# Condition-based sensor-health monitoring and maintenance in biomanufacturing

**Aditya Tulsyan, Chris Garvin, Cenk Undey**

*Digital Integration & Predictive Technologies, Amgen Inc., USA*
*E-mail: {tulsyan, garvinc, cundey}@amgen.com*

**Abstract:** In the Biotechnology 4.0 paradigm, process analytical technology (PAT) tools are being increasingly deployed in biomanufacturing to gain improved process insights through extensive use of advanced and automated sensing techniques. Critical parameters, such as pH, dissolved oxygen (DO), temperature, and metabolite concentrations, are routinely measured and controlled in a cell culture process. While these extensive networks of sensors generate critical process information and insights, they are also prone to failures and malfunctions. In this paper, we propose a condition-based maintenance (CbM) framework for real-time sensor-health management, with a focus on fault detection, diagnosis, and prognostics. To this effect, a slow-feature analysis (SFA)-based platform is proposed for the detection and diagnosis of sensor-health. For health prognostics, a Gaussian process (GP) model is proposed for forecasting the remaining useful life (RUL) of the sensor along with the probability of failure. The efficacy of the proposed sensor-heath management strategy is demonstrated in a biomanufacturing process.

*Keywords:* Condition-based monitoring, maintenance, diagnostics, prognostics, sensor-health

## 1. INTRODUCTION

Stable production of biotherapeutic protein requires real-time monitoring and control of cell culture processes. A commercial bioreactor is regularly fitted with a multitude of critical sensors that work in tandem to ensure consistent product quality and overall process safety. While these sensors generate critical process information and insights, they are also prone to failures and malfunctions. In biomanufacturing, engineers, process plant owners and operators are faced with the challenge of keeping their sensors calibrated and working efficiently, while reducing the overall maintenance cost.

A majority of sensors used in biomanufacturing follow a time-based maintenance (TbM) strategy; wherein, maintenance decisions (e.g., calibration times, sensor change-outs) are determined based on historical failure time analyses (Yam et al., 2001). For example, most of the sensors have a predetermined usage cycle, after which these sensors are replaced, irrespective of their remaining useful life (RUL). Although TbM reduces the probability of sensor failures and the frequency of unplanned maintenance, it cannot eliminate the occurrences of random failures. Furthermore, in TbM, most decisions are made by experienced planners according to suggestions from sensor manufacturers, historic breakdowns or failure data, operating experience, and judgment of maintenance staff and technicians (Peng et al., 2010). In recent years, growing maintenance costs and increased downtime caused by unexpected sensor failures have motivated biopharmaceutical companies to explore alternative preventive maintenance (PM) strategies for sensor-health management.

In the last decade, emerging technologies like big data analytics, the Internet of Things, and cloud data storage have enabled many industries to adopt condition-based maintenance (CbM) strategies in manufacturing (Shin and Jun, 2015). CbM is a PM program that makes maintenance decisions in real-time based on the information collected about the underlying system, allowing maintenance activities to be performed only when necessary (Prajapati et al., 2012; Jardine et al., 2006). CbM is a part of a broader maintenance framework, referred to as prognostics health management (PHM). PHM provides a robust framework for real-time measuring, recording, and monitoring the extent of deviation and degradation from normal operation conditions (Tsui et al., 2015). A typical work-flow in a PHM system includes three critical steps: (a) fault detection and diagnostics, (b) prognostics, and (c) condition-based maintenance. The first task is that of fault detection and diagnosis; wherein, a fault detection routine determines if a system is experiencing problems, which is then identified by a fault diagnostics routine. In the second task, fault prognostics provide the predictions of the future reliability of a system by assessing the extent of degradation of the system from its expected normal operations. Finally, the CbM program takes decisions based on the information collected from the prognostics model. All of these three tasks need to be executed in real-time.

Fault diagnosis and prognostics are the two critical components of PHM. Fault detection and diagnosis methods can be classified as model-based and data-based. In the model-based category, some forms of first principles models linking failure modes and observations are proposed. These techniques require the estimation of complete mechanistic knowledge, which might not be the most practical solution for most industrial applications. Several data-based methods based on hypothesis testing, cluster analysis, and statistical process control (SPC), support vector machines

(SVM) and neural networks have also been proposed. For example, Ma and Li (1995) proposed a composite hypothesis test for detection of localized defects in rolling element bearings and Fugate et al. (2001) proposed a vibration-based damage detection using SPC.

There are two main aspects of fault prognostics: (a) predicting RUL of a system before a failure occurs; (b) predicting the probability that a system operates without a fault until the next inspection. Most of the literature on prognostics deal with the former problem. Similar to diagnostics, prognostics methods can also be classified into model-based and data-based methods. Model-based RUL estimation methods have been proposed for several mechanical systems, including bearings (Li et al., 1999), gas turbines (Kacprzynski et al., 2001) and rotor shafts (Oppenheimer and Loparo, 2002). Several statistical methods have also been proposed for fault prognostics. For example, Yan et al. (2004) employed a logistic regression model to calculate the probability of failure for given condition variables and an ARMA time series model to trend the condition variables. Pecht (2010) proposed a Kalman filter to track failure probability vectors. Other prognostics methods include exponential smoothing (Ji et al., 2018) and autoregressive (Wang et al., 2017) models.

In this paper, we propose a PHM framework for sensor-health management in biomanufacturing, with a focus on fault detection, diagnosis, and prognostics. To this effect, a slow-feature analysis (SFA)-based monitoring framework is proposed for real-time fault detection and diagnosis of sensor-health. SFA is an unsupervised dimension reduction technique (Wiskott and Sejnowski, 2002). We project the sensor data onto a latent space based on the signal *slowness*, referred to as slow-feature space. A majority of sensor faults, such as calibration errors and malfunction have distinct fault dynamics that are monitored using their projections in the feature space. We use SFA-derived control charts for both real-time detection and diagnoses of various fault dynamics propagating through a sensor. SFA has been previously proposed for real-time monitoring of process systems and controller performance (Shang et al., 2015). In this paper, we extend the application of SFA to sensor-health monitoring. For sensor-health prognostics, we propose the use of Gaussian processes (GPs). As a Bayesian non-parametric method, GPs not only provide a forecast of the RUL of a sensor but also provides the confidence estimates for the RUL. This information can, in turn, be used by the CbM program to make any maintenance-related decisions. The efficacy of the proposed sensor-heath management strategy is demonstrated in the sterilization steps of industrial cell culture process.

## 2. PRELIMINARIES: SFA AND GP

**SFA**: We briefly introduce SFA in this section; however, for a detailed exposition, the reader is referred to Wiskott and Sejnowski (2002); Shang et al. (2015). The SFA is an unsupervised learning algorithm for dimension reduction in temporally correlated datasets. SFA is based on the slowness principle. Before introducing SFA, we first define the slowness of a stochastic signal, $x(t)$ as follows:

$$\Delta(x) := \langle \dot{x}(t)^2 \rangle_t, \tag{1}$$

where $\dot{x}(t) \approx x(t) - x(t-1)$ is the first-order time difference approximation of $\dot{x}(t)$, and $\langle x \rangle_t$ denotes the time-averaging of $x(t)$. Given $N$ data samples, $\{x(1), \ldots, x(N)\}$, the time averaging of $x(t)$ can be approximated as follows

$$\langle x \rangle_t \approx \frac{1}{N} \sum_{t=1}^{N} x(t). \tag{2}$$

Physically, $\Delta(x)$ in (1) provides a natural measure of how fast the signal $x(t)$ evolves in time.

Now, for a given $m$-dimensional input signal, $\mathbf{x}(t) \in \mathbb{R}^m$, SFA aims to map the input signal to an $m$-dimensional slow feature (SF) space, denoted as $\mathbf{s}(t) \in \mathbb{R}^m$, such that

$$s_i(t) = g_i(\mathbf{x}(t)), \tag{3}$$

for all $i = 1, \ldots, m$. SFA computes the function $g_i$ in (3) by solving the following constrained optimization problem:

$$g_i^\star = \underset{g_i(\cdot)}{\arg\min} \ \langle \dot{s}(t)^2 \rangle_t, \quad i = 1, \ldots, m \tag{4}$$

such that:

$$\langle s_i \rangle_t = 0; \qquad \text{(zero mean)} \tag{5a}$$

$$\langle s_i^2 \rangle_t = 1; \qquad \text{(unit variance)} \tag{5b}$$

$$\langle s_i s_j \rangle_t = 0 \ \forall i \neq j. \qquad \text{(decorrelation and order)} \tag{5c}$$

The optimization problem in (4) computes the functions $g_i$ $(1 \leq i \leq m)$ by projecting the input signal onto the feature space, where each feature, $s_i$ has as slow variation as possible. The constraints (5a) and (5b) eliminate the trivial solution $s_i(t) = \text{const.}$, while (5c) ensures that all SFs are independent of each other. For additional details, the reader is referred to Shang et al. (2015).

The optimization problem in (4) is in general difficult to solve, except in some constrained cases. For example, for a linear case, $s_i$ is a linear combination of inputs, such that $s_i$ $(1 \leq i \leq m)$ can be written as follows:

$$s_i(t) = \mathbf{w}_i^{\mathrm{T}} \mathbf{x}(t), \tag{6}$$

where $\mathbf{w}_i \in \mathbb{R}^m$ is the coefficient associated with the SF, $s_i$. In compact notation, we can rewrite (6) as follows:

$$\mathbf{s}(t) = \mathbf{W}\mathbf{x}(t), \tag{7}$$

where $\mathbf{s}(t) = [s_1(t), \ldots, s_m(t)]^{\mathrm{T}}$ and $\mathbf{W} \equiv [\mathbf{w}_1, \ldots, \mathbf{w}_m]^{\mathrm{T}}$ is the coefficient matrix. Solving (4) with (7) entails estimating the coefficient matrix $\mathbf{W}$ in (7). Fortunately, solving the constrained optimization problem in (7) can be avoided for the linear case through problem reformulation. Now, if each dimension of $\mathbf{x}(t)$ is scaled to zero mean then given (7), the optimization in (4) can be recast as the generalized eigenvalue decomposition (GED) problem, such that $\mathbf{W}$ satisfies the following (Shang et al., 2015):

$$\mathbf{R}_{\dot{x}\dot{x}} \mathbf{W} = \mathbf{R}_{\mathbf{xx}} \mathbf{W} \boldsymbol{\Omega}, \tag{8}$$

where $\mathbf{R}_{\dot{\mathbf{x}}\dot{\mathbf{x}}} = \langle \dot{\mathbf{x}}\dot{\mathbf{x}}^{\mathrm{T}} \rangle_t$ and $\mathbf{R}_{\mathbf{xx}} = \langle \mathbf{x}\mathbf{x}^{\mathrm{T}} \rangle_t$ denote the covariance matrices for $\dot{\mathbf{x}}(t)$ and $\mathbf{x}(t)$, respectively; $\mathbf{W}$ is the matrix of $m$ generalized eigenvectors, which are also the coefficients in (7); and $\Omega = \mathrm{diag}\{\omega_1, \ldots, \omega_m\}$ contains generalized eigenvalues along its diagonal, which satisfy $\omega_i = \langle \dot{s}_i^2 \rangle_t$ and are arranged in an ascending order. Note that $\Omega$ introduces an ordering in the SFs based on their slowness, such that $s_1$ denotes the slowest feature, $s_2$ denotes the second slowest feature and so on.

The procedure for solving the GED problem is standard; for the sake of brevity, it is not discussed here, but can be found detailed in Shang et al. (2015). In the remainder of this paper, it is assumed that for an $m$-dimensional input signal, $\mathbf{x}(t)$, the $m$-dimensional SFs, $\mathbf{s}(t)$ are available.

**GPs**: GP is a *non-parametric, probabilistic* approach to model the relationship between inputs $\mathbf{y} \in \mathbb{R}^y$ and outputs $z \in \mathbb{R}^z$. The use of GPs is sophisticated and reader is recommended to review Rasmussen (2003). The training set in GPs contains $N$ samples of outputs and inputs $\mathcal{D} \equiv \{\mathbf{z}, \mathbf{Y}\}$, which are related as follows:

$$z^i = f(\mathbf{y}^i) + \epsilon^i, \tag{9}$$

where $f$ is the underlying process model and $\epsilon^i \sim \mathcal{N}(0, \sigma^2)$ is a zero-mean Gaussian noise with variance $\sigma^2$. A GP is defined by a mean and a covariance function, such that

$$\mathbf{f}(\mathbf{Y}) \sim \mathcal{GP}(\boldsymbol{\mu}(\mathbf{Y}), \boldsymbol{\Sigma}(\mathbf{Y}, \mathbf{Y})). \tag{10}$$

The mean and covariance matrices are defined as $\boldsymbol{\mu}_\theta(\mathbf{Y})$ and $\boldsymbol{\Sigma}_\theta(\mathbf{Y}, \mathbf{Y})$, where $\theta \in \mathbb{R}^{n_\theta}$ represents hyperparameters. A common choice for $\Sigma$ is the Gaussian-RBF kernel:

$$\Sigma_\theta(\mathbf{y}^i, \mathbf{y}^j) = \beta \exp(-\frac{||y^i - y^j||_2^2}{2\alpha^2}). \tag{11}$$

The Gaussian-RBF kernel assigns a higher correlation if the inputs in the set $\{\mathbf{y}^i, \mathbf{y}^j\}$ are "close" in the sense of Euclidean distance. In (11), the hyperparameters $\theta$ include $\beta$ (length scale) and $\alpha$ (signal variance), and together they affect the "smoothness" of the function $f$. Given $\mathcal{D}$, the modelling objective is to learn both hyperparameters and any other unknown model parameters, denoted by the set $\Gamma$. The learning is accomplished by maximizing the following likelihood function:

$$p(\mathbf{z}|\mathbf{Y}) = \mathcal{N}(\mathbf{0}_N, \boldsymbol{\Sigma}_\theta(\mathbf{Y}, \mathbf{Y}) + \sigma^2 \mathbf{I}_N). \tag{12}$$

The unknown parameters $\gamma \equiv \{\theta, \sigma^2\} \in \Gamma$ can now be estimated by solving the following optimization problem:

$$\gamma^* \in \arg\max_{\gamma \in \Gamma} \log p(\mathbf{z}|\mathbf{Y}), \tag{13}$$

where $\gamma^* \in \Gamma$ is an optimal estimate, and

$$\log p(\mathbf{z}|\mathbf{Y}) = -\frac{1}{2}\mathbf{z}^{\mathrm{T}} \boldsymbol{\Sigma}_\gamma^{-1} \mathbf{z} - \frac{1}{2} \log |\boldsymbol{\Sigma}_\gamma| - \frac{N}{2} \log 2\pi, \tag{14}$$

where $\boldsymbol{\Sigma}_\gamma \equiv \boldsymbol{\Sigma}_\theta + \sigma^2 \mathbf{I}_N$. This is a non-convex function with multiple local optima; therefore, gradient descent can be used, but the initializations must be chosen with care.

Once a GP is trained, $\hat{z}$ is predicted by first constructing a joint density of all the training output set $\mathbf{z}$, and the test GP output $f(\hat{\mathbf{Y}})$ conditioned on both training input set $\mathbf{Y}$ and test input $\hat{\mathbf{Y}}$. This joint density is given by

$$p(\mathbf{z}, f(\hat{\mathbf{Y}})|\mathbf{Y}, \hat{\mathbf{Y}}) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \boldsymbol{\Sigma}_\gamma(\mathbf{Y}, \mathbf{Y}) & \boldsymbol{\Sigma}_\theta(\mathbf{Y}, \hat{\mathbf{Y}}) \\ \boldsymbol{\Sigma}_\theta(\hat{\mathbf{Y}}, \mathbf{Y}) & \Sigma_\theta(\hat{\mathbf{Y}}, \hat{\mathbf{Y}}) \end{bmatrix}\right), \tag{15}$$

where $\boldsymbol{\Sigma}_\gamma \equiv \boldsymbol{\Sigma}_\theta(\mathbf{Y}, \mathbf{Y}) + \sigma^2 \mathbf{I}_N$. Then a posterior distribution over $f(\hat{\mathbf{Y}})$ can be computed as:

$$p(f(\hat{\mathbf{Y}})|\mathcal{D}, \hat{\mathbf{Y}}) = \mathcal{N}(\hat{\mu}_\theta, \hat{\Sigma}_\theta), \tag{16}$$

and the mean and covariance matrices are computed as:

$$\hat{\mu}_\theta = \boldsymbol{\Sigma}_\theta(\hat{\mathbf{Y}}, \mathbf{Y})[\boldsymbol{\Sigma}_\gamma(\mathbf{Y}, \mathbf{Y})]^{-1}\mathbf{Y}, \tag{17}$$

$$\hat{\Sigma}_\theta = \Sigma_\theta(\hat{\mathbf{Y}}, \hat{\mathbf{Y}}) - \boldsymbol{\Sigma}_\theta(\hat{\mathbf{Y}}, \mathbf{Y})[\boldsymbol{\Sigma}_\gamma(\mathbf{Y}, \mathbf{Y})]^{-1}\boldsymbol{\Sigma}_\theta(\mathbf{Y}, \hat{\mathbf{Y}}). \tag{18}$$

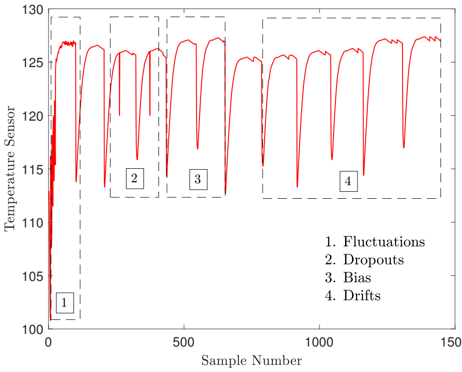

Fig. 1. Discrete and continuous sensor fault classification

Finally, the posterior for $\hat{z}$ can be predicted as follows:

$$p(\hat{\mathbf{z}}|\mathcal{D}, \hat{\mathbf{Y}}) = \mathcal{N}(\hat{\mu}_\theta, \hat{\Sigma}_\theta + \sigma^2). \tag{19}$$

For a single test input $\hat{\mathbf{y}}$, the GP prediction gives a distribution of outputs. However, for process control and monitoring, the point-estimate value $\hat{\mathbf{z}} = \hat{\mu}$ is preferred.

## 3. SENSOR FAULT CLASSIFICATION

A typical process sensor in biomanufacturing is prone to several types of faults. If a sensor reading is represented as $x_t + \epsilon_t$, where $x_t$ is the actual sensor reading and $\epsilon_t \in \mathbb{R}$ is the fault then we classify a sensor fault as follows:

(1) **Continuous:** After a certain time, the sensor returns constantly inaccurate readings, and it is possible to observe a pattern in sensor error in the form:

$$\epsilon_t = f_\alpha(t), \tag{20}$$

where $f_\alpha : \mathcal{T} \to \mathbb{R}$ is a continuous error-function parameterized by $\alpha \in \mathbb{R}^{n_\alpha}$. Based on $f_\alpha$, a continuous fault can be further classified into:

(a) **Bias:** The error-function is a constant, such that

$$f_\alpha(t) = \beta, \tag{21}$$

where $\beta \in \mathbb{R}$ is an arbitrary constant. $\beta$ can be a positive or a negative offset.

(b) **Drifts:** The error-function follows a slow-change, such as a polynomial change

$$f_\alpha(t) = \alpha_1 \epsilon_{t-1} + \alpha_2 \epsilon_{t-2} + \cdots + \alpha_n \epsilon_{t-n}, \tag{22}$$

where $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ are some constants.

(2) **Discrete:** Fault occurs at discrete time points, such that $\epsilon_t \neq 0$ for some $t \in \mathcal{T}$. Discrete faults can be further classified as follows:

(a) **Dropouts:** Sensor fails to record measurements at random time points, such that

$$\epsilon_t = \begin{cases} a_t & \text{for some } t \in \mathcal{T} \\ 0 & \text{otherwise} \end{cases}, \tag{23}$$

where $-x_t \leq a_t < 0$. For $a_t = -x_t$, sensor readings are not available.

(b) **Fluctuations:** Sensor exhibits random fluctuations around the expected sensor reading. Random fluctuations can be represented using probability densities. For example, $\epsilon_t \sim \mathcal{N}(\mu, \sigma)$, where $\epsilon_t$ follows a Gaussian distribution with mean and variance $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}_+$, respectively.

Figure 1 illustrates common discrete and continuous faults for a temperature sensor. Calibration errors are probably a key source of many continuous sensor faults since they generally manifest themselves as a bias or drift throughout the lifetime of the sensor node (Baljak et al., 2012). In practice, continuous sensor faults can be mitigated by recalibrating the sensor. For example, in Figure 1, the sensor bias subdues after a one-point calibration at $t = 600$.

In contrast, sensor malfunction is often the primary source of many of the discrete sensor faults. Note that fluctuations in sensor readings may also emanate as a result of process upsets or loss of controller performance; however, in this study, it is assumed that the process is well-controlled and any fault observed in the system – discrete or continuous – is a sensor-related fault. Further, $\epsilon_t$ or $f_\alpha$ are assumed to be unknown or unmeasured; however, it is assumed that the faults can be observed in readings through the effect they produce in the data. Given sensor data, the objective is to perform diagnostics and prognostics of sensor-health.

## 4. SENSOR MONITORING AND DIAGNOSIS

Unlike in process monitoring, where a fault in one variable affects other variables, in sensor-health monitoring, faults emanating in multiple sensors are often isolated and uncorrelated. For example, adjusting the bioreactor pressure might have an influence on the bioreactor temperature; however, it is unlikely that a calibration error in the pressure sensor would result in a calibration error in the temperature sensor. From a sensor-health monitoring perspective, this implies that the health of each sensor can be independently monitored. In this paper, we focus on univariate sensor-heath monitoring; however, in situations, where the health of a sensor is known to be correlated with other sensors in the network, the proposed approach can be extended to multivariate sensor-health monitoring.

In this paper, an SFA-based framework is proposed for real-time monitoring and diagnosis of the health of a multitude of sensors deployed in a biomanufacturing network. As discussed earlier, the choice of an SFA-based platform is motivated by the fact that sensor faults exhibit different dynamics for different fault-types. For example, continuous faults, such as bias and drifts, are slow-varying faults that cause gradual degradation of the sensor performance. In contrast, discrete faults, such as dropouts and fluctuations, exhibit faster dynamics and have an immediate effect on sensor performance. Informally, in the proposed framework, SFA is first applied to an incoming sensor signal projecting the raw signal onto multiple slow features, which are then trended on SFA-derived control charts for real-time monitoring and diagnosis of sensor-health.

For a given univariate sensor, $x(t) \in \mathbb{R}$, SFs can be readily computed (see Section 2); however, note that the SFs in (6) are calculated independent of past sensor recordings. In practice, underlying features should be related to sensor readings collected over a period of time due to the inherent sensor dynamics. To embed the dynamics in the feature space, SFA is extended to a dynamic SFA by appending $k$-lagged measurements, such that (Shang et al., 2015):

$$\mathbf{x}(t) := [x(t), x(t-1), \ldots, x(t-k)]^{\mathrm{T}} \in \mathbb{R}^k. \tag{24}$$

In (24), including $k$-lagged measurements converts a univariate sensor signal, $x(t)$ into a $k$-dimensional signal. Performing SFA on (24) yields $k$ slow features that are not only independent but also depend on the past sensor readings.

Based on the slowness value of each $s_i$, as indicated by $\omega_i$, the first $m_c$ ($m_c < k$) SFs are defined as dominant features for continuous faults and the last $m_d$ ($m_c + m_d \leq k$) SFs are the dominant features for discrete faults. Mathematically, if $\mathbf{s}(t) = [s_1(t), \ldots, s_k(t)]^{\mathrm{T}} \in \mathbb{R}^k$ denotes $k$ features computed by SFA then $\mathbf{s}_c(t) = [s_1(t), \ldots, s_{m_c}(t)]^{\mathrm{T}} \in \mathbb{R}^{m_c}$ and $\mathbf{s}_d(t) = [s_{k-m_d-1}(t), \ldots, s_k(t)]^{\mathrm{T}} \in \mathbb{R}^{m_d}$ denote dominant features for continuous and discrete faults, respectively. Physically, $\mathbf{s}_c$ models the slow-varying continuous faults or other incipient faults in $\mathbf{x}(t)$, while $\mathbf{s}_d$ captures fast-moving discrete faults and other random sensor noise.

Based on the SFA-based sensor model, monitoring indices can be established for sensor-health monitoring. Note that the dominant features, $\mathbf{s}_c$ and $\mathbf{s}_d$ form a partial orthogonal decomposition of a signal $\mathbf{x}$. To track the changes in continuous and discrete faults in the sensor, we apply the Hotelling's $T^2$ statistic to both $\mathbf{s}_c$ and $\mathbf{s}_d$, such that

$$T_c^2(t) = \mathbf{s}_c^{\mathrm{T}}(t)\mathbf{s}_c(t), \tag{25a}$$

$$T_d^2(t) = \mathbf{s}_d^{\mathrm{T}}(t)\mathbf{s}_d(t), \tag{25b}$$

where $T_c^2$ and $T_d^2$ are the Hotelling's statistics for $\mathbf{s}_c$ and $\mathbf{s}_d$, respectively. Assuming that $\mathbf{s}$ is (7) is a vector of independently Gaussian distributed random variables, then $T_c^2$ and $T_d^2$ both follow a $\chi^2$ distribution with $m_c$ and $m_d$ degrees of freedom, such that (Shang et al., 2015)

$$T_c^2(t) = \mathbf{s}_c^{\mathrm{T}}(t)\mathbf{s}_c(t) \sim \chi^2_{m_c}, \tag{26a}$$

$$T_d^2(t) = \mathbf{s}_d^{\mathrm{T}}(t)\mathbf{s}_d(t) \sim \chi^2_{m_d}. \tag{26b}$$

The $T_c^2$ and $T_d^2$ statistics in (26a) and (26b) measure the variations inside the subspace spanned by the dominant SFs for continuous and discrete faults, respectively. To use (26a) and (26b) for sensor-health monitoring, control limits are estimated using routine data from a healthy sensor. With $(1-\alpha)$ confidence level, the monitoring policy can be summarized as follows: (a) if $T_c^2 > \chi^2_{m_c,\alpha}$ then potential continuous faults are detected; and (b) if $T_d^2 > \chi^2_{m_d,\alpha}$ then potential discrete faults are detected. Note that these conditions not only provide a framework for real-time monitoring of sensor-health, but also a framework for fault diagnosis. This is possible because $T_c^2$ and $T_d^2$ individually capture continuous and discrete fault-types.

## 5. SENSOR HEALTH PROGNOSTICS

As discussed earlier, a prognostics algorithm predicts the future reliability of a sensor considering current and past sensor-health information. As a second step in PHM, a prognostic model relies on a *condition-monitoring* signal from the fault detection program that relates to the RUL. In this paper, we consider $T_c^2(t)$ and $T_d^2(t)$ in (25a) and (25b) as the condition-monitoring signals for continuous and discrete faults. Given signals $T_c^2(1:t) = \{T_c^2(1), \ldots, T_c^2(t)\}$ and $T_d^2(1:t) = \{T_d^2(1), \ldots, T_d^2(t)\}$, the goal is to find $t_f$, such that $T_c^2(t+t_f) > \chi^2_{m_c,\alpha}$ and $T_d^2(t+t_f) > \chi^2_{m_d,\alpha}$.

We propose a GP-based prognostics model to forecast the RUL of a sensor. Since both continuous and discrete faults affect the RUL, we use GP models to forecast the future values of $T_c^2(t)$ and $T_d^2(t)$ based on the current and past sensor-heath information. Mathematically, it is done as follows: Let $z_{c,t+p_c} = T_c^2(t + p_c)$ and $\mathbf{y}_{c,t} = [T_c^2(t), \ldots, T_c^2(t - q_c)]$, where $p_c, q_c \in \mathbb{N}$ are the prediction horizon and the Markov order, respectively, then we have

$$z_{c,t+p_c} = f_c(\mathbf{y}_{c,t}) + \epsilon_{c,t}, \tag{27}$$

where $f_c$ is a GP, given in (10). Note that (27) provides a $p_c$-step ahead prediction of $T_c^2$ based on its past $q_c$ values. Similarly, for $z_{d,t+p_d} = T_d^2(t + p_d)$ and $\mathbf{y}_{d,t} = [T_d^2(t), \ldots, T_d^2(t - q_d)]$, where $p_d, q_d \in \mathbb{N}$ are the prediction horizon and the Markov order, respectively, then we have

$$z_{d,t+p_d} = f_d(\mathbf{y}_{d,t}) + \epsilon_{d,t}, \tag{28}$$

where $f_d$ is a GP, given in (10). The prognostics models in (27) and (28) can be trained as discussed in Section 2.

## 6. INDUSTRIAL CASE STUDY

Steam-in-place (SIP) is a critical part of biomanufacturing. SIP is carried out before processing or between two batches to ensure process sanitization and prevent carryover of product or unwanted impurities. Failure to appropriately sanitize the cell culture equipment could lead to contamination resulting in the loss of the batch. SIP involves adding steam to elevate the temperature of the equipment, then maintaining a minimum temperature for a specified time, followed by equipment cool-down (Roy et al., 2014).

While temperature sensors play a critical role in maintaining the equipment at the prescribed temperature, they are also prone to fluctuations and malfunction due to high temperature and pressure environment inside the bioreactor. The current maintenance strategy for the temperature sensor in SIP is TbM, wherein the sensor is routinely calibrated or changed out after some predetermined usage cycles. In addition, a maintenance work order may also be submitted if there are any unusual treads visible in the sensor readings. Figure 2(a) shows the temperature data collected over 20 SIP cycles. Each SIP cycle is approximately 100 samples in length that include elevating the temperature of the equipment to 125° C, maintaining it for around 80 samples, and then cooling down the equipment. Visually, the sensor exhibits fluctuations in several SIP cycles during the heating phase.

We demonstrate the efficacy of the proposed framework in real-time health monitoring, diagnostics, and prognostics of a temperature sensor. Using data from a healthy sensor, first a linear SFA model is trained for $k = 100$ (based on trial-and-error). Next, using the model, the Hotelling's $T^2$ statistics are computed for Figure 2(a). Figures 2(c) and (d) show the $T^2_{c,\alpha}$ and $T^2_{d,\alpha}$ statistics, respectively, computed for $m_c = m_d = 2$ dominant SFs. This selection is based on the history for the sensor, where slow drifts and random high-frequency fluctuations have been a common issue. As discussed earlier, Figure 2(b) monitors continuous faults, and Figure 2(c) monitors discrete faults.

From Figure 2(c) and (d) it is clear that the sensor exhibits no critical faults (discrete or continuous) in the first 10 SIP cycles. For SIP cycles $11 - 13$, we have $T^2_{d,\alpha} > \chi^2_{m_d,\alpha}$. This indicates a potential discrete fault in the sensor. Note that this coincides with the fluctuations observed in SIP cycles $12 - 13$ shown in Figure 2(a). In addition, there is also a potential continuous fault during SIP cycles $12 - 13$, as indicated by $T^2_{c,\alpha} > \chi^2_{m_c,\alpha}$ in Figure 2(b). Similarly, for SIP cycles $18 - 21$, the sensor exhibits both discrete and continuous faults. Compared to SIP cycles $12 - 13$, the magnitude of fault in cycles $18 - 21$ is much more pronounced. Further, the growing $T^2_{d,\alpha}$ values in SIP cycles $18 - 21$ indicate that the discrete fault is increasing in magnitude. In summary, Figure 2 provides a practical approach for monitoring and diagnoses of sensor-health.

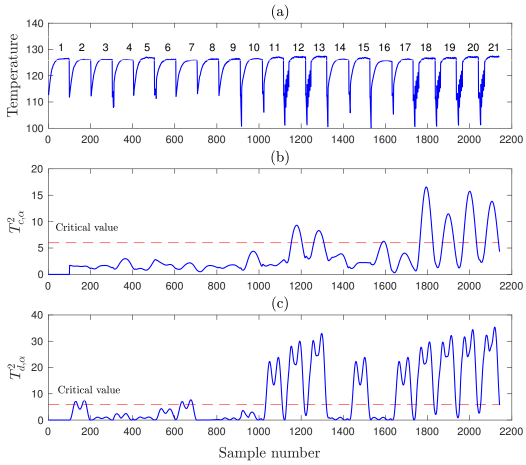

Fig. 2. SFA-based health monitoring of temperature sensor

Next, we train GP-based prognostics models to forecast the condition-monitoring signals for continuous and discrete faults. As discussed earlier, we forecast the future values of $T^2_{d,\alpha}$ and $T^2_{c,\alpha}$ based on the past health information available for the sensor. The GP prognostics model for continuous faults is trained for $q_c = 10$ and $p_d = 10$ samples. Similarly, the GP model for discrete faults is also trained for $q_d = 10$ and $p_d = 10$ samples. In other words, both the GP models use past 10 samples of $T^2_{d,\alpha}$ and $T^2_{c,\alpha}$ to provide a 10 sample ahead predictions of $T^2_{d,\alpha}$ and $T^2_{c,\alpha}$, respectively. Figures 3 and 4 provide a 10 step ahead forecast of $T^2_{c,\alpha}$ and $T^2_{d,\alpha}$, respectively. For reference, the actual values, represented by the black line, are also shown in Figures 3 and 4. Overall, the forecast from both the prognostics models are accurate. 95% confidence intervals (CIs) around the forecast are also shown in Figures 3 and 4 which further validates the accuracy of the prediction. The 10-step ahead root-mean square error (RMSE) for the continuous fault prognostics model in Figure 4 is 1.12, while the RMSE for discrete fault prognostics model is 29.20. A higher RMSE value for the continuous fault is expected, since the dominant SFs for discrete faults include faster signal dynamics, including the sensor noise. Nevertheless, the prognostics models provide a reliable 10 step ahead forecast for the RUL and also provide the probability of failure, in terms of CIs. Finally, note that these results can directly feed into the CbM program to take maintenance related decisions.

## 7. CONCLUSIONS

A commercial bioreactor is regularly fitted with a multitude of critical sensors that work in tandem to ensure consistent product quality and overall process safety. While these sensors generate critical process information and

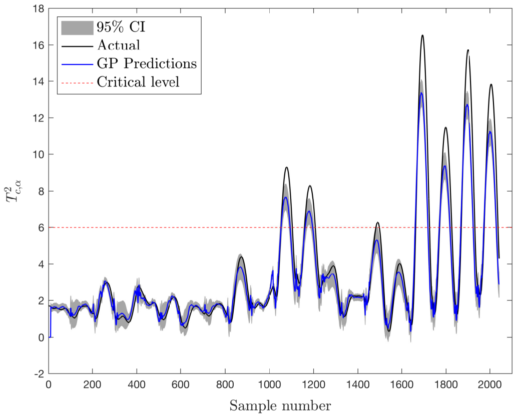

Fig. 3. Prognostics model for $T^2_{c,\alpha}$ for $p_c = 10$ and $q_c = 10$.

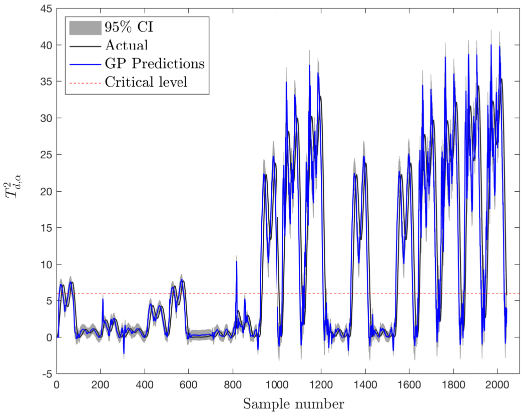

Fig. 4. Prognostics model for $T^2_{d,\alpha}$ for $p_d = 10$ and $q_d = 10$.

insights, they are also prone to frequent failures and malfunctions. In this paper, we proposed a novel framework for sensor-health management. The framework provides real-time health monitoring, diagnosis, and prognosis of critical sensors used in biomanufacturing. The proposed method is based on SFA, which separates continuous and discrete faults in a sensor. A robust GP-based prognostics model is also proposed to forecast the future reliability of the sensor. The efficacy of the proposed method was demonstrated in an industrial temperature sensor.

## REFERENCES

Baljak, V., Tei, K., and Honiden, S. (2012). Classification of faults in sensor readings with statistical pattern recognition. In *The Sixth International Conference on Sensor Technologies and Applications*, 270–276.

Fugate, M.L., Sohn, H., and Farrar, C.R. (2001). Vibration-based damage detection using statistical process control. *Mechanical Systems and Signal Processing*, 15(4), 707–721.

Jardine, A.K., Lin, D., and Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, 20(7), 1483–1510.

Ji, H., He, X., Shang, J., and Zhou, D. (2018). Exponential smoothing reconstruction approach for incipient fault isolation. *Industrial & Engineering Chemistry Research*, 57(18), 6353–6363.

Kacprzynski, G.J., Gumina, M., Roemer, M.J., Caguiat, D.E., Galie, T.R., and McGroarty, J.J. (2001). A prognostic modeling approach for predicting recurring maintenance for shipboard propulsion systems. In *ASME Turbo Expo 2001: Power for Land, Sea, and Air*. American Society of Mechanical Engineers Digital Collection.

Li, Y., Billington, S., Zhang, C., Kurfess, T., Danyluk, S., and Liang, S. (1999). Adaptive prognostics for rolling element bearing condition. *Mechanical systems and signal processing*, 13(1), 103–113.

Ma, J. and Li, J.C. (1995). Detection of localised defects in rolling element bearings via composite hypothesis test. *Mechanical Systems and Signal Processing*, 9(1), 63–75.

Oppenheimer, C.H. and Loparo, K.A. (2002). Physically based diagnosis and prognosis of cracked rotor shafts. In *Component and Systems Diagnostics, Prognostics, and Health Management II*, volume 4733, 122–132. International Society for Optics and Photonics.

Pecht, M.G. (2010). A prognostics and health management roadmap for information and electronics-rich systems. *IEICE ESS Fundamentals Review*, 3(4), 4_25–4_32.

Peng, Y., Dong, M., and Zuo, M.J. (2010). Current status of machine prognostics in condition-based maintenance: a review. *The International Journal of Advanced Manufacturing Technology*, 50(1-4), 297–313.

Prajapati, A., Bechtel, J., and Ganesan, S. (2012). Condition based maintenance: a survey. *Journal of Quality in Maintenance Engineering*, 18(4), 384–400.

Rasmussen, C.E. (2003). Gaussian processes in machine learning. In *Summer School on Machine Learning*, 63–71. Springer.

Roy, K., Undey, C., Mistretta, T., Naugle, G., and Sodhi, M. (2014). Multivariate statistical monitoring as applied to clean-in-place (cip) and steam-in-place (sip) operations in biopharmaceutical manufacturing. *Biotechnology progress*, 30(2), 505–515.

Shang, C., Yang, F., Gao, X., Huang, X., Suykens, J.A., and Huang, D. (2015). Concurrent monitoring of operating condition deviations and process dynamics anomalies with slow feature analysis. *AIChE Journal*, 61(11), 3666–3682.

Shin, J.H. and Jun, H.B. (2015). On condition based maintenance policy. *Journal of Computational Design and Engineering*, 2(2), 119–127.

Tsui, K.L., Chen, N., Zhou, Q., Hai, Y., and Wang, W. (2015). Prognostics and health management: A review on data driven approaches. *Mathematical Problems in Engineering*, 2015.

Wang, D., Tsui, K.L., and Miao, Q. (2017). Prognostics and health management: A review of vibration based bearing and gear health indicators. *IEEE Access*, 6, 665–676.

Wiskott, L. and Sejnowski, T.J. (2002). Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4), 715–770.

Yam, R., Tse, P., Li, L., and Tu, P. (2001). Intelligent predictive decision support system for condition-based maintenance. *The International Journal of Advanced Manufacturing Technology*, 17(5), 383–391.

Yan, J., Koc, M., and Lee, J. (2004). A prognostic algorithm for machine performance assessment and its application. *Production Planning & Control*, 15(8), 796–801.