurate predictions, regardless of whether the test data is chosen from the training dataset or freshly generated from a new run of the model. At low noise powers, the MDHMM-based scheme is also capable of performing accurate fault diagnosis on fresh data. On increasing the noise power, both schemes exhibit a drop in the accuracy.

But it can be seen from Table 1 that this reduction in performance is much more pronounced in the MDHMM-based scheme than the MCHMM-based scheme. At a noise power of 0.005, the MDHMM-based scheme has an error rate of 27.44%, rendering it an unreliable means of state estimation under such conditions. The proposed new scheme, conversely, makes reasonably good estimates with an error rate of 6% at a noise power of 0.005 and an error rate of 11% at a noise power of 0.01. It may be noted that the performance of the FDD schemes are generally better on the training dataset than on a freshly generated data as can be seen in the Table 1. This happens because the probabilistic relationships existing between the states and the observations are captured in the HMM parameters from the training dataset. As the fresh data may not exhibit these same relationships, the accuracy of predictions of HMM-based FDD schemes drops. This drop in accuracy becomes prominent at higher noise levels when there may be a larger disparity between the two datasets.

In addition to this, 60 batches of fresh data generated at 0.01 noise power, with 20 batches belonging to each mode of operation, were subjected to FDD using the proposed scheme and the MDHMM-based scheme. The algorithms were trained initially on a dataset consisting of merely nine batches with three batches per each mode of operation

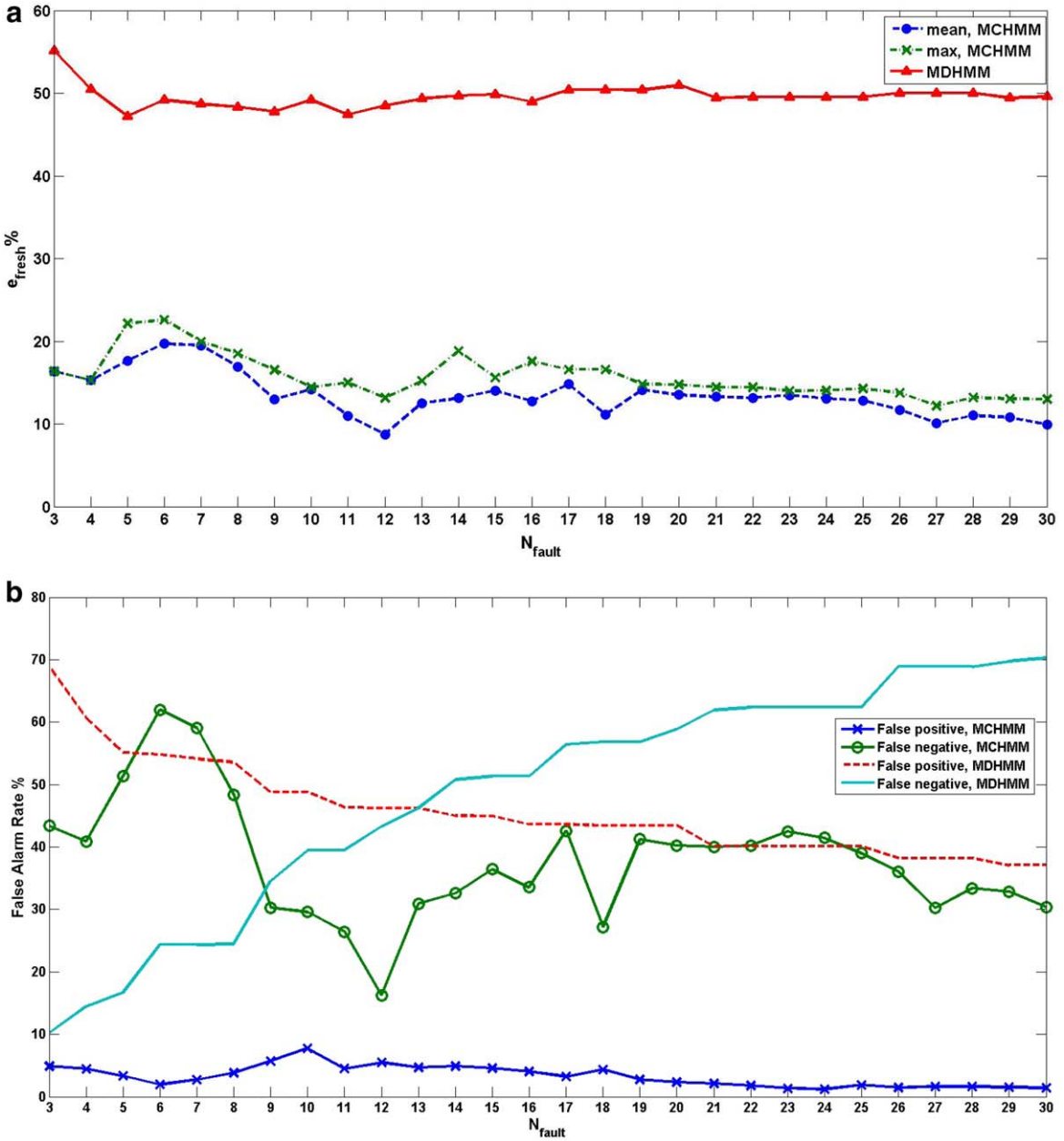


**Figure 5. Comparison of results between MDHMM and MCHMM for inverted pendulum FDD for noise power- = 0.01.**

**(a) Total error rates. (b) False alarm rates. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]**

($N_{\text{fault}}$). The size of training dataset was gradually increased to study its effect on the accuracy of detection, while keeping the constant noise power 0.01.

After training, the MDHMM-based FDD was performed on the fresh dataset and the error rate was noted.

Because the FJ algorithm is a stochastic clustering scheme, it might cluster the same dataset differently in different runs. Hence, the MCHMM-based FDD was trained and tested 40 times for each data size, recording the average and maximum error rates.

Figure 4a shows the states as identified by the two FDD schemes for a particular test batch from the inverted pendulum system. The actual state of the system at each instant for the same batch can be seen in Figure 4b. A plot showing the relationship between average $e_{\text{fresh}}\%$ and the number of training batches per mode of operation, $N_{\text{fault}}$, used in training is given in Figure 5a. It is seen that in the case of MCHMM, for lower values of $N_{\text{fault}}$, not only is the $e_{\text{fresh}}$ generally higher, but it also exhibits major variations from one value of $N_{\text{fault}}$ to the next. This is not the case with

**Table 2. Model IV FCC Results for Proposed Method**

| Case | Noise Power | $e_{train}$ % | $e_{fresh}$ % |
|---|---|---|---|
| 1 | 0.0000001 | 0.61 | 0.61 |
| 2 | 0.0001 | 0.61 | 0.61 |
| 3 | 0.01 | 0.64 | 0.64 |
| 4 | 5.0 | 0.64 | 0.64 |

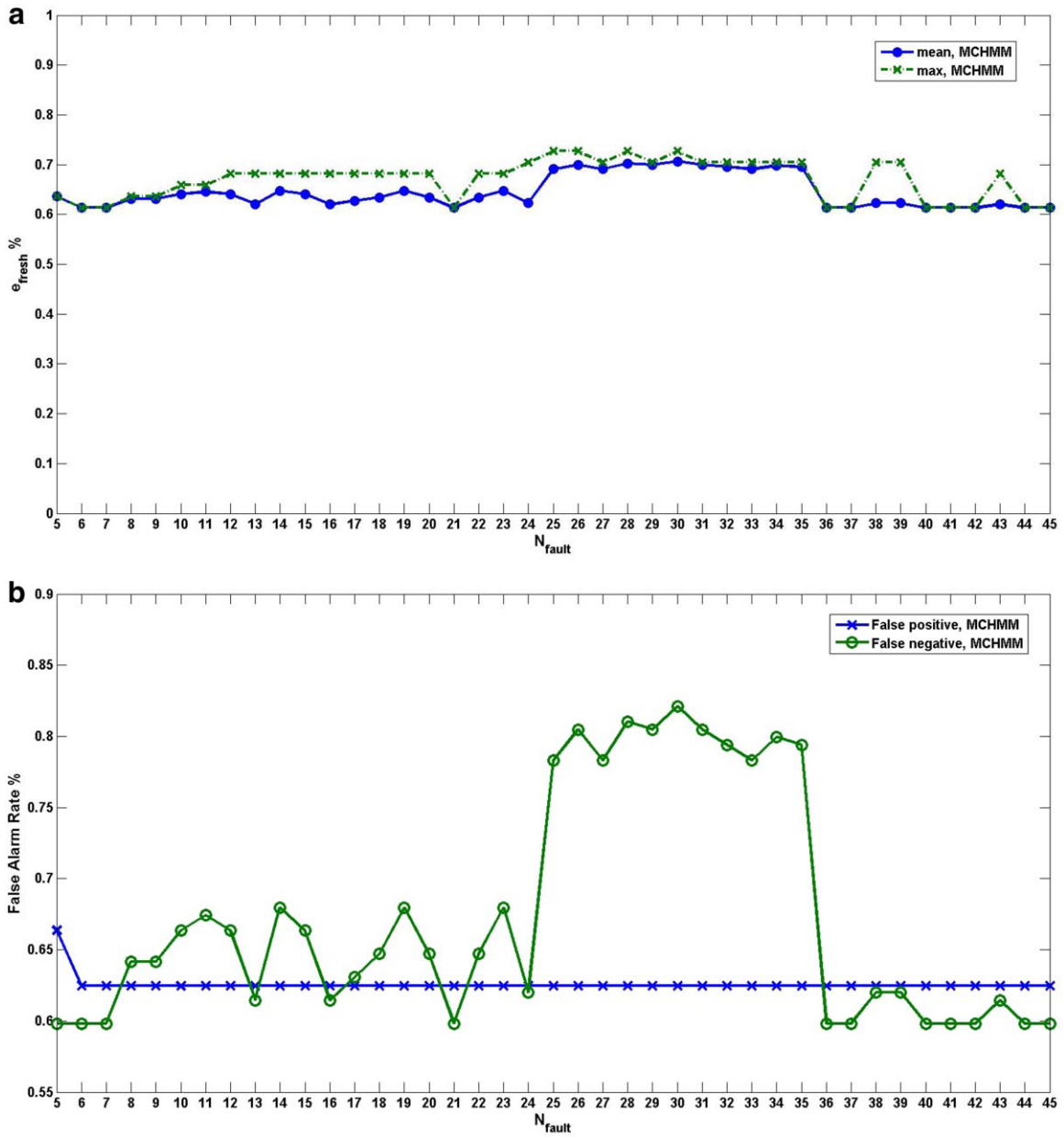


**Figure 6. Results of FDD using MCHMM on FCC Model.**

**(a) Variation of $e_{fresh}$% with $N_{fault}$ for FDD on FCC model. (b) False alarm rates for FDD on FCC model. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]**

MDHMM where such variations are relatively less. Nevertheless, MCHMM has an error rate of 11–12% as compared to MDHMM with an error rate of around 50% for $N_{fault} > 20$. It is also observed that both the average and the maximum error rates of MCHMM are smaller than the error rate of MDHMM at all values of $N_{fault}$. The difference between maximum and average error rates for the MCHMM-based scheme is seen to be always less than 10%. Moreover, this difference was seen to become smaller as the training data size was increased. The false positive and false negative rates for the two FDD schemes are plotted in Figure 5b. The false positive rate for MCHMM is found to be smaller than that of MDHMM at all values of $N_{fault}$. For the MDHMM-based scheme, the false negative rate is seen to increase and the false positive rate is seen to reduce as the training data size is increased. The number of false negatives ranges from 200 to 300 for most values of $N_{fault}$ in the case of MCHMM while it increases readily from 72 to 497 for MDHMM over the specified range of $N_{fault}$.

### *Results—Model IV FCC*

The fault detection and classification results for the FCC unit are charted in Table 2. It can be seen that the proposed MCHMM-based scheme could estimate the modes of operation with great accuracy. Trained on a dataset of 100 batches at a noise power of 0.0000001, the FGMM-based HMM could correctly estimate 4373 points out of 4400 observations. The proposed scheme produced estimates at the same error rate (0.61%) when tested on both training dataset and fresh data from a new run of the model. The error rate remained same even when the noise power was increased thousand fold to 0.0001. On increasing the noise power to 0.01, a slight increase of 0.03% in error rate was observed in both training data and

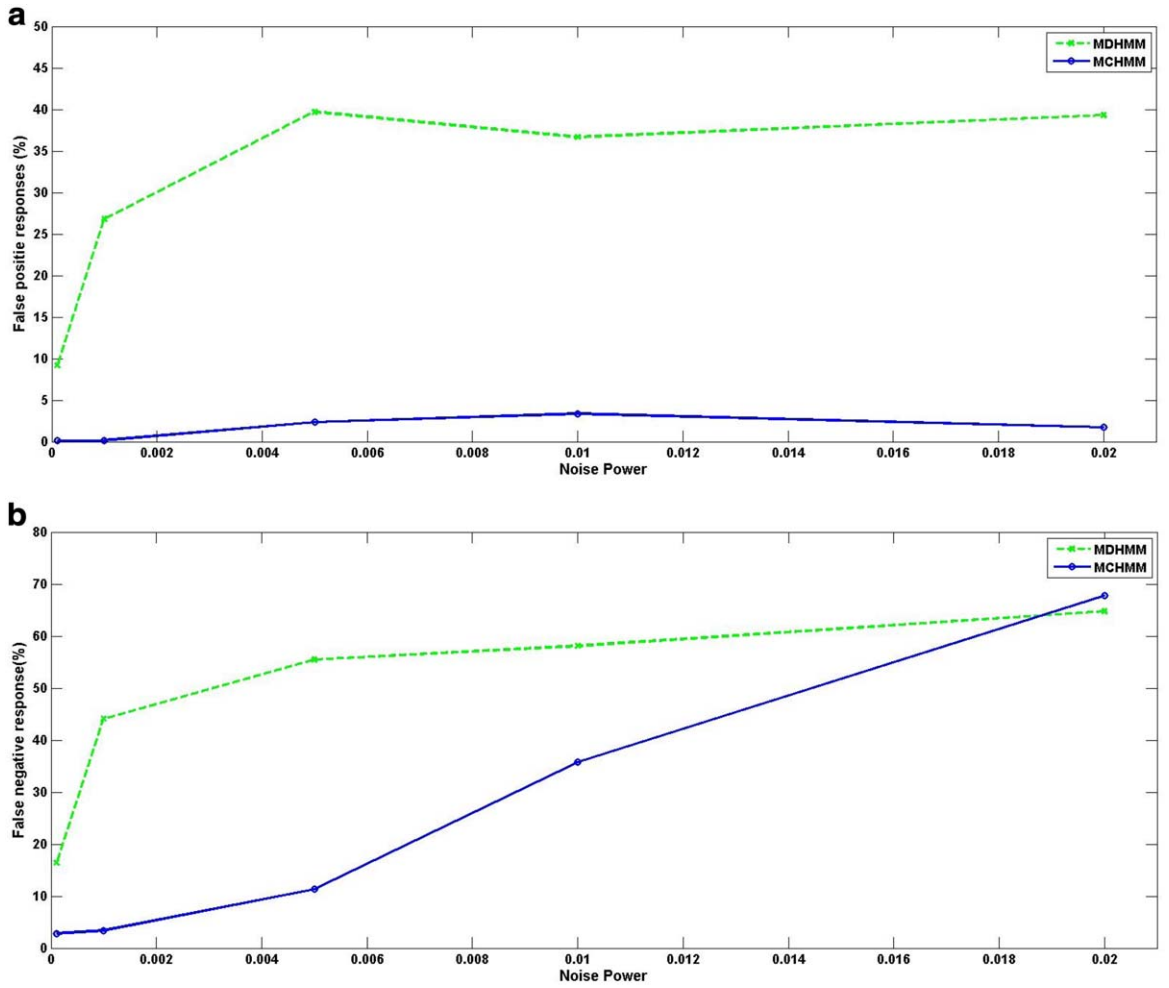


**Figure 7. Comparison of (a) false positive and (b) false negative rates for the MDHMM and MCHMM-based FDD schemes.**

**[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]**

fresh data tests. A similar trend was observed by changing the batch size. For the Amoco model of FCC system, the observation vector for the HMM contained 20 variables. Thus, the probability of occurrence of any particular observation vector, if computed by multiplying the individual probability of occurrence of each of the values, results in a diminutive number. Due to this, calculations involved in implementing the MDHMM-based method were often found to return the standard Not-a-Number response from MATLAB due to division operations by numbers rounded off as zero. Hence, attempts to compare the results of the proposed scheme to the prediction results of MDHMM-based FDD scheme were not successful.

Furthermore, 100 fresh batches with a noise power of 5.0, with 25 batches from each mode of operation, were monitored using the MCHMM algorithm using increasing values of $N_{\text{fault}}$ from 5 to 40. The $e_{\text{fresh}}\%$ is seen to remain considerably unchanged at around 0.64%. In the FCC case with 20 measurement variables, repetition of the algorithm on the same set of test data, using the same training set, are seen to yield more consistent results than in the case of the inverted pendulum where only four measurements variables were available. Also, the false positive rate for the proposed scheme on the FCC model is seen to remain almost constant when the training data size is increased.

The MDHMM-based scheme was not successfully implemented on the FCC model due to computational constraints as described above. The results obtained for MCHMM-based FDD scheme when tested on FCC model for increasing values of $N_{\text{fault}}$ are shown in Figure 6.

### *Comparison—False alarm rates*

The performance of the two FDD schemes at different noise powers were compared in terms of false positive and false negative rates. As the MDHMM-based FDD scheme was not successful in monitoring the FCC model, the comparison was done based on the monitoring results obtained from the inverted pendulum model. To compare the false alarm rates, both FDD schemes were first trained at a noise level of 0.0001 on a common training dataset consisting of 60 batches. Then, a fresh test dataset of the same noise power with 60 batches was generated. The two FDD schemes were tested on this dataset and the false positive and false negative rates were noted. The procedure was repeated by increasing the noise power of the datasets. The results of this study have been plotted in Figures 7a, b. It may be noted from the figures that, at all noise levels, false positive response rate given by the MCHMM based scheme is smaller than that given by the MDHMM-based scheme.

Also, the false negative rate is seen to increase with the increase in noise power for both schemes.

## Conclusions

In this work, we studied the effectiveness of a novel MCHMM-based FDD scheme. The training data was characterized by a FGMM. An optimal number of Gaussian components and their mixture parameters were computed by clustering the training data using the FJ algorithm. A continuous HMM was then generated by using segmental k-means algorithm for training. An updated Viterbi algorithm was used to estimate the hidden states in test datasets. Performance of the proposed MCHMM-based FDD scheme was compared with that of an MDHMM-based FDD scheme by testing on an inverted pendulum system and an FCC model. It was observed that at low noise powers, both MCHMM-based and MDHMM-based schemes are capable of accurate detection and diagnosis faults.

It was also observed that both schemes exhibit a loss in accuracy as the noise power is increased. For the cases studied, MCHMM-based FDD scheme was found to be considerably more accurate than MDHMM-based scheme at high noise powers. Also for the cases studied, the false positive rate for the MCHMM-based scheme was found to be smaller than that of MDHMM-based scheme at all noise powers. No significant trend was observed in false negative rates.

Relationship between the size of training dataset and the accuracy of fault diagnosis was studied. It was observed that the MCHMM-based fault detection scheme performed better than MDHMM-based scheme across all training data sizes used. The false positive rates for the MCHMM scheme were found to be lower than that for the MDHMM-based scheme at all training data sizes. No significant relationship was found between error rates and the training data size.

The results of FDD using this method is also dependent on the clustering of observation vectors which is carried out by the FJ algorithm. A certain level of randomness is introduced at this level. As the FJ algorithm is a stochastic clustering scheme that may cluster the same dataset differently on different trials. The proposed scheme can thus produce different monitoring results in different trials. Nevertheless it was found that these differences in monitoring results are reduced when the training data size is increased.

A major limitation of the proposed method lies in the requirement that the historical database that is used to train the MCHMM should include descriptions of all faults to be monitored. Performance of FJ algorithm used in clustering the data is known to reduce when Gaussian components with very different weights are present. The performance of the proposed FDD scheme in diagnosing such faults with significantly different Gaussian weights needs to be studied.

## Literature Cited


1. Chiang LH, Russell EL, Braatz RD. *Fault Detection and Diagnosis in Industrial Systems*. London: Springer, 2001.
2. Hwang I, Kim S, Kim Y, Seah CE. A survey of fault detection, isolation, reconfiguration methods. *IEEE Trans Control Syst Technol*. 2010;18(3):636–653.
3. Isermann R. *Fault-Diagnosis Applications*. Berlin Heidelberg: Springer-Verlag, 2011.
4. Yu J, Qin SJ. Statistical MIMO controller performance monitoring. *J Process Control*. 2008;18:277–296.
5. Qin SJ. Statistical process monitoring. *Comput Chem Eng*. 2003;17: 480–502.
6. Venkatasubramanian V, Rengaswamy R, Kavuri SN. A review of process fault detection and diagnosis. Part I: quantitative model-based methods. *Comput Chem Eng*. 2003;27:293–311.
7. Venkatasubramanian V, Rengaswamy R, Kavuri SN, Yin K. A review of process fault detection and diagnosis. Part III: process history based methods. *Comput Chem Eng*. 2003;27:327–346.
8. Dunia R, Qin SJ. Joint diagnosis of process and sensor faults using principal component analysis. *Control Eng Pract*. 1998;67: 457–469.
9. MacGregor JF, Jaeckle C, Kiparissides C, Koutoudi M. Process monitoring and diagnosis by multiblock PLS methods. *AIChE J*. 1994;40:826–838.
10. Chiang LH, Russell EL, Braatz RD. Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemom Intell Lab Syst*. 2000;50:243–252.
11. Wise BM, Gallagher NB, Butler SW, White DD, Barna GG. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. *J Chemom*. 1999;13: 379–396.
12. Nomikos P, MacGregor JF. Monitoring of batch processes using multi-way principal component analysis. *AIChE J*. 1994;40:1361–1375.
13. Chen J, Liu K. On-line batch process monitoring using dynamic PCA and dynamic PLS models. *Chem Eng Sci*. 2002;57:63–75.
14. Hwang DH, Han C. Real-time monitoring for a process with multiple operating modes. *Control Eng Pract*. 1999;7:891–902.
15. Chen J, Liu K. Mixture principal component analysis models for process monitoring. *Ind Eng Chem Res*. 1999;38:1478–1488.
16. Choi SW, Park JH, Lee IB. Process monitoring using a Gaussian mixture model via principal component analysis and discriminant analysis. *Comput Chem Eng*. 2004;28:1377–1387.
17. Thissen U, Swierenga H, deWeijer AP, Wehrens R, Melssen WJ, Buydens LMC. Multivariate statistical process control using mixture modeling. *J Chemom*. 2005;19:23–31.
18. Yu J, Qin SJ. Multimode process monitoring with Bayesian inference-based finite Gaussian mixture models. *AIChE J*. 2008;54: 1811–1829.
19. Figueiredo MAF, Jain AK. Unsupervised learning of finite mixture models. *AIChE J*. 2002;24:381–96.
20. Paalanen P, Kamarainen JK, Ilonen J, Kälviäinen H. Feature representation and discrimination based on Gaussian mixture model probability densities—practices and algorithms. *Pattern Recognit*. 2006; 39:1346–1358.
21. Chiang L, Kotanchek M, Kordon A. Fault diagnosis based on Fisher discriminant analysis and support vector machines. *Comput Chem Eng*. 2004;28:1389–1401.
22. Yeung DY, Ding Y. Host-based intrusion detection using dynamic and static behavioral models. *Pattern Recognit*. 2003;36:229–243.
23. Heck LP, McClellan JH. Mechanical system monitoring using hidden Markov models. *International Conference on Acoustics, Speech, and Signal Processing*. IEEE 1991:1697–1700.
24. Yu J. Multiway discrete hidden Markov model-based approach for dynamic batch process monitoring and fault classification. *AIChE J*. 2012;58:2714–2725.
25. Elliot RJ, Aggoun L, Moore JB. Discrete states and discrete observations. In: Karatzas I, Yor M, editors. *Hidden Markov Models Estimation and Control*. New York: Springer-Verlag, 1995.
26. Bhowmik T, van Oosten J, Schomaker L. Segmental k-means learning with mixture distribution for HMM based handwriting recognition. *Pattern Recognit Mach Intell*. 2011;6744:432–439.
27. Allahverdyan A, Galstyan A. Comparative analysis of Viterbi training and ML estimation for HMMs. In: Shawe-Taylor J, Zemel RS, Bartlett P, Pereira FCN, Weinberger KQ, editors. *Advances in Neural Information Processing Systems*, Curran Associates, Inc., New York, Vol. 24. 2011:1674–1682.
28. Dempster AP, Laird NM, Rubin DB. Maximum likelihood of incomplete data via the EM algorithm. *J R Stat Soc B*. 1977;39:1–38.
29. Bilmes JA. *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*. Tech. report ICSI-TR-97-021, ICSI, 1997.

30. Rabiner L. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc IEEE*. 1989;77:257–286.
31. Forney GD Jr. The Viterbi algorithm. *Proc IEEE* 1978;61:268–278.
32. Ding SX. Modelling of technical systems. *Model Based Fault Diagnosis Techniques*. Berlin Heidelberg: Springer, 2008:21–49.
33. Patton RJ, Klinkhieo S. Actuator fault estimation and compensation based on an augmented state observer approach. *48th IEEE Conference Decision and Control/28th Chinese Control Conference, CDC/CCC 2009*. IEEE: Shanghai, China, 2009:8482–8487.
34. McFarlane RC, Reineman RC, Bartee JF, Georgakis C. Dynamic simulator for a model IV fluidized catalytic cracker unit. *Comput Chem Eng*. 1993;17:275–300.
35. Ali E. *Simulink Module Archive*. Available at http://faculty.ksu.edu.sa/Emad.Ali/Pages/SimulinkModule.aspx. Accessed on December 12, 2013.

*Manuscript received Sept. 20, 2013, and revision received Dec. 24, 2013.*