

RESEARCH

Open Access

# A distributed multi-robot adaptive sampling scheme for the estimation of the spatial distribution in widespread fields

Muhammad F Mysorewala<sup>1\*</sup>, Lahouari Cheded<sup>1</sup> and Dan O Popa<sup>2</sup>

## Abstract

Monitoring widespread environmental fields is undoubtedly a practically important area of research with many complex and challenging tasks. It involves the building of models of the fields or natural phenomena to be monitored, the estimation of the spatio-temporal distribution of a variety of environmental parameters of interest, such as moisture or salinity in a crop field, or the spatial distribution of vital natural resources such as oil and gas, etc. Sampling, a key operation of the monitoring process, is a broad methodology for gathering statistical information about the phenomenon, or environmental variable, being monitored. To efficiently monitor widespread fields and estimate the spatio-temporal distribution of some particular environmental variable, calls for the use of a sampling strategy can fuse information from different scales of sensors. Such an attractive strategy is well catered for by both the capabilities and distributed nature of wireless sensor networks and the mobility of robots performing the sampling (sensing) tasks. This sampling strategy could even be rendered “adaptive” in that the decision of “where to sample next” evolves temporally with past measurements and is optimally computed. In this article, we examine various single-robot and multi-robot adaptive sampling schemes based on different extended Kalman filter filtering structures such as centralized and decentralized filters as well as our own novel decentralized and distributed filters. Our investigation shows that, whereas the first two filters suffer from a heavy computational or communication load, our proposed method, through its key feature of distributing the filtering task amongst the robots used, manages to reduce both loads and the total reconstruction time. It also enjoys the added attractive feature of scalability that allows the structure of the proposed monitoring scheme to grow with the complexity of the field under study. Our results are corroborated by our simulation work and offer ample encouragement for a further theoretical investigation of some properties of the proposed scheme and its implementation on a physical system. Both of these activities are currently underway.

**Keywords:** Spatial field estimation, Adaptive sampling, Information fusion, Multi-scale sensing, Robotic sensor network, Environmental monitoring

## Introduction

Mobile robots are being increasingly used as sensor-carrying agents to perform sampling missions, such as searching for harmful biological and chemical agents, search and rescue in disaster areas, and environmental mapping and monitoring. One of the objectives of these sampling missions is ‘Field Estimation’. Field estimation is the construction of an estimate of how a certain

parameter varies in space and time, i.e., an estimate of its spatio-temporal distribution, based on observed or sampled data. As the field of interest is spread over a wide area, using a dense and fixed sampling scheme for an efficient field mapping would simply be too costly and will involve a possibly prohibitive computational load. Instead, it is far more interesting to use a mobile sampling scheme that would collect samples at few judiciously selected locations, in a way that would enable it to gain enough information about the field to be able to infer, with significant accuracy, the value of the parameter of interest at the unsampled locations. A

\* Correspondence: mysorewala@kfupm.edu.sa

<sup>1</sup>Systems Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Full list of author information is available at the end of the article

multitude of research groups have published results on sampling using mobile robots for chemical plume source localization [1,2], soil-moisture mapping for crop monitoring [3], ocean sampling [4,5], forest-fire mapping [6], etc.

The sensor fusion schemes for sampling missions can broadly be classified into three categories based on (i) physical parametric models, (ii) feature-based inference techniques such as clustering algorithms, neural networks, etc., which are generally non-parametric in nature but can lead to black or grey box parametric representation of the process, and (iii) cognitive-based models, which use the inference processes of humans and animals and which are based on fuzzy logic rules, search techniques, information-theoretic approaches, etc. Models acquired using these three broad classes of approaches can be either purely deterministic or purely stochastic. In many cases, deterministic models affected by some random noise can also be assumed.

In the area of physical deterministic parametric modeling representing the first category of sampling missions, Christopoulos and Roumeliotis [2] presented an approach for estimating the parameters of the diffusion equation that describes the propagation of an instantaneously released gas. Cannell and Stilwell [4] presented two approaches for adaptive sampling (AS) of underwater processes using AUVs. The first one assumes a parametric model, while the second one uses an information-theoretic approach. A number of strategies for non-parametric AS can also be found in the literature. A solution for non-parametric ocean sampling is proposed in [7] based on a classification of the sampling area. The multi-robot path planning problem is addressed in [8] using the mutual information collected using different paths. The study of [5] is also similar to that of [8] in the sense that both deal with generating optimal trajectories for multiple underwater vehicles for sampling purposes. Rule-based non-parametric approaches are also used widely in chemical plume tracing on land and in water, odor sensing [2], mine detection, etc.

Forest fires, chemical source leaks, and temperature variations in oceans are examples of complex natural phenomena for which the exact nonlinear model descriptions are unattainable due to the high-level of complexity involved. Demetriou and Hussein [9] present a solution to the problem of estimating a spatial distribution when the process is described by a partial differential equation. In [10], a non-parametric model is considered, and a distributed scheme for field estimation is developed using a Kalman filter-like recursive scheme.

In geostatistics, spatial processes are generally modeled as random fields, and estimation is performed using Kriging Interpolation techniques [11,12]. Kriging is termed

“simple” if the mean of the distribution is also known, and “universal” if the mean is treated as an unknown linear combination of known basis functions. In [13], a distributed algorithm is presented for spatial estimation using the Kriged Kalman filter. Graham and Cortes [14] proposed a Kriged Kalman filter-based approach for a spatiotemporal field where the discrete-time evolution of the state is governed by the Kalman filter used. In [15], the authors represent the time-varying field with a random process with a covariance known up to a scaling parameter. They proposed gradient descent algorithm which can run in a distributed fashion on multiple robots. Olfati-Saber [16,17] developed a distributed Kalman filter approach along with consensus filters to estimate the state of a process and reach consensus of all nodes.

Due to the time and energy-critical nature of some of these sampling scenarios, simply requiring the robots to perform a raster scan or randomly sample the field of interest would clearly be a sub-optimal and highly inefficient sampling strategy. Moreover, many time-varying distributions of interest encompass a wide area, and must therefore be observed with sensors having variables characteristics such as multiple size scales, rates, and accuracies [18]. For example, a forest fire is monitored using satellite images which provide a large spatial field-of-view (FOV) but a low-resolution or fidelity. On the other hand, a plane flying at low altitude would provide a low-spatial FOV but high-fidelity information.

In order to effectively fuse these different types of measurements, we proposed a *Multi-scale Multi-rate Adaptive Sampling* approach with a parametric description of the field [6]. In this approach, sampling strategies continuously adapt in response to real-time measurements from sensors of different scales. This scheme relies on building parametric models of the field using spatial sensor measurements collected from a high-altitude, and which are thus less accurate, and then improving the models by using more accurate spot measurements. The extended Kalman filter (EKF) is used to derive a quantitative information measure that is needed for the selection of sampling locations that are mostly likely to yield optimal information. In this approach, the existing low-resolution information of the field is first used to acquire an initial parametric representation of the field whose parameters have a higher initial error covariance which gradually reduces as high-resolution samples are taken and processed.

In our previous work [6], we presented a framework that extends our estimation of a simple parametric field to that of complex time-varying (e.g., forest fires [6]) by representing these with sums of overlapping Gaussians. The resulting algorithm was called EKF-NN-GAS, and is based on (a) a Radial Basis Function (RBF) neural

network (NN) for the parameterization of the non-parametric field, (b) an EKF for parameter estimation, and (c) a heuristic search scheme called ‘Greedy Adaptive Sampling’ (GAS).

A further investigation of the AS algorithm using multiple robots is presented in this article. For widespread fields, it may be impractical and certainly inefficient for a single-robot to map the entire field by navigating to different sampling locations, even when guided by an efficient sampling algorithm. However, when using multiple robots, the sampling area is first divided into smaller regions, and then each sampling instance in a particular region gains information about the parameters which have a dominant effect in that region. Therefore, in order to distribute computations, we need to be able to fuse the parameter estimates in order to construct the map of the field density distribution.

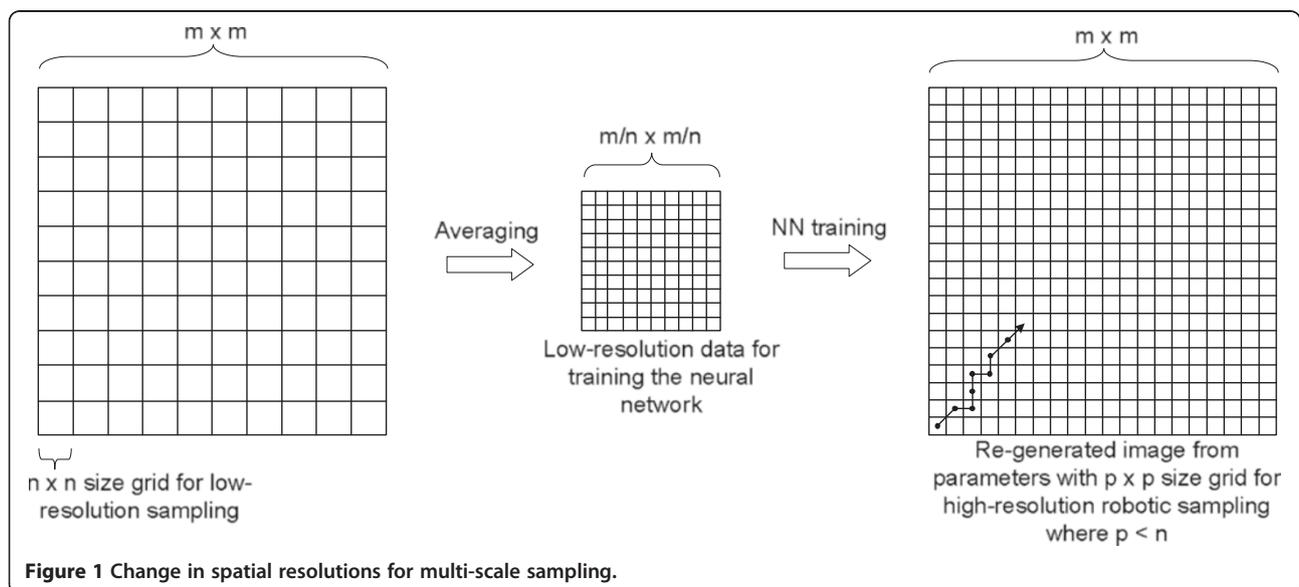
This problem is similar to reformulating the algorithm originally designed for a conventional single-sensor single-processor system to work on a more general multi-sensor, multi-processor system. Distributed algorithms have been used before in many applications, and the degree of parallelism used in them varies from one algorithm to another, depending on the application at hand. An example of distributing processing includes target location estimation using several sensors for data collection, and then fusing together the collected measurements either at the central station or at each sensor in a multi-sensor fusion algorithm [19-21].

Since complex fields are represented by hundreds of parameters [6], it is computationally cumbersome for a single-robot to compute and store all parameter estimates and the uncertainty measures. It also quickly becomes unfeasible for individual robots to run a large

AS algorithm, and share large covariance matrices wirelessly. Furthermore, with multi-robot sampling, the resources can be allocated efficiently if some resources are either busy or not available.

If the filter computation can be distributed among multiple robots, the number of computations performed by all the robots, i.e., the overall computational efficiency would be greater than the processing carried out by a single-robot having to carry-out both the sampling and computational tasks. Moreover, we expect that the concomitant advantages such as the flexible degree of parallelism, speed of convergence, and reduction in complexity that will be thus gained would be significant. With a single-robot, the total field estimation time includes the time necessary for navigation, sensing, and computation of the estimate (as there is no communication involved in this case). With multiple robots, the field estimation time includes the time taken for sensing, computation, communication, and final fusion to recover the field density distribution. We expect that the speed of convergence would increase by using multiple robots simply because of the sampling being done in parallel, and that the navigation time would be reduced significantly at the cost of modest increases in computation, communication, and fusion.

The rest of the article is organized as follows: in Section 2, we present the general formulation of the AS problem; Section 3 summarizes the existing centralized and decentralized filters, and their application to sensor network for field estimation; in Section 4, we present the novel federated distributed KF; Section 5 presents the simulation results for the proposed algorithm, and their discussion; finally Section 6 concludes the article.



**Figure 1** Change in spatial resolutions for multi-scale sampling.

### Formulation of multi-robot AS algorithm

As covered in our previous study, a single-robot-based AS algorithm for a 2D spatially stationary field  $g(x,y)$  can be described as follows [6] (Figure 1).

- (1) *Low-resolution sampling*: The field  $g(x,y)$  of size  $m \times m$  is divided into uniform square-sized grids  $n \times n$  such that  $n < m$ , and samples are collected at the centers of each of the  $n \times n$  grids. Hence,  $m/n \times m/n$  samples are collected as a low-resolution representation of the actual field.
- (2) *Parameterization*: Parametric representation of the field  $g(x,y)$  is achieved by training a B-neuron RBF neural network with the acquired low-resolution data. This results in a representation of the field as a sum of  $B$  Gaussians (one per neuron), and an offset (or bias) parameter  $b$ , with each neuron having its own parameters such as its peak  $a_i$ , variance  $\sigma_i$ , and center  $(x_{0i}, y_{0i})$ . Each of these parameters has an initial estimate value  $A_0$ , and an initial error covariance  $P_0$ . The number of neurons  $B$  is chosen depending on the complexity of the field and in such a way that the initial field estimation error is minimized to a value less than an acceptable threshold. Note at this stage that unlike the low-resolution samples which are uniformly distributed since they are acquired from uniformly distributed grids, the Gaussians (one Gaussian per RBF node) are distributed non-uniformly depending on the density of the field. We actually use more Gaussians in denser areas and fewer Gaussians in smoother areas of the field to be mapped. Further details on the relationship between the number of low-resolution samples and the number of neurons can be seen in [6].

Mathematically, a spatially stationary field is represented by the parameter vector  $A$  defined by

$$A = [b \quad a_1 \quad \sigma_1 \quad x_{01} \quad y_{01} \quad \dots \quad a_B \quad \sigma_B \quad x_{0B} \quad y_{0B}]_k^T \quad (1)$$

where  $A$  is the vector containing the true values of the parameters, which is not known due to (i) the resolution error between the actual field and the acquired low-resolution version, and (ii) RBF training error.

- (3) *High-resolution sampling*: In order to improve the field estimate, spot-measurements are made by a robotic vehicle which collects samples  $Z_k$  in a grid of size  $p \times p$  (where  $p \leq n$ ) based on a heuristic GAS algorithm [6]. According to the GAS algorithm, the next sampling location is searched within the vicinity of the currently sampled location, based on a criterion of minimization of the norm of the parameters' error covariance matrix.

The EKF governing and measurement equations are respectively given by

$$\begin{aligned} A_{k+1} &= A_k + \omega \\ &= [b_{0,k} \quad a_{1,k} \quad \sigma_{1,k} \quad x_{01,k} \quad y_{01,k} \quad \dots \quad a_{B,k} \\ &\quad \sigma_{B,k} \quad x_{0B,k} \quad y_{0B,k}]^T + \omega, \omega \sim N(0, Q) \\ Z_k &= h(A_k) + v_k \\ &= b + \sum_{i=1}^B a_i \exp[-\sigma_i \{(x_k - x_{0i})^2 \\ &\quad + (y_k - y_{0i})^2\}] + v, v \sim N(0, R) \end{aligned} \quad (2)$$

where  $Q$  is the process noise covariance,  $R$  is the measurement noise covariance and  $(x_k, y_k)$  are the robot sampling locations.

The multi-agent (or multi-robot) AS problem considered here can be described as follows:

*Assumptions:*

- (i) *A nonlinear spatio-temporal field variable is described via a parametric approximation  $Z = Z(A, X, t)$  depending on an unknown parameter vector  $A$ , position vector  $X$ , and time  $t$ .*
- (ii)  *$N$  robotic vehicles (agents) sample the field with sensing uncertainty in order to obtain higher resolution estimates of the field.*
- (iii) *The number of field parameters ( $L$ ) and their initial guesses are based on a hypothesis originating from prior knowledge of the field consistent with a low-resolution image of the entire field.*

As a complex spatial field is spread over a large area, its parameterization will require a large number of parameters. Therefore, it becomes unfeasible for a single-robot to navigate to different locations, collect samples, and improve parameter estimates in a short period of time. In addition to time constraints, the sampling problem also experiences constraints in the amount of energy available to the robot, as well as suffers from a considerable computational burden. These constraints limited the performance of our single-robot AS algorithm as described in [22]. Therefore, a key contribution of this article is to propose a better alternative that greatly alleviates the time and energy constraints imposed on the sampling process by the single-robot approach of mapping a spatio-temporal stationary field.

It is assumed here that only a single parameter  $Z$  vector is measured by all of the mobile robots used. However, in the case where multiple parameter vectors are to be measured, and the measurement model of each measured parameter vector is known, then the general EKF-based framework of AS presented [6] can be used. In

[23], we considered the scenario with two measurements only: the field measurement and the location of the robots.

It is important at this juncture to describe the following three main issues which underline the multi-robot sampling problem tackled here.

- (i) How can the sampling area be divided efficiently?  
 Section 2.1 discusses the above issue and suggests some efficient ways of tackling it.
- (ii) How can the density distribution be estimated through efficient data fusion when robots are collecting measurements in parallel?
- (iii) How can the computational and communication burden be distributed efficiently amongst the many robots used?

To address the last two issues, several possible algorithms are first presented in Sections 3&4, and then their respective simulation results presented and discussed in Section 5.

**Partitioning of sampling area**

A method is clearly needed to efficiently divide the sampling area into clusters, in order to run a parallel AS algorithm with multiple robots. Here, we propose an approach to efficiently divide the sampling area for parametric distributions using Fuzzy c-means clustering (FCM) and Centroidal Voronoi Tessellation (CVT) diagrams. FCM has frequently been used in the past for the classification of numerical data. CVT diagrams [24] have also been used for forming non-uniform size grids to better explore high-variance areas for non-parametric distributions [7]. Here, we employ a scheme to efficiently divide the sampling areas for parametric distributions

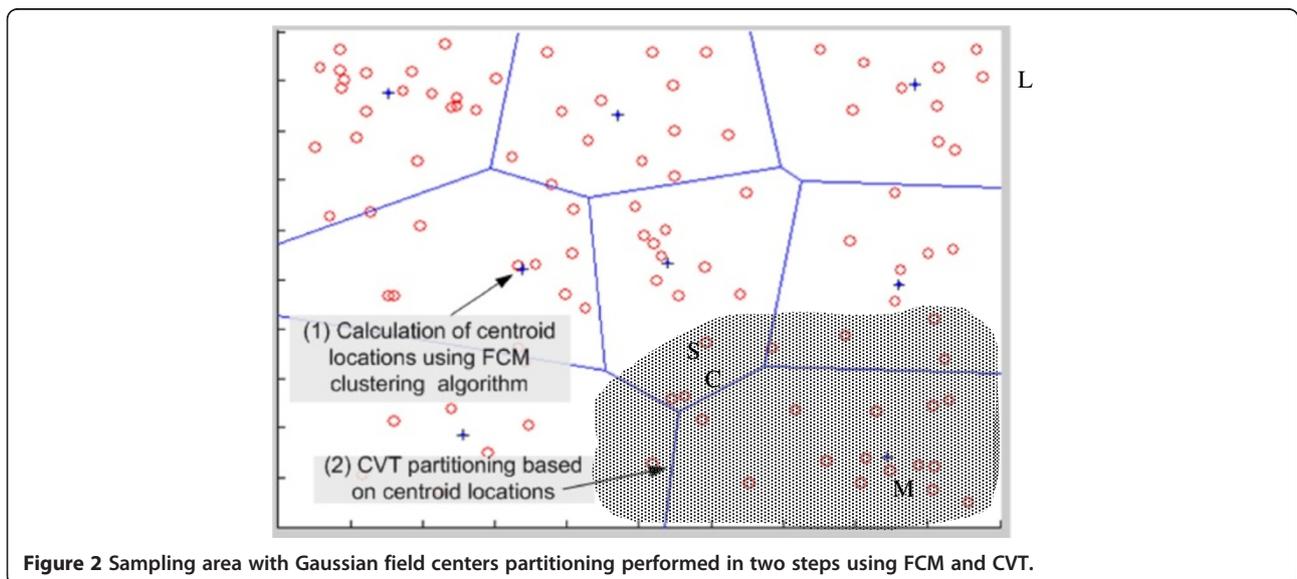
using both FCM and CVT. In this approach, FCM clusters samples based on the estimated centers of the approximating Gaussians used to map the field. Note here that we have assumed that the partitioning is performed once only at the beginning of the Fusion filter. For a time-varying field, further accuracy can be obtained by re-partitioning the field (and hence repositioning the Gaussians) after some samples to account for the field evolution in time.

As discussed in the beginning of this section, low-resolution samples from  $g(x, y)$  are used to train the RBF neural network which gives an estimate of the field as a sum of  $B$  Gaussians (neurons). This clustering approach is illustrated in Figure 2, where a field represented by  $B = 100$  Gaussians is partitioned into eight regions. The centers of these Gaussians shown in red circles are used for clustering.

As the clustering is fuzzy, it allows one piece of data to belong to several clusters via a membership grade  $u$  ranging between 0 and 1, and involves the iterative minimization of the cost function [25] given in Equation (3).

$$J_m = \sum_{i=1}^L \sum_{j=1}^N u_{ij}^m \|x_i - c_j\|^2, 1 \leq m < \infty, \quad (3)$$

where  $L = 4B + 1$  is the number of Gaussian centers,  $N$  is the number of clusters which is equal to the number of robots in this case,  $u_{ij}$  is the degree of membership of center  $x_i$  in cluster  $j$ ,  $c_j$  is the centroid of the cluster  $j$  and  $m$  is a real number greater than 1. Next, a CVT diagram based on Lloyd’s algorithm uses the centroid locations acquired by fuzzy clustering to classify all points in discrete space that are closest to the centroid, as a single group. Mathematically, given  $C$  clusters, each with a



**Figure 2** Sampling area with Gaussian field centers partitioning performed in two steps using FCM and CVT.

centroid denoted by  $c_s$ , then a point  $p$  on the field is said to be part of the cluster  $r$  if the following distance inequality is satisfied:  $|p - c_r| \leq |p - c_s|, s = 1, \dots, N, s \neq r$ .

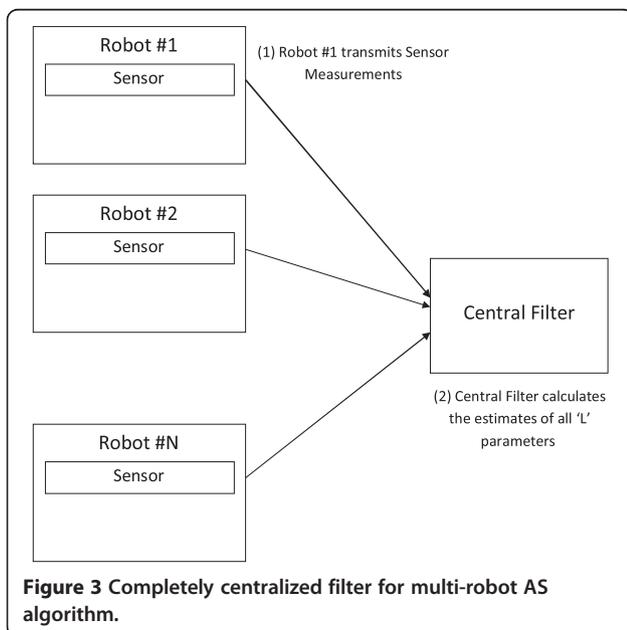
As a result of this mapping scheme, more Gaussians will overlap in areas where there are large field variations. The use of FCM and the CVT diagram for area classification may result in regions which have more variations and which must be as small as required in order to sample them thoroughly, i.e., so as not to miss out on any vital information. The areas with less variation, though they may be large, would require fewer samples, since they are represented by only a few parameters.

### Centralized, completely decentralized, and federated decentralized filters

In this section, we first examine completely centralized, completely decentralized, and federated decentralized filters, and their use in running the proposed multi-robot AS algorithm. We then argue that a new and efficient filter is needed for this application which will be discussed in detail in the following section.

#### Using completely centralized filter

In a completely centralized sampling approach, each robot  $j = 1, 2, \dots, N$  takes sensor measurement  $Z_{j,k+1}$  and transmits them to the central processor, which then calculates the required parameter estimates  $\hat{A}_{k+1}$  and error covariances  $P_{k+1}$ . The central processor computes these estimates, shown below in (4), using the 'KF equations for a single robot', (while single-handedly) taking on the task of fusing the multiple measurements it acquires from the  $N$  robots used.



**Figure 3** Completely centralized filter for multi-robot AS algorithm.

Figure 3 illustrates the completely centralized approach, in which all robots transmit their sensor measurement to the central filter, which then calculates the field estimate using Equation (4) given below where the superscript '-' in the vector  $A$  and matrix  $P$  indicates pre-measurement, while the lack of it indicates post-measurement:

EKF Pre-measurement update (a priori estimate) equations :

$$\hat{A}_{k+1}^- = \hat{A}_k = \begin{bmatrix} \hat{b}_{0,k} & \hat{a}_{1,k} & \hat{\sigma}_{1,k} & \hat{x}_{01,k} & \hat{y}_{01,k} & L \\ \hat{a}_{B,k} & \hat{\sigma}_{B,k} & \hat{x}_{0B,k} & \hat{y}_{0B,k} \end{bmatrix}^T$$

$$P_{k+1}^- = P_k$$

EKF Post-Measurement update (a posterior estimate) equations :

$$P_{k+1} = [P_k^{-1} + G_k^T R^{-1} G_k]^{-1}$$

$$= \left[ P_k^{-1} + \sum_{j=1}^N G_{j,k}^T R^{-1} G_{j,k} \right]^{-1}$$

$$\hat{A}_{k+1} = \hat{A}_k + P_{k+1} G_k^T R^{-1} \left[ \begin{pmatrix} Z_{1,k+1} \\ Z_{2,k+1} \\ M \\ Z_{N,k+1} \end{pmatrix} - \begin{pmatrix} g_k(\hat{A}_k) \\ g_k(\hat{A}_k) \\ M \\ g_k(\hat{A}_k) \end{pmatrix} \right]$$

where,

$$\hat{A}_k = \begin{bmatrix} \hat{b}_{0,k} & \hat{a}_{1,k} & \hat{\sigma}_{1,k} & \hat{x}_{01,k} & \hat{y}_{01,k} & \dots & \hat{a}_{B,k} \\ \hat{\sigma}_{B,k} & \hat{x}_{0B,k} & \hat{y}_{0B,k} \end{bmatrix}^T$$

$$g_k(\hat{A}_k) = b_{0,k} + \sum_{i=1}^B \hat{a}_{i,k} \exp \left\{ -\frac{(x - \hat{x}_{0i,k})^2 + (y - \hat{y}_{0i,k})^2}{2\hat{\sigma}_{i,k}^2} \right\}$$

$$G_k = \begin{bmatrix} \frac{\partial g_k}{\partial \hat{b}_{0,k}} & \frac{\partial g_k}{\partial \hat{a}_{1,k}} & \frac{\partial g_k}{\partial \hat{\sigma}_{1,k}} & \frac{\partial g_k}{\partial \hat{x}_{01,k}} & \frac{\partial g_k}{\partial \hat{y}_{01,k}} & \dots \\ \frac{\partial g_k}{\partial \hat{a}_{B,k}} & \frac{\partial g_k}{\partial \hat{\sigma}_{B,k}} & \frac{\partial g_k}{\partial \hat{x}_{0B,k}} & \frac{\partial g_k}{\partial \hat{y}_{0B,k}} \end{bmatrix}^T \quad (4)$$

Here we assume a stationary field and hence time prediction is not needed, i.e., the *a priori* estimates will be  $\hat{A}_{k+1}^- = \hat{A}_k$  and  $P_{k+1}^- = P_k$ . In [6], we assumed a slow time-varying field, a single sampling robot was used, and we included the prediction too considering the time evolution of the field.

This type of scheme is simple, as there is little communication involved and no redundant computations. But, the disadvantage is that the sensing robots do not carry any information on the field to be estimated. Therefore, this algorithm cannot be adaptive for every sample because the latest estimates are required to generate new sampling locations, and these estimates are

not calculated at every robot. Simulation results are shown in Section 5, where multiple sampling locations are chosen based on the current field estimate, and then all the measurement data collected are transmitted to the central filter for fusion, further processing and determination of the next sampling locations.

### Using completely decentralized filter

For a completely decentralized filter implementation, each robot not only takes the sensor measurement, but also runs locally the AS algorithm. However, it only calculates partial estimates of the field parameters and error covariance. It also generates new sampling locations within the vicinity of its current position. After every few samples, the robots communicate and share with each other their partial field estimate information, in order to calculate the complete estimates. The parameter estimate vector and the error covariance are the two terms each robot needs to transmit to the other robots. Each robot assimilates the received information using a decentralized EKF scheme formulated in [19,26].

Figure 4 illustrates the completely decentralized filter structure in which each robot has its own filter to compute partial estimates, and a fusion filter for assimilating the estimates acquired from other nodes to generate the complete field estimate.

If a completely decentralized approach is considered, then an AS algorithm running on each robot carries the information about all the field parameters, and thus there is no need at all for a global fusion filter in this case. Hence, each robot  $j$  can calculate the partial or

Local Estimate (LE),  $\hat{A}_{j,k+1,LE}$  and  $P_{j,k+1,LE}$  after  $(k+1)^{th}$  using Equation (5)

$$P_{j,k+1,LE} = \left[ P_{j,k,LE}^{-1} + G_{j,k,LE}^T R^{-1} G_{j,k,LE} \right]^{-1}$$

$$\hat{A}_{j,k+1,LE} = \hat{A}_{j,k,LE} + P_{j,k,LE} G_{j,k,LE}^T R^{-1} [Z_{j,k+1} - g_{j,k,LE}(\hat{A}_{j,k,LE})]$$

where,

$$\hat{A}_{j,k,LE} = \begin{bmatrix} \hat{b}_{0,k} & \hat{a}_{1,k} & \hat{\sigma}_{1,k} & \hat{x}_{01,k} & \hat{y}_{01,k} & \dots & \hat{a}_{B,k} \\ & \hat{\sigma}_{B,k} & \hat{x}_{0B,k} & \hat{y}_{0B,k} \end{bmatrix}_{j,LE}^T$$

$$g_{j,k,LE}(\hat{A}_{j,k,LE}) = \hat{b}_{0,j,k,LE} + \sum_{i=1}^B \hat{a}_{i,j,k,LE} \exp \left\{ -\frac{(x - \hat{x}_{0i,j,k,LE})^2 + (y - \hat{y}_{0i,j,k,LE})^2}{2\hat{\sigma}_{i,j,k,LE}^2} \right\}$$

$$G_{j,k,LE} = \begin{bmatrix} \frac{\partial g_k}{\partial \hat{b}_{0,k}} & \frac{\partial g_k}{\partial \hat{a}_{1,k}} & \frac{\partial g_k}{\partial \hat{\sigma}_{1,k}} & \frac{\partial g_k}{\partial \hat{x}_{01,k}} & \frac{\partial g_k}{\partial \hat{y}_{01,k}} \\ \dots & \frac{\partial g_k}{\partial \hat{a}_{B,k}} & \frac{\partial g_k}{\partial \hat{\sigma}_{B,k}} & \frac{\partial g_k}{\partial \hat{x}_{0B,k}} & \frac{\partial g_k}{\partial \hat{y}_{0B,k}} \end{bmatrix}_{j,LE}^T \quad (5)$$

Note that  $G_{j,k,LE}$ , where  $j, k, LE$  stand for the sensor number, sample number, and LE, respectively, is the Jacobian of the Gaussian vector  $g_{j,k,LE}$ , and is used in the above linearized EKF measurement update equation to estimate  $\hat{A}_{j,k,LE}$ .

To compute the  $r^{th}$  update, robot  $j$  calculates the total estimate  $(\hat{A}_{j,r}, P_{j,r})$  after each robot has collected its own  $q$  samples as explained next. First it (robot  $j$ ) acquires from the other robots their new partial estimates  $(\hat{A}_{i,rq,LE}, P_{i,rq,LE})$  which were computed from  $q$  new samples and then assimilates these new partial estimates

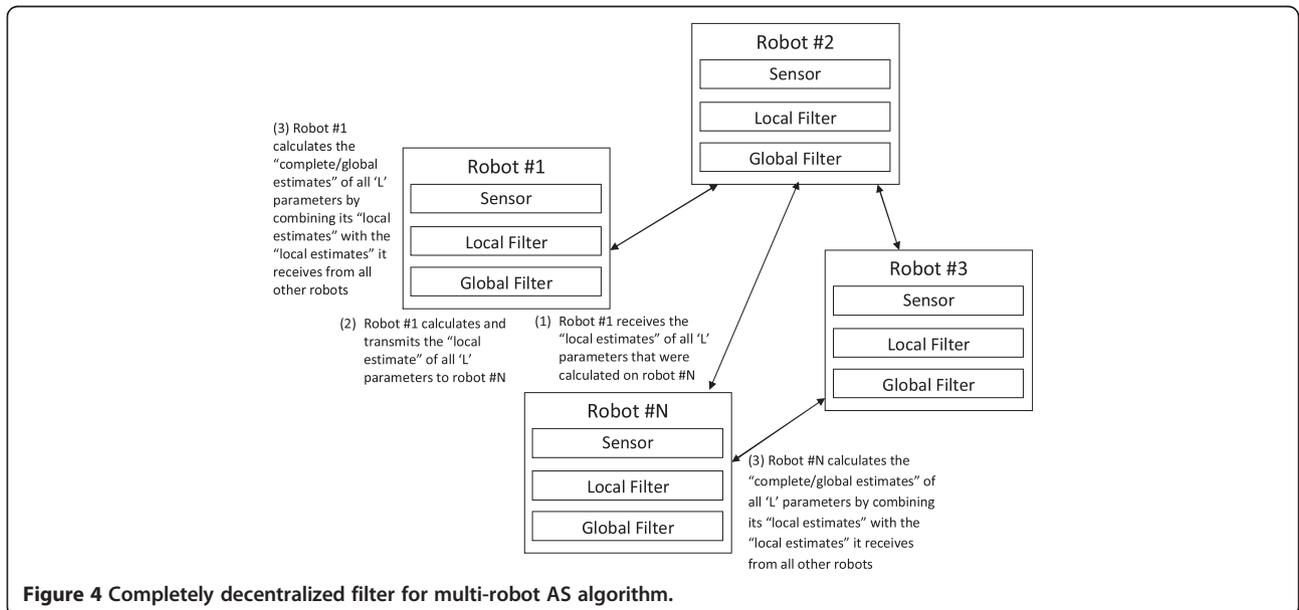


Figure 4 Completely decentralized filter for multi-robot AS algorithm.

with both its previous total estimates ( $\hat{A}_{j,r-1}P_{j,r-1}$ ) and its own new partial estimates. The complete  $r^{\text{th}}$  updates,  $P_{j,r}$  and  $\hat{A}_{j,r}$ , are finally computed by robot  $j$  using Equation (6) [19]:

$$(P_{j,r})^{-1} = (P_{j,r-1})^{-1} + \sum_{i=1}^N [(P_{i,rq,LE})^{-1} - (P_{i,(r-1)q,LE})^{-1}]$$

$$\hat{A}_{j,r} = P_{j,r} \left[ (P_{j,r-1})^{-1} \hat{A}_{j,r-1} + \sum_{i=1}^N [(P_{i,rq,LE})^{-1} \hat{A}_{i,rq,LE} - (P_{i,(r-1)q,LE})^{-1} \hat{A}_{i,(r-1)q,LE}] \right] \quad (6)$$

The advantage of this approach is that it does not involve any approximations, and there is no dependence on a central filter for computing the partial estimates. Also, the objective of sampling in parallel can be successfully achieved. The disadvantage of the algorithm is that it is demanding and inefficient in terms of communication and computational requirements when there are many parameters to estimate and heavy communication requirements to satisfy. This network has to be fully connected and there is excessive communication. This full parallelism (and complete distribution) of this type of algorithm can be taken advantage of in applications such as target tracking which involve the estimation of a few parameters (such as location, speed, etc., of the

target) only. When a large number of parameters are to be estimated, dividing the entire field of interest into several sampling areas and provided a sufficient number of robots is allocated to each area, then there will be no doubt that, through communication, this will enable different robots to carry better information about different parameters, thus resulting in an improvement of the overall estimation of the field. If only a few robots are used to sample a particular area, then each robot will have a larger sampling area to cover and it will take more time to calculate the local parameter estimates up to a certain degree of accuracy, from which it will then calculate the global estimate of the field parameters. This may not be possible under time constraint. This is clearly illustrated in Table 1 where reduction in number of robots from 4 to 1 resulted in an almost four fold increase in the total sampling time.

By the way of example, in adaptively sampling a field (shown in Figure 2) represented by  $B = 401$  parameters. The field is divided into  $N = 8$  partitions and the sampling operation is performed using 1 robot/partition. Running this decentralized algorithm would require each robot to calculate the partial estimate of 401 parameters, and to wirelessly transmit an error covariance matrix of size  $401 \times 401$ , and a parameter estimate vector of size  $401 \times 1$  to every other robot. Clearly, such a scheme would be very inefficient and not scalable.

**Table 1 Comparison of simulation results for single robot, multi-robot decentralized and federated decentralized filter**

	Single-robot	Multi-robot centralized KF (non-AS)	Multi-robot decentralized federated and non-federated fusion
Field size ( $m \times m$ )	300 × 300	300 × 300	300 × 300
Grid size for initial samples collection ( $n \times n$ )	30 × 30	30 × 30	30 × 30
Number of neurons ( $\beta$ )	40	40	40
RBF variances ( $\sigma$ )	30	30	30
Number of sampling robots ( $N$ )	1	4	4
Grid size for adaptive sampling ( $p \times p$ )	5 × 5	5 × 5	5 × 5
Horizon size (in grids) for next sample selection for each robot	10	30	10
Initial parameters error covariances $[b \ a_i \ s_i'x_{0i}y_{0i}]$	[200 50 10 <sup>-7</sup> 4 4]	[200 50 10 <sup>-7</sup> 4 4]	[200 50 10 <sup>-7</sup> 4 4]
Sensor measurement error covariance (R)	1	1	1
Initial norm of error covariance of all parameters ( $\ P_0\ $ )	375.9	375.9	375.9
Final norm of error covariance of all parameters ( $\ P_{k+1}\ $ )	13.25	241.0	17.27
Norm of error between original and initial estimated field ( $\ g - g_{est_0}\ $ )	25.05	25.05	25.05
Norm of error between original and final estimated field ( $E_{2F} = \ g - g_{est_{k+1}}\ $ )	19.67	48.0	19.33
Time taken to reach ( $\ g - g_{est_{k+1}}\  < 20$ )	11.92 min	5.48 min	2.89 min
No. of samples ( $qN$ )	300	300	320
No. of times KF runs for calculating the parameter estimates	300 (complete estimate)	1 (complete estimate)	320 (partial estimate)
# of samples/robot after which global estimate is calculated ( $q/r$ )	1	300	20
No. of times fusion is performed using LEs ( $t$ )	N/A	N/A	4

### Using a federated decentralized filter

In this approach, each robot takes some sensor measurements, estimates partial error covariances and field parameters, and transmits this information to a global fusion filter for assimilation, in a similar fashion to the approach proposed in [20,21,27]. Each robot runs Equation (5), but the fusion is done only at the fusion filter using Equation (6). Then these estimates are transmitted by the global fusion center (or filter) to all of the robots. So, the only difference between federated and completely decentralized approach is that in the federated case, these estimates are centrally calculated by the common global fusion filter while in complete decentralization, these are locally estimated at each robot.

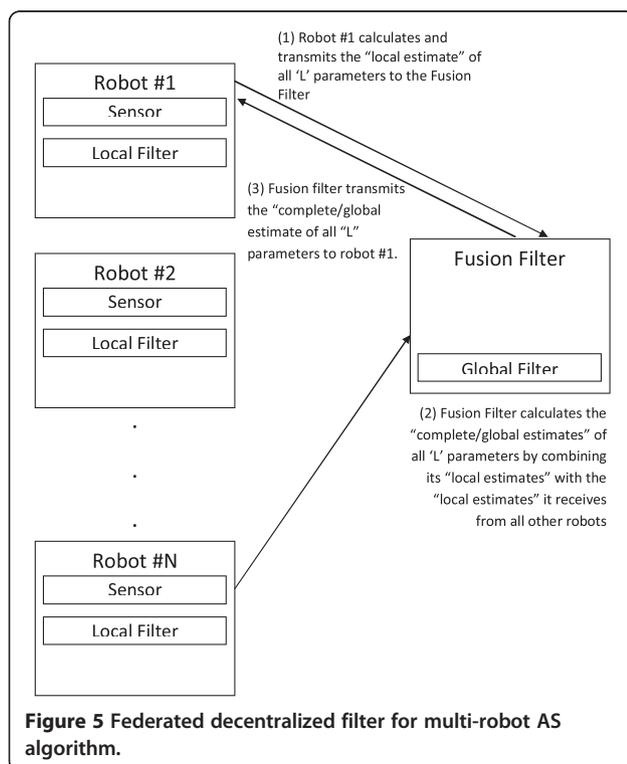
Figure 5 illustrates the federated decentralized filter in which each robot calculates partial field estimates, and transmits them to the global fusion filter, which then computes the complete field estimates. The advantage of this approach is that there is less communication compared to the completely decentralized case. Although in this case, none of the robots carries the complete information about all of the parameters all of the time, this approach will also be computationally more efficient than the completely decentralized implementation, simply because of the removal of the computational redundancy, due to fusion taking place at every robot, that was needed in the completely decentralized scheme. The disadvantage that this approach shares with the completely decentralized one is that partial estimates of all

parameters are still being carried by all of the robots all of the time, although information about these estimates might not be complete. Therefore, by federating the decentralized KF filtering scheme, the computational aspect of the problem has been mitigated but not the communication one. A thorough examination of the above three filtering schemes has therefore led us to take a novel and fruitful approach that would reduce both computational and communication overheads simultaneously. This novel approach is underpinned by a shift in focus from the mere decentralization of the KF filter to its distribution as described in the following section.

### Federated distributed Kalman filter

A decentralized and a distributed KF are two different formulations of the same KF algorithm [19]. In a decentralized algorithm, the filter is full-order, which means that every local filter carries partial information about all parameters, and the information is shared in a star topology to reach consensus amongst all robots on the final parameter estimates. The objective of distributed algorithms is to efficiently decompose the full-order filter into several reduced-order filters, in order to reduce the computational complexity and communication overhead, and hence improve the scalability. It can be said that decentralization is the first step toward efficient distribution. In case of no distribution, every collected sample is used to compute the estimates of all parameters in the field. But with distribution, this sample is used to compute the estimates of only those parameters which have significant impact on the region where this sample has been collected.

The objective of the work presented in this section is to modify the formulation of a federated decentralized scheme, in order to reduce both the communication overheads and the computational load involved. This formulation considers only the cross-covariance terms contributed by neighboring Gaussians only and ignores those contributed by distant Gaussians as a trade-off between accuracy and computational complexity. The decision behind ignoring distant Gaussians is supported by the analysis provided in Section 5, where a threshold of 0.001% in the relative contribution of each Gaussian was used in deciding the number of Gaussians to keep. An accurate DKF is not possible in this AS problem because local measurement models are not available. Furthermore, the use of global measurement models at each node requires the estimate of all parameters, which will contradict the motivation behind the implementation of DKF. There are other schemes that handle the error covariance terms "very lightly" such as Kalman Consensus schemes, which take the average of the error covariances of the parameter estimates in order to implement the DKF with only communication between neighboring nodes being used [16,17].



Decentralized approaches are good enough for applications involving a small number of states such as tracking of objects, etc. But problems such as parametric sampling involve hundreds of parameters, and hence distributing the KF filter becomes all the more important for an efficient operation.

### Approach to distributed computations and communications

Assume that we have a continuous field distribution within a certain perimeter, which means that there is discontinuity between the field and its surroundings. As shown in Figure 2, this field is represented by  $L$  parameters, where  $L = 4B + 1$ , and the field estimate is calculated at the central station based on the LEs received from  $N$  sampling robots. In the example shown in Figure 2,  $B = 100$ ,  $N = 8$ , and  $L = 401$ . The circles shown are the center  $(x_{0i}, y_{0i})$  of  $B$  Gaussians. One of the highlighted partitions has  $S$  parameters, the estimates of which are expected to change by collecting samples from that partition.  $S$  includes all the parameters inside a partition, as well as the surrounding parameters which have a significant impact on that partition. The collection of a single sample leads to the change in  $M$  parameter estimates, whereas collecting multiple samples results in the change of  $C$  parameter estimates. Hence, from a set-theoretic point of view, we can state that  $M < C < S < L$ . For the decentralized case,  $M = C = S = L$  and all the cross-covariance terms contributed by all the Gaussians are considered. However, for the distributed case, we have  $M < C < S < L$  and an increase in  $M$ ,  $C$ , and  $S$  will lead to a better accuracy at the cost of a higher number of computations.

The idea behind this approach is to run a reduced-order KF rather than a full-order one so as to reduce the computational load, as well as the communication overheads by transmitting only the smallest amount of information needed.

Given the following sizes of the variables involved:  $\hat{A}_M \in \mathfrak{R}^{M \times 1}$ ,  $P_M \in \mathfrak{R}^{M \times M}$ ,  $\hat{A}_C \in \mathfrak{R}^{C \times 1}$ ,  $P_C \in \mathfrak{R}^{C \times C}$ ,  $\hat{A}_S \in \mathfrak{R}^{S \times 1}$ ,  $P_S \in \mathfrak{R}^{S \times S}$ ,  $\hat{A}_L \in \mathfrak{R}^{L \times 1}$ ,  $P_L \in \mathfrak{R}^{L \times L}$ , this approach involves the following steps:

1. Transformation from  $(P_L, \hat{A}_L)$  to  $(P_S, \hat{A}_S)$  at the fusion filter.

The fusion filter evaluates the initial estimate of  $(P_S, \hat{A}_S)$  by first generating the binary transformation matrix  $U_{LS}$  (to transform  $L$  to  $S$ ), and transmitting  $(P_S, \hat{A}_S)$  to robot 1. The matrix  $U_{LS} = U_{SL}^T$  is kept in memory by the fusion filter for the final assimilation stage.

$$\begin{aligned} P_S &= U_{LS}^T P_L U_{LS}, \hat{A}_S = U_{LS}^T \hat{A}_L \\ \hat{A}_S &\in \mathfrak{R}^{S \times 1}, P_S \in \mathfrak{R}^{S \times S}, \hat{A}_L \in \mathfrak{R}^{L \times 1}, \\ P_L &\in \mathfrak{R}^{L \times L}, U_{LS} \in \mathfrak{R}^{L \times S} \end{aligned} \quad (7)$$

2. Transmit the estimates of  $S$  parameters  $(P_S, \hat{A}_S)$  to Robot # $j$
3. Collect the measurement- and estimate pair,  $(P_{M,k+1}, \hat{A}_{M,k+1})$

$$P_{M,k+1} = \left[ \left( U_{SM,k+1}^T P_S U_{SM,k+1} \right)^{-1} + G_{M,k+1}^T R_{k+1}^{-1} G_{M,k+1} \right]^{-1} \in \mathfrak{R}^{M_{k+1} \times M_{k+1}}$$

$$\begin{aligned} \hat{A}_{M,k+1} &= U_{SM,k+1}^T \hat{A}_S \\ &+ P_{M,k+1} G_{M,k+1}^T (R_{k+1})^{-1} [Z_{k+1} - g(\hat{A}_{M,k})] \\ &\in \mathfrak{R}^{M_{k+1} \times 1} \end{aligned}$$

$$G_{M,k+1} = \frac{\partial g(A_{M,k})}{\partial \hat{A}_{M,k}} \in \mathfrak{R}^{1 \times M_{k+1}} \quad (8)$$

$$\begin{aligned} \hat{A}_{M,k+1} &\in \mathfrak{R}^{M_{k+1} \times 1}, P_{M,k+1} \in \mathfrak{R}^{M_{k+1} \times M_{k+1}}, \\ U_{SM,k+1} &\in \mathfrak{R}^{S \times M_{k+1}} \end{aligned}$$

4. Transformation from  $(P_{M,k+1}, \hat{A}_{M,k+1})$  to  $(P_{C,k+1}, \hat{A}_{C,k+1})$

$$\begin{aligned} P_{C,k+1} &= U_{C,k+1}^T P_{C,k} U_{C,k+1} + U_{MC,k+1}^T (P_{M,k+1} \\ &- U_{SM,k+1}^T P_S U_{SM,k+1}) U_{MC,k+1} \\ \hat{A}_{C,k+1} &= U_{C,k+1}^T \hat{A}_{C,k} + U_{MC,k+1}^T (\hat{A}_{M,k+1} \\ &- U_{SM,k+1}^T \hat{A}_S) \end{aligned}$$

$$\begin{aligned} \hat{A}_{C,k+1} &\in \mathfrak{R}^{C_{k+1} \times 1}, P_{C,k+1} \in \mathfrak{R}^{C_{k+1} \times C_{k+1}}, \\ \hat{A}_{C,k} &\in \mathfrak{R}^{C_k \times 1}, P_{C,k} \in \mathfrak{R}^{C_k \times C_k}, \\ U_{MC,k+1} &\in \mathfrak{R}^{M_{k+1} \times C_{k+1}}, U_{C,k+1} \in \mathfrak{R}^{C_{k+1} \times C_k} \end{aligned} \quad (9)$$

where  $U_{MC,k+1}$  and  $U_{C,k+1}$  are the binary matrices for transformation from  $M$  to  $C$  for the  $(k+1)^{\text{th}}$  sample, and the transformation of  $C$  from  $k^{\text{th}}$  to  $(k+1)^{\text{th}}$  sample, respectively.

5. Repeat steps 3 and 4 until an update is requested from the fusion filter.
6. Transmit the pair  $(P_C, \hat{A}_C)$  to the fusion filter
7. The fusion filter then substitutes  $(P_C, \hat{A}_C)$  into  $(P_{j,LE}, \hat{A}_{j,LE})$  which is unique to each robot.

$$\begin{aligned} [P_{L,k+n}]_j &= \left[ P_{L,k} + U_{SL,k+n}^T (U_{CS,k+n}^T P_{C,k+n} U_{CS,k+n} \right. \\ &\left. - U_{CS,k}^T P_{C,k} U_{CS,k}) U_{SL,k+n} \right]_j \\ [\hat{A}_{L,k+n}]_j &= \left[ \hat{A}_{L,k} + U_{SL,k+n}^T (U_{CS,k+n}^T \hat{A}_{C,k+n} \right. \\ &\left. - U_{CS,k}^T \hat{A}_{C,k}) \right]_j \end{aligned} \quad (10)$$

8. The fusion filter finally runs the global update Equation (6) considering all the different pairs  $(P_{j,LE}, \hat{A}_{j,LE})$  to be local updates from different robots.

For clarification, an example is shown below with  $L = 401$ ,  $S = 10$ ,  $C = 5$ ,  $M = 3$  (for two samples).

$$P_S = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} & p_{16} & p_{17} & p_{18} & p_{19} & p_{1,10} \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} & p_{26} & p_{27} & p_{28} & p_{29} & p_{2,10} \\ p_{31} & p_{32} & p_{33} & p_{34} & p_{35} & p_{36} & p_{37} & p_{38} & p_{39} & p_{3,10} \\ p_{41} & p_{42} & p_{43} & p_{44} & p_{45} & p_{46} & p_{47} & p_{48} & p_{49} & p_{4,10} \\ p_{51} & p_{52} & p_{53} & p_{54} & p_{55} & p_{56} & p_{57} & p_{58} & p_{59} & p_{5,10} \\ p_{61} & p_{62} & p_{63} & p_{64} & p_{65} & p_{66} & p_{67} & p_{68} & p_{69} & p_{6,10} \\ p_{71} & p_{72} & p_{73} & p_{74} & p_{75} & p_{76} & p_{77} & p_{78} & p_{79} & p_{7,10} \\ p_{81} & p_{82} & p_{83} & p_{84} & p_{85} & p_{86} & p_{87} & p_{88} & p_{89} & p_{8,10} \\ p_{91} & p_{92} & p_{93} & p_{94} & p_{95} & p_{96} & p_{97} & p_{98} & p_{99} & p_{9,10} \\ p_{10,1} & p_{10,2} & p_{10,3} & p_{10,4} & p_{10,5} & p_{10,6} & p_{10,7} & p_{10,8} & p_{10,9} & p_{10,10} \end{bmatrix}, A_S = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \end{bmatrix} \quad (11)$$

Let the sample taken at time  $(k + 1), (k + 2)$  and  $(k + 3)$ , respectively, estimate the parameters  $(2,4,7)$ ,  $(3,4,6)$ , and  $(1,2,4,9)$ . Then,

For the first sample: parameters  $(2, 4, 7)$  changes. Therefore,

$$P_{M,k+1} = P_{C,k+1} = \begin{bmatrix} p_{22} & p_{24} & p_{27} \\ p_{42} & p_{44} & p_{47} \\ p_{72} & p_{74} & p_{77} \end{bmatrix},$$

$$A_{M,k+1} = A_{C,k+1} = \begin{bmatrix} a_2 \\ a_4 \\ a_7 \end{bmatrix}, \quad (12)$$

For the second sample: parameters  $(3, 4, 6)$  changes. Therefore,

$$P_{M,k+2} = \begin{bmatrix} p_{33} & p_{34} & p_{36} \\ p_{43} & p_{44} & p_{46} \\ p_{63} & p_{64} & p_{66} \end{bmatrix}, A_{M,k+2} = \begin{bmatrix} a_3 \\ a_4 \\ a_6 \end{bmatrix}$$

$$U_{C,k+2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$U_{C,k+2}^T P_{C,k+1} U_{C,k+2} = \begin{bmatrix} p_{22} & 0 & p_{24} & 0 & p_{27} \\ 0 & 0 & 0 & 0 & 0 \\ p_{42} & 0 & p_{44} & 0 & p_{47} \\ 0 & 0 & 0 & 0 & 0 \\ p_{72} & 0 & p_{74} & 0 & p_{77} \end{bmatrix},$$

$$U_{MC,k+2} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad (13)$$

$$P_{C,k+2} = P_{j,k+2,LE} = \begin{bmatrix} p_{22} & 0 & p_{24} & 0 & p_{27} \\ 0 & p_{33} & p_{34} & p_{36} & 0 \\ p_{42} & p_{43} & p_{44} - p_{44}^- & p_{46} & p_{47} \\ 0 & p_{63} & p_{64} & p_{66} & 0 \\ p_{72} & 0 & p_{74} & 0 & p_{77} \end{bmatrix},$$

$$\hat{A}_{C2} = \hat{A}_{j,k+2,LE} = \begin{bmatrix} a_2 \\ a_3 \\ a_4 \\ a_6 \\ a_7 \end{bmatrix},$$

For the third sample parameters  $(1, 2, 4, 9)$  changes. Therefore,

$$A_{M,k+3} = \begin{bmatrix} a_1 \\ a_2 \\ a_4 \\ a_9 \end{bmatrix}, P_{M,k+3} = \begin{bmatrix} p_{11} & p_{12} & p_{14} & p_{19} \\ p_{21} & p_{22} & p_{24} & p_{29} \\ p_{41} & p_{42} & p_{44} & p_{49} \\ p_{91} & p_{92} & p_{94} & p_{99} \end{bmatrix}$$

### Computational and communication complexities

EKF has an  $O(L^3)$  computational complexity if each sample updates all of the  $L$  parameters of the two-dimensional parametric field. However, as a first-order approximation, it can be assumed that a single sample affects only neighboring parameters. With this assumption, the algorithm can run in a distributed fashion, and the computational complexity at the sampling nodes can then be reduced. Only the fusion filter's complexity remains of order  $O(L^3)$ , because it needs to combine information about all the  $L$  parameters. However, this central field parameter fusion process occurs less frequently and hence will have only a small effect on the overall computational burden.

Table 2 illustrates a comparison of computations and communication complexity for a centralized, completely decentralized, federated decentralized and distributed filter. Let  $N$  be the number of sampling robots,  $L$  is the number of field parameters,  $q$  is the number of sensor measurements per robot, and  $r$  is the number of times robots communicate to share their information with each other.

For the centralized filter, the sensing robots do not perform any computation. Hence, the computational and communication complexity are  $O(qNL^3)$  and  $O(qN^3)$ , respectively.

For a completely decentralized filter, the computational complexity involved in calculating the LE at each robot is  $O(qL^3)$ , whereas that involved in calculating the global estimate at each robot is  $O((N - 1)rL^3)$ , after taking estimates from  $(N-1)$  robots at a frequency  $r$ . Hence, the combined computational complexity becomes

**Table 2 Comparison of computational complexity and communication overhead for centralized, decentralized, federated decentralized, and federated distributed filter**

	Computations			Communication
	Robot	Fusion center	Combined	
Centralized filter	–	$O(qNL^3)$	$O(qNL^3)$	$O(qN)$
Completely decentralized filter	$O(qL^3 + (N - 1)rL^3)$	–	$O(NqL^3 + N(N - 1)rL^3)$	$O(N(N - 1)r(L^2 + L))$
Federated decentralized filter	$O(qL^3)$	$O(rL^3)$	$O(NqL^3 + rL^3)$	$O(2Nr(L^2 + L))$
Federated distributed filter	$O(qM^3)$	$O(rL^3)$	$O(NqM^3 + rL^3)$	$O(Nr(C^2 + C + S^2 + S))$

$O(NqL^3 + N(N - 1)rL^3)$ . At the same time, the communication complexity is  $O(N(N - 1)r(L^2 + L))$ .

In order to reduce the communication overhead and computational complexity, a federated filter calculates the global estimate on the fusion filter only, which reduces the computational complexity to  $O(NqL^3 + rL^3)$ , and the communication complexity to  $O(2Nr(L^2 + L))$ .

Finally, for the proposed distributed version of the federated decentralized filter, instead of calculating the estimates of  $L$  states at a single robot, we simply calculate the estimates of  $M$  ( $M < L$ ) states at a single robot for each sample collected. This approach reduces the computational and communication complexity to  $O(NqM^3 + rL^3)$  and  $O(Nr(C^2 + C + S^2 + S))$ , respectively.

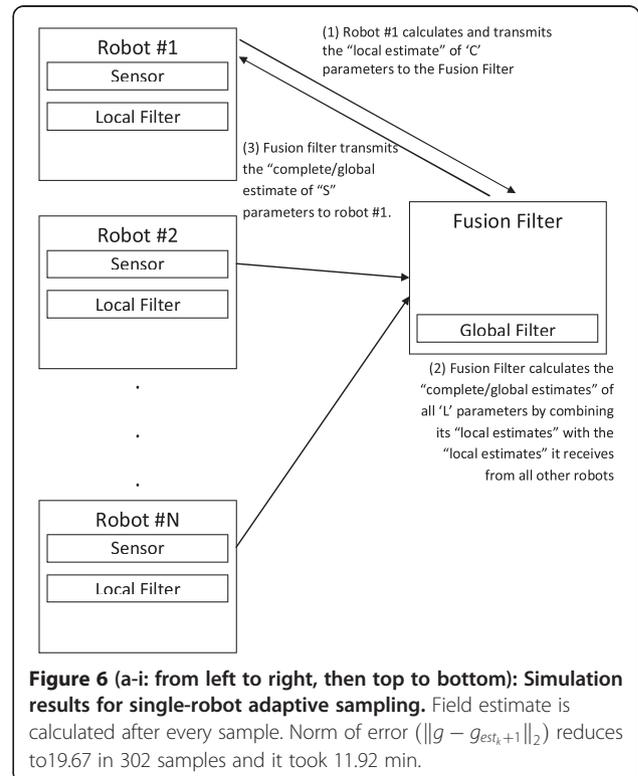
### Simulation results

In our previous work, we have shown simulation and experimental results for a single-robot AS procedure to validate our approach [6,22,23,28,29]. We now consider the multi-robot algorithm with centralized, decentralized, federated decentralized, and distributed filtering structures.

Here a complex field, of size  $m \times m = 300 \times 300$  pixels, is generated as the truth field, and is to be reconstructed by AS using  $N=4$  robots. The field is divided into uniformly-sized grids of size  $n \times n = 30 \times 30$  each, and  $m/n \times m/n = 10 \times 10 = 100$  low-resolution samples are initially collected by considering a sample from the middle of each grid. These samples provide a low-resolution description of the field. These initial samples are used for training the RBF neural network and the training method used is of the ‘Self-organized selection of centers’ type ([30]). We use the ‘new rb’ function of MATLAB to train the neural network assuming  $B=40$  neurons and a spread parameter of  $\sigma=30$ . This provides an initial estimate of the field with  $L \times B + 1 = 161$  parameters. Spot measurement-based AS is then performed by robots roaming in smaller grids, each of size  $p \times p = 5 \times 5$ , in order to improve the field estimate. All assumptions used and results obtained are shown in Table 1.

To estimate the field reconstruction accuracy, two convergence criteria are used. One is the 2-norm of the

error between the original and the estimated field, i.e.,  $E_{2F} = \|g - g_{est_{k+1}}\|_2$ , henceforth referred to as the field error, which is achieved by calculating the errors for all points  $(x,y)$  in the field, and then calculating the 2-norm of these point-wise error values. It is obvious that, for a fixed neural network structure, using more samples for the initial training would result in a smaller initial field estimation error. For example, as shown in Figure 6d, if  $m/n \times m/n = 10 \times 10 = 100$  low-resolution samples per uniform grid are used for RBF training, then the initial field error  $E_{2F} = 32$ , and the final field error after 302 samples is  $E_{2F} = 19.67$ . However, by increasing the number of low-resolution samples per a uniform grid from 100 to 900 (i.e., a nine fold increase), the initial value of the field error ( $E_{2F}$ ) decreases from 32 to 20 (i.e., a decrease of 37.5%).

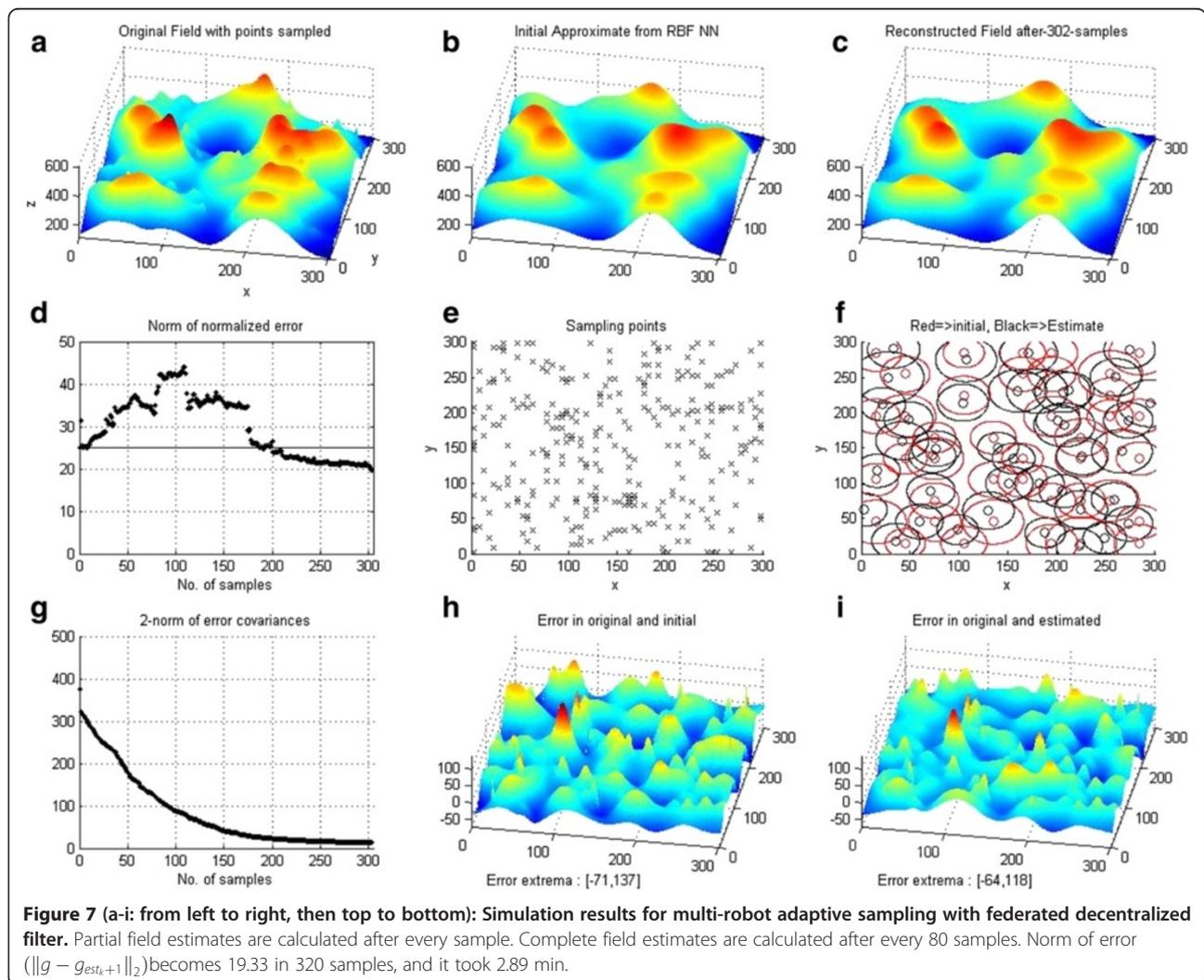


However, it is important to note here that, while the example here, based on a lower number (100) samples per grid, has a high initial field error, it achieves the same accuracy as the example (covered in [6]), which uses a higher number (900) of samples per grid. The accuracy achieved by this example is due to the fact that it relies on AS while using only a smaller total number of samples of 100 (initial samples) + 302(adaptively acquired) = 402 samples than the one used in the example of [6].

The other criterion is the 2-norm of the parameter error covariance matrix ( $\|P_{k+1}\|_2$ ).

Figures 6 and 7, respectively, show the simulation results when using a single-robot sampling and a multi-robot one. It can be seen from Figure 6d that the field estimation error first increases to a peak value before it starts to decrease. This initial increase in error seems to be caused by an apparent divergence of the EKF filter which is prone to divergence because of its dependence on the first-order linearization process that is

performed to calculate the new estimate. More detailed analysis of this can be found in [29] where we carried out a thorough comparison between various nonlinear filters such as the EKF, Second-order EK, Iterated EKF, and Unscented KF so as to study and highlight the limitations of the EKF filter. Another possible reason for this filter divergence could be the insufficient number of samples used. This can also be exacerbated by the fact that the further these few samples are apart, i.e., the larger the linearization step is, the worse the linearization error becomes. This increase in error could also be due to an insufficient coverage of the sampling area. This therefore reinforces our motivation to use multiple robots that ensure that different regions are adequately covered at the same time. The improvement brought about by the use of multiple robots can be seen in Figure 7d for the multi-robot case where, the initial error increase, although not completely eliminated, has been greatly reduced compared to the single robot case (Figure 6d).

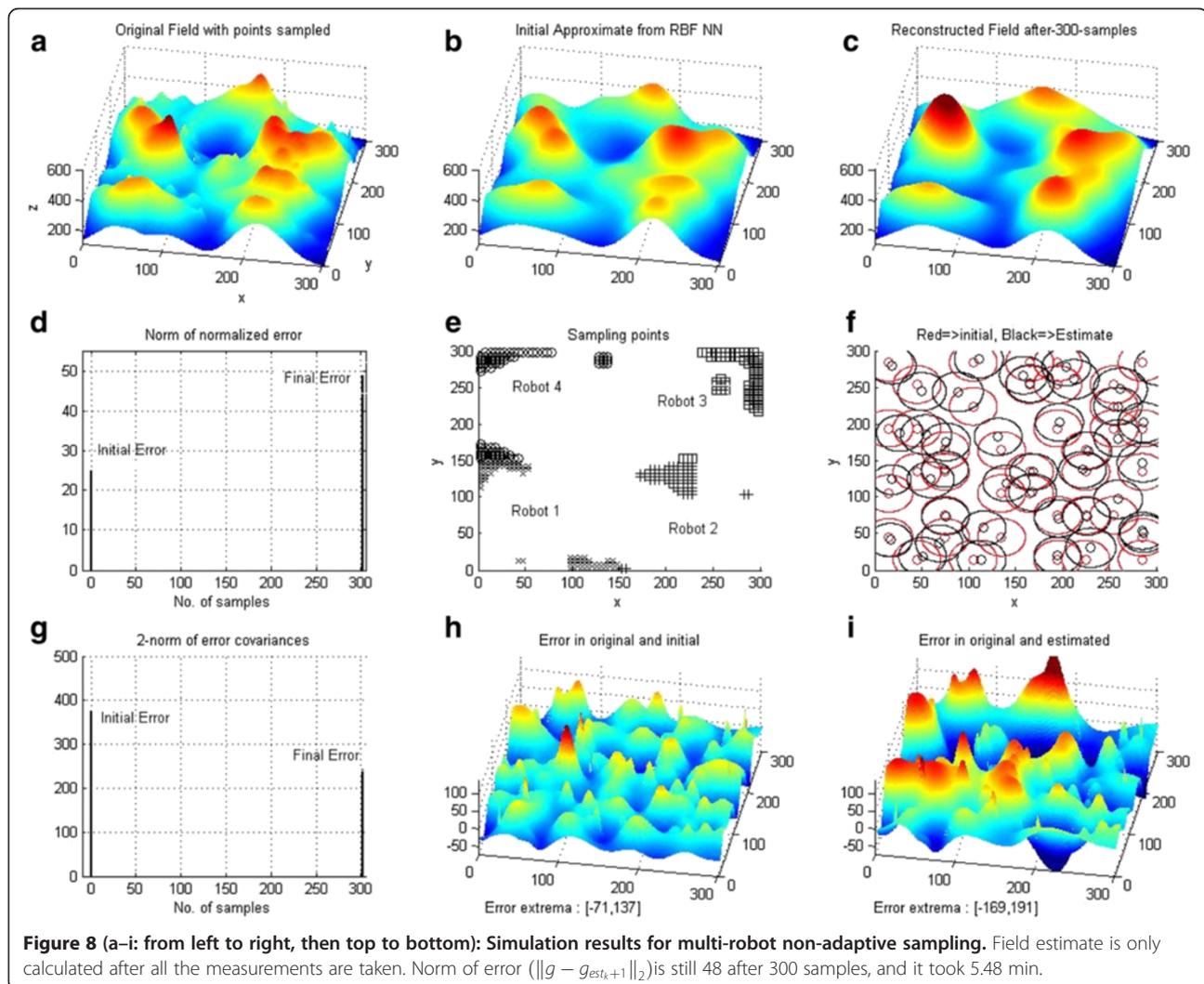


As discussed in previous sections, if the centralized filter is used for multi-robots, then AS is not possible. Figure 8 shows the simulation results when all the sampling locations for the four robots used are generated in advance based on the initial estimate. Hence, the sampling approach is non-adaptive in nature. Robots collect samples from these locations and transmit them to the central filter for fusion. It can be clearly seen from Figure 8e that if future sampling locations generated by the (non-adaptive) sampling algorithm are based on the initial estimate of error covariance only, then these locations would not provide much information about the global field distribution, as these locations are all closer to one another and hence would furnish only a localized knowledge of the field distribution. In fact, after collecting 300 samples, the error is still very high as shown in Figure 8d and Table 3. Moreover, it takes the non-negligible time of 5.48 min to perform this mission. Figure 7 shows the results for a federated decentralized approach which is equally

valid for a completely decentralized one. The only difference between these two approaches will be in the computation and communication load to be carried by the robots. For the completely decentralized approach, the total number of samples collected is  $q=320$ . After every 20 samples collected, each robot sends its partial (local) estimate to the global filter for fusion. This way this update is performed  $r=4$  times.

The use of four robots instead of one for sampling also reduces the time for field reconstruction from 11.92 to 2.98 min which amounts approximately to a fourfold reduction in time. The reason for this reduction can be explained intuitively since, by sampling using four robots, instead of one, not only does the number of samples collected by each robot gets reduced, but so does the navigation time as well because of the smaller sampling area allocated to each robot.

It is important to point out at this juncture that the process by which only the average number of the most influencing Gaussians is kept is based on their percent



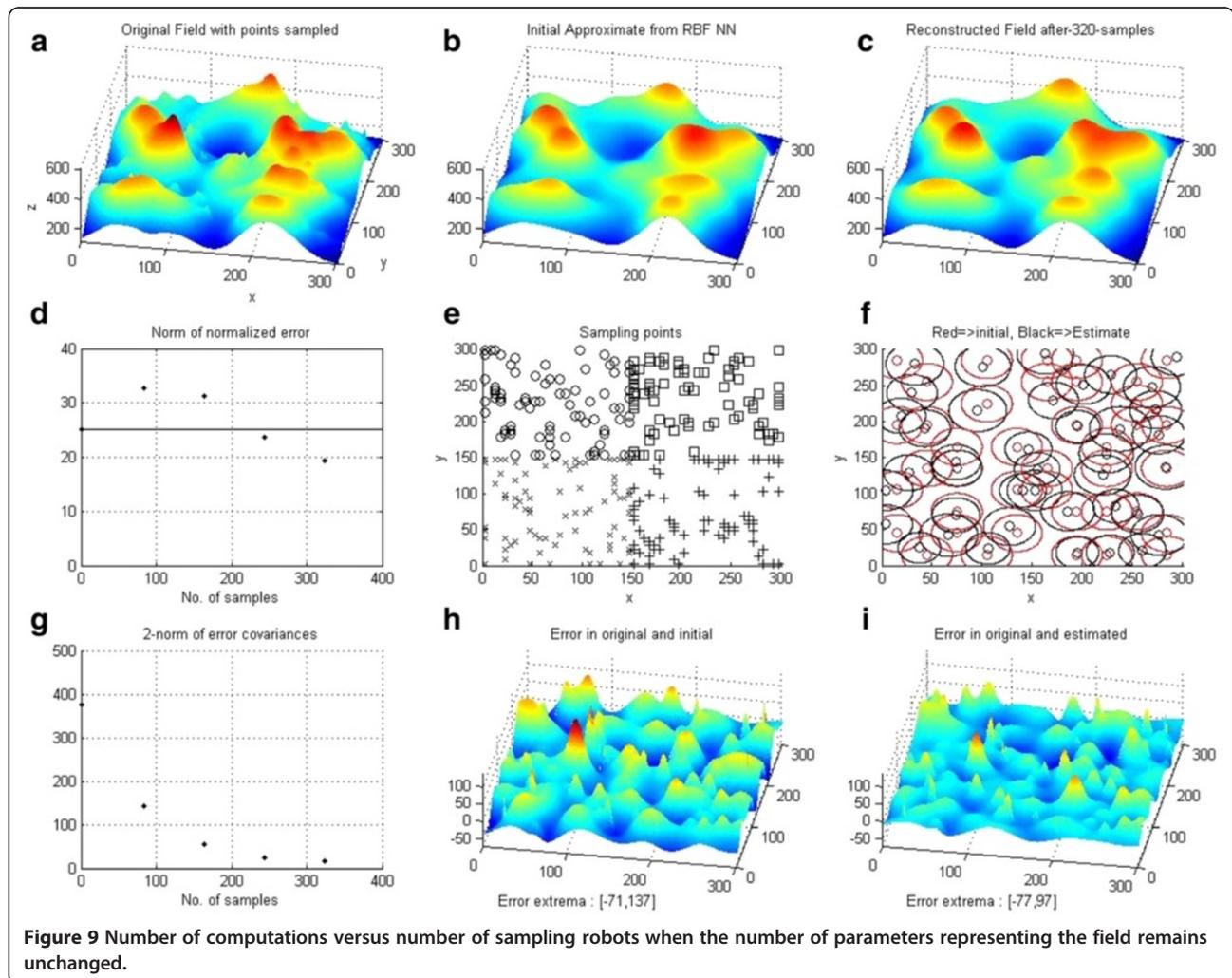
**Table 3 Comparison of computational loads and communication overheads for centralized, completely decentralized, federated decentralized and federated distributed filters for sampling of the complex field shown in Figures 6, 7, and 8**

	Computations			Communication
	Robot	Fusion Center	Combined	
Centralized filter	–	$1.34 \times 10^6$	$1.34 \times 10^6$	320
Completely decentralized filter	$383.94 \times 10^6$	–	$1,535.77 \times 10^6$	$1,251.94 \times 10^3$
Federated decentralized filter	$333.86 \times 10^6$	$16.69 \times 10^6$	$1,352.13 \times 10^6$	$834.62 \times 10^3$
Federated distributed filter	$5.51 \times 10^6$	$16.69 \times 10^6$	$38.73 \times 10^6$	$121.02 \times 10^3$

contribution relative to the total contribution of all the Gaussians. These influencing Gaussians are selected whenever their relative percent contributions exceed a very small threshold chosen to be equal to 0.001% in our simulation.

Table 3 illustrates the number of computations and communications involved in the above simulations. For the federated distributed filter, it is assumed that on the average, each collected sample influences the estimate of

10 neighboring Gaussians, and each communication update transmits the estimates of 15 Gaussians. Hence, the average number of parameters that can change after each sample is  $M = 41$ , since there are 10 Gaussians, 4 parameters per Gaussian and 1 free offset parameter (i.e.,  $M = 4B + 1 = 4 \times 10 + 1$ ). Furthermore in our simulation we are assuming that the number of all the parameters expected to change is equal to the number of all the parameters that actually change, i.e.,  $S = C =$



$4B + 1 = 4 \times 15 + 1$ . Using the formulae shown in Table 2 to calculate the number of computations and communication, the results we get for the federated decentralized case are, respectively, 1.14 and 1.5 times smaller than their counterparts in the completely decentralized case. Moreover, the number of computations and communication in the federated distributed case are, respectively, 35 and 7 times smaller than their counterparts in the federated decentralized case.

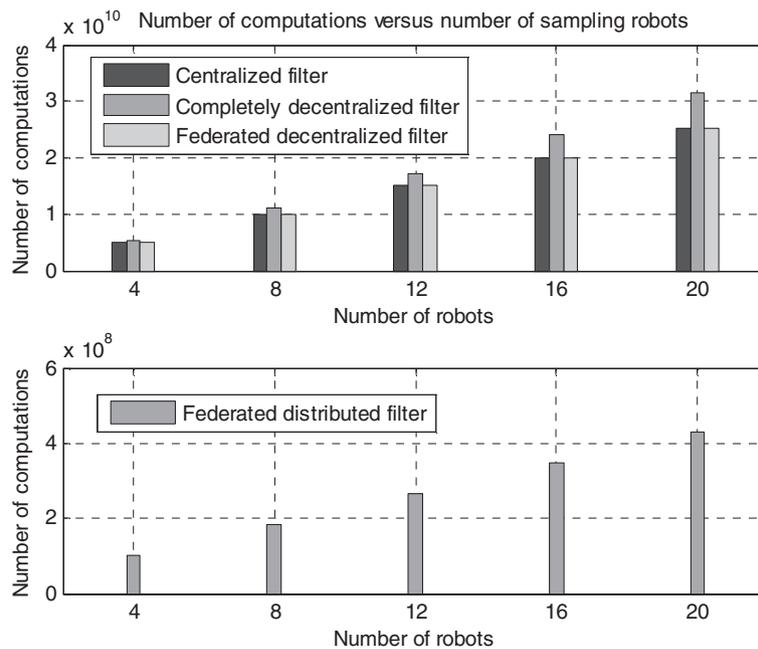
**Scalability**

The scalability of the proposed federated distributed algorithm is discussed here by comparing the numbers of computations and packets communicated (i.e., the computational and communication load) in two different scenarios, as explained below.

- i. The number of sampling robots increases but the number of field parameters is kept unchanged. As the number of sampling robots increases, the computational and communication load increases almost linearly in the case of both the federated decentralized and distributed filters, whereas for the completely decentralized filter, this load increases quadratically. Figures 9 and 10, respectively, show that the computational and communication loads increase when the number of robots used increases from 4 to 20 for all 4 types of filter structures.
- ii. The number of parameters representing the field increases but the number of robots remains

unchanged. This scenario may represent different cases where either a highly complex field is used which requires a large number of parameters for its description but does not necessarily cover a wide area or a field that is modestly complex but ranges over a very wide area or possibly a field that combines both features. If the field is spread over a wide area, and the number of robot is kept unchanged, then it would require more time to reconstruct the field and the number of computations and communications would depends on the number of parameters used to represent the field.

Figures 11 and 12, respectively, show the computational and communicational loads when the increasing numbers of parameters used are 161, 241, 321, 401, and 481. These five scenarios reflect the cases where the field is represented with 40, 60, 80, 100, and 120 Gaussians, respectively. As shown in Table 2, the computational complexity is related cubically to the number of parameters. But, in the case of the distributed KF algorithm, the rate of increase is far smaller than the one for the other three filters as shown in Figure 11. This result is expected since, for the distributed KF filter, the complexity is proportional to  $M^3$  rather than to  $L^3$ , and  $M < L$ . The computational complexity can be further reduced by increasing the number of robots as the number of parameters increases as this will reduce the factor  $M$ .



**Figure 10** Number of communications versus number of sampling robots when the number of parameters representing the field remains unchanged.

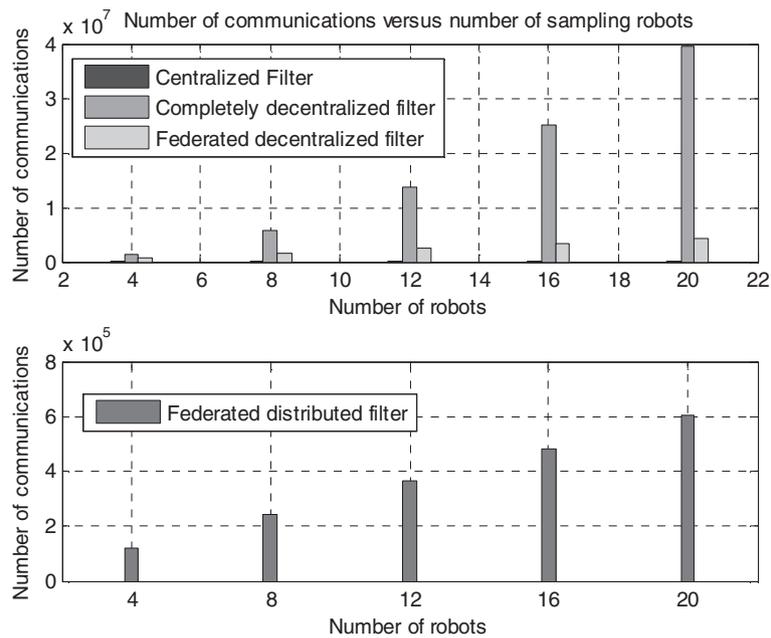


Figure 11 Number of computations versus number of field parameters when number of sampling robots remains unchanged.

The communication complexity is related quadratically to the number of parameters for the completely decentralized and federated decentralized filters. For the centralized filter, this complexity is not a function of the number of parameters, because it is the measurement  $Z$ , rather than the parameter estimate, that is transmitted. For the federated distributed filter, the communication complexity is

related quadratically to the number of parameters. However, when the number of parameters increases, the rate of growth of the communication load is smaller than the corresponding rate for the completely decentralized and federated decentralized filters. The reason for this is that it is  $M$  and  $C$ , rather than the larger  $L$ , that are, respectively, used in the last two entries in the

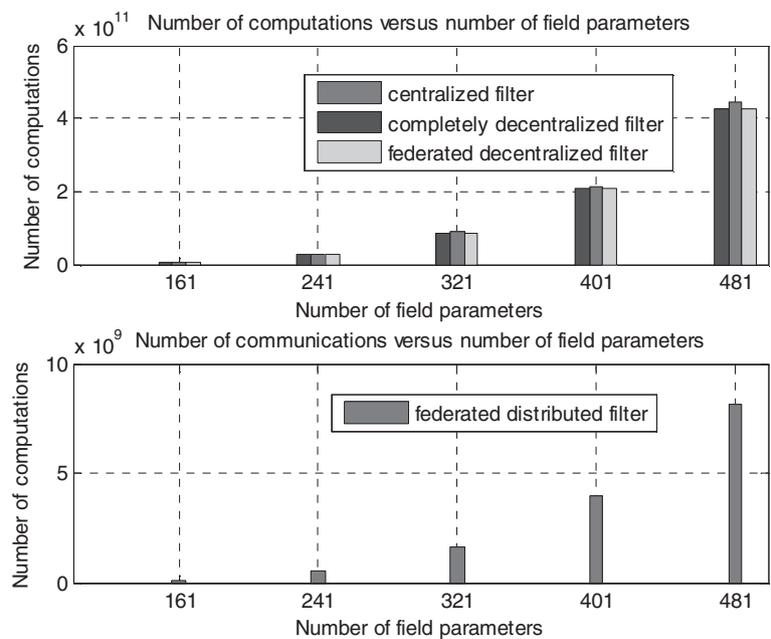


Figure 12 Number of communications versus number of field parameters when number of sampling robots remains unchanged.

columns titled: “Combined” and “Communications” in Table 2.

## Conclusion

In this article, we studied the problem of estimating the field distribution of some particular environmental variable (e.g., moisture, salinity, etc.) using both single-robot and multi-robot AS schemes and different filtering structures, such as the centralized and decentralized ones as well as our proposed federated distributed filtering structure. Our thorough simulation study, encompassing various AS schemes, clearly showed the superiority of using multi-robot-based AS schemes over their single-robot-AS counterparts.

These attractive advantages enjoyed by the multi-robot AS schemes are mainly due to their features of parallel sampling, a wider area coverage and a decentralization scheme offered by the multi-robot approach. We proposed a novel scalable structure termed the decentralized distributed filter approach where the full-order local KF filter used in the conventional decentralized approach has been distributed into several low-order KFs, thus leading to a further vital reduction in the field reconstruction time. Our simulation results corroborated very well our expectations of the higher performance of our novel decentralization-cum-distribution approach since the estimates of the communication and computational loads on the  $N$  robots used show that a dramatic in-excess of- $N$ -fold reduction in the sampling time can be achieved, leading to a similar reduction in the field reconstruction time. These very encouraging results provide us with ample encouragement to further investigate both the efficiency and convergence properties of our proposed distributed filter scheme. This analytical investigation as well as our ultimate goal of successfully testing our proposed approach on a physical multi-robot system is both currently under way.

## Competing interests

The authors declare that they have no competing interests.

## Acknowledgment

This study was supported by the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, Projects # JF090014 and SB101017.

## Author details

<sup>1</sup>Systems Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia. <sup>2</sup>Electrical Engineering Department, The University of Texas at Arlington, Arlington, TX 76019, USA.

Received: 21 July 2011 Accepted: 30 May 2012

Published: 18 July 2012

## References

1. W. Jatmiko, K. Sekiyama, T. Fukuda, A mobile robots PSO-based for odor source localization in dynamic advection-diffusion environment, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2006), pp. 4527–4532
2. V.N. Christopoulos, S. Roumeliotis, Adaptive sensing for instantaneous gas release parameter estimation, in *IEEE International Conference on Robotics and Automation* (2005), pp. 4450–4456
3. D.A. Robinson, C.S. Campbell, J.W. Hopmans, B.K. Hornbuckle, S.B. Jones, R.O. Knight, F. Ogden, J. Selker, O. Wendroth, Soil moisture measurement for ecological and hydrological watershed-scale observatories: a review. *Vadose Zone J.* **7**, 358–389 (2008)
4. C.J. Cannell, D.J. Stilwell, A comparison of two approaches for adaptive sampling of environmental processes using autonomous underwater vehicles, in *Proceedings of MTS/IEEE OCEANS* (2005), pp. 1514–1521
5. N.E. Leonard, D. Paley, F. Lekien, R. Sepulchre, D.M. Fratantoni, R. Davis, Collective motion, sensor networks and ocean sampling. *Proc. IEEE* **95**(1), 48–74 (2007)
6. M.F. Mysorewala, D.O. Popa, Multi-scale adaptive sampling with mobile agents for mapping of forest fires. *J. Intell. Robot. Syst.* **54**(4), 535–565 (2009)
7. V. Hombal, A.C. Sanderson, R. Blidberg, A non-parametric iterative algorithm for adaptive sampling and robotic vehicle path planning, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2006), pp. 217–222
8. A. Singh, A. Krause, C. Guestrin, W. Kaiser, Efficient informative sensing using multiple robots. *J. Artif. Intell. Res. (JAIR)* **34**, 707–755 (2009)
9. M.A. Demetriou, I.I. Hussein, Estimation of spatially distributed processes using mobile spatially distributed sensor network. *SIAM J. Control. Optim.* **48**, 266–291 (2009)
10. S. Martinez, Distributed interpolation schemes for field estimation by mobile sensor networks. *IEEE Trans. Control. Syst. Technol.* **18**(2), 491–500 (2010)
11. NAC Cressie, *Statistics for Spatial Data*, Revisedth edn. (Wiley, New York, 1993)
12. M.L. Stein, *Interpolation of Spatial Data. Some Theory for Kriging*. Springer Series in Statistics (Springer, New York, 1999)
13. J. Cortes, Distributed Krige Kalman filter for spatial estimation. *IEEE Trans. Automat. Control* **54**(12), 2816–2827 (2009)
14. R. Graham, J. Cortes, Spatial statistics and distributed estimation by robotic sensor networks, in *American Control Conference (ACC)* (2010), pp. 2422–2427
15. R. Graham, J. Cortes, Cooperative adaptive sampling of random fields with partially known covariance. *Int. J. Robust Nonlinear Control* **22**(5), 504–534 (2012)
16. R. Olfati-Saber, Distributed Kalman filter with embedded consensus filters, in *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05* (2005), pp. 8179–8184
17. R. Olfati-Saber, Distributed Kalman filtering for sensor networks, in *46th IEEE Conference on Decision and Control* **2007**(12–14), 5492–5498 (2007)
18. A. Singh, D. Budzik, W. Chen, M. Batalin, M. Stealey, H. Borgstrom, W. Kaiser, Multiscale sensing: a new paradigm for actuated sensing of high frequency dynamic phenomena, in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006* (2006), pp. 328–335
19. A.G. Mutambara, *Decentralized Estimation and Control for Multisensor Systems, Chapters 2–3* (CRC Press, Boca Raton, 1998), pp. 19–79. doi:19
20. H.R. Hashmipour, S. Roy, A.J. Laub, Decentralized structures for parallel Kalman filtering. *IEEE Trans. Automat. Control* **33**(1), 88–93 (1988)
21. Y. Gao, E.Y. Krakiwsky, M.A. Abousalem, J.F. McLellan, Comparison and analysis of centralized, decentralized, and federated filters. *Navigation* **40**(1), 69–86 (1993)
22. M.F. Mysorewala, L. Cheded, M.S. Baig, D.O. Popa, A distributed multi-robot adaptive sampling scheme for complex field estimation, in *11th International Conference on Control Automation Robotics & Vision (ICARCV) 2010, 7–10* (2010), pp. 2466–2471
23. D.O. Popa, M.F. Mysorewala, F.L. Lewis, EKf-based adaptive sampling with mobile robotic sensor nodes, in *International Conference on Intelligent Robots and Systems, 2006 IEEE/RSJ* (2006), pp. 2451–2456
24. Q. Du, V. Faber, M. Gunzburger, Centroidal voronoi tessellations: applications and algorithms. *SIAM Rev.* **41**, 637–676 (1999)
25. J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms* (Kluwer Academic Publishers (Norwell, MA, 1981)
26. B.S. Rao, H.F. Durrant-Whyte, Fully decentralized algorithm for multisensor Kalman filtering. *IEE Proc. Control Theory Appl.* **138**, 413–420 (1991)
27. N.A. Carlson, Federated square root filter for decentralized parallel processes. *IEEE Trans. Aerospace Electron. Syst.* **26**(3), 517–525 (1990)
28. D.O. Popa, A.C. Sanderson, R.J. Komerska, S.S. Mupparapu, D.R. Blidberg, S.G. Chappel, Adaptive sampling algorithms for multiple autonomous

underwater vehicles, in *Autonomous Underwater Vehicles* (2004), pp. 108–118

29. M.F. Mysorewala, L. Cheded, A. Qureshi, Comparison of nonlinear filters for the estimation of parametrized spatial field by robotic sampling, in *6<sup>th</sup> IEEE conference on Industrial Electronics and Applications* (2011), pp. 2005–2010
30. S. Haykin, *Neural Networks: A Comprehensive Foundation*, Secondth edn. (Prentice Hall PTR, 1998)

doi:10.1186/1687-1499-2012-223

**Cite this article as:** Mysorewala et al.: A distributed multi-robot adaptive sampling scheme for the estimation of the spatial distribution in widespread fields. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:223.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Immediate publication on acceptance
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

---

Submit your next manuscript at ▶ [springeropen.com](http://springeropen.com)

---