TIERED ARCHITECTURE FOR ON-LINE DETECTION, ISOLATION AND REPAIR OF FAULTS IN WIRELESS SENSOR NETWORKS

Sadaf Zahedi*, Marcin Szczodrak+, Ping Ji+, Dinkar Mylaraswamy**,
Mani Srivastava* and Robert Young***
*University of California, Los Angeles, CA USA, {*szahedi, mbs*}*@ee.ucla.ed*u
+City University of New York, NY USA, *mszczodrak@gmail.com, pji@jjay.cuny.edu*
**Honeywell, MN USA, *dinkar.mylaraswamy@honeywell.com*
***MoD, United Kingdom, *riyoung@dstl.gov.uk*

Abstract*—Wireless sensor networks fuse data from a multiplicity of sensors of different modalities and spatiotemporal scales to provide information for reconnaissance, surveillance, and situational awareness in many defense applications. For decisions to be based on information returned by sensor networks it is crucial that such information be of sustained high quality. While the Quality of Information (QoI) depends on many factors, perhaps the most crucial is the integrity of the sensor data sources themselves. Even ignoring malicious subversion, sensor data quality may be compromised by non-malicious causes such as noise, drifts, calibration, and faults. On-line detection and isolation of such misbehaviors is crucial not only for assuring QoI delivered to the end-user, but also for efficient operation and management by avoiding wasted energy and bandwidth in carrying poor quality data and enabling timely repair of sensors. We describe a two-tiered system for on-line detection of sensor faults. A local tier running at resource-constrained nodes uses an embedded model of the physical world together with a hypothesis-testing detector to identify potential faults in sensor measurements and notifies a global tier. In turn, the global tier uses these notifications on the one hand during fusion for more robust estimation of physical world events of interest to the user, and on the other hand for consistency checking among notifications from various sensors and generating feedback to update the embedded physical world model at the local nodes. Our system eliminates the undesirable attributes of purely centralized and purely distributed approaches that respectively suffer from high resource consumption from sending all data to a sink, and high false alarms due to lack of global knowledge. We demonstrate the performance of our system on diverse real-life sensor faults by using a modeling framework that permits injection of sensor faults to study their impact on the application QoI.

I. INTRODUCTION

Recent sensor networks are potential of advancing our knowledge about interesting features of phenomenon by monitoring the physical world at unprecedented scales and resolutions. However, for a sensor network to be useful, the information it provides must be of high integrity. In typical applications, a sensor network detects sources, reconstructs physical fields, triggers other events, and communicates the results to end users to estimate *event of interest* (E), and make decisions based upon it. The decision making process can go awry if the sensor network provides a misleading picture of the physical phenomenon, some causes of which are communication and environmental noise, faults in the sensors, unfavorable channels from source to detectors (e.g. obstacles) and insufficient sampling in time or space. Although *Quality of Information* (QoI) in decision making depends on many factors, sensor faults such as high noise, drifts and calibration have perhaps the most crucial effect on the integrity of sensor data. A sensor may misbehave due to an inherent failure or due to an inappropriate environmental condition [1], [2].

Recent increasing attention to QoI-aware systems in applications such as health monitoring and enemy activity surveillance has been the strong motivation behind this work. We have proposed a real-time on-line fault detection system to maintain the high integrity.

Research on reliable computing and fault tolerant systems has established before emerging the wireless sensor networks. An early example is [3], which studied fault tolerance in measurements by a group of sensors in process control applications. In [3], an abstract virtual sensor is proposed that averages values from multiple fault-prone sensors in a fault-tolerant fashion. The analysis in [3] assumes that each sensor measures the same physical variable with a certain uncertainty and fault specification. However, studying faults in wireless sensing systems differs from faults in process control, which makes the problem more difficult. One of the most important issues is that sensor networks may involve many more sensors over larger areas and sensors don't measure equal variables. Also, for a sensor network the phenomenon being observed is often not well defined and modeled, resulting in higher uncertainty when modeling sensor behavior and sensor faults.

In [4], an outlier detection method is presented which is based on Bayesian learning. They have assumed that conditional on the observed measurement, neighbors infor-

978-1-4244-2677-5/08/ $ 25.00 (c)2008 IEEE

mation and past information are independent, and with this assumption, the procedure learns a distribution for interval ranges of the measurements.

Fault detection can be done either in off-line or on-line fashion. Due to increasing demand of recent sensor network applications to real-time data management, we have focused on on-line detection. On-line fault detection and diagnosis are usually based on data quality analysis of sensor measurements. These methods use statistical models of sensor data and physical phenomenon, and leverage correlations in space, time, and sensing modalities.

Most of the previous on-line detection mechanisms, such as [5], use a centralized approach where the sink detects faults based on statistical model built at the sink based on the collected sensor data. The centralized quality checking is certainly not an efficient approach for application scenarios that only need to report the occurrence of some events or the presence of some features to the end user. Having to extract all the sensor samples to the back-end for the purpose of fault detection will impose needless energy overheads and also cause communication scalability problems. On the other hand, distributed fault detection approaches have been proposed in other works such as [6], [7], [8]. Approach in [6] lacks the information from other nodes that can help disambiguate faults from unexpected phenomena. In [7], a distributed fault-tolerant detection scheme has proposed which is based on the assumption that time varying failure probabilities of each node are known, and detection is based on threshold testing. In a more recent work [8], authors proposed a distributed fault detection system which is based on the techniques in sequential analysis and sequential change point detection. This scheme assumes that the probability distributions of the data is computable which is not always the case in general sensor network applications.

To satisfy the quality and cost constraints and consider their tradeoffs, we have investigated a two-tiered system for on-line detection of sensor faults. In this paper we propose a two-tiered system for on-line fault detection, isolation and possibly repair (FDIR) of it in wireless sensor networks and study its positive impact on the network QoI .

II. Tiered Architecture Overview

The proposed framework is illustrated in figure 1. Sensor fault characterization is done at two tiers: a ”local tier” and a ”global tier”. The local fault model is generic and independent of the sensor modality. It is assumed that the sampling frequency at an individual sensor is fast, compare to the time constant of the “event of interest”. Local FDIR exploits high frequency raw data available from a sensor or a sensor array to classify the incoming signal into one of pre-enumerated fault states. The local model may also be able to repair a faulty signal. A local fault model at the sensor level augments sensor measurements $y_i(t), i = 1, 2, ..$ with a *fault vector* that characterizes sensor behavior at the local level. A fault vector is described as a 2-tuple: $\{$fault state, fault signature parameters$\}$. It may have an additional attribute, namely a reconciled value for the sensor signal. Additional fault detection is performed at the global level exploiting the spacial correlation and/or analytic redundancy between the sensors. Information regarding misbehaving sensors (or bad actors) is used by a fault aware fusion block to estimate the event of interest. The estimated value is then fed to an ideal sensor model that calculates the *no-fault* sensor reading. This ideal sensor reading is fedback to the local FDIR module for fault discrimination at the next computation cycle. This information is then feedback to each sensor so that they can correct themselves appropriately.

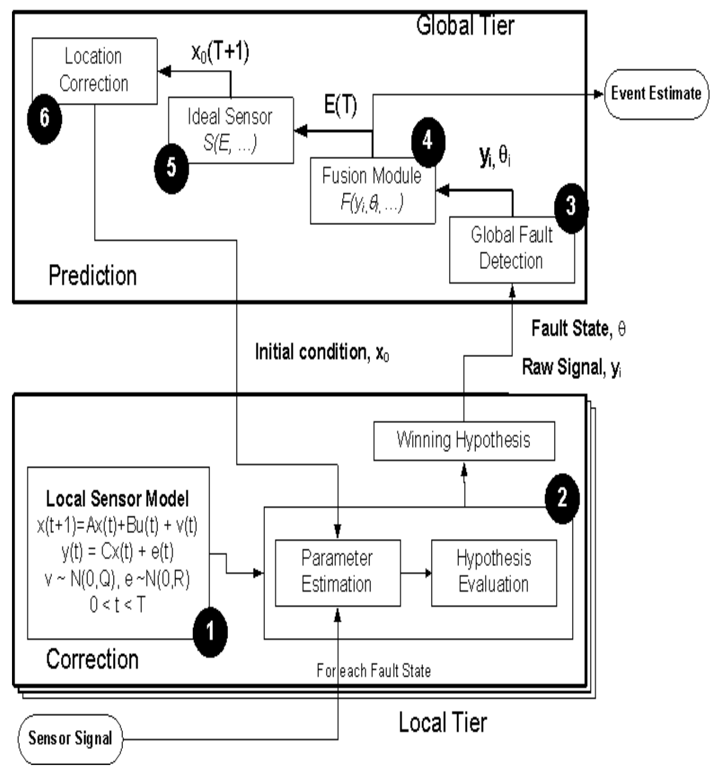

Fig. 1. The proposed two-tier framework for sensor fault detection

The framework defines a two-tier computational cycle—a faster fault diagnostic cycle at the sensor level and a slower data fusion cycle at the network level.

Continuous feedback from the global tier to the local tier formulates a predictor-corrector scheme wherein (1) a faulty sensor is progressively isolated (2) a fault-aware fusion progressively discounts information provided by a fault sensor; and (3) contributes positively towards network

QoI. In the following subsections we describe the components of the framework in detail.

A. Local Fault Detection and Characterization

The local sensor fault model assumes that measurand of interest $x(t)$ is an abstract or hidden variable, the presence of which can be established only through the observable variable $y(t)$. We further assume that the model holds for a small time window $0 \leq t \leq T$. We further postulate that for a non-faulty and accurate sensor, $y(t) \approx x(t)$ over this window subject to measurement delays and noise introduced within the transducers and digital conditioning electronics. An estimate of ground truth for the measurand is available only at $t = 0$. The relationship between the hidden measurand and the observed sensor signal is described using a first order linear model,

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + \nu(t) \\ y(t) &= Cx(t) + \epsilon(t) \\ 0 \leq t \leq T, x(0) &= x_0, u(t) = 1 \quad \forall t. \end{aligned} \tag{1}$$

Qualitatively, this formulation can be described as follows: over a time window T, we assume that the sensor responds to changes in the measurand x according to a linear dynamical model. The dynamics of the transducer response model is determined by A, while $\nu \sim N(0, Q)$ indicates the noise introduced in the transducer. $B \neq 0$ is indicative of bias introduced in the sensor. C is the calibration or the gain factor. Finally, $\epsilon \sim N(0, R)$ signifies the noise introduced by the sensor electronics.

A sensor is said to *misbehave* when the signal it provides to a network $y(t)$ does not reflect the measurand of interest $x(t)$. The larger is the deviation, the less reliable is the information provided by that sensor and thus should be discounted from a network fusion point-of-view. A *fault vector* is a mathematical characterization of this misbehavior. We further categorize the misbehavior into finite number of *fault states*, each of which is characterized using a parametric *fault signature*. Let θ and $FS(p)$ denote the fault state and the fault signature respectively. We use the notation $\theta_0, \theta_1, \theta_2, \ldots, \theta_*$ to denote the finite set of fault states such that θ_0 denotes a no-fault normal state for the sensor and θ_* denotes an unknown state for the sensor and $\theta_1, \theta_2, ...$ are the defined fault states in table I. Ω denotes the complete sensor fault set. At any given time, the ith sensor can be in $\theta \in \Omega$ state.

Combining equation (1) with the concept of a fault signature $FS(p)$, we note that: $p \iff \{A, B, C, Q, R\}$, and θ_k are *labeled regions* in this 5-dimensional parameter space. A sensor with no fault is defined as the one that responds to the measurand with a well defined time constant, well defined transducer noise and bias. Under these conditions, the response signal will contain noise induced because of the electronics. That is,

$$\begin{aligned} &\theta_0 : p_0 \iff \{A_0, B_0, C_0, Q_0, R_0\} \\ &a_l < A_0 < a_u; b_l < B_0 < b_u; c_l < C_0 < c_u; \\ &q_l < Q_0 < q_u; r_l < R_0 < r_u \end{aligned} \tag{2}$$

defines baseline values for no-fault sensor. Algorithmically, baseline values p_0 are established using no-fault data and the upper bound and lower bound thresholds are expressed as an acceptable bounds for the specific application. For any new set of measurements from sensors, the first step of the fault detection is to estimate the parameters of the proposed linear model in (1). Table I enumerates the misbehavior categories, namely θ_k and corresponding estimation and hypothesis testing steps. The proposed framework specifies that the local diagnosis module performs the basic task of fault state identification and fault signature estimation.

To make the estimate process easier, for any θ, we can partition p as $\{p_a; p_u\}$ where p_a denotes *assumed* parameters and p_u denotes *unknown* parameters for which we have an expectation given the fault state. Clearly, one needs to estimate p_u given a series of $y(t), 0 \leq t \leq T$, p_a and x_0. Conversely, given $\theta_k \leftrightarrow p_u$ for $k = 0, 1, 2, \ldots$ one can perform a series of estimations for each of the fault states. The error from each of these estimation can then be used as a basis for hypothesis testing. That is, the θ that gives the minimum error is the most likely fault state for the sensor. For completion as a candidate algorithm, we define this problem as a generic least squares minimization problem.

TABLE I
ENUMERATION OF MISBEHAVIOR AT EACH SENSOR LEVEL

Fault State	p_a	p_u	Hypothesis
θ_0, Normal	$A = A_0, B = B_0, C = C_0, Q = Q_0$	R	$r^l < R < r^u$
θ_1, Noisy	$A = A_0, B = B_0, C = C_0, R = R_0$	Q	$Q > q^u$
θ_2, Frozen	$A = A_0, B = B_0, C = C_0, R = R_0$	Q	$Q < q^l$
θ_3, Saturation	$A = A_0, B = B_0, Q = Q_0, R = R_0$	C	$C \neq [c^l, c^u]$
θ_4, Bias	$A = A_0, C = C_0, Q = Q_0, R = R_0$	B	$\|B\| > b^u$
θ_5, Spike	$B = B_0, C = C_0, Q = Q_0, R = R_0$	A	$\|A\| > a^u$
θ_6, Oscillation	$B = B_0, C = C_0, Q = Q_0, R = R_0$	A	$\|A\| \gg a^u$
θ_*, Unknown	None of the above		

$$\begin{aligned} &\min \sum_i^T (y(i) - \bar{y}(i))^2 \\ &\text{where:} x(t+1) = Ax(t) + Bu(t) + \nu(t) \\ &\qquad \bar{y}(t) = Cx(t) + \epsilon(t),\ 0 \leq t \leq T \\ &\qquad x(0) = x_0,\ u(t) = 1 \quad \forall t \\ &\text{such that} : p \iff \{A, B, C, Q, R\} = \{p_a; p_u\} \\ &\qquad p_a, \text{given.} \end{aligned} \tag{3}$$

It is easy to envision that sophisticated algorithm can generate a reconciled value $\hat{y}_i(t)$ for the ith sensor as a side effect of its calculation. To handle this case, we augment the fault vector to include this reconciled value, if available. To summarize, we have formulated the local sensor fault characterization as (1) an estimation step, (2) hypothesis testing step and (3) a decision making step wherein we select the most appropriate hypothesis based on the estimation error. It goes without saying that the linear dynamic model (equation 1) is the basis for this analysis.

Mathematically, the local fault characterization module defines an *operator* $\mathcal{A}$ or *algorithm* at each sensor level that operates on the raw sensor signal from the ith sensor and produces the following outputs. Note that the sensor generates these outputs (fault vector) once over T time interval.

1) Sensor Fault state, $\hat{\theta} \in \Omega$
2) Fault Signature parameters, $\hat{p}$ for the corresponding signature FS
3) (optional) A corrected or reconciled signal value $\tilde{y}_i(t)$

Now we can formally define the two forms of the fault vector.

$$fv(kT) = \{\theta, p, \emptyset\} \; or \; \{\theta, p, \hat{y}\}, \quad k = 1, 2, \ldots \tag{4}$$

B. Global Fault Detection

The local decisions or raw data (with possible lower sampling rates), augmented with the fault vectors will be send to the global tier. The global tier has the network-wide view of the data. It uses spatio-temporal correlations and also sample redundancies (for dense network topologies) to validate the fault detections at local tiers. As we have assumed that in our applications only small ratio of sensors can face faulty behaviors, in the case of detection of similar behavior from highly correlated sensors, global tier will consider it as an abrupt environmental change instead of faulty behavior. Since with the only local view of the measurements at sensors we can not distinguish between a sensor fault and an unexpected variation in the phenomena, a global fault detector is crucial to validate the fault notifications from local tiers.

C. Fault-aware Fusion

The previous section described a formulation for augmenting the signal from ith sensor with a fault vector. In this section, we describe a formulation wherein a network fusion algorithm can utilize faulty behavior information. It is clear that, sensor faults can significantly affect the accuracy of the fusion results.

Let E be an appropriate description for the "event of interest". In an abstract sense, data fusion can be defined as an operator $\mathcal{F}$ that takes the signals $y_i(t), i = 1, 2, \ldots N$ provided by N sensors to generate E. For the sake of completeness, we may say fusion generates $E(t), t > 0$. Symbolically, we can write the fusion as equation (6). One must note that the equation is not a mathematically precise definition considering asynchronous y_i signals.

Our objective is to propose a new operator $\mathcal{F}'$ when the ith sensor provides a fault vector every T units of time in addition to the raw signal y_i. Although we are investigating more advanced fusion schemes, for the purpose of this work we have started with a simple fault aware fusion based on averaging. In this work, event of interest is characterized by averaging various signals. The weights given to individual signals may depend on the reliability of the sensor (weighted averages) or ignorance of a sensor (Dempster-Shafer). Normalization of the signals to make them consistent is a pre-requisite.

The ith sensor has provided a fault vector $\{\theta, p\}$ in addition to the raw signal y_i at time t. We define a function $d : \Omega \rightarrow [0, 1]$ such that:

$$d(\theta \in \Omega) = \begin{cases} 1 & \text{include } i\text{th sensor} \\ 0 & \text{exclude } i\text{th sensor} \end{cases} \tag{5}$$

We include the ith sensor in the fusion depending on the fault state identified for the sensor. A simple approach could be $d = 1 \Leftrightarrow \theta = \theta_0$. If and only if the local fault diagnosis algorithm declares that the ith sensor is normal, we include it in the fusion, else we ignore it. This decision making represents an extreme choice among those available. In fact, one may argue why bother enumerating various θ if the fusion is not utilizing this information. Conversely, the motivation behind defining a Ω set and all the discussion presented in §II-A is to allow partial usage of the information and minimize making such extreme decisions. This is where having fault signature helps. In our future more advanced fault aware fusion schemes, we will use this extra information. Moreover, the discussion in subsection II-E shows how we can benefit from this information to model the faulty behavior of the sensors in the network over time.

D. Ideal Sensor Model

A sensor model is an idealized model that describes how a specific modality responds to an event E. In other words, this model establishes what the ground truth should be for the estimated $\hat{E}$. The main assumption here is that phenomena and event of interest are not fast changing. Clearly, the model will be specific to a sensor modality. The ground truth may be conditioned or corrected for the actual geo-spatial location of the sensor or the appropriate engineering units. In our framework, this module provides the appropriate x_0 for all sensors for the next cycle of fault detection. As shown by the block labeled 5 in figure 1, we can model it as following:

$$\hat{E}(t) = \mathcal{F}(y_i(t), \theta_{k,i}(t))), \ t > 0 \tag{6}$$

$$x_{0,i}(t+1) = \mathcal{S}(\hat{E}(t)), \ t > 0 \tag{7}$$

E. Network Fault Localization and Isolation

In this section, we outline a scheme for localizing the sensor fault in a sensor network and possibly isolation of those faults. Fault localization should not be confused with fault isolation. Fault isolation identifies the root cause of a sensor misbehavior. Fault isolation requires in-depth knowledge of the sensor internals which is not commonly known to an end user. Localization, on the other hand, identifies a sensor that may be misbehaving without providing a root cause. In most cases, localizing the problem helps the sensor manufacturer as well as the user to triage between a replace or a repair action. Often the localization is succeeded by troubleshooting procedure (manual or an automated test equipment), which quickly narrows down the root cause to make this decision.

We propose a simple fault localization scheme based on sequences of fault vector being generated by the ith sensor at each time interval T. That is, out starting point is a sequence of $\{\theta, p\}$, the proposed localization logic *annotates* the sensor misbehavior to assist the replace/repair decision. Misbehavior at the ith sensor is annotated as *persistent* or *intermittent* depending on duration of the fault. A persistent fault remains forever till a repair or replace action is performed. That is when the ith sensor enters a θ_k state, it remains there forever. An intermittent fault, on the other hand, switches between $\theta_k, k > 0$ and θ_0 fault states. This gives us a simple logic to classify the sequence of $\{\theta, p\}$ as intermittent or persistent fault. The annotation *progressive* is used to indicate the progression of the misbehavior over time. That is for any $\theta_k, k \neq 0$, we monitor p over successive visits to this same state. Progressive behavior is indicated either as a monotonic uptrend or downtrend in the estimated p values.

III. APPLICATION

In the next two subsections, we describe two application examples to demonstrate the effectiveness of the proposed framework for fault detection and study the impact of sensor faults on a pre-defined QoI metric for the network.

A. Acoustic Event Detection

In this application, the affect of fault detection and event detection coupling is studied. The scenario parameters are listed in table II.

TABLE II
ACOUSTIC SOURCE APPLICATION SCENARIO PARAMETERS

Source or Ground Truth	
Event signal, $E^*(t)$	$50, \ 0 \leq t \leq 10$
Event location	$(2, 2)$
Simulation period	$0 \leq t \leq 100$
Event occurrence times	$[20, 30], [70, 80], [90, 100]$
Received signal at k^{th} sensor, $E_k(t)$	$a_k E^*(t - \tau_k) + n_k(t)$
Attenuation, a_k	$1/(1 + d_k^2)$
Propagation Delay, τ_k	$d_k/\nu \ (\nu = 344m/s)$
Noise	$n_k(t)$
Network Characteristics	
Model update period, T	1
Sampling period,	0.05
Sensor numbers	4
Sensor locations	$(1, 1), (5, 1), (5, 5), (1, 5)$
Noise Var.σ_{s_k}	$1, \forall k$
Fault Characteristics	
Faulty sensor ID	1
Type	Multiplicative bias
Signature	0.5

Location of sensors, source and source signal are known while the time of event occurrence is unknown. In the specific studied scenario, we have assumed one of the worst cases where the closest sensor to the event has a multiplicative bias of less than one (0.5). A local tier running at nodes uses a Bayesian event detection hypothesis test along a local fault detector.

At each T, local tier detects faults by checking the estimated parameters with their acceptable bounds, as figure 2 shows, for sensor 1, parameter B doesn't stay in the acceptable bound, which shows the presence of fault at it.

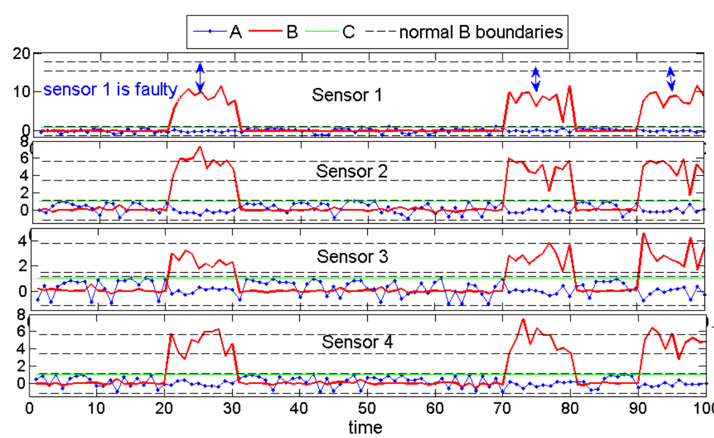

Fig. 2. Trace of model coefficients

After each T, global tier receives local detection decisions along with fault notification. Global tier cross validates the fault notifications and removes the faulty sensors from the process of event detection for that time frame. In global fusion of event detection different sensors have different weights, the closer sensors have higher weights in fusion. The global fault detector uses the fault notifications and also distance information to make a more robust final decision on event detection. In the simple fault-aware fusion that we have selected for this work, the sensor which is considered as a faulty is not considered in fusion process. If $\mathbf{S}$ is a set of all the non-faulty sensors then the global decision from the fusion process will be event occurrence if the following threshold test is satisfied,

$$\sum_{s_k \in \mathbf{S}} \left(\frac{I_k}{d_k^2}\right) \geq \sum_{s_k \in \mathbf{S}} \left(\frac{0.5}{d_k^2}\right) \quad (8)$$

where $I_k \in \{0, 1\}$ is the local decision on event occurrence from the non-faulty k^{th} sensor, and d_k is the relative distance of the k^{th} sensor to the event location.

Final decision which is either event occurrence or no event occurrence will be sent to ideal sensor models to estimate the ground truth for the local tiers. Sensor model in this case has used the propagation model of the acoustic signal, as it is shown in table II to estimate the initial state of the local models for the next cycle of fault detection. In the case of no event occurrence decision in global decision, noise is the only received signal at the sensor locations and the estimate of $x_0 = 0$ will be sent to the local sensors. In the case of event occurrence decision, having the signal of the event $E^*(t)$ and ideal sensor model, system can estimate the value of the received signal at place of the k^{th} sensor $(E_k(t))$.

As figure 3 shows for the whole scenario period (100 time units), event occurs 3 times. Figure 3 shows that our two-tiered architecture has efficiently improved the event detection at presence of fault and it showed that without that there is more chance to make wrong decisions due to faulty data from sensors with misbehavior. Figure 3 compares results of event detection with no fault detection with the event detection which is capable of fault detection. In this specific example, the multiplicative bias has attenuated the signal of the sensor 1, which results in missing the event.

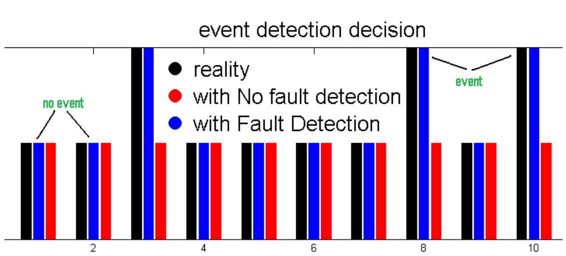

Fig. 3. comparing decisions for different cases

B. Hot Spot Intensity Estimation

In this application, the affect of fault detection and event intensity estimation is studied. The scenario parameters are listed in table III.

Location of sensors, source and source signal are known while the intensity of a hotspot is unknown. The location of the hotspot is known and the sensors are immobile. Figure 4(A) depicts the layout of the temperature sensors. A total of 10 sensors are used. Figure 4(B) shows the ground truth for the event of interest E_{truth}, namely the intensity of the hotspot as it evolves in time.

TABLE III
HOT SPOT APPLICATION PARAMETERS

Ground Truth	
Hot spot intensity, $E^*(t)$	$200 + 300(1 - e^{-t/3})$
Hot spot location	$(0, 0)$
Simulation period	$0 \leq t \leq 80$
Propagation model	$T_{diff} = (T_{hot} - T_{amb}) - \log(T_{diff})/(d_k^2 + e_k^2)$
Network Characteristics	
Sensor number	10
Sensor locations	$(-5, 27), (-10, 18), (4, 22), (17, 14)$ $(-24, -9), (-27, 10), (-15, 13), (25, 9),$ $(-30, 0), (12, 21)$
Noise Var. σ_{s_k}	$1, \forall k$
Fault Characteristics	
Faulty sensor ID	3
Type	Additive bias
Signature	$-4°C, 30 < t < 50$

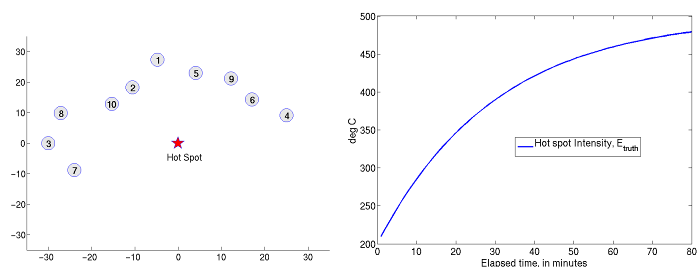

Fig. 4. (A): Sensor layout (B): The ground truth

Our task is to generates estimates for the hotspot intensity $\hat{E}$ using the sensor network, once every minute. The fault aware fusion function $\mathcal{F}$ is a 2D lookup table—which maps the hotspot temperature with the average value provided by all sensors in the network at time T. If the ith sensor was detected to be misbehaving, we did not use it in the average calculation. The ideal no-fault sensor model $\mathcal{S} = \mathcal{F}^{-1}$ and hence the same x_0 is fedback from the global tier to each local sensor FDIR model. The local FDIR model parameters are calculated using 10 samples within each 1 minute window. p_0 parameters are established using measurements within the first 7 minutes.

Figure 5 shows the trace of the model parameter B. It clearly shows that the "sensor bias" fault hypothesis is established between $30 - 50$ minutes. Figure 6 shows that the two-tier sensor FDIR framework consistently produces less estimation error and consequently higher network QoI.

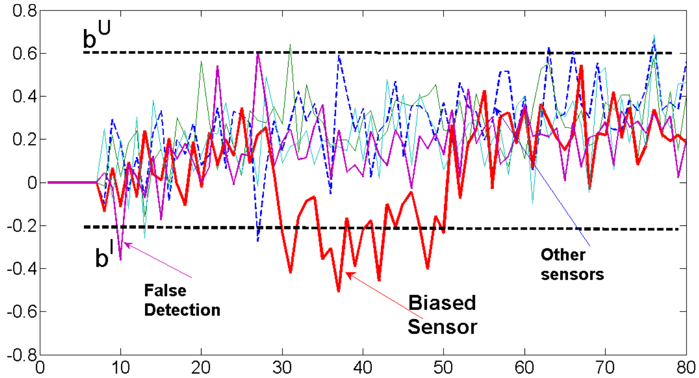

Fig. 5. Trace of model coefficient B

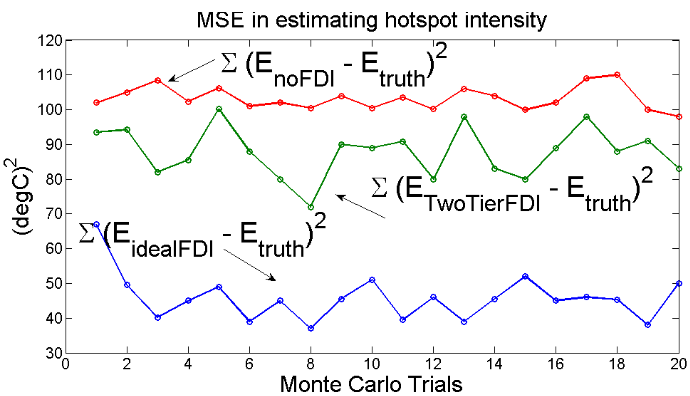

Fig. 6. Ground truth estimation error with and without sensor FDIR

To show the impact on network QoI and effectiveness of the proposed approach, we calculated traces of $\hat{E}$ under three conditions: no sensor FDI present, two-tier based sensor FDI and ideal sensor FDI (since we know the time period when the sensor is biased).

IV. CONCLUSIONS

The work presented in this paper, proposes a tiered fault detection system which detects faults in an online fashion. Tiered architecture of the proposed system makes it both effective in presence of faults and cost efficient. This system follows a sequence of repeating steps: fault analysis at the local sensor level to generate a fault vector; fault detection at global level that uses spatial correlation between sensors to distinguish faults from abrupt changes in the environment; global level fusion that uses faulty behavior information of the sensors to generate a robust estimate for the event of interest; the event estimate feeds back to an ideal sensor model to generate a reference signal (an instant of ground truth) for the i^{th} sensor; this reference signal is used by the fault analysis algorithm in its next update cycle. The continual feedback of an estimated measure of ground truth helps to increase the confidence of fault localization with time. Better fault characterization at the local level, will result in smarter fault-aware fusion rules at the network level, which in turn allows accurate ground truth estimation. This sequential approach will result not only in accurate fault localization in a sensor network, but also increasing the network QoI by systematically ignoring the information provided by misbehaving sensors.

V. ACKNOWLEDGEMENT

Research was sponsored by US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

[1] A. Benveniste and M. Basseville, *Detection of Abrupt Changes in Signals and Dynamical Systems*. Springer, 1984.

[2] S. Qin and W. Li, *Detection and identification of faulty sensors in dynamic processes with maximized sensitivity*. AICHE Journal, 2001, vol. 47:1581-1593.

[3] K. Marzullo, *Tolerating Failures of Continuous-Valued Sensors*. CM Transactions on Computer Systems (TOCS), November 1990, vol. 8.

[4] E. Elnahrawy and B. Nath, *Context-Aware Sensors*. Lecture Notes in Computer Science (LNCS), 2004, vol. 2920:7793.

[5] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli, *On-line fault detection of sensor measurements*. IEEE Sensors, 2003.

[6] J. Chen, S. Kher, and A. Somani, *Distributed fault detection of wireless sensor networks*. DIWANS, 2006.

[7] X. Luo, M. Dong, and Y. Huang, *On distributed fault-tolerant detection in wireless sensor networks*, 2006.

[8] R. Rajagopal, X. Nguyen, S. C. Ergen, and P. Varaiya, *Distributed Online Simultaneous Fault Detection for Multiple Sensors*. International Conference on Information Processing in Sensor Networks, 2008.