12th International Conference on Information Fusion
Seattle, WA, USA, July 6-9, 2009

Dynamic Clustering and Belief Propagation for Distributed Inference in Random Sensor Networks with Deficient Links

Amadou Gning
Department of Communication Systems,
InfoLab21 Lancaster University, LA1 4WA
e.gning@lancaster.ac.uk

Lyudmila Mihaylova
Department of Communication Systems,
InfoLab21 Lancaster University, LA1 4WA
mila.mihaylova@lancaster.ac.uk

Abstract – *A fundamental issue in real-world monitoring network systems is the choice of sensors to track local events. Ideally, the sensors work together, in a distributed manner, to achieve a common mission-specific task. This paper develops a framework for distributed inference based on dynamic clustering and belief propagation in sensor networks with deficient links. We investigate this approach for dynamic clustering of sensor nodes combined with belief propagation for the purposes of object tracking in sensor networks with and without deficient links. We demonstrate the efficiency of our approach over an example of hundreds randomly deployed sensors.*

Keywords: Belief propagation, distributed inference, dynamic clustering, sensor networks, object tracking, communication failures, Markov random fields

1 Introduction

Wireless sensor networks have been recently subject to an enormous interest. They consist of a large amount of small sensor nodes equipped with cheap devices distributed in the environment. The devices gather data in real time, process it and the result serves for different purposes: situation assessment, wild life monitoring, e-health monitoring of people, "health monitoring" of aircraft wings, fire monitoring, and others.

Recent works are focused on distributed data processing [3] which will make the network robust to sensor failures, and is feasible for large networks and areas. Belief propagation [12, 11, 5] is a promising theoretical framework that has a high potential to solve inference problems in a distributed way.

Target tracking in sensor networks based on cluster formations has been widely investigated (e.g., [4], [7, 8]). By organising sensors in clusters, any event in the network can be tackled locally. This allows activating only a reduced number of necessary sensors and, then, saving energy. A common approach is the election of a leading node (head) for each cluster. Each cluster head is in charge to gather local information from the cluster and to transmit it to another unit (for instance another cluster head or a unit in a higher level).

However with randomly distributed sensors, the tracking process and the cluster updating process should integrate possibilities of coping with uncovered regions (or even presence of obstructions). In this situation, the dynamic cluster updating process should be able to activate strategic sensors in order to provide a continuous tracking process (prediction only in uncovered regions) and to transmit the relevant information. In addition, many of the works are focused on energy saving only, without studying how the cluster updating process can ameliorate the tracking process. Realistic cases such as temporary missing links, or communication failures need to be taken into account in the cluster formation. In some applications [6], the collaboration between sensors is a key issue to achieve a desirable precision.

The approaches for clustering sensors in wireless sensor networks can be classified into two big groups: 1) *Static clustering* (the most studied case); 2) *Dynamic clustering* for information transmission. The process of clustering sensor nodes in sensor networks can be performed according to different criteria such as communications, distance between sensors and energy efficiency. Ji et al. [6] classify the existing clustering approaches as: naive, scheduled monitoring, continuous monitoring, dynamic clustering and prediction based.

In this paper we present a distributed approach, in randomly distributed sensor network, for nonlinear dynamic systems and in the presence of deficient sensor links. The distributed inference is based on dynamic clustering and belief propagation. The main contributions of this work are: *i*) in the developed approach for dynamic clustering taking into account several requirements, other than energy saving, such as uncovered regions and missing messages, and *ii*) in the proposed solution to a distributed inference problem with belief propagation.

The remaining part of this paper is organised as follows. Section 2 presents the proposed approach for dynamic clustering. Section 3 describes concisely the Markov random field approach for undirected graphical models. Section 4 formulates the problem of target tracking based on distributed sensor networks and belief propagation. Section 5 illustrates our approach over a sensor network with hundreds of randomly dispersed sensors. Finally, Section 6 provides the final discussion of our work and of the results.

2 Dynamic Clustering

In this section we describe a dynamic clustering method for the purposes of distributed detection and tracking of an evolving object in a monitored area with randomly distributed sensors, such as the example shown in Figure 1. Performing a distributed inference in the entire network is intractable and energy consuming. By introducing dynamic clusters, the objective is to have a good approximation of the joint probability density function with a small number of sensors and with significant energy savings. In the next section, rules are introduced that allow us to describe the dynamic clustering approach.

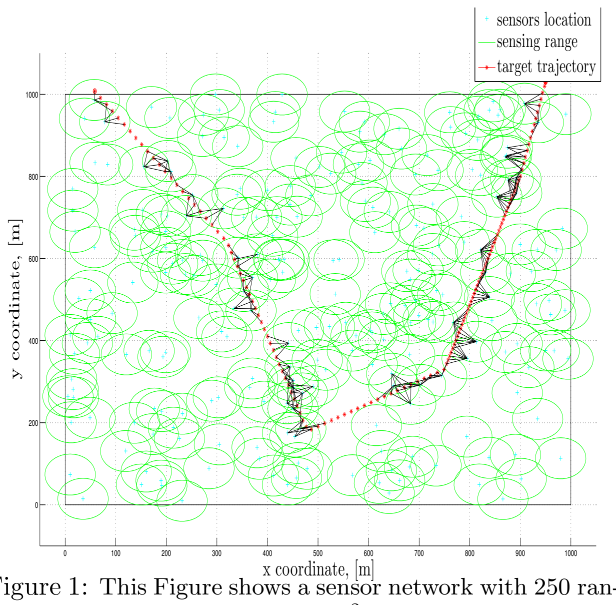

Figure 1: This Figure shows a sensor network with 250 randomly distributed sensors in 1 [km^2] two dimensional square region, the sensing range of the sensors and the trajectory of one moving object inside this square region.

2.1 Notations and Assumptions

Assume that there are N sensor nodes randomly distributed within a square region Λ (see Figure 1). Each sensor has a communication range r_c and a sensing range r_s (in general $r_c \ll r_s$). Using the communication range of each sensor, a graphical model can be introduced. Each node of the graph corresponds to a sensor. An undirected edge between two nodes means

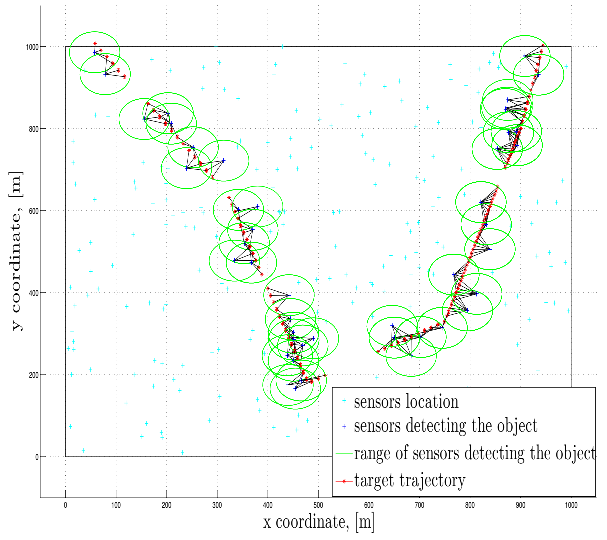

Figure 2: A sensor network with 250 randomly distributed sensors in 1 [km^2] two dimensional square region. The moving object trajectory is shown only at these places where there is at least one sensor containing the object in its range.

that the communication range for one selected node contains the other corresponding node. In this study, a directed edge is also possible since it allows to take into account the realistic case with different values for the sensors' communication range. It is then possible that, one node v_i can communicate with another node v_j and that, the reverse link is not possible due to the inferior communication range of v_j.

2.2 Cluster Model Requirements

At each time instant t, let define $\boldsymbol{C}_t \subset \{1, \ldots, N\}$ a cluster representing a group of sensors that are in charge to exchange information and track the object. In order to establish a way of choosing $\boldsymbol{C}_t$, at time instant t, we propose, in this section, to define desired properties for the clusters in terms of 4 primordial rules.

1. The first rule is trivial. $\boldsymbol{C}_t$ must contains all sensors detecting the target. The set of sensor detecting the object at time t is denoted by $\boldsymbol{C}_t^M$.

2. In randomly distributed sensor networks the first rule is not sufficient since during the evolution of the object, the set $\boldsymbol{C}_t^M$ might be empty (for example see Figure 2). There is no prediction in this case and the information from the previous steps is lost. The second rule means that at each time instant the clusters should contain nodes that ensure the cluster is not an empty set. At each time step, in some nodes of the network, an approximation of the object position exists.

3. The third rule means that two successive clusters in time, $\boldsymbol{C}_t$ and $\boldsymbol{C}_{t-1}$, must have an intersection $\boldsymbol{C}_t \cap \boldsymbol{C}_{t+1} = \varnothing$. This rule ensures that there is no loss of information between two time steps.

4. A fourth rule can be introduced related with the viability of the communication network. Due to various reasons such as obstacles, sensor age, bugs, sensors quality, the network links can have diverse probabilities of failure. Additional nodes are added to cluster $\boldsymbol{C}_t$ that provides redundancy or simply establishes new links and replaces faulty links.

2.3 The Clustering Model

This Section presents a dynamic clustering model. A cluster $\boldsymbol{C}_t$ at time t, is decomposed as follows:

$$\boldsymbol{C}_t = \boldsymbol{C}_t^M \cup \boldsymbol{C}_t^E \cup \boldsymbol{C}_t^P \cup \boldsymbol{C}_t^R, \qquad (1)$$

where $\boldsymbol{C}_t^M$ represents the set of nodes detecting the target at time t (as previously defined in Section 2.2); $\boldsymbol{C}_t^E$ (E for empty) consists of the set of nodes added to the cluster $\boldsymbol{C}_t$ determined with respect to the second rule; $\boldsymbol{C}_t^P$ (P stands for the previous time step) represents a set of nodes added to the cluster $\boldsymbol{C}_t$, with a procedure determined from the third rule. Finally, $\boldsymbol{C}_t^R$ (R stands for redundancy) represents a set of nodes added to the cluster $\boldsymbol{C}_t$, determined based on the fourth rule. Additionally, in order to make difference between $\boldsymbol{C}_t^M$ and all other components of $\boldsymbol{C}_t$ appearing in the union in equation (1), we introduce $\boldsymbol{C}_t^+$ as following:

$$\boldsymbol{C}_t^+ = \boldsymbol{C}_{t-1}^E \cup \boldsymbol{C}_{t-1}^P \cup \boldsymbol{C}_{t-1}^R. \qquad (2)$$

In the next sections, models and algorithms for these cluster components are presented.

2.3.1 How to Construct the Model for $\boldsymbol{C}_t^E$

The set $\boldsymbol{C}_t^E$ is designed to model nodes added to the cluster $\boldsymbol{C}_t$ in order to ensure that $\boldsymbol{C}_t$ is not empty. At time instant $t-1$, we propose, in our model, that each node in the previous cluster $\boldsymbol{C}_{t-1}$ is able to activate neighbouring nodes. The objective is, in case of prediction only (the target position is not measured by any sensor), to add in cluster $\boldsymbol{C}_t^E$ sensors that are likely to detect the target in the near future.

Let consider v_i a node in $\boldsymbol{C}_{t-1}$ and the set of all neighbours of v_i including the node v_i itself. Any node, in this neighbourhood set, containing the predicted target position in its sensing range (or more practically, which intersects the estimated predicted position and error according to a certain criterion, for instance the Mahalanobis distance) can be activated. If, the prediction is outside the sensing zone of v_i's neighbour, the closest node to the predicted target position can be chosen. Algorithm 1, given below, summarises the procedure for activation of nodes in $\boldsymbol{C}_t^E$ by a node $v_i \in \boldsymbol{C}_{t-1}$. This way of generating $\boldsymbol{C}_t^E$ can be refined by restricting the nodes that are allowed to activate other nodes. In fact two cases can be distinguished:

1. If $\boldsymbol{C}_{t-1}^M$ is not empty, the position of the target is at least known with a precision better than r_s.

Algorithm 1. Activation of nodes in $\boldsymbol{C}_t^E$ by a node v_i in $\boldsymbol{C}_{t-1}$

```
EVALUATE  $\boldsymbol{x}_{i,t|t-1}$, state prediction in node $v_i$
EVALUATE  $\boldsymbol{P}_{i,t|t-1}$, the corresponding
          covariance matrix of $\boldsymbol{x}_{i,t|t-1}$
FOR  $k \in \{i\} \cup \mathcal{N}(i)$
    IF the sensor range intersects the ellipsoid
      formed with $(\boldsymbol{x}_{i,t|t-1}, \boldsymbol{P}_{i,t|t-1})$
        ADD $k$ in $\boldsymbol{C}_t^E$
END
IF $\boldsymbol{C}_t^E = \varnothing$
    EVALUATE $l \in \{i\} \cup \mathcal{N}(i)$ the closest node
      to the target predictions $(\boldsymbol{x}_{i,t|t-1}, \boldsymbol{P}_{i,t|t-1})$
    ADD $l$ in $\boldsymbol{C}_t^E$
END
```

Activating nodes in $\boldsymbol{C}_t^E$ only by using nodes in $\boldsymbol{C}_{t-1}^M$ affords tracking the target with a convenient number of sensors. Without this restriction, the cluster size keeps growing since each node in $\boldsymbol{C}_{t-1}$ is able to generate at least on node in $\boldsymbol{C}_t^E$.

2. On the contrary, when $\boldsymbol{C}_{t-1}^M$ is empty, the target position uncertainty is higher and it is desired to activate more nodes for $\boldsymbol{C}_t^E$. In this case, all nodes in $\boldsymbol{C}_{t-1}$ can be used (except for nodes in $\boldsymbol{C}_{t-1}^R$ that are dedicated only for redundancy and for ameliorating communication links in cluster $\boldsymbol{C}_{t-1}$). Algorithm 2 describes the proposed procedure for activation of nodes in $\boldsymbol{C}_t^E$ using nodes in $\boldsymbol{C}_{t-1}$.

Algorithm 2. Activation of nodes in $\boldsymbol{C}_t^E$ using nodes in $\boldsymbol{C}_{t-1}$

```
IF  $\boldsymbol{C}_{t-1}^M \neq \varnothing$
    FOR  $k \in \boldsymbol{C}_{t-1}^M$
        APPLY Algorithm 1 to $k$
        ADD news elements in $\boldsymbol{C}_t^E$
    END
ELSE
    FOR  $k \in \boldsymbol{C}_{t-1}^E \cup \boldsymbol{C}_{t-1}^P$
        APPLY Algorithm 1 to $k$
        ADD news elements in $\boldsymbol{C}_t^E$
    END
END
```

2.3.2 The Model for $\boldsymbol{C}_t^P$

The set $\boldsymbol{C}_t^P$ is added to the cluster $\boldsymbol{C}_t$ to ensure that information from the previous cluster $\boldsymbol{C}_{t-1}$ is continuously propagated. A natural solution is to have nodes in common, i.e., $\boldsymbol{C}_{t-1}^P \cup \boldsymbol{C}_t^P = \varnothing$. The difficulty is to determine which particular nodes to add in order to satisfy this rule. Similarly to the previous section, we propose to distinguish two cases:

1. if $\boldsymbol{C}_{t-1}^M$ is not an empty set, then the nodes in $\boldsymbol{C}_{t-1}^M$ give a good representation of information at time

$t-1$. A simple solution is to consider nodes in $\boldsymbol{C}_{t-1}^M$ to be activated in the next step i.e., $\boldsymbol{C}_t^P = \boldsymbol{C}_{t-1}^M$.

2. if $\boldsymbol{C}_{t-1}^M$ is empty, the target position uncertainty is higher and in order to have the continuity between the cluster, $\boldsymbol{C}_{t-1}^E \cup \boldsymbol{C}_{t-1}^P$ is a good representation of the past (the set $\boldsymbol{C}_{t-1}^R$ is not added in $\boldsymbol{C}_t^P$ since $\boldsymbol{C}_{t-1}^R$ is dedicated for redundancy and for ameliorating communication links in cluster $\boldsymbol{C}_{t-1}$). Table 3 describes the proposed algorithm.

Algorithm 3. Activation of nodes in $\boldsymbol{C}_t^P$ using nodes in $\boldsymbol{C}_{t-1}$

```
IF  $\boldsymbol{C}_{t-1}^M \neq \varnothing$
    $\boldsymbol{C}_t^P = \boldsymbol{C}_{t-1}^M$
ELSE
    $\boldsymbol{C}_t^P = \boldsymbol{C}_{t-1}^E \cup \boldsymbol{C}_{t-1}^P$
END
```

2.3.3 The Model for $\boldsymbol{C}_t^R$

The set $\boldsymbol{C}_t^R$ is added to the cluster $\boldsymbol{C}_t$ in order to solve communications and failure issues. Therefore, the way of activating nodes in $\boldsymbol{C}_t^R$ is very dependent on the application. For example, in one application, nodes can be activated in order to create a path between two nodes in $\boldsymbol{C}_t^M$. In this case, creating this path can ameliorate the estimations. Another application can be seen in the case where, for each link in the network, an estimation of the link quality can be obtained. Then, optimisation algorithms can be introduced in order to choose supplementary nodes $\boldsymbol{C}_t^R$ that will ameliorate the communication in the cluster $\boldsymbol{C}_t^M \cup \boldsymbol{C}_t^E \cup \boldsymbol{C}_t^P$.

Algorithm 4. Activation of nodes in $\boldsymbol{C}_t^R$ using nodes in $\boldsymbol{C}_t^M \cup \boldsymbol{C}_t^E \cup \boldsymbol{C}_t^P$

```
FOR all existing links $(i,j)$ in $\boldsymbol{C}_t^M \cup \boldsymbol{C}_t^E \cup \boldsymbol{C}_t^P$
    IF  $p_{i,j} > \epsilon$
        FIND max$(p_{i,k} * p_{k,j})$ for nodes $k$
          having link both with $i$ and $j$
        IF max$(p_{i,k} * p_{k,j}) \leq \epsilon$
            CHOOSE one $k_{(i,j)}$
              maximising the path probability
            ADD $k_{(i,j)}$ to $\boldsymbol{C}_t^R$
            DEACTIVATE the link $(i,j)$
              in the corresponding graph of $\boldsymbol{C}_t$
        END
    END
END
```

In this paper, we are focused on the second case with missing messages between nodes in the clusters. Let assume that, in order to simulate deficient links, for each direct link (i,j), a probability $p_{i,j}$ of missing messages from i to j is associated. A simple model for $\boldsymbol{C}_t^R$, is presented as Algorithm 4. A threshold ϵ is introduced and an alternative path is searched if the probability $p_{i,j}$ of one link (i,j) is above the threshold ϵ. Algorithm 4 is able to cope with sensor deficient links when the sensor communication coverage of the sensor network allows finding common neighbourhood between any two nodes.

2.4 Clustering Evolution Model

This section resumes the clustering models in one evolution model $\boldsymbol{C}_t = f(\boldsymbol{C}_{t-1}, \boldsymbol{C}_t^M, \boldsymbol{X}_{t-1})$. The matrix $\boldsymbol{X}_{t-1}$ contains the state vectors of the targets at time instant $t-1$ and f denotes the desired evolution model.

The evolution model for the selected sensors can be described by the expression

$$\begin{cases} t = 0, \boldsymbol{C}_0 = \boldsymbol{C}_0^M \cup \boldsymbol{C}_0^R, \\ t > 0, \boldsymbol{C}_t = \boldsymbol{C}_t^M \cup \boldsymbol{C}_t^+, \text{where} \end{cases} \quad (3)$$

$$\boldsymbol{C}_t^+ = f^R(f^E(\boldsymbol{C}_{t-1}, \boldsymbol{X}_{t-1}) \cup f^P(\boldsymbol{C}_{t-1}, \boldsymbol{X}_{t-1})) \quad (4)$$

and the four distinctive rules (Algorithms 1-4) are taken into account. Additionally, f^E is the clustering prediction model defined in subsection 2.3.1 for $\boldsymbol{C}_t^E$; f^P is the model defined in subsection 2.3.2 for $\boldsymbol{C}_t^P$; f^R is the model defined in subsection 2.3.3 for $\boldsymbol{C}_t^R$.

For $t = 0$, $\boldsymbol{C}_0$ contains $\boldsymbol{C}_0^M$ the set of sensors measuring the target union $\boldsymbol{C}_0^R$ a set of nodes added for communication issues. For any other time $t > 0$, $\boldsymbol{C}_t$ is built using four sets satisfying the desired rules described in Section 2.2.

3 Markov Random Fields and Belief Propagation

3.1 Markov Random Fields

In Markov Random Fields (MRFs), the conditional probability density function (pdf) for the set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of latent variables given the set $\mathbf{Z}_C = \{\mathbf{z}_1, \ldots, \mathbf{z}_M\}$ of observed variables can be expressed as follows:

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N|\mathbf{Z}) = \frac{1}{\kappa} \prod_{i,j \in \mathcal{V}} \psi_i(\mathbf{x}_i, \mathbf{x}_j) \prod_{C \in \mathcal{C}} \psi_C(\mathbf{Z}_C|\mathbf{x}_i), \quad (5)$$

where κ is a normalisation constant, $\mathcal{V}$ is the set of vertices' indices; $\mathcal{C}$ denotes the set of cliques (a clique is defined as a subset of the nodes in a graph such that there exists a link between all pairs of nodes in the subset [2]), $\psi_i(\mathbf{x}_i, \mathbf{x}_j)$ is a compatibility function between nodes i and j and $\psi_C(\mathbf{Z}_C|\mathbf{x}_i)$ represents the effect of the local sensors on the belief in node i, i.e., this is the likelihood function of the ith sensor. This representation will be used in the remaining of the paper.

3.2 Belief Propagation

An ultimate goal for inference problems is to compute the posterior marginal probability density functions

$$p(\mathbf{x}_i|\mathbf{Z}) = \int_{\mathbf{x} \setminus \mathbf{x}_i} p(\mathbf{x}|\mathbf{Z}) d_{\mathbf{x} \setminus \mathbf{x}_i} \quad (6)$$

for each $\mathbf{x}_i$. Using these marginal pdfs, estimates of the state vector $\mathbf{x}_i$, given all the observations $\mathbf{Z}$, can be calculated on each node i.

For MRFs with a tree structured graph, a tractable exact inference can be computed using the belief propagation (BP) algorithm [10, 12] also called the sum-product algorithm. The idea of BP is to compute the marginal $p(\mathbf{x}_i|\mathbf{Z})$ at each node using a message passing process between nodes. In the case of tree structures, the implementation of the BP follows a two-pass form: first sending messages from the leaves to the node considered as the root, and then downwards from the root to the leaves. A message $\boldsymbol{m}_{j,i}$ from a node j to a node i can be expressed as follows

$$\boldsymbol{m}_{j,i}(\boldsymbol{x}_i) \propto \int_{\mathbf{x}_j} \psi_{j,i}(\boldsymbol{x}_j, \boldsymbol{x}_i)\psi_j(\boldsymbol{x}_j) \prod_{k \in \mathcal{N}(j)\backslash i} \boldsymbol{m}_{k,j}. \quad (7)$$

Then the belief function at node i that represents an approximation to the marginal distribution is given by

$$\boldsymbol{b}_i(\boldsymbol{x}_i) \propto \psi_i(\boldsymbol{x}_i) \prod_{k \in \mathcal{N}(i)} \boldsymbol{m}_{j,i}(\boldsymbol{x}_i). \quad (8)$$

For graphs with cycles, exact algorithms solving the MRF inference do exist but unfortunately they are intractable and their real-time applications are questionable. However, approximated algorithms can be implemented in the form of loopy belief propagation (LBP) [10, 12]. Using the same messages passing ideas, the messages can be computed in parallel in a loop until reaching convergence. A message $\boldsymbol{m}_{j,i}^{(n+1)}$ from a node j to a node i and the belief $\boldsymbol{b}_i^{n+1}$ at node i, at the $(n+1)$ iteration, similarly to (7), can be expressed as follows

$$\boldsymbol{m}_{j,i}^{(n+1)}(\boldsymbol{x}_i) \propto \int_{\mathbf{x}_j} \psi_{j,i}(\boldsymbol{x}_j, \boldsymbol{x}_i)\psi_j(\boldsymbol{x}_j) \prod_{k \in \mathcal{N}(j)\backslash i} \boldsymbol{m}_{k,j}^{(n)}, \quad (9)$$

$$\boldsymbol{b}_i^{n+1}(\boldsymbol{x}_i) \propto \psi_i(\boldsymbol{x}_i) \prod_{k \in \mathcal{N}(i)} \boldsymbol{m}_{j,i}^{n+1}(\boldsymbol{x}_i). \quad (10)$$

4 Target Tracking Based on Distributed Sensor Networks

Consider the problem of tracking the motion of a target moving in the sensor network area. Each node i is characterised by its state vector $\boldsymbol{x}_{t,i} = (x_{t,i}, \dot{x}_{t,i}, y_{t,i}, \dot{y}_{t,i})'$ (comprising the target positions $x_{t,i}$, $y_{t,i}$ and velocities $\dot{x}_{t,i}$, $\dot{y}_{t,i}$ in x and y directions, respectively); $'$ denotes the transpose operation. The state associated with all nodes in the network is denoted by $\boldsymbol{X}_t = (\boldsymbol{x}_{t,1}, \ldots, \boldsymbol{x}_{t,N})$. At time instant t, a set of N_t sensors denoted previously $\boldsymbol{C}_t^M$ (in Section 2.2), possibly empty, produces a measurement matrix $\boldsymbol{Z}_t = (\boldsymbol{z}_{t,1}, \ldots, \boldsymbol{z}_{t,N_t})$ of the evolving target. Assuming that measurement likelihood functions $p(\boldsymbol{z}_{t,i}|\boldsymbol{x}_{t,i})$ can be calculated in each node v_i belonging to C_t^M, the purpose is to compute sequentially the state pdf in each node contained in a dynamically estimated cluster $\boldsymbol{C}_t$. An augmented state vector is introduced constituted of the nodes state $\boldsymbol{X}_t$ and the cluster $\boldsymbol{C}_t$ at time instant t.

In the next Section, the joint pdfs resulting from dynamic clustering are studied.

4.1 Probability Density Functions for Clusters in Sensor Network

The process of dynamic clustering of the sensor nodes necessitates to estimate different joint pdfs. Sensors outside a cluster at time t are assumed to contain uniform distribution of the target position over a region. In other words, instead of the pairwise MRF expression (5), the aim is to estimate an approximated joint pdf which has the following expression:

$$p(\mathbf{X_t}|\boldsymbol{C}_t) = \frac{1}{\kappa} \prod_{i \in \boldsymbol{C}_t} \psi_i(\mathbf{x}_i) \prod_{(i,j) \in \mathcal{E}_{\boldsymbol{C}_t}} \psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) \prod_{i \notin \boldsymbol{C}_t} U(\mathbf{x}_i), \quad (11)$$

where κ denotes a normalisation constant; $\mathcal{E}_{\boldsymbol{C}_t}$ is the subset of edges linking nodes in $\boldsymbol{C}_t$; $U(\mathbf{x}_i)$ represents a uniform distribution of the target position over the considered region (a sufficiently wide region is equivalent to a lack of any information about the target).

4.2 Bayesian Formulation

Under the Markovian assumption for the state transition, the Bayesian *prediction* and *filtering steps* can be written respectively as follows:

$$\begin{aligned} p(\boldsymbol{X}_t, \boldsymbol{C}_t|\boldsymbol{Z}_{1:t-1}) &= \textstyle\int p(\boldsymbol{X}_t, \boldsymbol{C}_t|\boldsymbol{X}_{t-1}, \boldsymbol{C}_{t-1}) \times \\ &\quad p(\boldsymbol{X}_{t-1}, \boldsymbol{C}_{t-1}|\boldsymbol{Z}_{1:t-1}) d\boldsymbol{X}_{t-1} \\ &= \textstyle\int p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}, \boldsymbol{C}_t) \times p(\boldsymbol{C}_t|\boldsymbol{X}_{t-1}, \boldsymbol{C}_{t-1}) \times \\ &\quad p(\boldsymbol{X}_{t-1}, \boldsymbol{C}_{t-1}|\boldsymbol{Z}_{1:t-1}) d\boldsymbol{X}_{t-1} \end{aligned} \quad (12)$$

$$p(\boldsymbol{X}_t, \boldsymbol{C}_t|\boldsymbol{Z}_{1:t}) = \frac{p(\boldsymbol{Z}_t|\boldsymbol{X}_t, \boldsymbol{C}_t) \times p(\boldsymbol{X}_t, \boldsymbol{C}_t|\boldsymbol{Z}_{1:t-1})}{p(\boldsymbol{Z}_t|\boldsymbol{Z}_{1:t-1})}, \quad (13)$$

where $\boldsymbol{Z}_{1:t}$ is the set of measurements up to time t.

5 Results

In this section, the dynamic clustering process is combined with the belief propagation algorithm in order to track a moving target in a network of randomly distributed sensors. The main goal is to cope with the deficient communication links between nodes (which lead to missing messages). The performance of two filters is studied for three scenarios. In the first scenario, we assume that there are no deficient links and we investigate the performance of the first filter. In the second scenario, the same filter is validated with missing messages. For the first filter, the dynamic clustering update algorithm does not activate nodes in order to cope with the missing messages, i.e. $\boldsymbol{C}^R = \varnothing$. Finally, in the third scenario, the performance of the second filter is investigated with missing messages. In this case, the dynamic clustering process activates nodes in order to tackle missing messages, i.e. $\boldsymbol{C}^R \neq \varnothing$. The evolution and observation models of these filters are given respectively in the next subsections 5.1 and 5.2. A Nonparametric Loopy Belief Propagation (NPLBP) algorithm based on particles is used in the two filters. Note that the need of nonparametric message passing algorithm is motivated by the nonlinear observation model.

At time instant t, for each node included in cluster $\boldsymbol{C}_t^P$, a particle filtering algorithm is used to propagate information from the previous time $t-1$. In addition, if a node i is in $\boldsymbol{C}_t^P$ and also in $\boldsymbol{C}_t^M$ (i.e., the node i has the target on its sensor range), the particles on each node are corrected and reweighted using a Metropolis-Hastings (MH) Algorithm.

5.1 Evolution Model of the Target

The nearly constant velocity model [9, 1] is the target model implemented on each node of the sensor network. Then the state of the target of interest is given by:

$$\boldsymbol{x}_{t,i} = \boldsymbol{A}\boldsymbol{x}_{t-1,i} + \boldsymbol{\Gamma}\boldsymbol{\eta}_{t-1,i}, \qquad (14)$$

where $\boldsymbol{A} = diag(\boldsymbol{A}_1, \boldsymbol{A}_1)$, $\boldsymbol{A}_1 = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix}$,

$\boldsymbol{\Gamma} = \begin{pmatrix} T/2 & 1 & 0 & 0 \\ 0 & 0 & T/2 & 1 \end{pmatrix}'$, T is the sampling interval and $\boldsymbol{\eta}_{t-1,i}$ is the system dynamics noise. In order to cover a wide range of motions, the velocity should be approximately constant in a straight line and the velocity change should be abrupt at each turn (especially for the direction of the velocity). The system dynamics noise $\boldsymbol{\eta}_{t-1,i}$ is represented as a sum of two Gaussian components

$$p(\boldsymbol{\eta}_{t-1,i}) = \alpha\mathcal{N}(0, \boldsymbol{Q}_1) + (1-\alpha)\mathcal{N}(0, \boldsymbol{Q}_2), \qquad (15)$$

$\boldsymbol{Q}_1 = \text{diag}(\sigma^2, \sigma_1^2)$, $\boldsymbol{Q}_2 = \text{diag}(\sigma^2, \sigma_2^2)$; σ is a standard deviation assumed common and constant for x and y; $\sigma_1 \ll \sigma_2$ are standard deviations allowing to model respectively smooth and abrupt changes in the velocity. The fixed coefficient α has values in the interval $[0, 1]$.

5.2 Observation Model

Range and bearing observations are considered as measurements, i.e., the measurement vector $\boldsymbol{z}_{t,i}$ for the moving target contains the range $r_{t,i}$ and the bearing $\beta_{t,i}$. The measurement equation is of the form:

$$\boldsymbol{z}_{t,i} = h(\boldsymbol{x}_{t,i}) + \boldsymbol{w}_{t,i}, \qquad (16)$$

where h is the nonlinear function

$$h(\boldsymbol{x}_{t,i}) = \left(\sqrt{x_{t,i}^2 + y_{t,i}^2}, \tan^{-1}\frac{y_{t,i}}{x_{t,i}}\right) \qquad (17)$$

and the measurement noise $\boldsymbol{w}_{t,i}$ is supposed to be Gaussian, with known covariance matrix $\boldsymbol{R}$.

5.3 Description of the Simulated Sensor Network

For simplicity, the communication and sensing ranges are assumed to be the same for each node in the network. In order to simulate deficient links, for each direct link (i, j), we associate a probability $p_{i,j}$ of a missing message from i to j. We simulated randomly 30% of links having $p_{i,j} = 0.7$ and 70% of links having $p_{i,j} = 0$. Figure 3 gives the resulting graph of the sensor network. Table 1 contains the parameter values of the simulation.

Table 1: Simulation parameters

Parameters	Settings
Grid	1000m×1000m
Number of nodes	150
r_s	50m
r_c	100m
Simulation duration	120s
Sampling period	1s

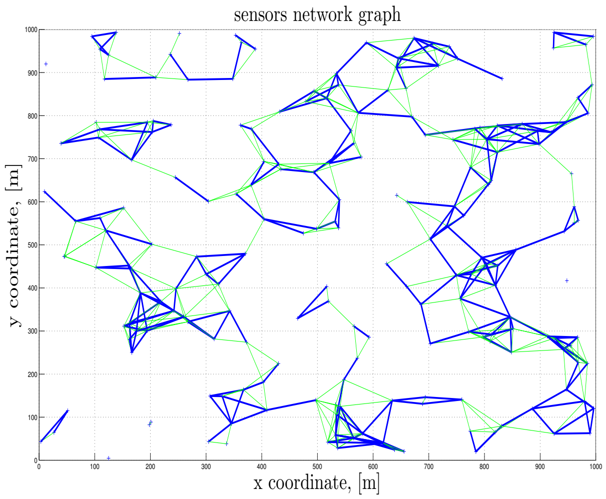

Figure 3: This Figure shows the simulated sensor network graph. The undirected links in bold are assumed to have a probability of missing message equal to 0. The others undirected links are deficient and have, at least, in one direction a probability of missing message equal to 0.7.

5.4 Filters and Simulation Results

In this section, first, the algorithm of the designed filters is given in Table 2. Then the performance of the filters is shown for one moving target, with changing velocity, evolving in the sensor network in Figure 3.

The algorithm in Table 2 combines the dynamic clustering model presented in Section 2.4 and uses a NPLBP for collaboration between the nodes in each cluster. The first step is to generate a cluster $\boldsymbol{C}_t$ at time t. Then, each node belonging to both $\boldsymbol{C}_t^P$ and $\boldsymbol{C}_t^M$ (i.e., the node that belongs to the cluster from the previous step, has a cloud of particles and contains the

Table 2: Combination of the dynamic clustering proces and nonparametric loopy belief propagation algorithn (NPLBP)

```
INITIALISATION.
t = 0, $C_0 = C_0^M$,
FOR all $i \in C_0$,
  DRAW samples from the measurement model in
    node $i$
END

LOOP.
FOR $t \geq 1$
   1. Cluster Updating.
  $C_t = f(C_{t-1}, C_t^M, X_{t-1})$ (see Section 2.4)

   2. Particles Initialisation for nodes $\in C_t$.
  FOR all $i \in C_t$,
    IF $i \in C_t^P$
      PREDICT particles from the previous
        cloud of node $i$
      IF $i \in C_t^M$
        SAMPLE new particles and
        CORRECT the weights using the
          measurements and MH steps
      END
    ELSEIF $i \in C_t^M$
      DRAW samples from the measurement
        available in node $i$
    ELSE
      The set of particles is empty
        at initialisation
    END

  3. Message passing algorithm: NPLBP.
  FOR a number of iteration = $nb_{iter}$
    RUN the message passing algorithm
    COMPUTE beliefs and estimations
      (see Section 3.2)
  END
END
```

target in its sensing range) computes a cloud of particles by propagating its cloud of particles and correcting them using MH steps. Next, nodes belonging to C_t^M draw samples only from the available current measurement. Finally, all other nodes in C_t are initialised without any particles. The last step is to use a NPLBP for the collaboration between nodes and for computing the estimates.

Two filters are developed which differ from each other in the mechanism for activation of nodes in C_t^R (see Section 2.3.3. For the first filter, the objective is to run an experiment without missing messages. Then a network with missing messages is simulated and the performance of the algorithms with and without missing messages is compared. The second filter is simulated to show the usefulness of adding nodes in C_t^R.

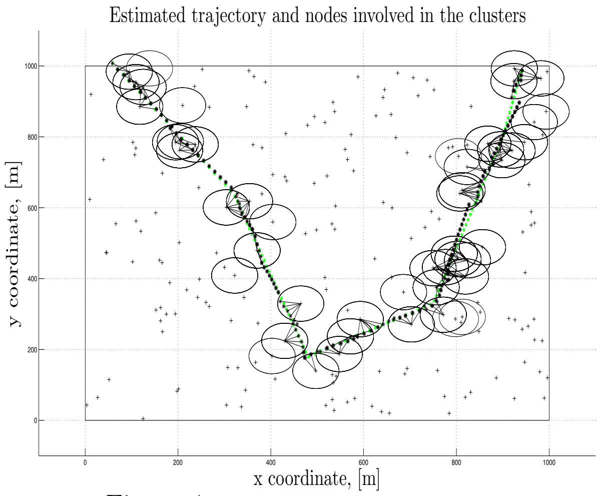

Figure 4: The estimated trajectory

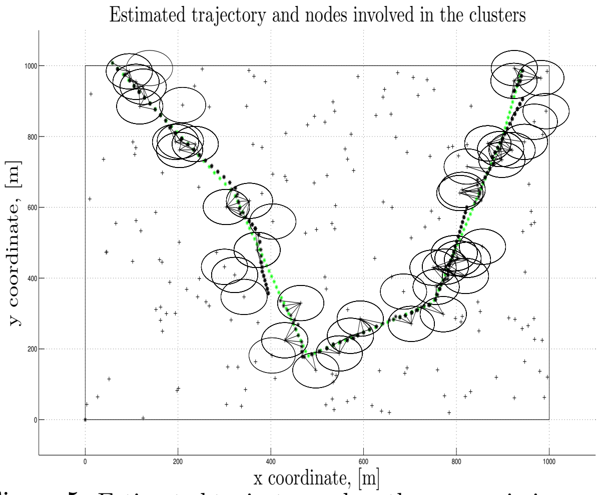

Figure 5: Estimated trajectory when there are missing messages.

Table 3: Parameters of the filters

Parameters	Settings
α	0.5
ϵ	0
Number of particles	30
Number of iterations of the MH algorithm	5
Number of iterations nb_{iter} of the NPLBP algorithm	5

Figures 4, 5 and 6 show the estimated trajectory of the target (for each time instant, estimates from the cluster are plotted). The estimated trajectory given in Figure 4 corresponds to an idealistic case without any missing message. Next, Figure 5 presents some expected effects of missing messages with respect to the

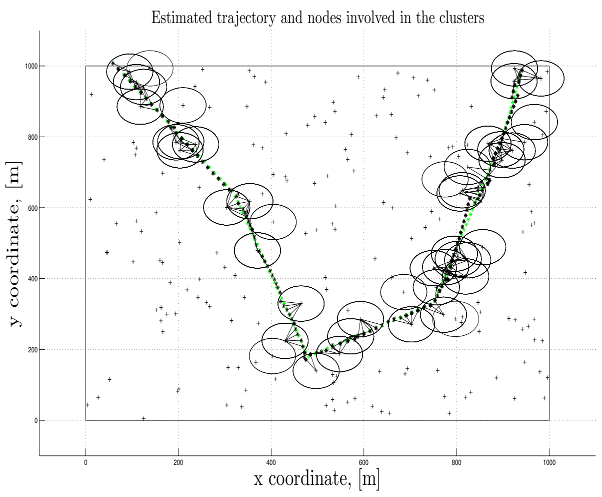

Figure 6: Estimated trajectory when there are missing messages. In order to tackle missing messages, a redundancy is introduced in the cluster updating process.

convergence rate and to the quality of the estimates (for example, around coordinates (400, 400)). Figure 6 shows that, simply by adding a rule in the cluster updating process, the effect of missing messages can be reduced.

6 Conclusions

In this paper we present an approach for dynamic clustering for a network of sensors aimed at distributed inference tasks. Dynamic clustering is combined jointly with belief propagation for the purposes of object tracking. Most of the existing works in the literature are not adapted to randomly distributed sensor networks. Dynamic clustering is a crucial part of the estimation process since, in addition to energy savings, it is tightly related with the calculation of the desired joint probability distribution function.

At every time instant the set of sensors providing the target state estimation are dynamically selected based on desired rules. First, sensors providing measurements are chosen. Then a groups of sensors from the previous time is also chosen in order to ensure that all the information from the past is continuously taken into account. Next a set of sensors is activated in order to predict target appearance in the case where there is no measurement. The last rule is related with existing deficient links and therefore missing messages.

The originality of this work is the ability to take into account the deficient links in the dynamic clustering process. Instead of trying to adapt the NPLBP algorithm, the objective is to change the joint probability density function to be estimated by adapting the clusters and their related graph.

Experiments with a challenging scenario prove the feasibility of the introduced dynamic clustering approach. In addition, by simulating an idealistic case without missing messages and a realistic case with missing messages, the experiments confirm the efficiency of the approach. In case of missing messages by adapting the cluster and hence adapting the joint probability density function to be estimated, the same performance as in the idealistic case is obtained.

Acknowledgements. We acknowledge the support of UK MOD Data and Information Fusion Defence Technology Centre (project DTC 375) and of the EPSRC, under project EP/E027253/1. We are grateful also Dr Leonid Sigal for providing us with a Matlab toolbox for the Nonparametric Belief Propagation. We would like also to thank Dr Simon Maskell for the useful discussions and advices.

References

[1] Y. Bar-Shalom and X.R. Li. *Estimation and Tracking: Principles, Techniques and Software.* Artech House, 1993.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[3] M. Çetin, L. Chen, J. Fisher, A. Ihler III, M. Wainwright, and A. Willsky. Distributed fusion in sensor networks. *IEEE Signal Proc. Mag*, 23(4):42– 55, 2006.

[4] W. P. Chen, J. C. Hou, and L. Sha. Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Trans. Mobile Comp.*, 3(3):258–271, 2004.

[5] A. Ihler. *Inference in Sensor Networks: Graphical Models and Particle Methods* . 2005 Massachusetts Institute of Technology, USA, 2005.

[6] X. Ji, H. Zha, J. J. Metzner, and G. Kesidis. Dynamic cluster structure for object detection and tracking in wireless ad-hoc sensor networks. In *IEEE International Conf. Communications, Vol.7*, pages 3807–3811, 2004.

[7] G.Y. Jin, X.Y. Lu, and M.-S. Park. Dynamic clustering for object tracking in sensor networks. In H. Y. Youn, M. Kim, and H. Morikawa, editors, *LNCS*, pages 200–209. 2006.

[8] R. Krishnan and D. Starobinski. Efficient clustering algorithms for self-organizing wireless sensor networks. *Ad Hoc Networks*, 4:3659, 2006.

[9] X. R. Li and V. Jilkov. A survey of maneuveuvering target tracking. Part I: Dynamic models. *IEEE Trans. on Aerosp. and Electr. Systems*, 39(4):1333–1364, 2003.

[10] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo, CA. Morgan Kaufmann, 1988.

[11] E. B. Sudderth, A. T. Ihler, W.T. Freeman, and A.S. Willsky. Nonparametric belief propagation. In *Proc. of Computer Visoon and Pattern Recognition, Vol I*, pages 605–612, June 2003.

[12] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. Technical report, No. 649. Dept. of Statistics, Univ. of California, Berkeley, 2003.