A Recursive Approach to Partially Blind Calibration of a Pollution Sensor Network

Thomas Becnel, Tofigh Sayahi, Kerry Kelly, and Pierre-Emmanuel Gaillardon
The University of Utah, Salt Lake City, Utah, USA
Email: thomas.becnel@utah.edu

***Abstract*—Distributed, low-cost sensor networks have become a widely used tool to aid in the interpolation of atmospheric measurements between regulatory grade monitoring stations, referred to as Golden Standards (GS). However, the quality of the data from these sensor networks can be questioned, especially in poorly correlated environments. Sensors can be individually calibrated in a laboratory environment before deployment of the network, but this approach is unfeasible for large networks. To overcome these shortcomings, we propose a novel online, autonomous approach to sensor calibration, by leveraging the ground truth measurements of the GS to calibrate neighboring nodes of the sensor network using a Recursive Least-Squares technique. Our algorithm percolates this calibration through the network such that every connected node will converge towards its ideal linear calibration. The algorithm outperforms a pre-deployment laboratory calibration and provides good tracking of quickly-changing environmental stimulus, which is known to change the inherent sensor calibration. Experimental results show a 45% improvement in estimated measurement error when compared to measurements corrected using a laboratory calibration.**

***Index Terms*—recursive least squares, wireless sensor network, blind calibration, measurement noise, distributed calibration algorithm**

I. Introduction

Wireless Sensor Networks (WSNs) have proven to be an invaluable tool to develop spatially and temporally dense datasets of large-scale heterogeneous environments [1], [2]. The price of wireless sensor nodes has dropped dramatically in the last decade, leading to the utilization of WSNs under many facets of environmental sensing, such as water quality monitoring, permafrost analysis, and localized urban air pollution events [3], [4], [5].

The inherent properties of low-cost, wireless sensor networks mean that they are typically noisy by nature, due to manufacturing variability as well as environmental measurement noise. Traditionally sensors are calibrated in a laboratory setting to determine the best linear fit before deployment. However, this approach is undesirable for several reasons. First, the cost of calibrating every sensor does not scale well as the network grows. Second, it is possible that the laboratory calibrations do not represent real environmental conditions and therefore may misrepresent the intrinsic error of the sensor. In many applications, such as optical particle counters, changes in the environment will drastically alter the gain and offset of a sensor, rendering the pre-deployment calibration useless [6]. Instead, it would be desirable to calibrate the sensors while online, where environmental conditions can be utilized to aid in the calibration [7].

Sensor calibration is a crucial step prior to deployment and has rightfully been a strong research topic for many years [7], [8]. If a ground-truth reference does not exist, something typical of remote networks, the sensors must look for redundancies in the data to estimate their calibrations, known as *blind* calibration [9]. If a reference signal is sparsely known by the network, the information can be leveraged to provide baseline measurements to improve calibration. The use of a ground-truth reference has been coined *partially-blind* calibration in previous works [10]. These calibration techniques become substantially more difficult given real-world engineering constraints such as affordability, extreme low power and longevity, spatial and temporal heterogeneity of the environment, and sensor calibrations that dynamically change with the environmental conditions. The work described above does not address these constraints. In this work, we explore a new technique for partially-blind calibration of an online network by utilizing stationary ground-truth sources as calibration seeds, in order to propagate and maintain a dynamic calibration of each sensor that reflects the current environment. This method was first introduced in [11] and was used to calibrate mobile sensor nodes when they came within proximity of one another. We build off of this concept with a lightweight algorithm suited for extremely low-power sensor nodes with limited communication.

In this paper, we propose a novel online calibration scheme that leverages sparse, but highly accurate data from one or more ground-truth references (such as a federally maintained “Golden Standard” particle counter in the case of air pollution monitoring) to seed the calibration of a distributed sensor network, which we will refer to as the *weighted neighborhood calibration algorithm*. We exploit the correlation and distance between sensors to compensate for the heterogeneous and dynamic environment in which the sensors are sampling. By implementing a localized *Recursive Least Squares* (RLS) algorithm on each sensor node, sensors estimate a linear regression based on stimulus data from their calibrated neighbors at an arbitrary time step. In this work, we demonstrate the capabilities of the algorithm in an ideal environment, then analyze its performance by applying it to AirU – a network of low-cost optical particle counters currently deployed in Salt

Lake City, Utah, USA.

More specifically, we make the following major contributions in this paper:

- We propose a novel two-step, decentralized calibration method based on recursive least squares minimization and sample correlation between neighboring nodes. This method uses sparse ground-truth measurements from dependable references to provide a calibration seed to neighboring low-cost sensors in the network.
- We provide experimental results on an actively deployed sensor network and demonstrate that the proposed method outperforms the laboratory calibration of individual sensors by 45%, and outperform the state-of-the-art distributed online calibration algorithms by 20%. Results also show that the algorithm is capable of efficiently tracking abrupt environmental changes and updating the calibration accordingly.

The remainder of this paper is organized as follows: Section II describes the background related to this work, Section III reviews the proposed algorithm in detail, Section IV describes the experimental setups and results of the algorithm on a currently deployed pollution monitoring network, and Section V concludes the work described here.

II. Background

In this section, we cover the current state of low-cost environmental sensor networks, how they are currently calibrated, and the current solutions to improving their calibrations while online.

A. Shortcomings of Offline Calibrations

Low-cost sensor networks have become a successful solution in monitoring large heterogeneous environmental events for long periods of time, given their unparalleled ability to cover large areas and collect measurements with high resolution [12]. Monitoring airborne pollution is one such use of sensor networks, with many networks, such as PurpleAir [13], AeroQual [14], and Alphasense [15]. In addition, many research groups have deployed their own networks [16], [17]. These low-cost sensors possess some intrinsic gain and offset when sampling data, and are therefore calibrated in a laboratory environment before deployment [6]. These offline calibrations are not ideal because a sensor network may span a broad land use model, covering urban, industrial, suburban, and geographically diverse regions. Each region produces a unique makeup of airborne particles, and will therefore produce different mappings of the sensor measurements to the actual data [18]. Consequently, static offline calibrations can misrepresent the actual online calibration of a sensor. It is then desirable to calibrate sensors online in order to account for the inevitable change of the calibration over time, due to drift and changing environmental factors.

B. Current Online Calibration Techniques

The rationale into why traditional calibration techniques are inadequate in ad-hoc localized systems, such as sensor networks, is strictly because of the vast amount of data each node is required to obtain to perform a calibration [19]. Since the recent popularity of low-cost sensor networks in the last decade, much work has been done to accomplish the task of online calibration [7], [8]. *Blind calibration* has recently gained traction as a viable calibration mechanism by leveraging redundant measurements in a densely deployed network. The authors in [20] show that through subspace matching, sensor gain can be extracted from over-sampled systems, and was used to estimate linear sensor drift over time. In [21], the authors improve on blind calibration and achieve minimized linear least squares parameters via network consensus algorithms. The authors in [22] were the first to demonstrate that a network of low-cost sensors in a mobile network could be calibrated using spatially sparse references. As sensors passed by the reference, or one another, they would exchange datasets and the uncalibrated sensor would calculate its gain and offset. In [23], the authors propose a solution to the positive relationship between calibration noise and hops from a reference by using the Geometric Mean Regression (GMR) in place of ordinary least squares. The advantage of GMR is the consideration of noise in the independent variable as well as the dependent variable, so the gain will not tend towards zero as noise increases in the system. GMR is intended to produce minimized linear estimates between datasets that exhibit low correlations, where the independent variable also contains noise. We show later that GMR is not well suited for networks with high inter-sensor correlation, such as optical particle counters used to measure airborne particulate concentrations. Recently there has been a development towards advanced estimation and calibration techniques by leveraging calibrated reference signals [24], [25], [26]. These techniques are undesirable for our problem statement because they require statistical assumptions about the network or are too computationally extensive to implement in a low-power, distributed approach.

In this work, we take into consideration the advantages and shortcomings of the previous work, and apply it to the specific problem of calibrating an urban low-cost sensor network, given the constraints of a low-cost, low-power, low-bandwidth, short-range, distributed network, sampling a highly heterogeneous process in both space and time. We leverage the existence of a regulatory-grade particle counter which is fairly centralized in the network topology.

III. Weighted Neighborhood Algorithm

In this section we first give an overview of the sensor network under consideration. Then we present the proposed weighted neighborhood calibration algorithm. Finally we conclude the section with a trivial example demonstrating the functionality of the algorithm.

A. Algorithm Rationale and Assumptions

Ideal Calibration. Consider a continuous, stochastic and heterogeneous time dependent signal, which may be an environmental measurand, such as temperature, humidity, pollution, *etc*. Now, consider a spatially static sensor which collects *observations* of this signal. We can define the observations as the discrete-time signal: $d_k = \alpha x_k + \beta + z_k$ where d_k is the observed signal, x_k is the environmental measurand, α, β are the ideal gain and offset of the sensing unit, and z_k is a normally distributed Gaussian noise with zero mean, representing the environmental and measurement noise.

Iterative Linear Regression. Now, consider n distributed sensors, each collecting their own set of measurements, $d_{i,k}$. The system of equations relating to these n sensors is overdetermined because of the additive noise, and therefore must be iteratively estimated.

Our weighted neighborhood calibration algorithm makes use of the *Recursive Least Squares* (RLS) filter, which iteratively approximates the gain and offset parameters as new measurements are added to the system, and is described in detail in [27].

Algorithm 1 Weighted Neighborhood Iterative Calibration

1: **procedure** WEIGHTEDNEIGHBORHOOD($\mathcal{G}$)
2:     **for all** k discrete time steps **do**
3:         Create breadth-first queue
4:         **while** queue not empty **do**
5:             $n_i \leftarrow$ pop from queue
6:             **for all** n_j in calibrated neighbors **do**
7:                 **if** n_j is gold standard **then**
8:                     $\mathbf{\Gamma}_j \leftarrow \Gamma$
9:                 $\boldsymbol{c}_j \leftarrow \mathrm{RLS}(d_{i,k}, x_{j,k})$
10:                 $\boldsymbol{r}_j \leftarrow \mathrm{SAMPLECORR}(d_{i,k}, x_{j,k})$
11:                 $\boldsymbol{v}_j \leftarrow \mathrm{DISTWEIGHT}(n_i, n_j)$
12:             $\boldsymbol{w} \leftarrow \boldsymbol{r} \cdot \boldsymbol{v} \cdot \mathbf{\Gamma}$
13:             $\alpha, \beta \leftarrow \mathrm{Mean}(\boldsymbol{c}, \boldsymbol{w})$

Additional Sensor Network Assumptions:

- We assume the n nodes form a graph, which we denote in this text as $\mathcal{G}$. We consider that two nodes form an edge if the Euclidean distance between them is less than some radius r. This intuitively means that we only consider the physical neighbors of a node. We require that every node in $\mathcal{G}$ have at least one edge.
- We assume that the constituent to be measured is perfectly known by at least one node in $\mathcal{G}$, which we refer to as the *Golden Standard*.
- We assume the graph follows a rational flow of communication typical for distributed sensor networks. Sensor nodes are configured in a mesh network and only communicate with their direct neighbors. If a new node is introduced to the network, it will follow an initialization process where it is made aware of its direct neighbors and will remember them.

B. Description of the Weighted Neighborhood Algorithm

Here we present the weighted neighborhood calibration algorithm, a partially-blind calibration algorithm for a distributed sensor network that uses one or more ground-truth reference nodes to seed the algorithm and percolate linear calibrations through the network. We refer to Algorithm 1 during the overview of the algorithm.

1) ***Overview****:* The weighted neighborhood calibration algorithm is a 2-step algorithm in which calibration is percolated through the network in a breadth-first manner, starting at the gold standard and calibrating nodes sequentially. In the first step, the considered sensor updates a linear regression between itself and its already calibrated neighbors. In the second step, the sensor computes an average of these regressions, weighted by correlation, distance, and sensor type. Once calibrated, the sensor can be used as a reference for calibration by its neighbor.

2) ***Percolation****:* We begin the algorithm by constructing a breadth-first queue containing the nodes in $\mathcal{G}$, with no duplicates, with the head of the queue being the node closest in distance to the gold standard node (Step 3). This queue will be reinitialized and emptied at every time step k (Step 2) . We consider the gold standard node, which we denote as n_{GS}, to be a *calibrated* node, because of our assumption that it perfectly measures the local stimulus.

3) ***Linear Regression****:* For the current sensor popped from the head of the queue (Step 5), call it n_i, there exists at least one neighbor of n_i that has already been calibrated at time step k. In the case of the completely full queue, the calibrated neighbor corresponds to n_{GS}. We say that there are m of these calibrated neighbors. Node n_i updates the RLS between itself and every one of its calibrated neighbors, so that we obtain a set of m linear regression equations, each containing gain (α) and offset (β) parameters (Step 9). At each iteration, the inputs to the algorithm are d_k, which is the measured value of n_i at time step k, and the vector $[x_j, 1]$, where x_j is the measurement from neighbor n_j after passing it through the current calibration equation for n_j. We will call this vector of linear regressions $\boldsymbol{c}$. The RLS algorithm is denoted as RLS in Algorithm 1. Now that node n_i has a set of calibration estimates from its neighbors, the next step is to score the importance of these estimates to better capture the dynamics of the environment.

4) ***Weighted Average****:* We consider three influences that impact the weights.

i **Skewing the gold standard.** The weight in $\boldsymbol{c}$ corresponding to the edge (n_i, n_{GS}) is heavily skewed to ensure a large portion of the calibration comes from here. We denote the gain applied to $\boldsymbol{c}_{GS}$ as Γ, and define the vector $\mathbf{\Gamma}$ as the all-ones vector with $\mathbf{\Gamma}_{GS} = \Gamma$.

ii **Correlation Based Weights.** We weight each element in $\boldsymbol{c}$ according to the Pearson correlation on the edge (n_i, n_j). In this way, estimates from highly correlated neighbors will be trusted more, and uncorrelated neighbors can be removed from the estimate. This is accomplished recursively using an iterative version of the Pearson correlation, called the *sample correlation coefficient*, which is described in detail in [28]. This is done in an identical fashion to the RLS, and we obtain the vector $\boldsymbol{r}$.

This is denoted as SampleCorr in Algorithm 1, and is referenced in Step 10.

iii **Distance Based Weights.** Highly heterogeneous environments may exhibit poor correlation on an edge. To this end the correlation-based weight distribution can be altered by including distance based weights in the model. Neighbors that are closer together will have a higher distance weight metric than neighbors at the edge of the communication range. This gain parameter is calculated as $(r/l)^2$, where r is the maximum communication radius and l is the Euclidean distance between sensors. We denote the distance based weight vector as $\boldsymbol{v}$. We also introduce an artificial bounding condition, λ, where $0 < \lambda < 1$ and $[r \cdot \lambda < l < r]$, to keep values in $\boldsymbol{v}$ reasonably constrained. Without the inclusion of λ co-located sensors may be completely dominated by proximity measurement, so l is limited to a degree. This is denoted as DistWeight in Algorithm 1, and is referenced in Step 11.

The resulting weight vector $\boldsymbol{w}$ is then the element-wise product of the three vectors: $\boldsymbol{w} = \boldsymbol{\Gamma} \cdot \boldsymbol{r} \cdot \boldsymbol{v}$. We then compute the weighted average of $\boldsymbol{c}$ using $\boldsymbol{w}$ as the weights (Steps 12 - 13):

$$\begin{bmatrix} \alpha & \beta \end{bmatrix} = \frac{1}{\sum \boldsymbol{w}} \boldsymbol{c} \bullet \boldsymbol{w}$$

Now (α, β) are the new calibration parameters for node n_i and n_i can be used as a calibrated neighbor for the next node in the network. In the next section, we show Algorithm 1 applied to a low-cost air pollution monitoring network currently deployed in an urban region.

C. *Proof-of-Concept Example*

Figure 1 shows a trivial network that may be used as a proof-of-concept example to demonstrate our algorithm. The gold standard node, labelled GS, serves as the seed for the algorithm at time-step k.

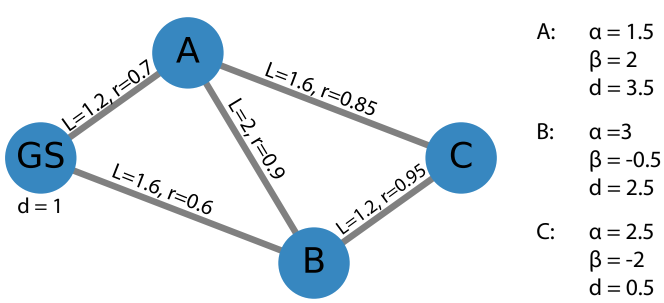

Fig. 1: Example network used to describe the process flow of the weighted neighborhood calibration algorithm. L refers to the Euclidean distance between nodes, and r refers to the Pearson correlation between nodes. The gain and offset parameters on the right-hand side represent the ideal calibration, which is unknown to the model. d refers to the value measured by the sensor. We arbitrarily define the maximum radius as $r = 2.5$.

1) We begin by queuing a list of nodes in a breadth-first approach, with A as the head because it is closest to GS. Our queue will then look like:

   $Q = \{A, B, C\}$

2) A computes one iteration of RLS with GS:

   $\text{RLS}(d_A = 3.5, x_{GS} = 1) = [1.5, 2]$

3) There are no other calibrated neighbors of A, so pop B from the queue. Both GS and A are calibrated neighbors, so compute one iteration of RLS with both.

   $\text{RLS}(d_B = 2.5, x_{GS} = 1) = [3, -0.5]$

4) B receives the calibrated measurement from A as input:

   $x_A = (d_A - \beta_A)/\alpha_A = 1$
   $\text{RLS}(d_B = 2.5, x_A = 1) = [3, -0.5]$

   B also updates the correlation and distance with GS and A:

   $\boldsymbol{r}_B = [0.6, 0.9]$
   $\boldsymbol{v}_B = [(2.5/1.6)^2, (2.5/2)^2]$

   We will use $\Gamma = 10$ for the gold standard weight, so that:

   $\boldsymbol{\Gamma}_B = [10, 1]$

5) Our weight vector is then $\boldsymbol{w}_B = [14.65, 1.4]$, and the normalized calibration parameters are:

   $\alpha = (3 * 14.65 + 3 * 1.4)/(14.65 + 1.4) = 3$
   $\beta = (-0.5 * 14.65 + -0.5 * 1.4)/(14.65 + 1.4) = -0.5$

6) Nodes A and B can then be used to calibrate C in the same fashion.

It should be noted that in the case of this example, the gain and offset parameters are calculated perfectly because of the absence of noise in the system.

IV. Experimental Setup and Results

In this section we produce and analyze two experiments performed using our weighted neighborhood calibration algorithm, and compare them to static laboratory calibrations, as well as the current state-of-the-art online calibration techniques. We simulate the algorithm using a network graph from an active low-cost pollution monitoring network, dubbed AirU.

A. *Methodology*

Our algorithm has been validated using a network of low-cost air pollution monitors currently deployed in Salt Lake City, Utah, USA – dubbed the AirU Low Cost Pollution Monitoring Network [29]. The network currently consists of 22 nodes and a single reference node (Gold Standard) that is centralized in the network. The degree of the Gold Standard is 7, so roughly 33% of the nodes are directly connected to the Gold Standard. We refer to the topology of this network as the graph $\mathcal{G}$. Figure 2 shows the graphical representation, $\mathcal{G}$, of the AirU network, where the gold circle refers to the

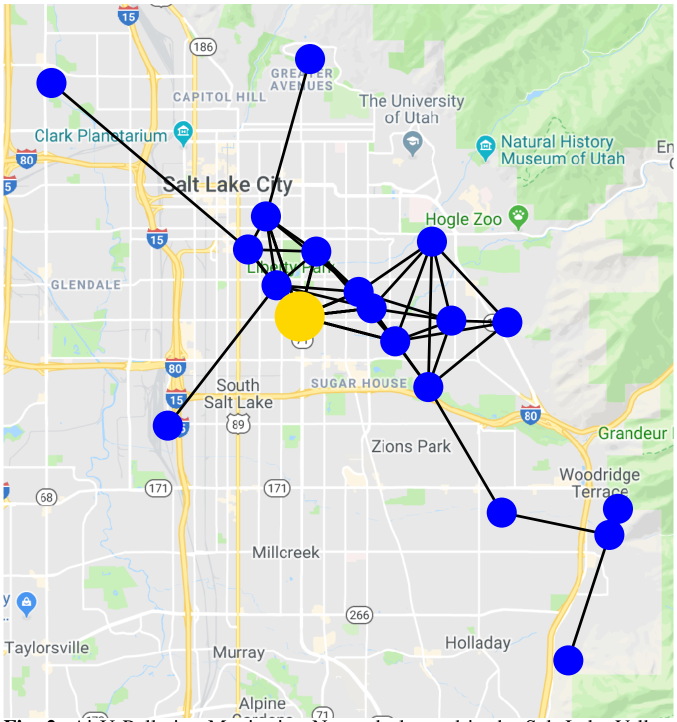

Fig. 2: AirU Pollution Monitoring Network, located in the Salt Lake Valley, USA. The network consists of 22 low-cost pollution monitoring stations, as well as a single "Gold Standard" monitor hosted by Utah Department of Air Quality [30].

Gold Standard. The dataset and network information were downloaded from the AirU open-source database.

The goal of our study is to quantify the measurement error between the AirU dataset calibrated using our algorithm and the ground-truth data. To this end, we consider a test dataset that fully captures the properties of the network, by comparing the laboratory calibrations to the real calibrations from a subset of 12 nodes collocated at the gold standard site. We also model the inter-sensor correlation using a multivariate normal distribution as additive measurement noise, which was derived from the measured AirU dataset.

Prior to deployment, the AirU particulate matter sensors were calibrated in a carefully scrutinized laboratory setting, using a regulatory grade reference. The sensors were assigned unique linear calibrations, which we will use in this section as a quality metric.

Throughout this section we refer to several datasets pertaining to the data collected from the AirU network.

- The *measured* dataset refers to values that the sensors measure. In the second experiment it refers to hourly measurements collected by the AirU nodes between July 01, 2018, and August 31, 2018.
- The *laboratory* dataset refers to the mapping of the measured dataset using the laboratory calibrations.
- The *ideal* dataset refers to actual pollution concentrations at the sensor locations.
- The *Weighted Neighborhood Calibration (WNC) Algorithm* dataset refers to the mapping of the measured dataset using calibrations from the weighted neighborhood calibration algorithm.
- The *GMR* dataset refers to the mapping of the measured dataset using calibrations produced with the GMR algorithm described in [23].
- The *Consensus* dataset refers to the mapping of the measured dataset using calibrations derived from a global average consensus, as proposed in [9].

Throughout the experiments we use the *Root Mean Squared Deviation* (RMSD) to refer to the error between two datasets. RMSD is a popular metric when comparing the difference of time-varying signals from the ideal signal [31].

B. Simulation in an ideal environment

We first perform an experiment on the AirU graph in a spatially homogeneous environment to investigate the properties of the weighted neighborhood calibration algorithm. The graph $\mathcal{G}$ was subjugated to a spatially homogeneous field that increased from 1 to 100 $\mu g/m^3$ over 1000 time steps.In this experiment, we refer to the measured dataset as the *ideal* dataset mapped to the ideal calibrations, with additive Gaussian noise modeled after the noise distribution of the sensors from laboratory experiments.

For each time step, the weighted neighborhood calibration algorithm was run on the modeled AirU network. The *improvement* is calculated as the ratio of error of the measured dataset over error of the given calibrated dataset. Results show that the weighted neighborhood calibration algorithm shows $4\times$ improvement over the raw, uncalibrated data, and also demonstrates a $2.8\times$ improvement over the laboratory dataset. With these results we prove that the weighted neighborhood algorithm captures the unique calibration of each sensor, even when that calibration changes based on the environment.

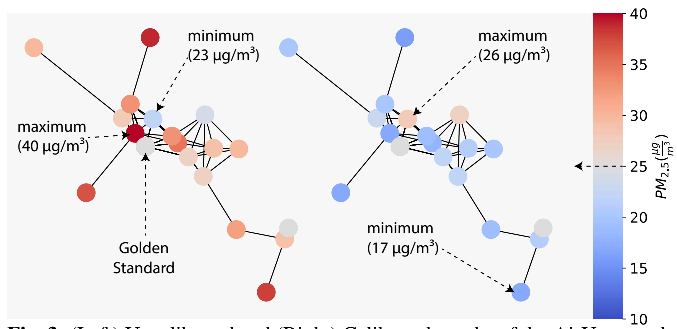

Fig. 3: (Left) Uncalibrated and (Right) Calibrated graphs of the AirU network. Calibration was performed using the weighted neighborhood calibration algorithm, in a homogeneous spatial field of 25 $\mu g \cdot m^{-3}$.

Figure 3 shows a graphical analysis of this simulation. The figure is a snapshot in time when the value of the spatial field to be measured is 25 $\mu g/m^3$. The graph on the left-hand side shows the measured dataset, and the right-hand side shows the algorithm dataset. The graph provides a good understanding of the difference in distributions of RMSD error between the uncalibrated and calibrated data. We also note that the nodes at the edge of the network are still misrepresenting their measurements more than nodes towards the center of the network, which contain many more edges, even after applying our algorithm. This can be attributed to the accumulation of distance and correlation weight degradation as the algorithm propagates from the reference node. This experiment also shows that a sensor with a large error (which corresponds to a large intrinsic gain and offset) in the raw measurements does not attribute a large error in the algorithm dataset. The RMSD in the algorithm dataset is instead largely due to the intrinsic additive noise from the sensor itself, as the gain and offset are well estimated.

In this experiment, we show that the weighted neighborhood calibration algorithm effectively converges to a close estimate of the ideal sensor calibrations, and outperforms the static, offline laboratory calibrations. The remaining RMSD from the algorithm dataset can be largely attributed to the intrinsic noise of each sensor.

C. Simulation using the AirU Dataset

An experiment was then performed on the simulated network using the AirU dataset. Here, we refer to the measured dataset as the AirU dataset, which corresponds to data downloaded directly from the AirU open-source database from July 01, 2018 to August 31, 2018. We now include the correlated noise between sensors as the additive sensor measurement noise to fully capture the properties of the AirU network deployed in a complex urban environment. The previous experiment was repeated on this dataset, and the results are shown in Table I.

TABLE I: RMSD results for different calibration techniques. *Measured* refers to the raw dataset of measured values. *Laboratory* refers to the measured dataset calibrated using the static, offline laboratory calibrations. *WNC$_C$* refers to the measured dataset calibrated using the weighted neighborhood (WNC) Algorithm with averaging weight based off sensor correlations. *WNC$_{C,D}$* refers to the measured dataset calibrated using the weighted neighborhood calibration algorithm, with averaging weight based off sensor correlations as well as distance-based weights. *Consensus* uses the global average across the network to calibrate the sensors [9]. *GMR* produces multi-hop calibrations from the reference to all nodes in the network [23].

Method	Min	Max	**Mean**	Improvement
Measured	10.378	30.803	16.508	-
Laboratory	9.273	27.509	13.586	1.215×
Consensus [9]	10.8	18.36	12.54	1.135×
GMR [23]	7.83	16.5	11.18	1.478×
WNC$_C$–This work	8.082	13.212	**9.410**	1.754×
WNC$_{C,D}$–This work	8.108	13.212	**9.402**	1.756×

The inclusion of sensor correlation statistics and the heterogeneous land-use model more accurately represent the actual error and improvement of our proposed algorithm, which still outperforms the static, offline laboratory calibrations, as well as current state-of-the-art techniques. We demonstrate that both WNC$_C$ and WNC$_C, D$ outperform macro-calibration via average consensus by 34%. Average consensus algorithms are used to compute global averages across distributed networks [32]. This technique is thus not well suited for our heterogeneous environment, as the results show. In previous work, a multi-hop GMR algorithm was able to significantly outperform an ordinary least squares approach [23]. However, that is not the case when applied to our model. In [23], a calibration would typically percolate through several nodes, thus producing a wide noise distribution between calibrations. The AirU network, and many other stationary, urban sensor networks, are generally within 1-2 hops of a reference sensor. It is therefore more appropriate to assume the calibrated sensor as an independent source in the model, and apply ordinary least squares. We see that our work outperforms the GMR approach by 20% when applied to the AirU network.

In Figure 4 we show the distribution of RMSD for each sensor as a result of this experiment under the measured dataset (a), laboratory calibrated dataset (b), and the weighted neighborhood calibrated dataset (c). The Weighted Neighborhood Algorithm decreased the RMSD relative to the laboratory calibrations on every node in the network. The average RMSD was lowered from 13.586 using the laboratory calibrations to 9.4 using the weighted neighborhood calibration algorithm. We see that the the weighted neighborhood calibration algorithm was able to effectively converge on the ideal calibration parameters of sensors with even the highest RMSD and bring them to roughly the same calibration estimate as sensors with an already low RMSD. We show that on average our proposed algorithm demonstrates 45% improvement over the laboratory dataset. We note here that the remaining RMSD error shown in Figure 4c can be largely attributed to the noise floor of each sensor, as is demonstrated by the uniform distribution of RMSD error between the sensors.

The drawbacks of the algorithm are demonstrated in Figure 5. After the experiment was performed, sensors were grouped based off the number of hops (number of intermediate graph edges) from the reference node to the sensor node. The RMSD for each sensor in the group for the AirU dataset was averaged and plotted. It is clear to see that the error follows a roughly linear increase as the path length between n_{GS} and n_i grows. This is expected and rational, as the error (mostly due to intrinsic noise of the sensor) introduced by each sensor is accumulated as the calibration percolates outward from the reference node. As the number of nodes in the network grows, the error per hop will tend towards a linear trend proportional to the average noise floor of the sensors. In many target applications, this error may be mitigated by additional ground-truth reference nodes in the network. For example, in the AirU network a second ground-truth reference exists in the northwest corner, but was omitted in the context of this research for simplicity of the simulations.

D. Algorithm Response to Nonlinear Changes

Figure 6 shows the calibration response of two sensors from the AirU network subjugated to an abrupt change in the sensing environment (denoted as *Field* in Figure 6). Prior to the change the environment was temporally static at 1 $\mu g/m^3$, then underwent a jump to 25 $\mu g/m^3$ in a single time step, and then was temporally static. This step is equivalent to a sharp change in the calibration parameters (offset of one and gain of zero), and is meant to simulate the dynamic change in environmental factors which directly affect the calibration of the sensors. Using the setup from the static environment

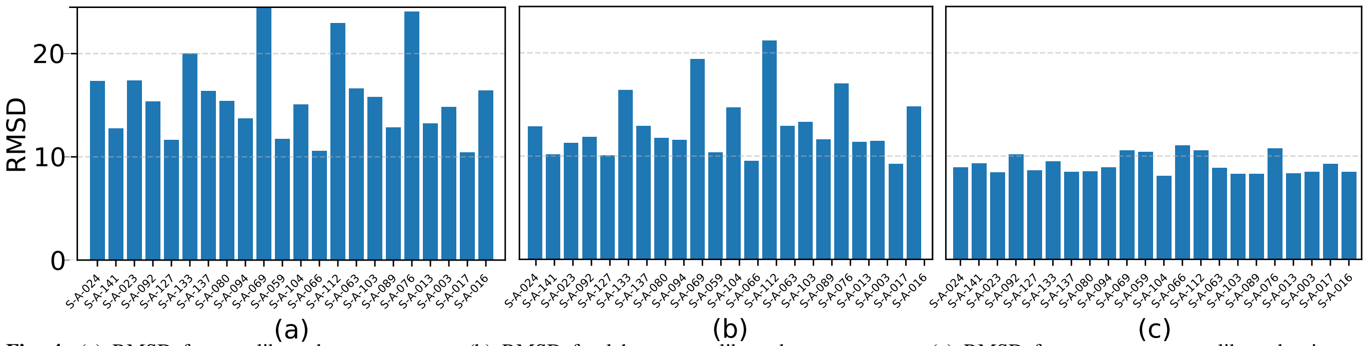

Fig. 4: (a) RMSD for uncalibrated measurements. (b) RMSD for laboratory-calibrated measurements. (c) RMSD for measurements calibrated using our algorithm. All RMSD metrics were computed by comparing the above data to the ideal calibration.

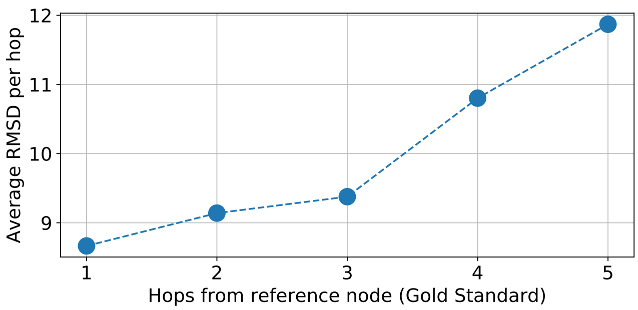

Fig. 5: Average RMSD, grouped by number of hops from gold standard. Measurements are from the AirU dataset simulation.

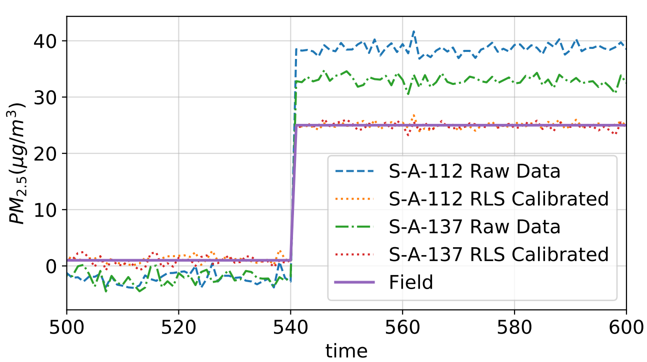

Fig. 6: Tracking capabilities of the weighted neighborhood calibration algorithm. The network is subjected to a homogeneous spatial field, that has an abrupt change in value. Where the uncalibrated sensors largely overestimate the actual field value, the weighted neighborhood calibration algorithm quickly updates the calibration to account for the sudden change.

simulation, we observe the fast tracking capabilities of the filter to abrupt changes in the environment. We arbitrarily pick two sensors from the network to demonstrate. We show the raw and calibrated responses of the sensors when subjugated to an initial concentration of 1 $\mu g/m^3$. We see that the calibration estimates a concentration much closer to the real value than the raw values. After the abrupt shift in concentration, which will inevitably change the sensor's calibration statistics, we see an extremely fast response from the algorithm to incorporate this change. In a temporally static environment such as the one demonstrated here, it only takes several iterations for a sensor to converge to its new calibration estimate, because the feedback error calculation in the RLS algorithm allows the sensor to correct the majority of its error in a single iteration. This experiment demonstrates the ability of the algorithm to dynamically update the sensors' calibrations, which may be necessary due to environmental changes that directly affect the sensing technique, or linear drift as the sensors age.

V. Conclusion

The calibration of wireless sensor networks is a necessary step to map a measured dataset to a close approximation of the actual stimulus. Sensors may be calibrated prior to deployment in a carefully controlled laboratory setting, but this approach does not scale well as the network grows, and is subject to error when deployed in a real environment that does not conform to the laboratory constituents. Instead, it is desirable to adaptively calibrate the network while it is online and deployed, where the environmental conditions may be taken into consideration. In this paper, we proposed a *partially*-blind calibration scheme in which a decentralized and distributed sensor network has access to at least one ground-truth reference (such as a federally maintained sensor which follows NIST standards), which is considered as an additional network node. The neighbors of the ground-truth node calibrate themselves directly using the reference. The calibration then percolates through the network as nodes use their calibrated neighbors to calibrate themselves. The algorithm described in this work was designed to operate on a distributed, low-power network that is measuring to a highly heterogeneous environment, such as urban airborne pollution microclimates. Calibrations produced via sample correlation and recursive least squares techniques keep the algorithm extremely lightweight while maintaining accuracy. Results using a real dataset from a currently deployed sensor network show a 45% improvement over pre-deployment laboratory calibrations, and 20% improvement over current state-of-the-art distributed online calibration solutions. We also demonstrate the ability of the algorithm to quickly adapt to abrupt changes in the environment, which would change the intrinsic gain and offset parameters of the sensors.

Acknowledgment

This work was supported by the National Science Foundation under Grant No. 1646408. The authors would like to thank Pascal Goffin and Tony Butterfield. Pierre-Emmanuel Gaillardon and Kerry Kelly have financial interest in the company Tetrad: Sensor Network Solutions, LLC, which manufactures air quality sensing solutions and provides engineering services.

References

[1] P. Corke *et al.* Environmental wireless sensor networks. *Proceedings of the IEEE*, 98(11), 2010.
[2] D. Estrin *et al.* Next century challenges: Scalable coordination in sensor networks. *International Conference on Mobile Computing and Networking*, 1999.
[3] T. Dinh *et al.* Design and deployment of a remote robust sensor network: Experiences from an outdoor water quality monitoring network. *IEEE Conference on Local Computer Networks*, 2007.
[4] J. Beutel *et al.* Permadaq: A scientific instrument for precision sensing and data recovery in environmental extremes. *International Conference on Information Processing in Sensor Networks*, 2009.
[5] J. Yick *et al.* Wireless sensor network survey. *Computer Networks*, 52(12), 2008.
[6] J. Binnig *et al.* Calibration of an optical particle counter to provide pm2.5 mass for well-defined particle materials. *Aerosol Science*, 38(3), 2007.
[7] R. Tan *et al.* System-level calibration for fusion-based wireless sensor networks. *IEEE Real-Time Systems Symposium*, 2010.
[8] M. Masinde and A. Bagula. A calibration report for wireless sensor-based weatherboards. *Journal of Sensor and Actuator Networks*, 4(1), 2015.
[9] J. Yang, X. Zhong, and W.P. Tay. A dynamic bayesian nonparametric model for blind calibration of sensor networks. *IEEE Internet of Things Journal*, 2018.
[10] J. Lipor and L. Balzano. Robust blind calibration via total least squares. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014.
[11] V. Bychkovskiy *et al.* A collaborative approach to in-place sensor calibration. *Information Processing in Sensor Networks*, 2003.
[12] A. Clements *et al.* Field performance evaluation of four low-cost particulate matter sensors. *Air Sensor International Conference*, 2018.
[13] AQ-SPEC. Field Evaluation: Purple Air (PA-II) PM Sensor. Technical report, South Coast Air Quality Management District, 2015.
[14] AQ-SPEC. Field Evaluation: aeroQUAL S500 OZU. Technical report, South Coast Air Quality Management District, 2015.
[15] AQ-SPEC. Field Evaluation: AlphaSense OPC-N2 Sensor. Technical report, South Coast Air Quality Management District, 2015.
[16] H. Zhi-gang and C. Cai-hui. The application of zigbee based wireless sensor network and gis in the air pollution monitoring. *2009 International Conference on Environmental Science and Information Application Technology*, 2, 2009.
[17] K. Kelly *et al.* A layered framework for integrating low-cost sensor data and for engaging citizens to understand pm2.5 exposure. *International Aerosol Conference*, 2018.
[18] K. Kelly *et al.* Receptor model source attributions for utah's salt lake city airshed and the impacts of wintertime secondary ammonium nitrate and ammonium chloride aerosol. *Journal of the Air & Waste Management Association*, 63(5), 2013.
[19] K. Whitehouse and D. Culler. Calibration as parameter estimation in sensor networks. *ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.
[20] L. Balzano and R. Nowak. Blind calibration of sensor networks. *International Symposium on Information Processing in Sensor Networks*, 2007.
[21] M. Stanković *et al.* Distributed calibration for sensor networks under communication errors and measurement noise. In *IEEE Conference on Decision and Control*, 2012.
[22] V. Bychkovskiy *et al.* A collaborative approach to in-place sensor calibration. *Information Processing in Sensor Networks*, 2003.
[23] O. Saukh *et al.* Reducing multi-hop calibration errors in large-scale mobile sensor networks. In *Conference on Information Processing in Sensor Networks*, New York, NY, USA, 2015. ACM.
[24] M. Stanković, S. Stanković, and K. H. Johansson. Asynchronous distributed blind calibration of sensor networks under noisy measurements. *IEEE Transactions on Control of Network Systems*, 5(1), 2018.
[25] P. Rathore *et al.* Maximum entropy-based auto drift correction using high- and low-precision sensors. *ACM Trans. Sen. Netw.*, 13(3), 2017.
[26] Y. Wang *et al.* A deep learning approach for blind drift calibration of sensor networks. *IEEE Sensors Journal*, 17(13), 2017.
[27] A.H. Sayed and T. Kailath. *Recursive Least-Squares Adaptive Filters*. CRC Press, IEEE Press, 1998.
[28] F. Croxton *et al.* *Applied General Statistics*. Pitman, 1968.
[29] Pascal Goffin. Air quality in salt lake city, http://www.aqandu.org/, 2019.
[30] Google. Salt Lake City, UT [Online]. Accessed Oct. 12, 2018.
[31] T. Chai and R. R. Draxler. Root mean square error or mean absolute error: Arguments against avoiding rmse in the literature. *Geoscientific Model Development*, 7(3), 2014.
[32] S. Kar and J. Moura. Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise. *IEEE Transactions on Signal Processing*, 57(1), 2009.