Fault Tolerance Techniques for Wireless Ad Hoc Sensor Networks

Farinaz Koushanfar
EECS Dept. UC Berkeley
Berkeley, CA, USA
farinaz@eecs.berkeley.edu

Miodrag Potkonjak
CS Dept. UCLA
Los Angeles, CA, USA
miodrag@cs.ucla.edu

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

Abstract

Embedded sensor network is a system of nodes, each equipped with a certain amount of sensing, actuating, computation, communication, and storage resources. One of the key prerequisites for effective and efficient embedded sensor systems is development of low cost, low overhead, high resilient fault-tolerance techniques. Cost sensitivity implies that traditional double and triple redundancies are not adequate solutions for embedded sensor systems due to their high cost and high energy-consumption.

We address the problem of embedded sensor network fault-tolerance by proposing heterogeneous back-up scheme, where one type of resources is substituted with another. First we propose a broad spectrum of heterogeneous fault-tolerance techniques for sensor networks including the ones where communication and sensing are mutually backing up each other. Then, we focus our attention on two specific approaches where we back-up one type of sensors with another type of sensor. In the first, we assume faults that manifest through complete malfunctioning and in the second, we assume sensors where fault manifest through high level of error. Specifically, we introduce techniques that enable efficient multimodal sensor fusion in presence of faults and errors. For each technique, we present efficient algorithms and demonstrate their effectiveness on a set of benchmark examples.

Keywords

Fault tolerance, wireless sensor networks, heterogeneous redundancy, and multimodal sensor fusion.

1. INTRODUCTION

Embedded sensor and actuator-based systems have potential to provide inexpensive and pervasive bridge between physical and computational worlds. At the same time, they are on the verge of redefining how computer-based systems are designed and used. Requirements for autonomous operation, localized information processing and storage, low energy, low cost, and, in particular, reliability and fault-tolerance emerged as premier and crucial system design desiderata. Although fault tolerance has been studied for several decades in computer and VLSI systems, tremendous intrinsic reliability of VLSI integrated circuits technology and operation in well-conditioned environments restricted the importance of fault tolerance in great majority of computing systems. However, sensor-based networks will often operate in potentially hostile, or at least harsh and unconditioned environments. Greater percentage of the applications of such networks will be mission critical, while they will have continuous mode of operation, higher structural complexity, and components such as sensor and actuators that have significantly higher fault rates than the traditional semiconductor integrated circuits-based systems. In addition, the maintenance and replacement of components will be often prohibitively expensive.

Therefore, requirements for low energy and in particular cost sensitivity imply that traditional double and triple redundancy fault tolerance techniques will not be adequate solutions for embedded sensor systems. Our goal in this paper is to analyze fault-tolerance related requirements in embedded sensor networks and develop techniques and algorithms to efficiently satisfy them.

We emphasize on the importance of heterogeneous fault-tolerance techniques, where a single type of resource backs up different types of resources. The key idea is to adapt application algorithms and/or operating system to match the available hardware and the applications needs. We envision that each of five primary types of resources: computing, storage, communication, sensing and actuating can replace each other with suitable change in system and application software. For example, if communication bandwidth is reduced and all of the computation power is available, the system can compress data using more computationally intensive compression schemes. Or, in the opposite situation, when computational power is reduced and communication is fully available, the node can transmit more raw data to other nodes for processing.

We focus our attention on how to back-up one type of sensor with another. There are two main reasons for this decision. The first is that technology trends indicate that sensing has by far the highest fault rates among the five types of resources. The second is that there is a wide consensus that multimodal sensor fusion is the key for successful and widespread use of embedded sensor networks.

The problem of fault-tolerant multimodal sensor fusion for digital binary sensors can be informally described in the following way using the example from Figure 1a-c. We first introduce the problem of multimodal sensor fusion. We assume that two types of sensors are given: one that measures height of the object and one that measures colors. Each object is unique in a sense that no two objects have simultaneously identical color and height. All sensors are binary. For example, each height sensor indicates the height of the observed object higher or lower than a particular value. Similarly, each color sensor uses a filter to indicate a color of particular object of a particular type or not. Suppose that we have 5 sensors as shown in Figure 1a. Furthermore, for the sake of simplicity, assume that both color and height sensors have the same cost. Finally, we assume that all measurements are exact. Sensor resource allocation and assignment problems ask to identify the minimal number of sensors needed to uniquely identify each object from the sensors readings. Sensor resource assignment is to identify the exact characteristics (in case of color sensors - color filter, in case of height sensors - height level) for each sensor so that classification can be conducted. Figure 1b shows one such solution that uses one color sensor and four height sensors.

Figure-1a)

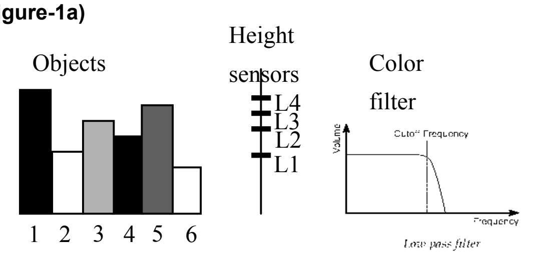

Figure-1b)

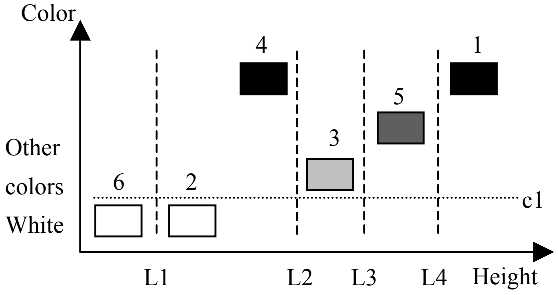

Figure-1c)

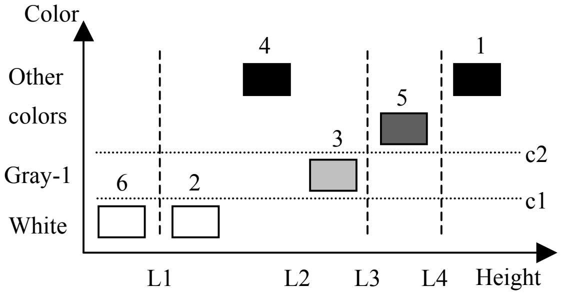

Figure 1 – an example for the multimodal (bi-modal) sensor fusion

What are our options if we want to design a fault tolerant solution where a single faulty sensor can be tolerated? The first one is to duplicate hardware. For embedded sensor networks this yields unacceptable overhead. In the case of our example, this means addition of 5 extra sensors. The second option is to add one extra sensor of each type as back-up. Now we can replace the failed sensor with the sensor of the same type. The overhead is significantly low - only two sensors.

However, we can do even better. Suppose that we allocate only four height and two color sensors. Therefore, the overhead is only one sensor. We are still able to properly function if any sensor gets faulty. This is because we can change our classification algorithm to adapt to available resources. Specifically, if one of color sensors gets faulty, we can use solution from Figure 1b. That solution uses 4 height and one color sensor. But, if one of height sensors gets faulty, we can use classification scheme from Figure 1c. Now, we use 3 height and 2 colors sensors. Therefore, flexibility in designing classification procedure enables us to have fault tolerant solution of multimodal sensor system with overhead of only one sensor.

We use the reminder of the paper to explain how one can systematically design low overhead heterogeneous back-up scheme for multimodal sensor systems. The technique is generic in the sense that it is applicable to both binary and multilevel sensors systems.

2. RELATED WORK

We survey the related work along three main lines: fault-tolerance computing techniques, wireless sensor networks, and multimodal sensor fusion.

Fault tolerance as a computer related research concept has been studied for almost half a century. In the beginning, low reliability of individual components created impetus for designing reliable systems by exploiting fault tolerance and redundancy. In particular, Moore and Shannon [Moo56] and in particular von Neumann [von56] built the first systematic approaches for modeling the redundancy techniques. Since then, the reliability of individual components has increased dramatically. However, exponentially increasing levels of integration created a need for fault-tolerant systems, in particular in DRAM. The summary of early developments in reliability engineering by fault tolerance is a book by von Alven [von64]. [Sie92] provides a more recent survey on reliable computing systems.

Wireless Sensor networks have recently emerged as a premier research topic. A number of high profile applications for wireless sensor networks have been envisioned [Wei91][Ten00][Est00]. Fault tolerance in measurements by a group of sensors, was first studied by Marzullo [Mar90]. Marzullo proposed a flexible control process program that tolerates individual sensor failures. Issues addressed include modifying specifications in order to accommodate uncertainty in sensor values and averaging sensor values in a fault-tolerant way. [Jay96] developed an algorithm that guarantees reliable and fairly accurate output from a number of different types of sensors when at most k out of n sensors are faulty. The results of the scheme are applicable only to certain individual sensor faults and traditional networks. They are not generalizeable to the reliability needs in complex network levels and most importantly; they do not address the reliability issues that are induced by the ad-hoc nature of the wireless sensor networks.

Multi-sensor data fusion is a problem that recently has attracted a great deal of attention in a number of scientific and engineering communities [Bro97][Var97][Hag90]. Majority of these works 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 [Cla90].

3. PRELIMINARIES

In this Section in order to make the paper self-contained, we briefly outline key facts and assumptions about fault models, fault detection, and embedded sensor networks.

Each sensor node has five components: computation, communication, storage, sensors, and often actuators. Widely accepted fault and error models for processors, FPGA-based components, SRAM and DRAM, non-volatile memory and disks, and communication systems are readily available. However, the situation for actuators and sensors is very different. Both type of resources are conceptually more complex and intrinsically more diverse to allow for simple, yet realistic and widely applicable fault and error models.

In this work, we restrict our attention on faults in sensors. We adopt two fault models. The first one is related to sensors that produce binary outputs. In this case, obviously, one can envision a number of applicable fault models. For example, one model can capture probability or statistics of erroneous reported result. Nevertheless, it appears that the most logical and with potentially largest applicability range is the permanent fault model where only possible outcomes are that either the sensor is functional or not. For this fault model, the fault detection procedure is often straightforward: usually just observing the output of the sensors.

The second fault model is related to the sensors with continuous (analog) or multilevel digital outputs. The fault models for this type of sensors are even additionally more complex and diverse. We propose to measure the level of discrepancy of the output of individual sensor with the multimodal model used for fusion as the indication of the level of error in that sensor.

The approach has two key advantages. The first is that our fault tolerance approaches are such that the developed technique is applicable to great variety of fault models. The approach is in particularly well suited for addresses transient errors and errors in measurements. The second advantage is that the approach simultaneously addresses fault detection and correction. Overall, we made only mild assumptions: the main being that majority of sensors are functioning correctly.

Wireless Embedded Sensor Networks (WESNs) are complex distributed systems deployed in an ad-hoc manner. WESN consists of a number of sensor nodes, each with significant amount of computation, communication, storage, sensing, and often actuating resources. While traditional wireless network architecture has been based on system of static base stations, it appears that the multihop networks where each node communicates with a few close nodes is the most efficient architecture in terms of energy saving and bandwidth reuse. In multihop networks, each node communicates with other nodes that are geographically distant using intermediate nodes to build communication path.

4. SENSOR RESOURCE ASSIGNMENT AND ALLOCATION

In this section, we formulate the sensor resource allocation (SRA) problem and establish the complexity of the proposed problem.

The SENSOR RESOURCE ASSIGNMENT (SRA) PROBLEM can be formulated in the following way.

INSTANCE: Set A_1 of points p_i $(x_{i1}, ..., x_{im})$, in m-dimensional space where $1 \leq i \leq n$, a positive integer J_1, set H that consists of m(n-1) [m-1]-dimensional hyperplanes that are perpendicular to one of the m axes, such that each hyperplane is separating two points p_i and p_j that have the closest coordinates along the axis to which the hyperplane is perpendicular.

QUESTION: Find a subset of selected hyperplanes H, such that any two points p_i and p_j are separated by at least one of the selected hyperplanes and also the cardinality of H is at most J_1.

Claim: SENSOR RESOURCE ALLOCATION (SRA) is NP-complete.

The proof of the NP-completeness of the SRA problem is outlined in [Kou02].

5. FAULT TOLERANT RESOURCE ASSIGNMENT PROBLEM

In this section, we present our approach and algorithms for fault tolerant sensor assignment. Although it is easy to envision a monolithic solution that simultaneously considers fault tolerance requirements and sensor allocation and assignment problem, following principles of separation of concerns and orthogonalilty, we designed fully modular system that has separate optimization mechanisms for the subtask: sensor assignment, sensor allocation, and fault-tolerance. These three steps are addressed in the following way.

We employ two different algorithmic engines to RSA problem: ILP-based and simulated annealing based. The rationale behind the integer linear programming (ILP) approach is that although ILP solvers are often not fast, they are attractive since they guarantee optimal solution. In addition, we expect that for many smaller instances of practical importance can be solved using this approach. The points is that we have to find the solution to the SRA problem before the deployment, so it is one time expense in computational time on workstation and may be acceptable. In the cases when ILP is not applicable, we provide option of using simulated annealing as optimization mechanism.

The ILP formulation for the SRA problem can be stated in the following way.

INPUTS: set of n, m-dimensional points $p_i(x_{i1}, x_{i2}, ..., x_{im})$, $1\le i \le n$. Set of all possible tests T, with elements t_k ($1\le t_k \le m(n-1)$), where the (l(n-1)+1) to (l+1)(n-1) tests are in dimension l, $1 \le l \le m$, each separating two closest point in that dimension. The cost of each test t_k is c_k.

We define the variable X_k as followed:

$X_k=1$ if test t_k is selected

$X_k=0$ otherwise.

The objective function is to minimize the total cost of all of the selected tests. In another word:

$$\text{OF:} \quad \sum_{k=1}^{m(n-1)} x_k \cdot c_k$$

The constraint of the problem is that for each pair of points p_i and p_j, there should be at least one test that has a different outcome when applied to these two points. We define an auxiliary matrix $A[n\times k(m-1)]$ with constant elements a_{ik},

$a_{ik}=1$ if the test t_k produces 1 on point p_i

$a_{ik}=0$ otherwise.

Using the matrix A and our variables, we find a linear expression that produces zero, if a test produces similar results on the two points p_i and p_j and one otherwise. One such expression is $X_k\times(a_{ik}+a_{jk}) \times (1-a_{ik}\times a_{jk})$ that has the required property. Therefore, to have a different test result on each set of two points pi and pj, we write the following constraints.

CONSTRAINTS: For each pair of points pi and pj,

$$\sum_{k=1}^{m(n-1)} x_k \cdot (a_{ik} + a_{jk}).(1 - a_{ik}.a_{jk}) \ge 2$$

We used standard simulated annealing code. The four components of simulated annealing (moves - neighborhood structure, objective function, cooling schedule, and stopping criteria) are defined in the following way. Move is the replacement of one sensor with another sensor of the same type. The goal is to maximize objective function. We use the standard geometric cooling schedule. Finally, as stopping criteria, we use the user specified number of steps in which the improvement did not occur.

We conduct resource allocation in the following way. We first propose as the initial solution as the number of sensors that is lower bound on the potential solution. The bound is calculated assuming that all dimensions have the same number of sensors and each n-dimensional compartment will eventually contain one point. After that, we run the simulated annealing RSA algorithm. During this running process, we modify the move so that one type of sensor can be replaced with another type of sensor. We accumulate statistics about which type of sensor helps the most to improve objective function after each move and use this information to decide which type of sensor to add or remove.

For fault-tolerance, one can envision three different mechanisms. The first is to specify in the ILP formulation or in the simulated annealing code that each two points have to be separated by at least r hyperplanes. Since this approach essentially doubles the redundancy, we did not accept this alternative. The second alternative is to add exactly one extra sensor of each type to the solution generated by the sensor resource allocation problem. When large number of sensor nodes of each type is used, the overhead is relatively low. Also, in this case the need for storing or communicating more than one resource assignment solution is eliminated. Therefore, if moderate levels of fault tolerance are needed, this can be an attractive alternative.

The final and most attractive alternative in terms of overhead is to leverage on heterogeneous back up of sensors of different modality. Here we generate allocation in the following way. We first calculate for each type of sensors for all allocations k from 1 to smaller than the number allocated in the best resource allocation solutions, the cost of overall solution. After that we plot the cost of all these solutions on y-axis on the graph where the x-axis is the number of allocated sensors of analyzed type. In such a way we obtain m graphs, where m is the number of sensors of different modality. Obviously, now we have to use the RSA algorithm to analyze only allocations that are worse in terms of cost than the optimal solution and better than the solution from the second alternative. We conduct this analysis in the order dictated by increasing cost of the proposed solution.

6. FAULT TOLERANCE IN MULTIMODAL SENSORY SYSTEMS

There are a number of ways to generalize and therefore enhance the applicability of the technique presented in the previous sections. One possibility is to characterize objects using statistical data and to build statistical model for decision making using data from sensors. Another, equally important and with equally large application domain option is to conduct multimodal sensor fusion in order to support decision process. As a matter of fact, the multimodal multilevel sensor fusion has emerged as one of canonical prob-

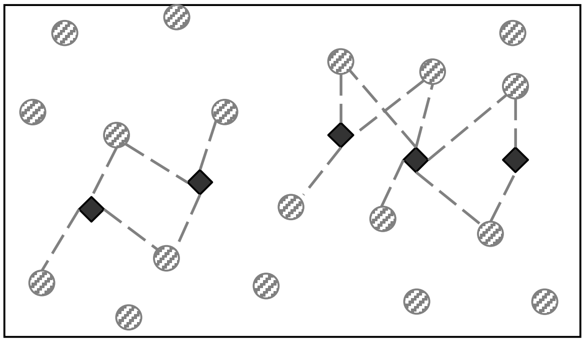

Figure 2 - tracking of an object by the sensors

lems in sensor networks. Informally it can be defined in the following way. A number of sensors, some of them with different modalities, are given. The goal is to extract as accurately as possible the information requested by a user from noisy measurements.

Although it the problem seems too general to be efficiently solved using a single approach, there is a systematic way to address the problem. What is needed is to develop or even better to find some already developed analytic models that are related to the measured quantities. Once the equations of an analytical model are assembled, the intriguing and important question is to try to figure out which measurements are faulty or have high degree of noise. One way to answer this question is to try to find a subset of measurements that produce consistent set of analytic models. Using this set of equation, we can calculate the value for all quantities of interest. Therefore, the key for providing fault tolerant multimodal sensor fusion is to generate rich enough model of physical world and in particular to ensure that the system is solvable even when some of the equations are not used. The main difficulty is that the systems of equations are often nonlinear and therefore it is very difficult to say in advance when the system is well defined in a sense that it can be uniquely solved.

Probably the best way to clarify the introduced approach is to take a closer look at an example. For this purpose we will use scenario illustrated at Figure 2. We see an object O that moves along its trajectory that includes points pi in embedded sensor network that consists of a number of nodes each represented by a shaded circle n_i. We have four types of sensors: RSSI-based distance discovery, speedometer, accelerometer, and compass that are used to measure the angle in 2D physical space. Three RSSI-based measurements can be used to locate the object O in any particular moment. Euclidian space, Newton mechanics, and trigonometry laws can be used to establish relationships between measurements. Specifically, Eq. 1-9 are trilateration equations, Eq. 10-13 are the Newton law equation and the Eq 14-15 are trigonometry laws. The key observation is that we have more equations (15) than variables (12) that may have errors. So, if one of sensor is not functioning, we can calculate it from the established system of equations. Also, for each variable, we can find how much it has to be altered in order to make the whole system of equations maximally consistent. The variables that have to be altered the most are most likely measured by faulty sensors. Therefore, one way to identify and correct sensor measurements is to try all scenarios where exactly one type of sensor measurements is not taken into account and compare the maximal error in the system. Another very important observation is that by sampling all operational sensors more often, we can compensate for faulty sensors.

$$(x_1 - s_1)^2 + (y_1 - t_1)^2 = R_1^2 \tag*{Eq 1}$$

$$(x_1 - s_2)^2 + (y_1 - t_2)^2 = R_2^2 \tag*{Eq 2}$$

$$(x_1 - s_3)^2 + (y_1 - t_3)^2 = R_3^2 \tag*{Eq 3}$$

$$(x_2 - s_4)^2 + (y_2 - t_4)^2 = R_4^2 \tag*{Eq 4}$$

$$(x_2 - s_5)^2 + (y_2 - t_5)^2 = R_5^2 \tag*{Eq 5}$$

$$(x_2 - s_6)^2 + (y_2 - t_6)^2 = R_6^2 \tag*{Eq 6}$$

$$(x_3 - s_7)^2 + (y_3 - t_7)^2 = R_7^2 \tag*{Eq 7}$$

$$(x_3 - s_8)^2 + (y_3 - t_8)^2 = R_8^2 \tag*{Eq 8}$$

$$(x_3 - s_9)^2 + (y_3 - t_9)^2 = R_9^2 \tag*{Eq 9}$$

$$\sqrt{(x_1 - x_2)^2 - (y_1 - y_2)^2} = {}^1\!/_2 \cdot a_1(\Delta t)^2 + v_1(\Delta t) \tag*{Eq 10}$$

$$\sqrt{(x_3 - x_2)^2 - (y_3 - y_2)^2} = {}^1\!/_2 \cdot a_2(\Delta t)^2 + v_2(\Delta t) \tag*{Eq 11}$$

$$a_1.\Delta t = v_1 - v_0 \tag*{Eq 12}$$

$$a_2.\Delta t = v_2 - v_1 \tag*{Eq 13}$$

$$\alpha_1 = \tan^{-1}\left({y_2 - y_1} \middle/ {x_2 - x_1} \right) \tag*{Eq 14}$$

$$\alpha_2 = \tan^{-1}\left({y_3 - y_2} \middle/ {x_3 - x_2} \right) \tag*{Eq 15}$$

7. EXPERIMENTAL RESULTS

There are three main reasons why the evaluation of the new approach and algorithms is a challenging task. The first is that the fault-tolerant sensor assignment is a NP-complete problem and therefore in general, one does not find the optimal solution for a given instance. The second reason is that there are no established benchmarks and previously published results for the addressed problem. The final complication is related to layered structure of the problem: one has to evaluate sensor resource allocation and fault tolerance. Nevertheless, it is still possible to evaluate the proposed algorithms in sound and convincing way.

We split the evaluation process in two subtasks: one where we evaluate the algorithms for sensor assignment and allocation and one where we evaluate the fault-tolerance approach. The key idea is to generate challenging instance for which the optimal solution is known. That, for example, can be accomplished in the following way. For the sake of simplicity, we assume that the cost of all sensors in all m-dimensions is equal. We first construct a solution. The solution consists of the equal number of sensors in each direction. Next, we place exactly one object in each of the m-dimensional hypercube defined by the selected sensors. Each object is placed in random location within the hypercube. It is easy to see that the selected sensors are optimal. We can additionally obscure solution by not placing objects in a small number of the hypercubes or by not using exactly the same number of sensors in each dimension. Furthermore, we can combine two or more instances of just created problems that to have disjointed ranges for objects in all dimensions to create new instances. Finally, note that we can also combine smaller arbitrary instances solved by our ILP-approach to create large new instances of the problem with known solution.

The evaluation of the simulated annealing-based algorithm is shown in Table 1. The first two columns indicate the number of objects and the number of dimensions. The next three columns indicate the size of solution generated by the simulated annealing program in 2 minutes on 1 GH Pentium processors. The final column indicates the size of the optimal solution. Each simulated annealing is run 10 times each time on different instance of the problem with the same characteristics.

Number of points	Dimension	SA-solution			Optimal
		worst	median	best	
100	2	22	20	19	18
100	3	15	13	12	12
200	2	33	30	28	28
300	3	25	20	18	18
500	4	23	19	16	16
800	4	26	22	19	18
1000	5	25	20	17	15

Table 1 – experimental results for the Simulated Annealing (SA)-based algorithm

For the evaluation of the fault tolerant approach, we used randomly generated instances solved using the simulated annealing approach. Table 2 summarizes our results. The first two columns indicate the number of objects and the number of dimensions. The last column shows the number of additional sensors used by the fault tolerance scheme on 20 instances of each type. We see that significant cost savings can be accomplished using the proposed approach.

Number of points	Dimension	Average overhead
200	4	3.0
300	5	3.8
500	5	4.0
800	6	4.6

Table 2- evaluation of the Fault tolerance using the SA-based approach

8. CONCLUSION

We have developed a new approach to design low overhead fault-tolerant sensor networks. The key idea is to use one type of sensor to back-up sensors of different types by exploiting flexibility during multimodal sensor data fusion. We formulated the problem for two different types of sensors, established computational complexity of associated problems, and have developed algorithms to solve them. Finally, we have demonstrated the effectiveness of our approach and algorithms of a set specially designed problems for which the optimal solution is known.

9. REFERENCES

[1] [Bro97] R. R. Brooks and S. S. Iyengar, "Multi-Sensor Fusion: Fundamentals and Applications With Software," Prentice-Hall, 1997.

[2] [Cla90] J. J. Clark and A. L. Yuille, Data Fusion for Sensory Information Processing Systems, Kluwer, 1990

[3] [Est00] D. Estrin, R. Govindan, and J. Heidemann. "Embedding the Internet: Introduction," Communications of the ACM, vol. 43, no. 5, pp. 38-42, 2000.

[4] [Hag90] G.D. Hager, "Task-Directed Sensor Fusion and Planning - A Computational Approach," Kluwer Academic Publishers, 1990.

[5] [Jay96] D.N. Jayasimha, "Fault tolerance in multi-sensor networks," IEEE Transactions on Reliability, vol. 45, no.2, pp. 308-15, June 1996.

[6] [Kou02] F. Koushanfar, M. Potkonjak, A. Sangiovanni-Vincentelli. "Fault tolerance techniques in wireless ad-hoc sensor networks", UC Berkeley technical reports, 2002.

[7] [Mar90] K. Marzullo, "Tolerating failures of continuous-valued sensors," ACM Transactions on Computer Systems, vol. 8, no.4, pp. 284-304, November 1990.

[8] [Moo56] E.F. Moore, C.E. Shannon, "Reliable Circuits Using Less Reliable Relays I," Journal Franklin Institute, vol. 262, pp. 191-208, September 1956.

[9] [Sie92] D.P. Siewiorek, R.S. Swartz. "Reliable Computer Systems: Design and Evaluation," 2nd edition, Digital Press, Burlington, MA, 1992.

[10] [Ten00] D. Tennenhouse, "Proactive Computing," Communications of the ACM, vol. 43, no. 5, pp. 43-50, 2000.

[11] [Var97] P. K. Varshney, "Distributed Detection and Data Fusion," N.Y.: Springer-Verlag, 1997.

[12] [von56] J. Von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," In C. E. Shannon and J. McCarthy, Eds. Automata Studies. Princeton: Princeton University Press, pp. 43-98, 1956.

[13] [von64] W. H. von Alven (edited book), "Reliability Engineering," Englewood Cliffs, NJ: Prentice-Hall, 1964.

[14] [Wei91] M. Weiser, "The Computer of the Twenty-First Century," Scientific American, pp. 94-100, September 1991.