sensors

MDPI

Article

Distributed Estimation of Fields Using a Sensor Network with Quantized Measurements

Chethaka Jayasekaramudeli [1], Alex S. Leong [2,*], Alexei T. Skvortsov [2], David J. Nielsen [2] and Omar Ilaya [2]

[1] Faculty of Engineering and Information Technology, University of Melbourne, Parkville 3010, Australia; chethakapethmin@gmail.com
[2] Defence Science and Technology Group, Fishermans Bend, Melbourne 3207, Australia; alexei.skvortsov@defence.gov.au (A.T.S.); david.nielsen2@defence.gov.au (D.J.N.); omar.ilaya@defence.gov.au (O.I.)
* Correspondence: alex.leong@defence.gov.au

Abstract: In this paper, the problem of estimating a scalar field (e.g., the spatial distribution of contaminants in an area) using a sensor network is considered. The sensors are assumed to have quantized measurements. We consider distributed estimation algorithms where each sensor forms its own estimate of the field, with sensors able to share information locally with its neighbours. Two schemes are proposed, called, respectively, measurement diffusion and estimate diffusion. In the measurement diffusion scheme, each sensor broadcasts to its neighbours the latest received measurements of every sensor in the network, while in the estimate diffusion scheme, each sensor will broadcast local estimates and Hessians to its neighbours. Information received from its neighbours will then be iteratively combined at each sensor to form the field estimates. Time-varying scalar fields can also be estimated using both the measurement diffusion and estimate diffusion schemes. Numerical studies illustrate the performance of the proposed algorithms, in particular demonstrating steady state performance close to that of centralized estimation.

Keywords: distributed estimation; field estimation; quantized measurements; sensor networks; time-varying systems

check for updates

Citation: Jayasekaramudeli, C.; Leong, A.S.; Skvortsov, A.T.; Nielsen, D.J.; Ilaya, O. Distributed Estimation of Fields Using a Sensor Network with Quantized Measurements. *Sensors* **2024**, *24*, 5299. https://doi.org/10.3390/s24165299

Academic Editor: Baris Fidan

Received: 14 July 2024
Revised: 7 August 2024
Accepted: 13 August 2024
Published: 15 August 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the event of a hazardous incident, e.g., an accidental or malicious release of chemical, biological, radiological, or nuclear (CBRN) materials, the timely acquisition of information related to the incident (position of the source and structure of associated contamination field) is vital in ensuring comprehensive situational awareness, prompt decision-making, and optimal mitigation strategy. In this regard, a well designed sensor system that can rapidly be deployed in the affected area can be considered as a valuable capability in gaining such information.

In recent years, the use of mobile autonomous vehicles (unmanned aerial vehicles (UAVs) or unmanned ground vehicles (UGVs)) with sensors mounted onboard have attracted increasing attention. These vehicles have been employed for hazard source localization/backtracking [1–10] and inference of field structure [11–17]. For the case of static or slowly varying scenarios, the application of these vehicles for hazard field estimation and reconstruction has proven to be favourable [11–17]. The problem becomes more challenging for the case of dynamic estimation when the field changes more rapidly with time. The issue is related to the logistical constraints preventing timely collection of measurements. Indeed, the number of measurements required to estimate the structure of the field in an operational scenario is usually at least on the order of hundreds [11,14,15,17]. Due to the finite speed of the UAVs/UGVs, this may result in a significant time for a vehicle (or vehicles) to travel to different locations and collect the informative measurements. Other potential issues may be related to constraints on the flight time of small UAVs (which is

often less than half an hour [13]), or the high operational cost of running multiple UGVs simultaneously [18]. These factors necessitate the investigation of some alternative solutions for the dynamic measurements that can subsequently be used as an input in a data fusion system for inferring the time-varying contaminated field structure. One of the favourable options, considered in the literature, is the application of a variation in sensor networks that can be promptly deployed [19,20] after the hazardous release.

We assume that the sensor network consists of a large number of small, low cost (and possibly disposable) sensor nodes with basic sensing, communication, and computation capabilities (Such sensors are also known as low size, weight, power, and cost (SWaP-C) sensors). In the event of a CBRN incident, many sensors could, for example, be dropped into an affected area from an air platform to form a dynamic sensor network [19,20]. In our study, we consider the scenario where each node calculates its own estimate of the field in a distributed manner, where nodes communicate locally with others within a given communication range. The sensors are assumed to have access only to coarsely quantized measurements, which is motivated by the use of low cost sensors with limited capabilities, as well as the fact that many chemical sensors only give output from a small number of discrete bars [21,22].

Quantization is a nonlinear process which is encountered in many areas such as signal processing [23], wireless communications [24], system identification [25], and feedback control [26]. Quantization is inevitable whenever physical quantities are represented numerically in digital systems [27]. While quantization can offer advantages such as reducing the number of bits required to represent information and hence reduce transmission bandwidth, the nonlinear nature of quantization also makes it challenging in the design of systems and algorithms, especially when the quantization is coarse (i.e., the number of quantization levels is small).

Previous studies on field estimation using sensor networks include [28–36]. Distributed estimation schemes are considered in [31,33,35], while [28–30,32,34,36] assume centralized estimation using a fusion centre. Poisson distributed measurements are used in [30], binary measurements in [36], general quantized measurements in [34], and noisy (non-quantized) measurements are assumed in [28,29,31–33,35]. Furthermore, time-varying fields are considered by assuming a PDE model for the field in [29,36]. However, field estimation algorithms, which are simultaneously (1) distributed, (2) can handle quantized measurements, and (3) can handle time-varying fields, appear to be lacking in the literature, which motivates the current work.

Our approach to field estimation entails approximation of the original field as a weighted sum of radial basis functions [37], where the coefficients of this sum are then estimated. To account for the local communication between neighbouring nodes, two conceptual schemes are considered. In the first scheme, which we call *measurement diffusion*, each node broadcasts to its neighbours the latest received measurement of every node in the network. After receiving the broadcasts from its neighbours, each node will then update their field parameter estimates using the newly acquired measurements via an online optimization approach [38]. In the second scheme, called *estimate diffusion*, each node first forms local (pre-)estimates and Hessians of the field parameters, which are then broadcast to its neighbours. Subsequently, each node will combine the received pre-estimates using an estimate fusion method such as covariance intersection [39] or inverse covariance intersection [40].

The key contributions of this paper are the following:

- We propose novel distributed estimation algorithms that can estimate the global field structure at each node in the sensor network, using quantized measurements and local communications.
- The proposed algorithms are iterative, and can handle time variations in the fields and adjust their estimates accordingly.

- We present comprehensive numerical studies of the algorithms, which demonstrate that the proposed algorithms can achieve steady state performance close to that of centralized estimation.

The paper is organized as follows. Section 2 presents the system model and problem statement. The measurement diffusion scheme is presented in Section 3. A brief overview of estimate fusion is given in Section 4, which serves as a preliminary to the estimate diffusion scheme that is considered in Section 5. Some discussions and extensions of the algorithms can be found in Section 6. Numerical results are given in Section 7. Section 8 draws conclusions.

Notation: Given a set $\mathcal{X}$, the cardinality of $\mathcal{X}$ is denoted by $|\mathcal{X}|$. A list of some commonly used symbols in this paper is given in Table 1.

Table 1. Commonly used symbols.

Symbol	Description
$\mathcal{R}$	Region of interest
$\mathbf{x}_m$	Location of node m
$\phi(\mathbf{x})$	Field value at location $\mathbf{x}$
$v_{m,k}(\mathbf{x}_m)$	Measurement noise of node m at time k
$q(x)$	Quantized value of x
$z_{m,k}(\mathbf{x}_m)$	Quantized measurement of node m at time k
$K_i(\mathbf{x})$	i-th radial basis function
$\mathcal{N}_m$	Neighbours of node m
$L(m,n)$	Number of hops between nodes m and n
$\mathcal{C}_{m,k}^{\text{latest}}$	Latest measurements available to node m at time k before broadcast
$\mathcal{C}_{m,k}^{\text{new}}$	New measurements for updating field estimate of node m at time k
$g_n(\boldsymbol{\beta})$	Per stage cost
$\hat{\boldsymbol{\beta}}_k^{(m)}$	Field parameter estimate of node m at time k
$\hat{\boldsymbol{\beta}}_k^{(m-)}$	Field parameter pre-estimate of node m at time k
$H_k^{(m-)}$	Approximate Hessian of node m at time k

2. System Model

Consider a region of interest $\mathcal{R}$. We have a field $\phi(.)$ taking values $\phi(\mathbf{x})$ at position $\mathbf{x} \in \mathbb{R}^2$. The field $\phi(.)$ could, for instance, represent the distribution of concentration levels of some contaminant that we are interested in estimating. For notational simplicity, we do not explicitly include time dependence on the field $\phi(.)$, although our developed algorithms will be able to handle time-varying fields.

We consider a sensor network with M nodes located within $\mathcal{R}$. We assume that the nodes are numbered $1, \ldots, M$ and each node knows its own ID. (We note that algorithms exist for assigning each node in a sensor network a unique ID, see, e.g., [41].) The sensor network can be represented as a graph $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, \ldots, M\}$ and vertex/node m is placed at location $\mathbf{x}_m$, $m = 1, \ldots, M$. We assume that accurate knowledge of the sensor locations can be obtained via GPS. (If GPS is unavailable, then location estimation will also need to be considered [42–44].) An edge (m, n) between nodes m and n exists if $||\mathbf{x}_m - \mathbf{x}_n|| < d$ for some communication radius $d > 0$. Given a node m, the direct neighbours of m are denoted by

$$\mathcal{N}_m \triangleq \{n : ||\mathbf{x}_m - \mathbf{x}_n|| < d,\ n \neq m\}.$$

We will assume that the network is connected.

At time step k (in practice, due to response and clear-down times, chemical sensors can obtain new measurements once every few seconds), the sensor at each node m will take noisy and quantized measurements $z_{m,k}(.)$ of the field, with

$$z_{m,k}(\mathbf{x}_m) = q\big(\phi(\mathbf{x}_m) + v_{m,k}(\mathbf{x}_m)\big). \tag{1}$$

Similar measurements models have been considered in, e.g., [21,22].

The term $v_{m,k}(.)$ in (1) is a noise term, which is assumed to be zero mean. The quantizer $q(.)$ is a quantizer of L levels, say $\{0, 1, \ldots, L-1\}$, of the form

$$q(x) = \begin{cases} 0, & x < \tau_0 \\ 1, & \tau_0 \leq x < \tau_1 \\ \vdots & \vdots \\ L-2, & \tau_{L-3} \leq x < \tau_{L-2} \\ L-1, & x \geq \tau_{L-2} \end{cases} \tag{2}$$

where the quantizer thresholds $\{\tau_0, \ldots, \tau_{L-2}\}$ satisfy $\tau_0 \leq \tau_1 \leq \cdots \leq \tau_{L-2}$. The case $L = 2$ corresponds to an 'on–off' model with binary measurements, while large L can be used to approximate real-valued measurements. Sensor saturation is naturally incorporated into the quantization model. The noise term $v_{m,k}(.)$ can also be used to take into account false positives (in the case of binary measurements) and uncertain quantizer thresholds [21].

Problem Statement

We wish to construct an estimate of the field $\phi(.)$ from the quantized measurements $\{z_{m,k}\}$ collected by the nodes. Distributed estimation schemes will be considered, where each node will form its own estimate of the field. We assume local communications, so that each node m can only transmit/receive information (e.g., measurements or estimates) to/from its direct neighbours $\mathcal{N}_m$. As we are assuming that measurements will be collected increasingly over time, we also want the estimation algorithms to be iterative, i.e., that the estimates will be progressively updated over time.

In order to estimate the field, in this paper, we consider that the field can be sufficiently well approximated by

$$\phi(\mathbf{x}) \approx \sum_{i=1}^{p} \beta_i K_i(\mathbf{x}), \tag{3}$$

where p is the number of radial basis functions, $K_i(\cdot), i = 1, \ldots, p$ is the i-th radial basis function, and β_i the i-th coefficient. Similar models to represent fields have been used in, e.g., [11,12,14–16,30,33]. (Other field models considered in the literature include spatial random processes [31,34,45], PDE models [29,32,36], and as a sum of Fourier components [46]). The use of model (3) is motivated by results from approximation theory, which show that many fields can be approximated arbitrarily closely provided a sufficiently large number of basis functions are used [37]. For the basis functions, we will use the Gaussian kernel

$$K_i(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{c}_i - \mathbf{x}\|^2}{\sigma_i^2}\right),$$

where $\mathbf{c}_i$ and σ_i can be regarded as the centre and width, respectively, of the i-th basis function.

For a given number of basis functions p, we assume that $\mathbf{c}_i$ and σ_i, $i = 1, \ldots, p$, are chosen. (The case where $\mathbf{c}_i$ and σ_i, $i = 1, \ldots, p$, are also estimated can also be considered, but in previous work using mobile sensors, this situation was found to suffer from identifiability

issues and sometimes give very unreliable results [15].) Estimation of the field $\phi(.)$ then becomes a problem of estimating the parameters

$$\boldsymbol{\beta} \triangleq (\beta_1, \ldots, \beta_p).$$

In this paper, we will consider two schemes for distributed estimation of the field $\phi(.)$, which we call measurement diffusion and estimate diffusion, which will be presented in Sections 3 and 5, respectively. The term 'diffusion' is used to convey the idea that information from any node is eventually diffused to the rest of the network via a sequence of local communications between direct neighbours [47,48].

3. Measurement Diffusion

In this section, we consider a scheme which we call *measurement diffusion*, where each node will, at each discrete time step k, broadcast to its direct neighbours the latest received measurement of *every* node in the network. We assume a single broadcast between two successive time steps, in order to reduce the amount of communication and hence energy usage of the nodes [47,48], noting that there exist other distributed estimation schemes based on consensus algorithms, which can involve multiple rounds of communication between successive time steps [49,50].

Consider a node m. At time k, measurement $z_{m,k}$ will be broadcast to the nodes in $\mathcal{N}_m$, since a new measurement has been collected at time k. While for a node $n \in \mathcal{N}_m$, the latest measurement of node n available at node m at time k will be $z_{n,k-1}$, as there is a one hop delay. In general, let $L(m,n)$ denote the length, in terms of the number of hops, of the shortest path between nodes m and n. Then, under this scheme, assuming that no transmissions are lost, the latest measurement of node n available to node m at time k will be $z_{n,k-L(m,n)}$. As an example, suppose we have the sensor network shown in Figure 1, and consider node 2. Nodes 1, 3, and 5 are one hop away from node 2, while node 4 is two hops away from node 2. Thus, at time k, the latest measurements available at node 2 are $z_{1,k-1}$, $z_{2,k}$, $z_{3,k-1}$, $z_{4,k-2}$, and $z_{5,k-1}$. It is not too difficult to see that for a connected network, assuming there are no transmission losses, each measurement from node m will eventually reach any other node n in the network, with a delay given by the number of hops $L(m,n)$ between nodes m and n.

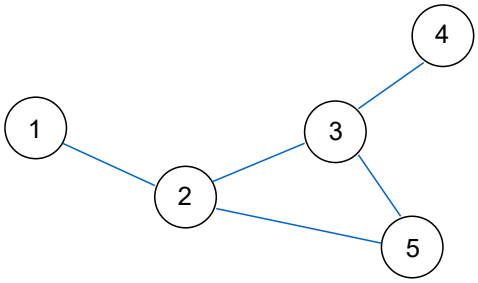


Figure 1. Sensor network example.

Let us be a bit more specific about the communication requirements. At time step k, node m will broadcast the set

$$\mathcal{C}_{m,k}^{\text{latest}} \triangleq \{(n, k_n, z_{n,k_n}) : n = 1, \ldots, M\} \tag{4}$$

to its direct neighbours, i.e., the node ID, time, and latest measurement of every node is transmitted. Here, k_n is the time of the most recent measurement from node n available to node m, and is equal to $k - L(m,n)$ if no transmissions are lost. The locations $\mathbf{x}_n$ of the nodes are not transmitted, as we assume that this information has been previously made available to every node beforehand and can be determined given the node ID.

As well as transmitting, each node will also receive the broadcasts of its direct neighbours. At time k, after receiving broadcasts of its neighbours, each node m will update $\mathcal{C}_{m,k}^{\text{latest}}$

to a set $\mathcal{C}_{m,k}^{\text{latest}+}$ of latest received measurements of every node, together with their corresponding IDs and times. (At time $k+1$, the set $\mathcal{C}_{m,k}^{\text{latest}+}$ updated with the new measurement $z_{m,k+1}$, will then become $\mathcal{C}_{m,k+1}^{\text{latest}}$ and be broadcast by node m.) Since the measurements are time-stamped, this update can be performed efficiently. In order to avoid possible double counting, we then further refine $\mathcal{C}_{m,k}^{\text{latest}+}$ to a set $\mathcal{C}_{m,k}^{\text{new}}$, which contains only new information that have not yet been incorporated into the field estimate of node m. This can also be performed efficiently if the times of the last received measurement of each of the nodes are stored [15], and updated whenever a new measurement is received. Note that if there are no transmission losses, then $\mathcal{C}_{m,k}^{\text{new}} = \mathcal{C}_{m,k}^{\text{latest}+}$.

We next present the estimation procedure for updating the field estimates. The estimation algorithm is based on a computationally efficient online optimization technique [38]. (A sequential Monte Carlo approach can also be used to estimate $\boldsymbol{\beta}$ [15]; however, the online optimization technique is considerably less computationally intensive, while having similar estimation performance [16].) The aim is to estimate $\boldsymbol{\beta}$ at node m by trying to iteratively minimize a cost function $J_{m,k}(.)$:

$$J_{m,k}(\boldsymbol{\beta}) \triangleq \sum_{t=0}^{k} \sum_{(n,k_n,z_{n,k_n})\in\mathcal{C}_{m,k}^{\text{new}}} g_n(\boldsymbol{\beta}) \tag{5}$$

where the following per stage costs are used:

$$g_n(\boldsymbol{\beta}) \triangleq \begin{cases} \log(1+\exp(\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_0))), & z=0 \\ \log(1+\exp(-\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_{z-1}))) & \\ \quad +\log(1+\exp(\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_z))), & z\in\{1,\ldots,L-2\} \\ \log(1+\exp(-\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_{L-2}))), & z=L-1, \end{cases} \tag{6}$$

with $\eta>0$ being a parameter in the logistic function $\ell(x) \triangleq 1/(1+\exp(\eta x))$, and

$$\mathbf{K}(\mathbf{x}) \triangleq \begin{bmatrix} K_1(\mathbf{x}) & K_2(\mathbf{x}) & \ldots & K_p(\mathbf{x}) \end{bmatrix}^T.$$

For notational simplicity. we have also used z in place of z_{n,k_n}. A similar per-stage cost to (6) has been previously shown to be suitable for multi-level quantized measurements [46].

The gradient and Hessian of $g_n(.)$ can be derived as, respectively,

$$\nabla g_n(\boldsymbol{\beta}) = \begin{cases} \frac{\eta}{1+\exp(-\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_0))}\mathbf{K}(\mathbf{x}_n), & z=0 \\ \Big(\frac{-\eta}{1+\exp(\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_{z-1}))} & \\ \quad +\frac{\eta}{1+\exp(-\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_z))}\Big)\mathbf{K}(\mathbf{x}_n), & z\in\{1,\ldots,L-2\} \\ \frac{-\eta}{1+\exp(\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_{L-2}))}\mathbf{K}(\mathbf{x}_n), & z=L-1, \end{cases} \tag{7}$$

and

$$\nabla^2 g_n(\boldsymbol{\beta}) = \begin{cases} \frac{\eta^2\exp(\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_0))}{(1+\exp(\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_0)))^2}\mathbf{K}(\mathbf{x}_n)\mathbf{K}(\mathbf{x}_n)^T, & z=0 \\ \Big(\frac{\eta^2\exp(\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_{z-1}))}{(1+\exp(\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_{z-1})))^2} & \\ \quad +\frac{\eta^2\exp(\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_z))}{(1+\exp(\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_z)))^2}\Big)\mathbf{K}(\mathbf{x}_n)\mathbf{K}(\mathbf{x}_n)^T, & z\in\{1,\ldots,L-2\} \\ \frac{\eta^2\exp(\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_{L-2}))}{(1+\exp(\eta(\boldsymbol{\beta}^T\mathbf{K}(\mathbf{x}_n)-\tau_{L-2})))^2}\mathbf{K}(\mathbf{x}_n)\mathbf{K}(\mathbf{x}_n)^T, & z=L-1. \end{cases} \tag{8}$$

We then iteratively estimate the parameters $\boldsymbol{\beta}$ at node m by performing an approximate Newton update similar to [15,46] using the new measurements in $\mathcal{C}_{m,k}^{\text{new}}$. The formal description of the measurement diffusion procedure that is run at node m is given as Algorithm 1 (each node in the network will perform the same operations), with the equations for the approximate Newton update given in lines 11–14. In Algorithm 1, G and H

represent approximate gradients and Hessians for the cost function (5). The quantity δ in line 12 is a forgetting factor [51] to allow for time-varying fields (where $\boldsymbol{\beta}$ can vary with k) to be estimated. In a slightly different context of sensors mounted on a moving vehicle, previous work in [46] has demonstrated that, for time-varying fields, using a forgetting factor close to but strictly less than 1 allows for quicker tracking of the time-variations than a forgetting factor equal to 1. The quantity ς in line 13 is a Levenberg–Marquardt modification parameter [52] to ensure that the Hessian approximation H is invertible at all times.

Algorithm 1 Field estimation at node m using measurement diffusion

1: **Algorithm Parameters**: Logistic function parameter $\eta > 0$, Levenberg–Marquardt parameter $\varsigma > 0$, forgetting factor $\delta \in (0, 1]$
2: **Outputs**: Parameter estimates $\{\hat{\boldsymbol{\beta}}_k^{(m)}\}$
3: Initialize $\tilde{H} = \mathbf{0}$, $\mathcal{C}_{m,0}^{\text{latest}+} = \emptyset$, and $\hat{\boldsymbol{\beta}}_0^{(m)}$
4: **for** $k = 1, 2, \ldots,$ **do**
5: Collect a measurement $z_{m,k}$
6: Update $\mathcal{C}_{m,k}^{\text{latest}}$ using $\mathcal{C}_{m,k-1}^{\text{latest}+}$ and $z_{m,k}$, and broadcast $\mathcal{C}_{m,k}^{\text{latest}}$ to neighbours $\mathcal{N}_m$
7: Receive $\{\mathcal{C}_{n,k}^{\text{latest}} : n \in \mathcal{N}_m\}$ from neighbours
8: Update $\mathcal{C}_{m,k}^{\text{latest}+}$ and construct $\mathcal{C}_{m,k}^{\text{new}}$
9: Set $\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}_{k-1}^{(m)}$
10: **for** $(n, k_n, z_{n,k_n}) \in \mathcal{C}_{m,k}^{\text{new}}$ **do**
11: $G = \nabla g_n(\tilde{\boldsymbol{\beta}})$ where $\nabla g_n(.)$ is computed using (7)
12: $\tilde{H} \leftarrow \delta\tilde{H} + \nabla^2 g_n(\tilde{\boldsymbol{\beta}})$ where $\nabla^2 g_n(.)$ is computed using (8)
13: $H = \tilde{H} + \varsigma I$
14: $\tilde{\boldsymbol{\beta}} \leftarrow \tilde{\boldsymbol{\beta}} - H^{-1}G$
15: **end for**
16: Update $\hat{\boldsymbol{\beta}}_k^{(m)} = \tilde{\boldsymbol{\beta}}$
17: **end for**

4. Estimate Fusion

In this section, we briefly describe the concept of estimate fusion, which will be used later in the estimate diffusion scheme of Section 5. Consider a quantity $\boldsymbol{\theta}$, which we wish to estimate. Suppose two unbiased estimates $\hat{\boldsymbol{\theta}}_A$ and $\hat{\boldsymbol{\theta}}_B$ of $\boldsymbol{\theta}$ are available, with corresponding estimation error covariances $\mathbf{P}_A$ and $\mathbf{P}_B$. We wish to fuse the two estimates $\hat{\boldsymbol{\theta}}_A$ and $\hat{\boldsymbol{\theta}}_B$ together to provide an improved estimate. In the case of unknown cross-correlations in the errors, elimination of common information [53] used in obtaining $\hat{\boldsymbol{\theta}}_A$ and $\hat{\boldsymbol{\theta}}_B$ is difficult, and one may encounter issues such as double counting, leading to overly confident fused estimates [54]. One way to overcome this issue is to design sub-optimal but consistent fusion rules that over-estimate the true error covariance, for which various different fusion methods have been proposed. In the following, we will describe the covariance intersection [39] and inverse covariance intersection methods [40] methods, which can be used as part of the estimate diffusion scheme to be presented in Section 5, noting that other consistent fusion rules can, in principle, also be considered.

4.1. Covariance Intersection

The most commonly used method for computing a consistent estimate when cross-correlations are unknown is covariance intersection [39]. In this method, the fused estimate and associated covariance are computed as

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{\text{CI}} &= \mathbf{P}_{\text{CI}}\left(\omega\mathbf{P}_A^{-1}\hat{\boldsymbol{\theta}}_A + (1-\omega)\mathbf{P}_B^{-1}\hat{\boldsymbol{\theta}}_B\right) \\ \mathbf{P}_{\text{CI}} &= \left(\omega\mathbf{P}_A^{-1} + (1-\omega)\mathbf{P}_B^{-1}\right)^{-1}, \end{aligned} \tag{9}$$

where the parameter $\omega \in [0,1]$ can be chosen to minimize quantities such as the trace or determinant of $\mathbf{P}_{\mathrm{CI}}$.

In the case where N estimates $\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_N$ with corresponding error covariances $\mathbf{P}_1, \ldots, \mathbf{P}_N$ are to be fused, covariance intersection can be extended to (see [55]):

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{\mathrm{CI}} &= \mathbf{P}_{\mathrm{CI}}\left(\omega_1 \mathbf{P}_1^{-1}\hat{\boldsymbol{\theta}}_1 + \cdots + \omega_N \mathbf{P}_N^{-1}\hat{\boldsymbol{\theta}}_N\right) \\ \mathbf{P}_{\mathrm{CI}} &= \left(\omega_1 \mathbf{P}_1^{-1} + \cdots + \omega_N \mathbf{P}_N^{-1}\right)^{-1}, \end{aligned} \tag{10}$$

where $\omega_n \in [0,1], \forall n$ and $\sum_{n=1}^{N} \omega_n = 1$.

4.2. Inverse Covariance Intersection

Inverse covariance intersection [40] follows a similar principle to covariance intersection, but by considering inverse covariance ellipsoids, it can produce less conservative estimates than covariance intersection. The fused estimate and associated covariance are now computed as

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{\mathrm{ICI}} &= \mathbf{P}_{\mathrm{ICI}}\left(\mathbf{P}_A^{-1} - \omega\left(\omega\mathbf{P}_A + (1-\omega)\mathbf{P}_B\right)^{-1}\right)\hat{\boldsymbol{\theta}}_A \\ &\quad + \mathbf{P}_{\mathrm{ICI}}\left(\mathbf{P}_B^{-1} - (1-\omega)\left(\omega\mathbf{P}_A + (1-\omega)\mathbf{P}_B\right)^{-1}\right)\hat{\boldsymbol{\theta}}_B \\ \mathbf{P}_{\mathrm{ICI}} &= \left(\mathbf{P}_A^{-1} + \mathbf{P}_B^{-1} - \left(\omega\mathbf{P}_A + (1-\omega)\mathbf{P}_B\right)^{-1}\right)^{-1}. \end{aligned} \tag{11}$$

The parameter $\omega \in [0,1]$ can also be optimized to minimize quantities such as the trace or determinant of $\mathbf{P}_{\mathrm{ICI}}$. We denote

$$(\hat{\boldsymbol{\theta}}_{\mathrm{ICI}}, \mathbf{P}_{\mathrm{ICI}}) = \mathtt{ICI}\left((\hat{\boldsymbol{\theta}}_A, \mathbf{P}_A), (\hat{\boldsymbol{\theta}}_B, \mathbf{P}_B)\right) \tag{12}$$

as the function that computes and returns $(\hat{\boldsymbol{\theta}}_{\mathrm{ICI}}, \mathbf{P}_{\mathrm{ICI}})$ using (11).

5. Estimate Diffusion

In this section, we will consider an alternative scheme to measurement diffusion, which we call *estimate diffusion*. In this scheme, after a new measurement has been collected at time step k, each node m will: (1) broadcast its own measurement $z_{m,k}$ and receive measurements $z_{n,k}$ from neighbours $n \in \mathcal{N}_m$, (2) form a local pre-estimate $\hat{\boldsymbol{\beta}}_k^{(m-)}$ (we follow the terminology of [47]) and (approximate) Hessian $H_k^{(m-)}$ using the received measurements, (3) broadcast $\left(\hat{\boldsymbol{\beta}}_k^{(m-)}, H_k^{(m-)}\right)$ and receive $\left(\hat{\boldsymbol{\beta}}_k^{(n-)}, H_k^{(n-)}\right)$ to/from neighbours $n \in \mathcal{N}_m$, (4) form an updated estimate $\hat{\boldsymbol{\beta}}_k^{(m)}$ using the received estimates and Hessians. For example, suppose we again have the sensor network shown in Figure 1, and concentrate on node 2. At time k, node 2 will broadcast $z_{2,k}$ to nodes 1, 3, and 5. After receiving $z_{1,k}$, $z_{3,k}$, and $z_{5,k}$ from nodes 1, 3, and 5, respectively, node 2 computes and broadcasts the pre-estimate $\hat{\boldsymbol{\beta}}_k^{(2-)}$ and Hessian $H_k^{(2-)}$. After receiving the broadcasts of its neighbours, node 2 will then combine $\hat{\boldsymbol{\beta}}_k^{(1-)}$, $\hat{\boldsymbol{\beta}}_k^{(2-)}$, $\hat{\boldsymbol{\beta}}_k^{(3-)}$, $\hat{\boldsymbol{\beta}}_k^{(5-)}$, $H_k^{(1-)}$, $H_k^{(2-)}$, $H_k^{(3-)}$, $H_k^{(5-)}$ in order to form $\hat{\boldsymbol{\beta}}_k^{(2)}$. In contrast to measurement diffusion, there are now two transmissions between successive time instances (Steps 1 and 3), which is similar to [47,48]. One could remove the broadcast of measurements in Step 1 and compute $\hat{\boldsymbol{\beta}}_k^{(m-)}$ using only $z_{m,k}$; however, we found that this degrades the estimation performance significantly.

The formal description of the procedure that is run at each node m is given as Algorithm 2. Similar to Algorithm 1, a forgetting factor δ is used to allow time-varying fields to be estimated. Below, we will provide additional details and explanations of the individual steps.

Algorithm 2 Field estimation at node m using estimate diffusion

1: **Algorithm Parameters**: Logistic function parameter $\eta > 0$, Levenberg–Marquardt parameter $\varsigma > 0$, forgetting factor $\delta \in (0, 1]$, FusionMethod
2: **Outputs**: Parameter estimates $\{\hat{\boldsymbol{\beta}}_k^{(m)}\}$
3: Initialize $\tilde{H} = \mathbf{0}$ and $\hat{\boldsymbol{\beta}}_0^{(m)}$
4: **for** $k = 1, 2, \ldots,$ **do**
5: Collect a measurement $z_{m,k}$
6: Broadcast $(m, z_{m,k})$ to neighbours $\mathcal{N}_m$
7: Receive measurements $\{(n, z_{n,k}) : n \in \mathcal{N}_m\}$ from neighbours
8: Set $\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}_{k-1}^{(m)}$
9: **for** $n \in \{m\} \cup \mathcal{N}_m$ **do**
10: $G = \nabla g_n(\tilde{\boldsymbol{\beta}})$ where $\nabla g_n(.)$ is computed using (7)
11: $\tilde{H} \leftarrow \delta \tilde{H} + \nabla^2 g_n(\tilde{\boldsymbol{\beta}})$ where $\nabla^2 g_n(.)$ is computed using (8)
12: $H = \tilde{H} + \varsigma I$
13: $\tilde{\boldsymbol{\beta}} \leftarrow \tilde{\boldsymbol{\beta}} - H^{-1} G$
14: **end for**
15: Update pre-estimate $\hat{\boldsymbol{\beta}}_k^{(m-)} = \tilde{\boldsymbol{\beta}}$ and $H_k^{(m-)} = H$
16: Broadcast $\left(\hat{\boldsymbol{\beta}}_k^{(m-)}, H_k^{(m-)}\right)$ to neighbours $\mathcal{N}_m$
17: Receive $\left\{\left(\hat{\boldsymbol{\beta}}_k^{(n-)}, H_k^{(n-)}\right) : n \in \mathcal{N}_m\right\}$ from neighbours
18: **if** FusionMethod is CovarianceIntersection **then**
19: Set $\omega_n = 1/(|\mathcal{N}_m| + 1)$ and compute $\mathbf{P}_{\mathrm{CI}} = \Big(\sum_{n \in \{m\} \cup \mathcal{N}_m} \omega_n H_k^{(n-)}\Big)^{-1}$
20: Update $\hat{\boldsymbol{\beta}}_k^{(m)} = \mathbf{P}_{\mathrm{CI}} \sum_{n \in \{m\} \cup \mathcal{N}_m} \omega_n H_k^{(n-)} \hat{\boldsymbol{\beta}}_k^{(n-)}$ and $\tilde{H} = \mathbf{P}_{\mathrm{CI}}^{-1} - \varsigma I$
21: **else if** FusionMethod is InverseCovarianceIntersection **then**
22: Set $\omega = 1/2$ and $(\hat{\boldsymbol{\beta}}_A, \mathbf{P}_A) = \left(\hat{\boldsymbol{\beta}}_k^{(m-)}, \left(H_k^{(m-)}\right)^{-1}\right)$
23: **for** $n \in \mathcal{N}_m$ **do**
24: Set $(\hat{\boldsymbol{\beta}}_B, \mathbf{P}_B) = \left(\hat{\boldsymbol{\beta}}_k^{(n-)}, \left(H_k^{(n-)}\right)^{-1}\right)$
25: $(\hat{\boldsymbol{\beta}}_A, \mathbf{P}_A) \leftarrow \texttt{ICI}\big((\hat{\boldsymbol{\beta}}_A, \mathbf{P}_A), (\hat{\boldsymbol{\beta}}_B, \mathbf{P}_B)\big)$ using (12)
26: **end for**
27: Update $\hat{\boldsymbol{\beta}}_k^{(m)} = \hat{\boldsymbol{\beta}}_A$ and $\tilde{H} = \mathbf{P}_A^{-1} - \varsigma I$
28: **end if**
29: **end for**

Step 1 involves a broadcast by node m of a *single* measurement $z_{m,k}$ to its neighbours, together with the node ID m, to allow recipients to determine where the measurement was taken.

Step 2 involves using the collected and received measurements $\{z_{m,k}\} \cup \{z_{n,k} : n \in \mathcal{N}_m\}$ to form a local pre-estimate $\hat{\boldsymbol{\beta}}_k^{(m-)}$ and Hessian $H_k^{(m-)}$. Similar to Section 3, this is performed using an approximate Newton update on the received measurements, shown in lines 10–13 of Algorithm 2.

Step 3 involves broadcasts of $\left(\hat{\boldsymbol{\beta}}_k^{(m-)}, H_k^{(m-)}\right)$ to neighbours. $\hat{\boldsymbol{\beta}}_k^{(m-)}$ is a vector of p numbers (recall p is the number of radial basis functions), while $H_k^{(m-)}$ is a $p \times p$ matrix, where the number of basis functions p is usually chosen to be less than the number of nodes M.

Step 4 combines the pre-estimates $\{\hat{\boldsymbol{\beta}}_k^{(m-)}\} \cup \{\hat{\boldsymbol{\beta}}_k^{(n-)} : n \in \mathcal{N}_m\}$ and Hessians to form the estimate $\hat{\boldsymbol{\beta}}_k^{(m)}$, as shown in lines 18 or 21 of Algorithm 2. It does this by using one of the fusion methods in Section 4, i.e., covariance intersection or inverse covariance intersection. In the case of covariance intersection, the batch form given in (10) is used, which is less computationally intensive than combining two estimates at a time using (9) in a sequential manner, while having similar estimation performance. In the case of inverse covariance intersection, the estimates are combined two at a time using (11). (Various ways to extend inverse covariance intersection to a batch form have been proposed

in [56,57]; however, we have been unable to get these methods to work in our situation). Note that (10) and (11) require error covariances, while the approximate online Newton method computes Hessians. In this paper, we will use the inverse Hessians as a surrogate for the covariances, which is motivated by the result that the Hessian matrix of the negative log-likelihood with respect to the parameters is approximately equal to the inverse of the covariance matrix [58], with equality holding for Gaussian random vectors [59]. For simplicity, we have set $\omega_n = 1/(|\mathcal{N}_m| + 1), \forall n \in \{m\} \cup \mathcal{N}_m$ for batch covariance intersection and $\omega = 1/2$ when combining two estimates sequentially in inverse covariance intersection. As mentioned in Section 4, one could also try to optimize ω_n or ω; however, this will increase the computational load of the algorithm. Furthermore, we note that the estimation diffusion scheme is actually agnostic to the particular estimate fusion method, and other consistent fusion rules can also be used. Optimizing ω_n/ω and the choice of fusion method will be a topic for future investigation.

6. Discussions and Extensions

In this section, we will first compare the communication requirements of the two proposed schemes, and a possible way to reduce communications for measurement diffusion. We then briefly discuss how the approximate Newton update can be run in batch form, and how the algorithms can be extended to handle sensor heterogeneity.

6.1. Communication Requirements

In measurement diffusion, at each time step k, each node m broadcasts $\mathcal{C}_{m,k}^{\text{latest}}$ given in (4), which in general is a set containing $3M$ numbers, which could be large when the number of nodes M in the network is large. One possible way to reduce communications is as follows. If $z_{n,k_n} = z_{n,k_n-1}$, i.e., the quantized measurement at a particular node has not changed at the next time step, then the sending of (n, k_n, z_{n,k_n}) when broadcasting $\mathcal{C}_{m,k}^{\text{latest}}$ can be eliminated, as the neighbours of node m can reconstruct this information. Note, however, that this requires that measurements are collected at every time step and there are no transmission losses, so that errors do not propagate.

For estimate diffusion, Step 1 involves broadcasting of two numbers. In Step 3, the pre-estimate is a vector of p numbers, while the Hessian is a symmetric $p \times p$ matrix with $\frac{p(p+1)}{2}$ unique entries in general. Thus, $\frac{p(p+1)}{2} + p + 2$ numbers are broadcast by each node at each time step for estimate diffusion. Note that the number of basis functions p will affect the quality of the field estimates, with larger values generally allowing us to capture finer field structure, but also requiring more measurements for accurate estimates [46]. For relatively simple fields, such as those considered in [15,16], one can use values as low as $p = 16$; however, for more complicated fields one may need to use values such as $p = 64$ or even higher (see Section 7), which will significantly increase the communication requirements.

6.2. Batch Newton Update

In Algorithm 1, at each k, an approximate Newton step is run for each new measurement in $\mathcal{C}_{m,k}^{\text{new}}$. In particular, in line 14, the operation $H^{-1}G$ needs to be carried out $|\mathcal{C}_{m,k}^{\text{new}}|$ times. A computationally less intensive alternative is to replace lines 9–16 with Algorithm 3, which computes a single approximation of the gradient and Hessian, and then performs a single approximate Newton step, on each batch of measurements in a partition of $\mathcal{C}_{m,k}^{\text{new}}$. This reduces the number of times the operation $H^{-1}G$ needs to be carried out to around $\lceil |\mathcal{C}_{m,k}^{\text{new}}|/N_{\text{batch}} \rceil$, where N_{batch} is the batch size. In simulations, we have found this batch update to also perform well, provided N_{batch} is not too large, e.g., less than 100. Estimate diffusion (Algorithm 2) can also be modified to use a batch Newton update in a similar fashion.

Algorithm 3 Batch Newton update

Partition $\mathcal{C}_{m,k}^{\text{new}} = \bigcup_i \mathcal{C}_i^{\text{batch}}$ into batches $\mathcal{C}_i^{\text{batch}}$ of size $\leq N_{\text{batch}}$, and set $\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}_{k-1}^{(m)}$
for each batch $\mathcal{C}_i^{\text{batch}}$ **do**
  Set $G = \mathbf{0}$
  for $(n, k_n, z_{n,k_n}) \in \mathcal{C}_i^{\text{batch}}$ **do**
    $G \leftarrow G + \nabla g_n(\tilde{\boldsymbol{\beta}})$ where $\nabla g_n(.)$ is computed using (7)
    $\tilde{H} \leftarrow \delta \tilde{H} + \nabla^2 g_n(\tilde{\boldsymbol{\beta}})$ where $\nabla^2 g_n(.)$ is computed using (8)
  end for
  $H = \tilde{H} + \varsigma I$
  $\tilde{\boldsymbol{\beta}} \leftarrow \tilde{\boldsymbol{\beta}} - H^{-1} G$
end for
Update $\hat{\boldsymbol{\beta}}_k^{(m)} = \tilde{\boldsymbol{\beta}}$

6.3. Heterogeneous Sensors

In this paper, we have assumed that the sensors are homogeneous, with identical capabilities. In some situations, it is more appropriate to assume heterogeneous sensors, e.g., there could be many cheap sensors with limited capabilities, plus a small number of more expensive but more capable sensors.

Suppose we consider a type of heterogeneity with regard to sensors having different quantization thresholds (number of thresholds and their values). The measurement diffusion scheme can be extended to this situation, provided each sensor knows the quantization thresholds of every other sensor in the network. While the estimate diffusion scheme can also be extended to this type of heterogeneity if each sensor knows the quantization thresholds of its direct neighbours only.

7. Numerical Studies

In this section, we will first describe the measures for performance evaluation of our algorithms in Section 7.1, and a centralized scheme for baseline comparison in Section 7.2, before presenting numerical examples in Sections 7.3–7.5. The simulations in this section are written in Python and run on an Intel Core i7 9700 PC with 16 GB of memory.

7.1. Performance Criteria

In this paper, we will consider two performance measures, the mean square error (MSE), and the structural similarity (SSIM) index [60]. The MSE is defined as

$$\text{MSE} \triangleq \frac{1}{|\mathcal{R}_d|} \sum_{\mathbf{x} \in \mathcal{R}_d} \Big(\phi(\mathbf{x}) - \sum_{i=1}^{p} \hat{\beta}_i K_i(\mathbf{x}) \Big)^2,$$

where $\phi(\mathbf{x})$ is the true field value at position $\mathbf{x}$, and $\mathcal{R}_d$ is a discretized set of points in the region $\mathcal{R}$.

The structural similarity index is a measure of the similarity between two images, first introduced in [60]. Suppose the discretized set of points $\mathcal{R}_d$ are located on a rectangular grid. We can then regard $\Phi = \{\phi(\mathbf{x}) : \mathbf{x} \in \mathcal{R}_d\}$ and $\hat{\Phi} = \{\hat{\phi}(\mathbf{x}) : \mathbf{x} \in \mathcal{R}_d\} = \{\sum_{i=1}^{p} \hat{\beta}_i K_i(\mathbf{x}) : \mathbf{x} \in \mathcal{R}_d\}$ as the image representations of the true and estimated fields, respectively, and compute the SSIM between these two images. The SSIM gives a scalar value between 0 and 1, with SSIM $= 1$ if the two images to be compared are identical. The specific equations to compute the SSIM can be found in [60]; see also [61].

7.2. Centralized Scheme

For comparison with our proposed algorithms, we also consider a centralized scheme, where at each time instant k, all of the measurements $z_{m,k}, m = 1, \ldots, M$ are transmitted to a central node or fusion centre which then performs the estimation. For large networks, this could involve information from some nodes needing to be transmitted over large distances, requiring either high transmission power or multi-hop communications. Nevertheless, the centralized scheme will serve as a useful benchmark on achievable performance. Estimation of $\boldsymbol{\beta}$ using the centralized scheme will be performed using the approximate Newton update, as in lines 11–14 of Algorithm 1, but now with all of the measurements $z_{m,k}, m = 1, \ldots, M$ collected at time k being used to update $\hat{\boldsymbol{\beta}}_k$.

7.3. Example 1: Effect of Number of Sensors on Estimated Field

In the process of forming a sensor network, knowing the minimum number of sensors needed to estimate fields to an acceptable level of accuracy could aid in reducing costs and network complexity. This subsection studies the effect that varying the number of sensors used can have on the estimated fields. We concentrate on the centralized scheme described in Section 7.2 above, in order to focus on the best performance achievable for a given number of sensors.

We consider a randomly generated example true field shown in Figure 2, in the region of interest $\mathcal{R} = [0, 1000] \times [0, 1000]$. Sensor measurements are noisy, with the measurement noise assumed to be Gaussian with zero mean and variance 0.1. The quantizer is a four-level quantizer, with quantizer thresholds $\tau = [1, 2, 3]$. For approximating the estimated fields, $p = 64$ basis functions were used with c_i spaced uniformly in a square grid within $\mathcal{R}$, with $\sigma_i = 1000/8 = 125$ for $i = 1, \ldots, p$. The logistic function parameter was chosen as $\eta = 5$. The Levenberg–Marquardt parameter was set to $\varsigma = 10$, while the forgetting factor was set to $\delta = 1$. The initial estimate $\hat{\beta}_0$ was initialized by randomly choosing each component from a uniform distribution between 0 and 1.

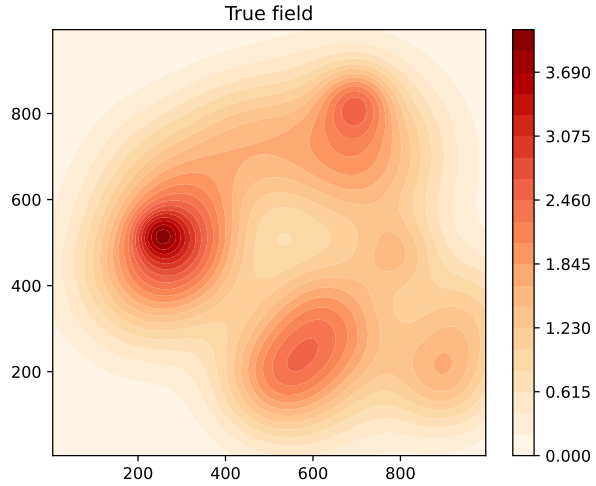


Figure 2. Example 1: true field.

Figure 3 depicts the estimated fields for the centralized scheme after 10 time steps, where the sensor number M ranges from 100 to 900 in increments of 100. The sensor locations were randomly generated, and shown as light blue dots in Figure 3. We see that a sensor number of 200–300 is already able to produce estimated fields to a reasonable level of accuracy, with the estimate reliability generally increasing as the number of sensors is increased.

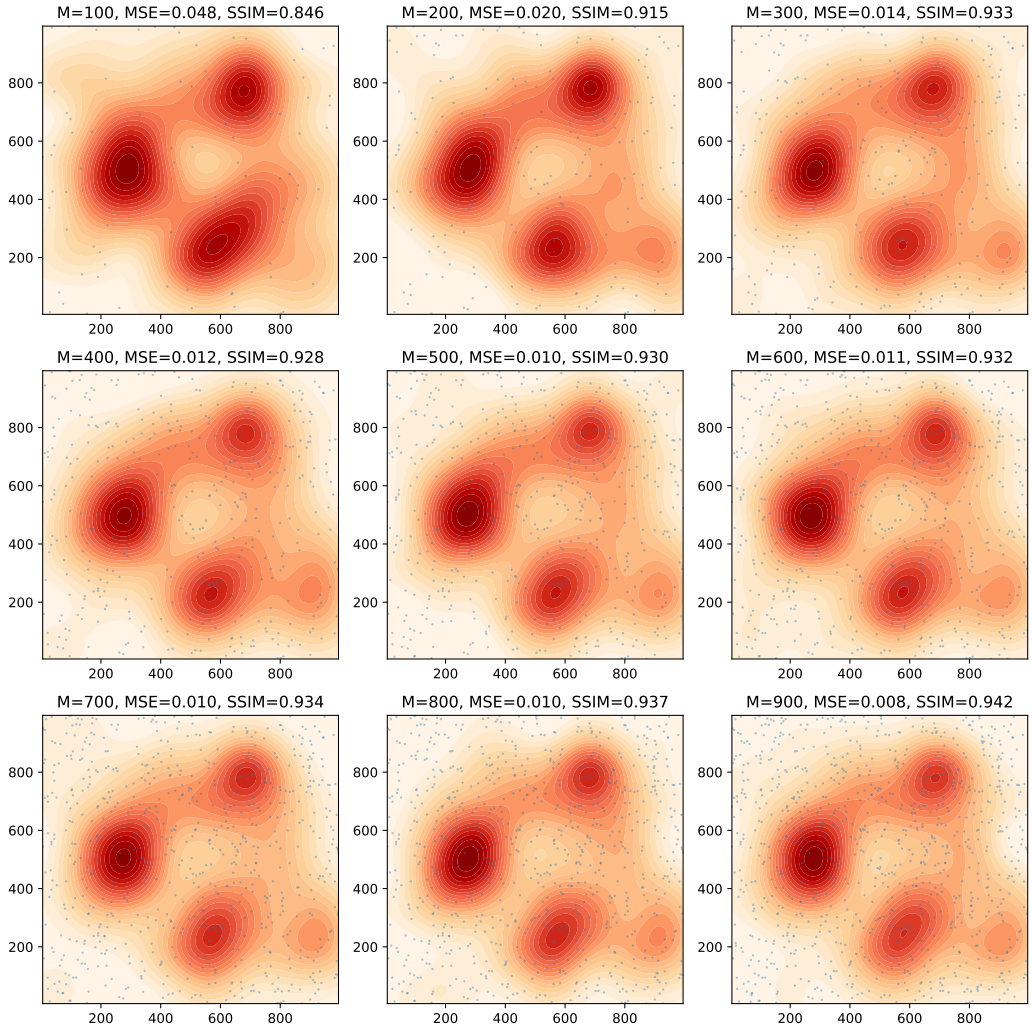


Figure 3. Example 1: estimated fields with varying numbers of sensors for the centralized scheme. The light blue dots represent the sensor locations.

7.4. Example 2: Estimation of Time-Varying Fields

For fields which are constantly evolving in real-time, it is crucial that the sensor system is able to adapt to these changing conditions without significant time delay. This subsection explores how each of the field estimation algorithms proposed in this paper responds to time-varying fields.

For modelling the time-varying fields, we consider the six fields shown in Figure 4, where each field will be held constant for 50 time steps before switching to the next field. We will consider the sensor network shown in Figure 5, consisting of 500 randomly placed nodes. The communication range is such that two sensors within a distance of $1000/12 = 83.33$ can communicate directly with each other. For this communication range and the chosen set of nodes, it was found that the network was connected. (In general, a larger communication range will make a network more likely to be connected; however, it will also require more transmission power). In this subsection, we will present field estimation results at the node represented by the large orange dot in Figure 5. The maximum number of hops $L(m,n)$ (recall the notation in Section 3) between the chosen node m and any other node n in this network can be found to be 13.

In the field estimation algorithms, the Levenberg–Marquardt parameter is set to $\varsigma = 10$ for the centralized and measurement diffusion schemes, and $\varsigma = 1$ for the estimate diffusion scheme. For time-varying fields the forgetting factor is now set to $\delta = 0.998$ for centralized

and measurement diffusion, and $\delta = 0.99$ for estimate diffusion. Other parameters were the same as those in Section 7.3.

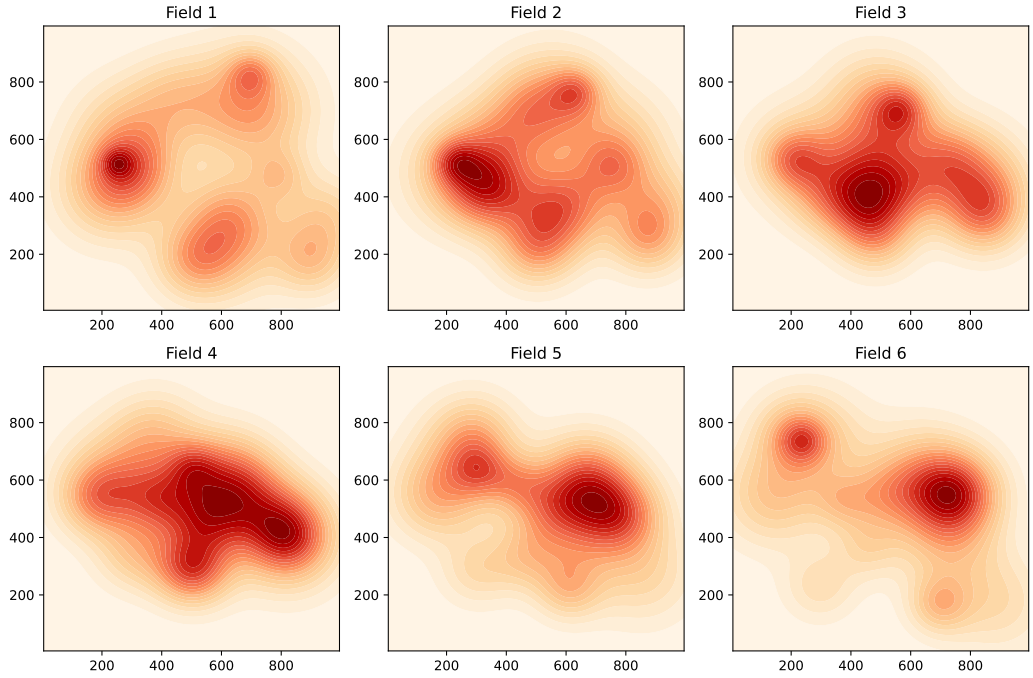


Figure 4. Example 2: time-varying fields.

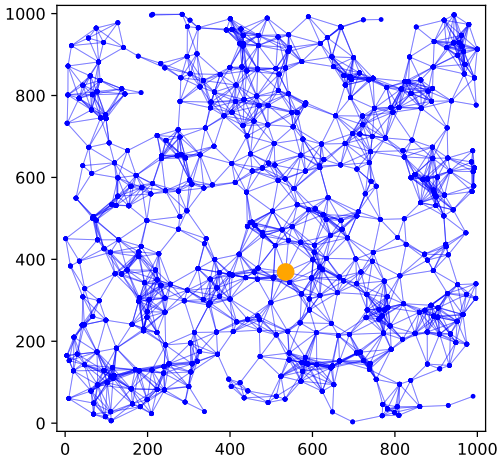


Figure 5. Example 2: sensor network.

Figures 6–9 plot the MSE and SSIM over time for centralized, measurement diffusion, estimate diffusion using covariance intersection, and estimate diffusion using inverse covariance intersection. As stated before, the fields change every 50 time steps, although we stress that the algorithms themselves do not know when the field changes occur. For the centralized scheme, the field estimate quality drops (MSE increases while SSIM decreases) immediately after the field changes, but after just 1–2 time steps (note that 500 measurements, equal to the number of sensors in the network, are collected at every time step) is able to quickly adapt to the new field. For the measurement diffusion scheme, adaptation to the field changes takes a bit longer, with the number of time steps needed to converge to a steady state roughly equal to 13, the maximum number of hops from the chosen node to any other node in the network. Comparing Figure 7 with Figure 6, we see that the MSE and SSIM values of measurement diffusion at steady state are close to that of centralized

estimation. Estimate diffusion is also able to adapt to field changes, although at a slower rate than measurement diffusion, with inverse covariance intersection performing better than covariance intersection and also able to achieve steady state performance close to that of centralized estimation.

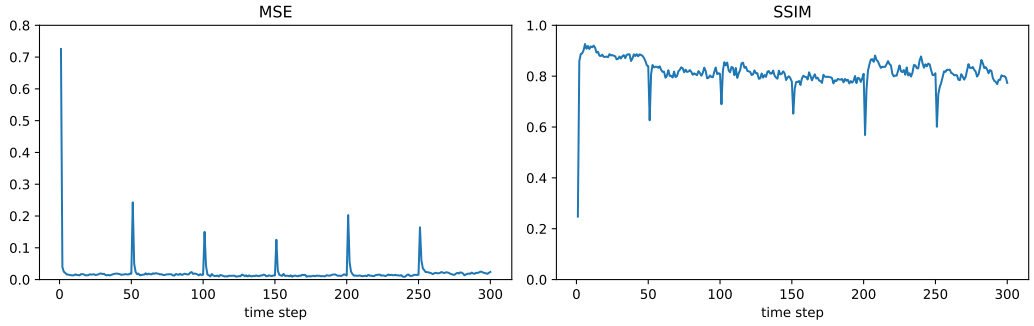


Figure 6. Example 2: centralized.

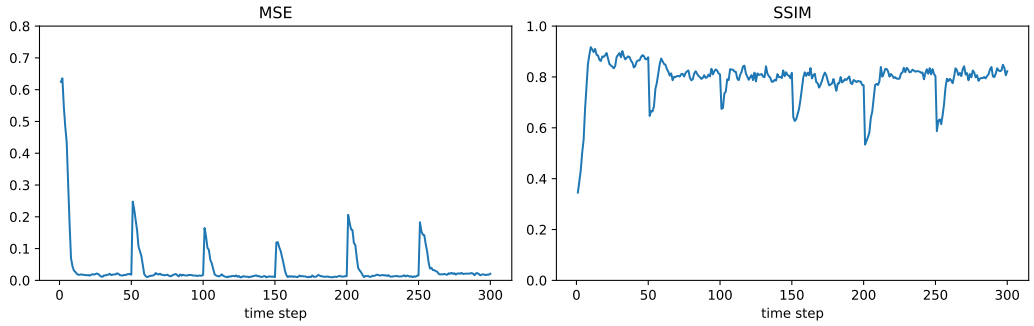


Figure 7. Example 2: measurement diffusion.

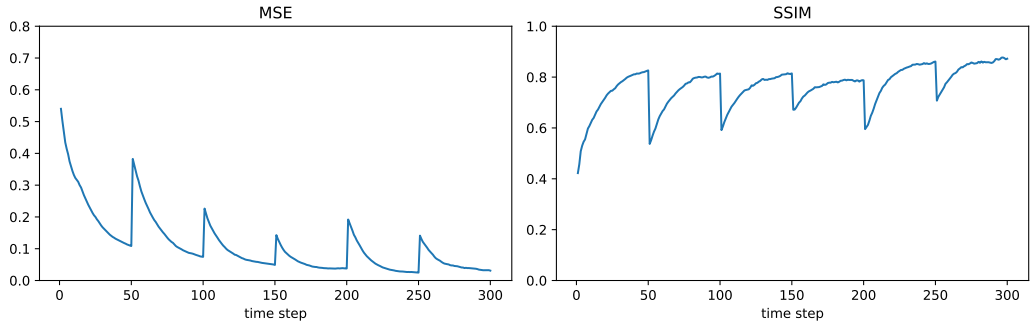


Figure 8. Example 2: estimate diffusion using covariance intersection.

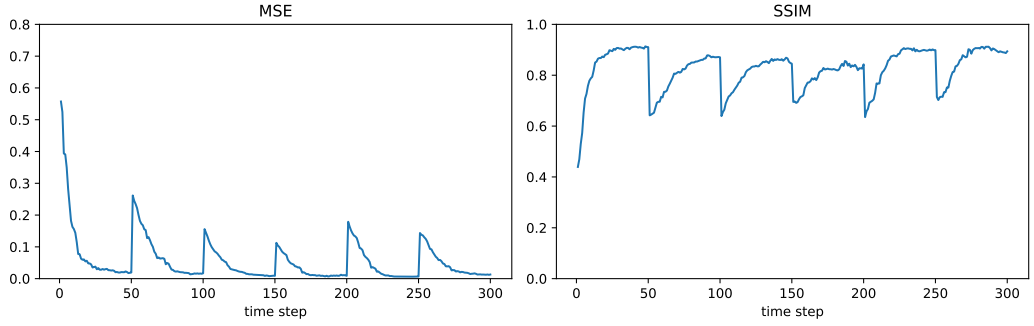


Figure 9. Example 2: estimate diffusion using inverse covariance intersection.

The delay in adapting to field changes is due to our assumption that each sensor communicates with its neighbours once (or twice for estimate diffusion) per time step, in order to limit the amount of communication. Whether this delay is tolerable for real time application will depend on the particular situation. Delay can be reduced if we allow for multi-hop communication within each time step, but this will increase the communication requirements. Thus, there exists a trade-off between delay and the amount of communication.

7.5. Example 3: Studies on a Real Dataset

We now consider the testing of our algorithms on a publicly available dataset provided by the European Environment Agency [62]. This dataset provides interpolated average concentration data in $\mu g/m^3$ for PM10 (particulate matter 10 µm or less in diameter) air pollutants in Europe during the year 2021. Figure 10 shows a plot of the available data.

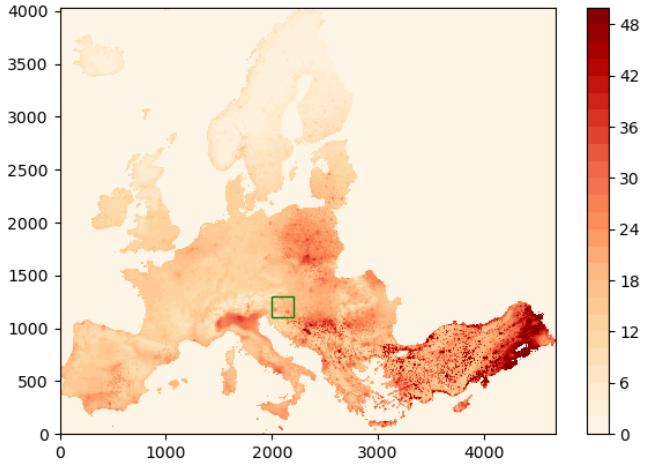


Figure 10. Example 3: average concentration data for PM10 air pollutants in Europe during the year 2021.

For testing of our field estimation algorithms, we consider the 200 km × 200 km area bounded by the green box in Figure 10. A close-up of this area is shown in Figure 11. We regard this as the true field that is to be estimated.

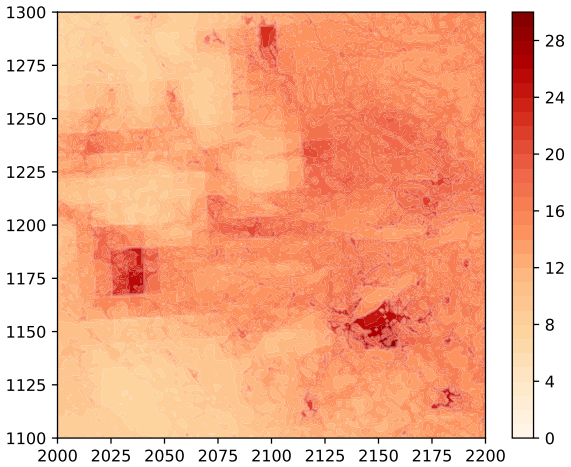


Figure 11. Example 3: close-up of the area bounded by the green box in Figure 10.

We use a sensor network consisting of 500 randomly placed sensors, a plot of which is shown in Figure 12. For this network, the maximum number of hops from the orange node to any other node is 19. We assume that the sensor measurements are noisy, with the noise being Gaussian of zero mean and variance 1, with the quantizer thresholds being $\tau = [5, 10, 15, 20, 25]$. The number of basis functions used is $p = 15 \times 15 = 225$, with c_i uniformly spaced in a square grid within the area and with $\sigma_i = 200/15 = 13.33$.

We will consider results for measurement diffusion and estimate diffusion using inverse covariance intersection, with logistic function parameter $\eta = 5$, and forgetting factor set to $\delta = 1$. For the orange node shown in Figure 12, Figure 13 plots the MSE and SSIM over time for the two schemes. For comparison, results for the centralized scheme are also plotted. Similar to Section 7.4, the performance of measurement diffusion reaches a steady state after around 19 time steps, which is the maximum number of hops from the orange node to any other node. While for estimate diffusion, the number of time steps needed to reach steady state is longer. For both schemes, we again see that the MSE and SSIM values at steady state are close to that of centralized estimation. The estimated fields at this node after 50 time steps are shown in Figure 14. We can see that qualitatively the estimated fields for both schemes look similar to smoothed versions of Figure 11.

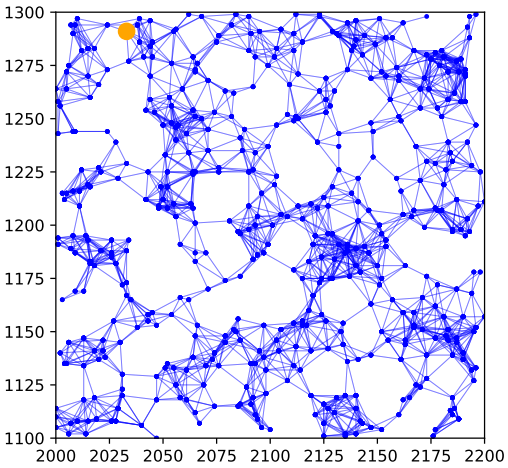


Figure 12. Example 3: sensor network.

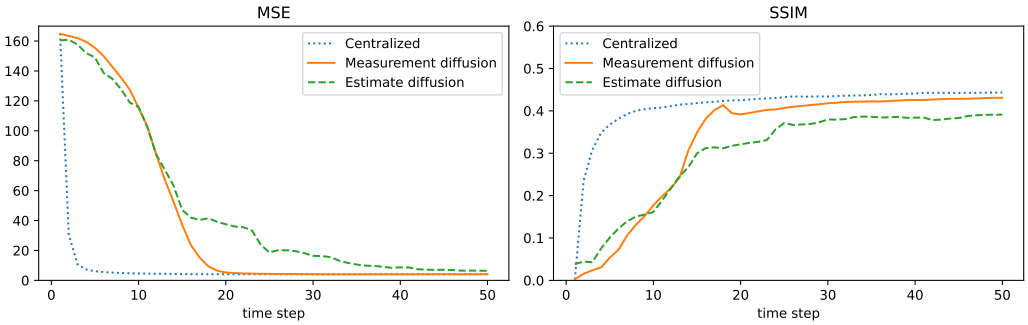


Figure 13. Example 3: performance comparison.

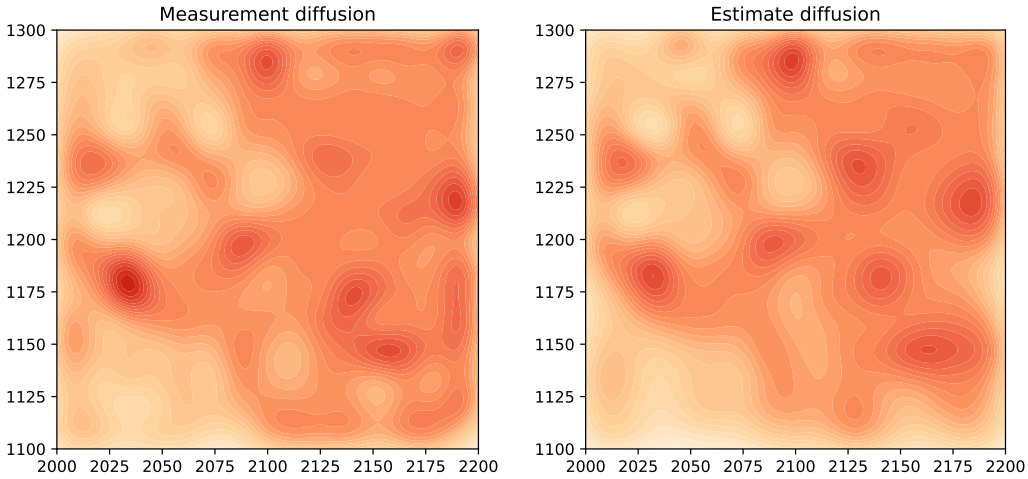


Figure 14. Example 3: estimated fields after 50 time steps.

8. Conclusions

This paper has studied the distributed estimation of scalar fields using a sensor network. Motivated by the possibility of utilizing large numbers of low cost sensors, we have considered the situation where the sensors have coarsely quantized measurements. We have proposed novel methods for distributed field estimation, which involve sharing either measurements or estimates between neighbouring sensors, and can adapt promptly to time-varying fields. Numerical studies have shown that our schemes can achieve steady state performance close to that of centralized estimation. Areas of future work include studying the impact of imperfect knowledge of sensor locations, and investigation of the use of both fixed and mobile sensors in field estimation.

Author Contributions: Conceptualization, A.S.L.; methodology, C.J. and A.S.L.; software, C.J.; validation, C.J.; investigation, C.J. and A.S.L.; resources, A.T.S., D.J.N. and O.I.; writing—original draft preparation, C.J. and A.S.L.; writing—review and editing, A.T.S., D.J.N. and O.I.; supervision, A.S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ristic, B.; Morelande, M.; Gunatilaka, A. Information driven search for point sources of gamma radiation. *Signal Process.* **2010**, *90*, 1225–1239. [CrossRef]
2. Neumann, P.P.; Hernandez Bennetts, V.; Lilienthal, A.J.; Bartholmai, M.; Schiller, J.H. Gas source localization with a micro-drone using bio-inspired and particle filter-based algorithms. *Adv. Robot.* **2013**, *27*, 725–738. [CrossRef]
3. Ristic, B.; Skvortsov, A.; Gunatilaka, A. A study of cognitive strategies for an autonomous search. *Inf. Fusion* **2016**, *28*, 1–9. [CrossRef]
4. Newaz, A.A.R.; Jeong, S.; Lee, H.; Ryu, H.; Chong, N.Y. UAV-based multiple source localization and contour mapping of radiation fields. *Robot. Auton. Syst.* **2016**, *85*, 12–25. [CrossRef]
5. Hutchinson, M.; Oh, H.; Chen, W.H. A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors. *Inf. Fusion* **2017**, *36*, 130–148. [CrossRef]
6. Selvaratnam, D.D.; Shames, I.; Dimarogonas, D.V.; Manton, J.H.; Ristic, B. Co-operative Estimation for Source Localisation using Binary Sensors. In Proceedings of the IEEE Conference on Decision and Control, Melbourne, Australia, 12–15 December 2017; pp. 1572–1577.

7. Hutchinson, M.; Liu, C.; Chen, W.H. Source term estimation of a hazardous airborne release using an unmanned aerial vehicle. *J. Field Robot.* **2019**, *36*, 797–817. [CrossRef]
8. Li, D.; Chen, F.; Wang, Y.; Wang, X. Implementation of a UAV-sensory-system-based hazard source estimation in a chemical plant cluster. *Proc. IOP Conf. Ser.* **2019**, *291*, 012043. [CrossRef]
9. Park, M.; Oh, H. Cooperative information-driven source search and estimation for multiple agents. *Inf. Fusion* **2020**, *54*, 72–84. [CrossRef]
10. Ristic, B.; Gilliam, C.; Moran, W.; Palmer, J.L. Decentralised multi-platform search for a hazardous source in a turbulent flow. *Inf. Fusion* **2020**, *58*, 13–23. [CrossRef]
11. La, H.M.; Sheng, W. Distributed Sensor Fusion for Scalar Field Mapping Using Mobile Sensor Networks. *IEEE Trans. Cybern.* **2013**, *43*, 766–778. [CrossRef]
12. La, H.M.; Sheng, W.; Chen, J. Cooperative and Active Sensing in Mobile Sensor Networks for Scalar Field Mapping. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *45*, 1–12. [CrossRef]
13. Shaw, J.T.; Shah, A.; Yong, H.; Allen, G. Methods for quantifying methane emissions using unmanned aerial vehicles: A review. *Phil. Trans. R. Soc. A* **2021**, *379*, 20200450. [CrossRef]
14. Razak, R.A.; Sukumar, S.; Chung, H. Scalar Field Estimation with Mobile Sensor Networks. *Int. J. Robust Nonlinear Control* **2021**, *31*, 4287–4305. [CrossRef]
15. Leong, A.S.; Zamani, M. Field Estimation Using Binary Measurements. *Signal Process.* **2022**, *194*, 108430. [CrossRef]
16. Leong, A.S.; Zamani, M.; Shames, I. A Logistic Regression Approach to Field Estimation Using Binary Measurements. *IEEE Signal Process. Lett.* **2022**, *29*, 1848–1852. [CrossRef]
17. Tran, V.P.; Garratt, M.A.; Kasmarik, K.; Anavatti, S.G.; Leong, A.S.; Zamani, M. Multi-gas source localization and mapping by flocking robots. *Inf. Fusion* **2023**, *91*, 665–680. [CrossRef]
18. Available online: https://www.chironix.com/robots-all (accessed on 14 July 2024).
19. Pounds, P.; Singh, S. Samara: Biologically inspired self-deploying sensor networks. *IEEE Potentials* **2015**, *34*, 10–15. [CrossRef]
20. Iyer, V.; Gaensbauer, H.; Daniel, T.L.; Gollakota, S. Wind dispersal of battery-free wireless devices. *Nature* **2022**, *603*, 427–433. [CrossRef]
21. Robins, P.; Rapley, V.; Thomas, P. A Probabilistic Chemical Sensor Model for Data Fusion. In Proceedings of the International Conference on Information Fusion, Philadelphia, PA, USA, 25–28 July 2005; pp. 1116–1122.
22. Cheng, Y.; Konda, U.; Singh, T.; Scott, P. Bayesian Estimation for CBRN Sensors with Non-Gaussian Likelihood. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 684–701. [CrossRef]
23. Zhu, J.; Lin, X.; Blum, R.S.; Gu, Y. Parameter Estimation From Quantized Observations in Multiplicative Noise Environments. *IEEE Trans. Signal Process.* **2015**, *63*, 4037–4050. [CrossRef]
24. Li, A.; Masouros, C.; Swindlehurst, A.L.; Yu, W. 1-Bit Massive MIMO Transmission: Embracing Interference with Symbol-Level Precoding. *IEEE Comm. Mag.* **2021**, *59*, 121–127. [CrossRef]
25. Leong, A.S.; Weyer, E.; Nair, G.N. Identification of FIR Systems with Binary Input and Output Observations. *IEEE Trans. Autom. Control* **2021**, *66*, 1190–1198. [CrossRef]
26. Fu, M.; Xie, L. Finite-Level Quantized Feedback Control for Linear Systems. *IEEE Trans. Autom. Control* **2009**, *54*, 1165–1170.
27. Widrow, B.; Kollár, I. *Quantization Noise*; Cambridge University Press: Cambridge, UK, 2008.
28. Nowak, R.; Mitra, U.; Willett, R. Estimating Inhomogeneous Fields Using Wireless Sensor Networks. *IEEE J. Sel. Areas Commun.* **2004**, *22*, 999–1006. [CrossRef]
29. Zhang, H.; Moura, J.M.F.; Krogh, B. Dynamic Field Estimation Using Wireless Sensor Networks: Tradeoffs between Estimation Error and Communication Cost. *IEEE Trans. Signal Process.* **2009**, *57*, 2383–2395. [CrossRef]
30. Morelande, M.R.; Skvortsov, A. Radiation field estimation using a Gaussian mixture. In Proceedings of the International Conference on Information Fusion, Seattle, WA, USA, 6–9 July 2009; pp. 2247–2254.
31. Cortés, J. Distributed Kriged Kalman Filter for Spatial Estimation. *IEEE Trans. Autom. Control* **2009**, *54*, 2816–2827. [CrossRef]
32. van Waterschoot, T.; Leus, G. Static Field Estimation Using a Wireless Sensor Network Based on the Finite Element Method. In Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, San Juan, PR, USA, 13–16 December 2011; pp. 369–372.
33. Bergamo, Y.P.; Lopes, C.G. Scalar Field Estimation Using Adaptive Networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Kyoto, Japan, 25–30 March 2012; pp. 3565–3568.
34. Nevat, I.; Peters, G.W.; Collings, I.B. Random Field Reconstruction with Quantization in Wireless Sensor Networks. *IEEE Trans. Signal Process.* **2013**, *61*, 6020–6033. [CrossRef]
35. Ramachandran, R.K.; Berman, S. The Effect of Communication Topology on Scalar Field Estimation by Large Networks with Partially Accessible Measurements. In Proceedings of the American Control Conference, Seattle, WA, USA, 24–26 May 2017; pp. 3886–3893.
36. Battistelli, G.; Chisci, L.; Forti, N.; Gherardini, S. MAP moving horizon estimation for threshold measurements with application to field monitoring. *Int. J. Adapt. Control Signal Process.* **2019**, *34*, 1–16. [CrossRef]
37. Park, J.; Sandberg, I.W. Approximation and Radial-Basis-Function Networks. *Neural Comput.* **1993**, *5*, 305–316. [CrossRef]
38. Lesage-Landry, A.; Taylor, J.A.; Shames, I. Second-order Online Nonconvex Optimization. *IEEE Trans. Autom. Control* **2021**, *66*, 4866–4872. [CrossRef]

39. Julier, S.J.; Uhlmann, J.K. A Non-divergent Estimation Algorithm in the Presence of Unknown Correlations. In Proceedings of the American Control Conference, Albuquerque, NM, USA, 4–6 June 1997; pp. 2269–2373.
40. Noack, B.; Sijs, J.; Reinhardt, M.; Hanebeck, U.D. Decentralized data fusion with inverse covariance intersection. *Automatica* **2017**, *79*, 35–41. [CrossRef]
41. Ould-Ahmed-Vall, E.; Blough, D.M.; Ferri, B.H.; Riley, G.F. Distributed global ID assignment for wireless sensor networks. *Ad Hoc Netw.* **2009**, *7*, 1194–1216. [CrossRef]
42. Kumar, V.; Arablouei, R.; Jurdak, R.; Kusy, B.; Bergmann, N.W. RSSI-based self-localization with perturbed anchor positions. In Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Montreal, QC, Canada, 8–13 October 2017.
43. Mei, X.; Han, D.; Saeed, N.; Wu, H.; Ma, T.; Xian, J. Range Difference-Based Target Localization Under Stratification Effect and NLOS Bias in UWSNs. *IEEE Wirel. Comm. Lett.* **2022**, *11*, 2080–2084. [CrossRef]
44. Khalafalla, M.; Jiang, K.; Tian, K.; Feng, H.; Xiong, Y.; Tang, B. Two-Dimensional Target Localization Approach via a Closed-Form Solution Using Range Difference Measurements Based on Pentagram Array. *Remote Sens.* **2024**, *16*, 1370. [CrossRef]
45. Rajasegarar, S.; Havens, T.C.; Karunasekara, S.; Leckie, C.; Bezdek, J.C.; Jamriska, M.; Gunatilaka, A.; Skvortsov, A.; Palaniswami, M. High-Resolution Monitoring of Atmospheric Pollutants Using a System of Low-Cost Sensors. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 3823–3832. [CrossRef]
46. Leong, A.S.; Skvortsov, A.T. Estimation of Scalar Field Distribution in the Fourier Domain. *Digital Signal Process.* **2024**, *146*. [CrossRef]
47. Cattivelli, F.S.; Lopes, C.G.; Sayed, A.H. Diffusion Recursive Least-Squares for Distributed Estimation over Adaptive Networks. *IEEE Trans. Signal Process.* **2008**, *56*, 1865–1877. [CrossRef]
48. Cattivelli, F.S.; Sayed, A.H. Diffusion Strategies for Distributed Kalman Filtering and Smoothing. *IEEE Trans. Autom. Control* **2010**, *55*, 2069–2084. [CrossRef]
49. Olfati-Saber, R. Distributed Kalman Filter with Embedded Consensus Filters. In Proceedings of the IEEE Conference on Decision and Control, Seville, Spain, 12–15 December 2005; pp. 8179–8184.
50. Olfati-Saber, R. Distributed Kalman Filtering for Sensor Networks. In Proceedings of the IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 5492–5498.
51. Manolakis, D.G.; Ingle, V.K.; Kogon, S.M. *Statistical and Adaptive Signal Processing*; Artech House: Boston, MA, USA, 2005.
52. Chong, E.K.P.; Żak, S.H. *An Introduction to Optimization*, 4th ed.; John Wiley & Sons: Hoboken, NJ, USA, 2013.
53. Liggins, M.E.; Chong, C.Y.; Kadar, I.; Alford, M.G.; Vannicola, V.; Thomopoulos, S. Distributed Fusion Architectures and Algorithms for Target Tracking. *Proc. IEEE* **1997**, *85*, 95–107. [CrossRef]
54. Khaleghi, B.; Khamis, A.; Karray, F.O.; Razavi, S.N. Multisensor data fusion: A review of the state-of-the-art. *Inf. Fusion* **2013**, *7*, 28–44. [CrossRef]
55. Julier, S.; Uhlmann, J.K. General Decentralized Data Fusion with Covariance Intersection. In *Handbook of Multisensor Data Fusion*, 2nd ed.; Liggins, M.E., Hall, D.L., Llinas, J., Eds.; CRC Press: Boca Raton, FL, USA, 2009; Chapter 14, pp. 319–343.
56. Ajgl, J.; Straka, O. Inverse Covariance Intersection Fusion of Multiple Estimates. In Proceedings of the International Conference on Information Fusion, Rustenburg, South Africa, 6–9 September 2020; pp. 2247–2254.
57. Liu, Y.; Deng, Z.; Hu, E. Multi-Sensor Fusion Positioning Method Based on Batch Inverse Covariance Intersection and IMM. *Appl. Sci.* **2021**, *11*, 4908. [CrossRef]
58. Murphy, K.P. *Probabilistic Machine Learning: An Introduction*; The MIT Press: Cambridge, MA, USA, 2022.
59. Yuen, K.V. *Bayesian Methods for Structural Dynamics and Civil Engineering*; John Wiley & Sons: Hoboken, NJ, USA, 2010.
60. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
61. Wang, Z.; Bovik, A.C. Mean Squared Error: Love It or Leave It? *IEEE Signal Process. Mag.* **2009**, *26*, 98–117. [CrossRef]
62. Available online: https://www.eea.europa.eu/en/datahub (accessed on 14 July 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.