DISTRIBUTED CONSENSUS-BASED TRACKING IN WIRELESS SENSOR NETWORKS: A PRACTICAL APPROACH

B. Béjar, P. Belanovic and S. Zazo

Dept. Señales, Sistemas y Radiocomunicaciones, UPM
ETSI de Telecomunicación C-303, 28040, Madrid, Spain
phone: + (34) 915 495 700 Ext. 4006, fax: + (34) 91 336 7350
web: www.gaps.ssr.upm.es

ABSTRACT

Tracking is a key application in Wireless Sensor and Ad-hoc Networks. The development of distributed, energy-efficient strategies for target tracking is of practical importance. In this contribution we propose a distributed tracking strategy based on cooperative localization using consensus. Nodes do not need to care about the network topology or the number of nodes present in the network. They use their own measurements as well as the jointly estimated position of the target in order to track it. The proposed approach appears to be robust against small errors in the node positions as well as in the presence of biased or misbehaving nodes.

1. INTRODUCTION

The deployment of a large number of scattered sensors in a certain area constitutes a very powerful tool for sensing and retrieving information from the environment (i.e. temperature, humidity, motion). The main features of Wireless Sensor Networks (WSN) are that of a large number of low-cost nodes with limited computational and power resources. WSNs must also be scalable and robust against changes in topology (i.e. node failure or addition of new nodes), as well as energy efficient. These are the major design issues in WSNs that make the development of simple and efficient algorithms a major issue. These limitations also make centralized approaches not very suitable for being used in WSNs. Localization is a key task (often mandatory) in many applications [1] and therefore, distributed localization and tracking algorithms are of high practical importance.

A great variety of methods exists for acquiring the position of a target node by fusing different measurement sources [1] like Time-of-Arrival (TOA), Time-difference of Arrival (TDOA), Angle of Arrival (AOA) or Received Signal Strength Indicator (RSSI). In this paper we focus on single antenna nodes without tight synchronization abilities, which leads us to the use of RSSI measurements for the localization task. One of the main challenges when using RSSI measurements is that the mapping between the measurement and target's position is nonlinear and hence, classical tracking strategies like the Kalman filter [2] are not suitable. More advanced techniques based on particle filtering [3] have also been proposed in the context of WSN with RSSI measurements [4–8]. In general, particle filtering approaches have shown very good performance when dealing with RSSI measurements but they are centralized and suffer from a high computational cost and hence, their applicability in a real scenario is questionable.

In this work we propose a two-step tracking strategy based on joint distributed localization and local tracking. Nodes perform local target tracking using the Unscented Kalman Filter [9–11] and then combine the smoothed data (coming from the tracking fileters) in a distributed fashion using consensus-based localization [12]. The obtained joint position is then used as feedback information to the local tracking filters. For that purpose, we use an augmented state variable that takes into account the previous target's position. The proposed algorithm has the following desirable properties: it is scalable (i.e. nodes do not care about the network topology), computationally simple (nodes only need to run a local UKF) and energy efficient (only 1-hop broadcast messages are used). Further, the use of weights in the joint estimation process make the proposed approach robust against uncertainties in nodes' positions and/or bias in the measurements. We validate the performance of the proposed approach by means of simulations and compare it with a standard (centralized, one-step) UKF where the tracking is performed directly from the measurements.

The paper is organized as follows: Section 2 describes the problem and the considered scenario. In Section 3 we describe the distributed consensus-based localization algorithm proposed in [12] as it will be used for distributed data fusion in the tracking algorithm of Section 4. Section 5 is devoted to validating the performance of the proposed solution using synthetic data. We draw some conclusions in Section 6 and provide references at the end of the paper.

2. PROBLEM FORMULATION AND DEFINITIONS

Consider a Wireless Sensor Network of N nodes randomly deployed on a certain area. Nodes are static and able to communicate with adjacent nodes that lie within a given range for communications. Assume the presence of a target node that moves within the network. The goal is to determine the location of the target node and be capable of tracking its position as time evolves. In the following we will provide a separate description of the behaviour of both the locating/tracking network and the target.

2.1 Network

For getting estimates of the target position, nodes employ RSSI measurements. The use of RSSI readings is of practical convenience when working with real hardware as they do not

This work was supported in part by the Spanish Ministry of Science and Innovation under the grant TEC2009-14219-C03-01; El Consejo Social de la UPM; the Spanish Ministry of Science and Innovation in the program CONSOLIDER-INGENIO 2010 under the grant CSD2008-00010 COMONSENS; the European Commission under the grant FP7-ICT-2009-4-248894-WHERE-2; the European Commission under the grant FP7-ICT-223994-N4C and the Spanish Ministry of Science and Innovation under the complementary action grant TEC 2008-04644-E; Spanish Ministry of Science and Innovation under the grant TEC2010-21217-C02-02-CR4HFDVL.

need tight synchronization requirements. We assume that the RSSI follows a linear relationship with the received power P_R. A common assumption, see [13] and references therein, is that the received power follows a lognormal distribution with a distance-dependent mean as

$$P_R[\mathrm{dB}] = P_0 - 10 n_p \log_{10}\left(\frac{d}{d_0}\right) + X, \tag{1}$$

where P_0 is the received power (in dB) at reference distance d_0, n_p is the path-loss exponent and X is a Gaussian random variable of zero mean and variance σ^2_{dB}. Let us denote $P_{R,n}$ as the measured power at the n-th locating node. The maximum likelihood estimate of the distance to the target is then given by

$$\hat{d}_n = d_0\, 10^{\left(\frac{P_0 - P_{R,n}}{10 n_p}\right)}. \tag{2}$$

Through the rest of the paper we will use $RSSI_n = P_{R,n}$ for all n.

2.2 Target

The target node can be placed at any arbitrary position in the network area. It is assumed that the target moves freely through the network by following a random force movement given by

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{v}^{(k)} T + \frac{1}{2}\mathbf{a}^{(k+1)} T^2 \tag{3}$$

$$\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} + \mathbf{a}^{(k+1)} T\ , \tag{4}$$

where $\mathbf{x}^{(k)}$ is the target position, $\mathbf{v}^{(k)}$ is the target speed, $\mathbf{a}^{(k)}$ is the acceleration at time instant k and T is the elapsed time between consecutive samples. It is assumed that the target is initially at some position $\mathbf{x}^{(0)} = [x_t^{(0)}, y_t^{(0)}]^\mathsf{T}$ with initial speed of $\mathbf{v}^{(0)} = [v_x^{(0)}, v_y^{(0)}]^\mathsf{T}$ and that the acceleration follows a Gaussian distribution, $\mathbf{a} \sim \mathcal{N}(\mathbf{0}, \sigma_a^2 \mathbf{I})$.

3. DISTRIBUTED LOCALIZATION

In order to locate a target in the network, nodes employ their estimated distances (2) in order to perform joint localization. Particularly, we have that the following set of equations should be satisfied

$$\begin{array}{c} d_1^2 = (x_1 - x_t)^2 + (y_1 - y_t)^2 \\ d_2^2 = (x_2 - x_t)^2 + (y_2 - y_t)^2 \\ \vdots \\ d_N^2 = (x_N - x_t)^2 + (y_N - y_t)^2 \end{array} \tag{5}$$

where d_n, $n = 1 \ldots N$ is the distance between the target and the n-th node, (x_n, y_n) are the node coordinates, and (x_t, y_t) are the target coordinates.

By further developing (5) we have that

$$\begin{array}{c} d_1^2 = x_1^2 + x_t^2 - 2x_1 x_t + y_1^2 + y_t^2 - 2y_1 y_t \\ d_2^2 = x_2^2 + x_t^2 - 2x_2 x_t + y_2^2 + y_t^2 - 2y_2 y_t \\ \vdots \\ d_N^2 = x_N^2 + x_t^2 - 2x_N x_t + y_N^2 + y_t^2 - 2y_N y_t \end{array} . \tag{6}$$

Rearranging terms we can express (6) in a more compact vector-matrix form as

$$\begin{bmatrix} d_1^2 - (x_1^2 + y_1^2) \\ \vdots \\ d_N^2 - (x_N^2 + y_N^2) \end{bmatrix} = \left(\mathbf{x}^\mathsf{T}\mathbf{x}\right)\cdot \mathbf{1} - 2 \underbrace{\begin{bmatrix} x_1 \ y_1 \\ \vdots \\ x_N \ y_N \end{bmatrix}}_{\mathbf{C} = [\mathbf{c}_1,\, \ldots,\, \mathbf{c}_N]^\mathsf{T}} \mathbf{x}\ , \tag{7}$$

where $\mathbf{1}$ is a $N \times 1$ vector of all ones and $\mathbf{x} = [x_t \ y_t]^\mathsf{T}$. However, we do not have the actual distances to the target but a noisy version (2). Define the vector $\mathbf{b} = \left[\hat{d}_1^2 - (x_1^2 + y_1^2),\, \ldots,\, \hat{d}_N^2 - (x_N^2 + y_N^2)\right]^\mathsf{T}$ and the vector-valued cost function

$$\tilde{\mathbf{f}}(\mathbf{x}) = \left(\mathbf{x}^\mathsf{T}\mathbf{x}\right)\cdot \mathbf{1} - 2\mathbf{C}\,\mathbf{x} - \mathbf{b}\ . \tag{8}$$

In order to incorporate robustness and make the localization task more applicable to realistic scenarios we propose to use a weighted version of the cost function (8). In a WSN it may happen that some of the nodes exhibit irregular behaviour (i.e. bias in their measurements). Additionally, nodes may not have precise information about their own locations instead, some inaccuracies may be present. The incorporation of weights will mitigate the effects of misbehaving or biased nodes and uncertainties in nodes' positions. So we define the cost function $\mathbf{f}(\mathbf{x})$ to be

$$\mathbf{f}(\mathbf{x}) = \mathbf{\Gamma}\tilde{\mathbf{f}}(\mathbf{x})\ , \tag{9}$$

where Γ is a weighting diagonal matrix. A proper choice for the weights would be inversely proportional to the variance of the measurements. As we are assuming the log-normal model for the measurements it is well known that the variance of the maximum likelihood estimate (2) is proportional to the square of the true distance to be estimated [13,14]. With this consideration in mind we choose to weight our measurements inversely proportional to the measured distance, that is

$$\mathbf{\Gamma} = \mathrm{diag}\left(1/\hat{d}_1,\, \ldots,\, 1/\hat{d}_N\right)\ . \tag{10}$$

It is important to mention that in our tracking algorithm, distances provided to the localization part do not coincide with the maximum likelihood distance estimates in (2). Instead, distances are obtained from the local tracking filters on each node as we will see in Section 4. However, we keep this choice of the weighting matrix for simplicity.

It is easy to observe that the cost function $\mathbf{f}(\mathbf{x})$ in (9) would be identically $\mathbf{0}$ (zero vector) under perfect distance estimation. Due to the presence of errors, function $\mathbf{f}(\mathbf{x})$ represents the disagreement between the measured and the estimated target position. It is clear then that a weighted nonlinear least-squares estimate of the target position can be then obtained as the solution to the following optimization problem

$$\hat{\mathbf{x}} = \min_{\mathbf{x}}\ \frac{1}{2}\mathbf{f}(\mathbf{x})^\mathsf{T}\mathbf{f}(\mathbf{x}) = \min_{\mathbf{x}}\ \frac{1}{2}\|\mathbf{f}(\mathbf{x})\|^2\ , \tag{11}$$

where $\|\cdot\|$ denotes Euclidean vector norm.

It has been recently shown [12] that the optimization problem (11) can be efficiently solved in a distributed fashion by means of consensus [15]. By only local communication (1-hop neighbourhood) nodes can approach the centralized solution of (11) by performing a distributed version of the Gauss-Newton algorithm. The weighted version of the consensus-based distributed localization method is then summarized in Algorithm 1, where K is the maximum number of iterations and $\mathbf{J}^{(k)}$ is the Jacobian matrix of $\mathbf{f}(\mathbf{x}^{(k)})$. We immediately note that the steps 3-6 and 11-12 can all be performed locally by each node. The only communication occurs in the steps 8 and 9 via standard average consensus algorithms [15]. After the consensus rounds, nodes can compute the Gauss-Newton descent direction locally and update their estimates (steps 11 and 12). Seeing how $\mathbf{\Delta}_n^{(k)} \in \mathbb{R}^{2\times 2}$ and is symmetric, and $\boldsymbol{\gamma}_n^{(k)} \in \mathbb{R}^2$, we conclude that each consensus round requires a broadcast of only 5 real values.

Algorithm 1 Distributed Gauss-Newton localization

$\hat{\mathbf{x}}^{(0)} \leftarrow$ same initial value $\forall\, n \in N$
for $k = 0$ to $K-1$ **do**
 $\mathbf{J}_n^{(k)} \leftarrow \frac{2}{\hat{d}_n} \left[\hat{x}_t^{(k)} - x_n \;\; \hat{y}_t^{(k)} - y_n\right]$
 $f_n(\hat{\mathbf{x}}^{(k)}) \leftarrow \frac{\hat{\mathbf{x}}^{(k)\top}\hat{\mathbf{x}}^{(k)} - 2\, \mathbf{c}_n^\top \hat{\mathbf{x}}^{(k)} + x_n^2 + y_n^2 - \hat{d}_n^2}{\hat{d}_n}$
 $\boldsymbol{\Delta}_n^{(k)} \leftarrow \mathbf{J}_n^{(k)\top}\mathbf{J}_n^{(k)}$
 $\boldsymbol{\gamma}_n^{(k)} \leftarrow \mathbf{J}_n^{(k)\top} f_n(\hat{\mathbf{x}}^{(k)})$
 consensus
  $\boldsymbol{\Delta}_*^{(k)} \leftarrow \frac{1}{N}\sum_{n=1}^N \boldsymbol{\Delta}_n^{(k)} = \frac{1}{N}\mathbf{J}^{(k)\top}\mathbf{J}^{(k)}$
  $\boldsymbol{\gamma}_*^{(k)} \leftarrow \frac{1}{N}\sum_{n=1}^N \boldsymbol{\gamma}_n^{(k)} = \frac{1}{N}\mathbf{J}^{(k)\top} f(\hat{\mathbf{x}}^{(k)})$
 end consensus
 $\mathbf{h}^{(k)} \leftarrow {\boldsymbol{\Delta}_*^{(k)}}^{-1} \boldsymbol{\gamma}_*^{(k)} = \left(\mathbf{J}^{(k)\top}\mathbf{J}^{(k)}\right)^{-1} \mathbf{J}^{(k)\top}\mathbf{f}(\hat{\mathbf{x}}^{(k)})$
 $\hat{\mathbf{x}}^{(k+1)} \leftarrow \hat{\mathbf{x}}^{(k)} + \mathbf{h}^{(k)}$
end for

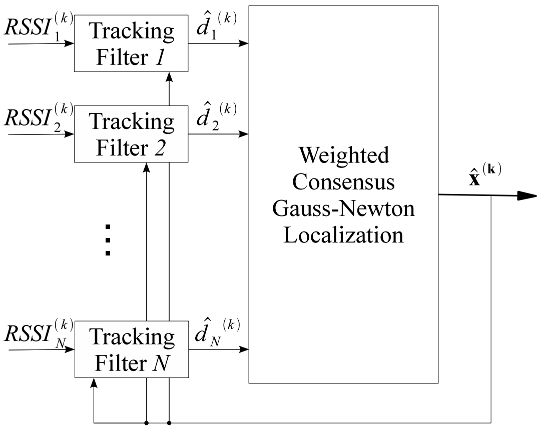

Figure 1: Distributed filtering structure

4. DISTRIBUTED TRACKING ALGORITHM

In order to track a target in the network we propose to use a two-step approach based on local data smoothing and joint localization using Algorithm 1. The idea is incorporate knowledge coming from the joint estimate into the local smoothing filter. The structure of the tracking strategy is depicted in Figure 1. As it can be observed, each node runs its local tracking filter and produces a smooth distance estimate in order to jointly compute the target position by means of consensus as described in Algorithm 1. The jointly estimated position is then fed back to the tracking filter.

It is important to mention that the filtering structure of Figure 1 is general in the sense that different local filters could be used (i.e. Kalman filter, particle filter). For the tracking filters we employ the Unscented Kalman Filter (UKF) [9, 10]. The use of the UKF is motivated by the fact that the available measurements are a nonlinear function of the target's position. The basic idea of the UKF is to deterministically sample or generate a set of points (called sigma points) from the covariance matrix of the state variable. Then this set of sigma points are propagated through the nonlinear function in order to approximate the statistics of the true state variable / measurement. We do not go into the details of the UKF because of space limitations. A detailed description of the UKF and related filters can be found in [11].

The general formulation of the tracking problem is represented by two equations, one describing the dynamics of the state variable (i.e. variable to be tracked) and another one that relates the state variable with some measurement. The state and measurement equations respectively, are given by

$$\mathbf{s}^{(k)} = \mathbf{f}\left(\mathbf{s}^{(k-1)}, \mathbf{w}^{(k)}\right) \tag{12}$$

$$\mathbf{z}^{(k)} = \mathbf{g}\left(\mathbf{s}^{(k)}, \mathbf{n}^{(k)}\right) \tag{13}$$

where $\mathbf{s}^{(k)}$ is the state variable at time instant k and $\mathbf{w}^{(k)}$ is the driving noise process. The variable $\mathbf{z}^{(k)}$ represents the measurement at time instant k and $\mathbf{n}^{(k)}$ is the measurement noise process. The functions $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ may be nonlinear functions of the state and noise processes.

In our particular case, define the state variable to be composed of the current target position and velocity and the previous position. The inclusion of the previous target position into the state variable has been done in order to incorporate the information coming from the joint estimation process (see Figure 1).

We then have the following representation of the state evolution

$$\mathbf{s}^{(k)} = \begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{v}^{(k)} \\ \mathbf{x}^{(k-1)} \end{bmatrix} = \mathbf{F}\mathbf{s}^{(k-1)} + \mathbf{W}\mathbf{a}^{(k)} \,, \tag{14}$$

where matrices $\mathbf{F}$ and $\mathbf{W}$ are given by

$$\mathbf{F} = \begin{bmatrix} \mathbf{I} & T\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} \,, \; \mathbf{W} = \begin{bmatrix} (T^2/2)\,\mathbf{I} \\ T\,\mathbf{I} \\ \mathbf{0} \end{bmatrix} \,. \tag{15}$$

The measurement available at each node is then composed by the received power from the target and the jointly estimated position with Algorithm 1 as illustrated in Figure 1.

Formally written, we have that the measurement available for the n-th node is then given by

$$\mathbf{z}_n^{(k)} = \begin{bmatrix} P_0 - 10\, n_p \log\left(\frac{\|\mathbf{G}\mathbf{s}^{(k)} - \mathbf{c}_n\|}{d_0}\right) + X_n^{(k)} \\ \hat{\mathbf{x}}^{(k-1)} \end{bmatrix} \,, \tag{16}$$

where $X_n^{(k)} \sim \mathcal{N}(0, \sigma_{\mathrm{d}B}^2)\,,\; n = 1, \ldots,\; N$ is the measurement noise variance and $\mathbf{G} = [\mathbf{I}\;\mathbf{0}\;\mathbf{0}]$. The joint position estimate at time instant k can also be expressed as $\hat{\mathbf{x}}^{(k)} = \mathbf{x}^{(k)} + \boldsymbol{\varepsilon}^{(k)}$, where $\boldsymbol{\varepsilon}^{(k)}$ is the error term at time instant k. As the number of nodes increases and by virtue of the Central Limit Theorem we can model the error $\boldsymbol{\varepsilon}^{(k)}$ to have a Gaussian distribution of zero mean and covariance $\sigma_\varepsilon^2\mathbf{I}$. We can now rewrite the measurement equation at node n as

$$\mathbf{z}_n^{(k)} = \begin{bmatrix} P_0 - 10\, n_p \log\left(\frac{\|\mathbf{G}\mathbf{s}^{(k)} - \mathbf{c}_n\|}{d_0}\right) \\ \mathbf{x}^{(k-1)} \end{bmatrix} + \mathbf{n}^{(k)} \,, \tag{17}$$

where $\mathbf{n}^{(k)}$ is the total measurement noise term given by

$$\mathbf{n}^{(k)} = \begin{bmatrix} X_n^{(k)} \\ \boldsymbol{\varepsilon}^{(k)} \end{bmatrix} \,, \tag{18}$$

As all components of $\mathbf{n}^{(k)}$ are Gaussian, so it is the total measurement noise (i.e. $\mathbf{n}^{(k)} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$). Note that the

Algorithm 2 Distributed tracking algorithm

$\hat{\mathbf{s}}^{(0)} = \mathbf{s}_0,\ \mathbf{P}_{\mathbf{s}}^{(0)} = \mathbf{P}_0,\ \hat{\mathbf{x}}^{(0)} = \mathbf{x}_0$ `/* Initialize */`

Get $RSSI_n$ `/* Sensing */`

$\mathbf{z}^{(k+1)} = \begin{bmatrix} RSSI_n \\ \hat{\mathbf{x}}^{(k)} \end{bmatrix}$ `/* Measurement Vector */`

`/* Unscented Kalman Filtering fed with` $\mathbf{z}^{(k+1)}$ `*/`

$\boldsymbol{\sigma}^{(k)} \leftarrow \pm\sqrt{(L+\kappa)\mathbf{P}_{\mathbf{s}}^{(k)}}$ `/* Draw Sigma Points */`

$\mathcal{S}^{(k)} \leftarrow \left\{\hat{\mathbf{s}}^{(k)},\ \hat{\mathbf{s}}^{(k)} + \boldsymbol{\sigma}^{(k)}\right\}$ `/* Sigma Set */`

$\mathcal{S}^{(k+1|k)} \leftarrow \mathbf{F}\,\mathcal{S}^{(k)}$ `/* Propagate */`

`/* Predicted Mean & Covariance */`

$\hat{\mathbf{s}}^{(k+1|k)} \leftarrow \alpha_0 \mathcal{S}_0^{(k+1|k)} + \frac{1}{2}\sum_{i=1}^{2L} \alpha_i \mathcal{S}_i^{(k+1|k)}$

$\mathbf{P}_{\mathbf{s}}^{(k+1|k)} \leftarrow \alpha_0 \mathbf{S}_0^{(k+1|k)} + \frac{1}{2}\sum_{i=1}^{2L} \alpha_i \mathbf{S}_i^{(k+1|k)}$

`/* Measurement Prediction */`

$\mathcal{Z}^{(k+1|k)} \leftarrow P_0 - 10\, n_p \log_{10}\left(\frac{\left\|\mathbf{G}\mathcal{S}^{(k+1|k)} - \mathbf{a}_n\right\|}{d_0}\right)$

$\hat{\mathbf{z}}^{(k+1|k)} \leftarrow \alpha_0\, \mathcal{Z}_0^{(k+1|k)} + \frac{1}{2}\sum_{i=1}^{2L} \alpha_i\, \mathcal{Z}_i^{(k+1|k)}$

$\mathbf{P}_{\mathbf{z}}^{(k+1|k)} \leftarrow \alpha_0\, \mathbf{Z}_0^{(k+1|k)} + \frac{1}{2}\sum_{i=1}^{2L} \alpha_i\, \mathbf{Z}_i^{(k+1|k)}$

$\mathbf{P}_{\mathbf{xz}}^{(k+1|k)} \leftarrow \alpha_0\, \boldsymbol{\Sigma}_0^{(k+1|k)} + \frac{1}{2}\sum_{i=1}^{2L} \alpha_i\, \boldsymbol{\Sigma}_i^{(k+1|k)}$

`/* Add Noise Covariance */`

$\mathbf{P}_{\mathbf{z}}^{(k+1|k)} \leftarrow \mathbf{P}_{\mathbf{z}}^{(k+1|k)} + \mathbf{Q}$

$\boldsymbol{\nu}^{(k+1|k)} \leftarrow \mathbf{z}^{(k+1)} - \hat{\mathbf{z}}^{(k+1|k)}$ `/* Innovation */`

$\mathbf{K}^{(k+1)} \leftarrow \mathbf{P}_{\mathbf{xz}}^{(k+1|k)}\, \mathbf{P}_{\mathbf{z}}^{(k+1|k)^{-1}}$ `/* Kalman Gain */`

`/* Update Phase */`

$\hat{\mathbf{s}}^{(k+1)} \leftarrow \hat{\mathbf{s}}^{(k+1|k)} + \mathbf{K}^{(k+1)}\, \boldsymbol{\nu}^{(k+1)}$

$\mathbf{P}_{\mathbf{s}}^{(k+1)} \leftarrow \mathbf{P}_{\mathbf{s}}^{(k+1|k)} - \mathbf{K}^{(k+1)}\, \mathbf{P}_{\mathbf{z}}^{(k+1|k)}\, \mathbf{K}^{(k+1)^{\mathsf{T}}}$

`/* Consensus Gauss-Newton Localization */`

$\hat{d}_n^{(k+1)} = \|\mathbf{G}\hat{\mathbf{s}}^{(k+1)} - \mathbf{c}_n\|$

$\hat{\mathbf{x}}^{(k+1)} \leftarrow$ **Algorithm 1** fed with $\hat{\mathbf{x}}^{(k)}$ and $\hat{d}_n^{(k+1)}$

components of $\mathbf{n}^{(k)}$ are correlated through the nonlinear operation of the optimization problem (11). However, as the number of nodes increases this correlation becomes negligible so that we can approximate the noise covariance to be diagonal with entries

$$\mathbf{Q} = \begin{bmatrix} \sigma_{\mathrm{d}B}^2 & 0 & 0 \\ 0 & \sigma_\varepsilon^2 & 0 \\ 0 & 0 & \sigma_\varepsilon^2 \end{bmatrix}. \tag{19}$$

Unfortunately we cannot provide a closed-form expression for the variance σ_ε^2. This value is considered as a filter parameter that has to be tuned. However, we have verified through simulations that the value of σ_ε^2 is not critical for the filter performance.

With all previous considerations, the proposed algorithm for distributed tracking is given in Algorithm 2, where L is the length of the state variable (i.e. 6), κ is a parameter of the UKF, $\alpha_0 = \kappa/(L+\kappa)$, $\alpha_i = 1/(L+\kappa)$, $i = 1, \ldots, 2L$ and the following matrix definitions are used

$$\mathbf{S}_0 = \left(\mathcal{S}_0^{(k+1|k)} - \hat{\mathbf{s}}^{(k+1|k)}\right)\left(\mathcal{S}_0^{(k+1|k)} - \hat{\mathbf{s}}^{(k+1|k)}\right)^{\mathsf{T}} \tag{20}$$

$$\mathbf{S}_i = \left(\mathcal{S}_i^{(k+1|k)} - \hat{\mathbf{s}}^{(k+1|k)}\right)\left(\mathcal{S}_i^{(k+1|k)} - \hat{\mathbf{s}}^{(k+1|k)}\right)^{\mathsf{T}} \tag{21}$$

$$\mathbf{Z}_0 = \left(\mathcal{Z}_0^{(k+1|k)} - \mathbf{z}^{(k+1|k)}\right)\left(\mathcal{Z}_0^{(k+1|k)} - \mathbf{z}^{(k+1|k)}\right)^{\mathsf{T}} \tag{22}$$

$$\mathbf{Z}_i = \left(\mathcal{Z}_i^{(k+1|k)} - \mathbf{z}^{(k+1|k)}\right)\left(\mathcal{Z}_i^{(k+1|k)} - \mathbf{z}^{(k+1|k)}\right)^{\mathsf{T}} \tag{23}$$

$$\boldsymbol{\Sigma}_0 = \left(\mathcal{S}_0^{(k+1|k)} - \hat{\mathbf{s}}^{(k+1|k)}\right)\left(\mathcal{Z}_0^{(k+1|k)} - \mathbf{z}^{(k+1|k)}\right)^{\mathsf{T}} \tag{24}$$

$$\boldsymbol{\Sigma}_i = \left(\mathcal{S}_i^{(k+1|k)} - \hat{\mathbf{s}}^{(k+1|k)}\right)\left(\mathcal{Z}_i^{(k+1|k)} - \mathbf{z}^{(k+1|k)}\right)^{\mathsf{T}} \tag{25}$$

5. SIMULATIONS

In order to analyze the performance of the proposed approach we have run several simulations with synthetic data. A network of 100 nodes scattered over a 100×100 [m^2] area has been generated. We have also generated 100 different target trajectories over the network area in order to test the tracking algorithm. For comparison purposes we also consider in our simulations a centralized version of the UKF where a central entity is assumed to collect all data coming from the nodes. In the simulations we label the proposed approach as Cooperative Filter (CF) to differentiate it from the UKF. A perfect consensus is assumed among the nodes. In Table 1 we summarize the values of the different parameters used for the simulations.

5.1 Uncertainties in node locations

We have performed a simulation where we have introduced a small variance in the actual node coordinates. That is, we replace $\mathbf{c}_i$, $i = 1, \ldots, N$ with $\tilde{\mathbf{c}}_i = \mathbf{c}_i + \mathcal{N}\left(\mathbf{0}, \sigma_L^2\mathbf{I}\right)$. We have varied σ_L in order to evaluate the robustness of the proposed approach against uncertainties in nodes' positions. For each trajectory new values for the uncertainties have been generated.

In Figure 2 we have represented the average error (over time & trajectories) for both the proposed approach and the centralized version of the UKF and for different values of σ_L. As it can be observed, when perfect knowledge about the nodes' positions is available, the UKF provides better results than the proposed approach. However, as the uncertainties increase the difference among the two approaches reduces and eventually, the proposed approach outperforms the centralized UKF when the uncertainties in the positions are high.

5.2 Presence of biased nodes

Another important fact that should be considered in WSN is the presence of biased or misbehaving nodes. For this purpose we have performed a simulation where for each trajectory, 10 nodes have been randomly selected to exhibit a bias in their measurements. The biases have been randomly (uniform distribution) chosen within the interval $(10, 20)$ dB.

The performance (average error over time) of the proposed filter for the different trajectories is illustrated in Figure 3. It can be clearly seen that, for the considered scenario, the proposed approach is more robust against the presence of biased nodes and that it outperforms the centralized UKF.

PARAMETER	VALUE	PARAMETER	VALUE
P_0	0 [dBm]	σ_a	0.1 [m/s^2]
d_0	1 [m]	T	0.1 [s]
n_p	2	κ	1
$\sigma_{\mathrm{d}B}$	2 [dBm]	σ_ε	1 [m]

Table 1: Simulation Parameters

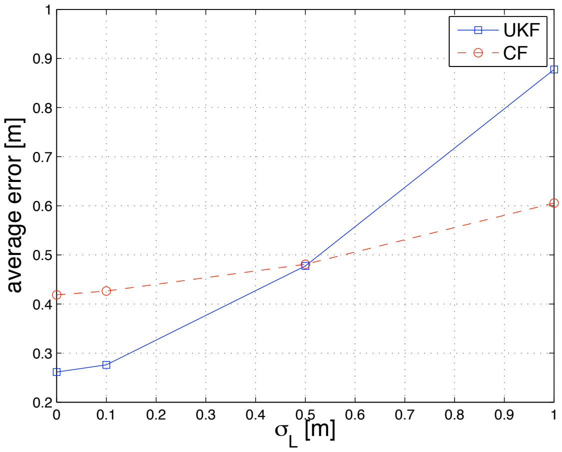

Figure 2: Performance degradation due to uncertainties in the actual node's positions

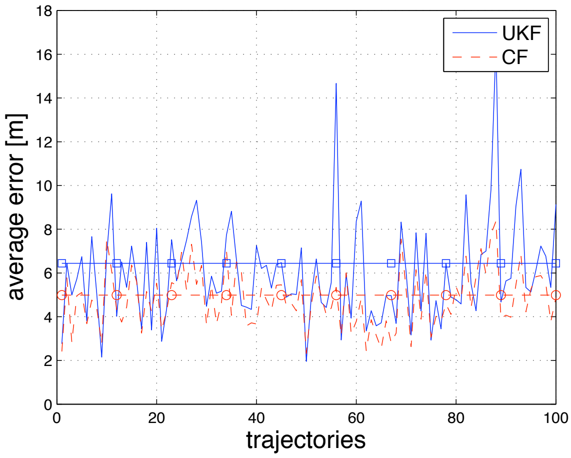

Figure 3: Performance when biased nodes are present

6. CONCLUSIONS

In this paper we have presented a practical tracking algorithm in the context of WSNs. The algorithm is based on a two-step approach that combines local tracking and consensus localization. The advantages of the proposed approach are that it is distributed, scalable and requires only local communication (1-hop neighbourhood). Further, it allows the use of other (local) tracking strategies like the Kalman filter or particle filters. We have verified by means of simulations that the introduction of the weighted joint localization makes the proposed approach robust against the presence of biased nodes and uncertainties in the actual nodes' positions, outperforming in some cases a standard tracker based on the UKF.

REFERENCES

[1] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network-Based Wireless Location," *IEEE Signal Processing Magazine*, vol. 22, pp. 24–40, July 2005.

[2] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.

[4] G. V. Zàruba, M. Huber, F. A. Kamangar, and I. Chlamtac, "Indoor location tracking using RSSI readings from a single Wi-Fi access point," *Wirel. Netw.*, vol. 13, no. 2, pp. 221–235, 2007.

[5] Y. Tai and Y. Bo, "Collaborative target tracking in wireless sensor network," in *Proc. 9th Int. Conf. Electronic Measurement & Instruments ICEMI '09*, 2009, pp. 2–1005.

[6] H. Wu, G. Tian, and B. Huang, "Multi-robot collaborative localization methods based on Wireless Sensor Network," in *Proc. IEEE Int. Conf. Automation and Logistics ICAL 2008*, 2008, pp. 2053–2058.

[7] H. Ren and M. Q.-H. Meng, "Power Adaptive Localization Algorithm for Wireless Sensor Networks Using Particle Filter," *IEEE Trans. Veh. Technol.*, vol. 58, no. 5, pp. 2498–2508, 2009.

[8] F. Aounallah, R. Amara, and M. T.-H. Alouane, "Particle Filtering Based on Sign of Innovation for Tracking a Jump Markovian Motion in a Binary WSN," in *Proc. Third Int. Conf. Sensor Technologies and Applications SENSORCOMM '09*, 2009, pp. 252–255.

[9] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A New Approach for Filtering Nonlinear Systems," in *Proceedings of the American Control Conference, Seattle, Washington*, 1995, pp. 1628 – 1632.

[10] S. Julier and J. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," in *in Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, 1997, pp. 182–193.

[11] R. van der Merwe, "Sigma-point kalman filters for probabilistic inference in dynamic state-space models," Ph.D. dissertation, OGI School of Science & Engineering, Oregon Health & Science University, Portland, OR, USA, April 2004.

[12] B. Bejar, P. Belanovic, and S. Zazo, "Distributed Gauss-Newton Method for Localization in Ad-Hoc Networks," in *43rd Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, November 2010.

[13] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the Nodes," *IEEE Signal Processing Magazine*, vol. 22, pp. 54–68, July 2005.

[14] N. Patwari, A. O. Hero, M. Perkins, and N. S. Correal, "Relative Location Estimation in Wireless Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 51, pp. 2137–2148, 2003.

[15] R. Olfati-Saber and R. Murray, "Consensus Problems in Networks of Agents with Switching Topology and Time-Delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, September 2004.