# On Precision Bound of Distributed Fault-Tolerant Sensor Fusion Algorithms

Buke Ao, Yongcai Wang, *member, IEEE* Richard Brooks, *Senior member, IEEE,* Iyengar S.S., *Fellow, IEEE,* and Yu Lu,

**Abstract**—Sensors have limited precision and accuracy. They extract data from their environment through physical interactions, which contain noise. The goal of fusion is to make the final decision more robust and minimize the influence of noise and system errors. One aspect of the fusion problem that has not been adequately addressed is establishing the bounds on fusion result precision. Precision is the maximum range of disagreement that can be introduced by one or more faulty inputs. This definition of precision is consistent both with Lamport's Byzantine General's problem and the minimax criteria commonly found in game theory. This paper considers the precision bounds of fault tolerant information fusion approaches, including Byzantine agreement, Marzullo's interval based approach and the Brooks-Iyengar fusion algorithm. We reviewed and derived the precision bounds for several fault-tolerant distributed sensor fusion algorithms. The analysis provides both insight into the limits imposed by fault tolerance and guidance for mapping fusion approaches to applications.

**Index Terms**—Sensor Fusion, Distributed Agreement, Fault Tolerance.

## 1 Introduction

Sensing applications are limited by a number of physical constraints, which make fault tolerant sensor fusion a critical problem in research and in applications. Sensors extract information from their environment by physical interaction with the environment to detect signals from the target traversing an ambient gas, fluid, or solid medium. If the medium is uniform, stochastic models can generally be applied to the sensor readings. So that statistical combination of samples, with increasing number of inputs, will usually increase the accuracy of the system output; increase confidence and bound variance [1].

Unfortunately the ambient medium is rarely uniform, which introduces additional factors, such as multi-path fading, shadowing and occlusion. A large number of noise factors exist in typical sensing scenarios [2]. These factors are not consistent with the assumptions used to create statistical models, which limits the practical utility of statistics.

In physical systems, component failures can lead to the introduction of arbitrary inputs instead of small magnitude noise factors of uniform variance. Loss of network connectivity can remove arbitrary subsets of inputs from calculations. In fact, networking errors can arbitrarily modify the inputs from any sensor traversing the affected network region [3]. Another fact is that sensors are generally distributed.


- *Buke Ao is with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, P.R. China. E-mail: aobuke@hotmail.com*
- *Yongcai Wang are with Tsinghua University, Beijing, P.R. China. E-mail: wangyc@tsinghua.edu.cn*
- *Iyengar S. S. are with Florida International University, Miami. FL. E-mail: iyengar@cis.fiu.edu*
- *Richard Brooks and Yu Lu are with Clemson University, SC. E-mail: rrb@acm.org,lyu@g.clemson.edu*


To counteract errors in systems like this, Lamport et al [4] proposed the Byzantine Generals Problem (BGP) where a number of decision makers strive to make the same decision in the presence of a limited number of purposely deceptive inputs. If an approach could reach correct consensus under these conditions, the system would be robust. The BGP can be considered as a fault tolerant logic problem. Lamport et al proved that agreement on the correct answer is always possible as long as (ignoring network topology constraints) fewer than $1/3$ of inputs are faulty. Where the original problem considered a binary choice, later researchers considered problems that included continuous variables.

Fault-tolerant sensor fusion shares the same goal as BGP, which is to achieve agreement (consensus) among the processing units (PEs) in the presence of faulty, noisy or malicious data. A critically important issue in evaluating algorithms quality in this area is how to define algorithm. There are two related, but subtly different, values that are used in defining the quality of the fusion algorithms, i.e., *accuracy* and *precision*. [1] In distributed sensor fusion,

- *Accuracy* – which we specify in this paper as $\delta$, measures the difference from a sensor's fusion result to the ground truth being measured [5].
- *Precision* – which we specify as $\varepsilon$, measures the degree of disagreement among sensors' outputs, which is the maximum distance between any two sensors' outputs due to different faulty inputs [6].

Although *accuracy* has been widely evaluated in various algorithm design, *precision*, especially the *precision bound*, which measures the worst case (maximum possible) dis-

1. We note that in the literature different variables have been used inconsistently to denote these concepts. The reader is warned to not assume that the variable notations we use will match the variable names in the papers we reference.

agreement among the PEs' outputs, is not thoroughly evaluated. In this paper, we investigated precision bounds for a set of distributed fault-tolerant sensor fusion algorithms, which suggest agreement properties of these algorithms for properly selecting them according to different input data type and output requirements.

The rest of the paper is organized as follows. Section II discusses related work and describes some important algorithms in this domain. Section III looks at the precision of these algorithms. Section IV gives a formal proof on the precision bounds of the Brooks-Iyengar algorithm. The paper finishes with Section V comparing the accuracy and precision of the algorithms discussed.

## 2 Background

Reaching consensus is a fundamental problem that has been applied in many domains [7]. One important application is fusing unreliable sensor inputs. Many algorithms have been proposed [3].

Consider a network of $N$ PEs $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$. Among the PEs, $\tau < N$ of them may be faulty and provide erroneous data. For the sake of analysis, we assume faulty PEs can maliciously conspire to create the worst possible set of inputs to force our algorithm to fail. This is a conservative assumption, that provides performance guarantees while also simplifying our analysis.

For each *non-faulty PE* $S_i$, $v_i$ measures some parameter with a noise factor that introduces a random, bounded deviation from the true value. For a *faulty PE* $S_f$, it can generate arbitrary $v_f$ values and broadcast different values to each collaborating PE. For reaching consensus, the number of faulty inputs must be bounded and proofs of these limits are in [4].

PEs exchange values with others. In this paper, we ignore the effects of the network topology on fusion. Let $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ be the data at a PE, which includes its own data and data received from other PEs. In some algorithms, inputs are scalar values, represented by $v_i$; in some other algorithms, inputs are intervals where $v_i$ is represented by an interval $[v_{i,l}, v_{i,h}]$. If all PE's were non-faulty, any deterministic algorithm should reach consensus in this case, since each PE would run the same algorithm with the same set of inputs.

The values sent from faulty PEs can arbitrarily differ from PE to PE. We will assume that the faulty PEs construct values in a way that maximizes the difference that the other PEs compute. The agreement algorithm should minimize the influence of the faulty inputs; reach consensus, and maximize the precision of the algorithm output. The agreement precision of a fusion algorithm is defined as:

***Definition 1 (Agreement Precision).*** Agreement precision of the fusion algorithm for scalar and interval inputs is:

$$P_A = \begin{cases} \max\limits_{\forall i,j=1,\cdots N, i\neq j} \left\{ \left| v'_i - v'_j \right| \right\}, \text{for scalar;} \\ \max\limits_{\forall i,j=1,\cdots N, i\neq j} \left\{ \left| v'_{i,h} - v'_{j,l} \right|, \left| v'_{j,h} - v'_{i,l} \right| \right\}, \text{for interval} \end{cases} \quad (1)$$

where $S_i$ and $S_j$ are non-faulty PEs with respective fusion results $v'_i$ and $v'_j$.

We now review the set of fusion approaches most relevant to this problem.

The Byzantine Generals Problem (BGP) for synchronous systems was addressed in [8] and [4]. However, Fischer [9] proved that in a complete asynchronous systems it is not possible to guarantee convergence. In [6], the authors showed that convergence was possible in the presence of some synchronous parameters in an incomplete asynchronous system.

Dolev et al's *approximate agreement* algorithm reached consensus within known precision bounds. Fekete [10] modified Dolev's approach by using a Byzantine agreement step to remove values that are inconsistent across PEs. Mahaney and Schneider proposed another approach that they call *inexact agreement*, which considers both accuracy and precision.

More recently, Vaidya [11] proposed an iterative approximate Byzantine consensus (IABC) algorithm to reach consensus in an arbitrary directed graph. This was followed by [12] Byzantine vector consensus (BVC) which, like the other algorithms we discuss in Section III ignored the network topology.

Marzullo [13] proposed a fault tolerance fusion approach that used intervals. It finds an interval where all non-faulty intervals intersect. In most cases, Marzullo's approach achieves better accuracy than individual sensor inputs. The fused interval is at least as accurate as the range of the least accurate individual sensor. Parhami [14] considered interval voting that combines either preference or uncertainty intervals.

The Brooks-Iyengar algorithm was used as a distributed tracking algorithm in the DARPA Sense-IT program and was then applied to a real-time extension of Linux [15]. This approach finds intervals where $N - \tau$ intervals overlap and performs a weighted average of the interval midpoints. This minimizes the influence of faulty inputs by only considering ranges where faulty inputs agree with a number of valid inputs.

Instead of assuming the presence of malicious inputs, other fusion approaches assume the data is contaminated by a limited amount of noise [16] [17] [18] that typically is Gaussian. Fusion typically uses tools from probability, such as maximum likelihood estimation. The PEs try to agree on a value only by iteratively exchanging information with neighbors.

## 3 Precision Bounds

The objective of sensor fusion is to achieve consensus among the PEs and minimize the impact of bad data. In this section, we review the precision bounds of a set of fusion algorithms. Frequently used notations in this section are listed in Table 1.

### 3.1 Naive Averaging

Let $S$ denote the set of PEs, we assume $|S| = N$. Each PE broadcasts its local measurement to all the other PEs and calculates its output as the average value of the measurements

TABLE 1: List of Notations

| Notations | Description |
|---|---|
| $N$ | number of PEs |
| $\tau$ | number of faulty PEs |
| $v_{j,i}$ | measurement received by PE $i$ from PE $j$ |
| $V_i$ | the set of values at PE $i$ |
| $U$ | the set of values from non-faulty PEs |
| $v'_i$ | estimation result at PE $i$ |
| $S_i$ | PE $i$ |
| $\mathcal{S}$ | set of PEs |
| $\delta(U)$ | $\max(U) - \min(U)$ |
| $\varepsilon$ | precision of fusion results |
| $\delta$ | accuracy of fusion results |
| $\kappa$ | the accuracy requirement of input in inexact agreement algorithm |
| $d$ | dimension of vector in Byzantine Vector Consensus algorithms |
| $[l_{j,i}, h_{j,i}]$ | the reading sent to PE $i$ by PE $j$ in interval agreement algorithms |

it has collected. Let $v_{n,i}$ denote the measurement received by PE $i$ from PE $n$ and $v'_i$ be its output, then

$$v'_i = \frac{v_{1,i} + v_{2,i} + \cdots + v_{N,i}}{N}$$

If we assume there is a malicious node $k$, who sends different readings to other PEs, the difference between PE $i$ and PE $j$ is written as

$$\begin{aligned} & \left|v'_i - v'_j\right| \\ = & \left|\frac{v_{1,i} + \cdots + v_{k,i} + v_{N,i}}{N} - \frac{v_{1,j} + \cdots + v_{k,j} + v_{N,j}}{N}\right| \\ = & \frac{|v_{k,i} - v_{k,j}|}{N} \end{aligned}$$

Since $v_{k,i}, v_{k,j} \in (-\infty, \infty)$, the above value is unbounded. So the naive averaging algorithm is not well fault-tolerant.

## 3.2 Approximate Byzantine Agreement

To reduce the impact of faulty inputs, Dolev et al. [19] propose an approximate agreement algorithm, aiming to filter out the extreme inputs.

### 3.2.1 Algorithm Introduction

The values received by PE $i$ can be written as an ordered vector $V_i = \{v_{1,i}, \cdots, v_{N,i}\}$, which is called the multiset at PE i. Note that faulty PEs may broadcast different values to different PEs. As a result, each PE may have different vectors. PE $i$ uses the following equation to estimate its output

$$v'_i = f_{k,\tau}(V_i) = mean(select_k(reduce^{\tau}(V_i))) \tag{2}$$

where $reduce^{\tau}(V_i)$ removes the largest $\tau$ values and smallest $\tau$ values from $V_i$; The reduced result is denoted by a sorted vector $W = \{w_0 \leq w_1 \leq \cdots, \leq w_m\}$. $select_k(W)$ makes a selection on $W$ with an interval $k$. Selection can be understood as a kind of downsample. $k$ is no smaller than the number of different elements between data of two PEs. The result of selection is $\{w_0, w_k, w_{2k}, \ldots, w_{jk}\}$, where $j = \lfloor (m-1)/k \rfloor$; and $mean(x)$ returns the mean over the selection.

### 3.2.2 Precision bound

***Theorem 1 (Maximum difference after one round).*** After one round of estimation, the maximum difference between the outputs of any two non-faulty PEs $S_i, S_j$ is [19]:

$$\max\left|v'_i - v'_j\right| = \frac{\delta(U)}{\lfloor \frac{N-2\tau-1}{k} \rfloor + 1}, k \geq \tau \tag{3}$$

***Proof 1.*** Please see Appendix A.

where $\delta(U) = max(U) - min(U)$ is the maximum distance in $U$. Furthermore, [19] shows that the lower bound of $\max\left|v'_i - v'_j\right|$ with all possible $k$ is obtained when $k = \tau$ and thus we have

$$\left|v'_i - v'_j\right| \leq \frac{\delta(U)}{\lfloor \frac{N-2\tau-1}{\tau} \rfloor + 1}, k = \tau \tag{4}$$

Since $\tau = \lfloor (N-1)/3 \rfloor$ is the maximum number of faulty PEs allowed in the system for obtaining a correct estimate, the precision bound after one round of estimation is given by

$$\max\left|v'_i - v'_j\right| \leq \frac{\delta(U)}{2} \tag{5}$$

### 3.2.3 $\epsilon$ Approximate Agreement

The agreement precision in Equation (5) can be made arbitrarily small by repeating the estimation in Equation (2) multiple times. A $\epsilon$-approximate agreement metric was defined to tolerate some inconsistency. It is proved in [19] that the algorithm can achieve *$\epsilon$-approximate agreement* after multiple rounds:

- *Agreement*: $\forall S_p, S_q \in \mathcal{S}$, $\left|v'_p - v'_q\right| \leq \epsilon$.
- *Validity*: $\forall S_x \in \mathcal{S}$,$v'_x \leq \delta(U)$

where *Validity* means the output of non-faulty PEs is in the range indicated by initial values of the non-faulty PEs.

## 3.3 Inexact Agreement

### 3.3.1 Algorithm Introduction

Mahaney and Scneider [5] assume the initial values $v_i$ of non-faulty PEs have bounded differences and bounded distances from the true value. Therefore, if $v_p, v_q$ are two initial estimates at non-faulty PEs with correct values, then

$$max\ |v_p - v_q| \leq \delta(U), max\ |v_p - \hat{v}| \leq \kappa,$$

Note that $U$ is the value of the set of non-faulty PEs. and $\hat{v}$ is the true value to be measured.

They proposed a Fast Convergence Algorithm (FCA), as shown in Algorithm 1, which is executed by each PE in parallel. Each PE finds a set of *$\tau$-acceptable* PE values from its multiset $V$. A value $v$ in a multiset $V$ *is acceptable* if $(\exists s, f \in \mathbb{R} :\ s \leq v \leq f\ and\ f - s \leq \delta : \#(V, [s, f]) \geq N - m)$, where $\#(V, [s, f])$ is the number of elements of $V$ that have values in the interval $[s, f]$. The algorithm replaces the unacceptable values with $e(V_{accept})$ (average, median or midpoint of $V_{accept}$) and computes the average of this set. The result is the output of inexact agreement.

**Algorithm 1** FCA in one round at a PE $p$

Collect values from other PEs to form a multiset $V$.
Construct an $\tau$-acceptable set $V_{accept}$ from $V$.
Compute $e(V_{accept})$
Replace any values in $V$ that is not in $V_{accept}$ by $e(V_{accept})$.
$v'_p \leftarrow mean(V_{accept})$
**return** $v'_p$;

### 3.3.2 Precision bound

FCA gives better precision than Approximate Byzantine Agreement [5][19].

***Theorem 2 (precision bound of FCA).*** FCA algorithm can lead to convergence as long as only fewer than 1/3 proportion of PEs are faulty. If less than 1/3 PEs are faulty, then the one-round precision and accuracy bounds [5] are :

- *precision*: $\left|v'_p - v'_q\right| \leq \frac{2\tau}{N}\delta(U)$
- *accuracy*: $\left|v'_p - \hat{v}\right| \leq \kappa + \frac{\tau}{N}\delta(U)$

After multi-rounds, the precision can converge to an arbitrarily small value, however the accuracy bound is larger than $\kappa$, which will not converge towards zero.

***Proof 2.*** Please see Appendix B.

## 3.4 Byzantine Vector Consensus (BVC)

Vaidya et al. [12] and Mendes et al.[20] consider multidimensional Byzantine agreement problems, where the input value at each PE is a vector of dimension $d$.

### 3.4.1 Algorithm Introduction

To measure agreement, Mendes used Euclidean distance between vectors, while Vaidya measured distance between each element in the vector[12]. We consider Vaidya's Byzantine Vector Consensus (BVC) algorithms for exact and approximate BVC.

Approximate Byzantine Vector Consensus redefines the $\varepsilon$-*Approximation Agreement* into the vector version. $\varepsilon$-*Approximation Agreement* is met if:

- $\varepsilon$-*Agreement*: For $1 \leq l \leq d$, the distance between the $l$-th elements of any two non-faulty PEs is within $\varepsilon$, where $\varepsilon$ is a pre-defined constant.
- *Validity*: The decision vector at each non-faulty PE is in the convex hull formed by the initially input non-faulty PEs.

Let function $\Gamma(Y)$ find the intersection of all subsets of $Y$. Let $\mathcal{H}(T)$ be the convex hull formed by a multiset $T$, then

$$\Gamma(Y) = \cap_{T \subseteq Y, |T|=|Y|-\tau}\mathcal{H}(T) \tag{6}$$

Then for a distributed systems with $N$ PEs, suppose each PE holds a vector $\mathbf{v}_p \in \mathbb{R}^d$. Suppose $\tau$ of them are faulty. The approximate BVC algorithm is:

Note that $|*|$ denotes the size of this multiset. $V$ contains at least $n - \tau$ elements. $|Z| \leq \mathrm{C}_n^{n-\tau}$.

**Algorithm 2** Approximate BVC in one step at $p$ PE

Each PE collects values from other PEs and forms a multiset $V$.
**for** each $C \subseteq V$ such that $|C| = N - \tau$ **do**
  Construct $\Gamma(C)$ and choose a point deterministically from $\Gamma(C)$ and add it to $Z$
**end for**
**return** $\mathbf{v}' = \frac{\sum_{\mathbf{z} \in Z} \mathbf{z}}{|Z|}$.

### 3.4.2 Precision bound

Vaidya et al. [12] proved the precision bound by finding two most divergent PEs. We show here the precision shrink of one round between $t-1$ and $t$. We first introduce some notations.

- $t$ denotes the rounds number
- $\mathbf{v}_i[t]$ is the vector of PE $S_i$ in $t$th round
- $\mathbf{v}_{il}[t]$ is the $l$-th element of $\mathbf{v}_i[t]$, where $1 \leq l \leq d$.
- $\Omega_l[t] = max_{1 \leq k \leq m}\ \mathbf{v}_{kl}[t]$ in $m$ non-faulty PEs
- $\mu_l[t] = min_{1 \leq k \leq m}\ \mathbf{v}_{kl}[t]$ in $m$ non-faulty PEs

***Theorem 3 (Precision bound of BVC).*** The most divergent element in two decision vectors after one round is:

$$\Omega_l[t] - \mu_l[t] \leq (1-\gamma)(\Omega_l[t-1] - \mu_l[t-1]) \tag{7}$$

$$\gamma = \frac{1}{n(\frac{n}{n-\tau})}, 1 \leq \gamma \leq 1 \tag{8}$$

***Proof 3.*** Please see Appendix C.

This result shows that the maximum difference of all $l$-th elements among $\mathbf{v}'_i (1 \leq i \leq N)$ will reduce by a scale factor $(1-\gamma)$.

## 3.5 Interval-based Sensor Fusion Algorithm

### 3.5.1 Algorithm Introduction

Marzullo [13] proposed an interval-based agreement sensor fusion algorithm. Interval-based agreement algorithms share the same assumptions like point-based algorithms, except that each PE gives an interval type measurement $[l_i, h_i]$ where the true value lies in. Let $[l_{j,i}, h_{j,i}]$ represent the reading sent to PE $i$ by PE $j$, the collected measurements at PE $i$ is written as $V_i = \{[l_{1,i}, h_{1,i}], [l_{2,i}, h_{2,i}], \cdots, [l_i, h_i], \cdots, [l_{N,i}, h_{N,i}]\}, 1 \leq i \leq N$. The interval type measurements overlap each other. We construct a *weighted region diagram (WRD)* to illustrate the scenario. An example is shown in Figure 1. The following terms are used to describe a WRD.

- *k-overlapping interval*: the interval overlapped by $k$ PEs.
- *Weight*: the number of PEs overlapping on an interval.
- *Region*: a set of continuous overlapping interval with weights no smaller than a certain number.
- $a_w$: the most left endpoint of a region consisting of overlapping intervals whose weight are no smaller than $w$.

- $b_w$: the most right endpoint of a region consisting of overlapping intervals whose weight are no smaller than $w$..
- $a = a_{N-2\tau}$,
- $b = b_{N-2\tau}$
- $U_i$: $U_i \subseteq V_i$ is set of correct measurements.

Consider the example shown in Figure 1 where $N = 6$, $\tau = 2$ ($\tau$ is the number of faulty PEs), we assume PE 1 and PE 2 are two malicious PEs, whose readings are shown by two red bars at the top of the figure, while the good measurements are represented by black bars. The union of all measurements, is divided into multiple overlapping areas according to the number of overlapping measurements. Each overlapping area shows a height with respect to the vertical axis, which is equal to its associated weight. The WRD is then constructed by connecting the overlapping areas together. The black stair-step line in Figure 1 is the WRD obtained by only considering the good measurements, which is transformed to the red stair-step line if the bad readings are included. For example, region $[a_3, b_3]$ consists of overlapping areas whose weights $\geqslant 3$.

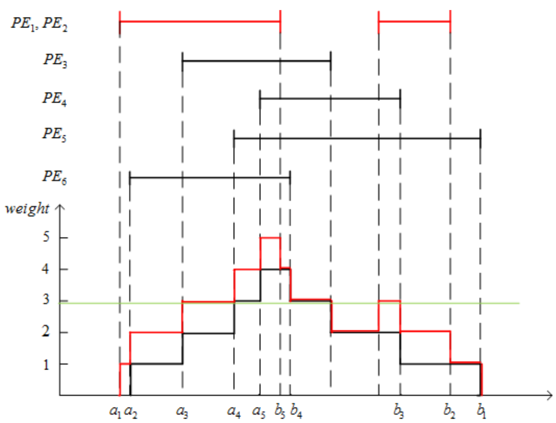


Figure 1: Interval fusion process

Given the WRD built out of the collected measurements, each PE outputs an interval estimate $[a_{N-\tau}, b_{N-\tau}]$. The interval agreement is said to be achieved, if all non-faulty PEs' output intervals contain a same region.

***Theorem 4 (Precision bound of Marzullo's algorithm).*** Let $I'_i$, $I'_j$ be output intervals of two non-faulty PEs $i$ and $j$, then the interval precision bound, as defined in (1) can be calculated as:

$$\max_{\forall i,j=1,\cdots N, i\neq j}\{|v'_{i,h} - v'_{j,l}|, |v'_{j,h} - v'_{i,l}|\} \leq |b_{w=N-2\tau} - a_{w=N-2\tau}| \quad (9)$$

***Proof 4.*** In Marzullo algorithm, regions with weights larger than or equal to $N-\tau$ will be chosen as the output. Since there are $\tau$ faulty intervals, only the regions with weights larger than or equal to $N - 2\tau$ have the possibilities to be chosen. So any two non-faulty output intervals $I_x, I_y$ satisfy $I_x, I_y \subseteq [a_{w=N-2\tau}, b_{w=N-2\tau}]$, then $\forall i, j = 1, \cdots N, i \neq j, v'_{i,h}, v'_{j,l}, v_{j,h}, v'_{i,l} \in [a_{w=N-2\tau}, b_{w=N-2\tau}]$. Therefore, the precision bound is the $distance(I_x, I_y) = |b_{w=N-2\tau} - a_{w=N-2\tau}|$.

### 3.6 Brooks-Iyengar Algorithm

#### 3.6.1 Algorithm Introduction

Brooks-Iyengar algorithm [21] is also interval-based. The output of the algorithm includes a point estimate and an interval estimate around the point estimate. The concrete steps of Brooks-Iyengar algorithm are shown in Algorithm 3. Each PE performs the estimation separately.

**Algorithm 3** Brooks-Iyengar Distributed Sensing Algorithm

**Input:**

The measurement sent by PE $k$ to PE $i$ is a closed interval $[l_{k,i}, h_{k,i}]$, $1 \leq k \leq N$.

**Output:**

The output of PE $i$ includes a point estimate and an interval estimate.

1: PE $i$ receives measurements from all the other PEs.

2: Divide the union of collected measurements into mutually exclusive intervals based on the number of measurements that intersect, which is known as the weight of the interval.

3: Remove intervals with weight less than $N - \tau$, where $\tau$ is the number of faulty PEs.

4: If there are $L$ intervals left, let $A_i$ denote the set of the remaining intervals. We have $A_i = \{(I_1^i, w_1^i), \cdots, (I_L^i, w_L^i)\}$, where interval $I_l^i = \left[l_{I_l^i}, h_{I_l^i}\right]$ and $w_l^i$ is the weight associated with interval $I_l^i$. We also assume $h_{I_l^i} \leq h_{I_{l+1}^i}$.

5: Calculate the point estimate $v'_i$ of PE $i$ as:

$$v'_i = \frac{\sum_l \frac{\left(l_{I_l^i} + h_{I_l^i}\right) \cdot w_l^i}{2}}{\sum_l w_l^i} \quad (10)$$

and the interval estimate is $\left[l_{I_1^i}, h_{I_L^i}\right]$

Consider an example of 5 PEs, in which PE 5 ($S_5$) is sending wrong values to other PEs. Table 2 is the values received by $S_1$.

TABLE 2: $S_1$ in Brooks-Iyengar algorithm

| | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ |
|---|---|---|---|---|---|
| $S_1$ values | [2.7, 6.7] | [0, 3.2] | [1.5, 4.5] | [0.8, 2.8] | [1.4, 4.6] |

As shown in Figure 2, we can determine $A_1$ for PE 1 according to Algorithm 3:

$$A_1 = \{([1.5, 2.7], 4), ([2.7, 2.8], 5), ([2.8, 3.2], 4)\} \quad (11)$$

which consists of intervals where at least $4 (= N - \tau = 5 - 1)$ measurements intersect. With Equation (10), the output of PE 1 is equal to

$$\frac{(4 * \frac{1.5+2.7}{2} + 5 * \frac{2.7+2.8}{2} + 4 * \frac{2.8+3.2}{2})}{13} = 2.625 \quad (12)$$

and the interval estimate is $[1.5, 3.2]$.

#### 3.6.2 Precision Bound

We analyze the precision bound of Brooks-Iyengar algorithm.

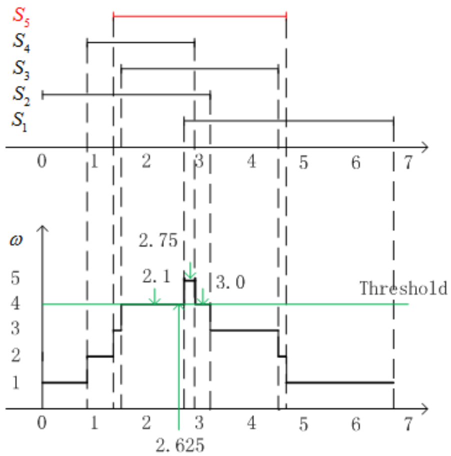


Figure 2: Brooks-Iyengar algorithm in $S_1$

***Theorem 5 (Precision of Brooks-Iyengar Algorithm).*** Given $N$ PEs, $\tau$ of which are faulty and $\alpha = \frac{N-\tau}{(2N-\tau)\tau}$, the precision bound of Brooks-Iyengar algorithm is

$$\frac{1}{1+\alpha} \left| b_{w=N-2\tau} - a_{w=N-2\tau} \right| \tag{13}$$

According to Theorem 5, the precision bound in the "worst case", i.e., when $\tau = \lceil \frac{N}{3} \rceil$, is equal to $\frac{N}{N+1.2}$. Furthermore, as $N \to \infty$, the precision bound approaches $(b-a)$.

#### 3.6.3 Accuracy Bound

The definition of accuracy bound here keeps consistency with inexact agreement. In addition to point estimation $v_i$, each non-faulty PE also outputs an interval $[a_{w=N-\tau}, b_{w=N-\tau}]$, which is the smallest interval that must contain the true value $\hat{v}$.

$$|v_i - \hat{v}| \le |b_{w=N-\tau} - a_{w=N-\tau}| \le min_{\tau+1}\{|u| : u \in U\} \tag{14}$$

$min_{\tau+1}\{|u| : u \in U\}$ (where $|u|$ denotes the length of $u$) indicates the length of the $(\tau+1)$th shortest interval in the set $U$. For example, if $U = \{[1,14],[2,16],[3,18]\}, \tau = 1, \zeta = \{|u| : u \in U\} = \{13,14,15\}$, then $min_{\tau+1}\{|u| : u \in U\} = min_2\{\zeta\} = 14$.

This statement also indicates the fusion accuracy since it describes the distance between the output value and the true value. For $|b_{w=N-\tau} - a_{w=N-\tau}| \le min_{\tau+1}\{|u| : u \in U\}$, an upper bound of fusion accuracy can be inferred by Marzullo [13] as given in Theorem 6.

***Theorem 6.*** Let $U$ be the (unknown) subset of $V$ that are non-faulty. if $\tau < \frac{N}{3}$, then $\left| \bigcap_{\tau,N}(V) \right| \le min_{\tau+1}\{|u| : u \in U\}$.

This theorem means that the derived interval is bounded by a non-faulty sensor that is $(\tau+1)$th best. Here $\bigcap_{\tau,N}(V) = [a_{w=N-\tau}, b_{w=N-\tau}]$ is the output interval of Marzullo's algorithm and Brooks-Iyengar algorithm, it is the smallest interval that is guaranteed to contain the correct true value.

#### 3.6.4 Robustness

Brooks-Iyengar algorithm can tolerant up to $\frac{\tau}{2}$ faulty PEs, the term "tolerant" means that the derived value and interval is bounded by non-faulty PEs. Marzullo [13] proved it in the paper (Theorem 1).

## 4 Proof of Precision Bound of Brooks-Iyengar Algorithm

In this section, we give the details to prove the precision bound of Brooks-Iyengar algorithm.

### 4.1 Model of Brooks-Iyengar Algorithm

The algorithm can be described as a flow in Figure 3. Non-faulty and faulty intervals fuse to generate region and weight. The *Threshold* remove all regions that weight is less than $N - \tau$. Last, all these passed regions' midpoint participate the weighted average and the output is derived.

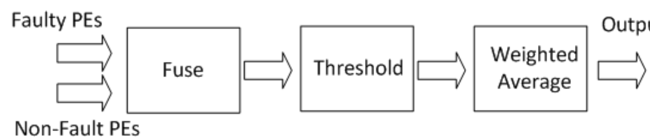


Figure 3: Flow chart of Brooks-Iyengar Algorithm

PE $S_i$ forms set $A_i$ in step 2 and 3 in Brooks-Iyengar algorithm. Table 3 shows an example of the regions, corresponding midpoints and weights in set $A_i$. $N - \tau \le w_p \le N, m_i = (r_{p,l} + r_{p,h})/2, 1 \le p \le M$, and $M$ is the number of regions that pass the threshold.

TABLE 3: Example of $A_i$ in Brooks-Iyengar algorithm

| Index | 1 | ... | $p$ | ... | $M$ |
|---|---|---|---|---|---|
| Region | $[r_{1,l}, r_{1,h}]$ | ... | $[r_{p,l}, r_{p,h}]$ | ... | $[r_{M,l}, r_{M,h}]$ |
| Mean | $m_1$ | ... | $m_p$ | ... | $m_M$ |
| Tuple | $(\omega_1, I_1)$ | ... | $(\omega_p, I_p)$ | ... | $(\omega_M, I_M)$ |

### 4.2 A framework of Proof solutions

#### 4.2.1 Notion and outline

- Interval: the input interval
- Region: a middle result of Brooks-Iyengar algorithm and also is subset of interval.
- $[l_{fi}, h_{fi}]$: $i$th faulty interval, where $1 \le i \le \tau$
- $g, f$: A set consists of $N - \tau$ non-faulty or $\tau$ faulty intervals.
- $G, F$: A set consists of all possible $g$ and $f$.
- $v = BI(g \cup f)$: $g, f$ are non-faulty and faulty intervals set, $v$ denotes the output value of Brooks-Iyengar algorithm with threshold $N - \tau$.
- $\delta(g,f) = BI(g \cup f) - BI(g)$, which is the output bias caused by $f$.
- $m_i$: variables in weighted avearge, midpoint of $i$th region.
- $p, q$: $p$th and $q$th region with weight larger than or equal to $N - 2\tau$, $1 \le p, q \le M$.

### 4.2.2 *Problem Transformation*

We use typical precision bound definition here, this point-based precision bound is maximum distance between non-faulty output values. In a $N, \tau$ distributed system, we can address the precision problem by considering the bias caused by two set of faulty inputs ($f_1$, $f_2$) that try to maximize and minimize the output $v$ respectively:

$$\forall g \in G, \forall f_1, f_2 \in F, max(\delta(g, f_1) - \delta(g, f_2)) \tag{15}$$

We can turn this problem into two similar optimization problems:

$$\begin{aligned} & \max_{\forall g \in G, \forall f_1, f_2 \in F} \delta(g, f_1) - \delta(g, f_2) \\ \leq \; & \max_{\forall g \in G, \forall f_1 \in F} \delta(g, f_1) + \max_{\forall g \in G, \forall f_2 \in F} (-\delta(g, f_2)) \end{aligned} \tag{16}$$

*Challenge*

The main problem in this precision bound problem is that it is hard to use an unified model or expression to describe $v = BI(g \cup f)$. While if we define that $M$ regions have pass the threshold, the output will follow this pattern:

$$v = BI(g, f) = \frac{\Sigma_{i=1}^{M} w_i m_i}{\Sigma_{i=1}^{M} w_i} \tag{17}$$

We use term "**item**" to denote each $w_i m_i$ in Equation 17.

Now we have transformed the problem into two similar optimization problems, we first consider how to positively maximize the bias $\delta(g, f_1)$ then obtain the bound. The following passage tries to find optimal solutions $g^* \in F$ and $f^* \in F$ and this process reflects how the interaction (e.g., position, length, placement strategies) between the non-faulty and faulty intervals affect the output.

***Assumption 1.***

$$\forall g \in G, \cap_{I \in g} I \neq \emptyset$$

This assumption means that all non-faulty intervals intersect on a same region, also the output $v = BI(g)$ is exist.

***Claim 1.*** $\forall g \in G$ and faulty intervals set $f'$ with $0 < |f'| < \tau$, and $v_0 = BI(g \cup f')$. If a new faulty interval $[r_l, r_h]$, where $r_l > v_0$ is added and let $v' = BI(g \cup f' \cup \{[r_l, r_h]\})$, then $v' \geq v_0$.

***Proof 5.*** Assume that the original result $v_0$ is weighted average of $M$ regions with weight $w_i$ and mean $m_i$. If the added interval $[r_l, r_h]$ can introduce new "**items**", then each "**item**" $Ax$ increase the output because $x \geq \frac{\Sigma_{i=1}^{M} w_i m_i}{\Sigma_{i=1}^{M} w_i}$ and

$$\begin{aligned} & \frac{\Sigma_{i=1}^{M} w_i m_i + Ax}{\Sigma_{i=1}^{M} w_i + A} - \frac{\Sigma_{i=1}^{M} w_i m_i}{\Sigma_{i=1}^{M} w_i} \\ = \; & A \frac{x \Sigma w_i - \Sigma w_i m_i}{(\Sigma w_i + A) \Sigma w_i} \geq 0 \end{aligned} \tag{18}$$

If the added interval $[r_l, r_h]$ can not introduce new "item", then the output remains the same.

We prove the precision bound by four Lemmas. In this proof, we first consider the affect of one faulty interval and then the combination of $\tau$ faulty intervals. Lemma 1,2,3 show how the movement of a faulty interval maximize the output bias. Lemma 4 finds some conditions that at least one optimal solution $g^*, f^*$ follows. The counterpart of minimize the output is similar, then the Theorem 5 gives the precision bound.

***Lemma 1.*** $\forall g \in G, f \in F$, $v_0 = BI(g \cup f)$ and if a faulty interval $[l_f, h_f] \in f, l_f \geq v_0$, then $BI(h_f)$ is nondecreasing.

***Proof 6.*** We use $BI(h_f)$ as the functional relation between algorithm output and interval end-point $h_f$. Whether or not the increase of $h_f$ will across the region, we have two cases.

*a):* When the increase of $h_f$ will not across the region, $BI'(h_f) \geq 0$ and the Lemma 1 is right. Suppose that $h_f$ increase on $q$th and $q'$th region in Figure 4 (1) and Figure 5 (2), $BI'(h_f) = 0$. Suppose that $h_f$ increase on $q'$th and $q$th region in Figure 4 (2) and Figure 5 (1), $BI'(h_f) = \frac{w_{q'}}{2} / \Sigma_{i=1}^{M} w_i m_i > 0$ and $BI'(h_f) = \frac{w_q}{2} / \Sigma_{i=1}^{M} w_i m_i > 0$ respectively.

*b):* When the increase of $h_f$ will across the region, the Lemma 1 is also right because of Claim 1. Suppose that $h_f$ increase on different regions, we consider whether or not this will introduce new "items". If no items are introduced, then $v = BI(h_f)$ remains unchanged. If new items are introduced, then $v = BI(h_f)$ will increase according to Claim 1. In Figure 4 (1)(2), $h_f$ across $q$th to $q'$th region and the introduced items cause $v = BI(h_f)$ to increase. In Figure 5 (1)(2), $h_f$ across $q$th to $q'$th region does not introduce new items, $v = BI(h_f)$ remains unchanged.

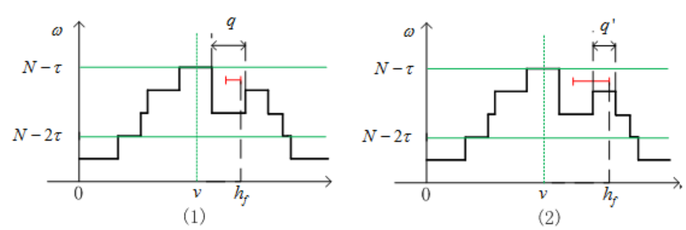


Figure 4: Example 1 of increase of $h_f$

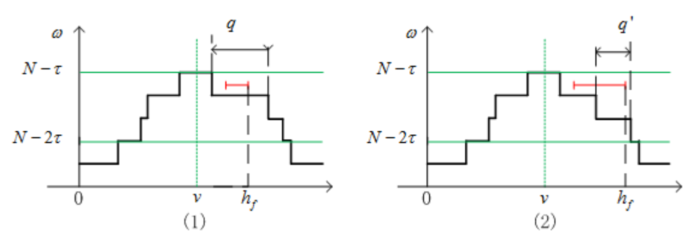


Figure 5: Example 2 of increase of $h_f$

***Lemma 2.*** $\forall g \in G, f \in F$, $v_0 = BI(g \cup f)$ and if a faulty interval $[l_f, h_f] \in f$, where $l_f \geq v_0$ and $l_f$ lies in region $[r_a, r_b)$, then $BI(l_f)$ is increasing on $[r_a, r_b)$.

***Proof 7.*** Like approach of Lemma 1 mentioned before, since $BI'(l_f) \geq 0$, where $l_f \in [r_a, r_b)$, then $BI(l_f)$ is increase

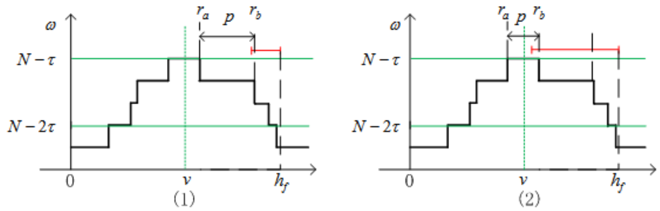


Figure 6: Example of increasing $l_f$

function. Figure 6 gives two examples that $BI(l_f)$ is increasing on $p$th region. $\lim_{l_f \to r_b} BI(l_f) = C$, where $C$ is the algorithm's output that allows single point region $[r_b, r_b]$ with weight $w_p + 1$ to participate the weighted average.

***Lemma 3.*** $\forall g \in G$, $v_0 = BI(g)$, and one faulty interval $[l_f, h_f]$, where $h_f \geq b_{w=N-2\tau}, l_f = b_{w=w_0} - \Delta$ and $w_0 \in \{N-\tau, N-\tau-1\}$, then $\delta(b_{w=w_0})$ is increase function.

***Proof 8.*** Lemma 3 indicates the $g \in G$ that is more vulnerable. Here we define $l_f \in [r_a, r_b)$ and use $r_b$ to substitute $b_{w=w_0}$. For different regions that $l_f$ is in (whether or not lies in the region with weight $N - \tau$ that formed all by non-faulty intervals), there are two cases should be considered. Suppose that the region with weight $N - \tau$ in Figure 6 is formed by all non-faulty intervals, then

*Case 1:* Figure 6 (1) shows an example that $w_0 = N - \tau - 1$, it means that $l_f$ is not in the region with weight $N-\tau$ that formed all by non-faulty intervals. In this case, $[l_f, h_f] = [r_b - \Delta, h_f]$:

$$\delta'(g \cup \{[r_b - \Delta, h_f]\}) = BI'(g \cup \{[r_b - \Delta, h_f]\}) \geq 0$$

*Case 2:* Figure 6 (2) shows an example that $w_0 = N - \tau$, it means that $l_f$ lies in the region with weight $N - \tau$ that is formed by all non-faulty intervals. In this case, Suppose that $(p+1)$th region's weight is $w_{p+1}$, since $l_f = r_b - \Delta$, then

$$\begin{aligned} & \delta'(g_0 \cup \{[r_b - \Delta, h_f]\}) \\ &= BI'(g_0 \cup \{[r_b - \Delta, h_f]\}) - BI'(g_0) \\ &= \frac{d}{dr_b}\Big(\frac{(N-\tau)\frac{r_a+r_b-\Delta}{2} + (N-\tau+1)\frac{r_b-\Delta+r_b}{2} + w_{p+1}\frac{r_b+r_{p+1}}{2}}{N-\tau+(N-\tau+1)+w_{p+1}} \\ &\quad - \frac{r_a+r_b}{2}\Big) \\ &= \frac{(N-\tau)\frac{1}{2} + (N-\tau+1) + w_{p+1}\frac{1}{2}}{(N-\tau)+(N-\tau+1)+w_{p+1}} - \frac{1}{2} > 0 \end{aligned} \tag{19}$$

We can easily extend Lemma 3 to multiple faulty intervals, it is similar that $g$ with larger $b_w$ has larger $\delta(g, f)$, where $N - 2\tau \leq w \leq N - \tau$ and $f$ share conditions with Lemma 3.

***Lemma 4.*** For all possible non-faulty and faulty intervals, problem $max\ \delta(g \cup f)$ exists an optimal solution $g^*$ and $f^*$ satisfy the following conditions:

- $\forall [l_{i,f}, h_{i,f}] \in f^*, h_{i,f} \geq b_{w=N-2\tau}$
- $\forall [l_{i,f}, h_{i,f}] \in f^*, l_{i,f} = b_{w=w_0} - \Delta_1$ where $N - 2\tau \leq w_0 \leq N - \tau$ and $\Delta_1$ is an arbitrary small value.
- $|b_{w=N-2\tau} - b_{w=N-\tau}| \leq \Delta_2$, where $\Delta_2$ is an arbitrary small value.

***Proof 9.*** We can use contradiction to prove it. If there are no $g, f$ that follow all these conditions, we can always make the $v = \delta(g \cup f)$ nondecreasing (increasing $h_f, l_f, b_{w=w_0}$)and find a larger than or equal to $v$ according to Lemma 1, Lemma 2, Lemma 3.

***Theorem 5 (Precision of Brooks-Iyengar Algorithm).*** Given $N$ PEs, $\tau$ of them are faulty, the precision bound of Brooks-Iyengar algorithm is $\frac{1}{1+\alpha}|b - a|$.

$$\alpha = \frac{N-\tau}{(2N-\tau)\tau}$$

***Proof 10.*** According to Lemma 4, we use $g^*, f^*$ to get maximum bias. For the weighted average in condition of Lemma 4, all regions' midpoint are within $[b_{w=N-2\tau} - \Delta_2, b_{w=N-2\tau}]$ except the region that weight is $w = N-\tau$ and is formed by all non-faulty intervals. We use $w_i$ to represent the region's weight and

$$\begin{aligned} & \max_{\forall g \in G, \forall f_1 \in F} \delta(g, f_1) \\ =\ & BI(g^* \cup f^*) - BI(g^*) \\ =\ & \frac{(N-\tau)\frac{a_{w=N-\tau}+b}{2} + \sum w_i b}{N-\tau+\sum w_i} - \frac{a_{w=N-\tau}+b}{2} \end{aligned} \tag{20}$$

To solve Equation 20, we should maximize $\sum w_i$ and minimize $a_{w=N-\tau}$ if $\sum w_i/(N - \tau + \sum w_i) \geq 1/2$. One possible strategy for maximize $\sum w_i$ is $b_{w=N-\tau} - \Delta_3 \leq l_{1,f} < \cdots < l_{\tau,f} < b_{w=N-\tau}$ where $\Delta_3$ is arbitrary small, and we can prove it by Mathematical Induction. In this case, $a_{N-\tau}$ can reach it's minimum $a$. For $\delta(g, f_2)$, we have similar conclusions. Assume that $b = b_{N-2\tau}, a = a_{N-2\tau}$ and $\Delta_3 \to 0$, Figure 7 shows a solution of Equation 16 and we find

$$\begin{aligned} & \max_{\forall g \in G, \forall f_1 \in F} \delta(g, f_1) - \max_{\forall g \in G, \forall f_2 \in F} \delta(g, f_2) \\ <\ & \frac{(N-\tau)\frac{a+b}{2} + (N-\tau+1+\ldots+N+\ldots+N-\tau)b}{(N-\tau)+N-\tau+1+\ldots+N+\ldots+N-\tau} \\ & - \frac{(N-\tau)\frac{a+b}{2} + (N-\tau+1+\ldots+N+\ldots+N-\tau)a}{(N-\tau)+N-\tau+1+\ldots+N+\ldots+N-\tau} \\ =\ & \frac{b-a}{1+\frac{N-\tau}{(2N-\tau)\tau}} \end{aligned} \tag{21}$$

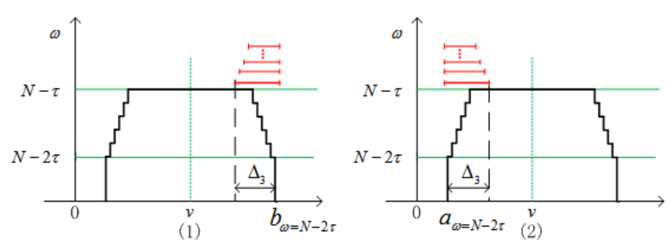


Figure 7: One case to achieve precision bound

TABLE 4: Comparison of agreement-based distributed sensing algorithms

| **Algorithm** | **Approximate agreement** | **FCA** | **Approximate BVC** | **Marzullo sensor fusion** | **Brooks-Iyengar algorithm** |
|---|---|---|---|---|---|
| Input | scalar | scalar | vector | interval | interval/hybrid |
| Faulty PEs tolerated | $< N/3$ | $< N/3$ | $\leq (N-1)/(d+2)$ | $< N/2$ | $< N/3$ |
| Maximum faulty PEs | $< N/3$ | $< 2N/3$ | $\leq (N-1)/(d+2)$ | $< N/2$ | $< N/2$ |
| Convergence rate [21] | $1/(1+\lfloor N-2\tau-1 \rfloor)$ | $2\tau/N$ | $(1-\gamma)$ | $2 * accuracy$ | $2\tau/N$ |
| Accuracy | $\delta(U)$ | $\kappa + \delta\tau/N$ | in the convex hull | $[a_{w=N-\tau}, b_{w=N-\tau}]$ | $[a_{w=N-\tau}, b_{w=N-\tau}]$ |
| Precision of each round | $\delta(U)/2$ | $2\tau\delta/N$ | $(1-\gamma)(\Omega_l[t-1]-\mu_l[t-1])$ | $\lvert b_{w=N-2\tau} - a_{w=N-2\tau} \rvert$ | $\lvert b_{w=N-2\tau} - a_{w=N-2\tau} \rvert / (1+\alpha)$ |

## 5 Conclusion

This paper surveyed a set of distributed, agreement-based sensor fusion algorithms and investigated their precision bounds. The precision bound indicates the disagreement level of the fusion results of a distributed fusion algorithm. We focused on point-based and interval-based distributed fusion algorithms. The characters of the investigated algorithms and their precision bounds were summarized in Table 4. An comparable summary is in paper [21], but our study include more performance metrics, including precision, faulty PEs tolerance etc.

From Table 4, i) regarding fault tolerance, FCA, Marzullo sensor fusion, and Brooks-Iyengar show better performances than other investigated algorithms, in terms of the tolerable faulty PEs. ii) regarding estimation accuracy, approximate agreement algorithm can bound output to within the accuracy bound of non-faulty input without any improvement; FCA can not guarantee the output be better than the input in the worst cases. By our definition of accuracy for interval-based fusion algorithms, Marzullo sensor fusion method and Brooks-Iyengar provides the same accuracy level. iii) for precision, all algorithms can iterate to improve the precision except Marzullo sensor fusion method; For interval-based sensor fusion, Brooks-Iyengar algorithm provides better precision bound than the Marzullo's. iv). No algorithms except Brooks-Iyengar algorithm conduct both a point estimation and an interval estimation in one round. Brooks-Iyengar algorithm considers different numbers of PEs and provides a point estimation in the presence of faults, which can be extended to solve problems in other areas [21] (e.g., floating-point computations, software reliability).

Since agreement-based distributed fusion plays important role in distributed computing, sensor networks and other distributed applications, selection of the fusion algorithm will be important for users, which can be based on the above mentioned characters of these investigated algorithms.

## Acknowledgment

The authors would like to thank Vasanth Iyer on their comments on paper presentation. This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61202360, 61033001, 61361136003.

## References


[1] Feng Zhao and Leonidas J Guibas. *Wireless sensor networks: an information processing approach*. Morgan Kaufmann, 2004.

[2] David Swanson. "environmental effects". In S Sitharama Iyengar and Richard R Brooks, editors, *Distributed Sensor Networks: Image and Sensor Signal Processing*. CRC press, 2012.

[3] Richard R Brooks and Sundararaja S Iyengar. *Multi-sensor fusion: fundamentals and applications with software*. Prentice-Hall, Inc., 1998.

[4] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

[5] Stephen R Mahaney and Fred B Schneider. Inexact agreement: accuracy, precision, and graceful degradation. In *Proceedings of the fourth annual ACM symposium on Principles of distributed computing (PODC)*, pages 237–249. ACM, 1985.

[6] Danny Dolev, Cynthia Dwork, and Larry Stockmeyer. On the minimal synchronism needed for distributed consensus. *Journal of the ACM (JACM)*, 34(1):77–97, 1987.

[7] Michael Barborak, Anton Dahbura, and Miroslaw Malek. The consensus problem in fault-tolerant computing. *ACM Computing Surveys (CSUR)*, 25(2):171–220, 1993.

[8] Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *Journal of the ACM (JACM)*, 27(2):228–234, 1980.

[9] Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.

[10] A D Fekete. Asymptotically optimal algorithms for approximate agreement. In *Proceedings of the Fifth Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 73–87, 1986.

[11] Nitin H Vaidya, Lewis Tseng, and Guanfeng Liang. Iterative approximate byzantine consensus in arbitrary directed graphs. In *Proceedings of the 2012 ACM symposium on Principles of distributed computing*, pages 365–374. ACM, 2012.

[12] Nitin H. Vaidya and Vijay K. Garg. Byzantine vector consensus in complete graphs. In *PODC*, pages 65–73, 2013.

[13] Keith Marzullo. Tolerating failures of continuous-valued sensors. *ACM Transactions on Computer Systems (TOCS)*, 8(4):284–304, 1990.

[14] Behrooz Parhami. Distributed interval voting with node failures of various types. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, pages 1–7. IEEE, 2007.

[15] Vijay Kumar. Impact of brooks-iyengar distributed sensing algorithm on real time systems. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, page 1, 2013.

[16] Lin Xiao, Stephen Boyd, and Sanjay Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 63–70. IEEE, 2005.

[17] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[18] Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, 49(9):1520–1533, 2004.
[19] Danny Dolev, Nancy A Lynch, Shlomit S Pinter, Eugene W Stark, and William E Weihl. Reaching approximate agreement in the presence of faults. *Journal of the ACM (JACM)*, 33(3):499–516, 1986.
[20] Hammurabi Mendes and Maurice Herlihy. Multidimensional approximate agreement in byzantine asynchronous systems. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing(STOC)*, pages 391–400. ACM, 2013.
[21] Richard R Brooks and S Sitharama Iyengar. Robust distributed computing and sensing algorithm. *Computer*, 29(6):53–60, 1996.

## APPENDIX

### 5.1 Proof of Theorem 1

***Proof 11.*** The following proof is based on the proof sketch in [19]. Let $S_i$ and $S_j$ formed multisets $V$ and $W$ respectively after the information exchange. Then $|V| = |W| = N$. Since there are at most $\tau$ faulty processes, $|V - U| \leq \tau$ and $|W - U| \leq \tau$. Moreover, since $V$ and $W$ both contain the $N-\tau$ non-faulty values , so $|V - W| = |W - V| \leq \tau$.

Let $c(N,k) = \lfloor \frac{N-2\tau-1}{k} \rfloor + 1$, $M = reduce^{\tau}(V)$ and $N = reduce^{\tau}(W)$. It is easily to known that $|select_k(M)| = |select_k(N)| = c(N-2\tau, k)$. Let $m_0 \leq m_1 \leq \dots m_{c-1}$ be the elements of $select_k(M)$, and let $n_0 \leq n_1 \leq \dots \leq n_{c-1}$ be the elements of $select_k(N)$. Notice that there are at least $ki+1$ elements in $M$ that are less than or equal to $m_i$, and at most $ki$ elements in $M$ that are strictly less than $m_i$; similarly for $N$.

$$\begin{aligned}
& |f_{k,\tau}(V) - f_{k,\tau}(W)| \\
= & |mean(select_k(M)) - mean(select_k(N))| \\
= & \frac{1}{c}\left|\sum_{i=0}^{c-1} m_i - \sum_{i=0}^{c-1} n_i\right| \\
= & \frac{1}{c}\left|\sum_{i=0}^{c-1} (m_i - n_i)\right| \\
\leq & \frac{1}{c}\sum_{i=0}^{c-1} |m_i - n_i| \text{ (by the triangle inequality)} \\
= & \frac{1}{c}\sum_{i=0}^{c-1} (max(m_i, n_i) - min(m_i, n_i)).
\end{aligned}$$

Showing that $max(m_i, n_i) \leq min(m_{i+1}, n_{i+1})$ for $0 \leq i \leq c-2$. It suffices to show that $m_i \leq n_{i+1}$; a symmetric argument demonstrates that $n_i \leq m_{i+1}$. We proceed by contradiction: Suppose that $m_i \geq n_{i+1}$. As noted above, there are at least $k(i+1)+1$ elements in $N$ less than or equal to $n_{i+1}$. By our supposition, these elements are strictly less than $m_i$. However, there are at most $ki$ elements in $M$ strictly less than $m_i$. Therefore, there are at least $k(i+1)+1-ki(=k+1)$ elements in $N$ that are not in $M$; thus, $|N - M| \geq k+1$. Now by hypothesis, $|W - V| \leq k$, so $|W \cap V| \geq N - k$. Then Lemma 2 in [19] shows $|N \cap M| \geq N - k - 2\tau$, and hence $|N - M| \leq (N - 2\tau) - (N - k - 2\tau) = k$. This is a contradiction, and we conclude that $m_i \leq n_{i+1}$ .

By the inequality demonstrated above, for $0 \leq i \leq c-2$, $(max(m_i, n_i) - min(m_i, n_i)) \leq (min(m_{i+1}, n_{i+1}) - min(m_i - n_i))$; so we get

$$\begin{aligned}
& |f_{k,\tau}(V) - f_{k,\tau}(W)| \\
\leq & \frac{1}{c}[max(m_{c-1}, n_{c-1}) - min(m_{c-1}, n_{c-1})] \\
& + \frac{1}{c}\sum_{i=0}^{c-2}(min(m_{i+1}, n_{i+1}) - min(m_i, n_i)) \\
\leq & \frac{1}{c}(max(m_{c-1}, n_{c-1}) - min(m_0, n_0)) \\
\leq & \frac{1}{c}(max(U) - min(U)) \\
= & \frac{1}{c}\delta(u)
\end{aligned}$$

The multiset $V, W, U$ satisfy above inequality. Thus,

$$|v'_i - v'_j| \leq \frac{\delta(U)}{\lfloor \frac{N-2\tau-1}{\tau} \rfloor + 1}, k = \tau$$

### 5.2 Proof of Theorem 2

We restate Theorem 2 as following: If $N \geq 3\tau + 1$, during execution of FCA the estimation precision in one round is bounded by $\frac{2\tau}{N}\delta(U)$ [5].

***Proof 12.*** The following proof is based on the proof sketch in [5]. Let $G$ and $F$ be the set of non-faulty and faulty PEs respectively, and $|F| = \tau$. Let $|F_{val}|$ be the number of non-faulty processors with faulty values and suppose $|F| + |F_{val}| \leq m$. Suppose PE $S_p$ is correct and formed $V_p$ after information exchange. Let $A_p$ be the set of acceptable values of $V_p$, $v_{rp}$ be the value $S_p$ obtains from PE $S_r$ and let $v^-_{rp}$ be defined by

$$v^-_{rp} = \begin{cases} e(A_p) & \text{if } v_{rp} \text{ is not in } A_p \\ v_{rp} & \text{otherwise} \end{cases}$$

Define $v^-_{rq}$ for some other correct PE $S_q$ similarly. The following proof is based on the proof framework in [5].

First, we prove that $\left|v^-_{rp} - v^-_{rq}\right| \leq 2\delta(U)$. By the definition of an estimator, if $v^-_{rp} = e(A_p)$ then $v^-_{rp}$ must lie in $intvl(A_p)$, which is the acceptable interval of $V_p$; otherwise, $v^-_{rp} = v_{rp}$ , which lies within $intvl(A_p)$ by definition. For the same reason, $v^-_{rp}$ lies within $intvl(A_q)$. Then by applying Lemma 1 in [5], we can yield $\left|v^-_{rp} - v^-_{rq}\right| \leq 2\delta(U)$.

Next, let $v'_p$ and $v'_q$ be the result value of $S_p$ and $S_q$ respectively computed by FCA.

$$\begin{aligned}
|v'_p - v'_q| &= \left|\frac{1}{N}\sum_{S_r \in F \cup G} v^-_{rp} - \frac{1}{N}\sum_{S_r \in F \cup G} v^-_{rq}\right| \\
&= \frac{1}{N}\left|\sum_{S_r \in F \cup G} (v^-_{rp} - v^-_{rq})\right| \\
&= \frac{1}{N}\left|\sum_{S_r \in F} (v^-_{rp} - v^-_{rq})\right|
\end{aligned}$$

since for $S_r \in G$, $\left|v_{rp}^{-} - v_{rq}^{-}\right| = 0$,

$$\begin{aligned} &\leq \frac{1}{N}\left|\sum_{S_r \in F}(v_{rp}^{-} - v_{rq}^{-})\right| \\ &\leq \frac{2\tau}{N}\delta(U) \end{aligned}$$

### 5.3 Proof of Theorem 3

The following proof is based on the proof sketch in [12]. The prove of precision boundis based on that $Z_i$ and $Z_j$ both contain one identical point. Suppose that $m$ processes $p_1, p_2, \ldots, p_m (m \geq n - \tau)$ are non-faulty and $v_i[t], v_j[t]$ are vectors of two non-faulty PEs at rount $t$ . In [12], Observations 1 and 3 in Part III of the proof of Theorem 5 imply that:

$$v_i[t] = \sum_{k=1}^{m} \alpha_k v_k[t-1] \tag{22}$$

$$\sum_{k=1}^{m} \alpha_k = 1, \alpha_k \geq 0, \alpha_g \geq \gamma \tag{23}$$

$$v_j[t] = \sum_{k=1}^{m} \beta_k v_k[t-1] \tag{24}$$

$$\sum_{k=1}^{m} \beta_k = 1, \beta_k \geq 0, \beta_g \geq \gamma \tag{25}$$

where $1 \leq k \leq m$ and $g$ is the index satisfies that $\alpha_g \geq \gamma$(the existence proof is in [12]), where $\gamma = 1/(n\mathrm{C}_{n-\tau}^{n})$.

$$\begin{aligned} v_{il}[t] &= \sum_{k=1}^{m} \alpha_k v_{kl}[t-1] \\ &\leq \alpha_g v_{gl}[t-1] + (1-\alpha_g)\Omega_l[t-1] \\ &\quad \textit{because } v_{kl}[t-1] \leq \Omega_l[t-1], \forall k \\ &\leq \gamma v_{gl}[t-1] + (\alpha_g - \gamma) v_{gl}[t-1] + (1-\alpha_g)\Omega_l[t-1] \\ &\leq \gamma v_{gl}[t-1] + (\alpha_g - \gamma)\Omega_l[t-1] + (1-\alpha_g)\Omega_l[t-1] \\ &\quad \textit{because } v_{gl}[t-1] \leq \Omega_l[t-1] \textit{ and } \alpha_g \geq \gamma \\ &\leq \gamma v_{gl}[t-1] + (1-\gamma)\Omega_l[t-1] \end{aligned} \tag{26}$$

$$\begin{aligned} v_{jl}[t] &= \sum_{k=1}^{m} \beta_k v_{kl}[t-1] \\ &\geq \beta_g v_{gl}[t-1] + (1-\beta_g)\mu_l[t-1] \\ &\quad \textit{because } v_{kl}[t-1] \geq \mu_l[t-1], \forall k \\ &\geq \gamma v_{gl}[t-1] + (\beta_g - \gamma) v_{gl}[t-1] + (1-\beta_g)\mu_l[t-1] \\ &\geq \gamma v_{gl}[t-1] + (\beta_g - \gamma)\mu_l[t-1] + (1-\beta_g)\mu_l[t-1] \\ &\quad \textit{because } v_{gl}[t-1] \geq \mu_l[t-1] \textit{ and } \beta_g \geq \gamma \\ &\geq \gamma v_{gl}[t-1] + (1-\gamma)\mu_l[t-1] \end{aligned} \tag{27}$$

Subtracting Equation (26) from Equation (27) and according to symmetry of $i$ and $j$ we get

$$|v_{il}[t] - v_{jl}[t]| \leq (1-\gamma)(\Omega_l[t-1] - \mu_l[t-1]) \tag{28}$$