Masking Failures of Multidimensional Sensors (extended abstract)

Paul Chew*
Keith Marzullo†
Cornell University
Department of Computer Science
Ithaca, New York 14853

December 6, 1990

Abstract

When a computer monitors a physical process, the computer uses *sensors* to determine the values of the physical variables that represent the state of the process. A sensor can sometimes fail, however, and in the worst case report a value completely unrelated to the true physical value. The work described in this paper is motivated by a methodology for transforming a process control program that cannot tolerate sensor failure into one that can. In this methodology, a reliable *abstract sensor* is created by combining information from several real sensors that measure the same physical value. To be useful, an abstract sensor must deliver reasonably accurate information at reasonable computational cost.

In this paper, we consider sensors that deliver multidimensional values (e.g., location or velocity in 3 dimensions, or both temperature and pressure). Geometric techniques are used to derive upper bounds on abstract sensor accuracy and to develop efficient algorithms for implementing abstract sensors

1 Introduction

One of the oldest techniques in fault-tolerance is using replication to mask failures [Sho68]. For example, TMR, the *triple module redundancy* scheme, masks the failure of a signal by feeding three independently computed copies of the signal into a majority voter [vN56]. TMR can be easily

*This work was supported by The Advanced Research Projects Agency of the Department of Defense under Office of Naval Research Contract N00014-88-K-0591, and by ONR Grant N00014-89-J-1946 and NSF Grant IRI-9006137.

†This work was supported by the Defense Advanced Research Projects Agency (DoD) under NASA Ames grant number NAG 2-593, Contract N00140-87-C-8904. The views, opinions, and findings contained in this report are those of the authors and should not be construed as an official Department of Defense position, policy, or decision.

extended to NMR, or *n-module redundancy*, whereby n independent copies are fed into a majority voter. With NMR, up to $f = \left\lfloor \frac{n-1}{2} \right\rfloor$ signal failures can be masked.

As stated here, NMR assumes a very weak failure model, making it a highly applicable technique. One doesn't, for example, need to know the nature of the faults, the frequency of faults, or the distribution of faulty signal values in order to design a system that uses NMR. The only time such properties are considered is when appropriate values of f and n are computed. This same weak failure model has been applied to several problems in distributed systems; for example, consensus [NT88] and reliable broadcast [CAS86], and has also been incorporated into a methodology for building fault-tolerant distributed programs [Sch90,Lam84].

One of us (Marzullo) has been working on the problem of writing provably correct programs that monitor and control physical processes. The state of a physical process is usually represented by a set of values for a corresponding set of continuous physical variables, such as the temperature or pressure of a reaction vessel. Physical values are usually measured by accessing sensors, such as thermometers or pressure gauges. A sensor, however, has a limited accuracy which gives some uncertainty in the value of the physical variable it senses, and the real-time nature of physical processes combined with uncertain execution times can increase the uncertainty in the measured value of the physical variable. If this uncertainty is too large or if the underlying sensor is faulty, then the measurement will be useless to a control program.

One can model the value of a sensor as as a random variable and then convolve the values of different sensors that measure the same physical variable. Doing so will improve the accuracy of the measured value, but it will also introduce a failure model that is expressed in terms of a (possibly unknown) probability distribution. Instead, in [Mar90] we have represented the value of a physical variable as a contiguous interval and applied the same weak failure model of assuming no more than f out of n sensors are incorrect. We have derived tight bounds on the accuracy of the resulting measured physical values and have presented efficient algorithms ($O(n \log n)$) for masking the faults of such sensors. The bounds for this problem are derived by considering *interval*

graphs [Gol80].

One limitation of the work in [Mar90] is that it is applicable only to sensors that measure a single, independent, real value. An example of a sensor that does not fit this model is one that measures the location of some physical object in 3D space. If such multidimensional sensors are used then a naive approach to masking failures is to consider the x component separately from failures of the y and z components, but doing so limits the accuracy of the resulting value. For example, any sensor found to be faulty by examining the x components should most likely be discarded when considering the y and z components. This paper extends [Mar90] by considering such multidimensional sensors.

We assume that real sensors have the following properties. Let s_i be a sensor of some physical variable $\overline{v}$. A measurement s_i is a continuous set of values that conform to some shape, such as a continuous interval, a rectangle, a sphere, etc. We say that s_i is *correct* if it is not too inaccurate and always includes the value of the actual physical variable. More precisely, for some upper bound acc on the accuracy of s_i,

$$s_i \text{ correct} \stackrel{\text{def}}{=} \overline{v} \in s_i \ \wedge \ |s_i| \leq acc$$

Thus, a real sensor can fail in two ways: it can fail to contain the true value or it can report a region so large as to be useless. For the purposes of this paper, we assume such large-region sensors can be detected and discarded by preprocessing the real sensor data (n and f will have to be adjusted). Thus for the remainder of this paper, we can assume without loss of generality that all sensors are accurate (report regions of reasonable size) and that a sensor can be incorrect only by failing to contain its corresponding true value.

Let s_i and s_j $(i \neq j)$ be the measurements by two abstract sensors for the same physical value $\overline{v}$. If s_i and s_j both contain the correct value, then the intervals s_i and s_j must intersect, and their intersection must contain the (unknown) value $\overline{v}$.

Consider a set $S = \{s_1, s_2, \ldots, s_n\}$ of n independent measurements of the same physical value. If f or less measurements do not contain the correct value, then any set of $n - f$ mutually intersecting

measurements may contain the correct value within their intersection, since they each share a common value. Conversely, any point not contained in at least $n - f$ measurements cannot be the correct value; if it were, then there would be more than f faulty sensors. So, the cover of all $(n - f)$-*cliques* must contain the correct value. (An $(n - f)$-*clique* corresponds to a value where at least $(n - f)$ sensor mesurements intersect.)

We have one further constraint: any program written to deal with a single measurement assumes that the sensor delivers a region of some expected shape (e.g., rectangle, sphere, cube, etc.), so we require the cover to also have this same shape. This constraint allows us to improve a program based on a single (unreliable) real sensor by changing only the sensor; the real sensor is replaced by several real sensors whose inputs are combined to produce a single abstract sensor. The program can use the resulting abstract sensor just as it originally used the single real sensor.

To summarize, we have the following goals for our abstract sensor:

1. It should be guaranteed (assuming no more than f failures) to deliver a region containing the true physical value.

2. It should deliver a shape that is within the same class as the shapes delivered by the individual real sensors.

3. It should be accurate. In other words, assuming no more than f failures, it should deliver a region that is not significantly larger than a region that might be delivered by a single, correct real sensor.

4. It should be efficient to compute. An abstract sensor is useless unless it can be computed in a reasonable amount of time.

It is useful to define $\mathcal{I}_{f,n}(S)$, the smallest region the satisfies goals 1 and 2. In other words, $\mathcal{I}_{f,n}(S)$ is the smallest figure of the correct shape that covers all $(n - f)$-cliques in S. For instance, if the individual sensors report intervals in one dimension then $\mathcal{I}_{f,n}(S)$ is the smallest interval that

contains all the $(n-f)$–cliques. It is clear that the (unknown) true value $\overline{v}$ is a member of $\mathcal{I}_{f,n}(S)$ as long as no more than f measurements are faulty.

Figure 1 illustrates $\mathcal{I}_{f,n}(S)$ for measurements that are rectangles. The left-hand figure shows four measurements, and the right-hand figure shows the rectangle that covers all 3–cliques of the measurements.

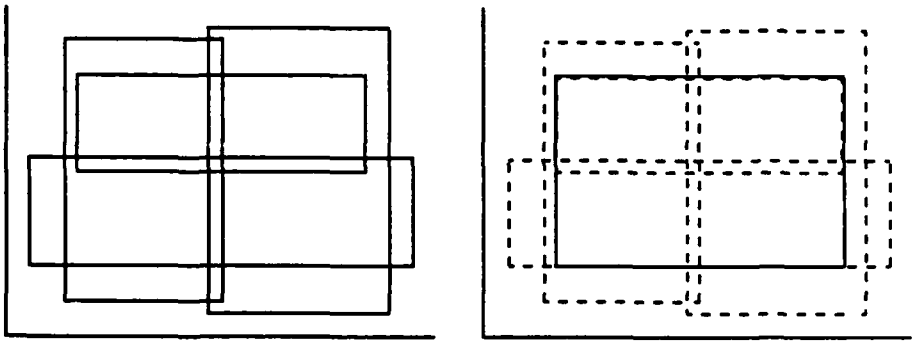

Figure 1: $\mathcal{I}_{1,4}(S)$ for Rectangular Measurements.

Although $\mathcal{I}_{f,n}(S)$ always contains the correct value and is defined for all $f : 0 \leq f < n$, it may be difficult to compute or its size $|\mathcal{I}_{f,n}(S)|$ may be too large to be of use to any control program.

In the following sections, we derive upper bounds on $|\mathcal{I}_{f,n}(S)|$ as a function of f, n, and the sizes of $s_i \in S$. We use this information to develop algorithms for abstract sensors. The results derived in [Mar90] for 1D intervals are summarized in Section 2. In Section 3 we derive upper bounds and algorithms for measurements that are d–dimensional rectangles, and in Section 4 we discuss abstract sensors for measurements that are d–dimensional circles. Note that the results on circles actually hold for any class of convex shapes in which the shapes are geometrically similar and share the same orientation.

2 Linear Sensors

In [Mar90], Marzullo shows that for linear sensors – sensors that report 1D intervals – $\mathcal{I}_{f,n}(S)$ can be found efficiently and that for $f < \frac{n}{2}$, $\mathcal{I}_{f,n}(S)$ has reasonable size. The upper bounds on $|\mathcal{I}_{f,n}(S)|$ are stated in the following two theorems.

First, we need some notation. Define the functions $\mathbf{min}_i$ and $\mathbf{max}_i$ to be the i^{th} smallest and largest values of a set of n values respectively. Note that $\mathbf{min}_i$ is the same as $\mathbf{max}_{n-i+1}$. For example, if $S = \{13, 14, 15\}$ then $\mathbf{min}_3(S) = \mathbf{max}_1(S) = 15$.

Theorem 1 *Let S be a set consisting of n intervals. If $0 \leq f < \frac{n}{2}$ then $|\mathcal{I}_{f,n}(S)| \leq \mathbf{min}_{2f+1}\{|\overline{s}| : \overline{s} \in S\}$.*

Thus, when $f < \frac{n}{2}$, the resulting abstract sensor is as accurate as one of the original sensors. $\mathcal{I}_{f,n}(S)$ can also be computed efficiently: $O(n \log n)$ time, by sorting the endpoints of the n intervals. then moving through the endpoints in order, keeping track of the depth at each instant.

The second theorem states that there is no upper bound on the size when $f \geq \frac{n}{2}$.

Theorem 2 *Given a set $\{\ell_1, \ell_2, ..., \ell_n\}$ of n lengths and $\frac{n}{2} \leq f < n$, then for any length $\Lambda \geq \mathbf{max}\{\ell_1, \ell_2, ..., \ell_n\}$, there exists a set of n intervals $S = \{\overline{s}_1, \overline{s}_2, ..., \overline{s}_n\}$ where $\forall i : 1 \leq i \leq n : |\overline{s}_i| = \ell_i$ and $|\mathcal{I}_{f,n}(S)| = \Lambda$.*

2.1 Multidimensional Sensors and Projection

The 1D results on intervals can be used directly to give results for multidimensional sensors. For a d–dimensional sensor, we project the region for sensor s_i onto each of the d orthogonal axes. We now have d separate 1D problems. These problems can be solved individually and then recombined to produce a d–rectangle.

There are several possible disadvantages to this approach:

1. Information may be lost. For example, the knowledge that a sensor's x–coordinate cannot possibly be correct should be used to toss out the entire sensor.

2. A d–rectangle is not necessarily the desired shape. For example, our abstract sensor may be required to report a circle.

3. The size of the resulting sensor may be larger than necessary (see Figure 2).

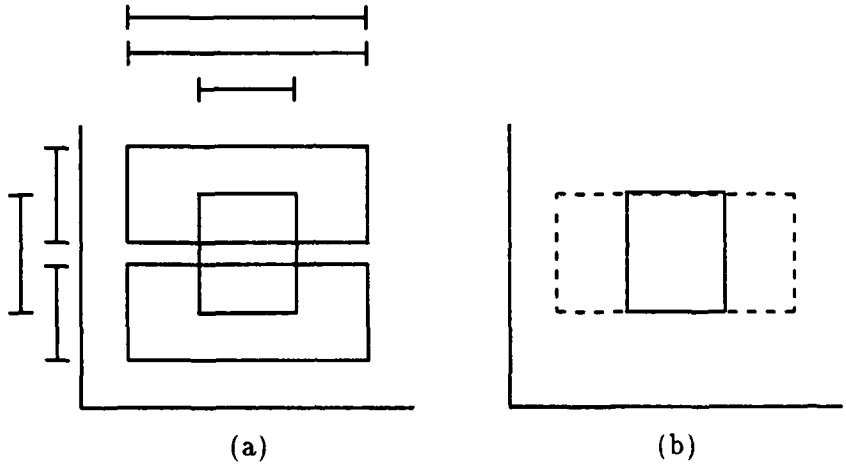

Figure 2: Intersection vs. Intersection of Projections ($n = 3, f = 1$)

In fact, projection techniques are the method-of-choice in some situations (see Section 3), but these situations depend on the shapes involved and the relationship between f and n.

3 d–Rectangles

If s_i is constrained to be a d–dimensional rectangle, then another upper bound can be placed on the size of $\mathcal{I}_{f,n}(S)$.

Theorem 3 *Let S be a set consisting of n d–dimensional rectangles. If $0 \leq f < \frac{n}{2d}$ then $|\mathcal{I}_{f,n}(S)| \leq \min_{2df+1}\{|\overline{s}| : \overline{s} \in S\}$.*

The proof of this theorem is based on a counting argument that shows $\mathcal{I}_{f,n}(S)$ is contained in at least $n - 2df$ of the original rectangles.

The bound on f given in the theorem is tight. Figure 1 shows a 2D example where $f = \frac{n}{2d}$ and $\mathcal{I}_{f,n}(S)$ is larger (in area) than any of the original rectangles. Similar examples can be built for any dimension d.

This theorem shows that increased accuracy comes with a price: if it is desired that $|\mathcal{I}_{f,n}(S)|$ be at least as accurate as some measurement in S, then the amount of replication needed increases quickly (linearly) with d. For example, in order to tolerate a single failure for measurements that are $3D$ rectangles, a sensor must be replicated at least 7 times.

3.1 Algorithms for Rectangles

For 2D problems (and for 1D problems), efficient algorithms exist to compute $\mathcal{I}_{f,n}(S)$ directly. Consider rectangles in two dimensions. The smallest rectangle containing all of the $(n-f)$–cliques can be found in $O(n \log n)$ time by using a sweep-line combined with Bentley's segment tree (see, for instance, [PS85]). Note that, although the entire boundary of the $(n-f)$–cliques can be of complexity n^2, we need only determine the left, right, top, and bottom boundaries. This can be done efficiently by keeping depth information within the segment tree.

Unfortunately, this technique does not generalize well to higher dimensions. For instance, 3D rectangles (rectangular parallelopipeds) require a sweep-plane with dynamic insertion and deletion of 2D rectangles.

There is however, an efficient algorithm that reports a d–rectangle for any d that is almost as good as the minimal d–rectangle that we desire. This uses the projection technique, converting a d-dimensional problem into d 1-dimensional problems. The results of these separate 1D problems are combined to produce the *projection rectangle*, a d-rectangle that is guaranteed to be of reasonable size. The algorithm is based on the following theorem.

Theorem 4 *Let S be a set consisting of n d-dimensional rectangles. If $0 \leq f < \frac{n}{2d}$ then the size of the projection rectangle is $\leq \min_{2df+1}\{|\bar{s}| : \bar{s} \in S\}$.*

Note that the projection rectangle can be computed in $O(dn \log n)$ time and has exactly the same size bound as $\mathcal{I}_{f,n}(S)$. Thus, if our goal is create an abstract sensor that is at least as accurate as some measurement in S, the projection rectangle is as good as $\mathcal{I}_{f,n}(S)$.

The full paper will include examples showing that neither $\mathcal{I}_{f,n}(S)$ nor the projection rectangle is necessarily larger than the other.

4 d–Circles

If s_i is constrained to be a d–dimensional circle (sphere in 3D) then the following upper bound can be placed on the size of $\mathcal{I}_{f,n}(S)$:

Theorem 5 *Let S be a set consisting of n d–circles. If $0 \leq f < \frac{n}{(d+1)}$ then $|\mathcal{I}_{f,n}(S)| \leq \min_{(d+1)f+1}\{|\overline{s}| : \overline{s} \in S\}$.*

The proof of this theorem will appear in the full paper. Note that this bound grows more slowly with d then does the bound of Theorem 3. For example, in order to tolerate a single failure for measurements that are spheres, a sensor must be replicated at least 4 times.

Algorithms for d–circles are not as efficient as algorithms for d–rectangles. Even in 2D, it appears that to find the $(n-f)$–cliques, it is necessary to build the entire arrangement of n circles. Since n circles can have $\Omega(n^2)$ intersections, building the arrangement must take time $\Omega(n^2)$. (The incremental algorithm for building an arrangement of circles takes worst-case time $O(n\lambda_4(n))$ where λ_4 is an almost-linear function related to Davenport-Schinzel sequences [EGPRSS]; using randomization, the arrangement can be built in expected time $O(m + n \log n)$ where m is the number of intersections [Mul89].) Of course, we can replace each d–circle by a d–square that contains it and use the rectangle techniques, but this may produce an answer less accurate than desired.

5 Discussion

We have shown how several real sensors (that measure the same multidimensional physical data) can be combined to produce a reliable *abstract sensor*. This process can be done efficiently, reporting a region guaranteed to be of reasonable size, for d–rectangles provided $f < \frac{n}{2d}$ where n is the number of real sensors and f is the number of real sensors that are faulty. For d–circles, an abstract sensor region of reasonable size exists provided $f < \frac{n}{d+1}$, but determining this region is considerably less efficient. As mentioned in the Introduction, the results on size bounds for circles actually hold for any class of convex shapes in which the shapes are geometrically similar and share the same orientation.

Improved results are possible if sensors are known to report d–rectangles that are all the same size and orientation. In this case, the projection technique can be used to create an abstract sensor which reports a d–rectangle of the standard size in $O(dn \log n)$ time provided $f < \frac{n}{2}$. Note that for this case, the required relation between f and n is independent of d. The reported rectangle may not correspond to any of the original rectangles, but it will be bounded by the correct size.

In contrast, for identically sized circles, the smallest circle covering all of the $(n - f)$–cliques may be larger than the initial circles even when $f < \frac{n}{2}$. Of course, the bound in Theorem 5 still applies; $|\mathcal{I}_{f,n}(S)|$ is bounded by the size of the initial circles when $f < \frac{n}{d+1}$.

In this shortened version of our work, we have room for only a brief mention of fast approximation techniques. A grid of equal-sized buckets can be used to detect $(n - f)$–cliques, leading to a linear-time abstract-sensor algorithm at the cost of some accuracy. This technique works for both d–rectangles and d–circles, but is more accurate for rectangles.

References

[CAS86] Flaviu Cristian, Houtan Aghili, and Ray Strong. Atomic broadcast: From simple message diffusion to Byzantine agreement. Technical Report RJ 5244 (54244), IBM Almaden Research Laboratory, July 1986.

[EGPRSS] Herbert Edelsbrunner, Leonidas J. Guibas, Janos Pach, Richard Pollack, Raimund Seidel, and Micha Sharir. Arrangements of curves in the plane – topology, combinatorics, and algorithms. Technical Report UIUCDCS-R-88-1477, University of Illinois at Urbana-Champaign, December 1988.

[Gol80] Martin C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, 1980.

[Lam84] Leslie Lamport. Using time instead of timeout for fault-tolerant distributed systems. *ACM Transactions on Programming Languages and Systems*, 6(2):254–280, April 1984.

[Mar90] Keith Marzullo. Tolerating failures of continuous-valued sensors. Technical Report TR 90-1156, Cornell University, September 1990.

[Mul89] Ketan Mulmuley. A fast planar partition algorithm, II. In *Proceedings of the Fifth Annual Symposium on Computational Geometry*, pages 33–43. ACM Press, June 1989.

[NT88] Gil Neiger and Sam Toueg. Automatically increasing the fault-tolerance of distributed systems. In *Proceedings of the Eighth Symposium on Principles of Distributed Computing*, pages 248–262. ACM SIGPLAN/SIGOPS, August 1988.

[PS85] Franco P. Preparate and Michael I. Shamos. *Computational Geometry.* Springer-Verlag, 1985.

[Sch90] Fred B. Schneider. The state machine approach: A tutorial. *Computing Surveys*, 22(3). September 1990.

[Sho68] R. A. Short. The attainment of reliable digital systems through the use of redundancy: A survey. *IEEE Computer Group News*, 2:2–17, March 1968.

[vN56] John von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. In C. E. Shannon and J. McMarthy, editors, *Automata Studies*, pages 43–98. Princeton University Press, 1956.