MOBILE AGENT BASED PROGRESSIVE MULTIPLE TARGET DETECTION IN SENSOR NETWORKS

Xiaoling Wang, Hairong Qi

University of Tennessee
Department of Electrical and Computer Engineering
Knoxville, TN 37916

ABSTRACT

In this paper, we study the multiple target detection problem in sensor networks. Due to the unique features possessed by sensor networks, the multiple target detection approach has to be energy-efficient and bandwidth-efficient. The existing centralized solutions cannot satisfy these requirements. We present a *progressive* decentralized detection approach based on the classic Bayesian source number estimation algorithm. The progressive approach is realized by a mobile agent framework, where instead of each sensor sending raw data to a central unit, each sensor processes data locally and a mobile agent is dispatched from the central unit and migrates in the network updating the estimation progressively. Experimental results show that this approach can achieve progressive accuracy with the migration of mobile agent and reduce data transmission dramatically.

1. INTRODUCTION

Multiple target detection in sensor networks remains a challenging problem due to the non-stationarity of signals and many unique features of sensor networks. Generally speaking, a sensor network is composed of hundreds or thousands of sensor nodes densely deployed in a field. Each sensor node can be an integrated entity with multiple sensing modalities, data processing capability, and wireless communication capability. Since sensor nodes are usually battery-powered and it is difficult to recharge or replace the battery on a regular base, energy conservation becomes one of the most important issues in the development of sensor network applications. The large amount of sensor nodes, the limited wireless communication bandwidth, and the energy efficiency requirement all present further difficulty to the multiple target detection problem in sensor networks.

If consider different targets in the field as sources and assume the signals they generate to be independent, multiple target detection in sensor networks can be solved using the traditional blind source separation (BSS) algorithm where the signal captured by individual sensor is a linear/nonlinear weighted mixture of the signals generated by the sources. The "blind" qualification in BSS refers to the fact that there is no *a priori* information available on the number of sources, the probabilistic distribution of source signals, or the mixing model [1, 2]. However, for conceptual and computational simplicity, the majority of BSS algorithms are developed based on a fundamental assumption: the number of sources equals the number of sensors.

Although the equality assumption is reasonable in small-size, well-controlled sensor array processing, it is not the case in sensor networks since the number of sensors is usually far more than the number of sources. Several source number estimation algorithms have been put forward in literature based on different principled approaches, such as Markov chain Monte Carlo (MCMC) method [3] and variational learning approximation [4].

The classic Bayesian estimation has been successfully implemented in several applications [5, 6]. The Bayesian framework has solid statistical basis and accommodates the usage of Bayesian fusion rule in the distributed detection hierarchy as described in our previous work [7]. However, its centralized structure is not suitable for sensor network applications as each sensor needs to send all its data to a central unit where the estimation is performed. This transmission of raw data from a large amount of sensors will occupy too much bandwidth and consume a lot of energy. This paper focuses on solving this problem by modifying the classic Bayesian estimation algorithm into a progressive detection approach based on the iterative relationship between sensors. A mobile agent framework is proposed to implement the progressive approach, where the mobile agent is transmitted between sensors and the processing of raw data is limited within each local node.

2. CLASSIC BAYESIAN SOURCE NUMBER ESTIMATION

First, as the theoretical basis of this paper, the classic Bayesian source number estimation algorithm is briefly summarized. Due to page limit, readers are referred to [5] for detailed derivation.

Suppose there are m targets in the field generating independent source signals $s_i^{(t)}, i = 1, \ldots, m$ and n sensors recording signals $x_j^{(t)}, j = 1, \ldots, n$, where $t = 1, \ldots, T$ indicates the time index of the discrete-time signals. Then the sources and the observed mixtures at time t can be denoted as vectors $\mathbf{s}^{(t)} = [s_1^{(t)}, \ldots, s_m^{(t)}]^T$ and $\mathbf{x}^{(t)} = [x_1^{(t)}, \ldots, x_n^{(t)}]^T$, respectively. If the mixing process is assumed to be linear, the observations are represented as $\mathbf{x}^{(t)} = \mathbf{A}\mathbf{s}^{(t)}$ and the sources are estimated as $\hat{\mathbf{s}}^{(t)} = \mathbf{W}\mathbf{x}^{(t)}$, where $\mathbf{A}_{n\times m}$ is the unknown non-singular scalar mixing matrix and the unmixing matrix $\mathbf{W}$ is calculated as the Moore-Penrose pseudo-inverse, $\mathbf{W} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$.

Based on this linear, instantaneous mixing model, the source number estimation can be considered as a hypothesis testing problem, where $\mathcal{H}_m$ denotes the hypothesis on a possible number of targets and the goal is to find $\hat{m}$ whose corresponding hypothesis $\mathcal{H}_{\hat{m}}$ maximizes the posterior probability given only the observation $\mathbf{x}^{(t)}$. According to Bayes' theorem, the posterior probability of the hypothesis can be written as

$$P(\mathcal{H}_m|\mathbf{x}^{(t)}) = \frac{p(\mathbf{x}^{(t)}|\mathcal{H}_m)P(\mathcal{H}_m)}{\sum_{all\ \mathcal{H}} p(\mathbf{x}^{(t)}|\mathcal{H})P(\mathcal{H})} \quad (1)$$

Assume the hypothesis $\mathcal{H}_m$ has a uniform distribution (i.e., equal prior probability), then the measurement of the posterior probability can be simplified to the calculation of likelihood $p(\mathbf{x}^{(t)}|\mathcal{H}_m)$.

In other words, the objective can be reformatted as finding $\hat{m}$ that maximizes the log-likelihood function (objective function).

As discussed in [5], maximizing the informativeness of the set of estimated sources may be achieved by making $\mathbf{W}$ as large as possible, which requires some form of constraint. An alternative approach is to linearly map the observations $\mathbf{x}^{(t)}$ to a set of latent variables, $\mathbf{a}^{(t)}$, of the form $\mathbf{a}^{(t)} = \mathbf{W}\mathbf{x}^{(t)}$, followed by a non-linear transform from this latent space to the set of source estimations, $\hat{\mathbf{s}}^{(t)} = \phi(\mathbf{a}^{(t)})$. The log-likelihood only depends on the calculation of marginal integrals of $p(\mathbf{x}^{(t)}|\mathbf{A}, \phi, \beta, \mathbf{a}^{(t)})$, which is the likelihood of $\mathbf{x}^{(t)}$ conditioned on the mixing matrix $\mathbf{A}$, the choice of non-linearity ϕ, the noise variance β, and the latent variables $\mathbf{a}^{(t)}$. By applying Laplace approximations on the marginal integrals, the log-likelihood function can be estimated as:

$$
\begin{aligned}
L(m) &= \log p(\mathbf{x}^{(t)}|\mathcal{H}_m) \\
&= \log \pi(\hat{\mathbf{a}}^{(t)}) + \frac{n-m}{2}\log(\frac{\hat{\beta}}{2\pi}) - \frac{1}{2}\log|\hat{\mathbf{A}}^T\hat{\mathbf{A}}| \\
&\quad - \frac{\hat{\beta}}{2}(\mathbf{x}^{(t)} - \hat{\mathbf{A}}\hat{\mathbf{a}}^{(t)})^2 - \frac{mn}{2}\log(\frac{\hat{\beta}}{2\pi}) \\
&\quad - \frac{n}{2}(\sum_{j=1}^{m}\log\hat{a}_j^2) - mn\log\gamma
\end{aligned} \quad (2)
$$

where $\pi(\cdot)$ is the marginal distribution of the latent variables, $\hat{\beta}$ is the noise variance on each component of sensor observations and $\hat{\beta} = \frac{1}{n-m}\langle(\mathbf{x}^{(t)} - \hat{\mathbf{A}}\hat{\mathbf{a}}^{(t)})^2\rangle$, $\hat{a}_j$ is the jth component of $\hat{\mathbf{a}}^{(t)}$, and $\gamma = 2\|\hat{\mathbf{A}}\|_\infty$.

3. A PROGRESSIVE ESTIMATION FRAMEWORK

As discussed earlier, the classic Bayesian source number estimation algorithm requires data from all the sensors to evaluate the log-likelihood of different source number hypotheses. This will generate a lot of traffic within the network and consume too much energy. In order to accommodate the unique energy efficiency requirement of sensor networks and perform target number estimation in real time, we derive a progressive approach based on the iterative relationship between sensors to evaluate the objective function in Eq. 2, i.e., to update the log-likelihood evaluation only based on its local observation and the information transmitted from its exact previous sensor.

3.1. Progressive Estimation of the Log-likelihood Function

In the progressive estimation framework, instead of all the sensors sending data to a central unit, data is processed locally at each sensor. After receiving a partial estimation result from its previous sensor, the current sensor will update the log-likelihood and the mixing matrix $\mathbf{A}$ corresponding to different source number hypotheses and then transmit the updated results to its next sensor. The updating rules at sensor i is denoted as $\mathcal{D}_i(x_i^{(t)}; I(i-1))$, where $x_i^{(t)}$ is the observation of sensor i at time t and $I(i-1)$ denotes the information received from sensor $i-1$.

Let's consider the derivation of updating rules to evaluate the log-likelihood function. From Eq. 2, it is clear that each individual term has a different iterative relation between sensors. Therefore, we derive the updating rule for each term separately.

Term 1: The first term in Eq. 2 accounts for the marginal distribution of the latent variables. In normal cases, the distribution of one latent variable $\hat{a}_k$ can be assumed to have the form $\pi(\hat{a}_k^{(t)}) = 1/(Z(\alpha)[\cosh(\alpha\hat{a}_k^{(t)})]^{\frac{1}{\alpha}})$, where $\log Z(\alpha) = a\log(\frac{c}{\alpha}+1)$, α is a scaling factor, $a = 0.522$, $b = 0.692$, and $c = 1.397$. Suppose the mixing matrix at sensor i is $\mathbf{A}_i$, then $\mathbf{W}_i = (\mathbf{A}_i^T\mathbf{A}_i)^{-1}\mathbf{A}_i^T$, and $\mathbf{w}_k$ is the kth row of $\mathbf{W}_i$. The updating rule for Term 1 can be written as:

$$
\begin{aligned}
[\log\pi(\hat{a}_k^{(t)})]_i &= [\log\pi(\hat{a}_k^{(t)})]_{i-1} - \frac{1}{\alpha}\exp(-2\alpha[\hat{a}_k^{(t)}]_{i-1}) \\
&\quad \cdot[\exp(-2\alpha w_{k,i}x_i^{(t)}) - 1] - w_{k,i}x_i^{(t)}
\end{aligned} \quad (3)
$$

Term 2: The second term takes into account the noise variance β, which is estimated by the squared errors between the real observations and their estimated counterparts. The updating rule for Term 2 is:

$$
\begin{aligned}
&[\frac{i-m}{2}\log(\frac{\hat{\beta}}{2\pi})]_i = \frac{i-m}{i-1-m}[\frac{i-1-m}{2}\log(\frac{\hat{\beta}}{2\pi})]_{i-1} \\
&+\frac{i-m}{2}\log(\frac{i-1-m}{i-m}) + \frac{i-m}{2}\cdot\frac{(x_i^{(t)} - \sum_{k=1}^{m}A_{i,k}\hat{a}_k^{(t)})^2}{\sum_{j=1}^{i-1}(x_j^{(t)} - \sum_{k=1}^{m}A_{j,k}\hat{a}_k^{(t)})^2}
\end{aligned} \quad (4)
$$

Term 3: The third term in Eq. 2 only depends on the updating rule of mixing matrix $\mathbf{A}$, which will be discussed later in this section.

Term 4: The updating rule for the fourth term in Eq. 2 is similar to that of Term 2, which is written as:

$$
\begin{aligned}
[-\frac{\hat{\beta}}{2}(\mathbf{x}^{(t)} - \hat{\mathbf{A}}\hat{\mathbf{a}}^{(t)})^2]_i &= \frac{i-1-m}{i-m}[-\frac{\hat{\beta}}{2}(\mathbf{x}^{(t)} - \hat{\mathbf{A}}\hat{\mathbf{a}}^{(t)})^2]_{i-1} \\
&+2(x_i^{(t)} - \sum_{k=1}^{m}A_{i,k}\hat{a}_k^{(t)})\cdot\sum_{j=1}^{i-1}(x_j^{(t)} - \sum_{k=1}^{m}A_{j,k}\hat{a}_k^{(t)})^2 \\
&+(x_i^{(t)} - \sum_{k=1}^{m}A_{i,k}\hat{a}_k^{(t)})^2
\end{aligned} \quad (5)
$$

Term 5: The fifth term is also affected by the noise variance $\hat{\beta}$. The updating rule is:

$$
\begin{aligned}
&[\frac{im}{2}\log(\frac{\hat{\beta}}{2\pi})]_i = \frac{i}{i-1}[\frac{(i-1)m}{2}\log(\frac{\hat{\beta}}{2\pi})]_{i-1} \\
&+\frac{im}{2}\log(\frac{i-1-m}{i-m}) + \frac{im}{2}\cdot\frac{(x_i^{(t)} - \sum_{k=1}^{m}A_{i,k}\hat{a}_k^{(t)})^2}{\sum_{j=1}^{i-1}(x_j^{(t)} - \sum_{k=1}^{m}A_{j,k}\hat{a}_k^{(t)})^2}
\end{aligned} \quad (6)
$$

Term 6: This term in Eq. 2 accounts for the estimated latent variables and its updating rule can be written as:

$$
\begin{aligned}
[\frac{i}{2}(\sum_{k=1}^{m}\log(\hat{a}_k^{(t)})^2)]_i &= \frac{i}{i-1}[\frac{i-1}{2}(\sum_{k=1}^{m}\log(\hat{a}_k^{(t)})^2)]_{i-1} + \frac{i}{2} \\
&\cdot\sum_{k=1}^{m}\frac{(w_{k,i}x_i^{(t)})^2 + 2w_{k,i}x_i^{(t)}[\hat{a}_k^{(t)}]_{i-1}}{([\hat{a}_k^{(t)}]_{i-1})^2}
\end{aligned} \quad (7)
$$

Term 7: The last term in Eq. 2 is $im\log\gamma$, where $\gamma = 2\|\hat{\mathbf{A}}\|_\infty$. This term only depends on the updating rule of matrix $\mathbf{A}$.

Progressive estimation of mixing matrix $\mathbf{A}$: At the first sensor, the mixing matrix $\mathbf{A}$ is initialized randomly. During the progressive implementation, sensor $(i-1)$ modifies matrix $\mathbf{A}_{(i-1)\times m}$ locally, and sends the resulting matrix to sensor i. After sensor i receives the information, it first adds one more dimension (an extra row) to $\mathbf{A}$ with random numbers, and then finds the optimal estimate of $\mathbf{A}_{i\times m}$ using the BFGS (Broyden-Fletcher-Goldfarb-Shanno) optimization method.

Estimation error: As the updating rules of the log-likelihood function require (Eq. 4, Eq. 5, Eq. 6), sensor i needs to update the accumulated estimation error as $error_i = error_{i-1} + (x_i^{(t)} - \sum_{k=1}^{m}A_{i,k}a_k^{(t)})^2$ and send it to its next sensor.

Program 3.1 Progressive source number estimation algorithm.

```
/*Initialization*/
At sensor 1, for each possible m:
    Initialize A (1 by m) using random numbers;
    Compute W and a;
    Compute estimation error;
    Compute each term in Eq.2;
    Compute L(m);

/*Sequential Update*/
While (max L(m)<threshold)
    Send A, latent variables a, estimation
    error and the seven terms in Eq.2 to the next
    sensor i;

    At sensor i, for each possible m:
      Add one row to A with random numbers;
      while (!converge)
        Update A using BFGS method;
      Update the accumulated estimation error;
      Update each term in Eq.2;
      Compute L(m);

/*Generate the final estimation*/
Decide m=arg max L(m);
Output m;
```

3.2. Algorithm

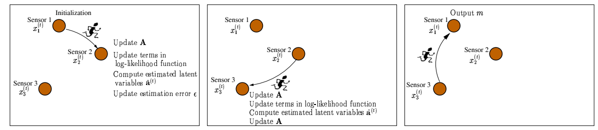
Based on the updating rules for each term in the log-likelihood function and the mixing matrix $\mathbf{A}$ discussed in Sec. 3.1, the iterative algorithm can be summarized as in Program 3.1.

4. MOBILE AGENT IMPLEMENTATION

The progressive estimation algorithm presented in Sec. 3 is potentially energy efficient in that it avoids the transmission of raw data from all the sensors to a central unit. In the progressive framework, each sensor processes its data locally and only sends the updated mixing matrix $\mathbf{A}$, the estimated latent variables $\hat{\mathbf{a}}^{(t)}$, the estimation error and the log-likelihood function. However, this approach also has some drawbacks: 1) Each sensor needs to keep a copy of the code to update matrix $\mathbf{A}$ and terms in the log-likelihood estimation. When receiving information from its previous sensor, a sensor needs a mechanism to execute the code; 2) The order of sensors in the progressive algorithm is fixed which limits its capability to adapt to the dynamically changing environment. These drawbacks make the algorithm not suitable for real-time estimation in complex environment. To solve these problems, we propose a mobile agent based approach to implement the progressive estimation.

Generally speaking, a mobile agent is a special kind of software. Once dispatched, it can migrate from sensor to sensor performing information processing autonomously. The structure of a mobile agent consists of four attributes: *identification*, *itinerary*, *data*, and *processing code* [8]. Identification uniquely identifies a mobile agent. Itinerary is the route of migration. It can be fixed or dynamically determined based on the current status of the network. Data is the mobile agent's data buffer which carries the information transmitted from one sensor to another. In progressive estimation, it carries the updated mixing matrix $\mathbf{A}$, estimated latent variables $\hat{\mathbf{a}}^{(t)}$, accumulated estimation error, and log-likelihood evaluation. Processing code carries out the updating of information when a mobile agent arrives at a local sensor.

Consider an example where the mobile agent migrates within a cluster of 3 sensors to implement the progressive estimation algorithm. The procedure of mobile agent based estimation is illustrated in Fig. 1. First, after initialization, sensor 1 will dispatch



(a) Step 1. (b) Step 2. (c) Step 3.

Fig. 1. Procedures of mobile agent based progressive estimation.

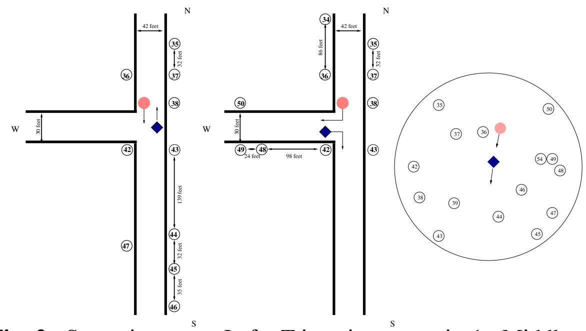


Fig. 2. Scenario setup. Left: T-junction scenario 1. Middle: T-junction scenario 2. Right: Circular parking lot scenario.

a mobile agent to sensor 2, carrying the information generated at sensor 1 ($\mathbf{A}$, $\hat{\mathbf{a}}^{(t)}$, accumulated error, log-likelihood function). As shown in Fig. 1(a), when the mobile agent arrives at sensor 2, it will read out the local observation $x_2^{(t)}$ and update the information. Before sending out the mobile agent to the next sensor, the maximum of log-likelihood corresponding to different source number hypotheses is evaluated and compared to a threshold. If it is beyond the threshold, which means the information available is sufficient to estimate the true number of sources, then the mobile agent decides that $\hat{m} = \arg\max L(m)$ and returns to sensor 1 carrying the final result $\hat{m}$. Otherwise, the mobile agent will continue its migration until it exceeds the threshold or all sensors have been visited (as shown in Fig. 1(b) and (c)). In the end, sensor 1 outputs the final source number estimation $\hat{m}$.

5. EXPERIMENTAL RESULTS

To evaluate the performance of the mobile agent based progressive estimation approach, we apply it to a test data set collected from a field demo, which is held at BAE Systems, Austin, TX in August, 2002. In the field demo, three scenarios are set up using civilian vehicles and in each case, the progressive estimation algorithm (Program 3.1) is implemented using the mobile agent framework.

In the first experiment, two civilian vehicles, a motorcycle and a diesel truck travel along the N-S road from opposite directions and intersect at the T-junction, as shown in the left figure of Fig. 2. There are 10 Sensoria WINS NG-2.0 sensors deployed along the road which capture acoustic signals generated from the targets. In our experiment, we choose sensor #38 at the T-junction to be the starting node which initiates the migration of mobile agent. The progressive estimation algorithm is applied to 1-second segments (500 samples) of the acoustic signals and the observation from each sensor is preprocessed component-wise to be zero-mean, unit-variance distributed. The progressively updated log-likelihoods at each sensor with mobile agent migration are shown in the left figure of Fig. 3. The different profiles in the figure correspond to different target number hypotheses (from

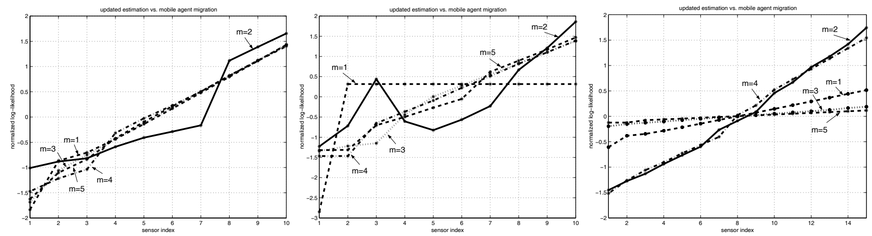


Fig. 3. Progressively estimated log-likelihood. Left: T-junction scenario 1. Middle: T-junction scenario 2. Right: Circular parking lot scenario. x-axis: Sensor index the mobile agent migrates to. y-axis: Normalized log-likelihood.

$m = 1$ to $m = 5$). Based on the average log-likelihood over the 500 samples, it is clear that the estimation accuracy is improved progressively with the migration of mobile agent and the hypothesis with the true source number ($m = 2$) has the greatest support. Actually, when the support is sufficient to make a decision, i.e., the estimated log-likelihood is beyond an appropriate threshold, the mobile agent can directly return to the first sensor and report the result instead of finishing the whole itinerary. Therefore, this approach can save response time and consume less energy.

The second scenario has the same sensor laydown as that in experiment 1. The estimation algorithm is performed when the motorcycle travels from west to south, the diesel truck from north to west and they intersect at the T-junction, as shown in the middle figure of Fig. 2. The log-likelihood estimations corresponding to different source number hypotheses averaged over 500 samples versus the migration of mobile agent are illustrated in the middle figure of Fig. 3. We observe from the figure that the log-likelihood of the hypothesis with the true source number ($m = 2$) is easily distinguished from others.

Experiment 3 uses a different scenario carried out in a parking lot with two civilian vehicles, a diesel truck and a pickup truck, traveling through a sensor network in a convoy. Fifteen sensors are deployed in a circular field with diameter of 1400 feet, as shown in the right figure of Fig. 2. The estimated log-likelihoods corresponding to different source number hypotheses versus mobile agent migration are shown in the right figure of Fig. 3. The figure exhibits similar growth pattern of the log-likelihood estimation accuracy as the mobile agent migrates among the sensor network and gives the greatest support to the true hypothesis ($m = 2$).

One significant advantage of the mobile agent based progressive source number estimation approach is the reduction of data transmission within the network. For example, in experiment 1 and 2, there are 10 sensors deployed along the road. The algorithm is performed over 1-second segments each of which consists of 500 samples. Therefore, in the classic estimation approach, 144,000 bits of data need to be transmitted, while in the mobile agent based progressive estimation approach, only 12,096 bits of data have to be transmitted, which is about 8.4% of the classic approach. In experiment 3, there are 15 sensors deployed in the field. 224,000 bits of data need to be transmitted in the centralized approach, while 27,200 bits of data need to be transmitted in the progressive approach which is 12.14% of the centralized one.

6. CONCLUSION

This paper studied the problem of source number estimation in sensor networks for multiple target detection. Due to the sheer amount of sensors deployed, the limited wireless communication bandwidth, and the battery-powered fact of the sensor node, the classic centralized approach would not provide satisfactory solution. In this paper, a progressive source number estimation algorithm is presented which is derived based on the classic Bayesian estimation algorithm using iterative relationship between sensors. A mobile agent framework is proposed to implement the progressive algorithm in sensor networks. Three experiments are conducted on the detection of multiple civilian vehicles using acoustic signals to evaluate the performance of the approach. It is shown that the mobile agent based progressive approach provides the greatest support to the true source number hypothesis in all cases and is able to achieve progressive accuracy with the migration of mobile agent. In addition, the progressive approach can reduce the amount of data transmission to 8.4% (10 sensors) and 12.14% (15 sensors) compared to the classic approach.

7. ACKNOWLEDGMENT

The authors would like to thank BAE researchers led by Joe Reynolds and Steve Beck for the tremendous effort they put in setting up the field demo.

8. REFERENCES

[1] A. J. Bell and T. J. Sejnowski, "An information-maximisation approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[2] J. Herault and J. Jutten, "Space or time adaptive signal processing by neural network models," in *Neural Networks for Computing: AIP Conf. Proc. 151*, J. S. Denker, Ed., New York, 1986, American Institute for Physics.

[3] S. Richardson and P. J. Green, "On Bayesian analysis of mixtures with an unknown number of components," *J. of the Royal Statistical Society, series B*, vol. 59, no. 4, pp. 731–758, 1997.

[4] H. Attias, "Inferring parameters and structure of latent variable models by variational Bayes," in *Proc. of the 15th Conf. on Uncertainty in Artificial Intelligence*, 1999, pp. 21–30.

[5] S. J. Roberts, "Independent component analysis: source assessment & separation, a bayesian approach," *IEE Proc. on Vision, Image, and Signal Processing*, vol. 145, no. 3, pp. 149–154, 1998.

[6] S. Roberts and R. Everson, Eds., *Independent Component Analysis: Principles and Practice*, Cambridge University Press, 2001.

[7] X. Wang, H. Qi, and H. Du, "Distributed source number estimation for multiple target detection in sensor networks," in *IEEE Workshop on Statistical Signal Processing*, St.Louis, MI, September 2003.

[8] H. Qi, X. Wang, S. S. Iyengar, and K. Chakrabarty, "High performance sensor integration in distributed sensor networks using mobile agents," *Intl. J. of High Performance Computing Applications*, vol. 16, no. 3, pp. 325–335, 2002.