1

Data Fusion Approach for Error Correction in Wireless Sensor Networks

Maen Takruri
Centre for Real-Time Information Networks (CRIN)
University of Technology, Sydney
Australia

Subhash Challa
NICTA Victoria Research Laboratory
The University of Melbourne
Australia

1. Introduction

Wireless Sensor Networks (WSNs) emerged as an important research area (Estrin et al., 2001). This development was encouraged by the dramatic advances in sensor technology, wireless communications, digital electronics and computer networks, enabling the development of low cost, low power, multi-functional sensor nodes that are small in size and can communicate over short distances (Akyildiz et al., 2002). When they work as a group, these nodes can accomplish far more complex tasks and inferences than more powerful nodes in isolation. This led to a wide spectrum of possible military and civilian applications, such as battlefield surveillance, home automation, smart environments and forest fire detection.

On the down side, the wireless sensors are usually left unattended for long periods of time in the field, which makes them prone to failures. This is due to either sensors running out of energy, ageing or harsh environmental conditions surrounding them. Besides the random noise, these cheap sensors tend to develop drift in their measurements as they age. We define the drift as a slow, unidirectional long-term change in the sensor measurement. This poses a major problem for end applications, as the data from the network becomes progressively useless. An early detection of such drift is essential for the successful operation of the sensor network. In this process, the sensors, which otherwise would have been deemed unusable, can continue to be used, thus prolonging the effective life span of the sensor network and optimising the cost effectiveness of the solutions.

A common problem faced in large scale sensor networks is that sensors can suffer from bias in their measurements (Bychkovskiy et al., 2003). The bias and drift errors (systematic errors) have a direct impact on the effectiveness of the associated decision support systems. Calibrating the sensors to account for these errors is a costly and time consuming process. Traditionally, such errors are corrected by site visits where an accurate, calibrated sensor is used to calibrate other sensors. This process is manually intensive and is only effective when the number of sensors deployed is small and the calibration is infrequent. In a large scale sensor network, constituted of cheap sensors, there is a need for frequent recalibration. Due to the size of such networks, it is impractical and cost prohibitive to manually calibrate them. Hence, there is a significant need for auto calibration (Takruri & Challa, 2007) in sensor networks.

The sensor drift problem and its effects on sensor inferences is addressed in this work under the assumption that neighbouring sensors in a network observe correlated data, i.e., the measurements of one sensor is related to the measurements of its neighbours. Furthermore, the physical phenomenon that these sensors observe also follows some spatial correlation. Moreover, the faults of the neighbouring nodes are likely to be uncorrelated (Krishnamachari & Iyengar, 2004). Hence, in principle, it is possible to predict the data of one sensor using the data from other closely situated sensors (Krishnamachari & Iyengar, 2004; Takruri & Challa, 2007). This predicted data provides a suitable basis to correct anomalies in a sensor's reported measurements. At this point, it is important to differentiate between the measurement of the sensor or the reported data which may contain bias and/or drift, and the corrected reading which is evaluated by the error correction algorithms. The early detection of anomalous data enables us not only to detect drift in sensor readings, but also to correct it.

In this work, we present a general and comprehensive framework for detecting and correcting both the systematic (drift and bias) and random errors in sensor measurements. The solution addresses the sparse deployment scenario of WSNs. Statistical modelling rather than physical modelling is used to model the spatio-temporal cross correlations among sensors' measurements. This makes the framework presented here likely to be applicable to most sensing problems with minor changes. The proposed algorithm is tested on real data obtained from the Intel Berkeley Research Laboratory sensor deployment. The results show that our algorithm successfully detects and corrects drifts and noise developed in sensors and thereby prolongs the effective lifetime of the network.

The rest of the chapter is organised as follows. Section 2 presents the related work on error detection and correction in WSNs literature. We present our network structure and the problem statement in Section 3. Sections 4 and 5 formulate the Support Vector Regression and Unscented Kalman Filter framework for error correction in sensor networks. Section 6 evaluates the proposed algorithm using real data and section 7 concludes with future work.

2. Related Work

The sensor bias and drift problems and their effects on sensor inferences have rarely been addressed in the sensor networks literature. In contrast, the bias correction problem has been well studied in the context of the multi-radar tracking problem. In the target tracking literature the problem is usually referred to as the *registration problem* (Okello & Challa, 2003; Okello & Pulford, 1996). When the same target is observed by two sensors (radars) from two different angles, the data from those two sensors can be fused to estimate the bias in both sensors. In the context of image processing of moving objects, the problem is referred to as *image registration*, which is the process of overlaying two or more images of the same scene taken at different times, from different viewpoints, and/or by different cameras. It geometrically aligns two images: the reference and sensed images (Brown, 1992). Image registration is a crucial step in all image analysis tasks in which the final information is gained from the combination of various data sources like in image fusion (Zitova & Flusser, 2003). That is, in order to fuse two sensor readings, in this case two images, the readings must first be put into a common coordinates systems before being fused. The essential idea brought forth by the solution to the registration problem is the augmentation of the state vector with the bias components. In other words, the problem is enlarged to estimate not only the states of the targets, using the radar measurements for example, but also the biases of the radars. This is the approach we consider in the case of sensor networks. Target tracking filters, in conjunction with sensor drift models are used to estimate the sensor drift in real time. The estimate is used for correction and as a feedback to the next estimation step. The presented methodology is a robust framework for auto calibration of sensors in a WSN.

A straightforward approach to bias calibration is to apply a known stimulus to the sensor network and measure the response. Then comparing the ground truth input to the response will result in finding the gain and offset for the linear drifts case (Hoadley, 1970). This method is referred to by (Balzano & Nowak, 2007) as non-blind calibration since the ground truth is used to calibrate the sensors. Another form of non-blind calibration is manually calibrating a subset of sensors in the sensor network and then allowing the non-calibrated sensors to adjust their readings based on the calibrated subset. The calibrated subset in this context form a reference point to the ground truth (Bychkovskiy, 2003; Bychkovskiy et al., 2003). The above mentioned methods are impractical and cost prohibitive in the case of large scale sensor networks.

The calibration problem of the sensor network was also tackled by (Balzano & Nowak, 2007; 2008) in a different fashion. They stated that after sensors were calibrated to the factory settings, when deployed, their measurements would differ linearly from the ground truth by certain gains and offsets for each sensor. They presented a method for estimating these gains and offsets using subspace matching. The method only required routine measurements to be collected by the sensors and did not need ground truth measurements for comparison. They referred to this problem as blind calibration of sensor networks. The method did not require dense deployment of the sensors or a controlled stimulus. However, It required that the sensor measurements are at least slightly correlated over space i.e. the network over sampled the underlying signals of interest. The theoretical analysis of their work did not take noise into consideration and assumed linear calibration functions. Therefore, the solution might not be robust in noisy conditions and will probably result in wrong estimates if applied in a scenario where the relationship between the measurement and the ground truth is nonlinear. The evaluations they presented showed that the method worked better in a controlled environment.

An earlier work on blind calibration of sensor nodes in a sensor network was presented in (Bychkovskiy, 2003; Bychkovskiy et al., 2003). They assumed that the sensors of the network under consideration were sufficiently densely deployed that they observed the same phenomenon. They used the temporal correlation of signals received by neighbouring sensors when the signals were highly correlated to derive a function relating the bias in their amplitudes. Another method for calibration was considered by (Feng et al., 2003). They used geometrical and physical constraints on the behaviour of a point light source to calibrate light sensors without the need for comparing the measurement with an accurate sensor (ground truth). They assumed that the light sensors under consideration suffered form a constant bias with time.

The authors in (Whitehouse & Culler, 2002; 2003) argued that calibrating the sensors in sensor networks is a problematic task since it comprises large number of sensor that are deployed in partially unobservable and dynamic environments and may themselves be unobservable. They suggested that the calibration problem in sensor/actuator networks should be expressed as a parameter estimation problem on the network scale. Therefore, instead of calibrating each sensor individually to optimise its measurement, the sensors of the network are calibrated to optimise the overall response of the network. The joint calibration method they presented calibrated sensors in a controlled environment. The method was tested on an ad-hoc localisation system and resulted in reducing the error in the measured distance from 74.6% to 10.1%. The authors claimed that the joint calibration method could be transformed into an auto calibration technique for WSNs in an uncontrolled environment i.e. some form of blind calibration where the value of the ground truth measurement (here the distance) is unknown. They formulated the problem as a quadratic programming problem. Similar to (Whitehouse & Culler, 2002; 2003), blindly calibrating range measurements for localisation purposes between sensors using received signal strength and/or time delay were considered in (Ihler et al., 2004; Taylor et al., 2006).

The work of (Elnahrawy & Nath, 2003) aimed to reduce the uncertainties in the sensors readings. It introduced a Bayesian framework for online cleaning of noisy sensor data in WSNs. The solution was designed to reduce the influence of random errors in sensors measurements on the inferences of the sensor network but did not address systematic errors. The framework was applied in a centralised fashion and on synthetic data set and showed promising results.

The author of (Balzano, 2007) described a method for in-situ blind calibration of moisture sensors in a sensor network. She used the Ensemble Kalman Filter (EnKF) to correct the values measured by the sensors, or in other words, to estimate the true moisture at each sensor. The state equation was governed by a physical model of moisture used in environmental and civil engineering and the measurements were assumed to be related to the real state by a certain offset and gain. The state (moisture) vector was augmented with the calibration parameters (gain and offset) and then the gains and offsets were estimated to recover the correct state from the measurements.

Another method for detecting a single sensor failure that is a part of an automation system (a sort of wired sensor network) was proposed by (Sallans et al., 2005). Using the incoming sensor measurement, a model for the sensor behaviour was constructed and then optimised using an online maximum likelihood algorithm. Sensor readings were compared with the model. In event that the sensor reading deviated from the modelled value by a certain threshold, the system labelled this sensor as faulty. On the other hand, when the difference was small, the system automatically adapted to it. This made the system capable of adapting to slow drifts.

A neural network-based instrument surveillance, calibration and verification system for a chemical processing system (a sort of wired sensor network) was introduced in (Xu et al., 1998). The neural network used the correlation in the measurements of the interconnected sensors to correct the drifting sensors readings. The sensors that were discovered to be faulty were replaced automatically with the best neural network estimate thus restoring the correct signal. The performance of the system depended on the degree of correlation of the sensors readings. It was also found that the robustness of the monitoring network was related to the amount of signal redundancies and the degree of signal correlations. The authors concluded that their system could be used to continuously monitor sensors for faults in a plant. However, they noted that retraining the entire network may be necessary for major changes in plant operating conditions

Support Vector Machines (SVM) were used in (Rajasegarar et al., 2007) to detect anomalies and faulty sensors of a sensor network. The data reported by the sensors were mapped from the input space (the space where the features are observed) to the feature space (higher dimensional space) using kernels. The projected data were then classified into clusters and the data points that did not lie in a normal data cluster were considered anomalous. The sensor that always reported anomalous data was considered faulty.

The authors of (Guestrin et al., 2004) presented a method for in-network modelling of sensor data in a WSN. The method used kernel linear regression to fit functions to the data measured by the sensors along a time window. The basis functions used were known by the sensors. Therefore, if a sensor knew the weights of its neighbour, it would be able to answer any query about the neighbour within the time window. So instead of sending the measured data of the whole window period from one sensor to another, sending the weights would considerably reduce the communication overhead. This was one of the aims of the method. The other aim was to enable any sensor in the network to estimate the measured variable at points within the network where there were no sensors using the spatial correlation in the network. An application for the introduced method is computing contour levels of sensor values as in (Nowak & Mitra, 2003). Even that the work in (Guestrin et al., 2004) considered the unreliable communication between distant sensors and the noise in sensor readings, it did not address the systematic errors (drift and bias) which can build up along time and propagate among sensors causing the continuously modelled functions to produce estimates that deviate from the ground truth values.

In addition to its superb capabilities in generalisation, function estimation and curve fitting, Support Vector Machines (SVR) is used in other applications such as forecasting and estimating the physical parameters of a certain phenomenon. In (Wang et al., 2003), SVR was utilised in medical imaging for nonlinear estimation and modelling of functional magnetic resonance imaging (fMRI) data to reflect their intrinsic spatio-temporal autocorrelations. Moreover, SVR was used in (Gill et al., 2006) to successfully predict the ground moisture at a site using meteorological parameters such as relative humidity, temperature average solar radiation, and moisture measurements collected from spatially distinct locations. A similar experiment to predict ground moisture was reported in (Gill et al., 2007). In addition to using the SVR to predict the moisture measurements ahead in time, they introduced the use of an EnKF to correct or match the predicted values with the real measurements at certain points of time (whenever measurements are available) to keep the predicted values close to the measurements taken on site and eventually reduce the prediction error.

The above survey, has introduced most of the work undertaken in the area of fault detection and fault detection/correction in wireless sensor networks. This research approaches the problem in a more comprehensive manner resulting in several novel solutions for detecting and correcting drift and bias in WSNs. It does not assume linearity of the sensor faults (drift) with time and addresses smooth drifts and drifts with sudden changes and jumps. It also considers the cases when the sensors of the network are densely and sparsely (non densely) deployed. Moreover, it introduces recursive online algorithms for the continuous calibration of the sensors. In addition to all of that, the solutions presented are decentralised to reduce the communication overhead. Some of the papers that have arisen from this research are surveyed below: (Takruri & Challa, 2007) introduced the idea of drift aware wireless sensor network which detects and corrects sensors drifts and eventually extends the functional life time of the network. A formal statistical procedure for tracking and detecting smooth sensors drifts using decentralised Kalman Filter (KF) algorithm in a densely deployed network was introduced in (Takruri, Aboura & Challa, 2008; Takruri, Challa & Chacravorty, 2010). The sensors of the network were close enough to have similar temperature readings and the average of their measurements was taken as a sensible estimate to be used by each sensor to self-assess. As an upgrade for this work, the KFs were replaced in (Takruri, Challa & Chacravorty, 2010; Takruri, Challa & Chakravorty, 2008) by interacting multiple model (IMM) based filters to deal with unsmooth drifts. A more general solution was considered in (Takruri, Rajasegarar, Challa, Leckie & Palaniswami, 2008). The assumption of dense sensor deployment was relaxed. Therefore, each sensor in the network ran an SVR algorithm on its neighbours' corrected readings to obtain a predicted value for its measurements. It then used this predicted data to self-assess its measurement, detect (track) its drift using a KF and then correct the measurement.

A more robust and reliable decentralised algorithm for online sensor calibration in sparsely deployed wireless sensor networks was presented in (Takruri, Rajasegarar, Challa, Leckie & Palaniswami, 2010). The algorithm represents a substantial improvement of method in (Takruri, Rajasegarar, Challa, Leckie & Palaniswami, 2008). By using an Unscented Kalman Filter (UKF) instead of the KF, the bias in the estimated temperature (system error) was dramatically reduced compared to that reported in (Takruri, Rajasegarar, Challa, Leckie & Palaniswami, 2008). This is justified by the fact that UKF is a better approximation method for propagating the mean and covariance of a random variable through a nonlinear transformation than the KF is. The algorithm was then upgraded in (Takruri et al., 2009) to become more adaptable for under sampled sensor measurements and consequently, allowing for reducing the communication between sensors and maintain the calibration. This led to reducing the energy consumed from the batteries. Unlike the work in (Balzano, 2007), statistical modelling rather than physical relations was used to model the spatio-temporal cross correlations among the sensors measurements. Similar to (Takruri, Rajasegarar, Challa, Leckie & Palaniswami, 2008), statistical modelling was achieved by applying SVR. This in principal made the framework applicable to most sensing problems without needing to find the physical model that describes the phenomenon under observation, and without the need to abide by the constraints of that physical formulation. The algorithm runs recursively and is fully decentralised. It does not make assumptions regarding the linearity of the drifts as opposed the work in (Balzano & Nowak, 2007). The implementation of the algorithm on real data obtained from the Intel Berkeley research laboratory (IBRL) showed a great success in detecting and correcting sensors drifts and extending the functional lifetime of the network.

In this chapter, we present another model for error detection and correction in sparsely deployed WSNs. Similar to (Takruri, Rajasegarar, Challa, Leckie & Palaniswami, 2010), SVR is used to model the spatio-temporal cross correlations among the sensors measurements to obtain a predicted value for the actual ground truth measurements and Unscented Kalman Filter is used to estimate the corrected sensors readings. However, both algorithms are substantially different in terms of the training data set used for training the SVR framework, the dynamic equations that govern the models and the estimated variables. The state transition function in the new model is taken to be linear resulting in much lower computational complexity than (Takruri, Rajasegarar, Challa, Leckie & Palaniswami, 2010) and comparable results.

3. Network Structure and Problem Statement

Consider a wireless sensor network with a large number of sensors distributed randomly in a certain area of deployment such as the one shown in Figure 1. The sensors are grouped in clusters (sub-networks) according to their spatial proximity. Each sensor measures a phenomenon such as ambient temperature, chemical concentration, noise or atmospheric pressure. The measurement, say temperature, is considered to be a function of time and space. As a result, the measurements of sensors that lie within the same cluster can be different from each other. For example, a sensor closer to a heat source or near direct sunlight will have readings higher than those in a shaded region or away from the heat source. An example of a cluster is shown using a circle in Figure 1. The sensors within the cluster are considered to be capable of communicating their readings among each other.

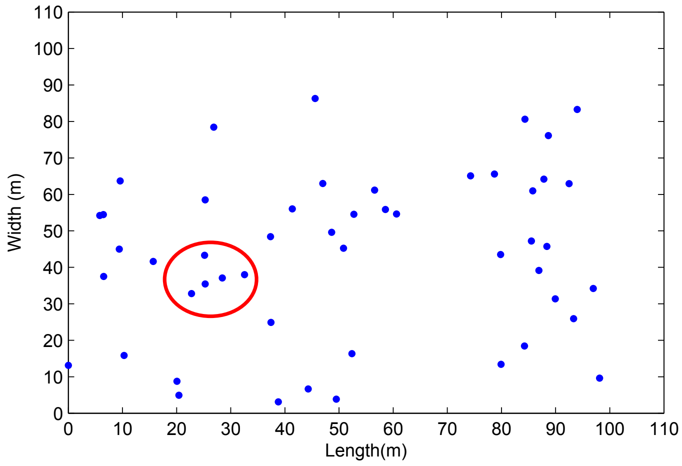

Fig. 1. Wireless sensor area with encircled sub-network

As time progresses, some nodes may start experiencing drift in their readings. If these readings are collected and used from these nodes, they will cause the users of the network to draw erroneous conclusions. After some level of unreliability is reached, the network inferences become untrustworthy. Consequently, the sensor network becomes useless. In order to mitigate this problem of drift, each sensor node in the network has to detect and correct its own drift using the feedback obtained from its neighbouring nodes. This is based on the principle that the data from nodes that lie within a cluster are correlated, while their faults or drifts instantiations are likely to be uncorrelated. The ability of the sensor nodes to auto-detect and correct their drifts helps to extend the effective (useful) lifetime of the network. In addition to the drift problem, we also consider the inherent bias that may exist within some sensor nodes. There is a distinct difference between these two types of errors. The former changes with time and often becomes accentuated, while the latter, is considered to be a constant error from the beginning of the operation. This error is usually caused by a possible manufacturing defect or a faulty calibration.

The sensor drift that we consider in this work is slow smooth drift that we model as linear and/or exponential function of time. It is dependent on the environmental conditions, and strongly relate to the manufacturing process of the sensor. It is highly unlikely that two electronic components fail in a correlated manner unless they are from the same integrated circuit. Therefore, we assume that the instantiations of drifts are different from one sensor to another in a sensor neighbourhood or a cluster. Figure 2 shows examples of the theoretical models for smooth drift.

Consider a sensor sub-network that consists of n sensors deployed randomly in a certain area of interest. Without loss of generality, we choose a sensor network measuring temperature, even though this is generally applicable to all other types of sensors that suffer from drift and bias problems. Let T be the ground truth temperature. T varies with time and space. Therefore, we denote the temperature at a certain time instance and sensor location as $T_{i,k}$ where i is the sensor number and k is the time index. At each time instant k, node i in the sub-network measures a reading $r_{i,k}$ of $T_{i,k}$. It then estimates and reports a *drift corrected* value $x_{i,k}$ to its neighbours. The corrected value $x_{i,k}$ should ideally be equal to the ground truth temperature $T_{i,k}$. If all nodes are perfect, $r_{i,k}$ will be equal to the $T_{i,k}$, and the reported values will ideally be equal to the readings, i.e., $x_{i,k} = r_{i,k}$.

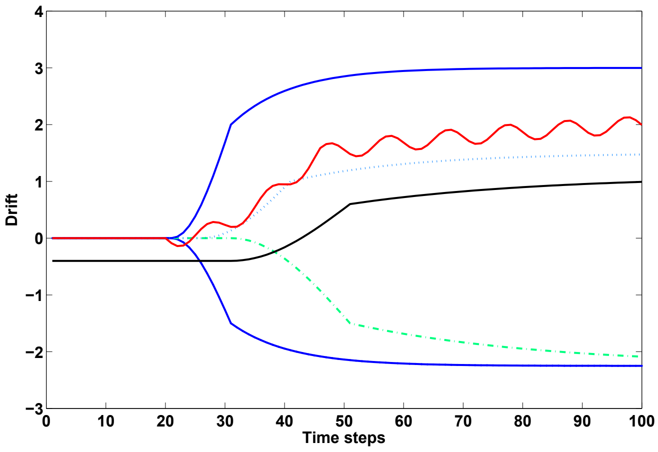

Fig. 2. Examples of smooth drifts

To estimate the corrected value $x_{i,k}$, each node i first finds a predicted value $\widetilde{x}_{i,k}$ for its temperature as a function of the corrected measurements collected from its neighbours in the previous time step using $\widetilde{x}_{i,k} = f(\{x_{j,k-1}\}_{j=1,j\neq i}^{n})$. Then it fuses this predicted value together with its measurement $r_{i,k}$ and the projected drift $d_{i,k}$ to result in an error corrected sensor measurement $x_{i,k}$. In practice, each sensor reading comes with an associated random reading error (noise), and a drift $d_{i,k}$. This drift may be null or insignificant during the initial period of deployment, depending on the nature of the sensor and the deployment environment. The problem we address here is how to account for the drift in each sensor node i, using the predicted value $\widetilde{x}_{i,k}$, so that the reading $r_{i,k}$ is corrected and reported as $x_{i,k}$.

In the following sections, $\widetilde{x}_{i,k}$ is computed using a support vector regression (SVR) modelled function that takes into account the temporal and spatial correlations of the sensor measurements. In this work, SVR approximates $\widetilde{x}_{i,k}$ using the previous corrected readings of all the sensors in the neighbourhood (cluster) excluding the sensor itself $\widetilde{x}_{i,k} = f(\{x_{j,k-1}\}_{j=1,j\neq i}^{n})$.

4. Modelling and predicting measurements using Support Vector Regression

The purpose of using Support Vector Regression (SVR) is to predict the actual sensor measurements $\widetilde{x}_{i,k}$ of a sensor node i at time instant k using the corrected measurements from neighbouring sensors. The intention is that each sensor learns a model function $f(.)$ that can be used for predicting its subsequent actual (error free) measurements through out the whole period of the experiment. SVR implements this in two phases, namely the *training phase* and the *running phase*. During the training phase, sensor measurements collected during the initial deployment period (training data set) are used to model the function $f(.)$. During the *running phase*, the trained model $f(.)$ is used to predict the subsequent actual sensor measurements $\widetilde{x}_{i,k}$.

We assume that the training data (collected during the initial periods of deployment) is void of any drift and can be used for training the SVR at each node. This is a reasonable assumption in practice, as the sensors are usually calibrated before deployment to ensure that they are working in order. Similar to our work in (Takruri, Rajasegarar, Challa, Leckie & Palaniswami, 2010), we use the widely used Gaussian kernel SVR for our evaluations (Scholkopf & Smola, 2002). However, the training data set used here is slightly different in that it comprises the corrected readings of the neigbours and does not take into consideration the corrected reading of node i itself. The training data set at each node i is given by $X_s = (TrX, TrZ)$, where $TrX = \{x_{j,k-1} : j = 1...n, k = 1...m, j \neq i\}$, $TrZ = \{x_{i,k} : k = 1...m\}$ and m is number of training data vectors. A detailed explanation of our implementation of the SVR can be found in (Takruri, Rajasegarar, Challa, Leckie & Palaniswami, 2010).

The model obtained via SVR training is then used during the *running phase* for predicting subsequent actual measurements $\widetilde{x}_{i,k}$. The difference between the sensor reading $r_{i,k}$ and the SVR modelled value $\widetilde{x}_{i,k}$, $y^{(2)}_{i,k}$, which we refer to as the drift measurement of node i at time instant k, is used by an *Unscented Kalman Filter* together with $r_{i,k}$ to estimate the corrected reading $x_{i,k}$ and the drift $d_{i,k}$ as will be shown in the following section.

5. Iterative measurement estimation and correction using an SVR-UKF framework

The solution to the smooth drift problem consists of the following iterative steps. At stage k, a reading $r_{i,k}$ is made by node i. The node also has a prediction for its corrected measurement (actual temperature at this sensor), $\widetilde{x}_{i,k} = f(\{x_{j,k-1}\}^n_{j=1,j\neq i})$, as a function of the corrected measurements of all neighbouring sensors in the cluster from the previous time step. Using this predicted value ($\widetilde{x}_{i,k}$) together with $r_{i,k}$, the corrected reading $x_{i,k}$ and the drift value $d_{i,k}$ are estimated. The node then sends the corrected sensor value $x_{i,k}$ to its neighbours. After that, each node collects the neighbourhood corrected measurements and computes $\widetilde{x}_{i,k}$ and so on. It is important here to emphasise that our main objective is to estimate $x_{i,k}$ the corrected reading which represents our estimate for the ground truth value $T_{i,k}$ at node i. Assuming that $x_{i,k}$ and $d_{i,k}$ change slowly with time the dynamics of $x_{i,k}$ and $d_{i,k}$ are mathematically described by:

$$\begin{aligned} x_{i,k} &= x_{i,k-1} + \eta^{(1)}_{i,k} \quad \eta^{(1)}_{i,k} \sim N(0, Q^{(1)}_{i,k}) & (1) \\ d_{i,k} &= d_{i,k-1} + \eta^{(2)}_{i,k} \quad \eta^{(2)}_{i,k} \sim N(0, Q^{(2)}_{i,k}) & (2) \end{aligned}$$

where $\eta^{(1)}_{i,k}$ and $\eta^{(2)}_{i,k}$ are the process noises. They are taken to be uncorrelated Gaussian noises with zero means and variances $Q^{(1)}_{i,k}$ and $Q^{(2)}_{i,k}$, respectively.

The value $x_{i,k}$ is never sensed or measured. What is really measured is $r_{i,k}$, the reading of the sensor. As we argued earlier, $r_{i,k}$ deviates from $x_{i,k}$ by both systematic and random errors. The random error is taken to be a Gaussian noise $w_{i,k} \sim N(0, R_{i,k})$ with zero mean and variance $R_{i,k}$ (measurement noise variance). The systematic error is referred to as the drift $d_{i,k}$. This leads to (3).

$$y^{(1)}_{i,k} = r_{i,k} = x_{i,k} + d_{i,k} + w_{i,k} \quad w_{i,k} \sim N(0, R_{i,k}) \tag{3}$$

We also define $y^{(2)}_{i,k}$ as the difference between the measurement $r_{i,k}$ and the SVR modelled value $\widetilde{x}_{i,k}$ and refer to $y^{(2)}_{i,k}$ as the drift measurement of node i at time instant k.

$$\begin{aligned} y^{(2)}_{i,k} &= y^{(1)}_{i,k} - f(\{x_{j,k-1}\}^n_{j=1,j\neq i}) \\ &= x_{i,k} + d_{i,k} + w_{i,k} - f(\{x_{j,k-1}\}^n_{j=1,j\neq i}) \\ &= x_{i,k} + d_{i,k} + w_{i,k} - \widetilde{x}_{i,k} \quad w_{i,k} \sim N(0, R_{i,k}) & (4) \end{aligned}$$

The model is expressed in vector notation as follows:

$$X_{i,k} = \begin{bmatrix} x_{i,k} \\ d_{i,k} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{i,k-1} \\ d_{i,k-1} \end{bmatrix} + \begin{bmatrix} \eta^{(1)}_{i,k} \\ \eta^{(2)}_{i,k} \end{bmatrix} \tag{5}$$

$$Y_{i,k} = \begin{bmatrix} y^{(1)}_{i,k} \\ y^{(2)}_{i,k} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_{i,k} \\ d_{i,k} \end{bmatrix} + \begin{bmatrix} w_{i,k} \\ w_{i,k} \end{bmatrix} - \begin{bmatrix} 0 \\ \widetilde{x}_{i,k} \end{bmatrix} \tag{6}$$

The noise component associated with $X_{i,k}$ is Gaussian with mean vector $\mu_{X_{i,k}} = [0 \quad 0]^T$ and covariance matrix $Qx_{i,k} = \begin{bmatrix} Q^{(1)}_{i,k} & 0 \\ 0 & Q^{(2)}_{i,k} \end{bmatrix}$. The noise component associated with $Y_{i,k}$ has a mean vector $\mu_{Y_{i,k}} = [0 \quad 0]^T$ and covariance matrix $Ry_{i,k} = \begin{bmatrix} R_{i,k} & R_{i,k} \\ R_{i,k} & R_{i,k} \end{bmatrix}$ which indicates that it is not White Gaussian. The system is clearly observable when $\widetilde{x}_{i,k} = x_{i,k}$, i.e. when $\widetilde{x}_{i,k}$ is a true, bias free, representation of $x_{i,k}$ and the difference between $x_{i,k}$ and $\widetilde{x}_{i,k}$ is zero.

Since the noise component associated with $Y_{i,k}$ is not White Gaussian, the KF cannot be used (Lu et al., 2007) to estimate $x_{i,k}$ and $d_{i,k}$. Another filter that can be used for solving such a problem is the Particle Filter. Unfortunately, the high computational complexity of the Particle Filter makes it unsuitable for the use in WSNs, where the sensors are limited in their energy and computational capabilities. A better alternative is to use the UKF. The Unscented Transformation (UT) was introduced by Julier et al. in (Julier et al., 1995) as an approximation method for propagating the mean and covariance of a random variable through a nonlinear transformation. This method was used to derive UKF in (Julier & Uhlmann, 1997). UKF can deal with versatile and complicated nonlinear sensor models and non-Gaussian noise that are not necessarily additive (Challa et al., 2008) with a comparable computational complexity to the Extended Kalman Filter (EKF) (Wan & van der Merwe, 2000). It also outperforms the EKF since it provides better estimation for the posterior mean and covariance to the third order Taylor series expansion when the input is Gaussian, whereas, the EKF, only achieves the first order Taylor series expansion (Wan & van der Merwe, 2000). Below, we explain the UKF algorithm in detail.

The UT as mentioned before is a method for finding the statistics of a random variable $Z = g(X)$ which undergoes nonlinear transformation. Let X of dimension L be the random variable that is propagated through the nonlinear function $Z = g(X)$. Assume that X has a mean $\hat{X}$ and a covariance P. According to (Challa et al., 2008), to find the statistics of Z using the scaled unscented transformation, which was introduced in (Julier, 2002), the following steps must be followed: First, $2L + 1$ (where L is the dimension of vector X) weighted samples or *sigma points* $\sigma_i = \{\mathcal{W}_i, \mathcal{X}_i\}$ are deterministically chosen to completely capture the true mean and covariance of the random variable X. Then, the sigma points are propagated through the function $g(X)$ to capture the statistics (mean and covariance) of Z. A selection scheme that satisfies the requirement is given below:

$$\begin{aligned}\mathcal{X}_0 &= \hat{X}, \quad \mathcal{W}_0^m = \frac{\lambda}{\lambda+L}\\ \mathcal{W}_0^c &= \frac{\lambda}{\lambda+L} + (1-\alpha^2+\beta)\\ \mathcal{X}_i &= \hat{X} + (\sqrt{(L+\lambda)P})_i, \quad \mathcal{W}_i = \frac{1}{2(\lambda+L)}\\ \mathcal{X}_{L+i} &= \hat{X} - (\sqrt{(L+\lambda)P})_i, \quad \mathcal{W}_{L+i} = \frac{1}{2(\lambda+L)} \end{aligned} \tag{7}$$

where $i = 1, ..., L$ and $\lambda = \alpha^2(L+\kappa) - L$ is a scaling parameter. α determines the spread of the sigma points around the mean $\hat{X}$ and is usually set to a small positive value (*e.g.*, 0.001). κ is a secondary scaling parameter which is usually set to 0, and β is used to incorporate prior knowledge of the distribution of X. The optimal value of β for a Gaussian distribution is $\beta = 2$ as stated in (Wan & van der Merwe, 2000). The term $(\sqrt{(L+\lambda)P})_i$ is the ith row of the matrix square root of matrix $(L+\lambda)P$. In our work here α, κ and β are taken to be equal to 0.001, 0, 2, respectively. The UKF is used to estimate $X_{i,k}$ for sensor i at time step k. The dimension L of $X_{i,k}$ is equal to 2. This means that we only have five sigma points for each node i. The steps of the UKF algorithm are given below as in (Challa et al., 2008):

Let $\hat{X}_{i,k-1|k-1}$ be the prior mean of the state variable and $P_{i,k-1|k-1}$ be the associated covariance for node i. To simplify the notation we write the prior mean of the state variable and the associated covariance as $\hat{X}_{k-1|k-1}$ and $P_{k-1|k-1}$ (without showing the sensor number i) keeping in mind that they refer to a certain sensor node i. This also applies for all the other parameters we use in describing the UKF algorithm.

The sigma points are calculated from (7) and then propagated through the state equation function $g(.)$. This results in $\mathcal{X}_{0,k|k-1}, \mathcal{X}_{1,k|k-1}, \mathcal{X}_{2,k|k-1}, \mathcal{X}_{3,k|k-1}$ and $\mathcal{X}_{4,k|k-1}$ as shown in (8).

$$\mathcal{X}_{k|k-1} = g(\mathcal{X}_{k-1}) = \mathcal{X}_{k-1} \tag{8}$$

The predicted mean and covariance of the state variable are given by (9) and (10), respectively.

$$\hat{X}_{k|k-1} = \mathcal{W}_0^m \mathcal{X}_{0,k|k-1} + \sum_{i=1}^{2L} \mathcal{W}_i \mathcal{X}_{i,k|k-1} \tag{9}$$

$$\begin{aligned} P_{k|k-1} &= \mathcal{W}_0^c(\mathcal{X}_{0,k|k-1} - \hat{X}_{k|k-1})(\mathcal{X}_{0,k|k-1} - \hat{X}_{k|k-1})^T\\ &+ \sum_{i=1}^{2L} \mathcal{W}_i(\mathcal{X}_{i,k|k-1} - \hat{X}_{k|k-1})(\mathcal{X}_{i,k|k-1} - \hat{X}_{k|k-1})^T + Qx_k \end{aligned} \tag{10}$$

The propagated sigma points are then passed through the measurement function $h(.)$ as shown in (11).

$$\mathcal{Y}_{k|k-1} = h(\mathcal{X}_{k|k-1}) \tag{11}$$

Then the predicted mean and covariance of each sensor measurement are given by (12) and (13), respectively.

$$\hat{Y}_{k|k-1} = \mathcal{W}_0^m \mathcal{Y}_{0,k|k-1} + \sum_{i=1}^{2L} \mathcal{W}_i \mathcal{Y}_{i,k|k-1} \tag{12}$$

$$
\begin{aligned}
P_{Y_kY_k} &= \mathcal{W}_0^c(\mathcal{Y}_{0,k|k-1} - \hat{Y}_{k|k-1})(\mathcal{Y}_{0,k|k-1} - \hat{Y}_{k|k-1})^T \\
&+ \sum_{i=1}^{2L} \mathcal{W}_i(\mathcal{Y}_{i,k|k-1} - \hat{Y}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{Y}_{k|k-1})^T + Ry_k
\end{aligned}
\tag{13}
$$

The cross covariance of the predicted state and sensor measurement is found by (14).

$$
\begin{aligned}
P_{X_kY_k} &= \mathcal{W}_0^c(\mathcal{X}_{0,k|k-1} - \hat{X}_{k|k-1})(\mathcal{Y}_{0,k|k-1} - \hat{Y}_{k|k-1})^T \\
&+ \sum_{i=1}^{2L}[\mathcal{W}_i(\mathcal{X}_{i,k|k-1} - \hat{X}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{Y}_{k|k-1})^T]
\end{aligned}
\tag{14}
$$

where

$$K_k = P_{X_kY_k}P_{Y_kY_k}^{-1} \tag{15}$$

The updated posterior mean and covariance of the state are then estimated by (16) and (17), respectively.

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k(Y_k - \hat{Y}_{k|k-1}) \tag{16}$$

$$P_{k|k} = P_{k|k-1} + K_kP_{Y_kY_k}K_k^T \tag{17}$$

where $\hat{X}_{k|k}$ and $P_{k|k}$ are the mean and covariance of the state of node i at time step k.

Figure 3 shows a block diagram of our drift correction algorithm. It clearly summarises the stages of the error detection and correction framework in one of the nodes in the cluster. The steps of the algorithm are stated below:

Decentralised error correction algorithm using the SVR-UKF framework

At time step k

- Each node i finds its predicted corrected measurement $\widetilde{x}_{i,k}=f(\{x_{j,k-1}\}_{j=1,j\neq i}^{n})$.
- Each node i obtains its reading $y_{i,k}^{(1)} = r_{i,k}$.
- Each node i calculates the drift measurement $y_{i,k}^{(2)}$.
- Each node i finds the sigma points $\sigma_i = \{\mathcal{W}_i, \mathcal{X}_i\}$ from $\hat{X}_{i,k-1|k-1} = [\hat{x}_{i,k-1|k-1}$ $\hat{d}_{i,k-1|k-1}]^T$.
- For each node i, the sigma points are propagated through the state equation function $g(.)$.
- The UKF estimates the corrected measurement and the drift using (9)-(17) and then sends the result to the neighbouring nodes.
- The algorithm reiterates.

6. Evaluation

Our aim is to evaluate the ability of our proposed framework to correct the drift experienced in sensor nodes and to extend the functional life of the sensor network. The data in our evaluation are a set of real sensor measurements gathered from a deployment of wireless sensors in the IBRL (2006, Accessed on 07/09/2006).

In 2004, a set of wireless sensors with 55 sensor nodes (including a gateway node) were deployed in the IBRL lab for monitoring the lab environment (refer to Figure 4). They recorded

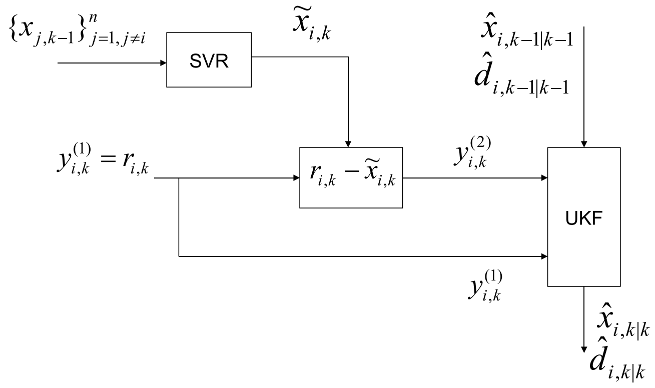

Fig. 3. The SVR-UKF Measurement correction framework at node *i*.

temperature, humidity, light and voltage measurements at 30 seconds intervals during the period starting from 28th February 2004 to 5th April 2004.

The data from the sensor nodes are re-sampled at seven minute intervals and the first 2000 samples are used for our evaluation purposes. This corresponds to the data collected during a ten day period from 28th February 2004 to 9th March 2004. We use the first 1000 samples (this corresponds to the first five days' data) as the training set for use in the *training phase.* An exponential drift is introduced to the real data in each node, starting randomly after the first 1000 samples. The data after 1000 samples and up to 2000 samples are used in the *running phase* for testing our algorithm for drift correction. These samples correspond to the next five days of the IBRL data. Temperature measurements are used in all our evaluations.

We formed a network of sensors using nodes selected from the IBRL deployment using sixteen sensor nodes. The node IDs used are {1,2,3,4,6,7,8,9,10,31,31,33,34,35,36,37}. Each sensor communicates only with its closest 8 neighbours.

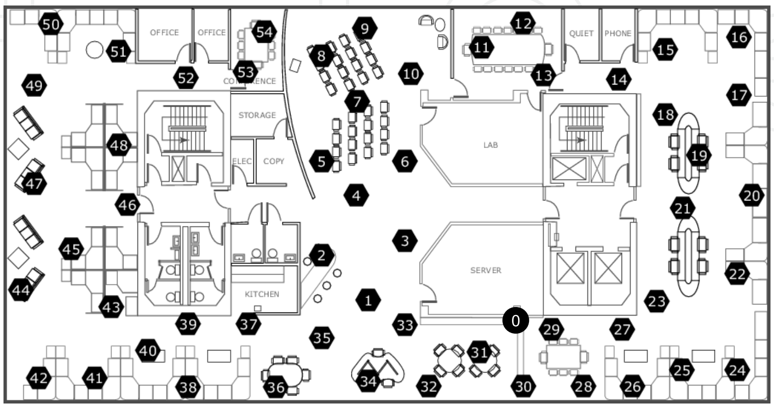

Fig. 4. Sensor nodes in the IBRL deployment. Nodes are shown in black with their corresponding node-IDs. Node 0 is the gateway node (2006, Accessed on 07/09/2006).

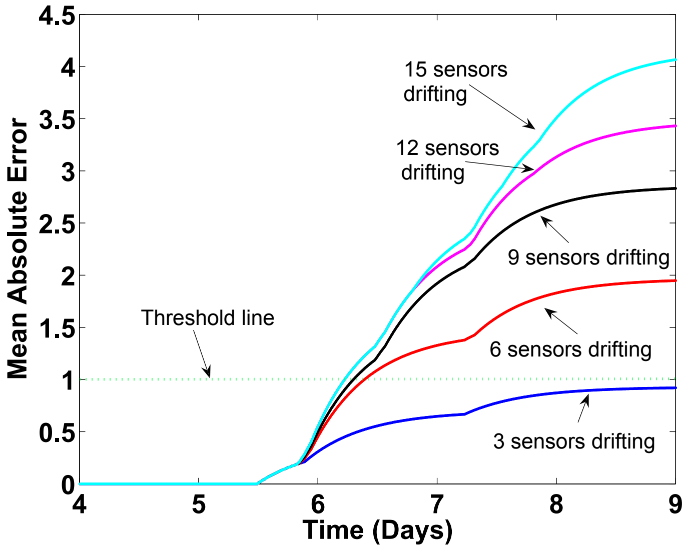

Fig. 5. Mean Absolute Error for the network without correction.

Our algorithm is implemented in MatLab, utilising the SVR toolbox from (Canu et al., 2005) and the UKF toolbox from (Särkkä & Hartikainen, 2007). For comparison purposes, we run the algorithm on two data sets. One, the data *without* the introduced drift (WOD), and the other, the data *with* drift introduced (WD). Initially, the SVR of each node is trained on the first 1000 samples of its neighbours readings.

The UKF parameters μ, κ and β are set to the default values as explained in Section 5. Through out our evaluations, we take $Q_{i,k}^{(1)} = Q_{i,k}^{(2)} = Q_{i,k}$. $Q_{i,k}$ and $R_{i,k}$ are tuned using trial and error for both cases. The values used in our evaluation are $Q_{i,k} = 0.001$ and $R_{i,k} = 0.02$. If $R_{i,k}$ is set to a high value, the estimated temperature will follow the reading (which may have drift) whereas if $R_{i,k}$ is set to a small value, the estimated temperature will not be able to follow the real temperature. Thus, it will not totally correct the error. On the other hand, a high value for $Q_{i,k}$ will result in oscillatory estimates and lead to an unstable state. Hence, a trade off has to be considered in selecting the values for $Q_{i,k}$ and $R_{i,k}$ to obtain the best results.

We have conducted two simulations using two data sets. One data set has no drifts introduced. We denote this data set by *R-WOD*, which stands for *'Readings Without Drift'* and represents the sensor measurements that only suffer from noise. The other is the same data set with drifts introduced in several scenarios. We denote the readings of this data set by *R-WD*, which stands for *'Readings With Drift'* and represents the sensor measurements that suffer from both drift and noise. The drift scenarios considered in *R-WD* are as follows: scenario 1 (SCN 1) being one node drifting, scenario 2 (SCN 2) being two nodes drifting and so on until the last scenario (SCN 16) having all nodes drifting. The resulting corrected measurements obtained when the algorithm is run on the *R-WD* data sets are denoted by *DCM-WD*, which stands for *'Drift Corrected Measurement for readings With Drift'*. Similarly, the corrected measurements obtained using data set *R-WOD* are denoted by *DCM-WOD*, which stands for *'Drift Corrected Measurement for readings WithOut Drift'*.

To evaluate the performance of our algorithm from the network's point of view, we compare the average absolute error of all the sensors of the network with and without implementing our drift correction algorithm.

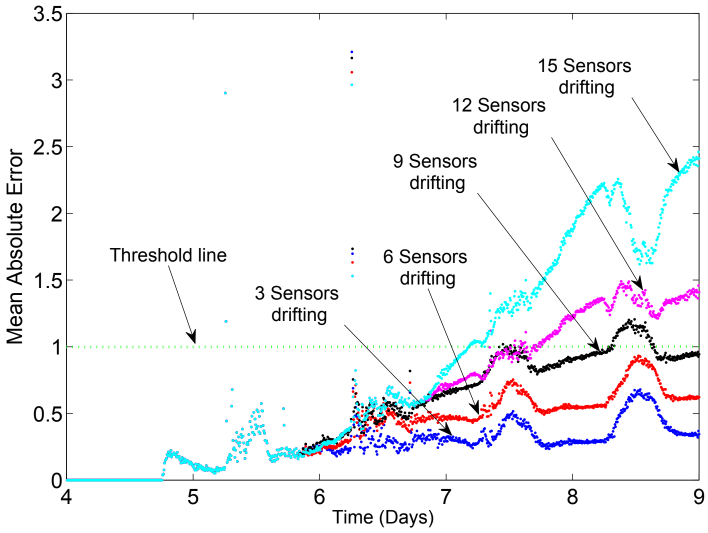

Fig. 6. Mean Absolute Error for the network with correction for 2001 samples in 10 days.

Figure 5 shows the mean absolute error between the true temperatures (R-WOD) and the values reported by the sensors (R-WD) for the whole network, for five different scenarios. The mean absolute error of the network is computed for each scenario as follows: for each node, at each instant of time, the absolute error between the true temperature (R-WOD) and the value reported by the sensors (R-WD) is computed. The average for all these nodes' absolute errors is then found. This gives the mean absolute error of the network. Similarly, the mean absolute error between the true temperatures (R-WOD) and the drift corrected measurements (DCM-WD) is calculated at each instant of time and plotted in Figure 6. By comparing Figures 5 and 6 it is evident that applying the drift correction algorithm results in less measurements error for all of the scenarios. For our evaluation purposes we assumed that the maximum mean absolute error that can be tolerated in the network is 1^oC. If the mean absolute error of the network exceeds that limit, the network is deemed to be useless or has broken down. This maximum limit is shown by a horizontal threshold line in Figures 5 and 6. The choice of the threshold is dependent on the error tolerance allowed by the application.

In Figure 5, it is evident that the curves for scenarios 6, 9, 12 and 15 cross the threshold line after the 6th day of the experiment. In contrast, in Figure 6, the curves for scenarios 6 and 9 do not cross the threshold line at all for the whole period of the experiment, while the curves of scenarios 12 and 15 cross the threshold line on the 8th day and the 7th day, respectively. This demonstrates that our algorithm extends the operational life of the network for all of the scenarios.

In another simulation we repeated the experiment after doubling the sampling rate. This resulted in 4001 samples for the 10 days experiment. Figure 7 shows the mean absolute error between the true temperatures (R-WOD) and the drift corrected measurements (DCM-WD) for the whole network for five different scenarios. The error is computed in a similar way to the method used for the Figures 5 and 6. By comparing Figures 5 and 7, it is evident that the application of the error correction algorithm results in less measurements errors for all of the scenarios.

Looking at Figures 6 and 7, we can notice that the performance when using 4001 samples is better for scenario 9 since the absolute error curve does not cross the 1^oC threshold line as it does in the 2001 samples case. This means that the operational lifetime has been extended from around 8 days in the case of 2001 samples, to more than 9 days for the case of 4001 samples. Moreover, we notice in Figure 7 that the curves for each scenario are smoother than the corresponding curves in Figure 6 and that the observed occasional jumps and peaks are smaller. The jumps in the curves are caused by the fast changes in the readings or the ambient temperature at some instants of time. An effective way of reducing the size of the jumps is to increase the sampling rate as we noticed in Figure 7. However, that would be at a cost of the increased communication overhead due to the increased data transmissions among the sensors. This means that a trade off between the smoothness of the curves and the communication overhead has to be made. Another important thing to note in both Figures 6 and 7 is that the mean absolute error of the network's estimated temperatures is proportional to the number of sensors developing drift.

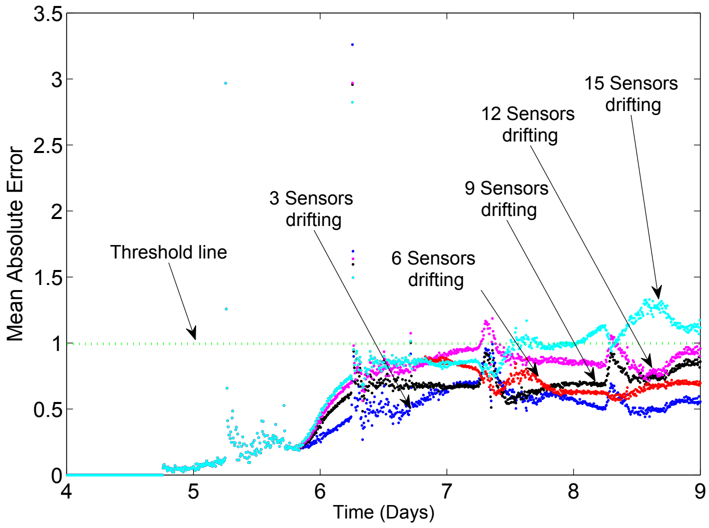

Fig. 7. Mean Absolute Error for the network with correction for 4001 samples in 10 days.

The choice of $R_{i,k}$ and $Q_{i,k}$ is crucial. It affects the accuracy of estimating the temperature and the induced error. In general, we can say that increasing $R_{i,k}$ improves the tracking of drift in the drifting sensors. However, it also increases both the induced error in drift estimation in the non drifting sensors and the fluctuations in the drifting sensors. Since, the error caused by the fast changes in temperature in the case of 4001 samples is less than that for the case of 2001 samples (as explained previously), $R_{i,k}$ is taken to be 0.05 for the case of 4001 and 0.02 for the 2001 samples case. This way, the drift tracking is improved for the 4001 samples case keeping error levels comparable to the case of 2001 samples. On the other hand, increasing $Q_{i,k}$ increases the fluctuations in the estimated drift in both drifting and non-drifting sensors and causes the response to become less stable. The $Q_{i,k}$ used in all our simulations is equal to 0.001.

It is important here to note that for comparison purposes, the data set used in the evaluations of this chapter is the same as the data set used in (Takruri, Rajasegarar, Challa, Leckie & Palaniswami, 2010) evaluations. Comparing figure 8, which is quoted from (Takruri, Rajasegarar, Challa, Leckie & Palaniswami, 2010), with figure 6, where both of them are for 2001 samples, it can be clearly noticed that SVR-UKF framework presented in (Takruri, Rajasegarar, Challa, Leckie & Palaniswami, 2010) outperforms the algorithm presented in this chapter in correcting sensor readings errors as it manages to keep the absolute error for the case of 9 sensors drifting below the threshold line. However, this is at the cost of substantially increased computational complexity. The state transition function used in this chapter is linear and the number of sigma points is 5 whereas the state transition function used in the algorithm in (Takruri, Rajasegarar, Challa, Leckie & Palaniswami, 2010) is the SVR modelled function which is highly nonliner and the number of sigma points is 19. This results in a computational complexity of several orders of magnitude.

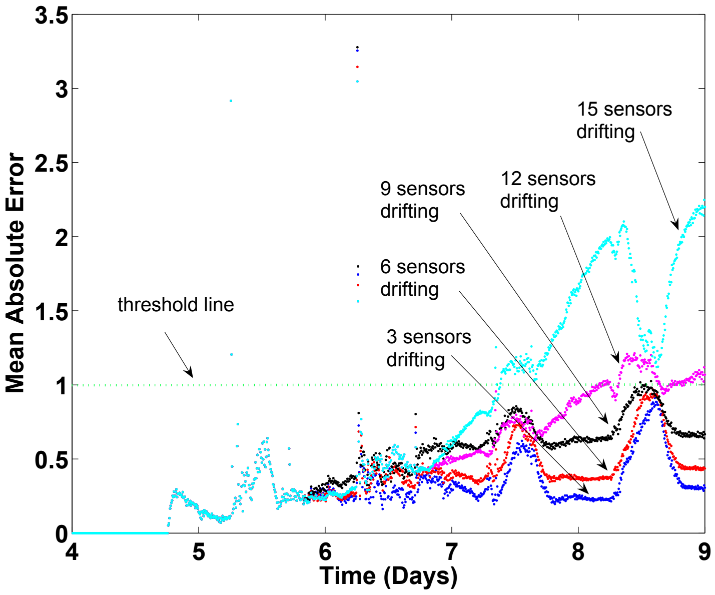

Fig. 8. Mean Absolute Error for the network with correction for 2001 samples in 10 days. Quoted from (Takruri, Rajasegarar, Challa, Leckie & Palaniswami, 2010).

The error correction performance for a sensor node in a cluster is dependent on the correlation of the actual temperature at the sensor under consideration with the actual temperatures at the neighbours. The SVR at a sensor predicts the actual temperature at the sensor $\tilde{x}_{i,k}$ using previous estimates of the neighbourhood $\{\hat{x}_{j,k-1|k-1}\}_{j=1,j\neq i}^{n}$. Therefore, low correlation will lead to a poor prediction, and result in poor estimate of the actual temperature at the sensor under consideration. In practice, the correlation among the nodes may change depending on their spatial proximity within the cluster and with the change in the observed phenomenon along time.

It can be observed in the IBRL sensor deployment that not all the sensors were subject to the same conditions. This is because of their physical locations. Some of the nodes were closer to air conditioning. Some were closer to windows and hence were affected by the sun. Some were closer to the kitchen and thus affected by the heat and humidity coming from there. Furthermore, the patterns followed by sensor measurements changed seasonally. As an example, during a week period, the pattern followed in week days was different than that followed in weekends. That was because the air conditioning was reduced or turned off in the laboratory on the weekends. This caused the interrelationship among the sensors to vary not only with their spatial locations, but also with time.

A solution to overcome such a problem is to choose the neighbour sensors of each node so that they are physically close and subject to similar conditions. An alternative solution is to upgrade the model to become incremental with time to account for phenomenal changes. This can be achieved using *incremental learning* of the SVR. The learning process can then be performed at each time step (incrementally) or at predefined short intervals, depending on how severe the change is. Incremental SVR learning algorithms (Platt, April 1998; Shilton et al., Jan. 2005) can be utilised with the UKF to perform adaptive drift correction in the network. Devising an adaptive drift correction framework by incorporating incremental learning is a direction for future work.

7. Conclusion

In this chapter we have proposed a formal statistical algorithm for detecting and correcting sensor measurement errors in sparsely deployed WSN based on the assumption that sensor nodes in a neighbourhood observing a certain physical phenomenon have correlated measurements and uncorrelated drifts and biases. We have used SVR to model the spatio-temporal correlations in the neighbouring sensor measurements to obtain predictions of the future sensor measurements. The predicted data have then been used by a UKF to estimate the actual value of the measured variable at the sensor under consideration. The algorithm runs recursively and is fully decentralised. Extensive evaluations of the presented algorithm on real data obtained from the IBRL have proved that it is effective in detecting and correcting sensor errors and extending the effective life of the network.

In future, we plan to upgrade our algorithm to become more adaptive to any phenomenal changes that may occur in the network deployment area by implementing an incremental SVR to periodically re-train the SVR. This will also be tested on sensor networks deployed both in a controlled Lab environment and uncontrolled outdoor environments.

8. References

2006 (Accessed on 07/09/2006). http://db.lcs.mit.edu/labdata/labdata.html, *[online]*.
URL: *http://db.lcs.mit.edu/labdata/labdata.html*

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. & Cayirci, E. (2002). Wireless sensor networks: a survey, *Comp. Networks* **38**: 393–422.

Balzano, L. (2007). *Addressing fault and calibration in wireless sensor networks*, Master's thesis, University of California, Los Angeles, California.

Balzano, L. & Nowak, R. (2007). Blind calibration of sensor networks, *Information Processing in Sensor Networks* .

Balzano, L. & Nowak, R. (2008). Blind calibration of networks of sensors: Theory and algorithms, *Networked Sensing Information and Control*, Springer US, pp. 9–37.

Brown, L. (1992). A survey of image registration techniques, *ACM Computing Surveys* **24**(4): 326–376.

Bychkovskiy, V. (2003). *Distributed in-place calibration in sensor networks*, Master's thesis, University of California, Los Angeles, California.

Bychkovskiy, V., Megerian, S., Estrin, D. & Potkonjak, M. (2003). A collaborative approach to in-place sensor calibration, *Int. Workshop on Information Processing in Sensor Networks* pp. 301–316.

Canu, S., Grandvalet, Y., Guigue, V. & Rakotomamonjy, A. (2005). Svm and kernel methods matlab toolbox, Perception Systï£¡mes et Information, INSA de Rouen, Rouen, France.
URL: *http://asi.insa-rouen.fr/enseignants/ arakotom/toolbox/index.html*

Challa, S., Evans, R., Morelande, M. & Musicki, D. (2008). *Fundamentals of Object Tracking*, Cambridge University Press.

Elnahrawy, E. & Nath, B. (2003). Cleaning and querying noisy sensors, *in Proceedings of ACM WSNAï£¡03*.

Estrin, D., Girod, L., Pottie, G. & Srivastava, M. (2001). Instrumenting the world with wireless sensor networks, *Int. Conference on Acoustics, Speech, and Signal Processing)* .

Feng, J., Megerian, S. & Potkonjak, M. (2003). Model-based calibration for sensor networks, *Sensors* pp. 737 – 742.

Gill, M. K., Asefa, T., Kemblowski, M. W. & McKee, M. (2006). Soil moisture prediction using support vector machines1, *J. of the American Water Resources Association* **42**(4): 1033–1046.
URL: *http://www.blackwell-synergy.com/doi/abs/10.1111/j.1752-1688.2006.tb04512.x*

Gill, M. K., Kemblowski, M. W. & McKee, M. (2007). Soil moisture data assimilation using support vector machines and ensemble kalman filter, *J. of the American Water Resources Association* **43**(4): 1004–1015.

Guestrin, C., Bodik, P., Thibaux, R., Paskin, M. & Madden, S. (2004). Distributed regression: an efficient framework for modeling sensor network data, *In IPSN04*, pp. 1–10.

Hoadley, B. (1970). A baysian look at inverse linear regression, *J. of the American Stats. Association* **65**(329): 356–369.

Ihler, A., Fisher, J., Moses, R. & Willsky, A. (2004). Nonparametric belief propagation for self-calibration in sensor networks, *In Proceedings of the Third international Symposium on Information Processing in Sensor Networks*.

Julier, S. (2002). The scaled unscented transformation, *American Control Conference* **6**: 4555–4559.

Julier, S. J., Uhlmann, J. K. & Durrant-Whyte, H. F. (1995). A new approach for filtering nonlinear systems, *American control Conference* pp. 1628 – 1632.

Julier, S. & Uhlmann, J. (1997). A new extension of the Kalman filter to nonlinear systems, *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*.

Krishnamachari, B. & Iyengar, S. (2004). Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks, *IEEE Tran. Computers* **53**(3): 241–250.

Lu, S., Cai, L., Lu, D. & Chen, J. (2007). Two efficient implementation forms of unscented kalman filter, *IEEE Int. Conference on Control and Automation* pp. 761 – 764.

Nowak, R. & Mitra, U. (2003). Boundary estimation in sensor networks: Theory and methods, *In IPSN*, pp. 80–95.

Okello, N. & Challa, S. (2003). Simultaneous registration and track fusion for networked trackers, *conference on information fusion*, Montrï£¡al, Canada.

Okello, N. & Pulford, G. (1996). Simultaneous registration and tracking for multiple radars with cluttered measurements, *IEEE Signal Processing Workshop on Statistical Signal and Array Processing* pp. 60–63.

Platt, J. (April 1998). Sequential minimal optimization: A fast algorithm for training support vector machines, *Technical Report 98-14, Microsoft Research, Redmond, Washington.* .
URL: *citeseer.ist.psu.edu/platt98sequential.html*

Rajasegarar, S., Leckie, C., Palaniswami, M. & Bezdek, J. (2007). Quarter sphere based distributed anomaly detection in wireless sensor networks, *Proceedings of the IEEE International Conference Communications (IEEE ICC '07)*, UK.

Sallans, B., Bruckner, D. & Russ, G. (2005). Statistical model-based sensor diagnostic for automation systems, *in* M. L. Chavez (ed.), *Fieldbus systems and their applications*, Elsevier, pp. 239–246.

Scholkopf, B. & Smola, A. (2002). *Learning with Kernels*, MIT Press.

Shilton, A., Palaniswami, M., Ralph, D. & Tsoi, A. C. (Jan. 2005). Incremental training of support vector machines, *IEEE Tran. on Neural Networks* **16**(1): 114–131.

Särkkä, S. & Hartikainen, J. (2007). Ekf/ukf toolbox for matlab v1.2, Centre of Excellence in Computational Complex Systems Research, Helsinki University of Technology (HUT), Finland. http://www.lce.hut.fi/research/mm/ekfukf/.

Takruri, M., Aboura, K. & Challa, S. (2008). Distributed recursive algorithm for auto calibration in drift aware wireless sensor networks, *in* K. Elleithy (ed.), *Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering*, Springer, pp. 21–25.

Takruri, M. & Challa, S. (2007). Drift aware wireless sensor networks, *Proceedings of the 10th international conference on information fusion*, Quebec City, Canada.

Takruri, M., Challa, S. & Chacravorty, R. (2010). Recursive bayesian approaches for auto calibration in drift aware wireless sensor networks, *Journal of Networks* **5**(7): 823–832.

Takruri, M., Challa, S. & Chakravorty, R. (2008). Auto calibration in drift aware wireless sensor networks using the interacting multiple model algorithm, *Mosharaka International Conference on Communications, Computers and Applications (MIC-CCA 2008)*, Amman, Jordan.

Takruri, M., Challa, S. & Yunis, R. (2009). Data fusion techniques for auto calibration in wireless sensor networks, *Proceedings of the 12th International conference on information fusion*, Seattle, USA.

Takruri, M., Rajasegarar, S., Challa, S., Leckie, C. & Palaniswami, M. (2008). Online drift correction in wireless sensor networks using spatio-temporal modeling, *Proceedings of the 11th international conference on information fusion*, Cologne, Germany.

Takruri, M., Rajasegarar, S., Challa, S., Leckie, C. & Palaniswami, M. (2010). Spatio-temporal modelling based drift aware wireless sensor networks, *IET Wirel. Sens. Syst. (under review)* .

Taylor, C., Rahimi, A., Bachrach, J., Shrobe, H. & Grue, A. (2006). Simultaneous localization, calibration, and tracking in an ad hoc sensor network, *In Proceedings of 5th International Conference on Information Processing in Sensor Networks (IPSN06)*, pp. 27–33.

Wan, E. & van der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation, *IEEE Symposium 2000 (AS-SPCC)* .

Wang, Y. M., Schultz, R. T., Constable, R. T. & Staib1, L. H. (2003). Nonlinear estimation and modeling of fmri data using spatio-temporal support vector regression, *Information Processing in Medical Imaging* **2732**: 647–659.

Whitehouse, K. & Culler, D. (2002). Calibration as parameter estimation in sensor networks, *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*.

Whitehouse, K. & Culler, D. (2003). Macro-calibration in sensor/actuator networks, *Mobile Networks and Applications Journal (MONET), Special Issue on Wireless Sensor Networks* .

Xu, X., Hines, J. W. & Uhrig, R. E. (1998). On-line sensor calibration monitoring and fault detection for chemical processes, *Maintenance and Reliability Conference (MARCON 98)*, pp. 12–14.

Zitova, B. & Flusser, J. (2003). Image registration methods: a survey, *Image and Vision Computing* **21**(1): 977–1000.

Wireless Sensor Networks: Application-Centric Design
Edited by Yen Kheng Tan

ISBN 978-953-307-321-7
Hard cover, 492 pages
Publisher InTech
Published online 14, December, 2010
Published in print edition December, 2010

Over the past decade, there has been a prolific increase in the research, development and commercialisation of Wireless Sensor Networks (WSNs) and their associated technologies. WSNs have found application in a vast range of different domains, scenarios and disciplines. These have included healthcare, defence and security, environmental monitoring and building/structural health monitoring. However, as a result of the broad array of pertinent applications, WSN researchers have also realised the application specificity of the domain; it is incredibly difficult, if not impossible, to find an application-independent solution to most WSN problems. Hence, research into WSNs dictates the adoption of an application-centric design process. This book is not intended to be a comprehensive review of all WSN applications and deployments to date. Instead, it is a collection of state-of-the-art research papers discussing current applications and deployment experiences, but also the communication and data processing technologies that are fundamental in further developing solutions to applications. Whilst a common foundation is retained through all chapters, this book contains a broad array of often differing interpretations, configurations and limitations of WSNs, and this highlights the diversity of this ever-changing research area. The chapters have been categorised into three distinct sections: applications and case studies, communication and networking, and information and data processing. The readership of this book is intended to be postgraduate/postdoctoral researchers and professional engineers, though some of the chapters may be of relevance to interested masterâ€™s level students.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820

www.intechopen.com

Fax: +385 (51) 686 166
www.intechopen.com

Fax: +86-21-62489821