COLLABORATIVE DIFFUSIVE SOURCE LOCALIZATION IN WIRELESS SENSOR NETWORKS

Sabina Zejnilovic,[1] *João Gomes*[2] *and Bruno Sinopoli*[1]

[1] Carnegie Mellon University, Department of Electrical and Computer Engineering, Pittsburgh, PA
sabinaz@cmu.edu, brunos@ece.cmu.edu

[2] Institute for Systems and Robotics – Instituto Superior Técnico (ISR/IST), Technical University of Lisbon (UTL), Portugal
jpg@isr.ist.utl.pt

ABSTRACT

We propose a collaborative, energy efficient method for diffusive source localization in wireless sensor networks. The algorithm is based on distributed and iterative maximum-likelihood (ML) estimation, which is very sensitive to initialization. As a part of the proposed method we present an approach for obtaining a "good enough" initial value for the ML recursion based on infinite time approximation and semidefinite programming. We also present an approach for determining the sensor node that initiates the estimation process. To improve the convergence rate of the algorithm, we consider the case where selected nodes collaborate with their neighbors. Simulation results are used to characterize the performance and energy efficiency of the algorithm. We also illustrate estimation accuracy/energy consumption trade-off by varying the communication radius of sensor nodes.

Index Terms— Diffusive source localization, distributed estimation, wireless sensor networks.

1. INTRODUCTION

Wireless sensor networks (WSN) have found application in various fields due to their collaborative sensing power, adaptability, low cost and rapid deployment [1]. Application areas are diverse and include, among others, environmental monitoring, pollution control and homeland security. A common task in the above-mentioned applications is the localization of a source of emissions (e.g., a source of pollution) from measurements of concentrations of its diffusive field taken at several locations. A standard approach for determining the location of a diffusive source involves transmitting all measurements from all sensor nodes (SNs) to a central location where processing takes place [2]. Although the source location thus computed is based on all available information, this approach has several disadvantages, especially prominent in large scale networks: abundant (and costly) communication, large processing power required at the central node, and increased vulnerability of the network to central node failure. In order to decrease the communication cost of estimation, a distributed algorithm for diffusive source localization proposed in [3] combines distributed ML estimation with information-based sensor node selection. In [4], to preserve energy resources, distributed detection and estimation of a diffusive source are performed in two stages and only local estimates are transmitted to the central location.

We propose a new, energy-efficient collaborative approach for diffusive source localization. A ML estimate of the source location is obtained through the distributed Gauss-Newton (GN) optimization method as in [3], but our approach additionally includes: a method for selection of the SN that initiates the estimation process, a method for obtaining a better starting point for the recursion, neighborhood based collaboration and addition of line-search method in local iterations. The estimate is updated recursively, with each recursion performed by a selected group of neighboring SNs.

In the next section the measurement model and the corresponding distributed ML estimation problem are discussed. Algorithm overview is given in Section 3.2. Section 4 contains simulation results and discussion, while conclusions are given in Section 5.

2. DISTRIBUTED ML ESTIMATION

A continuous diffusive point source, located at position $\boldsymbol{\theta}$, unknown to the sensors, starts emitting at time t_0. Each node i, located at position $\boldsymbol{r}_i$, measures concentration $c(\boldsymbol{r}_i, t_j)$ at time instants t_j, $j = 1, \ldots, N$. To model airborne diffusion above the ground, we assume the environment is a semi-infinite medium with impermeable boundary, where the ground represents the boundary [2]. Then the concentration of the substance at a position $\boldsymbol{r}_i$ at time t_j is given as

This research was partially supported by Fundação para a Ciência e a Tecnologia (FCT) through projects PEst-OE/EEI/LA0009/2011, PTDC/EEA-CRO/104243/2008, and a PhD grant from the Carnegie Mellon—Portugal program.

$$c\left(\boldsymbol{r}_i, t_j\right) = \frac{\mu}{2\pi\kappa\|\boldsymbol{r}_i - \boldsymbol{\theta}\|}\operatorname{erfc}\left(\frac{\|\boldsymbol{r}_i - \boldsymbol{\theta}\|}{2\sqrt{\kappa\left(t_j - t_0\right)}}\right), \quad (1)$$

where μ is the release rate of substance from the source and κ is the medium diffusivity. The statistical model for measurements y_{ij} for each node i at time instant t_j becomes

$$y_{ij} = a_{ij}\left(\boldsymbol{\theta}\right) + e_{ij},$$

where $a_{ij} = c\left(\boldsymbol{r}_i, t_j\right) + b$, e_{ij} is the SN's Gaussian measurement noise, independent from SN to SN and correlated in time, and b is the bias, which models the SN's response to the presence of a foreign substance. The bias can be estimated prior to localization, by averaging the measurements when there is no source present. Now, putting together all the measurements of sensor node i, a vector form of the measurement model is obtained

$$\boldsymbol{y}_i = \boldsymbol{a}_i(\boldsymbol{\theta}) + \boldsymbol{e}_i, \quad \boldsymbol{e}_i \sim \mathcal{N}\left(0, \Sigma_i\right), \quad i = 1, \ldots M. \quad (2)$$

The maximum-likelihood estimate (MLE) for model (2) corresponds to the nonlinear least-squares estimate

$$\hat{\boldsymbol{\theta}} = \arg\min_{\theta}\left\{\sum_{i=1}^{M}\left(\boldsymbol{y}_i - \boldsymbol{a}_i(\boldsymbol{\theta})\right)^T \Sigma_i^{-1}\left(\boldsymbol{y}_i - \boldsymbol{a}_i(\boldsymbol{\theta})\right)\right\}. \quad (3)$$

Due to the nonlinear dependence on $\boldsymbol{\theta}$, a closed-form expression for the MLE cannot be found, and the estimate must be obtained through an iterative approach, such as the Gauss-Newton method. The location of the diffusive source can be obtained iteratively through the distributed Gauss-Newton method

$$\begin{aligned}\hat{\boldsymbol{\theta}}_i &= \hat{\boldsymbol{\theta}}_{i-1} + \Gamma_i^{-1} J_i^T(\hat{\boldsymbol{\theta}}_{i-1})\Sigma_i^{-1}\left(\boldsymbol{y}_i - \boldsymbol{a}_i(\hat{\boldsymbol{\theta}}_{i-1})\right)\\ \Gamma_i &= \Gamma_{i-1} + J_i^T(\hat{\boldsymbol{\theta}}_{i-1})\Sigma_i^{-1} J_i(\hat{\boldsymbol{\theta}}_{i-1}),\end{aligned} \quad (4)$$

where $\Gamma_0 = 0$ and

$$J_i(\boldsymbol{\theta}) = \frac{\partial \boldsymbol{a}_i(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}$$

is the $N \times P$ Jacobian matrix, where P represents the dimension of $\boldsymbol{\theta}$ [3]. Each iteration i of the distributed GN algorithm updates the estimate $\hat{\boldsymbol{\theta}}_{i-1}$ to $\hat{\boldsymbol{\theta}}_i$ and requires the values from the previous iteration: $\hat{\boldsymbol{\theta}}_{i-1}$ and Γ_{i-1} and the locally known values y_i, a_i and Σ_i. This means that the process of sequentially updating the estimate of the source location can be distributed among the SNs and only the values $\hat{\boldsymbol{\theta}}_{i-1}$ and Γ_{i-1} need to be passed from SN to SN. In terms of energy consumption, communication is far more costly than computation, so the update (4) can be modified by allowing multiple local iterations, starting with $\phi_0 = \hat{\boldsymbol{\theta}}_{i-1}$ and $\Upsilon_0 = \Gamma_{i-1}$

$$\begin{aligned}\phi_j &= \phi_{j-1} + \Upsilon_j^{-1} J_i^T(\phi_{j-1})\Sigma_i^{-1}\left(\boldsymbol{y}_i - \boldsymbol{a}_i(\phi_{j-1})\right)\\ \Upsilon_j &= \Upsilon_{j-1} + J_i^T(\phi_{j-1})\Sigma_i^{-1} J_i(\phi_{j-1}), \quad j = 1, 2, \ldots\end{aligned}$$

Therefore, one SN repeats the update step multiple times, but each time using the same measurements. After this recursion converges, the values ϕ_j and Υ_j become the new estimates $\hat{\boldsymbol{\theta}}_i$ and Γ_i [3].

The performance of the centralized GN method is improved when line search is performed to ensure that there is sufficient decrease of the cost function in each iteration [5]. Similarly, we modify the recursions of distributed GN (4) by placing a weight α_i on the update at each iteration i. This weight α_i is calculated through standard backtracking line search, such that sufficient improvement in the local cost function at each iteration i is achieved [5]. The resulting update equation has the following form

$$\hat{\boldsymbol{\theta}}_i = \hat{\boldsymbol{\theta}}_{i-1} + \alpha_i \Gamma_i^{-1} J_i^T(\hat{\boldsymbol{\theta}}_{i-1})\Sigma_i^{-1}\left(\boldsymbol{y}_i - \boldsymbol{a}_i(\hat{\boldsymbol{\theta}}_{i-1})\right).$$

Although this increases the computation load of each iteration, overall the algorithm becomes more energy efficient, as each iteration improves the cost function more, requiring less total number of iterations, and consequently, the total number of costly transmission between the nodes is decreased.

3. ALGORITHM OVERVIEW

The distributed algorithm for diffusive source localization can be summarized in the following steps:

i. A node is selected to start the algorithm through Neyman-Pearson detection.

ii. Initial estimate is obtained by the starting node as a solution of SDP optimization program.

iii. Local update (4) is performed by the starting node to derive a new estimate of the source position.

iv. The quality of the updated estimate is evaluated to decide whether the algorithm should terminate or proceed with the next step.

v. Another node is selected to continue the recursion starting from step iii.

3.1. Starting node and initial estimate

Each SN has a clock that ticks at the times of a rate μ Poisson process. Once the clock of a SN ticks, it becomes a selected SN and performs a Neyman-Pearson (NP) detection test. Based only on its own measurements, it decides which one of the two models is more likely, one where the source is located at some relatively small distance from the SN, and the other where it is further away. If the SN detects that it is "close enough" to the source, it queries for measurements its neighboring nodes, i.e., those located within its communication range, and calculates the initial estimate. Even though energy is spent in querying and gathering the measurements

from the neighborhood, the initial estimate is based on more information, when compared with using only the SN's own measurements. This in turn results in fewer iterations of the algorithm and estimation remains energy efficient. If, on the other hand, a SN concludes, based on the NP test, that it has low signal to noise ratio and is far away from the source, it does nothing. In this way, SNs are chosen such that the whole area is monitored uniformly and only informative enough SNs initiate estimation.

Once a node is chosen to start the estimation process, it needs to obtain an initial estimate. The function which is minimized in order to obtain the MLE in (3) is highly non-convex in the source coordinates. The performance of GN heavily depends on the value with which the recursion (4) is initialized. If the initial value is not "close enough" to the true position of the source, the estimate may drift further away from the true value with each new iteration, even though more and more measurements are used to calculate the estimate.

We propose a method for initialization motivated by approaches used in localization of wireless radio sources based on received signal strength (RSS) [6]. If the concentration is measured for a long time, such that $t_j - t_0 \gg \|\boldsymbol{r}_i - \boldsymbol{\theta}\|^2/(4\kappa)$, the measurement model (1) takes on a simplified form, as the value of the complementary error function erfc becomes approximately one [2]

$$y_{ij} = \frac{\mu}{2\pi\kappa\|\boldsymbol{r}_i - \boldsymbol{\theta}\|} + b + e_{ij}. \tag{5}$$

Using the infinite time approximation model (5), we can obtain a crude estimate of the source position as the solution of the following optimization problem

$$\hat{\boldsymbol{\theta}}_0 = \arg\min_{\hat{\boldsymbol{\theta}}_0,\alpha} \sum_{i,j} |\,(y_{ij} - b)^2 \|\boldsymbol{r}_i - \hat{\boldsymbol{\theta}}_0\|^2 - \alpha|, \tag{6}$$

where $\alpha = (\mu/(2\pi\kappa))^2$. However, in practical scenarios where the location of the source needs to be estimated as soon as the source is detected, the time when the concentration is measured is not large enough to satisfy the infinite time approximation. To compensate for the approximation in our model in this case, we treat α as an additional optimization variable, even if the release rate μ and medium diffusivity κ are both known. In this way, deviations from the model (5) due to finite time are partly captured by variable α, still allowing this model simplification to be used. The cost function in (6) is not convex, but the problem can be relaxed to a semidefinite program (SDP) by introducing an auxiliary variable y such that $y = \boldsymbol{\theta}^T\boldsymbol{\theta}$ [6]. After relaxing the equality to inequality constraint $y \geq \boldsymbol{\theta}^T\boldsymbol{\theta}$, the problem becomes an SDP

$$\begin{array}{ll} \underset{\boldsymbol{\theta},\alpha,t_{ij},y}{\text{minimize}} & \sum_{i,j} t_{ij} \\ \text{subject to} & -t_{ij} \leq (y_{ij} - b)^2\left(y - 2\boldsymbol{r}_i^T\boldsymbol{\theta} + \|\boldsymbol{r}_i\|^2\right) - \alpha \leq t_{ij} \\ & y \geq \boldsymbol{\theta}^T\boldsymbol{\theta}. \end{array} \tag{7}$$

3.2. Sensor node collaboration

Once an initial value is obtained by solving the SDP (7), the starting node performs the first iteration of (4), using its own measurements and those from its neighbors, and obtains $\hat{\boldsymbol{\theta}}_1$. This value, together with the value Γ_1, needs to be passed on to another SN, so that the estimate can be further updated. To improve the estimate faster and consume less energy, the starting node needs to select an informative SN. Decision which SN is the most informative one cannot be based on explicit knowledge of the measurements residing at each SN, as that would require transmission of all the measurements.

As a measure of the informativeness of the node, the trace or determinant of the Fisher information matrix (FIM) can be used. The FIM is the inverse of the Cramér-Rao bound (CRB), the lower bound on the variance of any unbiased estimator. The FIM is calculated by averaging over all possible measurements, thus it does not require knowledge of their particular values. Additional incentive to use the FIM in the selection criterion comes from the fact that it can be calculated without increasing neither the number of communications nor the computational complexity of the distributed GN algorithm [3]. As measurements from different sensors are independent, the FIM can be updated recursively as

$$F_i(\boldsymbol{\theta}) = F_{i-1}(\boldsymbol{\theta}) + J_i^T(\boldsymbol{\theta})\Sigma_i^{-1}J_i(\boldsymbol{\theta}) \tag{8}$$

where $F_i(\boldsymbol{\theta})$ is the FIM of nonlinear parameter $\boldsymbol{\theta}$ after incorporating measurements $\{y_1, \ldots, y_i\}$. Let node i be the SN that was selected as the most informative SN after $i-1$ iterations of the algorithm. For node i to calculate the updated FIM, based on its own measurements, it needs to know the previous value, $F_{i-1}(\boldsymbol{\theta})$. However, this is the value of Γ_{i-1} that is already passed on to node i. Node i would also need to know the true value of $\boldsymbol{\theta}$ to evaluate (8), but instead it uses the currently available estimate $\hat{\boldsymbol{\theta}}_i$ as an approximation of (8).

As the communication cost also needs to be taken into consideration when choosing the most informative SN, only a subset of SNs is considered in the selection process. The energy spent communicating depends on the number of nodes transmitting and receiving. By constraining the next selected node to be a neighbor of the current node, the communication cost is kept at one transmission and one reception. Using the trace of the approximate FIM as the information measure, node i selects the best node $i+1$ as [3]

$$i+1 = \arg\max_{l\in\mathcal{S}} \text{Trace}\left\{J_l^T(\hat{\boldsymbol{\theta}}_i)\Sigma_l^{-1}J_l(\hat{\boldsymbol{\theta}}_i)\right\},$$

where $\mathcal{S}$ is the set of all the neighbors of current node i. The trace of the CRB approximation from equation (8) is also used as termination criterium, to determine after each recursion whether another SN should be selected to additionally improve the estimate, or if the algorithm should terminate.

Once a node is selected as the most informative node, it queries its neighbors for their measurements and computes

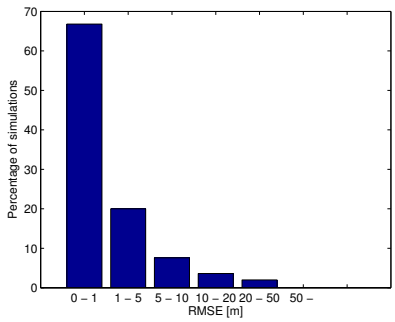
the new estimate using all the available measurements. The motivation for this neighborhood-based collaboration is again the increase in estimation accuracy, with the acceptable trade-off in increased energy consumption. With this collaboration, the proposed approach differs from typical distributed methods where SNs only process their own measurements. In our approach, estimate $\hat{\boldsymbol{\theta}}_i$ is calculated using the measurements of node i and its neighbors, one of which is node $i+1$ itself. In most practical WSN implementations, where nodes are randomly distributed over an area which is to be monitored, SNs have overlapping neighborhoods, meaning nodes i and $i+1$ might have common neighbors. To avoid transmissions of redundant information, only those neighboring nodes that have not been queried by node i respond to the query of node $i+1$ for measurements. Each iteration involves one transmission of previous estimates, one query for measurements and transmissions from the neighbors responding to the query, so the total transmission cost scales linearly with the number of iterations. Denoting with d_{avg} the average number of neighbors of a SN, with p the probability that two neighbors of one SN are neighbors themselves, which corresponds to the overall clustering coefficient of the network, and with C_k the expected number of transmissions over k iterations, we have

$$C_k = (d_{\text{avg}}(1-p)+2)\,k + \text{constant}. \tag{9}$$

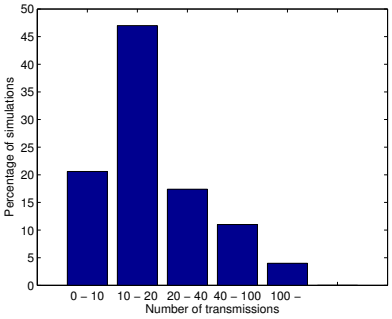
The constant reflects the fact that in the first iteration, unlike in the subsequent ones, the transmissions include one query from the starting node and responses from all the neighbors.

4. SIMULATION RESULTS

To evaluate the performance and energy consumption of the proposed approach for distributed diffusive source localization, a simulation study was performed. A sensor network with 100 nodes is set up to have measurements of the form (2). The source and SNs are placed uniformly at random in a square area 100m $\times$ 100m. The communication radius is set at 15 m, each SN has at least one neighbor and the resulting network is connected. Each SN makes 10 measurements, with sampling interval of 10s, starting 100s after the source became active. The parameters are set to $\mathrm{b} = 10^{-5}\,\mathrm{g/m^3}$, $\sigma = 6\times10^{-6}\,\mathrm{g/m^3}$, $\mu = 1\,\mathrm{g/s}$, $\kappa = 20\,\mathrm{m^2/s}$ and $t_0 = 0$ s. Using NP detection, SNs test if they are less than 25m away from the source, with probability of false alarm set to 0.1. For each simulation, we record the final root mean-squared estimation error (RMSE) and the total number of transmissions incurred during the estimation process. As the communication cost dominates over all other costs for SN, the number of transmissions gives us a metric to evaluate the energy consumption of the algorithm. To show the algorithm's performance for different scenarios in which the source could be located anywhere within the monitored area, we run 500 simulations, each with different source and SNs placement and noise realizations. For the same topologies and noise realization we also simulate the algorithm presented in [3]. Since this algorithm does not incorporate a method for initialization, we use the center of the monitored area as the initial estimate for the source position.



(a) Histogram of RMSE

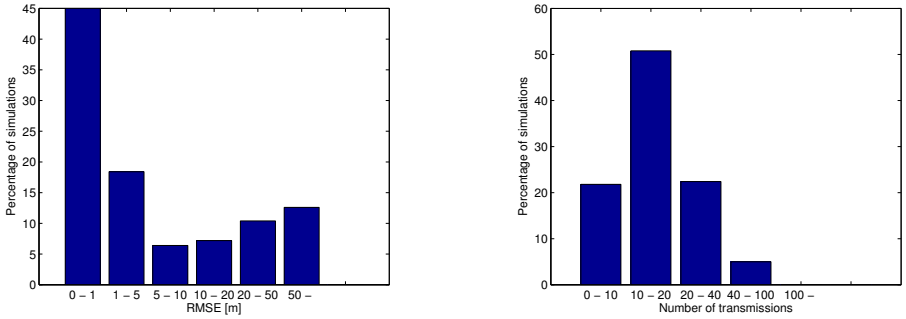


(b) Histogram of the number of transmissions

Fig. 1: Performance and energy consumption of the proposed algorithm

Figure 1a shows that in more than 65% of the simulations, for our aproach, the estimated source location was at most 1m away from the true position of the source. In comparison, Figure 2a shows that for the algorithm proposed in [3] this percentage is 45. Moreover, the percentage of simulations where RMSE was larger than 10m for our algorithm is less than 5%, whereas for the algorithm of [3] it is around 30%. To get this improvement in accuracy, our approach incurs higher number of transmissions, as SNs collaborate with their neighbors for each iteration of the distributed GN algorithm. However, as seen from Figures 1b and 2b, this increase in the number of transmissions is slight. In less than 5% of the simulations the number of transmission was higher than 100, which is the number of transmissions which would be incurred by any centralized approach where all 100 nodes need to transmit their measurements to the central location.

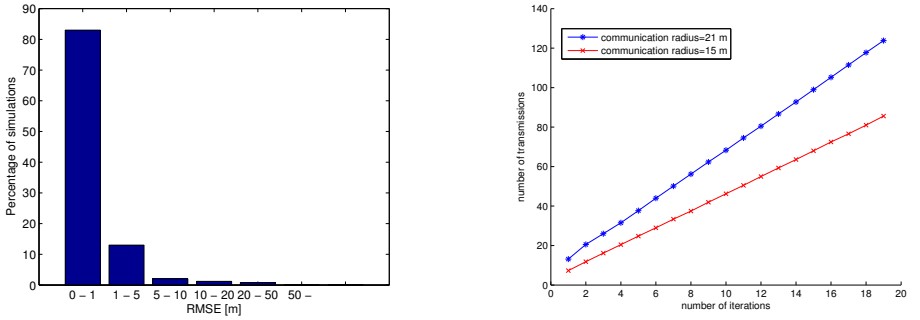
Next, we explore the effect of the communication radius on the performance and energy consumption of the algorithm. For the same source and SN placements and noise realization, simulations were repeated, but with the communication radius increased to 21 m. This results in the average number of neighbors almost doubling, from 6 to 11.6, which mod-



(a) Histogram of RMSE

(b) Histogram of the number of transmissions

Fig. 2: Performance and energy consumption of the algorithm proposed in [3]



(a) Histogram of RMSE

(b) Number of transmissions through iterations

Fig. 3: The effect of increased communication radius on the proposed algorithm

els networks with denser SN deployments. Figure 3a now shows increased estimation accuracy, while Figure 3b shows increased slope of the seemingly linear trend that represents how the number of transmissions varies with the number of iterations of the algorithm, modeled by (9). Higher radius increases the rate at which the number of transmissions increases with iteration number, but this increased energy consumption results in smaller RMSE. Hence, varying the density of SNs placement produces a tradeoff between estimation accuracy and energy consumption.

5. CONCLUSIONS

We presented a method for collaborative localization of a diffusive source, where a maximum likelihood estimate is iteratively calculated across groups of neighboring nodes. A method for obtaining the initial value for the recursive estimation was presented, as well as the method for selection of an informative node to start the estimation. Simulation results illustrated that the proposed approach gives higher estimation accuracy than a similar published approach with only a slight increase in energy consumption. Also, we analyzed the tradeoff in estimation accuracy and energy cost that occurs when the communication radius of the sensor nodes varies.

6. REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks (Elsevier) Journal*, vol. 38, no. 4, pp. 393–422, Mar 2002.

[2] A. Nehorai, B. Porat, and E. Paldi, "Detection and localization of vapor-emitting sources," *IEEE Transactions on Signal Processing*, vol. 43, no. 1, pp. 243–253, Jan 1995.

[3] T. Zhao and A. Nehorai, "Information-driven distributed maximum likelihood estimation based on Gauss Newton method in wireless sensor networks," *IEEE Trans on Signal Processing*, vol. 55, no. 9, pp. 4669–4682, Sep 2007.

[4] S. Aldalahmeh and M. Ghogho, "Robust distributed detection, localization and estimation of a diffusive target in clustered wirless sensor networks," *ICASSP*, 2011.

[5] J. Nocedal and S. Wright, "Numerical optimization," *Springer*, 1999.

[6] R. Vaghefi, M. Gholami, and E. Strom, "RSS-based sensor localization with unknown transmit power," *ICASSP*, 2011.