Distributed Fusion Algorithm for Passive Localization of Multiple Transient Emitters

WENBO DOU
YAAKOV BAR-SHALOM
LANCE KAPLAN
JEMIN GEORGE

This paper investigates the problem of deploying a network of passive sensors to estimate the positions of an unknown number of stationary transient emitters. Since a completely connected network, which has a link between every pair of nodes, is not feasible because of the power and bandwidth constraints, we developed a distributed algorithm that relies only on local communications between neighboring sensors. This distributed algorithm requires information diffusion within the network with the goal that every node achieves all target location estimates as accurate as a fusion center with centralized access to all information. The locations of the emitters are not completely observable by any single sensor since bearings and times of arrival with origin uncertainty are the only available measurements. These measurements are modeled as a realization of a Poisson point process at each sensor. The problem is formulated as a constrained optimization problem, which is solved via an alternating direction method of multipliers in a distributed manner based on the expectation maximization and averaging consensus algorithms. Consensus on the number of candidate targets as well as the inter-node estimate association are addressed so that the distributed algorithm converges to the maximum likelihood estimate. A likelihood function based approach using the estimated probability of detection is presented to determine the number of targets. Simulation results show that the distributed algorithm converges very fast and the root mean square error of target locations is almost as small as that obtained using the centralized algorithm. It is also shown that one can accurately determine the number of targets using the estimated probability of detection.

Manuscript received September 18, 2016; revised January 26, 2017 and March 1, 2017; released for publication March 7, 2017.

Refereeing of this contribution was handled by Chee-Yee Chong.

Authors' addresses: W. Dou and Y. Bar-Shalom, Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269 (E-mail: wenbo.dou@uconn.edu, ybs@engr.uconn.edu). L. Kaplan and J. George, U.S. Army Research Laboratory, 2800 Powder Mill Rd., Adelphi, MD 20783 (E-mail: {lance.m.kaplan.civ, jemin.george.civ}@mail.mil).

Supported by ARO Grant W991NF-10-1-0369.

1. INTRODUCTION

1.1. Background

This paper considers the problem of multiple transient emitter (target) localization using a wireless sensor network (WSN). One particular application is to utilize a network of acoustic gunfire detection systems mounted on a group of soldiers to localize adversaries in a battlefield [16][17]. It is assumed that the targets are stationary during the time window of interest but the number of the targets is unknown. The sensors can measure the line of sight (LOS) angles to the targets by detecting their emitted acoustic signals and record the times of arrival (TOAs) of the detected signals. This implies incomplete target location observability for any single sensor. Missed detections and false alarms are present due to the imperfection of the sensors. Furthermore, the associations between the measurements and the targets are unknown, that is, each sensor does not know from which target (or clutter) a particular measurement originates. Before estimating the position of any target, one has to associate the measurements from all the sensors. Therefore, the quality of data association is critical to the overall localization performance.

Two different fusion algorithms developed in our previous work [13] solved this problem using a centralized approach, i.e., we assumed that there is a fusion center collecting all the information from individual sensors either directly or by multi-hop relay, typically by wireless communication. Centralized access to all information can be difficult. For example, it requires a high transmission power to deliver the information from a single sensor directly to a fusion center in applications covering a large area. Moreover, the fusion center based approach is not robust, i.e., if the fusion center fails, the whole system fails. This has motivated a lot of work on distributed fusion or distributed optimization algorithms including the one presented in this paper.

One straightforward distributed solution is flooding, i.e., broadcasting the actual sensor measurements through the links in the network. In [7], a communication strategy of broadcasting new measurements was presented to allow distributed measurement fusion, which produces the optimal estimate at each node given all the measurements received up to any time for a linear dynamic system. For the localization problem considered in this paper, one has a nonlinear static system. The flooding approach still applies, by careful bookkeeping and a number of iterations of information exchange, each sensor would have all the information and can act as a fusion center to find the same global solution as a centralized approach. This method requires a large amount of data communication, storage memory, and bookkeeping overhead. For instance, it requires about S (the number of sensors) times the memory storage of the average consensus (AC) based approach.

TABLE I
Classification of the various versions of the shooter localization problem.

	Single target	Multiple targets
No missed detections or false alarms	P1	P5
Only missed detections exist	P2	P6
Only false alarms exist	P3	P7
Missed detections and false alarms exist	P4	P8

When it is used for the localization problem, the flooding approach is distributed in the sense that the information (all the measurements) is communicated in a distributed manner but it is centralized in the sense that the estimation algorithm including all computations is applied on all the information collected at every node, i.e., the flooding approach is a multiple replica of the centralized approach. In this paper, we present a consensus based algorithm that is different from the flooding approach and is distributed in the sense that both communication and estimation are performed in a distributed manner.

One of our approaches in [13] formulated the localization problem as an optimization problem and solved it using the expectation maximization (EM) algorithm. We observe that two types of subproblem are solved in the EM algorithm. One is to compute the average of variables with one variable from one sensor and the other is to solve a nonlinear least squares problem. Both subproblems can be formulated to optimize a global objective function, which can be written as a sum of local objective functions. Such problems can be solved using distributed optimization approaches whose goal is to recover the optimal global solution without any global coordination or interactions (like using a fusion center). Their solutions often contain a step where the sum or average of some quantity needs to be calculated and this can be achieved by an average consensus (AC) based approach.

The average consensus based approach with communication only between the one-hop neighbors scales well in that the communication overhead per sensor can be kept at an affordable level as the size of the network increases. Unlike the full flooding approach, which requires the local variables labeled with their origins, the average consensus approach does not need such labels and therefore uses less storage. If new nodes join the network, our consensus based distributed algorithm does not need to restart the whole process because the local variables can be updated following a (mini) flooding of only the new information.

In this paper, we assume that centralized access of all the information is not possible and we are interested in solving the problem of multiple transient emitter localization using an alternative algorithm that is different from the flooding approach and that is distributed in the sense that both communication and estimation are performed in a distributed manner. Since the goal is to have each sensor obtain a global estimate (which is a vector consisting of the number of targets and the position estimates of all targets) as good (or almost as good) as can be obtained by a fusion center using a centralized algorithm, information diffusion either in the form of raw measurements or in the form of some intermediate estimates (a function of raw measurements) within the network is necessary. Instead of using the raw measurement diffusion approach as in the flooding approach, we diffuse the intermediate estimates using the average consensus approach, i.e., the estimation is also performed in a distributed manner.

Without a fully connected network (each node can reach each other node via one or multiple "hops"), sending raw measurements to all nodes in order to achieve global optimal solution is a difficult task which requires "subnetwork" coordination, which is beyond the scope of this paper (multiple layers would be necessary). Therefore, we assume that the network is fully connected, i.e., there is a (not necessarily direct) path between every two sensors. If the network is not connected and has more than one connected subnetwork due to node or link failures, each subnetwork can be processed by our distributed algorithm independently. In such case, the consensus is achieved within each connected subnetwork.

Table I presents a classification of the various versions of the shooter localization problem. In view of the above discussion, it is necessary to develop a distributed algorithm to solve the problem P8 in Table I relying solely on local communications between one-hop neighboring sensors. Problems P3, P4, P6 and P7 are special cases of P8, therefore can be solved by the same distributed algorithm. Problem P1 is addressed in Section 2.7. Problems P2 are P5 are special extensions of P1 and will not be covered here.

1.2. Related Work

Distributed data fusion strategies, such as methods in [6], [8], [9], [10], [14], and [21] among others, are available for joint state estimation and data association in multi-sensor multi-target tracking scenarios. Since they are recursive algorithms that require sequential measurements and provide solutions to dynamic data association problems, they cannot be employed to solve joint parameter estimation and data association in a multi-sensor multi-target localization scenario (with incomplete observability at each sensor) considered in this paper. While most of the distributed estimation work in the literature assumes linear measurement models, our paper deals with nonlinear and incomplete target location measurements (direction of arrival and delayed arrival time). Although, one could imagine linearizing the localization problem and sharing messages between the nodes, we suspect that the linearization will probably

cause more errors than the distributed ADMM and will investigate this in our future work. Related work from robotics can be found in [19]. A recent comparison of optimal distributed estimation and consensus filtering for dynamic systems was done in [7].

A multi-dimensional assignment formulation assuming a Bernoulli measurement generation model that the number of measurements from each target received at each sensor is a Bernoulli random variable with parameter equal to the probability of detection as well as a cardinality selection formulation assuming a Poisson measurement generation model that the number of measurements from each target received at each sensor is a Poisson random variable with parameter equal to the probability of detection were considered in the centralized fusion algorithms [13] to solve the same problem of multiple transient emitter localization. This paper only considers developing a distributed algorithm to solve the cardinality selection problem assuming a Poisson measurement generation model[1] and leaves distributed multi-dimensional assignment algorithms for future work. While a list of measurements at each sensor was modeled as either realizations of a random variable with a mixture density or a Poisson point process (PPP) in [13], only PPP modeling is considered in this paper due to its simpler mathematical solution expression. Since the centralized algorithm solving the cardinality selection problem, which combines expectation maximization (EM) algorithm to estimating target parameters given a fixed number of targets and information criterion for selection of the best possible number of targets, is not amendable to a distributed implementation, it is necessary to develop a distributed EM algorithm.

Distributed EM algorithms have attracted a lot of attentions in sensor network applications for density estimation, data clustering and target tracking. For a fixed number of target, the localization problem can be considered as a density estimation problem. An incremental distributed EM algorithm presented in [23] is the first known scheme for density estimation and clustering in distributed sensor network. A distributed EM algorithm based on the averaging consensus filtering was developed in [18] for particle filter based target tracking. A distributed EM algorithm based on alternating direction method of multipliers (ADMM) was proposed in [15] for distributed data clustering. However, all these works assumes a *linear* generative model for their respective applications, which does not apply to a *nonlinear* generative model (see the measurement model in (78)) considered here due to the incomplete position measurement based on bearings and TOAs in the emitter localization scenario considered in this paper. Moreover, the parameters in these distributed EM algorithms are initialized to be either fixed values (zeros) or random values. This initialization approach was shown to be useless for our centralized EM algorithm, which requires an initialization based on the sequential m-best 2-D assignment algorithm applied on the lists of measurements from all sensors for the convergence to the global maximum.

1.3. Contributions

In this paper, we develop a distributed EM algorithm to solve the same problem as considered in [13] but in a distributed manner. The distributed processing introduces a number of challenges.

Firstly, the convergence of an EM algorithm (whether being centralized or distributed) depends highly on the initialization step. Previous studies on developing distributed EM algorithms assumed a linear measurement model and thus the initialization with fixed values (such as zeros) or random values, which is commonly used, works fine. This initialization does not work in the problem considered in this paper where the measurements (incomplete position observations) are nonlinear functions of target locations. Our earlier work shows that the assignment based initialization leads to global convergence. However, due to limited connections in a distributed setting, each sensor can only obtain a different EM initialization, which is a set of vectors, using the sequential m-best 2-D assignment algorithm on the measurement lists of its own and its neighbors (a subset of all the lists of measurements). For the global convergence of the EM algorithm, we developed a distributed set consensus algorithm ensuring that every node has the same initialization (the same number of targets and the same target locations).

Secondly, the maximization step in the standard EM algorithm has to be evaluated in a distributed manner. Although the probability of detection can be estimated by a distributed averaging consensus subroutine and the locations of the targets can be estimated by a distributed ADMM subroutine, this would result in a nested iterative algorithm with two subroutines being iterative algorithms themselves. Even more challenging, these two subroutines are needed for a number of iterations and at each iteration both of them requires local communications between sensors for a number of times, which would result in a very high communication cost. Instead, we manage to formulate a constrained optimization problem with equality conditions that force all local variables to be identical and developed a new distributed ADMM algorithm enabling a lower communication cost at the expense of additional local computation. The EM and AC based distributed ADMM algorithm is a generalization of previous distributed algorithms allowing

[1]The Bernoulli measurement generation model is more realistic than the Poisson measurement generation model. Therefore, the Bernoulli model is used to generate the synthetic data for the evaluation of the developed algorithm, whereas the Poisson model is assumed in the derivation of the developed algorithm. Using the Bernoulli model in the algorithm would make it excessively complicated because of the need to use multidimensional assignment.

the handling of the nonlinear and incomplete measurement models such as bearings in the passive sensing applications as here.

Last but not least, since we feel that a Bernoulli measurement generation model is a more realistic assumption and it reflects best the physical process of measurement generation, we used a likelihood function based thresholding approach to determine the number of targets.

1.4. Paper Organization

The remaining sections of this paper are organized as follows. Section 2 presents some preliminaries required for the development of the desired distributed algorithm. These include (i) graph modeling, (ii) a distributed AC algorithm for both single parameter estimation and multiple parameter estimation, (iii) data association test for two estimates as well as two sets of estimates, (iv) an algorithm of alternating direction method of multipliers and (v) a distributed nonlinear least squares algorithm, which can solve problem P1 in Table I. Section 3 formulates the problem by modeling each measurement set as a realization of a Poisson point process. Section 4 reviews a recently developed centralized algorithm that uses an EM algorithm to estimate the location and emission time parameters for a fixed number of targets. The distributed algorithm for problem P8 is presented in Section 5. The initialization issues of this algorithm—how to reach the consensus on the number of targets and how to reach the consensus on the target-estimate association—are discussed in Sections 5.1 and 5.2, respectively. An EM and AC based distributed ADMM algorithm is developed in Section 5.3. Section 5.4 describes a thresholding approach to distinguish real target estimates from false target estimates using the estimated probability of detection values. Section 6 presents and analyzes simulation results and Section 7 concludes the paper.

2. PRELIMINARIES

2.1. Graph Model

A wireless sensor network with S nodes (sensors) is deployed to collect data and perform data association and parameter estimation task. Every node is only able to communicate with its neighbors. Mathematically, this network can be modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the set of nodes

$$\mathcal{V} = \{v_1, v_2, \ldots, v_S\} \tag{1}$$

and the set of edges $\mathcal{E}$, where an edge $(v_i, v_j) \in \mathcal{E}$ is an unordered pair of distinct nodes, representing a two-way communication link between v_i and v_j. The graph $\mathcal{G}$ is assumed connected, meaning that there is a path between any two nodes. The set of neighbors of node v_i is defined as

$$\mathcal{N}_i = \{v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}\} \tag{2}$$

The degree of node v_i is defined as

$$d_i = |\mathcal{N}_i| \tag{3}$$

where $|\cdot|$ denotes the set cardinality. The maximum degree of the graph $\mathcal{G}$ is defined as

$$d_{\max} = \max_i d_i \tag{4}$$

The Laplacian matrix L of the graph $\mathcal{G}$ is defined as

$$L_{ij} = \begin{cases} -1 & \text{if } v_j \in \mathcal{N}_i \\ d_i & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

2.2. Distributed Averaging Consensus Algorithm

Suppose a wireless sensor network with S nodes is deployed to estimate an unknown constant parameter $x \in \mathbf{R}^n$. Each node v_i makes a measurement

$$z_i = x + w_i \tag{6}$$

where w_i are independent, identically distributed, normal, zero mean, and with a known identity covariance matrix I. The maximum likelihood estimate of x is $(1/S)\sum_{i=1}^{S} z_i$, which is the mean vector of all measurements z_i. This estimate can be obtained by the following distributed averaging consensus algorithm.

Let us denote an initial value (z_i for the estimate problem) at node v_i by $u_i(0) \in \mathbf{R}^n$ at time $t = 0$. The matrix formed by the column vectors at all nodes is denoted as

$$\mathbf{U}(0) = [u_1(0) \quad u_2(0) \quad \ldots \quad u_S(0)]^T \in \mathbf{R}^{S\times n} \tag{7}$$

The goal of distributed averaging consensus is to make every node obtain the mean vector $(1/S)\sum_{i=1}^{S} u_i(0)$ eventually after gradually updating its value with a linear combination of its previously stored value and the values of its neighbors. One iteration of the process can be represented with a weight matrix W as

$$u_i(t+1) = W_{ii}u_i(t) + \sum_{j\in\mathcal{N}_i} W_{ij}u_j(t) \quad i = 1, \ldots, S \tag{8}$$

where $t = 0, 1, \ldots$ is the discrete time index, and W_{ij} is the weight on u_j at node v_i. Setting $W_{ij} = 0$ for $j \notin \mathcal{N}_i$, this iteration can be written in matrix form as

$$\mathbf{U}(t+1) = W\mathbf{U}(t) \tag{9}$$

and W is selected such that

$$\lim_{t\to\infty} \mathbf{U}(t) = \frac{1}{S}\mathbf{1}\mathbf{1}'\mathbf{U}(0) \tag{10}$$

The best constant edge weight matrix is given by [27]

$$W = I - \beta L \tag{11}$$

with

$$\beta = \frac{2}{\eta_1(L) + \eta_{S-1}(L)} \tag{12}$$

where $\eta_1(L)$ and $\eta_{S-1}(L)$ are the largest and second smallest eigenvalues of L, respectively.

In some cases, each node only has the knowledge of its neighbors rather than the connectivity of the entire network. It is more suitable to use the Metropolis weight matrix, which is defined as [28]

$$W_{ij} = \begin{cases} \dfrac{1}{1+\max\{d_i,d_j\}} & \text{if } v_j \in \mathcal{N}_i \\ 1-\sum_{v_k\in\mathcal{N}_i} W_{ik} & \text{if } j=i \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

2.3. Distributed Averaging Consensus Algorithm for Multiple Parameter Estimation with Unknown Data Association

Suppose a WSN with S nodes is used to estimate a set of N unknown constant parameters

$$X = \{x_1, x_2, \ldots, x_N\} \tag{14}$$

with each $x_j \in \mathbf{R}^n$. Each node v_i has a set of N measurements

$$Z_i = \{z_{i1}, z_{i2}, \ldots, z_{iN}\} \tag{15}$$

with one for each x_j. Let Π_N denote all permutations of the set $\{1,2,\ldots,N\}$, then the jth measurement of node v_i is

$$z_{ij} = x_{\pi_i(j)} + w_i \tag{16}$$

where $\pi_i \in \Pi_N$ is a permutation[2] at node v_i, and w_i are independent, identically distributed, normal, zero mean measurement noises with a known identity covariance matrix I.

Since the second index j of z_{ij} in the set Z_i contains no labeling information, one needs to perform data association and weighted averaging update (8) simultaneously for multiple parameter estimation. Let us denote the stacked vector at node v_i at time t as

$$\mathbf{u}_i(t) = [u_{i1}^T(t), u_{i2}^T(t), \ldots, u_{iN}^T(t)]^T \tag{17}$$

and $u_{ij}(0)$ is initialized as z_{ij}. At time t, node v_i calculates an optimal permutation[3] π_{ji} for each of its neighbor nodes v_j as

$$\pi_{ji} = \arg\min_{\pi\in\Pi_N} \sum_{k=1}^{N} \|u_{ik}(t) - u_{j\pi(k)}(t)\|^2 \tag{18}$$

Then node v_i updates each segment of its stacked vector (17) as

$$u_{ik}(t+1) = W_{ii}u_{ik}(t) + \sum_{j\in\mathcal{N}_i} W_{ij}u_{j\pi_{ji}(k)}(t) \tag{19}$$

where the index $\pi_{ji}(k)$ refers to the segment of the stacked vector at node v_j that associates with the kth segment of the stacked vector at node v_i according to the permutation π_{ji} (18), and the weight matrix is given by (13).

[2]It is a one-to-one mapping function from an ordered set $\{1,2,\ldots,N\}$ to a particular permutation of this set.

[3]The second index i of π_{ji} indicates that the optimal permutation is obtained with respect to $\mathbf{u}_i(t)$.

2.4. Association Test for Two Estimates

Suppose that sensor v_i has an unbiased estimate $\hat{x}_i$ of the n-dimensional (unknown) parameter x_i with a covariance matrix P_i and sensor v_j has an unbiased estimate $\hat{x}_j$ of the n-dimensional (unknown) parameter x_j with a covariance matrix P_j. We are interested in testing whether $x_i = x_j$. Let us denote the difference of the two estimates as

$$\hat{\Delta}_{ij} = \hat{x}_i - \hat{x}_j \tag{20}$$

which is the estimate of the difference of the parameters

$$\Delta_{ij} = x_i - x_j \tag{21}$$

Since the estimation errors

$$\tilde{x}_i = x_i - \hat{x}_i \tag{22}$$

$$\tilde{x}_j = x_j - \hat{x}_j \tag{23}$$

are zero-mean, the estimation error of the difference of the parameters

$$\tilde{\Delta}_{ij} = \Delta_{ij} - \hat{\Delta}_{ij} = \tilde{x}_i - \tilde{x}_j \tag{24}$$

is also zero-mean and it has the covariance matrix

$$\begin{aligned} T_{ij} &= E\{\tilde{\Delta}_{ij}\tilde{\Delta}_{ij}^T\} = E\{(\tilde{x}_i - \tilde{x}_j)(\tilde{x}_i - \tilde{x}_j)^T\} \\ &= P_i + P_j - E\{\tilde{x}_i\tilde{x}_j^T\} - E\{\tilde{x}_j\tilde{x}_i^T\} \end{aligned} \tag{25}$$

If $\tilde{x}_i$ and $\tilde{x}_j$ are independent, then we have

$$T_{ij} = P_i + P_j \tag{26}$$

Assuming that $\tilde{x}_i$ and $\tilde{x}_j$ are Gaussian, the normalized estimation error squared (NEES) [2] for Δ

$$\epsilon_{ij} \triangleq \tilde{\Delta}_{ij}^T T_{ij}^{-1} \tilde{\Delta}_{ij} \tag{27}$$

is chi-square distributed with n degrees of freedom.

The null hypothesis that the two parameters are the same and the alternative hypothesis are

$$H_0 : \Delta = 0 \tag{28}$$

$$H_1 : \Delta \neq 0 \tag{29}$$

Under H_0 ($\Delta = 0$), we have the following

$$\tilde{\Delta}_{ij} = -\hat{\Delta}_{ij} \tag{30}$$

$$\epsilon_{ij} = \hat{\Delta}_{ij}^T T_{ij}^{-1} \hat{\Delta}_{ij} \tag{31}$$

Therefore, the test of H_0 vs. H_1 is as follows. If

$$\hat{\Delta}_{ij}^T T_{ij}^{-1} \hat{\Delta}_{ij} \leq F_{\chi_n^2}^{-1}(1-\alpha) \tag{32}$$

where $F_{\chi_n^2}^{-1}(\cdot)$ is the inverse of the cumulative distribution function (cdf) of a chi-square random variable with n degrees of freedom, we will not reject H_0 at a significance level of α. Then it is likely that $\hat{x}_i$ and $\hat{x}_j$ are estimates of the same parameter.

2.5. Association Test for Two Sets of Estimates

Suppose that there are N unknown n-dimensional constant parameters

$$X = \{x_1, x_2, \ldots, x_N\} \tag{33}$$

Sensor v_i has a set of N_i estimates with corresponding covariance matrices

$$\hat{x}_i = \{\hat{x}_{i1}, \hat{x}_{i2}, \ldots, \hat{x}_{iN_i}\} \tag{34}$$

$$\mathcal{P}_i = \{P_{i1}, P_{i2}, \ldots, P_{iN_i}\} \tag{35}$$

Similarly, sensor v_j has N_j estimates with corresponding covariance matrices

$$\hat{x}_j = \{\hat{x}_{j1}, \hat{x}_{j2}, \ldots, \hat{x}_{jN_j}\} \tag{36}$$

$$\mathcal{P}_j = \{P_{j1}, P_{j2}, \ldots, P_{jN_j}\} \tag{37}$$

We assume that each sensor has at most one estimate for a particular parameter and the estimation errors are independent.

If $\hat{x}_{ik}$ and $\hat{x}_{j\ell}$ are estimates of the same parameter, then the NEES

$$d_{k\ell} = (\hat{x}_{ik} - \hat{x}_{j\ell})^T (P_{ik} + P_{j\ell})^{-1} (\hat{x}_{ik} - \hat{x}_{j\ell}) \tag{38}$$

can be regarded as a distance measure between $\hat{x}_{ik}$ and $\hat{x}_{j\ell}$. A small value of $d_{k\ell}$ indicates a high probability of both being the estimates of the same parameter.

To deal with incomplete associations caused by missed detections, we add dummy estimates $\hat{x}_{i0}$ and $\hat{x}_{j0}$ to the sets $\hat{x}_i$ and $\hat{x}_j$, respectively [24]. The distance involving a dummy estimate is defined as

$$d_{k0} = d_{0\ell} = F^{-1}_{\chi^2_n}(1 - \alpha) \tag{39}$$

for a small value (say, 0.01) of α.

To associate the estimates in set $\hat{x}_i$ with those in set $\hat{x}_j$, we solve a generalized 2-D assignment problem

$$\min_{\rho_{k\ell}} \sum_{k=0}^{N_i} \sum_{\ell=0}^{N_j} \rho_{k\ell} d_{k\ell} \tag{40}$$

subject to

$$\sum_{\ell=0}^{N_j} \rho_{k\ell} = 1 \quad \forall\ k = 1, 2, \ldots, N_i \tag{41}$$

$$\sum_{k=0}^{N_i} \rho_{k\ell} = 1 \quad \forall\ \ell = 1, 2, \ldots, N_j \tag{42}$$

$$\rho_{k\ell} \in \{0, 1\} \quad k = 0, 1, \ldots, N_i;\ \ell = 0, 1, \ldots, N_j \tag{43}$$

The modified auction algorithm [24] can be applied to the above problem.

The association results of $\hat{x}_{ik}$ are determined as follows.

If

$$\rho_{k0} = 1 \tag{44}$$

then $\hat{x}_{ik}$ is assigned to the dummy estimate $\hat{x}_{j0}$, that is, the probability that no estimate in $\hat{x}_j$ comes from the same parameter as $\hat{x}_{ik}$ is 0.99 for $\alpha = 0.01$. In this case, $\hat{x}_{ik}$ is not associated.

If

$$\rho_{k\ell} = 1 \tag{45}$$

then $\hat{x}_{ik}$ is associated with $\hat{x}_{j\ell}$.

The association results of $\hat{x}_{j\ell}$ are determined in a similar way.

2.6. The Alternating Direction Method of Multipliers (ADMM) Algorithm

Consider the following equality-constrained optimization problem

$$\min_{z,y} \{f(z) + g(y)\} \tag{46}$$

subject to

$$Az + By = c \tag{47}$$

with variables $z \in \mathbf{R}^p$ and $y \in \mathbf{R}^q$, where $A \in \mathbf{R}^{m \times p}$, $B \in \mathbf{R}^{m \times q}$ and $c \in \mathbf{R}^m$ are given.

The augmented Lagrangian of (46) is defined as

$$L_\rho(z, y, \lambda) = f(z) + g(y) + \lambda^T (Az + By - c) + \frac{\rho}{2} \|Az + By - c\|_2^2 \tag{48}$$

where λ is the dual variable or Lagrange multiplier and $\rho > 0$ is the penalty parameter.

The ADMM algorithm [5] solves (46) by iterating the following 3 steps

$$z^{k+1} \triangleq \arg\min_z L_\rho(z, y^k, \lambda^k) \quad z\text{-minimization} \tag{49}$$

$$y^{k+1} \triangleq \arg\min_y L_\rho(z^{k+1}, y, \lambda^k) \quad y\text{-minimization} \tag{50}$$

$$\lambda^{k+1} \triangleq \lambda^k + \rho(Az^{k+1} + By^{k+1} - c) \quad \text{dual update} \tag{51}$$

where ρ is used as the step size for the dual update and the superscript is the iteration counter.

In the ADMM, the variables z and y are updated in an alternating or sequential fashion instead of being minimized jointly, which accounts for the term *alternating direction*. Separating the minimization over z and y into two steps is precisely what allows for decomposition when f (or g) is separable with respect to a partition of the variable z (or y) into subvectors.

2.7. Distributed Nonlinear Least Squares Algorithm

This subsection presents a distributed solution to the problem P1 in Table I. We are interested in localizing a single target using the network $\mathcal{G}$ without missed detections or false alarms. Suppose each node v_i has a scalar measurement a_i from the target, we need to solve the unconstrained optimization problem

$$\min_x \sum_{i=1}^{S} (h(x) - a_i)^2 \tag{52}$$

TABLE II
Averaging consensus based distributed ADMM algorithm.

1: Node v_i initializes x_i^1 and $\lambda_i^1 = 0$
2: Compute $\bar{x}^1 = \frac{1}{S}\sum_{i=1}^{S} x_i^1$ via a distributed averaging consensus algorithm
3: **for** $k = 1,2,\ldots$ **do** until convergence
4: **for all** v_i **do**
5: Compute x_i^{k+1} via (71)
6: Compute $\bar{x}^{k+1} = \frac{1}{S}\sum_{i=1}^{S} x_i^{k+1}$ via a distributed averaging consensus algorithm
7: Compute λ_i^{k+1} via (72)
8: **end for**
9: **end for**

where $x \in \mathbf{R}^2$ is the parameter to be estimated (or the variable for the minimization), $h(\cdot)$ is a nonlinear function of x (for instance, $h(x)$ is an arctan function in a bearing-only localization problem) and S is the number of sensors.

Consider the constrained optimization problem, which is equivalent to (52)

$$\min_{x_1,x_2,\ldots,x_S} \sum_{i=1}^{S} (h(x_i) - a_i)^2 \tag{53}$$

subject to

$$x_1 = x_2 = \ldots = x_S = w \tag{54}$$

We can put (54) in the form of (47) by setting

$$z = [x_1^T \quad x_2^T \ldots x_S^T]^T \tag{55}$$

$$y = w \tag{56}$$

$$f(z) = \sum_{i=1}^{S} (h(x_i) - a_i)^2 \tag{57}$$

$$g(y) = 0 \tag{58}$$

$$A = I_{2S} \tag{59}$$

$$B = [-I_2 \quad -I_2 \cdots -I_2]^T \in \mathbf{R}^{2S\times 2} \tag{60}$$

$$c = 0 \tag{61}$$

Therefore, the augmented Lagrangian is

$$L_\rho(x_1,x_2,\ldots,x_S,w,\lambda) = \sum_{i=1}^{S} \left[(h(x_i) - a_i)^2 + \lambda_i^T (x_i - w) + \frac{\rho}{2}\|x_i - w\|_2^2 \right] \tag{62}$$

where

$$\lambda = [\lambda_1^T \ \lambda_2^T \ldots \lambda_S^T]^T \tag{63}$$

The z-minimization step (49) is

$$(x_1^{k+1},x_2^{k+1},\ldots,x_S^{k+1}) = \arg\min_{x_1,x_2,\ldots,x_S} L_\rho(x_1,x_2,\ldots,x_S,w^k,\lambda^k) \tag{64}$$

which can be carried out in a distributed fashion as

$$x_i^{k+1} = \arg\min_{x_i} (h(x_i) - a_i)^2 + \lambda_i^{kT}(x_i - w^k) + \frac{\rho}{2}\|x_i - w^k\|_2^2 \quad i = 1,2,\ldots,S \tag{65}$$

The y-minimization step (50) is

$$\begin{aligned} w^{k+1} &= \arg\min_{w} L_\rho(x_1^{k+1},x_2^{k+1},\ldots,x_S^{k+1},w,\lambda^k) \\ &= \arg\min_{w} \sum_{i=1}^{S} \left[\lambda_i^{kT}(x_i^{k+1} - w) + \frac{\rho}{2}\|x_i^{k+1} - w\|_2^2 \right] \\ &= \frac{1}{S}\sum_{i=1}^{S} x_i^{k+1} + \frac{1}{S\rho}\sum_{i=1}^{S} \lambda_i^k \end{aligned} \tag{66}$$

The dual update step (51) is

$$\lambda_i^{k+1} = \lambda_i^k + \rho(x_i^{k+1} - w^{k+1}) \quad i = 1,2,\ldots,S \tag{67}$$

If we carry out the summation of (67) over i and substitute w^{k+1} from (66), then

$$\sum_{i=1}^{S} \lambda_i^{k+1} = \sum_{i=1}^{S} \lambda_i^k + \rho\sum_{i=1}^{S} x_i^{k+1} - S\rho w^{k+1} = 0 \quad k \neq 0 \tag{68}$$

which means that the dual variables have average value zero after the first iteration. If the dual variables are initialized such that

$$\sum_{i=1}^{S} \lambda_i^1 = 0 \tag{69}$$

then, the y-minimization step simplifies to

$$w^{k+1} = \frac{1}{S}\sum_{i=1}^{S} x_i^{k+1} \triangleq \bar{x}^{k+1} \tag{70}$$

The simplified ADMM steps, in a distributed form, become

$$x_i^{k+1} \triangleq \arg\min_{x_i} [h(x_i) - a_i]^2 + \lambda_i^{kT}(x_i - \bar{x}^k) + \frac{\rho}{2}\|x_i - \bar{x}^k\|_2^2 \quad i = 1,2,\ldots,S \tag{71}$$

$$\lambda_i^{k+1} \triangleq \lambda_i^k + \rho(x_i^{k+1} - \bar{x}^{k+1}) \quad i = 1,2,\ldots,S \tag{72}$$

Based on the above ADMM steps, we obtain an averaging consensus based distributed algorithm as shown in Table II. Each node v_i stores and updates two vectors x_i and λ_i. At iteration $k = 1$, each node initializes a local parameter estimate x_i^1 and obtains $\bar{x}^1$ via a distributed averaging consensus algorithm as discussed in Section 2.2. The dual variables $\lambda_i^1 = 0$ are also initialized. During the kth iteration, each node updates its local parameter estimate x_i^{k+1} using (71). Next, each node reaches the consensus on $\bar{x}^{k+1}$, and subsequently, updates its local dual variable λ_i^{k+1} using (72), which concludes the kth iteration.

Reformulations of (52) other than (53) include [26] and [22], which result in a bridge-sensor based distributed ADMM and a coloring-scheme based distributed ADMM, respectively. However, either prior assignment of bridge sensors [26] or colors [22] is required for the respective algorithm to function properly. However, in these versions it is difficult to make a new assignment in case of node or link failures. Whereas, the averaging consensus based distributed ADMM algorithm does not require any feature assignment to individual nodes since it relies solely on information diffusion across the network.

3. PROBLEM STATEMENT AND FORMULATION

3.1. Problem Statement

Consider a scenario where there are N stationary targets located in $\mathbb{R}^2$. The target locations are denoted as

$$\mathbf{T} = (\mathrm{T}_1, \mathrm{T}_2, \ldots, \mathrm{T}_N) = \left(\begin{bmatrix} T_{x_1} \\ T_{y_1} \end{bmatrix}, \begin{bmatrix} T_{x_2} \\ T_{y_2} \end{bmatrix}, \ldots, \begin{bmatrix} T_{x_N} \\ T_{y_N} \end{bmatrix} \right) \quad (73)$$

The number of targets and their locations are unknown quantities of interest, to be estimated. A wireless sensor network consisting of S stationary nodes is deployed at known locations

$$\mathbf{S} = (\mathrm{S}_1, \mathrm{S}_2, \ldots, \mathrm{S}_S) = \left(\begin{bmatrix} S_{x_1} \\ S_{y_1} \end{bmatrix}, \begin{bmatrix} S_{x_2} \\ S_{y_2} \end{bmatrix}, \ldots, \begin{bmatrix} S_{x_S} \\ S_{y_S} \end{bmatrix} \right) \quad (74)$$

to perform this estimation task. There is one transient event occurring at each target location. Each node is able to observe these transient events by detecting the acoustic signals arising from them and measure the bearings to the targets and the TOAs of the received acoustic signals. The acoustic signal emission times are denoted as

$$\mathbf{t}^{\mathrm{e}} = (t_1^{\mathrm{e}}, t_2^{\mathrm{e}}, \ldots, t_N^{\mathrm{e}}) \quad (75)$$

For notational simplicity, let us denote

$$\mathbf{\Phi} = [\phi_1^T \quad \phi_2^T \ldots \phi_N^T]^T \quad (76)$$

where

$$\phi_i = [T_{x_i} \quad T_{y_i} \quad t_i^{\mathrm{e}}]^T \quad (77)$$

denotes the unknown 3-dimensional parameter of ith target.

If the transient events are separated significantly in time, the measurements from the same event will be close in time and the measurements from different events will also be separated significantly in time, and then the target locations can be estimated one at a time using the algorithm presented in 2.7. Therefore, we assume a more challenging situation that the transient events are close in time. In this case, the data association between the measurements and the targets has to be addressed before the network can fuse the measurements from a common origin to estimate the corresponding target location.

It is assumed that all measurements fall within a short time window W. Let m_ℓ denote the number of measurements (one measurement is defined as a vector consisting of both a bearing and a TOA due to one acoustic signal in this context) obtained by the ℓth sensor within the time window W. The jth measurement received by the ℓth sensor, if it is from the ith target at t_i^{e}, is

$$\mathbf{z}_{\ell j} = \mathbf{h}_\ell(\phi_i) + \mathbf{w}_{\ell j} \quad i = 1, \ldots, N; \quad \ell = 1, \ldots, S; \quad j = 1, \ldots, m_\ell \quad (78)$$

where $\mathbf{w}_{\ell j}$ is a zero mean white Gaussian measurement noise with a known diagonal covariance matrix R_ℓ and

$$\mathbf{h}_\ell(\phi_i) = \begin{bmatrix} \theta_{\ell i} \\ t_{\ell i} \end{bmatrix} = \begin{bmatrix} \arctan\left[\dfrac{T_{y_i} - S_{y_\ell}}{T_{x_i} - S_{x_\ell}} \right] \\ t_i^{\mathrm{e}} + \dfrac{\sqrt{(T_{x_i} - S_{x_\ell})^2 + (T_{y_i} - S_{y_\ell})^2}}{c} \end{bmatrix} \quad (79)$$

where t_i^{e} is the unknown emission time of the acoustic signal from ith target and c is the known speed of sound.

To incorporate false alarms, we denote a clutter target (with index 0) as ϕ_0. A false measurement detected by the ℓth sensor consists of a bearing θ_0, which is uniformly distributed in the field of view of the ℓth sensor, and its arrival time t_0, which is uniformly distributed in the interval $[0, W]$. The number of false alarms from each sensor is assumed to be a Poisson random variable with mean

$$N_{\mathrm{fa}} = \lambda_{\mathrm{fa}} \Phi W \quad (80)$$

where Φ is the range of the field of view and is assumed to be the same for each sensor and λ_{fa} can be interpreted as the spatial-temporal density.

The likelihood function [2] of the target parameter[4] (location and emission time) based on the measurement $\mathbf{z}_{\ell j}$ is

$$\Lambda(\phi_0; \mathbf{z}_{\ell j}) \triangleq p(\mathbf{z}_{\ell j} \mid \phi_0) = p(\theta_0) p(t_0) = \frac{1}{\Phi W} \quad (81)$$

$$\Lambda(\phi_i; \mathbf{z}_{\ell j}) \triangleq p(\mathbf{z}_{\ell j} \mid \phi_i) = |2\pi R_\ell|^{-1/2} \cdot \exp\{-\tfrac{1}{2}[\mathbf{z}_{\ell j} - \mathbf{h}_\ell(\phi_i)]' R_\ell^{-1} [\mathbf{z}_{\ell j} - \mathbf{h}_\ell(\phi_i)]\} \quad i \neq 0 \quad (82)$$

where (81) is the probability density function (pdf) of a clutter-origin measurement (a false alarm), and (82) is the pdf of a real measurement from a true target with unknown ϕ_i.

The problem is to estimate N and $\mathbf{\Phi} = \{\phi_i, \ i = 1, \ldots, N\}$ (therefore knowing $\mathbf{T} = \{\mathrm{T}_i, \ i = 1, \ldots, N\}$) given the complete set of observations $\mathbf{Z} = \{\mathbf{z}_{\ell j}, \ \ell = 1, \ldots, S; \ j = 1, \ldots, m_\ell\}$ in the presence of missed detections and false alarms and without the knowledge of the true data association.

[4] If the source is clutter, it has no emission time, only an arrival time.

3.2. Poisson Point Process Measurement Modeling

Assume the number of targets, N, is given. The number of measurements m_ℓ and $\{\mathbf{z}_{\ell j}, j = 1,2,\ldots,m_\ell\}$ obtained at the ℓth sensor is jointly modeled as a realization of a Poisson Point Process (PPP) [11]. The measurement set at the ℓth sensor is denoted as

$$\psi_\ell = \{m_\ell, \mathbf{z}_{\ell 1}, \mathbf{z}_{\ell 2}, \ldots, \mathbf{z}_{\ell m_\ell}\} \tag{83}$$

In this case, the points $\mathbf{z}_{\ell j}$ occur in the space $\mathbb{S} = \{(\theta,t) : \theta \in \Phi, t \in [0,W]\}$ and their order is irrelevant. The PPP is fully parameterized by its spatial intensity function

$$\mu_\ell(\mathbf{z}) = \sum_{i=0}^{N} p_i^{\mathrm{d}} g_{\ell i}(\mathbf{z}) \tag{84}$$

where p_i^{d} is the probability of detection for the real target i ($i \neq 0$) and is assumed to be the same at each sensor and with abuse of notation

$$p_0^{\mathrm{d}} = N_{\mathrm{fa}} \tag{85}$$

is the expected number of false alarms at each sensor; the density $g_{\ell i}(\mathbf{z})$ is the conditional[5] pdf of a measurement $\mathbf{z}$ obtained by the ℓth sensor given that it is associated with the ith target and is given by

$$g_{\ell i}(\mathbf{z}) = \frac{1}{\Phi W} \quad i = 0 \tag{86}$$

$$g_{\ell i}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{h}_\ell(\phi_i), R_\ell) \quad i = 1,\ldots,N \tag{87}$$

For notational simplicity, we denote

$$\mathbf{p}^{\mathrm{d}} = [p_0^{\mathrm{d}} \quad p_1^{\mathrm{d}} \ldots p_N^{\mathrm{d}}]^T \tag{88}$$

which is assumed to be unknown and therefore the set of parameters to be estimated is expanded to $\boldsymbol{\theta} = [\boldsymbol{\Phi}^T \ \ \mathbf{p}^{\mathrm{d}T}]^T$ for a given N.

The number of points in the PPP is a Poisson random variable with mean $\int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z}$, that is, the probability mass function (pmf) of m_ℓ is

$$\begin{aligned} p(m_\ell) &= \frac{(\int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z})^{m_\ell}}{m_\ell!} \exp\left\{ - \int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z} \right\} \\ &= \frac{\left(\sum_{i=0}^{N} p_i^{\mathrm{d}}\right)^{m_\ell}}{m_\ell!} \exp\left(- \sum_{i=0}^{N} p_i^{\mathrm{d}} \right) \end{aligned} \tag{89}$$

The m_ℓ points are defined as independent and identically distributed (i.i.d.) samples of a random variable with probability density function

$$p(\mathbf{z}) = \frac{\mu_\ell(\mathbf{z})}{\int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z}} = \frac{\sum_{i=0}^{N} p_i^{\mathrm{d}} g_{\ell i}(\mathbf{z})}{\sum_{i=0}^{N} p_i^{\mathrm{d}}} \tag{90}$$

The joint pmf-pdf of ψ_ℓ from (83) is

$$p(\psi_\ell \mid \boldsymbol{\theta}) = \exp\left(- \sum_{i=0}^{N} p_i^{\mathrm{d}} \right) \prod_{j=1}^{m_\ell} \mu_\ell(\mathbf{z}_{\ell j} \mid \boldsymbol{\theta}) \tag{91}$$

where the conditioning (dependency) on $\boldsymbol{\theta}$ will be explicitly indicated hereafter. The factorial term $m_\ell!$ in (89) is canceled out because there are $m_\ell!$ permutations of an ordered list of measurements. Let $\boldsymbol{\Psi}$ denote the set of all measurement sets (from the S sensors), i.e.,

$$\boldsymbol{\Psi} = \{\psi_1, \psi_2, \ldots, \psi_S\} \tag{92}$$

The conditional independence of the S measurement sets yields

$$p(\boldsymbol{\Psi} \mid \boldsymbol{\theta}) = \prod_{\ell=1}^{S} p(\psi_\ell \mid \boldsymbol{\theta}) \tag{93}$$

Therefore, we can find the maximum likelihood estimate (MLE) of $\boldsymbol{\theta}$ by maximizing (93).

3.3. Data Association Modeling

Since the intensity function (84) is a mixture of uniform or Gaussian pdf and the association is unknown, we model the latent association variables as conditionally independent random variables

$$\kappa_{\ell j} \in \{0,1,2,\ldots,N\} \tag{94}$$

that identify which component generated $\mathbf{z}_{\ell j}$. Here $\kappa_{\ell j} = 0$ indicates that the measurement is generated by the clutter. The set of latent variables for the ℓth sensor is denoted as

$$\kappa_\ell = \{\kappa_{\ell 1}, \ldots, \kappa_{\ell m_\ell}\} \tag{95}$$

such that the complete set of latent variables for all sensors is

$$\boldsymbol{\kappa} = \{\kappa_1, \ldots, \kappa_S\} \tag{96}$$

The latent association variables may be regarded as "marks" associated with each of the points in the PPP. If we define a mark space

$$\mathbb{M} \triangleq \{0,1,2,\ldots,N\} \tag{97}$$

then the marked measurement set at the ℓth sensor denoted by

$$\psi_\ell^{\mathrm{M}} = \{m_\ell, (\mathbf{z}_{\ell 1}, \kappa_{\ell 1}), \ldots, (\mathbf{z}_{\ell m_\ell}, \kappa_{\ell m_\ell})\} \tag{98}$$

represents a realization of the marked[6] PPP for the ℓth sensor on the product space $\mathbb{S} \times \mathbb{M}$. It can be shown that the intensity function of ψ_ℓ^{M} is

$$\mu_\ell^{\mathrm{M}}(\mathbf{z}, \kappa \mid \boldsymbol{\theta}) = p_\kappa^{\mathrm{d}} g_{\ell\kappa}(\mathbf{z}) \tag{99}$$

The joint probability density function of ψ_ℓ^{M} is, similarly to (91), given by

$$\begin{aligned} p(\psi_\ell^{\mathrm{M}} \mid \boldsymbol{\theta}) &= \exp\left(- \sum_{\kappa=0}^{N} \int_{\mathbb{S}} \mu_\ell^{\mathrm{M}}(\mathbf{z}, \kappa \mid \boldsymbol{\theta}) d\mathbf{z} \right) \\ &\quad \cdot \prod_{j=1}^{m_\ell} \mu_\ell^{\mathrm{M}}(\mathbf{z}_{\ell j}, \kappa_{\ell j} \mid \boldsymbol{\theta}) \end{aligned} \tag{100}$$

[5] $g_{\ell i}(\mathbf{z} \mid \phi_i)$ will be used when the conditioning needs to be explicitly indicated.

[6] The superscript of ψ_ℓ^{M} indicates that the associations are known, i.e., "marked."

TABLE III
Centralized EM algorithm.

1: Initializes $\boldsymbol{\theta}^{(0)}$
2: **for** $n = 1,2,\ldots$ **do** until convergence
3: **E step** Evaluate

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(n-1)}) = \sum_{\boldsymbol{\kappa}} p(\boldsymbol{\kappa} \mid \mathbf{Z}, \boldsymbol{\theta}^{(n-1)}) \ln p(\boldsymbol{\Psi}^{\mathrm{M}} \mid \boldsymbol{\theta}) \tag{106}$$

4: **M step** Evaluate $\boldsymbol{\theta}^{(n)}$ as

$$\boldsymbol{\theta}^{(n)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(n-1)}) \tag{107}$$

5: **end for**

Let us denote the marked measurement sets from all sensors as

$$\boldsymbol{\Psi}^{\mathrm{M}} = \{\psi_1^{\mathrm{M}}, \psi_2^{\mathrm{M}}, \ldots, \psi_S^{\mathrm{M}}\} \tag{101}$$

The conditional independence of these S marked measurement sets yields the pmf-pdf for $\boldsymbol{\Psi}^{\mathrm{M}}$ as

$$p(\boldsymbol{\Psi}^{\mathrm{M}} \mid \boldsymbol{\theta}) = \exp\left(-S\sum_{i=0}^{N} p_i^{\mathrm{d}}\right) \prod_{\ell=1}^{S}\prod_{j=1}^{m_\ell} p_{\kappa_{\ell j}}^{\mathrm{d}} g_{\ell\kappa_{\ell j}}(\mathbf{z}_{\ell j} \mid \boldsymbol{\theta}) \tag{102}$$

where we have used the fact

$$\sum_{\kappa=0}^{N}\left(\int_{\mathbb{S}} p_\kappa^{\mathrm{d}} g_{\ell\kappa}(\mathbf{z} \mid \mathbf{T}, \mathbf{t}^{\mathrm{e}}) d\mathbf{z}\right) = \sum_{i=0}^{N} p_i^{\mathrm{d}} \tag{103}$$

Dividing (102) by (93) leads to the density of the marks conditioned on the observed measurements and the unknown parameters

$$p(\boldsymbol{\kappa} \mid \mathbf{Z}, \boldsymbol{\theta}) = \prod_{\ell=1}^{S}\prod_{j=1}^{m_\ell} p_\ell(\kappa_{\ell j} \mid \mathbf{z}_{\ell j}, \boldsymbol{\theta}) \tag{104}$$

where

$$p_\ell(\kappa_{\ell j} \mid \mathbf{z}_{\ell j}, \boldsymbol{\theta}) = \frac{p_{\kappa_{\ell j}}^{\mathrm{d}} g_{\ell\kappa_{\ell j}}(\mathbf{z}_{\ell j} \mid \boldsymbol{\theta})}{\mu_\ell(\mathbf{z}_{\ell j} \mid \boldsymbol{\theta})} \tag{105}$$

4. CENTRALIZED ALGORITHM

4.1. Centralized EM Algorithm

Given the joint distribution $p(\boldsymbol{\Psi}^{\mathrm{M}} \mid \boldsymbol{\theta})$ over observed $\boldsymbol{\Psi}$ and latent variables $\boldsymbol{\kappa}$, governed by the parameter $\boldsymbol{\theta}$, the maximum likelihood estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ from the likelihood function $p(\boldsymbol{\Psi} \mid \boldsymbol{\theta})$ can be found by the standard (named as centralized hereafter) EM algorithm [12] as shown in Table III.

Evaluation of the E step decomposes into two terms

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(n-1)}) = Q_p + Q_\phi \tag{108}$$

where

$$Q_p = Q(\mathbf{p}^{\mathrm{d}} \mid \boldsymbol{\theta}^{(n-1)}) = -S\sum_{i=0}^{N} p_i^{\mathrm{d}} + \sum_{\ell=1}^{S}\sum_{j=1}^{m_\ell}\sum_{i=0}^{N} \ln(p_i^{\mathrm{d}}) \alpha_{\ell ji}^{(n-1)} \tag{109}$$

$$Q_\phi = Q(\boldsymbol{\Phi} \mid \boldsymbol{\theta}^{(n-1)}) = \sum_{\ell=1}^{S}\sum_{j=1}^{m_\ell}\sum_{i=0}^{N} \ln(g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_i)) \alpha_{\ell ji}^{(n-1)} \tag{110}$$

where

$$\alpha_{\ell ji}^{(n-1)} = p_\ell(\kappa_{\ell j} = i \mid \mathbf{z}_{\ell j}, \boldsymbol{\theta}^{(n-1)}) = \frac{p_i^{\mathrm{d}(n-1)} g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_i^{(n-1)})}{\sum_{i=0}^{N} p_i^{\mathrm{d}(n-1)} g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_i^{(n-1)})} \tag{111}$$

The M step involves two separate maximizations with respect to $\mathbf{p}^{\mathrm{d}}$ and $\boldsymbol{\Phi}$. From the Karush-Kuhn-Tucker (KKT) conditions [20], we have

$$p_i^{\mathrm{d}(n)} = \begin{cases} \dfrac{1}{S}\sum_{\ell=1}^{S}\sum_{j=1}^{m_\ell} \alpha_{\ell j0}^{(n-1)} & \text{if } i = 0 \\ \min\left\{1, \dfrac{1}{S}\sum_{\ell=1}^{S}\sum_{j=1}^{m_\ell} \alpha_{\ell ji}^{(n-1)}\right\} & \text{if } i \neq 0 \end{cases} \tag{112}$$

Since Q_ϕ in (110) can be further decomposed into $N+1$ terms, the parameters of each target can be estimated independently as

$$\boldsymbol{\phi}_i^{(n)} = \arg\max_{\boldsymbol{\phi}_i} \sum_{\ell=1}^{S}\sum_{j=1}^{m_\ell} \ln(g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_i)) \alpha_{\ell ji}^{(n-1)} \tag{113}$$

5. DISTRIBUTED ALGORITHM

Note that it is possible to have a distributed implementation of the centralized EM algorithm if (i) every node has the consensus on the initialization and (ii) every node has the consensus on the parameter estimates at the end of each M step. The second condition can readily be satisfied if an averaging consensus based distributed ADMM is applied to solve (113), which is a nonlinear least squares problem, and a distributed averaging consensus algorithm is applied to obtain (112). However, it is not trivial to have the same initialization for $\boldsymbol{\theta} = [\boldsymbol{\Phi}^T \ \mathbf{p}^{\mathrm{d}T}]^T$ among all sensors, especially for the component $\boldsymbol{\Phi}$. Simulation results show that it is good enough to initialize each p_i^{d} to be 1. Whereas, equal initialization at some pre-fixed values (for instance, zero vectors) for $\boldsymbol{\Phi}$ could result in the convergence of the EM algorithm to estimates that are very different from the desired MLE.

There are two possible initialization approaches in a single target localization scenario. Assume that the data association is known and no missed detections or false alarms occur, we want to localize a single target using the algorithm in Table II. The first approach is to initialize the target location at each node using only its bearing measurement. The average distance from the wireless sensor network to the target is assumed to be D. Given a range R (probably unknown in a real scenario), each node initializes the target location along the measured line of sight in the direction towards the target randomly with a distance (between the initialized target location and the node itself) being uniformly distributed

TABLE IV
RMSE using different initializations for distributed localization of a single target.

	Centralized	Intersection	Random $R = 30$	Random $R = 60$
RMSE (m)	1.7036	1.7733	3.2988	5.2915

in $[D\text{-}0.5R, D\text{+}0.5R]$. The second approach is to obtain the LOS information (bearing and sensor location) from one of its neighbors and use the intersection of two lines of sight as the initial target location estimate.

Table IV lists the root mean square error (RMSE) of the target location (averaged over 100 Monte-Carlo runs) using different initialization approaches to localize a single target at the location (8.7 m, 99.6 m) in a scenario given in Section 6. It shows that the performance of the distributed algorithm with LOS intersection initialization is almost as good as the centralized algorithm, which assumes all bearing measurements available at a fusion center and also uses intersection initialization. With a random initialization based on some knowledge, which is likely unavailable, the distributed algorithm converges to local minimum point of (53) with $h(x)$ being an arctan function.

For a multiple target localization scenario with unknown data association, the random initialization approach will be worse. Therefore, in a similar way as what we did at the fusion center in a centralized fusion algorithm, each node obtains an initial position estimate for each target that is very close to the final MLE, by associating its local measurements with those from its neighbors using the sequential m-best 2-D assignment algorithm [1]. Another important reason that we choose the sequential m-best 2-D algorithm over the random initialization approach is that the position estimates obtained are completely observable with corresponding covariance matrices, which allows the use of the association method described in Section 2.5 to reach the consensus on the initialization.

If the probability of detection is low or the false alarm rate is high, then it is possible that the initial estimated numbers of targets at various nodes are different. Some nodes could have estimated more targets than there actually are due to false alarms, whereas other nodes could have estimated less due to missed detections. Section 5.1 discusses how to reach the consensus on the number of candidate targets[7] among all nodes.

Section 5.2 assumes that each node has an initial set consisting of the same number of target estimates which correspond to the same group of candidate targets, and discusses how to reach the consensus on on target-estimate association, that is, for a given ordered set of candidate targets, each node should know the association between its estimates and the targets. Note that the initial estimated value of parameter $\mathbf{\Phi}$ could still be different from node to node. However, the consensus on target-estimate association requires all the nodes have exactly the same set of target estimates.

In Section 5.3, we develop a distributed algorithm that assumes all the nodes have the same set of target estimates. The consensus on target-estimate association is required for convergence of the algorithm.

5.1. Consensus on the Number of Targets

If the probability of detection is low or the false alarm rate is high, then it is possible that the initial estimated numbers of targets at various nodes are different. Some nodes could have estimated more targets than there should be, whereas other nodes could have estimated less. In this subsection, we extend the problem solved in Section 2.3 to the case when missed detections and/or false alarms exist and develop a distributed set averaging consensus algorithm to expand some or all sets so that we end up with sets of estimates for the same number of candidate (real or false) targets. Each sensor gradually modifies its own set by performing the association test presented in Section 2.5 with the sets of its neighbors.

Let us denote the initial set of estimates with corresponding covariance matrices at node υ_i as

$$\hat{\mathbf{\Phi}}_i = \{\hat{\phi}_{i1}, \hat{\phi}_{i2}, \ldots, \hat{\phi}_{iN_i}\} \tag{114}$$

$$\mathcal{Q}_i = \{Q_{i1}, Q_{i2}, \ldots, Q_{iN_i}\} \tag{115}$$

where each $\hat{\phi}_{ik}$ corresponds to one candidate target and the number of candidate targets N_i is probably distinct for different nodes υ_i. Assume that there are N_c candidate targets with parameters

$$\mathbf{\Phi}_c = \{\phi_1, \phi_2, \ldots, \phi_{N_c}\} \tag{116}$$

of which only N parameters correspond to real targets and the remaining $N_c - N$ parameters correspond to false targets. The number N_c will only be known at the end of the algorithm. For any target parameter in $\mathbf{\Phi}_c$, there is at most one estimate $\hat{\phi}_{ik}$ at node υ_i.

Figure 1 illustrates the concept of candidate target. In this example, both sensors have detected true targets at coordinates (10 m, 10 m) and (20 m, 20 m), therefore each of these is a candidate target. Sensors 1 and 2 each also have an additional estimated target around (31 m, 31 m) and (29 m, 11 m), respectively. In this case, we assume that these two estimates fail to be associated. Therefore, two additional targets, which are assumed to be at coordinates (30 m, 30 m) and (30 m, 10 m), are also candidate targets.

Referring back to the same context in Section 2.5, the sets $\mathbf{\Phi}_c$ and $\hat{\mathbf{\Phi}}_i$ defined in (116) and (114) play the same roles as X and $\hat{x}_i$ in (33) and (34), respectively. The independent estimation error assumption is valid only when two estimates have no common source of error [3]. In the case that two neighboring nodes each have

[7]The concept of candidate target is discussed in detail in Section 5.1.

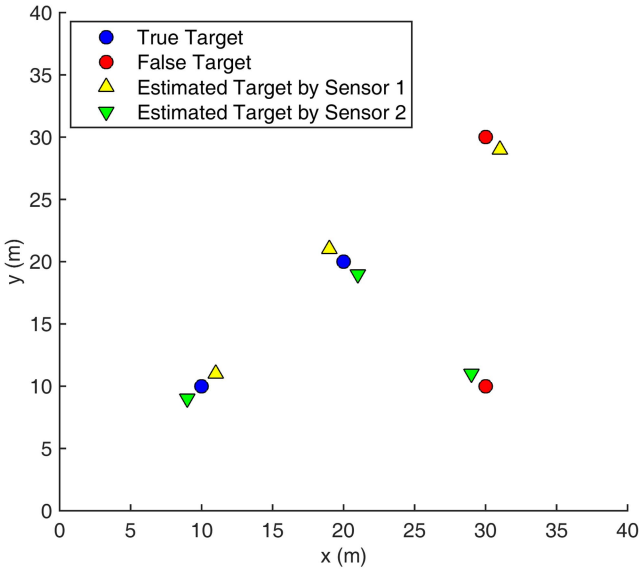


Fig. 1. An illustrative example: each sensor has three estimates, there are four candidate targets.

one estimate for the same target, it is quite likely that these two estimates are obtained using some common measurements and therefore they have correlated errors. Since it is difficult to calculate the cross covariance, we will use (26) as an approximation. This approximation only applies to true target location estimates that are supposed to be associated, and will not affect the decisions involving estimates that belong to false targets. In our approach, the independence assumption (of the errors in the estimated target locations at different sensors) is used only to build the consensus. We do not "fuse" the corresponding covariances, which pertain to errors that are dependent; fusing them under independence assumption would indeed be optimistic and unreasonable.

One iteration of the distributed set averaging consensus algorithm is described next. At iteration t, node v_i expands $\hat{\mathbf{\Phi}}_i(t)$ sequentially with each neighboring node $v_j \in \mathcal{N}_i$.

Firstly, for a given significance level α, the following generalized 2-D assignment problem

$$\min_{\rho_{k\ell}} \sum_{k=0}^{N_i} \sum_{\ell=0}^{N_j} \rho_{k\ell} d_{k\ell}(t) \tag{117}$$

subject to

$$\sum_{\ell=0}^{N_j} \rho_{k\ell} = 1 \text{ for all } k = 1,2,\ldots,N_i \tag{118}$$

$$\sum_{k=0}^{N_i} \rho_{k\ell} = 1 \text{ for all } \ell = 1,2,\ldots,N_j \tag{119}$$

$$\rho_{k\ell} \in \{0,1\} \text{ for all } k = 0,1,\ldots,N_i \text{ and } \ell = 0,1,\ldots,N_j \tag{120}$$

where, similarly as in (38) and (39), with the addition of a dummy estimate $\hat{\phi}_{i0}$ at each node v_i, the distance between two estimates are defined as

$$d_{k\ell}(t) = \begin{cases} (\hat{\phi}_{ik}(t) - \hat{\phi}_{j\ell}(t))^T (P_{ik}(t) + P_{j\ell}(t))^{-1} \\ \cdot (\hat{\phi}_{ik}(t) - \hat{\phi}_{j\ell}(t)) \quad \text{if } k > 0 \text{ and } \ell > 0 \\ F^{-1}_{\chi^2_n}(1-\alpha) \quad \text{if } k = 0 \text{ or } \ell = 0 \end{cases} \tag{121}$$

Next, $\hat{\mathbf{\Phi}}_i(t)$ could be expanded based on the solution $\rho_{k\ell}$ to the assignment problem. If

$$\rho_{0\ell} = 1 \tag{122}$$

which means that the estimate $\hat{\phi}_{j\ell}(t)$ is not associated with any estimate at node v_i, then $\hat{\mathbf{\Phi}}_i(t)$ is expanded to $\hat{\mathbf{\Phi}}_i(t) \cup \{\hat{\phi}_{j\ell}(t)\}$. If there are multiple estimates that are not associated, then they are all used to expand $\hat{\mathbf{\Phi}}_i(t)$.

The algorithm terminates when every node set has the same number of estimates and no set can be expanded further.

Note that if a sensor does not have a position estimate for target i, it will "copy" a position estimate for target i from one of its neighbors. If a sensor has a position estimate for a false target, then all its neighbors need to "copy" this estimate so that every sensor has a position estimate for the same false target. Since the total number of target estimates across all the sensors is a finite number $\sum_{i=1}^{S} N_i$, where S is the number of sensors, and each iteration expands at least one set, the algorithm will be terminated in a finite number of iterations.

5.2. Consensus on the Target-Estimate Association

Suppose that the initial sets of target estimates, either obtained directly via the assignment algorithm across all nodes or by means of the method described in Section 5.1, have the same number of target estimates.

The local variable of $\boldsymbol{\theta}_\ell$ has components $\mathbf{\Phi}_\ell$ and $\mathbf{p}^{\mathrm{d}}_\ell$. We initialize $\mathbf{p}^{\mathrm{d}}_\ell$ as a vector of ones. The component $\mathbf{\Phi}_\ell = [\phi^T_{\ell 1} \;\; \phi^T_{\ell 2} \ldots \phi^T_{\ell N}]^T$ will be initialized using the set obtained via the sequential assignment algorithm denoted by

$$\Phi_\ell = \{\varphi_{\ell 1}, \varphi_{\ell 2}, \ldots, \varphi_{\ell N}\} \tag{123}$$

There are $N!$ ways of initialization for node ℓ. We want to find a permutation for each set Φ_ℓ

$$\pi_\ell(\Phi_\ell) = \{\varphi_{\ell\pi_\ell(1)}, \varphi_{\ell\pi_\ell(2)}, \ldots, \varphi_{\ell\pi_\ell(N)}\} \tag{124}$$

such that for any $k = 1,2\ldots,N$, the set of estimates $\{\varphi_{\ell\pi_\ell(k)}, \; \ell = 1,2,\ldots,S\}$, one from each sensor, corresponds to the same target. For this purpose, we can apply the algorithm in Section 2.3 on the sets Φ_ℓ, $\ell = 1,\ldots,S$, and when the algorithm terminates, we have all the sets equal.

We initialize $\mathbf{\Phi}_\ell$ as

$$\mathbf{\Phi}_\ell = [\varphi^T_{\ell\pi_\ell(1)} \quad \varphi^T_{\ell\pi_\ell(2)} \ldots \varphi^T_{\ell\pi_\ell(N)}]^T \tag{125}$$

where, letting $i = \pi_\ell(k)$

$$\varphi_{\ell i} = [T_{x_i} \quad T_{y_i} \quad t_i^{\mathrm{e}}]^T \tag{126}$$

is ordered such that

$$T_{x_i} \le T_{x_j},\ \forall\ i \le j \tag{127}$$

$$T_{y_i} \le T_{y_j},\ \forall\ i \le j \text{ and } T_{x_i} = T_{x_j} \tag{128}$$

The use of the ordering rules (127) and (128) to label targets makes sense only when the sets of the estimates from all nodes are the same.

5.3. The EM and AC Based Distributed ADMM Algorithm

The centralized EM algorithm provides a method to solve the following optimization problem

$$\min_{\boldsymbol{\theta}} \{-\ln p(\boldsymbol{\Psi} \mid \boldsymbol{\theta})\} \tag{129}$$

where

$$\ln p(\boldsymbol{\Psi} \mid \boldsymbol{\theta}) = \sum_{\ell=1}^{S} \ln p(\boldsymbol{\psi}_\ell \mid \boldsymbol{\theta}) \tag{130}$$

To develop a distributed algorithm to solve the above problem, we consider an equivalent formulation with equality constraints between local variables $\boldsymbol{\theta}_\ell$ and a global variable $\boldsymbol{\theta}$

$$\min_{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_S} \sum_{\ell=1}^{S} -\ln p(\boldsymbol{\psi}_\ell \mid \boldsymbol{\theta}_\ell) \tag{131}$$

subject to

$$\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 = \ldots = \boldsymbol{\theta}_S = \boldsymbol{\theta} \tag{132}$$

The augmented Lagrangian is

$$L_\rho(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_S, \boldsymbol{\theta}, \lambda) = \sum_{\ell=1}^{S} \Big[-\ln p(\boldsymbol{\psi}_\ell \mid \boldsymbol{\theta}_\ell) + \lambda_\ell^T (\boldsymbol{\theta}_\ell - \boldsymbol{\theta}) + \frac{\rho}{2} \|\boldsymbol{\theta}_\ell - \boldsymbol{\theta}\|_2^2 \Big] \tag{133}$$

Following the similar derivations as presented in Section 2.7, we can obtain the ADMM steps, which are in a distributed form, as

$$\boldsymbol{\theta}_\ell^{k+1} = \arg\min_{z} \sum_{\ell=1}^{S} \Big[-\ln p(\boldsymbol{\psi}_\ell \mid \boldsymbol{\theta}_\ell) + \lambda_\ell^{kT} (\boldsymbol{\theta}_\ell - \boldsymbol{\theta}^k) + \frac{\rho}{2} \|\boldsymbol{\theta}_\ell - \boldsymbol{\theta}^k\|_2^2 \Big] \tag{134}$$

$$\boldsymbol{\theta}^{k+1} = \frac{1}{S} \sum_{\ell=1}^{S} \boldsymbol{\theta}_\ell^{k+1} \tag{135}$$

$$\lambda_\ell^{k+1} = \lambda_\ell^k + \rho(\boldsymbol{\theta}_\ell^{k+1} - \boldsymbol{\theta}^{k+1}) \tag{136}$$

TABLE V
EM and averaging consensus based distributed ADMM algorithm.

1: Node v_ℓ initializes $\boldsymbol{\theta}_\ell^1$ by a sequential m-best 2-D assignment algorithm and $\lambda_\ell^1 = 0$
2: Compute $\boldsymbol{\theta}^1 = \frac{1}{S} \sum_{\ell=1}^{S} \boldsymbol{\theta}_\ell^1$ by a distributed averaging consensus algorithm
3: **for** $k = 1, 2, \ldots$ **do** until convergence
4: **for all** v_ℓ **do**
5: Compute $\boldsymbol{\theta}_\ell^{k+1}$ via (134) by a local EM algorithm
6: Compute $\boldsymbol{\theta}^{k+1} = \frac{1}{S} \sum_{\ell=1}^{S} \boldsymbol{\theta}_\ell^{k+1}$ by a distributed averaging consensus algorithm
7: Compute λ_ℓ^{k+1} via (136)
8: **end for**
9: **end for**

Based on the above ADMM steps, we obtain an EM and averaging consensus based distributed algorithm as summarized in Table V. Each node v_ℓ stores and updates two vectors $\boldsymbol{\theta}_\ell$ and λ_ℓ. At iteration k=1, each node initializes a local parameter estimate $\boldsymbol{\theta}_\ell^1$ and reaches the consensus on the global variable $\boldsymbol{\theta}^1$ via a distributed averaging consensus algorithm. The local dual variable is initialized as $\lambda_\ell^1 = 0$. During the kth iteration, each node updates the local variable $\boldsymbol{\theta}_\ell^{k+1}$ via (134), which is solved by the local EM algorithm as in Table VI because of the term $\ln p(\boldsymbol{\psi}_\ell \mid \boldsymbol{\theta}_\ell)$. Next, each node obtains $\boldsymbol{\theta}^k$ via a distributed averaging consensus algorithm, and subsequently, updates its local dual variable λ_i^{k+1} using (136), which concludes the kth iteration.

In the local EM algorithm, the dual variable λ_ℓ is partitioned as

$$\lambda_\ell = \begin{bmatrix} \lambda_{\phi\ell} \\ \lambda_{p\ell} \end{bmatrix} \tag{143}$$

with respect to the components $\boldsymbol{\Phi}_\ell$ and $\mathbf{p}_\ell^{\mathrm{d}}$ of $\boldsymbol{\theta}_\ell$.

5.4. Determination of the Number of Real Targets

The Bayesian information criterion (BIC) was used in our previous paper [13] for the Poisson measurement generation model assumption because this assumption leads to a cardinality selection problem formulation, which is similar to the K-means clustering problem and BIC is one of the widely used and trusted approaches [25] to determine the number of clusters (the number of targets in the present paper).

In the present paper, we assume a Bernoulli measurement generation model, which is more realistic than the Poisson model in the multiple transient emitter localization problem. Therefore, we used the likelihood function (binomial, in view of the Bernoulli model) based thresholding approach to determine the number of targets.

In the distributed algorithm, the estimated probability of detection $\hat{p}_i^{\mathrm{d}}$ will converge to the value in (112) for $n = 10$. Assume that the true probability of detection is high (say, above 0.9) and the number of nodes is large, we expect that most of the nodes have a measurement associated with a particular target. Therefore, for a real

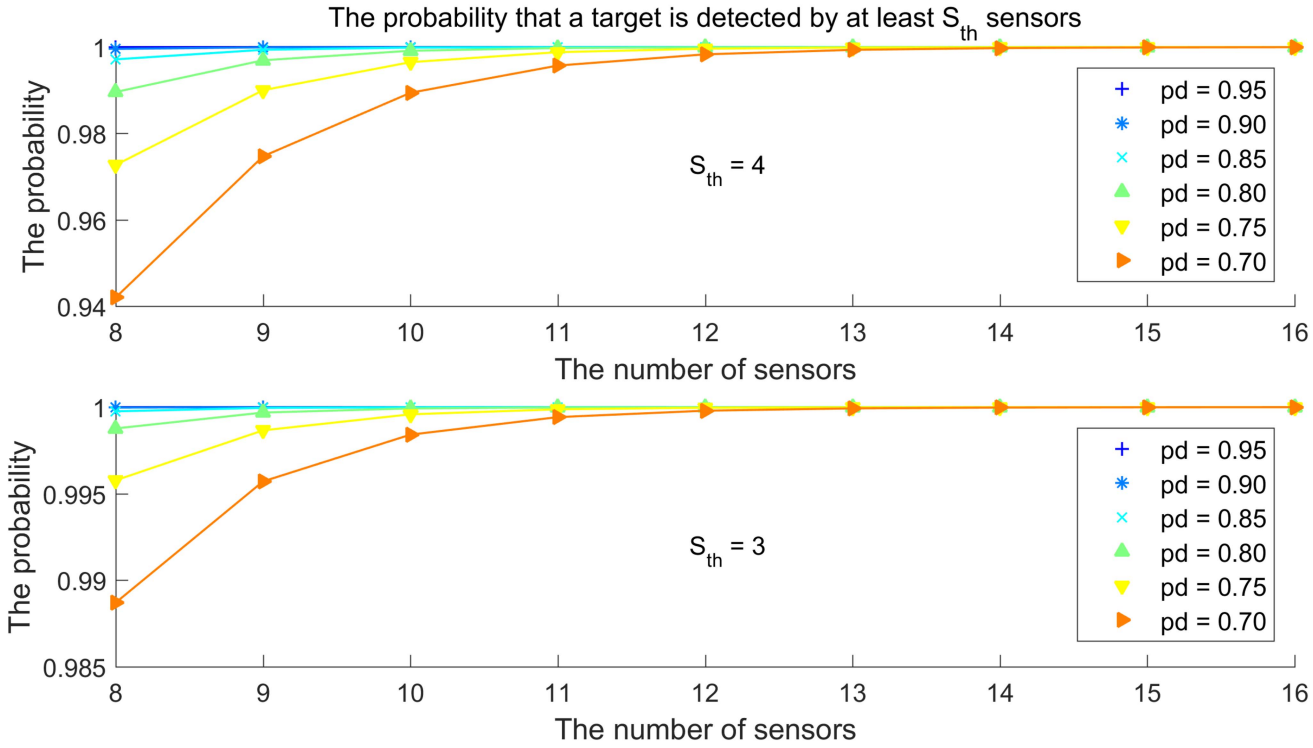


Fig. 2. The probability that a target is detected by at least S_{th} (3 or 4) sensors for varied values for the number of sensors and the probability of detection (pd).

TABLE VI
Local EM algorithm at node v_ℓ to find $\boldsymbol{\theta}_\ell^{k+1}$.

1: Initialization

$$\boldsymbol{\theta}_\ell^{(0)} = \boldsymbol{\theta}_\ell^{k} \tag{137}$$

2: **for** $n = 1,2,\ldots$ **do** until convergence
3: **E step**

$$\alpha_{\ell ji}^{(n-1)} = p_\ell(\kappa_{\ell j} = i \mid \mathbf{z}_{\ell j}, \boldsymbol{\theta}_\ell^{(n-1)}) = \frac{p_{i\ell}^{\mathrm{d}(n-1)} g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_{i\ell}^{(n-1)})}{\sum_{i=0}^{N} p_{i\ell}^{\mathrm{d}(n-1)} g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_{i\ell}^{(n-1)})} \tag{138}$$

$$Q(\boldsymbol{\theta}_\ell \mid \boldsymbol{\theta}_\ell^{(n-1)}) = \sum_{\kappa_\ell} p(\kappa_\ell \mid \psi_\ell, \boldsymbol{\theta}_\ell^{(n-1)}) \ln p(\psi_\ell^{\mathrm{M}} \mid \boldsymbol{\theta}_\ell) = Q(\mathbf{p}_\ell^{\mathrm{d}}) + Q(\boldsymbol{\Phi}_\ell)$$

$$Q(\mathbf{p}_\ell^{\mathrm{d}}) = -\sum_{i=0}^{N} p_{i\ell}^{\mathrm{d}} + \sum_{j=1}^{m_\ell} \sum_{i=0}^{N} \ln(p_{i\ell}^{\mathrm{d}}) \alpha_{\ell ji}^{(n-1)} \tag{139}$$

$$Q(\boldsymbol{\Phi}_\ell) = \sum_{j=1}^{m_\ell} \sum_{i=0}^{N} \ln(g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_{i\ell})) \alpha_{\ell ji}^{(n-1)} \tag{140}$$

4: **M step**

$$\mathbf{p}_\ell^{\mathrm{d}(n)} = \arg\min_{\mathbf{p}_\ell^{\mathrm{d}}} -Q(\mathbf{p}_\ell^{\mathrm{d}}) + \lambda_{p\ell}^{kT}(\mathbf{p}_\ell^{\mathrm{d}} - \mathbf{p}^{\mathrm{d}k}) + \frac{\rho}{2}\|\mathbf{p}_\ell^{\mathrm{d}} - \mathbf{p}^{\mathrm{d}k}\|_2^2 \tag{141}$$

$$\boldsymbol{\Phi}_\ell^{(n)} = \arg\min_{\boldsymbol{\Phi}_\ell} -Q(\boldsymbol{\Phi}_\ell) + \lambda_{\phi\ell}^{kT}(\boldsymbol{\Phi}_\ell - \boldsymbol{\Phi}^k) + \frac{\rho}{2}\|\boldsymbol{\Phi}_\ell - \boldsymbol{\Phi}^k\|_2^2 \tag{142}$$

5: **end for**

target estimate, $\hat{p}_i^{\mathrm{d}}$ is likely to end up with a value close to 1. For a false target estimate, $\hat{p}_i^{\mathrm{d}}$ is likely to end up with a value close to 0, since only a few nodes have a measurement associated with a false target (which is the "same" across sensors, i.e., approximately at the same location). Based on this difference between real targets

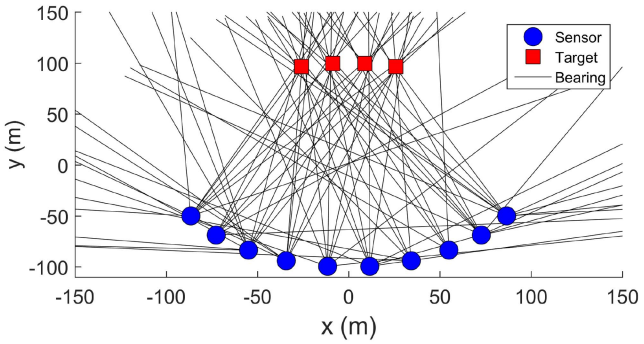


Fig. 3. A scenario with 10 targets and 4 sensors.

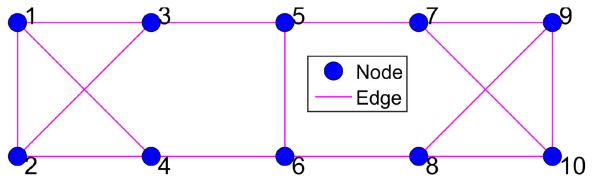


Fig. 4. The graph model of the wireless sensor network in Figure 3.

and false targets, it is reasonable to assume that there is a threshold value of $\hat{p}_i^{\mathrm{d}}$ that can be used to classify targets into either real or false.

If the number of sensors is known and the probability of detection is also known, then one can calculate the probability that a target is detected by at least S_{th} sensors. Figure 2 plots this probability for a range of values for the number of sensors and the probability of detection. Since even at p^{d}=0.7, the probability that a target is detected by at least 3 sensors is greater than 0.995 in most cases. We use the threshold value S_{th}=3. The corresponding threshold value of $\hat{p}_i^{\mathrm{d}}$ is

$$p_{\mathrm{th}}^{\mathrm{d}} = 0.3 \tag{144}$$

when $S = 10$ as in the simulation study. Therefore, we classify the targets with $\hat{p}_i^{\mathrm{d}}$ greater than 0.3 as real targets and otherwise the targets are deemed as false.

6. SIMULATION RESULTS

6.1. Scenario

Assume there are four targets ($N = 4$). The emission times of the acoustic events at the target locations are 0.4 s, 0.3 s, 0.1 s and 0.2 s, respectively. The speed of the acoustic signal is assumed to be 342 m/s. The measurement noise covariance matrix is

$$R_\ell = \begin{bmatrix} 7.6 \times 10^{-5} & 0 \\ 0 & 1 \times 10^{-4} \end{bmatrix} \tag{145}$$

i.e., the bearing standard deviation is $\sqrt{76}$ mrad = 0.5° and the TOA measurement standard deviation amounts to 10 ms, assumed to be the same for all sensors. The probability of detection for the targets is assumed to be 0.9 at all sensors. The time window W is chosen to be 1 s and the field of view of each sensor is from 0 to π. The density of the false alarms is set to be 1.27 $\mathrm{s}^{-1}\mathrm{radian}^{-1}$ such that the expected number of false alarms (N_{fa}) at each sensor is 4, which is equal to the number of real targets. Figure 3 shows one example using a wireless sensor network with 10 sensors numbered from left to right in an ascending order, which is represented by the graph model shown in Figure 4, to localize these 4 targets. Each node has three neighbors.

TABLE VII
CRLB and MSE with and without TOA measurements.

	Bearing	Bearing and TOA
CRLB (m^2)	2.6655	2.6464
MSE (m^2)	2.6396	2.6290

In the simulation, the targets and the sensors are located such that the angle between two LOS from two neighboring targets to any sensor is 5°, which is 10 times the standard deviation of LOS measurement noise, i.e. there are no unresolved measurements.

6.2. The significance of TOA measurements

The TOA measurements play an important role in the data association. The ghosting effect using bearing-only measurements is no longer present due to the additional estimation of a common signal emission time for the measurements associated with a single target. Here, we look at the improved estimation accuracy provided by the TOA measurements on top of the bearing-only measurements.

Assume that the data association is known and no missed detection or false alarms occurs, we want to localize a single target at the location (8.7 m, 99.6 m) with all measurements available at a fusion center. Table VII shows the Cramér-Rao lower bound (CRLB) and MSE of the target location using bearing-only measurements and bearing with TOA measurements. It shows that the improvement of the location estimation due to the additional TOA information is insignificant.

This implies that the TOA information should be only used in the sequential m-best assignment algorithm to obtain initial target estimates. Within the local EM algorithm, we can use only bearing measurements to reduce computational workload without significantly degrading the estimation accuracy.

6.3. Performance Metrics

In the following sections, we evaluate our distributed algorithm by two real-valued metrics for each Monte-Carlo run instead of averaging over all Monte-Carlo runs. These two metrics are the cardinality error for the number of targets and the root mean square (RMS) position error averaged over all targets. The latter is obtained by globally associating each location estimate to the nearest targets.

1) The cardinality error for the number of targets:

Given the true number of targets N_{t} and the estimated number of targets $\hat{N}$, the cardinality error is defined as

$$\tilde{N} = N_{\mathrm{t}} - \hat{N} \tag{146}$$

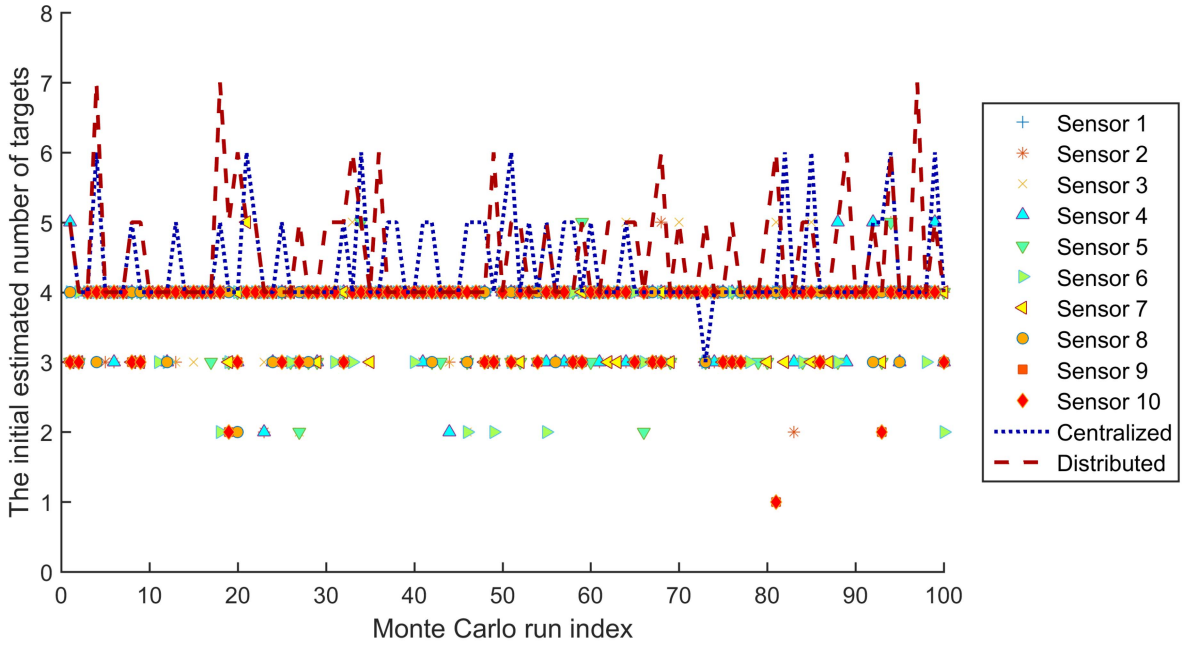


Fig. 5. The initially estimated number (the truth is 4) of targets by individual sensors, the centralized EM algorithm and the EM and AC based distributed ADMM algorithm.

2) The RMS position error:

Given the set of true positions of N_t targets

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_{N_t}, y_{N_t})\} \tag{147}$$

and the set of estimated positions of $\hat{N}$ targets

$$\{(\hat{x}_1, \hat{y}_1), (\hat{x}_2, \hat{y}_2), \ldots, (\hat{x}_{\hat{N}}, \hat{y}_{\hat{N}})\} \tag{148}$$

there are three cases. Let Π_N denote all permutations of the set $\{1, 2, \ldots, N\}$.

Case 1: $N_t = \hat{N}$. The RMS position error is defined as

$$\mathrm{RMS}_p = \min_{\pi \in \Pi_{N_t}} \sqrt{\frac{1}{N_t} \sum_{i=1}^{N_t} [(x_i - \hat{x}_{\pi(i)})^2 + (y_i - \hat{y}_{\pi(i)})^2]} \tag{149}$$

Case 2: $N_t < \hat{N}$. The RMS position error is defined as

$$\mathrm{RMS}_p = \min_{\pi \in \Pi_{\hat{N}}} \sqrt{\frac{1}{N_t} \sum_{i=1}^{N_t} [(x_i - \hat{x}_{\pi(i)})^2 + (y_i - \hat{y}_{\pi(i)})^2]} \tag{150}$$

Case 3: $N_t > \hat{N}$. The RMS position error is defined as

$$\mathrm{RMS}_p = \min_{\pi \in \Pi_{N_t}} \sqrt{\frac{1}{\hat{N}} \sum_{i=1}^{\hat{N}} [(\hat{x}_i - x_{\pi(i)})^2 + (\hat{y}_i - y_{\pi(i)})^2]} \tag{151}$$

Note that we need to combine these two real-valued metrics (146 and one of 149–151) in order to have a complete evaluation of the algorithm performance.

6.4. Performance of the EM and AC based distributed ADMM algorithm

For the algorithm evaluation, the target measurements are generated according to a Bernoulli measurement model, specifically, one measurement from each target is generated for each sensor with a probability p_d or nothing with a probability $1 - p_d$. The false alarms are generated for each sensor according to the Poisson model (80) and (81).

Note that the values of the probability of detection, p_d, and the expected number of false alarms, N_{fa}, are required to generate the target measurements. However, the EM and AC based distributed ADMM algorithm do not need to know the values of N_{fa} and p_d. They adapt to these values by "learning them."

We used 100 Monte-Carlo runs to evaluate the performance of our distributed algorithm and make comparisons with a modified version of the centralized algorithm in [13]. Both used the same threshold (144) to determine the number of targets.

Figure 5 shows the number of targets initially estimated by each sensor using the sequential m-best 2-D assignment algorithm on the measurements of its own and its one-hop neighbors. It can be observed that this number is different from sensor to sensor because of the missed detections and false alarms, which is the motivation for the development of the distributed set consensus algorithm described in the Sections V-A and V-B. In the same plot, the centralized algorithm (denoted by "Centralized") obtained the initial estimated number of targets by using the sequential m-best 2-D assignment algorithm on the measurements from all sensors. In contrast, the distributed algorithm obtained the initial estimate (the same for all sensors) of the number of targets via the distributed set consensus algorithm and this estimate is also the estimated number of candidate targets. Since the centralized and distributed algorithms

TABLE VIII
Evolution of each sensor's target location estimates (each row represents a target location estimate) at key stages of the initialization consensus process.

sensor index	initial estimates by SEQ[m(2-D)]	consensus on the number of targets: after 1 iteration	consensus on the number of targets: after 3 iterations	consensus on the estimates: after 1 iteration	consensus on the estimates: after 25 iterations
1	$\begin{bmatrix} -5.45 & 106.09 \\ 7.16 & 94.15 \\ 26.82 & 101.02 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -5.45 & 106.09 \\ 7.16 & 94.15 \\ 26.82 & 101.02 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -5.45 & 106.09 \\ 7.16 & 94.15 \\ 26.82 & 101.02 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -5.89 & 104.59 \\ 7.19 & 94.37 \\ 26.03 & 99.83 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.86 & 95.19 \\ -7.83 & 103.36 \\ 8.28 & 96.08 \\ 25.65 & 98.66 \\ -104.90 & -21.35 \end{bmatrix}$
2	$\begin{bmatrix} -5.45 & 106.09 \\ 7.16 & 94.15 \\ 26.82 & 101.02 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -5.45 & 106.09 \\ 7.16 & 94.15 \\ 26.82 & 101.02 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -5.45 & 106.09 \\ 7.16 & 94.15 \\ 26.82 & 101.02 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -5.89 & 104.59 \\ 7.19 & 94.37 \\ 26.03 & 99.83 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.86 & 95.19 \\ -7.83 & 103.36 \\ 8.28 & 96.08 \\ 25.65 & 98.66 \\ -104.90 & -21.35 \end{bmatrix}$
3	$\begin{bmatrix} -6.05 & 105.04 \\ 6.21 & 92.11 \\ 23.71 & 96.02 \end{bmatrix}$	$\begin{bmatrix} -6.05 & 105.04 \\ 6.21 & 92.11 \\ 23.71 & 96.02 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -6.05 & 105.04 \\ 6.21 & 92.11 \\ 23.71 & 96.02 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -26.72 & 94.12 \\ -6.29 & 104.05 \\ 7.04 & 93.26 \\ 25.84 & 99.59 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.85 & 95.22 \\ -7.86 & 103.37 \\ 8.29 & 96.08 \\ 25.64 & 98.64 \\ -104.90 & -21.35 \end{bmatrix}$
4	$\begin{bmatrix} -27.05 & 91.42 \\ -6.63 & 101.14 \\ 8.24 & 97.05 \\ 26.75 & 101.26 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -6.63 & 101.14 \\ 8.24 & 97.05 \\ 26.75 & 101.26 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -6.63 & 101.14 \\ 8.24 & 97.05 \\ 26.75 & 101.26 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -26.72 & 94.12 \\ -5.89 & 101.15 \\ 7.78 & 99.08 \\ 26.21 & 99.98 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.85 & 95.22 \\ -7.86 & 103.37 \\ 8.29 & 96.08 \\ 25.64 & 98.64 \\ -104.90 & -21.35 \end{bmatrix}$
5	$\begin{bmatrix} -8.22 & 98.99 \\ 7.64 & 92.64 \\ 26.01 & 100.30 \end{bmatrix}$	$\begin{bmatrix} -25.73 & 102.20 \\ -8.22 & 98.99 \\ 7.64 & 92.64 \\ 26.01 & 100.30 \end{bmatrix}$	$\begin{bmatrix} -25.73 & 102.20 \\ -8.22 & 98.99 \\ 7.64 & 92.64 \\ 26.01 & 100.30 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.83 & 97.86 \\ -6.92 & 98.83 \\ 7.51 & 97.09 \\ 24.78 & 98.37 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.79 & 95.33 \\ -7.98 & 103.42 \\ 8.35 & 96.10 \\ 25.63 & 98.58 \\ -104.90 & -21.35 \end{bmatrix}$
6	$\begin{bmatrix} -25.73 & 102.20 \\ -6.02 & 91.28 \\ 8.56 & 110.96 \\ 24.45 & 96.61 \end{bmatrix}$	$\begin{bmatrix} -25.73 & 102.20 \\ -6.02 & 91.28 \\ 8.56 & 110.96 \\ 24.45 & 96.61 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.73 & 102.20 \\ -6.02 & 91.28 \\ 8.56 & 110.96 \\ 24.45 & 96.61 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.11 & 96.95 \\ -7.50 & 98.81 \\ 8.69 & 99.10 \\ 25.70 & 98.53 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.79 & 95.33 \\ -7.98 & 103.42 \\ 8.35 & 96.10 \\ 25.63 & 98.58 \\ -104.90 & -21.35 \end{bmatrix}$
7	$\begin{bmatrix} -24.83 & 95.60 \\ -7.40 & 100.00 \\ 24.94 & 100.57 \end{bmatrix}$	$\begin{bmatrix} -24.83 & 95.60 \\ -7.40 & 100.00 \\ 7.64 & 92.64 \\ 24.94 & 100.57 \end{bmatrix}$	$\begin{bmatrix} -24.83 & 95.60 \\ -7.40 & 100.00 \\ 7.64 & 92.64 \\ 24.94 & 100.57 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.51 & 98.36 \\ -10.26 & 105.19 \\ 8.97 & 94.20 \\ 25.53 & 98.47 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.73 & 95.44 \\ -8.10 & 103.47 \\ 8.41 & 96.12 \\ 25.61 & 98.52 \\ -104.90 & -21.35 \end{bmatrix}$
8	$\begin{bmatrix} -21.92 & 91.96 \\ -9.15 & 103.81 \\ 10.31 & 95.76 \\ 25.59 & 95.97 \end{bmatrix}$	$\begin{bmatrix} -21.92 & 91.96 \\ -9.15 & 103.81 \\ 10.31 & 95.76 \\ 25.59 & 95.97 \end{bmatrix}$	$\begin{bmatrix} -21.92 & 91.96 \\ -9.15 & 103.81 \\ 10.31 & 95.76 \\ 25.59 & 95.97 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -24.78 & 97.45 \\ -10.15 & 104.22 \\ 9.87 & 99.56 \\ 25.30 & 96.40 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.73 & 95.44 \\ -8.10 & 103.47 \\ 8.41 & 96.12 \\ 25.61 & 98.52 \\ -104.90 & -21.35 \end{bmatrix}$
9	$\begin{bmatrix} -25.74 & 97.82 \\ -12.72 & 110.89 \\ 25.58 & 96.51 \end{bmatrix}$	$\begin{bmatrix} -25.74 & 97.82 \\ -12.72 & 110.89 \\ 10.31 & 95.76 \\ 25.58 & 96.51 \end{bmatrix}$	$\begin{bmatrix} -25.74 & 97.82 \\ -12.72 & 110.89 \\ 10.31 & 95.76 \\ 25.58 & 96.51 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -24.56 & 95.80 \\ -10.50 & 106.40 \\ 9.64 & 94.98 \\ 25.43 & 97.39 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.72 & 95.47 \\ -8.13 & 103.49 \\ 8.43 & 96.12 \\ 25.61 & 98.50 \\ -104.90 & -21.35 \end{bmatrix}$
10	$\begin{bmatrix} -25.74 & 97.82 \\ -12.72 & 110.89 \\ 25.58 & 96.51 \end{bmatrix}$	$\begin{bmatrix} -25.74 & 97.82 \\ -12.72 & 110.89 \\ 10.31 & 95.76 \\ 25.58 & 96.51 \end{bmatrix}$	$\begin{bmatrix} -25.74 & 97.82 \\ -12.72 & 110.89 \\ 10.31 & 95.76 \\ 25.58 & 96.51 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -24.56 & 95.80 \\ -10.50 & 106.40 \\ 9.64 & 94.98 \\ 25.43 & 97.39 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.72 & 95.47 \\ -8.13 & 103.49 \\ 8.43 & 96.12 \\ 25.61 & 98.50 \\ -104.90 & -21.35 \end{bmatrix}$

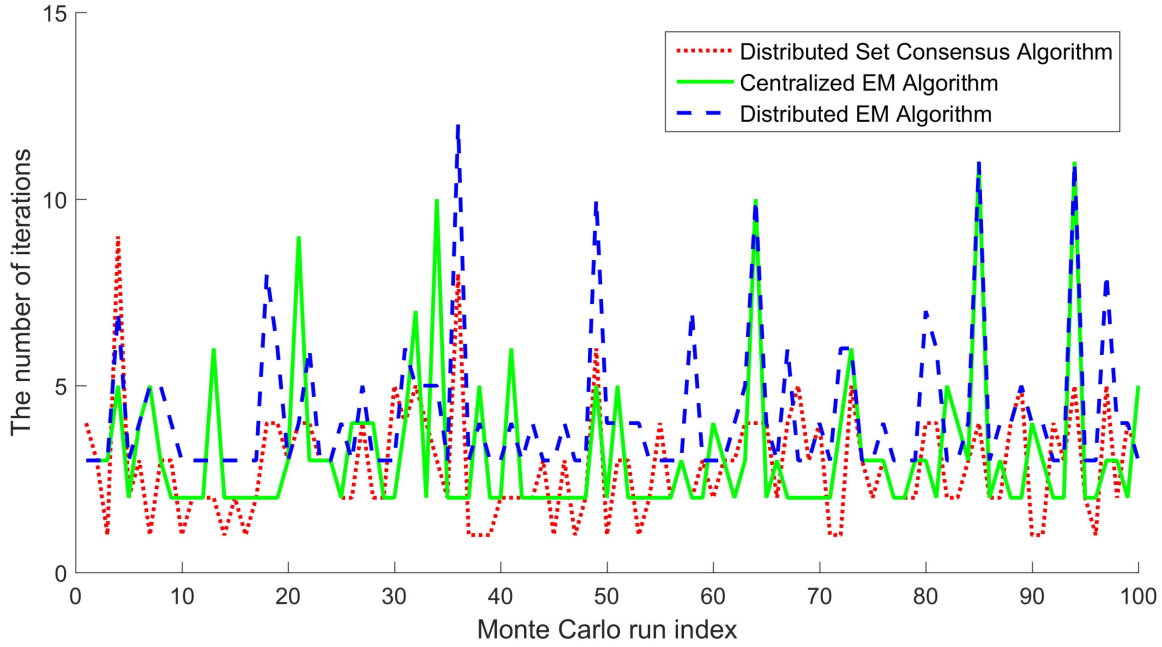


Fig. 6. The number of iterations of the distributed set consensus algorithm, the centralized EM algorithm and the EM and AC based distributed ADMM algorithm.

TABLE IX
Evolution of the target location estimates (same for each sensor) throughout the distributed EM algorithm.

Initial estimates (final from initialization consensus, 60 iterations)	After 2 iterations (converged)	After removing false targets
$\begin{bmatrix} -25.79 & 95.33 \\ -7.98 & 103.42 \\ 8.35 & 96.10 \\ 25.63 & 98.58 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.29 & 97.01 \\ -7.56 & 100.05 \\ 8.95 & 98.45 \\ 25.19 & 98.32 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.29 & 97.01 \\ -7.56 & 100.05 \\ 8.95 & 98.45 \\ 25.19 & 98.32 \end{bmatrix}$

use different initialization approaches, the initially estimated number of targets is different for the two algorithms. False targets appear in 40 runs, where the estimated number of candidate targets is greater than the true number of targets. Tables VIII and IX illustrate the consensus and distributed EM processes.

Figure 6 shows the number of iterations required for the convergence of the different iterative algorithms presented in this paper. All the algorithms terminate in a few iterations. The EM and AC based distributed algorithm, being itself an iterative algorithm, consists of three steps, two of which are iterative algorithms themselves (steps on Lines 5 and 6 in Table V). By close examination, we found that average number of iterations for these two algorithms is around 3 and 9, respectively. Since local communication only occurs at the AC step (Line 6 in Table V), the average number of communications for each sensor is approximately 50.

Figure 7 plots the number of targets estimated by the centralized and distributed algorithms before and after thresholding. Since the initialization is different for these two EM algorithms, the estimated number of targets is slightly different. In the shooter localization application, the priority is to avoid any missed target and then try to avoid as many false targets as possible. There are two possible sources for the false targets in the final solution. One is that the false alarm rate is high, which can inevitably cause the presence of some false targets. The other is that a target is split into two close targets due to the association test. While the former may cause confusion in the decision making, we may prioritize the targets based on the estimated p^{d} such that the low $\hat{p}^{\mathrm{d}}$ targets have the low priority. The latter may be solved by looking at whether two close targets with low estimated probabilities of detection have their sum close to 1.

The top plot in Figure 8 shows the RMS position error (averaged over all targets) for different cases before we remove the predicted false targets. The "Known Association," which refers to the situation when we know the number of targets and the association between measurements and targets, is meant to serve as a baseline or a lower bound (which is unachievable). In this case, the position estimates can be obtained separately for each target by solving a nonlinear least squares problem, and subsequently the position error can easily be obtained. From the same plot, it can be observed that the distributed algorithm yields the same position error as the centralized algorithm most of the time. While the baseline serves as a lower bound in most cases, it is interesting to note that the performance of the centralized algorithm or the distributed algorithm is better than the baseline in a few situations, which is due to "useful" false measurements.

The bottom plot in Figure 8 shows the RMS position error (averaged over all targets) for different cases

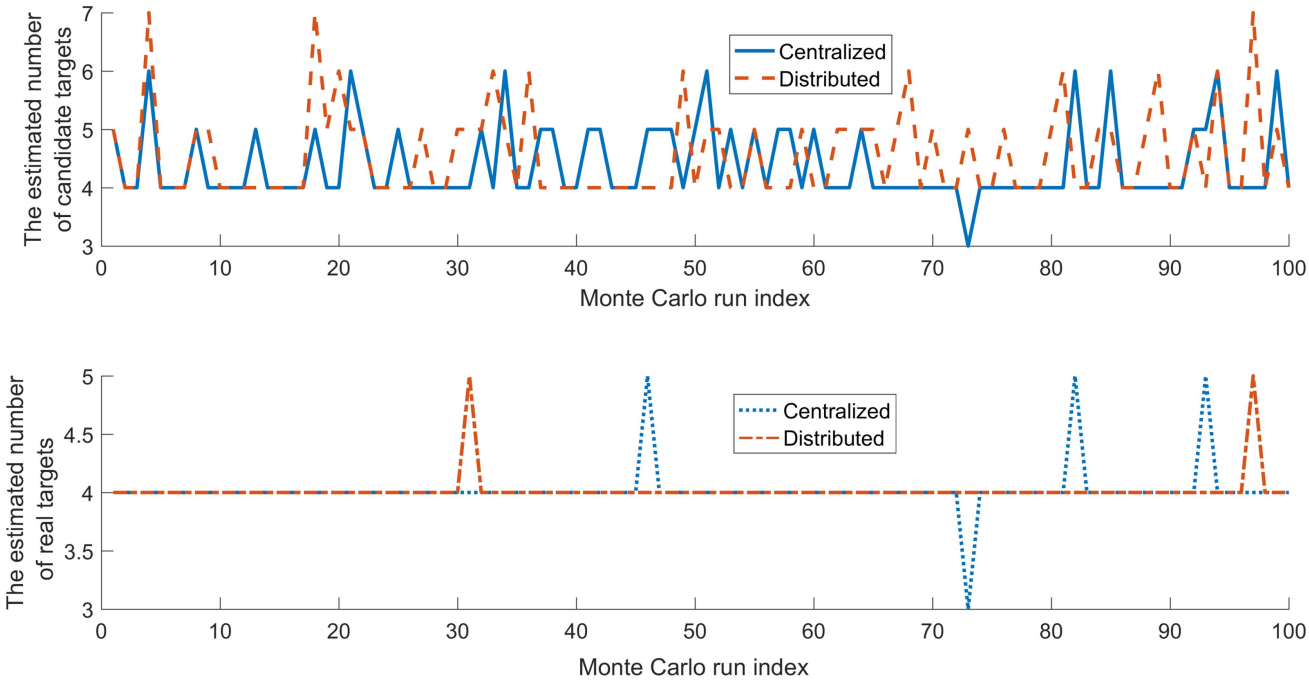


Fig. 7. The number of targets (the truth is 4) estimated by the centralized and distributed algorithms before (top plot) and after (bottom plot) removing false targets using the threshold (144).

Fig. 8. The RMS position error per target evaluated by assuming known target-measurement association, using the centralized EM algorithm and the EM and AC based distributed ADMM algorithm before and after removing the false targets.

after we remove the low $\hat{p}^{\mathrm{d}}$ false targets. This is also a measure of accuracy of the final position estimates provided by the centralized and distributed algorithm. For a clearer comparison, the range of the RMS ratio of the distributed algorithm over the centralized algorithm is also shown as in Table X. While the distributed algorithm can produce a higher error than the centralized algorithm occasionally, in most cases (84%), it yields practically the same localization result as the centralized algorithm. It is interesting to note that the distributed algorithm can be slightly better than the centralized algorithm due to a different initialization.

7. CONCLUSION

This paper considers passive localization of multiple transient emitters using a wireless sensor network and develops a distributed algorithm, which relies solely on local communications between one-hop neighboring

TABLE X
The final (after removing low pd targets) RMS ratio among assuming known association (KA), the centralized algorithm (C) and the distributed algorithm (D).

Interval	(0.9,0.99)	[0.99,1.01]	(1.01,1.1)	[1.1,1.3)	[1.3,1.5)	[1.5,8]
D versus C	3	84	3	6	1	3
C versus KA	12	75	3	4	2	2
D versus KA	15	60	6	9	6	4

sensors. A distributed implementation of the centralized EM based algorithm is not possible unless the consensus on the initial set of estimates can be reached among all sensors. It is shown by simulation that even with the knowledge of data association, we need to carry out the initialization carefully because of the bearing measurements. Random initializations based on individual sets of bearing measurements could converge to a local minimum, therefore it is necessary to use the bearing measurements from neighboring sensors. As in the centralized EM based algorithm, each node uses a sequential m-best 2-D assignment algorithm on measurements from itself and its neighbors to obtain an initial set of target estimates.

Since initially estimated target set can be different from node to node in terms of both the cardinality (this happens when the probability of detection is low or the false alarm rate is high) and the values of estimates (since each sensor uses different measurements for initialization), we developed a distributed set consensus algorithm to reach consensus on the number of candidate targets before each node can reach consensus on the target-estimate association so that a proper initialization is obtained for the EM and AC based distributed ADMM algorithm. Since a Bernoulli measurement generation model is a more realistic assumption as it reflects best the physical process of measurement generation, we presented a likelihood function based thresholding technique to determine the number of targets.

Simulation results show that the EM and AC based distributed ADMM algorithm converges very fast and yields the target location estimates that are almost as good as those of the centralized algorithm. The estimated probability of detection is shown to be able to effectively distinguish real targets from false targets.

REFERENCES

[1] J. Areta, Y. Bar-Shalom, M. Levedahl, and K. Pattipati
Hierarchical track association and fusion for a networked surveillance system.
Journal of Advances in Information Fusion, 1, 2 (Dec. 2006), 140–157.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.
Wiley, 2001.

[3] Y. Bar-Shalom, P. Willett, and X. Tian
Tracking and Data Fusion.
YBS Publishing, 2011.

[4] C. M. Bishop
Pattern Recognition and Machine Learning.
Springer-Verlag, 2006.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein
Distributed optimization and statistical learning via the alternating direction method of multipliers.
Foundations and Trends in Machine Learning, 3, 1 (Jan. 2011), 1–222.

[6] C. Y. Chong
Hierarchical estimation.
In *2nd MIT/ONR Workshop on Distributed Information and Decision Systems Motivated by Naval Command-Control-Communication (C3) Problems*, July 1979.

[7] C. Y. Chong, K. C. Chang and S. Mori
Comparison of optimal distributed estimation and consensus filtering.
In *Proceedings of 19th International Conference on Information Fusion*, Heidelberg, July 2016.

[8] C. Y. Chong, S. Mori and K. C. Chang
Distributed Multitarget Multisensor Tracking.
Multitarget-Multisensor Tracking: Applications and Advances, Volume I, Yaakov Bar-Shalom, Editor, Chapter 8, Artech House, 1990.

[9] C. Y. Chong, S. Mori, E. Tse, and R. P. Wishner
Distributed estimation in distributed sensor networks.
In *Proceedings of the 1982 American Control Conference*, 1982.

[10] C. Y. Chong, E. Tse, and S. Mori
Distributed estimation in networks.
In *Proceedings of the 1983 American Control Conference*, 1983.

[11] D. Daley, and D. Vere-Jones
An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods.
Springer-Verlag, 2003.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin
Maximum likelihood from incomplete data via the em algorithm.
Journal of the Royal Statistical Society, Series B, 39, 1 (Aug. 1977), 1–38.

[13] W. Dou, J. George, L. Kaplan, R. W. Osborne, and Y. Bar-Shalom
Evaluation of fusion algorithms for passive localization of multiple transient emitters.
Journal of Advances in Information Fusion, under review.

[14] H. Durrant-Whyte and M. Stevens
Data fusion in decentralised sensing networks.
In *Proceedings of 4th International Conference on Information Fusion*, 2001.

[15] P. A. Forero, A. Cano, and G. B. Giannakis
Distributed clustering using wireless sensor networks.
IEEE Journal of Selected Topics in Signal Processing, 5, 4 (Aug. 2011), 707–724.

[16] J. George, and L. Kaplan
Shooter localization using a wireless sensor network of soldier-worn gunfire detection systems.
Journal of Advances in Information Fusion, 8, 1 (June 2013), 15–32.

[17] J. George, L. Kaplan, S. Deligeorges, and G. Cakiades
Multi-shooter localization using finite point process.
In *Proceedings of Seventeenth International Conference on Information Fusion*, July 2014.

[18] D. Gu
Distributed EM algorithm for Gaussian mixtures in sensor networks.
IEEE Transactions on Neural Networks, 19, 7 (July 2008), 1154–1166.

[19] V. Indelman, E. Nelson, J. Dong, N. Michael and F. Dellaert
Incremental distributed inference from arbitrary poses and unknown data association: using collaborating robots to establish a common reference.
IEEE Control Systems Magazine, 36, 2(Apr. 2016), 41–74.

[20] H. W. Kuhn, and A. W. Tucker
Nonlinear Programming.
In *Proceedings of Second Berkeley Symposium on Mathematical Statistics and Probability*, 1951, 481–492.

[21] T. W. Martin and K. C. Chang
A distributed data fusion approach for mobile ad hoc networks.
In *Proceedings of 7th International Conference on Information Fusion*, 2005.

[22] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel
D-ADMM: a communication-efficient distributed algorithm for separable optimization.
IEEE Transactions on Signal Processing, 61, 10 (May 2013), 2718–2723.

[23] R. D. Nowak
Distributed EM algorithms for density estimation and clustering in sensor networks.
IEEE Transactions on Signal Processing, 51, 8 (Aug. 2003), 2245–2253.

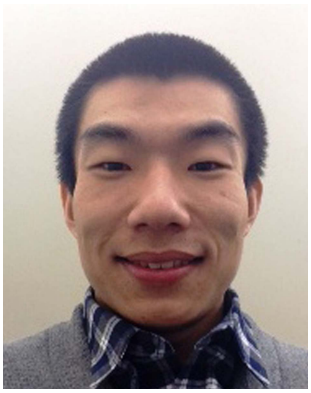
[24] K. R. Pattipati, S. Deb, Y. Bar-Shalom, and R. B. Washburn
A new relaxation algorithm and passive sensor data association.
IEEE Transactions on Automatic Control, 37, 2 (Feb. 1992), 198–213.

[25] D. Pelleg and A. W. Moore
X-means: extending K-means with efficient estimation of the number of clusters.
In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

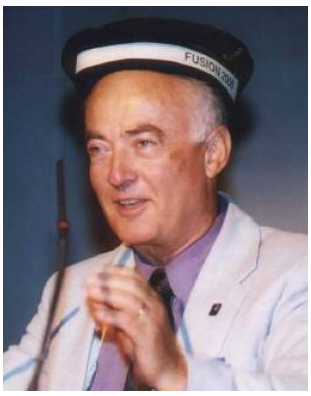
[26] I. D. Schizas, A. Ribeiro, and G. B. Giannakis
Consensus in Ad Hoc WSNs with noisy links—part I: distributed estimation of deterministic signals.
IEEE Transactions on Signal Processing, 56, 1 (Jan. 2008), 350–364.

[27] L. Xiao, and S. Boyd
Fast linear iterations for distributed averaging.
Systems and Control Letters, 53 (2004), 65–78.

[28] L. Xiao, S. Boyd, and S. Lall
A scheme for robust distributed sensor fusion based on average consensus.
In *Fourth International Symposium on Information Processing in Sensor Networks*, Apr. 2005, 63–70.

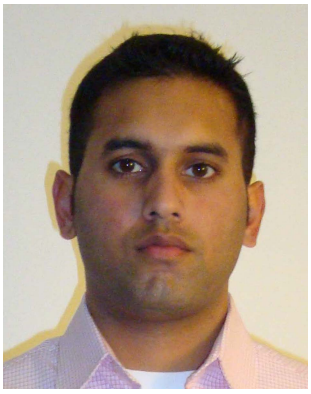


Wenbo Dou is a graduate student in the Department of Electrical and Computer Engineering at the University of Connecticut. He received the Bachelor of Engineering in Electrical and Electronic Engineering with First Class Honors from Nanyang Technological University in 2011. His current research interests include bias estimation, multi-target localization and tracking, data fusion and machine learning.



Yaakov Bar-Shalom was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in electrical engineering. From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, California. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. He is also Director of the ESP (Estimation and Signal Processing) Lab. His current research interests are in estimation theory, target tracking and data fusion. He has published over 500 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate texts *Estimation and Tracking: Principles, Techniques and Software* (Artech House, 1993; translated into Russian, MGTU Bauman, Moscow, Russia, 2011), *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction* (Wiley, 2001), the advanced graduate texts *Multitarget-Multisensor Tracking: Principles and Techniques* (YBS Publishing, 1995), *Tracking and Data Fusion* (YBS Publishing, 2011),and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Artech House, Vol. I, 1990; Vol. II, 1992; Vol. III, 2000). He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multi-target tracking." He has been consulting to numerous companies and government agencies, and originated the series of Multitarget-Multisensor Tracking short courses offered via UCLA Extension, at Government Laboratories, private companies and overseas. During 1976 and 1977 he served as Associate Editor of the *IEEE Transactions on Automatic Control* and from 1978 to 1981 as Associate Editor of *Automatica*. He was Program Chairman of the 1982 American Control Conference, General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–87 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–89 was a member of the Board of Governors of the IEEE CSS. He was a member of the Board of Directors of the International Society of Information Fusion (1999–2004) and served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications in 2004–13. In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS and has given numerous keynote addresses at major national and international conferences. He is co-recipient of the M. Barry Carlton Award for the best paper in the *IEEE Transactions on Aerospace and Electronic Systems* in 1995 and 2000 and recipient of the 1998 University of Connecticut AAUP Excellence Award for Research. In 2002 he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. In 2008 he was awarded the IEEE Dennis J. Picard Medal for Radar Technologies and Applications, and in 2012 the Connecticut Medal of Technology. He has been listed by academic.research.microsoft (top authors in engineering) as #1 among the researchers in Aerospace Engineering based on the citations of his work. He is the recipient of the 2015 ISIF Award for a Lifetime of Excellence in Information Fusion.

Lance M. Kaplan received the B.S. degree with distinction from Duke University, Durham, NC, in 1989 and the M.S. and Ph.D. degrees from the University of Southern California, Los Angeles, in 1991 and 1994, respectively, all in Electrical Engineering. From 1987–1990, Dr. Kaplan worked as a Technical Assistant at the Georgia Tech Research Institute. He held a National Science Foundation Graduate Fellowship and a USC Dean's Merit Fellowship from 1990–1993, and worked as a Research Assistant in the Signal and Image Processing Institute at the University of Southern California from 1993–1994. Then, he worked on staff in the Reconnaissance Systems Department of the Hughes Aircraft Company from 1994–1996. From 1996–2004, he was a member of the faculty in the Department of Engineering and a senior investigator in the Center of Theoretical Studies of Physical Systems (CTSPS) at Clark Atlanta University (CAU), Atlanta, GA. Currently, he is a researcher in the Networked Sensing and Fusion branch of the U.S Army Research Laboratory (ARL). Dr. Kaplan serves as Editor-In-Chief for the *IEEE Transactions on Aerospace and Electronic Systems* (AES) and as VP of Conferences for the International Society of Information Fusion (ISIF). Previously, he served on the Board of Governors of the IEEE AES Society (2008–2013) and on the Board of Directors of ISIF (2012–2014). He is a three time recipient of the Clark Atlanta University Electrical Engineering Instructional Excellence Award from 1999–2001. He is a Fellow of IEEE and ARL. His current research interests include signal and image processing, information/data fusion, network science and resource management.



Jemin George received his M.S. ('07), and Ph.D. ('10) in Aerospace Engineering from the State University of New York at Buffalo. In 2008, he was a Summer Research Scholar with the U.S. Air Force Research Laboratory's Space Vehicles Directorate and in 2009, he was a National Aeronautics and Space Administration Langley Aerospace Research Summer Scholar. From 2009–2010 he was a Research Fellow with the Department of Mathematics, Technische Universität Darmstadt, Darmstadt, Germany. Since 2010, he has been with the U.S. Army Research Laboratory. His principal research interests include stochastic systems, control theory, nonlinear filtering, information fusion, adaptive networks, distributed sensing and estimation.