Target Tracking in Confined Environments with Uncertain Sensor Positions

(article starts on next page)

Target Tracking in Confined Environments with Uncertain Sensor Positions

Vladimir Savic, Henk Wymeersch, and Erik G. Larsson

Abstract—To ensure safety in confined environments such as mines or subway tunnels, a (wireless) sensor network can be deployed to monitor various environmental conditions. One of its most important applications is to track personnel, mobile equipment and vehicles. However, the state-of-the-art algorithms assume that the positions of the sensors are perfectly known, which is not necessarily true due to imprecise placement and/or dropping of sensors. Therefore, we propose an automatic approach for simultaneous refinement of sensors' positions and target tracking. We divide the considered area in a finite number of cells, define dynamic and measurement models, and apply a discrete variant of belief propagation which can efficiently solve this high-dimensional problem, and handle all non-Gaussian uncertainties expected in this kind of environments. Finally, we use ray-tracing simulation to generate an artificial mine-like environment and generate synthetic measurement data. According to our extensive simulation study, the proposed approach performs significantly better than standard Bayesian target tracking and localization algorithms, and provides robustness against outliers.

Index Terms—confined environments, tunnels, sensor network, simultaneous localization and tracking, belief propagation, hidden Markov model, ray tracing, time of arrival.

I. INTRODUCTION

A. Background and Motivation

A confined environment represents a constrained and irregularly-shaped area, consisting of a series of tunnels or passages that connect different rooms or halls. Typical examples are underground mines, caves, steel factories and subways. In these environments, the working conditions may be hazardous due to the possibilities of traffic accidents, machine collisions, wall collapses, fires and explosions. These environments require continuous monitoring using sensors deployed all over the area. The sensors may be wired and connected to control rooms, but to improve the safety and reduce operational costs, recently the industry is developing robust wireless communication systems for this kind of environments [2]–[4].

A wireless sensor network (WSN) can be deployed across the area to monitor the environmental conditions such as

stability, temperature and gas levels. The information obtained from the sensors can be used to control the ventilation system, and determine the unsafe areas and rescue paths. Beyond this ability, a WSN can be used to track the personnel, mobile equipment and vehicles. The problem is very challenging due to the unavailability of GPS signals and the characteristics of the propagation environment. The knowledge of the last location of an employee is especially important in the aftermath of accidents such as a wall collapse, explosion, or water inundation, but can be also used for task optimization, production monitoring and traffic management. For instance, according to the MINER act [5], created in response to the many mine tragedies in the United States during 2006, the emergency response plan "shall provide for above-ground personnel to determine the current or immediately pre-accident location of all underground personnel". This problem, that also exists in many other confined environments, is the main motivation behind the work reported in this paper.

B. Related Work

Contemporary techniques for localization and tracking in confined environments are very basic. They are typically based on manual reporting of the employee's location using paging phones or video surveillance [6], [7]. Moreover, there are few proposals in the literature, based on fingerprinting [8]–[11], trilateration [6], [12], centroid [13] and Bayesian filtering [14], [15].

More specifically, in [8], a fingerprinting technique was proposed, in which seven relevant parameters (including mean excess delay, total received power, and delay spread) were learned offline from wideband impulse responses measured at hundreds of locations. Then, these 7D vectors were used as the input to an artificial neural network pattern-matching algorithm. The measurements were conducted in a gallery of the CANMET mine, a former gold mine located in Quebec, Canada. This method was then improved in [9] by using more receivers with known positions. The fingerprinting techniques [10], [11], based on WiFi signals, have been also applied in subway tunnels in Seoul, S. Korea. The main problem of these algorithms is that they are not well suited for dynamic propagation environments (e.g., caused by movement of heavy machinery) in which the fingerprints have to be updated very frequently.

In [6], [12], ultra-wideband (UWB) measurements were used for positioning. They were motivated by a high ranging accuracy in cluttered environments and low-cost implementation of the communication system. To solve the trilateration

problem, many types of algorithms have been applied, including linearized least-squares, Gauss-Newton and bounding-box methods. The measurements were performed in the same environment as the one studied in [8]. The main drawbacks of these algorithms are that the sensor nodes have to be precisely deployed and maintained, and that the algorithms are sensitive to outliers.

In [13], a centroid algorithm was proposed, in which the miner's location was found by averaging the coordinates of the detected anchors. The algorithm is a part of a structure-aware self-adapting (SASA) WSN, which is capable of detecting structure variations caused by mine collapses. The main problem of this approach is that it requires a high density of uniformly deployed sensor nodes.

In [14], Bayesian point-mass (grid-based) filtering was applied to track mine vehicles. The main goal was monitoring and control of ore extraction from the draw points in a mine in Australia. Since the draw points are very close to each other, high tracking accuracy is required. The main problem of this approach is that it requires many grid points in order to obtain sufficiently accurate estimates.

Finally, the results of the measurement campaign [15], carried out in a basement tunnel of Linköping university and an iron-ore mine in Kiruna, Sweden, indicated that UWB time-of-arrival (TOA) allows very accurate ranging in line-of-sight (LOS) and non-LOS (NLOS) scenarios caused by thin obstacles. However, if the direct path is blocked by a thick tunnel wall, the TOA-based ranging leads to a relatively large bias. Moreover, the analysis showed that NLOS conditions cannot be accurately discriminated from LOS conditions, which means that (Bayesian) soft-decision algorithms are required for accurate ranging and positioning in these environments.

The previously described state-of-the-art algorithms assume that the positions of the sensors are perfectly known, which is not necessarily the case due to imprecise placement and/or sensor drops caused by vibrations or wall collapses.[1] One possible solution to this problem is to manually and periodically verify that the sensors positions are correct. However, this approach may be too costly and even infeasible in some areas due to the on-going activities.

C. Technical Contributions

In this paper we propose an automatic approach for target tracking with uncertain sensor positions, which involves both simultaneous refinement of the sensors position estimates (localization) and target tracking (SLAT). Our specific technical contributions are as follows:

- We divide the considered area into a finite number of cells, and define appropriate dynamic and TOA measurement models that take into account the quantization effects associated with this division.
- We formulate the localization and target tracking problem in a Bayesian setting and apply a discrete variant of belief propagation (BP). The resulting proposed algorithm

(referred to as SLAT-BP) can efficiently handle the high dimensionality of the problem and the non-Gaussian uncertainties.

- To demonstrate the performance of our SLAT-BP algorithm, we perform an extensive simulation study using synthetic impulse responses obtained from ray-tracing simulation of a mine-like environment. Our results show that SLAT-BP performs significantly better than standard Bayesian target tracking and localization algorithms, and provides robustness against outliers.

D. Paper organization

The remainder of this paper is organized as follows. In Section II, we formulate the problem and define the dynamic and measurement models. In Section III, we propose the algorithm for simultaneous localization and tracking, based on real-time belief propagation. TOA error modelling using ray-tracing simulation and performance analysis are provided in Section IV. Finally, conclusions and proposals for future work are provided in Section V.

II. SYSTEM MODEL

A. Problem formulation

We consider N_s sensors with fixed 3D positions $\mathbf{z}_n = (z_{n,1}, z_{n,2}, z_{n,3})$, $n = 1, 2, \ldots, N_s$, and one target, with 3D position $\mathbf{x}_t = (x_{t,1}, x_{t,2}, x_{t,3})$, at time t, $t = 1, 2, \ldots, N_T$, moving through the confined area. Fig. 1a illustrates the scenario. The sensors are usually placed on the walls or the ceiling, but their positions are not perfectly known. A moving target periodically emits a signal (including a unique identifier)[2] that can be detected by a subset of the sensors, with a sampling interval T_s. The target is equipped with an inertial measurement unit (IMU), so it periodically communicates its measured velocity. We also assume that there are one or more fusion centers (FCs) (e.g., a computer in a control room or a target itself), which have available the priors of the sensors' and target's positions, and periodically collect measurements from the sensors and the target.

A confined environment is naturally a continuous 3D space, but typically it is irregular and it is impossible to analytically describe the shape of its borders. On the other hand, using an unrestricted 3D continuous space would lead to decreased computational efficiency, and more importantly, significant loss of performance in that position estimates could end up, for example, behind a wall. Therefore, we propose to use a *discrete* 3D space, in which the environment is divided into a finite number of cells. The 3D position of the cell $\mathbf{l}_c = (l_{c,1}, l_{c,2}, l_{c,3})$ ($c = 1, ..., N_c$, where N_c is the number of cells) is represented, in Cartesian coordinates, by the approximation of its geometrical center. It is thereby assumed that the FCs have available a detailed floor plan of the whole area. The cell size must be chosen based on a trade-off between computational complexity and performance, and it is

---

[1] Although probably not available in confined environments nowadays, we also envision that uncertain sensors' positions can be an outcome of some (cooperative) sensor network localization algorithm [16], [17].

[2] That means that our algorithm can be also used for multi-target tracking, simply by running the same algorithm multiple times. Otherwise, different algorithms, e.g., with data association [18], would be necessary.

$$u_{t,\kappa}^m \sim p_m(u^m) = \mathcal{N}(u^m; 0, \sigma_u^2),$$

where $u_{t,\kappa}^q$ and $u_{t,\kappa}^m$ are samples (at time t, and dimension κ) of the quantization and measurement noise, drawn from the uniform $\mathrm{Unif}(\cdot)$, and the Gaussian distribution $\mathcal{N}(\cdot)$, respectively. The parameter σ_u^2 represents the variance of the measurement noise of the IMU, and $-D/T_s$, D/T_s represent the boundaries of the uniform distribution. The quantization noise is introduced into the model here because we have to relate the continuous quantity $v_{t,\kappa}$ with the discrete quantity $(x_{t,\kappa} - x_{t-1,\kappa})$. Since the boundaries of the irregularly-shaped cell are difficult to compute, we approximated the cell with a circumscribed cube with length D ($D = \max_{c,\kappa} D_{\kappa,c}$, where $D_{\kappa,c}$ is the maximum possible distance over dimension κ in cell c). Note that as the cell size D decreases, the quantization noise will eventually vanish.

The distribution of the total IMU noise $u = u^q + u^m$ is given by following convolution:

$$
\begin{aligned}
p_u(u) &= \int p_m(u - u^q) p_q(u^q) du^q \quad (2)\\
&\propto \Phi(u + \frac{D}{T_s}) - \Phi(u - \frac{D}{T_s}),
\end{aligned}
$$

where $\Phi(u) = 0.5\left(1 + \mathrm{erf}(u/(\sigma_u\sqrt{2}))\right)$ is the cumulative Gaussian distribution. Now, we can model the dynamics[3] of the target as:

$$
\begin{aligned}
p(\mathbf{v}_t, \mathbf{x}_t | \mathbf{x}_{t-1}) &= p(\mathbf{v}_t | \mathbf{x}_t, \mathbf{x}_{t-1}) p(\mathbf{x}_t | \mathbf{x}_{t-1}) \quad (3)\\
&\propto p(\mathbf{v}_t | \mathbf{x}_t, \mathbf{x}_{t-1})\\
&= \prod_{\kappa=1,2,3} p(v_{t,\kappa} | x_{t,\kappa}, x_{t-1,\kappa})\\
&= \prod_{\kappa=1,2,3} p_u(v_{t,\kappa} - (x_{t,\kappa} - x_{t-1,\kappa})/T_s),
\end{aligned}
$$

where we assumed independence between the coordinates, and that the target's mobility model is unknown (i.e., $p(\mathbf{x}_t | \mathbf{x}_{t-1}) \propto 1$).[4] Recall also that $\mathbf{x}_t$ is discrete, so (3) gives us information about the cell of the target at time t, given the cell of the target at time $t-1$, and the measured velocity $\mathbf{v}_t$.

For our framework (see Section III-A), it is convenient to formally define the "dynamics" of the sensors:

$$p(\mathbf{z}_{n,t} | \mathbf{z}_{n,t-1}) = \delta(\mathbf{z}_{n,t} - \mathbf{z}_{n,t-1}), \quad (4)$$

where $\delta(\cdot)$ is the Dirac delta impulse, which enforces that the sensors are static ($\mathbf{z}_{n,t} = \mathbf{z}_{n,t-1} = \mathbf{z}_n$).

Finally, we note that, if $p_0(\mathbf{x}_0)$ is very informative (e.g., the target's initial cell is given), the target's dynamic model (given by (3)) already constitutes sufficient information for a tracking algorithm known as *dead reckoning* [19], but this approach would suffer from error accumulation over time. Therefore, we need to use a WSN which will provide periodic measurements w.r.t. their positions.

Fig. 1: Illustration of target tracking in a confined environment: (a) deployment of the sensors and a target track, (b) division of the area into 20 cells.

preferable that all the cells have approximately the same size. With this model, in which $\mathbf{z}_n$ and $\mathbf{x}_t$ are discrete variables, our goal is to identify in which cells the target and sensors are located. This approach also facilitates the application of belief propagation (see Section III-B) without applying Monte Carlo or other approximations. Fig. 1b illustrates a confined environment divided into cells.

Finally, we assume that the prior knowledge of the sensors and the target positions is defined by probability mass functions (PMFs) $p_{n,0}(\mathbf{z}_{n,0})$, and $p_0(\mathbf{x}_0)$. While the target's prior represents information about its initial position, the priors of the sensors' positions represent the available information on the real positions of the sensors. In practice, this information may be provided by the deployment team, and may be imperfect due to imprecise deployment or drops of sensors.

B. Dynamic Model based on IMUs

We assume that the target is equipped by an IMU, consisting of a 3-axis accelerometer, and a 3-axis gyroscope, that aids the tracking. IMUs are relatively insensitive to environmental conditions, and a good choice in the motivating applications. The IMU periodically provides (each T_s second) a velocity estimate in all 3 directions $\mathbf{v}_t = (v_{t,1}, v_{t,2}, v_{t,3})$. The coordinate in the κ-th dimension ($\kappa = 1, 2, 3$) of the measured velocity at time t can be written as a function of the target's positions as:

$$v_{t,\kappa} = \frac{x_{t,\kappa} - x_{t-1,\kappa}}{T_s} + u_{t,\kappa}^q + u_{t,\kappa}^m, \quad (1)$$

$$u_{t,\kappa}^q \sim p_q(u^q) = \mathrm{Unif}(u^q; -\frac{D}{T_s}, \frac{D}{T_s}),$$

---

[3]In some of the literature, the quantity $p(\mathbf{v}_t, \mathbf{x}_t | \mathbf{x}_{t-1})$ is denoted by $p(\mathbf{x}_t | \mathbf{x}_{t-1})$. We prefer to write out the measured velocity explicitly.

[4]Even without this assumption, the IMU typically provides much more precise information than the mobility model, so $p(\mathbf{v}_t | \mathbf{x}_t, \mathbf{x}_{t-1}) p(\mathbf{x}_t | \mathbf{x}_{t-1}) \approx p(\mathbf{v}_t | \mathbf{x}_t, \mathbf{x}_{t-1})$.

C. TOA Measurement Model

We assume that round-trip TOA (RT-TOA) measurements are obtained, at each time slot, by a subset of the sensors in proximity to the target. We decided to use RT-TOA (instead of one-way TOA), in order to avoid the need for clock synchronization between the sensors and the target. We did not consider received signal strength (RSS) since the distance estimates will be highly erroneous, due to the severe multipath in confined environments, as shown in [12], [20], [21]. RT-TOA can be obtained using many techniques [22], but we assume that a simple thresholding is performed. More exactly, the RT-TOA is taken to be the arrival time of the first multipath component in the measured impulse response that exceeds a predefined threshold. Note that a signal should have very large bandwidth (WB or UWB), in order to provide high TOA resolution [23].

The measured distance between sensor n and the target at time t, can be written as:

$$d_{t,n} = c\tau_{t,n} - d_{\mathrm{PT}} = \|\mathbf{x}_t - \mathbf{z}_{n,t}\| + w_{t,n}^q + w_{t,n}^m, \quad (5)$$

$$w_{t,n}^q \sim p_q(w^q) = \mathrm{Unif}(w^q; 0, D\sqrt{3}), \ w_{t,n}^m \sim p_m(w^m),$$

where d_{PT} is a known bias caused by processing time on a target, $\tau_{t,n}$ is measured TOA, $c = 3 \cdot 10^8$ m/s is the speed of light, and $w_{t,n}^q$, $w_{t,n}^m$ are samples of the quantization noise, and measurement noise, respectively. The distribution of the quantization noise is not available in a parametric form, so we choose the least informative (i.e., a uniform) distribution to keep the algorithm tractable.

While the measurement noise depends on many factors (such as thermal noise, bandwidth, and the quality of the sensors), the bias arising from multipath propagation in NLOS conditions is usually the most critical source of error. The most common approach [24], [25] is to identify NLOS measurements, and discard them, or alternatively mitigate the effect of the multipath bias. However, these techniques require an identification of NLOS conditions, which cannot be always accurately done (see, for example, [15], [24]). Therefore, we prefer a model that does not require NLOS identification, but only knowledge of the probability of having NLOS.

According to previous results [6], we can roughly model line-of-sight (LOS) measurement noise with a Gaussian distribution, and NLOS noise (in which the walls block the direct path) with a Weibull or an exponential distribution. However, in severe multi-path environments, it is expected that NLOS noise has multiple modes (see also Section IV-A). Therefore, we use a Gaussian mixture (GM), which is capable of approximating arbitrary probability distributions. Moreover, we also need to take into account other sources of NLOS, such as vehicles and machinery. Since these objects are usually dynamic, made of different materials, and have different sizes and thicknesses, it is difficult to model their effect. Therefore, we assume that this error is uniformly distributed [14], but that it appears with very small probability. In total, the model

for $p_w(w^m)$ is then given by the following mixture:

$$\begin{aligned} p_m(w^m) \ &= P_{\mathrm{LOS}} \cdot \mathcal{N}(w^m; 0, \sigma_{w,0}^2) + \qquad\qquad (6) \\ &\quad P_{\mathrm{NLOS}} \cdot \sum_{i=1,\dots,N_M} \rho_{w,i} \mathcal{N}(w^m; \mu_{w,i}, \sigma_{w,i}^2) + \\ &\quad P_{\mathrm{OBS}} \cdot \mathrm{Unif}(w^m; 0, D_{\max}), \end{aligned}$$

where $\sigma_{w,0}$ is the standard deviation of the LOS component of the noise; $\rho_{w,i}$, $\mu_{w,i}$, $\sigma_{w,i}$ are the weights, means and standard deviations of the NLOS noise caused by tunnel walls; N_M is the number of GM components; and $D_{\max}$ is the maximum distance error. P_{LOS}, P_{NLOS} and P_{OBS} are the probabilities of LOS, NLOS caused by tunnel walls, and NLOS caused by other obstacles, respectively (with $P_{\mathrm{LOS}} = 1 - P_{\mathrm{NLOS}} - P_{\mathrm{OBS}}$). While P_{NLOS} and P_{OBS} can be approximately estimated by examining the floor plan of the deployment area, the GM parameters ($\rho_{w,i}$, $\mu_{w,i}$ and $\sigma_{w,i}$) can be estimated by applying the expectation-maximization (EM), generalized EM, or the k-means algorithm [26, Chapter 9] on training samples.

The distribution of the total noise $w = w^m + w^q$ is given by the following convolution:

$$\begin{aligned} p_w(w) \ &= \int p_m(w - w^q) p_q(w^q) dw^q \qquad\qquad (7) \\ &= \frac{P_{\mathrm{LOS}}}{D\sqrt{3}} \left(\Phi_0(w) - \Phi_0(w - D\sqrt{3}) \right) + \\ &\quad \frac{P_{\mathrm{NLOS}}}{D\sqrt{3}} \sum_{i=1,\dots,N_M} \rho_{w,i} \left(\Phi_i(w) - \Phi_i(w - D\sqrt{3}) \right) + \\ &\quad g(w) P_{\mathrm{OBS}}, \end{aligned}$$

where $\Phi_i(\cdot)$ is the shorthand notation for $\Phi(\cdot; \mu_{w,i}, \sigma_{w,i}^2)$, and the distribution $g(w)$ is found by convolution of two uniform distributions:

$$g(w) = \begin{cases} w/(D_{\max} D\sqrt{3}), & 0 < w < D\sqrt{3} \\ 1/D_{\max}, & D\sqrt{3} < w < D_{\max} \\ (D_{\max}' - w)/(D_{\max} D\sqrt{3}), & D_{\max} < w < D_{\max}' \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $D_{\max}' = D\sqrt{3} + D_{\max}$. Finally, the likelihood function is given by:

$$p(d_{n,t}|\mathbf{x}_t, \mathbf{z}_{n,t}) = p_w(d_{n,t} - \|\mathbf{x}_t - \mathbf{z}_{n,t}\|). \tag{9}$$

The likelihood functions can be computed for all sensors which detect the target. However, we assume that the sensors can perform measurements with the target if and only if $d_{n,t} < d_{\mathrm{TH}}$ where d_{TH} is a predefined *sensing radius*. The set of sensors that perform measurements at time t is denoted by G_t^s. The sensing radius is chosen so as to ensure that a sufficient number of sensors can detect the target, but should be small enough that the model in (6) remains valid.

III. SIMULTANEOUS SENSOR LOCALIZATION AND TARGET TRACKING

Our goal is to obtain the posterior marginal PMFs (referred to as the *beliefs*), $p(\mathbf{x}_t|\mathbf{e}_{1:t})$ and $p(\mathbf{z}_{n,t}|\mathbf{e}_{1:t})$, of the following
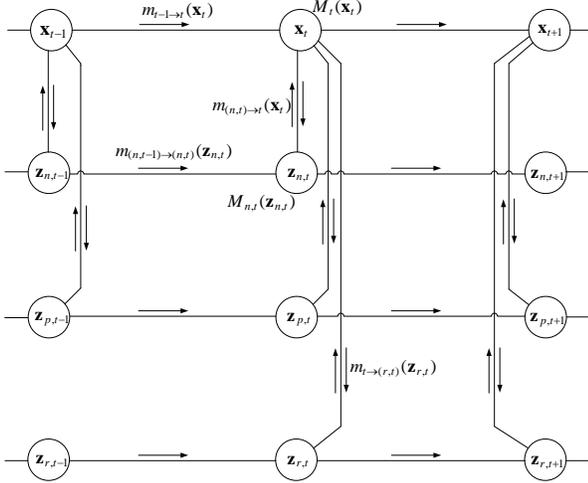
Fig. 2: An example of a MRF, with three sensors (n, p, and r) and a target at three time instants ($t-1$, t, and $t+1$). The messages are represented with arrows.

joint distribution:

$$p(\mathbf{x}_0, ..., \mathbf{x}_t, \mathbf{z}_{1,0}, ..., \mathbf{z}_{N_s,t}|\mathbf{e}_{1:t}) \propto \qquad (10)$$

$$p_0(\mathbf{x}_0) \prod_{n=1...N_s} p_{n,0}(\mathbf{z}_{n,0}) \prod_{\substack{t'=1...t \\ \forall n \in G_t^s}} p(d_{n,t}|\mathbf{x}_{t'}, \mathbf{z}_{n,t'}) \cdot$$

$$\prod_{t'=1...t} p(\mathbf{v}_{t'}, \mathbf{x}_{t'}|\mathbf{x}_{t'-1}) \prod_{\substack{t'=1...t \\ n=1...N_s}} p(\mathbf{z}_{n,t'}|\mathbf{z}_{n,t'-1}),$$

where $\mathbf{e}_{1:t}$ is all available evidence up to time t (i.e., measured TOAs and velocities). The previous factorization is obtained using Bayes' rule and standard assumptions [27], such as independence of the measurements/priors and memoryless movement. Since the marginalization of (10) is intractable, we resort to message-passing on a graphical model.

A. Graphical model

We use an undirected graphical model [28], also known as a Markov random field (MRF),[5] to represent the factorization in (10). In a MRF, each vertex represents a random variable with an associated *single-node* potential (a local evidence), and each edge represents a *pairwise* potential (a likelihood function). An example is shown in Fig. 2. Using the models defined in Sections II-B and II-C, the potentials are given by:

$$\psi_t(\mathbf{x}_t) = \begin{cases} p_0(\mathbf{x}_0), & \text{if } t = 0 \\ 1, & \text{otherwise} \end{cases}, \qquad (11)$$

$$\psi_{n,t}(\mathbf{z}_{n,t}) = \begin{cases} p_{n,0}(\mathbf{z}_{n,0}), & \text{if } t = 0 \\ 1, & \text{otherwise} \end{cases}, \qquad (12)$$

$$\psi_{t,(n,t)}(\mathbf{x}_t, \mathbf{z}_{n,t}) = p(d_{n,t}|\mathbf{x}_t, \mathbf{z}_{n,t}), \qquad (13)$$

$$\psi_{t-1,t}(\mathbf{x}_{t-1}, \mathbf{x}_t) = p(\mathbf{v}_t, \mathbf{x}_t|\mathbf{x}_{t-1}), \qquad (14)$$

$$\psi_{(n,t-1),(n,t)}(\mathbf{z}_{n,t-1}, \mathbf{z}_{n,t}) = p(\mathbf{z}_{n,t}|\mathbf{z}_{n,t-1}). \qquad (15)$$

[5] An equivalent algorithm can be derived using the forward phase of the *forward-backward* algorithm (also known as BCJR) in a hidden Markov model [29]. However, we prefer to use a much more flexible framework, valid for discrete, continuous and mixed variables.

Note that these potentials, as well as all messages and beliefs (defined in further text), are not necessarily normalized.

B. SLAT via Real-Time Belief Propagation (BP)

We adapt the standard BP (see [30, eqs. (8)-(9)]) for our real-time and discrete problem. To ensure real-time execution, we do not send the messages backward in time (see also Fig. 2), and since the variables are discrete, we replace integration with summation. The beliefs (denoted with $M_{n,t}(\mathbf{z}_{n,t})$ and $M_t(\mathbf{x}_t)$, respectively) are initialized with $M_{n,0}(\mathbf{z}_{n,0}) = \psi_{n,0}(\mathbf{z}_{n,0})$ and $M_0(\mathbf{x}_0) = \psi_0(\mathbf{x}_0)$. The algorithm (for time slot t, $t > 0$) is summarized below:

Step 1. Compute the sensor-to-target and target-to-target messages:

$$m_{(n,t)\to t}(\mathbf{x}_t) = \sum_{\mathbf{z}_{n,t}} \psi_{t,(n,t)}(\mathbf{x}_t, \mathbf{z}_{n,t}) M_{n,t-1}(\mathbf{z}_{n,t}), \quad (16)$$

$$m_{t-1\to t}(\mathbf{x}_t) = \sum_{\mathbf{x}_{t-1}} \psi_{t-1,t}(\mathbf{x}_{t-1}, \mathbf{x}_t) M_{t-1}(\mathbf{x}_{t-1}). \quad (17)$$

Step 2. Update the target's belief:

$$M_t(\mathbf{x}_t) = m_{t-1\to t}(\mathbf{x}_t) \prod_{n \in G_t^s} m_{(n,t)\to t}(\mathbf{x}_t). \qquad (18)$$

Step 3. Compute the target-to-sensor and sensor-to-sensor messages:

$$m_{t\to(n,t)}(\mathbf{z}_{n,t}) = \sum_{\mathbf{x}_t} \psi_{t,(n,t)}(\mathbf{x}_t, \mathbf{z}_{n,t}) \frac{M_t(\mathbf{x}_t)}{m_{(n,t)\to t}(\mathbf{x}_t)}, \quad (19)$$

$$m_{(n,t-1)\to(n,t)}(\mathbf{z}_{n,t}) = M_{n,t-1}(\mathbf{z}_{n,t}). \qquad (20)$$

Step 4. Update the beliefs of the sensors:

$$M_{n,t}(\mathbf{z}_{n,t}) = \begin{cases} M_{n,t-1}(\mathbf{z}_{n,t}) m_{t\to(n,t)}(\mathbf{z}_{n,t}), & \text{if } n \in G_t^s \\ M_{n,t-1}(\mathbf{z}_{n,t}), & \text{otherwise} \end{cases}$$
$$\qquad (21)$$

Step 5. (optional) Compute the estimates using the k-nearest neighbour (kNN) approach [31]:

$$\hat{\mathbf{z}}_{n,t} = \frac{\sum\limits_{\mathbf{z}_{n,t} \in C_{\mathbf{z}_{n,t}}^k} \mathbf{z}_{n,t} M_{n,t}(\mathbf{z}_{n,t})}{\sum\limits_{\mathbf{z}_{n,t} \in C_{\mathbf{z}_{n,t}}^k} M_{n,t}(\mathbf{z}_{n,t})}, \qquad (22)$$

$$\hat{\mathbf{x}}_t = \frac{\sum\limits_{\mathbf{x}_t \in C_{\mathbf{x}_t}^k} \mathbf{x}_t M_t(\mathbf{x}_t)}{\sum\limits_{\mathbf{x}_t \in C_{\mathbf{x}_t}^k} M_t(\mathbf{x}_t)}, \qquad (23)$$

where $C_{\mathbf{z}_{n,t}}^k$ and $C_{\mathbf{x}_t}^k$ are the set of k cells with highest beliefs, $M_{n,t}(\mathbf{z}_{n,t})$ and $M_t(\mathbf{x}_t)$, respectively. The special cases $k = 1$ and $k = N_c$, correspond to MAP and MMSE estimates, respectively. Note that this phase is optional, since the main output of this algorithm (beliefs) have been already computed in steps 2 and 4.

For comparison purposes, we also consider two specific instances of the proposed SLAT algorithm: i) Bayesian point-mass target tracking (by excluding all target-to-sensor messages, i.e., $m_{t\to(n,t)}(\mathbf{z}_{n,t}) = 1$), and ii) Bayesian point-mass target localization (by excluding all target-to-sensor

and target-to-target messages, i.e., $m_{t\to(n,t)}(\mathbf{z}_{n,t}) = 1$ and $m_{t-1\to t}(\mathbf{x}_t) = 1$). The former one uses the sensors' priors to track the target, while the latter one uses the sensors' priors to locate the target independently in each time slot (i.e., without a dynamic model). Note that the target tracking algorithm in [14] can be considered as a special case of SLAT-BP (although their measurement and dynamic models are different).

C. Implementation Issues

In this section, we discuss some important issues that can arise during the implementation of the proposed SLAT-BP algorithm.

- *Complexity of the algorithm:* The complexity of the SLAT-BP algorithm at time t is $\mathcal{O}(|G_t^s| N_c^2)$, since the message computations dominates the cost. This complexity is significantly less than that of naive marginalization of (10) which would require $\mathcal{O}(N_c^{N_s+t-1})$ operations at time instant t. Although this is a significant reduction, the complexity is still high if there are many cells. The complexity can be further reduced by considering only beliefs with a probability larger than a predefined *belief threshold* ϵ_M ($\epsilon_M < 1$). For example, in (16), only cells c, which satisfy the following constraint:

$$M_{n,t-1}(\mathbf{l}_c)/\sum_{c'=1...N_c} M_{n,t-1}(\mathbf{l}_{c'}) > \epsilon_M/N_c, \quad (24)$$

should be considered for summation. An analog constraint is then used for (17) and (19). Denoting the number of these cells by $N_{\mathbf{z}_{n,t}}^\epsilon$ and $N_{\mathbf{x}_t}^\epsilon$, and $N_t^\epsilon = \max(\max_{n\in G_t^s}(N_{\mathbf{z}_{n,t}}^\epsilon), N_{\mathbf{x}_t}^\epsilon)$, the complexity at time t is reduced to $\mathcal{O}(|G_t^s| N_c N_t^\epsilon)$.

- *Non-synchronized measurements:* In Section II-B, we assumed that the IMU is configured to operate at the same rate as the sensors (reporting every T_s second). In practice, the rate of the IMU may be much higher than that of the sensors. The proposed message-passing algorithm can be easily adapted to a situation where the rates are different. Assuming that the algorithm operates at the IMU rate and that we do not know the rate of the sensors' measurements, we just need to do following at each time slot: i) if the sensors' measurements are available, we run the algorithm in Section III-B, and ii) if the sensors' measurements are unavailable, simply exclude all sensor-to-target messages (i.e., $m_{(n,t)\to t}(\mathbf{x}_t) = 1$). In other words, we run simultaneous localization and dead reckoning in all time slots in which only the IMU measurements are available.

- *Routing data to the FCs:* All collected measurements should be routed to the FCs as soon as they become available. Although there are many well-known routing protocols [32], we recommend a hybrid system based on a *leaky-feeder system* (LFS) and a *wireless mesh network* (WMN), similar to one installed at a coal mine in West Virginia, US [7]. LFS consists of a coaxial-type cable, which emits and receives radio waves (i.e, it behaves as a distributed antenna). It has many power supplies, and a backup battery in an explosion-proof enclosure. Therefore, all data transmitted by the sensors or the target, will be available to FCs using a one-hop communication link and without any routing protocol. Since LFS typically cannot provide coverage all over the deployment area, it should be complemented with a WMN, which should consist of the subset of WSN not in the vicinity of the LFS cables. In the WMN, sensors communicate in a multi-hop fashion using an optimal path, computed in real-time. If one or a few sensors fails, the system simply recomputes the path. Therefore, this system is capable of routing the data as long as there is one path between a sensor and a FC or LFS cable.

- *Online calibration:* Although the measurement model in (6) provides robustness against dynamic obstacles, it will not be good enough if a permanent change is made to the environment (e.g., new pillars are formed, or the tunnels are extended). In that case, it would be necessary to repeat the calibration (especially, to re-estimate the GM parameters for the NLOS error model), which is a cumbersome task. An alternative, and preferable option, is to update the measurement model online using already deployed sensors. This is feasible since the fusion center knows the current estimates of the sensors' positions, and consequently all inter-sensor distances. The additional requirement is that the sensors are randomly deployed so that they can provide sufficient statistics for parameter estimation. This calibration should be done periodically (e.g., once per day), or manually triggered once some change in the deployment area is reported.

IV. NUMERICAL RESULTS

In this section, we analyze the accuracy and the robustness of the proposed approach using ray-tracing simulation.

A. TOA error modelling using ray-tracing

For TOA error modelling, we decided to use REMCOM's Wireless InSite ray-tracing simulator [33], [34]. Wireless InSite is a flexible, powerful tool for accurately predicting the effects of the environment on the propagation of electromagnetic waves. It models the physical characteristics of the environment (including the effects of terrain, urban buildings and tunnel features), performs the electromagnetic calculations, and evaluates the signal propagation characteristics. The calculations are made by shooting rays from the transmitters, and propagating them through the environment, until they arrive at the receivers. The rays interact with environmental features via: *reflections* at object faces, *diffraction* around objects, and *transmission* (penetration) through objects. Wireless InSite can provide quantities such as electric and magnetic field strength, received signal strength, time of arrival, path-loss, delay spread, direction of arrival, impulse response and power delay profile.

We designed an artificial mine tunnel using Wireless InSite (see Fig. 3a), by creating many small pieces, and connecting them together. The dimensions of the tunnel are similar to a real mine tunnel [8], and each piece contains expected irregularities. We chose concrete as material for the tunnel

(a)

(b)
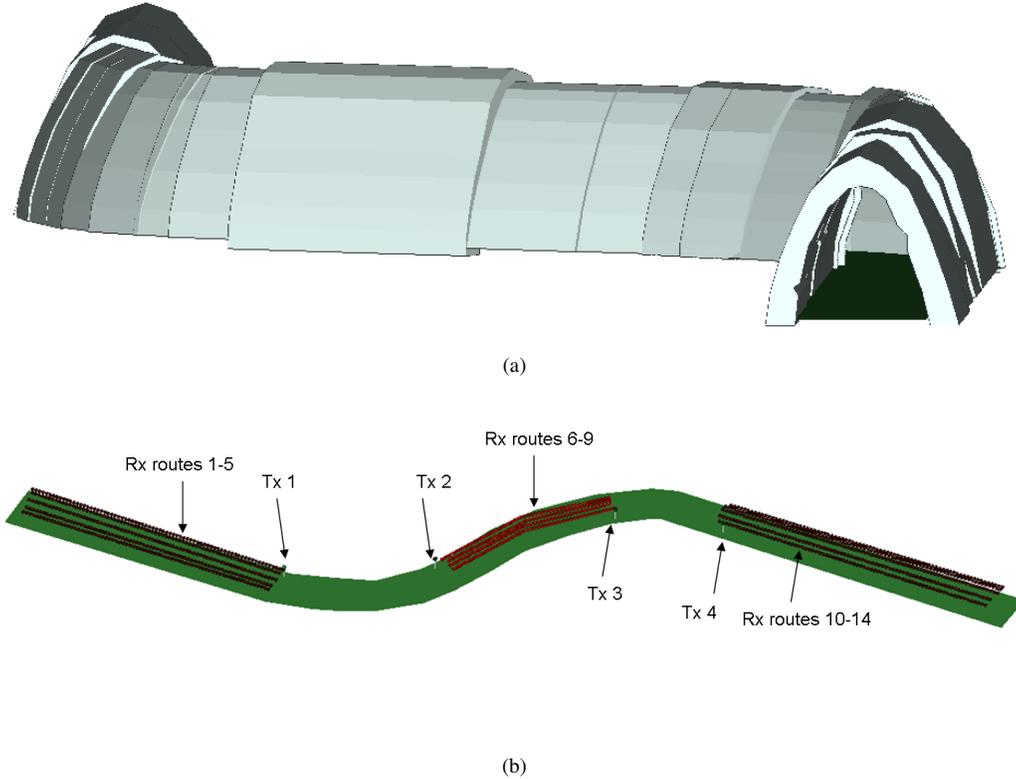
Fig. 3: Illustrations of the designed tunnel in Wireless InSite: (a) 3D illustration, and (b) deployment of Tx/Rx through the tunnel. The dimensions of the tunnel are approx. 110m (length) x 5m (width) x 5m (height) (corresponding to example in Fig. 1a).

walls, which corresponds to the areas in the mine reinforced to increase stability. Then, we deployed transmitters (Tx) and receivers (Rx) routes along the tunnel, as shown in Fig. 3b. Our goal is to obtain TOA at each receiver, which has NLOS to the transmitter caused by the tunnel walls. LOS links are not considered since they provide ground-true estimates (the TOA estimates from Wireless InSite do not include other sources of errors, such as thermal noise, or limited bandwidth). Finally, we set the parameters[6] as shown in Table I. We chose the Full-3D Shooting and Bouncing Ray (SBR) method, which includes the effect of reflections, transmissions, and diffractions on the electric field in 3D environment, without any restriction on the object shapes (more details about this method, and possible alternatives can be found in [34, Chapter 15]). Regarding maximum number of reflections, transmissions, and diffractions, taking into account the analyses in [21], we chose the values which provide a good trade-off between the performance and complexity (i.e., any further increase of these values would lead to a negligible difference in the results, while the computational time would grow dramatically). We used short-dipole antennas with vertical polarization in order to ensure a near-omni-directional radiation pattern. Finally, the values for input power, transmission line loss, and received power threshold were chosen to ensure sufficient range for ray propagation.

From the simulation, we collected 1130 impulse responses,

TABLE I: Main parameters used for Wireless InSite simulations

Ray-tracing method	Full-3D SBR
Antenna (Tx/Rx)	short dipole
Polarization (Tx/Rx)	vertical
Relative permittivity of the wall	7
Conductivity of the wall	$0.015 \text{ m}^{-1}\Omega^{-1}$
Waveform (Tx/Rx)	sinusoidal
Central frequency	2.4 GHz
Input power (Tx)	15 dBm
Received Power Threshold (Rx)	-110 dBm
Transmission line loss (Tx/Rx)	0 dB
Altitude of antenna (Tx/Rx)	1.3 m
Maximum number of reflections	8
Maximum number of transmissions	1
Maximum number of diffractions	1

and obtained the same number of TOA samples (in the absence of noise, the threshold for TOA estimation is set to zero). The samples were used to find the distance estimates, and the samples of the ranging error (the difference between the true and the estimated distance), in order to create a GM model. Observing the histogram in Fig. 4, we can see that the error is positive (as expected in the absence of the other noise), and that it can be well approximated with 5 modes. Moreover, we can see that the error is not exponentially distributed, which stands in contrast to results [35], [36] for other environments. Therefore, we set the number of GM components to $N_M = 5$,

---

[6]Many other parameters (see [34]), not shown in Table I, are kept to default values since they are irrelevant for this analysis.
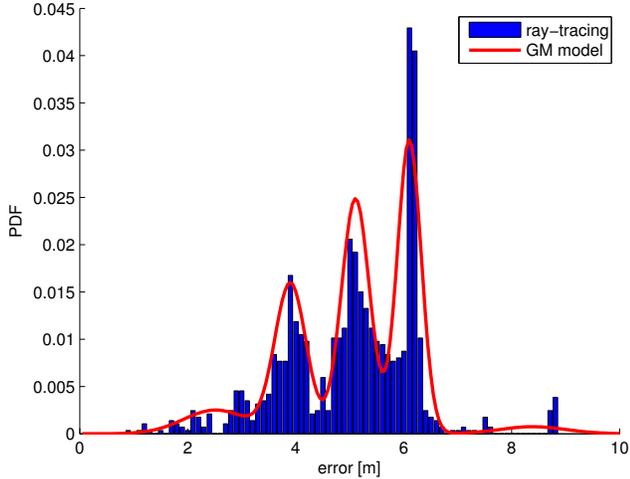
Fig. 4: Histogram of the NLOS ranging error, and corresponding Gaussian mixture model.

TABLE II: Summary of parameters used in simulations

Parameter	Value
number of cells (N_c)	44
number of sensors (N_s)	25
number of time slots (N_T)	40
sampling interval (T_s)	1 s
std. deviation of sensors' positions (σ_S)	6 m
prob. of NLOS (tunnel wall) (P_{NLOS})	0.17
prob. of NLOS (obstacle) (P_{OBS})	0.03
std. deviation of IMU noise (σ_u)	0.5 m/s
std. deviation of LOS noise ($\sigma_{w,0}$)	1 m
sensing radius (d_{TH})	30 m
maximum distance error ($D_{\max}$)	d_{TH}
quantization noise (D)	5 m
kNN parameter (k)	2
belief threshold (ϵ_M)	0.05
number of Monte Carlo runs (N_{MC})	100

and run the k-means algorithm to obtain its parameters.[7]

According to Fig. 4, this model provides a good (but not perfect) approximation of the real PDF. We did not include other error sources in the NLOS model, since they typically are negligible compared to the multipath error (recall that the signal-to-noise-ratio is very high for short-range sensing, and that we target WB/UWB signals). For the LOS error model, we did not analyze the precise impact of the transmit power, bandwidth and other sources of the error. We simply assumed that it can be well approximated with a zero-mean Gaussian distribution with a sufficiently large standard deviation $\sigma_{w,0} = c \cdot 3.33$ ns $= 1$ m.

Finally, we considered a scenario with the same deployment as in Fig. 3b, but with different types of the obstacles (objects created of metal, water, sand, etc.) placed in front of the transmitters. We found that the bias caused by these obstacles is very uncertain, so our assumption that it is uniformly distributed (as a least informative option) is reasonable. Assuming that these obstacles will block the LOS path in 3% of the cases, we increased the size of the NLOS database to 1164 samples by adding (randomly chosen) samples from the NLOS scenarios with obstacles. This database, along with the GM model, will be used in the next sections as an input to the performance analysis simulations.

B. Simulation Setup

We considered the tunnel in Fig. 3a divided in $N_c = 44$ cells, and with $N_s = 25$ sensors randomly deployed in these cells. The sensors' priors were given by $p_{n,0}(\mathbf{z}_{n,0}) = \mathcal{N}(\mathbf{z}_{n,0}; \mathbf{l}_n, \Sigma_S)$ where $\mathbf{l}_n$ ($n = 1, \ldots, N_s$) is the reported location provided by the deployment team, and $\Sigma_S = \mathrm{diag}(\sigma_S^2, \sigma_S^2, \sigma_S^2)$ ($\sigma_S = 6$ m) is the empirical measure of the precision of the sensor placement. We assume that we know the initial cell of the target, i.e., $p_0(\mathbf{x}_0) = \delta(\mathbf{x}_0 - \mathbf{l}_1)$. There are $N_T = 40$ time slots, and the sampling interval is

[7]The obtained parameters are not provided since they are valid only for this, artificially generated, tunnel. For other environments, the error modelling should be repeated.

$T_s = 1$ s. The target position at time t is generated using the following mobility model:

$$\mathbf{x}_t = \begin{cases} \mathbf{l}_{2t+\eta}, & \text{for } 1 \le t \le N_T/2 + 1, \\ \mathbf{l}_{2(N_c - 1 - t) + \eta}, & \text{for } N_T/2 + 2 \le t \le N_T \end{cases}, \quad (25)$$

where η is a random integer between -1 and 1 (i.e., $\eta \sim \mathrm{Unif}\{-1, 0, 1\}$), which adds uncertainty to the mobility of the target. This model assumes that the target is moving forward through the tunnel during the first half of the time, and then going backward until the end of the period. Recall that this model is not known to our algorithm, but the target's velocity is measured by the IMU.

We set the remaining parameters for the measurement noise as follows: $P_{\mathrm{NLOS}} = 0.17$, $P_{\mathrm{OBS}} = 0.03$, and $\sigma_u = 0.5$ m/s. The measurements are generated using these models, except in the NLOS case, in which we pick a randomly-chosen sample from the ray-tracing database. To take into account the cell size, we set the quantization noise parameter to $D = 5$ m. Moreover, the sensing radius was set to $d_{\mathrm{TH}} = 30$ m, the maximum distance error to $D_{\max} = d_{\mathrm{TH}} = 30$ m, the parameter for kNN estimation to $k = 2$, and the belief threshold to $\epsilon_M = 0.05$. All results were averaged over at least $N_{\mathrm{MC}} = 100$ Monte Carlo runs. In each run, we generated different observations, target tracks and sensor positions.

All parameters, summarized in Table II, were used unless otherwise stated in the following text.

C. Performance Analysis

Our goal is to analyze the accuracy of the proposed SLAT and compare it with corresponding tracking and localization algorithms (defined in Section III-B). The target's (sensor's) position error is defined as the Euclidean distance between the position of the true and the estimated cell of the target (sensor).

We first analyze the root-mean-square error (RMSE) as a function of time, as shown in Fig. 5. Regarding the estimates of the target positions (Fig. 5a), SLAT provides the best performance, followed by tracking, and localization which provides the worst performance. The difference between
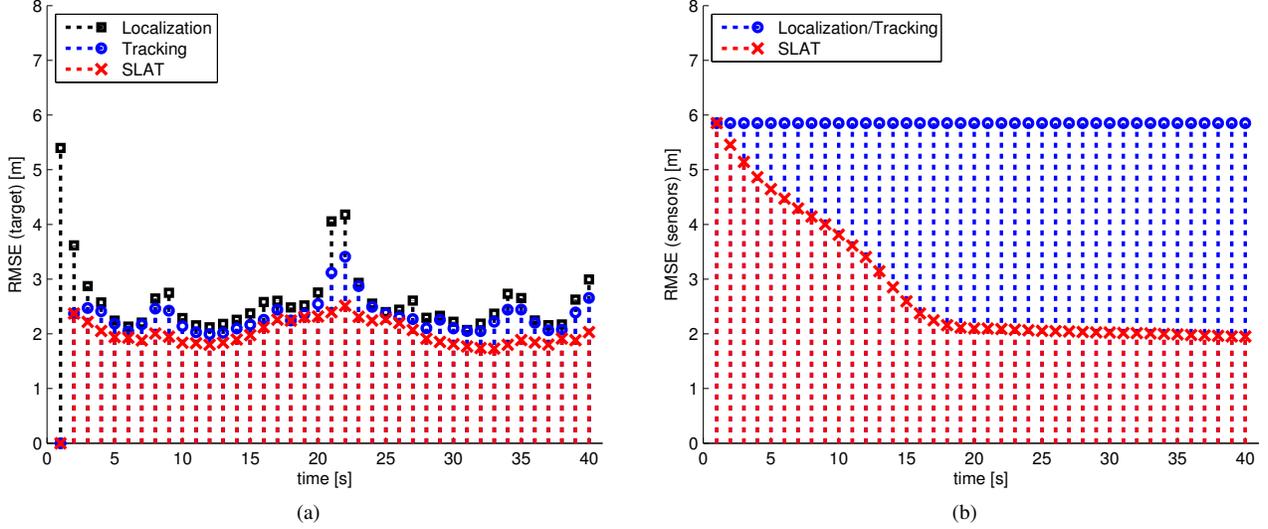
Fig. 5: RMSE as a function of time for (a) the target's position error, and (b) the sensors' position error. For the sensors, there is an additional averaging over all sensors' position errors.
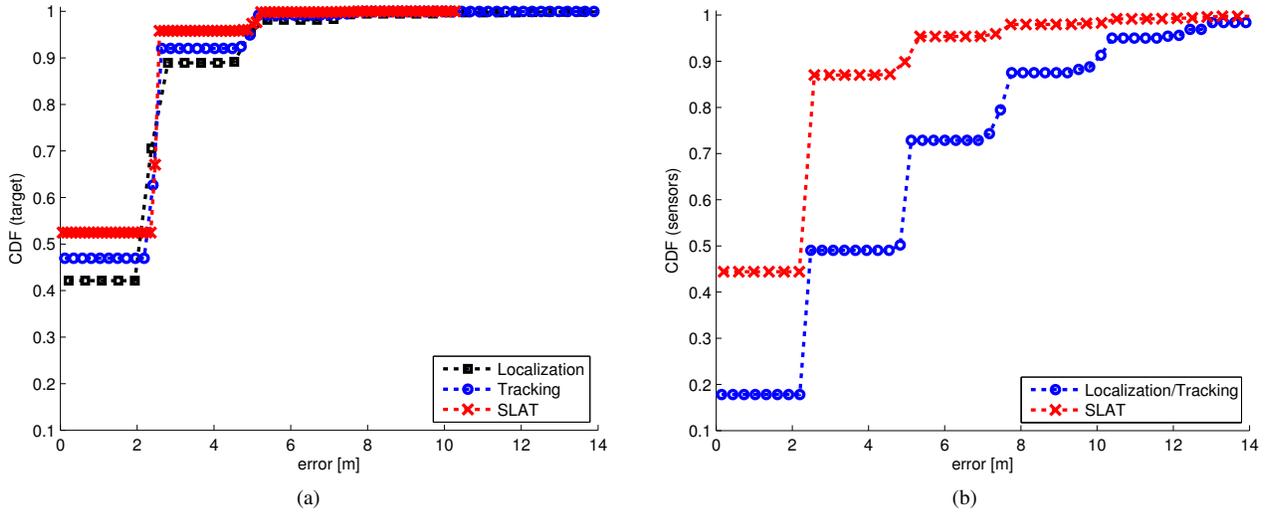
Fig. 6: CDF of (a) the target's position error, and (b) the sensors' position error.

tracking and localization is caused by information from the IMU. The SLAT performs better than tracking because the target uses the information from improved sensors' PMFs, while only the information from initial sensors' PMFs in the tracking algorithm. We also note that the performance of all algorithms is the worst when the target is close to the tunnel edges ($t = 1, 21, 22, 40$). This behavior is caused by the small number of the sensors in proximity of the target. Regarding the estimates of the sensors' positions (Fig. 5b), the SLAT consistently improves their estimates comparing with localization/tracking which do not update the sensors' PMFs. This can be explained by the fact that the state of the same variable (in contrast to tracking) is estimated in each time slot. However, the RMSE will not converge to zero because the NLOS measurement model is not fully consistent with the measurements (see Fig. 4).

To draw more precise conclusions, we analyze the cumulative distribution function (CDF)[8] of the target's and the sensors' position error in Fig. 6. We can see in Fig. 6a that the localization, tracking and SLAT algorithms correctly estimate the target's cell in about 42%, 46%, and 53% of the considered tests, respectively. Regarding the sensor position estimates (Fig. 6b), the localization/tracking algorithms correctly estimate the sensors' cells in only 18% of the tests, while SLAT does so in about 45% of the tests. More importantly, the 95th percentile error of the SLAT (for both the sensors' and target's RMSE) is at most the half of the corresponding localization/tracking algorithms. Therefore, we can conclude that SLAT outperforms tracking/localization in terms of both the target's and the sensors' error. In both cases,

[8]The CDF contains discrete steps due to the finite number of possible error values.
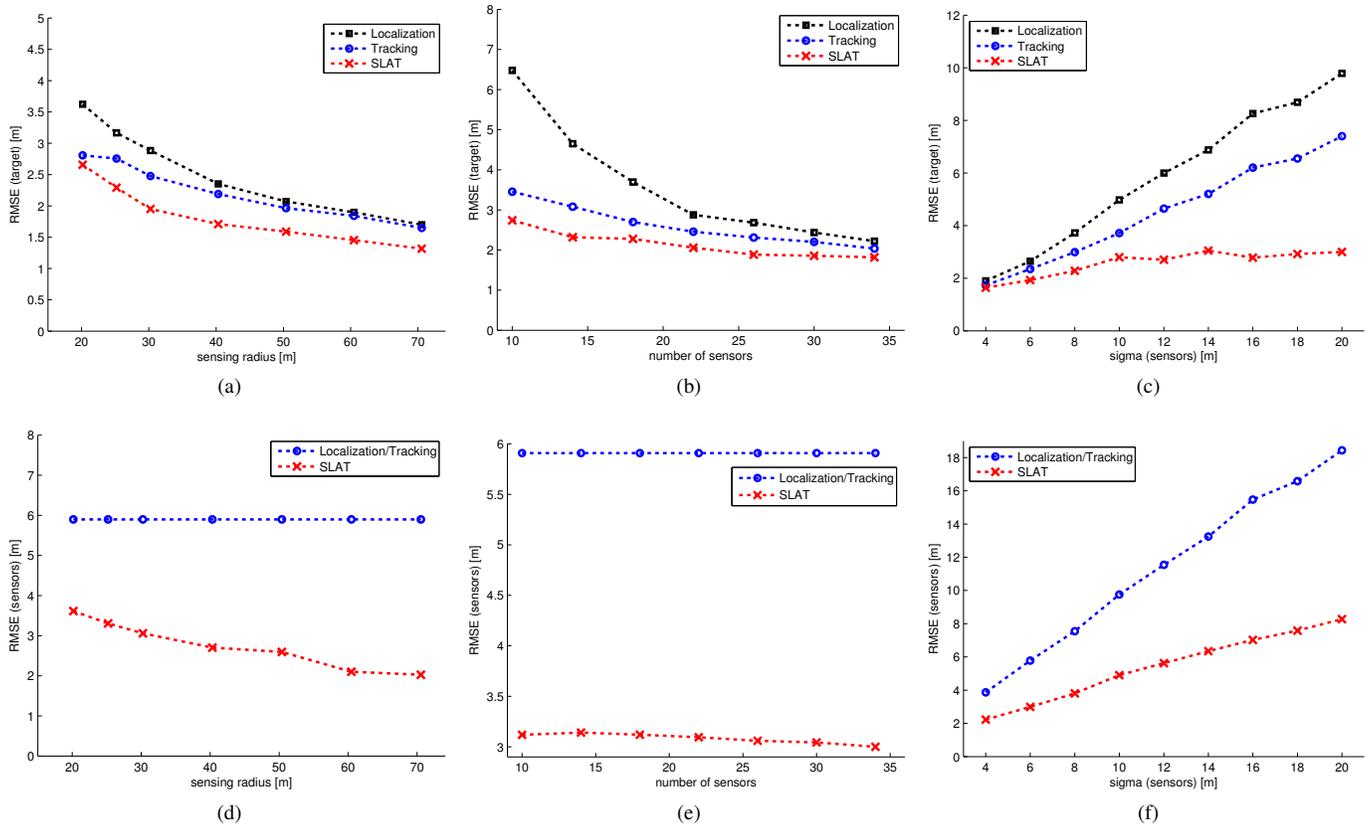
Fig. 7: RMSE as a function of: (a), (d) sensing radius, (b), (e) number of sensors, and (c), (f) standard deviation of the sensors position. Plots with RMSE of the target's position error are shown in the first row, while plots with RMSE of the sensors' positions error are shown in the second row.

SLAT is consistently (at any percentile), better than the other two algorithms. It is worth noting that these conclusions are consistent with our previous results [1] based on models from the CANMET mine.

We now analyze the effect of the different parameters on the performance. The results are shown in Fig. 7. We make the following observations:

- *Effect of varying the sensing radius (Figs. 7a and 7d):* Increasing the sensing radius consistently leads to lower RMSE of the target position estimates (for any of the considered algorithms), and of the sensor position estimates (only for SLAT). This behavior is expected since a higher radius allows more sensors to perform measurements, which are then used to update both the target's and the sensors' positions. However, recall that the sensing radius can be only increased up to a limit defined by the maximum sensing threshold for which the measurement model is valid. We also note that the difference between the tracking and localization algorithms is decreasing with an increasing sensing radius. This can be explained by the fact that the IMU provides a smaller proportion of the total information when more sensors provide measurements.

- *Effect of varying the number of sensors (Figs. 7b and 7e):* Increasing the number of deployed sensors will obviously lead to better performance of all considered algorithms. Thanks to the measurements from the IMU, the sensitivity

is not significant for the target position estimates of the tracking and SLAT algorithms. Regarding the sensors' SLAT estimates, the sensitivity is extremely low because each sensor receives the same number of messages from the target regardless of the number of sensors (see Fig. 2). Nevertheless, these messages are more informative since they are functions of improved target's beliefs. Finally, we can conclude that the SLAT algorithm can be used with a lower density of deployed sensors than the corresponding localization/tracking algorithms (e.g., if an RMSE of 3 m is acceptable, we need 10 sensors in case of SLAT, 15 sensors in case of tracking, and 22 sensors in case of localization). This is especially important if the sensors are expensive.

- *Effect of varying the standard deviation of the sensors' positions (Figs. 7c Fig. 7f):* In all analyzed cases, the RMSE is nearly a linear function of σ_s. However, the most important difference between the algorithms is the slope of the curve, which is the lowest in case of SLAT for both the sensor and the target position estimates. That basically means that SLAT is the most useful in scenarios in which the sensors are very imprecisely deployed, or if most of the sensors drop far from their original positions (which may happen in the aftermaths of accidents).

Finally, we analyze the robustness of the SLAT algorithm against outliers in the distance measurements. Therefore, we assume that the measured distance is contaminated with an
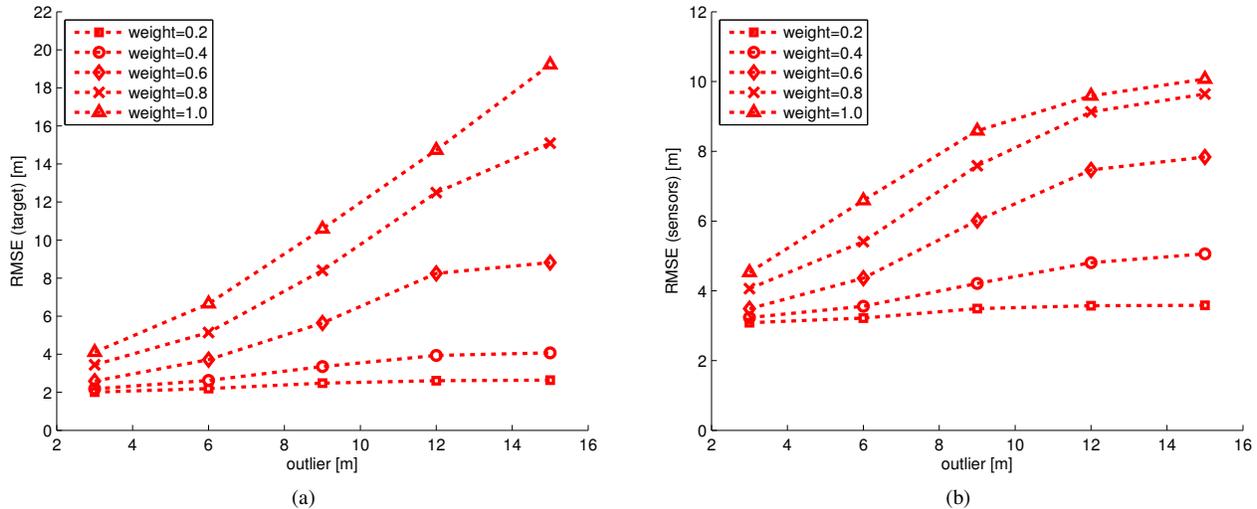
Fig. 8: RMSE of the SLAT algorithm as a function of parameters of the contamination noise for (a) the target's position error, and (b) the sensors' position error.

outlier d_o, with weight P_o. In other words, the contaminated measured distance is given by $d_{\mathrm{o},t,n} = d_{t,n} + n_\mathrm{o}$ where n_o is contamination noise distributed as:

$$n_\mathrm{o} \sim (1 - P_\mathrm{o})\delta(n_\mathrm{o}) + P_\mathrm{o}\delta(n_\mathrm{o} - d_\mathrm{o}). \qquad (26)$$

Note that the presence of these outliers is unknown to the SLAT algorithm, so it is not part of the model in (5). These outliers may be caused by additional NLOS conditions, interference, or other sources of errors. According to the results, shown in Fig. 8, we find the following: i) higher values of P_o and/or d_o would lead to increased RMSE, ii) small value of P_o would not lead to performance degradation regardless of the value of d_o, iii) the sensor position estimates are less sensitive to outliers than the target position estimates, and iv) for $P_\mathrm{o} > 0.4$ and large values of d_o, the sensor position estimates are worse than the corresponding tracking/localization estimates, in which the RMSE is equal to 5.9 m (see Fig. 7e) even in the presence of outliers. In principle, the algorithm is very robust against outliers, especially for small weights P_o, and this is achieved thanks to the uniformly distributed tail in the distribution of the measurement noise (see (6)). The unique problem is that SLAT should not be used if these weights are large, but this situation should not be expected in reality. If the most of the links are contaminated with an outlier (this situation can be detected by comparing tracking and SLAT), it is necessary to perform re-calibration, as explained in Section III-C.

D. Experimental Example

We consider a small part of the CANMET mine [6] to test the performance of our algorithm. The tunnel is divided into $N_c = 14$ (non-equal) cells, and in each of them there is one sensor (i.e., $N_s = N_c$). The sensors' priors are given by $p_{n,0}(\mathbf{z}_{n,0}) = \mathcal{N}(\mathbf{z}_{n,0}; \mathbf{l}_n, \Sigma_S)$ where $\mathbf{l}_n$ ($n = 1, \ldots, N_s$) is the expected location of the sensors, and $\Sigma_S = \mathrm{diag}(25\,\mathrm{m}^2, 25\,\mathrm{m}^2, 0)$. There are $N_T = 30$ time slots,

the sampling interval is $T_s = 1$ s, and the quantization noise parameter is $D = 6$ m. Taking the results from [6], the measurement noise follows a Gaussian distribution for LOS, and a Weibull distribution for NLOS. More details about the considered scenario, and all other parameters, are available in [1].

We analyze the CDF of the target's and the sensors' position error, shown in Fig. 9. As we can see, SLAT is consistently (at any percentile) more accurate than corresponding tracking and localization algorithms, which is consistent with our results based on ray-tracing (Fig. 6). One minor difference is that CDF is smoother than in Fig. 6, which is caused by variable cell sizes.

E. Other solutions in literature

Although not for the same type of the environment, there are other solutions in literature that try to address a similar problem. They are mainly based on fingerprinting techniques [8], [37], [38], and SLAT algorithms over continuous space [39]–[41]. In general, fingerprinting is able to outperform our proposed SLAT method, assuming that enough fingerprints are available. However, this approach would require an exhaustive calibration that sometimes may be infeasible. An additional problem of fingerprinting is that sensors may fall from the walls during the training phase (which have to be repeated very frequently), making a subset of the fingerprints invalid. Regarding other SLAT algorithms, to the best of our knowledge, none of them are adapted to non-Gaussian measurement models and for confined areas. For example, the SLAT methods in [39]–[41] provide the posterior in Gaussian form, and use an unrestricted continuous space. Consequently, they would either fail in the presence of high levels of NLOS bias or provide invalid estimates (e.g., behind the walls).
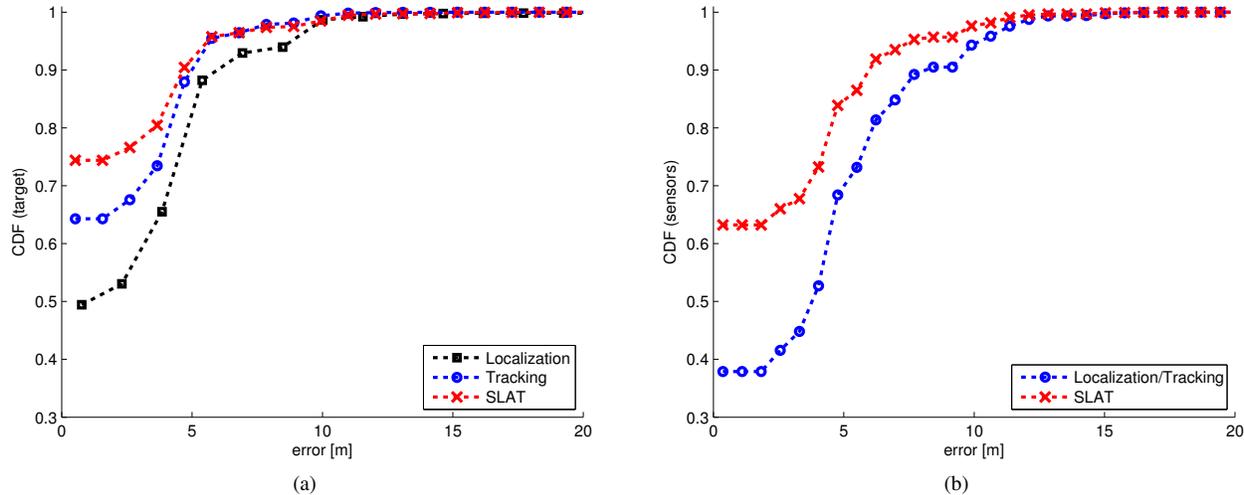
Fig. 9: CDF of (a) the target's position error, and (b) the sensors' position error. The results are based on models from the CANMET mine.

V. CONCLUSIONS AND FUTURE WORK

We presented a novel approach to target tracking in confined environments in the presence of uncertain sensors positions. It is based on a SLAT-BP algorithm, which can simultaneously, and in real-time, refine the sensors' positions and track the target. This algorithm can: i) efficiently solve high-dimensional problems, ii) handle all non-Gaussian uncertainties, and iii) provide only valid position estimates thanks to the use of cells with predefined positions. According to our simulation results, both the sensor and the target position estimates are improved even after a very short tracking period. Moreover, SLAT-BP can be used with a very low density of deployed sensors, and can keep performing well even if most of the sensors drop far from their original position. Finally, the algorithm is also very robust against outliers as long as they appear with reasonable probabilities.

By no means the study in this paper provides the solutions for all problems in confined environments. There remain many open directions for future work. First, it would be useful to provide a method that learns and adapts the ranging distribution online. As a result, the SLAT algorithm can be re-used for a wide variety of environments. Second, a distributed implementation of the SLAT algorithm, in which the sensors estimate the target's position by cooperating only with the neighboring sensors, may be useful to increase the scalability and robustness. Finally, cooperative infrastructure-free self-localization algorithm would be of high interest for search-and-rescue operations.

ACKNOWLEDGMENT

REFERENCES

[1] V. Savic, H. Wymeersch, and E. G. Larsson, "Simultaneous sensor localization and target tracking in mine tunnels," in *IEEE Proc. of 13th Conf. on Information Fusion (FUSION)*, July 2013.

[2] P. Stenumgaard, J. Chilo, J. Ferrer-Coll, and P. Ängskog, "Challenges and conditions for wireless machine-to-machine communications in industrial environments," *IEEE Communications Magazine*, vol. 51, pp. 187–192, June 2013.

[3] P. Misra, S. Kanhere, D. Ostry, and S. Jha, "Safety assurance and rescue communication systems in high-stress environments: A mining case study," *IEEE Communications Magazine*, vol. 48, pp. 66 –73, April 2010.

[4] S. Yarkan, S. Guzelgoz, H. Arslan, and R. Murphy, "Underground mine communications: A survey," *IEEE Communications Surveys Tutorials*, vol. 11, no. 3, pp. 125 –142, 2009.

[5] "Mine Improvement and New Emergency Response (MINER) Act." United States Public Laws, PL 109-236, June 2006.

[6] A. Chehri, P. Fortier, and P. M. Tardif, "UWB-based sensor networks for localization in mining environments," *Ad Hoc Networks*, vol. 7, pp. 987–1000, July 2009.

[7] T. Novak, D. P. Snyder, and J. L. Kohler, "Post-accident mine communications and tracking system," in *IEEE Industry Applications Society Annual Meeting*, pp. 1–8, Oct. 2009.

[8] C. Nerguizian, C. Despins, and S. Affes, "Geolocation in mines with an impulse response fingerprinting technique and neural networks," *IEEE Transactions on Wireless Communications*, vol. 5, pp. 603 – 611, March 2006.

[9] S. Dayekh, S. Affes, N. Kandil, and C. Nerguizian, "Cooperative localization in mines using fingerprinting and neural networks," in *Proc. of IEEE WCNC*, pp. 1 –6, April 2010.

[10] J.-S. Lim, W.-H. Jang, G.-W. Yoon, and D.-S. Han, "Radio map update automation for WiFi positioning systems," *IEEE Communications Letters*, vol. 17, pp. 693–696, April 2013.

[11] I. Lee, G. Yoon, and D. Han, "Nerimi: WiFi-based subway navigation system," in *IEEE Proc. of Intl. Microwave Workshop Series on Intelligent Radio for Future Personal Terminals*, 2011.

[12] A. Chehri, P. Fortier, and P. M. Tardif, "Characterization of the ultra-wideband channel in confined environments with diffracting rough surfaces," *Wireless Personal Communications (Springer)*, vol. 62, pp. 859–877, Feb. 2012.

[13] M. Li and Y. Liu, "Underground coal mine monitoring with wireless sensor networks," *ACM Trans. Sensor Networks*, vol. 5, pp. 1–29, April 2009.

[14] M. Hedley and I. Gipps, "Accurate wireless tracking for underground mining," in *Proc. of ICC Workshop on Advances in Network Localization and Navigation (ANLN)*, June 2013.

[15] V. Savic, J. Ferrer-Coll, P. Angskog, J. Chilo, P. Stenumgaard, and E. G. Larsson, "Measurement analysis and channel modeling for TOA-based ranging in tunnels," *IEEE Trans. on Wireless Communications*, vol. 14, pp. 456–467, Jan. 2015.

[16] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, III, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, pp. 54–69, July 2005.

[17] H. Wymeersch, F. Penna, and V. Savic, "Uniformly reweighted belief propagation for estimation and detection in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 11, pp. 1587–1595, April 2012.

[18] M. Cetin, L. Chen, I. Fisher, J. W., A. T. Ihler, R. L. Moses, M. J. Wainwright, and A. S. Willsky, "Distributed fusion in sensor networks," *IEEE Signal Processing Magazine*, vol. 23, pp. 42–55, July 2006.

[19] Y. Jin, W.-S. Soh, M. Motani, and W.-C. Wong, "A robust indoor pedestrian tracking system with sparse infrastructure support," *IEEE Trans. on Mobile Computing*, vol. 12, pp. 1392–1403, July 2013.

[20] M. Boutin, A. Benzakour, C. Despins, and S. Affes, "Radio wave characterization and modeling in underground mine tunnels," *IEEE Transactions on Antennas and Propagation*, vol. 56, pp. 540 –549, Feb. 2008.

[21] S. Chuasomboon, "A comparison of ranging and localization techniques in indoor, urban and tunnel environments," Master's thesis, Linkoping University, Sweden, June 2013.

[22] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Z. Win, "Ranging with ultrawide bandwidth signals in multipath environments," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 404–426, 2009.

[23] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE Signal Processing Magazine*, vol. 22, pp. 70–84, July 2005.

[24] S. Marano, W. Gifford, H. Wymeersch, and M. Win, "NLOS identification and mitigation for localization based on UWB experimental data," *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 1026–1035, Sept. 2010.

[25] H. Miao, K. Yu, and M. Juntti, "Positioning for NLOS propagation: Algorithm derivations and Cramer-Rao bounds," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 2568–2580, Sept. 2007.

[26] C. M. Bishop, *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[27] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. of the IEEE*, vol. 97, no. 2, pp. 427–450, 2009.

[28] M. Wainwright and M. Jordan, *Graphical models, exponential families, and variational inference*. Now Publishers Inc., 2008.

[29] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. of the IEEE*, vol. 77, pp. 257–286, Feb. 1989.

[30] A. T. Ihler, J. W. I. Fisher, R. L. Moses, and A. S. Willsky, "Non-parametric belief propagation for self-localization of sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 809–819, April 2005.

[31] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: indoor location sensing using active RFID," in *Proc. of 1st IEEE Int. Conf. on Pervasive Computing and Communications*, pp. 407–415, 2003.

[32] E. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications*, vol. 6, pp. 46 –55, apr 1999.

[33] REMCOM, "Wireless InSite® - Wireless EM Propagation Software." http://www.remcom.com/wireless-insite.

[34] REMCOM, "Wireless InSite® Reference Manual." Version 2.6.3, April 2012.

[35] P.-C. Chen, "A non-line-of-sight error mitigation algorithm in location estimation," in *IEEE Proc. of WCNC*, pp. 316–320, 1999.

[36] S. Venkatesh and R. Buehrer, "NLOS mitigation using linear programming in ultrawideband location-aware networks," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 3182–3198, Sept. 2007.

[37] M. Bshara, U. Orguner, F. Gustafsson, and L. V. Biesen, "Fingerprinting localization in wireless networks based on received-signal-strength measurements: A case study on WiMAX networks," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 283–294, Jan. 2010.

[38] S. C. Ergen, H. S. Tetikol, M. Kontik, R. Sevlian, R. Rajagopal, and P. Varaiya, "RSSI-fingerprinting-based mobile phone localization with route constraints," *IEEE Transactions on Vehicular Technology*, vol. 63, pp. 423–428, Jan. 2014.

[39] C. Taylor, A. Rahimi, J. Bachrach, H. Shrobe, and A. Grue, "Simultaneous localization, calibration, and tracking in an ad hoc sensor network," in *Proc. of IEEE/ACM Int. Conf. on Information Processing in Sensor Networks (IPSN)*, pp. 27–33, April 2006.

[40] J. Teng, H. Snoussi, C. Richard, and R. Zhou, "Distributed variational filtering for simultaneous sensor localization and target tracking in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 61, pp. 2305 –2318, June 2012.

[41] N. Kantas, S. Singh, and A. Doucet, "Distributed maximum likelihood for simultaneous self-localization and tracking in sensor networks," *IEEE Transactions on Signal Processing*, vol. 60, pp. 5038–5047, Oct. 2012.

Vladimir Savic received the Dipl.Ing. degree in electrical engineering from the University of Belgrade, Serbia, in 2006, and the M.Sc. and Ph.D. degrees in communications technologies and systems from the Universidad Politecnica de Madrid, Spain, in 2009 and 2012, respectively. He was a Digital IC Design Engineer with Elsys Eastern Europe, Belgrade, from 2006 to 2008. From 2008 to 2012, he was a Research Assistant with the Signal Processing Applications Group, Universidad Politecnica de Madrid. He spent three months as a Visiting Researcher at the Stony Brook University, NY, and four months at the Chalmers University of Technology, Gothenburg, Sweden. In 2012, he joined the Communication Systems (CommSys) Division, Linköping University, Sweden, as a Postdoctoral Researcher. His research interests include localization and tracking, wireless channel modeling, Bayesian inference, and distributed and cooperative signal processing in wireless networks.

Henk Wymeersch (S'99, M'05) received the Ph.D. degree in Electrical Engineering/Applied Sciences in 2005 from Ghent University, Belgium. He is currently an Associate Professor with the Department of Signals and Systems at Chalmers University of Technology, Sweden. Prior to joining Chalmers, he was a Postdoctoral Associate with the Laboratory for Information and Decision Systems (LIDS) at the Massachusetts Institute of Technology (MIT). He served as Associate Editor for *IEEE Communication Letters* (2009–2013), *IEEE Trans. on Wireless Communications* (2013–present), and *Trans. on Emerging Telecommunications Technologies* (2011–present).

Erik G. Larsson received his Ph.D. degree from Uppsala University, Sweden, in 2002. Since 2007, he is Professor and Head of the Division for Communication Systems in the Department of Electrical Engineering (ISY) at Linköping University (LiU) in Linköping, Sweden. He has previously been Associate Professor (Docent) at the Royal Institute of Technology (KTH) in Stockholm, Sweden, and Assistant Professor at the University of Florida and the George Washington University, USA.

His main professional interests are within the areas of wireless communications and signal processing. He has published some 100 journal papers on these topics, he is co-author of the textbook *Space-Time Block Coding for Wireless Communications* (Cambridge Univ. Press, 2003) and he holds more than 10 issued and many pending patents on wireless technology. He has served as Associate Editor for several major journals, including the *IEEE Trans. on Communications* (2010-2014) and *IEEE Trans. on Signal Processing* (2006-2010). He serves as chair of the IEEE Signal Processing Society SPCOM technical committee in 2015–2016. He also serves as chair of the steering committee for the *IEEE Wireless Communications Letters* in 2014–2015. He is active in conference organization, for example as General Chair of the Asilomar Conference on Signals, Systems and Computers in 2015 (he was Technical Chair in 2012). He received the *IEEE Signal Processing Magazine* Best Column Award twice, in 2012 and 2014.