RF Localization and tracking of mobile nodes in Wireless Sensors Networks: Architectures, Algorithms and Experiments

M. Bertinato, G. Ortolan, F. Maran, R. Marcon, A. Marcassa, F. Zanella, P. Zambotto, L. Schenato, A. Cenedese

DEI, Department of Information Engineering, University of Padua, Italy

Abstract. In this paper we address the problem of localizing, tracking and navigating mobile nodes associated to operators acting in a fixed wireless sensor network (WSN) using only RF information. We propose two alternative and somehow complementary strategies: the first one is based on an empirical map of the Radio Signal Strength (RSS) distribution generated by the WSN and on the stochastic model of the behavior of the mobile nodes, while the second one is based on a maximum likelihood estimator and a radio channel model for the RSS. We compare the two approaches and highlight pros and cons for each of them. Finally, after implementing them into two real-time tracking systems, we analyze their performance on an experimental testbed in an industrial indoor environment.

1 Introduction

Wireless sensor networks (WSNs), large networks of spatially distributed electronic devices (nodes) capable of sensing, computation and wireless communication, are becoming very popular non only within the academic world as a prototype of multiagent system but also in the industry. In fact, they can offer access to an unprecedented quality and quantity of information that can revolutionize our ability in controlling the environment. In particular, location-based applications are among the first and most popular applications of WSNs since they can be employed for tracking enemies in battlefield, locating moving objects in buildings (e.g. warehouses, hospitals), and tracking people inside buildings.

The work in this paper is motivated by one of these applications, i.e. the design of a real-time system that can support fireman rescue squads to locate themselves and to navigate inside a building during emergency scenarios. To improve coordinated searching strategies, there may be a need also to communicate the fireman position to external monitoring centers supervising the operations. To achieve these objectives, we propose to deploy a static wireless sensor network whose nodes are placed in known positions. Each fireman is provided with a pocket-PC, or similar device, with a mobile node that can communicate with the static network. The position of firemen is estimated only by using the radio signals (Received Signal Strength Indication, RSSI, and Link Quality Indicator, LQI) provided by a standard IEEE 802.15.4 radio chip, without resorting to any other special sensor or devices such as IR motion sensors, ultrasound, or directional antennas.

The entire network of both fixed and mobile nodes, realizes a distributed intelligence system, relying on the computational capabilities of the nodes. In

[*] This research is supported by CaRiPaRo Foundation under project "Wise-Wai".

this framework, though, the exact location of each fireman is performed by the pocket-PC rather than by the static network since the former is provided with more computational capabilities to elaborate complex algorithms and at the same time it is scalable, since the static network simply keeps on transmitting their own location[1] regardless of the number of firemen. Nevertheless, we also propose simple localization algorithms that can be performed with the limited computational capabilities of sensor nodes at the price of a lower performance, but can be employed in case of failure of the Pocket-PCs. Each pocket-PC computes it own position and displays it on a screen with a map of the environment similarly to commercial GPS-based navigators. The position is also retransmitted to the static network which routes it back to a gateway and then to the firemen coordination center. Finally, the coordination center can plan a set of way-points for each fireman that are retransmitted back to the pocket-PC for navigation.

In addition, the navigation system must also comply with a number of constraints which are common to WSNs. In fact, it needs to be power efficient if the nodes of the static network are battery-powered, robust to packet drop, real-time, and finally it should maintain an acceptable localization accuracy even in the event of some static nodes failure.

Before proceeding, to avoid confusion, some definitions are in order: we define *localization* as the process of estimating the position of a node, *tracking* as the process of estimating the trajectory of a moving node, possibly adopting a model of object dynamics to reduce localization error, and *navigation* as the process to provide a mobile node with a trajectory or way-points to be followed.

1.1 Previous Work

Any tracking and navigation system is based on some form of localization of the object of interest. The are many different strategies to localize an object. Within the context of WSNs, localization algorithms can be classified according to the parameter used to get relative position information between nodes, as, among others, the *Angle of Arrival* AoA (an estimate of the relative angles between nodes), the *Time of Arrival* ToA (time taken by the radio signal to propagate from one node to another), the *Time Difference of Arrival* TDoA (time interval between the reception of a radio signal and an ultrasound emitted by a beacon), the *Received Signal Strength Indicator* RSSI (an index of received signal power).

Approaches based on the first three quantities require specific devices such as array antennas for the AoA [1], ultrasound for TDoA [2], dedicated HW and SW to maintain nodes synchronization [3], or motion detection sensors such as magnetometers, IR motion sensor [4] and cameras [5]. Although successful implementations of systems based on these approaches exist, they are not widespread because the specific localization hardware is too expensive or fragile in cluttered and dynamic environments.

Differently, RSSI-localization systems are much more popular since most of today's radio chips for WSNs provide it at no extra hardware cost. In an ideal medium with an ideal antenna, there should be an information preserving correspondence between each RSSI value and the relative distance between two nodes; however in an indoor environment multipath fading, reflections, diffraction, interference, and the presence of dynamics highly affect this relation. Although the localization accuracy is poorer than the one achieved with the non-RSSI

[1] or simply node ID if pocket-PCs have a stored map of node IDs and their locations.

localization systems, it is sufficient in many applications such as the one we address. Most of the algorithms coping with the high variability of RSSI measurements can be grouped into two distinct classes that we refer as *RSSI map matching*-based localization and *RSSI channel model*-based localization.

RSSI-map-based localization. This class of algorithms is based on the RSSI-maps generated by each static wireless node. In particular, this map takes into account the location of the static nodes as well as the topology and morphology of the environment including wall and static objects. The most popular tracking systems in this class are the RADAR [6] and MoteTrack [7] tracking systems.

The RSSI-map is obtained either through an analytical model of propagation of the radio signal (like *wall attenuation factor model* [8]) or by a series of measurements. The *analytical* RSSI-map can be provided by standard tools like ray-tracing [9], but it does not take into account small objects like tables and shelves, or dynamics in the environment, thus possibly leading to poor performance. The *empirical* RSSI-map, instead, better represents the real situation, but requires an extensive set of experiments. Moreover, the RSSI-map can be *deterministic* or *stochastic*, the former associating to each point (x, y, z) of the environment a single RSSI value for each node, the latter resorting to a probability distribution of RSSI values. The stochastic RSSI-map is more realistic than the determinist map as it takes into account possible dynamic variations of the environment, at the price of a more complex model.

This RSSI-map is computed off-line for each static node and gives for each point in the environment the corresponding RSSI value or its probabilistic distribution. Position estimate is then calculated on-line by searching for the value nearest to the current measurement in terms of some defined metric. Searching for the best match in the RSSI map can be computationally expensive if the map is finely gridded. To overcome this problem, for example, RADAR adopts a technique called *Nearest Neighbor in Signal Space* (NNSS) to identifies a subset of possible positions so reducing the computation time required for the location estimate. Similarly to RADAR, also MoteTrack adopts the same strategy based on an RSSI-map, called reference signature, but it adds some additional features which reduce estimation errors from about 3[m] for RADAR to 1-1.5[m] and improve robustness and scalability.

Both RADAR and MoteTrack are based on a deterministic model for the RSSI-map. Recently, Morelli et al. [10] [11] proposed a probabilistic approach for modeling the RSSI-map by including not only mean RSSI values but also variance information to take into account the variability of environment. In this framework, the location of the tracked node is obtained by computing the most likely location based on the received measurements using a Bayesian approach. This strategy is more computationally expensive, but provides good performance also in highly cluttered indoor environments.

RSS-channel model based localization. Differently from the previous class of localization algorithms , the algorithms based an a RSS-channel model first try to estimate the relative distance of the moving mode from the static nodes and then they triangulate the moving node location using a geometric approach. The advantage of this strategy is that it does not require any a-priori detailed RSSI-map of the environment and that the localization algorithms are computationally efficient. In particular, the distance-vs-RSSI model is based on a fading-channel model with gaussian noise and the distance is derived using a maximum likelihood estimator (MLE) [12]. Once the relative distances from each anchor node is obtained, there are few geometric approaches that can be used to estimate the location. The triangulation technique calculates the location of the moving target by solving a linear set of equations which results from geometric constraints, i.e. it gives the least square location estimate. This is the technique adopted by the GPS-system [13] and it requires at least three ancor nodes to estimate the location on a plane. Although it is computational attractive, it is quite sensitive to errors on the relative distance estimates, as it is usually the case in WSNs. Another very natural localization strategy, based on maximum ratio combining of measurements [14], is to consider the position of any static node as an estimate of the location of the moving agent. The location of the target is then obtained by a convex weighted combination of the static nodes positions where the weight associate to each static node is proportional to the RSSI-strength or inversely proportional to the estimated distance. This approach is even simpler than triangulation, but gives wrong answers if the target node is close to the border of the WSN or outside, since the location is a convex combination of static nodes location.

Dynamics-based tracking. The previous two classes of algorithms simply provide en estimate of position of a target based on the measurements at some time instant t. However, better estimates can in principle be obtained by using also the past measurements and a stochastic modelization of mobile node dynamics like a Markovian homogeneous first-order process, which gives rise to a random walk. Classical approaches for tracking are based on Kalman filters [15] or more general Bayesian filters like particle filters [11] which act similarly to low-pass filters on the instantaneous position estimates. This dynamics-based tracking pairs particularly well with the stochastic RSSI-map based localization since they are both probabilistic models.

1.2 Contribution

As discussed above, the localization error is not the only important factor that needs to be considered when designing an appropriate tracking systems. Other aspects are important such as number of nodes to achieve a desired localization error, the computational requirements necessary to run the proposed algorithms in real-time, the installation and maintenance costs, the system lifetime, robustness to packetdrop and node failures, and scalability. Therefore, it is necessary to study different architectures and compare them across all their different features. Motivated by these considerations, we propose two different tracking architectures, the first, named *ARIANNA* , which belongs to the RSSI-map based localization systems, and the second *TESEO* , which belongs to the RSSI-channel-model localization systems. In particular, *ARIANNA* uses a stochastic model based on empirical RSSI measurements collected in the real environment similarly to the one proposed in [11], but also models packet dropout and adopts a novel interpolation for the RSSI-map. Differently, *TESEO* does not need any extensive set of experimental measurements but identifies the RSSI-channel parameters based on local measurements of the static nodes, as proposed in [12]. Moreover, it implements three different geometric localization algorithms: the maximum ratio combining (MRC) [14], the least square estimator (LSE) [13], and a novel maximum likelihood (ML) estimator based on gradient descent. These different algorithms present a trade-off between computational complexity and localization error and they are implemented simultaneously to provide different level of robustness to system failures.

We implemented both architectures using commercial-off-the-shelf wireless sensor nodes, hardware and software, and we performed *real-time* tracking experiments in a rather hostile and large scale indoor environment which included large walls, high electromagnetic interference and metallic structures. These experiments provide a fair comparison between two fundamentally different architectures which have different structural characteristic in terms of the factors listed above. In particular, we show that *ARIANNA* is a suitable architecture if HW cost is the most important aspect, but it requires extensive set of measurements before installation and a rather powerful computation unit to track moving objects. Differently, *TESEO* provided smaller localization error estimates, can be installed very rapidly without time-consuming experimental measurements, and requires fewer computational resources to implement the algorithms. However, it requires much larger number of nodes as compared to *ARIANNA* .

In the following of the paper we describe in details *ARIANNA* and *TESEO* architecture and algorithms including some simulations. Then we describe the implementation and real time-time tracking performance of the two architectures in an industrial indoor environment. Finally we provide some considerations on the results and discuss about future opportunities of research.

2 ARIANNA Tracking System

This algorithm is based on the measured RSSI over the Anchor Nodes (ANs)-Mobile Node (MN) links, sensed by the MN; a suitable *a priori map* containing the RSSI distribution over the interested area is built. The MN dynamics is modeled as a Markovian homogeneous first-order process, so allowing Bayesian filtering to estimate and track the MN's position. The WSN we are referring to is a centralized one, since we suppose that the MN is a smart sensor or it is connected to a smarter processing unit (e.g. a Pocket PC). The problem we are considering deals with a single MN, but this approach can be easily extended to a generic case with more than one MN, as described in the Introduction.

2.1 RSSI-map stochastic modeling

The WSN is formed by L ANs and is located in a region $\mathcal{X} \subset \mathbb{R}^2$. Every beacon's position is fixed and known, while at a discrete time $t \in \mathbb{N}$ the MN position $\mathbf{x}_t = [x_{1,t}\ x_{2,t}]'$ has to be estimated. In order to minimize the multipath and shadowing effects, we use a log-normal stochastic model for the RSSI

$$y_l(\mathbf{x}_t) = \overline{y}_l(\mathbf{x}_t) + v_{l,t} \text{ [dBm]}, \tag{1}$$

where $\overline{y}_l(\mathbf{x}_t)$ is deterministic and concerns about the attenuation due to static obstacles and signal propagation, and $v_{l,t} \sim \mathcal{N}\big(0, \sigma_l^2(\mathbf{x}_t)\big)$ indicates the random effects of people and small objects moving through the environment; both $\overline{y}_l(\mathbf{x}_t)$ and $\sigma_l(\mathbf{x}_t)$ are based on empirical data. This model has been validated by ray tracing techniques [8]. Since the MN cannot measure RSSI values out of a specific range $[T_1, T_2]$, weak signals are often ignored by the MN; moreover, due to low fidelity of radio channel and hardware bugs, RSSI measurements are sometimes completely wrong: that's why (1) is too scanty. The following random variable is therefore introduced to account for the packet loss:

$$\gamma_l(\mathbf{x}_t) = \begin{cases} 1 & \text{if } y_l(\mathbf{x}_t) \in [T_1, T_2] \\ 0 & \text{if } y_l(\mathbf{x}_t) \notin [T_1, T_2]. \end{cases}$$

We assume that the arrival process is stationary and i.i.d., and it holds:

$$P[\gamma_l(\mathbf{x}_t) = 1] = \lambda_l(\mathbf{x}_t),\ \ \lambda_l(\mathbf{x}_t) \in [0, 1]\ \forall t, l;$$

hence $\gamma_l(\mathbf{x}_t)$ is a Bernoullian process with success probability $\lambda_l(\mathbf{x}_t)$. A measurement is then given by $\mathbf{y}_t \in \mathbb{R}^L,\ \mathbf{y}_t = [\tilde{y}_{1,t}\ \dots\ \tilde{y}_{L,t}]'$, where

$$\tilde{y}_{l,t} = \begin{cases} y_l(\mathbf{x}_t) & \text{if } \gamma_l(\mathbf{x}_t) = 1 \\ \text{ID} & \text{if } \gamma_l(\mathbf{x}_t) = 0, \end{cases} \tag{2}$$

and ID is an appropriate flag corresponding to the packet loss.

The MN dynamics is assumed to be a 2-D random walk

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{v}_t, \tag{3}$$

where $\mathbf{v}_t$ denotes the driving process. The probabilistic model of the state evolution accords to $p(\mathbf{x}_t|\mathbf{x}_{t-1}) = f_{\mathbf{v}}(\mathbf{x}_t - \mathbf{x}_{t-1})$, where $f_{\mathbf{v}}$ is the $\mathbf{v}$ probability density function (pdf).

2.2 Bayesian filtering

From (2), the probability of $\tilde{y}_{l,t}$ conditioned to state $\mathbf{x}_t$ is given by

$$p(\tilde{y}_{l,t}|\mathbf{x}_t) = \begin{cases} \lambda_l(\mathbf{x}_t)\frac{1}{(2\pi)^{1/2}\sigma_l(\mathbf{x}_t)}\,\mathrm{e}^{-\frac{(y_{l,t}-\overline{y}_l(\mathbf{x}_t))^2}{2\sigma_l^2(\mathbf{x}_t)}} & \text{if } \gamma_l(\mathbf{x}_t) = 1 \\ 1 - \lambda_l(\mathbf{x}_t) & \text{if } \gamma_l(\mathbf{x}_t) = 0; \end{cases}$$

since we suppose the L measurements to be independent each other, the following formula holds for the whole observation $\mathbf{y}_t$:

$$p(\mathbf{y}_t|\mathbf{x}_t) \propto \prod_{l\in Q}\big(1 - \lambda_l(\mathbf{x}_t)\big) \cdot \prod_{l\in R}\lambda_l(\mathbf{x}_t)\frac{1}{(2\pi)^{1/2}\sigma_l(\mathbf{x}_t)}\,\mathrm{e}^{-\frac{(y_{l,t}-\overline{y}_l(\mathbf{x}_t))^2}{2\sigma_l^2(\mathbf{x}_t)}}, \tag{4}$$

where R represents the subset of received ANs and Q the subset of missed ANs at time t. Applying the MLE criterion, an estimate for the state $\mathbf{x}_t$ could be found as $\widehat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t\in\mathcal{X}} p(\mathbf{y}_t|\mathbf{x}_t)$; but according to the Bayesian approach the whole measurement series $\mathbf{y}_{1:t}$ is employed to estimate the state position. The a posteriori pdf therefore depends on a memory-less term and on a memory-bearing one, following

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1}). \tag{5}$$

The Chapman-Kolmogorov equation yields for the second term (a priori pdf)

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int_{\mathcal{X}} p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_t|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1},\ t > 1 \tag{6}$$

where $p(\mathbf{x}_1)$ contains the available information about the MN initial position[2]. The position estimate can then be obtained with the MMSE criterion: $\widehat{\mathbf{x}}_t = \mathbb{E}[\mathbf{x}_t|\mathbf{y}_{1:t}]$.

[2] If the initial position is known precisely, an impulsive pdf can be used; otherwise an uniform pdf is chosen.

Since resolving (6) analitically is impracticable, we use a particle filter [16] to get an approximation of the a priori pdf; this is obtained as an equally weighted sum of S Dirac pulses centered on a set of particles $\{\mathbf{x}_t^{(s)}\}_{s=1}^S$:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \approx \frac{1}{S}\sum_{s=1}^{S}\delta(\mathbf{x}_t - \mathbf{x}_t^{(s)}); \tag{7}$$

note that particle filtering allows an irregular sampling of the pdf, unlike the detecting-tracking algorithms (D/TA). So the a posteriori pdf arises from (5) and (7):

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{s=1}^{S} w_t^{(s)}\delta(\mathbf{x}_t - \mathbf{x}_t^{(s)}),$$

where $w_t^{(s)} \propto p(\mathbf{y}_t|\mathbf{x}_t^{(s)})$ and $\sum_{s=1}^{S} w_t^{(s)} = 1$ (weight normalization). So, the position estimate is given by the MMSE criterion:

$$\widehat{\mathbf{x}}_t = \mathbb{E}[\mathbf{x}_t|\mathbf{y}_{1:t}] \approx \sum_{s=1}^{S} w_t^{(s)}\mathbf{x}_t^{(s)}.$$

The particle filter adopted is a Sequential Importance Resampling Particle Filter (SIR), and its implementation follows the pseudo-code indicated in [16]. At every discrete time t the algorithm computes recursively a set of S particles and their weights $w_t^{(i)}$, $i = 1, \ldots, S$, so approximating the a posteriori pdf $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. To avoid degeneracy problems, i.e. a few particles with heavy weights, a resampling step is done at every iteration: more equally weighted particles are placed instead of a "heavy" single particle, so that every state probability does not change[3].

2.3 A priori maps

In (4) the quantities $\overline{y}_l(\mathbf{x}_t)$, $\sigma_l(\mathbf{x}_t)$, $\lambda_l(\mathbf{x}_t)$ must be known for every possible state position $\mathbf{x} \in \mathcal{X}$ and for every AN $l = 1, \ldots, L$. Their values are obtained through an offline gathering: the MN is placed in N different positions $\{\mathbf{z}^{(n)}\}_{n=1}^N \subsetneq \mathcal{X}$ listening to the broadcasting ANs, and records the number and the RSSI of packets received from every AN in a time slot. Then the sample average provides $\widehat{y}_l(\mathbf{z}^{(n)})$, the sample standard deviation gives $\widehat{\sigma}_l(\mathbf{z}^{(n)})$ and the success probability is deduced from

$$\widehat{\lambda}_l(\mathbf{z}^{(n)}) = \max\left[\lambda_{min}, \min\left(\left.\frac{\#\text{ received packets}}{\#\text{ total packets}}\right|_{\mathbf{z}^{(n)}}, \lambda_{MAX}\right)\right],$$

where λ_{min} and λ_{MAX} stand for the minimum and the maximum allowed values for the success probability respectively and must be appropriately chosen depending on the time slot of the gathering.

[3] Other resampling approaches, such as non-systematic resampling (whenever a significant degeneracy is observed) or log-resampling [17] have been tried out, but with no satisfactory results.

2.4 Interpolation

A high precision for the estimate needs to define a fine grid of possible a priori particle positions $\mathbf{x}^{(m)}$; but in this way the whole offline gathering would become too expensive. So we build a wider and comprehensive mesh for the experimental data surveying and then we interpolate the data for fitting on the finer grid (neither the finer grid must be necessarily regular):

$$\{\mathbf{z}^{(n)}\}_{n=1}^{N} \subsetneq \{\mathbf{x}^{(m)}\}_{m=1}^{M} = \{[h\Delta\ k\Delta]'\}_{\substack{h\in[0,\dots,H]\\ k\in[0,\dots,K]}} \subseteq \mathcal{X},\ M > N.$$

For the sake of simplicity, let us now just consider the average RSSI $\overline{y}_l$. From the data gathering we get a set of values $\{\widehat{y}_l(\mathbf{z}^{(n)})\}_{n=1}^{N}$ that can be interpreted as an irregular sampling of a sample sequence of a bidimensional stochastic stationary zero-mean signal:

$$\overline{y}_l(\mathbf{x}^{(m)}) = \sum_{n=1}^{N} a_n(\mathbf{x}^{(m)})\widehat{y}_l(\mathbf{z}^{(n)}) = \langle \mathbf{Y}_l, \mathbf{a}(\mathbf{x}^{(m)})\rangle,\ l = 1,\dots,L,\ m = 1,\dots,M,$$

where $\mathbf{Y}_l$ is the measurements vector from the l-th AN. We impose that the stochastic signal has an isotropic correlation function $r(\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|) = r(d) = \mathrm{e}^{d/d_0}$, where d_0 must be suitably chosen. The coefficients $a_n(\mathbf{x}^{(m)})$ are calculated applying the *orthogonality principle* that gives *Wiener-Hopf* simultaneous equations [18]. Solving the system of equations we obtain

$$\overline{y}_l(\mathbf{x}^{(m)}) = \mathbf{Y}_l'\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}^{(m)})$$

where $(\mathbf{R})_{ij} = r(\|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\|)$ and $\mathbf{r}(\mathbf{x}^{(m)}) = \begin{bmatrix} r(\|\mathbf{x}^{(m)} - \mathbf{z}^{(1)}\|) \\ \vdots \\ r(\|\mathbf{x}^{(m)} - \mathbf{z}^{(N)}\|) \end{bmatrix}'$. The same method is applied for σ_l and for λ_l.

Fig. 1 shows the interpolated map of the RSSI mean and variance, and the packet transmission success probability corresponding to a single AN. Note how it clearly shows that the node is located at the conjunction of two long corridors forming a T, very similar to what a ray-tracing analytical tool would provide.

2.5 Simulations

We validate *ARIANNA* using a MATLAB simulation. The simulations are based on a RSSI-map stochastic model obtained from measurements performed in the indoor environment described in Section 4. The whole area has been covered using 12 motes, whose arrangement has been designed using a ray tracing software [19]. The entire gathering counts $N = 115$ measurements, each of them during 100 s for 100 broadcasted packets amount. For the interpolation, we have chosen a regular grid with $\Delta = 0.5$ m, $d_0 = 20$ m, $\lambda_{min} = 0.03$, $\lambda_{MAX} = 0.97$ and we have assumed that missing the $l-$th AN (for all measurement long) in a position $\mathbf{z}^{(i)}$ corresponds to $\overline{y}_l(\mathbf{z}^{(i)}) = -70$ dBm, $\sigma_l^2(\mathbf{z}^{(i)}) = 25$ dBm^2: the first value is justified by a cutted off radio propagation model, while the second one is inferred by the gathering data.

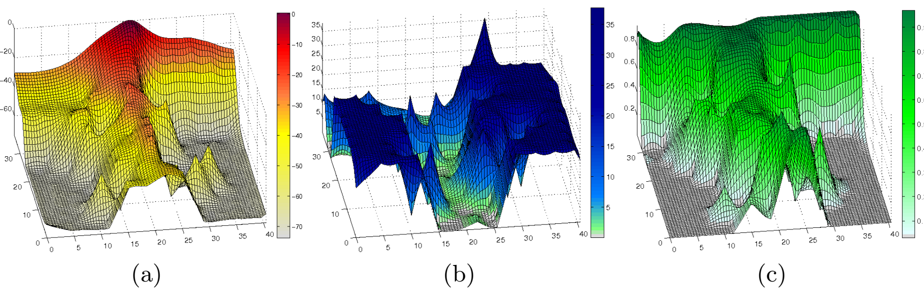


Fig. 1: Interpolated data from AN #2. (a) mean RSSI [dBm]. (b) Variance RSSI [dBm^2]. (c) Packet success probability.

A MN trajectory is created using (3), $t = 1, \dots, 70$, with various driving processes $\mathbf{v}_t$ [10]: a classic 2D Gaussian pdf, a Gaussian ring and a Beta ring. All the driving process are expressed in polar coordinates. The measurements $\overline{y}_l(\mathbf{x}_t)$ are created using (1) where $v_{l,t} \sim \mathcal{N}(0, \sigma_l^2(\mathbf{x}_t))$. The particle filter used $S = 500$ particles. Figure 2.5 shows the simulated tracking performance of ARIANNA on the model of the real environment (left panel) and the corresponding error estimate as a function of time (right panel).

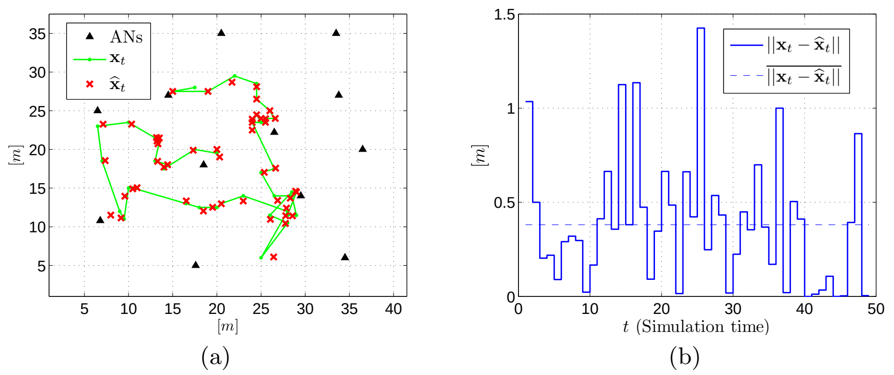


Fig. 2: *ARIANNA* simulation performance in the whole environment. (a) Comparison between real and estimated trajectories. (b) Estimation error (Euclidean distance) and his mean. Black triangles represent ANs location.

3 TESEO Tracking System

3.1 Radio Channel Models for RSS

The adopted statistical model that relates distance from mobile-node with received signal strength is

$$P(d) = P(d_0) - 10n_p \log\left(\frac{d}{d_0}\right) + \chi_\sigma \tag{8}$$

where, $P(d)$ is the received power (RSS) in [dBm], $P(d_0)$ is the received power at a fixed distance from the emitter d_0, expressed in [dBm], n_p is the path-loss exponent, typically $n_p > 2$ because channel dissipates energy, χ_σ is a gaussian aleatory variable with zero mean and variance σ^2 [dBm^2] used to model, and d is the distance from the emitter (in [m]). Therefore $P(d)$ is a gaussian variable with mean $\bar{P}(d) = P(d_0) - 10n_p \log\left(\frac{d}{d_0}\right)$ and variance σ^2 (being $P(d)$ expressed as a linear affine transformation of a gaussian r.v.). An alternative formulation of (8) is

$$P(d) = P_{TX} + A - 10n_p \log(d) + \chi_\sigma$$

where P_{TX} is the well-know transmission power and A is the attenuation factor.

3.2 Architecture and Algorithms

The *TESEO* architecture is specifically designed to support both centralized and decentralized localization algorithms to cope with possible system failures. In particular, the localization is performed using two distinct algorithms: the *main algorithm*, based on an extension of the maximum likelihood (ML) estimator proposed in [12] for autolocalization, uses when the pocket-PC/central-node serial link works, while and the *recovery algorithm*, used in case the serial link was interrupted, adopts the Least Square Estimator [13]. We also tested the Maximum Ratio Combining Approach in simulations, but since it showed poor performance as compared to the other two methods, it has not been implemented in the real-time experimental system.

The **main algorithm** follows these steps

1. *initialization*: all the nodes wake-up, set communication channel and transmission power, start sending radio signals;
2. *broadcast*: the mobile-node broadcasts to every fixed-node N_{pck} packets every T_{broad} [ms], while fixed-nodes wait for packets;
3. *listen*: starting from the first packet received, every fixed node $j = 2, \ldots, N$ stores the received packets if the received signal power $P_{1,j}$ and LQI are greater than the pre-calculated thresholds P_{th}, LQI_{th};
4. *cell making*: every fixed-node $j = 2, \ldots, N$ computes the average $\breve{P}_{1,j}$ of the powers $P_{1,j}$ on the *really* arrived packets (some may be lost) given by (11), and then calculates the M.L. distance estimate from the mobile-node $\breve{d}_{1,j}$ given by (12), based on $\breve{P}_{1,j}$; so, a cell of N_s nodes near the mobile-one is formed, defined by the set

$$A_s = \Big\{ j > 1 \,\Big|\, P_{1,j} > P_{th}, \breve{d}_{1,j} < \text{Range and } LQI_{1,j} > LQI_{th} \Big\},$$

where Range indicates the maximum distance from the mobile-node;

5. *sending distance estimates*: nodes included in A_s send the mobile-node packets containing their ID j and distance estimate $\breve{d}_{1,j}$;
6. *position estimate*: the mobile-node collects all the estimates $\breve{d}_{1,j}$ and sends them through serial link (USB) to the pocket-PC, which calculates the M.L. position estimate $\breve{\mathbf{z}}_1(kT_c)$ given by (14); the mobile-node iterates from point 2.

If serial link is interrupted for some reason, a **recovery algorithm** is provided: it is analogous to the principal one but the target position estimate is calculated by the target node since we used an estimator which is simpler than the M.L. from a computational point of view. Position estimate $\hat{\mathbf{z}}_1(kT_c)$ is calculated using the least squares (L.Q.) method given by (16) if $|A_s| > 2$, otherwise by maximum ratio combining (M.R.C.) method given by (17); finally $\hat{\mathbf{z}}_1(kT_c)$ is sent to the supervising system.

The *Cell Making* step of the main algorithm is a fundamental feature of the algorithm. In fact, the exclusion of nodes which do not belong to the set A_s has two benefits: reduces number of nodes used to estimate position, thus reducing also computational burden, and more importantly, it discards wrong distance estimates. In fact, distance estimates are reliable only when the RSS-channel model given by Equation 8 is valid, and we have empirically observed that the linear power model considered is a good approximation for radio channel only if all constraints defining the set A_s are satisfied. In particular, we noticed that also the use of LQI, and not only RSSI signal $P_{1,j}$, helped reducing estimation errors.

In the next sections, all three location estimators mentioned above are illustrated.

M.L. distance estimator. The power (in [dBm]) transmitted by the j-th sensor and received by the mobile-sensor, $P_{1,j}$, has *gaussian* probability density function (pdf)

$$f(p \,|\, x_1, y_1, z_1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2} \left(\frac{p - \bar{P}(d_{1,j})}{\sigma} \right)^2 \right\} \tag{9}$$

where $d_{1,j} = \|\mathbf{z_1} - \mathbf{z_j}\| = \sqrt{(\mathbf{x_1} - \mathbf{x_j})^\mathbf{2} + (\mathbf{y_1} - \mathbf{y_j})^\mathbf{2} + (\mathbf{z_1} - \mathbf{z_j})^\mathbf{2}}$.

The M.L. distance estimator based on Equation (9) is given by [12]

$$\hat{d}_{1,j} = d_0 \cdot 10^{\frac{P(d_0) - P_{1,j}}{10 n_p}} \tag{10}$$

which can be considerably improved using by substituting $P_{1,j}$ in the previous equation with its mean $\bar{P}_{1,j} = P(d_0) - 10 n_p \log\left(\frac{d_{1,j}}{d_0}\right)$, thus obtaining the (pseudo-)M.L. estimator

$$\breve{d}_{1,j} = d_0 \cdot 10^{\log(d_{1,j}/d_0)} = d_{1,j}.$$

The expected value $\bar{P}_{1,j}$ can be approximated by the sample mean using the ergodic theorem[4]

$$\breve{P}_{1,j}(N_{pck}) \triangleq \frac{1}{N_{pck}} \sum_{i=1}^{N_{pck}} P_{1,j}(i) \longrightarrow \bar{P}_{1,j} \text{ for } N_{pck} \to \infty \tag{11}$$

where N_{pck} is the number of measurements used to calculate the sample mean and $i = 1, \ldots, N_{pck}$ is the temporal index for power sampling: so, the (pseudo-)M.L. estimator is

$$\breve{d}_{1,j}^{N_{pck}} \triangleq d_0 \cdot 10^{\frac{P(d_0) - \breve{P}_{1,j}(N_{pck})}{10 n_p}} . \tag{12}$$

M.L. position estimator. Let $\mathbf{p} \in \mathbb{R}^{|A_s|}$ be the vector obtained by arranging the power values received by the cell-nodes in columns ($|A_s|$ is the cardinality of set A_s); assuming the communication channels as indipendent, $\mathbf{p}$ joint probability density function is

$$f(\mathbf{p} \,|\, \mathbf{x_1}, \mathbf{y_1}, \mathbf{z_1}) = \frac{\mathbf{1}}{(\mathbf{2}\pi\sigma^{\mathbf{2}})^{|\mathbf{A_s}|}} \exp\left\{ -\frac{\mathbf{1}}{\mathbf{2}} \frac{||\mathbf{p} - \mathbb{E}[\mathbf{p}]||^{\mathbf{2}}}{\sigma^{\mathbf{2}}} \right\} \tag{13}$$

Substituting equation (8) into (13), calculating its negative logarithm and then its minimum, we have that M.L. position estimator is given by:

$$\hat{\mathbf{z}}_1 = (\hat{x}_1, \hat{y}_1, \hat{z}_1) \triangleq \arg \max_{(x_1, y_1, z_1)} \ln f(\mathbf{p} \,|\, \mathbf{x_1}, \mathbf{y_1}, \mathbf{z_1}) = \arg \min_{(\mathbf{x_1}, \mathbf{y_1}, \mathbf{z_1})} \sum_{\mathbf{j} \in \mathbf{A_s}} \left(\ln \frac{\hat{\mathbf{d}}^{\mathbf{2}}_{\mathbf{1,j}}}{\mathbf{d}^{\mathbf{2}}_{\mathbf{1,j}}} \right)^{\mathbf{2}} ;$$

the minimum is numerically calculated by conjugate gradient algorithm [20].

The estimator actually implemented in the algorithm makes use of the M.L. distance estimator given by (12), which utilizes power mean; so, M.L. position estimator becomes

$$\breve{\mathbf{z}}_1 = (\breve{x}_1, \breve{y}_1, \breve{z}_1) = \arg \min_{(x_1, y_1, z_1)} \sum_{j \in A_s} \left(\ln \frac{\breve{d}^2_{1,j}}{d^2_{1,j}} \right)^2 . \tag{14}$$

L.S. position estimator [13]. In this case, to obtain position estimate $\mathbf{z}_1(t)$ we solve the overdetermined system

$$\begin{cases} \|\mathbf{z_1(t)} - \mathbf{z_{i_1}}\|^{\mathbf{2}} &= d^2_{1,i_1} \\ &\vdots \\ \|\mathbf{z_1(t)} - \mathbf{z_{i_M}}\|^{\mathbf{2}} &= d^2_{1,i_M} \end{cases} \tag{15}$$

where indexes $i_1, \ldots, i_M$ indicate the anchor-nodes included in A_s which contribute to position estimate. System (15) can be reduced to a linear system,

[4] During a broadcast time lag T_{broad}, the real distance $d_{1,j}$ is approximately costant and the RSS measurements can be considered as random variables composing a random i.i.d. process $P_{1,j}(iT_c)$, $i \in \mathbb{Z}$, which respects the conditions for applying the ergodic theorem.

having as unknown quantities $x_1(t),\ y_1(t),\ z_1(t)$, by subtracting from the first equation all the other ones, obtaining

$$\underbrace{\begin{bmatrix} x_{i_1}-x_{i_2} & y_{i_1}-y_{i_2} & z_{i_1}-z_{i_2} \\ \vdots & \vdots & \vdots \\ x_{i_1}-x_{i_M} & y_{i_1}-y_{i_M} & z_{i_1}-z_{i_M} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1(t) \\ y_1(t) \\ z_1(t) \end{bmatrix}}_{\mathbf{z_1(t)}} = \underbrace{\frac{1}{2}\begin{bmatrix} d_{i_2}^2-d_{i_1}^2-x_{i_2}^2+x_{i_1}^2-y_{i_2}^2+y_{i_1}^2-z_{i_2}^2+z_{i_1}^2 \\ \vdots \\ d_{i_M}^2-d_{i_1}^2-x_{i_M}^2+x_{i_1}^2-y_{i_M}^2+y_{i_1}^2-z_{i_M}^2+z_{i_1}^2 \end{bmatrix}}_{\mathbf{b}}$$

this system can be solved by L.S. method, i.e.

$$\hat{\mathbf{z}}_1(t) = \arg\min_{\mathbf{z_1(t)}} ||A\mathbf{z_1(t)} - \mathbf{b}||^2 = \left(\mathbf{A^T A}\right)^{-1} \mathbf{A^T b} \tag{16}$$

where M.L. distance estimate $\breve{d}_{1,i_k}$ (given by (12)) is used instead of real distance d_{1,i_k}.

M.R.C. position estimator [14]. Position estimate is obtained by a convex combination of fixed-nodes coordinates belonging to the cell, i.e.

$$\hat{\mathbf{z}}_1(t) = \sum_{j\in A_s} w_j \mathbf{z_j} \tag{17}$$

where w_j are positive scalars satisfying the condition $\sum_{j\in A_s} w_j = 1$. The variables w_j are chosen in order to give more weight to anchor-nodes coordinates closer to mobile node; possible choices are

$$w_j = \frac{1/\hat{d}_{1,j}^2}{\sum_{i\in A_s}\left(1/\hat{d}_{1,i}^2\right)} \tag{18}$$

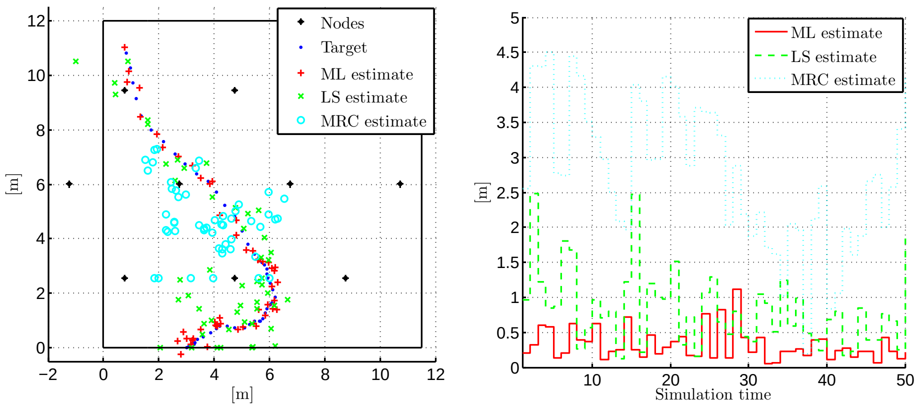
or

$$w_j = \frac{1/\mathbb{E}\left[\hat{d}_{1,j}^2\right]}{\sum_{i\in A_s}\left(1/\mathbb{E}\left[\hat{d}_{1,i}^2\right]\right)} \tag{19}$$

where statistical powers are calculated from samples using ergodic theorem, i.e. $\mathbb{E}\left[\hat{d}_{1,j}^2\right] \simeq \frac{1}{N_{pck}}\sum_{k=1}^{N_{pck}} \hat{d}_{1,j}^2(k)$, where each k-th distance estimate (10) is calculated using only one packet. Weights (19) make use of the fact that estimates statistical power increases with distance: as a matter of fact, it can be shown that $\mathbb{E}\left[\hat{d}_{1,j}^2\right] = e^{2\gamma} d_{1,j}^2$.

3.3 Simulations

We implemented a MATLAB simulator to test the previous localization algorithms. We choose the following parameters based on experimental measurements and simulations results: *cell range* $= 6$ [m], $P_{th} = -80$ [dBm] (RSS power threshold), $N_{meanMax} = 40$ (maximum number of packets for RSS mean calculation), $n_p = 2.12$, $A = -63.67$, $d_0 = 1$ [m] and $\sigma = 7.57$ [dBm]. We tested the algorithms within an area of 11.5×12 [m^2], with 10 fixed nodes distant 4 [m] from each other on a regular triangular grid: the comparison between the three methods is in Figure 3a, the norm-error $\|\tilde{\mathbf{z}}_1\|$ trend is reported in Figure 3b. M.L. estimator is clearly the best one, while M.R.C. is the worst one, and this motivates the choice of L.S. estimator as recovery algorithm. We used as weights for M.R.C. estimator the ones shown in (18), because there are no significant improvements brought by using (19). Finally, it's interesting to see the effect of an ill-conditioned matrix in L.S. method: consider figure 3a, it is easy to see that the first L.S. estimates have y coordinate set to 0. This is due to the fact that the cell for the first trajectory steps is formed by the two nodes positioned on the bottom left of the map which have the same y coordinate: this leads to a null column in matrix A and so to an undetermined solution, which is set to 0 by the algorithm. In this case, M.R.C. estimator is a better solution, as it sets the estimated y coordinate the same as the nodes one (clearly visible in Figure 3a), leading, generally, to a smaller estimate error.



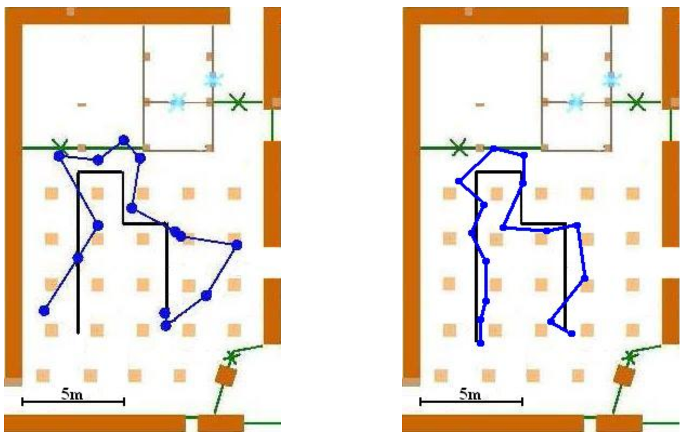
(a) (b)

Fig. 3: Comparison between methods. (a) True and estimated trajectories. (b) Estimation error trends.

4 Implementation and experimental results

In our experiment the used sensors nodes Tmote Sky produced by Moteiv Corporation [21]. The area of interest spanned 1150 m^2 within an indoor environment which included reinforced concrete walls and pillars, about twenty uninterrupted power supplies and high-voltage electric cables, and several metal storage cages.

In both architectures, the wireless sensor nodes have been programmed with a high-level language like C/C++, called NESC, and their algorithms are managed by the open-source operating system TINYOS 2.0.1, designed for WSNs. The *ARIANNA* tracking system has been implemented using Java and C coding for data acquisition and MATLAB for the localization algorithms and visual rendering. In particular, the MN was connected to a laptop and exchanges data with it on a serial link. *ARIANNA* also sent the current position estimate over the WSN, so allowing remote monitoring of the MN. Differently, *TESEO* has been implemented using only C/C++ and Java for data acquisition, localization algorithms and visualization, thus providing faster localization estimates. The mobile node was connected to a notebook and communicated with it through a USB (serial) port. The user side on the laptop is a GUI programmed with JAVA (J2SE 1.5.0.06). The user-friendly JAVA frame has been realized to provide a useful and smart interface, both for the monitoring and the configuration of the system. All the work has been performed on UBUNTU FEISTY. The entire track used to test the two algorithms was about $195m$ long and has



(a) (b)

Fig. 4: Experimental tracking results of real-time implementation of *ARIANNA* localization system (left) and *TESEO* localization system (right) in indoor environment. True trajectory is shown in think solid line and reconstructed trajectory in thin line.

been covered in about $140s$. Figure 4 shows experimental results obtained on a smaller portion of the whole trajectory for better visualization. Walls, pillars and doors are clearly recognizable in the map, but other large and small objects were present. The actual environment where experiments were performed can be better viewed in the videos corresponding to real-time tracking and visualization of *ARIANNA* `http://www.youtube.com/v/uXGqQu18yIs` and *TESEO* `http://www.youtube.com/v/3j_RscaluxU`, respectively.

5 Conclusions and future work

In this work we proposed two different tracking system architectures based on two different approaches to RF localization and tracking in WSNs: one, named *ARIANNA* , is based on a stochastic RSSI-map model built from experimental measurements, while the other, named *TESEO* , is based on the RSSI-channel model and three different geometric localization algorithms. These two architectures try to implement the best algorithms proposed in the literature so far, and they also included some novel features. Each architecture, or more generally each localization approach they represent, has pros and cons as illustrated in Table 5. In fact, *TESEO* provides about three times smaller estimation errors than *ARIANNA* . Moreover, *TESEO* requires lower computational resources since algorithms are simpler, and it does not require time-consuming in-locus measurements gathering at the time of installation. On the other hand, *ARIANNA* , when compared to *TESEO* , requires only a sixth of the number of nodes to cover the whole area. Clearly this comparison shows that it is hard to claim that one architecture is better then the other.

	installation complexity	computational complexity	mean − max error	node density
TESEO	Low	Low-Medium	0.7m-1.5m	$16m^2$/node
ARIANNA	High	High	2m-4m	$90m^2$/node

Besides the features mentioned in the table, many other factor should be compared such as robustness to anchor node failures, robustness to environmental changes such as new disposition of the furniture of the presence of many people, and scalability of the system when multiple moving nodes are present. These are some of the aspects we are currently exploring. Other important aspects are the implementation of a whole navigation and coordination system as mentioned in the introduction, and a more detailed comparison between the two different architectures from a computational complexity point of view, by implementing them on the same HW platform.

References

1. R. Peng and M. L. Sichitiu, “Angle of arrival localization for wireless sensor networks,” in *Proc. of IEEE Conference on Sensor and Ad Hoc Communications and Networks*, Reston, VA, USA, September 2006.
2. A. Smith, H. Balakrishnan, M. Goraczko, and N. B. Priyantha, “Tracking Moving Devices with the Cricket Location System,” in *Int. Conf. on Mobile Systems, Applications and Services (Mobisys 2004)*, Boston, MA, June 2004.
3. Y.-T. Chan, W.-Y. Tsui, H.-C. So, and P. chung Ching, “Time-of-arrival based localization under nlos conditions,” *IEEE Transactions on Vehicular Technology*, vol. 55, no. 1, pp. 17–24, 2006.
4. S. Oh, L. Schenato, P. Chen, and S. Sastry, “Tracking and coordination of multiple agents using sensor networks: system design, algorithms and experiments,” *To appear in Proceedings of IEEE.*
5. J. Park and A. C. Kak, “Distributed online localization of wireless camera-based sensor networks by tracking multiple moving objects,” *submitted for publication.*
6. P. Bahl and V. N. Padmanabhan, “Radar: an in-building rf-based user location and tracking system,” in *Proceedings of IEEE Infocomm 2000*, January 2000, pp. 775–784.
7. K. Lorincz and M. Welsh, “Motetrack: a robust, decentralized approach to rf-based location tracking,” *Personal and Ubiquitous Computing*, vol. 11, no. 6, pp. 489–503, 2006.
8. U. Spagnolini and A. Bosisio, “Indoor localization by attenuation maps: model-based interpolation for random medium,” in *ICEAA Internation Conference on Electromagnetics in Advanced Application*, Sept. 2005.
9. R. A. Valenzuela, “Ray tracing prediction of indoor radio propagation,” in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 1, 1994, pp. 140–144.

10. C. Morelli, M. Nicoli, V. Rampa, and U. Spanolini, “Hidden markov models for radio localization in mixed los/nlos conditions,” *IEEE transaction on signal processing*, vol. 55, no. 4, 2007.
11. C. Morelli, V. Rampa, M. Nicoli, U. Spagnolini, and C. Alippi, “Particle filters for rss-based localization in wireless sensor networks: an experimental study,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2006.
12. N. Patwari, “Location estimation in sensor networks,” Ph.D. dissertation, University of Michigan, 2005.
13. Z. Li, W. Trappe, Y. Zhang, and B. Nath, “Robust statistical methods for securing wireless localization in sensor networks,” in *Symposium on Information Processing in Sensor Networks (IPSN’05)*, 2005, pp. 91–98.
14. L. Song and D. Hatzinakos, “A cross-layer architecture of wireless sensor networks for target tracking,” *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 145–158, February 2007.
15. Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.* J. Wiley and Sons, 2001.
16. M. S. Arulampalam, S. Maskell, N. Gornod, and T. Clapp, “A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking,” *IEEE transaction on signal processing*, vol. 50, no. 2, 2002.
17. N. D. Freitas, “Sequential sampling-importance resampling (sir).” [Online]. Available: http://www.cs.ubc.ca/~nando/software.html
18. G. Bolondi, F. Rocco, and S. Zanoletti, “Automatic contouring of faulted subsurfaces,” *Geophysics*, vol. 41, no. 6, pp. 1377–1393, 1976.
19. *Radioplan RPS User Manual - version 5.3*, Radioplan GmbH.
20. W. V. B. F. W.H. Press, S.A. Teukolsky, *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1992.
21. *Tmote Sky: datasheet*, Moteiv Corporation, November 2006.