Network Calibrarion of Embedded Sensors

Chris Papachristou, Swarup Bhunia, Frank Wolff
Case Western Reserve University
Cleveland, Ohio 44106, USA

Abstract

The focus of this project is to provide methods for calibration of sensor nodes in sensor networks. The importance of the calibration problem is to compensate for the sensor reading drifts that occur due to systematic errors, noise or sensor degradation. The objective is to provide calibration techniques that apply on collaborative sensors autonomously, i.e. without supervision. Our work involves: a) distributed procedures to identify erroneous sensors; and b) a simulator framework for a sensor net drift calibration setups.

1. Overview

• **Motivation**. The deployment of sensor networks has been growing in scale and scope with new applications in embedded and challenging environments. This progress has been facilitated by advances in nanoscale electronics that can be integrated with MEMS optical and biochemical technologies to build tiny sensor nodes, [1]. However, there are several issues and difficulties in building such sensor nets. In addition to reliability concerns of the sensor nodes and their communications, there is a significant problem with the *drift* of sensor readings from their correct values. A variety of sensor applications use MEMS for realizing the transduction mechanism. Along with degradations in the micromechanical parts due to environmental effects, wear and tear [2, 3], the signal conditioning and read-out circuitry also suffer from both systematic and random degradations inducing errors in the sensor reads.

Sensor calibration has been used over time to correct the reading drifts. Traditionally, calibration is applied on sensors at the micro level, meaning on individual sensor nodes either at the factory or in the network off-line. However, micro-level calibration may not be feasible due to many difficulties such as remote access, security, size of sensor net and device degradations. There is need for network based calibration, i.e. autonomous calibration by collaborating sensor nodes.

• **Related Work**. The problem of calibration of sensor nodes has received considerable attention in research works. Earlier calibration was performed on each sensor at the factory or in the field using simple built-in techniques [4]. Calibration in traditional sensor networks is still being done individually.

Most of the recent research has focused on location discovery of sensor nodes. The early work on SpotON [5] modeled the signal to distance relation between transmitting and receiving sensors off-line. Another project targeting location discovery is Calamari, [6], which formulates a calibration approach to sensor localization as a parameter estimation problem. A two-phase collaborative technique is used in [7] where first all pair-wise calibration functions are found and then they are optimized to produce the global calibration function for the net. In the SCAAT project [8], a mathematical technique is used for tracking position and orientation while performing sensor autocalibration. Dynamic fine-grain localization is reported in [9]. More recently. a calibration approach to location estimation is taken in [10] using nonlinear least-squares optimization. Nonparametric statistical techniques are used in [11] to introduce an error model of the calibration process.

Calibration research addressing tracking based on signal processing and filtering is reported in [13]. Additional work on sensor ranging issues is reported in [14, 15]. The exposure problem in wireless sensor nets is discussed in [16]. Calibration work on robotic sensors is in [17, 18]. The work in [17] exploits the intrinsic sensor mobility to effect the calibration process. Calibration issues concerning chemical and biosensors are discussed in [19, 20]. Methods for drift compensation in humidity, pressure and gas sensors have been developed based on Neural Networks [21, 22], using wavelet transforms [23], Kalman filtering [24], and other learning techniques [25]. These methods are computationally too expensive for low-cost sensors.

• **Objectives**. The goal of this work is to develop a calibration methodology for embedded or dense sensor nets to address error readings due to sensor degradations. Our approach uses distributed techniques, driven by collaborating nodes, to discover erroneous sensors. These techniques are employed on the network autonomously to identify sensors with errors, Our research project provides calibration procedures and design techniques to address these sensor drift problems. The following significant differences distinguish our work from the ones cited above: a) Most of the previous sensor calibration research do not deal with the network-based and the self-calibration problem; b) the few proposed self calibration techniques [16] do not provide distributed algorithms to address the self calibration problem; c) a simulator framework for a sensor net drift calibration setups.

2. Background

Sensors are small devices positioned in possibly adverse environment for monitoring and measuring phenomena occurring, for example, in physical, chemical, or biological processes. Often sensors operate collaboratively in a network communicating their measurements to remote gateways or stations. Sensor networks can be wireless, for instance sensors monitoring environmental conditions of an open area, or embedded, e.g. biosensors linked into monitoring human or animal body parts. Miniature sensors or microsensors concen-

978-1-4244-6578-1/10/$26.00 ©2010 IEEE

trate in dense networks to provide collaborative measurements of localized points. Examples of dense sensor networks are in applications such as smart dust and smart clothing [26].

It is well known that the measured values of sensors may deviate from the actual values due to the following reasons: a) manufacturing tolerance of electromechanical components [4]; b) process variation and degradation in the electronic device [27]; c) external effects such as noise [4]. Traditionally, calibration has been used to address these errors. Calibration of equipment or devices is a systematic method to adjust a measured value to the correct value. Depending on the nature of reading errors, there are at least three kinds of calibration approaches addressing the following errors: 1) Systematic errors. This is the usual calibration approach correcting the bias inaccuracies due to manufacturing tolerances. Moreover, semiconductor process variation may also produce device variance and reading errors. 2) Device degradation errors. There are several device characteristics that can be degraded after manufacturing, i.e. in the field, which may affect both the performance and the output values. 3) Noise errors. These may be due to transient events, hardware noise or other effects that influence sensor readings.

Normally, calibration due to systematic errors is initially performed on single sensors during manufacturing. This concerns both component tolerance and device process variation. Calibration is still needed at setup time in the field to account for the environment. However, there are issues with single sensor calibration in large sensor networks. Some of them according to [6] are: a) limited access to the sensor network and b) complex dynamic environmental effects. A network calibration for systematic bias is proposed in [7] based on sensor redundancy. We need to make a note about two issues, orientation and distance, that may affect sensor readings. Except in [16], previous works do not account for the "angularization" issue of the sensors in the field. Suppose there is a collection of sensors s_1, s_2, ... all measuring the same physical variable with respect to a reference point X. Depending on the topology, each sensor has its own angular orientation to the reference point. Our view is that the effect of this orientation may be reflected as bias on the sensor readings and corrected by systematic calibration. The distance between sensors and the reference point may also affect their reading. In other words, two sensors s_1 and s_2 that have the same orientation towards X but different distances from X may well get different readings. In many cases, the reading values may simply be proportional to the distance. If the relation between sensor reading and distance is known from the factory we may also be able to use systematic calibration of the distance effect on the sensor readings. About noise, we may assume that it can be filtered out, e.g. by the well known technique of averaging over time.

The focus of this work is on calibration of errors due to device degradation effects. It has been observed that several device characteristics can be degraded after manufacturing leading to degradation of important parametric aspects such as latency, energy consumption of the sensor device level components. This degradation not only influences the sensor output readings but it may also affect the sensor performance in terms of power and timing. We consider such degradations of the output value, the power, and the timing as sensor errors that need calibration. Systematic calibration can not be used to correct device degradation effects. Note, calibration addressing sensor power or performance has not been reported. In the following we discuss a new network calibration method that applies to sensor errors due to device degradations.

3. Network Based Calibration

• **Concepts**. Suppose $\mathcal{S} = \{s_1, ..., s_n\}$ is a set of n sensors in a dense sensor network. The objective is to measure collectively a certain physical variable with respect to a reference point X. We assume that each sensor is equipped with limited computational capability (e.g. no division), limited memory and limited energy. Sensors can communicate one-to-one wirelessly, or via a fabric matrix (e.g. smart clothing). We also assume that $\mathcal{S}$ has been calibrated for systematic bias, sensor orientation and distance.

We now consider the device degradation effects on sensor readings. Suppose $s_k(t)$ denotes the output reading of the k-th sensor at time t. Let $s'_k(t)$ denote the true output value of the k-th sensor at time t. We observe that because of systematic calibrations all true sensor outputs at time t with respect to X should be equal, i.e. $s'_1(t) = s'_2(t) = ... = s'_n(t)$. The deviation of the sensor s_k output at time t is $\Delta_k(t) = s_k(t) - s'_k(t)$ whereas its *drift* is defined as $\delta_k(t) = |\Delta_k(t)|$. Note the deviation $\Delta_k(t)$ may be positive or negative as the degradation affecting s_k may produce larger or smaller output values than the true value. We assume that the sensor drift is changing over time but it does so slowly. This means that the sensor drift can be stationery within a reasonable time window. In other words, it varies like a staircase step function in time. The drift of a sensor is *acceptable* if $\delta_k(t) = |s_k(t) - s'_k(t)| \leq \epsilon$ where ϵ, the *threshold* drift, is a small positive number established by the sensor specifications. By definition, all sensors satisfying the above relation constitute a sensor *bundle*. In what follows we will relate sensor reading statistics to the threshold drift and sensor bundles.

In this work, we assume that the statistical distribution of the sensor readings over a time window t is close to normal. In support of this claim, we remark that if the sensors in $\mathcal{S}$ comprise an unbiased sample of sensors from the overall factory production, then their readings may well fit into the normal distribution, in accord with the statistical sampling theory. The real scenario we expect is that some of the sensor readings will be *outliers*, i.e. either "too low" or "too high". Under this scenario, the sensor readings have a focused distribution with low standard deviation σ. This also means that the mean μ and the median of the readings are close to each other within ϵ, and further, they are also close to the true sensor readings s'_k, for all $k = 1, ..., n$. It should be noted, however, that despite our assumption about near normality of the readings distribution, we do not actually know their statistics μ and σ apriori. Nonetheless, it would be interesting to consider the relationship between the threshold drift ϵ and the standard deviation σ of the sensor statistics. In some cases it would suffice to assume $\epsilon \approx 2\sigma$, at least if we go by the well known 68% rule

of normal distributions. Since we do not actually know μ and σ, one idea is to use *hints* on ϵ values by experimentation.

3.1. Calibration Procedures and Strategies

The first problem we have concerning the sensor calibration under device degradation is as follows: given a set of readings of the sensor net $\mathcal{S}$ over a time window t and a threshold drift ϵ, find an optimal sensor bundle, i.e. a maximal subset of sensors $\{s_k\}$ that satisfy the above threshold drift relation. By "maximal subset" we mean a subset containing a maximal number of sensors.

The rationale of this problem goes as follows. Initially, we do not assume that we know what is the true s'_k value. However, under the normal distribution assumption, the mean μ is close to s'_k. Therefore, an optimal bundle (maximal subset) of sensors should include μ and hence the true s'_k. All sensors within the optimal bundle will be considered as having acceptable readings and thus should not need calibration. The rest of sensors outside the optimal bundle are outliers and would need calibration.

Before discussing our strategy to solve this problem, we remark that the intent of our approach is to use distributed algorithms as they apply to a dense sensor network, autonomously. Clearly, distributed techniques over the net are more fault tolerant than centralized ones employed on a single node. Our minimal sensor requirements are:
a) point-to-point message communication – limited data transfers; b) simple arithmetic operations (e.g. no multiplication and division) – limited memory and energy consumption; c) broadcast to all, or selectively some, sensors certain signals; d) distributed time awareness.

We assume that all sensors are identified by indexing IDs such as $1, 2, ..., n$. Initially, each sensor contains a numerical label or token, $Z(s_k) = z_k, \quad k = 1, 2, ..., n$. It is understood that the labels are ordered as $z_1 < z_2 < ... < z_n$. In other words, the sensors can identify that $z_5 < z_6$, thus z_1 and z_n are the minimum and maximum labels, respectively.
Our approach involves three major procedures.
Procedure 1 – find the sensor with the maximum reading;
Procedure 2 – sorting the sensors by their readings;
Procedure 3 – constructing sensor bundles and sliding the bundles until discovering the optimal one.
In the following we propose distributed based solutions to the above parts of the problem.
Procedure 1. We propose a distributed searching algorithm based on a heap binary tree composed of the sensor nodes. The idea is to pairwise compare the sensor readings and selectively change label pairs. Following are the steps of this procedure illustrated by the example of Fig. 1.
1) The first step of the algorithm begins by engaging the following sensor pairs {1,2}, {3,4} ,..., $\{(n-1), n\}$, concurrently, with each node pair comparing their readings. There two procedural parts: a) Relabelling and b) Re-indexing. a) Whenever a node pair realizes that the comparison result is in reverse order to their corresponding labels, then they interchange their labels. More precisely, if $s_k \geq s_{k+1}$ and $z_{k+1} < z_k$, then the nodes exchange their labels, i.e $Z(s_k) = z_{k+1}$, and $Z(s_{k+1}) = z_k$; otherwise, the nodes keep the same labels. At the end of this step, the node with the largest label in each pair is the *winner* node. b) The winner node is issued a new (additional) index ID for broadcasting purposes. The new index is the maximum index of the two engaging nodes divided by 2 (easily done by shifting). Thus if nodes 3 and 4 are engaged, the winner of 3 and 4 gets a new index equal to 2. The winners still keep their original indices.
2) We continue with the next steps by engaging the winner nodes of the previous step, relabelling and reindexing as needed. The node pair engagement will be done on the basis of newly issued index IDs. Thus in step k the node pairs are: $\{1, 2\}, \{3, 4\}, ..., \{2^{m-k-1} - 1, 2^{m-k-1}\}$ assuming for simplicity that $n = 2^m$. There is only one set of indices issued in each step, previous indices are discarded.
3) In every step, there is need for broadcasting synchronization signals, as follows. i) A winner node broadcasts a signal to the other nodes indicating its readiness for engagement. ii) The appropriate winner node, i.e. its mate in the corresponding node pair, responds when ready. iii) This is done by all winners that are not engaged.
4) The procedure terminates when a single winner node remains whose label is then z_n, the largest label. Because of point-to-point communication, the complexity of this algorithm in the number of steps is $\mathcal{O}(logn)$.

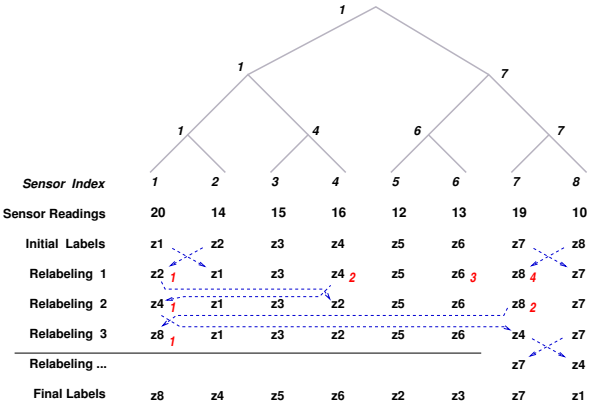

Figure 1. Distributed maximum search and sorting example

The above algorithm is illustrated by the example of Fig. 1 showing eight sensors, $1, ..., 8$, their reading values and their initial labels, $z_1, ..., z_8$. Relabelling 1 occurs for sensor pairs $1, 2$ and $7, 8$. After two more relabelling, the process terminates with node 1 obtaining label z_8 corresponding to the minimum reading value 10. The re-indexing of the winner nodes is also shown, for example, the node with original index 7 is a winner in step 2 and re-indexed as 4. Note, relabelling of two nodes may require additional relabelling to maintain the validity of the labelling rules, as shown in Fig. 1, last line.
Procedure 2. For the sorting part of the problem, we can work on the same node heap structure of Fig. 1 proceeding to finding the second largest sensor reading, and so on. More specifically, once the maximum reading node is found, here node 1, this node is removed from the heap. Then, the searching process for the next maximum resumes on the remaining nodes. This step, i.e. Procedure 1, is repeated through iterations till

the heap contains one node, the minimum reading node. Note each maximum node needs to broadcast a reset signal to all others to restart the process. The overall complexity of both methods is at worst $\mathcal{O}(nlogn)$. The intermediate steps of this process are not shown in Fig. 1.

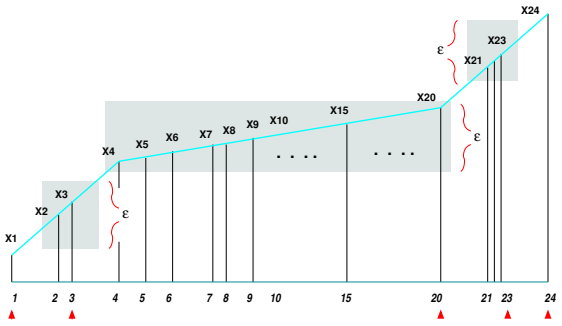

Figure 2. sensor bundle groupings

Procedure 3 – **Optimal Bundle**. We will now propose a distributed procedure to determine bundles of sensors within the threshold drift ϵ. First we assume that the sensors have been already sorted by their reading values as described earlier. The procedure is best described by reference to the example Fig. 2. This figure represents the sensors $s_1, ..., s_{24}$ and their readings $X_1, X_2, ..., X_{24}$ in incremental order, respectively. Note the sensor indices $1, 2, ...$ now correspond to their ranking labels z_k (dropping z), as discussed earlier, shown in the horizontal axis of Fig. 2. The procedure may begin with the smallest or the largest sensor readings. The steps follow.
1) Begin with sensor s_1 (smallest reading), send $X_1+\epsilon$ to sensors $s_2, s_3, ..., s_{k-1}$; test in each sensor that $X_1 + \epsilon < X_{k-1}$. This step terminates at the first sensor s_{q_1} that $X_1+\epsilon \geq X_{q_1}$. This occurs at $q_1 = 2$, Fig. 2.
2) Mark sensor s_{q_1-1} by M_{1,q_1-1} indicating the sensor bundle $B_1 = \{s_1, ..., s_{q_1-1}\}$. The size of the bundle is $|B_1| = q_1-1$. In Fig. 2 the first bundle contains only s_1.
3) Continue with sensor s_k, $k \geq q_1$ repeating steps 1) and 2) as above until finding the next marker sensor, s_{k+q_2-1} marked as M_{q_1,q_2-1}. A new bundle is obtained namely $B_2 = \{s_{q_1}, ..., s_{q_2-1}\}$ with size $|B_2| = q_2 - q_1$. In our running example, this marker sensor is s_3 and the bundle is $B_2 = \{s_2, s_3\}$.
4) At the end of step 3) the two successive markings M_{1,q_1-1} and M_{q_1,q_2-1} are compared. If $|B_2|-|B_1| = q_2-q_1 < q_1-1$ then M_{q_1,q_2-1} is discarded whereas M_{1,q_1-1} is to be passed along to the sensors for the next bundle. Otherwise, marking M_{q_1,q_2-1} is maintained as active for the next steps. In our example, marking $M_{2,3}$ remains active while M_1 is removed.
5) The procedure continues forming bundles until reaching the last one which includes the last sensor s_n. There is only one marker active from bundle to bundle, in the end the remaining marker corresponds to the optimal bundle.

Continuing with our example, the third and fourth bundles are $B_3 = \{s_4, ..., s_{20}\}$, $B_4 = \{s_{21}, ..., s_{24}\}$, with markers $M_{4,20}$, $M_{21,23}$, respectively, and marker $M_{4,20}$ wins, meaning B_3 is the optimal bundle. After deriving the optimal bundle, any sensor within it could be used as reference for calibration, for example, the marker sensor reading. Incidently, the sensor net can not directly compute the mean or other statistics because its nodes lack arithmetic capability, e.g. division.

The complexity of the above procedure appears to be linear with n. However, there is an implicit assumption that we do not consider overlapping bundles. The latter can be considered by modifying step 3) above to include the omitted sensors. Although some of the bundles may be subsets of others, in general there may well be overlapping bundles. If we include all bundles, then the complexity may be $\mathcal{O}(n^2)$ at worst. However, we conjecture that if the distribution of sensor readings is about normal, there is very high chance that the above linear procedure will find a near optimal bundle which will include all sensors of the optimal except few outliers.

One more comment about the bundle approach. Procedures 1 and 2 do not require knowledge of the sensor reading statistics whereas Procedure 3 is very efficient for near normal distributions. Nonetheless, the proposed bundle approach can be adjusted to apply to other reading distributions that are not normal, for example, bimodal type of distributions, or distributions with several peak concentrations, or monotonically incrementing characteristics.

3.2. Sensor Mesh Calibration

A mesh or array of sensors is sometimes an attractive topology for a sensor net. This is the case for sensor fabrics in smart clothing but also of acoustic or ultrasound sensors spread over mesh like nets on the ground or underwater for security applications. Other applications concern array radar sensors such as phased array radars or MIMO array radars. In our case, we will consider the calibration problem of a mesh like sensor net as illustrated in Fig. 3.

The calibration problem of this net can be approached by the distributed searching and sorting methods provided earlier. However, we will address this problem taking into consideration the mesh topology using near-neighbor communication of sensors rather than point to point communications. This means that sensors can transmit directly horizontally, vertically and diagonally in the network mesh. Assuming a $m \times m$ mesh, our approach is as follows.
Step 1: Perform distributed searching on each row mesh.
Step 2: Perform inter row searching in the mesh.
Performing row searching in m sensor row can be done in k sessions where k=$logm$. As shown in Fig. 3 (left), session 1 compares adjacent sensors to produce a winner; session 2 generates session 2 winners, and session k, $k = logm$, generates the row winner. Note the number of sensor transmission hops over all k sessions is
$T_m = 1 + (2^2 - 1) + ... + (2^k - 1) = 2^{k+1} - (k + 1) = 2m - (1 + logm) = \mathcal{O}(m)$.
For inter row searching, we compare the row winners using k session comparisons, as shown in Fig. 3 (right). In each session a winner needs to traverse at most $m - 1$ hops to reach a corresponding winner. For example, if sensors s_{00} and s_{37} are winners in session 1, then 7 traversal hops are needed for comparison to produce the winner in session 2. The total number of hops are: $(m - 1)logm$; Thus the overall number of hops

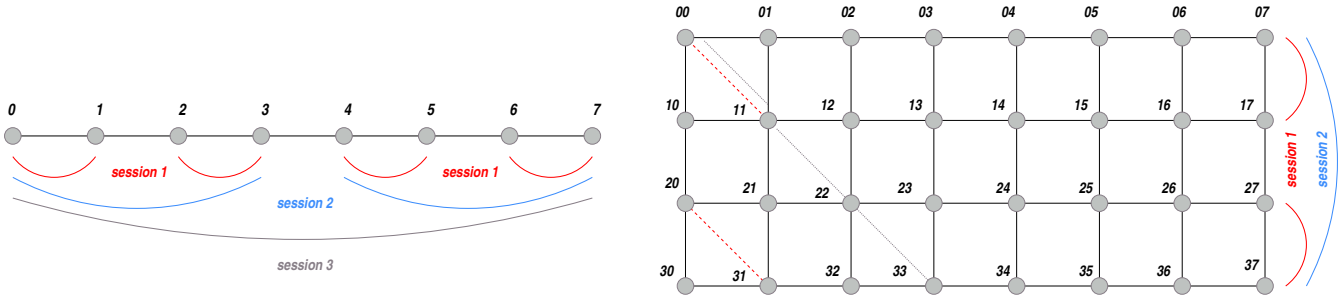

Figure 3. Calibration of mesh like sensor net. a) Row processing; b) inter-row processing

for the $n = m \times m$ mesh is
$T_n = 2m - (1 + logm) + (m - 1)logm = \mathcal{O}(mlogm)$
Now to complete the sorting we will need at most $n \times mlogm$ hops and since $n = m^2$ we need $\mathcal{O}(n^{1.5}logn)$ overall hops. Although the distributed processing on the mesh seems to be longer than in point to point communication, however, the mesh is more realistic and it does account for variation in distances among the sensors.

• **Sensor parameter gradient**. Our basic assumption about our calibration process is that the sensor readings of a parameter are invariant over the net within a time window. However, this assumption may not be valid for some applications such as temperature sensors over a vineyard field. In other words, there may be a significant temperature gradient over the sensor net. This is an important issue affecting calibration in some practical applications and requires further investigation. We have made some observations which may help out in this regard.

Let $s_k(t)$ be the k-th sensor reading, e.g. temperature, *without* any additional gradient. If we assume there is an extra gradient spreading over the sensor mesh from row to row we can express this by an extra value $Y_k(t)$. Thus the actual sensor reading is $Y_k(t), =, s_k(t) + Y_k(t)$. Then, $E[Z] = E[s] + E[Y]$, where $E[Z]$, $E[s]$ and $E[Y]$ are the expected values of the distributions of the variables Z, s and Y, respectively. This means if the statistics of the gradient parameter Y are known then the previously proposed bundle searching method can be applied since $E[s] = E[Z] - E[Y]$. To estimate these statistics, one idea is to sample each sensor to obtain readings within a short time window right after installation, when the drift is zero. This will capture average reading for each sensor and lead to an average gradient estimate. This technique can be repeated after each calibration when the drift is again zero.

4. Discussion of Other Calibration Issues

The previous procedures separate from the sensor net those sensors that have potential degradation problems. We will consider several methods to deal with these outlier sensors. a) Treat the problem by systematic calibration; b) discard the sensors as being uncalibratable;

a) Temporary Systematic Calibration. It may be possible to treat the unacceptable sensor drifts by systematic calibration using the base sensor readings obtained by Procedure 3 as reference. This entails mapping of a degraded sensor reading, for example, s_2 in Fig. 2, to the base reading in the optimal bundle, s_{10}. For this to work, we will need to perform this action over a range of sensor readings to obtain a meaningful calibration mapping table. One could also use the curve fitting method of [11] for mapping. However, these mappings may only be valid over a time window, because the degradation effects may progressively increase in some sensors, which then requires the whole process to be repeated. Thus systematic calibration can only be used as a temporary fix.

b) Uncalibratable sensors. In some cases, it may be determined that a sensor is not calibratable, for instance if some parametric characteristic has deteriorated, beyond recovery. This of course depends on the sensor net operating policy. In that case, it may be worth decommissioning those sensors.

Size and Scalability. There is a question as to how Procedures 1, 2 and 3 scale with the size of the net n. We address this issue for both the point to point and the mesh like sensor nets. Given a sensor net $\mathcal{S}$ with $|\mathcal{S}| = n$, consider $\mathcal{R}$ being a random sample of sensors taken from $\mathcal{S}$ with size $|\mathcal{R}| = m$ such that $n \gg m$. Assuming again that the probability distribution of readings is close to normal, then for sure the probability distribution of sensors from $\mathcal{R}$ will also be close to normal. This means, Procedures 1, 2 and 3 would apply on $\mathcal{R}$ with the benefit of much smaller sensor subnet size m. Then we would have an optimal bundle of sensors in $\mathcal{R}$ which could be used as good reference readings to calibrate the entire sensor net.

Another idea to address scalability is to introduce monitor sensors. These sensors are not used to measure any physical phenomena, their sole purpose is to monitor the rest of sensors in the net. Then the entire sensor net may be clustered around each monitor sensor based on some proximity distance sensor-to-monitor scheme. For example in Fig. 3, sensor 11 could be a monitor of the sensors on the surrounding rectangle. We may assume that the monitors have greater computational capability than the main sensors and thus Procedures 1, 2, 3 could be implemented on each monitor. The entire processing will now require two steps. a) Centralized processing: the sensors in each cluster will transfer their readings to their monitors to locally perform Procedures 1, 2, 3. b) Distributed processing: Once the monitors get their results, the distributed Procedures 1,2,3 will be performed as stated earlier. The validity of the 2-step process is based on the fact that the mean of the entire distribution is equal to the mean of its constituent cluster

means – captured by the monitors.

Consideration of Power and Heat type Errors. There is a possibility that there are no output reading drifts in the sensor net, yet there may still be some sensors exhibiting parameter errors, for example, overheating due to power consumption. Apparently, this behavior is undesirable as it may lead to exhaustion of the sensors power source, or breakdown due to overheating. We now address the problem of such power or heating type of sensor errors while their readings and other functions are not impaired. Before discussing approaches to solve this problem, the following questions need to be settled. a) Can a sensor measure its own heat? b) can two sensors measure the heat of one another? c) is the sensor's ability to measure heat impaired by its overheating condition?
The first two questions are raised because the sensor net may have been designed to measure certain external physical parameters, thus the above heat measuring is additional capability to be embedded in the sensors. The answer to the third question is not clear.

Our first approach to this problem is to assume that sensor measuring capabilities are available and not impaired by overheating. Then, this problem can be solved by adaptation of the techniques used in Procedures 1, 2 and 3 for normal distributions of the sensor heatings. These procedures will identify the overheated sensors from their normal heating distribution. One difference here is that we would not expect any "underheated" sensors hence the distribution statistics σ and ϵ should be carefully adjusted. The problem is more difficult if there are both heat and heat measurement errors. This problem may be very difficult to solve if the sensors can only measure their own heat and then "lie" about their readings. However, it may be possible to solve this problem if we assume that the sensors have the capability to measure the heat of one another. it will be necessary then to find a method that can filter out the erroneous readings.

Note, if we assume the availability of the monitor sensors discussed earlier, then the heat error problem can be addressed differently by exploiting the monitors. Assuming that the monitors can read the heat of the other sensors, the problem can be formulated and solved by a similar technique discussed earlier. Of course, the monitor sensors will need to be more robust in their design fabrication properties.

In conclusion, this work provides a calibration methodology for embedded or dense sensor nets to address error readings due to sensor degradations. Our approach uses distributed techniques, driven by collaborating nodes, to discover erroneous sensors. This investigation provides a simulator framework for sensor net drift calibration setups.

References

[1] P. R. Hauptmann, "Selected examples of intelligent (micro) sensor systems: state-of-the-art and tendencies," *Meas. Sci. Technol.*, 17, No 3, pp. 459-466, 2006.

[2] S. Tadigadapa and N. Najafi, "Reliability of microelectromechanical systems (MEMS)," *Proceedings of SPIE*, vol. 4558, 2001.

[3] D. M. Tanner, *et al*., "The Effect of Humidity on the Reliability of a Surface Micromachined Microengine," Proc. of *IRPS*, 1999.

[4] K. Stum, "Sensor accuracy and calibration theory and practical application," *National Conf. on Building Commissioning*, April 19-21, 2006.

[5] J. Hightower, C. Vakili, G. Borriello, "Design and calibration of the SpotON ad-hoc location sensing system," *UW CSE*, University of Washington, Seattle, August 2001.

[6] K. Whitehouse, D. Culler, "Calibration as parameter estimation in sensor networks and applications," *ACM Internat. Workshop on Wireless Sensor Networks and Applications (WSNA-02)*, 67, 2002.

[7] V. Bychkovskiy, S. Megerian, D. Estrin, M. Potkonjak, A collaborative approach to in-place sensor calibration, *2nd Internat. Workshop on Information Processing in Sensor Networks (IPSN'03)*, April 2003.

[8] G. Welch, G. Bishop, "SCAAT: Incremental tracking with incomplete information," *Proc. ACM SIGGRAPH*, pp. 333-344, Aug 1997.

[9] A. Savides, C. Han, M.B. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," *MobiCom'01*, pp. 104-111, July 2001.

[10] A. T. Ihler, J. W. Fisher, "Nonparametric belief propagation for self-calibration in sensor networks," *IPSN'04*, pp. 225-233, April 2004.

[11] J. Feng, G. Qu, M. Potkonjak, "Sensor calibration using nonparametric statistical characterization of error models," *IEEE Sensors*, pp. 1456-1459, October 2004.

[12] W. Tsujita, et al, "Sensor network for atmospheric environmental monitoring and its auto-calibration," *Proc. Int. Conf. Electrical Engineering*, Vol. 3-1, pp. 132 135, 2004.

[13] V. Cevher, J. McClellan, "Sensor array calibration via tracking with the extended Kalman filter," *IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing* Vol. 5, 2001.

[14] W. Merrill, L. Girod, J. Elson, K. Sohrabi, F. Newberg, W. Kaiser, "Autonomous position location in distributed, embedded, wireless systems," *IEEE CAS Workshop on Wireless Communications and Networking*, September 2002.

[15] W. Merrill, F. Newberg, L. Girod, K. Sohrabi, "Battlefield ad-hoc LANs: a distributed processing perspective," *29th Annual GOMATech Conference*, Monterey, March 2004.

[16] S. Meguerdichian, F. Koushanfar, G. Qu, M. Potkonjak, "Exposure in wireless ad-hoc sensor networks," *7th Internat. Conf. on Mobile Computing and Networking*, 2001.

[17] M. M. Howard, G. Sukhatme, "Relaxation on a mesh: a formalism for generalized localization," *IROS* 2001.

[18] D. Ma, J. M. Hollerbach, "Identifying mass parameters for gravity compensation and automatic torque sensor calibration," *IEEE Intl. Conf. Robotics and Automation*, pp. 661-666, April 1996.

[19] R. Byrne, D. Diamond, "Chemo/bio-sensor networks," *Nature Materials*, Vol 5, June 2006.

[20] A. Orsini, A. D Amico, "Chemical sensors and chemical sensor systems: Fundamentals limitations and new trends," *Advances in Sensing with Security Applications*, NATO Advanced Study Institute, July 2005.

[21] T. Islam and H. Saha, "Study of long-term drift of a porous silicon humidity sensor and its compensation using ANN technique," *Sensors and Actuators A*, 133, pp. 472-479, 2007.

[22] J. C. Patra, G. Panda and R. Baliarsingh, "Artificial neural network-based nonlinearity estimation of pressure sensors," *IEEE Trans. Instrum. Meas.* 43, 874-881, 1994.

[23] J. Ma, J. Q. Zhang, and Y. Yan, "Wavelet transform based sensor validation," *IEE Colloquium on Intelligent and Self-Validation Sensors*, pp. 10/1-10/4, 1999.

[24] D. Gobbo, M. Napolitano, P. Famouri, and M. Innocenti, "Experimental application of extended Kalman filtering for sensor validation," *IEEE Trans. Control Syst. Technol.*, vol. 9, no. 2, pp. 376-380, Mar. 2001.

[25] K. Goebel and W. Yan, "Correcting sensor drift and intermittency faults with data fusion and automated learning," *IEEE Systems Journal*, Vol. 2, No. 2, Jun 2008.

[26] S. Mann, "Smart clothing: The wearable computer and wearcam," *J. Personal and Ubiquitous Computing*, Vol. 1, pp. 21-27, March 1997.

[27] S Borkar, T Karnik, V De, "Design and reliability challenges in nanometer technologies," *Design Automation Conf. (DAC)*, June 2004.