A dynamic Bayesian nonparametric model for blind calibration of sensor networks

Yang, Jielong; Zhong, Xionghu; Tay, Wee Peng

2018

Yang, J., Zhong, X., & Tay, W. P. (2018). A dynamic Bayesian nonparametric model for blind calibration of sensor networks. *IEEE Internet of Things Journal*, 5(5), 3942-3953.
doi:10.1109/JIOT.2018.2847697

<https://hdl.handle.net/10356/102693>

<https://doi.org/10.1109/JIOT.2018.2847697>

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at:
<https://doi.org/10.1109/JIOT.2018.2847697>.

Downloaded on 26 Apr 2023 11:42:54 SGT

A Dynamic Bayesian Nonparametric Model for Blind Calibration of Sensor Networks

Jielong Yang, *Student Member, IEEE*, Xionghu Zhong, *Member, IEEE*, and Wee Peng Tay, *Senior Member, IEEE*

Abstract—We consider the problem of blind calibration of a sensor network where the sensor gains and offsets are estimated from noisy observations of unknown signals. This is in general a non-identifiable problem, unless restrictive assumptions on the signal subspace or sensor observations are imposed. We show that if each signal observed by the sensors follows a known dynamic model with additive noise, then the sensor gains and offsets are identifiable. We propose a dynamic Bayesian nonparametric model to infer the sensors' gains and offsets. Our model allows different sensor clusters to observe different unknown signals, without knowing the sensor clusters a priori. We develop an offline algorithm using block Gibbs sampling and a linearized forward filtering backward sampling method that estimates the sensor clusters, gains and offsets jointly. Furthermore, for practical implementation, we also propose an online inference algorithm based on particle filtering and local Markov chain Monte Carlo. Simulations using a synthetic dataset, and experiments on two real datasets suggest that our proposed methods perform better than several other blind calibration methods, including a sparse Bayesian learning approach, and methods that first cluster the sensor observations and then estimate the gains and offsets.

Index Terms—Blind calibration, sensor network, dynamic Bayesian nonparametrics

I. INTRODUCTION

Because of differences in the materials and electrical components that make up a sensor and other environmental factors, each sensor has a gain and offset that need to be estimated or calibrated before its observations can be interpreted meaningfully [1], [2]. For example, in using sensors to perform monitoring of a physical phenomenon, modeling of the distributions governing the sensor observations is required in various distributed inference methods [3], [4]. A sensor can be calibrated in a controlled environment by utilizing its observations of a known signal to estimate its gain and offset. However, this is a painstaking task, which may be impractical if the sensor network is large, and sensors may be placed in inaccessible locations. Macro calibration has thus been proposed to calibrate an entire sensor network based on observations from all sensors in the network [5], [6]. Furthermore, the calibration is done without first observing the underlying signals that the sensors are observing. This is known as *blind* calibration [1], [7]. Since without knowing the sensors' observed signals, the sensor gains and offsets are non-identifiable if estimation is to be done solely based on the sensors' noisy observations. As such, additional assumptions are required.

In [1], a blind calibration method is proposed based on the assumptions that the underlying signals observed by all sensors are deterministic and lie in a known subspace, and the time-averaged measurements of every sensor approaches a known mean value. Subsequently, principal component analysis (PCA) is introduced to learn such a subspace, and a sparse Bayesian learning (SBL) method [7] and Kalman filter [8] are developed to recover the observation drifts of a subset of sensors. However, a calibration step in which the gains are assumed to be known a priori is required, and sensors not in the estimation subset are assumed to have no drifts. The paper [9] proposes a distributed blind calibration algorithm and assumes that all the sensors observe the same stochastic underlying signal. In some applications in which sensors observe multiple signals, the aforementioned approach cannot be applied if we do not know a priori which sensor is observing which signal. In [10], [11], blind calibration methods are based on redundancy information provided by co-located sensors, which are classified into a group and assumed to have correlated measurements. Both of these two methods assume that sensors are densely deployed. In [12], an informed nonnegative matrix factorization method is proposed based on the assumption that the sensed physical phenomenon can be sparsely represented in a given dictionary, which can be used to regularize the calibration even if the sensors are not densely deployed. However, evolution of the underlying signals observed by the sensors is not considered.

The compressive sensing (CS) framework has also been adopted to jointly recover the sensor gains and signals. In the compressive sensing framework, it is commonly assumed that the signal is projected by a known measurement matrix and then distorted by unknown gains [13], [14] or some transfer function with unknown distortion parameters [15]. In [13] and [14], the blind calibration problem is formulated as a convex optimization problem. In [15], the sensors distort the ground truth signals in other ways than multiplication by a gain and a approximate message passing (AMP) algorithm used in CS is adopted to solve the blind calibration problem. However, these methods do not consider the evolution of the sensor signals.

In this paper, we consider blind calibration when sensors may observe different signals, and the number of the underlying time-varying signals as well as the signals themselves are unknown. We say that two sensors are in the same cluster if they observe the same underlying signal, and we assume that each underlying signal follows a known dynamic model with additive noise. Unlike [1], [10], [11], we do not require sensors to be densely deployed or to know the time-averaged measurements of each sensor. Our main contributions are as

J. Yang, Xionghu Zhong and W. P. Tay are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Emails: jyang022@e.ntu.edu.sg, xhzhong@ntu.edu.sg, wptay@ntu.edu.sg.

follows:

- We prove sufficient conditions for identifiability of sensor gains and offsets when each ground truth signal follows a known dynamic model with additive noise.
- We propose a dynamic Bayesian nonparametric model to model the number of underlying signals as well as the evolution of each signal. At each time t , the sensor observations follow a Bayesian nonparametric model, and the model at time t depends on the model at time $t - 1$. To the best of our knowledge, the use of a dependent Bayesian nonparametric model for sensor network blind calibration is novel. We develop both online and offline inference algorithms to estimate the sensor parameters. The offline inference approach is a linearized forward filtering backward sampling based block Gibbs sampling method. The online algorithm uses particle filter (PF) [16] and local Markov chain Monte Carlo (MCMC), which updates the posterior distribution of the sensor gains and offsets sequentially as the data become available. The computational complexities of both our proposed offline and online algorithms are analyzed.
- We perform simulations as well as experiments on real data, which suggest that our proposed methods perform better than several other methods including the sparse Bayesian learning approach of [7], and methods that first cluster the sensor observations, and then estimate the gains and offsets.

Sensor clustering has been widely assumed in the calibration literature [6], [10], [17]–[21]. Nonparametric Bayesian methods have also been applied when the number of clusters is unknown [22]. We refer the readers to [23]–[25] and the references therein for an overview.

A preliminary version of this paper has been presented in our conference paper [26]. Three new developments are included in this paper: (i) we extend our identifiability analysis to include nonlinear dynamic models; (ii) we develop an online inference algorithm; and (iii) real data experiments are performed.

The rest of this paper is organized as follows. In Section II, we present our problem formulation and sufficient conditions under which the sensor gains and offsets are identifiable. In Section III, we propose a dynamic Bayesian nonparametric model for blind calibration. In Section IV and Section V, we develop offline and online inference methods, respectively. Simulation and experimental results are presented in Section VI, and we conclude in Section VII.

Notations. We use boldfaced characters to represent vectors and matrices. We use $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to represent the Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. We use $\mathcal{TN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to represent a Gaussian distribution whose realized values are truncated to be non-negative, and the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance of the Gaussian distribution before truncation. The notation $y \mid x$ denotes a random variable y conditioned on x , and $p(y \mid x)$ denotes its conditional probability density function. The sequence $y(1), y(2), \dots, y(t)$ is written as $y(1 : t)$. Suppose $f(\boldsymbol{\theta}) = [f^1(\boldsymbol{\theta}), \dots, f^m(\boldsymbol{\theta})]$, where each $f^i(\boldsymbol{\theta})$ is a multi-

variate function of $\boldsymbol{\theta}$. We use $Df(\cdot)$ to denote the Jacobian $[\nabla f^1(\boldsymbol{\theta}), \dots, \nabla f^m(\boldsymbol{\theta})]$ of $f(\boldsymbol{\theta})$.

II. PROBLEM FORMULATION AND IDENTIFIABILITY

In this section, we present our system model and assumptions, and show that the sensor gains and offsets are identifiable under our assumptions.

Consider N sensors monitoring an unknown number of signals $\{\boldsymbol{\theta}_k(t) : k \geq 1\}$ over a period of time $t = 1, 2, \dots, T$. At each time t , sensor n observes

$$\mathbf{y}_n(t) = \alpha_n \boldsymbol{\theta}_{s_n(t)}(t) + \beta_n + \mathbf{w}_n(t), \quad (1)$$

where $\alpha_n > 0$ and β_n are the gain and the offset of the sensor n , respectively, $s_n(t)$ is the index of the signal that sensor n is observing at time t , and $\mathbf{w}_n(t)$ is an additive zero mean observation noise with covariance matrix $\mathbf{R}_n(t)$. The observation noises $\mathbf{w}_n(t)$ are independent across sensors and time. Our objective is to estimate α_n and β_n without knowing $\boldsymbol{\theta}_{s_n(t)}(t)$ a priori. This is in principle a non-identifiable estimation problem since $\alpha_n \boldsymbol{\theta}_{s_n(t)}(t) = (c\alpha_n)(\boldsymbol{\theta}_{s_n(t)}(t)/c)$ for any constant $c \neq 0$. Therefore, assumptions on the structure of the sensor signals are required. For example, in [1], the signals are assumed to belong to a known signal subspace and do not evolve over time. In the following, we consider instead a time-varying sensor signal, and provide a sufficient condition for identifiability of the sensor gains and offsets. We assume that for each signal index k , and for all $t \geq 1$,

$$\boldsymbol{\theta}_k(t) = f_k(\boldsymbol{\theta}_k(t-1)) + \boldsymbol{\epsilon}_k(t), \quad (2)$$

where the function $f_k(\cdot)$ characterizing the signal's evolution is known, $\boldsymbol{\epsilon}_k(t)$ is additive noise with positive definite covariance matrix $\mathbf{Q}_k(t)$, and $\boldsymbol{\theta}_k(0)$ is a random variable with mean $\mathbf{M}_k(0)$. In practice, based on domain expertise, we may know a priori a collection of possible signal models that the sensors can observe. Alternatively, we may have high quality sensors with known drifts and offsets that can be used to characterize the signal models. Such sensors are usually expensive, and can only be deployed sparsely.

Proposition 1. *Suppose (1) and (2) hold for all $k \geq 1$ and $t \geq 1$. For each $n \geq 1$, suppose for some $t \geq 1$ and some $k \geq 1$, we have $s_n(t') = k$ for $t-1 \leq t' \leq t+1$. Then, the parameters (α_n, β_n) are identifiable.*

Proof: Suppose that α_n is not identifiable. Then at time t , there exist different parameters (α_n, β_n) and (α'_n, β'_n) inducing the same linear least squares (LLS) predictor of $\mathbf{y}_n(t) \mid \mathbf{y}_n(t-1)$. Let $\boldsymbol{\Phi}_k(t)$ and $\boldsymbol{\Phi}'_k(t)$ be the covariance matrix of the LLS estimator of $\boldsymbol{\theta}_k(t) \mid \mathbf{y}_n(t-1 : t)$ when the sensor parameters are (α_n, β_n) and (α'_n, β'_n) , respectively. Let $\mathbf{C}_n(t)$ be the covariance matrix of the LLS predictor of $\mathbf{y}_n(t) \mid \mathbf{y}_n(t-1)$. Let $\mathbf{A}_k(t-1) \triangleq Df_k(\boldsymbol{\theta}_k(t-1))$. Then from equations (18.86)–(18.92) of [27], we have

$$\begin{aligned} \mathbf{C}_n(t) &= \alpha_n^2 (\mathbf{A}_k(t-1) \boldsymbol{\Phi}_k(t-1) \mathbf{A}_k(t-1)^T + \mathbf{Q}_k(t)) + \mathbf{R}_n(t) \\ &= (\alpha'_n)^2 (\mathbf{A}_k(t-1) \boldsymbol{\Phi}'_k(t-1) \mathbf{A}_k(t-1)^T + \mathbf{Q}_k(t)) + \mathbf{R}_n(t). \end{aligned} \quad (3)$$

(4)

For simplicity, denote $\mathbf{\Lambda}(t) = \mathbf{A}_k(t-1)\mathbf{\Phi}_k(t-1)\mathbf{A}_k(t-1)^T + \mathbf{Q}_k(t)$ and $\mathbf{\Lambda}'(t) = \mathbf{A}_k(t-1)\mathbf{\Phi}'_k(t-1)\mathbf{A}_k(t-1)^T + \mathbf{Q}_k(t)$. From (3) and (4), we have

$$\mathbf{\Lambda}(t) = \frac{(\alpha'_n)^2}{\alpha_n^2} \mathbf{\Lambda}'(t). \quad (5)$$

From [27], the covariance matrix of the LLS estimator of $\theta_k(t) \mid \mathbf{y}_n(t-1:t)$ is

$$\mathbf{\Phi}_k(t) = \mathbf{\Lambda}(t) - \alpha_n^2 \mathbf{\Lambda}(t) [\alpha_n^2 \mathbf{\Lambda}(t) + \mathbf{R}_n(t)^T]^{-1} \mathbf{\Lambda}(t).$$

Similar to (3) and (4), the covariance matrix of the LLS predictor at time $t+1$ with sensor gain α_n is

$$\begin{aligned} \mathbf{C}_n(t+1) &= \alpha_n^2 (\mathbf{A}_k(t) \mathbf{\Phi}_k(t) \mathbf{A}_k(t)^T + \mathbf{Q}_k(t+1)) + \mathbf{R}_n(t+1) \\ &= \alpha_n^2 [\mathbf{A}_k(t) (\mathbf{\Lambda}(t) - \alpha_n^2 \mathbf{\Lambda}(t) [\alpha_n^2 \mathbf{\Lambda}(t) + \mathbf{R}_n(t)^T]^{-1} \mathbf{\Lambda}(t)) \\ &\quad \cdot \mathbf{A}_k(t)^T] + \alpha_n^2 \mathbf{Q}_k(t+1) + \mathbf{R}_n(t+1) \\ &= \alpha_n^2 \mathbf{A}_k(t) \mathbf{\Lambda}(t) \mathbf{A}_k(t)^T \\ &\quad - \alpha_n^4 \mathbf{A}_k(t) \mathbf{\Lambda}(t) [\alpha_n^2 \mathbf{\Lambda}(t) + \mathbf{R}_n(t)^T]^{-1} \mathbf{\Lambda}(t) \mathbf{A}_k(t)^T \\ &\quad + \alpha_n^2 \mathbf{Q}_k(t+1) + \mathbf{R}_n(t+1) \\ &= (\alpha'_n)^2 \mathbf{A}_k(t) \mathbf{\Lambda}'(t) \mathbf{A}_k(t)^T \\ &\quad - (\alpha'_n)^4 \mathbf{A}_k(t) \mathbf{\Lambda}'(t) [(\alpha'_n)^2 \mathbf{\Lambda}'(t) + \mathbf{R}_n(t)^T]^{-1} \mathbf{\Lambda}'(t) \mathbf{A}_k(t)^T \\ &\quad + (\alpha_n)^2 \mathbf{Q}_k(t+1) + \mathbf{R}_n(t+1), \end{aligned} \quad (6)$$

where the fourth equality follows from (5). Similarly, when the sensor gain is α'_n , we have

$$\begin{aligned} \mathbf{C}_n(t+1) &= (\alpha'_n)^2 \mathbf{A}_k(t) \mathbf{\Lambda}'(t) \mathbf{A}_k(t)^T \\ &\quad - (\alpha'_n)^4 \mathbf{A}_k(t) \mathbf{\Lambda}'(t) [(\alpha'_n)^2 \mathbf{\Lambda}'(t) + \mathbf{R}_n(t)^T]^{-1} \mathbf{\Lambda}'(t) \mathbf{A}_k(t)^T \\ &\quad + (\alpha'_n)^2 \mathbf{Q}_k(t+1) + \mathbf{R}_n(t+1). \end{aligned} \quad (7)$$

Equating (6) and (7), since both $\alpha_n, \alpha'_n \neq 0$, we have $\alpha_n = \alpha'_n$. Hence α_n is identifiable.

If the sensor parameters are (α_n, β_n) , the mean of the LLS predictor of $\mathbf{y}_n(1)$ is $\alpha_n f_k(\mathbf{M}_{s_n(1)}(0)) + \beta_n$, whereas the mean is $\alpha'_n f_k(\mathbf{M}_{s_n(1)}(0)) + \beta'_n$ if the sensor parameters are (α'_n, β'_n) . Since $\alpha_n = \alpha'_n$, we obtain $\beta_n = \beta'_n$. The proof is now complete. ■

Proposition 1 shows that although sensors can switch clusters across time, we require that each sensor monitors a particular signal for at least 3 sampling time intervals. In practice, this requirement is easily satisfied since sampling intervals are typically very short. In our model, we will introduce a parameter to model the ‘stickiness’ of the sensor signals.

A similar identifiability problem is discussed in [28]–[30], where the evolution model is assumed to be linear but unknown. However, a specific canonical form is assumed.

III. DYNAMIC BAYESIAN NONPARAMETRIC MODEL

In this section, we propose a dynamic Bayesian nonparametric model to model the blind calibration problem described in (1) and (2). In the rest of this paper, we assume that the observation noise $\mathbf{w}_n(t)$ in (1) and the process noise $\epsilon_k(t)$ in

Fig. 1. Graphical representation of the dynamic Bayesian nonparametric model for our blind calibration problem.

(2) are Gaussian distributed with zero mean and known covariance matrices $\mathbf{R}_n(t)$ and $\mathbf{Q}_k(t)$, respectively. Fig. 1 shows the graphical representation of our model. We let $\alpha = (\alpha_n)_{n=1}^N$, $\beta = (\beta_n)_{n=1}^N$, $\theta(t) = (\theta_k(t))_{k=1}^\infty$, $\mathbf{s}(t) = (s_n(t))_{n=1}^N$, $\mathbf{y}(t) = (\mathbf{y}_n(t))_{n=1}^N$, and $\mathbf{R}(t) = (\mathbf{R}_n(t))_{n=1}^N$.

The priors of α_n and β_n are

$$\alpha_n \sim \mathcal{TN}(\mu_{1,n}, \Sigma_{1,n}), \quad \beta_n \sim \mathcal{N}(\mu_{2,n}, \Sigma_{2,n}), \quad (8)$$

where $\mu_{1,n}, \sigma_{1,n}, \mu_{2,n}, \sigma_{2,n}^2$ are known hyper-parameters. In this work, we assume the gains are positive and follow a truncated Gaussian distribution $\mathcal{TN}(\mu_{1,n}, \Sigma_{1,n})$ [31], where $\mu_{1,n}$ and $\Sigma_{1,n}$ are the mean and variance of the Gaussian distribution before truncation. We note that Gaussian priors for the gains and offset are widely used in the literature of sensor calibration problem; see [10], [32], [33]. The signal $\theta_k(0)$ is also assumed to be Gaussian distributed with mean $\mathbf{M}_k(0)$ and covariance $\mathbf{V}_k(0)$.

At each time t , sensor observations due to the same signal $\theta_k(t)$ can be clustered into the same group. Let $s_n(t) = k$ indicate that the n -th sensor observation is in the k -th cluster, and $\pi_{n,k}(t)$ be the probability of $s_n(t) = k$, i.e.,

$$s_n(t) \mid \pi(t) \sim \pi_n(t), \quad (9)$$

where $\pi_n(t) = (\pi_{n,k}(t))_{k \geq 1}$. We let $\pi_n(t)$ be generated according to a Dirichlet Process, i.e., $\pi_n(t) \sim \text{GEM}(\gamma)$, where GEM stands for the Griffiths, Engen and McCloskey stick-breaking process and γ is a concentration hyperparameter [24]. However, sampling $\pi_n(t)$ is difficult as $\pi_n(t)$ has infinite dimension. Following [34], [35], we approximate the Dirichlet process with its K -degree weak limit given by

$$\pi_n(t) \sim \text{Dir}(\gamma/K, \dots, \gamma/K), \quad (10)$$

where $\text{Dir}(\cdot)$ is the Dirichlet distribution.

As shown in Proposition 1, a certain ‘stickiness’ in the signal index is required for identifiability. In practice, the sensors tend to monitor the same signal over a period of time. We introduce a parameter κ in the Dirichlet distribution such that an amount of ‘stickiness’ can be preserved between $\mathbf{s}(t-1)$ and $\mathbf{s}(t)$. We use the following model:

$$\pi_n(t) \mid s_n(t-1), \gamma, \kappa \sim \text{Dir}(p_1(t), \dots, p_K(t)), \quad (11)$$

where

$$p_k(t) = \begin{cases} \gamma/K + \kappa & \text{if } s_n(t-1) = k, \\ \gamma/K & \text{otherwise,} \end{cases}$$

for all $k \in \{1, 2, \dots, K\}$. Note that κ is a parameter that controls the probability of a sensor changing to a different cluster at time t . The larger the κ , the less likely a sensor switches its cluster.

In summary, for any time step $t \geq 1$, our generative model can be summarized as follows:

$$s_n(t) \mid \pi(t) \sim \pi_n(t), \quad (12)$$

$$\pi_n(t) \mid s_n(t-1), \gamma, \kappa \sim \text{Dir}(p_1(t), \dots, p_K(t)), \quad (13)$$

$$\theta_k(t) \mid \theta_k(t-1) \sim \mathcal{N}(f_k(\theta_k(t-1)), \mathbf{Q}_k(t)), \quad (14)$$

$$\mathbf{y}_n(t) \mid \theta_k(t), s_n(t) = k, \alpha_n, \beta_n \sim \mathcal{N}(\alpha_n \theta_k(t) + \beta_n, \mathbf{R}_n(t)). \quad (15)$$

IV. OFFLINE INFERENCE ALGORITHM

In this section, we present an offline inference algorithm for the Bayesian nonparametric model in Section III. As closed-form solutions for the posterior distributions of some of the latent variables and parameters are not readily available, we adopt a block Gibbs sampling method, which uses a number of samples to approximate the distributions of the parameters θ, α, β . Conditioning on these parameters, an MCMC inference method based on forward filtering backward sampling (MCMC-FFBS) is applied to estimate the signal parameters. The top-level implementation of the offline inference algorithm for the i -th iteration is given in Algorithm 1, where we use the superscript i to indicate that a quantity is the sample from the i -th iteration. Algorithm 1 is then repeated for a sufficiently large number of iterations to achieve good mixing [23]. The detailed procedure for drawing samples are discussed in next subsections.

Algorithm 1 MCMC-FFBS (i -th iteration)

Input: $\alpha^{i-1}, \beta^{i-1}, \theta^{i-1}, \pi^{i-1}, \mathbf{s}^{i-1}$ from $i-1$ -th iteration and data $\mathbf{y} = (\mathbf{y}(t))_{t=1}^T$.

Output: new set of samples: $\alpha^i, \beta^i, \theta^i, \pi^i, \mathbf{s}^i$

for $t = 1$ to T **do**

 Sample $\mathbf{s}^i(t)$ using (16) and $\pi^i(t)$ using (17).

end for

Block sample $\theta^i(1:T)$ using (18).

Sample α^i using (22).

Sample β^i using (23).

return $\alpha^i, \beta^i, \theta^i, \pi^i$ and $\mathbf{s}^i$.

A. Sampling $\pi^i(t)$ and $\mathbf{s}^i(t)$

At the i -th iteration, we sample $s_n(t)$ using the following posterior distribution:

$$\begin{aligned} p(s_n(t) = k \mid \mathbf{y}_n(t), \pi_n(t), \pi_n(t+1), \theta_k(t), \alpha_n, \beta_n) \\ \propto p(s_n(t) = k \mid \pi_n(t)) p(\pi_n(t+1) \mid s_n(t)) \\ \cdot p(\mathbf{y}_n(t) \mid s_n(t) = k, \theta_k(t), \alpha_n, \beta_n) \\ = \pi_{n,k}(t) p(\pi_n(t+1) \mid s_n(t)) p(\mathbf{y}_n(t) \mid \theta_k(t), \alpha_n, \beta_n), \end{aligned} \quad (16)$$

where $\pi_n(t)$, $\pi_n(t+1)$, $\theta_k(t)$, α_n and β_n are set to their respective sample values at the $(i-1)$ -th iteration, and $p(\mathbf{y}_n(t) \mid \theta_k(t), \alpha_n, \beta_n)$ and $p(\pi_n(t+1) \mid s_n(t))$ are obtained from (1) and (11) respectively.

Let $l_{n,k}(t) \triangleq \sum_{n=1}^N \delta(s_n(t) - k)$ with $\delta(\cdot)$ representing the Dirac delta function. Because the Dirichlet distribution prior of $\pi_n(t)$ (11) is conjugate to the categorical distribution likelihood (9), the posterior distribution is a Dirichlet distribution with updated parameters:

$$\pi_n(t) \mid \mathbf{s}(t), s_n(t-1), \gamma, \kappa \sim \text{Dir}(p'_1(t), \dots, p'_K(t)), \quad (17)$$

where

$$p'_k(t) = \begin{cases} \gamma/K + l_{n,k}(t) + \kappa & \text{if } s_n(t-1) = k, \\ \gamma/K + l_{n,k}(t) & \text{otherwise.} \end{cases}$$

We then sample $\pi_n(t)$ according to (17). It can be observed from (17) that the mixture weight is determined by the Dirichlet concentration parameter γ , the stickiness parameter κ , and the sensor association to each cluster.

B. Sampling $\theta^i(1:T)$

We sample $\theta^i(1:T)$ using the forward filtering backward sampling method. Let $\mathbf{y}_{|k}(t) \triangleq \{\mathbf{y}_n(t) : s_n(t) = k\}$, $\alpha_{|k}(t) \triangleq \{\alpha_n(t) : s_n(t) = k\}$, $\beta_{|k}(t) \triangleq \{\beta_n(t) : s_n(t) = k\}$, $\mathbf{R}_{|k}(t) \triangleq \text{diag}(\{\mathbf{R}_n(t) : s_n(t) = k\})$, and $\psi_{|k} \triangleq \{\psi_{|k}(t)\}_{t=1}^T$ where $\psi_{|k}(t) \triangleq \{\alpha_{|k}(t), \beta_{|k}(t), \epsilon_k, \mathbf{R}_{|k}(t), \mathbf{A}_k(t)\}$, and

$$\mathbf{V}_k(t) = (\alpha_{|k}(t)^T \mathbf{R}_{|k}(t)^{-1} \alpha_{|k}(t) + \mathbf{b}_k(t)^{-1})^{-1},$$

$$\begin{aligned} \mathbf{M}_k(t) = \mathbf{V}_k(t) \left(-\alpha_{|k}(t)^T \mathbf{R}_{|k}(t)^{-1} \beta_{|k}(t) \right. \\ \left. + \alpha_{|k}(t)^T \mathbf{R}_{|k}(t)^{-1} \mathbf{y}_{|k}(t) + \mathbf{a}_k(t) \mathbf{b}_k(t)^{-1} \right), \end{aligned}$$

$$\mathbf{a}_k(t) = f_k(\mathbf{M}_k(t-1)),$$

$$\mathbf{b}_k(t) = \mathbf{A}_k(t-1) \mathbf{V}_k(t-1) \mathbf{A}_k(t-1)^T + \mathbf{Q}_k(t-1),$$

$$\mathbf{A}_k(t) = \text{Df}_k(\mathbf{M}_k(t-1)).$$

Set $\alpha_{|k}(t)$ and $\beta_{|k}(t)$ to their respective sample values from the $(i-1)$ -th iteration. Then,

$$\begin{aligned} p(\theta_k(1:T) \mid \mathbf{y}_{|k}(1:T), \psi_{|k}) \\ = p(\theta_k(T) \mid \mathbf{y}_{|k}(1:T), \psi_{|k}) \\ \cdot \prod_{t=1}^{T-1} p(\theta_k(t) \mid \theta_k(t+1), \mathbf{y}_{|k}(1:t), \psi_{|k}), \end{aligned} \quad (18)$$

where

$$\begin{aligned} p(\theta_k(t) \mid \theta_k(t+1), \mathbf{y}_{|k}(1:t), \psi_{|k}) \\ \propto p(\theta_k(t+1) \mid \theta_k(t), \psi_{|k}) p(\theta_k(t) \mid \mathbf{y}_{|k}(1:t), \psi_{|k}). \end{aligned} \quad (19)$$

To sample from (18), we need to first obtain $p(\theta_k(t) \mid \mathbf{y}_{|k}(1:t), \psi_{|k})$, where $t \in \{1, 2, \dots, T\}$. Adopting the extended Kalman filter approach [27], we approximate

$$\theta_k(t) \mid \mathbf{y}_{|k}(1:t), \psi_{|k} \sim \mathcal{N}(\mathbf{M}_k(t), \mathbf{V}_k(t)), \quad (20)$$

Using (20) with t replaced by T , we can obtain $p(\theta_k(T) \mid \mathbf{y}_{|k}(1:T), \psi_{|k})$, from which $\theta_k(T)$ can be sampled.

From (19), (20), and the conjugacy of the Gaussian prior, we have

$$p(\boldsymbol{\theta}_k(t) \mid \boldsymbol{\theta}_k(t+1), \mathbf{y}_{|k}(1:t), \boldsymbol{\psi}_{|k}) \sim \mathcal{N}(\mathbf{h}_k(t), \mathbf{H}_k(t)), \quad (21)$$

where

$$\begin{aligned} \mathbf{B}_k(t) &= \mathbf{V}_k(t) \mathbf{A}_k(t) (\mathbf{b}_k(t+1))^{-1}, \\ \mathbf{H}_k(t) &= \mathbf{V}_k(t) - \mathbf{B}_k(t) \mathbf{b}_k(t+1) \mathbf{B}_k(t)^T, \\ \mathbf{h}_k(t) &= \mathbf{M}_k(t) + \mathbf{B}_k(t) (\boldsymbol{\theta}_k(t+1) - \mathbf{a}_k(t+1)). \end{aligned}$$

Hence, conditioned on the sample for $\boldsymbol{\theta}_k(t+1)$, $\boldsymbol{\theta}_k(t)$ can be sampled according to (21), starting from $t = T - 1$.

C. Sampling α^i and β^i

Let $\phi_n(t) \triangleq \boldsymbol{\theta}_{s_n(t)}(t)$. The posterior distribution of α_n is given by

$$\begin{aligned} & p(\alpha_n \mid \beta_n, \mathbf{y}_n(1:T), \phi_n(1:T), \mathbf{R}_n(t)) \\ & \propto p(\mathbf{y}_n(1:T) \mid \alpha_n, \beta_n, \phi_n(1:T), \mathbf{R}_n(t)) p(\alpha_n) \\ & = \prod_{t=1}^T p(\mathbf{y}_n(t) \mid \alpha_n, \beta_n, \phi_n(t), \mathbf{R}_n(t)) p(\alpha_n). \end{aligned}$$

where β is set to their respective sample values from the $(i-1)$ -th iteration. From (1) and (8), we obtain the following distribution:

$$\alpha_n \mid \beta_n, \mathbf{y}_n(1:T), \phi_n(1:T), \mathbf{R}_n(t) \sim \mathcal{TN}(m_n, v_n), \quad (22)$$

where

$$\begin{aligned} v_n &= \left[\sum_{t=1}^T \phi_n(t)^T \mathbf{R}_n(t)^{-1} \phi_n(t) + \frac{1}{\Sigma_{1,n}} \right]^{-1}, \\ m_n &= v_n \left[\sum_{t=1}^T \mathbf{y}_n(t)^T \mathbf{R}_n(t)^{-1} \phi_n(t) + \frac{\mu_{1,n}}{\Sigma_{1,n}} \right. \\ & \quad \left. - \sum_{t=1}^T \beta_n^T \mathbf{R}_n(t)^{-1} \phi_n(t) \right]. \end{aligned}$$

Similarly,

$$\beta_n \mid \alpha_n, \mathbf{y}_n(1:T), \phi_n(1:T), \mathbf{R}_n(t) \sim \mathcal{N}(\mathbf{T}_n, \mathbf{G}_n), \quad (23)$$

where

$$\begin{aligned} \mathbf{G}_n &= \left[\sum_{t=1}^T \mathbf{R}_n(t)^{-1} + \Sigma_{2,n}^{-1} \right]^{-1}, \\ \mathbf{T}_n &= \mathbf{G}_n \left[\sum_{t=1}^T \mathbf{y}_n(t)^T \mathbf{R}_n(t)^{-1} + \boldsymbol{\mu}_{2,n}^T \Sigma_{2,n}^{-1} \right. \\ & \quad \left. - \sum_{t=1}^T \alpha_n \phi_n(t)^T \mathbf{R}_n(t)^{-1} \right]. \end{aligned}$$

V. ONLINE INFERENCE ALGORITHM

In this section, we develop an online inference algorithm that allows us to perform sequential inference as more sensor observations become available. We then discuss the advantages and disadvantages of the online inference approach versus the offline approach in Section IV.

Since our model is nonlinear with respect to all the variables we need to infer, we can not apply Kalman filter directly to perform online inference. Instead, we use the particle filter (PF) and local MCMC (PF-MCMC) approach. The PF uses point mass to approximate the probability distribution of the states and updates the target distributions sequentially. MCMC is then used to draw samples from the target joint distribution. Compared with the aforementioned MCMC-FFBS method, we need much less iterations to draw samples from target distribution, which is important for online data processing. Our proposed approach is summarized in Algorithm 2. The subscript l refers to the l -th particle, while the superscript i refers to the i -th iteration in the MCMC step. We provide the detailed descriptions in the following subsections.

Algorithm 2 PF-MCMC (Time instant t)

Input: Observation at time t , $\mathbf{y}(t)$; and L particles obtained at time $t-1$, $\{\mathbf{s}_l(t-1), \boldsymbol{\theta}_l(t-1), \boldsymbol{\alpha}_l, \beta_l, \boldsymbol{\pi}_l(t-1)\}_{l=1}^L$.

Output: New particles at t , $\{\mathbf{s}_l(t), \boldsymbol{\theta}_l(t), \boldsymbol{\alpha}_l, \beta_l, \boldsymbol{\pi}_l(t)\}_{l=1}^L$.

for $l = 1 : L$ **do**

 Sample $\boldsymbol{\pi}_l(t)$ from $p(\boldsymbol{\pi}_l(t) \mid \mathbf{s}_l(t-1))$, see (27).

for $i = 1 : I$ **do**

 Sample $\mathbf{s}_l^i(t)$ from

$p(\mathbf{s}_l^i(t) \mid \boldsymbol{\alpha}_l^{i-1}, \beta_l^{i-1}, \boldsymbol{\theta}_l(t-1), \mathbf{y}(t), \boldsymbol{\pi}_l(t))$, see (29),

 Sample $\boldsymbol{\theta}_l^i(t)$ from

$p(\boldsymbol{\theta}_l^i(t) \mid \boldsymbol{\alpha}_l^{i-1}, \beta_l^{i-1}, \boldsymbol{\theta}_l(t-1), \mathbf{y}(t), \mathbf{s}_l^i(t))$, see (30).

 Sample $\boldsymbol{\alpha}_l^i$ and β_l^i from (24) and (25).

end for

 Update weight, see (31) for details.

end for

Calculate effective sample size $N_{\text{eff}} = \frac{1}{\sum_{l=1}^L (w_l(t))^2}$.

if $N_{\text{eff}} < 0.6L$ **then**

 Resample particles using a multinomial sampling procedure. (cf. Section V-B) to obtain $\{\mathbf{s}_l(t), \boldsymbol{\theta}_l(t), \boldsymbol{\alpha}_l, \beta_l, \boldsymbol{\pi}_l(t)\}_{l=1}^L$

else

 Set

$$\begin{aligned} & \{\mathbf{s}_l(t), \boldsymbol{\theta}_l(t), \boldsymbol{\alpha}_l, \beta_l, \boldsymbol{\pi}_l(t)\}_{l=1}^L \\ & = \{\mathbf{s}_l^I(t), \boldsymbol{\theta}_l^I(t), \boldsymbol{\alpha}_l^I, \beta_l^I, \boldsymbol{\pi}_l^I(t)\}_{l=1}^L \end{aligned}$$

end if

A. Particle sampling

We use Gibbs sampling to sample particles. The posterior distribution of $\boldsymbol{\alpha}_l$ and β_l can be sampled directly in a sequential way. The posterior distribution of $(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t), \boldsymbol{\pi}_l(t))$ is approximated by a proposal distribution, from which we draw samples.

1) *Sampling α_l and β_l* : α_l is sampled from the following distribution:

$$\begin{aligned}
& p(\alpha_l | \beta_l, \mathbf{y}(1:t), \mathbf{s}_l(1:t), \boldsymbol{\theta}_l(1:t)) \\
& \propto p(\mathbf{y}(1:t) | \alpha_l, \beta_l, \mathbf{s}_l(1:t), \boldsymbol{\theta}_l(1:t)) p(\alpha_l) \\
& = \prod_{t=1}^t p(\mathbf{y}(t) | \alpha_l, \beta_l, \mathbf{s}_l(t), \boldsymbol{\theta}_l(t)) p(\alpha_l) \\
& = p(\mathbf{y}(t) | \alpha_l, \beta_l, \mathbf{s}_l(t), \boldsymbol{\theta}_l(t)) \\
& \quad \cdot \prod_{t=1}^{t-1} p(\mathbf{y}(t) | \alpha_l, \beta_l, \mathbf{s}_l(t), \boldsymbol{\theta}_l(t)) p(\alpha_l) \\
& \propto p(\mathbf{y}(t) | \alpha_l, \beta_l, \mathbf{s}_l(t), \boldsymbol{\theta}_l(t)) \\
& \quad \cdot p(\alpha_l | \beta_l, \mathbf{y}(1:t-1), \mathbf{s}_l(1:t-1), \boldsymbol{\theta}_l(1:t-1)). \quad (24)
\end{aligned}$$

The first term in the product in (24) comes from (15). Equation (24) thus provides a sequential scheme to update the posterior distribution $p(\alpha_l | \beta_l, \mathbf{y}(1:t), \mathbf{s}_l(1:t), \boldsymbol{\theta}_l(1:t))$, where at time $t = 1$, we set $p(\alpha_l | \beta_l, \mathbf{y}(1), \mathbf{s}_l(1), \boldsymbol{\theta}_l(1)) \propto p(\mathbf{y}(1) | \alpha_l, \beta_l, \mathbf{s}_l(1), \boldsymbol{\theta}_l(1)) p(\alpha_l)$.

Similarly, we sample β from the following distribution

$$\begin{aligned}
& p(\beta_l | \alpha_l, \mathbf{y}(1:t), \mathbf{s}_l(1:t), \boldsymbol{\theta}_l(1:t)) \\
& \propto p(\mathbf{y}(t) | \alpha_l, \beta_l, \mathbf{s}_l(t), \boldsymbol{\theta}_l(t)) \\
& \quad \cdot p(\beta_l | \alpha_l, \mathbf{y}(1:t-1), \mathbf{s}_l(1:t-1), \boldsymbol{\theta}_l(1:t-1)). \quad (25)
\end{aligned}$$

2) *Sampling $\boldsymbol{\theta}_l(t)$, $\mathbf{s}_l(t)$, and $\pi_l(t)$* : The posterior distribution of $(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t), \pi_l(t))$ can not be directly sampled in a sequential way, but it can be approximated by a proposal distribution that can be sampled sequentially. The proposal distribution q is the probability of assignments $\mathbf{s}_l(t)$, parameters $\boldsymbol{\theta}_l(t)$, and component weight $\boldsymbol{\theta}_l(t)$ given previous assignments $\mathbf{s}_l(t-1)$, parameters $\boldsymbol{\theta}_l(t-1)$, gains α_l , offsets β_l and observations $\mathbf{y}(t)$, where

$$\begin{aligned}
& q(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t), \pi_l(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{s}_l(t-1), \mathbf{y}(t)) \\
& = q(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{y}(t), \pi_l(t)) \\
& \quad \cdot q(\pi_l(t) | \alpha_l, \beta_l, \mathbf{s}_l(t-1), \mathbf{y}(t)).
\end{aligned}$$

The particle filter method provides some choices of proposal distribution [16]. Here we choose the posterior distribution of $(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t))$ as their proposal distribution, with

$$\begin{aligned}
& q(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{y}(t), \pi_l(t)) \\
& \triangleq p(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{y}(t), \pi_l(t)), \quad (26)
\end{aligned}$$

and choose the prior distribution of $\pi_l(t)$ as its proposal distribution, i.e.,

$$q(\pi_l(t) | \alpha_l, \beta_l, \mathbf{s}_l(t-1), \mathbf{y}(t)) \triangleq p(\pi_l(t) | \mathbf{s}_l(t-1)). \quad (27)$$

Then we can sample $\pi_l(t)$ from $p(\pi_l(t) | \mathbf{s}_l(t-1))$ and sample $\boldsymbol{\theta}_l(t), \mathbf{s}_l(t)$ from

$$\begin{aligned}
& p(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{y}(t), \pi_l(t)) \\
& = p(\boldsymbol{\theta}_l(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{y}(t), \mathbf{s}_l(t)) \\
& \quad \cdot p(\mathbf{s}_l(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{y}(t), \pi_l(t)), \quad (28)
\end{aligned}$$

where

$$\begin{aligned}
& p(\mathbf{s}_l(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{y}(t), \pi_l(t)) \\
& \propto p(\mathbf{s}_l(t) | \pi_l(t)) p(\mathbf{y}(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{s}_l(t)) \\
& = p(\mathbf{s}_l(t) | \pi_l(t)) \\
& \quad \cdot \int p(\mathbf{y}(t) | \mathbf{s}_l(t), \boldsymbol{\theta}_l(t), \alpha_l, \beta_l) p(\boldsymbol{\theta}_l(t) | \boldsymbol{\theta}_l(t-1)) d\boldsymbol{\theta}_l(t), \quad (29)
\end{aligned}$$

and

$$\begin{aligned}
& p(\boldsymbol{\theta}_l(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{y}(t), \mathbf{s}_l(t)) \\
& \propto p(\boldsymbol{\theta}_l(t) | \boldsymbol{\theta}_l(t-1)) p(\mathbf{y}(t) | \alpha_l, \beta_l, \mathbf{s}_l(t), \boldsymbol{\theta}_l(t)). \quad (30)
\end{aligned}$$

B. Particle weights

A common problem of particle filtering is the degeneracy phenomenon, where all but one particle have negligible weights after a few iterations. In order to eliminate this problem, we resample by deleting particles having small weights. In our model, the weight of particle l is defined as:

$$\begin{aligned}
& w_l(t) \\
& = \frac{p(\boldsymbol{\theta}_l(1:t), \mathbf{s}_l(1:t), \alpha_l, \beta_l, \pi_l(1:t), \mathbf{y}(1:t))}{q(\boldsymbol{\theta}_l(1:t), \mathbf{s}_l(1:t), \alpha_l, \beta_l, \pi_l(1:t) | \mathbf{y}(1:t))} \\
& = w_l(t-1) \frac{p(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t), \mathbf{y}(t), \pi_l(t) | \boldsymbol{\theta}_l(t-1), \mathbf{s}_l(t-1), \alpha_l, \beta_l)}{q(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t), \pi_l(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{s}_l(t-1), \mathbf{y}(t))} \\
& = w_l(t-1) \frac{p(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t), \mathbf{y}(t) | \boldsymbol{\theta}_l(t-1), \alpha_l, \beta_l, \pi_l(t))}{q(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{y}(t), \pi_l(t))} \\
& \quad \cdot \frac{p(\pi_l(t) | \mathbf{s}_l(t-1))}{q(\pi_l(t) | \alpha_l, \beta_l, \mathbf{s}_l(t-1), \mathbf{y}(t))} \\
& = w_l(t-1) \frac{p(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t), \mathbf{y}(t) | \boldsymbol{\theta}_l(t-1), \alpha_l, \beta_l, \pi_l(t))}{p(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{y}(t), \pi_l(t))}.
\end{aligned}$$

The last equality follows from (26) and (27). Furthermore, we have

$$\begin{aligned}
& \frac{p(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t), \mathbf{y}(t) | \boldsymbol{\theta}_l(t-1), \alpha_l, \beta_l, \pi_l(t))}{p(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t) | \boldsymbol{\theta}_l(t-1), \alpha_l, \beta_l, \pi_l(t), \mathbf{y}(t))} \\
& = \frac{p(\mathbf{y}(t) | \boldsymbol{\theta}_l(t), \mathbf{s}_l(t), \alpha_l, \beta_l)}{p(\boldsymbol{\theta}_l(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{s}_l(t), \mathbf{y}(t))} \\
& \quad \cdot \frac{p(\boldsymbol{\theta}_l(t), \mathbf{s}_l(t) | \boldsymbol{\theta}_l(t-1), \alpha_l, \beta_l, \pi_l(t))}{p(\mathbf{s}_l(t) | \boldsymbol{\theta}_l(t-1), \alpha_l, \beta_l, \pi_l(t), \mathbf{y}(t))} \\
& = \frac{p(\mathbf{y}(t) | \boldsymbol{\theta}_l(t), \mathbf{s}_l(t), \alpha_l, \beta_l)}{p(\boldsymbol{\theta}_l(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{s}_l(t), \mathbf{y}(t))} \\
& \quad \cdot \frac{p(\boldsymbol{\theta}_l(t) | \boldsymbol{\theta}_l(t-1)) p(\mathbf{s}_l(t) | \pi_l(t))}{p(\mathbf{s}_l(t) | \boldsymbol{\theta}_l(t-1), \alpha_l, \beta_l, \pi_l(t), \mathbf{y}(t))}. \quad (31)
\end{aligned}$$

The terms in the numerator $p(\mathbf{y}(t) | \boldsymbol{\theta}_l(t), \mathbf{s}_l(t), \alpha_l, \beta_l)$, $p(\boldsymbol{\theta}_l(t) | \boldsymbol{\theta}_l(t-1))$ and $p(\mathbf{s}_l(t) | \pi_l(t))$ are respectively the output distribution obtained from (1), the transition distribution obtained from (2), and the prior of cluster index (9). The terms in the denominator $p(\boldsymbol{\theta}_l(t) | \alpha_l, \beta_l, \boldsymbol{\theta}_l(t-1), \mathbf{s}_l(t), \mathbf{y}(t))$ and $p(\mathbf{s}_l(t) | \boldsymbol{\theta}_l(t-1), \alpha_l, \beta_l, \pi_l(t), \mathbf{y}(t))$ can be obtained from (30) and (29), respectively.

After the particle weights are computed, we normalize each weight to obtain the normalized weight $\tilde{w}_l(t) = w_l(t) / \sum_k w_k(t)$ for particle l . Then, we resample by drawing L particles using the multinomial distribution with weights $(\tilde{w}_l(t))_{l=1}^L$ [36].

C. Comparison with offline inference method

As both PF-MCMC and MCMC-FFBS are sampling based inference methods, computing the posterior distributions at each step dominates the time complexity of the procedure. Consider a sensor network with K clusters and N sensors. Suppose we perform our online PF-MCMC method in Algorithm 2 using L particles and I local Gibbs sampling iterations over T observation time instances. The number of posterior distribution calculation of each local Gibbs iteration and each particle weight computation are of order $NK + K + 2N$ and $NK + 2N$, respectively. The complexity of PF-MCMC is thus $\mathcal{O}(TLI(2NK + K + 4N)) = \mathcal{O}(LITNK)$.¹

For the offline MCMC-FFBS method, the number of operations incurred to compute the posterior distributions for sampling $\alpha, \beta, \theta, \pi$, and s in each iteration are of order N, N, TK, TK , and TNK , respectively. Thus, the complexity for each iteration is $\mathcal{O}(TNK + TK + TK + 2N) = \mathcal{O}(TNK)$, and with I' iterations of Gibbs sampling, the overall complexity of MCMC-FFBS is $\mathcal{O}(I'TNK)$. We see that the time complexity of MCMC-FFBS and PF-MCMC are related to the number of particles and number of iterations in the Gibbs sampling procedure.

The main advantage of the online PF-MCMC method is that data can be processed sequentially and parameter estimates can be updated in real time as more observations become available. Furthermore, we do not need to store the observation data over T time instances, which is required for the offline MCMC-FFBS method. However, these advantages are achieved at the cost of estimation accuracy. The offline Gibbs sampling method in MCMC-FFBS collects all the data and estimate the ground truth signal with both filtering and smoothing. By conditioning on both past and future data, the variances of the smoothed estimates are reduced [27].

VI. SIMULATION AND EXPERIMENT RESULTS

In this section, we present simulations and real data experiments to evaluate our proposed offline MCMC-FFBS and online PF-MCMC methods. To the best of our knowledge, there are no other existing blind calibration algorithms for our considered setting, where different sensors may switch signals across time, and the signals are time-varying. Therefore, for comparison, we adapt some state-of-the-art signal clustering methods to first perform sensor clustering, and then use MCMC to estimate gains and offsets. As a baseline comparison, we also compare against the case where the sensor clusters are known, which is expected to give the best performance. We conducted 100 simulations for all the methods and the average square error (ASE) is used to measure the accuracy. The ASE is defined as the square deviation of an estimated parameter from its true value, averaged over all the sensors.

A. Sensors remain in same clusters

We consider the case where each sensor observes the same signal over all time instances, i.e., the sensors do not switch

to different clusters. We simulate $N = 40$ sensors observing 4 signals. At $t = 0$, the signals are chosen to be $\theta_1(0) \sim \mathcal{N}(30, 1)$, $\theta_2(0) \sim \mathcal{N}(20, 1)$, $\theta_3(0) \sim \mathcal{N}(10, 1)$ and $\theta_4(0) \sim \mathcal{N}(-10, 1)$. The observation time period is $T = 40$. We use the following model to generate the simulation data:

$$\begin{aligned} \alpha_n &\sim \mathcal{TN}(1, \sigma_1^2), \beta_n \sim \mathcal{N}(0, \sigma_2^2), \\ y_n(t) &= \alpha_n \theta_{s_n(t)}(t) + \beta_n + \omega_n(t), \\ \omega_n(t) &\sim \mathcal{N}(0, R_n(t)), R_n(t) \sim \mathcal{N}(0.3, 0.1), \\ \theta_k(t) | \theta_k(t-1) &\sim \mathcal{N}(a_k \sin(\theta_k(t-1)) + \theta_k(t-1), Q_k), \\ (a_k)_{k=1}^4 &= (2, 3, 4, 5), \\ Q_k &\sim \mathcal{N}(0, 1), \text{ for all } k = 1, \dots, 4 \text{ and } t = 1, \dots, T. \end{aligned}$$

Six different values of (σ_1^2, σ_2^2) are chosen to generate simulation data, namely $(0.14^2, 1.4^2)$, $(0.13^2, 1.3^2)$, $(0.12^2, 1.2^2)$, $(0.11^2, 1.1^2)$, $(0.1^2, 1)$, and $(0.09^2, 0.9^2)$.

In both the offline MCMC-FFBS method and the online PF-MCMC method, we set the following parameters: $K = 10$, $\alpha_n = \mu_{1,n} = 1$ and $\beta_n = \mu_{2,n} = 0$ for all n , $\{M_k(0)\}_{k=1}^K = \{55, 45, 30, 20, 10, 0, -10, -25, -35, -45\}$, and assume that $(a_k)_{k=1}^K = (2, 3, 4, 5, 6, 7, 8, 9, 10, 11)$. In the PF-MCMC method, the number of particles is set to $L = 100$.

We compare the performance of our proposed approaches with the following methods:

- 1) The gain and offset of each sensor are estimated according to its own observations using MCMC (MCMC-single).
- 2) The sensor clusters are assumed to be known a priori, and the sensor gains and offsets are estimated using MCMC with correctly clustered sensor observations (MCMC-cluster). This benchmark is expected to give the best ASE performance.
- 3) The number of sensor clusters is assumed to be known a priori. We apply a k -means method to cluster the sensor observations, and then use MCMC to estimate the sensor gains and offsets.
- 4) We use the affinity propagation (AP) method [37] to perform the clustering and then MCMC to estimate the sensor gains and offsets.
- 5) The T-SBL method by [7]. The sensors are assumed to have unit gains and no offsets during an initial calibration period, and a subset of sensors are assumed to start drifting at some known time. Sensor observations from the initial calibration period are used to estimate the signal subspace in which the sensor signals lie, and this subspace is assumed to be invariant over time. The sensor drifts are then estimated using a sparse Bayesian learning approach.

Among the methods in the literature on blind calibration, the assumptions adopted in [8] and [7] are closest to ours. The reference [7] is an extension of [8] with better performance. Therefore, we only compare our approaches with that proposed in [7], in which the drift of sensor n is equivalent to $(\alpha_n - 1)\theta_{s_n(t)}(t) + \beta_n$ in our model. Hence, we use T-SBL to estimate the drift of sensor n first and then use the ground truth values of $\theta_{s_n(t)}(t)$ and β_n to obtain α_n . As [7] requires an initial calibration period, we simulate an initial calibration

¹We say that $f(n) = \mathcal{O}(g(n))$ if $\limsup_{n \rightarrow \infty} f(n)/g(n) < \infty$.

period consisting of 400 samples only for T-SBL, but not the other methods. After the initial calibration period, we let 50% of the sensors drift over time for T-SBL, while 100% of the sensors drift over time for the other methods.

Fig. 2. Mean ASE of different methods for sensor gain estimation when sensors remain in same cluster over time.

In Fig. 2, we plot the mean ASE of the estimated sensor gains of different methods versus different prior variances for the sensor gains. It can be observed that given the correct clustering of the sensor observations, the estimation accuracy can be significantly improved. The results also show that our proposed methods outperform the methods that first perform clustering and then calibration, as well as the T-SBL method. Both the k -means and AP regard the time-varying observations as different vectors and the clustering is done using these vectors, while our proposed methods are better able to model the time-varying behavior of the observations and capture the evolutionary information of the sensor signals. Unlike our approaches, the T-SBL approach does not utilize the evolutionary information of the underlying sensor signals.

Box-plots of the simulation results are shown in Fig. 3. It can be observed that the proposed offline MCMC-FFBS method performs better than the k -means, AP, and T-SBL methods in both median ASE and number of ASE outliers. The online PF-MCMC method has less outliers, but higher median ASE compared to the offline methods. This is because the offline methods are based on batch processing and use all available data to estimate the parameters. Note that for the k -means method, the cluster number is assumed to be known and the ground truth of the gains and offsets are employed for the initialization.

Compared with the T-SBL method in [7], the advantages of our methods are thus: (i) they do not require any calibration period, (ii) all the sensors can be low-cost sensors that drift over time, and (iii) they can estimate the sensor gains and offsets even if the subspace in which the sensor signals lie in change over time. However, our methods require that the signals' evolution models to be known.

Fig. 3. Box-plots of ASE when sensors do not switch clusters. Numerical value indicated is the median.

B. The influence of the parameter κ

In many real applications, the underlying signals tend to remain the same or change slowly over observation time instances. The parameter κ introduced in (11) is used to characterize the stickiness of a sensor to a particular signal. In order to show the impact of κ , the performance of the proposed online PF-MCMC method with $\kappa = 0$ and $\kappa = 10$ are compared in Table I. It can be observed that the performance when $\kappa = 10$ outperforms that when $\kappa = 0$. In general, larger κ can better characterize the *sticky* behavior of the underlying signals. In practice, κ is experimentally selected according to different applications.

TABLE I
THE EFFECT OF κ ON MEAN ASE.

Variance	Mean ASE when $\kappa = 0$	Mean ASE when $\kappa = 10$
0.9^2	5.8×10^{-3}	5.5×10^{-3}
1.0^2	7.9×10^{-3}	7.4×10^{-3}
1.1^2	8.4×10^{-3}	7.5×10^{-3}
1.2^2	9.1×10^{-3}	8.3×10^{-3}
1.3^2	11.0×10^{-3}	10.0×10^{-3}
1.4^2	11.5×10^{-3}	10.4×10^{-3}

C. Sensors switch clusters across time

In this subsection, we evaluate the performance of our proposed methods when sensors switch clusters over time. Note that the cluster change points are unknown a priori. In our simulation, $N = 40$ sensors are deployed to observe 4 signals.

The observation time period is $T = 40$. Each sensor changes its observed signal randomly every 10 sampling intervals. The same initialization and distributions given in VI-A are employed to generate the observations.

Similar to the case that sensors remain in the same clusters over time, we also compare MCMC-FFBS and PF-MCMC with the MCMC-single, MCMC-cluster, k -means and AP based methods. We assume that the change points are known a priori for these latter methods. Given the known change points, the MCMC-single, MCMC-cluster, the k -means and AP based methods are implemented for unchanged data segments and the error is computed by averaging the results over all segments. We do not compare with the T-SBL method because it assumes that sensors monitor the same signal throughout. The ASE of the different methods are shown in Fig. 4. The results show that the performance of PF-MCMC is worse than MCMC-FFBS but better than both AP and k -means based methods. The performance of MCMC-FFBS is close to the MCMC-cluster, which is the best performance we can achieve.

Fig. 4. Comparison of different methods for sensor gain estimation when sensors switch clusters across time.

Fig. 5 shows the box-plots of the ASE due to MCMC-FFBS, PF-MCMC, k -means and AP. It can be observed that MCMC-FFBS performs the best, and the estimation accuracy of PF-MCMC is better than the k -means and AP based methods.

D. Real data experiment 1

In this subsection, we test our algorithms on the Sensorscope dataset collected by LCAV [38]. The dataset was collected on the rock glacier located at 2500m on the top of Le Généri above Martigny in Switzerland, and consists of the temperature, solar radiation, wind speed and some other information. In this paper, we only use the temperature observations in four files named ‘genepi-meteo-6’, ‘genepi-meteo-17’, ‘genepi-meteo-18’, and ‘genepi-meteo-19’. Since we do not know the ground-truth temperature values, we treat the collected data as calibrated signals, and different gains and offsets are artificially added to produce the sensor observations. The same initialization and parameter distributions given in Section VI-C are adopted. In the dataset, each

Fig. 5. Median and distribution of ASE with different methods when sensors switch clusters across time.

sensor took 3000 temperature readings. These observations are downsampled to 40 observations, and each sensor switches its observed signal randomly every 10 samples. For the T-SBL method, we use the same settings as in Section VI-A, in which sensors remain in the same clusters throughout.

Following [39], we assume the temperature data follows a first order AR process:

$$\theta_k(t) = a_k \theta_k(t-1) + \epsilon_k(t), \quad (32)$$

where $\epsilon_k(t)$ is additive noise with zero mean and covariance Q_k . Both a_k and Q_k are constants, which we estimate from the dataset. For every signal k , we estimate a_k and Q_k from its data by performing AR fitting. Averaging over all the estimated a_k and Q_k , we obtain an average value of approximately 1 and 0.6, respectively. We then take these as the model parameter values in our algorithms. From our AR fitting, we also note that the R squared values for the 4 signals are 0.28, 0.25, 0.62, and 0.94, respectively. This indicates that at least 3 of the signals do not follow our model assumption (32) closely. As seen in Fig. 6, our proposed algorithms achieve better performance than the other methods, which demonstrate our model is robust to the mismatch in our model assumption (32).

E. Real data experiment 2

In our next experiment, we collected measurements from 10 high quality light sensors. We use two sensors as reference

Fig. 6. Comparison of different methods for sensor gain estimation.

sensors to provide the ground truth and perturbed the others by covering them with paper. The dataset we have collected is available at [40]. The data is collected in 3 different areas of a passageway. The differences of the light intensity values between areas are greater than those within the same area. In each area, we let the sensors move around together and all the observations collected correspond to the same signal. The perturbations were different in different areas, and thus the gains and offsets of each sensor were different in different areas. Using the ground truth observations from reference sensors and the noisy observations from each perturbed sensor, we estimate the gains and offsets (α_n, β_n) for each sensor n using least squares, and use these as the ground truth sensor parameters.

In our inference algorithms, using the dynamic light scattering model from [41], we model each light signal as following an AR process: $\theta_k(t) | \theta_k(t-1) \sim \mathcal{N}(a_k \theta_k(t-1), Q_k)$. We perform AR fitting on the reference observations and find $(a_k)_{k=1}^3$ and $(Q_k)_{k=1}^3$ to be $(0.995, 0.998, 1.000)$ and $(1.3, 1.9, 1.5)$, respectively. However, in applying our algorithms, the number of clusters is unknown. We assume the maximum number of clusters $K = 10$, $Q_k = 1.5$ for all k , and $(a_k)_{k=1}^K = (0.989, 0.992, 0.995, 0.998, 1.000, 1.003, 1.006, 1.009, 1.012, 1.015)$. The variance of the measurement noise of sensor n is estimated by calculating the variance of time series $(y_n(t) - \alpha_n \theta_{s_n(t)}(t) - \beta_n)_{t=1}^T$. For T-SBL, we use an initial calibration period consisting of 80 samples.

The mean ASEs of different methods are shown in Table II. We observe that MCMC-FFBS and PF-MCMC have better performance than AP, k -means, MCMC-single, and T-SBL.

VII. CONCLUSION

We have proposed a dynamic Bayesian nonparametric model for blind calibration of networked sensors. The sensor signals are assumed to be evolving according to a known model and noisy observations are collected. We provided

TABLE II
ASE OF DIFFERENT METHODS ON LIGHT DATA

Methods	ASE
MCMC-cluster	7.0×10^{-3}
MCMC-FFBS	7.6×10^{-3}
PF-MCMC	8.5×10^{-3}
AP	9.9×10^{-3}
k -means	9.8×10^{-3}
MCMC-single	11.5×10^{-3}
T-SBL	16.0×10^{-3}

sufficient conditions for the identifiability of the calibration parameters. We developed both offline and online inference methods based on Gibbs sampling and particle filtering, respectively. The proposed methods have the advantage that the number of underlying signals and the signals themselves need not be known a priori. Simulations and real data experiments demonstrate that our proposed methods perform better than several other methods, including those which first perform sensor clustering and then parameter estimation, and a method based on sparse Bayesian learning.

For future research, it would be interesting to consider the case where sensor observations may be correlated with each other. Additional information like the sensor network topology or sensor locations may be incorporated into our model and algorithms to improve estimation accuracy.

ACKNOWLEDGMENT

This research is supported in part by the Singapore Ministry of Education Academic Research Fund Tier 2 grant MOE2014-T2-1-028, and the Delta-NTU Corporate Lab for Cyber-Physical Systems with funding support from Delta Electronics Inc and the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme. We thank Yuzhi Wang of [7] for supplying the source codes for T-SBL.

REFERENCES

- [1] L. Balzano and R. Nowak, "Blind calibration of sensor networks," in *ACM/IEEE Int. Conf. Inform. Process. Sensor Networks*, 2007, pp. 79–88.
- [2] S. Guo, H. Zhang, Z. Zhong, J. Chen, Q. Cao, and T. He, "Detecting faulty nodes with data errors for wireless sensor networks," *ACM Trans. Sensor Networks*, vol. 10, no. 3, p. 40, 2014.
- [3] W. P. Tay, J. N. Tsitsiklis, and M. Z. Win, "On the impact of node failures and unreliable communications in dense sensor networks," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2535–2546, 2008.
- [4] —, "Bayesian detection in bounded height tree networks," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 4042–4051, 2009.
- [5] K. Whitehouse and D. Culler, "Macro-calibration in sensor/actuator networks," *Mobile Netw. Applicat.*, vol. 8, no. 4, pp. 463–472, 2003.
- [6] R. Tan, G. Xing, Z. Yuan, X. Liu, and J. Yao, "System-level calibration for fusion-based wireless sensor networks," in *IEEE Real-Time Syst. Symp.*, 2010, pp. 215–224.
- [7] Y. Wang, A. Yang, Z. Li, X. Chen, P. Wang, and H. Yang, "Blind drift calibration of sensor networks using sparse Bayesian learning," *IEEE Sensors J.*, vol. 16, no. 16, pp. 6249–6260, 2016.
- [8] Y. Wang, A. Yang, Z. Li, P. Wang, and H. Yang, "Blind drift calibration of sensor networks using signal space projection and Kalman filter," in *IEEE Int. Conf. Intelligent Sensors, Sensor Networks and Inform. Process.*, 2015, pp. 1–6.
- [9] M. S. Stanković, S. S. Stanković, and K. H. Johansson, "Distributed blind calibration in lossy sensor networks via output synchronization," *IEEE Trans. Autom. Control*, vol. 60, no. 12, pp. 3257–3262, 2015.

- [10] B. Lee, S. Son, and K. Kang, "A blind calibration scheme exploiting mutual calibration relationships for a dense mobile sensor network," *IEEE Sensors J.*, vol. 14, no. 5, pp. 1518–1526, 2014.
- [11] V. Bychkovskiy, S. Megerian, D. Estrin, and M. Potkonjak, "A collaborative approach to in-place sensor calibration," in *ACM/IEEE Int. Conf. Inform. Process. Sensor Networks*, 2003, pp. 301–316.
- [12] C. Dorffler, M. Puigt, G. Delmaire, and G. Roussel, "Blind mobile sensor calibration using an informed nonnegative matrix factorization with a relaxed rendezvous model," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2016, pp. 2941–2945.
- [13] R. Gribonval, G. Chardon, and L. Daudet, "Blind calibration for compressed sensing by convex optimization," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, 2012, pp. 2713–2716.
- [14] Ç. Bilen, G. Puy, R. Gribonval, and L. Daudet, "Convex optimization approaches for blind sensor calibration using sparsity," *IEEE Trans. Signal Process.*, vol. 62, no. 18, pp. 4847–4856, 2014.
- [15] C. Schulke, F. Caltagirone, F. Krzakala, and L. Zdeborová, "Blind calibration in compressed sensing using message passing algorithms," in *Advances in Neural Inform. Process. Syst.*, 2013, pp. 566–574.
- [16] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.
- [17] J. J. Cho, Y. Ding, Y. Chen, and J. Tang, "Robust calibration for localization in clustered wireless sensor networks," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 1, pp. 81–95, 2010.
- [18] N. Moon and Y. Kim, "Optimized thermal compensation method using clustering and drifted response stability for total power radiometer calibration," *IEEE Sensors J.*, vol. 17, no. 5, pp. 1269–1276, 2017.
- [19] H. Medeiros, H. Iwaki, and J. Park, "Online distributed calibration of a large network of wireless cameras using dynamic clustering," in *ACM/IEEE Int. Conf. Distributed Smart Cameras*, 2008, pp. 1–10.
- [20] K. R. Smith, P. Edwards, M. J. Evans, J. D. Lee, M. D. Shaw, F. A. Squires, and A. Lewis, "Clustering approaches to improve the performance of low cost air pollution sensors," *FARADAY DISCUSSIONS*, pp. 1–15, 2017.
- [21] C. Luo, H. Hong, and M. C. Chan, "Piloc: A self-calibrating participatory indoor localization system," in *Int. Symp. Inform. Process. Sensor Networks*, 2014, pp. 143–154.
- [22] S. J. Gershman and D. M. Blei, "A tutorial on bayesian nonparametric models," *Journal of Mathematical Psychology*, vol. 56, no. 1, pp. 1–12, 2012.
- [23] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *J. Computational and Graphical Stat.*, vol. 9, no. 2, pp. 249–265, 2000.
- [24] Y. W. Teh, "Dirichlet process," in *Encyclopedia of Machine Learning*. London: Springer, 2011, pp. 280–287.
- [25] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet processes," *J. Amer. Stat. Assoc.*, 2012.
- [26] J. Yang, W. P. Tay, and X. Zhong, "A dynamic Bayesian nonparametric model for blind calibration of sensor networks," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, New Orleans, USA, 2017.
- [27] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press, 2012.
- [28] L. Ljung, *System identification*. Wiley Online Library, 1999.
- [29] P. Papadopoulos and V. Digalakis, "Identification of linear systems in canonical form through an EM framework," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, 2010, pp. 4110–4113.
- [30] G. Tsonzoz, V. Diakouloukas, C. Koniaris, and V. Digalakis, "Estimation of general identifiable linear dynamic models with an application in speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, vol. 4, 2007, pp. 453–456.
- [31] N. Chopin, "Fast simulation of truncated Gaussian distributions," *Stat. and Computing*, vol. 21, no. 2, pp. 275–288, 2011.
- [32] D. Kumar, S. Rajasegarar, and M. Palaniswami, "Automatic sensor drift detection and correction using spatial kriging and Kalman filtering," in *IEEE Int. Conf. Distributed Computing in Sensor Systems*, 2013, pp. 183–190.
- [33] N. Ramakrishnan, E. Ertin, and R. L. Moses, "Gossip-based algorithm for joint signature estimation and node calibration in sensor networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 665–673, 2011.
- [34] H. Ishwaran and M. Zarepour, "Exact and approximate sum representations for the Dirichlet process," *Canadian J. Stat.*, vol. 30, no. 2, pp. 269–283, 2002.
- [35] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "A sticky HDP-HMM with application to speaker diarization," *Ann. Appl. Stat.*, pp. 1020–1056, 2011.
- [36] R. Douc and O. Cappé, "Comparison of resampling schemes for particle filtering," in *Proc. Int. Symp. Image and Signal Process. and Anal.*, 2005, pp. 64–69.
- [37] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.
- [38] M. Vetterli. (2007, August) Sensorscope. [Online]. Available: <http://lcav.epfl.ch/cms/lang/en/pid/86035>
- [39] F. J. Expósito, A. González, J. C. Pérez, J. P. Díaz, and D. Taima, "High-resolution future projections of temperature and precipitation in the Canary islands," *J. Climate*, vol. 28, no. 19, pp. 7846–7856, 2015.
- [40] J. Yang, W. P. Tay, and X. Zhong, "Light sensors data," 2017. [Online]. Available: <http://dx.doi.org/10.21979/N9/QNV6CW>
- [41] Z. Li, J. Shen, X. Sun, and Y. Wang, "Nanoparticle size measurement from dynamic light scattering data based on an autoregressive model," *Laser Physics Lett.*, vol. 10, no. 9, p. 095701, 2013.