Appeared in FUSION 2001 conference

Value-Fusion versus Decision-Fusion for Fault-tolerance in Collaborative Target Detection in Sensor Networks

Thomas Clouqueur, Parameswaran Ramanathan, Kewal K. Saluja , Kuang-Ching Wang
Department of Electrical and
Computer Engineering
University of Wisconsin-Madison
Madison, WI 53706, USA
saluja@ece.wisc.edu

Abstract – *Collaborative signal processing algorithms in sensor networks must be robust to device failures because one expects a large number of failures due to the harsh conditions in which they are usually deployed. In this paper, we study two distinct approaches, value-fusion and decision-fusion, for achieving fault-tolerance in collaborative target detection algorithms. In value-fusion, sensor devices first exchange their measured values to arrive at a fault-tolerant consensus on the measurement. Then each device makes an independent decision as to whether or not a target is present based on the consensus measurement. In contrast, in decision-fusion, each device first makes an independent decision as to whether or not a target is present and then the devices exchange their decisions to arrive at a fault-tolerant consensus decision. In this paper, we compare the performance of value and decision fusion using two measures: probability of correct detection and probability of false alarm. The results show that if fault-tolerance is not required, then value-fusion is better than decision-fusion and whereas if fault-tolerance is essential, then decision-fusion is better than value-fusion.*

Keywords: Collaborative signal processing, data fusion, decision, fault tolerance, sensor network.

1 Introduction

Wireless devices are becoming an integral part of electronic systems in everyday use. Today's wireless devices are not just the cell phones, but can be intelligent and smart devices with sensing, processing and communication capabilities. Such devices may consist of communication modules using wireless technologies such as bluetooth [2] or IEEE 802.11 standard [5]. They may also contain special purpose sensors and processors as in the case of SensIT program [8]. These small and smart devices termed as *sensors or microsensors*, can form networks and collectively perform tasks that no single device may perform by itself. Examples of such tasks include detection, classification, and tracking of an object in a region. Such tasks may use all three, namely sensing, processing, and communication capabilities of these micro sensors. When performing a target detection task, multiple sensors in a region detect the presence of an object using sound, motion, or heat associated with the object of interest. The sensors may exchange information about the presence or absence of the object or energy level associated with the object and after performing collective signal processing, may reliably determine the nature of the object and arrive at a common conclusion. Below we describe a wireless sensor network architecture that is being developed in the SensIT program to support these objectives.

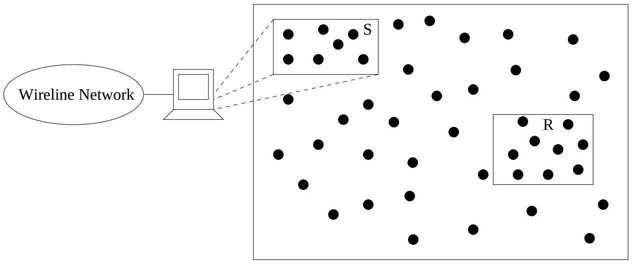

Figure 1: Wireless sensor network

The basic micro sensor network system is shown in Figure 1. In this figure, each black dot represents an inexpensive microsensor with positioning, multiple sensing, processing, and communication capabilities. Consider a task assigned to the sensors in region R to collectively make a decision about the presence of an object in this region. The sensors using their varied sensing capabilities can make a collective decision using one the following two alternatives. 1) Each sensor may make its independent decision using its own measured values and then sensors may exchange their decisions among each other to arrive at a consensus by fusing all decisions. 2) All sensors exchange their measured values and then each sensor makes its own individual and independent decision by fusing the collected values. We call the first method *decision fusion* and the second method *value fusion*.

Comparative study of these two methods of fusing information is the focus of this paper. In particular, we compare these two methods for their accuracy under the following two conditions: 1) All sensors are assumed to be fault free and 2) some of the sensors may be faulty. Note that in the second case, faulty sensors may provide incorrect values or decisions to the other sensors in the sensor network. We must add that the two approaches can also be compared using other parameters such as power consumption, the number of message exchanges, or the communication bandwidth required to arrive at consensus. However, these are not the subject of this paper. Some of these parameters have been studied in literature [1, 9].

This paper is organized as follows. In section 2, we present the system model and introduce the metrics used to assess the collaborative detection algorithms. Section 3 describes the algorithms for value and decision fusion for fault free and faulty systems. Section 4 introduces the simulator used to evaluate performance of the algorithms and section 5 presents the simulation results. The paper concludes with section 6.

2 System model and problem formulation

In this section we introduce the model of sensor network used for target detection as well as the fault model and metrics used to evaluate the system performance.

2.1 Network model

As mentioned in section 1, we assume that a set of sensors deployed in a region R, shown in Figure 1, is to determine if a specified target is present or not in the region. To detect the target, each sensor can measure an energy level that is a function of its distance to the target and the background noise. We assume the noise to be Gaussian with zero mean and independent at different sensors. We also assume that sensors can communicate with each other. The goal of the detection algorithms is to estimate if a target is present or not in a region. This requires collaboration among the sensors deployed in the region since sensors have only a limited perception of the complete region. For example, if the target lies in the corner of the region or in a neighboring region, it may be detected by a small subset of all sensors in the region. However, the final decision for the region needs to be “present” in the first case and “absent” in the second case. The all set of sensors can come to this decision only by fusing their information into a global description of the region.

The two approaches proposed to solve this problem are value fusion and decision fusion. These approaches are illustrated in Figure 2. In value fusion, sensors communicate their energy measurement values to each other and decide using the set S of all values whether a target is present in the region. In decision fusion, each sensor makes a decision first by using its own energy measurement value, the sensors then communicate decisions to each other and finally decide using the set of decisions.

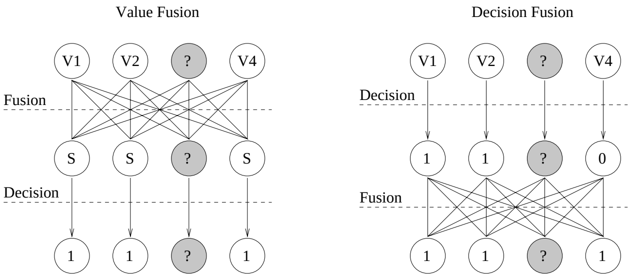

Figure 2: Two approaches for fusion. The grey sensors are faulty and have indeterminate behavior.

When collaborating, the decision made by the sensors can be corrupted by faulty sensors present in the region. We describe in the next subsection the type of faulty behavior assumed in this study and a solution that can be adopted to tolerate such behavior.

2.2 Fault model

The network considered is likely to contain some faulty sensors due to harsh environmental conditions. The behavior of faulty sensors is assumed to be arbitrary or malicious, e.g. they can send incorrect information and can even be inconsistent when sending information to different sensors as shown in Figure 3.

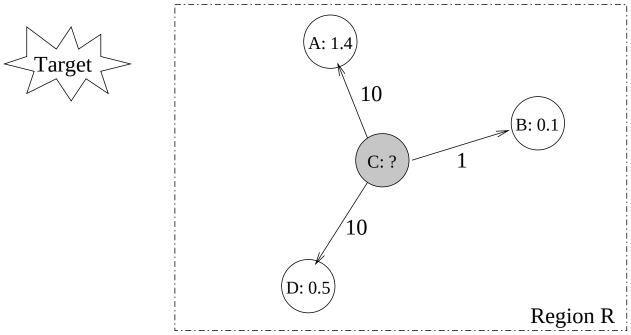

Figure 3: Byzantine faulty behavior

Four sensors (A, B, C and D) are deployed in the region as a target object is present in the neighboring region. Sensor A measures an energy level of 1.4 (including noise) whereas sensor B and D measure an energy level of 0.5 and 0.1 respectively. Sensor C is assumed to be faulty and sends different measurements to the other sensors (10, 1 and 10 to A, B and D respectively). As a result, non faulty sensors obtain different pictures of the region and may conclude differently on the presence of the target (e.g. sensor A and D may conclude that a target is present while sensor B concludes that no target is present). This faulty behavior is referred as Byzantine [6]. In the presence of such faults, agreement needs to be performed for all the non faulty sensors to arrive at the same final decision. Numerous studies have been conducted on agreement and it is proven that to reach agreement in the presence of m Byzantine faulty sen-

	A	B	C	D
Value	1.4	0.1	?	0.5
Value received from C	10.0	1.0	-	10.0
Set S after exact agreement	$\left\{\begin{array}{c}1.4\\0.1\\10.0\\0.5\end{array}\right\}$	$\left\{\begin{array}{c}1.4\\0.1\\10.0\\0.5\end{array}\right\}$	?	$\left\{\begin{array}{c}1.4\\0.1\\10.0\\0.5\end{array}\right\}$
Final decision	1	1	?	1

Table 1: Example of exact agreement on values, sensor C being faulty

sors, the network must contain $N \geq 3m + 1$ sensors [6]. In this paper, we use the exact agreement algorithm developed by Lamport et al. in [6]. This algorithm guarantees that when exchanging values, all the non faulty sensors obtain the same set of values and all the values sent by non faulty sensors are part of this set. Inconsistent values sent by faulty sensors are replaced by a majority vote or a default value. An example of exact agreement performed on the four sensors of Figure 3 is presented in Table 1. After exact agreement is performed, inconsistent values sent by sensor C are replaced by a common value (i.e. 10.0). Note that in this example, the final decisions of the non faulty sensors are incorrect but they are consistent.

2.3 Performance metrics

The performance of the algorithms can be measured in terms of precision and accuracy [3, 6]. As shown in the previous subsections, sensors need to fuse their values to make a decision representative of the complete region and faults can lead to inconsistent fused values obtained at different sensors. Precision measures the closeness of decisions from each other, the goal being that all non faulty sensors make the same decision. Accuracy measures how well sensor values represent the environment, the goal being that the decision of non faulty sensors is “0” whenever the target is absent and “1” whenever it is present. Note that we have no control on the decision made by faulty sensors. The algorithms developed for fusion in the presence of faults use exact agreement to solve the inconsistency problem. Therefore, all the sensors obtain the same set of values or local decisions and make the same final decisions and both approaches for fusion have perfect precision. On the other hand, the accuracy is not perfect and is measured by the false alarm probability and the detection probability as defined below.

> The *false alarm probability* is the conditional probability that the sensors report the presence of an object given that there is no object in the region.

> The *detection probability* is the conditional probability that the sensors report the presence of an object given that there is an object in the region.

3 Algorithms

In this section, we describe the target detection algorithms used for value fusion and decision fusion in the absence and in the presence of faults, respectively.

3.1 Value fusion algorithms

In value fusion, the sensors in the network exchange their local energy values and fuse them by finding the average. The final detection decision is made by comparing this final value to a threshold η_v. The algorithm for value fusion in the absence of faults (Alg VF_NFS) is described below.

```
// value fusion - no faulty sensors (VF_NFS)
at each node{
  compute energy;
  exchange values;
  compute average of values;
  compare average to threshold $\eta_v$ for final decision;
}
```

The energy measurements of all sensors contain independent Gaussian noise with zero mean and variance σ^2. The average of such noise over N sensors is a Gaussian zero mean noise with variance σ^2/N [4]. Therefore, provided that N is large enough, the fused value has a low noise and simple comparison to a threshold gives an accurate decision.

In the presence of faulty sensors, extra steps must be added to the fusion algorithm to achieve precision and accuracy. As mentioned in the previous section, we use exact agreement to achieve precision in the system. Exact agreement guarantees that all the non faulty sensors obtain the same set S of values and the values sent by the non faulty sensors are part of this set. However, consistent outlying values can remain in the set, as shown in Table 1. To prevent corruption of the decision by these outliers, the largest m and smallest m values are dropped from the set S and the average value is computed over the remaining $N - 2m$ values. The algorithm for value fusion in the presence of faults (Alg VF_FS) is described below.

```
// value fusion - faulty sensors (VF_FS)
at each node{
  compute energy;
  exchange values with exact agreement;
  drop largest $m$ and smallest $m$ values;
  compute average of remaining values;
  compare average to threshold $\eta_v$ for final decision;
}
```

Since the fused value is the average over $N - 2m$ values, lower accuracy is expected from the fault tolerant value fusion than the non fault tolerant value fusion. Furthermore, many meaningful values may get dropped (e.g. the high energies measured by the sensors closest to the target). Therefore, more sensors need to detect the target for the system to make an accurate decision. Thus, the SNR must be higher to obtain similar performance with faults as without faults.

3.2 Decision fusion algorithms

In decision fusion, the sensors in the network make a local decision on the presence of the target by comparing their own energy measurement to a threshold η_d. Then they exchange their local decision and fuse them by averaging. The final detection decision is made by comparing this fused decision to a threshold α. The algorithm for value fusion in the absence of faults (Alg DF_NFS) is described below.

```
// decision fusion - no faulty sensors (DF_NFS)
at each node{
    compute energy;
    compare to $\eta_d$ to arrive at a local decision;
    exchange decisions;
    compute average of local decisions;
    compare average to $\alpha$ for final decision;
}
```

As in value fusion, the fused data is obtained by averaging data received from all the sensors. Different values of α lead to different performance in term of detection probability for constant false alarm probability. Through simulation, we found that the best performance were achieved when $\alpha = 1/N$. That means that the final decision is "detect" as soon as one of the N sensors reports a detection.

In the presence of faults, exact agreement is used to achieve precision in the system. However, as opposed to value fusion, no data is dropped since the corruption capability of the faulty sensors is limited to sending wrong binary decisions. The algorithm for decision fusion in the presence of faults (Alg DF_FS) is described below.

```
// decision fusion - faulty sensors (DF_FS)
at each node{
    compute energy;
    compare to $\eta_d$ to arrive at a local decision;
    exchange local decisions with exact agreement;
    compute average of local decisions;
    compare average to $\alpha$ for final decision;
}
```

The value of the second threshold α needs to be increased compared to the non faulty case. Indeed, as m sensors out of N can be faulty, the final decision cannot rely on fewer than m sensors and α must be between m/N and $(N-m)/N$. For example, if $\alpha \leq m/N$, the incorrect decisions sent by m faulty sensors in the absence of a target result into a "detect" final decision, whatever the decision of non faulty sensors are, and therefore the false alarm probability is one, which is undesirable. We found that best performance was obtained for $\alpha = .43$.

4 Simulator design

We used simulation to compare value and decision fusion under various environmental conditions such as varying number of sensors, varying SNR, and varying number of faulty sensors. The sensors are assumed to be evenly distributed over a region of size 4×4, the distance unit being left undefined, and the energy measured by sensor i is a function of its distance to the target object d_i, as defined by the following equation:

$$E(d_i) = \frac{K}{(1+d_i)^3} \qquad (1)$$

where K is the maximum energy at the target object. For our energy model to be valid for very small d_i, we used the term $(1 + d_i)$ in the denominator. Note that the constant 1 is relative to the distance unit and for large distances d_i, $1 + d_i \simeq d_i$ and the energy model becomes similar to the standard energy models for signal transmission [7]. The zero mean Gaussian noise is generated with variance $\sigma^2 = 1$ and the SNR is defined as the peak SNR at the target object.

$$SNR = 10\log_{10}(\frac{K}{\sigma^2}) = 10\log_{10}(K) \quad (dB) \qquad (2)$$

The Byzantine faulty behavior is generated as follows. In the absence of target, faulty nodes report a high value and in the presence of target in the region, they all report a low value. To compare value fusion and decision fusion, we measured the detection probability for constant false alarm rates. This requires first to find the adequate thresholds to obtain a given false alarm probability and then use these thresholds to measure the detection probability. Finally, the simulation results are averaged over a large number of iterations to obtain 80% confidence that the results are within 10% of the mean values.

5 Comparative performance

We now present simulation results for the detection algorithms proposed without and with faults in the system.

5.1 Without faulty sensors

Figures 4 and 5 show the relative performance of the two algorithms for 4, 25 and 49 sensors and varying SNR between 5dB and 19dB. The graph of Figure 4 is derived for a false alarm probability of 3% and the graph of Figure 5 for a false alarm probability of 8%. We notice that detection probabilities increase as the SNR increases and detection probabilities are higher when the false alarm probability is

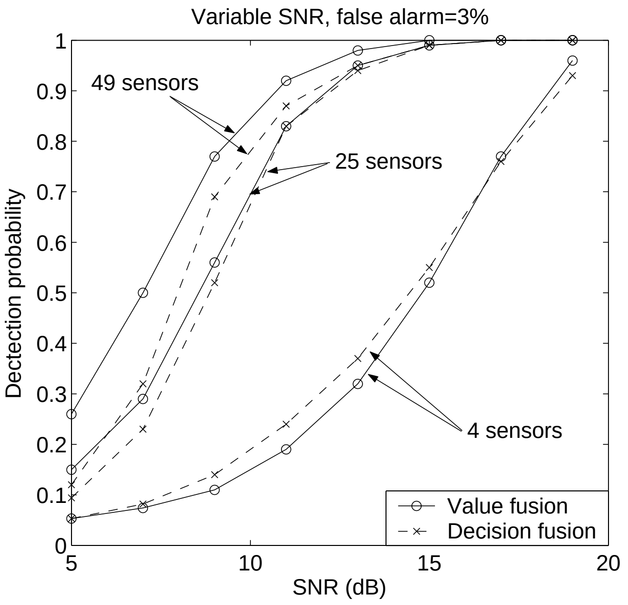

Figure 4: Non Fault Tolerant Algorithms (a)

8% as opposed to when it is 3%. As far as comparing the two methods of fusion, the graphs show that decision fusion and value fusion perform almost equally when there are only four sensors in the region of interest. However, value fusion is substantially better when sensor density is medium, 25 sensors, to high, 49 sensors.

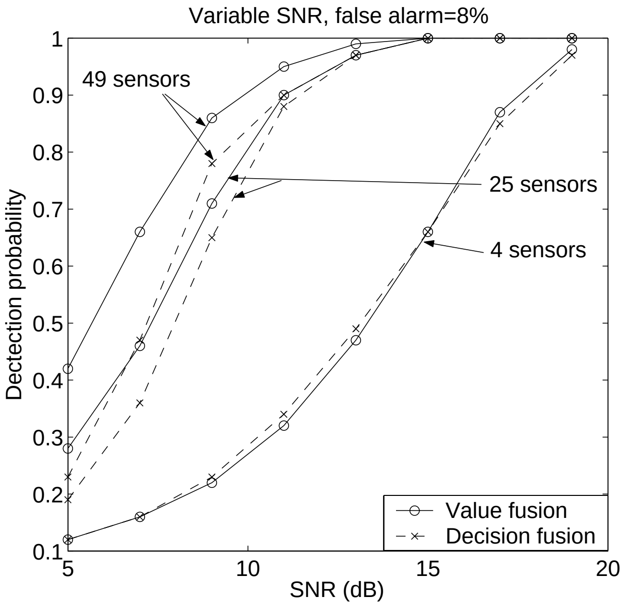

Figure 5: Non Fault Tolerant Algorithms (b)

5.2 With faulty sensors

Two parameters specify the simulation system in the presence of faults: m, the number of faulty sensors that the algorithm can tolerate; and t, the number of faulty sensors actually present in the system. In all the simulation runs, $0 \leq t \leq m \leq \lfloor \frac{N-1}{3} \rfloor$. We first evaluate the performance of value and decision fusion when $m = t = \lfloor \frac{N-1}{3} \rfloor$. The performance for 4, 25 and 49 sensors as a function of the SNR is presented in the graphs of Figure 6. The thresholds are set to obtain a constant false alarm probability of 3% and the graphs show that decision fusion performs consistently better with 4, 25 or 49 sensors in the system. The same behavior was also observed when comparing the two methods with a higher false alarm probability of 8%, but the graphs are not shown here.

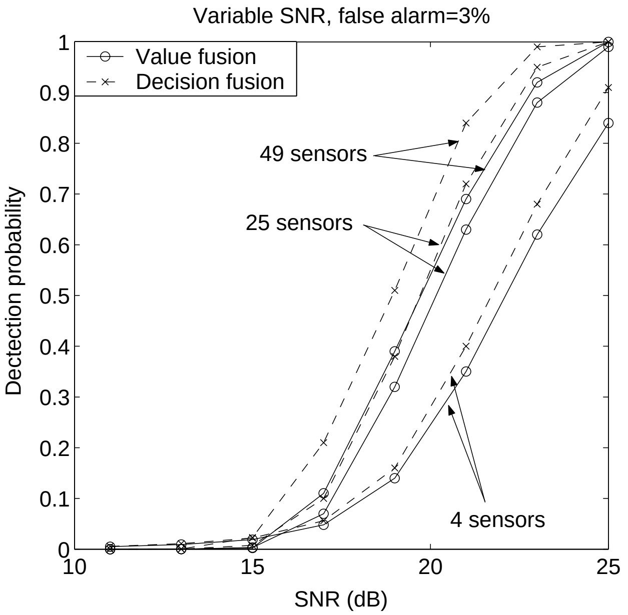

Figure 6: Fault Tolerant Algorithms with $m = t = \lfloor \frac{N-1}{3} \rfloor$

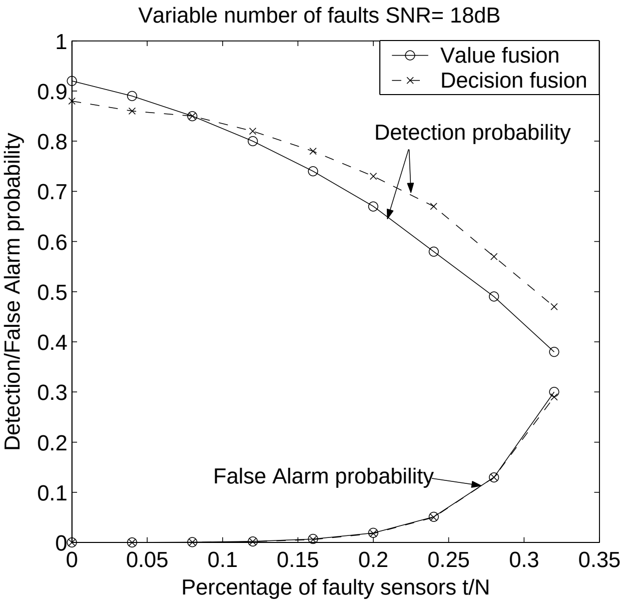

Figure 7: Fault Tolerant Algorithms with $m = \lfloor \frac{N-1}{3} \rfloor$ and variable t

Our second simulation measures the effect of the number of faulty sensors present on the performance when trying to tolerate as many faulty sensors as possible. Thus,

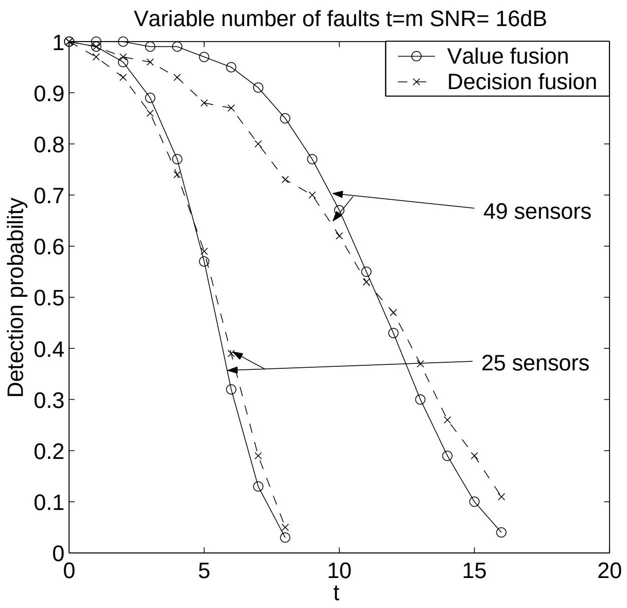

Figure 8: Fault Tolerant Algorithms with variable $m = t$

we set $m = \lfloor \frac{N-1}{3} \rfloor$ and varied t from 0 to m, working with 25 sensors and a SNR of 18dB. The thresholds are set to obtain a false alarm probability of 5% when $t = 6$ and we measure both the detection probability and false alarm probability for varying t, as reported in the graph of Figure 7. We notice that the false alarm probabilities of value and decision fusion stay almost equal for varying t. When comparing detection probabilities, we find again that decision fusion performs better than value fusion for $t = m$ but the conclusion reverses when t/N is below 8%. Value fusion is superior to decision fusion in the general case when no or only a few sensors are faulty. However, the number of faulty sensors is usually not know a priori and decision fusion offers a graceful degradation when t increases.

Our last simulation evaluates which fusion method is superior if one knows a priori how many faulty sensors are present in the system. Indeed, we showed in section 5.1 that when $t = 0$ value fusion is superior to decision fusion (provided that the sensor density is not too low). And we showed in the previous paragraphs that when $t = \lfloor \frac{N-1}{3} \rfloor$ decision fusion is superior to value fusion. If t is known a priori, the best performance is obtained by setting $t = m$ so as to drop as few values as possible. Thus we set $m = t$ and varied t from 0 to $\lfloor \frac{N-1}{3} \rfloor$ for 25 and 49 sensors in the system and a SNR of 16dB. The thresholds are set to obtain a constant false alarm probability of 5%. The results are presented in the graph of Figure 8. As observed in previous simulations, value fusion is superior to decision when $m = t = 0$ and the conclusion is reversed when $m = t = \lfloor \frac{N-1}{3} \rfloor$. The two approaches are equivalent for $t/N \simeq 20\%$ and the gap between value and decision fusion is larger when sensor density is larger.

6 Conclusion

In this paper, we studied the problem of collaborative target detection in a sensor network without and with faulty sensors. We introduced and compared two methods for fault tolerant data fusion, namely value fusion and decision fusion. Comparison of the two methods is presented under various environmental conditions and we study in particular the effect of the number of faulty sensors in the system on the fusion performance.

Our simulation results show that value fusion is superior to decision fusion when the sensor network is highly reliable and fault free. However, as faulty sensors are introduced in the system, the performance of value fusion degrade faster than the performance of decision fusion and decision fusion becomes superior to value fusion.

Acknowledgments

The work reported here is sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Material Command, USAF, under agreement number F30602-00-2-0555. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), the Air Force Research Laboratory, or the U.S. Government.

References

[1] R. Blum, S. Kassam, and H.V.Poor, “Distributed Detection with Multiple Sensors: Part II-Advanced Topics,” in *Proc. of the IEEE*, pp. 64–79, January 1997.

[2] J. Bray and C. F. Sturman, *Bluetooth: Connect Without Cables*, Prentice Hall PTR, 2000.

[3] R. Brooks and S. Iyengar, “Robust Distributed Computing and Sensing Algorithms,” *IEEE Computer*, vol. 29, no. 6, pp. 53–60, June 1996.

[4] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, Oxford Science Publications, 1992.

[5] IEEE Std 802.11-1997, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1997.

[6] L. Lamport, R. Shostak, and M. Pease, “The Byzantine Generals Problem,” *ACM Transaction on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, July 1982.

[7] T. Rappaport, *Wireless communications*, Prentice Hall PTR, 1996.

[8] Sensor Information Technology Website, *http://www.darpa.mil/ito/research/sensit/index.html.*

[9] R. Viswanathan and P. Varshney, “Distributed Detection with Multiple Sensors: Part I-Fundamentals,” in *Proc. of the IEEE*, pp. 54–63, January 1997.