

# On-line Fault Detection of Sensor Measurements

Farinaz Koushanfar  
EECS Dept., UC Berkeley  
Berkeley, CA 94720  
Email: farinaz@eecs.berkeley.edu

Miodrag Potkonjak  
CS Dept., UC Los Angeles  
Los Angeles, CA 90095  
Email: miodrag@cs.ucla.edu

Alberto Sangiovanni-Vincentelli  
EECS Dept., UC Berkeley  
Berkeley, CA 94720  
Email: alberto@eecs.berkeley.edu

**Abstract**—On-line fault detection in sensor networks is of paramount importance due to the convergence of a variety of challenging technological, application, conceptual, and safety related factors. We introduce a taxonomy for classification of faults in sensor networks and the first on-line model-based testing technique. The approach is generic in the sense that it can be applied on an arbitrary system of heterogeneous sensors with an arbitrary type of fault model, while it provides a flexible trade-off between accuracy and latency. The key idea is to formulate on-line testing as a set of instances of a non-linear function minimization and consequently apply nonparametric statistical methods to identify the sensors that have the highest probability to be faulty. The optimization is conducted using the Powell non-linear function minimization method. The effectiveness of the approach is evaluated in the presence of random noise using a system of light sensors.

## I. INTRODUCTION

### A. Motivation

Testing and fault detection has emerged early as an important aspect of the design and analysis of computer and communication systems. In traditional computer systems, a significant percentage of circuitry is dedicated to testing, either in the form of built-in self-test (BIST) subsystems or, in the form of signature verification subsystems. Also, in DRAM memories, built-in self-repair (BISR) techniques are regularly used to significantly enhance the yield. Modern testers are very expensive and therefore speed of testing is critical for economic viability. For example, recent IBM studies indicate that typical wireless systems, such as cell phones, have to be tested for less than one-third of a second in order to be economically competitive. It is important to note that testing of wireless devices is greatly complicated due to the presence of the radio frequency (RF) components.

There is a wide consensus that both off-line and on-line testing of wireless sensor network nodes is a very important and demanding task. There are at least four major groups of reasons for the importance of testing:

- 1) Technological trends. Technological aspects of a design make testing a critical step. On one hand, several components of a sensor network node, such as wireless communication components and sensors and actuators are significantly more prone to faults than standard VLSI computing components. On the other hand, wireless sensor network nodes are significantly more complex systems than traditional computing systems, in particular in terms of their diversity.

- 2) Application trends. Requirements of applications often render testing of sensor network nodes important and/or difficult. Sensor network nodes are often deployed in uncontrollable or even harsh or hostile environments. Furthermore, the typical applications are often under the strict cost and energy constraints and many sensor networks have to operate in a real-time mode where error detection has to be conducted with low latency or high throughput.
- 3) Conceptual Novelty. From a conceptual point of view, it is difficult to address testing issues before a solid understanding of both hardware and software issues of sensor nodes is available. There is still a long series of disagreements among system designers about the most suitable organization and the proportion of hardware and software components of a node. Therefore, it is difficult to envision what kind of requirements and constraints will be imposed on fault detection procedures.
- 4) Critical Functionality. From a functionality point of view, it is important to recognize that a significant percentage of sensor networks will be involved in safety critical applications such as the mechanic parts of automotive systems and nuclear reactors. In this type of network continuous frequent on-line testing is of paramount importance.

### B. Objectives and Paper Organization

Our primary goal is to present the first non-parametric statistics-based technique for on-line testing of the nodes in a sensor network. In order to accomplish this task, we start by defining four phases of a test procedure. After that, we define several sensor fault models. The technical core of the paper is an on-line testing approach that consists of five steps and is conducted simultaneously with sensor fusion. In the first phase, test data is collected. In the next two phases, each piece of data is properly weighted and analyzed. After that, separate steps are used to best distinguish the importance of a particular dataset by considering properties of the environment. Finally, statistical percentile resubstitution methods are used to establish the interval of confidence for testing decisions. Multimodal sensor fusion is conducted by mapping the original problem to an instance of a non-linear function minimization. The effectiveness of the technique is analyzed both experimentally and through simulations using the light sensors platform. We conclude the paper by outlining

the most-likely directions for the future research for testing in sensor networks.

## II. PRELIMINARIES

In order to make the presentation self-contained we present background material on sensor networks and testing and introduce all relevant assumptions and abstractions.

### A. Sensor Networks

Sensor networks consist of a number of communicating nodes. Each node is equipped with computation, communication, storage, supply, sensing and in some cases actuating subsystems. Sensor networks have the potential to change the way humans interact with the physical world, by adding a new dimension of sensing (and eventually actuating) to the network of communicating, recording and computing devices. For example, by combining several different dimensions of a sensed phenomenon (multi-modal sensor fusion), one can deduce the properties of that phenomena. However, the large number of heterogeneous nodes tremendously adds to the complexity of the network and calls for a paradigm shift for the design of such networks with respect to conventional computing systems. For instance, nodes of a sensor network have to be deployed in a physical environment and adapt to the often harsh conditions in such environments. The limited battery supply on the wireless nodes dictates a need to have short range communications and localized interactions between the nodes. Note that short range communication is proven to consume orders of magnitude less energy than its long range counterpart [1]. Furthermore, manual maintenance, error detection, and debugging of the nodes become cumbersome if not impossible tasks in sensor networks. Therefore, there is a need to develop a means to automatically test the nodes in a sensor network in order to detect errors and faults without the involvement of a human.

### B. Testing

One can identify five phases in the testing of sensor based systems. The first phase is test vector generation where the goal is to generate inputs to sensors that are most likely to excite a particular subset of faults. In the case of sensor fusion-based testing, excitation of faults can be defined as the process of inducing a subset of sensors to have a particular relationship between their readings that indicate the presence. Test vector generation is in particular a demanding task when testing is conducted on-line. In this situation, the most attractive alternative is to use one or more actuators in addition to the already available sources of excitations. The actuators should create readings that maximize the chances for detection of faults in the maximum number of sensors. For example, the actuator that is at a symmetric position with respect to three or more sensor can be used to test the differences in two consecutive snapshots, one when the additional actuator is not active and one where it active, identical within some limit.

The second phase is error detection where the goal is to identify all sensors that have incorrect readings due to

systematic sources of errors. There are two main sources of difficulties related to error detection. The first one is a consequence of the fact that a single fault is not realistic for sensor networks and therefore, faults at different sensors can have compound effect on the sensor fusion task. The second problem is that any type of sensor is always also subject to random transient fluctuations that can mask non-transient systematic faults. Therefore, in many cases the only viable alternative to distinguish between the transient and non-transient faults is to consider multiple snapshots of the sensor readings at different time instances.

Once the errors are detected, we enter a error diagnosis stage, where the nature of the fault and how it impacts reading is being identified. The other phase related to the error diagnosis phase is a phase of error evaluation and validation that establishes the interval of confidence on the result of the analysis with respect to each particular error. Note that due to random fluctuations, we rarely can fully guarantee that some nodes are fault-free. However, the reverse situation is not true and for some type of error models, we can fully prove that a particular sensor is faulty.

The final phase of testing a distributed sensor system is the error correction step where raw sensor readings are remapped in such a way that correct or approximately correct readings are obtained. Note that the error correction phase is only applicable to some error models. Furthermore, note that our effort is currently restricted only to faults in sensors and does not consider faults in other components such as computation and communication units of a node. At the same time, it is important to emphasize that our methodology and techniques can be readily applied to controllable actuators too.

### C. Additional Assumptions

There are two main assumptions with respect to the observed physical phenomena and density and placement of sensors. With respect to the phenomena, the main assumption is that phenomena can be adequately described by a set of equations. We do not place any restrictions on the type or on the way the equations are specified. However, we do place a restriction on number of the sources of excitation to be small enough so that it can be properly captured by a given number of sensors. With respect to the density and placement of the sensors, we assume that sensors are dense enough in the sense that number of non-faulty sensors is sufficient to accurately specify the phenomena and the impact of the environment. Finally, the sensors are placed in such a way that the equations are not linearly dependent or at least, there exists a subset of equations that are sufficient to uniquely capture the phenomena.

In our experimentation and simulations, we translate the problem of solving a system of non-linear equations into a non-linear function minimization problem. The non-linear function minimization problem is being solved using the Powell method [2], [3]. There is a great variety of techniques that map solving a non-linear system of equations into a problem of minimizing a non-linear function subject to a

set of nonlinear constraints. While obviously this later option greatly facilitates easiness of specifying various constraints it is important to note that solvers for this second problem are significantly less efficient than for the first formulation. They are also much more complex and therefore may not be so well-suited for the deployment in low cost low energy sensor nodes.

### III. RELATED WORK

In this Section, we outline the related work along the following lines: wireless ad-hoc sensor networks, sensor fusion, testing, fault detection in sensor networks, and used statistical and numerical techniques.

Wireless ad-hoc sensor networks have recently emerged as a high impact research topic. A number of economically and scientifically important applications for wireless sensor networks have been envisioned [4], [5]. Due to the harsh environmental conditions, majority of measurements in such networks are subject to errors. Techniques for calibration to adjust for errors in wireless sensor networks have been recently proposed [6], [7]. Error detection and correction has been studied in the context of traditional wireless network. For example, Eckhardt et. al, presented a measurement and analysis of faults in wireless networks [8].

Multi-sensor data fusion is a problem that recently has attracted a great deal of attention in a number of scientific and engineering communities [9], [10], [11]. Majority of these efforts are restricted to sensor fusion of sensors of the same modality. Constraints, in addition to statistical models and analytical equations, are one of main building blocks for our approach. Constraint-based sensor fusion for vision has been advocated in [12]. The results of the scheme are applicable only to certain individual sensor faults and traditional networks. They are not applicable to the reliability needs in complex network organization and most importantly; they do not address the reliability issues that are induced by the ad-hoc nature of the wireless sensor networks. An approach for fault-tolerant multi-sensor fusion was presented in [13].

During the 70's, research on fault diagnosis and fault prevention started to diverge along several lines. Initially, the two most important and influential were correctness of VLSI-based systems [14] and fault-prevention in distributed system [15]. Concurrent system-level error detection techniques using a watchdog processor are surveyed in [16]. A watchdog processor is a small and simple coprocessor that detects errors by monitoring the behavior of a system. Like replication, it does not depend on any fault model for error detection. Fault detection and identification (FDI) has long been subject of active research in distributed systems. References [17], [18], [19] provide a comprehensive surveys on both experimental and theoretical based FDI. Stelling et. al. [20], propose a fault detection service designed to be incorporated, in a modular fashion, into distributed computing systems, tools, or applications.

A survey on fault identification in the context of fault management in the networks can be found in [21]. Automated

fault identification is a must, when the size of the network increases. Most of the related work in traditional networks such as the Internet is monitoring the network. There two ways monitor a network: time driven monitoring and event driven monitoring. The former type of monitoring collects multiple snapshots at certain times. The latter approach produces alert triggered by interesting events. The common practice is to use event-driven monitoring since it significantly reduces the traffic in the network [22], [23]. On-line fault diagnosis is also an important issue for automotive industry. For example, on-line fault diagnostic system requirements are introduced to control the emission of the vehicles [24].

Resampling techniques are statistical validation tools that often use statistics in which one or more data points are eliminated from the data in order to calculate bias, variance, interval of confidence and other measures of accuracy of the developed models. Usually either systematic exhaustive or random resampling are used. The idea is to calculate the statistics of interest using the subsets of data after a part of data is eliminated to evaluate the consistency of the obtained models. Jackknife (cross-validation) is a statistical resampling validation technique that was introduced by Quenouille [25]. The standard references include [26], [27], [28]. Jackknife eliminates all data points one at the time and calculates the statistical measures of interest.

Nonlinear programming involves three types of entities: variables, objective function, and constraints. The goal is to find a set of assignments to the variables in such a way that the objective function is minimized and at the same time, all the constraints remain satisfied. Linear programming problems are the ones that only contain linear objective function and constraints; nonlinear programming problems, on the other hand, do not pose this restriction of linearity on the objective function and constraints. [3] is a widely reference source. More specifically, for nonlinear programming, the useful sources include [29], [30]. There is a variety of linear and nonlinear commercial solvers for public usage. We used the conjugate gradient algorithm for an unconstrained nonlinear function optimization routine [3].

### IV. SENSOR FAULT MODELS

We introduce fault classification taxonomy for sensors. In addition to already suggested and used one, we also identify several new types of sensor faults. The proper starting point for testing of sensor-based systems is to establish adequate fault models. Fault models have to be expressive enough to properly capture all essential features of the most common faults. At the same time, they have to be conceptually and computationally simple enough so that they can be tractable. Distributed sensor-based systems are complex systems according to at least two criteria. They have a large number of components and they employ a layered and interactive system software and application software components. From the hardware point of view, we completely restrict our attention to sensors and actuators. From the functional level view we assume that all system software as well as the application software are already

fault tolerant. One way to accomplish software fault tolerance is n-versioning [31]. Another way is to use software self-checking techniques.

The most common sensor error fault is offset bias. Offset bias alters the sensor readings uniformly by a certain value. The generalization of the bias model is a drift fault model, where the correct value is subject to alternation that is a time invariant function of the correct value. For example, the error could result that a light sensor always reads only 60% of the correct luminance due to dust or other obstacles accumulated on a subset of its sensor cells. Probably the simplest model from both conceptual and difficulty to detect point of view is when a sensor is frozen to a particular value. Another fault model that is relatively easy to detect is one where the sensor is dead and does not report any values. Two types of error models are very complex in terms of their demands how to be treated. The first is one when the sensor readings variance is subject to degradation often due to the aging of the sensor. The other demanding model is the one where the sensor readings are transformed by a particular function that may or may not be dependent on the sensor readings. For example, in the case of light sensors, if there is an error in angle measurement, all readings will be transformed by a cosine function.

## V. CROSS-VALIDATION-BASED PROCEDURE FOR ON-LINE TESTING FOR SENSOR FAULTS

In this Section, we outline the cross-validation based real-time fault detection procedure for sensor networks. The key idea is to consider the impact of readings of a particular sensor on the consistency of multi-sensors fusion. If the elimination of a particular sensor significantly improves the consistency of the results, the sensor is most likely faulty.

The practical method in fault detection to distinguish a random noise is to run a maximum likelihood or Bayesian approach on the multi sensor fusion measurements. A random noise would exist, if running these procedures improves the accuracy of the final results of multi-sensor fusion. While there have been several efforts to minimize random errors, very little has been done for fault detection. In multi-sensor fusion, the measurements from different sensors are combined in a model for consistent mapping of the sensed phenomena. Although the new fault detection technique is generic and can be applied to an arbitrary system of sensors that use an arbitrary type of data fusion, for the sake of brevity and clarity we focus on equation-based sensor fusion.

We start by summarizing our sensor fusion process. Assume that we have a set of sensors  $s_i$  ( $0 \leq i \leq n$ ), each measuring a value  $x_i, t$  at a time  $t$ . The multimodal sensor fusion model equations are  $f_1, \dots, f_n$  are typically non-linear functions, and have the following forms:  $f_j(x_1, y_1, y_2, \dots, x_n) = 0$ , ( $0 \leq j \leq n$ ). For example, in the case of light, the reading of each sensor is proportional to superposition of the product of the incident angle and inverse of the square of the distance over all light sources. The system of equations has to be over-constrained in order to be solved in presence of errors. We solve the system by translating it into an instance of non-linear

function minimization and consequently applying the Powell method to obtain the solution [3]. The nonlinear function is the weighted sum of errors for all sensor readings. Note, the obtained solution provides both characteristics of the excitation sources as well as an indication of the estimated inconsistency of the solution.

The sensor fusion process is the basis for the cross-validation based on-line testing. The cross-validation-based on-line test procedure can be summarized using the pseudo-code in the following figure.

```

Procedure cross-validation-based on-line testing:
1. Conduct Sensor Fusion using non-linear function minimization;
2. while (stopping-criteria == NO)
3. {
4.   For (i=1, i=n, i++)
5.   {
6.     Eliminate sensor i from consideration;
7.     Conduct Sensor Fusion using non-linear function minimization;
8.   }
9.   Identify most discrepant sensor;
10.  Update stopping-criteria;
11. }
12. Establish Intervals of Confidence for Eliminated Sensors;
13. Reinstate Eliminated Sensors with high Intervals of Confidence;

```

We solve the system  $n + 1$  times, each time after a slight modification. First, we solve with all the equations in the original format and then, we ignore each variable one at time and solve a less constrained system with  $n - 1$  variables ( $n$  times). We compare the values for each variable  $x_n$  in all  $n + 1$  scenarios. The procedure is iteratively applied to detect  $s$  faulty sensors (one with the highest discrepancy) at the time. The value for  $s$  is set by the user. In our experimentation, we noted that the best results are obtained when  $s$  is set to a very small integer value.

In order to improve the accuracy of fault detection, the system can be solved for  $m$  measurements by each sensor in  $m$  different time steps. At last, we conduct statistical analysis on the data for each sensor. If the obtained values for a sensor are not consistent within a confidence interval calculated by the

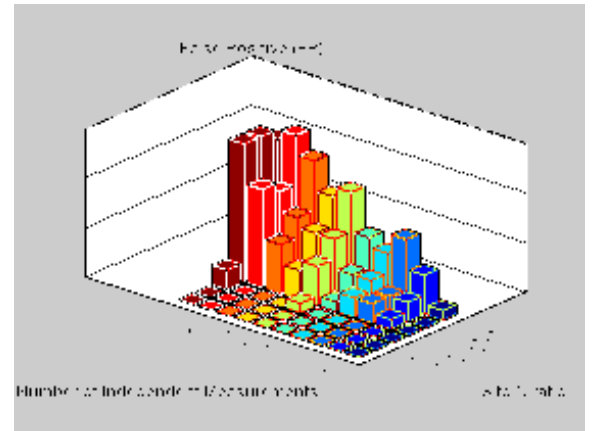


Fig. 1. Number of false positive reporting during fault detection for sensor network of 200 light sensors and 40 faulty sensors.

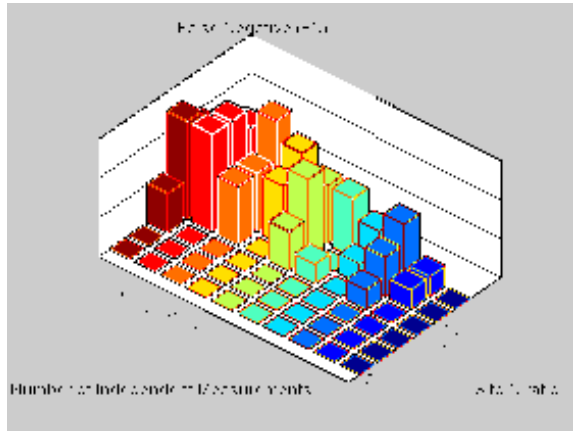


Fig. 2. Number of false negatives reporting during fault detection for sensor network of 200 light sensors and 40 faulty sensors.

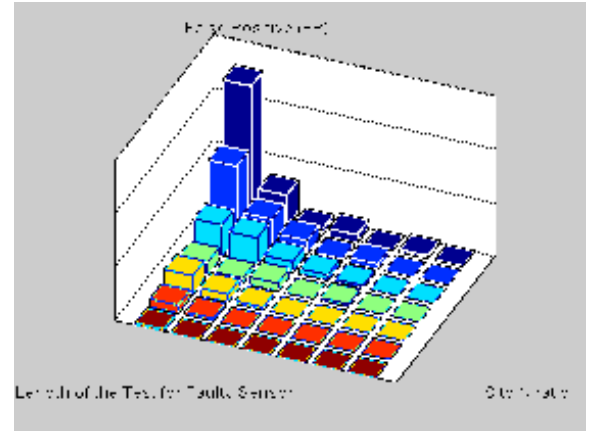


Fig. 3. Number of false positive reporting during fault detection for sensor network of 200 light sensors and 60 faulty sensors.

percentile method, the sensor is considered faulty. The interval of confidence is dynamically set according to the interval of confidence for the system that uses all sensor readings.

There are several minor algorithmic and software issues that significantly reduce run time and improve the quality of the obtained solutions. For example, runtime can be significantly reduced if starting points for the iterative non-linear function minimization during re-substitution are provided by adding a small uniformly distributed random noise around the solution obtained when all sensors are considered. Furthermore, one can use solutions obtained in step 7, as starting points for consequent invocation of sensor fusion procedure. Also, the accuracy of the fault detection can be significantly improved if the delayed binding scheme is used [32].

## VI. EXPERIMENTAL RESULTS

In order to evaluate the accuracy of the proposed on-line testing technique, we conducted a series of simulation and experimental studies. The primary goal of the evaluation was to analyze the impact of the percentage of faulty sensors, the signal-to-noise ratio, and the length of test on the false positive and negative error fault detections. Our driver example was the detection of faults in optical sensors in environments with point lights as the sources of excitation.

Simulations were organized in the following way. We generated scenarios with  $k$  light sources.  $k$  was uniformly randomly selected from the range [1, 12]. The intensity and the location of each source was randomly selected for rooms of 5 different sizes. For each measurement of each sensor, a Gaussian random noise was superimposed. The characteristics of the noise were specified as indicated in Figures 1, 2, 3 and, 4. On the specified percentage of the nodes an additional modification was superimposed. This modification had the same sign and an intensity that was 10 times large than the Gaussian random noise.

Figures 1 and 2 show the number of reported false positive (fault free sensor was indicated as faulty) and negative fault (a sensor with fault was not detected) on the vertical axis. In both

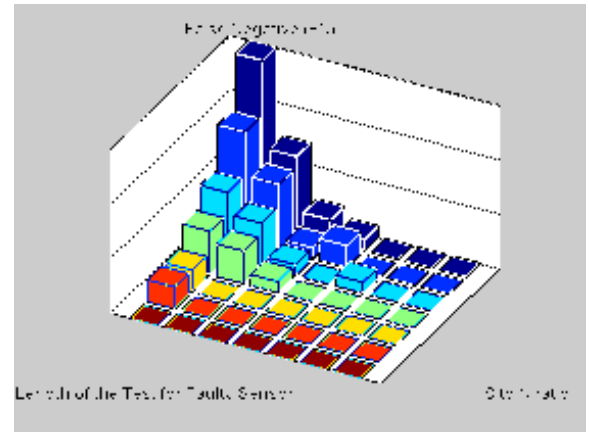


Fig. 4. Number of false negative reporting during fault detection for sensor network of 200 light sensors and 60 faulty sensors.

Figures, one of the horizontal axis indicates the the number of measurements for each sensor and the other shows signal to noise ratio. Figures 3 and 4 present the same evaluation information for the case where 60 out of 200 sensors were faulty. It is apparent that even for low average signal to noise ratios (intensity of correct signal vs. amount of superimposed noise), very few false reports were conducted. In particular, false negative readings were rare. It is also clear that the difficulty of fault detection significantly increases with large percentage of faulty sensors.

In our experimentation we used systems of 8 and 16 sensors. Figure 5 shows a subset of main components of the experimental platform. We have used light bulbs with very small diameter as the light source and conducted all the experiments in a dark room. We created faults in a particular subset of sensors by covering a part of the sensor using opaque film. In all of the situations where the sensor readings of a particular sensor was altered by more than 7%, we correctly identified all the impacted sensors in a smaller system configuration. In all of the situations where the sensor reading of particular sensor was altered by more than 4%, we correctly identified all the



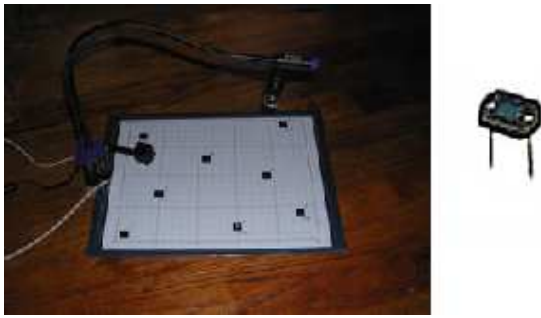


Fig. 5. Our testbed, consisting of a number of photovoltaic sensor elements (shown separately on the right) and different configurations of the sensors. The data is processed at a PC.

impacted sensors in a smaller system configuration.

## VII. CONCLUSION

We have developed a cross-validation-based technique for on-line detection of sensor faults. The approach can be applied to a broad set of fault models. We define a fault as an arbitrary type of inconsistent measurement by a sensor, which cannot be compensated systematically. In particular, we consider faults associated with the incorrect measurements that cannot be corrected using calibration techniques. The approach is based on two ideas: (i) comparing the results of multi-sensor fusion with and without each of the sensors involved; and (ii) using non-parametric statistical techniques to identify the measurements that are not correctable, regardless of the used mapping function between the measured and accepted values. Comprehensive simulations indicate high accuracy of the approach in presence of Gaussian noise even for relatively sparse sensor networks.

## ACKNOWLEDGMENT

This material is based upon work supported in part by the National Science Foundation under Grant No. ANI-0085773 and the NSF CENS Grant.

## REFERENCES

- [1] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy aware wireless microsensor networks," *IEEE Signal Processing*, March 2002.
- [2] R. P. Brent, *Algorithms for Minimization without Derivatives*. Englewood Cliffs, NJ: Prentice-Hall, 1973.
- [3] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C++*. Cambridge University Press, 2002.
- [4] I. F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cyirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [5] D. Estrin, R. Govindan, and J. Heidemann, "Embedding the internet: Introduction," *Communications of the ACM*, vol. 43, no. 5, pp. 38–42, 2000.
- [6] K. Whitehouse and D. Culler, "Calibration as parameter estimation in sensor networks," in *ACM WSNA*, 2002, pp. 59–67.
- [7] S. M. V. Bychkovskiy, D. Estrin, and M. Potkonjak, "Colibration: A collaborative approach to in-place sensor calibration," in *2nd International Workshop on Information Processing in Sensor Networks (IPSN)*, 2003, pp. 301–316.

- [8] D. Eckhardt and P. Steenkiste, "Measurement and analysis of the error characteristics of an in building wireless network," in *ACM SIGCOMM*, 1996, pp. 243–254.
- [9] R. R. Brooks and S. Iyengar, *Multi-Sensor Fusion: Fundamentals and Applications With Software*. Englewood Cliffs, NJ: Prentice-Hall, 1997.
- [10] P. K. Varshney, *Distributed Detection and Data Fusion*. N.Y.: Springer-Verlag, 1997.
- [11] G. Hager, *Task-Directed Sensor Fusion and Planning - A Computational Approach*. Kluwer Academic Publishers, 1990.
- [12] J. J. Clark and A. L. Yuille, *Data Fusion for Sensory Information Processing Systems*. Kluwer Academic Publishers, 1990.
- [13] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli, "Fault tolerance in wireless ad-hoc sensor networks," in *IEEE Sensors*, 2002, pp. 1491–1496.
- [14] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Kluwer Academic Publishers, 2000.
- [15] P. Jalote, *Fault Tolerance in Distributed Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
- [16] A. Mahmood and E. McCluskey, "Concurrent error detection using watchdog processors - a survey," *IEEE Transactions on Computer*, vol. 37, no. 2, pp. 160–174, 1988.
- [17] J. J. Gertler, "Survey of model-based failure detection and isolation in complex plants," *IEEE Control Systems Magazine*, vol. 12, no. 6, pp. 3–11, 1988.
- [18] R. Isermann, "Process fault detection based on modeling and estimation method - a survey," *Automatica*, no. 20, pp. 387–404, 1984.
- [19] A. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, no. 12, pp. 601–611, 1976.
- [20] P. Stelling, C. DeMatteis, I. Foster, C. Kesselman, C. A. Lee, and G. von Laszewski, "A fault detection service for wide area distributed computations," *Cluster Computing*, vol. 2, no. 2, pp. 117–128, 1999.
- [21] T. Oates, "Fault identification in computer network A review and a new approach," University of Massachusetts Amherst, Tech. Rep. UM-CS-1995-113, 1995.
- [22] D. Haban and D. Wybraniec, "A hybrid monitor for behavior and performance analysis of distributed systems," *IEEE Transactions on Software Engineering*, vol. 16, no. 2, pp. 197–211, 1983.
- [23] J. Joyce, "Monitoring distributed systems," *ACM Transactions on Computer Systems*, vol. 5, no. 2, pp. 121–150, 1987.
- [24] J. J. Gertler, M. Costin, X. Fang, R. Hira, Z. Kowalczyk, M. Kunwer, and R. Monajemy, "Model based diagnosis for automotive engines - algorithm development and testing on a production vehicle," *IEEE Transactions on Control Systems Technology*, vol. 3, no. 1, pp. 61–69, 1995.
- [25] M. Quenouille, "Notes on bias in estimation," in *Biometrika*, no. 43, 1956, pp. 353–360.
- [26] A. Davison and D. Hinkley, *Bootstrap Methods and their Application*. Cambridge University Press, 1997.
- [27] B. Efron and R. Tibshirani, *An Introduction to Bootstrap*. Chapman-Hall, 1993.
- [28] P. I. Good, *Resampling methods*. N.Y.: Springer-Verlag, 1999.
- [29] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory & Applications*. N.Y.: John Wiley & Sons, 1993.
- [30] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.
- [31] A. Avizienis, "The n-version approach to faulttolerant software," *IEEE Transactions on Software Engineering*, vol. SE-11, no. 12, pp. 1491–1501, 1985.
- [32] J. Wong, F. Koushanfar, S. Meguerdichian, and M. Potkonjak, "A probabilistic constructive approach to optimization problems," in *ACM/IEEE ICCAD*, 2001, pp. 453–456.