

Chapter 8

Smart Building Sensor Drift Calibration



Tinghuan Chen, Bingqing Lin, Hao Geng, and Bei Yu

8.1 Introduction

In modern smart building, the temperature measurement is a key step for smart temperature management implemented by a cyber-physical system (CPS) [1, 2]. CPS is a complex, heterogeneous distributed system with seamlessly integrated and closely interacted cyber components (e.g., sensors, sink nodes, control centers, and actuators) and physical processes (e.g., temperature) [3]. As shown in Fig. 8.1, the physical world is sensed by corresponding sensors and the acquired data is sent to a sink node or control center. Then the sink node or control center will send an instruction to actuators to control the physical world after the data is analyzed. In smart building, the in-building temperatures are monitored by several spatially distributed and immovable temperature sensors.

©T. Chen et al. | ACM 2019. This is a minor revision of the work published in DAC'19, June 2–6, 2019, Las Vegas, NV, USA <https://doi.org/10.1145/3316781.3317909>

T. Chen · H. Geng · B. Yu (✉)
CSE Department, Chinese University of Hong Kong, Hong Kong, China
e-mail: tchen@cse.cuhk.edu.hk; hgeng@cse.cuhk.edu.hk; byu@cse.cuhk.edu.hk

B. Lin
Shenzhen University, Shenzhen, China
e-mail: bqlin@szu.edu.cn

© Springer Nature Switzerland AG 2020
S. Hu, B. Yu (eds.), *Big Data Analytics for Cyber-Physical Systems*,
https://doi.org/10.1007/978-3-030-43494-6_8

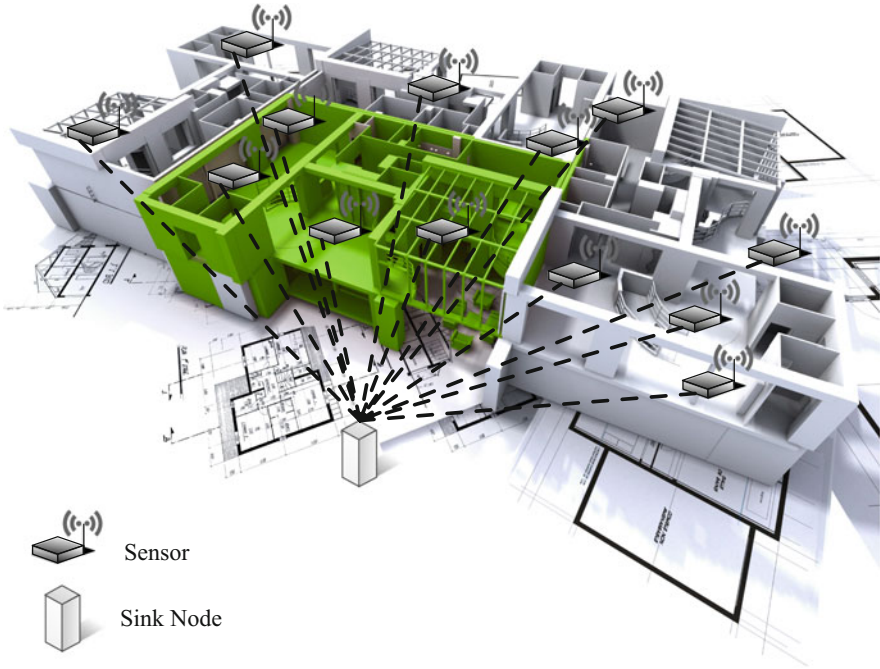


Fig. 8.1 CPS in modern smart building

Although advanced technologies in the semiconductor industry and micro-electromechanical systems are developed in recent years, in practice, sensors outputs exist errors, which are one of the major barriers to the use of sensor networks. There are three main types of errors: gain, drift, and noise [4]. Compared with gain and noise, the sensor drift is considered with vital importance since it has significantly negative effect on measurement accuracy [5]. Although sensors with high accuracy can be deployed, these sensors always have expensive price. As shown in Fig. 8.2a, the temperature sensor AD590JH with $\pm 0.5^\circ\text{C}$ accuracy is sold at more than tenfold price of TMP100 with $\pm 2^\circ\text{C}$ accuracy.

Sensor drift calibration has been studied in many literatures. Without further assumption, calibration cannot be performed. In [7–9], at most one sensor is assumed to have an unknown drift, which is estimated by Kalman filter. In practice, this assumption is hard to satisfy. Therefore, the calibration problem is naturally studied extensively to be a sparse reconstruction problem, where a sparse set of sensors are assumed to have significant drifts. These drift calibration works mainly depend on subspace prior, which is first proposed by Balzano and Nowak to perform calibration when variational sources are over-sampled by sensors [10]. The projection matrix is obtained by singular value decomposition (SVD) [10, 11]. In [11], Wang et.al. adopt temporal sparse Bayesian learning (TSBL) [12] to calibrate time-variant and incremental drifts for the sparse set of sensors. However, due to the

Part Number	Temp. Range	Accuracy	Price [12]
SMT172	-45 ~ 130 °C	±0.25 °C	\$ 35.13
AD590JH	-50 ~ 150 °C	±0.5 °C	\$ 17.91
TMP100	-55 ~ 125 °C	±2.0 °C	\$ 1.79
MCP9509	-40 ~ 125 °C	±4.5 °C	\$ 0.88
LM335A	-40 ~ 100 °C	±5.0 °C	\$ 0.75

(a)



(b)



(c)

Fig. 8.2 (a) Comparison of different temperature sensors; (b) Sensor MCP9509; (c) Sensor LM335A

sparsity assumption, not all sensors can be calibrated. In addition, since observation matrix is directly determined by drift-free measurement, the method cannot calibrate drifts if signals lie in time-variant subspace.

Very recently, in order to calibrate all sensors, Ling and Strohmmer presented three models, which are formulated as bilinear inverse problems [13]. However, these models heavily rely on the partial information about the sensing matrix. For the temperature sensor calibration in a smart building, the sensing matrix depends on weather, the position of sensors, and parameters of the building, e.g., material characteristics, geometry, and equipment power per area [1, 2, 14]. In practice, it is hard to obtain these complex and tedious information. As a result, these models cannot be directly used to calibrate temperature sensors in a smart building.

In this paper, we focus on the temperature sensor drift calibration. Several low-cost sensors with low accuracy are deployed to sense in-building temperatures (see Fig. 8.2b,c). Unlike traditional arts, we build a sensor spatial correlation model whose coefficients only depend on measurements, and we assume that all sensors have drifts. Our model coefficients are optimally determined by statistically extracting prior information from drift-free measurement model coefficients and *maximum-a-posteriori* (MAP) estimation. As a result, our proposed sensor drift calibration framework allows that the signals lie in time-variant subspace. MAP estimation is formulated as a non-convex problem with three hyper-parameters. We propose an alternating-based optimization algorithm to handle the non-convex formulation. Cross-validation and *expectation-maximization* (EM) with Gibbs sampling are adopted to determine hyper-parameters, respectively.

Experimental results show that on benchmarks simulated from EnergyPlus, compared with state-of-the-art method, the proposed framework with EM can achieve a better trade-off between accuracy and runtime.

The rest of this paper is organized as follows. In Sect. 8.2, we provide a problem formulation about sensor drift calibration and broadly introduce our proposed whole flow. In Sect. 8.3, we build a drift calibration model based on sensor spatial correlation and deliver mathematical formulation with three hyper-parameters.

In Sect. 8.4, we propose a more efficient method to handle the mathematical formulation. In Sect. 8.5, three hyper-parameters are determined by cross-validation and EM with Gibbs sampling, respectively. Section 8.6 presents experimental results with comparison and discussion, followed by conclusion in Sect. 8.7.

8.2 Preliminary

8.2.1 Problem Formulation

Several low-cost sensors are deployed to sense in-building temperatures. Due to a slow-aging effect, all sensors have unknown time-invariant drifts. As shown in Fig. 8.3, unlike communication channels [12], for a sensor signal to be output, e.g., current, it is contaminated by a time-invariant drift. In order to achieve high-accurate measurements, drifts need to be estimated and calibrated. Specifically, the mean absolute percent error (MAPE) is used to evaluate drift calibration accuracy.

Based on the above description, we define the sensor drift calibration problem as follows.

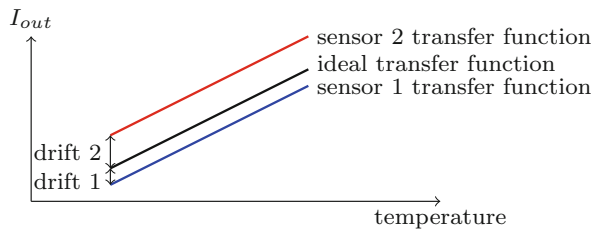
Problem 1 (Sensor Drift Calibration) Given the measurement values sensed by all sensors during several time-instants, drifts will be accurately estimated and calibrated.

8.2.2 Overall Flow

The overall flow of our proposed sensor drift calibration is shown in Fig. 8.4, which consists of three parts: model optimization, cross-validation, and EM with Gibbs sampling.

After drift-free measurements model coefficients and several temperature measurements with drifts are input, an alternating-based optimization algorithm is proposed to handle sensor drift calibration formulation in model optimization. In addition, cross-validation and EM with Gibbs sampling are adopted to induce

Fig. 8.3 Drift vs. temperature [15]



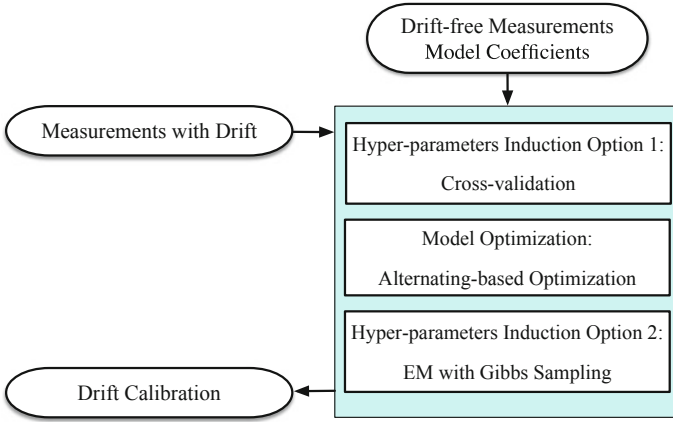


Fig. 8.4 The proposed sensor drift calibration flow

hyper-parameters, respectively. By using the proposed sensor drift calibration, it is expected to accurately calibrate sensor drifts.

8.3 Mathematical Formulation

We assume that n sensors are deployed to sense in-building temperatures. During a short time after new sensors are deployed, the drift is assumed to be insignificant. Furthermore, like [11], we assume all sensors are drift-free during m_0 initial time-instants. Due to over-sampling, as illustrated in [10, 11], signals measured by sensors lie in a low dimensional subspace. Furthermore, in a smart building, all actual temperatures measured by sensors have a high correlation, for example, the dense deployment of sensors. Therefore, we build a linear model among all actual temperatures as follows:

$$x_i^{(k)} \approx \sum_{j=1, j \neq i}^n a_{i,j} x_j^{(k)} + a_{i,0}, \quad k = 1, 2, \dots, m_0, \quad (8.1)$$

where $x_i^{(k)}$ is the ground-truth temperature sensed by i th sensor at k th time-instant. $a_{i,j}$ is the drift-free model coefficient. We define $\mathbf{x} = [x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}, \dots, x_n^{(m_0)}]^\top$, $\mathbf{a}_i = [a_{i,0}, \dots, a_{i,i-1}, a_{i,i+1}, \dots, a_{i,n}]^\top \in \mathbb{R}^n$, $\mathbf{a} = [\mathbf{a}_1^\top, \mathbf{a}_2^\top, \dots, \mathbf{a}_n^\top]^\top \in \mathbb{R}^{n^2}$.

Due to a slow-aging effect, all sensors have unknown time-invariant drifts. During m time-instants, Eq. (8.1) is naturally extended as

$$\hat{x}_i^{(k)} + \epsilon_i \approx \sum_{j=1, j \neq i}^n \hat{a}_{i,j} \left(\hat{x}_j^{(k)} + \epsilon_j \right) + \hat{a}_{i,0}, \quad k = 1, 2, \dots, m, \quad (8.2)$$

where $\hat{x}_i^{(k)}$ is the measurement value sensed by i th sensor at k th time-instant. In particular, in order to obtain enough information, we assume $m_0, m > n$. For i th sensor, ϵ_i is a time-invariant drift calibration, which is independent of time-instant k . $\hat{a}_{i,j}$ is the model coefficient when all sensors have unknown time-invariant drifts. We vectorize these variables as $\hat{\mathbf{x}} = [\hat{x}_1^{(1)}, \hat{x}_2^{(1)}, \dots, \hat{x}_n^{(m)}]^\top$, $\hat{\mathbf{a}}_i = [\hat{a}_{i,0}, \dots, \hat{a}_{i,i-1}, \hat{a}_{i,i+1}, \dots, \hat{a}_{i,n}]^\top \in \mathbb{R}^n$, $\hat{\mathbf{a}} = [\hat{\mathbf{a}}_1^\top, \hat{\mathbf{a}}_2^\top, \dots, \hat{\mathbf{a}}_n^\top]^\top \in \mathbb{R}^{n^2}$, and $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \dots, \epsilon_n]^\top \in \mathbb{R}^n$.

Note that Eq. (8.2) is essential in our proposed sensor spatial correlation model. Furthermore, the model error in Eq. (8.2) is assumed to follow identical independent Gaussian distribution with zero-mean and unknown precision (inverse variance) δ_0 . Therefore, the likelihood function $\mathcal{P}(\hat{\mathbf{x}}|\hat{\mathbf{a}}, \boldsymbol{\epsilon})$ is defined as follows:

$$\mathcal{P}(\hat{\mathbf{x}}|\hat{\mathbf{a}}, \boldsymbol{\epsilon}) \propto \exp \left(-\frac{\delta_0}{2} \sum_{i=1}^n \sum_{k=1}^m \left[\hat{x}_i^{(k)} + \epsilon_i - \sum_{j=1, j \neq i}^n \hat{a}_{i,j} \left(\hat{x}_j^{(k)} + \epsilon_j \right) - \hat{a}_{i,0} \right]^2 \right). \quad (8.3)$$

However, the likelihood function $\mathcal{P}(\hat{\mathbf{x}}|\hat{\mathbf{a}}, \boldsymbol{\epsilon})$ cannot be directly used to calibrate drifts using *maximum-likelihood-estimation* (MLE) since it has not enough information. Therefore, we need to give two priors in development.

For all sensors, drifts are assumed to follow identical independent Gaussian distribution with zero-mean and unknown precision δ_ϵ as follows:

$$\mathcal{P}(\boldsymbol{\epsilon}) \propto \exp \left(-\frac{\delta_\epsilon}{2} \sum_{i=1}^n \epsilon_i^2 \right). \quad (8.4)$$

In addition, we assume that the model coefficient $\hat{a}_{i,j}$ follows identical independent Gaussian distribution. Intuitively, $\hat{a}_{i,j}$ has high dependency on $a_{i,j}$ in statistics. Furthermore, the probability density function of $\hat{a}_{i,j}$ is assumed to take a maximum value at $a_{i,j}$. Therefore, the prior mean of $\hat{a}_{i,j}$ is $a_{i,j}$. In addition, in order that each model coefficient $\hat{a}_{i,j}$ is provided with a relatively equal probability to deviate from the corresponding drift-free model coefficient $a_{i,j}$, the precision of model coefficient $\hat{a}_{i,j}$ is defined to be $\lambda a_{i,j}^{-2}$, where λ is a nonnegative hyper-parameter to control the precision of $\hat{a}_{i,j}$. Therefore, each model coefficient $\hat{a}_{i,j}$ follows identical independent Gaussian distribution with $a_{i,j}$ mean and $\lambda a_{i,j}^{-2}$ precision [16–18]. For all model coefficients, we have

$$\mathcal{P}(\hat{\mathbf{a}}) \propto \exp \left(- \sum_{i=1}^n \sum_{j=0, j \neq i}^n \frac{\lambda}{2a_{i,j}^2} (\hat{a}_{i,j} - a_{i,j})^2 \right). \quad (8.5)$$

In order to calibrate drifts for all sensors, the posterior $\mathcal{P}(\hat{\mathbf{a}}, \boldsymbol{\epsilon} | \hat{\mathbf{x}})$ needs to be maximized in MAP estimation manner. According to Bayes' rule, the posterior $\mathcal{P}(\hat{\mathbf{a}}, \boldsymbol{\epsilon} | \hat{\mathbf{x}})$ can be expressed by two priors and the likelihood function as follows:

$$\mathcal{P}(\hat{\mathbf{a}}, \boldsymbol{\epsilon} | \hat{\mathbf{x}}) \propto \mathcal{P}(\hat{\mathbf{x}} | \hat{\mathbf{a}}, \boldsymbol{\epsilon}) \cdot \mathcal{P}(\hat{\mathbf{a}}) \cdot \mathcal{P}(\boldsymbol{\epsilon}). \quad (8.6)$$

Taking the logarithm, MAP can be transferred to the equivalent formulation as follows:

$$\begin{aligned} \min_{\hat{\mathbf{a}}, \boldsymbol{\epsilon}} \quad & \delta_0 \sum_{i=1}^n \sum_{k=1}^m \left[\hat{x}_i^{(k)} + \epsilon_i - \sum_{j=1, j \neq i}^n \hat{a}_{i,j} (\hat{x}_j^{(k)} + \epsilon_j) - \hat{a}_{i,0} \right]^2 \\ & + \lambda \sum_{i=1}^n \sum_{j=0, j \neq i}^n \frac{1}{a_{i,j}^2} (\hat{a}_{i,j} - a_{i,j})^2 + \delta_\epsilon \sum_{i=1}^n \epsilon_i^2. \end{aligned} \quad (8.7)$$

There are two challenges for Formulation (8.7): how to handle Formulation (8.7) and how to induce hyper-parameters λ , δ_0 , and δ_ϵ .

8.4 Alternating-Based Optimization

Formulation (8.7) is a non-convex problem; thus, it is difficult to obtain an optimal solution. In this section, we propose a fast and efficient alternating-based optimization methodology to handle Formulation (8.7) by alternatively updating in each iteration.

According to the alternating-based methodology, at each iteration, the values of $\hat{\mathbf{a}}$ and $\boldsymbol{\epsilon}$ are updated by optimizing Formulation (8.7) *w.r.t.* $\hat{\mathbf{a}}$ and $\boldsymbol{\epsilon}$. Furthermore, note that with fixed drift calibration variable $\boldsymbol{\epsilon}$, Formulation (8.7) *w.r.t.* $\hat{\mathbf{a}}$ is regarded as a convex unconstrained quadratic programming (QP) problem. In addition, Formulation (8.7) *w.r.t.* $\hat{\mathbf{a}}$ can be decomposed into n independent sub-formulations *w.r.t.* $\hat{\mathbf{a}}_i$ as follows:

$$\begin{aligned} \min_{\hat{\mathbf{a}}_i} \quad & \delta_0 \sum_{k=1}^m \left[\hat{x}_i^{(k)} + \epsilon_i - \sum_{j=1, j \neq i}^n \hat{a}_{i,j} (\hat{x}_j^{(k)} + \epsilon_j) - \hat{a}_{i,0} \right]^2 \\ & + \lambda \sum_{j=0, j \neq i}^n \frac{1}{a_{i,j}^2} (\hat{a}_{i,j} - a_{i,j})^2, \end{aligned} \quad (8.8)$$

with the first-order optimality condition:

$$\delta_0 \sum_{k=1}^m \left(\hat{x}_t^{(k)} + \epsilon_t \right) \left[\sum_{j=1}^n \hat{a}_{i,j} \left(\hat{x}_j^{(k)} + \epsilon_j \right) + \hat{a}_{i,0} \right] + \lambda \frac{(\hat{a}_{i,t} - a_{i,t})}{a_{i,t}^2} = 0, \quad (8.9)$$

where $t = 0, 1, \dots, i-1, i+1, \dots, n$. In particular, we define $\hat{a}_{i,i} \triangleq -1$ and $\hat{x}_0^{(k)} + \epsilon_0 \triangleq 1$. The system of linear equations (8.9) can be handled by Gaussian elimination [19].

In the same manner, with fixed model coefficients $\hat{\mathbf{a}}$, Formulation (8.7) *w.r.t.* the drift calibration ϵ can also be regarded to be a convex unconstrained QP problem as follows:

$$\min_{\epsilon} \delta_0 \sum_{i=1}^n \sum_{k=1}^m \left[\hat{x}_i^{(k)} + \epsilon_i - \sum_{j=1, j \neq i}^n \hat{a}_{i,j} \left(\hat{x}_j^{(k)} + \epsilon_j \right) - \hat{a}_{i,0} \right]^2 + \delta_{\epsilon} \sum_{i=1}^n \epsilon_i^2, \quad (8.10)$$

with the corresponding first-order optimality condition:

$$\delta_0 \sum_{i=1}^n \sum_{k=1}^m \left[\hat{a}_{i,t} \left(\sum_{j=1}^n \hat{a}_{i,j} \left(\hat{x}_j^{(k)} + \epsilon_j \right) + \hat{a}_{i,0} \right) \right] + \delta_{\epsilon} \epsilon_t = 0, \quad (8.11)$$

where $t = 1, 2, \dots, n$.

Algorithm 1 Alternating-based method

Input: Sensor measurements $\hat{\mathbf{x}}$, prior \mathbf{a} and hyper-parameters $\lambda, \delta_0, \delta_{\epsilon}$.

- 1: Initialize $\hat{\mathbf{a}} \leftarrow \mathbf{a}$ and $\epsilon \leftarrow \mathbf{0}$;
 - 2: **repeat**
 - 3: **for** $i \leftarrow 1$ to n **do**
 - 4: Fix ϵ , solve the system of linear equations (8.9) using Gaussian elimination to update $\hat{\mathbf{a}}_i$;
 - 5: **end for**
 - 6: Fix $\hat{\mathbf{a}}$, solve the system of linear equations (8.11) using Gaussian elimination to update ϵ ;
 - 7: **until** Convergence
 - 8:
 - 9: **return** $\hat{\mathbf{a}}$ and ϵ .
-

A local optimum can be obtained by the proposed alternating-based method while the convergence speed and solution quality depend on the initialization of variables. In our proposed framework, two priors are given for model coefficients $\hat{\mathbf{a}}$ and drift calibration ϵ . Therefore, in order to achieve a better convergence speed and solution quality, the prior means \mathbf{a} and $\mathbf{0}$ are used to initialize variables $\hat{\mathbf{a}}$ and ϵ . We continue to update $\hat{\mathbf{a}}$ and ϵ until convergence. The convergence condition is that the relative difference of drift calibration ϵ between current and previous iterations is

less than a threshold. In summary, our proposed alternating-based method is shown in Algorithm 1.

8.5 Estimation of Hyper-Parameters

It is important to induce the aforementioned three hyper-parameters so that drifts can be accurately calibrated and meanwhile the over-fitting can be avoided. In this section, cross-validation and EM with Gibbs sampling are used to induce hyper-parameters, respectively.

8.5.1 Unsupervised Cross-Validation

Cross-validation is a simple method to select hyper-parameters. Although there are three hyper-parameters λ , δ_0 , δ_ϵ in Formulation (8.7), only two ratios λ/δ_0 and δ_ϵ/δ_0 need to be determined instead of individual hyper-parameters by cross-validation. We partition temperature measurements during m time-instants into s non-overlapping parts. Given each combination of ratios candidates λ/δ_0 and δ_ϵ/δ_0 , in each run, one of the s parts is used to estimate the model error and all other $s - 1$ parts are used to calculate model coefficients and drift calibration. In the same manner, each run gives a model error e_r ($r = 1, 2, \dots, s$) estimated from a part of temperature measurements. The final model error is computed as the average $\bar{e} = (e_1 + e_2 + \dots + e_s)/s$. Then two ratios λ/δ_0 and δ_ϵ/δ_0 corresponding to the minimum average model error are chosen.

Note that unlike conventional cross-validation [1, 2, 14, 16–18], not any golden value of drift calibration is used in metrics to choose hyper-parameters in model fitting stage. Therefore, in our proposed framework, cross-validation is adopted in an unsupervised-learning-like fashion.

Cross-validation is time-consuming since Algorithm 1 has to be performed for multiple times. Thus, we propose a fast and efficient EM algorithm to determine hyper-parameters in statistical model.

8.5.2 Monte Carlo Expectation Maximization

In this section, MLE is used to determine individual hyper-parameters δ_0 , λ , and δ_ϵ . MLE of hyper-parameters is formulated as follows:

$$\max_{\delta_\epsilon, \delta_0, \lambda} \mathcal{P}(\hat{\mathbf{x}}; \delta_0, \lambda, \delta_\epsilon). \quad (8.12)$$

However, the likelihood function $\mathcal{P}(\hat{\mathbf{x}}; \delta_0, \lambda, \delta_\epsilon)$ is intractable. EM algorithm is leveraged to efficiently find a solution to Formulation (8.12). According to EM algorithm, Formulation (8.12) can be taken the logarithm and transferred to be its auxiliary lower bound function [20]. Then, the auxiliary lower bound function is optimized by **E-step** and **M-step** iteratively after the term independent of hyper-parameters is omitted. The detailed derivation can be found in [21]. For convenience, all hyper-parameters are collected as a set Ω .

8.5.2.1 Expectation Step with Gibbs Sampling

In **E-step**, the auxiliary lower bound function can be simplified to be a quantity defined as follows:

$$Q(\Omega|\Omega^{\text{old}}) = \int \int \mathcal{P}(\hat{\mathbf{a}}, \boldsymbol{\epsilon}|\hat{\mathbf{x}}; \Omega^{\text{old}}) \ln \mathcal{P}(\hat{\mathbf{x}}, \hat{\mathbf{a}}, \boldsymbol{\epsilon}; \Omega) d\hat{\mathbf{a}} d\boldsymbol{\epsilon}, \quad (8.13)$$

where Ω^{old} denotes estimated hyper-parameters in the previous iteration.

However, the posterior $\mathcal{P}(\hat{\mathbf{a}}, \boldsymbol{\epsilon}|\mathbf{x}; \Omega^{\text{old}})$ is intractable. There are two main methods to approximate the posterior $\mathcal{P}(\hat{\mathbf{a}}, \boldsymbol{\epsilon}|\mathbf{x}; \Omega)$: variational inference and *Markov chain Monte Carlo* (MCMC). Compared with variational inference, MCMC has the advantage of being non-parametric and asymptotically exact [22]. Therefore, Monte Carlo method is utilized to approximate the quantity as follows:

$$Q(\Omega|\Omega^{\text{old}}) \approx \frac{1}{L} \sum_{l=1}^L \ln \mathcal{P}(\hat{\mathbf{x}}, \hat{\mathbf{a}}^{(l)}, \boldsymbol{\epsilon}^{(l)}; \Omega), \quad (8.14)$$

where samples $\hat{\mathbf{a}}^{(l)}$ and $\boldsymbol{\epsilon}^{(l)}$ are obtained from the distribution $\mathcal{P}(\hat{\mathbf{a}}, \boldsymbol{\epsilon}|\hat{\mathbf{x}}; \Omega^{\text{old}})$. L is total amount of samples. In MCMC, there are two main algorithms to obtain samples from the desired distribution $\mathcal{P}(\hat{\mathbf{a}}, \boldsymbol{\epsilon}|\hat{\mathbf{x}}; \Omega^{\text{old}})$: Metropolis Hastings algorithm and Gibbs sampling. Since the rejection rate will be high in complex problems, Metropolis Hastings algorithm has very slow convergence [21]. Therefore, Gibbs sampling is used to obtain samples $\hat{\mathbf{a}}^{(l)}$ and $\boldsymbol{\epsilon}^{(l)}$.

Gibbs sampling has the behavior that one or batch variables are cyclically and repeatedly updated in some particular order at random from conditional distribution. Sampling order is arranged to be $\hat{a}_{1,0}^{(l)}, \dots, \hat{a}_{1,n}^{(l)}, \hat{a}_{2,0}^{(l)}, \dots, \hat{a}_{n,n-1}^{(l)}, \epsilon_1^{(l)}, \dots, \epsilon_n^{(l)}$. In Gibbs sampling, one of key points is derivation of the conditional distribution for each variable. Note that according to Formulation (8.7), the log conditional distribution *w.r.t.* individual variable is quadratic. Therefore, the conditional distribution of each variable is Gaussian distribution as follows:

$$\begin{aligned} \hat{a}_{p,q} &\sim \mathcal{P}(\hat{a}_{p,q}|\boldsymbol{\epsilon}, \hat{\mathbf{a}}_{/\hat{a}_{p,q}}, \hat{\mathbf{x}}; \delta_\epsilon, \delta, \lambda) = \mathcal{N}(\mu_{\hat{a}_{p,q}}, \sigma_{\hat{a}_{p,q}}^{-1}), \\ \epsilon_t &\sim \mathcal{P}(\epsilon_t|\boldsymbol{\epsilon}_{/\epsilon_t}, \hat{\mathbf{a}}, \hat{\mathbf{x}}; \delta_\epsilon, \delta, \lambda) = \mathcal{N}(\mu_{\epsilon_t}, \sigma_{\epsilon_t}^{-1}), \end{aligned} \quad (8.15)$$

in agreement with (8.4) and (8.5). μ is mean and σ is precision. $\hat{\mathbf{a}}_{/\hat{a}_{p,q}}$ and $\epsilon_{/\epsilon_t}$ denote $\hat{\mathbf{a}}$ but with $\hat{a}_{p,q}$ omitted and ϵ but with ϵ_t omitted.

Before Gibbs sampling, in order to converge to the desired posterior, the warm-start has to be performed if there is no reasonable initialization for samples. Furthermore, it is very hard to judge whether the warm-start is enough [21]. In order to waive the warm-start, a reasonable initialization for samples is adopted in Gibbs sampling. Note that Gibbs sampling is used to obtain samples from the desired posterior $\mathcal{P}(\hat{\mathbf{a}}, \epsilon | \hat{\mathbf{x}}; \Omega^{\text{old}})$ (8.6). As we discussed in Sect. 8.2, Formulation (8.7) is equivalent to MAP estimation of $\hat{\mathbf{a}}$ and ϵ . Thus given hyper-parameters Ω^{old} and measurement values $\hat{\mathbf{x}}$, Gibbs sampling can be initialized by handling Formulation (8.7) to obtain initial samples $\hat{\mathbf{a}}^{(0)}$ and $\epsilon^{(0)}$ satisfying the distribution $\mathcal{P}(\hat{\mathbf{a}}, \epsilon | \hat{\mathbf{x}}; \Omega^{\text{old}})$. As a result, the warm-start can be totally waived.

8.5.2.2 Maximization Step

After L samples are obtained by Gibbs sampling, in **M-step**, we will maximize the approximated quantity as follows:

$$\max_{\Omega} \frac{1}{L} \sum_{l=1}^L \ln \mathcal{P}(\hat{\mathbf{x}}, \hat{\mathbf{a}}^{(l)}, \epsilon^{(l)}; \Omega). \quad (8.16)$$

With the first-order optimality condition, that is $dQ/d\Omega = 0$, hyper-parameters λ , δ_0 , δ_ϵ can be updated as follows:

$$\lambda = \frac{n^2 L}{\sum_{i=1}^n \sum_{j=0, j \neq i}^n \sum_{l=1}^L \frac{(\hat{a}_{i,j}^{(l)} - a_{i,j})^2}{a_{i,j}^2}}, \quad (8.17)$$

$$\delta_0 = \frac{Lmn}{\sum_{l=1}^L \sum_{i=1}^n \sum_{k=1}^m \left[\sum_{j=1}^n \hat{a}_{i,j}^{(l)} (\hat{x}_j^{(k)} + \epsilon_j^{(l)}) + \hat{a}_{i,0}^{(l)} \right]^2}, \quad (8.18)$$

$$\delta_\epsilon = \frac{nL}{\sum_{l=1}^L \sum_{i=1}^n \epsilon_i^{(l)2}}. \quad (8.19)$$

Here, $\hat{a}_{i,i}^{(l)} \triangleq -1$ and $\hat{x}_0^{(k)} + \epsilon_0^{(l)} \triangleq 1$. We continue to alternate between E-step and M-step until convergence. The convergence condition is that the relative difference of three hyper-parameters between current and previous iterations is less than a threshold. Then hyper-parameters λ , δ , δ_ϵ can be determined.

For convenience, all variables are collected as a set $\Psi = \{\psi_1, \psi_2, \dots, \psi_{n^2+n}\} = \{\hat{a}_{1,0}, \dots, \hat{a}_{1,n}, \dots, \hat{a}_{n,n-1}, \epsilon_1, \dots, \epsilon_n\}$. In summary, our proposed EM with Gibbs sampling is shown in Algorithm 2.

Algorithm 2 EM with Gibbs sampling

Input: Sensor measurements $\hat{\mathbf{x}}$, prior \mathbf{a} ;

- 1: Initialize hyper-parameters Ω ;
 - 2: **repeat**
 - 3: Initialize samples $\Psi^{(0)}$ by Algorithm 1;
 - 4: **for** $l \leftarrow 1$ to L **do**
 - 5: **for** $i \leftarrow 1$ to $n^2 + n$ **do**
 - 6: Sample $\psi_i^{(l)}$ from the desired conditional distribution $\mathcal{N}(\mu_{\psi_i}, \sigma_{\psi_i})$ (8.15) with $\psi_1^{(l)}, \dots, \psi_{i-1}^{(l)}, \psi_{i+1}^{(l-1)}, \dots, \psi_{n^2+n}^{(l-1)}$;
 - 7: **end for**
 - 8: **end for**
 - 9: Update hyper-parameters Ω by Equations (8.17), (8.18) and (8.19);
 - 10: **until** Convergence
 - 11:
 - 12: **return** hyper-parameters Ω .
-

8.6 Experimental Results

The in-building temperature data are used to test our proposed framework. We use several sensors to calibrate drifts. All data is directly generated from EnergyPlus as shown in Fig. 8.5. As shown in Fig. 8.6, two building benchmarks, Hall [23] with Washington, D.C weather and Secondary School [24] with Chicago weather, are simulated by EnergyPlus to generate the ground-truth in-building temperatures, which are used to test our proposed framework. The temperature sampling period is set to be 1 h.

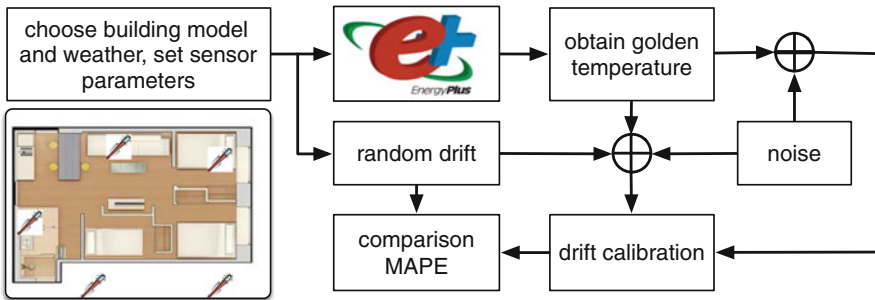


Fig. 8.5 The generated simulation data

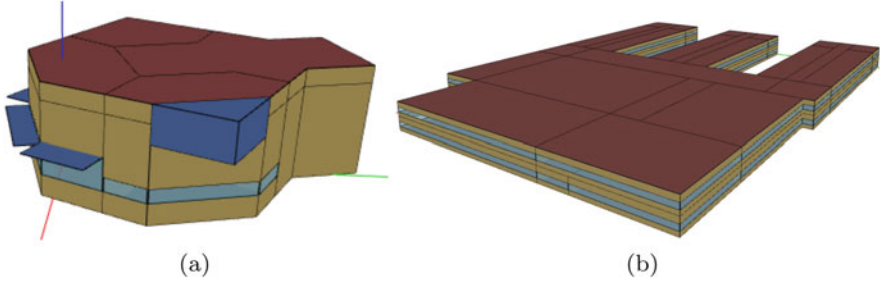


Fig. 8.6 Benchmark: (a) Hall; (b) Secondary School

In practice, both drift and measurement noise need to be carefully considered and reasonably set to close to real temperature measurement. Because of a slow-aging effect, time effects on sensor performance are not considered in our experiments. Drift is set to be time-invariant while measurement noise is set to be time-variant. According to the sensors' performance shown in Fig. 8.2a, two low-cost temperature sensors, MCP9509 with accuracy ± 4.5 °C and LM335A with accuracy ± 5 °C as shown in Fig. 8.2b,c, are chosen to set drift variance, respectively. According to the triple standard deviation, we set two drift variances to be $\sigma^2 = (4.5/3)^2 = 2.25$ and $\sigma^2 = (5/3)^2 = 2.78$. In addition, according to our survey, the noise variance is set to be 0.001. All temperature measurements are generated by adding noise.

The time-instant number needs to be reasonably set to meet practical application and accurately calibrate sensor drifts. We assume the temperature measurements are drift-free during first $m_0 = 240$ time-instants (first 10 days). And during $m = 60$ time-instants (60 h), the temperature measurements with drifts are used to test our proposed framework.

TSBL [11] and the proposed framework with cross-validation and EM are used to calibrate sensor drifts, respectively. All methods are implemented by Python 2.7 on 12-core Linux machine with 256 G RAM and 2.80 GHz. 100 combinations of hyper-parameters ratios and $s = 5$ folds are set in cross-validation. Since the warm-start is waived in Gibbs sampling, in order to achieve a better trade-off between accuracy and runtime, only $L = 10$ samples are generated to perform Monte Carlo approximation (8.14), and three hyper-parameters λ , δ_0 , δ_ϵ are initialized to be 10^3 , 10^{-4} , and 10^{-3} in EM. The convergence criterion thresholds are set to be 10^{-8} and 10^{-2} in Algorithms 1 and 2.

As mentioned in Sect. 8.2, the drift calibration accuracy is evaluated by using MAPE defined as follows:

$$\text{MAPE} = \frac{1}{nm} \sum_{k=1}^m \sum_{i=1}^n \left| \frac{\hat{\epsilon}_i^{(k)} - \epsilon_i}{\epsilon_i} \right|, \quad (8.20)$$

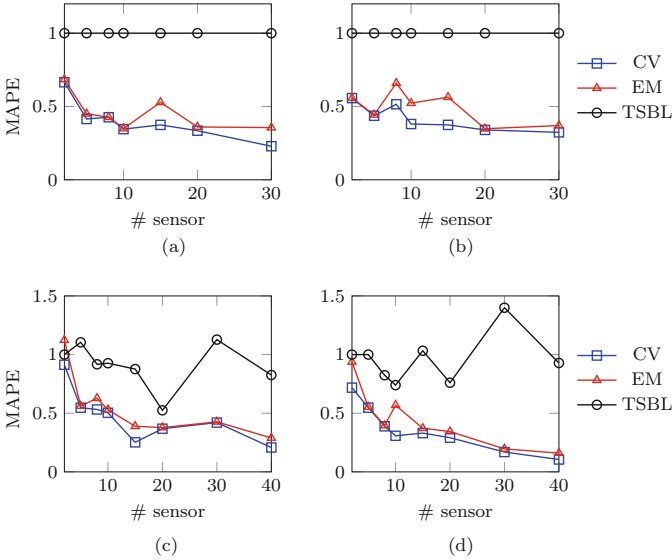


Fig. 8.7 Drift variance is set to (a,c) 2.25; (b,d) 2.78; Benchmark: (a,b) Hall; (c,d) Secondary school

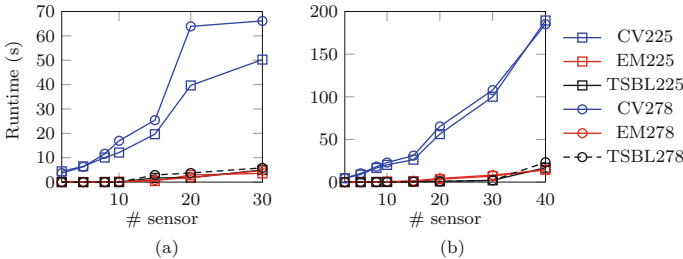


Fig. 8.8 Runtime vs. # sensor on (a) Hall; (b) Secondary school

where $\hat{\epsilon}_i^{(k)}$ is the estimated calibration. Specifically, in our proposed framework, $\hat{\epsilon}_i^{(k)} = \hat{\epsilon}_i$. The sensor drift calibration performances of accuracy and runtime are shown in Figs 8.7 and 8.8.

As shown in Fig. 8.8, TSBL has acceptable computational overhead even if its computational complexity is dominated by multiple matrix inversion operations. However, as shown in Fig. 8.7, TSBL has the worst performance and robust for drifts calibration. In fact, temperature signals lie in time-variant subspace since in-building temperatures are influenced by multiple time-variant factors, e.g., weather. As a result, TSBL cannot achieve an obvious drift calibration.

Unlike TSBL, the proposed spatial correlation model can calibrate drifts even if temperature signals lie in time-variant subspace. Therefore, as shown in Fig. 8.7, the proposed framework with either cross-validation or EM outperforms TSBL in

accuracy. Besides, the proposed drift calibration framework with cross-validation can achieve the best accuracy. However, as shown in Fig. 8.8, cross-validation has heavy computational overhead since we need to run Algorithm 1 for multiple times. Compared with cross-validation and TSBL, EM with Gibbs sampling has lower computation complexity since less samples are generated to perform Monte Carlo approximation and EM can achieve fast convergence. However, as shown in Fig. 8.7, the proposed framework with EM cannot achieve the best accuracy since EM with Gibbs sampling is an approximation method.

As shown in Fig. 8.7, because of incremental correlation, the more sensors can achieve the more accuracy of drift calibration by using our proposed framework. In practice, when less sensors need to be calibrated, in order to achieve a better accuracy, cross-validation can be used to determine hyper-parameters within a reasonable response time, e.g., 1 min. While more sensors need to be calibrated, EM with Gibbs sampling can be used to determine hyper-parameters so that sensor measurement accuracy can be improved to a tolerable level within acceptable runtime. The proposed calibration framework with EM can achieve robust drift calibration and a better trade-off between accuracy and runtime.

8.7 Conclusion

In this paper, a sensor spatial correlation model has been proposed to perform drift calibration. Thanks to spatial correlation, the unknown actual temperature measured by each sensor is linearly expressed by all other sensors. The priors for model coefficients and drift calibration are applied to MAP estimation. MAP estimation is then formulated as a non-convex problem with three hyper-parameters, which is handled by the proposed alternating-based method. Cross-validation and EM with Gibbs sampling are used to determine hyper-parameters, respectively. Experimental results show that on benchmarks simulated from EnergyPlus, the proposed framework with EM can achieve a robust drift calibration and better trade-off between accuracy and runtime.

References

1. X. Chen, X. Li, S.X.-D. Tan, Overview of cyber-physical temperature estimation in smart buildings: from modeling to measurements, in *INFOCOM Workshops* (2016), pp. 251–256
2. B. Lin, B. Yu, Smart building uncertainty analysis via adaptive lasso. *IET Cyber-Phys. Syst. Theory Appl.* **2**(1), 42–48 (2017)
3. Q. Zhu, A. Sangiovanni-Vincentelli, S. Hu, X. Li, Design automation for cyber-physical systems. *Proc. IEEE* **106**(9), 1479–1483 (2018)
4. K. Ni, N. Ramanathan, M.N.H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, M. Srivastava, Sensor network data fault types. *ACM Trans. Sens. Netw. (TOSN)* **5**(3), 25 (2009)

5. *Engineer's Guide to Accurate Sensor Measurements* (2016). http://download.ni.com/evaluation/daq/25188_Sensor_WhitePaper_IA.pdf
6. *Findchips* (2018). <https://www.findchips.com>
7. Y. Wang, A. Yang, Z. Li, P. Wang, H. Yang, Blind drift calibration of sensor networks using signal space projection and Kalman filter, in *IEEE 10th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)* (2015), pp. 1–6
8. M. Takruri, S. Challa, R. Yunis, Data fusion techniques for auto calibration in wireless sensor networks, in *International Conference on Information Fusion* (2009), pp. 132–139
9. M. Takruri, S. Rajasegarar, S. Challa, C. Leckie, Online drift correction in wireless sensor networks using spatio-temporal modeling, in *International Conference on Information Fusion* (2008), pp. 1–8
10. L. Balzano, R. Nowak, Blind calibration of sensor networks, in *International Conference on Information Processing in Sensor Networks (IPSN)* (2007), pp. 79–88
11. Y. Wang, A. Yang, Z. Li, X. Chen, P. Wang, H. Yang, Blind drift calibration of sensor networks using sparse Bayesian learning. *IEEE Sensors J.* **16**(16), 6249–6260 (2016)
12. Z. Zhang, B.D. Rao, Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning. *IEEE J. Sel. Top. Sign. Proces.* **5**(5), 912–926 (2011)
13. S. Ling, T. Strohmer, Self-calibration and bilinear inverse problems via linear least squares. *SIAM J. Imag. Sci. (SIIMS)* **11**(1), 252–292 (2018)
14. X. Chen, X. Li, S. X.-D. Tan, From robust chip to smart building: CAD algorithms and methodologies for uncertainty analysis of building performance, in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (2015), pp. 457–464
15. *2-Terminal IC Temperature Transducer* (2013). <https://www.analog.com/media/en/technical-documentation/data-sheets/AD590.pdf>
16. F. Wang, P. Cachecho, W. Zhang, S. Sun, X. Li, R. Kanj, C. Gu, Bayesian model fusion: large-scale performance modeling of analog and mixed-signal circuits by reusing early-stage data. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD)* **35**(8), 1255–1268 (2016)
17. Q. Huang, C. Fang, F. Yang, X. Zeng, X. Li, Efficient multivariate moment estimation via Bayesian model fusion for analog and mixed-signal circuits, in *ACM/IEEE Design Automation Conference (DAC)* (2015), p. 169
18. Q. Huang, C. Fang, F. Yang, X. Zeng, D. Zhou, X. Li, Efficient performance modeling via dual-prior Bayesian model fusion for analog and mixed-signal circuits, in *ACM/IEEE Design Automation Conference (DAC)* (2016), pp. 1–6
19. G.H. Golub, C.F. Van Loan, *Matrix Computations* (JHU Press, Baltimore, 2012)
20. K. Ganchev, B. Taskar, J. Gama, Expectation maximization and posterior constraints, in *Conference on Neural Information Processing Systems (NIPS)* (2008), pp. 569–576
21. C. Robert, *Machine Learning, a Probabilistic Perspective* (Taylor & Francis, London, 2014)
22. T. Salimans, D. Kingma, M. Welling, Markov chain Monte Carlo and variational inference: bridging the gap, in *International Conference on Machine Learning (ICML)* (2015), pp. 1218–1226
23. *OpenStudio®* (2018). <https://www.openstudio.net>
24. *National Renewable Energy Laboratory OpenStudio Standards* (2018). <https://github.com/NREL/openstudio-standards>