Algorithms for distributed source localization in Wireless Sensor Networks

Soma Sekhar Dhavala

Contents

1 Introduction

Wireless sensor network is Network in which wireless sensors form the nodes and any useful information is shared across the nodes in the networks. Data gathered by these sensors is processed to extract useful information to make predictions/ decisions or for such related purposes. In this article, we address the problem of identifying acoustic source location using sensor networks. The sensors in the N/W make measurements of the acoustic signature impinged on it by the *source* whose position we want to estimate.

2 Model and assumptions

The acoustic source signature impinged on a sensor is modeled as [1, 2]:

$$s_j(\boldsymbol{\theta}) = \frac{a}{\|\boldsymbol{\theta} - \boldsymbol{x}_j\|^\beta} \tag{2.1a}$$

$$y_j^\kappa(i) = s_j(\boldsymbol{\theta}) + e_j^\kappa(i) \tag{2.1b}$$

where

- $y_j^\kappa(i)$ is the i^{th} measurement at the j^{th} sensor during the κ^{th} cycle,
- a is the acoustic energy of the source
- $\boldsymbol{\theta}$ and $\boldsymbol{x_j}$ are the locations of the source and the j^{th} sensor in cartesian coordinates, respectively
- $e_j^\kappa(i)$ are i.i.d. samples of zero-mean Gaussian noise process with variance σ^2 and
- β is a parameter dependent on the transmission medium

It should be noted that though the measurements represent energy, they can take still take on negative values because e_j^κ is detrended [2, Eqn. 4].

For the experimental set-up, a field of 100×100 units is chosen wherein all the sensors are scattered randomly. Sensors within the range of fifteen units are considered active and ascertain themselves to collaborate to estimate the source location. Further,to circumvent numerical instability in the model, no sensor is in the range of 1 unit. This assumption is realistic since the *source* is often away from the sensor. We also assume that $\beta = 2$ in our simulations.

3 Location estimation

The sensors can not, individually, estimate the location. In order to estimate the location, the measurements made by each of these sensors has to be shared across the networks for further processing. Based on the way this information is shared/ processed, source location estimation algorithms can be broadly classified in to: 1) Centralized and 2) Decentralized/ distributed algorithms. In centralized processing, either source location or measurements were to be made available to a central processing unit, which processes the data gathered by all

the sensors. Where as, in the distributed processing algorithms, the sensors share the location estimated modified by its own measurement to its neighbors following certain protocols. This minimizes total communication cost and henceforth bandwidth. However, it is assumed that the sensors are capable of performing some computations. We review a recently proposed decentralized source location algorithm and propose some modifications to that as well as a new approach based sequential Bayesian analysis.

In order to accomplish that, we first derive the likelihood function. Based on this likelihood function, we can formulate the location estimation problem based on both the Maximum Likelihood (ML) and sequential Bayesian analysis (SBA) methods. The likelihood function of the entire data up to the k^{th} (from here onwards referred to as global likelihood) cycle, $\boldsymbol{y}^k$, is given by:

$$L(\boldsymbol{\theta}; \boldsymbol{y}^k) = c \exp(-\frac{1}{\sigma^2} \sum_{j=1}^{M} \sum_{\kappa=1}^{k} \gamma^{k-\kappa} \sum_{i=1}^{N} (y_j^{\kappa}(i) - s_j)^2) \tag{3.1}$$

- c is a proportionality constant
- M is the number of sensors
- k is the number of cycles
- N is the number measurements being considered in one cycle (a sensor contributes only once in a cycle) at each sensor and
- γ is the recency factor ($0 < \gamma \leq 1$)

This model generalizes what was proposed in [1], i.e., for $\gamma \approx 0$, the likelihood function considers the data only from the current (k^{th}) cycle and for $\gamma = 1$, it considers all the data up to k^{th} cycle at a particular sensor. Thus, we can account for the moving nature of the source effectively. Based on this global likelihood function, we develop ML and SBA methods below.

4 Maximum Likelihood estimation

The log-likelihood function is given by:

$$\ell(\boldsymbol{\theta}; \boldsymbol{y}^k) = \ln(c) - \frac{1}{\sigma^2} \sum_{j=1}^{M} \sum_{\kappa=1}^{k} \gamma^{k-\kappa} \sum_{i=1}^{N} (y_j^{\kappa}(i) - s_j)^2 \tag{4.1}$$

We obtain the ML estimate by maximizing the above log-likelihood function. We can immediately recognize that, in fact, the ML estimate is also a least-squared estimate and we can apply any non-linear optimization to minimize least-squared cost function. Below we present a steepest-decent based algorithm.

4.1 Incremental steepest decent method

The incremental subgradient method proposed in [3] can be used to minimize functions of the form

$$f^k(\boldsymbol{\theta}) = \sum_{j=1}^{M} f_j^k(\boldsymbol{\theta}) \tag{4.2}$$

where

$$f_j^k(\boldsymbol{\theta}) = \sum_{\kappa=1}^{k} \sum_{i=1}^{N} \gamma^{k-\kappa} \left(y_j^\kappa(i) - s_j\right)^2 \tag{4.3}$$

is the local cost function. Then the parameter estimates are updated as:

$$\hat{\boldsymbol{\theta}}_j^k = \hat{\boldsymbol{\theta}}_{j-1}^k - \alpha_j^k \nabla f_j^k(\hat{\boldsymbol{\theta}}_{j-1}^k) \tag{4.4a}$$

$$\nabla f_j^k(\boldsymbol{\theta}) = \frac{4a(\boldsymbol{\theta} - \boldsymbol{x}_j)\, s_j(\boldsymbol{\theta})}{\|\boldsymbol{\theta} - \boldsymbol{x}_j\|} \sum_{\kappa=1}^{k} \gamma^{k-\kappa} \sum_{i=1}^{N} y_j^\kappa(i) - s_j(\boldsymbol{\theta}) \tag{4.4b}$$

where α_j^k is the step size in the steepest decent direction $\angle(\boldsymbol{\theta} - \boldsymbol{x_j})$. At the end of the cycle, estimate from the last sensor in the previous cycle becomes the initial estimate for the first snesor in the current cycle, i.e.,

$$\hat{\boldsymbol{\theta}}_1^{k+1} = \hat{\boldsymbol{\theta}}_M^k - \alpha_1^{k+1} \nabla f_1^{k+1}(\hat{\boldsymbol{\theta}}_M^k) \tag{4.5}$$

Now we derive the optimal step size by rewriting (2.1a) by representing the location of the source with respect to a sensor in polar coordinates as:

$$s_j(\boldsymbol{\theta}) = \frac{a}{r_j^2(\boldsymbol{\theta})} \tag{4.6}$$

where

$$\begin{aligned} r_j(\boldsymbol{\theta}) &= \|\boldsymbol{\theta} - \boldsymbol{x}_j\| \text{ and} \\ \phi_j(\boldsymbol{\theta}) &= \angle(\boldsymbol{\theta} - \boldsymbol{x}_j) \end{aligned}$$

Then, the local least squared cost function in the context of subgradiant optimization is given by:

$$f_j^k(\boldsymbol{\theta}) = \sum_{\kappa=1}^{k} \gamma^{k-\kappa} \sum_{i=1}^{N} \left[y_j^\kappa(i) - \frac{a}{r_j^2(\boldsymbol{\theta})} \right]^2 \tag{4.7}$$

We note that the only estimable quantity, given the data at the j^{th} sensor is only r_j. We can not estimate the quantity ϕ_j by solving the normal equations for the above local cost function. Henceforth, we will first estimate r_j by solving the normal equation given by:

$$\nabla f_j^k(\boldsymbol{\theta}) = \frac{4a}{r_j^3(\boldsymbol{\theta})} \sum_{\kappa=1}^{k} \gamma^{k-\kappa} \sum_{i=1}^{N} \left[y_j^\kappa(i) - \frac{a}{r_j^2(\boldsymbol{\theta})} \right] \tag{4.8}$$

The solution to this normal equation is given as:

$$\hat{r}_j^k = \left(\frac{a \sum_{\kappa=1}^{k} \gamma^{\kappa-k}}{\sum_{\kappa=1}^{k} \gamma^{\kappa-k} \bar{y}_j^k} \right)^{1/2} \tag{4.9}$$

where $\bar{y}_j^\kappa$ is the sample average (also the sufficient statistic) of the measurements at the j^{th} sensor in the κ^{th} cycle. When we represent the source location with respect to the current sensor, the uncertainty in estimating the source location

is reduced to only estimating the phase of the source location. However, we note that from (4.4b), the steepest descent direction given as

$$\hat{\phi}_j^k = \angle(\boldsymbol{\theta}_{j-1}^k - \boldsymbol{x_j}) \tag{4.10}$$

Now, we can combine both Eqn. (4.9) and Eqn. (4.10) to obtain the final estimates as:

$$\begin{array}{rcl} \hat{\theta}_{1j}^k & = & x_{1j} + \hat{r}_j^k \cos(\hat{\phi}_j^k) \\ \hat{\theta}_{2j}^k & = & x_{2j} + \hat{r}_j^k \sin(\hat{\phi}_j^k) \end{array} \tag{4.11}$$

where

$$\boldsymbol{\theta} = [\theta_1\ \theta_2]^T \text{and } \boldsymbol{x}_j = [x_{1j}\ x_{2j}]^T \tag{4.12}$$

Note that (4.11) is same as (4.4) except that α is chosen adptively:

Define d_j^{k-1} as

$$d_j^k = \|\hat{\boldsymbol{\theta}}_{j-1}^k - \boldsymbol{x}_j\|$$

then

$$\hat{\boldsymbol{\theta}}_j^k = \hat{\boldsymbol{\theta}}_{j-1}^k - \frac{d_j^k - \hat{r}_j^k}{d_j^k} [\hat{\boldsymbol{\theta}}_{j-1}^k - \boldsymbol{x}_j] \tag{4.13}$$

However, it is possible that the step size can be imaginary whenever ${}_w\bar{y}_j^k$ is negative due to noise, where, ${}_w\bar{y}_j^k$ (weighed moving average of sample averages) is given as:

$${}_w\bar{y}_j^k = \frac{1}{k}\left(\frac{k-1}{\gamma}\, {}_w\bar{y}_j^{k-1} + \bar{y}_j^k\right) \tag{4.14}$$

To combat this problem, we implement a simple threshold detector (out of many other possibilities) of the form given by:

$$\hat{r}_j^k = \begin{cases} 0 & \text{if } {}_w\bar{y}_j^k \leq th \\ \left(a \sum_{\kappa=1}^k \gamma^{k-\kappa} / (k\ {}_w\bar{y}_j^k)\right)^{1/2} & \text{otherwise} \end{cases} \tag{4.15}$$

where th is a threshold chosen based on σ^2 or it could be set zero. In either case, we leave the initial estimate of the source location unmodified whenever ${}_w\bar{y}_j^k$ is less than the threshold. This proves benifitial especially while updating the location estimate with measurements taken from a sensor having low SNR. This also suggests that the ordering of the sensors may not be as important as it were without the *forwarding* mechanism. Below, we discuss about the convergence of algorithm.

The locus of the estimate by a sensor is a circle of radius r_j with center (x_{1j}, x_{2j}). The source location is the point of intersection of sensor estimates loci. When we have the exact estimates of r_j's from at least three sensors, finding out the location becomes deterministic. Thus, the rate of convergence of location estimate depends on rate of convergence of r_j's to its true value. It is true that $\hat{r^2}_j^k$ estimated using Eqn. (4.9) is also the maximum likelihood estimate. Thus asymptotic optimality of ML estimation guarantees convergence

besides providing a vehicle to obtain bounds on the convergence rate of r_j^2. From the asymptotic properties of ML, r_j^k's convergence depends on the individual noise variances at every sensor. We can obtain the worst case estimate by considering the farthest sensor which could be away from the source by 15 units. The minimum number of measurements required can be computed since we know the variance of the noise samples a priori,given the desired accuracy of the estimate specified in terms of the confidence level.

4.2 Guass-Newton algorithm

The local log-likelihood (ignoring the constant term) is given by:

$$l_j^k(r_j) \quad = \quad -\frac{1}{\sigma^2}\sum_{\kappa=1}^{k}\gamma^{k-\kappa}\sum_{i=1}^{N}\left[y_j^\kappa(i) - \frac{a}{r_j^2(\boldsymbol{\theta})}\right]^2 \tag{4.16}$$

We will first estimate r_j by the Gauss-Newton method:

$$\hat{r}_j^k = \hat{r}_j^{k-1} - \alpha\left[\frac{\partial^2 l_j^k(r_j)}{\partial\, {r_j}^2}\right]^{-1}\left.\frac{\partial\, l_j^k(r_j)}{\partial\, r_j}\right|_{r_j=\hat{r}_j^{k-1}} \tag{4.17}$$

where

$$\begin{aligned}\hat{r}_j^{k-1} &= \|\hat{\boldsymbol{\theta}}_{j-1}^k - \boldsymbol{x}_j\| \text{ and}\\ \hat{\phi}_j^k &= \angle(\hat{\boldsymbol{\theta}}_{j-1}^k - \boldsymbol{x}_j)\end{aligned}$$

and α is a damping factor.

$$\frac{\partial\, l_j^k(r_j)}{\partial\, r_j} = -\frac{2akN}{\sigma^2 r_j^5}\left(\bar{y}_j^k r_j^2 - \bar{\gamma}^k a\right) \tag{4.18}$$

$$\frac{\partial^2\, l_j^k(r_j)}{\partial\, {r_j}^2} = \frac{2akN}{\sigma^2 r_j^6}\left(3\bar{y}_j^k r_j^2 - 5\bar{\gamma}^k a\right) \tag{4.19}$$

Finally,

$$\hat{r}_j^k = \hat{r}_j^{k-1}\left(1 + \alpha\,\frac{\bar{y}_j^k(\hat{r}_j^{k-1})^2 - \bar{\gamma}^k a}{3\bar{y}_j^k(r_j^{k-1})^2 - 5\bar{\gamma}^k a}\right) \tag{4.20}$$

where

$$\begin{aligned}\bar{y}_j^k &= (1/kN)\textstyle\sum_{\kappa=1}^{k}\gamma^{k-\kappa}\sum_{i=1}^{N} y_j^\kappa(i)\\ \bar{\gamma}^k &= (1/k)\textstyle\sum_{\kappa=1}^{k}\gamma^{k-\kappa}\end{aligned} \tag{4.21}$$

The final location estimates can be obtained by Eqn. (4.11)

4.3 Iterated extend Kalman filter

Gauss-Newton algorithms have quadratic convergence as compared to steepest decent methods. Besides the convergence aspects, they also provide an approximation of covariance matrix that can be used to analyze the performance of the estimator. The iterated extended Kalman filter (iEKF) [4] is an incremental

version of the Gauss-Newton type of algorithms. The iEKF minimizes functions of the form:

$$\boldsymbol{g}^k(\boldsymbol{\theta}) = \sum_{j=1}^{M} \|\boldsymbol{g}_j^k(\boldsymbol{\theta})\|^2 \tag{4.22}$$

In the present context, $\boldsymbol{g}_j^k(\boldsymbol{\theta})$ is given by:

$$\boldsymbol{g}_j^k(\boldsymbol{\theta}) = [\boldsymbol{g}_j^{k,1}(\boldsymbol{\theta})^T, \boldsymbol{g}_j^{k,2}(\boldsymbol{\theta})^T, \ldots, \boldsymbol{g}_j^{k,k}(\boldsymbol{\theta})^T]^T \tag{4.23a}$$

$$\boldsymbol{g}_j^{k,\kappa}(\boldsymbol{\theta}) = \sqrt{\boldsymbol{\gamma}^{k-\kappa}}(\boldsymbol{y}_j^k - \boldsymbol{s}_j(\boldsymbol{\theta})) \tag{4.23b}$$

where

- $\boldsymbol{y}_j^k$ and $\boldsymbol{s}_j^k$ are $N \times 1$ vectors
- $\boldsymbol{g}_j^k$ is $kN \times 1$ vector
- $\boldsymbol{g}_j^{k,\kappa}$ is $N \times 1$ vector

The updation equation for the parameters is given as [4, Eqn. 21]:

$$\hat{\boldsymbol{\theta}}_j^k = \hat{\boldsymbol{\theta}}_{j-1}^k - (\boldsymbol{H}_j^k)^{-1} \nabla \boldsymbol{g}_j^k(\hat{\boldsymbol{\theta}}_{j-1}^k) \boldsymbol{g}_j^k(\hat{\boldsymbol{\theta}}_{j-1}^k) \tag{4.24a}$$

$$\boldsymbol{H}_j^k = \lambda \boldsymbol{H}_{j-1}^k + \nabla \boldsymbol{g}_j^k(\hat{\boldsymbol{\theta}}_{j-1}^k) \nabla^T \boldsymbol{g}_j^k(\hat{\boldsymbol{\theta}}_{j-1}^k) \tag{4.24b}$$

where λ is a recency factor. We can use the *matrix inversion lemma* to compute the inverse as:

$$(\boldsymbol{H}_j^k)^{-1} = \frac{1}{\lambda}(\boldsymbol{H}_{j-1}^k)^{-1} - \boldsymbol{H}_g \boldsymbol{H}_g^T (\boldsymbol{I} - \nabla^T \boldsymbol{g}_j^k(\hat{\boldsymbol{\theta}}_{j-1}^k) \boldsymbol{H}_g) \text{ where} \tag{4.25a}$$

$$\boldsymbol{H}_g = \frac{1}{\lambda}(\boldsymbol{H}_{j-1}^k)^{-1} \nabla^T \boldsymbol{g}_j^k(\hat{\boldsymbol{\theta}}_{j-1}^k) \tag{4.25b}$$

We should note that there is a subtle difference between the two recency factors introduced thus far: λ weighs the samples across sensors with in a cycle and γ weighs the data across the sensors. If we choose $\lambda = 1$, the above algorithms reduces to regular Gauss-Newton algorithm and if set λ to zero, we need not pass any covariance information. It would be interesting to study the affect of different combinations of these recency factors.

5 Sequential Bayesian estimation

In the Bayesian framework, we express our *belief* or *knowledge* about the quantity to be estimated in terms of a *pdf*. There can be many ways in which one carry-out Bayesian analysis based on the way we assign as well as update the priors. We consider three approaches suggested in [5, Sec. 3.2]

- Independent Likelihood pool
- Independent Opinion pool
- Linear Opinion pool

5.1 Independent Likelihood pool

The simplest of the three methods, assigns a common prior to all the information sources (sensors). This is most appropriate when the information sources observe *i.i.d* data. We always assume normal priors and approximate posteriors also with a normal for computational purposes as well as for using minimal information in communicating the posterior.

The location estimate provided by a neighboring sensor acts as a prior to the current sensor. The prior is modified by the local likelihood function and the posterior moments are passed to the next sensor. This is the central idea in the sequential Bayesian analysis. The results should be identical irrespective of the sequential nature of processing if the data is *i.i.d*. Below, we introduce the notation:

$$\begin{aligned} \mu_1 &= E[\theta_1] \\ \mu_2 &= E[\theta_2] \\ \sigma_1^2 &= E[(\theta_1 - \mu_1)^2] \\ \sigma_2^2 &= E[(\theta_2 - \mu_2)^2] \\ \rho &= \tfrac{E[(\theta_1-\mu_1)(\theta_2-\mu_2)}{\sigma_1\sigma_2}, \end{aligned} \tag{5.1}$$

We assume normal prior with mean and covariance matrix defined above as:

$$\pi(\boldsymbol{\theta}) = \pi(\theta_2)\pi(\theta_1|\theta_2) \tag{5.2a}$$

$$\pi(\theta_2) = \frac{1}{(2\pi\sigma_2^2)^{1/2}} \exp(-\frac{(\theta_2 - \mu_2)^2}{2\sigma_2^2}) \tag{5.2b}$$

$$\pi(\theta_1|\theta_2) = \frac{1}{(2\pi\sigma_1^2(1-\rho^2))^{1/2}} \exp(-\frac{(\theta_1 - \mu_1 - \rho\frac{\sigma_1}{\sigma_2}(\theta_2 - \mu_2))^2}{2\pi\sigma_1^2(1-\rho^2)}) \tag{5.2c}$$

The local likelihood function is given by:

$$L_j^k(\boldsymbol{\theta}; \boldsymbol{y}_j^k) = c\, \exp(-\frac{1}{\sigma^2} \sum_{\kappa=1}^{k} \gamma^{k-\kappa} \sum_{i=1}^{N} (y_j^\kappa(i) - s_j)^2) \tag{5.3}$$

Here, $\boldsymbol{y}_j^k$ represents the data at the j^{th} sensor up to the k^{th} cycle. Then, the posterior density is given by,

$$p(\boldsymbol{\theta}|\boldsymbol{y}_j^k) = \frac{L_j^k(\boldsymbol{\theta}; \boldsymbol{y}_j^k)\, \pi(\boldsymbol{\theta})}{\int L_j^k(\boldsymbol{\theta}; \boldsymbol{y}_j^k)\, \pi(\boldsymbol{\theta}) d\,\boldsymbol{\theta}} \tag{5.4}$$

Under the squared loss function, posterior mean minimizes the average risk. Henceforth, we treat the posterior mean as the estimate for the quantity of interest, i.e., the location of the source. This posterior mean at the j^{th} sensor during the k^{th} cycle becomes the prior for the $(j+1)^{\text{th}}$ sensor in the same cycle. This is the basic idea behind sequential Bayesian analysis. We also compute the posterior covariance matrix and pass it as a prior in the subsequent estimation process, i.e.,

$$\pi_j^{k+1} \leftarrow p_j^k \tag{5.5a}$$

$$\hat{\boldsymbol{\theta}}_j^k \leftarrow \mu_j^k \tag{5.5b}$$

In general, it is quite cumbersome to calculate the posterior moments. By noting the fact that the prior is a normal density, we can apply a classical result in computing these posterior moments, which we will review below: The posterior expectation of a function $g(\boldsymbol{\theta})$ is given as :

$$E_j^k(g(\boldsymbol{\theta})) = \int g(\boldsymbol{\theta}) L(\boldsymbol{\theta}; \boldsymbol{y}_j^k)\, \pi_j^k(\boldsymbol{\theta})\, d\boldsymbol{\theta} \tag{5.6}$$

Upon expanding, we get,

$$= \int \pi_j^k(\theta_2) \int g(\boldsymbol{\theta})\, \pi_j^k(\theta_1|\theta_2)\, L_j^k(\boldsymbol{\theta}; \boldsymbol{y}_j^k) d\theta_1\, d\theta_1 \tag{5.7}$$

Quadrature method

A numerical approximation to the above equation is sought using the Hermite polynomials [6] which leads us to the following:

$$E_{post}[g(\theta_1, \theta_2)] \approx \sum_{P=1}^{N1} m_{1,p} \pi_j^k(z_{2,p}) \sum_{q=1}^{N2} m_{2,q}\, g(z_{1,p}, z_{2,q}) L(z_{1,p}, z_{2,q}; \boldsymbol{y}_j^k) \pi_j^k(z_{2,q}|z_{1,p}) \tag{5.8}$$

where

$$\begin{aligned}
\mu_1' &= \mu_1 + \rho \tfrac{\sigma_1}{\sigma_2}(\theta_2 - \mu_2) \\
\sigma_1' &= \sigma_1 (1 - \rho^2)^{0.5} \\
m_{1,p} &= w_{1,p}\, \exp(t_{1,p}) \sqrt{2} \sigma_2 \\
z_{1,p} &= \mu_2 + \sqrt{2}\, \sigma_2\, t_{1,p} \\
m_{2,q} &= w_{2,q}\, \exp(t_{2,q}) \sqrt{2}\, \sigma_1' \\
z_{2,q} &= \mu_1' + \sqrt{2}\, \sigma_1'\, t_{2,q}
\end{aligned} \tag{5.9}$$

and $w_{1,p}, t_{1,p}, w_{2,q}, t_{2,q}$ are the weights and abscissa of Hermite polynomials of orders $N1$ and $N2$, respectively. We can estimate posterior mean vector and covariance matrix by choosing an appropriate $g(\theta_1, \theta_2)$. To be specific,

$$\begin{aligned}
\hat{\theta_1} &= E_{post}[\theta_1] \\
\hat{\theta_2} &= E_{post}[\theta_2] \\
\hat{\sigma_1^2} &= E_{post}[\theta_1^2] - \hat{\theta_1}^2 \\
\hat{\sigma_2^2} &= E_{post}[\theta_2^2] - \hat{\theta_2}^2 \\
\hat{\rho} &= \tfrac{E_{post}[\theta_1 \theta_2] - \hat{\theta_1}\hat{\theta_2}}{\hat{\sigma_1}\hat{\sigma_2}}
\end{aligned} \tag{5.10}$$

adaptive Quadrature method

In the adaptive quadrature methods, we approximate the posterior with a normal near the mode and use this normal approximation as the quadrature kernel [7]. More precisely,

$$E[g(\theta)] = \int g(\theta) \frac{L(\theta)\,\pi(\theta)}{\psi(\theta)} \psi(\theta)\, d\theta \tag{5.11}$$

where $\psi(\theta)$ is a normal approximation to the posterior at the MAP estimate given by:

$$\theta_{MAP} = \arg\max_{\theta} p(\theta) \tag{5.12a}$$

$$\psi(\theta) = \mathcal{N}(\theta_{MAP}, -\left[\frac{\partial^2 \log\, p(\theta)}{\partial\theta\partial\theta^T}|\theta = \theta_{MAP}\right]^{-1}) \tag{5.12b}$$

We can use Gauss-hermite quadrature methods by replacing $\pi(\theta)$ with $\psi(\theta)$. The derivation is given below:

Let $x = [x_1, x_2]^T, \theta_l = (\theta - x)$ and $\theta_\pi = (\theta - \mu)$, then

$$\log p(\theta) \propto -\frac{N}{\sigma^2}\left(\bar{y} - \frac{A}{\theta_l^T\theta_l}\right)^2 - \left(\theta_\pi^T \Sigma^{-1} \theta_\pi\right) \tag{5.13}$$

$$\frac{\partial \log p(\theta)}{\partial\theta} = -\frac{2N}{\sigma^2}\left(\bar{y} - \frac{A}{\theta_l^T\theta_l}\right)\frac{2A\theta_l}{\left(\theta_l^T\theta_l\right)^2} - 2\Sigma^{-1}\theta_\pi \tag{5.14}$$

$$\frac{\partial^2 \log p(\theta)}{\partial\theta\partial\theta^T} = -\frac{2N}{\sigma^2}\frac{4A^2\theta_l\theta_l^T}{\left(\theta_l^T\theta_l\right)^4} - \frac{4AN}{\sigma^2}\left(\bar{y} - \frac{A}{\theta_l^T\theta_l}\right)\frac{\theta_l^T\theta_l I - 4\theta_l\theta_l^T}{\left(\theta_l^T\theta_l\right)^3} - 2\Sigma^{-1} \tag{5.15}$$

We can use some optimization techniques to obtain an estimate of θ_{MAP}.

This completes the estimation process.

5.2 Independent Opinion pool

It in the previous section, we have assumed a common prior to all the sensors. Hence, the joint likelihood is the product of the individual likelihoods. However, we were vary vague in specifying a common prior to all the sensors. For example, based on prior knowledge, a sensor can identify a target well with-in a region and tries identify itself in participating in the estimation process only if its received signl strength is above a certain threshold. Hence, it is reasonable to assume a prior whose mean is near the sensor it self and variance being inversely proportional to the threshold. We compute local posteriors at each sensor and make inference from the joint posteriors. Now $\pi_j^k(\boldsymbol{\theta})$ refers to the local prior of the location at the jth sensor and $p_j^k(\boldsymbol{\theta})$ is the posterior as given by (5.4). The posterior distribution of previous sensor is $p_{j-1}^k(\boldsymbol{\theta})$. Since the posteriors are approximated as normal densities, Independent Opinion pool is also normal, given by:

$$p_{new} \propto p_j^k(\boldsymbol{\theta}) p_{j-1}^k(\boldsymbol{\theta}) \tag{5.16a}$$

$$p_{new}(\boldsymbol{\theta}) \sim N(\mu_{new}, \Sigma_{new}) \tag{5.16b}$$

$$\Sigma_{new} = (\Sigma_j^{k\,-1} + \Sigma_{j-1}^{k,\,-1})^{-1} \tag{5.16c}$$

$$\mu_{new} = \Sigma_{new}\left(\Sigma_j^{k\,-1}\mu_j^k + \Sigma_{j-1}^{k,\,-1}\mu_{j-1}^k\right) \tag{5.16d}$$

We assign p_{new} as the prior for the jth sensor and pass this as the jth sensors posterior to $j+1$th sensor and the recursion follows i.e.:

$$p_j^k \leftarrow p_{new} \tag{5.17a}$$
$$\pi_j^{k+1} \leftarrow p_{new} \tag{5.17b}$$

5.3 Linear Opinion pool

In this method, we assign some weights to all the posteriors from the information sources. The posterior in the Linear Opinion pool is given as:

$$p_{new} = \sum_{j=1}^{M} w_j p_j^k \tag{5.18}$$

subject to the constraint that $\sum_{j=1}^{M} w_j = 1$. In sequential processing, we have two posterior densities at a time. Hence, the above equation reduces to

$$p_{new} = w_1 p_j^k + w_2^k p_{j-1}^k \tag{5.19}$$

Since the posteriors are normal, we obtain a closed form expression for p_{new} by minimizing symmetric Kullback-Leibler (KL) distance given by [8, Sec. 4.3]:

$$p_{new}(\boldsymbol{\theta}) \sim N(\mu_{new}, \Sigma_{new}) \tag{5.20a}$$
$$\boldsymbol{\mu}_{new} = w_1 \boldsymbol{\mu}_j^k + w_2 \boldsymbol{\mu}_{j-1}^k \tag{5.20b}$$
$$\boldsymbol{\Sigma}_{new} = w_1(\boldsymbol{\Sigma}_j^k + \boldsymbol{\mu}_j^k \boldsymbol{\mu}_j^{T\,k}) + w_2(\boldsymbol{\Sigma}_{j-1}^k + \boldsymbol{\mu}_{j-1}^k \boldsymbol{\mu}_{j-1}^{T\,k}) - \boldsymbol{\mu}_{new} \boldsymbol{\mu}_{j,new}^{T\,k} \tag{5.20c}$$

It is possible to approximate the covariance matrix in the above equation in the following manner:

$$\boldsymbol{\Sigma}_{new} = w_1 \boldsymbol{\Sigma}_j^k + w_1 w_2 (\mu_j^k - \boldsymbol{\mu}_{j-1}^k)(\boldsymbol{\mu}_j^k - \boldsymbol{\mu}_{j-1}^k)^T \tag{5.21}$$

We found from our simulations that the above approximation is serving well and results in huge savings since only posterior mean is communicates.

6 Results and discussion

We have simulated the sensor network with a stationary source located as shown in Fig. 1. We replicate Nowaks' algorithm for comparison purposes (shown in Fig. 2). In Fig. 3, we show the convergence properties of different algorithms. Of all, the SBA with local prior performs better than other. It should also be noted that only mean information is passed from one sensor to the other unlike full covariance information sent in the SBA with global prior.

A Centralized estimation Algorithms

In this Section, we provide algorithms for centralized estimation. In particular, we calculate the Fisher information matrix and use it in deriving the Fisher

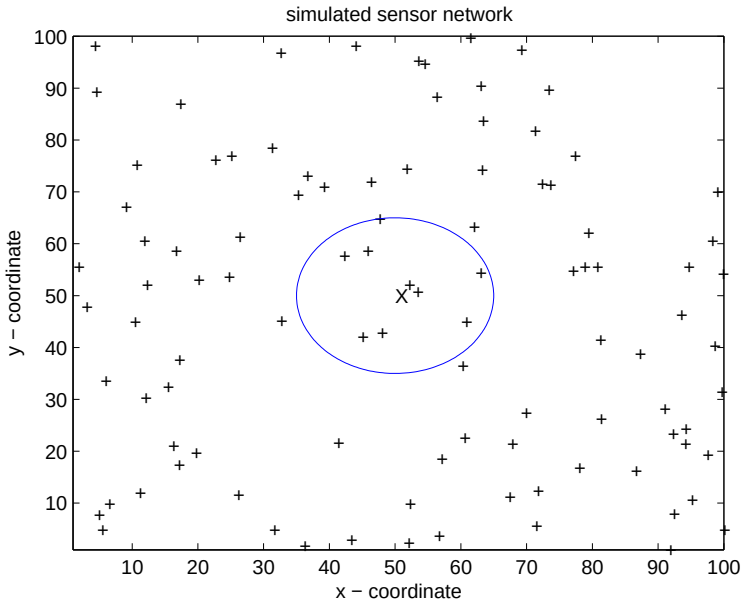

Figure 1: RMSE of different algorithms

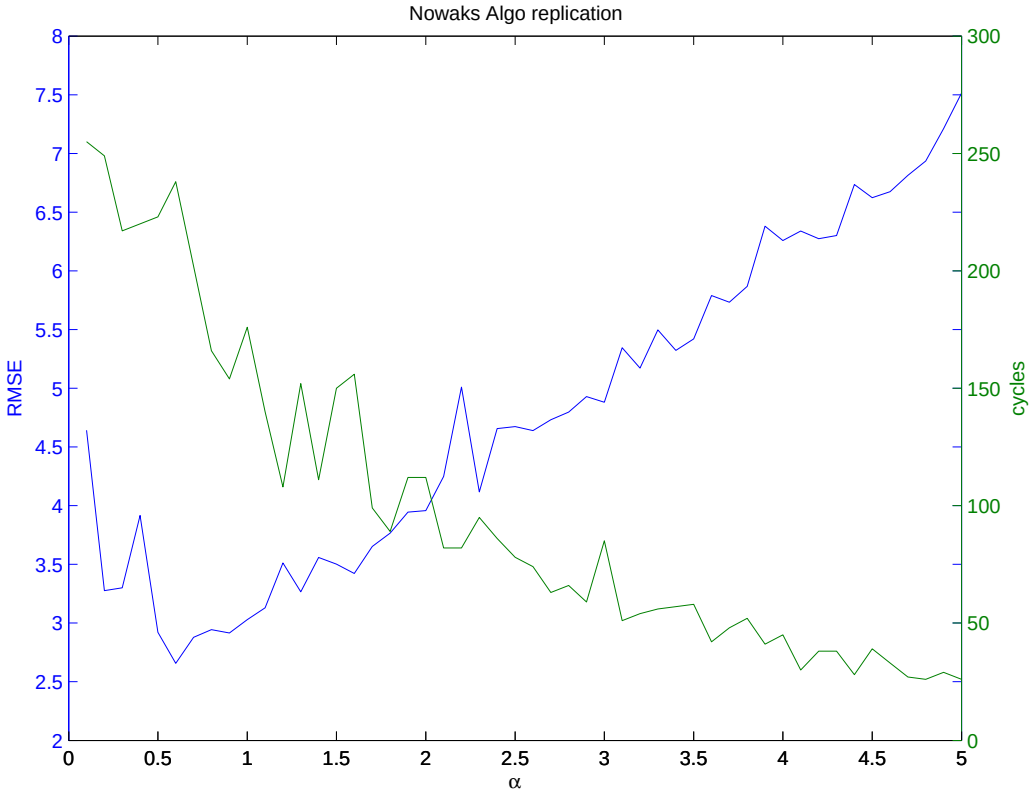

Figure 2: Nowak's algorithm

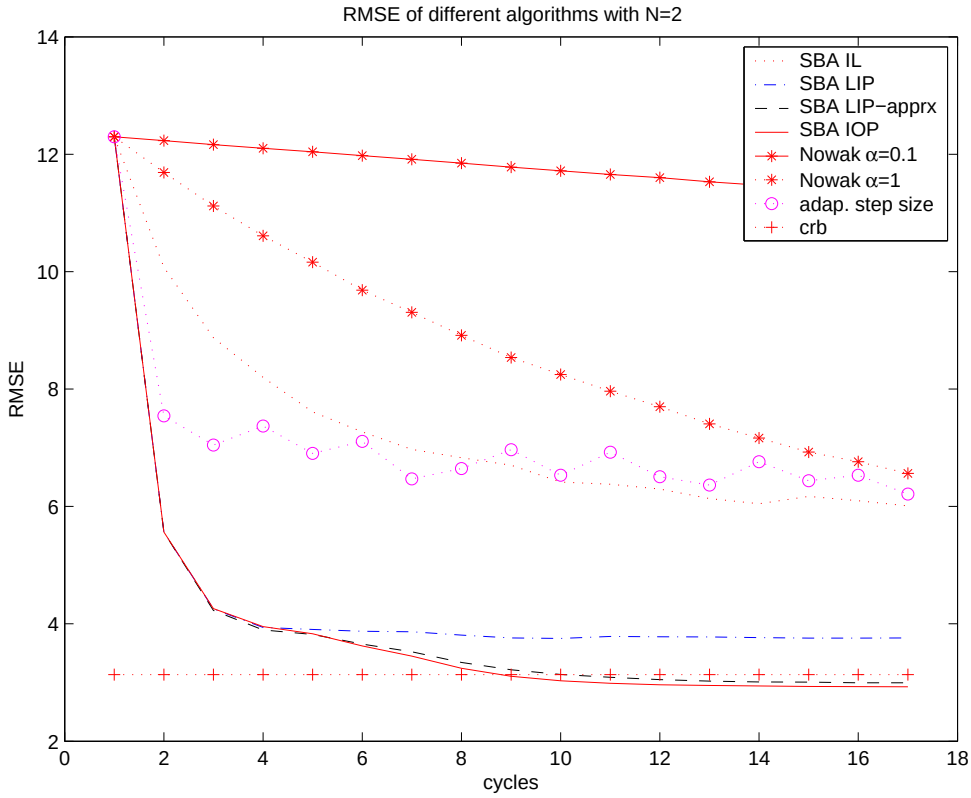

Figure 3: RMSE of different algorithms: $\gamma = 0$ for SBA $crb = E(\sqrt{((\theta - \hat{(\theta)}))^T(\theta - \hat{(\theta)})})$

scoring method and Cramer-Rao lower bound. In the Bayesian frame-work, we model the sensor network Hierarchical non-linear Bayes model and consider some advantages of such a modeling mechanism.

A.1 Fisher scoring method

The log-likelihood function is given in Eqn. (4.1). The Fisher method of scoring computes the estimates according to the following equation:

$$\hat{\boldsymbol{\theta}}_j^k = \hat{\boldsymbol{\theta}}_{j-1}^k - I(\boldsymbol{\theta})^{-1} \frac{\partial}{\partial \boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \boldsymbol{y}^k)_{|\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{j-1}^k} \tag{A.1}$$

where $I(\boldsymbol{\theta})$ is the Fisher information matrix given by

$$I(\boldsymbol{\theta}) = -E\left[\frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^H} \ell(\boldsymbol{\theta}; \boldsymbol{y}^k))\right] \tag{A.2}$$

and $\frac{\partial}{\partial \boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \boldsymbol{y}^k)$ is the score function. We can obtain the quantiles after a straight forward algebraic manipulation:

$$I(\boldsymbol{\theta}) = \frac{2\bar{\gamma}_k a^2 kN}{\sigma^2} \sum_{j=1}^{M} \frac{1}{r_j^8} \begin{bmatrix} (\theta_1 - x_{1j})^2 & (\theta_1 - x_{1j})(\theta_2 - x_{2j}) \\ (\theta_1 - x_{1j})(\theta_2 - x_{2j}) & (\theta_2 - x_{2j})^2 \end{bmatrix} \tag{A.3}$$

and the scoring function is given as:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \boldsymbol{y}^k) = \frac{2akN}{\sigma^2} \sum_{j=1}^{M} \frac{\bar{y}_{j,\gamma}\, r_j^2 - \bar{\gamma}_k a}{r_j^6} (\boldsymbol{\theta} - \boldsymbol{x}_j) \tag{A.4}$$

where $\bar{y}_j^k$ and $\bar{\gamma}_k$are defined as in Eqn. (4.21).

Cramer-Rao lower bound is given by the inverse of the Fisher information matrix.

A.2 Hierarchical non-linear Bayes model

The location estimation problem under consideration is interesting in the sense that it is impossible to estimate the source location based on the observations by a single sensor. In this respect, the sensor information can not be considered independent. However, in the global prior method, we have implicitly assumed independence in the samples across the sensors and the results appear promising. This might seem true if the sensors are large in number. In the local prior method, we have assumed lack of independence across the sensors. In fact, we have considered to form linearly dependent pool. Hence, we fused the estimate given by the previous sensor with the estimate given by the current sensor. This method is also promising. These are two extreme cases of pooling the information given by different groups of observations. Hierarchical modeling is natural way to weigh the information across the sensors.

The following model is considered

$$y_j^k(i) \sim N(s_j(\theta), \sigma^2) \tag{A.5a}$$

$$s_j(\theta) \sim N(\boldsymbol{\theta}, \Sigma) \tag{A.5b}$$

with some priors on $\boldsymbol{\theta}$ and Σ We are using winBUGS to compute the posterior density of $\boldsymbol{\theta}$.

WinBUGS model is given below:

```
model {
    for (j in 1:J) {
        for (i in 1:N){y[j, i] ~- dnorm(mu[j], tauC)}
        mu[j] ← A/(pow(x1[j]-theta[j,1],2)+pow(x2[j]-theta[j,2],2))
        theta[j, 1:2] ~ dmnorm(phi[1:2], tau[1:2, 1:2])}
        phi[1:2] ~ dmnorm(mean[1:2], prec[1:2, 1:2])
        tau[1:2, 1:2] ~ dwish(R[1:2, 1:2], 2)
        sigma2[1:2, 1:2] ← inverse(tau[1:2, 1:2])
        for (i in 1 : 2) {sigma[i] ¡- sqrt(sigma2[i, i])      }
}
```

B Extensions

In this Section, we generalize the model to include multiple sources both with known and unknown source energies. The model is given as:

$$s_j = \sum_{t=1}^{T} \frac{a_t}{\|\boldsymbol{\theta}_t - \boldsymbol{x}_j\|^2} \tag{B.1a}$$

$$y_j^{\kappa}(i) = s_j + e_j^{\kappa}(i) \tag{B.1b}$$

where $\boldsymbol{\theta}_t$ is the location of the t^{th} source and T is the total number of sources. All other parameters are as defined in (2.1). In the next Section, we consider the case where a_t's, the sources energies, are known.

B.1 Known source energies

This is a classical problem in parameter estimation of super imposed signals. In [9], the authors apply the EM algorithm to estimate the individual components (E-step) and then maximize the conditional likelihood (M-step) to obtain the parameter estimates of individual components in the the superimposed signal model. In the present context, if we can estimate the individual component, we can use the algorithms developed in the earlier sections to obtain the individual parameter estimates. The outline of the algorithm is as follows:

E-step: for t = 1,2,...,T

$$\hat{y}_{j,t}^{k} = s_{j,t}(\hat{\boldsymbol{\theta}}_{j-1,t}^{k}) + \xi_t \left[y_j^k - \sum_{l=1}^{T} s_{j,l}(\hat{\boldsymbol{\theta}}_{j-1,l}^{k}) \right] \tag{B.2}$$

where ξ_t are chosen such that $\sum_{t=1}^{T} \xi_t = 1$.

M-step: for t = 1,2,...,T

$$\hat{\boldsymbol{\theta}}_{j,t}^{k} = \min_{\boldsymbol{\theta}_t} \sum_{\kappa=1}^{k} \sum_{i=1}^{N} \gamma^{\kappa-k} \left(\hat{y}_{j,t}^{\kappa}(i) - s_{j,t}(\boldsymbol{\theta}_t) \right)^2 \tag{B.3}$$

We can use the method described in Section 4 to solve the above minimization. In the next Section, we consider an even more general problem where the source energies are assumed unknown.

B.2 Unknown source energies

If the source energies are unknown, then we can not reduce the dimensionality of the problem. Instead, we use the extended Kalman filter (EKF) method proposed and analyzed in [4]. The EKF can be applied to minimize functions of the form

$$\boldsymbol{g} = \sum_{j=1}^{M} \|\boldsymbol{g}_j(\boldsymbol{\psi})\|^2 \tag{B.4}$$

In the present context, $\boldsymbol{g}_j$ is given by:

$$\boldsymbol{g}_j(\boldsymbol{\psi}) = \boldsymbol{y}_j^k - \boldsymbol{s}_j^k(\boldsymbol{\psi}) \tag{B.5}$$

where $\boldsymbol{y}_j^k$ and $\boldsymbol{s}_j^k$ are $N \times 1$ vectors and $\boldsymbol{\psi}$ is a $3T \times 1$ vector of parameters (T source energies and $2T$ source location parameters). From here onwards, explicit dependence of $\boldsymbol{s}$ and $\boldsymbol{y}$ on j and k will not be shown. The updation equation for the parameters is given as [4, Eqn. 21]:

$$\hat{\boldsymbol{\psi}}_j = \hat{\boldsymbol{\psi}}_{j-1} - \boldsymbol{H}_j^{-1} \nabla \boldsymbol{g}_j(\hat{\boldsymbol{\psi}}_{j-1}) \boldsymbol{g}_j(\hat{\boldsymbol{\psi}}_{j-1}) \tag{B.6a}$$

$$\boldsymbol{H}_j = \lambda \boldsymbol{H}_{j-1} + \nabla \boldsymbol{g}_j(\hat{\boldsymbol{\psi}}_{j-1}) \nabla^T \boldsymbol{g}_j(\hat{\boldsymbol{\psi}}_{j-1}) \tag{B.6b}$$

However, we note that the source energies enter linearly into the model equation. Hence, it is possible reduce the dimensionality [10]. We partition the parameters as: $\boldsymbol{a}$ (a $T \times 1$ vector of source energies) and $\boldsymbol{\varphi}$ (a $2T \times 1$ vector of source locations). We can rewrite the model as:

$$\boldsymbol{s} = \boldsymbol{D}(\boldsymbol{\varphi})\boldsymbol{a} \tag{B.7a}$$

$$\boldsymbol{D}(\boldsymbol{\varphi}) = \boldsymbol{J}\boldsymbol{d}^T(\boldsymbol{\varphi}) \tag{B.7b}$$

where $\boldsymbol{J}$ is a $N \times 1$ vector of ones, $\boldsymbol{d}$ is $T \times 1$ vector with $d(i) = \frac{1}{\|\boldsymbol{\theta}_i - \boldsymbol{x}\|^2}$ as its elements. Then the BLUE estimate of $\boldsymbol{a}$ is given as:

$$\boldsymbol{a}(\boldsymbol{\varphi}) = (\boldsymbol{D}^T(\boldsymbol{\varphi})\boldsymbol{D}(\boldsymbol{\varphi}))^{-1}\boldsymbol{D}^T(\boldsymbol{\varphi})\boldsymbol{y} \tag{B.8}$$

Also, define the projection matrix as:

$$\boldsymbol{P}_d(\boldsymbol{\varphi}) = \boldsymbol{D}(\boldsymbol{\varphi})(\boldsymbol{D}^T(\boldsymbol{\varphi})\boldsymbol{D}(\boldsymbol{\varphi}))^{-1}\boldsymbol{D}^T(\boldsymbol{\varphi}) \tag{B.9}$$

Then the nested EKF updation equations can be written by performing block cholskey decomposition as:

$$\hat{\boldsymbol{\varphi}}_j = \hat{\boldsymbol{\varphi}}_{j-1} - (\boldsymbol{H}_{j,22} - \boldsymbol{H}_{j,21}\boldsymbol{H}_{j,11}^{-1}\boldsymbol{H}_{j,12})^{-1} \nabla_{\boldsymbol{\varphi}} \boldsymbol{g}_j(\hat{\boldsymbol{\varphi}}_{j-1}) \boldsymbol{P}_d^{\perp}(\hat{\boldsymbol{\varphi}}_{j-1}) \boldsymbol{y} \tag{B.10}$$

where

$$\boldsymbol{H}_j = \begin{pmatrix} \boldsymbol{H}_{j,11} & \boldsymbol{H}_{j,12} \\ \boldsymbol{H}_{j,12}^T & \boldsymbol{H}_{j,22} \end{pmatrix} \tag{B.11a}$$

$$\boldsymbol{H}_{j,11} = \lambda \boldsymbol{H}_{j-1,11} + \boldsymbol{D}^T(\hat{\boldsymbol{\varphi}}_{j-1})\boldsymbol{D}(\hat{\boldsymbol{\varphi}}_{j-1}) \tag{B.11b}$$

$$\boldsymbol{H}_{j,12} = \lambda \boldsymbol{H}_{j-1,12} + \boldsymbol{D}^T(\hat{\boldsymbol{\varphi}}_{j-1}) \nabla_{\boldsymbol{\varphi}} \boldsymbol{g}_j(\hat{\boldsymbol{\varphi}}_{j-1}) \tag{B.11c}$$

$$\boldsymbol{H}_{j,22} = \lambda \boldsymbol{H}_{j-1,22} + \nabla_{\boldsymbol{\varphi}} \boldsymbol{g}_j(\hat{\boldsymbol{\varphi}}_{j-1}) \nabla_{\boldsymbol{\varphi}}^T \boldsymbol{g}_j(\hat{\boldsymbol{\varphi}}_{j-1}) \tag{B.11d}$$

We can use *matrix inversion lemma* to compute $\boldsymbol{H}_{j,11}^{-1}$. This completes the derivation of iterated separable EKF algorithm to be used in the estimation of multiple source parameters in distributed mode.

Acknowledgements

The authors wishes to thank Dr. Aleksander Dogandžić for many suggestions on this topic.

References

[1] Michel Rabbat and Robert Nowak, "Decentralized source localization and tracking," in *ICASSP'04*. 2004, vol. 3, pp. 921–924, IEEE.

[2] Xiaohong Sheng and Yu-Hen Hu, "Energy based acoustic source localization," in *The 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03)*, April 2003, pp. 285–300.

[3] Angelia Nedic and Dimitri P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM J. on Optimization*, vol. 12, pp. 109–138, 2001.

[4] Dimitri P. Bertsekas, "Incremental least squares methods and the extend kalman filter," *SIAM J. on Optimization*, vol. 6, pp. 1–15, 1996.

[5] Olena Punska, *Bayesian Methods for Multi-Sensor fusion*, M. Phil, Univ. of Cambridge, August 1999, url: www-sigproc.eng.cam.ac.uk/ op205/mphil.pdf.

[6] J. C. Naylor and A. F. M. Smith, "Applications of a method for the efficient computation of posterior distributions," *Applied Statistics*, vol. 31, 1982.

[7] Sophia Rabe-Hesketh and Anders Skrondal, "Reliable estimation of generalized linear mixed models using adaptive quadrature," *The Stata Journal*, vol. 2, no. 1, pp. 1–21, 2002.

[8] M̃. Figueiredo J. Leitao and A. K. Jain, "On fitting mixture models," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, vol. 1654, pp. 54–69. Springer-Verlag, 1999.

[9] Meir Feder and Ehud Weinstein, "Parameter estimation of superimposed signals using the EM algorithm," *IEEE Trans. on ASSP*, vol. 36, no. 4, pp. 477–489, April 1988.

[10] Gordon K Smith, "Partitioned algorithms for maximum likelihood and other nonlinear estimation," *Statistics and Computing*, vol. 6, pp. 201–216, 1996.