COOPERATIVE PARTICLE FILTERING FOR EMITTER TRACKING WITH UNKNOWN NOISE VARIANCE

Stiven S. Dias

Embraer S.A.
São José dos Campos SP 12227-901, Brazil
+55 12 3927-5428; stiven.dias@embraer.com.br

Marcelo G.S. Bruno

Instituto Tecnológico de Aeronáutica
São José dos Campos SP 12228-900, Brazil
+55 12 3947-6906; bruno@ita.br

ABSTRACT

We introduce in this paper a novel cooperative particle filter algorithm for tracking a moving emitter using received-signal strength (RSS) measurements with unknown observation noise variance. In the studied scenario, multiple RSS sensors passively observe independently attenuated and perturbed versions of the same broadcast signal transmitted by an emitter which is moving through the sensor field and cooperate to estimate the emitter state. The new algorithm differs from previous methods by employing a parametric approximation to reduce the associated communication burden.

Index Terms— Distributed Algorithms, Particle Filters, Emitter Tracking, RSS, Wireless Sensor Networks

1. INTRODUCTION

Distributed algorithms aim at dispersing the associated computational load across several cooperative processing units, ideally approximating the performance of a centralized equivalent method. In the context of cooperative sensor networks, most methods found in the literature assume conditional independence of the measurements available at each node. Thus, each node may calculate its local likelihood and spread this information to the network in a way that the remaining nodes can fuse them to form a global posterior.

In previous work on distributed particle filtering (PF), nodes may be able to communicate only with its immediate neighbors, e.g. in [1], [2], [3], [4], or may be able to broadcast messages to the entire network, e.g. in [5]. However, regardless of the network topology, distributed algorithms often rely on approximations to reduce communication burden, see e.g. [2], [3], [5]. In this paper, we propose a new low-communication-cost distributed PF scheme that allows us to successfully track a moving emitter from independent RSS observations corrupted by Gaussian noise with *unknown* variance, in contrast to previous literature [1], [2], [3] where the observation model parameters are perfectly known.

The remaining text is organized as follows: Sec. 2 setups the problem, Sec. 3 and 4 introduce two distributed approaches to its solution. Sec. 5 offers the simulations results and Sec. 6 summarizes the conclusions.

2. EMITTER TRACKING USING MULTIPLE RSS SENSORS

The emitter trajectory is described by the white noise acceleration model

$$\mathbf{x}_{n+1} = \mathbf{F}\mathbf{x}_n + \mathbf{w}_n \tag{1}$$

where $\mathbf{x}_n \triangleq \begin{bmatrix} x_n & \dot{x}_n & y_n & \dot{y}_n \end{bmatrix}^T$, $\mathbf{F}$ is the state transition matrix and $\{\mathbf{w}_n\}$ is a sequence of independent, identically distributed (i.i.d.) zero-mean random vectors with covariance matrix $\mathbf{Q}$. Matrices $\mathbf{F}$ and $\mathbf{Q}$ are given

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$\mathbf{Q} = \sigma_{accel}^2 \begin{bmatrix} T^3/3 & T^2/2 & 0 & 0 \\ T^2/2 & T & 0 & 0 \\ 0 & 0 & T^3/3 & T^2/2 \\ 0 & 0 & T^2/2 & T \end{bmatrix} \tag{3}$$

where T is the sampling period and σ_{accel} is the acceleration standard deviation.

2.1. Observation Model

The measurements $z_{r,0:n} = \{z_{r,0}, \ldots, z_{r,n}\}$ in dBm at the r-th node of a network of R RSS sensors are modeled as [6]

$$z_{r,n} = g_r(\mathbf{x}_n) + v_{r,n} \tag{4}$$

where $v_{r,n}$ represents an additive zero-mean i.i.d. Gaussian noise process with *unknown* variance σ_r^2 and $g_r(\cdot)$ is a non-linear function given by

$$g_r(\mathbf{x}) = P_0 - K_r 10 \log\left(\frac{\|\mathbf{H}\mathbf{x} - \mathbf{x}_r\|}{d_0}\right) \tag{5}$$

where $\mathbf{x}_r$ represents sensor position, (P_0, d_0, K_r) are known model parameters (see [6] for details), and $\mathbf{H}$ is a 2×4 projection matrix such that $H(1,1) = H(2,3) = 1$ and $H(i,j) = 0$ otherwise. Using a Bayesian approach, we also model the unknown noise variance as a realization of a random variable distributed a priori as $\sigma_r^2 \sim \mathcal{IG}(\sigma^2|\alpha, \beta)$ where $\mathcal{IG}$ denotes the inverse Gamma distribution and $\{\alpha, \beta\}$ are the model's hyperparameters.

3. CENTRALIZED SOLUTION VIA PARTICLE FILTERS

Particle Filters approximate the minimum mean-square error (MMSE) estimate of the emitter state, $\mathbf{x}_n$, at instant n, given all present and past observation $z_{1:R,0:n}$ in the network as [7]

$$\hat{\mathbf{x}}_{n|n} = E\left\{\mathbf{x}_n | z_{1:R,0:n}\right\} \approx \sum_{q=1}^{Q} w_n^{(q)} \mathbf{x}_n^{(q)} \tag{6}$$

where Q is the number of particles $\mathbf{x}_n^{(q)}$, sampled from an importance function $\pi(\cdot)$, and $w_n^{(q)}$ are the importance weights.

The so-called *blind importance function* can be written as $\pi(\mathbf{x}_n|\mathbf{x}_{0:n-1}^{(q)}, z_{1:R,0:n}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(q)})$, which is determined as $p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(q)}) = \mathcal{N}(\mathbf{x}_n|\mathbf{F}\mathbf{x}_{n-1}^{(q)}, \mathbf{Q})$, where $\mathcal{N}(\cdot|\mathbf{m}, \Sigma)$ denotes a multivariate Normal distribution with mean vector $\mathbf{m}$ and covariance matrix Σ. The importance weights can in turn be recursively propagated as

$$w_n^{(q)} \propto w_{n-1}^{(q)} p(z_{1:R,n}|\mathbf{x}_{0:n}^{(q)}, z_{1:R,0:n-1}). \tag{7}$$

On the other hand, assuming that $p(z_{1:R,0:n}|\mathbf{x}_{0:n}, \sigma_{1:R}^2) = \prod_{r=1}^{R} p(z_{r,0:n}|\mathbf{x}_{0:n}, \sigma_r^2)$ and that $p(\sigma_{1:R}^2|\mathbf{x}_{0:n}) = \prod_{r=1}^{R} p(\sigma_r^2|\mathbf{x}_{0:n}) = \prod_{r=1}^{R} p(\sigma_r^2)$, we can show that

$$p(z_{1:R,n}|\mathbf{x}_{0:n}^{(q)}, z_{1:R,0:n-1}) = \prod_{r=1}^{R} p(z_{r,n}|\mathbf{x}_{0:n}^{(q)}, z_{r,0:n-1}). \tag{8}$$

Finally, under the model assumptions, it follows after some algebraic manipulations that

$$\begin{aligned} p(z_{r,n}|\mathbf{x}_{0:n}^{(q)}, z_{r,0:n-1}) &= \\ &= \int_0^\infty p(z_{r,n}|\mathbf{x}_n^{(q)}, \sigma_r^2) p(\sigma_r^2|\mathbf{x}_{0:n}^{(q)}, z_{r,0:n-1}) d\sigma_r^2 \\ &\propto \frac{\left[\beta_{r,n-1}^{(q)}\right]^{\alpha_{n-1}}}{\Gamma(\alpha_{n-1})} \frac{\Gamma(\alpha_n)}{\left[\beta_{r,n}^{(q)}\right]^{\alpha_n}} \end{aligned} \tag{9}$$

where $\beta_{r,n}^{(q)}$ and α_n can be recursively computed by

$$\alpha_n = \alpha_{n-1} + \frac{1}{2} \tag{10}$$

$$\beta_{r,n}^{(q)} = \beta_{r,n-1}^{(q)} + \frac{1}{2}\left[z_{r,n} - g_r(\mathbf{x}_n^{(q)})\right]^2, \tag{11}$$

$g_r(\cdot)$ is calculated as indicated in (5), and $\Gamma(\cdot)$ denotes the Gamma function defined by the improper integral $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, \mathrm{d}t$.

In a centralized approach, all observations are sent to a central node to be incorporated in the posterior probability density function (p.d.f.). This approach requires that the central node know the observation model parameters of the entire sensor network and be able to process all measurements simultaneously. Moreover, if the fusion center becomes unavailable, the system collapses.

3.1. Cooperative Approach

An alternative to the aforementioned centralized approach is to use a cooperative scheme where the computation of the importance weights is distributed over different network nodes. Substituting (8) into (7) leads to the weight update rule

$$w_n^{(q)} \propto w_{n-1}^{(q)} \prod_{r=1}^{R} \lambda_{r,n}^{(q)}(\mathbf{x}_n^{(q)}) \tag{12}$$

where $\lambda_{r,n}^{(q)}(\mathbf{x}_n^{(q)}) \triangleq p(z_{r,n}|\mathbf{x}_{0:n}^{(q)}, z_{r,0:n-1})$ is calculated according to (9) through (11).

An exact decentralized implementation of (7) is as follows: at a given iteration, each node $r \in \boldsymbol{\mathcal{R}} \triangleq \{1, \ldots, R\}$ maintains a set of weighted particles describing the emitter estimated state and broadcasts its local updated model's hyperparameters $\beta_{r,n}^{(q)}$, $q \in \boldsymbol{\mathcal{Q}} \triangleq \{1, \ldots, Q\}$, to the other nodes. Conversely, each node receives the hyperparameters from the other nodes and incorporates them in its posterior p.d.f. by means of equation (12). Synchronized sampling from the prior transition p.d.f. and synchronized resampling according to the weights, see [2], guarantee that all nodes have the same set of particles.

Algorithm 3.1 is a decentralized regularized particle filter (DCRPF) that meets the aforementioned approach. This algorithm reduces the computational burden per filter (compared to the centralized solution) by spreading it over the network nodes, but requires in turn the transmission of Q real numbers per node at each time stamp n.

Algorithm 3.1: DCRPF$(z_{s,n}, \{(\mathbf{x}_{n-1}^{(q)}, w_{n-1}^{(q)})\}, \Theta_{s,n-1})$

$\langle \alpha_{n-1}, \{\beta_{r,n-1}^{(q)}\} \rangle \leftarrow \Theta_{s,n-1}$; where $q \in \boldsymbol{\mathcal{Q}}$ and $r \in \boldsymbol{\mathcal{R}}$
for each $q \in \boldsymbol{\mathcal{Q}}$
do $\begin{cases} \bar{\mathbf{x}}_n^{(q)} \sim p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(q)}) \\ \text{Calculate } \beta_{s,n}^{(q)} \text{ and broadcast it} \end{cases}$
Block until receive all $\{\beta_{r,n}^{(q)}\}$, $r \neq s$
for each $q \in \boldsymbol{\mathcal{Q}}$
do $\left\{ \bar{w}_n^{(q)} \propto w_{n-1}^{(q)} \prod_{r=1}^{R} \lambda_{r,n}^{(q)}(\bar{\mathbf{x}}_n^{(q)}) \right.$
$\{(\mathbf{x}_n^{(q)}, w_n^{(q)}, l^{(q)})\} \leftarrow$ REGULARIZE$(\{(\bar{\mathbf{x}}_n^{(q)}, \bar{w}_n^{(q)})\})$
$\Theta_{s,n} \leftarrow \langle \alpha_n, \{\beta_{r,n}^{(l^{(q)})}\} \rangle$
return $(\{(\mathbf{x}_n^{(q)}, w_n^{(q)})\}, \Theta_{s,n})$

where $\lambda_{r,n}^{(q)}(\bar{\mathbf{x}}_n^{(q)})$ is calculated as previously stated and the regularization step [8] is performed as shown in Algorithm 3.2 taking D_n such that $D_n D_n^T$ corresponds to the empirical covariance of the weighted particles $\{(\bar{\mathbf{x}}_n^{(q)}, \bar{w}_n^{(q)})\}$ and $h = h_{opt}/2$ (see [8] for further details).

Algorithm 3.2: REGULARIZE($\{(\bar{\mathbf{x}}_n^{(q)}, \bar{w}_n^{(q)})\}$)

for each $q \in \mathcal{Q}$

do $\begin{cases} l^{(q)} \sim \{1, 2, \ldots, Q\};\ P(\{L^{(q)} = l\}) = \bar{w}_n^{(q)} \\ \mathbf{x}_n^{(q)} \leftarrow \bar{\mathbf{x}}_n^{(l^{(q)})} + hD_n\mathbf{x}^*;\ where\ \mathbf{x}^* \sim \mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{I})) \\ w_n^{(q)} \leftarrow 1/Q \end{cases}$

return $(\{(\mathbf{x}_n^{(q)}, w_n^{(q)}, l^{(q)})\})$

At each node r, the filter must be initialized from a prior distribution of the emitter state, i.e. $\mathbf{x}_0^{(q)} \sim p(\mathbf{x}_0)$ and $w_0^{(q)} = 1/Q$ for all $q \in \mathcal{Q}$, and assuming an initial inverse Gamma distribution $\mathcal{IG}(\sigma^2|\alpha_0, \beta_0)$ for the unknown parameter σ_r^2.

4. APPROXIMATE DISTRIBUTED PARTICLE FILTER

As discussed before, despite its asymptotic optimality, the DCRPF algorithm has inter-node communication requirements that may be prohibitive in most practical scenarios. As observed in [5], the communication burden may be considerably reduced though by making suboptimal parametric approximations that allows one to drop the dependence of the coefficients $\lambda_{r,n}^{(q)}$ in (12) on the particle label q.

Assuming now a Monte Carlo representation of the posterior $p(\mathbf{x}_{n-1}|z_{1:R,0:n-1})$ by the properly weighted set $\{(\mathbf{x}_{n-1}^{(q)}, w_{n-1}^{(q)})\}$ and from (8), one can show that the marginal $p(\sigma_r^2|z_{1:R,0:n-1})$ is approximated by the weighted sum

$$p(\sigma_r^2|z_{1:R,0:n-1}) \approx \sum_{q=1}^{Q} w_{n-1}^{(q)} \mathcal{IG}(\sigma_r^2|\alpha_{n-1}, \beta_{r,n-1}^{(q)}). \quad (13)$$

To reduce the communication requirements, we propose to further approximate this marginal by a single inverse Gamma distribution with parameters $(\tilde{\alpha}_{r,n-1}, \tilde{\beta}_{r,n-1})$ chosen such that the approximated distribution $\mathcal{IG}(\sigma_r^2|\tilde{\alpha}_{r,n-1}, \tilde{\beta}_{r,n-1})$ matches the first and second moments of the marginal (13).

After some algebraic calculations, it follows that $\tilde{\alpha}_{r,n-1} = 2 + \widehat{E}_{n-1}^2\left[\sigma_r^2\right] / \widehat{VAR}_{n-1}\left[\sigma_r^2\right]$ and $\tilde{\beta}_{r,n-1} = (\tilde{\alpha}_{r,n-1} - 1)\widehat{E}_{n-1}\left[\sigma_r^2\right]$, where $\widehat{E}_{n-1}\left[\sigma_r^2\right]$ and $\widehat{VAR}_{n-1}\left[\sigma_r^2\right]$ are obtained from $\left\{(w_{n-1}^{(q)}, \beta_{r,n-1}^{(q)})\right\}$ and α_{n-1} using the expressions in Ref. [5].

At a given node s, the weight update rule (12) is approximated then by

$$\bar{w}_n^{(q)} \propto w_{n-1}^{(q)} \left[\prod_{r=1, r\neq s}^{R} \tilde{\lambda}_{r,n}(\mathbf{x}_n^{(q)})\right] \lambda_{s,n}^{(q)}(\mathbf{x}_n^{(q)}) \quad (14)$$

where

$$\tilde{\lambda}_{r,n}(\mathbf{x}_n^{(q)}) = \frac{\left[\tilde{\beta}_{r,n-1}\right]^{\tilde{\alpha}_{r,n-1}}}{\Gamma(\tilde{\alpha}_{r,n-1})} \frac{\Gamma(\tilde{\alpha}_{r,n})}{\left[\tilde{\beta}_{r,n}^{(q)}\right]^{\tilde{\alpha}_{r,n}}} \quad (15)$$

$$\tilde{\alpha}_{r,n} = \tilde{\alpha}_{r,n-1} + \frac{1}{2} \quad (16)$$

$$\tilde{\beta}_{r,n}^{(q)} = \tilde{\beta}_{r,n-1} + \frac{1}{2}\left[z_{r,n} - g_r(\mathbf{x}_n^{(q)})\right]^2. \quad (17)$$

Note that, once the particle update rule is different at each node, this approach no longer guarantees that particles at each node are the same.

Algorithm 4.1 outlines the approximate distributed regularized particle filter (ADRPF).

Algorithm 4.1: ADRPF($z_{s,n}, \{(\mathbf{x}_{n-1}^{(q)}, w_{n-1}^{(q)})\}, \Theta_{s,n-1}$)

$\langle \alpha_{n-1}, \{\beta_{s,n-1}^{(q)}\}\rangle \leftarrow \Theta_{s,n-1}$; where $q \in \mathcal{Q}$

Calculate $(\tilde{\alpha}_{s,n-1}, \tilde{\beta}_{s,n-1})$

Broadcast $(\tilde{\alpha}_{s,n-1}, \tilde{\beta}_{s,n-1}, z_{s,n})$ and block until receive all $\{(\tilde{\alpha}_{r,n-1}, \tilde{\beta}_{r,n-1}, z_{r,n})\}$, $r \neq s$

for each $q \in \mathcal{Q}$

do $\begin{cases} \bar{\mathbf{x}}_n^{(q)} \sim p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(q)}) \\ \bar{w}_n^{(q)} \propto w_{n-1}^{(q)} \left[\prod_{r=1, r\neq s}^{R} \tilde{\lambda}_{r,n}(\bar{\mathbf{x}}_n^{(q)})\right] \lambda_{s,n}^{(q)}(\bar{\mathbf{x}}_n^{(q)}) \end{cases}$

$\{(\mathbf{x}_n^{(q)}, w_n^{(q)}, l^{(q)})\} \leftarrow$ REGULARIZE($\{(\bar{\mathbf{x}}_n^{(q)}, \bar{w}_n^{(q)})\}$)

$\Theta_{s,n} \leftarrow \langle \alpha_n, \{\beta_{s,n}^{(l^{(q)})}\}\rangle$

return $(\{(\mathbf{x}_n^{(q)}, w_n^{(q)})\}, \Theta_{s,n})$

At a given iteration, a node s calculates the approximated model's hyperparameters $(\tilde{\alpha}_{s,n-1}, \tilde{\beta}_{s,n-1})$ and broadcasts them to the other nodes. Furthermore, it receives the hyperparameters $(\tilde{\alpha}_{r,n-1}, \tilde{\beta}_{r,n-1})$ from the other nodes $(r \neq s)$. Afterwards, it uses (14) to update the importance weights.

In contrast to the DCRPF algorithm, ADRPF requires that, at time stamp n, each node r broadcasts just three real numbers: the local observation $z_{r,n}$ itself and the approximated hyperparameters $(\tilde{\alpha}_{s,n-1}, \tilde{\beta}_{s,n-1})$, assuming, without loss of generality, the same (P_0, d_0) for all nodes. Moreover, each node knows the observation model parameters $\{(K_r, \mathbf{x}_r)\}$ of the entire sensor network.

5. SIMULATION RESULTS

The performance of both algorithms were assessed via simulations consisting of 100 independent Monte Carlo runs. We computed the root-mean-square (RMS) error norm associated with the estimated emitter position at each time stamp n given all realizations. The simulated system has $R = 4$ RSS sensors with parameters $P_0 = 1dBm$, $d_0 = 1m$, $K_r = 2$. Fig. 1 indicates the sensor positions. It also shows one realization of the

emitter trajectory generated for $T = 1s$, $\sigma_{accel} = 0.05m/s^2$ and $\mathbf{x}_0 = \begin{bmatrix} 25m & 0.5m/s & 50m & 0.5m/s \end{bmatrix}^T$.

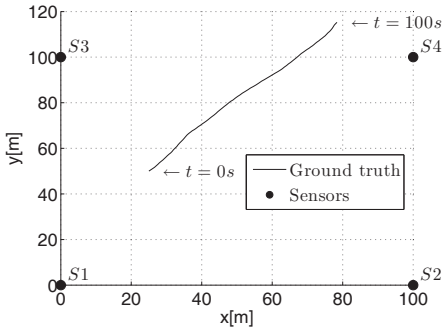

Fig. 1. Evaluated scenario.

Both filters employed $Q = 500$ particles to estimate the emitter state. Particles were initialized considering a Gaussian prior distribution, with mean $\begin{bmatrix} x_0 & y_0 \end{bmatrix}^T$ and covariance matrix $diag(20^2, 20^2)$, for the emitter position and a Gaussian prior distribution, with mean $\left[\sqrt{\dot{x}_0^2 + \dot{y}_0^2} \quad \arctan\left(\dot{y}_0/\dot{x}_0\right)\right]^T$ and covariance matrix $diag(0.3^2, (5\pi/8)^2)$, for the emitter velocity. Furthermore, at each node, the hyperparameters were initialized such that $\alpha_0 = 2$ and $\beta_{r,0}^{(q)} = 0.1$ for all particles q. Finally, the unknown observation model parameters were kept fixed during all simulations and were chosen as $\sigma_1 = 0.09$, $\sigma_2 = 0.08$, $\sigma_3 = 0.1$ and $\sigma_4 = 0.11$.

In Fig. 2, we show the evolution of the RMS error norm of the emitter position estimates for both algorithms. As one can observe, the ADRPF performs slightly worse initially than DCRPF, specially, at node $S4$, since it is the farthest node from emitter at that time. Although being nearly the same for both algorithms, the stationary error increases since the emitter tends to leave the sensor field as suggested in Fig. 1. Finally, it is worth mentioning that the cooperative scheme significantly outperformed the isolated nodes, which were not able to individually estimate the emitter state successfully without cooperation.

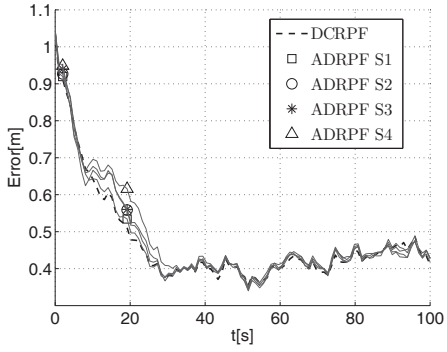

Fig. 2. Evolution of the estimated position RMS error norm.

6. CONCLUSIONS

In this work, we presented two distributed algorithms for tracking a moving emitter cooperatively by means of independent RSS observations. We also introduced a suitable parametric approximation to reduce the inter-node communication burden. The simulations show that the proposed approximation leads to a modest degradation in performance compared to an exact decentralized approach, but reduces the associated communication cost in our particular scenario by a factor of two orders of magnitude.

7. REFERENCES

[1] S. Farahmand, S. I. Roumeliots, and G. B. Giannakis, "Particle filter adaptation for distributed sensors via set membership," in *IEEE Intl. Conf. on Acoustics Speech and Sig. Proc.*, Dallas, TX, March 2010.

[2] D. Uztebay, M. Coates, and M. Rabbat, "Distributed auxiliary particle filter using selective gossip," in *IEEE Intl. Conf. on Acoustics Speech and Sig. Proc.*, Prague, Czech Republic, May 2011.

[3] O. Hlinka, O. Sluciak, F. Hlawatsch, P. Djuric, and M. Rupp, "Distributed gaussian particle filtering using likelihood consensus," in *IEEE Intl. Conf. on Acoustics Speech and Sig. Proc.*, Prague, Czech Republic, May 2011.

[4] C. J. Bordin Jr. and M. G. S. Bruno, "Consensus-based distributed particle filtering algorithms for cooperative blind equalization in receiver networks," in *IEEE Intl. Conf. on Acoustics Speech and Sig. Proc.*, Prague, Czech Republic, May 2011.

[5] C. J. Bordin Jr. and M. G. S. Bruno, "Nonlinear distributed blind equalization using network particle filtering," in *IEEE/SP 15th Workshop on Statistical Signal Processing*, August 2009.

[6] N. Patwari, A.O. Hero III, M. Perkins, N.S. Correal, and R.J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 3, pp. 2137–2148, 2003.

[7] Arnaud Doucet, Simon Godsill, and Christophe Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[8] C. Musso, N. Oudjane, and F. Le Gland, "Improving regularised particle filters," in *Sequential Monte Carlo Methods in Practice*, chapter 12. New York: Springer-Verlag, 2001.