Fault Detection for Systems with Multiple Unknown Modes and Similar Units - Part I*

Anwer Bashi
Computrols, Inc.
New Orleans, LA, U.S.A.
anwerb@gmail.com

Vesselin P. Jilkov X. Rong Li
Department of Electrical Engineering
University of New Orleans
New Orleans, LA, U.S.A.
vjilkov@uno.edu, xli@uno.edu

Abstract – This paper considers fault detection for large-scale practical systems with many nearly identical units operating in a shared environment.

A special class of hybrid system model is introduced to describe such multi-unit systems, and a general approach for estimation and change detection is proposed. A novel fault detection algorithm is developed based on estimating a common Gaussian-mixture distribution for unit parameters whereby observations are mapped into a common parameter-space and clusters are then identified corresponding to different modes of operation via the Expectation-Maximization algorithm. The estimated common distribution incorporates and generalizes information from all units and is utilized for fault detection in each individual unit.

The proposed algorithm takes into account unit mode switching, parameter drift, and can handle sudden, incipient, and preexisting faults. It can be applied to fault detection in various industrial, chemical, or manufacturing processes, sensor networks, and others. Two illustrative examples are presented, and a discussion on the pros and cons of the proposed methodology is provided.

Keywords: fault detection, FDD, hybrid system, multiple model, estimation, expectation-maximization, EM, HVAC.

1 Introduction

This paper considers fault detection for a system with many units operating in a shared environment such that the parameters of the units are somehow correlated with each other. Practical implementation and results are not discussed here, but appear in the companion paper [1]. The primary focus of this research is on large-scale practical systems such as heating, ventilation, and air-conditioning (HVAC) systems, some industrial, chemical, or manufacturing processes, or specialized sensor networks.

Numerous approaches have been suggested for fault detection in HVAC or industrial processes, [2, 3, 4]. Currently, most automatic fault detection in HVAC is of the model-free, non-parametric form, such as issuing alarms on out-of-range sensors. Model-based fault detection is hard to make generally applicable since detailed equipment models tend to be manufacturer and model dependent, and manufacturers do not generally provide reference models. One of the main problems with model-based methods for HVAC is that parameters for unit models often drift, sometimes quite dramatically, given different externalities. While detailed models can account for such a drift they would require data that is almost surely not readily available to “key” the model to a particular unit. Another complicating issue is that units often have more than one mode and they operate under a different parameter distribution for each mode, in general. A basic example of different modes is a three-way valve which mixes cold and hot fluids. We can consider each model to be representative of a different mode for operation.

Failures can be classified as sudden or incipient. Sudden failures are often the simplest form to diagnose since they usually have a dramatic impact on performance, which can be detected at a number of downstream sensors. However, incipient or gradual failures are difficult to detect since they present as a slow degradation in performance which can only be seen over time. Techniques based on identifying a model and using performance or error metrics to detect failure will find it difficult to identify a failure which occurs at a rate slower (often, much slower) than the model drifts under normal conditions. Another class of failure which has largely been ignored in the fault detection literature is a “preexisting” failure. This is because a model of correct operation is impossible to identify simply by looking at a unit after it has already failed, and so most practical fault detection techniques are simply inapplicable to this type of fault.

In this paper we propose an approach that addresses the above practical issues. A special class of stochastic hybrid system model is introduced to describe large-scale systems with many nearly identical units operating in a shared envi-
ronment and a general algorithm for estimation and change detection is proposed. It is based on estimating a common Gaussian-mixture distribution for unit parameters whereby observations are mapped into a common parameter-space and clusters are then identified corresponding to different modes of operation via the Expectation-Maximization algorithm. The estimated common distribution incorporates information from all units and is utilized for fault detection in each individual unit. The proposed algorithm takes into account unit mode switching, parameter drift, and is potentially capable of handling sudden, incipient, and preexisting faults.

2 Problem Formulation & Approach

2.1 Formulation

We consider a class of large-scale dynamic systems incorporating (a large number of) subsystems that have similar (or possibly, identical) structure and operate in a coupled manner as explained below.

Each subsystem, referred to as a unit, is described by the following model

\[
\begin{align*}
\xi_{k+1}^{(s)} &= f_k^{(s)} \left( \xi_k^{(s)}, \theta_k^{(s)}, w_{\xi_k}^{(s)} \right) \\
\theta_{k+1} &= g_k \left( \theta_k, m_k^{(s)} \right) \\
z_k^{(s)} &= h_k^{(s)} \left( \theta_k, \xi_k^{(s)} \right)
\end{align*}
\]

where \( k = 0, 1, 2, \ldots \) is the time index, \( s = 1, 2, \ldots, S \) is the unit index, \( x_k^{(s)} = \begin{bmatrix} \xi_k^{(s)} \theta_k^{(s)} \end{bmatrix} \) is the base-state vector of unit \( s \) (the components of which are discussed below), \( w_{\xi_k}^{(s)} \sim N(0, Q_{\xi_k}^{(s)}) \), \( w_{\theta_k}^{(s)} \sim N(0, Q_{\theta_k}) \) and \( v_k^{(s)} \sim N(0, R_{k}^{(s)}) \) denote white and mutually independent Gaussian process and measurement noises, respectively, and \( f_k^{(s)}, g_k, h_k^{(s)} \) are known state and measurement functions. The mode of operation of unit \( s \) is modeled through a Markov chain \( \{ m_k^{(s)} \}_{k=0,1,\ldots} \) with states \( m_k^{(s)} \in \mathbb{M} = \{ 1, 2, \ldots, M \} \) and initial and transition probabilities as given below, respectively

\[
P \left\{ m_0^{(s)} = i \right\} = \mu_0
\]

\[
P \left\{ m_{k+1} = j \mid m_k = i \right\} = \pi_{ij}
\]

for \( i, j = 1, \ldots, M \).

The ultimate aim is to detect whether a fault has occurred in one or more of the units \( s = 1, 2, \ldots, S \), given observation data \( Z_k = \{ z_k^{(s)} : s = 1, 2, \ldots, S, \} \) \( k=0 \).

The special structure and features of the above multi-unit hybrid system model can be exploited to achieve this fault detection goal. First, note that the state \( x_k^{(s)} = \begin{bmatrix} \xi_k^{(s)} \theta_k^{(s)} \end{bmatrix} \) of each unit is separated into two parts \( \xi_k^{(s)} \) and \( \theta_k^{(s)} \), the latter of which does not depend on the former. While the process \( \xi_k^{(s)} \) is determined through the unit-specific state transition model \( f_k^{(s)} \) in (1), the process \( \theta_k^{(s)} \) is determined through the state transition model \( g_k \) which is common for all units. Thus \( g_k \) models the common part in the structures of all units, meaning that for all units this part operates in the same manner. Furthermore, as seen from (4)–(5) the Markov transition model for the modal state \( m_k^{(s)} \), as well as the initial probabilities, is common for all units, and the statistical properties and parameters of \( \theta_0^{(s)} \) and \( w_{\theta_k}^{(s)} \) are also assumed to be the same for all units. Under these circumstances the processes \( \theta_k^{(s)}, s = 1, 2, \ldots, S \), will have the same statistical properties. If a fault occurs in a unit \( s' \) it will inevitably exhibit itself in changed statistical properties of \( \theta_k^{(s')} \) as compared to the remaining processes \( \theta_k^{(s)}, s \neq s' \).

2.2 General Approach

- **Multiple Model Filtering**

  For each unit \( s \) a multiple model filter is run to obtain the conditional estimates, covariances, and model weights \( \{ \tilde{\xi}_k^{(m,s)}, \tilde{\theta}_k^{(m,s)}, \tilde{\mu}_k^{(m,s)} \}_{m=1}^M \).

- **Common Distribution Identification**

  As justified above, it is assumed that \( \theta_k^{(s)}, s = 1, \ldots, S, \) have the same Gaussian-mixture distribution

\[
f(\theta_k; \alpha_k) = \sum_{i=1}^M \mu_k^{(i)} N \left( \tilde{\theta}_k^{(i)}, \tilde{\Sigma}_k^{(i)} \right)
\]

where the means and covariances \( \tilde{\theta}_k^{(i)}, \tilde{\Sigma}_k^{(i)} \) under mode \( i \) as well as the mixture weights \( \mu_k^{(i)} \) are unknown.

Estimation of the mixture parameters \( \alpha_k = \{ \tilde{\theta}_k^{(i)}, \tilde{\Sigma}_k^{(i)}, \mu_k^{(i)} \}_{i=1}^M \) can be done via an iterative Expectation-Maximization (EM) algorithm [5].

**E-step**

\[
P \left\{ m_k^{(s)} = i \mid \tilde{\theta}_k^{(s)}, \alpha_k^{\text{old}} \right\} = \frac{f \left( \tilde{\theta}_k^{(s)} \mid m_k^{(s)} = i, \alpha_k^{\text{old}} \right) P \left\{ m_k^{(s)} = i \mid \alpha_k^{\text{old}} \right\}}{\sum_{s=1}^S f \left( \tilde{\theta}_k^{(s)} \mid m_k^{(s)} = i, \alpha_k^{\text{old}} \right) P \left\{ m_k^{(s)} = i \mid \alpha_k^{\text{old}} \right\}}
\]

**M-step**

\[
\begin{align*}
\mu_k^{(i)\text{new}} &= \frac{1}{S} \sum_{s=1}^S P \left\{ m_k^{(s)} = i \mid \tilde{\theta}_k^{(s)}, \alpha_k^{\text{old}} \right\} \\
\tilde{\theta}_k^{(i)\text{new}} &= \frac{\sum_{s=1}^S P \left\{ m_k^{(s)} = i \mid \tilde{\theta}_k^{(s)}, \alpha_k^{\text{old}} \right\} \tilde{\theta}_k^{(s)}}{\sum_{s=1}^S P \left\{ m_k^{(s)} = i \mid \tilde{\theta}_k^{(s)}, \alpha_k^{\text{old}} \right\}} \\
\tilde{\Sigma}_k^{(i)\text{new}} &= \frac{\sum_{s=1}^S P \left\{ m_k^{(s)} = i \mid \tilde{\theta}_k^{(s)}, \alpha_k^{\text{old}} \right\} \tilde{\theta}_k^{(s)\text{'}i}}{\sum_{s=1}^S P \left\{ m_k^{(s)} = i \mid \tilde{\theta}_k^{(s)}, \alpha_k^{\text{old}} \right\}}
\end{align*}
\]
where $\tilde{\theta}^{(s,i)}_k = \hat{\theta}^{(s)}_k - \bar{\theta}^{(i)\text{new}}_k$.

- **Hypothesis Testing**
  Once the parameters of the common Gaussian-mixture distribution $\alpha_k = \left\{ \tilde{\rho}^{(i)}_k, \tilde{\Sigma}^{(i)}_k, \tilde{\mu}^{(i)}_k \right\}_{i=1}^M$ are identified, the test statistic $f(z^{(s)}_k; \alpha_k)$ can be computed for each unit $s$. Then the following unitary hypothesis is tested for detection of a fault in unit $s$

$$H_0^{(s)} : \tilde{z}^{(s)}_k \sim f(z^{(s)}_k; \alpha_k)$$

(11)

A fault in unit $s$ is declared when $H_0^{(s)}$ is rejected.

Detailed description of the overall algorithm and discussion on the implementation are provided next.

3. **Algorithm**

3.1 **Estimation for Static Mode Systems**

If we assume that the mode does not change during the estimation process, i.e., the transition probability matrix $\Pi = 1$, then only the current estimate is needed for estimating the mixture distribution at the current time, and since the current estimate includes information from all measurements up to that point in time, earlier estimates can be ignored.

Figure 1 shows a system with four modes and two parameters (A and B). Initially the unit mode-membership estimates are quite poor, but improve as time progresses.

This simply means that some initialization period is required before the estimates can be considered accurate. The estimation of the common distribution depends only on information contained in $\{\tilde{\theta}^{(s)}_k\}_{s=1}^S$, although the estimate of the common process contains information which could improve local estimates.

3.2 **Estimation for Dynamic Mode Systems**

In reality, the mode of a unit can change either gradually or suddenly. For the purpose of this discussion, we define a gradual change as one in which the parameters can successfully be tracked by the dynamic system model being used to represent the unit. A sudden change is one in which the period of adaptation for the parameters is not insignificant (i.e. several samples would be required before the estimated parameters can be said to represent the unit).

For gradual mode changes, parameter estimation can progress as described above for a static mode system with the understanding that mode changes are simply represented as a directed parameter drift. However, the mode probabilities used for EM estimation should explicitly take into account the mode transition probabilities.

For a sudden mode change, some technique should be used to properly initialize the parameter for the new mode otherwise the period of sudden parameter mismatch might be interpreted as a fault. Using standard hybrid estimation techniques may have problems since almost all of them assume that the new mode is somehow represented within the model set, or (less commonly) by a convex combination of the models as in [6]. To avoid this problem a different unit parameter reinitialization is implemented in the proposed algorithm. We try to retain the fast reinitialization of states and probabilities present in an interacting multiple model estimator, but still use a single estimator for $\tilde{x}^{(s)}_k, F^{(s)}_k$.

3.3 **Algorithm**

Notation for different estimates:

- $\tilde{x}^{(m,s)}_k \triangleq [\tilde{z}^{(m,s)}, \tilde{\mu}^{(m,s)}]$ = [local, local]
- $\tilde{x}^{(m,s)}_k \triangleq [\tilde{z}^{(m,s)}, \tilde{\mu}^{(m,s)}]^{\prime}$ = [local, global]
- $\tilde{x}^{(0,s)}_k \triangleq $ active mode matched estimator for unit $s$

- **Initialization**: In the absence of any relevant data, random initial parameter and state starting points and sufficiently diffuse covariances are chosen. Each unit is assumed to be in a mode, sampled according to the initial probability masses, until later reassigned by the mode change hypothesis test.

- **Estimation**: For each unit $s = 1, \ldots, S$ perform

  - **Mode Matched Estimation**: For each mode $m = 1, \ldots, M$ perform one-step filter update using the global (estimated mixture) parameters, and evaluate the likelihood $L^{(m,s)}_k$

    $$L^{(m,s)}_k \tilde{z}^{(m,s)}_k = \text{Filt} \left[ \tilde{z}^{(s)}_k, \tilde{x}^{(0,s)}_k, \tilde{\Sigma}^{(0,s)}_k \right]$$

    (12)

  - **Active Mode Matched Estimation**: Perform one-step filter update of the active mode estimator

    $$\tilde{x}^{(0,s)}_k = \text{Filt} \left[ \tilde{z}^{(s)}_k, \tilde{x}^{(0,s)}_k, \tilde{\Sigma}^{(0,s)}_k \right]$$

    (13)

- **Active Mode Change Detection**: For each unit $s = 1, \ldots, S$ perform a statistical test using the model likelihoods $\{L^{(m,s)}_k : m = 1, \ldots, M\}$ to establish whether a mode change has occurred

  $$\text{Detect} \left\{ L^{(m,s)}_k : m = 1, \ldots, M \right\}$$

(14)

Upon detection of an active mode change from $m^{(s)}_\kappa$ to $m^{(s)}_\kappa$, that occurred at time $\kappa \leq k$, reinitialize:

$$\tilde{x}^{(0,s)}_k, \tilde{\Sigma}^{(0,s)}_k = \text{Reinit} \left[ \tilde{x}^{(m^{(s)}_\kappa)}, \tilde{\Sigma}^{(m^{(s)}_\kappa)} ; \kappa, k \right]$$

(15)
• **Common Distribution Estimation:** Compute the common Gaussian-mixture parameters \(\hat{\mu}_k^{(m)}, \hat{\theta}_k^{(m)}, \hat{\Sigma}_k^{(m)}\), \(m = 1, 2, \ldots M\) via one iteration of the EM algorithm

\[
\hat{\mu}_k^{(m)} = \frac{1}{S} \sum_{s=1}^{S} \mu_k^{(m,s)} \\
\hat{\theta}_k^{(m)} = \frac{\sum_{s=1}^{S} \theta_k^{(m,s)}}{\sum_{s=1}^{S} \mu_k^{(m,s)}} \\
\hat{\Sigma}_k^{(m)} = \frac{\sum_{s=1}^{S} \mu_k^{(m,s)} (\hat{\theta}_k^{(0,s)} - \hat{\theta}_k^{(m)}) (\hat{\theta}_k^{(0,s)} - \hat{\theta}_k^{(m)})'}{\sum_{s=1}^{S} \mu_k^{(m,s)}}
\]

where the mode probabilities are given by

\[
\hat{\mu}_k^{(m,s)} = \frac{\mu_k^{(m,s)} L_k^{(m,s)}}{\sum_{m=1}^{M} \mu_k^{(m,s)} L_k^{(m,s)}}
\]

\[
\hat{\mu}_k^{(m,s)} = \sum_{i=1}^{M} \mu_k^{(s,i)} \pi_{im}
\]

• **Fault Detection:** For each unit \(s = 1, \ldots, S\) through the active mode estimator compute the likelihood

\[
L_k^{(s)} = f \left( z_k^{(s)} \left| \hat{x}_k^{(m,s)}, \hat{\Sigma}_k^{(m,s)}, \hat{\mu}_k^{(m)} \right|_{m=1, \ldots, M} \right)
\]

(21)

3.4 Remarks

Here we discuss some details of the algorithm while noting that there are obviously various methods for implementing its different parts such as estimation, change detection, reinitialization.

**Estimation (Filt):** For linear systems, a Kalman filter could be used for mode conditional filtering, however, most of the real-life parameter estimation problems we are interested in are non-linear, and thus an efficient and effective treatment of non-linear systems is critical. For non-linear systems, we use an unscented transform filter (UTF) because of its simplicity and effective treatment of non-linear systems, as well as its relatively low computational burden.

**Mode Change Detection (Detect):** Function Detect detects a change in the mode using likelihood and/or probability information about the unit. An SSPRT or CUSUM algorithm may be used [7], [8], [9], both of which are optimal but require exact knowledge of the models.

Alternately, one can use a simpler change detector such as a simple probability test: A mode change is declared when the probability of one of the (non-active) modes becomes greater than all others by some threshold value, i.e., iff

\[
\hat{P}_k^{(m,s)} > T_s + \hat{P}_k^{(m,s)} \forall m \neq m^*
\]

A threshold is used to avoid switching “chatter” for poorly mode-differentiated units (those which are nearly equidistant, in terms of likelihood, from more than one mode) in the presence of noise.

**Reinitialization (Reinit):** When a mode change occurs, we need to use our best estimate of parameters and state to reinitialize the active mode matched estimator. For systems where units gradually drift between modes, we can simply identify the new mode and allow the mode matched estimators to identify the parameters normally. For these systems, the reinitialization algorithm can be left blank since the active mode matched estimator is likely better than any of the mode matched estimators.
However, for systems where the mode jumps suddenly, we should initialize the parameters to our best estimate for that mode. One way of doing this is to use the parameter estimates from the mode-matched estimator for the new mode, \( \tilde{x}_k^{(m^*)}, \Sigma_k^{(m^*)} \), to reininitialize the active filter:

\[
\begin{align*}
\bar{m}_k^{(s)} &\leftarrow m^*
\bar{\theta}_k^{(0,s)} &\leftarrow \bar{\theta}_k^{(m^*)}
\tilde{\Sigma}_k^{(0,s)} &\leftarrow \tilde{\Sigma}_k^{(m^*)}
\end{align*}
\]

\[
\begin{bmatrix}
\tilde{x}_k^{(0,s)}, \tilde{\Sigma}_k^{(0,s)}
\end{bmatrix} = \text{Filt} \begin{bmatrix}
\tilde{x}_k^{(s)}, \tilde{x}_k^{(0,s)}, \tilde{\Sigma}_k^{(0,s)}
\end{bmatrix}
\]

In some cases, local states exist which are mode-related, but not globally mode-correlated (i.e., they change with respect to mode, however are not useful in clustering modes globally). For these cases, the relevant local states (which act more like parameters) are also reinitialized from the appropriate running estimator.

Optionally, if the mode-switching algorithm provides a jump-time, smoothing may be performed to improve the estimate. Given the multi-mode nature of the problem, multiple-model smoothing techniques must be used, as in [10, 11, 12, 13].

**Fault Detection:** The no-fault hypothesis \( H_0 \) is represented by the Gaussian-mixture distribution

\[
L\left( z_k^{(s)} \right) = \sum_{m=1}^{M} \mu_{k-1}^{(m,s)} \mathcal{N}\left( z_k^{(s)}; \bar{x}_k^{(m,s)}, \tilde{\Sigma}_k^{(m,s)} \right)
\]

However, since we have no model for \( H_1 \), we cannot use detection algorithms based on likelihood ratios, such as the Neyman-Pearson (NP) or Generalize Likelihood Ratio (GLR) tests. Detection algorithms based on input estimation are not really suitable for systems where the input is known with more certainty than the model. See [14], [15], [16] for details on measurement residual-based, model-probability-based, and input-based detection algorithms. In such a situation our test statistic is simply the log likelihood of the measurement residual with an window of size \( W \):

\[
T_k^{(s)} = \frac{1}{W} \log \left( L\left( z_k^{(s)} \right) \right)
\]

with \( T_0^{(s)} = \infty \) set to a value large enough to cause the initially poor parameter estimates not to register as failures.

If \( T_k^{(s)} < \gamma_f \), where \( \gamma_f \) is the failure threshold, a fault is declared.

**Unit Exclusion:** When attempting to detect faults based on set statistics, it is useful to be able to look at the statistics without the contribution of failed units. This allows for detection of small faults without degrading the estimate through measurements originating from the faulty unit. Unit exclusion aids in incipient and preexisting failure detection.

Units with a total likelihood less than a threshold \( T_c \) are excluded from the cluster statistics calculation (16)–(20). \( T_c \) may be set to a small number (e.g. \( 10^{-4} \)) for simplicity, or for a more sophisticated treatment, assume different thresholds depending on the number of units estimated to be in a particular mode, their likelihoods, and the desired number of units to exclude.

Obviously, the selective exclusion of some data points changes the sample mean and variance of the cluster. In order to regenerate the original statistics, the covariance can be boosted as noted in the Appendix. If the truncation is symmetric (as it is when truncating based on likelihood), then the mean is not changed.

### 4 Examples

A couple of simple examples are given to illustrate possible choices of \( \xi \) and \( \theta \). The examples assume \( w_k, v_k \sim \mathcal{N}(0, 1^2) \) are standard noise, while the process and measurement noise covariance matrices \( Q \) and \( R \) are selected appropriately.

#### 4.1 Static System - Valve Stiction

Stiction is the resistance to the start of motion caused by static friction and is a common problem in real-world valves. Excessive stiction can be considered a fault condition.

Given an applied force \( F_u \), a static resistance force \( F_s \), and the current velocity \( v \), we can model the resulting velocity after stiction as:

\[
v_s(v, F_u, F_s) = \begin{cases} 
0, & v = 0 \text{ and } |F_u| < F_s \\
v + K F_u, & \text{otherwise}
\end{cases}
\]

If we define our state, measurement, and input vectors as:

\[
\begin{align*}
x &\triangleq \begin{bmatrix} s \\ v \\ F_s \\ K \end{bmatrix} \\
z &\triangleq \begin{bmatrix} s \end{bmatrix} \\
u &\triangleq [F_u]
\end{align*}
\]

we can describe the system dynamics using the following equations:

\[
x_k = \begin{bmatrix} s + \Delta T v \\ v_s(v, F_u, F_s) \\ F_s \\ K \end{bmatrix} + \sqrt{Q} w_k \\
z_k = \begin{bmatrix} s \end{bmatrix}_k + \sqrt{R} v_k
\]

Here, the global parameters are the static resistance and action gain, while the local states are valve position and velocity:

\[
\theta \triangleq [F_s, K], \quad \xi \triangleq [s, v]^T
\]
For this example, we assume an installation with 300 pneumatic valves of a particular size, from 4 different manufacturers. In our example, the valves from the different manufacturers have slightly different dynamic statistics, which we represent using 4 different static modes.

4.2 Dynamic System: Three-Way Valve

A set of simple 2nd-order auto-regressive moving average (ARMA) models may be used to describe the behavior of a three-way combination heating-cooling valve with a dead-band. We designate cooling mode as mode 1, dead-band as mode 2, and heating mode as mode 3.

We define $a$, $b$ to be the autoregressive and moving average parameters, respectively. Valve position and temperature are given by $s$ and $t$. $q^{-i}x_k$ is the delay operator which returns $x_{k-i}$. The output we are interested in is temperature $z$. We have:

$$x_k = [a_1 \ a_2 \ b_0 \ b_1]^T$$

$$F_k = I_{44}, \quad H_k = [q^{-i}z \ q^{-2}z \ q^0s \ q^{-1}s]^T_k$$

$$x_k = F_kx_{k-1} + \sqrt{Q}w_k$$

$$z = H_kx_k + \sqrt{R}v_k$$

The parameters and local states for modes 1 and 3 are given by

$$\xi_k^{(m,s)} \triangleq [\text{empty}]$$

$$\theta_k^{(m,s)} \triangleq [a_1 \ a_2 \ b_0 \ b_1]^{(m,s)}_k$$

For mode 2 (dead-band), we fix the parameters as:

$$\theta_k^{(2,s)} \triangleq [1 \ 0 \ 0 \ 0]^T$$

5 Limitations and Capabilities

The assumptions underlying the presented algorithm are novel. Because of this, the algorithm has a number of interesting limitations and capabilities.

5.1 Limitations

**Ensemble Requirement** The algorithm is dependent on the availability of multiple correlated units. It is completely inapplicable to systems with a single unit, and suffers performance degradation for systems with few units (e.g., less than twenty units for a 2-mode system).

**Mode Representation Requirement** In order to detect jumps from one mode to another, the new mode must have some representation, i.e., one or more units must already be in that mode, otherwise the jump could be considered a fault. Obviously, this is not an issue with single-mode systems, however one of the most significant strengths of the algorithm is the ability to effortlessly adapt as units jump from mode to mode.

**Slower Detection Than Uni-Mode, No-Drift Algorithms** Since the distribution of the global parameters may have a larger covariance than that of a local parameter estimate, algorithms that address only a single mode and assume that parameter drift is either non-existent, or can be ignored over the time in which a fault occurs, will generally perform better (if the stated assumptions are true). Effectively, this limits applicability of single-mode slow-drift parameter algorithms to detection of sudden failures only, which may be sufficient for many problem domains. However, it should be noted that the proposed solution can be used to augment single-mode slow-drift algorithms, thereby allowing fast detection using such algorithms, while providing for detection of incipient or preexisting failures.

**Central Processing Requirement** As presented, the solution assumes a central location where information must be collected for analysis, which may be problematic for sensor networks. However, the information transfer requirement is minimal, consisting not of actual sample data but rather parameter statistics estimates (such as mean and covariance). Additionally, complete system synchronization is not an absolute requirement, but information from most nodes should be collected at a rate faster than parameter drift is expected to occur.

5.2 Capabilities

**Multi-Mode Capable** The proposed solution excels at addressing multi-mode systems, even when models are initially unavailable for all the modes.

**Incipient and Preexisting Fault** Parameter estimation explicitly accounts for parameter drift, and since fault detection does not depend on sudden deviation of local parameters, incipient and preexisting faults can be detected. This differs from techniques that account for gradual parameter drift, and assume that a fault is detectable by a sudden parameter change.

**Can Include Tertiary Mode Information** If information is available on which mode is active for a particular unit, this information can be incorporated into the algorithm through the transition probability matrix or the initial mode probabilities.

**Fast Parameter Initialization After Jumps** The probability distribution for global parameters is being constantly estimated using data from every unit at the site. So parameter estimation after the sudden jump of a unit has the advantage of being able to use the global information to set its initial values, even if that unit has never before been seen in that mode. This reduced the “improbability”-spike which would occur otherwise, greatly reducing false positives.
6 Conclusion

A novel technique has been proposed for fault detection in large-scale systems with many nearly identical units operating in a shared environment.

A fault detection algorithm has been developed based on estimating a common Gaussian-mixture distribution for unit parameters via the Expectation-Maximization algorithm. The estimated common distribution incorporates and generalizes information from all units and is utilized for fault detection in each individual unit.

The proposed algorithm can function when information about the system modes, system faults, active mode for a unit, and structure and dynamics of the underlying data generation mechanism (the “true model”) is incomplete or missing. The algorithm takes into account unit mode switching, parameter drift, and can handle sudden, incipient, and pre-existing faults. It can be applied to fault detection in various industrial, chemical, or manufacturing processes, sensor networks, and others.

No simulation results have been presented here, but a companion paper [1] demonstrates algorithm implementation and performance.

Appendix

The calculation of parameters for a truncated normal distribution (TND) is a recurring problem in real-world testing. A Gaussian distribution assumes limits of $\pm \infty$. In practice, there are usually upper and lower limits imposed on an experiment, so that the actual distribution being measured is a truncated normal distribution (see, for example, [17], [18]).

Consider the multivariate normal distribution given by:

$$
\mathcal{N}\left(\tilde{z}, P \right) \triangleq \frac{1}{(2\pi)^{\frac{N}{2}} |P|^{\frac{1}{2}}} \exp \left[ -\frac{1}{2} \tilde{z}^T P^{-1} \tilde{z} \right] 
$$

(27a)

where $\tilde{z} = z - \bar{z}$.

If we condition the observation on $\tilde{z}^T \tilde{z} \leq t$, the total probability contained in this region can be denoted by,

$$
\Phi(t) = \int_{\tilde{z}^T \tilde{z} \leq t} \mathcal{N}(\tilde{z}) d\tilde{z}
$$

(28a)

Hence, after scaling, the probability distribution becomes:

$$
\mathcal{T}(\tilde{z}; t) = \begin{cases} 
\mathcal{N}(\tilde{z}) / \Phi(t) & \text{if } \tilde{z}^T \tilde{z} \leq t \\
0 & \text{elsewhere}
\end{cases}
$$

(29a)

Using Likelihood as a Gating Threshold

The focus of most TND work considers arbitrary thresholds described in terms of left- and right-cutoff values. For multimodal TND’s, a matrix of cutoff values describes an $N$-dimension box outside of which observations are rejected.

For our purposes, we wish to define the threshold in terms of a probability boundary, beyond which observations are rejected. This results in a mean- and covariance-invariant boundary, meaning that the observation does not have to be pre-whitened or translated into a zero-mean, unit covariance form before truncation since the likelihood explicitly considers the distribution statistics. The mean of the original distribution is equal to the mean of the truncated distribution if likelihood is chosen as the truncation threshold measure since the Gaussian distribution is symmetric.

Unfortunately, an analytical solution for the mean and covariance of arbitrarily dimensioned TND’s is very difficult to come by. For univariate TND’s, the relationship between the mean and covariance of the truncated sample and the original (untruncated) distribution is well known [17], [19], as are the statistics of bivariate [20], [21], [22], and trivariate [23] distributions, however this does not help in the general multivariate case.

Nevertheless, while no simple closed form solution exists, Monte-Carlo simulations can be used to generate a lookup table which is independent of the actual covariance of the original distribution.

References


