Optimal Fault-Tolerant Data Fusion in Sensor Networks: Fundamental Limits and Efficient Algorithms

Marian Temprana Alonso, Farhad Shirani, S. Sitharama Iyengar

Abstract—Distributed estimation in the context of sensor networks is considered, where distributed agents are given a set of sensor measurements, and are tasked with estimating a target variable. A subset of sensors are assumed to be faulty. The objective is to minimize i) the mean squared estimation error at each node (accuracy objective), and ii) the mean squared distance between the estimates at each pair of nodes (consensus objective). It is shown that there is an inherent tradeoff between the former and latter objectives. Assuming a general stochastic model, the sensor fusion algorithm optimizing this tradeoff is characterized through a computable optimization problem, and a Cramér-Rao type lower bound for the achievable accuracy-consensus loss is obtained. Finding the optimal sensor fusion algorithm is computationally complex. To address this, a general class of low-complexity Brooks-Iyengar Algorithms are introduced, and their performance, in terms of accuracy and consensus objectives, is compared to that of optimal linear estimators through case study simulations of various scenarios.

I. INTRODUCTION

Distributed estimation arises naturally in various application scenarios including sensor networks [1]–[4], robotics [5], navigation, tracking and radar networks [6], and monitoring and surveillance applications [7]. In recent years, there has been renewed interest in distributed estimation scenarios involving sensor fusion due to advances in sensor and communication technologies, which has enabled the application of massive non-homogeneous collections of sensors including radar, LiDAR, camera, ultrasonic, GPS, IMU, and V2X in applications including tracking, navigation, and autonomous driving [8].

We consider the distributed estimation scenario shown in Figure 1, where $m$ distributed agents receive a set of $n$ sensor measurements, in the form of confidence intervals $[L_{i,j}, U_{i,j}], i \in [n], j \in [m]$, and are tasked with finding the best estimate of a target variable $X$ with respect to fidelity criteria on both individual accuracy and collective consensus among sensors. In general, the sensor measurements are noisy, and additionally a subset of sensors may be faulty, where a faulty sensor provides measurements which are independent of the target variable. Furthermore, faulty sensors transmit independent measurements to each of the distributed agents. Faulty sensors model both sensor malfunction as well as adversarial interference in the sensor measurement and transmission phases [9], [10]. The distributed estimation system has two objectives: i) **Accuracy objective**: to minimize the mean squared error of the estimate of the target variable by each agent, and ii) **Consensus objective**: to minimize the squared distance between pairs of estimates of the target variable by each of the distributed agents. The former objective is a local performance objective focusing on the individual accuracy of each sensor, whereas the latter objective is a global performance objective focusing on the collective agreement among distributed sensors. The consensus objective is of interest in applications such as clock synchronization, robot convergence and gathering, autonomous driving, blockchain technologies, and distributed voting, among others. The consensus objective has been studied extensively in the context of the Byzantine Agreement Problem [11]–[14] as well as the study of consensus-based filters [15], [16].

The study of the distributed estimation scenario considered in this paper was initiated in [17], [18]. Marzullo proposed a sensor fusion algorithm in [19] and derived the corresponding worst case performance guarantees. An improved algorithm was proposed by Brooks and Iyengar (BI) in [20], and worst-case theoretical guarantees for both accuracy and consensus objectives were derived in [21]. It was shown that the BI algorithm improves upon Marzullo’s algorithm in terms of the worst-case performance in the consensus objective. On the theory side, prior works have considered various other formulations of the distributed estimation problem under Bayesian, mean squared error, and Dempster–Shafer theory paradigms [15], [16], [22], [23]. In this paper, we first formulate a general distributed estimation problem, and provide an analytical framework to evaluate the performance limits in terms of...
the accuracy and consensus objectives. We show that there is a fundamental tradeoff between these two objectives, and characterize the fusion operation optimizing the tradeoff in the form of a convex optimization problem. Furthermore, we provide a lower bound on the accuracy-consensus loss in terms of the average Fisher information of the sensor output variables via a Cramér-Rao type inequality. Analytical characterization of the fusion operation optimizing the aforementioned tradeoff requires prior knowledge of the underlying statistics, which is not possible in many real-world applications. Furthermore, deriving the optimal decision function has high computational complexity. In the second part of the paper, we propose a generalized class of practical low-complexity Brooks-Iyengar fusion algorithms and evaluate their performance through various simulations of practical scenarios.

Notation: The set \( \{1,2,\cdots,n\}, n \in \mathbb{N} \) is represented by \([n]\). An \( n \)-length vector is written as \([x]_{[n]}\) and an \( n \times m \) matrix is written as \([h]_{[n] \times [m]}\). We write \(x\) and \(h\) instead of \([x]_{[n]}\) and \([h]_{[n] \times [m]}\), respectively, when the dimension is clear from context. The vector \(e_m\) is the \( m \)-length all-ones vector. \( \mathbb{B} \) denotes the Borel \( \sigma \)-field. For the event \( E \), the variable \( 1(E) \) denotes the indicator of the event. For random variable \( X \) with probability space \((X,\mathcal{F},\mathbb{P}_X)\), the Hilbert space of functions of \( X \) with finite variance is denoted by \( \mathcal{L}_2(X) \).

II. PROBLEM FORMULATION

We consider a distributed estimation problem involving \( m \) agents and \( n \) sensors. Each agent wishes to estimate the target variable \( X \) defined on the probability space \((\mathbb{R},\mathcal{B},\mathbb{P}_X)\). Each sensor measures the target variable \( X \) separately, and the measurement output is assumed to be in the form of an interval \([L_i, U_i]\), where \( L < U \). The \( i \)-th sensor transmits the pair \((L_i, U_i)\) to the \( j \)-th agent, where \( i \in [n] \), \( j \in [m] \) and \( L_i \leq U_i \). For brevity, we denote the set of measurements \((L_i, U_i)_{i \in [n]} \) by \( I \), and the subset of measurements \((L_i, U_i)_{i \in [n]} \) by \( I_j \) for \( j \in [m] \). This setup is formalized below.

Definition 1 (Distributed Estimation Setup). A distributed estimation setup is characterized by \((n,m,P_{X|I})\), where \( n, m \in \mathbb{N} \) and \( P_{X|I} \) is a probability measure defined on \( \mathbb{R}^{n+1} \) such that \( L_i \leq U_i \), \( i \in [n], j \in [m] \) with probability one.

Definition 2 (Fusion Algorithm). For a distributed estimation setup characterized by the tuple \((n,m,P_{X|I})\), a fusion algorithm \( f \) consists of a collection of functions \( f_j : \mathbb{R}^2 \rightarrow \mathbb{R}, j \in [m] \). The estimate of \( X \) at the \( j \)-th agent is \( \hat{X}_j \equiv f_j(I_1), j \in [m] \).

Definition 3 (Fusion Objectives). For a given a distributed estimation setup \((n,m,P_{X|I})\) and fusion algorithm \( f(I) = (f_j(I_1), j \in [m]) \), the accuracy is parametrized by the vector \( \text{mse}(f_j), j \in [m] \), where:

\[
\text{mse}(f_j) = \mathbb{E} \left( \left| X - \hat{X}_j \right|^2 \right), \quad j \in [m].
\]

The consensus is parametrized by the vector \( \text{cns}(f_j, f_j'), j, j' \in [m] \), where:

\[
\text{cns}(f_j, f_j') = \mathbb{E} \left( \left| \hat{X}_j - \hat{X}_{j'} \right|^2 \right), \quad j, j' \in [m].
\]

Given parameter \( \lambda \in \mathbb{R} \) such that \( 0 \leq \lambda \leq 1 \), the fusion algorithm \( f^\lambda \) is called \( \lambda \)-optimal if it is the solution to the following optimization problem:

\[
f^\lambda = \arg \min_{f \in \mathcal{F}(I_1 \times \cdots \times I_m)} \lambda \sum_{j \in [m]} \text{mse}(f_j) + \frac{\lambda}{m-1} \sum_{j, j' \in [m], j \neq j'} \text{cns}(f_j, f_j'),
\]

where the minimum is taken over all \( f = (f_j)_{j \in [m]} \) such that \( f_j \in \mathcal{L}_2([\prod_{i \in [n]} L_{i,j} \times U_{i,j}], j \in [m]) \) and \( \lambda \doteq 1 - \lambda \).

The parameter \( \lambda \) in the definition of \( \lambda \)-optimal sensor fusion algorithm captures the priority of accuracy over consensus and vice versa. For instance, the 1-optimal fusion algorithm is the optimal estimator in the absence of a consensus objective. It is well-known that, due to the orthogonality principle, the 1-optimal fusion algorithm is given by \( f_j(I) = (\mathbb{E}(X|I_j), j \in [m]) \) e.g., see [24]. On the other hand, the 0-optimal fusion algorithm maximizes consensus without an accuracy objective, so that all of the estimators output the same constant value. In Section III, we evaluate the accuracy-consensus tradeoff under the general distributed estimation model described above. We characterize the \( \lambda \)-optimal sensor fusion algorithm in the form of a computable optimization problem. The optimization requires knowledge of the underlying distribution and has high computational complexity. In Section IV, we restrict our study to a specific subset of distributed estimation scenarios involving faulty sensor measurements, and provide practical fault-tolerant sensor fusion algorithms with low computational complexity whose performance is evaluated through simulations in Section V.

Remark 1. It can be noted that the objective function in the optimization in Equation (1) is a convex function and the Hilbert space \( \mathcal{L}_2([\prod_{i \in [n]} L_{i,j} \times U_{i,j}], j \in [m]) \) is a convex set. So, Equation (1) describes a convex optimization problem and a unique \( \lambda \)-optimal fusion algorithm \( f^\lambda \) always exists.

Remark 2. Define the set of all achievable accuracy and consensus values \( O \equiv \{m(c) \in \mathbb{R}^2 : \exists f_c(I) \in \mathcal{L}_2([\prod_{i \in [n]} L_{i,j} \times U_{i,j}], j \in [m]) : \text{mse}(f_c) = m, \text{cns}(f_c) = c, \} \), where \( \text{mse}(f_c) \equiv \sum_{j} \text{mse}(f_j) \) and \( \text{cns}(f_c) \equiv \sum_{j,j'} \text{cns}(f_j, f_{j'}) \). The set \( O \) is a convex set. Hence, it is characterized by its supporting hyperplanes [25]. So, characterizing \( O \) is equivalent to solving (1) for all \( \lambda \in [0,1] \).

III. ANALYTICAL DERIVATION OF THE \( \lambda \)-OPTIMAL FUSION ALGORITHM

We derive an analytical expression of the \( \lambda \)-optimal fusion algorithm, under general assumptions on the joint distribution \( P_{X|I} \). An application of this is given in Proposition 4 where the \( \lambda \)-optimal linear estimator is characterized analytically. In Section V, we use the characterization in Proposition 4 to numerically evaluate optimal linear estimators in a specific sensor fusion scenario.

Theorem 1 (\( \lambda \)-Optimal Fusion Algorithm). For a distributed estimation setup characterized by the tuple \((n,m,P_{X|I})\), and given \( 0 \leq \lambda \leq 1 \), the \( \lambda \)-optimal fusion algorithm \( f^\lambda(I) = (f^\lambda_j(I_j), j \in [m]) \) is given by \( f^\lambda_j(I_j) = c^\lambda_j f^\lambda_j(I_j) + b^\lambda_j, j \in [m] \).

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
for constant vectors \( c^* = (c_1^*, \ldots, c_m^*) \) and zero-mean, unit-variance functions \( \tilde{f} = (\tilde{f}_1, \tilde{f}_2, \ldots, \tilde{f}_m) \) given by:

\[
\tilde{f} = \arg \max_{\mathbf{e} \in \mathcal{F}} \Theta(\mathbf{A} \mathbf{e})^{-1} \mathbf{e}^T \\
\mathbf{c}^* = \mathbf{A}^{-1} \mathbf{e}^T \\
\mathbf{b}^* = \mathbb{E}(\mathbf{X} \mathbf{e}_m)
\]

where \( \mathbf{A} = [a_{i,j}]_{i,j \in [m]} \), \( a_{i,j} \triangleq 1 \) if \( i = j \) and \( -\frac{1}{m-1} \) if \( i \neq j \).

Proposition 1 (Optimal Bias Vector). For a fixed \( \bar{f} \in \mathcal{F} \) and \( \mathbf{r} > 0 \), the bias vector \( \mathbf{b}^* \) optimizing Equation (1) is given by:

\[
\mathbf{b}^* = \mathbb{E}(\mathbf{X}) \mathbf{e}_m.
\]

Proof. The proof follows by noting that

\[
\text{mse}(\mathbf{f}_j) = \mathbb{E} \left( \left( (\mathbf{X} - \mathbb{E}(\mathbf{X}) - \tilde{X}_j + \mathbb{E}(\tilde{X}_j))^2 \right) + (\mathbb{E}(\mathbf{X}) - \mathbb{E}(\tilde{X}_j))^2 \right),
\]

which is minimized if \( \mathbb{E}(\mathbf{X}) = \mathbb{E}(\tilde{X}_j) \). Similarly,

\[
\text{cns}(\mathbf{f}_j, \mathbf{f}_j) = \mathbb{E} \left( \left( (\tilde{X}_j - \mathbb{E}((\tilde{X}_j) - \tilde{X}_j + \mathbb{E}(\tilde{X}_j))^2 \right) + (\mathbb{E}(\tilde{X}_j) - \mathbb{E}(\tilde{X}_j))^2 \right),
\]

which is minimized if \( \mathbb{E}(\tilde{X}_j) = \mathbb{E}(\tilde{X}_j) \). So, all terms in (1) are minimized simultaneously if \( \mathbb{E}(\tilde{X}_j) = \mathbb{E}(\tilde{X}_j), j \in [m] \).

In the rest of the paper, without loss of generality, we assume \( \mathbb{E}(\mathbf{X}) = 0 \) so that \( \mathbf{b}^* = 0 \). Next, we fix \( \tilde{f}(\mathbf{l}_i) \) and optimize the amplitude vector \( \mathbf{c} = (c_j, j \in [m]) \). The following proposition characterizes the optimal amplitude vector \( \mathbf{c} \).

Proposition 2 (Optimal Amplitude Vector). For a fixed \( \tilde{f} \in \mathcal{F} \), the amplitude vector \( \mathbf{c}^* \) optimizing (1) is given by:

\[
\mathbf{c}^* = \mathbf{A}^{-1} \mathbf{e}^T.
\]

Proof. Let us fix \( \bar{f} \), and consider Equation (1):

\[
\arg \min_{\mathbf{c}} \lambda \sum_{j \in [m]} \text{mse}(c_j \tilde{f}_j) + \frac{\lambda}{m-1} \sum_{i,j \in [m], i \neq j} \text{cns}(i,j \mathbf{c}_j, \mathbf{c}_j \tilde{f}_j).
\]

For \( j \in [m] \), we take the derivative of the objective function with respect to \( c_j \) and set it equal to 0. Note that

\[
\frac{\partial}{\partial c_j} \text{mse}(c_j \tilde{f}_j) = \mathbb{I}(j = j') (2c_j \mathbb{E}(\tilde{f}_j^2) - 2\mathbb{E}(X^2 \tilde{f}_j))
\]

Furthermore,

\[
\frac{\partial}{\partial c_j} \text{cns}(c_j \tilde{f}_j) = \mathbb{I}(j = j') (2c_j - 2\mathbb{E}(X \tilde{f}_j)).
\]

Let \( L_4 \) denote the objective function in (1). We have:

\[
\frac{\partial}{\partial c_j} L_4 = 2\lambda(c_j - \mathbb{E}(X \tilde{f}_j)) + \frac{2\lambda}{m-1} \sum_{j \in [m]} (c_j - \mathbb{E}(\tilde{f}_j \tilde{f}_j))
\]

for all \( j \in [m] \). This provides a system of \( m \) linear equalities, and solving for \( \mathbf{c}^* \) yields, \( \mathbf{c}^* = \mathbf{A}^{-1} \mathbf{e}^T \) as desired.

Next, we find the optimal \( \tilde{f}() \) given that \( \mathbf{c}^* = \mathbf{A}^{-1} \mathbf{e}^T \).

Proposition 3 (Optimal Direction Vector). Given an arbitrary \( \bar{f} \in \mathcal{F} \), let the corresponding amplitude vector be \( \mathbf{c} = \mathbf{A}^{-1} \mathbf{e}^T \). Then, \( \bar{f} - \mathcal{F} \) optimizing (1) is given by:

\[
\bar{f} = \arg \max_{\mathbf{f} \in \mathcal{F}} \Theta(\mathbf{A} \mathbf{e})^{-1} \mathbf{e}^T.
\]

The proof of this proposition is provided in [26]

B. Optimization over Real-valued Matrices

From Theorem 1 it follows that the optimal fusion algorithm is given by \( \arg \max_{\mathbf{f} \in \mathcal{F}} \Theta(\mathbf{A} \mathbf{e})^{-1} \mathbf{e}^T \). This optimization, taken over all zero-mean and unit-variance functions, can be reformulated as an optimization over real-valued matrices as follows. Let \( \mathbf{F} = (\tilde{f}_{ij}, \tilde{f}_{ij}, \tilde{f}_{ij}, \ldots) \) be an orthonormal basis for the Hilbert space \( \mathcal{L}_2(\mathbb{P}(U_{ij} \times U_{ij}) \in [m]), j \in [m] \). Then, the optimization can be re-written as follows:

\[
\bar{f} = \arg \max_{\mathbf{f} \in \mathcal{F}} \Theta(\mathbf{A} \mathbf{e})^{-1} \mathbf{e}^T.
\]

where \( \tilde{f}_{ij}(\mathbf{l}_i) = \sum_{k=0}^{m} \epsilon_{ik} f_{kj}, j \in [m] \), so that \( a_{i,j} = \mathbb{I}(j = j') - \mathbb{I}(j = j') - \frac{1}{m-1} \sum_{k \in [m]} \epsilon_{ik} f_{kj} \mathbb{E}(X \tilde{f}_{ij}) \). For \( j, j' \in [m] \) and \( \theta_j = \sum_{k} \epsilon_{ik} \mathbb{E}(X \tilde{f}_{ij}) \).

To provide an example, let us consider the 1-optimal fusion algorithm. As mentioned in Section II, it is well-known that the 1-0ptimal fusion algorithm is given as \( \mathbf{f}^* = \mathbb{E}(\mathbf{X} | \mathbf{L}_{ij}, U_{ij}, i \in [m]) \). This result can be re-derived from Equation (4) by taking the orthonormal basis \( \mathbf{F} \) such that

\[
\tilde{f}_{ij}(\mathbf{l}_i) = \frac{\mathbb{E}(\mathbf{X} | \mathbf{L}_{ij}, U_{ij}, i \in [m])}{\sqrt{\mathbb{V}(\mathbb{V}(\mathbf{X} | \mathbf{L}_{ij}, U_{ij}, i \in [m]))}}, j \in [m].
\]

It suffices to show that \( \epsilon_{i,j} = 1, \epsilon_{k,j} = 0, j \in [m], k \neq 1 \). To see this, note that since \( \lambda = 1 \), we have \( a_{i,j} = \mathbb{I}(j = j') \), \( j, j' \in [m] \) so that \( A \) is a diagonal matrix, and \( \Theta = [\epsilon_{1,1}, \epsilon_{2,1}, \ldots, \epsilon_{m,m}] \) due to the fact that \( X \) is orthogonal to the subspace generated by \( (\tilde{f}_{ij}, \tilde{f}_{ij}, \tilde{f}_{ij}, \ldots) \) which follows by the smoothing property of expectation, the fact that \( \tilde{f}_{ij} = \frac{\mathbb{E}(\mathbf{X} | \mathbf{L}_{ij}, U_{ij}, i \in [m])}{\sqrt{\mathbb{V}(\mathbb{V}(\mathbf{X} | \mathbf{L}_{ij}, U_{ij}, i \in [m]))}}, \) and that \( \mathbf{F} \) is an orthonormal basis. So, \( \Theta(\mathbf{A} \mathbf{e})^{-1} \mathbf{e}^T = \sum_{k=1}^{m} \epsilon_{ik} \mathbb{E}(X \tilde{f}_{ik}) \), which is maximized by taking \( \epsilon_{i,j} = 1, \epsilon_{k,j} = 0, j \in [m], k \neq j \), since we must have \( \sum_{k=1}^{m} \epsilon_{ik} \mathbb{E}(X \tilde{f}_{ik}) = 1, j \in [m] \) in the constrained optimization in Equation (4).


C. Cramér-Rao Type Bound on Accuracy-Consensus Loss

In the previous section, we characterized the λ-optimal fusion algorithm. Next, we provide a Cramér-Rao type bound on the performance of the algorithm with respect to the accuracy-consensus loss. To present the resulting bound concisely, we make additional assumptions on the structure of the sensor measurements’ statistics. Specifically, we assume that a subset \( \tau \in [n] \) of the sensors are faulty, and the rest of the sensors are non-faulty. Faulty sensors report independent measurements to each agent, that is, if the \( i \)th sensor is faulty, the pair \( (L_{i,j}, U_{i,j}) \) is independent of \( (L_{i’,j'}, U_{i’,j'}) \), \( (i, j) \neq (i’, j’) \) and \( X \). Otherwise, if the sensor is non-faulty, the sensor reports the same measurement to all agents. That is, there exits \( L_{i}, U_{i} \) such that \( L_{i,j} = L_{i}, j \in [m] \) and \( U_{i,j} = U_{i}, j \in [m] \), and \( X \) is in the interval \([L_{i}, U_{i}]\) with probability one. It is assumed that each sensor is equally likely to be faulty, i.e., the \( \tau \) faulty sensors are chosen randomly and uniformly among the \( n \) sensors. Formally, we consider a joint distribution \( P_{X(L_{i}, \bar{U}_{i})} \) defined on \( \mathbb{R}^{m+1} \) such that \( P(X \in [L_{i}, \bar{U}_{i}]) = 1, i \in [n] \). The pair \( (L_{i}, \bar{U}_{i}), i \in [n], j \in [m] \) represent the ground-truth measurement at the \( i \)th sensor. The joint measure on \( X \) is given as follows:

\[
P_{X(A, (L_{i,j}, U_{i,j}))}(e_{i[j]}, j \in [m]) = \prod_{j=1}^{m} P_{X(A)} \prod_{j=1}^{m} P_{L_{i,j}, \bar{U}_{i,j}}(L_{i,j}, U_{i,j})
\]

Then, \( m^2 \lambda / \overline{I}_f(X) \leq L^*_f, \lambda \in [0, 1] \), where \( \overline{I}_f(X) = \frac{m^{2} \lambda}{\text{Var}(f(X))} \) is the average Fisher information defined as:

\[
\overline{I}_f(X) = \sum_{j=1}^{m} \frac{1}{\lambda} \sum_{j=1}^{m} \text{Var}(f_j(X)) \left( \frac{\partial \log f_j(U_{i,j})(X)}{\partial X} \right)^2.
\]

where the expectation is taken over \( X \). Particularly, if the non-faulty sensors’ outputs are identically distributed given \( X \), we have:

\[
\frac{m^2 \lambda }{(n - \tau) \text{Var}_X(\frac{\partial \log f_j(U_{i,j})(X)}{\partial X})^2} \leq L^*_f.
\]

Proof. Please refer to [26].

IV. LOW-COMPLEXITY FAULT-TOLERANT FUSION ALGORITHMS

Theorem 1 quantifies the accuracy-consensus tradeoff under general assumptions on the target and measurement statistics. Solving Equation (2) requires knowledge of the underlying distribution and has high computational complexity. Next, we restrict our study to the specific subset of distributed estimation scenarios described in Section III-C and provide practical fault-tolerant sensor fusion algorithms. A common approach in the estimation literature is to restrict the search to specific classes of estimation algorithms, e.g., linear and limited-degree polynomial estimation algorithms. One can use Equation (4) to find the optimal fusion algorithm over a given subspace of \( F_{\tau}(\sum_{i=1}^{\tau} f_i(X)) \), \( j \in [m] \). To provide an example of this procedure, we introduce a new class of fusion algorithms called Generalized Brooks-Iyengar Algorithms (GBI), and numerically evaluate their performance and compare with that of linear fusion algorithms, Marzullo’s Algorithm [19], and the original Brooks-Iyengar (BI) Algorithm [20].

The class of linear fusion algorithms \( F_{lin} \) is defined as:

\[
f_{j}(I_j) = \sum_{i=1}^{n} \epsilon_{i,j} \Delta_{i,j} + \gamma_{j}, j \in [m].
\]

The class of GBI algorithms \( F_{GBI} \) is defined as:

\[
f_{j}(I_j) = \sum_{i=1}^{n} \epsilon_{i,j} \Delta_{i,j} + \gamma_{j}, j \in [m].
\]

Remark 3. Finding the optimal GBI fusion algorithm requires solving the following:

\[
\hat{f}_{GBI} = \arg \max_{\theta} \Theta(AA')^{-1} \theta',
\]

where \( f_{j}(I_j) \neq \emptyset \in \mathbb{R} \), \( : f_{j}(I_j) \neq \frac{\sum_{i=1}^{n} \epsilon_{i,j} \Delta_{i,j} + \gamma_{j}}{\sum_{i=1}^{n} \epsilon_{i,j} \Delta_{i,j} + \gamma_{j}} \), \( \mathbb{R} \), \( : f_{j}(I_j) \neq \frac{\sum_{i=1}^{n} \epsilon_{i,j} \Delta_{i,j} + \gamma_{j}}{\sum_{i=1}^{n} \epsilon_{i,j} \Delta_{i,j} + \gamma_{j}} \)

The following proposition characterizes the λ-optimal linear fusion algorithm for \( m = 2 \) and \( \mathbb{E}(X) = 0 \). The characterization is used in the numerical simulations in the subsequent section.

Proposition 4. Consider the distributed estimation setup described by Equation (5). Let \( m = 2, \mathbb{E}(X) = 0, \lambda \in (0, 1) \). The parameters \( (\epsilon_{i,j}, \delta_{i,j}, \gamma_{j}, i \in [n], j \in [m] \) characterizing the λ-optimal linear fusion satisfy:

1) \( \exists (\epsilon_{i,j}, \delta_{i,j}, \gamma_{j}, i \in [n], j \in [1, 2] \)
2) \( \gamma_{j} = n(\epsilon_{i,j} \mathbb{E}(L_{i,j}) + \delta_{i,j} \mathbb{E}(U_{i,j})), j \in [1, 2] \)
3) \( \delta_{j} \) is a root of \( \xi_{1,j} - \xi_{2,j} + \xi_{3,j} \), where

\[
\xi_{1,j} = n \mathbb{E}(U_{i,j}) + n(n-1) \mathbb{E}(L_{i,j})
\]

\[
\xi_{2,j} = 2n \epsilon_{i,j} (n \mathbb{E}(L_{i,j}) + n(n-1) \mathbb{E}(L_{i,j}))
\]

\[
\xi_{3,j} = n \mathbb{E}(L_{i,j}) + n(n-1) \mathbb{E}(L_{i,j})
\]

4) \( (\epsilon_{1}, \epsilon_{2}) \) is given as:

\[
(\epsilon_{1}, \epsilon_{2}) = \arg \min_{\epsilon_{1}, \epsilon_{2}, \epsilon_{3}} \frac{\theta_{1}^2 + \theta_{2}^2 + 2z(\theta_{1} + \theta_{2})^2}{(1 - z^2)^2},
\]

where \( z \) is a root of \( z^2 + 2z + 1 = 0 \), \( z \neq 0 \)

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
where $\theta_j \triangleq n\epsilon_j \text{Cov}(L_1, X) + n\delta_j \text{Cov}(U_1, X)$ and 
\[
z \triangleq - (1 - \lambda) \times (e_1 \epsilon_2 (\text{Var}(L_1) + n(n - 1) \text{Cov}(L_1, L_2)) + \delta_1 \delta_2 (\text{Var}(U_1) + n(n - 1) \text{Cov}(U_1, U_2)) + (e_1 \delta_2 + e_2 \delta_1) (n \text{Cov}(L_1, U_1) + n(n - 1) \text{Cov}(L_1, U_2))).
\]

The proof follows by optimizing (4) on the set of linear fusion algorithms. We provide a summary of the proof arguments: 1) follows by the convexity of the objective function in (4) and its symmetry with respect to the variables $e_i$, $e_i^f$ and $\delta_i, \delta_i^j$ under the distribution given in (5), 2) follows from the fact that $\mathbb{E}(\delta_i) = 0$ and $b = \mathbb{E}(X)e_m = 0$, 3) follows from the fact that $\mathbb{E}(\delta_i^j) = 1$, and 4) follows by setting $A = \begin{bmatrix} 1 & -z \\ -z & 1 \end{bmatrix}$ and simplifying the optimization in Equation (4).

V. CASE STUDY: UNIFORMLY DISTRIBUTED VARIABLES

In order to numerically study the accuracy-consensus tradeoff characterized in the prequel, in this section we consider a specific distributed estimation example and perform numerical simulations of various fusion algorithms. In particular, we consider the distributed estimation setup described by Equation (5) and assume that $X$ is uniformly distributed over the interval $[-x_{\text{max}}, x_{\text{max}}]$, where $x_{\text{max}} \in \mathbb{N}$. Let the precision of the $j$th sensor be parametrized by the random variable $\delta_j$, which is uniformly distributed over the set $\{1, 2, \ldots, x_{\text{max}}\}$. Let the set of possible outputs for the $j$th sensor be $P_j = \{[-x_{\text{max}} + 2 \frac{d - 1}{\delta_j} x_{\text{max}}, -x_{\text{max}} + 2 \delta_j x_{\text{max}}], d \in \delta_j\}$. Note that by this construction, $L_j, U_j, i \in [n]$ are discrete variables taking value from $L_j = \{[-x_{\text{max}} + 2 \frac{d - 1}{\delta_j} x_{\text{max}}, -x_{\text{max}} + 2 \delta_j x_{\text{max}}], d \in \delta_j\}$ and $U_j = \{-x_{\text{max}} + 2 \delta_j x_{\text{max}}, d \in \delta_j\}$, respectively. Given $X \in [-x_{\text{max}}, x_{\text{max}}]$, the variables $L_i, U_i, i \in [n]$ take the unique values in $L_j$ and $U_j$, respectively, for which $X \in [L_i, U_i]$.

The following proposition shows that the 1-optimal fusion algorithm for this setup is a GBI algorithm.

**Proposition 5.** For the distributed estimation setup described above, the 1-optimal linear fusion algorithm is equal to the GBI algorithm with $w_j = \min u_j - \max \ell_j, \prod_{j: i \neq 0} 1, i \in T^0$.

**Proof.** Please refer to [26]. □

In order to compare the performance of the sensor fusion algorithms introduced in Section IV, we numerically simulate their performance in a scenario with $n = 10$ sensors, $m = 2$ agents, $1 \leq \tau \leq 7$ faulty sensors, and $x_{\text{max}} = 5$. We consider the $\lambda$-optimal linear fusion algorithm given in Proposition 4 for $\lambda \in \{0.1, 0.5, 0.9\}$, the original BI algorithm [20], Marzullo’s algorithm [19], and the 1-optimal GBI algorithm introduced in Proposition 5. Figure 2 shows the resulting mean squared error (MSE) and consensus cost (CNS). It can be noted that the 0.1-optimal linear fusion algorithm has the worst MSE and the best CNS among the simulated algorithms. This is expected, as the fusion algorithm prioritizes consensus over accuracy. On the other hand, the 0.9-optimal linear fusion algorithm has the best MSE and worst CNS among the three linear fusion algorithms since it prioritizes accuracy over consensus. Furthermore, the BI and Marzullo’s algorithms perform well for $\tau \leq 2$. This is in agreement with prior works (e.g., [21]) which provide worst-case performance guarantees for BI and Marzullo’s algorithms when $\tau \leq 2$. The simulation shows that in this scenario, the average-case performance is good as well. The 1-optimal GBI has the best MSE performance as it is the optimal sensor fusion algorithm in terms of MSE as shown in Proposition 5. It also outperforms the BI and Marzullo’s algorithms in terms of CNS for $\tau \leq 2$. It is of note that the highly non-linear BI, Marzullo, and GBI algorithms outperform the best linear estimators in terms of MSE in this scenario.

VI. CONCLUSION

Distributed estimation in the context of sensor networks was considered, where a subset of sensor measurements is faulty. Faulty sensors model both sensor malfunctions, as well as adversarial interference in measurement and transmission phases. A computable characterization of the fusion algorithm optimizing the accuracy-consensus tradeoff was provided. Several classes of fusion algorithms were studied, and their performance was evaluated in multiple case study simulations.

**Acknowledgements:** We would like to thank Prof. Azad Madni and Prof. S. Sandeep Pradhan for helpful discussions.
References

[1] I. F. Akyildiz and M. C. Vuran, Wireless sensor networks. John Wiley & Sons, 2010.
[2] W. Zhong and J. Garcia-Frias, “Combining data fusion with joint source-channel coding of correlated sensors,” in Information Theory Workshop. IEEE, 2004, pp. 315–317.
[3] S. He, H.-S. Shin, S. Xu, and A. Tsourdos, “Distributed estimation over a low-cost sensor network: A review of state-of-the-art,” Information Fusion, vol. 54, pp. 21–43, 2020.
[4] J. A. Gubner, “Distributed estimation and quantization,” IEEE Transactions on Information Theory, vol. 39, no. 4, pp. 1456–1459, 1993.
[5] N. Ahmed, J. Cortes, and S. Martinez, “Distributed control and estimation of robotic vehicle networks: Overview of the special issue,” IEEE Control Systems Magazine, vol. 36, no. 2, pp. 36–40, 2016.
[6] J. Yan, W. Pu, S. Zhou, H. Liu, and M. S. Greco, “Optimal resource allocation for asynchronous multiple targets tracking in heterogeneous radar networks,” IEEE Transactions on Signal Processing, vol. 68, pp. 4055–4068, 2020.
[7] F. Pasqualetti, R. Carli, and F. Bullo, “Distributed estimation via iterative projections with application to power network monitoring,” Automatica, vol. 48, no. 5, pp. 747–758, 2012.
[8] A. Kim, A. Olep, and L. Leal-Taixé, “Eagermot: 3d multi-object tracking via sensor fusion,” in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021, pp. 11315–11321.
[9] R. Ivanov, M. Pajic, and I. Lee, “Attack-resilient sensor fusion for safety-critical cyber-physical systems,” ACM Transactions on Embedded Computing Systems (TECS), vol. 15, no. 1, pp. 1–24, 2016.
[10] H. Shin, D. Kim, Y. Kwon, and Y. Kim, “Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications,” in International Conference on Cryptographic Hardware and Embedded Systems. Springer, 2017, pp. 445–467.
[11] L. Lamport, R. Shostak, and M. Pease, “The byzantine generals problem,” ACM Transactions on Programming Languages and Systems, vol. 4, no. 3, pp. 382–401, 1982.
[12] R. Indrakumari, T. Poongodi, K. Saini, and B. Balamurugan, “Consensus algorithms—a survey,” in Blockchain Technology and Applications. Auerbach Publications, 2020, pp. 65–78.
[13] N. Sangwann, V. Narayanan, and V. M. Prabhakaran, “Byzantine consensus over broadcast channels,” in 2022 IEEE International Symposium on Information Theory (ISIT). IEEE, 2022, pp. 1157–1162.
[14] X. He, H. Dai, and P. Ning, “A byzantine attack defender: The conditional frequency check,” in 2012 IEEE International Symposium on Information Theory Proceedings. IEEE, 2012, pp. 975–979.
[15] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” IEEE Transactions on Automatic Control, vol. 49, no. 9, pp. 1520–1533, 2004.
[16] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, “Distributed kalman filtering based on consensus strategies,” IEEE Journal on Selected Areas in Communications, vol. 26, no. 4, p. 622, 2008.
[17] T. R. Mahaney and F. B. Schneider, “Inexact agreement: Accuracy, precision, and graceful degradation,” in Proceedings of the fourth annual ACM symposium on Principles of distributed computing, 1985, pp. 237–249.
[18] D. Dolev, N. A. Lynch, S. S. Pinter, E. W. Stark, and W. E. Weihl, “Reaching approximate agreement in the presence of faults,” Journal of the ACM (JACM), vol. 33, no. 3, pp. 499–516, 1986.
[19] K. Marzullo, “Tolerating failures of continuous-valued sensors,” ACM Transactions on Computer Systems (TOCS), vol. 8, no. 4, pp. 284–304, 1990.
[20] R. R. Brooks and S. S. Iyengar, “Robust distributed computing and sensing algorithm,” Computer, vol. 29, no. 6, pp. 53–60, 1996.
[21] B. Ao, Y. Wang, L. Yu, R. R. Brooks, and S. Iyengar, “On precision bound of distributed fault-tolerant sensor fusion algorithms,” ACM Computing Surveys (CSUR), vol. 49, no. 1, pp. 1–23, 2016.
[22] P. K. Varshney, “Multisensor data fusion,” Electronics & Communication Engineering Journal, vol. 9, no. 6, pp. 245–253, 1997.
[23] R. R. Murphy, “Dempster-shafer theory for sensor fusion in autonomous mobile robots,” IEEE Transactions on Robotics and Automation, vol. 14, no. 2, pp. 197–206, 1998.
[24] R. Durrett, Probability: theory and examples. Cambridge university press, 2019, vol. 49.
[25] A. Gray, E. Abbena, and S. Salamon, Modern differential geometry of curves and surfaces with Mathematica®. Chapman and Hall/CRC, 2017.
[26] M. Temprana Alonso, F. Shirani, and S. S. Iyengar, “Distributed fault-tolerant sensor fusion: Fundamental limits and efficient algorithms,” arXiv preprint, 2022.