Blind Drift Calibration of Sensor Networks using Signal Space Projection and Kalman Filter

Yuzhi Wang, Anqi Yang, Zhan Li, Pengjun Wang, Huazhong Yang
Department of Electronic Engineering
Tsinghua National Laboratory for Information Science and Technology (TNList)
Tsinghua University, Beijing, China
Email: {yz-wang12, yang-aq14, lizhan13}@mails.tsinghua.edu.cn,
{wangpj, yanghz}@tsinghua.edu.cn

***Abstract*—As wireless sensor network (WSN) technologies become mature, an increasing number of large-scale WSN-based long-term monitoring systems are deployed. However, data quality, especially sensor drift, is affecting the trustworthiness of sensor data. In this paper, we proposed an online algorithm to blindly calibrate sensor drift using signal space projection and Kalman filter. By utilizing data correlation among sensors, the proposed method neither requires sensors to be densely deployed nor needs prior knowledge of data models. Simulation results showed the proposed method can detect and calibrate sensor drift successfully. The mean square error of estimated drift is less than 1%, which is more accurate than existing prediction-based methods. The proposed method is also robust to measurement noise, multiplicative drift, and signal subspace estimation error.**

***Index Terms*—WSN, Blind Calibration, Signal Space Projection, PCA, Kalman Filter**

I. Introduction

A wireless sensor network (WSN) consists of a number of autonomous sensor nodes with the ability of sensing, computing, and wireless communication working together to monitor physical or environmental conditions, and cooperatively passing the obtained data through an ad-hoc network to a main location [1, 2]. In recent years, the advance of micro-electromechanical systems (MEMS) and technology of System-on-Chip (SoC) has made it possible to manufacture low-cost and low-power sensor nodes with small physical size and adequate accuracy in measurement. Meanwhile, standardized WSN protocols such as ZigBee [3] and 6LoWPAN [4] have been proposed and verified. As a result, mature WSN technologies enable large-scale WSNs to be deployed in many industrial and consumer applications. In our previous work [5, 6], hundreds of small and low-cost sensor nodes were deployed to monitor animal activities, environmental conditions and bridge structural health.

However, with the spreading deployment of long-term monitoring systems, sensor drift and event faults reveal to be a serious problem affecting the trustworthiness of sensor data. According to Ni et al. [7], even costly high-accuracy sensors may still produce faulty data. In traditional monitoring systems, few manually operated sensors were deployed to regions which are relatively easy to reach, hence they can be manually calibrated annually or more frequently. But in many WSN applications, hundreds of nodes are deployed to harsh environment and at nearly inaccessible locations, so it's almost impossible to unmount and manually calibrate these sensors device-by-device. Therefore, there is an urgent need to automatically identify and calibrate faulty or drifted sensors without unmounting them.

Usually, it's almost impossible to measure the ground-truth data of sensing region, and the sensing model is hardly known, so sensors need to be calibrated without ground-truth data. This calibration method is called *blind calibration* [8]. Many existing blind calibration approaches [9–12] assumed that WSN was deployed very densely or the sensor nodes were to monitor the same phenomenon, so the sensors should read the same value when no drift occurs. But unfortunately, in many practical WSN projects, the density requirement is hard to satisfy.

This work is supported in part by NSFC under grant #61271269 and #61321061.

In this paper, we extend the idea of *signal subspace* proposed by Balzano and Nowak [8] to utilize the correlation among sensors, and use Kalman filter [13] to track and calibrate sensor drift. For sensor networks deployed to monitor physical environment, if some redundant sensors were deployed, i.e., physical values are slightly oversampled in sensing field, there would exist correlation among sensors. Thus the ground-truth of sensor measurement should lie in a lower-dimensional signal subspace of the measurement space, and we can calculate the orthogonal projection of sensor measurements onto the orthogonal complement of the signal subspace. We take sensor drift as system state, and the projected sensor measurements as observation, and in this way, Kalman filter can be used to track sensor drift. Hence, the sensors can be calibrated.

The main contributions of this paper are:

- We propose a novel two-phase online blind calibration method to detect and calibrate sensor drift without the requirement of dense deployment or prior knowledge of the data model, and we prove that the sensor network should over sample the signal space by at least 2 dimensions;
- The proposed method does not need to predict the ground-truth, but project actual sensor drift to a known space, and it's more accurate than existing prediction-based blind calibration approaches;
- Previous work [8, 11] modelled the mapping function between measurement and ground-truth as time-independent gain and offset, while in this paper sensor data error is simply modelled as additive drift which varies over time. Experiment results show that our method is robust to both additive and multiplicative drift.

The rest of the paper is organized as follows. Section II provides a literature review of some related work. In Section III, we formulate the problem and then describe our calibration algorithm in Section IV. In Section V, we simulate a measurement system to evaluate the proposed algorithm. Future work is discussed in Section VI and finally we draw conclusions in Section VII.

II. Related Work

Fault tolerance of sensor networks is a fundamental research topic. In [7, 14], sensor data faults and their causes were categorized by analyzing data collected from real-world WSN projects. We further categorize sensor data faults to: a) *out-of-range fault*: samples that significantly deviates from normal value, which is likely caused by hardware fault or random error; and b) *in-range error*: samples with high noise or offset from normal value but still in a reasonable range of the data model, which is likely caused by sensor drift, low-battery or out of working environment.

Normally, sensor data with out-of-range fault has no correlation with true measurement, hence this kind of fault can only be detected and abandoned.

We focus on in-range data error, which is usually caused by sensor drift, where erroneous data still has correlation with the ground truth, so it's possible to recover or estimate true measurement from erroneous data. Many previous works calibrate sensors by using prior knowledge or data models. In [14], an environment monitoring WSN system was deployed, where the authors proposed a physical dynamical model for the measurand, then use Ensemble Kalman filter or Particle Filter to filter measurement with noise and error. In [15], inspired by expectation maximization (EM) method, the authors proposed an iterative method to maximize the likehood of parameters of a pollution source detection system.

However, in many applications, such as environment monitoring and structural health monitoring, the data model is either unavailable or too expensive to build. Meanwhile, these applications need long-term monitoring, where sensor drift is likely to happen and cause in-range data error. Therefore, sensor data need to be calibrated blindly.

Many existing blind calibration approaches [9–12] are based on an assumption that sensors were densely deployed, or even monitoring the same phenomenon. But the density requirements limited the scope of application of these approaches.

Balzano and Nowak [8, 16] proposed a blind calibration method without requirement of dense deployment. They assumed that the sensors measurements lay in a lower dimensional "*signal subspace*" of the measurement space, that is to say, the sensor network is slightly oversampling the signals of interest. They modeled sensor error as $y = (x - \beta)/\alpha$, where x is the ground-truth and y is the measurement, α and β are *gain* and *offset*. The calibration algorithm is to find the gain and offset which can satisfy a linear equation system. They further proved that the sensor gains can be perfectly recovered, while the offset need more assumptions to be calibrated.

Kumar et al. [17] followed the idea of drift-aware WSN introduced by Takruri et al. [18]. They use Kriging as an interpolation algorithm to predict the ground-truth of a sensor, and take the difference between predicted and measured value as an estimation for sensor drift, then they use Kalman filter (KF) to track the drift. However, the accuracy of this approach is limited by the prediction phase, where prediction error could cause erroneous results.

In this paper, we follow the concept of signal subspace introduced by Balzano and Nowak to detect sensor drift, and use Kalman filter to track and calibrate sensors. Our approach does not need to predict any true value to estimate sensor drift by substracting the measurement and prediction, but takes the projection of drift to the orthogonal complement to the signal subspace as the observation, hence achieve better accuracy and robustness.

III. Problem Description

We follow Balzano's idea of signal subspace [8] to model the measurement and error of sensor networks. Consider a sensor network with n sensors, at each time instant t, let $\boldsymbol{x}_t = [x_{1,t},\ x_{2,t},\ \dots,x_{n,t}]^T$ be the ground-truth signal supposed to be measured by the sensor, where $x_{i,t}$ represents the signal value of sensor i at time instant t with no drift and noise. Obviously, $\boldsymbol{x}$ lies in an n-dimension *measurement space*, denoted as $\mathcal{M}$.

Each sensor would have an unknown drift, we denote the drifted sensor measurement as y, and assume that $y_{i,t} = x_{i,t} + d_{i,t}$, where $y_{i,t}$ is the sensor reading, $x_{i,t}$ is the unknown ground-truth, and $d_{i,t}$ is the unknown sensor drift. To summarize using vector notation:

$$\boldsymbol{y}_t = \boldsymbol{x}_t + \boldsymbol{d}_t \tag{1}$$

The blind drift calibration problem is to recover unknown ground-truth $\boldsymbol{x}$ from drifted measurement $\boldsymbol{y}$.

Now that neither $\boldsymbol{x}$ nor $\boldsymbol{d}$ can be measured, we need some assumptions for blind recovering. Assume that the ground-truth signal $\boldsymbol{x}$ lies in a signal subspace $\mathcal{S}$ of lower dimension r, where $0 < r < n$. The term "signal subspace" means, there exist some linear constrains among elements of $\boldsymbol{x}$, which cannot be an arbitrary value in $\mathcal{M}$.

For example, assuming that sensor signal outputs were driven by a group of signal sources: $\boldsymbol{s} = [s_1, s_2, \dots, s_r]^T$, and the signal at each sensor's position is a linear combination of signal sources, then $x_i = \sum_{j=1}^{r} a_{ij} s_j$, or in vector form:

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$$

where a_{ij} is the element at i-th row and j-th column of $n \times r$ matrix $\boldsymbol{A}$. Obviously, $\boldsymbol{x}$ should lie in the column space of matrix $\boldsymbol{A}$, or, the column vectors of $\boldsymbol{A}$ spanned the signal space $\mathcal{S}$.

We then denote $\mathcal{S}^\perp$ as the *orthogonal complement* to the signal subspace $\mathcal{S}$, which means $\mathcal{S}^\perp = \{\boldsymbol{v} \in \mathcal{M} \mid \forall \boldsymbol{u} \in \mathcal{S}, \langle \boldsymbol{u}, \boldsymbol{v} \rangle = 0\}$. Let $\boldsymbol{P}$ represent the *orthogonal projection* onto $\mathcal{S}^\perp$, hence $\mathcal{S}$ is the null space of $\boldsymbol{P}$. Therefore, for any $\boldsymbol{x} \in \mathcal{S}$,

$$\boldsymbol{P}\boldsymbol{x} = \boldsymbol{P}(\boldsymbol{y} - \boldsymbol{d}) = \boldsymbol{0} \tag{2}$$

From equation (2) we can get

$$\boldsymbol{P}\boldsymbol{d} = \boldsymbol{P}\boldsymbol{y} \tag{3}$$

Here we assume $\boldsymbol{P}$ is known, which we'll discuss later how to calculate, and since $\boldsymbol{y}$ is the known sensor readings, it seems $\boldsymbol{d}$ is recoverable. But actually the rank of $\boldsymbol{P}$ is only $n - r$, hence, this system is underdetermined. Balzano and Nowak have proved in [8] that $\boldsymbol{d}$ is recoverable only if at least $n - r$ elements of $\boldsymbol{d}$ is known.

In this paper, we deal with this problem by assuming the sensors would not drift simultaneously, or, at a certain time instant, at most one sensor could have unknown drift. We prove that if $n - r \geq 2$, the drifted sensor could be detected and calibrated.

IV. Online Drift Correction Algorithm

A. Assumptions and Algorithm Description

First, as we've discussed in section III, we assume the ground-truth signal lies in a lower-dimension linear subspace to the measurement space, and the signal subspace is time invariant. This assumption is appropriate since in many applications, such as structural health monitoring, the signal space is decided by the structure and sensors' geographic locations, which changes very slowly over time.

We then assume that all sensors are pre-calibrated before deployment. Hence, within a short period after the sensors were deployed, the drift should be zero or insignificant. Therefore, we can extract features of the signal subspace using sensor data collected within the initial period of deployment.

In order to detect the drifted sensor, we need to assume that sensors do not start drifting simultaneously, so at a certain time instant, at most one sensor could have unknown drift.

Based on these assumptions, we propose a blind drift calibration algorithm using signal space projection (SSP) and Kalman filter (KF).

Our algorithm has two phases: *learning phase* and *running phase*. In learning phase, we use principle component analysis (PCA) to calculate the orthogonal projection matrix $\boldsymbol{P}$ onto $\mathcal{S}^{\perp}$, the orthogonal complement of the signal subspace. And in running phase, we first detect whether there exists drift in sensor data, if so, find which sensor is drifting, then Kalman filter is used to track the drift value, as shown in figure 1. In this way, we can calibrate sensor data by substracting sensor reading and estimated drift.

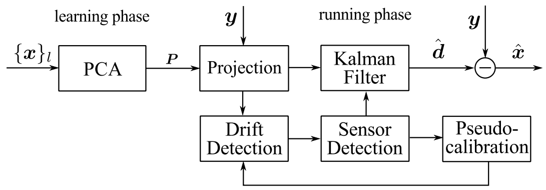

Fig. 1. SSP-KF Blind Drift Detection and Calibration Framework: $\{\boldsymbol{x}\}_l$ is the measurement signal set in learning phase, $\hat{\boldsymbol{d}}$ and $\hat{\boldsymbol{x}}$ are estimated drift and calibrated signal

B. Learn Signal Subspace from Data at Deployment

In learning phase, using sensor data collected within the initial period of deployment, where drift was considered to be null or insignificant, features of the signal subspace are estimated.

According to equation (3), the feature of the signal subspace needed for calibration is the orthogonal projection matrix $\boldsymbol{P}$ from $\mathcal{M}$ onto the orthogonal complement of the signal subspace.

$\boldsymbol{P}$ can be calculated from an orthogonal projection matrix $\boldsymbol{Q}$ from $\mathcal{M}$ onto $\mathcal{S}$. It's easy to prove that $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{Q}$ is an orthogonal projection matrix from $\mathcal{M}$ onto $\mathcal{S}^{\perp}$, where $\boldsymbol{I} = \mathrm{diag}([1, 1, \ldots, 1]^T_{1\times r})$.

Proof: $\forall \boldsymbol{u}, \boldsymbol{v} \in \mathcal{M}$, let $\boldsymbol{Q}$ be an orthogonal projection matrix on to $\mathcal{S}$, then $\boldsymbol{Q}\boldsymbol{u} \in \mathcal{S}$, let $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{Q}$ Then

$$\begin{aligned}\langle \boldsymbol{P}\boldsymbol{v}, \boldsymbol{Q}\boldsymbol{u}\rangle &= ((\boldsymbol{I}-\boldsymbol{Q})\boldsymbol{v})^T\boldsymbol{Q}\boldsymbol{u} = \boldsymbol{v}^T(\boldsymbol{I}-\boldsymbol{Q})^T\boldsymbol{Q}\boldsymbol{u} \\ &= \boldsymbol{v}^T(\boldsymbol{Q}-\boldsymbol{Q}^T\boldsymbol{Q})\boldsymbol{u} \end{aligned} \tag{4}$$

Since $\boldsymbol{Q}$ is an orthogonal projection matrix, there is $\boldsymbol{Q}^T = \boldsymbol{Q} = \boldsymbol{Q}^2$, so $\boldsymbol{Q} - \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{Q} - \boldsymbol{Q} = \boldsymbol{0}$, hence, $\langle \boldsymbol{P}\boldsymbol{v}, \boldsymbol{Q}\boldsymbol{u}\rangle = 0$, or $\boldsymbol{P}\boldsymbol{v} \in \mathcal{S}^{\perp}$.

Obviously, $\boldsymbol{P}^T = \boldsymbol{P} = \boldsymbol{P}^2$, therefore, $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{Q}$ is an orthogonal projection matrix from $\mathcal{M}$ onto $\mathcal{S}^{\perp}$. ■

We can construct $\boldsymbol{Q}$ using a set of orthogonal basis of $\mathcal{S}$, let $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_r]_{n\times r}$, where $\boldsymbol{\phi}_i$ is a set of orthogonal unit vectors that span $\mathcal{S}$, then $\boldsymbol{Q}$ can be constructed as $\boldsymbol{Q} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$ [19].

Since $\boldsymbol{x}$ lies in $\mathcal{S}$, hence a number of samples during the learning phase $\{\boldsymbol{x}\}_l$ can be considered to span the signal subspace, and we can extract a set of unit orthogonal basis from $\{\boldsymbol{x}\}_l$ by PCA [20].

There are three reasons of using PCA:

- PCA is a non-parametric method, which can extract unit orthogonal basis of the signal subspace using only a set of data samples;
- It's easy to estimate the dimension of the signal subspace using PCA. In practice, using few principle components associated with largest singular values is enough to construct an even lower-dimensional estimation of the signal subspace;
- Without drifting, the measured signals would still be noisy. While PCA method can construct a best matching subspace of the signal subspace from least-square meaning, so PCA is robust to measurement noise.

We use singular value decomposition(SVD) to calculate the principal components, or, the basis of the signal subspace. Let $\boldsymbol{X} = [\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_\tau]_{n\times\tau}$, where $\boldsymbol{x}_i$ is the signal vector measured by the sensor network at time instant i. Then we calculate it's SVD:

$$\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* = \mathrm{svd}(\boldsymbol{X}) \tag{5}$$

where the columns of $\boldsymbol{U}_{n\times n}$ are left-singular vectors, which is a set of unit orthogonal basis to the signal subspace, $\boldsymbol{\Sigma}_{n\times\tau}$ is a diagonal matrix consists of singular values; $\boldsymbol{V}$'s columns are called right-singular vectors.

The dimension $\hat{r}$ of the estimation of signal subspace is determined as:

$$\sum_{e\in\boldsymbol{\Sigma}, i=0}^{\hat{r}} e_{i,i} \geq T_{\mathrm{r}} \cdot \sum_{e\in\boldsymbol{\Sigma}} e \tag{6}$$

where T_{r} is the threshold. In practice, we can choose 0.99 as a proper value of T_{r}.

The first $\hat{r}$ columns of $\boldsymbol{U}$ are the unit orthogonal basis of the estimated signal subspace, or, $\hat{\boldsymbol{\Phi}} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{\hat{r}}]$, where $\boldsymbol{u}_i$ is the i-th column of $\boldsymbol{U}$.

Finally, the orthogonal projection matrix onto the orthogonal complement of the estimated signal subspace can be constructed as:

$$\boldsymbol{P} = \boldsymbol{I} - \hat{\boldsymbol{\Phi}}\hat{\boldsymbol{\Phi}}^T \tag{7}$$

C. Drift Tracking using Kalman filter

Before discussing how to detect the drifted sensor, we suppose the drifted sensor had been detected. Let k be the identifier of the drifted sensor, we model its drift as:

$$d_{k,t} = d_{k,t-1} + v_{k,t} \quad v_{k,t} \sim N(0, Q_t) \tag{8}$$

where $d_{k,t}$ is the drift of sensor k at time instant t, and $v_{k,t}$ is a random variable that obeys Gaussian distribution.

At each time instant t of running phase, we calculate the offset of measured signal projected onto $\mathcal{S}^{\perp}$, according to equation (1) and equation (3):

$$\begin{aligned}\boldsymbol{P}\boldsymbol{y}_t &= \boldsymbol{P}(\boldsymbol{x}_t + \boldsymbol{d}_t + \boldsymbol{w}'_t) \quad \boldsymbol{w}'_t \sim N(0, \boldsymbol{R}'_t) \\ &= \boldsymbol{P}\boldsymbol{d}_t + \boldsymbol{w}_t \quad \boldsymbol{w}_t \sim N(0, \boldsymbol{R}_t)\end{aligned} \tag{9}$$

where $\boldsymbol{P}$ is the orthogonal projection matrix calculated in learning phase, $\boldsymbol{y}_t$ is sensors' measurements with unknown drift, $\boldsymbol{d}_t$ is sensors' drift and $\boldsymbol{w}'_t$ and $\boldsymbol{w}_t$ are additive Gaussian noise, and $\boldsymbol{R}_t = \boldsymbol{P}\boldsymbol{R}'_t$.

Since we assumed sensors do not start drifting at the same time, hence at most one sensor has an undetermined drift, if other sensors had already drifted, it should be already calibrated. Therefore, at most one element of $\boldsymbol{d}_t$ is non-zero, and that would be $d_{k,t}$. So $\boldsymbol{d}_t = [0, \ldots, d_{k,t}, 0, \ldots, 0]^T$, as a result,

$$\boldsymbol{P}\boldsymbol{d}_t = \boldsymbol{P}[0, \ldots, d_{k,t}, 0, \ldots, 0]^T = \boldsymbol{p}_k \cdot d_{k,t} \tag{10}$$

where $\boldsymbol{p}_k$ is the k-th column of $\boldsymbol{P}$.

Let $\boldsymbol{z}_t = \boldsymbol{P}\boldsymbol{y}_t$, combined with equation (9) and equation (10),

$$\boldsymbol{z}_t = \boldsymbol{p}_k \cdot d_{k,t} + \boldsymbol{w}_t \quad \boldsymbol{w}_t \sim N(0, \boldsymbol{R}_t) \tag{11}$$

Equation (8) and equation (11) consist a kalman state-observation model, where the system state is the sensor drift, and the observed value is measurement signal's offset projected onto $\mathcal{S}^{\perp}$. Hence we can use Kalman filter [13] to optimally estimate the drift iteratively shown in algorithm 1, the estimated drift is denoted as $\hat{d}_{k,t}$.

Algorithm 1: Drift Tracking using Kalman filter

function KF (k, t_0, τ, Q_t, $\boldsymbol{R}_t$, $P_{t_0}^+ \leftarrow 0$, $\hat{d}_{k,t_0} \leftarrow 0$)
  // 0 is the default initial value of $P_{t_0}^+$ and $\hat{d}_{k,t_0}$
  for $t \leftarrow t_1$ **to** τ **do**
    $P_t^- \leftarrow P_{t-1}^+ + Q_t$
    $\boldsymbol{K}_t \leftarrow P_t^- \boldsymbol{p}_k^T (\boldsymbol{p}_k P_t^- \boldsymbol{p}_k^T + \boldsymbol{R}_t)^{-1}$
    $\hat{d}_{k,t} \leftarrow \hat{d}_{k,t-1} + \boldsymbol{K}_t(\boldsymbol{z}_k - \boldsymbol{p}_k \hat{d}_{k,t-1})$
    $P_t^+ \leftarrow (\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{p}_k) P_t^-$
  return $\hat{\boldsymbol{d}}_k$

D. Drift Sensor Detection

As discussed before, sensor drift could be optimally estimated using Kalman filter if the drifted sensor could be detected. Let k be the identifier of the drifted sensor, and j be another sensor. Without consideration of additive noise:

$$\boldsymbol{P}\boldsymbol{y}_t = \boldsymbol{p}_k \cdot d_{k,t} \tag{12}$$

If sensor j was mistaken for the drifted sensor, trying to solve the equation $\boldsymbol{P}\boldsymbol{y}_t = \boldsymbol{p}_j \cdot d_{j,t}$ would lead to

$$\boldsymbol{p}_j \cdot d_{j,t} = \boldsymbol{p}_k \cdot d_{k,t} \quad k \neq j \tag{13}$$

In most cases $\boldsymbol{p}_i$ are not linear correlated to each other, so equation (13) has a solution if and only if $\text{rank}(\boldsymbol{P}) = n - r = 1$. That is to say, if $n - r \geq 2$, equation (13) would have no solution. In this way, the drifted sensor can be detected by enumerating k and test whether equation (12) has a solution.

However, because all measurements comes with noise, a more robust method is to suppose j was the drifted sensor, then use kalman filter to pseudo-calibrate the drift, hence, a series of pseudo-calibrated samples $\hat{\boldsymbol{x}}^{(j)}$ can be calculated. If and only if $j = k$, $\boldsymbol{P}\hat{\boldsymbol{x}} \approx \boldsymbol{0}$. In other words, the j to minimize $\boldsymbol{P}\hat{\boldsymbol{x}}$ is supposed as k.

Algorithm 2 shows our drift sensor detection method.

Algorithm 2: Drifted Sensor Detection

function project ($\boldsymbol{Y}$)
  /* $\boldsymbol{Y}$ is a matrix consists of several signal samples over time, each column of $\boldsymbol{Y}$ is a signal sample */
  $\boldsymbol{o} \leftarrow \boldsymbol{P}\boldsymbol{Y}$
  return $||\sum_i \boldsymbol{o}_i||$ // $\boldsymbol{o}_i$ is the i-th column of $\boldsymbol{o}$
function detect (t_0, τ, $P_{t_0}^+ \leftarrow 0$, $d_{k,t_0} \leftarrow 0$)
  drift $\leftarrow$ **false**
  for $t \leftarrow t_0 \cdots \tau$ **do**
    proj $\leftarrow$ project ($\boldsymbol{y}[t \rightarrow t + w]$) // w is a window
    if *proj* $> T_h$ **then**   // T_h is threshold
      break
  if $t = \tau$ **then**
    return NoDriftDetected
  min $\leftarrow T_h$, k $\leftarrow$ **null**
  for $j \leftarrow 1 \cdots n$ **do**
    $\boldsymbol{d_j} \leftarrow$ KF (j, t, $t + w$, Q_t, $\boldsymbol{R}_t$)
    $\hat{\boldsymbol{x}}^{(j)} \leftarrow \boldsymbol{y} - \boldsymbol{d_j}$
    proj$_j$ $\leftarrow$ project ($\hat{\boldsymbol{x}}^{(j)}$)
    if *proj$_j$* $<$ *min* **then**
      $k \leftarrow j$, min $\leftarrow$ proj$_j$
  if $k =$ **null then**
    return Error
  return k

E. Summary

In this section, we first discussed the assumptions of our algorithm. Then the method of learning the orthogonal projection matrix onto the orthogonal complement of the signal subspace using PCA was introduced. After that, we modeled sensor drift and projection as a kalman state-observation process, and hence the drift can be tracked with a Kalman filter. Finally, we proved that if $n - r \geq 2$, the drifted sensor can be detected.

V. Evaluation

In real-world sensor networks, it's difficult to get the ground-truth values of each sensor, and although there exist several open sensor network datasets available online [21], they come with no ground-truth. It's not convincing to prove the correctness of calibration algorithm if the ground-truth is unknown, so we conduct the evaluation of our proposed algorithm via simulation implemented with SciPy [22].

A. Simulation Scenario

To simulate an n-dimensional measurement space with r-dimensional signal subspace, we simulate a 10×10 sensing field with n sensors and r signal source randomly placed in the field.

In our experiment, we let $n = 15$ and $r = 10$, sensor and source positions are shown in figure 2.

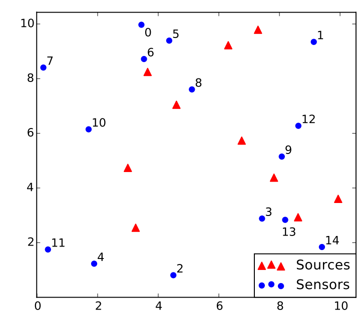

Fig. 2. Sensing Field: Positions of signal sources and sensors

The source signals were generated using several samples of low-pass filtered ARMA process [23], each of 10 the series was generated and sampled independently with the length of 8192.

Sensors work as follows, for each sensor i, it's measured signal is

$$x_{i,t} = \sum_{j=1}^{r} a_{ij} s_{j,t} \tag{14}$$

where s_j is the source signal value, a_{ij} is the contribution of source j to sensor i, a_{ij} is calculated by:

$$a_{ij} = (\delta_{ij} + 1)^{-1.5} \tag{15}$$

where δ_{ij} is the distance between the source j and sensor i. Hence, the signal subspace is spanned by vectors of these coefficients totally decided by sensors' positions.

The data samples used in following experiments are shown in figure 3, where the upper figure shows the source signals, and the lower one plots the sensor readings. From the figure we can see, the source signals are independent from each other, while the sensor readings are correlated but different from each other.

B. Result and Comparison

As introduced above, the rank of signal subspace is 10, and 15 sensors were deployed. The time range of signals is 0 to 8191.

We first add noise and drifting to measurements. Additive Gaussian noise at 45.0 dB of signal-noise-ratio (SNR) was generated. The

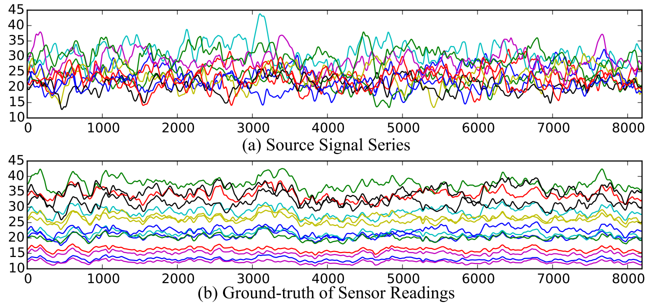

Fig. 3. Simulated Signal of Sources and Sensors: Sources signals are independent, while sensor signals are correlated

learning phase is time range $[0, 499]$, and one random sensor starts drifting from a random time instant in $[500, 999]$. Q_t is set to 0.002 and $\boldsymbol{R}_t$ is set to $\text{diag}([0.1, 0.1, \dots, 0.1]_n)$. In practice, Q and $\boldsymbol{R}$ should be estimated from measurements.

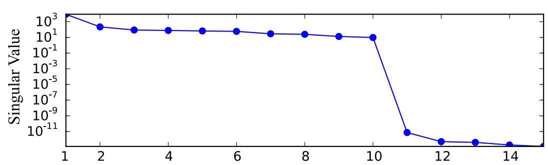

Fig. 4. Singular Values in Learning Phase: The sharp falling at 11-th singular value verified the dimension of signal subspace is 10

In learning phase, we set the estimation threshold to 0.99, and 6 vectors was chosen as the basis to estimate the signal subspace. The singular values are shown in figure 4.

We then run the proposed SSP-KF and compare the results with Kriging-KF [17], shown in figure 5. The dark-blue solid line is the ground-truth signal, the green line is the drifted measurement, the orange dashed line is the calibrated signal estimation using the proposed SSP-KF method and the pink dotted line is the result using Kriging-KF method.

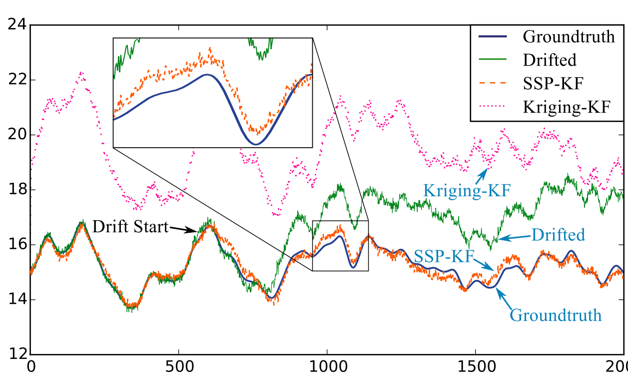

Fig. 5. Simulation Result and Comparison to Kriging-KF: SSP-KF is more accurate in this scenario, and calibrated signal is very closed to the ground-truth

From the figure we can see: a) The result of Kriging-KF is biased, and less accurate than that of SSP-KF method; b) The calibrated signal using our SSP-KF method is very near to the ground-truth; c) The estimated 6-dimensional space has kept enough information we need for calibration, and using $1/16$ of samples to estimate the signal subspace does not lead to accumulative error.

Recalling equation (11), the observation of kalman state model is derived from the ground-truth of sensor drift, while Kriging-KF needs to predict and estimate sensor drift as observation. Hence, the Kalman filter in Kriging-KF is actually tracking the estimated drift, but in our method, KF is tracking the true drift. That's why SSP-KF is more accurate than Kriging-KF.

C. Robustness

We further test the robustness of our proposed approach. First, we test whether our additive drift model is enough for sensor drift; then we test its noise-tolerance by calculating its accuracy under different SNRs; finally, as shown in previous experiment, we estimated the 10-dimensional signal subspace with a 6-dimensional space and got fairly accurate result, so we would test the results' accuracy under different dimension of signal subspace estimations.

1) Drift Model: Previous work [8, 11] modelled sensor as offset and gain, while our SSP-KF method only considers additive drift, which corresponds to "offset" in Balzano's model. We verify our method by simulating sensor measurements with drift in both gain and offset, while calibrate only considering drift (offset), and compare the mean-square-error (MSE) between real and estimated drift with the scenario where measurements were only drifted. We name the two drifting models as "with gain" (WG) and "without gain" (WOG), and set SNR at 50 dB, and for each model, we run 5 times in case of random error. For WOG group, the MSE is 0.426%, and for WG group, the MSE is 0.452%. So modelling error can lower the accuracy of our calibration approach, but the influence is very limited.

Balzano and Nowak's model did not take time into consideration, and the calibration method is essentially a parameter-estimation to a mapping function from measurements to the ground-truth. So it's necessary to model the mapping with gain and offset, or even more complex non-linear function. Our method works iteratively, and the drift parameter is time-variant. As long as the Kalman filter could track up the difference between measurement and ground-truth, the final result should not be erroneous. Therefore, it's enough to model sensor data error as additive drift, more complex model might lead to more complex algorithm and more uncertainty.

2) Noise: Noise can influence the quality of signal subspace estimation, here we test whether the proposed approach is noise-tolerant. We run our simulation under different SNR settings and use MSE as the indicator of accuracy, and for each SNR value, the simulation is run 5 times.

As the results shown in table I, the MSE is lower than 1% until SNR gets lower than 35 dB. Since both PCA and Kalman filter are optimal in meaning of least square error, the proposed approach is noise-robust.

TABLE I
MSE UNDER DIFFERENT SNR

SNR (dB)	60	50	40	35	30	25	20
MSE (%)	0.409	0.425	0.577	0.938	2.10	5.65	14.1

3) Signal Subspace Estimation: In our simulation settings, the dimension of the signal subspace is 10, since we set PCA threshold to 0.99, the estimated signal subspace only has dimension of 6, while the results have shown that a non-precise estimation can result in acceptible accuracy. Here we test the robustness of the proposed approach to signal subspace esitmation error by setting the estimated dimension to different values and use MSE as accuracy indicator.

TABLE II
MSE UNDER DIFFERENT SIGNAL SPACE DIMENSION

$\hat{r}$	1	3	4	6	7	8	10	11	12
E(%)	93.8	97.1	97.9	99.2	99.5	99.8	100	100	100
MSE(%)	2.98	2.91	0.55	0.42	0.22	0.04	0.04	0.05	0.06

As the results shown in table II, the "E" row is the energy (sum of chosen singular values) of total energy, in this scenario, since sensor values are highly correlated with their geographical location, the first singular value has made up 93.8% of the total energy. When estimated dimension is higher than 4, the MSE is under 1%. An interesting phenomenon is that when estimated dimension is larger than 10, the accuracy instead reduced. This is because the first 10 dimensions has kept almost all the information needed, higher dimensions would bring in more noise interference. Meanwhile, since $\boldsymbol{P}$ has rank of $n - r$, larger r value would lower redundancy of the algorithm, thus making it less noise-robust.

D. Summary

In this section, we first introduced our simulation settings, including the method to generate sensor signals and the sensing field. Then we ran our SSP-KF method to calibrate a drifted sensor and compared with the Kriging-KF method proposed in [17], where our SSP-KF performed better. We further conducted robustness evaluation in three aspects: drift model, additive noise and signal subspace estimation, and the results showed that the proposed SSP-KF approach is robust to all these factors.

VI. Future Work

We proposed a signal space projection based blind calibration method for sensor drift named SSP-KF, which performs well in our simulation. While there are still some issues.

a) Time-variance of signal subspace: SSP-KF is based on the assumption that the signal subspace is time invariant. Although we have proved via simulation that this assumption is appropriate in our supposed scenario, in practice, the signal subspace is decided by both sensors' location and the monitoring field, which may slowly change over time. Future work on sensor network calibration with time variant signal subspace is needed.

b) Testbed Evaluation: : We'll setup a testbed with ground-truth readings and use it to evaluate calibration algorithms.

c) Application: We'll try to apply our method to real-world sensor network applications, and in turn improve our approach that can fit real-world scenarios.

VII. Conclusion

Blind calibration has become one of the key issues of sensor networks when large-scale and long-term sensor network applications increase. In this paper, we proposed a blind drift calibration method using signal subspace projection and Kalman filter named SSP-KF, which can run without prior knowledge of measurand data model or prediction of ground-truth value, and it does not require the sensors to be densely deployed.

By using signal subspace projection, the proposed method can use a projection of the true value of drift as the observation, then use Kalman filter to optimally estimate sensor drift. Hence, it's more accurate than the approaches that need to predict sensors' ground-truth and estimate a drift observation.

Experiments showed that the proposed method is not only accurate, but also robust to model error, additive noise and signal subspace estimation error. Therefore, the SSP-KF calibration algorithm should be capable to be applied in real-world sensor network systems.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292 – 2330, 2008.

[2] Wikipedia, "Wireless Sensor Network." [Online]. Available: https://en.wikipedia.org/wiki/Wireless_sensor_network

[3] "The ZigBee Alliance." [Online]. Available: http://zigbee.org

[4] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," IETF RCC 4619, Aug 2007.

[5] W. Liu, X. Fei *et al.*, "Design and implementation of a hybrid sensor network for milu deer monitoring," in *ICACT '12*, Feb 2012, pp. 52–56.

[6] X. Fei, Q. Zheng *et al.*, "A reliable transfer protocol for multi-parameter data collecting in wireless sensor networks," in *ICACT '13*, Jan 2013, pp. 569–573.

[7] K. Ni, M. Srivastava *et al.*, "Sensor network data fault types," *ACM TOSN*, vol. 5, no. 3, pp. 1–29, May 2009.

[8] L. Balzano and R. Nowak, "Blind Calibration of Sensor Networks," *IPSN '07*, pp. 79–88, Apr. 2007.

[9] V. Bychkovskiy, S. Megerian, and D. Estrin, "A Collaborative Approach to In-Place Sensor Calibration," in *IPSN '03*, 2003, pp. 301–316.

[10] M. Takruri, K. Aboura, and S. Challa, "Distributed Recursive Algorithm for Auto Calibration in Drift Aware Wireless Sensor Networks," in *Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering*, K. Elleithy, Ed., 2008, pp. 21–25.

[11] M. S. Stanković, S. S. Stanković, and K. H. Johansson, "Distributed Blind Calibration via Output Synchronization in Lossy Sensor Networks," pp. 1–34, Jul. 2013. [Online]. Available: http://arxiv.org/abs/1307.6982

[12] B. Lee, S. Son, and K. Kang, "A Blind Calibration Scheme Exploiting Mutual Calibration Relationships for a Dense Mobile Sensor Network," *IEEE Sensors Journal*, vol. 14, no. 5, pp. 1518–1526, May 2014.

[13] G. Welch and G. Bishop, "An introduction to the kalman filter," 1995.

[14] L. K. Balzano, "Addressing Fault and Calibration in Wireless Sensor Networks," Master's thesis, University of California, Los Angeles, 2007.

[15] X. Chaocan, P. Yang *et al.*, "Calibrate without Calibrating: An Iterative Approach in Participatory Sensing Network," *IEEE Transactions on Parallel and Distributed Systems*, vol. 9219, no. c, pp. 1–1, 2014.

[16] J. Lipor and L. Balzano, "Robust blind calibration via total least squares," in *ICASSP '14*. IEEE, May 2014, pp. 4244–4248.

[17] D. Kumar, S. Rajasegarar, and M. Palaniswami, "Automatic Sensor Drift Detection and Correction Using Spatial Kriging and Kalman Filtering," *DCOSS '13*, pp. 183–190, May 2013.

[18] M. Takruri and S. Challa, "Drift aware wireless sensor networks," in *2007 10th International Conference on Information Fusion*. Ieee, Jul. 2007, pp. 1–7.

[19] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra Book and Solutions Manual*, 2001.

[20] J. Shlens, "A Tutorial on Principal Component Analysis," Apr. 2014. [Online]. Available: http://arxiv.org/abs/1404.1100

[21] S. Madden, "Intel Lab Data," accessed: 2015-02-25. [Online]. Available: http://db.lcs.mit.edu/labdata/labdata.html

[22] SciPy developers, "Scipy." [Online]. Available: http://www.scipy.org/

[23] J. D. Cryer and K.-S. Chan, *Time Series Analysis: With Applications in R (2nd Edition)*. Springer, 2010.