Sensor Calibration and Diagnostics Under Parameter Uncertainty: A Smart Building Application [⋆]

Gregory Provan and Seamus Ó Buaudacháin [*]

[*] *Department of Computer Science, University College Cork, Cork, Ireland*
(Tel: 353-21-490-1816; e-mail: g.provan, sdob@ cs.ucc.ie).

Abstract: One key problem with control systems in general, and control systems for buildings in particular, is that mis-calibration of sensors/actuators is commonplace and causes significant problems, such as suboptimal performance and diagnostics false alarms. This paper describes a methodology for calibrating sensors that can reduce these problems. We show how we can take sensor outputs and continuously calibrate them by applying expectation-maximization (EM) learning and recent gossip-based algorithms. We apply our approach to the domain of sustainable buildings, in particular temperature sensors in shared zones in a large commercial building. We empirically show that our approach can correctly either diagnose faults that render sensors impossible to calibrate, or can perform appropriate calibration.

1. INTRODUCTION

In many building applications, the building that is constructed differs from the design models. This discrepancy between the as-designed and actual performance of buildings (Elkhuizen et al., 2006; Piette et al., 1994) arises due to a variety of reasons, such as improper system design and installation, inadequate commissioning and maintenance practices, and inadequate use of operational performance data (Piette et al., 2001). A variety of studies have shown that sub-optimal building performance occurs widely due to this design/usage disconnect (Piette et al., 1994). For example, Piette et al. (2001) reports that, in the process of commissioning new buildings, 81% of the building owners surveyed encountered problems with new heating and air conditioning systems; Piette et al. (1994) reports that, in a survey of 60 buildings, 50% had controls problems, 40% had HVAC equipment problems, 15% had missing equipment, and 25% had energy management control systems (EMCS), economizers, and/or variable speed drives that were not functioning properly.

When a building is first set up, the *commissioning* process consists of using BMS test signals to define the operational parameters of individual components and the system as a whole, in order to create a well-understood (and hopefully optimal) operating condition. After the commissioning process, the BMS uses a monitoring/diagnostics process to identify anomalous operating conditions, by assessing system control/sensor data to identify component/system state without altering operating conditions (House and Kelly, 1999). However, the development and application of building monitoring/diagnostics has been slower and lower-tech than that in other industries, e.g., process control or aerospace, because the industry is cost-sensitive, the benefits are difficult to quantify, and design has focused on user comfort (but not human safety). The focus on low-cost solutions has meant that low-accuracy state estimation and fault isolation has been adequate to date (Braun, 2007).

One key issue in the as-designed vs. as-operated discrepancy is that important building parameters are not precisely known. This results in false alarms and poor isolation accuracy during monitoring and diagnostics, since existing BMS rule-based systems typically use thresholds for determining faults, and these thresholds are rarely updated after installation.

We assume that we have a system model Φ with parameters $\boldsymbol{\theta}$. FDI approaches typically assume that $\boldsymbol{\theta}$ is known in the nominal system state. With such an assumption, fault detection concerns deciding whether the vector θ has changed, and isolation concerns identifying which components have caused the observed anomaly in $\boldsymbol{\theta}$. In our case, we assume there is nontrivial uncertainty in $\boldsymbol{\theta}$; under these assumptions, FDI is not a straightforward process.

This article describes an approach that can be applied to (1) learning accurately calibrated "nominal"' sensor outputs, and to (2) reducing false alarms from inaccurate monitoring and diagnostics isolation. We apply the expectation maximization (EM) algorithm (Dempster et al., 1977) to learn nominal parameter values, and we then extend this approach to residual generation for fault detection and isolation (FDI).

This approach can be used throughout the lifetime of the building as a means for improving the "continuous commissioning" process. The practice in the buildings industry of "commissioning" a building once is changing, as it is now acknowledged that "continuous commissioning" is a key ingredient for cost-effective building operations. A key part of continuous commissioning is matching the existing building parameters to the embedded code that monitors the building. From example, this includes parameter estimation, in order to continuously tune system parameters.

In this article, we show how we can continuously calibrate sensor outputs by applying expectation-maximization (EM) learning and sensor parameter-sharing, via "gossip-based" algorithms. We apply our approach to temperature sensors in shared zones, e.g., as in a large commercial building. We em-

[⋆] Supported by SFI grant 06/SRC/I1091.

pirically show that our approach can correctly either diagnose faults that render sensors impossible to calibrate, or can perform appropriate calibration. This approach is general, and can be applied to any suite of sensors and actuators.

Our contributions are as follows:

- We apply EM and a recent gossip-based (Ramakrishnan et al., 2011) algorithm to continuously calibrate sensors;
- We describe a model-driven method for gossip-based calibration to improve the computational efficiency of the approach;
- We apply our approach to the domain of sustainable buildings, in particular addressing temperature monitoring (via multiple sensors) in a large commercial building;
- We empirically show how our approach can improve overall FDI accuracy, given that the system parameters $\boldsymbol{\theta}$ are continuously updated and inoperable sensors are accurately diagnosed.

2. APPLICATION DOMAIN

For our application domain, we focus on the task of temperature monitoring in an indoor environment, such as a multi-user open space office. We model a room in which a suite of sensors are placed at random (but unique) positions. We assume that the underlying base temperature of the environment is affected by some regularly varying heat source (for example, solar radiation), while several stationary hotspots (which might be computers) are distributed throughout the zone.

We assume that sensors are not initially calibrated, i.e., that each sensor is subject to a certain level of calibration offset error. These calibration offsets are randomly generated from a uniform distribution over $[-5.625, 5.625]$. We consider these minor offsets to be qualitatively different from calibration offset *faults*, since they are effectively mitigated during the continuous calibration process. Further, we allow sensor calibration and fault status to dynamically change over time; the algorithm we will present will mitigate the effects of these dynamics.

We aim to track changes in temperature over a 24-hour period in an office-like environment: the baseline temperature varies sinusoidally between $20°C$ and $25°C$, while we model hotspots using 20 2-dimensional Gaussians whose means are drawn uniformly from $\mu \in [20, 50]$ and standard deviations drawn uniformly from $\sigma \in [4, 16]$. Because our simulated nodes measure their environment once every 30 seconds, we divide the total simulation time into 2880 discrete epochs ($2880 = 24 \times 60 \times 2$).

We assume that the sensor nodes can be in a nominal state, or one of two fault states. A "stuck-at" fault causes a node to return a constant value of either $\pm 40°$ (chosen randomly). A "calibration offset fault" causes a node to track the underlying phenomenon at an offset of $\pm 40°$ (chosen randomly).

Our application has several novel aspects. One novelty is the tracking of temperatures that vary throughout the zone; in contrast, most other papers, e.g., Ramakrishnan et al. (2011), assume a fixed temperature across the zone. Second, we will diagnose fault states as well as calibrate those sensors with calibration offset faults.

Our fault-injection methodology is as follows. We generate a set of 100 fault events. Of these, 50 are 'stuck-at' faults, and the remaining half are 'calibration offset faults'. Every 28 epochs ($28 \approx 2880/100$) we take one fault event from the set and apply it to a currently fault-free node. We assume that faults are static and uncorrected; in other words, after a sensor begins to exhibit faulty behaviour, it continues to exhibit that behaviour until the end of the simulation.

3. NOTATION

This section describes our notation. We assume that we have a non-linear system that has a nonlinear state-space model structure given by:

$$\boldsymbol{x}_{t+1} = f_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\theta}) + \epsilon_t, \tag{1}$$

$$\boldsymbol{y}_t = g_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\eta}_t, \boldsymbol{\theta}) + \delta_t, \tag{2}$$

$$\boldsymbol{u}_t = h_t(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{\theta}) + \zeta_t. \tag{3}$$

Here, $\boldsymbol{x} \in \Re^{n_x}$ denotes the state variable, with $\boldsymbol{y} \in \Re^{n_y}$ and $\boldsymbol{u} \in \Re^{n_u}$ denoting (respectively) observed output and input responses. $\boldsymbol{\eta}_t$ denotes a discrete-valued fault variable denoting the fault status of a sensor generating the output. Furthermore, $\boldsymbol{\theta}$ is a vector of (unknown) parameters that specifies the mappings $f_t(\cdot)$, $g_t(\cdot)$ and $h_t(\cdot)$, which may be nonlinear and time-varying. Finally, ϵ_t, δ_t and z_t represent mutually independent vector i.i.d. processes described by probability density functions (pdfs) $Pr_\epsilon(\cdot)$, $Pr_\delta(\cdot)$ and $Pr_z(\cdot)$, respectively. These are assumed to be of known form (Gaussian) but parameterized (e.g., mean and variance) by values that can be absorbed into $\boldsymbol{\theta}$ for estimation if they are unknown.

Throughout this paper, lower-case bold variables represent vectors, uppercase bold variables represent matrices, and non-bold lower-case variables are scalars.

Due to the random components ϵ_t and δ_t, the model (equation 1) can also be represented via the stochastic description

$$\boldsymbol{x}_{t+1} \sim Pr_\theta(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t), \tag{4}$$

$$\boldsymbol{y}_t \sim Pr_\theta(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\eta}_t), \tag{5}$$

where $Pr_{\boldsymbol{\theta}}(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)$ is the pdf describing the dynamics for given values of $\boldsymbol{x}_t$ and $\boldsymbol{\theta}$, and $Pr_\theta(\boldsymbol{y}_t|\boldsymbol{x}_t)$ is the pdf describing the measurements.

Given the system, we aim to measure the value of x_t over time using a sensor network with n nodes. Our objective is to use the sensor network to minimize the objective function (derived from (Feng et al., 2003))

$$J = \frac{\sum_{i=1}^{n} |\tilde{y}_{i,t} - x_{i,t}|}{\sum_{i=1}^{n} |y_{i,t} - x_{i,t}|} \tag{6}$$

subject to the system model, where $\tilde{y}_{i,t}$ is the calibrated measurement of the i^{th} sensor at time t and $y_{i,t}$ is the uncalibrated measurement (where u is null). Note that our model allows the system state x_t to evolve without being controlled; what we are controlling is the sensor measurement process.

We will use the following notation for temporal histories: $\boldsymbol{Y}_{t:N}$ denotes the history $[\boldsymbol{y}_t, ..., \boldsymbol{y}_N]$. However, for brevity we may also denote $\boldsymbol{Y}_{1:N}$ simply as $\boldsymbol{Y}_N$, where we implicitly assume that the index starts at 1. We will call a window W the temporal history for a sensor that can be stored locally at the sensor node.

4. PARAMETER ESTIMATION

This section describes our method for continuous updating of system parameters. Having sensor values that reflect current building (temperature) conditions, rather than the as-designed conditions, is a key part of continuous commissioning. To achieve this, we update the "nominal" value of key system parameters with each subsequent measurement of that value.

We now show how we can compute an estimate $\hat{\boldsymbol{\theta}}$ of the parameter vector $\boldsymbol{\theta}$ based on N measurements $\boldsymbol{U}_N = [u_1, ..., u_N]$, $\boldsymbol{Y}_N = [\boldsymbol{y}_1, ..., \boldsymbol{y}_N]$ of observed system input/output responses.

4.1 Estimation Using EM

The expectation-maximization (EM) algorithm Dempster et al. (1977) is an iterative method for maximum likelihood parameter estimation. The EM algorithm alternates between two steps: (1) the *expectation* (E) step computes the expectation of the log-likelihood, using the current estimate for the parameters; and (2) the *maximization* (M) step computes the parameters that maximize the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

The key idea in EM is to define the joint likelihood function $L_{\boldsymbol{\theta}}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = logPr_\theta(\boldsymbol{X}_N, \boldsymbol{Y}_N)$, with respect to both the observations $\boldsymbol{Y}_N$ and the unknown state variables $\boldsymbol{X}_N$. Underlying this strategy is an assumption that maximizing the "complete" log-likelihood $L_{\boldsymbol{\theta}}(\boldsymbol{X}_N, \boldsymbol{Y}_N)$ is easier than maximising the incomplete one $L_{\boldsymbol{\theta}}(\boldsymbol{Y}_N)$. The EM algorithm then copes with X_N being unavailable by forming an approximation $Q(\theta, \theta_k)$ of $L_{\boldsymbol{\theta}}(\boldsymbol{X}_N, \boldsymbol{Y}_N)$. The approximation used is the minimum variance estimate of $L_{\boldsymbol{\theta}}(\boldsymbol{X}_N, \boldsymbol{Y}_N)$ given the observed available data $\boldsymbol{Y}_N$, and an assumption $\boldsymbol{\theta}_k$ of the true parameter value. This minimum variance estimate is given by the conditional mean:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k) \equiv E_{\boldsymbol{\theta}_k} L_{\boldsymbol{\theta}}(\boldsymbol{X}_N, \boldsymbol{Y}_N) | \boldsymbol{Y}_N). \tag{7}$$

Choosing θ so that $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k) > Q(\boldsymbol{\theta}_k, \boldsymbol{\theta}_k)$ implies that the log-likelihood is also increased in that $L_{\boldsymbol{\theta}}(\boldsymbol{Y}_N) > L_{\boldsymbol{\theta}_k}(\boldsymbol{Y}_N)$. The EM algorithm exploits this to deliver a sequence of values designed to be increasingly good approximations of the maximum-likelihood estimate (equation 7), also called the Most Probable Explanation (MPE), using the following algorithm.

Algorithm 1 EM algorithm

```
 1: procedure EM($\boldsymbol{Y}_N, \boldsymbol{\theta}$)                ▷ The MPE of $\boldsymbol{\theta}$
 2:     $k \leftarrow 0$
 3:     initialise $\boldsymbol{\theta}_k$ such that $L_{\boldsymbol{\theta}}(\boldsymbol{Y}_N)$ is finite
 4:     while not converged do
 5:         Expectation (E) step: Calculate $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$
 6:         (Maximisation (M) step: Compute
                $\boldsymbol{\theta}^{(t+1)} = \overset{argmax}{\boldsymbol{\theta}} \quad Pr(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$
 7:         $k \leftarrow k + 1$
 8:     end while
 9:     return $\hat{\boldsymbol{\theta}}$                         ▷ The MPE of $\boldsymbol{\theta}$
10: end procedure
```

The termination decision for convergence of the equation shown in line 6 of Algorithm 1 is performed using a standard criterion such as the relative increase of $L_{\boldsymbol{\theta}}(\boldsymbol{Y}_N)$ or the relative increase of $Q(\boldsymbol{\theta}_k, \boldsymbol{\theta}_k)$ falling below a pre-defined threshold.

4.2 Joint Estimation and Calibration

We consider the case where we want to simultaneously calibrate a set $\mathcal{S}$ of sensors and estimate the signal $\boldsymbol{s}$. For computational efficiency, we assume a linear model relating output values $\boldsymbol{y}$ and true signal $\boldsymbol{s}$. We further assume a Gaussian distribution for $Pr(\boldsymbol{y}; \boldsymbol{\theta}_{true})$, where $\boldsymbol{\theta}_{true}$ is the true parameter vector.

The EM algorithm has been extended (Kowalczyk and Vlassis, 2005) to incorporate a gossip-based protocol, i.e., a protocol in which sensors/actuators can communicate their values in order to achieve a joint MPE, rather than just a set of independent MPE values. We extend a recent gossip-based EM (Ramakrishnan et al., 2011) that enables us to simultaneously estimate sensor correction parameters for a collection $\mathcal{S}$ of sensors, as well as a hidden signal jointly estimated by $\mathcal{S}$.

Model Formulation We assume we have a set $\mathcal{S}$ of sensors that measure a signal, where we assume each sensor may have an individual offset, and hence will require individual calibration factors. We model this as follows:

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_N \end{bmatrix} = \boldsymbol{A}(\boldsymbol{\sigma}, \boldsymbol{\eta})\boldsymbol{s} + \boldsymbol{\delta} = \begin{bmatrix} A_1(\sigma_1, \eta_1) \\ A_2(\sigma_2, \eta_2) \\ \vdots \\ A_N(\sigma_N, \eta_N) \end{bmatrix} \boldsymbol{s} + \begin{bmatrix} \boldsymbol{\delta}_1 \\ \boldsymbol{\delta}_2 \\ \vdots \\ \boldsymbol{\delta}_N \end{bmatrix} \tag{8}$$

where $\boldsymbol{s} \in \Re^D$ is the signal to be estimated, and $\boldsymbol{\eta}$ denotes the diagnosis state. The vector $\boldsymbol{y}_n \in \Re^M$ represents the noisy observation vector at node n, where $\boldsymbol{\delta}_n \in \Re^M$ is additive Gaussian noise with zero mean and covariance $\sigma^2 I$.

The matrix for $\boldsymbol{A}_n$ corresponds to the measurement matrix

$$\boldsymbol{A}_n(\boldsymbol{\sigma}_n, \boldsymbol{\eta}_n) = \begin{bmatrix} a_{11}^n & \cdots & a_{1D}^n \\ \vdots & \ddots & \vdots \\ a_{M1}^n & \cdots & a_{MD}^n \end{bmatrix} \tag{9}$$

$\boldsymbol{A}_n$ is parameterized by a calibration parameter vector $\boldsymbol{\alpha}_n$ and $\boldsymbol{\eta}_n$. The signature $\boldsymbol{s}$, the diagnosis state $\boldsymbol{\eta}$, and the calibration vectors $\{\boldsymbol{\alpha}_n\}_{n=1}^N$ are the unknown parameters to be estimated.

Joint Maximum Likelihood Estimation This section describes a method for solving the joint maximum-likelihood estimation problem of simultaneously calibrating a set of sensors, estimating the signal and computing the diagnosis state. We use an extension of the EM algorithm to solve this joint estimation problem (Ramakrishnan et al., 2011). We alternate E and M steps, but adopt a gossip-based distributed method for the M step. The pseudocode for this is shown as Algorithm 2.

We extend an alternating optimization-based approach (Ramakrishnan et al., 2011). In the equivalent of the E step, this approach chooses a subset of the parameter space to minimize while keeping the other parameters fixed. Here, the parameter vector is $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2]$, where $\boldsymbol{\theta}_1 = \{\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_n\}$ and $\boldsymbol{\theta}_2 = \boldsymbol{s}, \boldsymbol{\eta}$. We assume that the observation random vector $\boldsymbol{Y}$ has a Gaussian probability density $Pr(\boldsymbol{y}; \boldsymbol{\theta}_{true})$.

Given an observation $\boldsymbol{Y} = \boldsymbol{y}$, the log-likelihood estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}_{true}$ is defined by $\hat{\boldsymbol{\theta}} = \overset{argmax}{\hat{\theta} \in \boldsymbol{\theta}} \quad log\ g(\boldsymbol{y}; \boldsymbol{\theta})$

Since the above joint maximum-likelihood optimization is non-convex, we use a distributed alternating optimization approach.

At the i^{th} iteration when we update $\boldsymbol{\theta}_1$, we first maximize the log-likelihood by fixing $\boldsymbol{\theta}_2^t$. For each of the sensor nodes, given $\hat{\mathbf{s}}^t$, the maximization over the corresponding $\boldsymbol{\alpha}_n$ decouples and hence can be solved locally.

$$\begin{aligned}\boldsymbol{\alpha}_n^{t+1} = \overset{argmin}{\boldsymbol{\alpha}_n} \; &\{(A_n(\boldsymbol{\alpha}_n)\boldsymbol{s}^{t+1})^T(A_n(\boldsymbol{\alpha}_n)\boldsymbol{s}^{t+1}) \\ &- 2(A_n(\boldsymbol{\alpha}_n)\boldsymbol{s}^{t+1})^T\boldsymbol{y}_n\} \end{aligned} \qquad (10)$$

Computing equation 10 can be done locally for each sensor, without communicating with its neighboring sensors. In the generalization of the M step, we employ alternating optimization,in which $\boldsymbol{\theta}_2 = \boldsymbol{s}$ needs to be updated for fixed $\boldsymbol{\theta}_1^t = \{\hat{\boldsymbol{\alpha}}_n^t\}_{n=1}^N$, i.e.,

$$\boldsymbol{s}^{t+1} = \overset{argmax}{l \leq \boldsymbol{\theta} \leq u} \left\{ \boldsymbol{s}^T\boldsymbol{A}^T(\hat{\boldsymbol{\alpha}}^t)\boldsymbol{A}(\hat{\boldsymbol{\alpha}}^t)\boldsymbol{s} - 2\boldsymbol{s}^T\boldsymbol{A}^T(\hat{\boldsymbol{\alpha}}^t)\boldsymbol{y} \right\}$$

5. SENSOR CALIBRATION/DIAGNOSIS

This section describes our calibration approach under parameter uncertainty. In the following, we assume that the health status of all sensors/actuators are independent. Assume that we obtain a sequence of τ sensor readings $\boldsymbol{Y}_\tau = \{\boldsymbol{y}_1, ..., \boldsymbol{y}_\tau\}$ from times 1 to τ, from which we want to compute a model that denotes the "nominal" sensor values.

The key idea behind this alternative approach is to use correlations among observations to cross-calibrate the sensors. In other words, given a set of sensors $\{S^1, ..., S^m\}$ generating an observation vector $\boldsymbol{y}_t = \{y_t^1, ..., y_t^m\}$, where sensor S^i generates $\boldsymbol{y}_t^i$, if all observations in $\boldsymbol{y}_t$ confirm a nominal hypothesis, this hypothesis is more likely; however, if only $\boldsymbol{y}_t^k$ of these multiple observations is abnormal, it is most likely that S^k is faulty and the other sensors are nominal. The challenge is to identify subsets of correlated observations, and to compute the joint estimates. In the following we assume spatial correlation of sensors, as described below. [1]

This section describes how we control the process of sensors performing fault-isolation and calibration. Our control algorithm (Algorithm 2) takes the following parameters:

- τ, a distance threshold that defines the greatest distance between two geo-spatially correlated sensors;
- c and d, threshold factors controlling the sensitivity of the diagnosis of stuck-at and calibration faults, respectively;
- W, a maximum window size;
- λ, the number of epochs between each invocation of distributed calibration; and
- ρ, the number of iterations of gossip during each calibration.

Once every epoch t, each node i adds its most recent measurement $y_{i,t}$ to its window (removing the oldest measurement if the window is full). After filling its window for the first time (at epoch W) each node estimates the distribution of noise affecting its readings as the standard deviation of its window, denoted by δ_i. Algorithm 2 shows the pseudo-code for the control algorithm, with inputs $d \in \mathbb{R}_{>0}$ and $\rho \in \mathbb{N}_1$.

After epoch W, each node compares the current standard deviation of its window, $\sigma[w_i]$, with its stored estimate of fault-free behaviour. We interpret an excessively small value of $\sigma[w_i]$ (relative to δ_i) as indicating that i is in an anomalous state.

Algorithm 2 Fault detection control algorithm

```
procedure DIAGNOSE/CALIBRATE($\tau, c, d; W, \lambda, \rho$)
    $\Phi \leftarrow \{i \rightarrow 0 \colon i \in S\}$                ▷ Calibration factors
    $\delta \leftarrow \{i \rightarrow \emptyset \colon i \in S\}$      ▷ Noise-distribution estimates
    $\eta \leftarrow \{i \rightarrow \emptyset \colon i \in S\}$        ▷ Diagnosed fault states
    for $t = 1$ do$k$
        $w_i \leftarrow y_{i,t}$
                                                      ▷ Estimate noise
        if $t = W$ then
            for $i \in S$ do
                $\delta_i \leftarrow \sigma[w_i]$
            end for
        end if
                                                      ▷ Detect stuck-at faults
        if $t > W$ then
            for $i \in S$ do
                if $\sigma[w_i] < \frac{\delta_i}{c}$ then
                    $F_i \leftarrow stuckAt$
                end if
            end for
        end if
                                                      ▷ Perform distributed calibration
        if $t \equiv 0 \pmod{\lambda}$ then
            $Y \leftarrow \{i \rightarrow \overline{w_i} + \Phi_i \colon i \in S\}$
            for $p \in S$ do
                $S' \leftarrow \{j \in S \colon \text{distance}(p, j) \leq \tau\}$
                if $S' \neq \emptyset$ then
                    for $u = 1$ do$\rho$
                        $q$ ← randomly-chosen member of
$S'$
                        $Y_p \leftarrow Y_q \leftarrow \frac{Y_p + Y_q}{2}$
                    end for
                end if
                $\Phi_i \leftarrow Y_i - \overline{w}_i$
                if $\Phi_i > \delta_i d$ then
                    $\eta_i \leftarrow calibrationFault$
                end if
            end for
            $\Phi \leftarrow \{Y_i - \overline{w_i} \colon i \in S\}$
        end if
        for $i \in S$ do
            $\tilde{y}_{i,t} \leftarrow y_{i,t} + \Phi_{i,t}$
        end for
    end for
end procedure
```

Once every λ epochs, the system performs distributed calibration. Each node first calculates the mean of its window Y_i, then repeatedly (over ρ iterations) updates this value by exchanging data with one randomly-chosen nearby node and updating Y_i to compute an aggregate mean of its neighbourhood of nodes. We take the difference between the aggregate mean and a node's local mean as the *calibration factor* for that node, and infer from a sufficiently large calibration factor (again relative to δ_i) that i is faulty.

6. EXPERIMENTAL ANALYSIS

Our experiments measure the capacity of the system first to distinguish a faulty sensor from a fault-free sensor, and second to distinguish calibration faults (where the reported data may

[1] Note that one can use a data-driven approach to compute the correlations present in the data using clustering techniques. We leave this approach for future work.

be recoverable if the calibration offset can be identified) and stuck-at faults from stuck-at faults (where the reported data are of no use).

Figure 1 shows the sensitivity and specificity of our algorithm when distinguishing faulty sensors from fault-free sensors. Our algorithm exhibits both extremely high sensitivity (sensitivity ≥ 0.959) and high specificity (specificity ≥ 0.99).

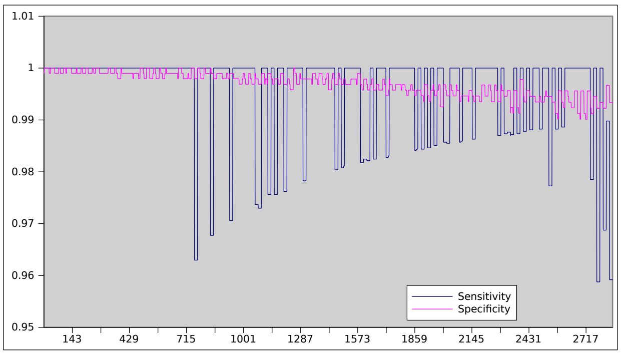

Fig. 1. Sensitivity and specificity of Algorithm 2

Figure 2 shows the accuracy with which our algorithm classifies those sensors which it has correctly identified as faulty. Again, the results are impressive (accuracy ≥ 0.8).

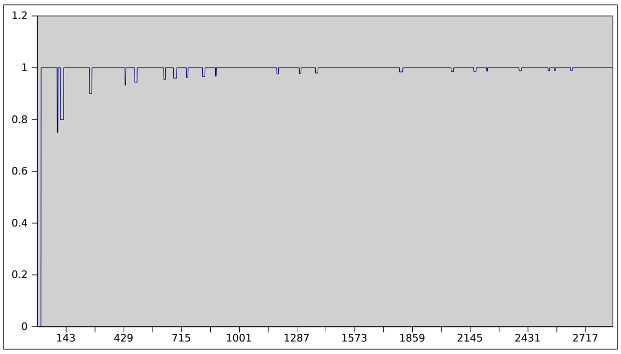

Fig. 2. Accuracy of classification of correctly-diagnosed faulty sensors

6.1 *Parameter sensitivity*

Figures 3–5 demonstrate the effect of varying parameters on the algorithm's performance: specifically, window size, threshold factors, and the number of epochs between invocations of calibration, respectively.

Specificity is uniformly high, uncorrelated with variations in any parameters. Accuracy of classification is correlated with window size (with a Pearson's coefficient of 0.77) but is otherwise uniformly high.

Sensitivity and window size are closely correlated (with a Pearson's coefficient of 0.85), as are sensitivity and threshold multiplier (with a Pearson's coefficient of 0.93). The situation with respect to varying values of λ is less clear (sensitivity and λ are correlated with a Pearson's coefficient of 0.18).

7. RELATED WORK

Calibration of sensors has received little attention in either the sustainable energy or sensor-network communities. Self-tuning systems have been examined in (Nassif et al., 2008), who use genetic algorithms to tune energy models, but only under ideal and nominal conditions. Ma and Wang (2011) use genetic algorithms for an adaptive optimal control strategy for central chiller plants. Simplified models generate ideal performance predictors, against which model parameters are continuously updated using RLS estimation with exponential forgetting. This approach assumes a centralized supervisory control model with no faults. In contrast, our work focuses on distributed methods that can handle both nominal and faulty system models.

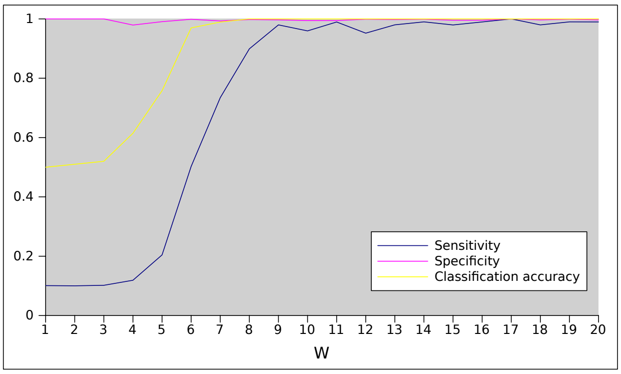

Fig. 3. Algorithm performance with respect to W

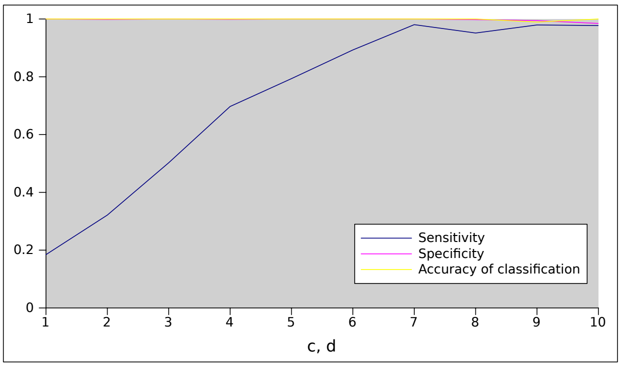

Fig. 4. Algorithm performance with respect to c, d, where $c = d$

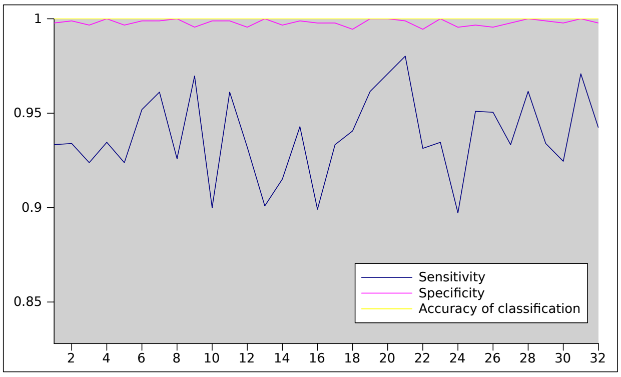

Fig. 5. Algorithm performance with respect to λ

This work is also related to calibration of sensor networks. We provide a means for automated calibration, circumventing manual calibration of large-scale sensor networks, which is impractical for large networks (Whitehouse and Culler, 2002). Automatic calibration has been studied in a number of contexts, including localization (Whitehouse and Culler, 2002), time synchronization (Elson and Estrin, 2001), and motion estimation (Welch and Bishop, 1997). The exploitation of the dense deployment of sensor nodes to perform calibration is investigated in (Bychkovskiy et al., 2003). Methods for automatic calibration which exploit conditions where the mean value of the phenomenon is constant or known are described in (Balzano and Nowak, 2007; Ramakrishnan et al., 2011).

We base our sensor fault models on the frameworks presented in (Ramanathan et al., 2006; Sharma et al., 2010; Ni et al., 2009). (Ramanathan et al., 2006) devised a rule-based system for classifying sensor faults while analysing data from a network deployed to measure the level of arsenic in groundwater. This classification formed the basis for the fault-detection survey in (Sharma et al., 2010), which estimates the prevalence of faults in several real-world data sets.

We extend prior work on gossip protocols for model calibration. A good introduction to the general topic of gossip and epidemic models in distributed systems is given in (Eugster et al., 2004). The use of gossip-based communication to compute aggregate information from values shared among a large network is described in (Jelasity et al., 2005).

8. CONCLUSIONS

We have described a method for calibrating sensors to reduce false alarms and automatically different conditions on system operational status. We used the expectation maximization (EM) algorithm (Dempster et al., 1977), in which the nominal value of a room's state (e.g., temperature) can be estimated using a maximum likelihood estimate (MLE). We extended EM with a gossip-based approach to enable distributed calibration through sharing of sensor data. This capability is very important for a range of applications, including smart buildings, in which sensors are often poorly calibrated and lead to diagnostics false alarms. We have shown empirically that the dependent estimation/calibration produced by the gossip-based method can generate accurate corrective actions, either to indicate stuck-at faults, or to enable sensor calibration. However, the gossip-based method is computationally complex, and requires a method to identify the correlated sensors. We have employed a simple distance-based approach for identifying the correlated sensors/actuators. In future work we plan to examine learning-based approaches for this task, and use models for virtual sensors.

REFERENCES

Laura Balzano and Robert Nowak. Blind calibration of sensor networks. In *Proc. 6th Int. Conf. on Information Processing in Sensor Networks*, pages 79–88, 2007.

J.E. Braun. Intelligent building systems-past, present, and future. In *American Control Conference, 2007. ACC'07*, pages 4374–4381. IEEE, 2007.

Vladimir Bychkovskiy, Seapahn Megerian, Deborah Estrin, and Miodrag Potkonjak. A collaborative approach to in-place sensor calibration. In *Proc. 2nd Int. Conf. on Inform. Process. in Sensor Networks*, pages 301–316, 2003.

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B*, 39: 1–38, 1977.

B. Elkhuizen, E. Rooijakkers, JE Scholten, and J. Knegt. De kwaliteit van installaties in gebouwen. *Verwarming & Ventilatie*, April 2006.

J. Elson and D. Estrin. Time synchronization for wireless sensor networks. In *Proc. 15th Int. Parallel and Distributed Processing Symp.*, pages 1965–1970, April 2001.

P. T Eugster, R. Guerraoui, A. -M Kermarrec, and L. Massoulie. Epidemic information dissemination in distributed systems. *Computer*, 37(5):60–67, May 2004.

J. Feng, S. Megerian, and M. Potkonjak. Model-based calibration for sensor networks. In *Proc. 2nd IEEE Int. Conf. on Sensors*, volume 2, pages 737–742, October 2003.

J.M. House and G.E. Kelly. An overview of building diagnostics. *Diagnostics for Commercial Buildings: Research to Practice*, 1999.

M. Jelasity, A. Montresor, and O. Babaoglu. Gossip-based aggregation in large dynamic networks. *ACM Trans. Comput. Syst.*, 23(3):219–252, 2005.

W. Kowalczyk and N. Vlassis. Newscast EM. *Advances in neural information processing systems*, 17, 2005.

Z. Ma and S. Wang. Supervisory and optimal control of central chiller plants using simplified adaptive models and genetic algorithm. *Applied Energy*, 88(1):198–211, 2011.

N. Nassif, S. Moujaes, and M. Zaheeruddin. Self-tuning dynamic models of hvac system components. *Energy and Buildings*, 40(9):1709–1720, 2008.

Kevin Ni, Nithya Ramanathan, Mohamed Nabil Hajj Chehade, Laura Balzano, Sheela Nair, Sadaf Zahedi, Eddie Kohler, Greg Pottie, Mark Hansen, and Mani Srivastava. Sensor network data fault types. *ACM Trans. on Sensor Networks*, 5 (3):25:1–25:29, June 2009.

M.A. Piette, S.K. Kinney, and P. Haves. Analysis of an information monitoring and diagnostic system to improve building operations. *Energy and Buildings*, 33(8):783–791, 2001.

M.A. Piette, B. Nordman, and S. Greenberg. Quantifying energy savings from commissioning: preliminary results from the pacific northwest. In *Proceedings of the second national conference on building commissioning*, 1994.

N. Ramakrishnan, E. Ertin, and R. Moses. Gossip-based algorithm for joint signature estimation and node calibration in sensor networks. *Selected Topics in Signal Processing, IEEE Journal of*, 5(4):665–673, 2011.

Nithya Ramanathan, Laura Balzano, Marci Burt, Deborah Estrin, Tom Harmon, Charlie Harvey, Jenny Jay, Eddie Kohler, Sarah Rothenberg, and Mani Srivastava. Rapid deployment with confidence: Calibration and fault detection in environmental sensor networks. Technical report, CENS, UCLA, 2006.

Abhishek B. Sharma, Leana Golubchik, and Ramesh Govindan. Sensor faults: detection methods and prevalence in real-world datasets. *ACM Trans. on Sensor Networks*, 6(3):23:1–23:39, June 2010.

Greg Welch and Gary Bishop. SCAAT: Incremental tracking with incomplete information. In *Proc. 24th Annu. Conf. on Computer Graphics and Interactive Techniques*, pages 333–344, 1997.

Kamin Whitehouse and David Culler. Calibration as parameter estimation in sensor networks. In *Proc. 1st ACM Int. Workshop on Wireless Sensor Networks and Applications*, pages 59–67, 2002.