Contents

Acknowledgements

During my PhD I've met several people; most of them were kind and helped me, others were not. But I would like to thank both, because the former taught me how pleasant life can be, from the latter I learnt how difficult and grasping life can be. I don't know but I suspect that they have the same relevance.

Other people, my family, old and new friends, have been in at least a part of my life of paramount importance in what I am now - good and evil! - and deserve a special acknowledgement.

I won't write down boundless lists of names since everybody knows, or knew, my feelings. Here, I prefer to express my enthusiasm for the new challenges taking shape on the horizon.

Thanks God, a day is over, but a new day is already born!

Luigi Bruno

Introduction

Positioning means to gain awareness, within a certain accuracy, of a mobile device location. It is not a stand alone application, but rather a service that is ground to several applications. Hundreds of research groups worldwide have been developing automated location systems for several decades, first for military, then for civilian purposes.

To date the standard solution to outdoors positioning is provided by the Global Navigation Satellite Systems (GNSS), that is the GPS and the forthcoming Galileo.

Challenges are different indoors and require micro-detailed georeferencing to satisfy users' growing needs. To geo referentiate a building is not enough if the position of users inside the building is also relevant to the application. Objects are used as landmarks, and relationships among the objects are crucial for symbolic representation of the whole system. The applications to this scenario are manifold and range from logistics to routing, from personal safety to emergency response.

Unluckily, GNSS do not work properly indoors for the absence of Line Of Sight (LOS) propagation between the satellites and the mobile, such that deep multipath effects randomize the times of arrival of the signals. This prevents GNSS from achieving a satisfactory degree of accuracy, to the extent that it cannot be used for determining whether a person stays inside or outside a certain building, and it is by no means possible to locate it within a room or a floor. The impossibility of using satellite systems was already clear in the mid '90s and has driven toward the exploitation of local technologies; consequently, a plethora of choices has been explored.

None of them has shown optimal or even sufficient performances in all settings, so the research in this field keeps on gathering much interest inside the Navigation Community.

The main interest has been drawn to 'local technologies', that are non invasive, cheap and under the full control of the provider; in particular, the reuse of technologies already deployed for wireless communication purposes yields a great opportunity. Thus, RFID, Bluetooth, WiFi and ZigBee have been employed into locating systems, some of which being proposed as popular commercial solutions, like LandMARC or ActiveBadge. Our choice falls on the Wireless Local Area Network (WLAN) and, concerning the architecture and the experimental results described in the thesis, on its main example: the standard IEEE 802.11 also named WiFi.

To avoid or, at least, attenuate the multipath problem and, above all, the addition of further costly hardware as, for example, accurate clocks or directional antennas, we adopt, as the relevant parameter to measure, the power propagated between transmitter and receiver, that can go over walls and obstacles. However, we must counteract a variety of unpredictable effects which make it difficult to properly use the signals and to achieve good accuracies.

Such physical limits boost the contribution of optimization techniques provided by the discrete signal processing theory. Among the others, we rather prefer Bayesian techniques borrowed by the Probability Theory because they provide elegant and powerful tools by formulating the positioning problem in terms of estimation theory.

Anyway, even the most sophisticated processing techniques have strong limits: indoors accuracies tighter than 2 meters are achieved only in very peculiar conditions and after heavy training stages. The reason lies in the environment, that is not only difficult to understand but also changes rapidly.

That is the reason why this thesis has been thought, designed and written: we believe that the actual borders in indoor positioning can be overcome and the only way of doing it is by *looking at the environment*.

Environment profiling

The indoor environment can change drastically from place to place and also from time to time. Some elements of environment profiling are thus necessary in order to improve or, sometimes, to get satisfactory performances. Even if the classic positioning techniques assume perfect knowledge of the environment, yet the eldest studies about indoor localization put in evidence the need to catch the environmental most relevant changes by means of adaptive algorithms.

Several techniques can be used to this aim are many, depend on the framework adopted, on the model of the transmission channel and so forth. Also, the environmental 'discovery' can be considered as side information in the positioning problem or can consist in a set of variables of interest for some applications.

We cast this problem, after having provided suitable models, onto a probabilistic problem to use the Bayesian framework, whose algorithms are properly handled and modified to meet the new challenge.

Co-localization

Then we address a scenario fitting more practical considerations, although it is formalized and solved in a mathematical setup. In indoor positioning the mobile device must be located, while the transmitters are considered known and perfectly located in a local reference system.

But is this hypothesis reasonable? In many case it is, but not always. The transmitters are positioned according to 'communication' requirements rather than to the 'positioning' task. When the communication needs change (or for whatever reason), they can be moved accordingly. In this case someone could look after their new position and update the algorithms, but providing algorithms that automatically find the transmitters, or at least some of them, and use their measurements with a degree of reliability proportional to their provenience knowledge would surely be a step forward.

Consider now another example: we are getting used at large buildings with tens of WLAN transmitters. Some of them, perhaps, are managed by the same person who puts on the locating system, but most of them is not. This latter's measurements can be of great interest in order to boost the positioning accuracy and should be accounted for.

Co-localization, as this topic is called in the literature, is very recent and is here explored by means of Bayesian tools, for which we will resort to further modifications of the corresponding algorithms.

Simultaneous Localization And Mapping

Being localization an estimation problem, the map of the building is a set of constraints on the estimate. This set can be of poor relevance in open spaces, but it is of great interest in buildings full of offices and hallways. The question is: can we use properly the map?

The answer is yes, of course, but only if we knew the map. Otherwise, the most elegant and cheap solution we can exploit consists in estimating it together with the mobile's position. This topic, arisen in the robotic literature in the mid '80s, has been experimenting strong theoretical improvements during the last 20 years and, let aside some peculiar cases, is normally undertaken by means of the Bayesian theory.

Simultaneous Localization And Mapping (SLAM) is different from co-localization since it aims at determining a building map, useful also in the future for other users or other applications. Even if some authors perform SLAM by employing only WLAN based measurements, the fusion with other sensors' measurements has been gaining favour. In this view, data from Inertial Measurements Units (IMU) are exploited. The mix of these two types of data is interesting for they are extremely heterogeneous: WLAN measurements are consistent with a reference system anchored to the building, while IMU's data refer to the user's system. Their fusion is challenging, but if well done it can highly boost the per-

formances, being one type of measurements 'strong' in the weak points of the other one.

Summary of the thesis

The thesis is organized as follows: a basic review of estimation theory is proposed in Chapter 1; in Ch. 2 we present indoor localization instead, as well as the main techniques adopted so far in the literature. The thesis' original contribution is provided starting from Ch. 3 where the environment profiling is considered. Co-localization is instead the topic addressed in Ch. 4 mainly from a theoretical point of view; it provides a strong formalization that is also useful in Ch. 5, where the final scenario with the fusion of WLAN and IMU measurements within the SLAM set up is considered.

Among thousands of other considerations that could be made here, we'd rather say what is our long term project: we would like to provide a contribution to a piece of the scientific research that walks, and sometimes sprints, towards the provision of a better quality of life to humans, all humans, conscious that *'real progress happens only when advantages of a new technology become available to everybody'*.

Chapter 1

Basics of Estimation Theory

Localization is an estimation problem and thus it is appropriate, in the first chapter, to describe some general elements of estimation theory. Since the topic is very spread, we will deal with just a part of it; some guidance in our choices is given by next chapters' content and thus more attention will be paid to those concepts which are applicated in indoor localization.

Among a plethora of choices, none of which is wrong, we split the estimation theory in two branches: static and dynamic estimation. This division, despite being arbitrary, is particularly important in the view of indoor localization because it reflects the division between their fields of application, techniques being used and, usually, research groups working on positioning.

Besides the estimation criteria and algorithms, we also provide the main theoretical bounds commonly used as benchmarks. The importance of such bounds is essential for the reseachers to understand how much room for improvements they have.

The chapter is so composed: static estimation is dealt with in Sect. 1.1, followed by the presentation of the corresponding Cramer-Rao based bounds in Sect. 1.2; similarly, we present dynamic estimation and its bounds in Sects. 1.3 and 1.4 respectively; some considerations are finally pointed out in Sect. 1.5.

1.1 Static estimation

We first deal with static estimation, in which a suitable function of the measurements - the *estimator* - is used to infer the value assumed by a variable - the *estimatee*. A large variety of frameworks is possible, based on what is assumed about the estimatee and the measurement models. In detail, the estimatee may be assumed either continuous- or discrete-valued: in the latter case the estimation is rather named *classification* and, in absence of models, nonparametric methods based on prefilled databases have to be used. In the other case probabilistic techniques are employed and the assumed nature of the estimatee, whether deterministic parameter or random variable, leads to Bayesian or classic statis-

tical techniques rispectively.

1.1.1 Probabilistic estimators

Probabilistic estimators are divided into two families, i.e. Bayesian and non Bayesian. The difference is that Bayesian estimation treats the estimatee like a random variable with an assigned pdf, while non Bayesian estimation deals with non random, but unknown, variables. In principle, they represent two completely different approaches, but most times the choice between them is based on convenience: if some prior information on the variable is available, Bayesian statistics allows its exploitation; on the other hand, non Bayesian techniques are often easier.

Only continuous valued estimatees are considered in this section; nevertheless, the extension to the discrete case is straightforward.

Non Bayesian estimators

Let $\boldsymbol{\theta} \in \mathcal{R}^{n_x}$ be a nonrandom variable and $\mathbf{y} \in \mathcal{R}^{n_y}$ a set of measurements. If the estimator is $\hat{\boldsymbol{\theta}}(\mathbf{y})$, the estimation error is defined as

$$\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(\mathbf{y})$$

and the corresponding Mean Squared Error (MSE) is

$$C = E\left[\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}'\right].$$

An important case is met when the estimator is *unbiased*, i.e.

$$E\left[\hat{\boldsymbol{\theta}}\right] = \boldsymbol{\theta};$$

then the MSE corresponds to the estimator covariance and their minimizations are equivalent.

Minimum Variance Unbiased Estimator (MVUE) An unbiased estimator with minimum covariance

$$\hat{\boldsymbol{\theta}}(\mathbf{y}) = \arg\min_{\hat{\boldsymbol{\theta}}} \operatorname{cov}(\hat{\boldsymbol{\theta}}(\mathbf{y}))$$

is the MVUE and is automatically optimal w.r.t. MSE. MVUEs are not very popular, indeed, because they may not exist[1] and, anyway, they are often very difficult to carry out.

Best Linear Unbiased Estimator (BLUE). MVUE can be an arbitrary function of data, but if we add a linearity constraint

$$\hat{\boldsymbol{\theta}}(\mathbf{y}) = H\mathbf{y},$$

where H is a $(n_x \times n_y)$ matrix, we obtain a suboptimal estimator, called BLUE, much more tractable from a mathematical point of view. That is why BLUE is very common, even if it leads to errors typically higher than MVUE (unless the MVUE is already linear).

Maximum Likelihood Estimator (MLE). Let us define the likelihood function of data as

$$L(\boldsymbol{\theta}) = p(\mathbf{y}|\boldsymbol{\theta}),$$

and the log-likelihood function, more useful in the presence of exponential pdfs

$$l(\boldsymbol{\theta}) = \log p(\mathbf{y}|\boldsymbol{\theta}).$$

The maximization of both w.r.t. $\boldsymbol{\theta}$ (it is equivalent since the logarithm is a monotonically increasing function) leads to MLE

$$\hat{\boldsymbol{\theta}}(\mathbf{y}) = \arg\min_{\boldsymbol{\theta}} l(\boldsymbol{\theta}).$$

MLE is usually biased and does not imply bounds on the MSE. Besides its comfortable tractability, the main reason for its popularity is that it is asymptotically efficient: with the addition of new independent measurements, MLE becomes unbiased and its covariance tends to zero.

[1] MVUE must exhibit the minimum MSE for all values of $\boldsymbol{\theta}$.

Bayesian estimators

Let $\boldsymbol{\theta} \in \mathcal{R}^{n_x}$ be a random variable with prior distribution $p(\boldsymbol{\theta})$ and $\mathbf{y} \in \mathcal{R}^{n_y}$ a set of measurements. The estimator is again a function $\hat{\boldsymbol{\theta}}(\mathbf{y})$, whose error covariance is

$$C = E\left[\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}'\right],$$

with $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(\mathbf{y})$. Note that now the expectation is done w.r.t. the joint distribution $p(\boldsymbol{\theta}, \mathbf{y})$.

Minimum Mean Square Error (MMSE). The MMSE estimator minimizes the MSE

$$\hat{\boldsymbol{\theta}}(\mathbf{y}) = \arg\min_{\hat{\boldsymbol{\theta}}} C = \arg\min_{\hat{\boldsymbol{\theta}}} E\left[\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}'\right].$$

It can be shown that the MMSE estimator of $\boldsymbol{\theta}$ is the expectation of $\boldsymbol{\theta}$ after that the data vector $\mathbf{y}$ is observed

$$\hat{\boldsymbol{\theta}}(\mathbf{y}) = E\left[\boldsymbol{\theta}|\mathbf{y}\right].$$

MMSE is usually difficult to find and thus other solutions are preferred. As in the non Bayesian case, a linear restriction is also used.

Linear Minimum Mean Square Error (LMMSE). The restriction of the MMSE estimators to the class of linear (or affine) functions of data leads to the LMMSE estimators

$$\hat{\boldsymbol{\theta}} = H\mathbf{y} + G,$$

with H and G matrices of dimensions $(n_x \times n_y)$ and $(n_x \times 1)$ respectively. LMMSE is much more common than MMSE for its mathematical tractability, even if a higher MSE is usually achieved.

Maximum A Posteriori Estimator (MAP). The estimator that maximizes the posterior pdf is called MAP

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{y}) = \arg\max_{\boldsymbol{\theta}} p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

in which, like in MLE, the logarithm can be taken. Note that the effective difference from MLE is due to the presence of the prior distribution of $\boldsymbol{\theta}$; nevertheless, MLE and MAP involve different probabilistic frameworks.

1.1.2 Non probabilistic estimators

In absence of models, the estimation can be based on a set of labeled measurements, generated at known values of $\boldsymbol{\theta}$ and collected during a training stage. Then, we can basically either estimate a probabilistic measurement model as a function of $\boldsymbol{\theta}$, or use non probabilistic techniques.

In this section we evaluate two non probabilistic approaches: the former is used in classification and does not require an explicit statement of the measurement model, the latter assumes a non random model and minimizes its divergence from the real data.

Fingerprinting

If $\boldsymbol{\theta}$ is a non random variable with values in a finite set of cardinality N

$$\boldsymbol{\theta} \in \mathcal{C} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$$

and no measurement model is supposed, fingerprinting based methods represent an important alternative. Several variants have been proposed which differ in some part, but the central point is always the 'fingerprint'. The fingerprint of a class is essentially a set of measurements collected in the corresponding hypothesis that is stored in a database during an off-line training stage.

The fingerprint from the estimatee during the estimation stage is thus compared with those ones stored into the database, in order to find the 'nearest' one in some sense. For this task k-Nearest Neighbours (k-NN) based algorithms are easily applicable.

k - Nearest Neighbours

The k-NN is a classification algorithm first presented in [FH89][2]. During a former training stage, M records about N classes, with $M > N$, are collected with known prior association. Then, when an unlabeled record is available, its 'distance' from the previous records is computed according to some metric space (e.g. Euclidean distance). The resulting vector of distances $\mathbf{d}$ is then sorted in ascending order and the first k entries are selected (closest records): the algorithm opts for the most represented class among them.

The power of k-NN is well expressed by its convergence features, whose complete description, made in terms of Information Theory, is proposed in [CH67].

Least Squares Estimator (LSE)

LSE is based on a linear measurement model assigned with an additive non random error $\mathbf{n}$

$$\mathbf{y} = H\boldsymbol{\theta} + \mathbf{n}$$

and the sum of the errors is then minimized

$$J = (\mathbf{y} - H\boldsymbol{\theta})'(\mathbf{y} - H\boldsymbol{\theta}).$$

If $\boldsymbol{\theta}$ is discrete (with finite cardinality) the search of the minimum can be done in an exhaustive way, otherwise the solution is found by setting the gradient of J w.r.t. $\boldsymbol{\theta}$ to zero

$$\nabla_{\boldsymbol{\theta}} J = -2H'(\mathbf{y} - H\boldsymbol{\theta})' = 0,$$

which yields

$$\boldsymbol{\theta} = (H'H)^{-1}H'\mathbf{y}.$$

Alternative LS methods allow to weight each error component in a different way[3], yielding the Weighted LSE. If the model is not

[2]The original technical report is dated 1951.

[3]For example if the covariance of the errors is known it could be used as a measure of data informativeness.

linear, we have to use instead an iterative procedure, based on the gradient descent algorithm [SA05, Ch. 3].

1.2 Cramer Rao Lower Bound

The Cramer Rao Lower Bound (CRLB) provides a lower bound for the covariance (and MSE) of an unbiased estimator when the estimatee is a non random parameter. Being it just a theoretical bound, the existence of an unbiased estimator whose covariance achieves CRLB is not ensured at all and is case dependent.

The CRLB is defined as the inverse of the Fisher Information Matrix (FIM), that is a classic measure for the information contained in some data $\mathbf{y}$ about the estimatee $\boldsymbol{\theta}$. We now start with the definition of the FIM and then we state the main result about the CRLB.

Definition 1 (FIM for nonrandom parameter)**.** The FIM for a nonrandom parameter $\boldsymbol{\theta}$ given data $\mathbf{y}$ is provided by (logarithms meant in the natural base)

$$J(\boldsymbol{\theta}) = E\left[\left(\frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{y}|\boldsymbol{\theta})\right)^T \left(\frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{y}|\boldsymbol{\theta})\right)\right]. \tag{1.1}$$

An alternative form of the FIM is given by

$$J(\boldsymbol{\theta}) = -E\left[\frac{\partial^2}{\partial \boldsymbol{\theta}^2} \log p(\mathbf{y}|\boldsymbol{\theta})\right]. \tag{1.2}$$

The FIM is evaluated at the true value of $\boldsymbol{\theta}$ and thus it is a function of $\boldsymbol{\theta}$. The inverse of FIM is the CRLB, that is in turn the lower bound of the MSE of any unbiased estimators.

Theorem 1 (CRLB for nonrandom parameters)**.** *If $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ is an unbiased estimator of the non random parameter $\boldsymbol{\theta}$ and in the hypothesis that*

$$\frac{\partial p(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \qquad \textit{and} \qquad \frac{\partial^2 p(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}$$

exist and are absolutely integrable, then the MSE satisfies

$$P = E\left[\left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(\boldsymbol{y})\right)^T \left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(\boldsymbol{y})\right)\right] \geq J\left(\boldsymbol{\theta}\right)^{-1} \tag{1.3}$$

where $J(\boldsymbol{\theta})^{-1}$ *is the CRLB.*

The proof is a simple application of the Schwarz inequality and is reported in several books, see for example [VT01].

If the MSE of an unbiased estimator achieves the CRLB, then this estimator is said to be efficient. An efficient estimator must be a MVUE. Since the CRLB may depend on the true value $\boldsymbol{\theta}$, which is unavailable (otherwise there is no need for estimation), the efficient estimator may not exist; however, the MVUE may still exist in that case [Kay93].

1.2.1 CRLB for Bayesian estimators

The Posterior CRLB (PCRLB) is an extension of the CRLB valid in the Bayesian estimation theory [VT01]. In this framework the estimatee $\boldsymbol{\theta}$ is a random variable and we refer to the joint distribution $p(\boldsymbol{\theta}, \mathbf{y})$ instead of the likelihood pdf $p(\mathbf{y}|\boldsymbol{\theta})$.

Definition 2 (FIM for random parameters)**.** The FIM for a random parameter $\boldsymbol{\theta}$ given data $\mathbf{y}$ is provided by (logarithms meant in the natural base)

$$J\left(\boldsymbol{\theta}\right) = E\left[\left(\frac{\partial}{\partial \boldsymbol{\theta}} \log p(\boldsymbol{\theta}, \mathbf{y})\right)^T \left(\frac{\partial}{\partial \boldsymbol{\theta}} \log p(\boldsymbol{\theta}, \mathbf{y})\right)\right]. \tag{1.4}$$

An alternative form of FIM is given by

$$J\left(\boldsymbol{\theta}\right) = -E\left[\frac{\partial^2}{\partial \boldsymbol{\theta}^2} \log p(\boldsymbol{\theta}, \mathbf{y})\right]. \tag{1.5}$$

In both cases the expectation is done w.r.t. $\boldsymbol{\theta}$ and $\mathbf{y}$ jointly. The inverse of FIM is the PCRLB, that is in turn the lower bound of the MSE of an estimator.

Theorem 2 (CRLB for random parameters)**.** *If $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ is an estimator of the random parameter $\boldsymbol{\theta}$ and in the hypotheses that*

$$\frac{\partial p(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \qquad \textit{and} \qquad \frac{\partial^2 p(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}$$

exist and are absolutely integrable,

$$\lim_{\boldsymbol{\theta}\to\infty} B(\boldsymbol{\theta}) p_0(\boldsymbol{\theta}) = 0, \tag{1.6}$$

$$\lim_{\boldsymbol{\theta}\to-\infty} B(\boldsymbol{\theta}) p_0(\boldsymbol{\theta}) = 0, \tag{1.7}$$

with

$$B(\boldsymbol{\theta}) = \int_{-\infty}^{\infty} \left[\hat{\boldsymbol{\theta}}(\boldsymbol{y}) - \boldsymbol{\theta}\right] p(\boldsymbol{y}|\boldsymbol{\theta})\, d\boldsymbol{y},$$

then the MSE satisfies

$$P = E\left[\left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(\boldsymbol{y})\right)^T \left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(\boldsymbol{y})\right)\right] \geq J\left(\boldsymbol{\theta}\right)^{-1} \tag{1.8}$$

where $J(\boldsymbol{\theta})^{-1}$ is the CRLB and $p_0(\boldsymbol{\theta})$ is the prior distribution of $\boldsymbol{\theta}$.

The conditions (1.6)-(1.7) are the extension of the unbiasedness to the random parameter case, but they are much weaker. In fact, the intuitive counterpart of unbiasedness in the Bayesian estimation would be

$$E[\boldsymbol{\theta}] = E[\hat{\boldsymbol{\theta}}(\mathbf{y})]$$

with expectation w.r.t. data likelihood, that is equivalent to require $B(\boldsymbol{\theta}) = 0,\ \forall\boldsymbol{\theta}$; the conditions (1.6)-(1.7) of the Theorem 2 require instead that the asymptotic behaviour w.r.t. $\boldsymbol{\theta}$, weighted on the tails of the prior pdf, be regular.

1.3 Bayesian dynamic estimation

The main problem when addressing a dynamic system is mathematical tractability. Complexity requirements prompt sequential

solutions whose complexity does not grow over time, otherwise their field of applicability is dramatically reduced to few instants. To achieve these results, Hidden Markov Models (HMM [Rab89]) are usually employed in a Bayesian framework. In HMMs a n_x-dimensional (hidden) state $\mathbf{x}_k \in \mathcal{R}^{n_x}$ evolves over time according to a process model and the n_y-dimensional observable $\mathbf{y}_k \in \mathcal{R}^{n_y}$ is some noisy function of the state. The peculiarities are two; at k

- the next state $\mathbf{x}_{k+1}$ is independent of the past given the present $\mathbf{x}_k$ (one step memory), viz.

$$p\left(\mathbf{x}_{k+1}|\mathbf{x}_{0:k}\right) = p\left(\mathbf{x}_{k+1}|\mathbf{x}_k\right); \tag{1.9}$$

- each observable depends only on the contemporary state, viz.

$$p\left(\mathbf{y}_k|\mathbf{x}_{0:k}, \mathbf{y}_{0:k-1}\right) = p\left(\mathbf{y}_k|\mathbf{x}_k\right). \tag{1.10}$$

Condition in eq. (1.9) brings to a comfortable application of the chain rule to the hidden state distribution (before the measurements observation); starting from a known prior pdf $p(\mathbf{x}_0)$ for the initial state $\mathbf{x}_0$, we obtain

$$p(\mathbf{x}_{1:k}) = p(\mathbf{x}_0) \prod_{s=1}^{k} p(\mathbf{x}_s|\mathbf{x}_{s-1}). \tag{1.11}$$

Condition in eq. (1.10), instead, greatly simplifies the likelihood statement of the measurements series, i.e.

$$p(\mathbf{y}_{1:k}|\mathbf{x}_{1:k}) = \prod_{s=1}^{k} p(\mathbf{y}_s|\mathbf{x}_s). \tag{1.12}$$

The resulting system statement consists of two equations

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{v}_k), \tag{1.13}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{n}_k), \tag{1.14}$$

named process and measurement models, respectively. In detail $\mathbf{v}_k$ and $\mathbf{n}_k$ are the process and the measurement noises and are

assumed white and mutually independent. Their distributions together with the state prior and the functions $\mathbf{f}$ and $\mathbf{h}$ are known.

The solution to Bayesian dynamic estimation is based on the computation of the hidden state pdf *after* the measurements observation; more precisely, since we aim at online estimation, we are interested in the posterior pdf

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \tag{1.15}$$

for a generic $k > 0$ and a sequential approach is required, i.e. $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ should be computed from $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ with constant time complexity.

In contexts denoted by the absence of real time requirements, we can also estimate a whole sequence jointly; if K is the final instant, the posterior pdf of interest is then

$$p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}). \tag{1.16}$$

After that the pdf of either eq. (1.15) or eq. (1.16) is available, Bayesian estimators like MAP, MMSE and LMMSE can be used, according to the definitions provided in the Sect. 1.1.1.

1.3.1 Theoretical solution

System (1.13)-(1.14) admits an elegant theoretical solution, expressed in the form of a sequential procedure; given posterior pdf (1.15) at the generic instant $k-1$ and a new measurement $\mathbf{y}_k$, we first compute the *prediction* pdf

$$p\left(\mathbf{x}_k|\mathbf{y}_{1:k-1}\right) = \int p\left(\mathbf{x}_k|\mathbf{x}_{k-1}\right) p\left(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}\right) \, \mathrm{d}\mathbf{x}_{k-1}, \tag{1.17}$$

and then the new posterior is provided by the *update* step

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p\left(\mathbf{y}_k|\mathbf{x}_k\right) p\left(\mathbf{x}_k|\mathbf{y}_{1:k-1}\right)}{p\left(\mathbf{y}_k|\mathbf{y}_{1:k-1}\right)}, \tag{1.18}$$

where

$$p\left(\mathbf{y}_k|\mathbf{y}_{1:k-1}\right) = \int p\left(\mathbf{y}_k|\mathbf{x}_k\right) p\left(\mathbf{x}_k|\mathbf{y}_{1:k-1}\right) \, \mathrm{d}\mathbf{x}_k \tag{1.19}$$

is a normalization factor and does not need to be computed. Equations (1.17)-(1.18) are clearly derived through a marginalization over $\mathbf{x}_{k-1}$ and Bayes theorem respectively, after the application of the Markov properties of eqs. (1.9)-(1.10). Nevertheless, practical implementations of this procedure are strongly case dependent: except very peculiar cases, suitable approximations are required.

1.3.2 Linear Gaussian system: Kalman Filter

In the following we will denote the estimator of $\mathbf{x}_k$ given the measurements $\mathbf{y}_{1:s}$ by $\hat{\mathbf{x}}_{k|s}$. Let us consider a Linear Gaussian (LG) Model

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{v}_k, \tag{1.20}$$
$$\mathbf{y}_k = G\mathbf{x}_k + \mathbf{n}_k, \tag{1.21}$$

with F and G known matrices of suitable dimensions, $\mathbf{v}_k \sim \mathcal{N}(0, Q)$ and $\mathbf{n}_k \sim \mathcal{N}(0, R)$. In this case the MMSE estimator for the state sequence is provided by a very popular algorithm: the Kalman Filter (KF). KF is a sequential algorithm which follows the prediction-update procedure. Indeed, a solution in closed form is possible because of a singular result on the recursive computation of the posterior pdf. In fact, it can be easily shown [Hay01] with recursive arguments that for a LG system the posterior pdf is Gaussian distributed, if the prior pdf was too. More in detail, if

$$p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \equiv \mathcal{N}\left(\hat{x}_{k-1|k-1}, P_{k-1|k-1}\right)$$

we have for the state pdf after the prediction step of eq. (1.17)

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \equiv \mathcal{N}\left(\hat{x}_{k|k-1}, P_{k|k-1}\right),$$

and for the pdf after the update step of eq. (1.18)

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \equiv \mathcal{N}\left(\hat{x}_{k|k}, P_{k|k}\right).$$

The starting point, of course, is that a Gaussian prior is given

$$p(\mathbf{x}_0) \equiv \mathcal{N}\left(\hat{x}_0, P_0\right).$$

Therefore, computing recursively the posterior pdf means to propagate its mean and covariance matrix over time.

KF does exactly this. The result, instant by instant, is an unbiased estimator that is the conditional mean $\hat{\mathbf{x}}_{k|k}$ of the posterior pdf. Recalling results in the Sect.1.1.1, this estimator corresponds to the MMSE estimator for $\mathbf{x}_k$. Its covariance matrix $P_{k|k}$ is also computed and is equivalent to the MSE. The algorithm is now briefly summarized.

Algorithm 1 (Kalman)**.**

For $k = 0$ set

- $\hat{\mathbf{x}}_0 = E[\mathbf{x}_0]$;
- $P_0 = E[(\mathbf{x}_0 - E[\mathbf{x}_0])(\mathbf{x}_0 - E[\mathbf{x}_0])']$.

For $k = 1, 2, \ldots$, compute:

- State prediction: $\hat{\mathbf{x}}_{k|k-1} = F\hat{\mathbf{x}}_{k-1|k-1}$;
- Covariance prediction: $P_{k|k-1} = FP_{k-1|k-1}F' + Q$;
- Kalman gain matrix: $K_k = P_{k|k-1}G'\left[GP_{k|k-1}G' + R\right]^{-1}$;
- State update: $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k\left(\mathbf{y}_k - G\hat{\mathbf{x}}_{k|k-1}\right)$;
- Covariance update: $P_{k|k} = (I - K_kG)\,P_{k|k-1}$.

Note that in this case the KF estimates are optimal also in the MAP sense, because posterior pdfs are Gaussian and their means coincide with their corresponding modes.

1.3.3 Alternative derivation of the Kalman Filter

The KF optimality can be stated also in an alternative form, that also corresponds to its first development and is useful in the sequential estimation in Non Linear Gaussian (NLG) and Non Linear Non Gaussian (NLNG) systems.

The posterior pdf of the state at k results in

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{x}_k, \mathbf{y}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}.$$

Neglecting $p(\mathbf{y}_k|\mathbf{y}_{1:k-1})$ because it is independent of $\mathbf{x}_k$, a further factorization of the left term yields

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \propto p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1}), \tag{1.22}$$

where white noise is assumed. Now if the LG model (1.20)-(1.21) is assumed, both pdfs on the right side of eq. (1.22) are Gaussians, viz.

$$\begin{aligned} p(\mathbf{y}_k|\mathbf{x}_k) &= \mathcal{N}\left(C\mathbf{x}_k, R_n\right), \\ p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) &= \mathcal{N}\left(\hat{\mathbf{x}}_{k|k-1}, P_{k|k-1}\right), \end{aligned}$$

where $\mathbf{x}_{k|k-1}$ and $P_{k|k-1}$ are computed like in Algorithm 1. The value $\mathbf{x}_k$ at which the posterior is maximized (or averaged) is clearly the minimum point of the cost function

$$J(\mathbf{x}_k) = (\mathbf{y}_k - C\mathbf{x}_k)' R_n^{-1} (\mathbf{y}_k - C\mathbf{x}_k) + (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})' P_{k|k-1}^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}), \tag{1.23}$$

that is the sum of the errors derived from the exploitation of the new measurement and from the prediction step, weighted on their respective covariance matrices. Similarly, we can define the batch cost function for $\mathbf{x}_{0:k-1}$

$$\begin{aligned} J(\mathbf{x}_{0:k}) = \sum_{s=1}^{k} &(\mathbf{y}_s - C\mathbf{x}_s)' R_n^{-1} (\mathbf{y}_s - C\mathbf{x}_s) + \\ &(\mathbf{x}_s - \hat{\mathbf{x}}_{s|s-1})' P_{s|s-1}^{-1} (\mathbf{x}_s - \hat{\mathbf{x}}_{s|s-1}). \end{aligned} \tag{1.24}$$

This derivation, although being less strong from a theoretical point of view, is much more general than the other one, since eq. (1.23) can be always stated provided that the dynamic system is known. This means that KF application does not require the Gaussian assumptions in the model (1.20)-(1.21), but on the other hand only in this case it achieves Bayesian optimality. In the general non linear non Gaussian (NLNG) framework the optimal solution does not exist in close form, and suboptimal approaches are being used.

Another important remark is that this derivation is particularly common when one deals with partially unknown systems: in the next chapter we will see that a reformulation of the metrics of eq. (1.23) is sufficient to extend KFs to those cases.

1.3.4 Non Linear Gaussian system: Kalman based filters

If the system is non linear but admits Gaussian additive noises

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{v}_k, \tag{1.25}$$
$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k) + \mathbf{n}_k, \tag{1.26}$$

the KF in the form of the Algorithm 1 cannot be applicated as it is. Indeed optimal algorithms in the MMSE or MAP sense do not exist in closed form and suboptimal solutions are in order. Next we present two algorithms, both based on KF, that tackle the non linearity in different ways: the Extended Kalman Filter and the Unscented Kalman Filter.

Extended Kalman Filter

The Extended Kalman Filter (EKF) is the most popular approach to Non Linear Gaussian (NLG) dynamic systems and is based on a first order approximation of the non linear functions. More precisely, a time variant linear system is obtained at each instant by stopping the Taylor series of the non linear functions at the

first order term, that is

$$\mathbf{x}_{k+1} = \tilde{F}_k \mathbf{x}_k + \mathbf{v}_k, \tag{1.27}$$
$$\mathbf{y}_k = \tilde{G}_k \mathbf{x}_k + \mathbf{n}_k, \tag{1.28}$$

where the matrices $\tilde{F}_k = \nabla_{\mathbf{x}_k} \mathbf{f}(\mathbf{x}_k)$ and $\tilde{G}_k = \nabla_{\mathbf{x}_k} \mathbf{g}(\mathbf{x}_k)$, given by the gradient of the model and measurement functions respectively, are evaluated at the best available estimation of $\mathbf{x}_k$.

Algorithm 2 (Extended Kalman)**.**

For $k = 0$ set

- $\hat{\mathbf{x}}_0 = E[\mathbf{x}_0]$;
- $P_0 = E[(\mathbf{x}_0 - E[\mathbf{x}_0])(\mathbf{x}_0 - E[\mathbf{x}_0])']$.

For $k = 1, 2, \ldots$, compute:

- Process model: $\tilde{F}_k = \nabla_{\mathbf{x}_k} \mathbf{f}(\mathbf{x}_k)|_{\hat{\mathbf{x}}_{k-1|k-1}}$;
- State prediction: $\hat{\mathbf{x}}_{k|k-1} = \tilde{F}_{k-1} \hat{\mathbf{x}}_{k-1|k-1}$;
- Covariance prediction: $P_{k|k-1} = \tilde{F}_{k-1} P_{k-1|k-1} \tilde{F}'_{k-1} + Q$;
- Measurement model: $\tilde{G}_k = \nabla_{\mathbf{x}_k} \mathbf{g}(\mathbf{x}_k)|_{\hat{\mathbf{x}}_{k|k-1}}$;
- Kalman gain matrix: $K_k = P_{k|k-1} \tilde{G}'_k \left[\tilde{G}_k P_{k|k-1} \tilde{G}'_k + R \right]^{-1}$;
- State update: $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k \left(\mathbf{y}_k - \tilde{G}_k \mathbf{x}_{k|k-1} \right)$;
- Covariance update: $P_{k|k} = \left(I - K_k \tilde{G}_k \right) P_{k|k-1}$.

EKF is not very accurate and it should be avoided when high performance is required. In fact, EKF is based on the assumption that each posterior pdf is approximated by Gaussians, whose mean and covariance matrix are propagated. In a NLG model, indeed, EKF ensures the exact propagation of the mean only, while the estimated covariance matrix is not equal to the true one. This leads to unpredictable errors and, furthermore, an unstable behaviour could be experienced.

On the other hand, if this is not the case, EKF ensures simplicity and low complexity and this is the reason why EKF is still very used.

Unscented Kalman Filter

A different approach to NLG systems is provided by UKF, proposed by [JUDW95] in 1995, that is based on a deterministic sampling of the state space. Like EKF, this method is also based on the Gaussian posterior approximation, whose mean and covariance matrix are propagated. Nevertheless, it is generally more accurate than EKF because it can be shown that EKF propagates exactly only the mean (first order approximation of the posterior), while UKF can propagate both exact mean and covariance matrix (second order approximation of the posterior).

In the practice the performances improvement, if any, is paid with an increment in complexity, since multiple sampling points must be handled. The net result is that EKF is still used wherever it shows a stable and accurate enough behaviour.

The theoretical setup in UKF is based on the Unscented Transformation (UT), that is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation. A deeper analysis of both UT and UKF goes beyond the thesis' concern and the interested reader can refer to [Hay01, Ch. 7].

1.3.5 Non Linear Non Gaussian systems

In the general case of NLNG systems, represented by the model (1.13)-(1.14) and here restated for clarity

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{v}_k), \tag{1.29}$$
$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{n}_k), \tag{1.30}$$

the KF based algorithms should be avoided. As the main problem is related to non Gaussianity, the same arguments are still valid for the class of LNG systems.

Particle Filters (PFs) are drawing more and more interest concerning the solution of general dynamic systems [AMGC02]- [FHL$^+$03]. Even if conceptually simple, their exploitation has been put off until this last decade for technological issues: their heavy computational burden requires an amount of hardware resources that was made available only recently.

PFs deal with non linearity and, above all, with non Gaussianity by sampling in a random way the state space. For this reason they belong to the family of Monte Carlo sampling techniques.

In PFs, the posterior pdf is approximated by

$$p\left(\mathbf{x}_k|\mathbf{y}_{1:k}\right) \approx \sum_{i=1}^{N_p} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i), \tag{1.31}$$

where $\{\mathbf{x}_k^i\}_{i=1,\ldots,N_p}$ is the set of state hypotheses or particles. Two main problems arise: i) how to (randomly) sample particles and ii) how to compute weights in order to obtain the asymptotic convergence of the sampled distribution to the real one.

The main issue is that it is not always easy to draw samples from a generic distribution. The first algorithm we are going to present is not used in the practice but solves this problem. After that, the most popular PF will be described.

1.3.6 Sequential Importance Sampling

Sequential Importance Sampling (SIS) algorithm is a PF that overcomes the problem of sampling from a generic distribution

by means of the Importance Sampling.

Suppose that the estimatee $\mathbf{x}$ has a distribution $p(\mathbf{x})$ from which it is difficult to draw realizations but that is easy to evaluate in some points of its domain. Therefore, instead of drawing particles $\{\mathbf{x}_k^i\}_{i=1,\dots,N_p}$ from $p(\mathbf{x})$ and approximate the posterior as in eq. (1.31) with equal weights

$$w^i = \frac{1}{N_p},$$

we rather draw them from an *importance density* $q(x)$, chosen in a suitable way. In this case the approximation of eq. (1.31) is fulfilled with

$$w^i \propto \frac{p(x^i)}{q(x^i)}.$$

An iteration of the SIS algorithm is summarized in Alg. 3; its sequential extention is then straightforward.

Algorithm 3 (Sequential Importance Sampling)**.**

At $k > 0$

- For any $i = 1, \dots, N_p$
  - Draw $\mathbf{x}_k^i \sim q(\mathbf{x}_k | \mathbf{x}_{1:k-1}^i, \mathbf{y}_k)$;
  - Assign the particle with a weight

$$w_k^i \propto \frac{p(\mathbf{x}_k^i | \mathbf{x}_{1:k-1}^i, \mathbf{y}_k)}{q(\mathbf{x}_k^i | \mathbf{x}_{1:k-1}^i, \mathbf{y}_k)};$$

- Normalize the weights such that their sum is 1.

The choice of the sampling function $q(\cdot)$ depends on the particular application and eventually contains the new measurement $\mathbf{y}_k$. If so, the PF is also called 'likelihood' PF.

1.3.7 Sequential Importance Sampling with Resampling

SIS has a big drawback that prevents people from using it. This problem is known like 'particle degeneracy' and consists in the fact that after few steps of the sequential procedure most particles will be far from the truth (and have nearly zero weights) and thus the 'effective' number of particles is much lower than N_p. A good measurement for the number of effective particles is supposed to be [AMGC02]

$$\bar{N}_{eff} = \frac{1}{\sum_{i=1}^{N_p}(w_k^i)^2}. \tag{1.32}$$

Sequential Importance Sampling with Resampling (SIR) overcomes the particle degeneracy problem by means of a resampling step. The idea is to resample the particles in accordance with a probability measure proportional to the weights when $\bar{N}_{eff}$ is lower than a threshold N_T. A common scheme for SIR is given below.

Algorithm 4 (Sequential Importance Resampling)**.**

At $k > 0$

- For any $i = 1, \ldots, N_p$
  - Draw $\mathbf{x}_k^i \sim q(\mathbf{x}_k|\mathbf{x}_{1:k-1}^i, \mathbf{y}_k)$;
  - Assign the particle with a weight

$$w_k^i \propto \frac{p(\mathbf{x}_k^i|\mathbf{x}_{1:k-1}^i, \mathbf{y}_k)}{q(\mathbf{x}_k^i|\mathbf{x}_{1:k-1}^i, \mathbf{y}_k)};$$

- Normalize the weights such that their sum is 1;
- Compute $\bar{N}_{eff}$ as in eq. (1.32);

- If $\bar{N}_{eff} < N_T$, resample the particles according to a multinomial distribution using the weights as probabilities, and then set uniform weights

$$w_k^i = \frac{1}{N_p}.$$

A common threshold is $N_T = \frac{2}{3}N_p$; note that the resampling introduces further uncertainty, so that, even if it allows to avoid degeneracy, the best pdf approximation is the one before resampling.

Many variants of SIR have been proposed over the past years to overcome some kinds of drawbacks. For example Auxiliary Sampling Importance Resampling (ASIR) is a type of likelihood PF in which a first weight is assigned to the particle relative to the new datum likelihood and, after a resampling step, classic SIR is applicated. The Regularized PF, instead, substitutes 'impulsive' functions of the particles in eq. (1.31) with smooth functions taken from a suitable kernel (Gaussian or other), as a further remedy for particle degeneracy.

All PF variants, however, show some advantages only in particular applications, so that the choice of the best PF is strongly dependent of the application. Further details can be found in [AMGC02] that, although it was written in 2002, is still today the most cited and popular work about particle filters, and in [RAG04] that is a collection of articles about the recent developments in this field.

We conclude this section with the description of an interesting technique that is very used today to reduce PF complexity: Rao-Blackwellized Particle Filters.

1.3.8 Rao-Blackwellized Particle Filters

Whatever application and PF algorithm are handled, the critical point lays in the state space sampling. What has been noted by

several authors is that the processing complexity of PFs grows more than linearly with the state space cardinality, that should be reduced as much as possible.

Rao-Blackwellized Particle Filters (RBPFs) are the most cited way to reduce the sampling complexity. Let $\mathcal{X}$ denote the state space of caridinality $|\mathcal{X}|$. To sample $\mathcal{X}$ means to draw a $|\mathcal{X}|$-dimensional vector $\mathbf{x} \in \mathcal{X}$. The RBPF exploits constraints over $\mathcal{X}$, if any, to decompose the state space in two disjointed subspaces, say $\mathcal{X}_S$ and $\mathcal{X}_D$, with

$$\begin{aligned} \mathcal{X}_S \cap \mathcal{X}_D &= 0, \\ \mathcal{X}_S \cup \mathcal{X}_D &= \mathcal{X}, \\ |\mathcal{X}_S| + |\mathcal{X}_D| &= |\mathcal{X}|, \end{aligned}$$

where 0 is here the empty set. Each particle sampling takes place in two subsequent steps:

- first sample into $\mathcal{X}_S$ to obtain the $i-th$ state hypothesis $\mathbf{x}_S^i$;
- then, given $\mathbf{x}_S^i$, determine the corresponding $\mathbf{x}_D^i \in \mathcal{X}_D$ in a deterministic way.

The advantage of this scheme is straightforward: only a part of the state space is effectively sampled and then deterministic techniques are employed to complete the particle. It is intuitive that it cannot be done in general, but only in the presence of some conditions. The most common cases are

- $\mathbf{x}_D^i \in \mathcal{X}_D$ is deterministically obtained given $\mathbf{x}_S^i \in \mathcal{X}_S$, viz.

$$\mathbf{x}_D^i = \mathbf{f}(\mathbf{x}_S^i);$$

- an optimal (or even sub-optimal) estimation $\mathbf{x}_D^i$ according to a suitable criterium is achievable given $\mathbf{x}_S^i$, ex. by means of KF-based algorithms.

1.4 Sequential PCRLB

The PCRLB of Sect. 1.2.1 can fit dynamic estimation by means of a recursive procedure that was presented by Tichavsky et al. in [TMN98] and it is here briefly reported. A general extension to the presence of unknown parameters in the dynamic models is also reported. The notation about matrices used throughout this section is a bit less intuitive than before and is shown in App. B.

1.4.1 Bound with known models

Let $\mathbf{\Theta}_k = \{\boldsymbol{\theta}_i\}_{i=0,2,\dots,k}$ be the target trajectory until the instant k, $\mathbf{Y}_k = \{\mathbf{y}_i\}_{i=1,2,\dots,k}$ the set of measurements, $\mathbf{X}_k = [\mathbf{\Theta}_k^T, \dot{\mathbf{\Theta}}_k^T]^T = \{\mathbf{x}_i\}_{i=1,2,\dots,k}$ the state vector sequence and $\hat{\mathbf{x}}_{k|k} = [\hat{\boldsymbol{\theta}}_{k|k}^T, \hat{\dot{\boldsymbol{\theta}}}_{k|k}^T]^T$ the estimated state (given $\mathbf{Y}_k$). We first start by computing the Fisher Information Matrix (FIM), whose inverse is the CRLB, relative to the whole trajectory $\mathbf{X}_k$, using $\mathbf{Y}_k$ like data. Since $\mathbf{X}_k$ is a random parameter, the FIM is defined like

$$I_k = E\left\{-\nabla_{\mathbf{X}_k}[\nabla_{\mathbf{X}_k} \log p(\mathbf{X}_k, \mathbf{Y}_k)]^T\right\}, \tag{1.33}$$

where the expectations are over both $\mathbf{X}_k$ and $\mathbf{Y}_k$, and it holds true that in mild hypotheses

$$E\left\{(\hat{\mathbf{X}}_{k|k} - \mathbf{X}_k)(\hat{\mathbf{X}}_{k|k} - \mathbf{X}_k)^T\right\} \geq I_k^{-1}, \tag{1.34}$$

for any trajectory estimates $\hat{\mathbf{X}}_{k|k}$.

We are actually interested in the sequential state estimation $\hat{\mathbf{x}}_{k|k}$. In this case, it can be derived from (1.33) and (1.34) that the Cramer Rao bound becomes

$$P_k = E\left[(\hat{\mathbf{x}}_{k|k} - \mathbf{x}_k)(\hat{\mathbf{x}}_{k|k} - \mathbf{x}_k)^T\right] \geq J_k^{-1}. \tag{1.35}$$

The inequality in eq. (1.35) means that the difference $P_k - J_k^{-1}$ is a positive semidefinite matrix. The information matrix J_k is $n_x \times n_x$ sized, with n_x the number of state variables, and is obtained

from I_k. In [TMN98] an effective sequential computation of J_k is performed by

$$J_{k+1} = D_k^{22} - D_k^{21}(J_k + D_k^{11})^{-1}D_k^{12}, k \geq 0, \tag{1.36}$$

where the matrices D are square, sized $n_x \times n_x$ and independent of k

$$\begin{cases} D_k^{11} &= E\left\{-\nabla_{\mathbf{x}_k}\left[\nabla_{\mathbf{x}_k}^T \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right]\right\}, \\ D_k^{12} &= E\left\{-\nabla_{\mathbf{x}_k}\left[\nabla_{\mathbf{x}_{k+1}}^T \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right]\right\}, \\ D_k^{21} &= E\left\{-\nabla_{\mathbf{x}_{k+1}}\left[\nabla_{\mathbf{x}_k}^T \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right]\right\}, \\ D_k^{22} &= E\left\{-\nabla_{\mathbf{x}_{k+1}}\left[\nabla_{\mathbf{x}_{k+1}}^T \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right]\right\} + \\ & \quad E\left\{-\nabla_{\mathbf{x}_{k+1}}\left[\nabla_{\mathbf{x}_{k+1}}^T \log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})\right]\right\}, \end{cases}$$

where the expectations are done w.r.t. $\mathbf{x}_k$, $\mathbf{x}_{k+1}$ and, in the last line, $\mathbf{y}_{k+1}$.

REMARK 1. Eq. 1.36 represents a recursive way in which the PCRLB of Sect. 1.2.1 can be computed for dynamic models and thus the conditions required for the validity of the bound are exactly the same, but now they are applied to the whole trajectory.

REMARK 2. In the case of LG systems, when the KFs are optimal as they minimize the MSE of the estimator, the FIM of eq. (1.36) is the inverse of the estimator covariance matrix $P_{k+1|k+1}$ of Alg. 1. In fact it is well known that the sequence of matrices $P_{k|k}$ can be computed in advance, being it independent of the true trajectory followed by the user. For the other systems, this procedure is very similar to the EKF; the only difference is that in this latter the gradients of the model functions are evaluated at the estimated state, rather than at the true state.

REMARK 3. The prior distribution of the state is typically considered a Gaussian whose parameters reflect the knowledge about the starting conditions; its FIM J_0, in this case, is the inverse of the prior covariance matrix. A uniform prior pdf cannot be used, since it is discontinuous on its lips, unless different bounds are being used. At this aim, the Ziv-Zakai Bound (ZZB) (presented

in [ZZ69], improved and extended by other authors) is similar to the CRLB, but it allows hard limitations on the prior distribution. Unluckily it is still not avilable for sequential procedures and thus it is not suitable in dynamic problems.

REMARK 4. The computation of the matrices D can be done exactly only in some trivial cases (e.g. for a LG model), but normally some of them require numerical solutions, e.g. Monte Carlo and deterministic sampling techniques[4].

1.4.2 Bound with partially unknown models

In the case of partially unknown models we need some further insight. We deal with a deterministic and unknown static parameters vector $\mathbf{p}$ which we include into the previously used joint distribution of the state trajectory and the data. Stating for simplicity a fictitious dynamic model for $\mathbf{p}$

$$\mathbf{p}_{k+1} = \mathbf{p}_k,$$

we define accordingly the Fisher information matrix of eq. (1.33):

$$I_k = E\left\{-\nabla_{\{\mathbf{X}_k,\mathbf{p}\}}[\nabla_{\{\mathbf{X}_k,\mathbf{p}\}}\log p(\mathbf{X}_k,\mathbf{p},\mathbf{Y}_k)]^T\right\}, \tag{1.37}$$

where the expectations are over $\mathbf{X}_k, \mathbf{p}$ and $\mathbf{Y}_k$. The solution requires defining the following matrices

$$\begin{cases} H_k^{11} &= E\left\{-\nabla_{\mathbf{x}_k}[\nabla^T_{\mathbf{x}_k}\log s_k]\right\}, \\ H_k^{12} &= E\left\{-\nabla_{\mathbf{x}_k}[\nabla^T_{\mathbf{p}_k}\log s_k]\right\}, \\ H_k^{13} &= E\left\{-\nabla_{\mathbf{x}_k}[\nabla^T_{\mathbf{x}_{k+1}}\log s_k]\right\}, \\ H_k^{22} &= E\left\{-\nabla_{\mathbf{p}_k}[\nabla^T_{\mathbf{p}_k}\log s_k]\right\}, \\ H_k^{23} &= E\left\{-\nabla_{\mathbf{p}_k}[\nabla^T_{\mathbf{x}_{k+1}}\log s_k]\right\}, \\ H_k^{33} &= E\left\{-\nabla_{\mathbf{x}_{k+1}}[\nabla^T_{\mathbf{x}_{k+1}}\log s_k]\right\}, \end{cases}$$

[4]The employment of the unscented sampling in the computation of the matrices is performed with interesting results in [LLP09].

where

$$\begin{aligned} s_k &= p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{p}_k)p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{p}_k) \\ &= p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}, \mathbf{p}_k), \end{aligned} \tag{1.38}$$

and the expectations are over $\mathbf{x}_k$, $\mathbf{x}_{k+1}$, $\mathbf{p}_k$ and $\mathbf{y}_{k+1}$.

The required FIM results in the block matrix

$$J_k = \begin{bmatrix} J_k^{11} & J_k^{12} \\ J_k^{21} & J_k^{22} \end{bmatrix},$$

where the blocks are computed recursively

$$J_{k+1}^{11} = H_k^{33} - (H_k^{13})^T[J_k^{11} + H_k^{11}]^{-1}H_k^{13}, \tag{1.39}$$

$$J_{k+1}^{12} = [H_k^{23}]^T - (H_k^{13})^T[J_k^{11} + H_k^{11}]^{-1}(J_k^{12} + H_k^{12}), \tag{1.40}$$

$$J_{k+1}^{21} = [J_{k+1}^{12}]^T, \tag{1.41}$$

$$J_{k+1}^{22} = J_k^{22} + H_k^{22} - (J_k^{12} + H_k^{12})^T[J_k^{11} + H_k^{11}]^{-1}(J_k^{12} + H_k^{12}). \tag{1.42}$$

The remarks made in the case with known models are still valid here. A further relevant consideration is that the block relative to the Fisher information of the parameter normally diverges

$$J_k^{22} \overset{k}{\to} \infty.$$

This is due to the fact that the information about the static parameters grows unbounded and thus an efficient and consistent estimator, if any, would produce a perfect output asymptotically.

1.5 Concluding remarks

In the end, we would like to stress that all estimation techniques are usually based on different models of the same reality. Since the final aim in estimation is the real object, the model, and hence the estimation technique, is just a design choice performed by whom is addressing the estimation problem. This choice is not unique at all and is usually the main challenge in an engineering problem, in which the designer's experience is essential.

Chapter 2

A Survey of WLAN Indoor Positioning Techniques

Over the last decades several researchers have investigated indoor navigation, exploring any possible technologies and approaches. The topic is still far from a satisfying, even sub-optimal, solution.

An incredible amount of factors makes indoor navigation difficult to meet. The modelization of the environment, the user motion, the possibility to use some devices and their real technical features are all issues mixing into the analysis of the problem in an inextricable way.

But, perhaps, the most challenging point is that such factors change dramatically in different experimental testbeds and even for the same testbed over time. For this reason, the simplest techniques, meant to be more general, keep prevailing over the others, above all in Industry, while a wider variety of approaches has been proposed in Academy.

In this chapter we provide an overview of the state of the art in indoor localization. At the moment, we only deal with classic localization: estimation of side parameters or maps is put off to the next chapters.

Since now, when we describe an approach we will always highlight the reasons why it has been proposed and what kind of needs are met. In particular, each argument is considered in the light of the next chapters' content. This can appear limitative, but on the other hand it provides a suitable base for the thesis' work.

Anyway, a point should be clear in our minds: whatever models and techniques are used, the underlying problem is always the same, i.e. somebody, walking around, must be detected and located. The design choices always pursue a suitable ratio amongst attainance to real world, complexity and availability of resources.

After a brief analysis of the main ad-hoc localization systems in the Sect. 2.1, we will pay our attention mostly to the reuse of RSS signals in WLANs (Sect. 2.2), whose most valuable example is IEEE 802.11 (WiFi). We will formalize the problem (Sects. 2.3 - 2.4) and then WLAN localization techniques are classified in static (Sect. 2.5) and dynamic systems (Sect. 2.6); this choice reflects not only their deep technical differences, but also the opposite

point in which the trade-off between performance and complexity is fixed. In the end, in Sect. 2.7 a brief description of the technologies more related to this thesis' contribution is given and in Sect. 2.8 we make our last considerations.

2.1 Indoor localization: earlier works

One of the first approaches for indoor localization was based on Infrared (IR) signals. In the system ActiveBadge [WHFaG92], developed at Olivetti Research Laboratory (now AT&T Cambridge) between 1989 and 1992, a badge worn by the user sends a IR pulse every 10 seconds[1]. Some sensors can detect those pulses, recognize the users and transmit information to a central server. ActiveBadge has been adopted by many universities in Great Britain and USA; its main advantage lays in its low costs, but it is affected by some drawbacks. Mainly the point is that receivers act as proximity sensors and thus the estimated user's position is identified with the receiver's position. Good quality performances can be only obtained as receivers' density is high enough and a suitable power management is allowed. Some versions of badges are represented in Fig. 2.1.

Other approaches employ the time of arrivals of ultrasound pulses. Many problems arise from reflections, from the lack of LOS and when the distance between transmitter and receiver exceeds few meters. For these reasons it is more reliable for the off-the-shelf systems to make use of combined RF-ultrasound signals and to measure the time between the arrivals of these two pulses. Cricket [PCB00], developed at MIT since 2000[2], and Active Bat [HHS+99], developed at Cambridge since 1997, are the major examples of ultrasound based locating systems.

[1] In the original version the badge only sent a 5 bit code on identification purposes; in the most recent versions a bi-directional communication is allowed and the badge is equipped with a microprocessor.

[2] Cricket current version is dated 2004.

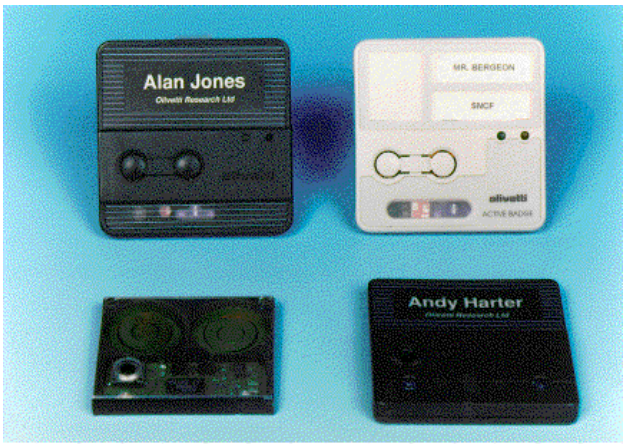


Figure 2.1 Active Badge modules: bottom left, the first version, with a unique five bit code; bottom right, the second version, with a ten bit code; top left, the third, current, version, with a forty-eight bit code, bi-directional capabilities, and an on-board 87C751 microprocessor. Source: *http://www.cl.cam.ac.uk*.

Spot On [HVBW01] and LandMarc [NLLP04] are the most famous systems based on the use of active RFIDs. Basically, some RFID readers in known locations can detect an active tag worn by a close user and measure the power received. Trilateration is thus employed to estimate the user's location.

2.2 RSS based localization: why?

It is worthy to note that all of the systems described above, although quite costly, are not used for communication tasks, being them dedicated to localization. The use of opportunistic signals from communication systems, instead, takes advantage from infrastuctures that are already available and constitutes the cheapest solution to the localization problem indoors.

In this context, the use of RSS measurements within wireless communication systems has drawn a constant interest over the last decade. The main idea is to extract distance information from the RF signals used in many wireless communication systems and then

to use trilateration schemes to localize the user.

Outdoors, this idea is being explored by mobile telephony. 4G mobile networks are expected to provide their subscribers with location aware services and thus must be equipped with localization capabilities, either stand alone or in cooperation with GNSS. Their precision, anyway, is too coarse for indoor environments. The exploitation of WLANs appears more promising in such contexts in order to obtain a fine user's localization inside a building by a simple reuse of already available data. The most common example is given by IEEE 802.11 (WiFi) technology, capable of performing localization without any additive hardwares.

2.3 WLAN indoor positioning

Some formalization is now needed. To preserve a good degree of readibility, we provide here only a minimal setup, valid all over the thesis, postponing further details until they are used.

Throughout this thesis we will assume that N_{US} users and N_{AP} APs are present in the area under surveillance, say $\mathcal{A} \in \mathcal{R}^2$, limited in the plane and a Cartesian coordinates system is defined. Time is discrete, due to the use of digital processing techniques, and is denoted by the variable $k = 0, 1, 2, \ldots$. The *i-th* user's position at k is the vector $\boldsymbol{\theta}_{i,k} \in \mathcal{A}$, where subscript i is usually omitted in single user frameworks, while the $j-th$ AP's position is denoted by $\mathbf{x}_{j,AP} \in \mathcal{A}$ and assumed still over time[3]. Moreover each AP is identified by a unique *ID*, e.g. the MAC address or some local label.

The RSS measurements vector $\mathbf{s}_k$ available at k is composed of at most N_{AP} tuples whose $j-th$ element is

$$s_{j,k} =< \mathrm{ID}_j, y_{1,j,k}, \ldots, y_{N_{US},j,k} >,\ j = 1, \ldots, N_{AP} \tag{2.1}$$

where $y_{i,j,k}$ is the RSS, in dBm if not stated otherwise, measured by the $i-th$ user and delivered by the $j-th$ AP at k. In absence

[3]The pointwise approximation for both users and APs is extremely common in localization; if the size of the user is small compared to the accuracy achieved this is not limiting.

of ambiguity, we will use the vector $\mathbf{y}_{i,k} = \{y_{i,j,k}\}_{j=1,\dots,N_{AP}}$ with implicit reference to IDs. Again, the subscript i will be omitted in single user frameworks.

Note that in eq. (2.1) the ID does not imply the knowledge of the AP's position. It only refers to the fact that measurements are always 'labeled', but some information about APs can be missing. More in general, we include a set of information about AP's and radio propagation into a WLAN map $\mathbf{W}$, whose exact composition is case sensitive and is put off until real cases are undertaken.

A similar concept holds for the building map, say $\mathbf{M}$, which can consist of walls, rooms, furniture, with a degree of refinement that is strictly case dependent.

In the end, we will consider for simplicity the case $N_{US} = 1$, if not stated otherwise.

2.3.1 Application architecture: server based vs. server free

RSS measurements can be processed either by the user or by the network. In the former case, namely self-positioning, problems as computational and memory capabilities of the user device (for example a smart phone) or its limited battery life should be cared, yielding the choice of simple and time saving algorithms. On the other hand this approach can be very effective in the case of multitarget positioning, since each user can compute its own position regardless of the others' presence. In the latter case the measurements can be transmitted using the WLAN network itself, and a location server can estimate the user's location. Meaningfully a server-based architecture, being less constrained in terms of energy consumption and computational capabilities than a mobile user, allows the application of more complicated techniques which account for the environmental variability. However, multitarget positioning is more critical in this scenario. In [ABG+10] an example of the potential of a server based architecture is provided: in this case the diffraction effects due to the presence of other users (cars in a parking lot) are tackled.

2.4 Indoor RF channel

The indoor RF channel is characterized by so several effects that it is almost impossible to list them as well as to cope with them. That is why we focus our attention on those issues that are reported in the literature as the most relevant for our aims.

The first effect we consider is key in RSS-based indoor positioning. While the RF signal is moving forward, its energy spreads on a larger surface and thus the energy absorbed by the receiving antenna is inversely proportional to the propagation distance. Unluckily this effect is deterministic only in ideal conditions but in a real scenario it is affected by several unpredictable causes.

Indoors, the effect of small objects or moving people is to randomize the signal propagation. In fact they cause a plethora of events, like (multiple) reflections, diffractions and attenuations which depend on the obstacle size and composition in relation to the operative frequency. The combination af all these effects leads to multipath and, thus, to fading [Par00]; an example of RSS measurements plotted against the distance between user and APS is in Fig. 2.2. Anyway, a separation is necessary between the cases of slow and rapid fluctuations of the signals, since different characterizations are usually employed. A deeper analysis of fading is proposed in the Ch. 3; here, after an introduction on the signal features at the receiver, we describe some popular propagation models.

2.4.1 Signal amplitude

In a digital transmission, a popular and effecive way to modelize the effects due to fading is to multiply the transmitted signal by a random process relative to the presence of multipath components. In terms of complex envelopes, the received signal $\tilde{u}(t)$ can be written as [BB99, p.699]

$$\tilde{u}(t) = r(t)e^{j\theta(t)}\tilde{s}(t) + \tilde{n}(t),$$

where $r(t)e^{j\theta(t)}$ is the cited complex process and $\tilde{n}(t)$ is a complex Gaussian noise with zero mean and PSD equal to N_0.

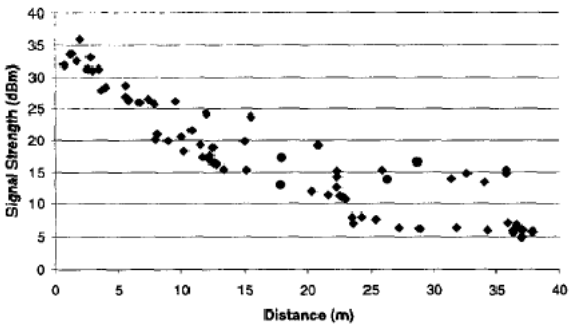


Figure 2.2 Path loss model validation with noise in evidence; on the abscissa we report the distance between user and AP. Borrowed from [BP00b].

Here, we assume a (frequency) nonselective fading, i.e the coherence bandwidth B_{cb} of the channel is much greater than the signal bandwidth W, leading to a slowly changing fading. In this hypothesis the fading term can be considered constant within the signal transmission. Furthermore the constant phase can be estimated with sufficient accuracy, yielding coherent detection. Consequently, the complex envelope $\tilde{u}(t)$ of the received signal can be now written as [BB99]

$$\tilde{u}(t) = r\tilde{s}(t) + \tilde{n}(t).$$

Note that this model differs from fading free transmission since r is a random rather than a deterministic, even unknown, attenuation. Accordingly, the SNR can be defined as [SA05]

$$\gamma = \frac{r^2 E_S}{N_0},$$

wherein E_s is the energy of $s(t)$.

The mean SNR $E[\gamma]$ is

$$E[\gamma] = \frac{E[r^2] E_S}{N_0}$$

in which the mean squared value of r, depending on the adopted distribution, is the mean power expected in the location under

test. Indeed we focus on the fading effect only, so our performance figures could neglect the symbol energy and the PSD of noise, considering them constant. In other words, for us

$$SNR = E[\gamma] = E[r^2].$$

In many cases the decay models are given in terms of the power in dB

$$y = 20 \log_{10} r \stackrel{\triangle}{=} 2\xi \ln r = \zeta \ln r,$$

where $\xi = \zeta/2 = 10/\ln 10$.

2.4.2 Propagation models

Exploitation of RSS measurements has prompted research about wireless RF signals propagation. The reference law in this case is known as Friis formula and describes the power attenuation in the free space signal propagation. Given a power P_T delivered by an antenna with gain $G_T(\phi)$ (ϕ is the direction of interest), the power P_r received at distance d by another antenna with gain G_R is

$$P_r = G_T(\phi) G_R(\phi) \left(\frac{\lambda}{4\pi d} \right)^2 P_T, \tag{2.2}$$

with λ wavelength of the propagated EMF and provided that far field conditions hold[4]. Eq. (2.2), depicted in Fig. 2.3, represents a watershed in wireless communications and is still today fundamental in research. But propagation in real world is different from what it is foretold in the free space condition and required further investigation as mobile telephony systems began a reality. As a result, most works are relative to the frequency bands used in telephony, from 900 MHZ up to 2 GHz, which are a bit lower than our frequencies of interest (ISM band at about 2.45 GHz).

Indoor propagation is also investigated with two scopes: penetration of mobile telephony signals in buildings and coverage of internal WLANs.

[4]In the ISM band, for example, $\lambda \sim 10$ cm, so that after a couple of meters the assumption is true.

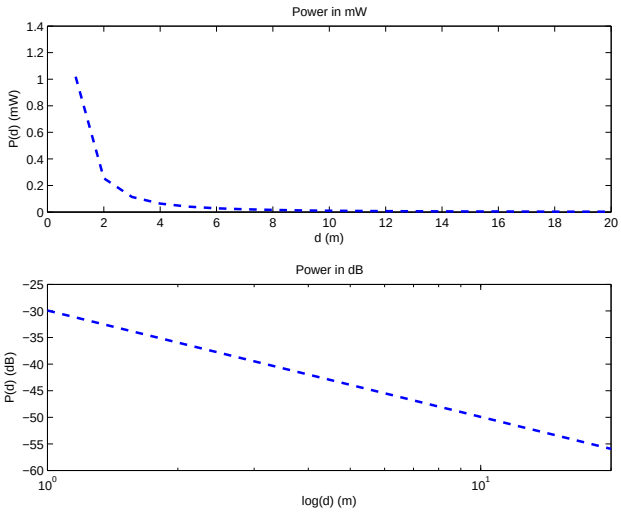


Figure 2.3 Path loss model without wall effects - eq.(2.3); on the top power in mW and linear axes, on the bottom power in dB against distance in log scale.

The basic idea is to extend eq. (2.2) in order to account for as many effects as possible in a simple way. Taking the logarithm, the received power is expressed by

$$P_{r,dB} = h - 10\alpha \log\left(\frac{d}{d_0}\right), \tag{2.3}$$

where h includes the effects of antennas and frequency together with the delivered power[5], d_0 is a reference distance (e.g. far field limit) and $\alpha = 1 \div 4$ is the decaying factor ($\alpha = 2$ in free space).

Other deterministic effects such as floor or wall attenuation can be met with additive loss factors. In the latter case, the *Wall Attenuation Factor* - WAF - model has been presented in [BP00b] and used by many other authors

$$P_{r,dB} = h - 10\alpha \log\left(\frac{d}{d_0}\right) - c \times WAF, \tag{2.4}$$

[5]That is why we prefer to avoid common definitions like the power delivered at distance d_0.

where WAF is the power loss due to a single wall and c is the number nW of walls between transmitter and receiver up to a limit, since the effect of new walls is less and less meaningful. As for the floors attenuation, a fully similar model was proposed in [Par00] and here omitted. In both cases the loss is heavily dependent of the building materials: this is why their values are case dependent.

The model of eq. (2.3) is useful to describe the power law in the far field conditions, that is $1 \div 2$ m at microwaves. At closer distances the model is not accurate at all and it diverges for $d \to 0$. To avoid degeneracies in a real situation some tricks must be used, for example we can assume that the AP is raised of $1 \div 2$ m on the user's plane. An alternative propagation model rarely adopted in the literature that overcomes this problem is presented in [NV05] and also used in [BMM11]

$$P_r = \frac{h}{1 + \beta d^{\alpha}}, \tag{2.5}$$

where β is a scaling factor for the distances. According to this model, in far field conditions the propagation law is the same as in the Friis formula, while for $d \to 0$ the field is limited by h and does not diverge. Furthermore, this model is remarkably continuous and infinitely derivable for $d > 0$.

Other models are less common in the literature and can be found in [Rap96].

2.5 Static localization

In static localization users are assumed to be somewhere in the area under surveillance and information about their previous trajectory is not employed at all. Another implicit hypothesis is that the user keeps still at least during sensing and processing operations. The time indication is only used to indicize a sequence of independent estimations.

Static localization is reasonable when saving hardware resources is a must and easy-to-go techniques are employed; historically it

was the first framework to be investigated and today it is pretty common in self-positioning where energy constrained devices are used, e.g. smartphones.

2.5.1 Ray-tracing

In some niche applications where hardware resources are not a problem at all, the most performing solution to static localization is ray tracing. It denotes a family of techniques based on the optics approximation: the EMF is partitioned in a beam of mutually independent rays[6] and effects like diffraction and reflections can be taken into account by providing the software with detailed information on the environment. The hardware requirements of such techniques are usually high, above all in indoor environments, due to the contemporary presence of a large number of rays and to their interaction with several objects. However approaches with different compromises between complexity and performance are available.

Other (even more complicated) techniques avoid optics approximation, as they look for a numerical solution of Maxwell equations at RF frequencies of the EMF.

For a complete analysis of ray tracing see [Par00, Sect 7.4] and references therein. We now say that besides the already cited computational requirements, softwares implementing ray tracing are typically protected by copyrights and are thus quite expensive. Therefore most applications should be tackled by other less expensive, despite sufficiently accurate techniques.

2.5.2 Fingerprinting techniques

With no doubts, the approach 'leader' in WLAN static localization is fingerprinting, due to its simplicity and effectiveness.

Several variants have been proposed which differ in some part, but the central point is always the radio 'fingerprint'. The finger-

[6] This approximation is almost exact only at optical frequencies but in this case it is used at RF frequencies.

print of a place is essentially a set of RSS measurements collected from there. Some meaningful locations are selected in the area under surveillance and their fingerprints are stored in a database called Radio Map (RM) during an off-line training stage.

The fingerprint of the position to estimate in the locationing stage is thus compared with those stored in the RM, in order to find the 'nearest' one in some sense. K-Nearest Neighbours (k-NN) algorithm, described in the Sect. 1.1.2, fits this scenario very well, but complexity issues limit its applicability. A large dataset to store and handle is a nightmare for terminals with few hardware resources. K-NN based solutions which face this problem try to include into the RM only a 'summary' of the training measurements without losing precision. The more concise the summary is, the more versatile the resulting algorithm is.

RADAR: simplicity rules

RADAR was presented in 2000 by [BP00b] and still today is the most popular k-NN based algorithm. It employs $k = 1$ (in [CH67] the admissibility of 1-NN is also stressed) and is based on an approximation of the RM: the fingerprints stored are not represented by the list of all measurements collected, but by their means. Thus, any training point is characterized by only a vector of mean RSS values disregarding the true number of measurements collected.

From a different point of view, RADAR is equivalent to a first order characterization of the empirical distributions at every training point. It is evident that higher order features could be extracted, reaching a halfway solution between standard 1-NN and RADAR (see Sect. 2.5.3). Nevertheless, for its simplicity RADAR keeps being very popular in WLAN based indoor localization and it is usually the benchmark for the other localization systems.

SVM based fingerprinting

Fingerprinting does not assume any measurement model, and this is both good and evil at the same time. In fact on one hand a

model would bridle reality by neglecting most effects difficult to describe analytically; on the other hand the (also online) tuning of parameters within an assigned model can be a rail in which classic algorithms can work to increase the algorithms accuracy in real world applications.

In [BB05] Support Vector Machines (SVM), a powerful family of algorithms in the framework of the Statistical Learning Theory, are employed in order to set the parameters of an assigned measurement model. This latter is not a propagation model based on the EMF properties, but is rather a heuristic model, linear or not, whose parameters are set up according to training stages. It is used instead of fingerprinting to reduce the complexity of the training stage whilst preserving the accuracy.

2.5.3 Probabilistic techniques

Several positioning techniques require a probabilistic characterization of the indoor channel. While for tracking, as we will see, probabilistic approaches are very exploited, in static localization fingerprinting techniques are considered simpler and sufficiently accurate, so the statistical techniques are less used. Nonetheless, advanced Bayesian approaches for WLAN localization are presented also in [MEM+05] and in [RMT+02]. More recently, the Horus system in [YA08] builds a radio map including the empirical pdfs of the RSS collected in some test locations and, then, the online RSS likelihood is computed. Also the autocorrelation of the measurements coming from the same AP is taken into account.

A brief survey of the most relevant problems regarding the probabilistic modelization of the Wlan indoor channel follows. One of the most difficult issues to tackle is the fading effect modeling that, as said above, is responsible for the randomness of the measurements. Both slow and fast fading can be described by probabilistic models, with different degrees of complexity, whose suitability is strongly dependent of the case.

Recalling the notation of Sect. 2.3, the pdfs of interest are

- the RSS likelihood with respect to $\boldsymbol{\theta}$

$$p(\mathbf{y}|\boldsymbol{\theta}) \tag{2.6}$$

- the a-posteriori pdf of $\boldsymbol{\theta}$ given RSS

$$p(\boldsymbol{\theta}|\mathbf{y}), \tag{2.7}$$

where we denote all pdfs with $p(\cdot)$. Note that time is neglected but this does not imply that each AP provides one RSS only: the underlying assumption is that the user's position does not vary over time during the collection of measurements.

According to Sect. 1.1, maximizing the RSS likelihood over $\boldsymbol{\theta}$ leads to MLE within a non Bayesian framework, while the posterior expectation and maximization lead to MAP and MMSE estimators, respectively. It is worthy to say that in suitable hypotheses, ML and MAP criteria bring to the same estimator. In fact, given the prior pdf $p_0(\boldsymbol{\theta})$ of the user's location, we can apply the Bayes Theorem to eq. (2.7), yielding

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta})p_0(\boldsymbol{\theta})}{p(\mathbf{y})}. \tag{2.8}$$

In the maximization of eq. (2.8) $p(\mathbf{y})$ is not involved and, if the prior pdf is uniform, we can write

$$\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{y}) = \max_{\boldsymbol{\theta}} p(\mathbf{y}|\boldsymbol{\theta}).$$

Note that ML and MAP criteria are drawn from different probabilistic frameworks: throughout this thesis we are mostly interested to a Bayesian framework and thus to the MAP (or even MMSE) rule.

We now present the main models proposed for the likelihood function. A general model for the RSS vector is analogous to the one of eq. (1.14)

$$\mathbf{y} = \mathbf{g}(\boldsymbol{\theta}, \mathbf{n}), \tag{2.9}$$

where $\mathbf{g}(\cdot) = \{g_j(\cdot)\}_{j=1,\ldots,N_{AP}}$ is a vectorial function of the user's position $\boldsymbol{\theta}$ corrupted by noise, say $\mathbf{n}$[7].

This means that $\mathbf{y}$ is in turn a random vector whose characteristics, given $\boldsymbol{\theta}$, depends on the noise $\mathbf{n}$. Assuming further that $\mathbf{n}$ has N_{AP} independent components $\{n_j\}_{j=1,\ldots,N_{AP}}$ such that $y_j = g_j(\boldsymbol{\theta}, n_j)$, RSS from different APs are also independent given $\boldsymbol{\theta}$, and thus

$$p(\mathbf{y}|\boldsymbol{\theta}) = \prod_{j=1}^{N_{AP}} p(y_j|\boldsymbol{\theta}), \tag{2.10}$$

where $p(y_j|\boldsymbol{\theta})$ is given by the fading model adopted.

2.6 Dynamic localization

If the user is in a building and is localized periodically, the final result is its temporal sequence of positions, i.e. its *trajectory*. Intuition can help us in stating that the trajectory contains further information that static localization cannot exploit. In particular we refer to the strong correlation that exists between estimates close in time.

The goal of dynamic localization is the estimation of a whole trajectory rather than a sequence of single positions. It appears straightforward that a more powerful framework is necessary to exploit user's positions correlation also preserving mathematical tractability. The Bayesian dynamic estimation described in Sect. 1.3 is a natural, despite not univoque, setup for this scenario.

2.6.1 Bayesian approach

The model (1.13)-(1.14) of Sect. 1.3 is here restated for clarity

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{v}_k), \tag{2.11}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{n}_k). \tag{2.12}$$

[7]In the practice, noise is dependent of the position, even if it is often assumed position-independent in probabilistic models.

While for the measurements model the considerations of Sect. 2.4 are still valid, the process model useful in our application can be expressed as

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{v}_k \tag{2.13}$$

where F is a constant matrix of dimensions $(n_x \times n_x)$, called state transition matrix, and $\mathbf{v}_k$ is a Gaussian noise with zero mean and covariance matrix Q $(n_x \times n_x)$

$$\mathbf{v}_k \sim \mathcal{N}(0, Q)\,.$$

The state $\mathbf{x}$ includes but is not limited to the user's position. Indeed, several choices are possible about the state composition and this affects the matrices F and Q. Markov hypothesis on $\mathbf{x}$ does not allow to use variables related to different instants. Thus the only way to increase the 'memory' effect is to employ derivatives of the position at the same instant, i.e. user's velocity and acceleration.

Independently of the state variables choice, two approaches can be derived with relevant differences in the noise properties; the former is based on the sampling of a continuous-time (CT) model, the latter assumes directly a discrete time (DT) model [BSLK01, Ch. 6].

CT white noise acceleration

Let the scalar $\theta(t)$ be the user's position in one dimension at the continuous time instant t. If we imposed

$$\ddot{\theta} = 0,$$

with reference to the second derivative w.r.t. time, the user's velocity would be constant and its position linearly related to time. We impose instead the condition

$$\ddot{\theta} = \tilde{v}(t), \tag{2.14}$$

where $\tilde{v}(t)$ is a continuous time zero mean white noise with

$$\begin{aligned} E[\tilde{v}(t)] &= 0 \\ E[\tilde{v}(t)\tilde{v}(s)] &= \sigma_v^2 \delta(t-s), \end{aligned}$$

and σ_v^2 is the power spectral density of the white noise. The corresponding state vector is composed of the user's position and velocity

$$\mathbf{x} = \left[\theta, \dot{\theta}\right]^T$$

and the CT movement model is

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + D\tilde{v}(t), \tag{2.15}$$

with

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Sampling the system in eq. (2.15) with a constant sampling time $\tau > 0$ yields

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{v}_k, \tag{2.16}$$

where

$$F = e^{A\tau} = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix},$$

and

$$\mathbf{v}(k) = \int e^{A(\tau - s)} D\tilde{v}_{k\tau+s} \text{ d } s.$$

The covariance matrix of $\mathbf{v}_k$ is thus

$$Q = E\left[\mathbf{v}_k \mathbf{v}_k^T\right] = \begin{bmatrix} \frac{1}{3}\tau^3 & \frac{1}{2}\tau^2 \\ \frac{1}{2}\tau^2 & \tau \end{bmatrix} \sigma_v^2.$$

The velocity changes over a sampling period τ are of the order of

$$\sqrt{Q_{22}} = \sigma_v \sqrt{\tau},$$

that can help again to set σ_v. In particular a Nearly Constant Velocity Model (NCVM) is obtained when expected changes in velocity are much smaller than the actual velocity. The cases of $\boldsymbol{\theta} \in \mathcal{R}^2$ and $\boldsymbol{\theta} \in \mathcal{R}^3$ are simple extensions: the elements in the matrices F and Q are multiplied each by an identity matrix of order 2 and 3 respectively. As a final remark, we say that the velocity is a Wiener process, being the integral of a white process.

DT white noise acceleration

Another common kinematic model is directly defined in discrete time

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \Gamma\mathbf{v}_k, \tag{2.17}$$

in which the discrete-time process noise $\mathbf{v}_k$ is a scalar-valued zero-mean white sequence

$$E[v(k)v(j)] = \sigma_v^2\delta_{kj}, \tag{2.18}$$

δ_{kj} is the Kronecker function and Γ a suitable gain matrix.

In this case we have for the one dimensional case and scalar v_k (the extension is the same as before)

$$F = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, \qquad \Gamma = \begin{bmatrix} \frac{1}{2}\tau^2 \\ \tau \end{bmatrix}.$$

In other words, this is a piecewise constant white acceleration model, because the noise is considered constant into the sampling period. The covariance of the process noise multiplied by the gain, Γv_k, is

$$Q = E[\Gamma v_k v_k \Gamma^T] = \Gamma\sigma_v^2\Gamma^T = \begin{bmatrix} \frac{1}{4}\tau^4 & \frac{1}{2}\tau^3 \\ \frac{1}{2}\tau^3 & \tau^2 \end{bmatrix}\sigma_v^2.$$

From a physical point of view v_k and σ_v are accelerations. A NCVM is obtained if the following condition is true: the changes in the velocity over a sampling interval, which are of the order of $\sigma_v\tau$, are small compared to the actual velocity.

CT Wiener process acceleration

We can include also acceleration into the state vector

$$\mathbf{x} = \left[\theta, \dot{\theta}, \ddot{\theta}\right]^T$$

and assume that it is a Wiener process, by imposing

$$\dddot{\theta} = \tilde{v}(t).$$

Again, the movement model is

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + D\tilde{v}(t), \tag{2.19}$$

with

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} D = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Sampling the system in eq. (2.19) with a constant sampling time $\tau > 0$ yields

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{v}_k, \tag{2.20}$$

where

$$F = \begin{bmatrix} 1 & \tau & \tau^2/2 \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix}.$$

The covariance matrix of $\mathbf{v}_k$ results in

$$Q = E\left[\mathbf{v}_k \mathbf{v}_k^T\right] = \begin{bmatrix} \frac{1}{20}\tau^5 & \frac{1}{8}\tau^4 & \frac{1}{6}\tau^3 \\ \frac{1}{8}\tau^4 & \frac{1}{3}\tau^3 & \frac{1}{2}\tau^2 \\ \frac{1}{6}\tau^3 & \frac{1}{2}\tau^2 & \tau \end{bmatrix} \sigma_v^2.$$

The changes in the acceleration over a sampling period τ are of the order of

$$\sqrt{Q_{33}} = \sigma_v \sqrt{\tau},$$

that is a guideline for the choice of σ_v. In particular a Nearly Constant Acceleration Model (NCAM) is obtained when expected changes in acceleration are much smaller than the actual acceleration.

DT Wiener noise acceleration

Similarly to DT white noise velocity model, but with the state composed by user's position, velocity and acceleration, the matri-

ces of model (2.17) are

$$F = \begin{bmatrix} 1 & \tau & \tau^2/2 \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix}, \qquad \Gamma = \begin{bmatrix} \frac{1}{2}\tau^2 \\ \tau \\ 1 \end{bmatrix}.$$

The covariance matrix is

$$Q = E[\Gamma v_k v_k \Gamma^T] = \Gamma \sigma_v^2 \Gamma^T = \begin{bmatrix} \frac{1}{4}\tau^4 & \frac{1}{2}\tau^3 \\ \frac{1}{2}\tau^3 & \tau^2 \end{bmatrix} \sigma_v^2.$$

For this model, σ_v should be of the order of the magnitude of the maximum acceleration increment over a sampling period.

2.6.2 KF based approaches

All movement models presented so far are linear with Gaussian noise. On the contrary, the measurement model is not linear at all and, if Rice likelihoods are employed, neither Gaussian. This is the reason why standard Kalman Filter cannot be applied and its suboptimal variants are needed.

The EKF algorithm of Sect. 1.3.4 deals with non linearity in a simple way, even if the performance must be then evaluated. Let us consider first the Lognormal model, viz., in dBm,

$$\mathbf{y}_k = g(\boldsymbol{\theta}) + \mathbf{n}_k \sim \mathcal{N}\left(g(\boldsymbol{\theta}), R\right).$$

To apply the Alg. 2 we have to linearize $\mathbf{g}(\cdot)$ at every time step, according to

$$G = \left.\frac{\partial \mathbf{g}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{k|k-1}}.$$

This can be done if $g(\boldsymbol{\theta})$ is derivable, as it is the case of the model of eq. (2.3). The WAF model of eq. (2.4), on the other hand, is singular since the number of walls is a non continuous function of the position. UKF can be also applicated to indoor localization straightforwardly as shown in [PW08].

In the case of RSS likelihoods modeled according to a Rice r.v., the non Gaussianity is a further source of approximation.

2.6.3 Particle Filter based approaches

In principle, Particle Filters (PFs) are keen on dealing both with non linearity and, above all, with non Gaussianity. Since resampling is necessary to avoid particle degeneracy, SIR algorithm (and its variants) is going to be used.

The state is $\mathbf{x}_k$ whose samples are drawn from a suitable importance density

$$q(\mathbf{x}_k|\mathbf{x}^i_{0:k-1}, \mathbf{y}_{1:k})$$

and, subsequently, are properly weighted. A popular choice is to use a 'predictive' function as importance density and then the RSS likelihood as weight. In view of this we factorize the joint pdf of the state sequence

$$\begin{aligned} p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) &= p(\mathbf{y}_k|\mathbf{x}_{0:k}, \mathbf{y}_{1:k-1})p(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k-1})p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}) \\ &= p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}). \end{aligned} \tag{2.21}$$

By defining the importance density as the prediction pdf

$$q(\mathbf{x}_k|\mathbf{x}^i_{0:k-1}, \mathbf{y}_{1:k}) = p(\mathbf{x}_k|\mathbf{x}^i_{k-1}),$$

the weight results in

$$w^i_k = p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}).$$

Applying recursive arguments to $p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})$, a useful formula for the weight is obtained, viz.

$$w^i_k = p(\mathbf{y}_k|\mathbf{x}^i_k)w^i_{k-1}.$$

A different choice is required by SIR; in this case the new RSS measurement is included in the importance density

$$q(\mathbf{x}_k|\mathbf{x}^i_{0:k-1}, \mathbf{y}_{1:k}) = p(\mathbf{x}_k|\mathbf{x}^i_{k-1}, \mathbf{y}_k),$$

which deals with the factorization

$$p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) = p(\mathbf{x}_k|\mathbf{x}^i_{k-1}, \mathbf{y}_k)p(\mathbf{y}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k-1})p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}), \tag{2.22}$$

and thus the recursively computed weight is

$$w_k^i = p(\mathbf{y}_k|\mathbf{x}_{0:k-1}^i, \mathbf{y}_{1:k-1}) w_{k-1}^i.$$

With this formulation, state sampling is expected to be more precise, but the computation of both importance density and weights are not straightforward anymore. In fact the importance density can be expressed, thanks to Bayes Theorem, as

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{y}_k) \propto p(\mathbf{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{x}_{k-1}^i),$$

from which $\mathbf{x}_k$ can be drawn only in some cases: if Gaussian likelihoods in dB are employed the product of two Multivariate Gaussians is again Gaussian and thus sampling can be done; otherwise it could encompass serious difficulties.

As for the weight, marginalization over $\mathbf{x}_k$ is necessary

$$p(\mathbf{y}_k|\mathbf{x}_{0:k-1}^i, \mathbf{y}_{1:k-1}) = \int p(\mathbf{y}_k|\mathbf{x}_k^i) p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)\, \mathrm{d}\mathbf{x}_k,$$

whose solution is available at least in a numerical way. ASIR can represent a way to perform something similar by avoiding these problems, but it employs a more complex procedure.

2.7 Survey of the technologies of interest

The contribution of this thesis is relative to WLAN indoor localization and the experimental part has been fully developed by means of IEEE 802.11 (WiFi) compliant devices. In this section, we provide a brief survey of the WiFi standard with its main technical features. Also Inertial Measurement Units (IMUs) are shortly presented, since the argument proposed in Ch. 5 is based on a data fusion framework between WiFi RSS measurements and IMU's derived step measurements.

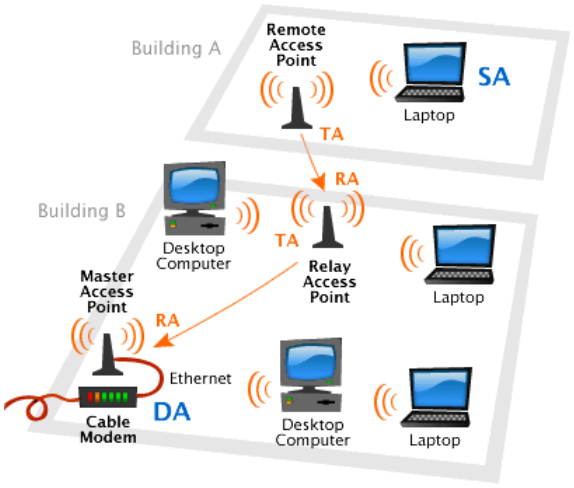


Figure 2.4 Typical WiFi infrastructure

2.7.1 IEEE 802.11 standard

IEEE 802.11 is today the most used WLAN technology. In the infrastructure topology the AP is the base station that forwards data to the user devices or to a connected network (see fig. 2.4).

There are many versions of the standard, indicated by the letters a,b,g and n respectively, in which the differences are mainly relative to their bit rates and other features. The most used versions are 802.11b and 802.11g, since many receivers are compatible with both. They operate with a Direct Sequence Spread Spectrum (DSSS) based modulation in the license free ISM band ($\approx$ 2.45 GHz). The maximum allowed bit rates are 11 and 54 Mbps for the 'b' and 'g' versions respectively; the last standard '802.11n' can operate also at 5 GHz with higher data rates and is equipped with MIMO antennas.

The maximum transmission power is set to 100 mW, allowing to cover up to one hundred meters, depending on the environment. The frequency band is split in 11 channels, but only 3 of them are fully not overlapped; it is thus important in the design of a WiFi network with at least 4 APs to assign the same channel to the farthest APs.

Beacon frames are periodically emitted by the APs for network tasks, such as the synchronization of the receivers' clocks to the AP's clock. Since the resolution of this latter in off-the-shelf APs ($1\,\mu s$) is too coarse to yield an accurate distance estimation, time of arrivals (TOA) based techniques are usually avoided, unless employing additive hardware, with a raising of the costs. For example, Ciurana et al. propose an AP prototype to perform TOA based positioning and tracking in [ICB+06] and in [CBC06] respectively[8]. Also, a slightly different system is proposed in [YOT+05], where time differential of arrival (TDOA) techniques are used. As for angle of arrivals (AOA) based algorithms, MIMO antennas are required, so that only the version 'n' of the standard can be employed without additive hardware.

Anyway, the RSS of the beacon frame emitted by the access point is measured by the receiver and made available to high level applications. Therefore, such information can be exploited by a localization system. Note that, even if the standard indicates a 8 bit (256 levels) quantization for the RSSI measurement, it does not define its resolution nor accuracy, that are typically unavailable to the user. Common resolutions are, however, -100 dBm to 0, with 1 dBm sized steps.

In [VSK+10] an anechoic chamber is used to characterize the RSSI measurement features for some popular products such as IPod and IPad. In detail, some practical aspects should be taken into account:

- even if antennas are usually specified as omni-directional, some meaningful imperfections often occur, yielding a bias in the measurements which depends on the direction of arrival;

- sometimes, the rate at which the device provides RSSI measurements can be higher than the sampling rate and in this case the same measurement can be provided several times;

[8] In the practice they add a network receiver with better time resolution to APs.

Figure 2.5 A foot mounted IMU. Taken from *http://www.wearable.ethz.ch*

- even if different RSSI are given, they could be statistical correlated due to hardware problems, above all if the sampling rate is too high.

Again, such information is usually not reported anywhere.

2.7.2 Inertial Movement Units

An inertial measurement unit (IMU) worn by a pedestrian who is walking in a building is of great help in localization. A comfortable and popular solution is to attach the IMU to a user's shoe, as in Fig. 2.5. Such sensor essentially provides a sequence of measurements (at a relatively high rate $\sim 100Hz$) of the acceleration (in 2 or 3 dimensions) and its utility in localization systems has been widely proved. In fact, a single integration filtering on those data provides velocity, and a double one makes it possible to infer the user's position. Given a kinematic (and usually probabilistic) model for the pedestrian movement and another one for the measurements, the algorithm often employed in order to obtain location inference is EKF. The result is a sequence of step measurements at the rate of about 1 step per second.

Although this method promises a great accuracy, some drawbacks should be tackled. Unluckily, the actual measurements are

always affected by a drift that, when integrated twice, makes the position estimate to diverge very rapidly with time. Therefore, in the practice an 'open loop' procedure can be adopted only for few seconds and more complex 'stable' algorithms must be expoited. More details about this will be given in the Ch. 5.

2.8 Concluding remarks

In this chapter we gave a survey of the main approaches adopted in indoor localization. Some choices were necessary for space reasons and we have neglected the techniques less addressed in the positioning literature. We have not considered aspects like the estimation of side parameters and maps, that can have a dramatic impact on positioning techniques and performances. This will be the next chapters' focus.

Chapter 3

Adaptive Techniques for Indoor Positioning

After a review of the main mathematical, algorithmic and technical aspects involved in the thesis, we now start describing the original contribution we aim to provide. This chapter proposes a basilar analysis of the environment profiling, with most emphasis drawn to the theoretical issues which make this task difficult, although necessary at the same time. After stating the models for the pedestrian movement and the power propagation that will be used, with minor variants, throughout the whole thesis, we will identify the environment uncertainties that usually characterize the indoor positioning, and we develop various solutions by means of Bayesian techniques. For this reason, some problems concerning the application of the algorithms presented in the previous chapters must be fixed with suitable changes. For example we deal with the presence of barriers in the environment to formulate a more realistic model of the true user motion.

After that, we propose a simulative study in a simplified framework to figure out the main criticisms in this kind of analysis and, finally, an example of experimental case study characterized by very advers environmental conditions.

It should be clear that in this chapter we are not supposed to provide operative algorithms, nor exhaustive analyses of the environmental profiling. Just, we want to approach the problem and explore the possible solutions offered by the Bayesian Probability Theory.

Briefly, we refine the user's movement model to account for the obstacles and propose a study about its parameters in Sect. 3.1 and 3.2 respectively. In Sect. 3.3 the Bayesian setup is applied to formalize the environmental profiling and then we present the corresponding solutions in Sect. 3.4; in Sect. 3.5 we introduce the variance noise filter. Simulative experiments corroborating the effectiveness of the proposed approaches are carried out in Sects. 3.6-3.7 and the corresponding theoretical bound is described in Sect. 3.8. Finally, experimental results and some concluding remarks are proposed in Sect. 3.9 and Sect. 3.10 respectively.

3.1 Walking pedestrian: a tractable model

As it was assumed in Ch. 2 the pedestrian is modeled like a material point whose two dimensional position at the time instant $k \geq 0$ is encoded in the vector $\boldsymbol{\theta}_k \in \mathcal{A} \subseteq \mathcal{R}^2$.

The user's motion is modeled through the NCVM introduced in Sect. 2.6.1; for clarity, we recall that it prospects a state vector composed of the user's position and velocity (both in 2D)

$$\mathbf{x}_k = \left[\boldsymbol{\theta}_k^T, \dot{\boldsymbol{\theta}}_k^T\right]^T,$$

where the superscript T means transpose, and a dynamic model sampled at the time instants $k\tau$, $k \geq 0$

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{v}_k.$$

The matrix F is defined like

$$F = \mathbf{I}_2 \otimes \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix},$$

while the zero mean white process $\mathbf{v}_k$, hence supposed Gaussian, has a covariance matrix Q

$$Q = E\left[\mathbf{v}_k \mathbf{v}_k'\right] = \sigma_v^2 \mathbf{I}_2 \otimes \begin{pmatrix} \frac{1}{3}\tau^3 & \frac{1}{2}\tau^2 \\ \frac{1}{2}\tau^2 & \tau \end{pmatrix}.$$

Above we have introduced for clarity the identity matrix $\mathbf{I}_2$ of order 2 and the Kronecker product $\otimes$.

3.1.1 How to model physical barriers

Barriers in real environments are walls, relevant pieces of furniture or whatelse is capable of limiting the user's movement. The pedestrian's visual system, that is liable for the barrier avoidance, cannot be sensed by an external system and thus suitable models for such barriers should be introduced. In other words, this is

equivalent to a set of constraints on the estimation problem that, beside causing bias, refine the overall performances.

A way to address this problem is to step up to a more general movement equation

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{v}_k.$$

This approach was adopted, among the others, in [BU11] to include the routes information into the maritime tracking algorithms. Although several simple models could be proposed for the barriers, we want to add a further constraint relative to the mathematical properties of the model; in detail, we require that both the function $\mathbf{f}(\cdot)$ and its gradient be continuous (the existence of the gradient itself is assumed as well). Furthermore, when the user is far from the obstacle it holds

$$\mathbf{f}(\mathbf{x}_k) \cong F\mathbf{x}_k$$

like in the unbounded case. The resulting 'soft' model is compliant with the mathematical assumptions typically required by the tracking algorithms and by the Cramer Rao based analysis.

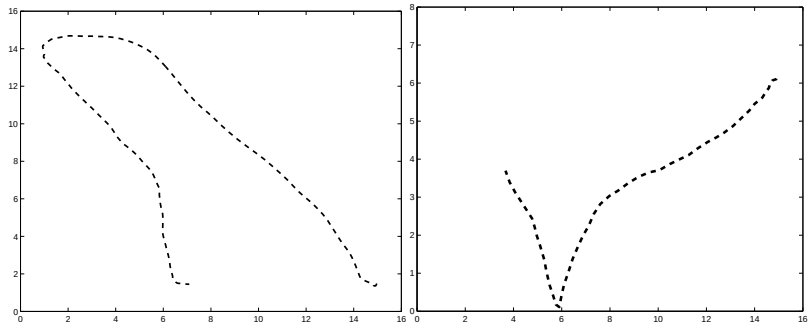
Our proposal aims at reducing and eventually inverting the user's velocity according to a sigmoid function when the user approaches the barrier. Consider for simplicity the one dimensional case, with the user's position limited to be $\theta_k < \theta_B$. The velocity $\dot{\theta}_k$ is influenced by the distance $\theta_B - \theta_k$ from the obstacle through a sigmoid function

$$\dot{\theta}_{k+1} = (1 - 2s_\zeta(\theta_B - \theta_k))\,\dot{\theta}_k + \mathbf{v}_{\dot{\theta},k}, \tag{3.1}$$

where

$$s_\zeta(t) = \frac{1}{1 + e^{-t/\zeta}}$$

is a sigmoid function and $\mathbf{v}_{\dot{\theta},k}$ is the component of $\mathbf{v}_k$ relative to the velocity. This model ensures that when the user is bound to impact on the obstacle, its velocity is first reduced and then inverted, so that it moves away from it. We say that this constraint is soft, because θ_k can be greater than θ_B for some k . This trick, however, for the low velocities of our applications has a very little impact on



(a) Soft constraint (b) Reflection

Figure 3.1 Typical trajectories generated by (a) the soft bounded model and the (b) reflection model.

the degree of approximation of the model, whilst making it more easy to handle. A typical trajectory generated by this model is shown in Fig. 3.1.a.

All obstacles can be modeled like similar barriers. Note that the choice of ζ should be case sensitive: for walls $\zeta \approx 1$ looks proper, even if a study for the optimal setting of the sigmoid paramaters (i.e. ζ and the sigmoid center since here set to θ_B) would be of interest.

As a special case we mention that $\zeta = 0$ resumes the reflective model, whose typical example is depicted in Fig. 3.1.b: the velocity is simply inverted in the case of barrier crossings, i.e. the matrix F is switched between two choices in which only the velocity sign is changed. Further details are omitted for this model is not used anymore.

3.1.2 Measurement Model

The fading effect has a high impact on the signal propagation, especially indoors. Following the analysis in [Par00], the signal amplitude r can be expressed like

$$r = r_S \cdot r_F,$$

where r_S and r_F account for slow and fast fading respectively. In Appendix A we propose a deep study of the main probabilistic models for slow and fast fading, aimed at providing a common analysis framework. Concerning this, slow fading is usually well characterized by a Lognormal model

$$r_S \sim p_L(r) = \frac{1}{\sqrt{2\pi}\sigma r} \exp\left(-\frac{(\log r - \mu)^2}{2\sigma^2}\right), \tag{3.2}$$

where μ and σ are dependent of the distance between the user and the AP; a suitable model for fast fading is instead given by the Rice (or Nagakami-n) pdf

$$\begin{aligned} r_F \sim p_R(r) =& \frac{2(1+K_f)r}{\Omega} \exp\left(-K_f - \frac{(K_f+1)r^2}{\Omega}\right) \\ & \cdot I_0\left(2r\sqrt{\frac{K_f(K_f+1)}{\Omega}}\right), \; r \geq 0 \end{aligned} \tag{3.3}$$

whose parameters are $K_f \geq 0$ and $\Omega = E[r^2]$ and $I_0(\cdot)$ is the zeroth order modified Bessel function of the first kind. The Rice pdf is quoted a suitable model for fast fading in case of LOS propagation conditions; if the LOS is not present, the Rayleigh pdf is best suited, that is a Rice pdf with $K_f = 0$ [Par00]. The K_f factor has a strong physical meaning, since it represents the ratio between the 'main' signal component and the 'floor' (due to multipath) powers.

Since the RSS measurements are usually available in dBm, we are particularly interested in the power characterization in logarithms, since in far field conditions the power results in the square of the signal amplitude; this allows us to split the slow fading effect from the fast fading one:

$$y = 20\log r = 20\log r_S + 20\log r_F. \tag{3.4}$$

Concerning the slow fading term, it is easy to show that $y_S = 20\log r_S$ is distributed according to a Gaussian pdf

$$y_S \sim \mathcal{N}\left(\mu_{dB}, \sigma_{dB}^2\right), \tag{3.5}$$

with mean $\mathrm{E}[y_S] = \mu_{dB} = \kappa\mu$ and variance $\mathrm{VAR}[y_S] = \kappa^2\sigma^2$, where $\kappa = 20/\log(10)$.

The fast fading term $y_F = 20\log r_F$ is expressed interestingly by means of the function

$$\begin{aligned} y_F \sim p_R(r_F) = & \frac{2(1+K_f))}{\kappa} \exp\left(\frac{2(y-\Omega_{dB})}{\kappa} - K_f \right. \\ & \left. -(K_f+1)\exp\left(\frac{2(y-\Omega_{dB})}{\kappa}\right)\right) \\ & \cdot I_0\left(2\sqrt{K_f(K_f+1)}\exp\left(\frac{y-\Omega_{dB}}{\kappa}\right)\right), \end{aligned} \tag{3.6}$$

where $\Omega_{dB} = \kappa/2\log(\Omega)$ is a shift parameter and thus affects only the expectation, while the variance is almost only dependent of K_f:

$$\begin{aligned} \mathrm{E}[20\log r_F] &= \Omega_{dB} - e(K_f); \\ \mathrm{VAR}[20\log r_F] &= v(K_f). \end{aligned}$$

The functions $e(K_f)$ and $v(K_f)$ have been computed numerically for $K_f \in [0, 20]$, because this range covers the typical values of K_f (see also [GME99]), and they are shown in Fig. 3.2; suitable analytical approximations can be performed by means of either polynomial fits or the Nagakami-m pdf [KA07].

Finally, it results that

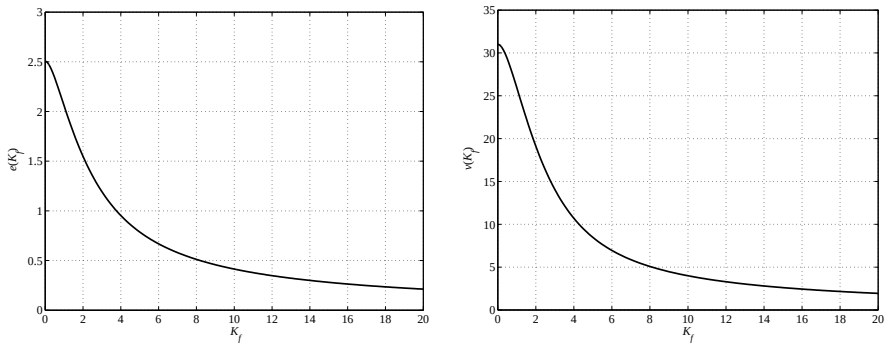
$$\mathrm{E}[y] = \mathrm{E}[y_S] + \mathrm{E}[y_F], \tag{3.7}$$

and, assuming independence between slow and fast fading effects,

$$\mathrm{VAR}[y] = \mathrm{VAR}[y_S] + \mathrm{VAR}[y_F]. \tag{3.8}$$

The mean power in dB is tuned according to a deterministic propagation model; in our case we use the path loss model [BP00b]

$$\mathrm{E}[y] = P_{d_j} = h_j - 10\alpha_j \log\left(\frac{d_j}{d_0}\right), \tag{3.9}$$



(a) (b)

Figure 3.2 Plot of the functions $e(K_f)$ (a) and $v(K_f)$ (b) vs. K_f.

where d_j the Euclidean distance between user and the *j-th* AP, h_j and α_j are the reference power and the propagation exponent respectively. Since now we will denote briefly

$$\mathbf{h} = \{h_j\}_{j=1,\ldots,N_{AP}}\,, \qquad \boldsymbol{\alpha} = \{\alpha_j\}_{j=1,\ldots,N_{AP}}\,,$$

that will be subject to further analysis later.

3.2 Effect of the parameters in the movement process

The parameters of the movement model have a remarkable impact on the performances of the algorithms. As an example, we here investigate the behavior of EKF when variations for different values of σ_v and τ within the NCVM assumed in this chapter.

In each subplot of Fig. 3.3 only a parameter of the movement model varies, whereas the other one is kept fixed (see the true values in the corresponding caption). The results confirm what we said above: an increase of the randomness in the target motion due to a higher noise power reduces the overall performances; in this case a knee is present between 1 and 1.5 $m/s^{1.5}$. This gives a numerical evaluation for the condition on σ_v required by the

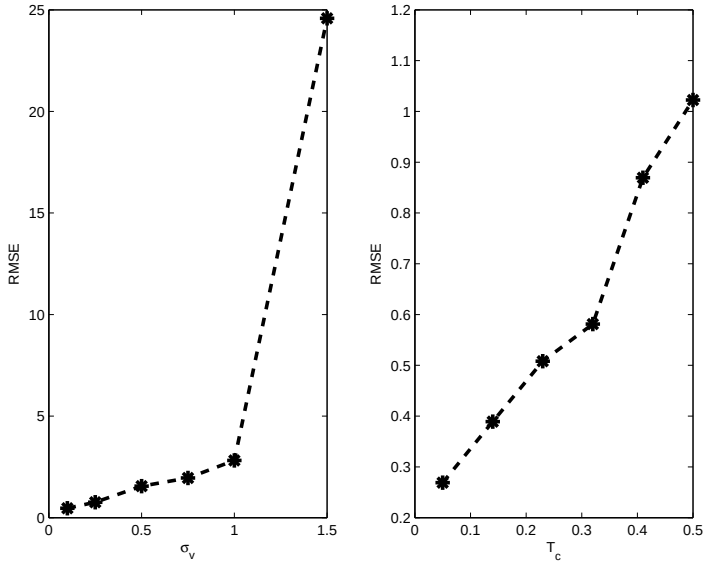


Figure 3.3 RMSE plotted according to variations of the movement model: in the left plot σ_v varies in the range $[0, 1.5]$ $m/s^{1.5}$, with a constant $\tau = 0.2$ s; on the right τ varies in the range $[0, 0.5]$ s, $\sigma_v = 0.1$ $m/s^{1.5}$

NCVM:

$$\sigma_v \ll \frac{1}{\sqrt{\tau}} \approx 2.23.$$

Furthermore, from the analysis of the right side plot of Fig. 3.3, we can state that a larger sampling period τ acts in the same way. These two effects are both visible in the expression of the covariance matrix Q, whose components are all proportional to both σ_v and τ.

3.3 Introducing uncertainty in the RSS propagation model

A further step within RSS based indoor localization concerns adaptivity to environmental uncertainties or unknown parameters. In the propagation model of eq. (3.9) there are at least two parameters that can be of interest to us: the reference power **h** and

the propagation exponent $\boldsymbol{\alpha}$; moreover also the variance of the measurement noise is usually uncertain.

In this section we provide a preliminary study about the on line estimation of the above parameters together with the user's trajectory, by employing only the RSS measurements. First we consider each parameter alone, and then we address the issues arising from the multi-parameter estimation.

Before stating our proposals, we briefly review the existing techniques.

3.3.1 Existing techniques

Yet from the eldest studies about indoor localization the need for adaptive algorithms was stressed. The first, quite rough, attempt to profile the environment was an enhancement of RADAR, presented in [BP00a]; it consists in building up several RMs in different conditions (at various day and night times or different activities) and in choosing the best one on line. This last decision is taken using each AP like a probe for the others.

Other RADAR based approaches explore similar solutions (e.g. in [YYN08]): for all of them the main drawback lays in the fact that the training phase must be iterated, and thus the complexity gets worse and worse. The use of probes and suitable interpolation techniques can boost the situation but additive hardware is still necessary.

Tracking algorithms open new doors to researchers who want to account for the environmental fluctuations. Among the batch algorithms, those based on the Expectation-Maximization principle are used to provide ML estimates of the parameters [MS00]. These methods are typically not keen to follow fluctuations but only to estimate static unknown parameters and it is overcome by resorting to some shortcomings, like e.g. splitting data into windows; in this case, of course, optimality is no more pursued.

If an online 'dual' estimation is needed, instead, other algorithms must be addressed. In a recent work some authors propose

an on-line calibration of the RSS measurements SNR [CWYS09], used to correct data before deploying EKF with a logarithmic propagation model. For this aim, a number of tags are deplyed in known positions along with the APs.

3.3.2 Bayesian approach

Let

$$\{\mathbf{p}_k\}_{k=0,1,2,\ldots}, \quad \mathbf{p}_k \in \mathcal{R}^{N_{par}}$$

be a (vectorial) parameter of dimension N_{par}, that states a Non-Linear Gaussian (NLG) system:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + \mathbf{v}_k \tag{3.10}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{p}_k) + \mathbf{n}_k. \tag{3.11}$$

The parameter can be static, i.e. $\mathbf{p}_k = \mathbf{p}, \forall k$, or can admit fluctuations; in either cases, we can assign a fictitious probabilistic dynamic model

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{r}_k \tag{3.12}$$

in which $\mathbf{r}_k$ is assumed, for convenience, to be a Gaussian white noise with zero mean, suitable covariance matrix R_k^r and arbitrary, but known, prior $\mu_p(\mathbf{p})$.

In our localization problem we have two aims: to determine i) a (causal) estimate $\hat{\boldsymbol{\theta}}_{0:k}$ of the user's trajectory $\boldsymbol{\theta}_{0:k}$ that minimizes the root mean square error (RMSE)

$$\sum_{t=0}^{k} \sqrt{E\left[||\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t||^2\right]}, \tag{3.13}$$

and ii) a final parameter pdf that can be used afterwards by other users as a refined prior pdf.

We explore now three approaches: in the first one we define an extended model by stacking the user's state and the parameters into a new state vector and we perform the MSE optimization in terms of this new state; the second one, based on the Rao Blackwellized Particle Filters (RBPFs), focuses on the $\boldsymbol{\theta}_{0:k}$ optimal

estimation using the parameters as side information, and providing at the end their distribution; the last approach approximates the RBPF to deal with computational issues.

3.4 Proposed solutions

3.4.1 Joint filtering

If we are interested in the optimal (causal) estimate of the parameters, the Bayesian score function to adopt is the joint posterior pdf of $\mathbf{x}_{0:k}$ and $\mathbf{p}_{0:k}$

$$p(\mathbf{x}_{0:k}, \mathbf{p}_{0:k}|\mathbf{y}_{1:k}). \tag{3.14}$$

The most straightforward way to undertake this problem is to jointly infer $\mathbf{x}_k$ and $\mathbf{p}_k$ by stacking them into a new extended state $\mathbf{z}_k$

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{p}_k \end{bmatrix}.$$

The dynamic model for $\mathbf{z}_k$ is thus expressed by

$$\mathbf{z}_{k+1} = \tilde{f}(\mathbf{z}_k) + \tilde{\mathbf{v}}_k, \tag{3.15}$$

$$\mathbf{y}_k = \tilde{\mathbf{g}}(\mathbf{z}_k) + \mathbf{n}_k. \tag{3.16}$$

When the user is far from obstacles, the linear approximation $\tilde{f}(\mathbf{z}_k) \approx \tilde{F}\mathbf{z}_k$ holds, where

$$\tilde{F} = \begin{bmatrix} F & 0 \\ 0 & I_{N_{par}} \end{bmatrix}$$

is a new transition matrix of dimensions $(N_x \cdot N_{par}) \times (N_x \cdot N_{par})$, whose blocks have consistent dimensions; otherwise

$$\tilde{f}(\mathbf{z}_k) = [f^T(\mathbf{x}_k), \mathbf{p}_{k-1}^T]^T.$$

Moreover

$$\tilde{\mathbf{g}}(\mathbf{z}_k) = \mathbf{g}(\mathbf{x}_k, \mathbf{p}_k),$$

and

$$\tilde{\mathbf{v}}_k = \begin{bmatrix} \mathbf{v}_k \\ \mathbf{r}_k \end{bmatrix}$$

is again a Gaussian random vector with zero mean and covariance matrix, given by $\tilde{Q}$

$$\tilde{Q} = \begin{bmatrix} Q & 0 \\ 0 & R_k^r \end{bmatrix}.$$

The choice for R_k^r is not unique and can meet several needs. Usually a better choice consists in decreasing the noise power over time, rather than employing a constant matrix: in fact a greater uncertainty is needed for the initial setting of the parameters; subsequently the adaptability can be reduced to deal with rarer variations.

Algorithms

Standard tracking algorithms for Non Linear Non Gaussian (NLNG) systems can be employed [Hay01,AMGC02]; altough not very keen to deal with NLNG systems, Kalman Filter based approaches can be taken in consideration for their simplicity. In [PAR11] the EKF has been tested in his joint version, and gave mediocre results. The PF based techniques, instead, are preliminary shown to be much more effective and in this chapter we focus on them and, more in detail, with SIR algorithm, denoted in this framework by the name Joint SIR (JSIR), that is here briefly summarized.

The i-th particle at instant k contains a hypothesis $\{\mathbf{x}_{0:k}^i, \mathbf{p}_{0:k}^i\}$; the initial values ($k = 0$) are simply drawn from the priors

$$\begin{aligned} \mathbf{x}_0^i &\sim \mu_x(\mathbf{x}_0), \\ \mathbf{p}_0^i &\sim \mu_p(\mathbf{p}_0); \end{aligned} \tag{3.17}$$

then the JSIR is based on the following factorization of eq. (3.14):

$$\begin{aligned} p(\mathbf{x}_{0:k}, \mathbf{p}_{0:k}|\mathbf{y}_{1:k}) \propto & p(\mathbf{x}_k|\mathbf{x}_{k-1}) \cdot p(\mathbf{p}_k|\mathbf{p}_{k-1}) \\ & \cdot p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{p}_k) \cdot p(\mathbf{x}_{0:k-1}, \mathbf{p}_{0:k-1}|\mathbf{y}_{1:k-1}). \end{aligned} \tag{3.18}$$

In the above, the importance density that is used to sample the particles is

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) \cdot p(\mathbf{p}_k|\mathbf{p}_{k-1})$$

while the new RSS likelihood updates the weights; in summary:

- for the $i-th$ particle draw independently

$$\mathbf{x}_k^i \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^i),$$
$$\mathbf{p}_k^i \sim p(\mathbf{p}_k|\mathbf{p}_{k-1}^i);$$

- update the weight

$$w_k^i = w_{k-1}^i \cdot p(\mathbf{y}_k|\mathbf{x}_k^i, \mathbf{p}_k^i); \tag{3.19}$$

- normalize the weights such as $\sum_i w_k^i = 1$;
- resample if need be.

Resampling can be made accordingly to the method proposed in [AMGC02], based on a heuristic measure of the particles depletion and is aimed at attaining a suitable diversity.

3.4.2 Rao Blackwellized Particle Filters

Joint filtering is affected by an increase in the state space cardinality as new parameters are estimated, for example due to the presence of more APs. In the PF case this is a limitation, since the number of particles (and thus the complexity) necessary to ensure the convergence to optimal values is strongly related to it. A solution can be represented by the Rao Blackwellized Particle Filters (RBPF), presented in a general form in [RAG04], in which the state space is split into two subspaces: one is explored through Monte Carlo Methods (as in classic PFs) and the other one is subsequently explored in a deterministic way. In the case of RBPF approach the score function assumes the form

$$p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}). \tag{3.20}$$

that is slightly different from the one in eq. (3.14), as the parameters are not explicitely present, but added through a further marginalization. Therefore, the $i-th$ particle at k contains only a hypothesis for $\{\mathbf{x}^i_{0:k}\}$; the initial value ($k=0$) are again drawn from the prior

$$\mathbf{x}^i_0 \sim \mu_x(\mathbf{x}_0);$$

then the algorithm is based on the following factorization of eq. (3.20):

$$\begin{aligned} &p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) \propto \\ &p(\mathbf{y}_k|\mathbf{x}_{0:k},\mathbf{y}_{1:k-1})\cdot p(\mathbf{x}_k|\mathbf{x}_{0:k-1},\mathbf{y}_{1:k-1})\cdot p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}). \end{aligned} \tag{3.21}$$

For the $i-th$ particle at k the following steps are in order:

- the user's state is drawn like before from the predictive importance density

$$\mathbf{x}^i_k \sim p(\mathbf{x}_k|\mathbf{x}^i_{0:k-1}\mathbf{y}_{1:k-1}) = p(\mathbf{x}_k|\mathbf{x}^i_{k-1}); \tag{3.22}$$

- the particle's weight is updated with the RSS likelihood

$$w^i_k = w^i_{k-1}\cdot p(\mathbf{y}_k|\mathbf{x}^i_{0:k},\mathbf{y}_{1:k-1}); \tag{3.23}$$

- the weights are normalized such as $\sum_i w^i_k = 1$;
- the resampling is performed if needed.

The main difference with JSIR is that now the RSS likelihood has to be marginalized over the parameters:

$$\begin{aligned} &p(\mathbf{y}_k|\mathbf{x}^i_{0:k},\mathbf{y}_{1:k-1}) = \\ &\int p(\mathbf{y}_k|\mathbf{x}_k,\mathbf{p}_k)p(\mathbf{p}_k|\mathbf{x}^i_{0:k-1},\mathbf{y}_{1:k-1})d\mathbf{p}_k, \end{aligned} \tag{3.24}$$

where suitable simplifications were employed. To compute the parameters distribution we use a KF-based filter (the KF in the case of linear models, Extended or Unscented KF if not); if

$$p(\mathbf{p}_{k-1}|\mathbf{x}^i_{0:k-1},\mathbf{y}_{1:k-1}) \sim \mathcal{N}\left(m_{k-1|k-1}, S_{k-1|k-1}\right), \tag{3.25}$$

then the prediction step results in

$$p(\mathbf{p}_k|\mathbf{x}^i_{0:k-1}, \mathbf{y}_{1:k-1}) \sim \mathcal{N}\left(m_{k|k-1}, S_{k|k-1}\right), \tag{3.26}$$

and the update step provides

$$p(\mathbf{p}_k|\mathbf{x}^i_{0:k}, \mathbf{y}_{1:k}) \sim \mathcal{N}\left(m_{k|k}, S_{k|k}\right). \tag{3.27}$$

The expressions for $m_{k|k-1}$, $S_{k|k-1}$, $m_{k|k}$ and $S_{k|k}$ are computed like in [Hay01] as well as the extensions for the EKF and the Unscented KF (UKF).

The solution to the integral in eq. (3.24) depends on the exact definition of $\mathbf{p}_k$ and on the RSS likelihood model. A special case is obtained when $\mathbf{p} = \mathbf{h}$, i.e. we want only to estimate the reference Signal Strength (SS), and the RSS likelihood is assumed Lognormal (Gaussian in dB). In this case the integrand function in eq. (3.24) is the product of two Gaussian distributions and is in turn a Gaussian-like (although a normalization is required) function whose mean and variance can easily be obtained [1]

$$p(\mathbf{y}_k|\mathbf{h}_k, \mathbf{x}^i_k)p(\mathbf{h}_k|\mathbf{x}^i_{0:k-1}, \mathbf{y}_{1:k-1}) = cf(\mathbf{h}_k), \tag{3.28}$$

where $f(\cdot)$ represents Gaussian pdfs and $0 < c < 1$ is a normalization constant. Note that the first term in eq. (3.28) is Gaussian with respect to the variable $\mathbf{y}_k$: here we are exploiting the fact the in the propagation model of eq. (3.9) the variables $\mathbf{y}_k$ and $\mathbf{h}_k$ are exchangeable. For this reason, the recursive weight results in

$$w^i_k \propto w^i_{k-1} \cdot c,$$

with c being on turn the ratio

$$\frac{p(\mathbf{y}_k|\mathbf{h}_k, \mathbf{x}^i_k)p(\mathbf{h}_k|\mathbf{x}^i_{0:k-1}, \mathbf{y}_{1:k-1})}{f(\mathbf{h}_k)}$$

at an arbitrary value of the variable $\mathbf{h}_k$, e.g. its expected value. A similar consideration holds if $\mathbf{p} = \boldsymbol{\alpha}$.

[1] If $f_i(x) \sim \mathcal{N}(\mathbf{m}_i, \Sigma_i)$, $i = 1, 2$ the function $f(x) = f_1(x) \cdot f_2(x)$ is proportional to a Multivariate Gaussian function with mean and covariance matrix

$$\mathbf{m} = \left(\Sigma_1^{-1} + \Sigma_2^{-1}\right)^{-1}\left(\Sigma_1^{-1}\mathbf{m}_1 + \Sigma_2^{-1}\mathbf{m}_2\right), \quad \Sigma = \left(\Sigma_1^{-1} + \Sigma_2^{-1}\right)^{-1},$$

respectively.

3.4.3 Comments on JSIR and RBPF

JSIR is optimal in the sense that it converges to the optimal solution (w.r.t the extended state) as the number of particles approaches infinite. The RBPF has the same property only if the KF algorithm computes the exact distribution of the parameter, namely when $\mathbf{p}_k$ is expressed by means of a Linear Gaussian (LG) Model given $\mathbf{x}_k^i$, viz.[2]

$$\begin{aligned} \mathbf{p}_{k+1} &= F_p \mathbf{p}_k + \mathbf{r}_k, \\ \mathbf{y}'_k &= C_p \mathbf{p}_k + \mathbf{n}_k, \end{aligned} \tag{3.29}$$

where F_p and C_p are constant matrices, $\mathbf{r}_k$ and $\mathbf{n}_k$ are Gaussian white (and uncorrelated) noise processes and $\mathbf{y}'_k$ is the observable. This holds if either $\mathbf{h}$ or $\boldsymbol{\alpha}$ have Gaussian RSS likelihoods, but not in general.

Another point of interest in NLG models is the need of a numerical evaluation of the integral (3.24), thus representing a bottleneck from a computational point of view. For there reasons we now provide an approximation of the RBPF that is computationally suitable even in NLNG models.

3.4.4 Discrete model

A simplified representation of the parameters state space can help to handle the complexity problems. We propose to decompose the parameters space into a finite number of disjoint cells $\tilde{P}_j$, each represented by its mean value, say $\tilde{\mathbf{p}}_j$; accordingly, the random process of eq. (3.12) is approximated by

$$\mathbf{p}_k = \left\{ \tilde{\mathbf{p}}_{j,k} \right\}_{j=1,\ldots,N_J} \tag{3.30}$$

and the corresponding prior probabilities result in

$$\mu_j = \int \mu_p(\mathbf{p}_0) \, \mathrm{d} \, \tilde{P}_j.$$

[2]The parameters prior is also assumed Gaussian.

Resorting again to the score function of eq. (3.20) and to the factorization of eq. (3.21), we address the RBPF algorithm, but we compute differently the terms which involve the parameters. In particular, in such hypotheses, the parameter distribution inside the $i-th$ particle is approximated by

$$p(\mathbf{p}_k|\mathbf{x}^i_{0:k}, \mathbf{y}_{1:k}) \approx \sum_{j=1}^{N_J} \Pr\left\{\mathbf{p}_k \in \tilde{P}_j|\mathbf{x}^i_{0:k}, \mathbf{y}_{1:k}\right\} \delta\left(\mathbf{p}_k - \tilde{\mathbf{p}}_{j,k}\right).$$

The recursive computation of the above distribution can be performed by means of the Approximated Grid Based (AGB) algorithm presented in [AMGC02], that is the counterpart of the KF with a discrete state space.

3.5 Variance filter

If the unknown (or partially known) variable is the variance of the measurement noise, joint estimation is not possible anymore. The reason lies in the fact that it produces a 'low-pass' version of the state, filtering out the noise: in this case, instead, we should do the opposite. In this case a dual filter approach is necessary, in which two competing filters alternate state and variance estimations sequentially in time. The variance estimation is based on the auxiliary state space model

$$\sigma^2_{k+1} = \sigma^2_k + r_k, \tag{3.31}$$

in which σ^2_k is a state variable and r_k is a fictitious Gaussian process, and on the maximization of the metrics

$$J^{ML}(\sigma^2) = -\sum_{k=1}^{N}\left(\log(2\pi\sigma^2_{\epsilon_k}) + \frac{(\epsilon_k)^2}{\sigma^2_{\epsilon_k}}\right), \tag{3.32}$$

where $\epsilon_k = y_k - h(\hat{\mathbf{x}}_{k|k-1})$ is the prediction error and its variance $\sigma^2_{\epsilon_k}$ depends on σ^2. Note that also the predicted state $\hat{\mathbf{x}}_{k|k-1}$, and accordingly ϵ_k, depend on the noise variance and this complicates

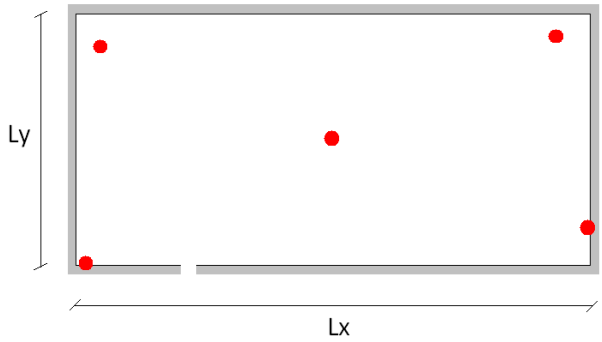


Figure 3.4 Testbed adopted in the simulations; in the figure $L_x = 40$ m and $L_y = 20$ m and the APs are in the positions denoted by red circles.

the solution. On the other hand we could neglect this latter dependence, leading to a simpler approximated solution, that is however reported to be unreliable by some authors (see [Nel00]). The filter is explained in [Hay01, Nel00]: the derivation is quite tedious and some spots are put off to the App. C, while some simulative experiments are shown in Sect. 3.7.3.

3.6 Simulation of tracking systems

Several computer experiments were designed in order to show and deeply analyse our proposal's performances. We chose to separate the fast fading from the slow fading, to avoid combined effects which would be difficult to discriminate. The testbed, represented in Fig. 3.4 is composed by a 40×20 m open area where a user walks according to the model of Sect. 3.1 with $\sigma_v = 0.1\ m/s^2$ and $\tau = 1$ s and 5 APs denoted by red circles periodically emit beacon signals. The mean received power is given by eq. (3.9), where the reference SS vector $\mathbf{h}$ is affected by uncertainty. In particular we assume the starting value of $\mathbf{h}$ being drawn from a Multivariate Gaussian prior distribution with known mean $\mathbf{h}_0$ and diagonal covariance matrix with elements σ_h^2; in some simulation settings a step-wise variation of some component of $\mathbf{h}$ is also impressed.

The starting user's state $\mathbf{x}_0$ is drawn from a Multivariate Gaussian prior distribution with diagonal covariance matrix, whose nonzero terms are set to 1 for the position variables ($\sigma^2_{\theta 1}$ and $\sigma^2_{\theta 2}$) and 0.1 for the velocity variables ($\sigma^2_{\dot{\theta} 1}$ and $\sigma^2_{\dot{\theta} 2}$). Finally, all results are averaged on series of independent data and are presented in terms of a sampling approximation of the positioning RMSE in eq. (3.13).

We drop the simulative characterization of the tracking algorithms in the case of fully known models because it is redundant, since it is provided by many authors and it is not our focus at all. Indeed in the next section we will use the 'clairvoyant' system to give a tight benchmark to the performances obtained by our adaptive algorithms. Nevertheless, we can state some considerations carried out after extensive simulative studies:

- there is no real gain, in this case, in using UKF instead of EKF unless in very high measurement variance conditions;
- the same holds for ASIR with respect to SIR, both for Log-normal and Rice fading;
- as for PFs, we found that in our simulative frameworks we have almost convergent results with a number of particles in the interval $N_p = 500 \div 1000$.

3.7 Simulative analysis of partially unknown models

We now present some simulative results, performed in the framework of Sect. 3.6; the results concerning the JEKF applied to the reference SS profiling are presented in [ABR10] and omitted here to give more space to PF-based algorithms: this choice is also confirmed by the fact that PFs are much more robust with real measurements and thus represent the standard in today localization systems.

Therefore, the PF-based algorithms of Sect 3.4 will be tested with both slow fading and fast fading. We provide a deep characterization in the case of the $\mathbf{p} = \mathbf{h}$, and further considerations about the cases $\mathbf{p} = \boldsymbol{\alpha}$ and $\mathbf{p} = [\mathbf{h}, \boldsymbol{\alpha}]$.

3.7.1 Reference SS estimation

Slow fading

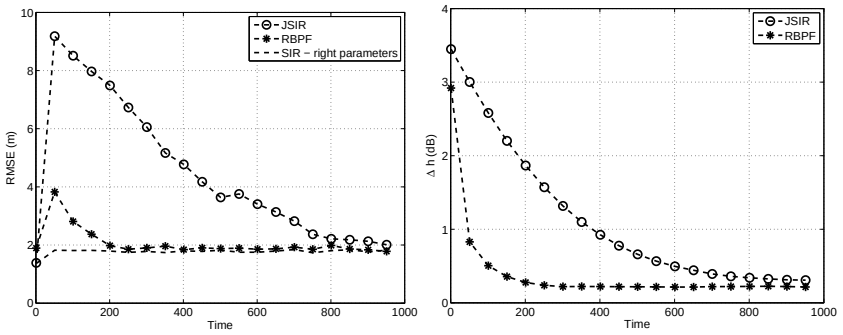
To test the slow fading effects, the measurements in dB are drawn according to the Gaussian model of eq. (3.5); beside JSIR, we test RBPF and not AGB because RBPF is not affected by computational problems in this case, as seen in Sect. 3.4.3. The first test is obtained with $\sigma_h^2 = 9$ and compares the results of JSIR and RBPF; as a reference we also draw the corresponding results obtained by the clairvoyant SIR algorithm (no parameter estimation) fed up with the true values of the reference SS. All algorithms are applied with 1000 particles and their RMSEs are plotted against time in Fig. 3.5.a. The initial RMSE value is related for all of them to the covariance matrix of the state prior, viz.

$$\sqrt{E\left[||\hat{\boldsymbol{\theta}}_0 - \boldsymbol{\theta}_t||^2\right]} = \sqrt{\sigma_{\theta 1}^2 + \sigma_{\theta 2}^2} = \sqrt{2};$$

then, both adaptive algorithms have a transient in which, by estimating the parameters, they retrieve the same performance shown by the non adaptive algorithm fed up with the true parameter values. The differences between JSIR and RBPF lay in the overshoot amplitude and in the speed of convergence: in both cases very relevant benefits are achieved by RBPF. This is a direct consequence of the algorithms adaptivity: as it is shown in Fig. 3.5.b the mean error

$$\Delta h = |h - \hat{h}|$$

on the estimated reference SS is rapidly torn down in the RBPF case to a steady state value that is slightly greater than zero, but nevertheless it does not affect the RMSE. This behavior is tightly foretold in [TMN98] in terms of the related Posterior Cramer Rao



(a) Positioning RMSE (b) Error on the parameter

Figure 3.5 Slow fading effect shown by means of computer experiments on the testbed of Fig. 3.4: JSIR and RBPF are applied with $\sigma_h^2 = 9$, $\sigma_y^2 = 5$ and number of particles in the range $N_p = 1000$; we show (a) the positioning RMSE and (b) mean estimation error $\Delta h = |h - \hat{h}|$ on the parameter.

Lower Bound (PCRLB), whose steady state values both in the presence and in absence of unknown parameters are the same; even if in our case this is not strictly possible since we adopt in the estimation a fictitious dynamic model for the parameter with nonzero noise variance, the performance is very similar. It would be of interest to optimize the algorithms w.r.t. the noise variance of the parameters but this issue goes beyond our aims.

Let us dig deeper into the algorithms evaluation. In Fig. 3.6 we show the results of our algorithms applied in the same conditions as in Fig. 3.5, but with a variable number of particles in the range $N_p = 200 \div 1000$: even 200 particles are sufficient for RBPF in order to overcome JSIR applied with as many as 1000 particles.

We also carried out an analysis of the performances related to variations of σ_h^2 and σ_y^2 in Fig. 3.7, subplots (a) and (b) respectively. We depict the results concerning RBPF which show a little variability in the positioning RMSE ; only a 100 % increase in the measurement variance σ_y^2 deteriorates the performances of 25 %. In the measurement variance case, looking at the steady state values evaluated in Table 3.12, a pejorative effect is reported due to variations of σ_y^2 (but it is a less than linear behavior).

As a final test, we imposed a downside step variation on the

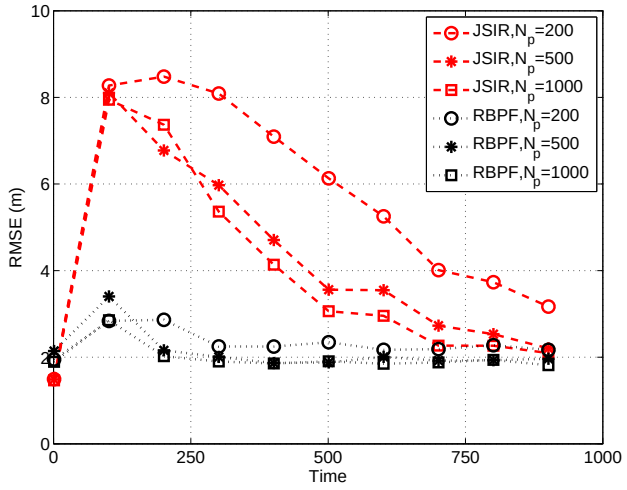
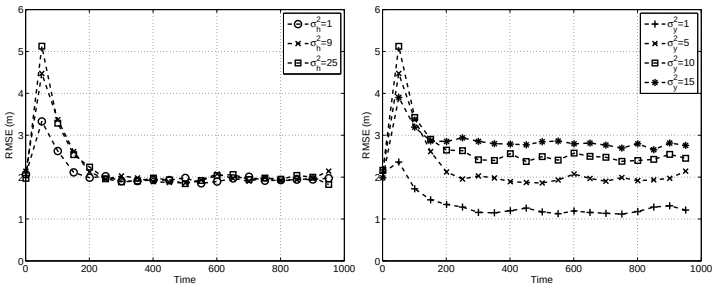


Figure 3.6 Positioning RMSE (both JSIR and RBPF) related to computer experiments concerning slow fading in the setup of Fig. 3.4 with number of particles in the range $N_p = 200 \div 1000$; here, $\sigma_h^2 = 9$, $\sigma_y^2 = 5$ and $N_p = 500$.



(a) σ_h^2 (b) σ_y^2

Figure 3.7 Computer experiments concerning slow fading for RBPF applied to the testbed of Fig. 3.4: positioning RMSE plotted vs time with different values of (a) the prior variance of the parameter in the range $\sigma_h^2 = 1 \div 25$ and (b) the measurement variance $\sigma_y^2 = 1 \div 10$; 500 particles were employed always.

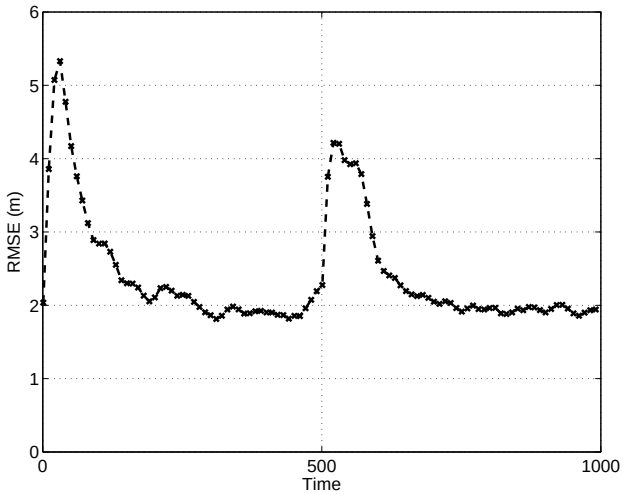


Figure 3.8 Computer experiments concerning slow fading in the setup of Fig. 3.4 with a -5 dB step variation on the parameter of a single AP at $k = 500$ in order to mimic a sudden shadowing; here $\sigma_h^2 = 9$, $\sigma_y^2 = 5$ and $N_p = 500$.

reference SS of one AP in order to simulate a sudden obstruction due, for example, to an obstacle. As before, this produces only a transient but the steady state RMSE is unchanged, as shown in Fig. 3.8 for RBPF (see the caption for the simulation details).

Fast fading

The fast fading effects are modeled by means of a Rice pdf as described in Sect. 3.1.2; beside JSIR, we test the AGB algorithm instead of the RBPF for the reasons explained in Sect. 3.4.3. Fig. 3.9 highlights a comparison between our proposals and the clairvoyant SIR algorithm, all applied with $\sigma_h^2 = 9$, $\sigma_y^2 = 5$ and $N_p = 1000$ to the testbed of Fig. 3.4; it is clearly shown the effectiveness of AGB due to a very sharp convergence with respect to JSIR, although the steady state value is slightly greater than JSIR's. This is due to the discrete set of parameter values assumed in AGB, whose choice is key in the algorithm performance. We prefer a uniform sampling of $\mathbf{h}$ in a suitable set, to account for

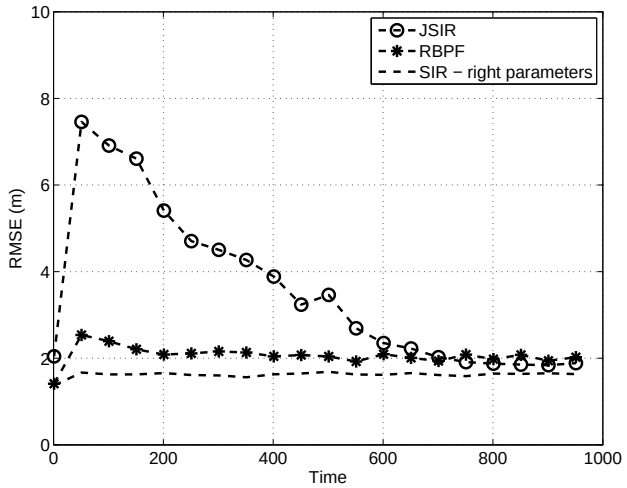


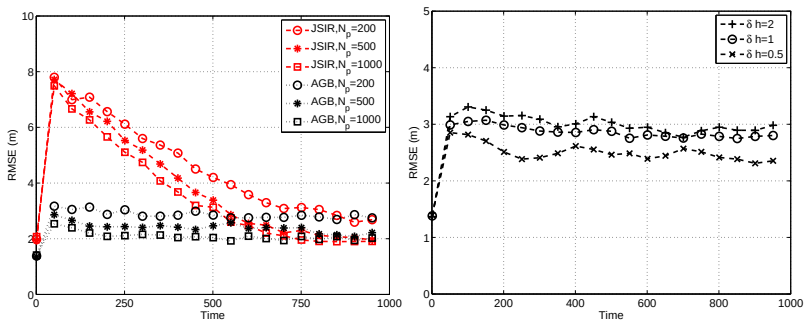
Figure 3.9 Fast fading effect onto RMSE shown by means of computer experiments on the testbed of Fig. 3.4: JSIR and AGB with $\delta h = 1$ dBm are applied with $\sigma_h^2 = 9$, $\sigma_y^2 = 5$ and number of particles in the range $N_p = 1000$.

sudden changes during the estimation. The step size, say δh, can be tuned by considering the full mismatch case: the maximum difference between the true value of the parameter and the closest discretized value is $\delta h/2$ and must be lower than the expected error Δh. Since we have found out in the computer experiments that $\Delta h \approx 0.5$ dBm, than we choose

$$\delta h = 1 \text{ dBm},$$

as a suitable compromise between algorithm complexity and performance (less 10 % of steady state RMSE increase with respect to JSIR).

The results of the analysis with respect to number of particles ($N_p = 200 \div 1000$), step size ($\delta h = 0.5 \div 2$) and down side variation of one AP's reference SS are shown, all in terms of positioning RMSE, in Fig. 3.10 and Fig. 3.11. In detail, Fig. 3.10.b confirms that there is room for improvement by setting a lower δh. The results about variations of σ_h^2 and σ_y^2 do not present relevant differences in relation to the slow fading case; we only report in Tab.



(a) Number of particles (b) Stepsize of the parameter

Figure 3.10 Computer experiments concerning fast fading applied to the testbed of Fig. 3.4: we report the positioning RMSE plotted vs time obtained (a) by both JSIR and RBPF with different set of particles in the range $N_p = 200 \div 1000$ and (b) by AGB with step size in the range $\delta h = 0.5 \div 2$; where not otherwise reported we use $\sigma_h^2 = 9$, $\sigma_y^2 = 5$, $N_p = 200$ and $\delta h = 1$ dBm.

	Slow Fading (RBPF)	Fast Fading (AGB)
$\sigma_y^2 = 1$	1.1936	-
$\sigma_y^2 = 3$	1.6961	2.2424
$\sigma_y^2 = 5$	1.9600	2.8125
$\sigma_y^2 = 10$	2.4623	2.8657
$\sigma_y^2 = 15$	2.7777	2.9516

Figure 3.12 Steady state values of the positioning RMSE for different values of the measurement variance in the range $\sigma_y^2 = 1 \div 15$; here $\sigma_h^2 = 9$, $N_p = 500$ and $\delta h = 1$ dBm.

3.12 the RMSE steady state values vs the measurement variance σ_y^2.

3.7.2 Other cases

The exponent α can be sometimes an unknown parameter as well. We have tested our algorithms in scenarios where only $\boldsymbol{\alpha}$ is unknown or variable, that is

$$\mathbf{p} = \boldsymbol{\alpha},$$

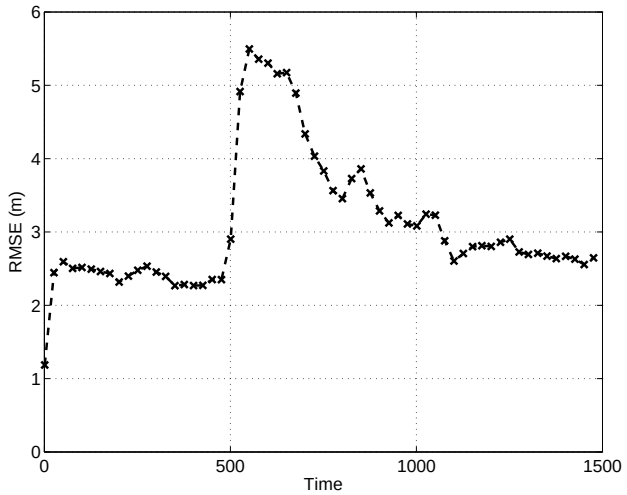


Figure 3.11 Fast fading effect: RMSE restoration after a -5 dB step variation on one AP's h at the time instant $k = 500$; we use $\sigma_h^2 = 9$, $\sigma_y^2 = 5$, $N_p = 200$ and $\delta h = 1$ dBm.

similarly to the reference power in Sect. 3.7.1. The results are omitted because they are fully similar to the previous case. More interestingly, in the composite case

$$\mathbf{p} = \{\mathbf{h}, \boldsymbol{\alpha}\}$$

severe identifiability problems arise. We shall introduce the point through a simple example: let y be a scalar observation, a and b the (partially) unknown parameters. It is straightforward that if $y = a + b$, we cannot discriminate between a and b observing only y and thus the identifiability problem cannot be solved (unless specific side information is available). On the other hand, if $y = a + b \cdot t$, with t known, the variations of a and b have different impacts on y, since the former is an additive constant on y and the latter is a slope term. In our case we have instead

$$y = a + b\hat{t} + n,$$

where $\hat{t}$ is the estimate of t based on the same observations y. When also a random noise is added all the algorithms tested have shown an unstable behavior. This is why in the next chapters we will always deal with only the reference power estimation.

3.7.3 Variance estimation

The first scenario being discussed in this section requires the estimation of only the AP's noise variances. In Fig. 3.13 we show the tracking of the variances with different forgetting factors and the corresponding positioning RMSE. The comparison with the performances of the algorithm without the variances tracking highlights the action of the variance filter. The forgetting factor sets the trade off between velocity of the convergence and residual uncertainty and too low values can lead to stability problems.

More interestingly, we have tested the variance filter matched with the reference power estimation and the results are in Fig. 3.14. Even though in the transients some coupling between the estimators is evident, an enhanced stability is shown in the end both in the parameters tracking and in the positioning performances. Analogous results are found in the case of contemporary estimation of variances and $\boldsymbol{\alpha}$, while severe identifiability problems are again reported in the estimation of all parameters.

3.8 Evaluation of the Cramer Rao Lower Bound

The sequential Posterior CRLB described in Sect. 1.4 was cast on our framework to study its theoretical limits. The matrices defined in Sect. 1.4.1 and in Sect. 1.4.2 w/o and with unknown parameters respectively are here specified and implemented numerically.

In the case we assume a Lognormal measurement model the matrices D of Sect. 1.4.1 can be expressed like follows:

$$\begin{cases} D_k^{11} &= E_{\mathbf{x}_k}\left\{ f_x^T(\mathbf{x}_k) R_v^{-1} f_x(\mathbf{x}_k) \right\}, \\ D_k^{12} &= -E_{\mathbf{x}_k}\left\{ f_x^T(\mathbf{x}_k)^T \right\} R_v^{-1}, \\ D_k^{21} &= \left[D_k^{12} \right]^T, \\ D_k^{22} &= R_v^{-1} + E_{\mathbf{x}_{k+1}}\left[g_x^T(\mathbf{x}_{k+1}) R_n^{-1} g_x(\mathbf{x}_{k+1}) \right]. \end{cases}$$

The functions $f_x(\cdot)$ and $g_x(\cdot)$ are the first order linearization of

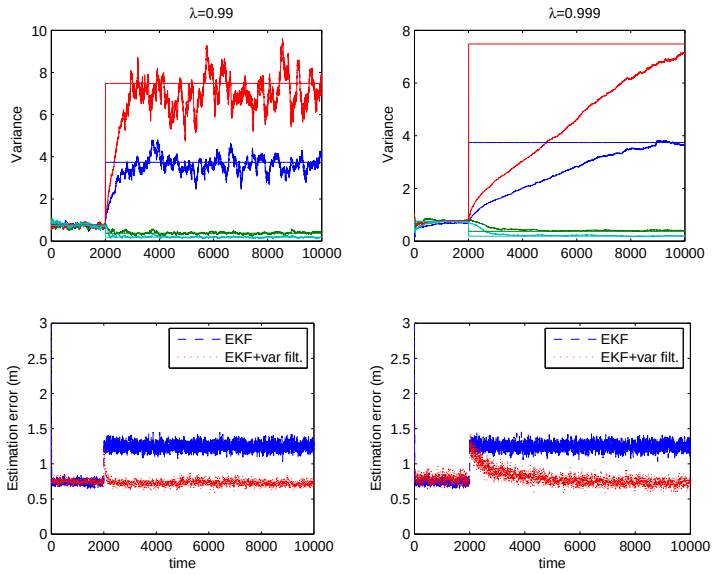


Figure 3.13 Variance estimation for all 4 APs by means of the exact filter, on the left $\lambda = 0.99$, on the right $\lambda = 0.999$: variances tracking (1 run) and mean experimental error (over 200 MC trials) in the target location; non-stationary SNR, starting from $SNR = 20$ dB, $\alpha = 1.7$.

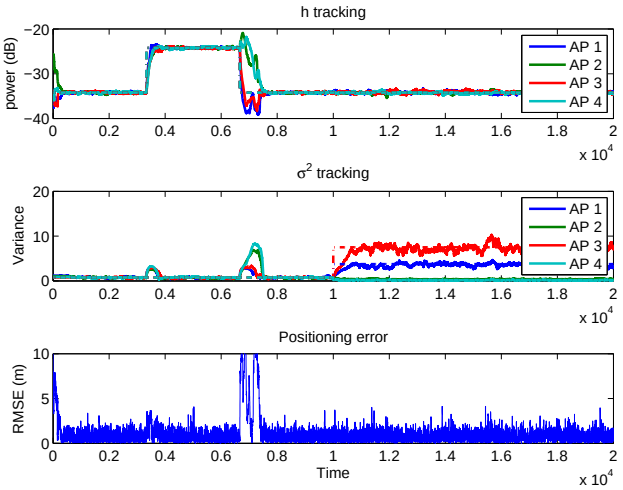


Figure 3.14 h (top) and variances (middle) estimation for all 4 APs by means of the exact filter: in both cases $\lambda = 0.99$; on the bottom there is the mean experimental error in the target location; non-stationary SNR, starting from $SNR = 20$ dB, $\alpha = 1.7$.

the functions $f(\cdot)$ and $g(\cdot)$ respectively:

$$\begin{aligned} f_x(\mathbf{x}_k) &= \nabla_{\mathbf{x}_k} f(\mathbf{x}_k), \\ g_x(\mathbf{x}_{k+1}) &= \nabla_{\mathbf{x}_{k+1}} g_{k+1}(\mathbf{x}_{k+1}). \end{aligned}$$

The computations involve matricial operations and are described in App. D; we notice that $f_x(\mathbf{x}_k) \approx F\mathbf{x}_k$ when the user is far away from the obstacles (see Sect. 3.1.1). The FIM J_k, whose inverse provides the CRLB, is thus given by eq. (1.36).

In the case of unknown parameters the H matrices of Sect. 1.4.2 result in

$$\begin{cases} H_k^{11} &= E_{\mathbf{x}_k}\left\{ f_x^T(\mathbf{x}_k) R_v^{-1} f_x(\mathbf{x}_k) \right\}, \\ H_k^{12} &= 0, \\ H_k^{13} &= -E_{\mathbf{x}_k}\left\{ f_x^T(\mathbf{x}_k)^T \right\} R_v^{-1}, \\ H_k^{22} &= E\left\{ g_p^T(\mathbf{x}_{k+1}, \mathbf{p}_k) R_n^{-1} g_p(\mathbf{x}_{k+1}, \mathbf{p}_k) \right\}, \\ H_k^{23} &= E\left\{ g_p^T(\mathbf{x}_{k+1}, \mathbf{p}_k) R_n^{-1} g_x(\mathbf{x}_{k+1}, \mathbf{p}_k) \right\}, \\ H_k^{33} &= R_v^{-1} + E\left\{ g_x^T(\mathbf{x}_{k+1}, \mathbf{p}_k) R_n^{-1} g_x(\mathbf{x}_{k+1}, \mathbf{p}_k) \right\}, \end{cases}$$

where

$$\begin{aligned} f_x(\mathbf{x}_k) &= \nabla_{\mathbf{x}_k} f(\mathbf{x}_k), \\ g_p(\mathbf{x}_{k+1}, \mathbf{p}_k) &= \nabla_{\mathbf{p}_k} g(\mathbf{x}_{k+1}, \mathbf{p}_k), \\ g_x(\mathbf{x}_{k+1}, \mathbf{p}_k) &= \nabla_{\mathbf{x}_{k+1}} g(\mathbf{x}_{k+1}, \mathbf{p}_k). \end{aligned}$$

The FIM is thus computed like in the Sect. 1.4.2.

In the case we assume a Rice measurement model, the computation of the gradients is much more tedious in $D_k^{22}, H_k^{22}, H_k^{23}$ and H_k^{33}; further details are proposed in App. D.

We give two simulative shots about the CRLB evaluation. In Fig. 3.15 the (root) CRLB is compared to the RMSE obtained from the JEKF with SNR=$10, 20$, given the Gaussian likelihood model. The most interesting point to figure out is that there is a divergence between the bound and the real performances by lowering the SNR.

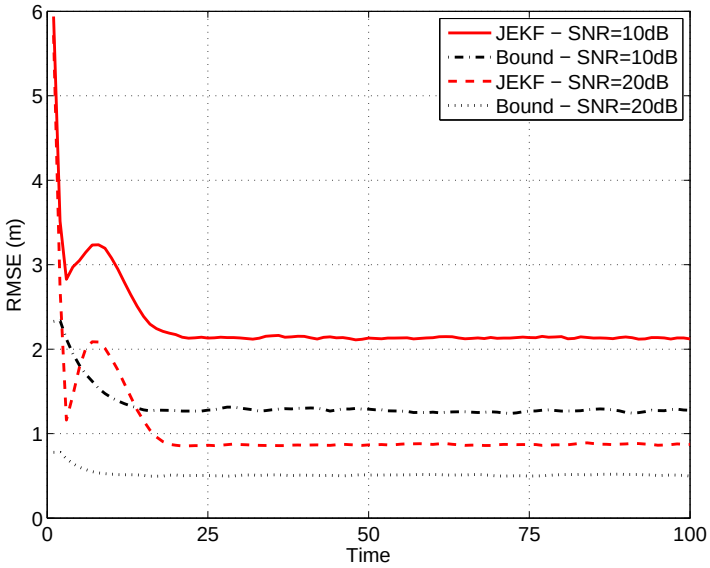


Figure 3.15 Bayesian (root) CRLB for the sequential tracking, compared with the EKF empirical RMSE. Here $\tau = 1$, $\sigma_v = 0.1$, data SNR (Gaussian case) equal to 10 and 20 dB; 10000 Monte Carlo trials.

In Fig. 3.16 we show the limits in the cases described throughout this chapter: full known model (black continuous line), reference power estimation (dotted red), exponent estimation (green semi-dotted) and both (dotted blue). As foretold in Sect. 1.4.2, they all converge to the same value, since the information describing the parameters grows endless. This is true also in the case of both $\mathbf{h}$ and $\boldsymbol{\alpha}$ estimation, where the algorithms fail: before saying that the limit makes no sense in this case (that is possible as well), we should remember that the identifiability problem found in the Sect. 3.7.2 could arise from a lack of sensibility in the algorithms rather than being a theoretical problem.

3.9 Experimental Results

Experiments on real datasets have been carried out in the scenary depicted in Fig. 3.17: it consists of an underground car park sized

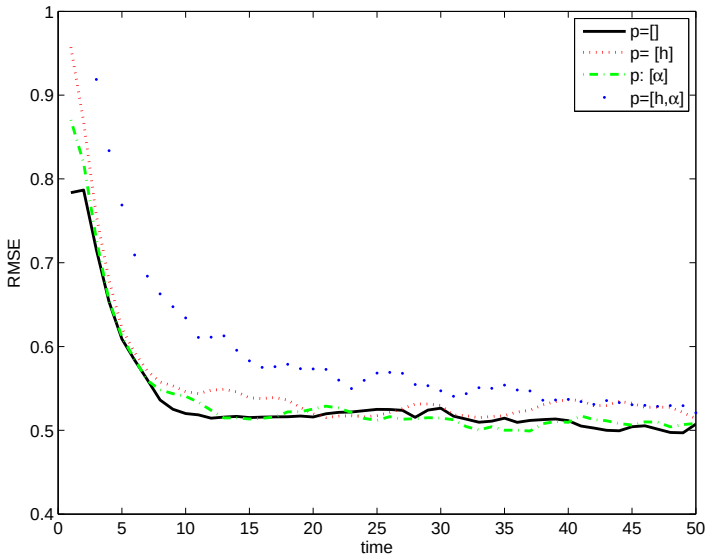


Figure 3.16 Bayesian (root) CRLB for the sequential tracking in all the examined cases. Here the initial state $\mathbf{x}_0$ has a Gaussian distribution on the measure area centered in the true position and with unitary covariance matrix, $\tau = 1$, $\sigma_v = 0.1$, $SNR = 20$ dB.

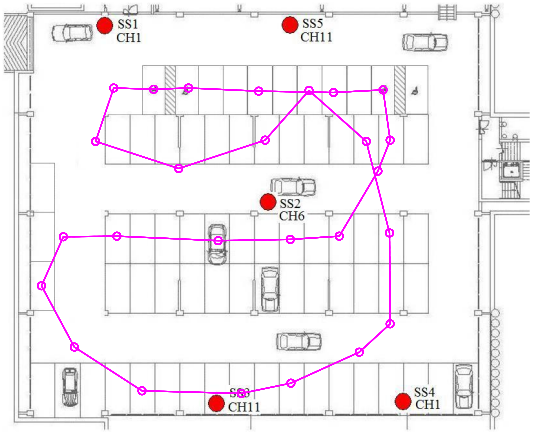


Figure 3.17 Experimental setting: the APs are denoted by red circles and are indicated with $SSi, i = 1, \ldots, 5$. Also the channel of the 802.11 band used by any AP is indicated at a 2.4GHz frequency. Note that only 3 channels are available, thus, in order to avoid interferences, we have exploited the fact that the distances between SS3 and SS5 and between SS1 and SS4 are greater than the APs range in reception.

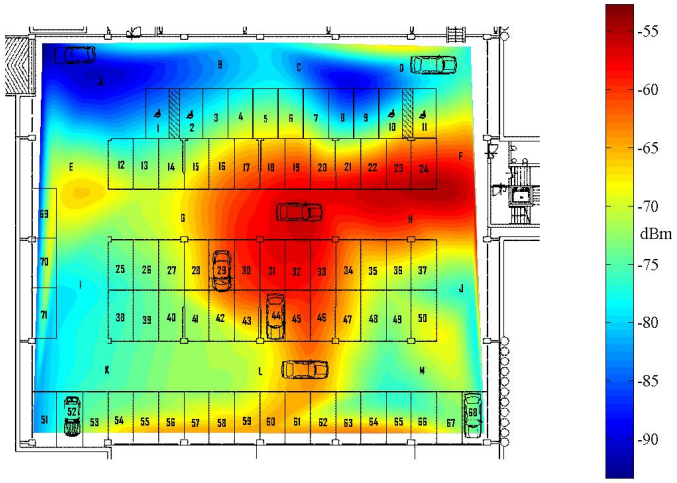


Figure 3.18 Graphical RSS from SS2; smoothing through splines.

about 45×40 m, in which a 802.11 (WiFi) network with 5 APs 3COM 7760 operates. In the parking area there are obstacles of reinforced concrete and cars. The number, type and position of the cars in the assigned sites are absolutely random and cannot be put under control. The sampling frequency is about 1 measurement per second, and the measurements are the power levels expressed in dB. Furthermore, they are quantized in order to be integer values, but we will not explicitly take account of the consequent rounding in our algorithms.

A starting analysis of our data shows how difficult the inference is because of the very high dispersion in the received powers. In fact, in Fig. 3.18 we show the mean power measurements collected in a training stage throughout the whole setting and relative to the AP called SS2; an interpolation based on splines is adopted to smooth the picture. We can see how the signal propagation is not circular at all, but is highly influenced by the obstacles (on the ground truth no information was noted down about the cars).

The training dataset consisted of 1000 measurements vectors

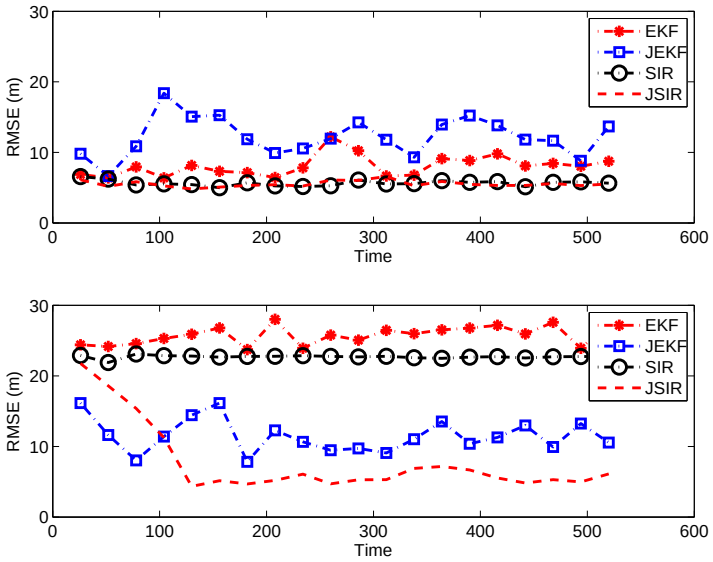


Figure 3.19 RMSE of KF, JKF, SIR and JSIR applied to the setting in Fig.3.17: (top) the estimated path loss model is employed, (bottom) h is alterated with an error of +10 dB for all APs.

(50 points, 20 measurements per point), useful to estimate the parameters $\mathbf{h}$ and $\boldsymbol{\alpha}$ of the path loss model of eq. (2.4) for each AP through a Minimum Squares optimization rule. After that, a virtual path has been simulated in the car park (marked line in Fig. 3.17), sorting the measurements collected in various points. In this particular scenario we have observed a bad behavior of JEKF approach (see Fig. 3.19). This is due to the significant departure from Gaussianity (in dB) hypothesis (the Rice distribution better fits the channel variability in the considered scenario) and to a very high variance affecting the channel noise.

The results, shown in Fig. 3.19, highlight the adaptive behavior of the joint estimator. Indeed, using the parameters of the propagation model estimated on real data as the actual value for SIR and the initial estimation for JSIR, the two algorithms show the same performances (about 5 meters of error, see the plot on top). By corrupting the same parameters with an increase of 10 dB for all APs, the initial error in both algorithms significantly

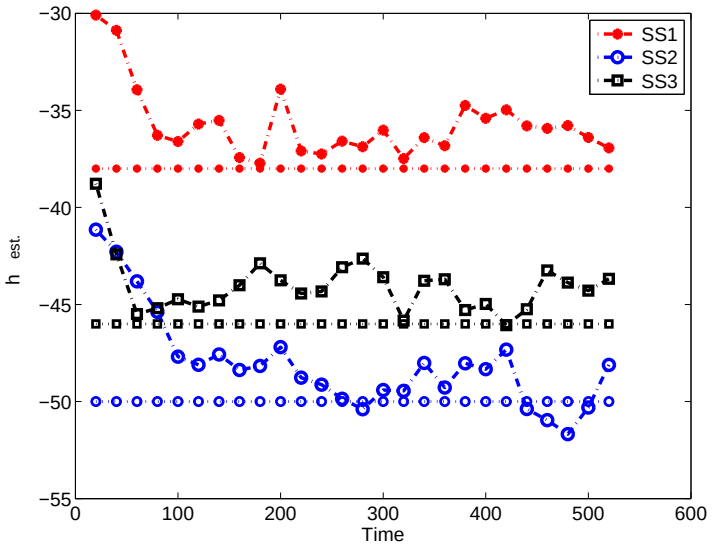


Figure 3.20 h tracking performed by the approximated JSIR algorithm for SS1, SS2 and SS3. The uncorrupted values are indicated with dotted lines.

gets worse. However the JSIR quickly retrieves the performances it had in the former case (see the plot on bottom), due to its tracking capability. An example of learning curve regarding h relative to some APs is presented in Fig. 3.20. Further experiments on adaptive behavior, performed, for example, by imposing different errors among the APs, have yielded quite similar results.

3.10 Concluding remarks

We have shown the basics in the application of the Bayesian Probability Theory to indoor localization with the addition of the environment profiling. We modified some classic algorithms to account for some propagation parameters involved in the chosen models. Some further issues have been undertaken, like the obstacle avoidance and the comparison between measurements generated under different probabilistic models. We have evaluated our algorithms' effectiveness and robustness by means of extensive computer experiments, as well as a very advers experimental scenario.

In the next chapters, these techniques will be further developed to address more elaborate and attractive problems. Thus also many practical details, here omitted because beyond our actual aims, will be dealt with.

Chapter 4

Co-Localization: Exploiting WLAN Signals of Opportunity

So far, we analyzed positioning systems based on the assumption that we know precisely where the APs are. Nevertheless, in several practical situations this information is not available or, sometimes, it is not accurate enough. In fact, in the present chapter we drop this hypothesis and undertake the topic known in the literature like co-localization.

Consider this problem, depicted in Fig. 4.1: a person having a WiFi compliant device, such as a smartphone, enters a building where a WiFi network is active. After the association procedure, the user equipement (UE) receives from the fixed network some information about the building structure, such as the floor map, position and features of the active APs and so on. After that, the UE starts measuring the RSS and can either perform the localization on its own end (self-positioning) or send the measurements to the server end (server based positioning) by means of a suitable protocol (e.g. see the localization architecture proposed in [ABG+10]). Suppose, now, that it detects pilot signals also from APs whose positions are unmanaged (for example, because they either belong to other networks or have been moved recently): we wonder whether we can usefully exploit such measurements or we should just ignore them.

The answer is of course yes; in this chapter we are going to develop a Bayesian framework aimed at exploiting the RSS from APs located in unknown positions. We will find that an estimation of the unmanaged APs is made in terms of the position pdf, that is used to 'weight' their measurements. The result is that as our knowledge about those APs grows, the corresponding RSS are even more relevant in the Bayesian algorithms.

After a literature review about co-localization (Sect. 4.1), we examine in Sect. 4.2 a possible architecture of the practical positioning system. Then, the theoretical framework will be developed in Sects. 4.3 and 4.4; an intuitive study on mapping is proposed in Sect. 4.5 and we develop the Bayesian filter in Sect. 4.6, whose approximation is dealt with in Sect. 4.7. Simulative and real world experiments are provided in Sects. 4.8 and 4.9 respectively.

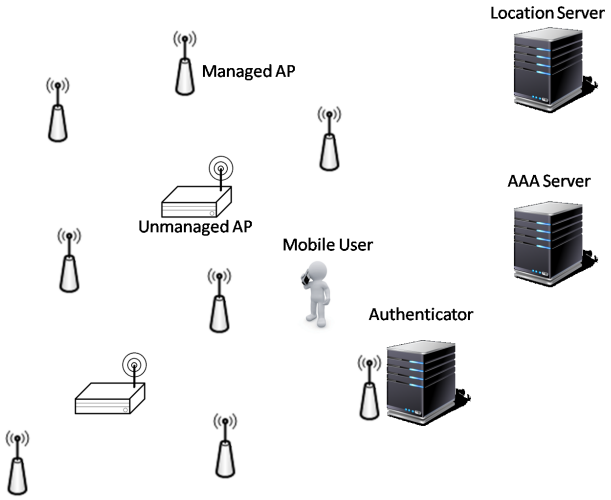


Figure 4.1 Scheme representing the architecture of a positioning system based on both managed and unmanaged WLANs.

4.1 Literature review

The idea of employing further information to improve the accuracy of WLAN based positioning algorithms is already present in the literature. *Cooperative* localization refers to scenarios in which different UEs are allowed to exchange some data [DRFF10]. In *opportunistic* or *redundant* localization different kinds of infrastructures are jointly used to infer the position of a mobile user [ZZ09]. Moreover, wireless nodes mapping by means of a mobile agent equipped with a GPS receiver (outdoors) is exploited in [SR04] by means of a probabilistic framework.

The co-localization paradigm is instead introduced in [PPY+12], where the authors develop a first machine-learning-based approach that combines collaborative filtering with graph based semi supervised learning. A sequential extension of the algorithm is also provided that can meet also environmental changes and, in a further scenario, they also integrate IMU's data. The results are interesting, even if not easily evaluable, being their experimental scenarios not accurately specified and no absolute measures of the error provided.

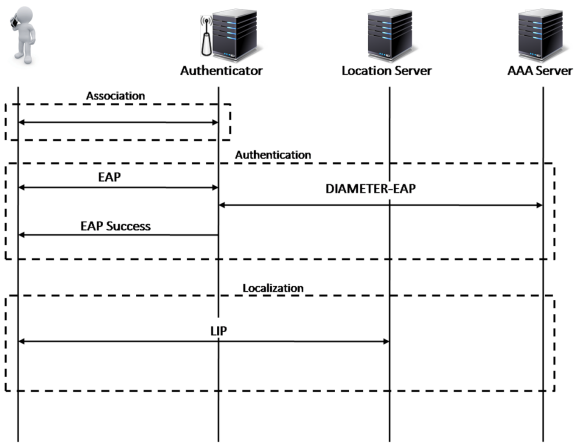


Figure 4.2 Protocols involved in our scenario for authentication and localization.

4.2 Architecture

A simple architecture which allows the exploitation of unmanaged APs makes use of a dedicated Locations Server (LS) and a suitable protocol, called Location Information Protocol (LIP) [ABG+10]. We will consider next a IEEE 802.11 (WiFi) network.

Fig. 4.2 depicts the procedures which take place for a WLAN network before starting location:

1. IEEE 802.11 association process takes place when UE enters in the network radio coverage. UE starts sending an Association Request to the AP. After the AP receives the association request successfully, it will reply with Association Reply. When UE receives the Association Reply message, it changes its status from a new station to a registered station.

2. The second step is the WLAN Direct IP Access procedure. In this case, the network will authenticate a user on the basis of the credentials.

The solution we propose is founded on a localization protocol, LIP, for the exchange of signaling data between UE and LS. LIP is

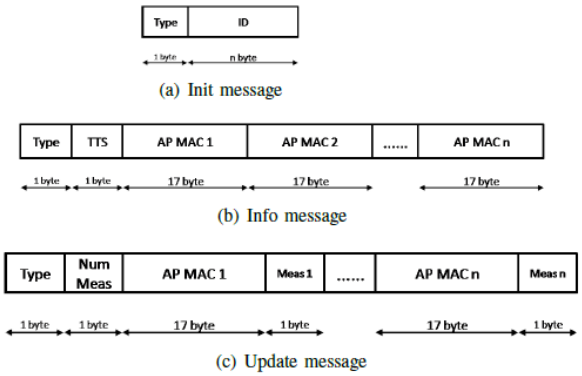


Figure 4.3 Types and corresponding structure of LIP messages.

an application layer protocol based on UDP, employing 3 types of messages showed in Fig. 4.3 and illustrated in the next paragraph.

4.2.1 Location message exchange

We now explain how the location process takes place in our design. After a successfully completed WLAN Direct IP Access procedure between UE and AAA server, the Authenticator sends a LIP Init message (Fig. 4.3.a) to the LS, containing the new user ID. The LS checks whether the user is allowed to access the network services, and then it sends to UE a LIP Info message (Fig. 4.3.b) containing a list of the managed APs list with the relative MAC and, eventually, the measurement rate to adopt. If this number is set to zero no measurement is collected in this phase, otherwise the user starts to collect measurements at the given rate and sends them to the LS, through LIP Update messages (Fig. 4.3.c). The latter elaborates the received data in order to estimate the user's location. If need be, the LS can send a LIP Info message to the user in order to stop the RSS measurements dispatching. The succession of LIP messages within a session is listed in Fig. 4.4.

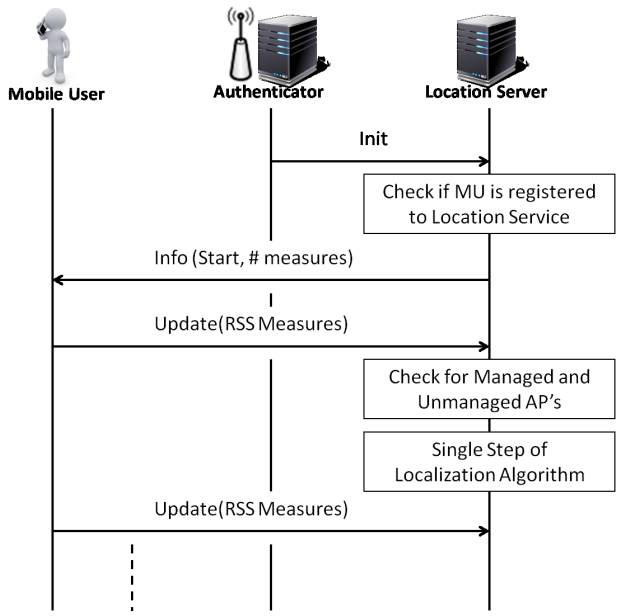


Figure 4.4 Flow of LIP messages within a session.

4.3 Theoretical setting

Let us focus now on the derivation of the Bayesian algorithms aimed at co-localization, provided the needed degree of formalization.

A pedestrian user is assumed to walk inside an indoor area $\mathcal{A} \subseteq \mathcal{R}^2$ of interest; its two dimensional position $\boldsymbol{\theta}_t \in \mathcal{A}$ is expressed in terms of a locally defined, Cartesian reference system (x, y) and evolves in time from $t_0 = 0$ on. The user's movement is driven by its intention and, at a lower level, by its sensorial systems that interact with the building map to avoid obstacles and choose a path leading to the goal; since, differently from what happens with the robots, these variables are not measurable, we ought to use general models exploiting the continuity of the user's trajectory. In particular, we refer to the NCVM presented in Sect. 2.5.3. The only difference is that now we'd rather sample the time in a non regular way, i.e. at the arbitrary instants $0 = t_0 < t_1 < \ldots < t_{k-1} < t_k$ and build a user's state vector into which we stack its

position $\boldsymbol{\theta}_t$ and velocity $\dot{\boldsymbol{\theta}}_t$:

$$\mathbf{x}_k \hat{=} \left[\boldsymbol{\theta}_k^T, \dot{\boldsymbol{\theta}}_k^T\right]^T, \tag{4.1}$$

where T means transpose and we denote the time instants by their indices. The dynamic model of the user is thus modeled by the stochastic process:

$$\mathbf{x}_k = F_k \mathbf{x}_{k-1} + \mathbf{v}_k$$

starting from an arbitrarily distributed initial state $\mathbf{x}_0$ with prior pdf

$$\mathbf{x}_0 \sim \mu_x(\mathbf{x}_0). \tag{4.2}$$

The prediction matrix F_k, provided $\tau_k \hat{=} t_k - t_{k-1}$, is defined by

$$F_k = \mathbf{I}_2 \otimes \begin{pmatrix} 1 & \tau_k \\ 0 & 1 \end{pmatrix},$$

while the zero mean white process $\mathbf{v}_k$, hence supposed Gaussian, has a covariance matrix Q_k

$$Q_k = E\left[\mathbf{v}_k \mathbf{v}_k'\right] = \sigma_v^2 \mathbf{I}_2 \otimes \begin{pmatrix} \frac{1}{3}\tau_k^3 & \frac{1}{2}\tau_k^2 \\ \frac{1}{2}\tau_k^2 & \tau_k \end{pmatrix}.$$

The continuous time process noise intensity σ_v and τ_k must always match the requirements for NCVMs, i.e. the changes in velocity during τ_k must be much smaller than the actual user's velocity (in norm)

$$\sigma_v \sqrt{\tau_k} \ll ||\dot{\boldsymbol{\theta}}_k||. \tag{4.3}$$

The area under test is equipped with N_{AP} Access Points (APs); the arbitrary j-th AP, located statically in $\boldsymbol{\theta}_{j,AP} \in \mathcal{A}$, broadcasts periodically its unique $SSID_j$ inside a beacon signal. The user WLAN device is able to measure the amplitude $y_{j,k}$ of the beacon signal emitted at time k by the j-th AP and has available a more complex Wlan measurement meant for convenience as the n-uple

$$s_{j,k} =< SSID_j, y_{j,k}, \boldsymbol{\theta}_{j,AP}, \mathbf{p}_j >,$$

where $\mathbf{p}_j$ is a set of static informations about the AP useful in the positioning application, as we will see shortly. The set of WLAN measurements collected at k will be denoted by $\mathbf{s}_k$. Our main assumption is that the measurements $s_{j,k}$ are independent both in time and among different APs, given the user's state.

For simplicity, we will typically refer to the set of RSS measurements at $k > 0$ and to the set of the APs' positions and parameters as

$$\mathbf{y}_k = \{y_{j,k}\}\,,\ \boldsymbol{\theta}_{AP} = \{\boldsymbol{\theta}_{j,AP}\}\,,\ \mathbf{p} = \left\{\mathbf{p}_j\right\}, \tag{4.4}$$

for $j = 1, \ldots, N_{AP}$, respectively.

4.3.1 Propagation model and parameters

We assume the Lognormal fading model and path loss propagation model as described in the Sect. 3.1.2. Like before, we give attention to the parameters $\mathbf{h}$ and $\boldsymbol{\alpha}$ that are the reference SS and the decaying exponent, respectively.

4.4 Problem statement

The positioning techniques based on the models of Sect. 4.3 require the knowledge of both the APs' positions $\boldsymbol{\theta}_{j,AP}$ and the parameters involved; in our case the parameters are h_j, α_j and σ_j that will be stacked into the vector $\mathbf{p}_j$. In our setting we assume that we have only partial knowledge about some of the APs. In this view, we split the matrices in eq. (4.4) after a suitable sorting:

$$\mathbf{y}_k = \left\{\mathbf{y}_k^F, \mathbf{y}_k^P\right\}\,,\ \boldsymbol{\theta}_{AP} = \left\{\boldsymbol{\theta}_{AP}^F, \boldsymbol{\theta}_{AP}^P\right\}\,,\ \mathbf{p} = \left\{\mathbf{p}^F, \mathbf{p}^P\right\}, \tag{4.5}$$

where the superscript F denotes the fully known APs and P denotes those partially known APs. The partial knowledge of these latters is expressed by defining prior pdfs for both $\boldsymbol{\theta}_{AP}^P$ and $\mathbf{p}^P$, viz.

$$\mu_\theta(\boldsymbol{\theta}_{AP}^P), \qquad \mu_p(\mathbf{p}^P), \tag{4.6}$$

that can be factorized, being position and parameters for different APs assumed mutually independent:

$$\mu_\theta(\boldsymbol{\theta}_{AP}^P) = \prod_j \mu_{\theta_j}(\boldsymbol{\theta}_{j,AP}^P), \tag{4.7}$$

$$\mu_p(\mathbf{p}^P) = \prod_j \mu_{p_j}(\mathbf{p}_j^P). \tag{4.8}$$

Note that the independence assumption for the parameter vectors $\mathbf{p}_j$ is not straightforward and is more accurate if the APs are far enough each other, otherwise the propagation features can be correlated also for different APs.

The Bayesian score function (at k) is represented by

$$p\left(\mathbf{x}_{1:k}, \boldsymbol{\theta}_{AP}^P | \boldsymbol{\theta}_{AP}^F, \mathbf{p}^F, \mathbf{y}_{1:k}\right); \tag{4.9}$$

it is causal, since it refers only to past and present measurements, and it needs a recursive computation. The variables bearing the superscript F are inserted for clarity and will be omitted henceforth. Note that the parameters set $\mathbf{p}^P$ does not appear explicitly but is nevertheless necessary in the computation of the pdf in eq. (4.9); our choice is to consider them by means of the marginalization:

$$\int p\left(\mathbf{x}_{1:k}, \boldsymbol{\theta}_{AP}^P, \mathbf{p}^P | \mathbf{y}_{1:k}\right) \text{ d } \mathbf{p}^P, \tag{4.10}$$

that will be detailed later on.

The variables and their mutual correlations are graphically shown in the Dynamic Bayesian Network of Fig. 4.5, where two adjacent time slices are depicted. The user's intention **Int**, the visual system **Vis** and the building Map are introduced for clarity, but are not evaluated in this paper; the DBN shows clearly that the AP's position and parameters are considered static in our framework.

4.4.1 Approaching the solution

We propose a solution to eq. (4.9) that is based on the Rao Blackwellized Particle Filters [DFG01]. The key idea is to factorize the

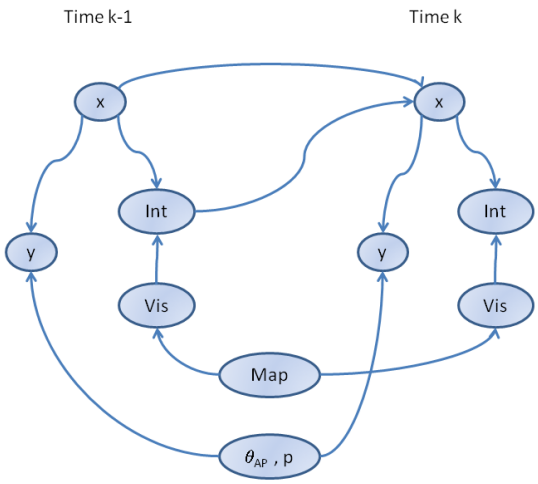


Figure 4.5 Dynamic Bayesian Network concerning colocalization.

pdf by splitting the localization part from the mapping part

$$p\left(\mathbf{x}_{1:k}, \boldsymbol{\theta}_{AP}^{P} | \mathbf{y}_{1:k}\right) = p\left(\boldsymbol{\theta}_{AP}^{P} | \mathbf{x}_{1:k}, \mathbf{y}_{1:k}\right) \cdot p\left(\mathbf{x}_{1:k} | \mathbf{y}_{1:k}\right). \tag{4.11}$$

Inside a particle, only the user's trajectory is drawn and then used to update the mapping pdf deterministically; therefore the particle will not contain a hypothesis about the partially known AP's positions, but their pdf. The mapping pdf will be at first very widespread (unless sharp prior pdfs are assumed), and thus the measurements from such APs will be almost useless to the localization purpose. Nevertheless, since the mapping pdf can be seen like the residual error in the AP's knowledge and its breadth as a measure for it, we can expect that it gets even sharper at any new measurements, leading to a better exploitation of the measurements.

There are two possible causes of drawbacks in this approach: i) co-localization is attractive only if the map pdf converges (and thus the measurements from the partially known APs are profitably used) quickly in relation to the walking time; ii) even supposing

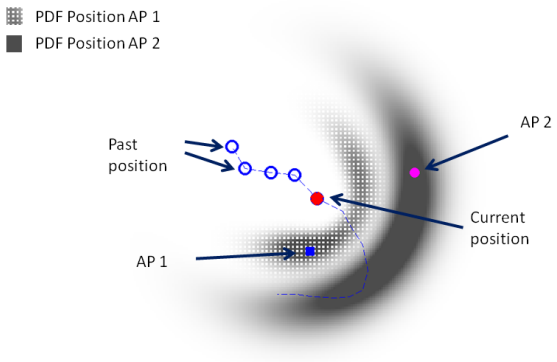


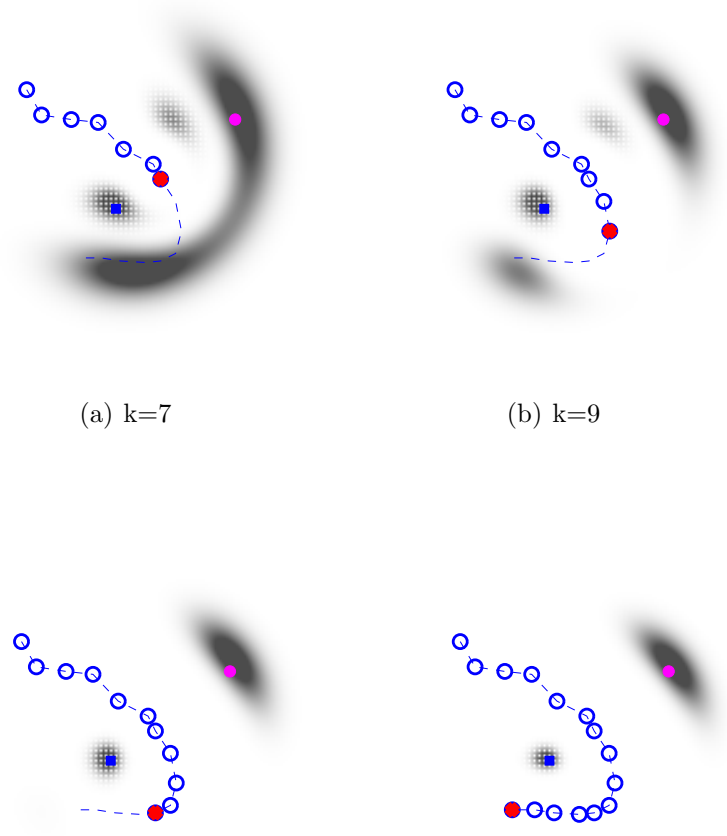
Figure 4.6 Mapping via simulated experiments: a user is assumed to walk on the blue sketched trajector and the actual position is the red full circle; APs indicated by squares and their measurements are collected when the user lays in the positions denoted by blue circles. The first AP is denoted by a small blue square and its pdf is graphically reported by means of a dotted texture (darker color means higher pdf); similarly the second AP's position is stressed by a magenta circle and its pdf by a plain texture. The notation we use is also summarized in Fig. 4.6, where the maps after $k = 5$ measurements are provided; here $\alpha = 2$ and $\sigma = 8$ for both APs.

that the number of particles is sufficiently high in order to properly explore the state space, nobody ensures us about the convergence of the algorithm to the 'true' trajectory and to the true 'map', even if valuable guidance in this direction is provided by several authors in other problems [MTKB02, BMM11].

Now we carry out a preliminary simulative analysis to get an answer to this issues. As for the point (i) we do not need to develop the full algorithm, but only to test the mapping part. Then, more theoretical insight will be provided, before approaching the latter point.

4.5 How mapping works

The consistency of mapping by means of RSS measurements was already shown in [SR04] where the positions of some sensors within a wireless network were investigated by means of a mobile probe



(a) k=7 (b) k=9

(c) k=11 (d) k=15

Figure 4.7 Mapping evolution in the simulative scenario reported in Fig. 4.6 for $k = 7, 9, 11, 15$.

whose position was intended to be exactly known. They used a probabilistic approach similar to ours, but we propose further insight.

We assume that the parameters in $\mathbf{p}_j^P$, corresponding to the j-th AP, are known and we denote its position pdf with

$$p_k(\boldsymbol{\theta}_j^P) \hat{=} p(\boldsymbol{\theta}_{j,AP} | \mathbf{p}_j^P, \mathbf{x}_{1:k}, \mathbf{y}_{j,1:k}); \tag{4.12}$$

a simple application of the Bayes Theorem leads to the recursive computation:

$$p_k(\boldsymbol{\theta}_{j,AP}) \propto p(y_{j,k} | \mathbf{x}_k, \mathbf{p}_j^P, \boldsymbol{\theta}_{j,AP}) \cdot p_{k-1}(\boldsymbol{\theta}_{j,AP}), \tag{4.13}$$

where we implicitly exploit the relations between the variables there are encoded in the DBN of Fig. 4.5. We shall note that the likelihood function of the single measurement, being Gaussian w.r.t. $y_{j,k}$, has a circular symmetry in the space of $\boldsymbol{\theta}_{j,AP}$, Log-normally distributed in range and uniform in angle. After having collected k measurements, the map is simply the (normalized) product of those k circular non-concentric functions which results in a narrow peak on the AP's location, if k is sufficiently high.

In Figg. 4.6-4.7 we show the results of a simulative experiment in which a user is assumed to walk on the blue sketched trajectory. The measurements from two static APs are collected when the user lays in the positions denoted by blue circles, according to the models of Sect. 4.3, with $\alpha = 2$ and $\sigma = 8$. The red full circle indicates the user's actual position; the first AP is denoted by a small blue square and its pdf is graphically reported by means of a dotted texture (darker color means higher pdf); similarly the second AP's position is stressed by a magenta circle and its pdf by a plain texture. The notation we use is also summarized in Fig. 4.6, where the maps after $k = 5$ measurements are provided. Here we can see the two pdfs have resulted into semicircular pdfs, narrower for the AP 1, that is positioned closer to the trajectory, and more widespread for the AP 2. Fig. 4.7 shows how the same experiment evolves for $k = 7, 9, 11, 15$; the notation is the same as in Fig. 4.6. As for the AP 1, yet at $k = 7$ the pdf is composed

of two quite narrow peaks: one overlaps the right AP's position, while the latter lays in the symmetric position with respect to the almost linear trajectory. This complies with the fact that RSS measurements are basically noisy distance measurements. When the users start to turn on his right, he breaks the symmetry and the specular peak vanishes; at the same time, the right peak gets even narrower.

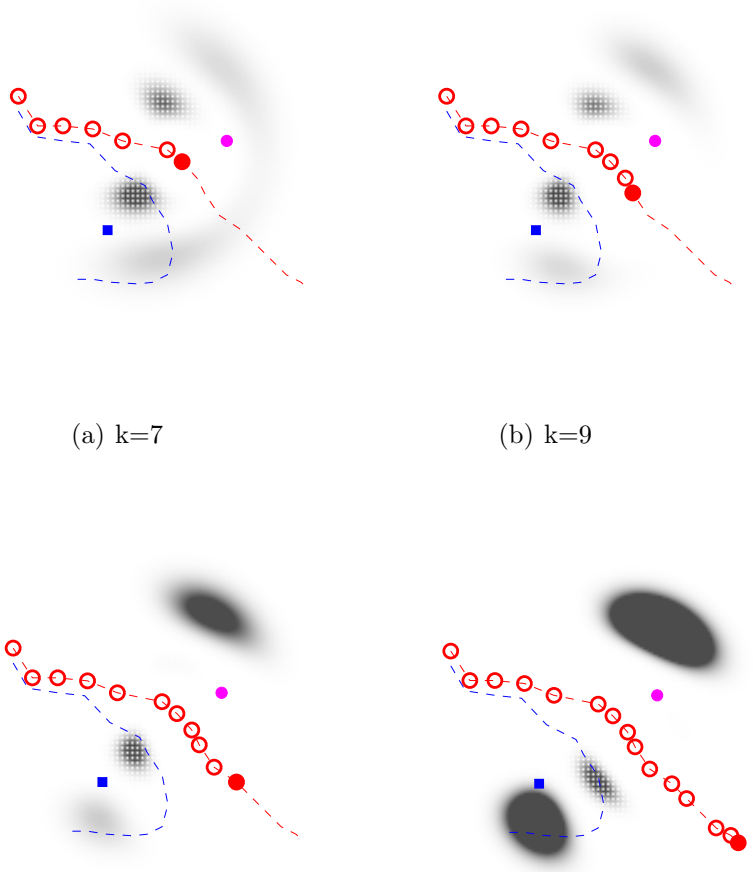
Focusing our attention on the AP 2, the only difference we can note is that the resulting peak in its position pdf is much less narrow, since it is farther from the user's trajectory. In fact, it is implicit in the RSS propagation model that the uncertainty over the distance measurement obtained from the RSS is proportional to the distance itself.

Referred to the same experiment we shall also consider a wrong trajectory hypothesis and the map pdfs that are consequently generated. In Fig. 4.8 the blue path is, like before, the true user's trajectory, used in the generation of the measurements, while the red path is the estimated one, used in the map computation. Comparing Fig. 4.8 to Fig. 4.7 we can figure out some remarkable differences: for both APs the map peaks are far away from the true AP's positions and, even more important in the view of the co-localization, they are much more widespread that in the true path case of Fig. 4.7. This pdf breadth is due to the fact that subsequent measurement likelihoods do not overlap sufficiently, causing large peaks and, as we will see, lower weights in the particle filter.

To provide a measure of the distributions' breadth we can resort to entropy, defined for a continuous variable $x \sim p(x)$ as [CT01]

$$H = -\int p(x) \log p(x) \, \mathrm{d}\, x \tag{4.14}$$

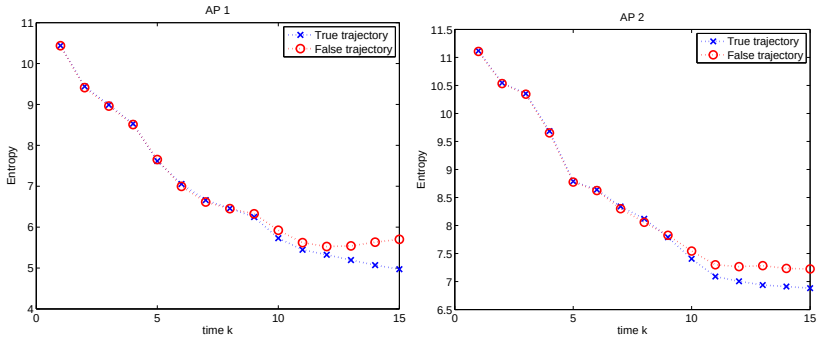
In Fig. 4.9 we compare the entropies of the AP's map pdfs generated in the example above; the entropies are computed by means of a sufficient numerical approximation. In Fig. 4.9.a we depict the entropies relative to the AP 1 mapping: the one corresponding to the true trajectory achieves lower values, i.e. narrower pdf, yet after few measurements; the same considerations are still true for



(a) k=7 (b) k=9

(c) k=11 (d) k=15

Figure 4.8 Mapping evolution in the simulative scenario reported in Fig. 4.6 for $k = 7, 9, 11, 15$ but employing a wrong hypothesis of the user's trajectory (in red); the true trajectory which is involved in the measurement generation (in blue) is the same as in Fig. 4.7.



(a) AP 1 (b) AP 2

Figure 4.9 Numerical approximation of the entropy (eq. (4.14)) of the APs' position pdf: (a) AP 1, (b) AP 2.

the AP 2 in Fig. 4.9.b. The minimization of the entropy as a result of the Bayesian algorithm can be also found in [SGB05].

4.6 Bayesian filter

We exploit the implications encoded in the DBN of Fig. 4.5 to develop our algorithm. At this aim, we factorize the posterior pdf of the state in the following way:

$$\begin{aligned} p\left(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}\right) &= p(\mathbf{y}_k^F|\mathbf{x}_k)\cdot p(\mathbf{y}_k^P|\mathbf{x}_k,\mathbf{y}_{1:k-1}^P)\cdot \\ &\quad p(\mathbf{x}_k|\mathbf{x}_{k-1})\cdot p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}), \end{aligned} \tag{4.15}$$

where the likelihood term which involves the measurements $\mathbf{y}_k^P$ associated to the partially unknown APs must be further processed to figure out the map contribution

$$\begin{aligned} &p(\mathbf{y}_k^P|\mathbf{x}_k,\mathbf{y}_{1:k-1}^P) = \\ &\int p(\mathbf{y}_k^P|\mathbf{x}_k,\boldsymbol{\theta}^P,\mathbf{y}_{1:k-1}^P)p(\boldsymbol{\theta}^P|\mathbf{x}_{0:k-1},\mathbf{y}_{1:k-1}^P)\ \mathrm{d}\ \boldsymbol{\theta}^P. \end{aligned} \tag{4.16}$$

A further marginalization involves the parametes $\mathbf{p}^P$, still not defined in our case.

4.6.1 Map and parameter set estimation

Being the APs' positions assumed mutually independent, we can factorize the mapping pdf in

$$p(\boldsymbol{\theta}^P|\mathbf{x}_{0:k}, \mathbf{y}^P_{1:k}) = \prod_j p(\boldsymbol{\theta}^P_j|\mathbf{x}_{0:k}, \mathbf{y}^P_{j,1:k}), \tag{4.17}$$

which allows us to account for one arbitrary AP j only. If the models are fully specified, i.e. void $\mathbf{p}^P_j$, the solution to the mapping problem is already provided in the Sect. 4.5, showing how the mapping pdf can be computed recursively by multiplying the likelihood when a new measurement is available and, then, normalizing it. Otherwise, a marginalization on $\mathbf{p}^P_j$ is necessary.

For simplicity, we try to reduce at our best the complexity of $\mathbf{p}^P_j$; we found that the sensibility to α_j and σ_j is not as relevant as the fact that we do not know h^P_j in advance [BMM11]. This is the reason why we assume here

$$\mathbf{p}^P_j = h^P_j$$

and assign fixed values to the other parameters. Furthermore, we model h^P_j like a discrete random variable with values in a suitable finite set $\{h_h\}_{h=1,\dots,N_H}$. So the map of the AP j can be expressed like

$$\begin{aligned} p(\boldsymbol{\theta}^P_j|\mathbf{x}_{0:k}, \mathbf{y}^P_{j,1:k}) &= \int p(\boldsymbol{\theta}^P_j, \mathbf{p}^P_j|\mathbf{x}_{0:k}, \mathbf{y}^P_{j,1:k}) \text{ d } \mathbf{p}^P_j \\ &= \sum_{j=1}^{N_H} \Pr\left\{h_h|\mathbf{x}_{0:k}, \mathbf{y}^P_{j,1:k}\right\} p\left(\boldsymbol{\theta}^P_j|h_h, \mathbf{x}_{0:k}, \mathbf{y}^P_{j,1:k}\right), \end{aligned} \tag{4.18}$$

that is a mixture of the distributions $p\left(\boldsymbol{\theta}^P_j|h_h, \mathbf{x}_{0:k}, \mathbf{y}^P_{j,1:k}\right)$ given by eq. (4.13) at different values h_h with coefficients resulting in the reference SS probabilities. Defining for simplicity

$$p_k(h_h) = \Pr\left\{h_h|\mathbf{x}_{0:k}, \mathbf{y}^P_{j,1:k}\right\},$$

we obtain the recursive computation of the $\mathbf{p}_j^P$ probabilities

$$\begin{aligned} p_k(h_h) &\propto p_{k-1}(h_h) \cdot p(y_{j,k}^P | \mathbf{x}_{0:k}, h_h) \\ &= p_{k-1}(h_h) \int p(y_{j,k}^P | \boldsymbol{\theta}_j^P, \mathbf{x}_{0:k}, h_h) p(\boldsymbol{\theta}_j^P | \mathbf{x}_{0:k}, h_h) \, \mathrm{d}\, \boldsymbol{\theta}_j^P, \end{aligned} \quad (4.19)$$

that is a mixture of pdfs as those achieved in Figg. 4.6-4.8 at different values h_h. The proportionality sign in the first line of eq. (4.19) refers to the fact that the probabilities must be normalized w.r.t. h.

4.6.2 Rao Blackwellized Particle Filter

In RBPF the particle i is composed of a hypothesis for the state trajectory and the *mapping pdf*:

$$\left\{ \mathbf{x}_{0:k}^i, p^i \left(\boldsymbol{\theta}_j^P | \mathbf{x}_{0:k}^i, \mathbf{y}_{j,1:k}^P \right) \right\}_{i=1,\ldots,N_p}.$$

The initial state vector $\mathbf{x}_0^i$ is drawn according to the prior $\mu_x(\mathbf{x}_0)$ of eq. (4.2) and a uniform distribution has been used

$$w_0^i = \frac{1}{N_p}.$$

Then, at the arbitrary time instant k, we can employ the factorization of eq. (4.15) to obtain the sampling step

$$\mathbf{x}_k^i \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}^i), \quad (4.20)$$

and the weighting step

$$w_k^i \propto w_{k-1}^i \cdot p(\mathbf{y}_k^F | \mathbf{x}_k^i) \cdot p(\mathbf{y}_k^P | \mathbf{x}_{0:k}^i, \mathbf{y}_{1:k-1}^P). \quad (4.21)$$

Some insight is required by the last innovation term in eq. (4.21), hereafter denoted for simplicity with I_k. First we can note that, in our hypotheses, it can be factorized into single APs terms (i.e. each AP provides an independent contribution to the particle weight):

$$I_k^i = \prod_j I_{j,k}^i = \prod_j p(y_{j,k}^P | \mathbf{x}_{0:k}^i, y_{j,1:k-1}^P); \quad (4.22)$$

then the $j-th$ term results, by substituting eq. (4.18) into eq. (4.16), in

$$I^i_{j,k} = \sum_h p^i_{k-1}(h_h) \int p(y^P_{j,k}|\mathbf{x}^i_k, \boldsymbol{\theta}^P_j, h_h) p^i_{k-1}(\boldsymbol{\theta}^P_j) \text{ d } \boldsymbol{\theta}^P_j \quad (4.23)$$

Finally, resampling can be performed according to a heuristic criterium, as described e.g. in [AMGC02].

4.6.3 Summary of the algorithm

The algorithm is now summarized in detail.

Algorithm 5 (RBPF Co-localization)**.**

For $k = 0$ and every particle i

- draw $\mathbf{x}^i_0 \sim \mu_x(\mathbf{x}_0)$;
- set $p_0(h_h) = \mu_p(h_h)$.

For $k = 1, 2, \ldots$, compute:

- For any $i = 1, \ldots, N_p$
  - Draw $\mathbf{x}^i_k \sim p(\mathbf{x}_k|\mathbf{x}^i_{k-1})$;
  - Compute I^i_k as in eq. (4.22), with the factors $I^i_{j,k}$ given by eq. (4.23);
  - Assign a weight to the particle as in eq. (4.21);
  - Update the h_h probabilities according to eq. (4.19) and, thus, update the map pdf as in eq. (4.18);
- Normalize the weights such that their sum is 1;
- Resample if required.

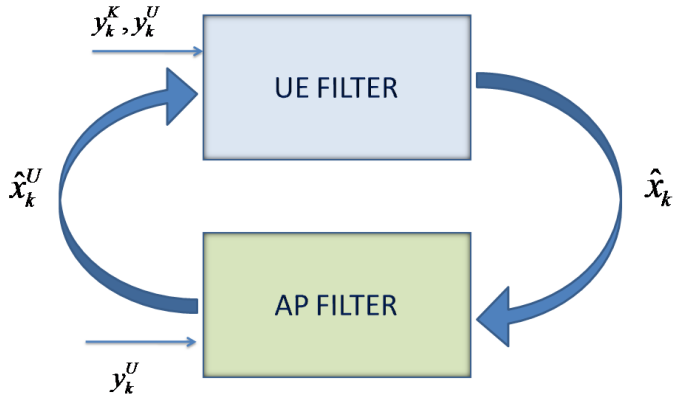


Figure 4.10 Dual estimation scheme for the algorithms proposed.

4.7 An approximation based on concurring filters

Alg. 5 may be prohibitive for computational issues, above all with several APs to care. An alternative approach is based on the following marginalization of the posterior pdf at k, i.e.

$$p(\mathbf{x}_k, \boldsymbol{\theta}_{AP}^P | \mathbf{y}_{1:k}) = p\left(\mathbf{x}_k | \boldsymbol{\theta}_{AP}^P, \mathbf{y}_{1:k}\right) p\left(\boldsymbol{\theta}_{AP}^P | \mathbf{y}_{1:k}\right). \tag{4.24}$$

We choose to maximize the two terms on the right-end side of eq. (4.24) in a decoupled way through a dual estimation algorithm. Such scheme is sketched in the block diagram of Fig. 4.10 and summarized in the following pseudocode:

Algorithm 6 (Dual Filter Co-localization)**.**

- Initialization
  - 1. $[\hat{\mathbf{x}}_1]$ =UEfilter$\left(\mathbf{y}_1^F, [\,], \hat{\boldsymbol{\theta}}_{AP}^F, [\,]\right)$;

– 2. $[\hat{\boldsymbol{\theta}}_1^P]$ =APfilter$(\hat{\mathbf{x}}_1, [\,], \mathbf{y}_1^P)$;

- For $k = 2, 3, \ldots$

  – 3. $[\hat{\mathbf{x}}_k]$ =UEfilter$(\mathbf{y}_k^K, \mathbf{y}_k^P, \boldsymbol{\theta}_{AP}^F, \hat{\boldsymbol{\theta}}_{k-1}^P)$;

  – 4. $[\hat{\boldsymbol{\theta}}_k^P]$ =APfilter$(\hat{\mathbf{x}}_k, \hat{\boldsymbol{\theta}}_{k-1}^P, \mathbf{y}_k^P)$;

- End

This means that initially the first UE's position $\mathbf{x}_1$ is inferred only by employing fully known APs' measurements $\mathbf{y}_1^F$, and an initial estimation $\hat{\boldsymbol{\theta}}_1^{P\,1}$ of $\boldsymbol{\theta}_{AP}^P$ is then carried out; indeed in this step also previous partially known APs' estimates can play a role. For $k > 1$:

- the UE filter computes the UE's position estimate $\hat{\mathbf{x}}_k$, given the measurements $\mathbf{y}_k^P$ and the partially known APs' locations estimates $\hat{\boldsymbol{\theta}}_{k-1}^P$, available at the end of the previous step;

- the AP filter uses the last $\hat{\mathbf{x}}_k$ and the previous $\hat{\boldsymbol{\theta}}_{k-1}^P$ to get a new estimation $\hat{\boldsymbol{\theta}}_k^P$.

We now propose a solution for the UE filter, and two different implementations of the AP filter.

[1]The subscript AP is omitted for simplicity.

4.7.1 UE filter

The UE filter has to maximize the first term on the right end side of eq. (4.24). Since we use fingerprinting we only process present measurements, yielding

$$\hat{\mathbf{x}}_k = \arg\max_{\mathbf{x}_k} p\left(\mathbf{x}_k|\boldsymbol{\theta}^P, \mathbf{y}_k\right). \tag{4.25}$$

According to our approach, $\boldsymbol{\theta}^P$ in (4.25) is substituted by its estimate $\hat{\boldsymbol{\theta}}^P_{k-1}$ available at the end of the previous step

$$p\left(\mathbf{x}_k|\boldsymbol{\theta}^P, \mathbf{y}_k\right) \approx p\left(\mathbf{x}_k|\hat{\boldsymbol{\theta}}^P_{k-1}, \mathbf{y}_k\right). \tag{4.26}$$

In this case, we can assume a uniform prior pdf of $\mathbf{x}_k$ and apply the Bayes rule

$$\hat{\mathbf{x}}_k = \arg\max_{\mathbf{x}_k} p\left(\mathbf{x}_k|\mathbf{y}_k, \hat{\boldsymbol{\theta}}^P_{k-1}\right) = \arg\max_{\mathbf{x}_k} p\left(\mathbf{y}_k|\mathbf{x}_k, \hat{\boldsymbol{\theta}}^P_{k-1}\right). \tag{4.27}$$

RADAR is the most widespread and simple way to solve eq. (4.27) without assigning an explicit propagation model to RSSs and we will use it.

4.7.2 AP filter

The positions $\boldsymbol{\theta}^P$ of the partially known APs should be estimated from the measurements $\mathbf{y}_{1:k}$ by maximizing the second term in the right end side of eq. (4.24)

$$\hat{\boldsymbol{\theta}}^P = \arg\max_{\boldsymbol{\theta}^P} p\left(\boldsymbol{\theta}^P|\mathbf{y}_{1:k}\right). \tag{4.28}$$

The RSSs measurements only provide distance information between the related AP and the UE, so in principle the whole UE's trajectory till k must be used in evaluating eq. (4.28), yielding

$$\hat{\boldsymbol{\theta}}^P_k = \arg\max_{\boldsymbol{\theta}^P} p\left(\boldsymbol{\theta}^P|\hat{\mathbf{x}}_{1:k}, \mathbf{y}_{1:k}\right). \tag{4.29}$$

Clearly, the partially known APs' locations are mutually independent, and thus from now on we are going to consider just one of them.

We briefly describe two algorithms, based on different models for the partially known AP's position and then we figure out their performances. Both exploit a sequential Bayesian approach that permits to refine the estimate at every step. This method turns out to be particularly useful not only when the positioning is performed at the UE side, but also when, in a server based localization architecture, few users with roughly initial position estimates do not allow for the localization of unknown APs by proximity or triangulation techniques.

Discrete state

In this approach we assume that the APs can be located in a discrete set of points in the area of interest (for example we could use the same grid as in the radio map, but this is not mandatory). The partially known AP's trajectory is trivially represented by one position repeated in time, thanks to their stationarity.

Let us model this trajectory as the underlying state of a Hidden Markov Model, whose data are the RSS measurements. Technically, we set a unitary transition matrix for the hidden state $\boldsymbol{\theta}_{AP}^{P}$ (dealing with stationarity), and a non-informative uniform prior distribution.

Moreover, each RSS (expressed in dBm) is assumed both to be conditional independent of the others and to cope with a Gaussian distribution, whose mean is related to the physical distance between user and AP and whose variance must be set. Further details about the parameters of the proposed RSS likelihood function will be given in Sect. 4.8, along with a simple propagation model.

The well known Viterbi Algorithm (VA) can be applied straightforwardly in this context for limiting the computational complexity when the optimal maximum a posteriori probability estimate is sought. Following the analysis in [sti00], in our case it results

in cumulating the RSS log-likelihood given each test point and in choosing, instant by instant, the trajectory with the maximum 'weight'.

Continuous state

If a continuous model for the AP's position is given, PFs are easily applicable. In this case the AP's kinematics has to include a fictitious random model, e.g. a white Gaussian noise $\mathbf{r}_k$ with zero mean and given covariance matrix Q_r [Nel00]

$$\boldsymbol{\theta}_k^P = A\,\boldsymbol{\theta}_{k-1}^P + \mathbf{r}_k, \tag{4.30}$$

where A is the unitary matrix. Now, given the UE estimation $\hat{\mathbf{x}}_k$, the SIR algorithm can be applied with a suitable number of particles [AMGC02]. This technique promises better performances than the former approach, but is computationally heavier, and this can be critical for both energy constrained UEs and real-time processing. Nonetheless, a server based location server, if available, overcomes this problem.

4.7.3 Complexity & Scalability

The UE filter has RADAR's pros and cons: quick in run time, but filling the database can be really heavy. Moreover the system is easily scalable with new users but requires a RM for each new known AP. As for the AP filter, VA needs basically the sampling of the RSS likelihood functions on the chosen grid for any APs; therefore even if its complexity depends upon the grid density, the overall burden is very low. In PF, instead, the complexity is linearly related to the amount of particles times the number of detected APs. In the practice this latter approach turns out to be quite heavier in both memory storage and CPU activity.

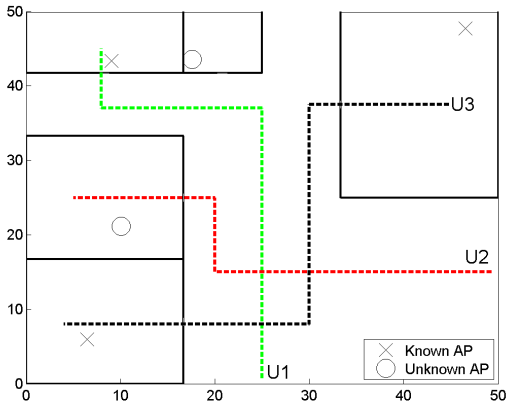


Figure 4.11 Simulative testbed.

4.8 Simulation results

The scenario in Fig. 4.11 has been adopted to test our approximated algorithms. It mimics a building floor 50×50 m sized with 3 fully known APs (denoted by 'X'), 2 unknown APs ('O'), and 3 users who walk in it. A server based architecture is supposed to be in force; the location server has at its disposal the RSS measurements of all the users in real time, and can use them to estimate the unknown APs' positions in a joint way. Later, in the section with the experimental results, we will show a self-positioning scheme that is even more challenging.

The RM has been built up on a 2 m spaced grid by using a theoretical propagation model in which the power decays with the distance d in accordance with the WAF model, here restated for clarity

$$P(d) = P(d_0) - 10\alpha \log_{10}\left(\frac{d}{d_0}\right) - nW \times WAF, \qquad (4.31)$$

where

- d_0 is a reference distance (the APs are supposed to be raised of d_0 from the users' plane to avoid singularities);

- α is the decaying exponent, $\alpha = 2$ in free space, $\alpha = 1 \div 4$ indoors;
- nW is the number of walls between UE and AP and WAF is the attenuation factor due to each wall.

In our simulations, we set

$$d_0 = 1.6\,\text{m},\ P(d_0) = -34.22\,\text{dBm},\ \alpha = 2,\ WAF = 3\,\text{dBm}.$$

The fading model is Lognormal, i.e. Gaussian in dBm with unitary variance.

In Fig. 4.12 the empirical cumulative probability functions (ECDF) for the localization error (in terms of RMSE) are depicted for the proposed algorithms; here 100 particles are used for the PF. Each plot refers to one of the users and shows the ECDF obtained with four implementations:

- in blue continuous line, the basic system (BS) only based on the exploitation of known APs (it is a lower bound for the performances);
- in red dotted line, a clairvoyant (CV) system that relies also on unknown APs' true positions (it is an upper bound);
- in black line with circles, the system based on VA that exploits RSS from all APs;
- in green line with squares, the PF based system.

In the legend we also report empirical mean and standard deviation for all distributions. In all cases the proposed algorithms perform better than the basic system, with PF achieving performances very close to the CV system. We recall, however, that PF is heavier than VA in terms of hardware resources requirements. The estimation error for the unknown APs' positions is plotted as a function of time in Fig. 4.13; it is also visible that VA suffers a discretization effect, not present in PF.

In PF we can increase the number N of particles to improve the performances. This has been verified in the previous scenario and the results are depicted in Fig. 4.14 that, however, shows only a light effect as N is tuned in the range $100 \div 500$.

4.9 Experimental results

The experimental scenario, depicted in Fig. 4.15, is a 1-floor 11×17 m sized apartment, with 4 APs. The measurements are collected by means of a laptop equipped with an external device ORINOCO 802.11 b/g compliant. A free software is used to collect approximately 1 RSS per second for any detected APs. All computations are made off line on the measuring laptop. For simplicity, no attention has been paid to the orientation of the antennas, or to the measurements resolution of 1 dBm, despite the fact that a suitable modeling of these effects can improve the performances of the localization algorithm.

We tuned the parameters in the WAF model by means of a radio map, consisting of 100 measurements per point times 43 points (the intersections of the lines within the grid, less the points marked by a cross), and the entire procedure took about two hours. The measurements in the localization phase were collected in the positions denoted by red points in Fig. 4.15. Real data differ from theoretical models for several reasons. First, the adopted path loss model is very simple and does not consider the obstacles. Similarly, the Lognormal model for RSS measurements is quite reliable, but cannot account for all the effects causing noise. Nevertheless, the experimental results obtained with these simple models fit our expectations well.

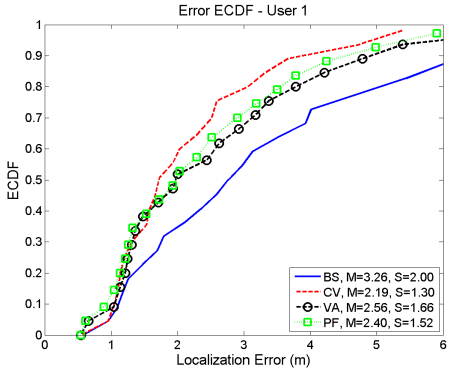
Fig. 4.16 shows the results of the proposed algorithms in the experimental testbed of Fig. 4.15, where AP 1 is supposed to be an unknown AP, while the other APs are known. In the top plot (a) the ECDF of the localization error is displayed, showing good performances for the proposed algorithms, while the bottom plot (b) reports the error in unknown AP estimation against time. No-

ticeably, PF is able to achieve a reliable estimate of the AP's position quickly and thus allows a meaningful improvement in the user localization. The covariance matrix Q_r of the fictitious noise $\mathbf{r}_k$ is supposed to be diagonal with equal entries, whose square roots are $\sigma_r = 0.05$. The price paid in the real case is that a greater number of particles is necessary; in our setup, for example, 1000 particles are needed to achieve good results. The VA algorithm, despite achieving a worse estimation of the AP, also performs well.

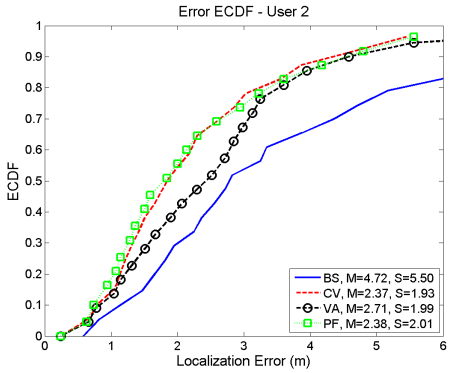
4.10 Concluding remarks

Colocalization provides the interesting opportunity to improve positioning accuracy in dense scenarios. We have set up a Bayesian framework providing the exact solution and a good approximation. Simulative and experimental results have confirmed the convergence of our algorithms.

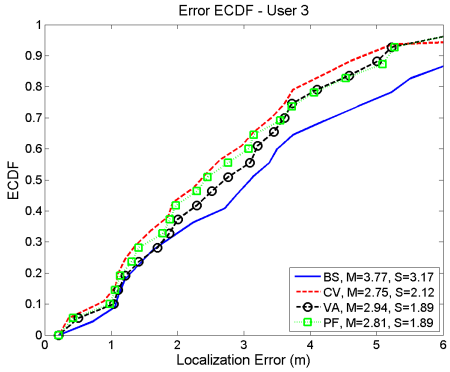
Strong similarities hold between colocalization and SLAM, that is the next chapter's content. We have preferred, however, to keep them separated since they refer to different goals: in colocalization we only deal with signals from unmanaged WLANs to improve the positioning accuracy and we determine the AP's position (in probability) to help next users but we do not aim to estimate the building map at yet.



(a) User 1



(b) User 2



(c) User 3

Figure 4.12 Localization error for all the users and described algorithms, referred to the simulative testbed in Fig. 4.11. For each curve empirical mean (M) and standard deviation (S) are also reported.

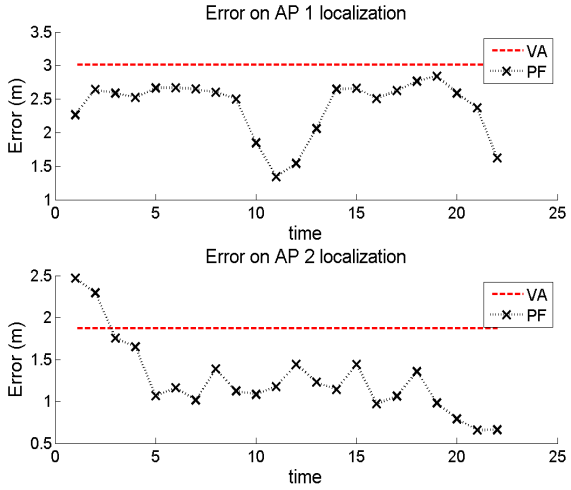


Figure 4.13 Unknown APs' position RMSE in meters for both algorithms, referred to the simulative testbed of Fig. 4.11.

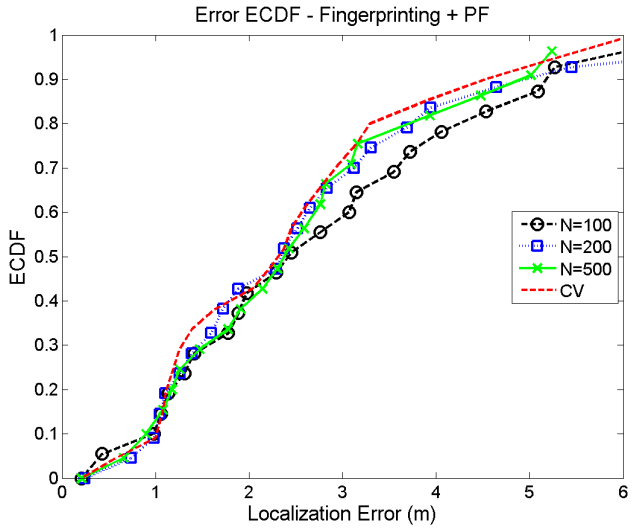


Figure 4.14 ECDF of the localization error for one UE in the simulative testbed in Fig. 4.11; the UE filter implements RADAR and the AP filter is based on PF with different amounts of particles. For comparison the curve referring to the clairvoyant (CV) case is also shown.

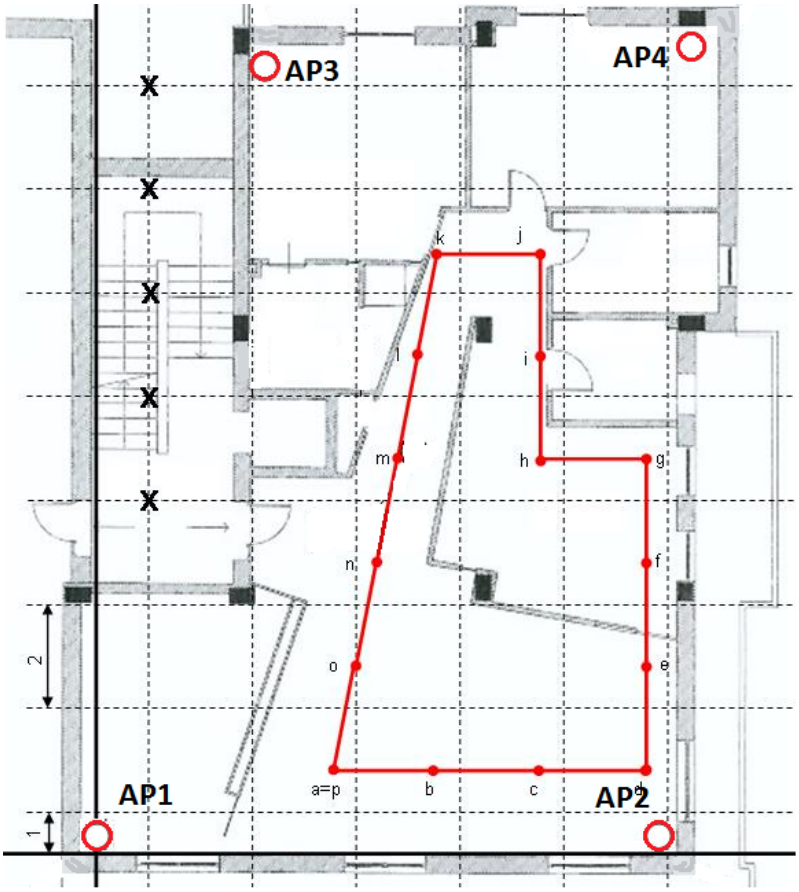
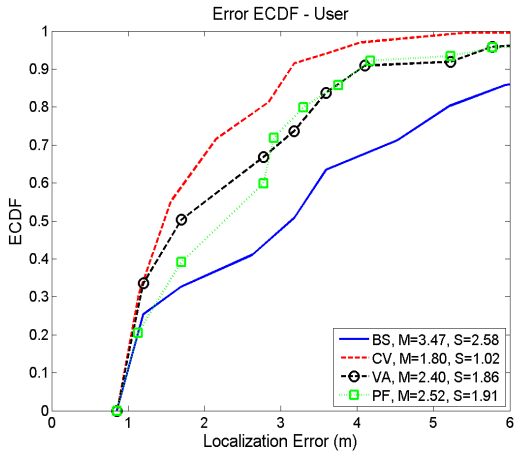
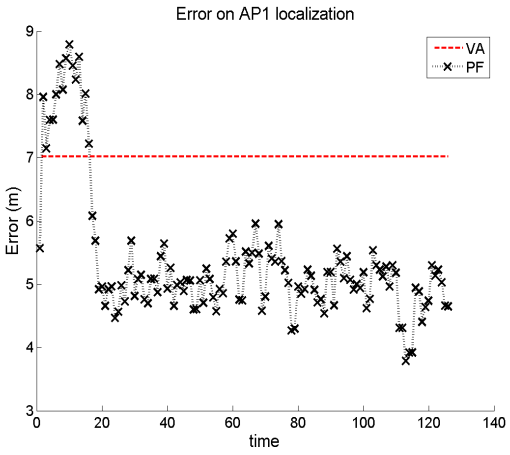


Figure 4.15 Experimental testbed.



(a) Empirical CDF



(b) AP estimation error

Figure 4.16 ECDF (a) of the localization error for one user and all described algorithms (PF employs 1000 particles), referred to the experimental testbed in Fig. 4.15 with *AP* 1 supposed unknown; the UE filter implements RADAR. For each curve mean (M) and standard deviation (S) are also reported. In (b) the RMSE for the unknown AP's position is reported vs time.

Chapter 5

WLAN aided Simultaneous Localization And Mapping

While driving a car with a GNSS navigator on our side, we can typically see our position overlapped to a map of the region on the screen. The aim is not to improve the localization algorithm, but rather to provide a guidance to the user in a more intuitive way than just the absolute coordinates.

Indoors the situation is very different, since the knowledge of the map in the estimation process has a striking effect on the performances. This is valid not only in the RSS-based localization but in all cases of indoor tracking, how it is well known in the literature.

In a more abstract way, the point is that the map represents a set of constraints, typically very tight, on both the user's true trajectory and the estimator. This is equivalent, of course, to bound the error, leading to a more accurate estimation and, unfortunately in some contexts, to a biased estimator, also when the unconstrained one was unbiased.

Employing the map in the algorithm design is, therefore, a precious tool which helps lowering the error but, on the contrary, there is a critical drawback: the map is often not available at all, at least with a sufficient degree of accuracy. One sometimes could depict it manually with a quite big effort, but it would be much more efficient and elegant if we could retrieve it by processing the same data employed for the user's localization.

This paradigm is known in the literature as Simultaneous Localization And Mapping (SLAM). Altough it was first defined within the robotics community in the mid '80s, an increasing interest is being paid over this last years to SLAM for pedestrians. Even if, at a first sight, one could think that SLAM is always the same for robots as well as for pedestrians, it will be soon clear that the frameworks are totally different, so that independent approaches and, thus, solutions are needed.

5.1 SLAM for robots

Robots can use a large gamma of sensors, ranging from inertial units to cameras and lasers. Furthermore, some robots can easily exchange some selected information and measure their mutual distance, as proposed and analized in many papers, e.g. in [MZZ+09, GABS10]. Several experiments employing heterogeneous robots were carried out in the past. Here the key point is how to control the robot movement to avoid obstacles and explore the whole environment, while the sensing part is less constrained, due to the large number of sensors used.

5.2 SLAM for pedestrians

There are two main differences between SLAM for robots and for human users: the type of information to exploit and the movement models to adopt. We shall now examine the former difference, while the movement models are beyond the aims of this chapter.

A pedestrian cannot provide his own intention ('what he wants to do') or sight ('what he sees') to a processing unit. Moreover, such invasive sensors like cameras or lasers are not likely to be used by common people walking somewhere in a building. We think that three kinds of information can be mostly used:

- step measurements
- proximity measurements,
- distance measurements.

We will now briefly discuss these types of measurements and give some examples about how they can be collected in the practice. Their fusion in a single framework will be then discussed later throughout this chapter.

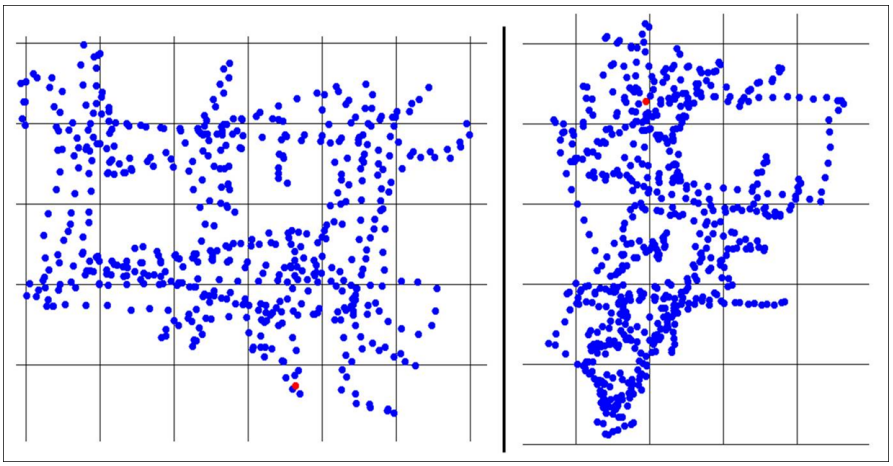


Figure 5.1 Maps retrieved by processing IMUs' data with the ZUPT algorithm [Fox05].

5.2.1 Step measurements

Quite recent work has shown remarkable advances in the area of indoor positioning with low cost inertial sensors worn by pedestrians. The work of Foxlin on foot mounted Inertial Measurement Units (IMUs) [Fox05] has shown how zero velocity updates - ZUPTs- during the rest phase of a pedestrians foot can be used to mitigate the problem of non-linear error growth in inertial integration over time. Nevertheless, errors still accrue over time, especially the heading error which is only weakly observable from the zero velocity update. Fig. 5.1 shows the data from the IMU attached to a shoe processed in a typical Inertial Navigation System (INS) using ZUPTs: the drifts (particularly in heading) lead to large divergence from the true path. Three groups independently showed that this problem can be addressed by employing known building layouts [KR08, BWK08, WH08]. All used PF methods where the building layout information is used to constrain particle movement to within the areas accessible to a pedestrian. As a result, long term error stability can be achieved when the map is sufficiently accurate and the layout sufficiently constrains the motion.

5.2.2 Proximity measurements

Proximity information is being used within indoor localization since the first contributions in this field. It indicates the possibility to sense that the user is getting close to a known *place*. There are many practical ways to do this:

- the user can manually report the place when he sees it;
- automatic proximity sensors deployed in the environment, e.g. RFID tags, infrared sensors, magnetic sensors, etc...

A further issue concerns whether the places can be uniquely identified or only a binary information is available; in the latter case the user detects a place, but does not know which one. Deeper considerations are provided later in Sect. 5.5.

5.2.3 Distance measurements

Distance measurements from some reference points are much more accurate than simple proximity detections, but more difficult to obtain for pedestrians, since lasers and cameras are not exploitable. One of the most popular choices in indoor environments is provided by RSS measurements from WLANs that are related to the distance from the APs, as seen in the previous chapters. Of course, since we are interested in probabilistic approaches, models for both propagation channel and RSS likelihood are needed.

To make our line of thought clear, we aim at developing and testing a probabilistic framework in which those heterogeneous measurements are fused together and used for SLAM. Particular attention, of course, will be paid to all the aspects about the integration of RSS measurements in a wider framework, partly made available by previous works.

Before dealing with that, a review of the existing literature is proposed.

5.2.4 Literature review

SLAM for pedestrian is quite a recent topic. To the best of our knowledge there are two main approaches, both converging to our final framework:

- SLAM based on Wlan measurements;
- SLAM based on step measurements provided by IMUs.

Both approaches bring, in the end, to the fusion of these two types of measurements but, interestingly, with different solutions. Moreover, also proximity information can be exploited.

We now present the main results achieved in pedestrian SLAM, starting from single sensor frameworks; afterwards multisensor extensions will be analyzed. Mostly, we will focus deep on the combined IMU based SLAM, and new advances will be proposed concerning the integration with WLAN measurements.

5.3 WiFi based SLAM

Ferris et al. in [FFL07] propose a technique for solving the WiFi SLAM problem using the Gaussian Process Latent Variable Model (GPLVM) to determine the latent-space locations of signal strength data from unknown APs. They show that GPLVM, in combination with an appropriate motion dynamics model, can be used to reconstruct a topological connectivity graph from a signal strength sequence which, in combination with the learned Gaussian Process signal strength model, can be used in SLAM.

The work has severe limits: the motion model relies on very simplistic assumptions and consequently the results, beside showing physical consistency, require experimental setups with very simple structures (hallways and only 90 degrees turns). Nevertheless it is the base for a more recent work in which the fusion with odometry data, which basically provide a motion model that is much more accurate, heavily increases the performances [HMQ+11]; moreover, connections with the colocalization technique in [PPY+12] are also noted.

5.4 IMU's based SLAM

Over the last few years several research groups have dealt with the stabilization of the IMU's based indoor navigation: to this end, the residual cumulative error of the resulting odometry in heading over time could be mitigated by using map information [KR08, BWK08, WH08]. Of course, we are more interested in estimating the map along with the user's trajectory, that is the case of FootSLAM algorithm, developed in 2009. It is based on a probabilistic framework in which particle filters and suitable approximations are used to implement the Bayesian solution.

FootSLAM [RAK09] uses a Bayesian estimation approach, where the state is the user's pose $\mathbf{P}$ (position and heading) and step measurements allow to update both the user's trajectory and the environment map over time. The implementation employs the RBPF, where each particle is composed of both a user trajectory instance and its related map. This latter is obtained by partitioning the environment into hexagonal cells and estimating all the transitions probabilities for each visited cell. Extensive experiments show that convergence of both mapping and localization occurs when the user walks on closed loops and enough particles are used. The fusion of several datasets (Collaborative FootSLAM) is also dealt with in [RGPA11].

We now analyze and discuss the main points of the algorithm, together with some interesting results and limitations. Also PlaceSLAM will be acquainted as the first example of data fusion in the FootSLAM framework. Finally, the integration of RSS measurements will be handled, leading to the algorithm WiSLAM, that is the main original contribution of this chapter.

5.4.1 Dynamic Bayesian Network

The key concepts involved in the FootSLAM derivation are summarized in the DBN, of which the Fig. 5.2 represents two time slices. The pose $\mathbf{P}$ is a set of user's features, in this case his po-

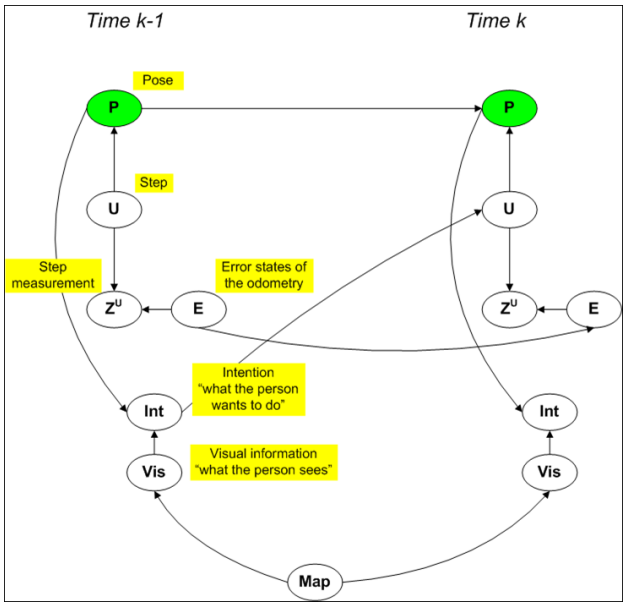


Figure 5.2 Two time slices of the Dynamic Bayesian Network for FootSLAM framework. The full description is reported in the text.

sition in 2 dimensions and heading[1]; it evolves according to the intention **Int** of the pedestrian and his current sight **Vis**. These are both strongly influenced by the floor map **M** and can be neither measured nor processed, but are included in the model for clarity. The **M** itself is of course not known in advance but will be estimated on the fly.

The step **U** is a vector that links deterministically two subsequent poses, viz.

$$\mathbf{P}_k = \mathbf{U}_k + \mathbf{P}_{k-1} \tag{5.1}$$

and is the variable that can be measured at each step. The measurement $\mathbf{Z}^U$ is corrupted by a correlated noise encoded in the variable **E**. Note that the IMU makes inertial measurements (distance and angle) with a rate of about 100-200 Hz and some preliminary processing is then performed to obtain the $\mathbf{Z}^U$ (whose frequence is down to 0.5-1 Hz), as described in Sect. 2.7.2 and

[1]No conceptual difference should show up when dealing with 3D scenarios: to our knowledge such extension is currently in progress.

therein references. Nevertheless, it is common to consider $\mathbf{Z}^U$ as a direct measurement and this is what we will do since here.

Finally we can note that the **MAP** is assumed to be static: in this case it appears obvious, but when applied to a more generic concept of map it is less straightforward and constitutes an assumption.

5.4.2 Bayesian formulation

In the Bayesian formulation the joint posterior pdf of both the state variables and the map at instant k given the step measurements is

$$p(\mathbf{P}_{0:k}\mathbf{U}_{0:k}\mathbf{E}_{0:k}, \mathbf{M}|\mathbf{Z}^U_{1:k}). \tag{5.2}$$

By exploiting the relationships implied in the DBN of Fig. 5.2, it factorizes into

$$p(\mathbf{M}|\mathbf{P}_{0:k})\cdot p(\mathbf{P}_{0:k}\mathbf{U}_{0:k}\mathbf{E}_{0:k}|\mathbf{Z}^U_{1:k}), \tag{5.3}$$

which splits localization from mapping. Note that in the right term the influence of the map is implicit: thanks to the knowledge of the map, estimated from the measurements, better accuracy (and stability) is achievable in the user's localization.

Let us focus on the last term of eq. (5.3). It allows a recursive formulation based on the independence relationships encoded in the DBN:

$$\begin{aligned}
&p\left(\{\mathbf{PUE}\}_{0:k}|\mathbf{Z}^U_{1:k}\right)\\
\propto&p\left(\mathbf{Z}^U_k|\{\mathbf{PUE}\}_{0:k}, \mathbf{Z}^U_{1:k-1}\right)\cdot p\left(\{\mathbf{PUE}\}_k|\{\mathbf{PUE}\}_{0:k-1}, \mathbf{Z}^U_{1:k-1}\right)\\
&\cdot p\left(\{\mathbf{PUE}\}_{0:k-1}|\mathbf{Z}^U_{1:k-1}\right)\\
=&p\left(\mathbf{Z}^U_k|\{\mathbf{UE}\}_k\right)\cdot p\left(\{\mathbf{E}_k|\mathbf{E}_{k-1}\}\right)\cdot p\left(\{\mathbf{PU}\}_k|\{\mathbf{PU}\}_{0:k-1}\right)\\
&\cdot p\left(\{\mathbf{PUE}\}_{0:k-1}|\mathbf{Z}^U_{1:k-1}\right).
\end{aligned} \tag{5.4}$$

The term in the first line of eq. (5.4) represents the likelihood of the k-th step measurement given the state at the same instant (the correlation with the past is encoded into $\mathbf{E}_k$), while the terms

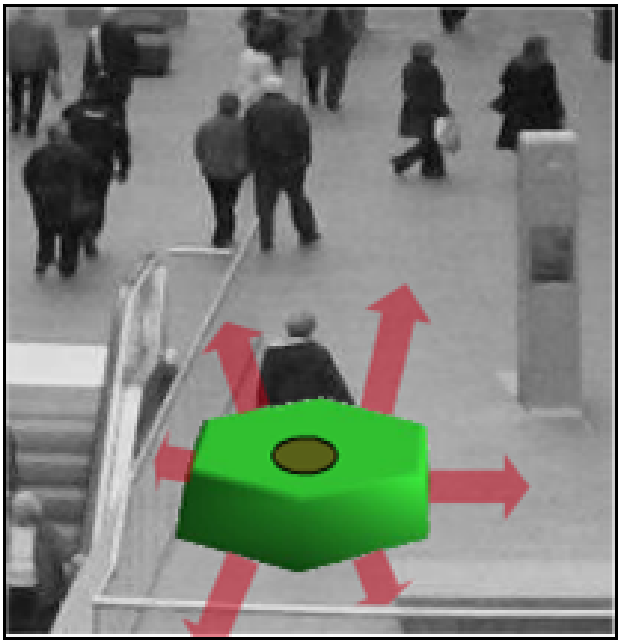


Figure 5.3 Hexagon based map.

on the second line are the state transition at time k. In detail the map plays a role into the right end term of the second line, providing prior information for the new user's pose $\mathbf{P}_k$. This can be expressed from a mathematical point of view by marginalizing over $\mathbf{M}$

$$\begin{aligned} I_M &\hat{=} p\left(\{\mathbf{PU}\}_k \,|\, \{\mathbf{PU}\}_{0:k-1}\right) \\ &= \int_{\mathbf{M}} p\left(\{\mathbf{PU}\}_k \,|\mathbf{P}_{k-1}, \mathbf{M}\right) p\left(\mathbf{M}|\mathbf{P}_{0:k-1}\right)\, d\mathbf{M} \end{aligned} \tag{5.5}$$

It is obvious that the solution to integral eq. (5.5) needs a formal definition for $\mathbf{M}$. Indeed the whole algorithm must be approximated in a suitable way.

5.4.3 Probabilistic map

The area of the floor is discretized in a set of hexagons (this shape has been chosen because they can fill the area without overlap-

pings, as for the base stations coverage in mobile communications) and the map is the set of transition probabilities between couples of adjacent hexagons.

The important concept is that here the map is meant as a 'probabilistic map', as for the colocalization in Ch. 4. Therefore, to estimate a map means to estimate all of the transition probabilities, i.e. the probability of crossing at k the edge e between the arbitrary hexagon H_h and H_j (see Fig. 5.3)

$$\mathbf{M}^e_{h(\mathbf{P}_{k-1})} = P\left(\mathbf{P}_k \in H_j | \mathbf{P}_{k-1} \in H_h\right) \tag{5.6}$$

for any h and j. Moreover, not adjacent hexagons are assumed mutually independent: this is a good approximation if the radius of the hexagon is large enough (0.5 m is the typical value reported in the experiments) and it also prevents from multiple crossings in one step.

5.4.4 Filter implementation

The Bayesian filter for FootSLAM is implemented by means of the RBPF (see Sect. 1.3.8), in which a hypothesis for the state $\{\mathbf{P}, \mathbf{U}, \mathbf{E}\}_{0:k}$ is drawn for all particles, based on a 'likelihood PF'. The Rao-Blackwellization is chosen to reduce the state space to sample, so that the particle number necessary for the convergence of the PF to the Bayesian filter is minimized. In detail, the sampling is performed in three steps; for the arbitrary particle i we

- sample $\mathbf{E}^i_k$ from $p\left(\mathbf{E}_k | \mathbf{E}^i_{k-1}\right)$
- sample $\mathbf{U}^i_k$ from $p\left(\mathbf{U}_k | \mathbf{Z}^U_k, \mathbf{E}^i_k\right)$
- compute $\mathbf{P}^i_k = \mathbf{P}^i_{k-1} + \mathbf{U}^i_k$.

Then, basing on the state hypothesis the map $p\left(\mathbf{M} | \mathbf{P}^i_{0:k}\right)$ is updated in the way described in [RAK09]; the transition probabilities are estimated with a frequentist approach, in terms of the number of times, say N^e_h, that the edge e linking the hexagon h to a neighbor is crossed by the user in either way.

In the end, each particle encompasses

$$\left\{\mathbf{P}_k^i, \mathbf{U}_k^i, \mathbf{E}_k^i, p\left(\mathbf{M}|\mathbf{P}_{0:k}^i\right)\right\}. \tag{5.7}$$

The final step of the algorithm lays in the weighting step

$$w_k^i \propto w_{k-1}^i \cdot I_M^i, \tag{5.8}$$

where I_M^i is the integral in eq. (5.5) evaluated for the particle hypothesis of both the state and the map. The computation is done by choosing a beta distribution for $p\left(\mathbf{M}_h^e|\mathbf{P}_{0:k}^i\right)$, that also allows for the introduction of prior information

$$I_M^i \propto \frac{N_h^e + \alpha_h^e}{N_h + \alpha_h}, \tag{5.9}$$

where $N_h = \sum_e N_h^e$, α_h^e is a prior count assigned to the edge e and $\alpha_h = \sum_e \alpha_h^e$.

5.4.5 Summary of the algorithm

FootSLAM is now briefly summarized.

Algorithm 7 (FootSLAM)**.**

- Initialize all N_P particles to $\mathbf{P}_0^i = (x, y, h = 0)$ where x, y and h denote the pose location and heading in two dimensions; draw $\mathbf{E}_0^i$ from a suitable initial distribution for the odometry error state.

For each time step increment k:

- Draw U_k^i, E_k^i from the proposal density

  $$p\left(\mathbf{U}_k|\mathbf{Z}_k^U, \mathbf{E}_k^i\right) \cdot p\left(\mathbf{E}_k|\mathbf{E}_{k-1}^i\right)$$

  and compute $\mathbf{P}_k^i$ by adding the vector $\mathbf{U}_k^i$ to $\mathbf{P}_{k-1}^i$.

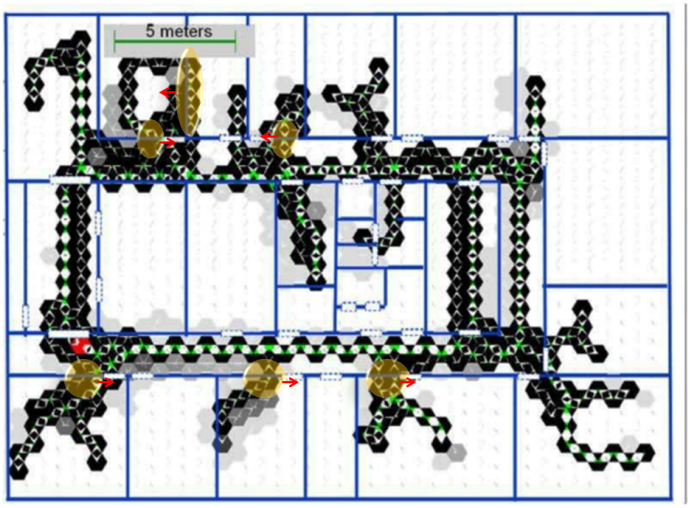
- Compute the new weights w_k^i by normalizing on i the products (5.8), where I_M^i is given by eq. (5.9) and relies on the counts until $k-1$.
- Update the map, adding the new count for all particles.
- Resample if necessary.

5.4.6 Real data experiments

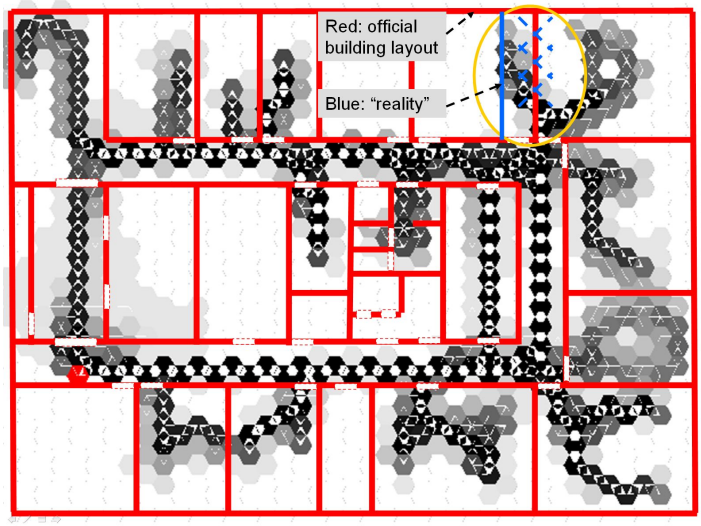
The algorithm is shown to recover the heading error cumulation and to provide a good quality map of the floor, with the consequence that the user is well tracked. Experiments are relative to 10 minutes walks around an office layout with the user wearing a foot mounted IMU. All the processings were made off line, due to the high number of particles implied by the complexity of the state space. In fact in all experiments at least 10000 particles were necessary for esuring the convergence.

In Fig. 5.4.a one such example of map is given. The darkness of the hexagons represents the density of particles crossing it (darker stays for more particles), with the main errors being highlighted by orange ovals and the arrows used as a rough graphic version of the error vectors. The ground truth is overlapped to perceive the quality of the estimated map. The main errors are inherent in the door crossings and do not exceed a couple of meters in the worst case, while better accuracy is shown in the corridors that are frequented more often. This difference is not casual: the frequentist approach used in the map estimation leads to the need for loop closure. More insight about this concept and its consequences will be provided later, together with some possible solutions.

In Fig.5.4.b the map from another experiment is shown to highlight the potentialities of FootSLAM. Here the resulting map is used to even correct inaccuracies in the ground truth: in fact the



(a)



(b)

Figure 5.4 Maps estimated by FootSLAM: in (a) the main errors, altough of scarce importance, are highlighted, in (b) the algorithm can help correcting errors in the ground truth.

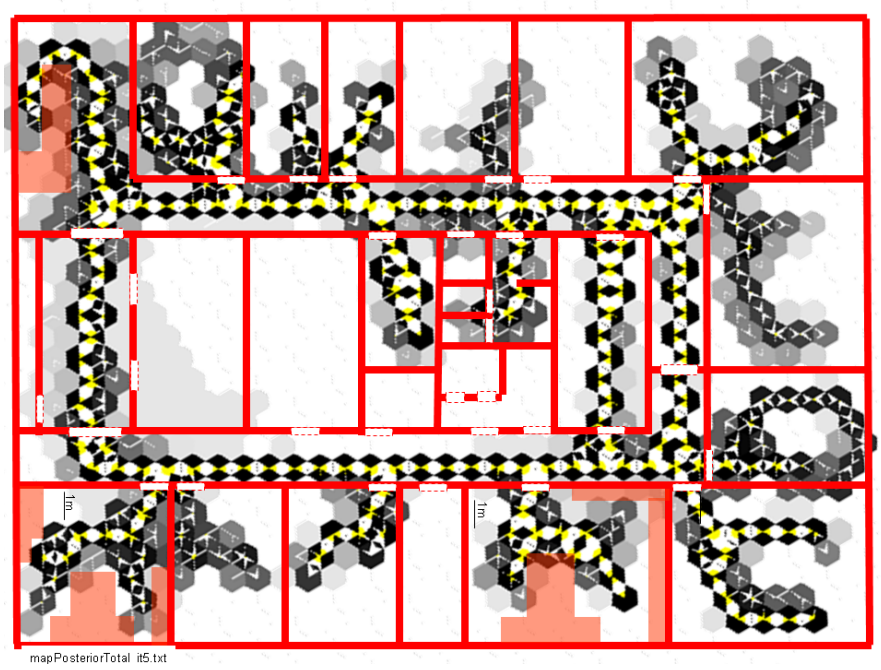


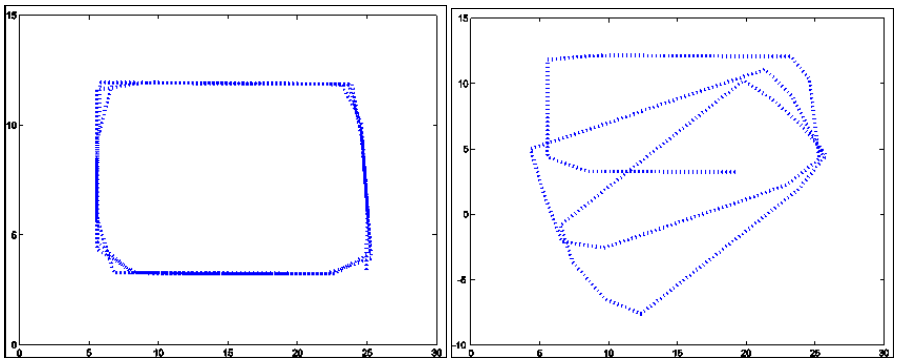
Figure 5.5 Map processed by FeetSLAM, derived from 4 independent maps obtained through FootSLAM. The elaboration is offline, and encompasses only the best particle.

room in the up-right corner of the ground truth has been enlarged over time and the blue line shows the actual position of the wall.

5.4.7 Collaborative FootSLAM: FeetSLAM

An extension of FootSLAM in a multiuser scenario is represented by FeetSLAM [RGPA11], where cooperation between more users is used to improve tha map accuracy. Here, the maps of some users, as provided by FootSLAM, are fused together by means of an iterative offline algorithm.

In Fig. 5.5 a map resulting from the fusion of 4 datasets is depicted. For computational issues, only the best weighted particle is involved in the fusion rather than the weighted average of all particles. This is due to the fact that SLAM is invariant for translation, rotation and scale, so that each map should be modified



(a) (b)

Figure 5.6 Intuitive particle selection in FootSLAM based algorithms: loop closure in (a) is preferred than the 'open area' map in (b).

properly to fit the others before the merging algorithm can take place. This operation is quite time consuming, so it is done only on one particle, the 'best' one, per user.

5.4.8 Considerations

FootSLAM results are really interesting but undergo remarkable limitations which promote new lines of research. To this end, an intuitive analysis of the algorithm is useful.

As said above, step measurements are affected by a noisy process, correlated and difficult to model, that can accrue over time leading to meaningless maps, as the ones in Fig. 5.1. The estimation of the map is used to correct those errors so that the results of Fig. 5.4 can be achieved. To understand how the correction takes place, we shall consider the following case study: a user walks around a rectangular path enclosing three loops, and two of the particles hypothesize the trajectory as depicted in Fig. 5.6.

Now we wonder which one is selected by the algorithm, i.e. has the highest weight. The crucial point is to evaluate I_M^i of eq. (5.9) in the two cases; note that in principle both could be true, but it is intuitive that the one in Fig. 5.6.a is much more likely than the other one, since it represents better an indoor layout. Indeed

FootSLAM gives priority to well structured and connected maps in which loops tend to overlap. Mathematically this is done by considering that the integrand function in eq. (5.9) is enhanced by the overlapping of the two distributions involved: the previously estimated map and the new likelihood function[2]. On the other hand, if they do not overlap that particle is considered unlikely and thus it is penalized.

Now, FootSLAM limitations should be clear:

- it fails in open spaces, i.e. where situation of Fig. 5.6.b could be true;
- its convergence is strictly linked to loop closure (or even several loop closures).

This can also be seen as a set of constraints on the movement of the user. The addition of other sensors in a suitable data fusion framework can mitigate or even overcome these problems, leading to a more versatile algorithm.

Of course, we need a framework that is scalable with highly heterogeneous data. Whatever sensor we add, we can expect that:

- the state of the algorithm is augmented with a new map related to the new type of measurement and independent of the other maps;
- its contribution to the particle weights is independent of the other sensors'.

5.5 Integrating proximity information: PlaceSLAM

A first example of heterogeneus data fusion is the integration of proximity information into FootSLAM. Let us consider a set of 'places' recognizable by the user, manually or by means of some

[2]The map distribution acts here as prior information.

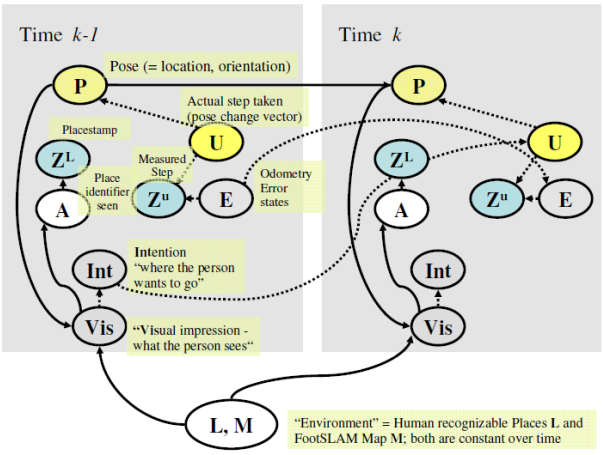


Figure 5.7 Combined Dynamic Bayesian Network for FootSLAM and PlaceSLAM showing two time slices. Arcs that only pertain to the FootSLAM aspect are drawn dotted. Measurements **Z** are in blue.

automatic equipment, and assume that when the user gets close to a place, a notification is available in the algorithm. This is possible with either manual notifications (the user himself presses a key after seeing a place) or automatic equipment, e.g. RFID tags.

PlaceSLAM [RAK10] is an algorithm that extends FootSLAM to include also proximity information. The most interesting case is when the algorithm ignores the places' number and locations: in this case a new map, say **L**, is supposed to include each discovered place. As described in the DBN of Fig. 5.7, the map **L** influences the newly defined state variable, say $\mathbf{A}_k$, indicating if a place is detected at instant k: the related measurement $\mathbf{Z}_k^L$ can be either the ID of the place detected, if for example a RFID tag is being used, or just a boolean variable if no information on the place ID is available.

When the user starts walking the map **L** is empty, until the first place is detected and a Gaussian pdf is assigned to the place position[3]. Note again that the map is probabilistic: the position

[3]The Gaussian pdf, employed for simplicity, is centered on the user current

pdf, in terms of its parameters, is stored for each place. From now on, any detection should be associated either to a place previously included in the map or to a new one; in the case of ID transmission this is straigthforward, otherwise some choice criterium is required[4].

5.5.1 Bayesian filter & implementation

The score function in PlaceSLAM is the joint posterior pdf

$$p(\mathbf{P}_{0:k}\mathbf{U}_{0:k}\mathbf{E}_{0:k}, \mathbf{A}_{0:k}, \mathbf{M}, \mathbf{L}|\mathbf{Z}^U_{1:k}, \mathbf{Z}^L_{1:k}) \tag{5.10}$$

of the state augmented with the new variable $\mathbf{A}_{0:k}$ and the maps given all measurements. This is implemented with a RBPF, similar to FootSLAM's approach (see Alg. 7), except for two main differences:

- the particle hypotheses include now

$$\left\{\mathbf{P}^i_k, \mathbf{U}^i_k, \mathbf{E}^i_k, \mathbf{A}^i_k, p\left(\mathbf{M}|\mathbf{P}^i_{0:k}\right), p\left(\mathbf{L}|\mathbf{A}^i_{0:k}, \mathbf{P}^i_{0:k}\right)\right\}. \tag{5.11}$$

  where the map $\mathbf{L}$ is updated at each measurement $\mathbf{Z}^L_{1:k}$;

- the weight w^i_k of each particle is

$$w^i_k \propto w^i_{k-1} \cdot I^i_M \cdot I^i_L, \tag{5.12}$$

  where I^i_M is the same as in FootSLAM (see eq. (5.9)) and it is thus related to the floor map $\mathbf{M}$, while I^i_L, related to the

position and has a standard deviation related to the visibility of the place itself.

[4] In [RAK10] a distance d_{min} is arbitrarly set: if the current pose is farther from any other known place than d_{min}, then a new place is added to the map $\mathbf{L}$.

places map $\mathbf{L}$, is a numerical approximation[5] for

$$\begin{aligned} I_L &= p\left(\mathbf{Z}_k^L|\mathbf{P}_{0:k}^i, \mathbf{A}_{0:k}^i, \mathbf{Z}_{0:k-1}^L\right) \\ &= \int_{\mathbf{L}} p\left(\mathbf{Z}_k^L|\mathbf{L}, \mathbf{P}_{0:k}^i, \mathbf{A}_{0:k}^i\right) \cdot p\left(\mathbf{L}|\left\{\mathbf{P}^i, \mathbf{A}^i, \mathbf{Z}^L\right\}_{0:k-1}\right) d\,\mathbf{L}. \end{aligned} \tag{5.13}$$

5.5.2 Considerations

An experimental analysis about PlaceSLAM is presented in [RAK10] and here only summarized. The authors show how the places mapping is robust even with binary measurements. The net result is that convergence is sped up, since passing by a place suddenly reduces the particles variance. Nonetheless loop closure limitation still holds, since a single detection of a place is not sufficient for a meaningful contribution.

5.6 WiSLAM: improving FootSLAM with WiFi

PlaceSLAM is a first attempt to overcome FootSLAM limitations, but is still limited by some factors as the number and disposition of places and the way of detecting them: either additive hardware or manual effort is needed. We have already seen that opportunistic signals like WLAN power measurements are a chance to reuse a technology already available in indoor environments. Furthermore they also provide distance instead of proximity information in a wider range, so that also one AP is supposed to give a valuable contribution.

In the case of WLAN measurements the map should include information about the APs, such as their position, as well as other

[5]The approximation is easily computable by assuming a Gaussian distribution for the distance at which the place is detected; moreover, a missed detection is considered possible.

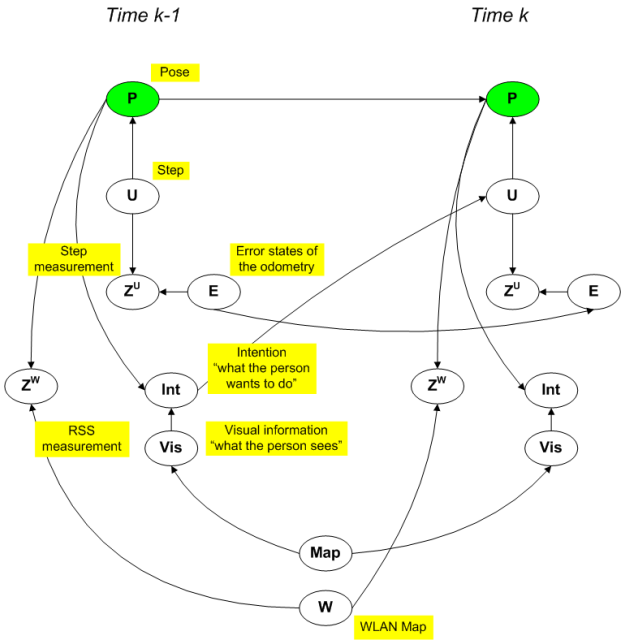


Figure 5.8 DBN for WiSLAM showing two time slices. The **Map** can include any features and information to let the pedestrian choose **Int**, while **W** is the WLAN map, in our case composed of APs' position and emitted power.

parameters that could be useful. Of course, introducing new variables can increase the complexity of the algorithm, so that a compromise is necessary. In this case we introduce only one further parameter per AP, i.e. the emitted power, its knowledge being of capital importance in WLAN based indoor navigation.

The resulting algorithm, named WiSLAM [BMM11], is the main original contribution of this chapter and is next analyzed in detail starting from the theorical approach up to extensive experimental results.

5.6.1 Preliminaries

We represent the estimation problem through a DBN similar to FootSLAM's and PlaceSLAM's ones (see Fig. 5.8 for two adjacent time slices). The pedestrian is assumed to be guided by his in-

tention **Int** (e.g. to reach a room or to exit the building), that is influenced by the human visual system **Vis** and thus by the environment **Map** (**M**). The intention, neither measurable nor processed in our algorithm, determines the step $\mathbf{U}_k$ that leads to the new pose $\mathbf{P}_k$, given the past one $\mathbf{P}_{k-1}$. The step measurement $\mathbf{Z}_k^U$ is observable (odometry) and is affected by IMU's correlated errors encoded in the state variable $\mathbf{E}_k$. In WiSLAM there is the RSS vector $\mathbf{Z}_k^W$ as well, that is basically a noisy measurement of the current distance between the user and any number of APs. It is influenced by the WLAN Map **W**, concerning position and emitted power of the APs. Note that the length of $\mathbf{Z}_k^W$ is variable, since the user can either enter or exit APs' ranges. This is easily handled since the RSS measurements are always labeled because the APs transmit their MAC address.

RSS Models

Our main assumptions are that RSS from different APs are independent given the user's position and, furthermore, the APs' maps **W** are also independent. This allows us to compute the contribution of each AP independently. Moreover, different measurements from the same AP are also conditionally independent. Despite this assumption be quite common in the literature, it should be validated with real data. Given the current Euclidean distance $\mathbf{r}_k$ of the user from the AP, located at $\mathbf{x}_{AP}$, the RSS likelihood is assumed to be Gaussian with variance σ^2 and mean $h(\mathbf{r}_k)$ given by the simple model [BP00b]

$$h\left(\mathbf{r}_k\right) = \mathbf{h} - 20\alpha \log_{10}\left(\mathbf{r}_k/d_0\right), \tag{5.14}$$

where **h** is the reference signal strength emitted by the AP, accounting also for the antenna orientation and gain, α is the propagation exponent, varying from 2 (free space) up to 4, and d_0 is a known reference distance. Note that both **h** and α are usually unknown, and **h** is found to vary strongly for different APs with dramatic effects on the mapping, unless it is learnt. This is why we introduce both $\mathbf{x}_{AP}$ and **h** in the WiFi map **W**. We found less

sensitivity to α and thus for simplicity we set $\alpha = 2$. Finally, it was convenient to model $\mathbf{h}$ as a discrete random variable, independent from AP to AP, with N_h equally spaced values $\{h_h\}_{h=1..N_h}$ for any APs.

5.6.2 Bayesian filter

Extending the approach of Sect. 5.4.2 we are interested in the joint posterior

$$p(\mathbf{P}_{0:k}\mathbf{U}_{0:k}\mathbf{E}_{0:k}, \mathbf{W}, \mathbf{M}|\mathbf{Z}^U_{1:k}, \mathbf{Z}^W_{1:k}) \tag{5.15}$$

of the state histories, floor and WiFi maps given step and RSS measurements, which factorizes into

$$p(\mathbf{M}|\mathbf{P}_{0:k}) \cdot p(\mathbf{W}|\mathbf{P}_{0:k}, \mathbf{Z}^W_{1:k}) \cdot p(\mathbf{P}_{0:k}\mathbf{U}_{0:k}\mathbf{E}_{0:k}|\mathbf{Z}^U_{1:k}, \mathbf{Z}^W_{1:k}). \tag{5.16}$$

Following the FootSLAM derivation, the last term in eq. (5.16) admits a recursive formulation based on the independence relationships encoded in the DBN:

$$\begin{aligned} & p\left(\{\mathbf{PUE}\}_{0:k}|\mathbf{Z}^W_{1:k}, \mathbf{Z}^U_{1:k}\right) \propto \\ & \quad p\left(\mathbf{Z}^U_k|\{\mathbf{UE}\}_k\right) \cdot p\left(\mathbf{Z}^W_k|\mathbf{P}_{0:k}, \mathbf{Z}^W_{1:k-1}\right) \cdot \\ & \quad p\left(\{\mathbf{E}_k|\mathbf{E}_{k-1}\}\right) \cdot p\left(\{\mathbf{PU}\}_k|\{\mathbf{PU}\}_{0:k-1}\right) \cdot \\ & \quad p\left(\{\mathbf{PUE}\}_{0:k-1}|\mathbf{Z}^U_{1:k-1}, \mathbf{Z}^W_{1:k-1}\right). \end{aligned} \tag{5.17}$$

The novelty in WiSLAM with respect to FootSLAM is the RSS likelihood term.

From eq. (5.16) it is clear that the $\mathbf{W}$ map can have a strong influence on the posterior. We define

$$\begin{aligned} I_W &\hat{=} p\left(\mathbf{Z}^W_k|\mathbf{P}_{0:k}, \mathbf{Z}^W_{1:k-1}\right) \\ &= \prod_{j=1}^{N_k} p\left(Z^W_{j,k}|\mathbf{P}_{0:k}, Z^W_{j,1:k-1}\right) \\ &= \int_{\mathbf{W}} p\left(\mathbf{Z}^W_k|\mathbf{W}, \mathbf{P}_k\right) \cdot p\left(\mathbf{W}|\mathbf{P}_{0:k-1}, \mathbf{Z}^W_{1:k-1}\right) \, d\mathbf{W}. \end{aligned} \tag{5.18}$$

The above integral is over a 3 dimensional space: the spatial dimensions are continuous (the AP's position), the last one is the discrete-valued signal strength. These considerations allow us to marginalize over $\mathbf{h}$

$$I_W = \sum_{h=1}^{N_h} \Pr\left(h_h|\mathbf{P}_{0:k-1}, \mathbf{Z}^W_{1:k-1}\right) \cdot \int_{\mathbf{x}_{AP}} p\left(\mathbf{Z}^W_k|\mathbf{x}_{AP}, h_h, \mathbf{P}_k\right) \cdot p\left(\mathbf{x}_{AP}|h_h, \mathbf{P}_{0:k-1}, \mathbf{Z}^W_{1:k-1}\right) \, d\mathbf{x}_{AP}. \tag{5.19}$$

WLAN map learning

The last point to consider is the map learning. Of course $\mathbf{M}$ is evaluated like in FootSLAM, Sect. 5.4.2. With the factorization

$$p(\mathbf{W}|\mathbf{P}_{0:k}, \mathbf{Z}^W_{1:k}) = p(\mathbf{x}_{AP}|\mathbf{h}, \mathbf{P}_{0:k}, \mathbf{Z}^W_{1:k}) \cdot p(\mathbf{h}|\mathbf{P}_{0:k}, \mathbf{Z}^W_{1:k}), \tag{5.20}$$

we split the WLAN map estimation into two separate tasks. To determine the probabilities for h_h and assuming a suitable prior, e.g. uniform, we apply the Bayes rule to express:

$$\Pr\left(h_h|\mathbf{P}_{0:k}, \mathbf{Z}^W_{1:k}\right) \propto p\left(\mathbf{Z}^W_k|h_h, \mathbf{P}_{0:k}, \mathbf{Z}^W_{1:k-1}\right) \cdot \Pr\left(h_h|\mathbf{P}_{0:k-1}, \mathbf{Z}^W_{1:k-1}\right). \tag{5.21}$$

More insight is needed when looking at the estimation of the AP's position $\mathbf{x}_{AP}$, given $\mathbf{h}$. The arguments explained in the Sect. 4.5 are still valid here: given the parameter, the AP's location pdf results in

$$\begin{aligned} &p\left(\mathbf{x}_{AP}|h_h, \mathbf{P}_{0:k}, \mathbf{Z}^W_{1:k}\right) \\ &\quad \propto p\left(\mathbf{Z}^W_k|\mathbf{x}_{AP}, h_h, \mathbf{P}_k\right) \cdot p\left(\mathbf{x}_{AP}|h_h, \mathbf{P}_{0:k-1}, \mathbf{Z}^W_{1:k-1}\right) \\ &\quad \propto \prod_{s=1}^{k} p\left(\mathbf{Z}^W_s|\mathbf{x}_{AP}, h_h, \mathbf{P}_s\right), \end{aligned} \tag{5.22}$$

that is the normalized product of k non concentric donut-shaped functions.

The complete $p(\mathbf{W}|\mathbf{P}_{0:k}, \mathbf{Z}^W_{1:k})$ is thus a mixture of N_H 'donuts' products, in which the coefficients, that are the probabilities for h_h (last term in eq. (5.20)), also evolve over time, viz.

$$p\left(\mathbf{W}|\mathbf{P}_{0:k}, \mathbf{Z}^W_{1:k}\right) \propto \sum_{h=1}^{N_h} \left\{ Pr\left(h_h|\mathbf{P}_{0:k-1}, \mathbf{Z}^W_{1:k-1}\right) \left[\prod_{s=1}^{k} p\left(\mathbf{Z}^W_s|\mathbf{x}_{AP}, h_h, \mathbf{P}_s\right)\right] \cdot \delta\left(\mathbf{h} - h_h\right) \right\}, \tag{5.23}$$

where $\delta(\cdot)$ denotes the Dirac function.

5.6.3 Filter implementation

Like in FootSLAM, for a RBPF implementation of the Bayesian filter, we sample from the 'likelihood PF' proposal density:

$$p\left(\mathbf{U}_k|\mathbf{Z}^U_k, \mathbf{E}^i_k\right) \cdot p\left(\mathbf{E}_k|\mathbf{E}^i_{k-1}\right). \tag{5.24}$$

The RSS contribution is a further factor in the particle weights

$$w^i_k \propto w^i_{k-1} \cdot I^i_M \cdot I^i_W \tag{5.25}$$

where I^i_M is relative to $\mathbf{M}$ estimation (Sect. 5.4.4) and I^i_W is a convenient numerical approximation for I_W in eq. (5.19). The nature of this approximation is pivotal from the implementation point of view. The first method we propose is useful for some preliminary results, then a more rough approximation will be required.

WLAN Map sampling

Let us consider a set of N_W points $\left\{\mathbf{x}^j_{AP}\right\}_{j=1\ldots N_W}$ uniformly drawn in the area under test (or deterministically sampled on a grid). We can now approximate the distribution of $\mathbf{W}$ given data and user's poses up to k [AMGC02]

$$p\left(\mathbf{x}_{AP}|h_h, \mathbf{P}^i_{0:k}, \mathbf{Z}^W_{1:k}\right) \approx \sum_{j=1}^{N_W} \tilde{a}^j_k \delta\left(\mathbf{W} - \mathbf{W}^j\right) \tag{5.26}$$

where the weights

$$\tilde{a}_k^j \propto p\left(\mathbf{x}_{AP}^j | h_h, \mathbf{P}_{0:k}^i, \mathbf{Z}_{1:k}^W\right) \tag{5.27}$$

are normalized over j.

We can approximate the integral in eq. (5.19) at the arbitrary instant k by

$$\begin{aligned}
& \int p\left(\mathbf{Z}_k^W | \mathbf{P}_k^i, \mathbf{x}_{AP}, h_h\right) p\left(\mathbf{x}_{AP} | h_h, \mathbf{P}_{0:k}^i, \mathbf{Z}_{1:k-1}^W\right) \, d\mathbf{x}_{AP} \\
\approx & \int p\left(\mathbf{Z}_k^W | \mathbf{P}_k^i, \mathbf{x}_{AP}, h_h\right) \sum_{j=1}^{N_W} \tilde{a}_{k-1}^j \delta\left(\mathbf{x}_{AP} - \mathbf{x}_{AP}^j\right) \, d\mathbf{x}_{AP} \\
= & \sum_{j=1}^{N_W} \tilde{a}_{k-1}^j \int p\left(\mathbf{Z}_k^W | \mathbf{P}_k^i, \mathbf{x}_{AP}, h_h\right) \delta\left(\mathbf{x}_{AP} - \mathbf{x}_{AP}^j\right) \, d\mathbf{x}_{AP} \\
= & \sum_{j=1}^{N_W} \tilde{a}_{k-1}^j p\left(\mathbf{Z}_k^W | \mathbf{P}_k^i, \mathbf{x}_{AP}^j, h_h\right),
\end{aligned} \tag{5.28}$$

yielding for I_W^i

$$\begin{aligned}
I_W^i = & \int p\left(\mathbf{Z}_k^W | \mathbf{W}, \mathbf{P}_k^i\right) p\left(\mathbf{W} | \mathbf{P}_{0:k-1}^i, \mathbf{Z}_{1:k-1}^W\right) \, d\mathbf{W} \\
= & \sum_{h=1}^{N_h} \Big\{ Pr\left(h_h | \mathbf{P}_{0:k-1}^i, \mathbf{Z}_{1:k-1}^W\right) \\
& \cdot \int p\left(\mathbf{Z}_k^W | \mathbf{x}_{AP}, h_h, \mathbf{P}_k^i\right) p\left(\mathbf{x}_{AP} | h_h, \mathbf{P}_{0:k-1}^i, \mathbf{Z}_{1:k-1}^W\right) \, d\mathbf{x}_{AP} \Big\} \\
\approx & \sum_{h=1}^{N_h} \left\{ Pr\left(h_h | \mathbf{P}_{0:k-1}^i, \mathbf{Z}_{1:k-1}^W\right) \left[\sum_{j=1}^{N_W} \tilde{a}_{k-1}^j p\left(\mathbf{Z}_k^W | \mathbf{P}_k^i, \mathbf{x}_{AP}^j, h_h\right) \right] \right\},
\end{aligned} \tag{5.29}$$

that is easily computable for each particle.

5.6.4 Summary of the algorithm

The algorithm is now briefly summarized.

Algorithm 8 (WiSLAM - sampled WLAN map)**.**

- For $k = 0$
  - Initialize all N_P particles to $\mathbf{P}_0^i = (x = 0, y = 0, h = 0)$, where x, y and h denote the pose location and heading in two dimensions; draw $\mathbf{E}_0^i$ from a suitable initial distribution for the error state.
  - Draw N_W points $\mathbf{x}_{AP}^j$ from the two dimensional area under test (like i.i.d. realizations of a uniform r.v. or deterministically on a grid).
  - Set $\forall\, i, j, h$, $p\left(\mathbf{x}_{AP}^j | h_h, \mathbf{P}_0^i\right) = 1/N_W$, and $Pr\left(h_h | \mathbf{P}_0^i\right) = r_h \geq 0$ (r_h arbitrary but with $\sum_{h=1}^{N_h} r_h = 1$).
- for each time step increment k:
  - Particles U_k^i, E_k^i are drawn from the proposal density (5.24).
  - The pose $\mathbf{P}_k^i$ is computed by adding the vector $\mathbf{U}_k^i$ to $\mathbf{P}_{k-1}^i$.
  - The particle weight updates are simply

    $$w_k^i \approx w_{k-1}^i \cdot I_M^i \cdot I_W^i$$

    where I_M^i is computed like in FootSLAM and I_W^i like in eq. (5.29), with $\tilde{a}_{k-1}^j$ given in eq. (5.27).
  - Compute $p\left(\mathbf{M} | \mathbf{P}_{0:k}^i\right)$, $\forall i$, like in FootSLAM.
  - Update $p\left(\mathbf{x}_{AP}^j | h_h, \mathbf{P}_{0:k}^i, \mathbf{Z}_{1:k}^W\right)$, $\forall\, i, j, h$.
  - Update $Pr\left(h_h | \mathbf{P}_{0:k}^i, \mathbf{Z}_{1:k}^W\right)$, $\forall\, i, h$, like (5.21), with the integrals approximation (5.28) and normalize.
  - Resampling can be performed if required.

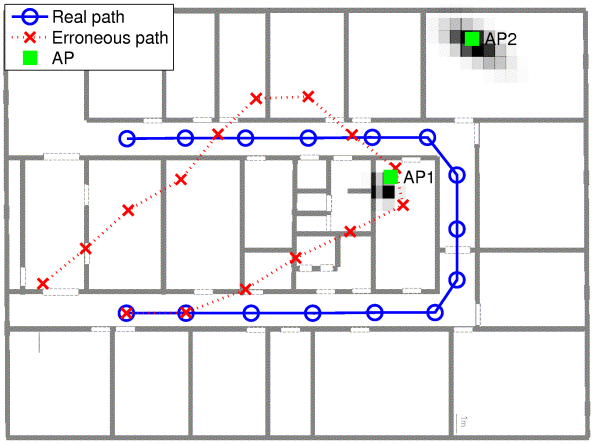


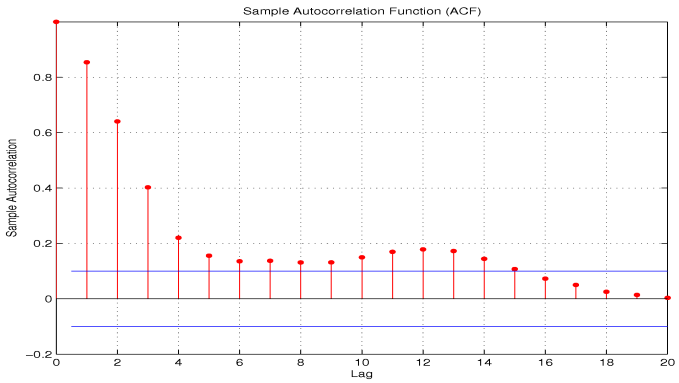
Figure 5.9 Experimental testbed adopted for results shown in Sect. 5.6.5. In evidence the final pdfs for both APs' positions produced by one of the datasets.

5.6.5 Real world experiments

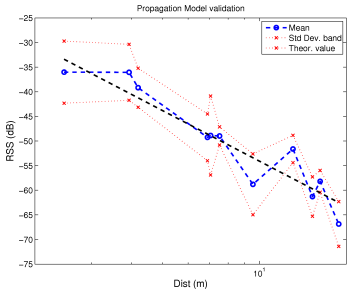
Extensive real data measurements were carried out to validate the fundamentals of WiSLAM in an indoor area of about 20×40 m and occupied by offices (refer to Fig. 5.9). We used a laptop equipped with an internal network device Link 5100, compliant with IEEE 802.11 a/b/g, and carried by a human operator. The measurements have been collected using a freeware working under Windows 7 OS. We employed two APs (green squares in Fig. 5.9), a Cisco AiroNet 1130 and an Apple Airport Extreme A1301 respectively, both IEEE 802.11 a/b/g compliant.

Statistical analysis

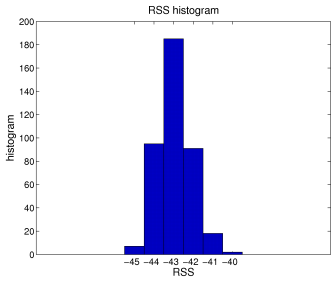
Our first analysis aims at validating our hypotheses about measurements. We start with the Autocorrelation Function (ACF) of the measurement noise. In fact the WiFi API allows to read measurements at whatever rate (typical rates for freewares are 1 RSS/s), but it does not ensure statistical independence. Fig. 5.10.a shows the (normalized) autocorrelation function $R(\tau)$ of the RSS



(a) Noise autocorrelation



(b) Propagation model



(c) Likelihood model

Figure 5.10 Model validation: (a) RSS noise autocorrelation, (b) propagation model, (c) likelihood model.

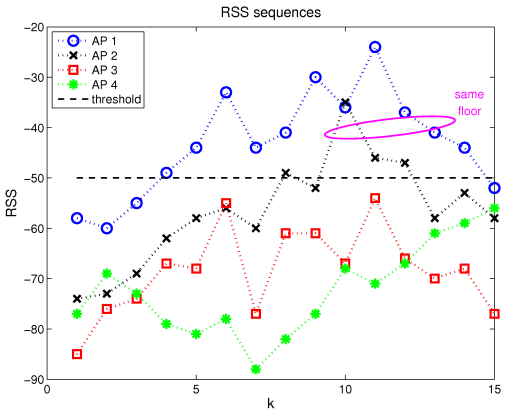


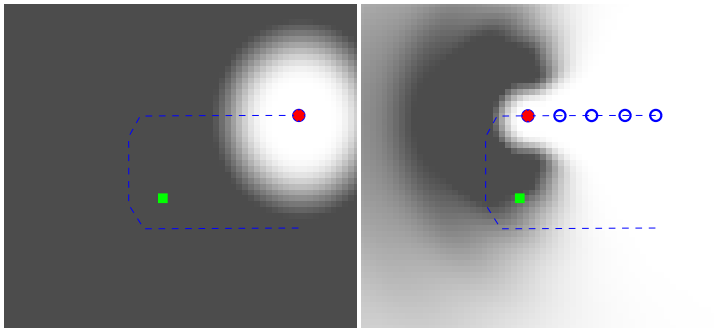
Figure 5.11 RSS sequences from different APs (testbed of Fig. 5.9): AP 1-2 are the positions depicted in Fig. 5.9, AP 3 is downstairs, AP 4 is far from the user's trajectory.

noise collected, with

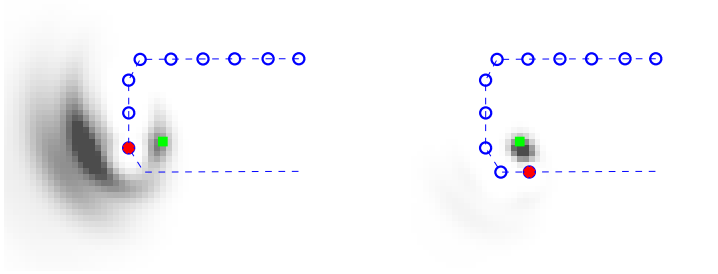
$$R(\tau) \propto \sum_k (\mathbf{Z}_k^W - \bar{\mathbf{Z}}^W)(\mathbf{Z}_{k-\tau}^W - \bar{\mathbf{Z}}^W),$$

against time; it has been computed on a sequence of 100 measurements collected at a rate of 1 RSS/s, with $\bar{\mathbf{Z}}^W$ being the sample mean. We can note that low values for $R(\tau)$ are achieved for measurements 3 or 4 seconds apart and this will be our collecting rate, even if some filtering could be useful in this case.

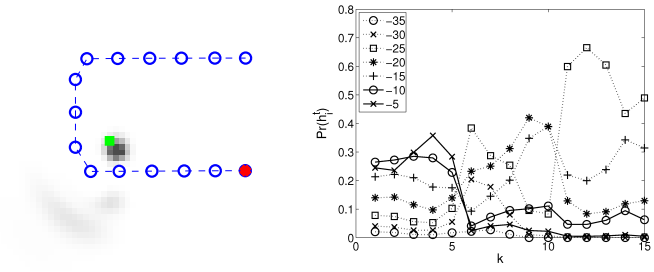
Fig. 5.10.b illustrates the fit of the propagation model presented in eq. (5.14). For this purpose a radio map was built in the environment under test, considering also different directions of the laptop; the resulting curve, obtained by a least squares approach is depicted over the distance. The comparison with the theoretical curve provides a measurement of the model mismatch, as well as the best values for the model parameters. Moreover, in Fig. 5.10.c, the histogram of the RSS collected in a static position presents a 'bell' shape that corroborates the Gaussian hypothesis.



(a) k=1 (b) k=5

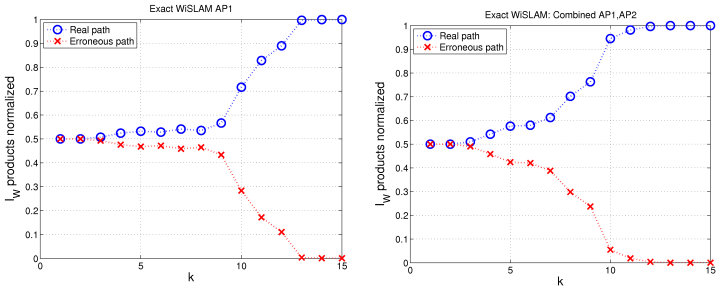


(c) k=9 (d) k=11



(e) k=15 (f) h_h probabilities

Figure 5.12 Mapping for a single AP: real data collected during a walk are employed to map the AP's position (a-e) and reference signal strength (f). For the meaning of the symbols see Fig. 4.6. The environment is the one depicted in Fig. 5.9 and is here omitted for clarity.



(a) Exact - Only AP 1 (b) Exact - AP 1 + AP 2

Figure 5.13 Competing paths: products (normalized) of the I_W terms for both paths in Fig. 5.9, averaged on 10 datasets, in the cases of (a) only AP 1 involved and (b) both APs involved. The line related to the real path is dotted with blue circles.

RSS prefiltering

Fluctuations in the output power of the AP and the non-ideal behavior of the receiver lead to outlying measurements (about 2% of the total) of tens of dBm in the received signal. Since a single outlier can have a negative effect on the convergence of the algorithm and recalling that WiSLAM is an off-line algorithm, we chose to prefilter the RSS series, with a basic approach, in which a RSS measurement differing from both the previous and the following ones by more than 10 dBm is removed.

Another kind of RSS prefiltering is related to multiple APs. Since each AP needs the allocation of new processing resources, it is of interest to process only those APs with stronger RSS measurements, i.e. with a smaller noise on the related distance estimations. An example is depicted in Fig. 5.11: the RSS sequences relative to 4 APs are shown together with a heuristic threshold (see Fig. 5.9 for the testbed) and AP 3 and AP 4 are rejected. The former is located on another floor of the same building. To neglect non-planar APs is doubly important, because their RSS measurements are strongly attenuated by the floor and also because distances are distorted by the third dimension. The latter AP is on the same floor as the user but far from his trajectory, so that it is not worth being processed. Moreover, we neglect the 802.11a standard since

at higher frequencies the effects of obstacles are more pronounced.

Preliminary results

Both the mapping and the localization part of WiSLAM were separately tested. To test a realistic scenario we took 10 datasets following the same path during office hours, with the APs fully operative. All the processings took place off-line, allowing non-causal procedures like RSS prefiltering.

First we map the AP using the actual user's positions and estimating also the reference signal strength (7 values in the range $[-35, -5]$ dB, RSS standard deviation $\sigma = 3$ dB). In Fig. 5.12.a we see that the **W** pdf after the first RSS is very spread and only at $k = 5$ a better resolution is shown (Fig. 5.12.b). Two turns are necessary to avoid ambiguity (Fig. 5.12.c-e) and the reason is visible in Fig. 5.12.f, where the corresponding h_h probabilities are presented: the mapping is well performed when one reference strength (in this case -25 dB) wins over the others (after about $k = 10$ steps). This is the price paid for the **h** estimation.

Mapping is just a crucial part of SLAM, but not the only one. RSS measurements allow us to distinguish between the real user's path from a competing one, affected by the heading error typical of odometry (Fig. 5.9 - the right path is in blue). As a figure of merit we use the product of the weights I_W over time, normalized for simplicity and averaged on all the datasets available. This is shown in Fig. 5.13, highlighting the capability of our algorithm to discriminate between the two paths after few steps. Furthermore, we have considered a case with only the contribution of AP 1 (Fig. 5.13.a) and a case with both (Fig. 5.13.b), in which there are clear benefits.

5.7 Simplified WiSLAM

Results in the Sect. 5.6.5 are promising, but meaningless since only two particles are handled: but what about the final software platform operating with thousand of particles?

In this case the computational burden due to the sampling and updating of the WLAN map for all particles prevents from a practical implementation. Let us dig deeper this point.

In the Bayesian filter the key point is represented by the function

$$p\left(\mathbf{W}|\mathbf{P}^i_{0:k-1}, \mathbf{Z}^W_{1:k-1}\right) \cdot p\left(\mathbf{Z}^W_k|\mathbf{P}^i_k, \mathbf{W}\right) \propto \prod_{s=1}^{k} p\left(\mathbf{Z}^W_s|\mathbf{P}^i_s, \mathbf{W}\right). \tag{5.30}$$

When integrated, in fact, it provides the WLAN contribution to the weight and, when normalized on $\mathbf{W}$, it is the new Wlan map.

A concise way to express the last term of eq. (5.30) for a give h_h, recalling that likelihoods are Gaussian pdfs, is

$$\prod_{s=1}^{k} p\left(\mathbf{Z}^W_s|\mathbf{P}^i_s, \mathbf{W}\right) \propto \exp\left\{-\sum_{s=1}^{k} \frac{\left(\mathbf{Z}^W_s - h_h + 20\alpha \log(r_s/d_0)\right)^2}{2\sigma^2}\right\}, \tag{5.31}$$

whose number of parameters grows in time, preventing from using tabular approximations for the integral at least. The most direct solution, the sampling (either static or dynamic) technique presented above, is computationally unmanageable, so that a more rough, but still accurate, method is necessary. In this section we provide our solution, based on a Gaussian Mixture approximation of the WLAN map.

5.7.1 Wlan map approximation

To provide a good approximation for the map we need some guidance. First, if we have only a few RSS, then the AP's position pdf is too spread to be useful in localization. That is why we can start the algorithm when sufficient information is available about the AP's position. Its pdf tends to a set of quite narrow peaks (given the power hypothesis h_h), suggesting our main assumption: we approximate each peak with a bivariate Gaussian function, so that the whole pdf is expressed by means of a Gaussian Mixture Model (GMM).

Assumption 1 (GMM for $\mathbf{x}_{AP}$ pdf)**.** *The pdf of the AP's position $\boldsymbol{x}_{AP}$ given h_h of eq. (5.22) can be approximated at the arbitrary instant k by a GMM*

$$p\left(\boldsymbol{x}_{AP}|h_h, \boldsymbol{P}_{0:k}, \boldsymbol{Z}_{1:k}^W\right) \approx \hat{p}\left(\boldsymbol{x}_{AP}|h_h, \boldsymbol{P}_{0:k}, \boldsymbol{Z}_{1:k}^W\right) \hat{=} \sum_{p=1}^{N_{peak}} \tilde{u}_{p,k} f_{p,k}(\boldsymbol{x}_{AP}, h_h), \tag{5.32}$$

where

$$f_{p,k}(\boldsymbol{W}) \sim \mathcal{N}\left(\mu_{p,k}, S_{p,k}\right),$$

$\tilde{u}_{p,k}$ is a set of suitable coefficients whose sum is 1, $\mu_{p,k}$ and $S_{p,k}$ are mean and covariance matrix of the p^{th} peak respectively.

Mainly, the GMM approximation allows us to express a non-parametric pdf in terms of a parametric pdf that is much more manageable. In App. E we will also sketch a theoretical justification for it. Now, we develop a simple implementation for WiSLAM that preserves the performance but overcomes its computational problems. In detail we have to discuss the following three steps:

1. initialize the GMM when the algorithm is started;
2. update it recursively when a new RSS is available;
3. compute the weights I_W^i and update h_h probabilities.

5.7.2 Initialization

The initialization step (see Fig. 5.14) is triggered when sufficient measurements, say T, from a new AP are collected. Its goal is to build the approximated WLAN map given the collected RSS and the path hypothesized by each particle.

Let us recall the factorization of eq. (5.20) evaluated at step T:

$$\hat{p}(\mathbf{W}|\mathbf{P}_{0:T}, \mathbf{Z}_{1:T}^W) = \hat{p}\left(\mathbf{x}_{AP}|\mathbf{h}, \mathbf{P}_{0:T}, \mathbf{Z}_{1:T}^W\right) \cdot p\left(\mathbf{h}|\mathbf{P}_{0:T}, \mathbf{Z}_{1:T}^W\right).$$

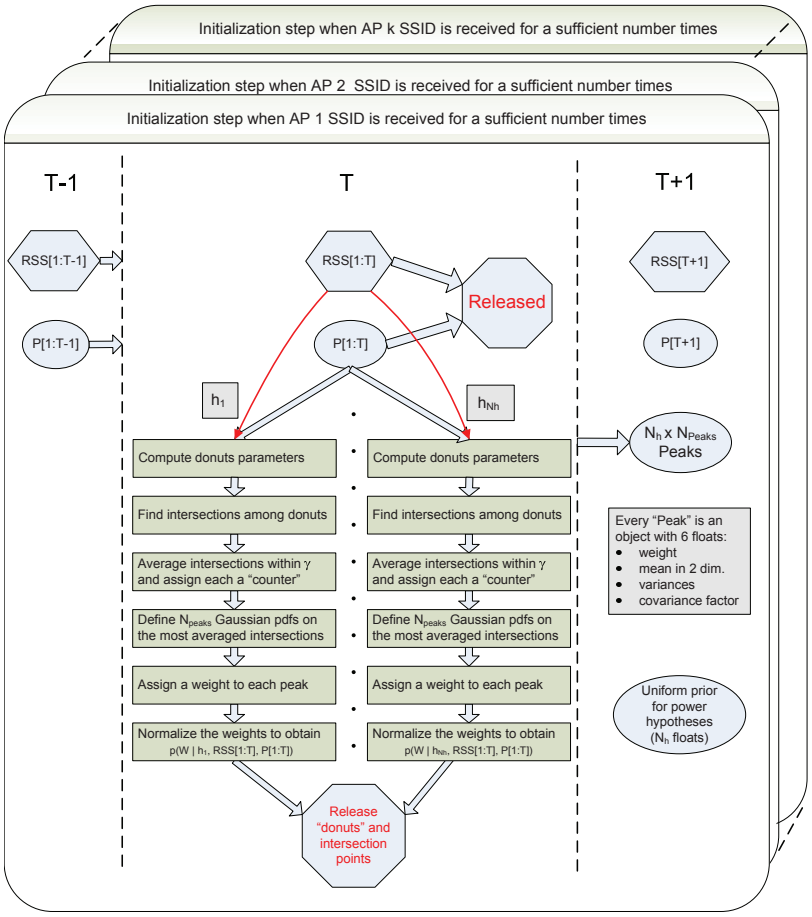


Figure 5.14 Simplified WiSLAM: block scheme for initialization with implementation tips. Variables in hexagons are global, the ones in ovals must be created for any particles. To 'release' a variable means that it is not used anymore and thus the related instance in the program can be erased.

The **h** distribution can be assigned uniformly, while the first term is given by the GMM in eq. (5.32). The main problem is to find the GMM parameters, i.e. to guess peaks' positions and parameters only by employing the sequence of measurements and poses. The solution found is not optimal but very reasonable: given h_h each RSS is equivalent to a distance measurement and thus it specifies a circle of the plane. The N_{peaks} points in the plane in which more donuts intersect are selected as locations for peaks. Fig. 5.15 shows an example for this, adding some implementation details. The next step would be to sample $p\left(\mathbf{x}_{AP}|h_h, \mathbf{P}_{0:T}, \mathbf{Z}^W_{1:T}\right)$ in a neighborhood of each selected point in order to extract sample mean $\mu_{p,T}$ and covariance matrix $S_{p,T}$ for the peaks (we omit the subscript h for simplicity). The coefficients $\tilde{u}_{p,T}$ in eq. (5.32) are given by the ratio between the exact pdf and the value of the p^{th} Gaussian pdf at the mean $\mu_{p,T}$ (a normalization over p is then necessary)

$$\tilde{u}_{p,T} \propto 2\pi \cdot \|S_{p,T}\|^{1/2} \cdot p\left(\mu_{p,T}|h_h, \mathbf{P}_{0:T}, \mathbf{Z}^W_{1:T}\right). \tag{5.33}$$

In the scheme in Fig. 5.14 we highlight that in the software implementation the RSS history is a global variable, while the poses history is different for any particles. Anyway, both can be released at the end of the step, together with all other variables created: in the next step we will need only new RSS and pose and the parameters of the **W** posterior (peaks and power probabilities).

5.7.3 Recursion

At $k > T$ a new RSS measurement $\mathbf{Z}^W_k$ is available, triggering the WiFi map updating. To compute the new $\mathbf{x}_{AP}$ pdf given **h** we apply the GMM approximation to eq. (5.22), i.e.

$$\begin{aligned}
&\hat{p}\left(\mathbf{x}_{AP}|h_h, \mathbf{P}_{0:k}, \mathbf{Z}^W_{1:k}\right) \\
&\propto p\left(\mathbf{Z}^W_k|\mathbf{x}_{AP}, h_h, \mathbf{P}_k\right) \cdot \hat{p}\left(\mathbf{x}_{AP}|h_h, \mathbf{P}_{0:k-1}, \mathbf{Z}^W_{1:k-1}\right) \\
&= \sum_{p=1}^{N_{peaks}} \tilde{u}_{p,k-1} \cdot f_{p,k-1}(\mathbf{x}_{AP}, h_h) \cdot p\left(\mathbf{Z}^W_k|\mathbf{x}_{AP}, h_h, \mathbf{P}_k\right).
\end{aligned} \tag{5.34}$$

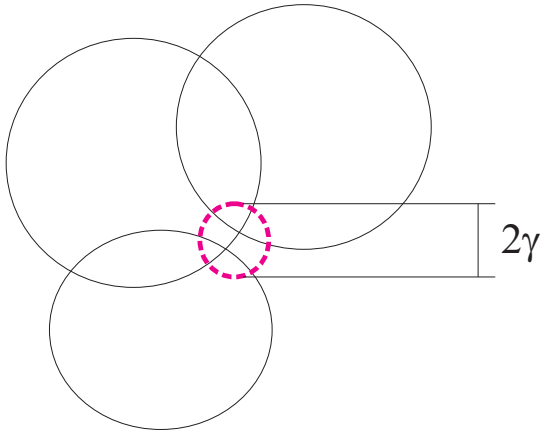


Figure 5.15 Example of intersection points between 3 donuts relative to 3 different measurements; since the points lie in a circle with radius γ (in this paper $\gamma = 2$ m) they are considered a single point. A sparse sampling in its neighborhood is performed to extract the peak parameters.

This new mixture is not generally a GMM but, under certain assumptions about the RSS likelihood, it is. Therefore, Ass. 1 is automatically true after the initialization.

We set $\mathbf{P}_k = 0$ for simplicity (a simple translation of the axes is always sufficient) and represent the pdf in polar coordinates (r, θ). We already noted that the radial component r is a Lognormal r.v., while the angular component θ is uniform. The following assumption is central in our approximation.

Assumption 2 (Gaussian band distribution)**.** *From now on we will consider the curvature of the 'donut' shaped pdf*

$$p\left(\boldsymbol{x}_{AP}|h_h, \boldsymbol{P}_{0:1}, \boldsymbol{Z}_1^W\right)$$

large enough so that it can be locally approximated by a 'band' distribution (as in Fig. 5.17). Moreover, we approximate the range pdf with a Gaussian r.v., instead of a Lognormal one, viz.

$$p\left(\boldsymbol{Z}_k^W|\boldsymbol{x}_{AP}, \boldsymbol{P}_k, h_h\right) \propto \exp -\frac{(r-\hat{r})^2}{2\sigma_G^2} \tag{5.35}$$

where $\hat{r}$ and σ_G are respectively mean and standard deviation of the original Lognormal pdf.

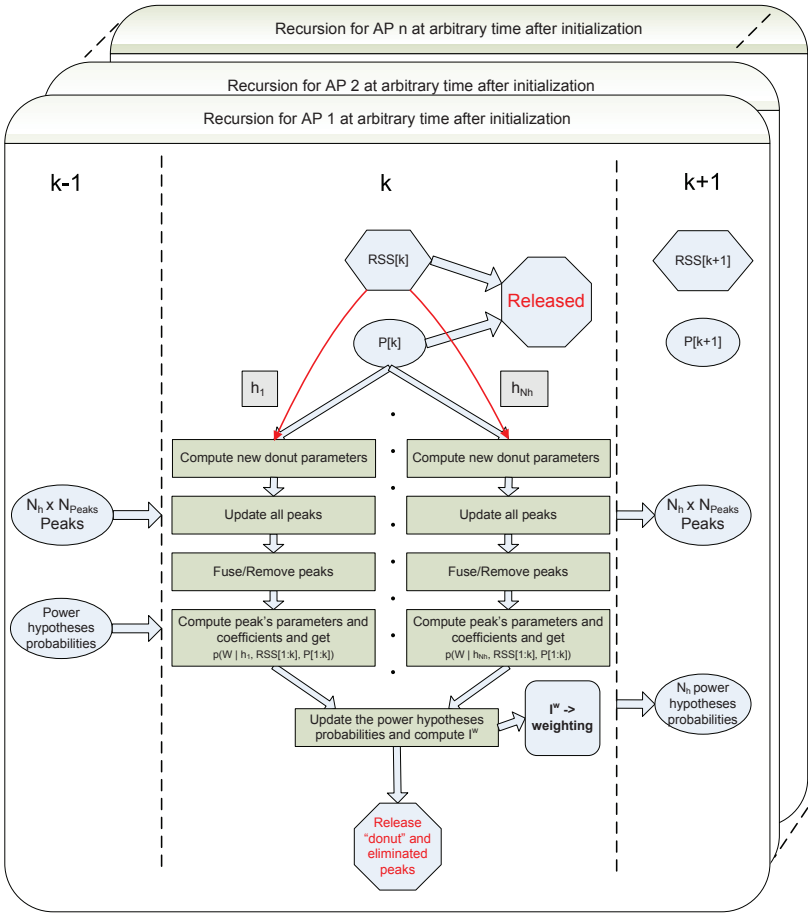


Figure 5.16 Simplified WiSLAM: block scheme for recursion with implementation tips. Variables in hexagons are global, the ones in ovals must be created for any particles. To 'release' a variable means that it is not used anymore and thus the related instance in the program can be erased.

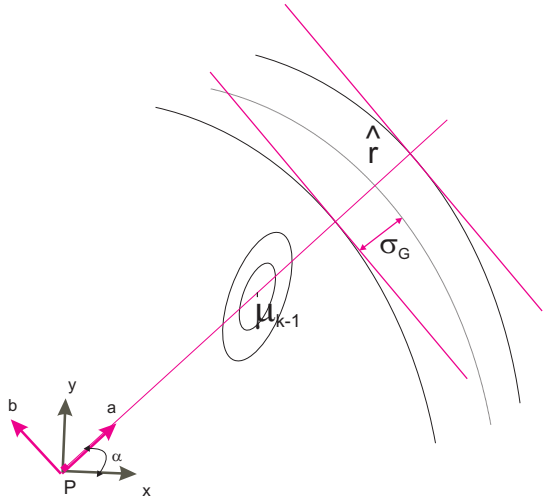


Figure 5.17 Gaussian band distribution: linearization of the RSS likelihood.

Ass. 2 is well explained by Fig. 5.17: we simply linearize the measurement donut in the neighborhood of its intersection with a peak. This is reasonable for narrow peaks, as we could expect if k is large enough. We now show that under these hypotheses the updated peak is again Gaussian, and so the new $\mathbf{x}_{AP}$ pdf is a new GMM. Of course we also provide, in the proof, a recursive way to compute the parameters for the new GMM.

Proposition 1 (Recursive computation of the peaks)**.** *Employing Ass. 1 for the distribution of* $\boldsymbol{x}_{AP}$ *at* $k-1$ *and Ass. 2 for the* k^{th} *RSS likelihood, their product can be expressed as a mixture of Gaussian function*

$$p\left(\boldsymbol{Z}_k^W|\boldsymbol{x}_{AP}, h_h, \boldsymbol{P}_k\right)\cdot \hat{p}\left(\boldsymbol{x}_{AP}|h_h, \boldsymbol{P}_{0:k-1}, \boldsymbol{Z}_{1:k-1}^W\right) = \sum_{p=1}^{N_{peaks}} u_{p,k} f_{p,k}(\boldsymbol{x}_{AP}, h_h), \tag{5.36}$$

and thus the updated distribution for $\boldsymbol{x}_{AP}$ *at* k *is again a GMM*

(its parameters are reported in the proof)

$$\hat{p}\left(\boldsymbol{x}_{AP}|h_h, \boldsymbol{P}_{0:k}, \boldsymbol{Z}^W_{1:k}\right) = \sum_{p=1}^{N_{peaks}} \tilde{u}_{p,k} f_{p,k}(\boldsymbol{x}_{AP}, h_h), \tag{5.37}$$

with

$$\tilde{u}_{p,k} = \frac{u_{p,k}}{\sum_{p=1}^{N_{Peaks}} u_{p,k}}. \tag{5.38}$$

Proof. The proof is reported in App. F. □

5.7.4 I^i_W approximation

Recalling the definition for I_W and the $\mathbf{h}$ update, the integral in eq. (5.19) is approximated by the sum of the unnormalized coefficients in eq. (5.36) over p

$$\begin{aligned}&\int_{\mathbf{x}_{AP}} p\left(\mathbf{Z}^W_k|\mathbf{x}_{AP}, h_h, \mathbf{P}_k\right) \cdot \hat{p}\left(\mathbf{x}_{AP}|h_h, \mathbf{P}_{0:k-1}, \mathbf{Z}^W_{1:k-1}\right)\, d\mathbf{x}_{AP} \\ &= \sum_{p=1}^{N_{Peaks}} u_{p,k},\end{aligned} \tag{5.39}$$

that is used also in the h_h probabilities update of eq. (5.21).

5.7.5 Summary of the algorithm

The simplified WiSLAM is now briefly summarized. Note that odometry and RSS' contributions from each AP are independent. Therefore they are processed in a separate way to obtain independent maps and they are only fused in the particle weights.

Algorithm 9 (Simplified WiSLAM)**.**

Initialization:

- Initialize all N_P particles to $\mathbf{P}_0^i = (x, y, h = 0)$, where x, y and h denote the pose location and heading in two dimensions; draw $\mathbf{E}_0^i$ from a suitable initial distribution for the odometry error state.

Then, for each time step increment k and any particles:

- Draw U_k^i, E_k^i from the proposal density in eq. (5.24), compute $\mathbf{P}_k^i$ by adding the vector $\mathbf{U}_k^i$ to $\mathbf{P}_{k-1}^i$.
- For all previously initialized APs, and all peaks:
  - Update peaks' parameters by using eqs. (F.2)-(F.6);
  - Compute the unnormalized $u_{p,k}$ of eq. (5.36) like in App. F.
- Normalize the $u_{p,k}$ over p to obtain $\tilde{u}_{p,k}$ (eq. (5.38)).
- Compute a factor I_W^i and the new $\mathbf{h}$ distribution (eqs. (5.19) and (5.21) respectively, with the integrals approximated like in eq. (5.39)) for each processed AP.
- Update the particle weights as in eq. (5.25), where I_M^i are computed as in FootSLAM [RAK09, eq. (5)].
- Decide if any detected but not yet employed AP should be processed and, if so, initialize new APs' posterior

$$p\left(\mathbf{W}|\mathbf{P}_{0:k}, \mathbf{Z}_{1:k}^W, \mathbf{h}\right)$$

  by applying the algorithm in Fig. 5.14.
- Update the map $\mathbf{M}$ as in FootSLAM [RAK09, eq. (4)].

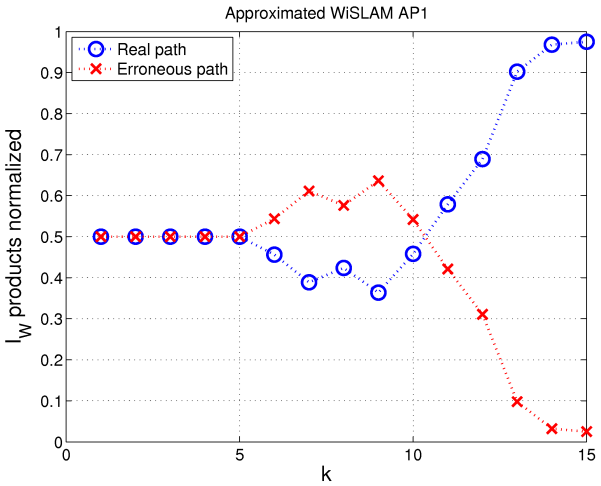


Figure 5.18 Competing paths: products (normalized) of the I_W terms for both paths in Fig. 5.9, averaged on 10 datasets, in the cases of (a) only AP 1 involved and (b) both APs involved. Plot (c) shows the performance obtained by the approximated WiSLAM in the same case as in (a) (algorithm started after 5 measurements). The line related to the real path is dotted with blue circles.

- Resampling can be performed if required.

5.7.6 Preliminary results

Simplified WiSLAM's effectiveness has been supported by our experiments: as an example, we show in Fig. 5.18 the results in the same case as in Fig. 5.13.a, relative to the average over 10 independent datasets. Here, we start the algorithm at $k = 5$ and we can see that with a little delay the expected performance is achieved.

We also show in Fig. 5.19 the performance of both versions of WiSLAM over a single dataset, to highlight that the simplified algorithm consistency: even the emitted power probabilities share

the same ongoing behavior.

5.7.7 Final results

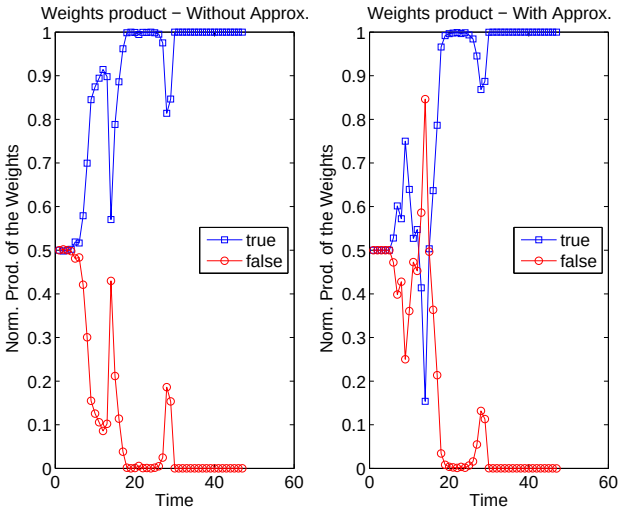
The full algorithm (in its approximated version) was run in the testbed of Fig. 5.9 over two independent walks of about 5 minutes duration each, with the number of particles from about 20.000 to 50.000. The results are shown in Fig. 5.20 where the comparison with the ground truth (known building layout) shows the accuracy of the estimated FootSLAM map. Also the final estimated positions of the APs are remarkably close to the true ones.

We have also tested the algorithm employing *only the particle weights due to the WiFi measurements*, i.e. we set a constant I_M^i in eq. (5.25), meaning that the FootSLAM update was not being used in the weight. By observing Fig. 5.21 one verifies that this algorithm, whose weighting relies only on the RSS measurements from two APs, converges to the true map, even if the positioning of one of the APs is less accurate. This implies that WiSLAM can be a useful approach even when FootSLAM will be expected to experience problems in convergence, such as in buildings with large open spaces, or few true loops during the walk.

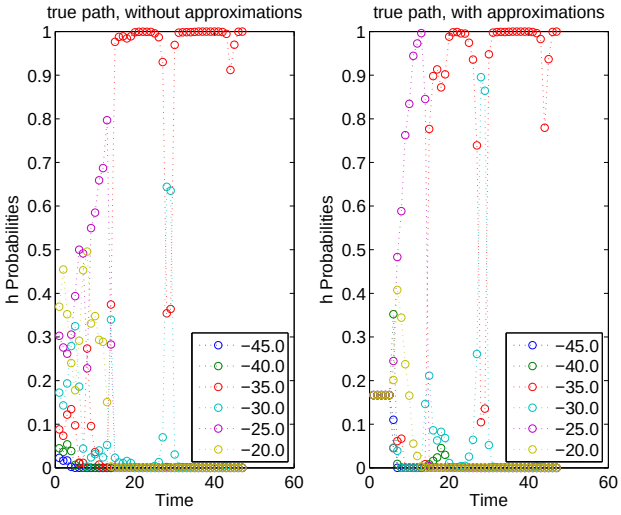
A further experiment employing the full algorithm and 4 APs is shown in Fig. 5.22, where an almost perfect fit between estimated and true APs' positions is achieved.

5.8 A further explanation for WiSLAM

FootSLAM, PlaceSLAM and WiSLAM provide optimal algorithms in Bayesian sense; nevertheless, this does not ensure good performances nor convergence. The following analysis does not pretend to be rigorous, but represents an attempt to answer the following question: why do they work in the end?

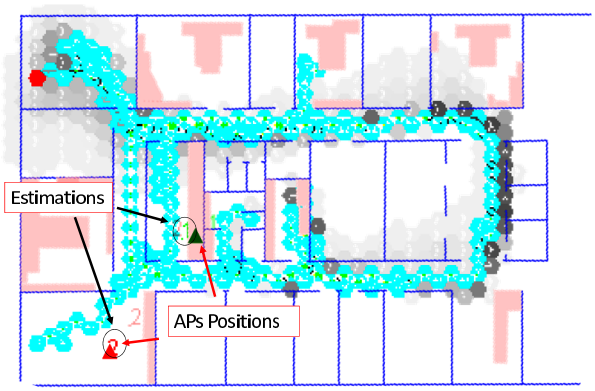


(a) Weights product

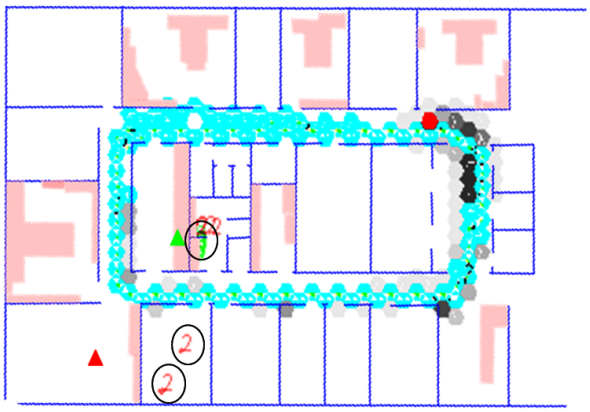


(b) Hypotheses Probabilities

Figure 5.19 Experimental validation: products of the weights related to a single dataset.



(a) Walk 1



(b) Walk 2

Figure 5.20 Two maps learnt by using full WiSLAM. Shown are an overlap of the posterior (i.e. weighed average) map (in shades of grey) and the more useful MAP map (i.e. the map of the 'best' particle) - blue. Hexagons which the pedestrian has visited more often are more open, unvisited ones are not shown. More white within the hexagons means more frequently crossed transitions. Triangles indicate the real AP positions, while the circles highlight their estimations. Parameters: 9 power hypotheses 3-dB spaced, $\sigma = 5$ dB, $N_{Peaks} = 14$, (a) 50000 and (b) 20000 particles.

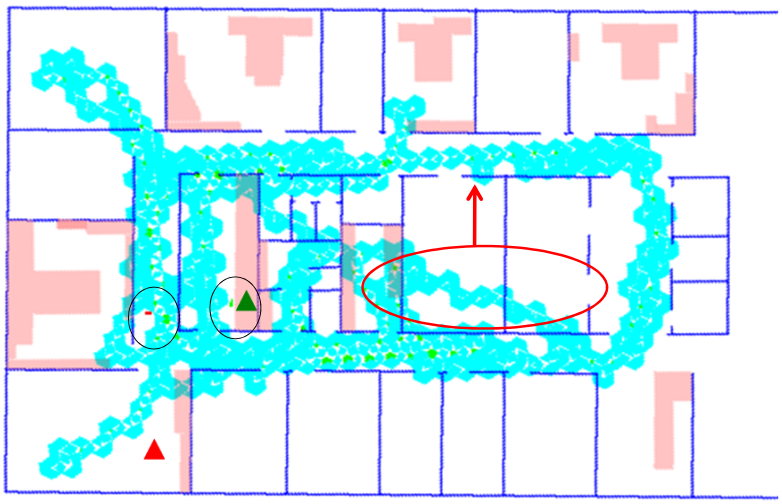


Figure 5.21 Map learnt by using WiSLAM with I_M constant (best particle only). Triangles indicate the real AP positions, while the circles highlight their estimations. The main error is indicated by an oval - the arrow showing roughly the error vector. The parameters are the same as in Fig. 5.20.a.

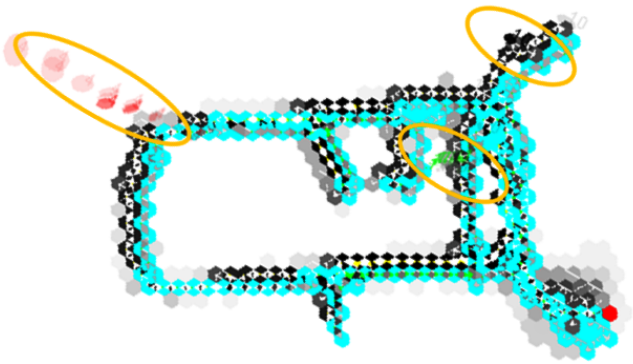


Figure 5.22 Map learnt by using WiSLAM with 4 APs. The AP positions and estimates are remarked by yellow ovals, while the circles highlight their estimations. The parameters are the same as in Fig. 5.20.a.

FootSLAM

In [RAK10] authors report a connection between 'best weighted' particles and the entropy of the map distribution: in particular they remark that lower entropy maps are automatically preferred by the algorithm rather than the others, even if a formal analysis is missing. This is conceptually sounding because the entropy of the map at k can be seen like the residual uncertainty on its estimation after k measurements. In other words, the algorithm would be looking for particles which extract as information as possible from data, minimizing the entropy of the estimated map (recall it is a pdf).

Following [SGB05], the entropy $H(x, m|z)$ for the joint estimation of a state vector x and map m for a RBPF approach given measurements z results in

$$H(m, x|z) = H(x|z) + E_{x|z}\left[H(m|x, z)\right], \tag{5.40}$$

that accounts for the entropy of the state (first term on the right side) and the entropy of the map averaged on state (second term).

Indeed the situation is quite more complex than it can appear: the movement model acts like a constraint (or better a cost function) such that the state space in which the entropy should be minimized is reduced in size. This avoids 'too' concentrated maps which are solutions of the unconstrained minimization: it would be trivial in fact to find that the 'back-and-forth' map, allowing only a transition between two adjacent hexagons, has the lowest entropy at all.

PlaceSLAM and WiSLAM

In PlaceSLAM this concept is going to fail: first, adding new places to the map is inherently a high entropy solution, but sometimes it is required. Second, even when the algorithm decides to update an old place, its new map results in the normalized product of two bivariate Gaussians (the old map and the new measurement pdf). The result is a new bivariate Gaussian whose entropy is related to the covariance matrix, which depends in turn on the covariance

matrices of the starting Gaussians; in other words the entropy for all particles' maps evolve in the same way. Their weights, instead, are different, being them the *unnormalized* product's integral and thus the algorithm works the same. For this reason the 'engine' in PlaceSLAM is not the minimization of the map entropy, but rather the 'matching' of the new measurement to the old map.

WiSLAM is conceptually very similar to PlaceSLAM. It only differs in the shape of the distributions, as the map is a product of k non-concentric 'donut' shaped pdfs and the new measurement likelihood is a further 'donut'. This brings to difficult evaluations of the entropy, but if we use the approximations of simplified WiSLAM, we have the same considerations as in PlaceSLAM.

The difference from FootSLAM is that the new place or RSS measurement is not involved in the state sampling but only in the weight. As a result the trajectory hypothesis is due to step measurements, and the other ones are used in the weighting.

Let us focus on the weights. Using the notation of eq. 5.40, PlaceSLAM and WiSLAM weights can be expressed in a unified way

$$\begin{aligned} I_k &= \int_m p\left(z_k|x_{0:k},m\right) p\left(m|x_{0:k-1},z_{1:k-1}\right) dm \\ &= < p\left(z_k|x_{0:k},m\right), p\left(m|x_{0:k-1}\right) >, \end{aligned} \tag{5.41}$$

where $< \cdot,\cdot >$ denotes the inner product in the pdf functional space. This implies that the weights follow a similarity score. Furthermore, taking the logarithm and adopting the Jensen inequality, we find

$$\begin{aligned} \log I_k &= \log E_{m|x_{0:k-1},z_{1:k-1}}\left[p\left(z_k|x_{0:k},m\right)\right] \\ &\geq E_{m|x^i_{0:k-1},z^W_{1:k-1}}\left[\log p\left(z^W_k|x^i_{0:k},m\right)\right] \\ &= -D\left(p(m|x^i_{0:k-1},z^W_{1:k-1})//p\left(z_k|x_{0:k},m\right)\right) \\ &\quad -H\left(p(m|x_{0:k-1},z_{1:k-1})\right), \end{aligned} \tag{5.42}$$

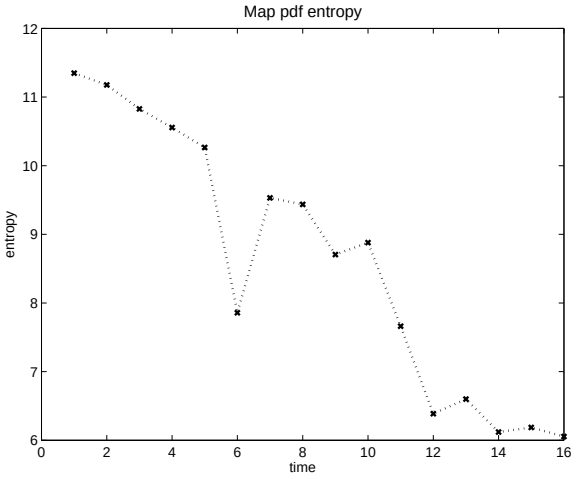
where $D(\cdot\,//\cdot)$ and $H(\cdot)$ denote the Kullback Leibler (KL) distance and the entropy for continuous variables respectively [CT01].

Consequently, we can state that a particle is assigned a high weight in the cases of:

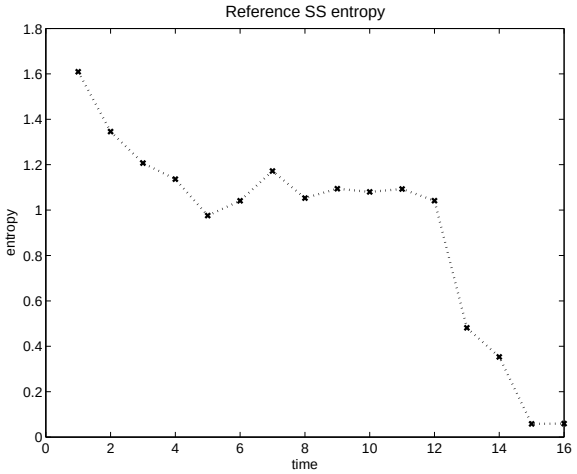
- low KL distance between the old map and the new RSS likelihood function, that is an entropy based matching criterium,
- low entropy of the old map.

This means that, besides the matching criterium, a low entropy rule for the map holds, i.e. narrower maps are chosen. It finds an intuitive explanation by considering that the map pdf represents the residual uncertainty over the APs' position, that is measured by its entropy: minimizing this entropy means to extract as more information about the map as possible from the RSS measurements.

Referring to WiSLAM, in Fig. 5.23 we show the entropy evolution of the pdf of the **W** map agaist the time (in Fig. 5.23.a, we report only a numerical approximation) and the **h** discrete distribution (Fig. 5.23.b) obtained from a set of simulated data. Although some variability is present, their tendency to decrease is clearly shown. Finally, we notice that this is consistent with the entropy based considerations made in other papers about RBPF SLAM, in particular in [SGB05], where a general expression for the (floor) map entropy is obtained, and in [RAK10] about PlaceSLAM.



(a) Map entropy



(b) **h** entropy

Figure 5.23 Entropy of (a) the **W** map and (b) the **h** distribution for a simulated dataset; the values in (a) are numerical approximations.

Conclusion

This thesis offers a theoretical and experimental analysis of some relevant aspects concerning indoor WLAN-based positioning from the Bayesian perspective. Our approach aims to avoid expensive ad-hoc hardware, but rather to extract all the information needed from the received power measurements already available within WLANs for the communication task.

Our main claim is that the indoor environment is a source of precious information that greatly improves the accuracy of all locating systems. In fact, several properties of the environment are heavily variable among different testbeds and in time: accounting for those ones affecting the positioning accuracy is essential in order to design high accuracy systems. Also, opportunity signals can be profitably used, even though some care is necessary.

We choose three cases of interest. The first one requires algorithms which can detect and follow the variations of some environmental properties; to this aim the signal propagation has been suitably modeled and the variability concerns some parameters like the power emitted by the transmitters and the power loss due to the signal propagation across the space.

The second scenario of interest finds its relevance in the high diffusion of WLAN networks, several of which can co-exist into the same area or even the same building. The consequent availability of measurements of signals emitted by unmanaged networks, whose base stations' position and properties are unknown, is a source of information about positioning that is not exploited by the classical techniques. Therefore, we suggest a framework in which these measurements are employed with a degree of reliability that

varies according to the knowledge available on the transmitters.

The last case considered looks at the building map as a set of constraints which, if used, can tighten up the accuracy. Although some attempts employing stand alone WLANs were made, it is much more profitable to merge power measurements from WLANs with other sensors' outputs. The consequent framework based on data fusion is of extreme interest for both the theoretical challenges offered and the actual improvements yielded.

All the previously described topics are met by means of the Bayesian theory since this latter is a powerful and elegant framework which conjugates the theoretical search for optimal solutions with a strong physical interpretation of the variables involved. The results, in terms of positioning accuracy and robustness, are always validated through extensive computer and real world experiments, showing the effectiveness of the proposed solutions.

Appendix A

RSS likelihood models

A.1 Rice fading

In the Rice fading model the fading envelope r_R is described by a *Rice distribution* (or *n-Nakagami*) [SA05]

$$p\left(r_R\right) = \frac{r_R}{\sigma^2} \exp\left(-\frac{\left(r_R^2 + V^2\right)}{2\sigma^2}\right) I_0\left(\frac{r_R V}{\sigma^2}\right), \quad r_R \geq 0,$$

in which V is the amplitude of the Line-of-Sight component, σ_R^2 is the variance of the two components of a complex scattered contribution and $I_0\left(\cdot\right)$ is the modified Bessel function of zeroth order. An alternative expression is in terms of the mean square value $E\left[r_R^2\right] = V^2 + 2\sigma_R^2$ and of the *Rice factor* $K_f \triangleq V^2/2\sigma_R^2$ that quantifies the ratio among the power of the two components:

$$p\left(r_R\right) = \frac{2\left(1 + K_f\right) r_R}{E\left[r_R^2\right]} \exp\left(-K_f - \frac{\left(K_f + 1\right) r_R^2}{E\left[r_R^2\right]}\right) \times I_0\left(2 r_R \sqrt{\frac{K_f\left(K_f + 1\right)}{E\left[r_R^2\right]}}\right), \quad r_R \geq 0.$$

The power observation (in natural scale) is given by

$$y\left(t\right) = s\left(t\right) = \left|r_R\left(t\right)\right|^2$$

and the AF is [SA05, [p. 23]]

$$AF = \frac{1+2K}{(1+K)^2}.$$

A.2 Lognormal fading

A.2.1 Lognormal distribution for powers

In the Lognormal fading the power observations in linear scale $s(d)$ are described by a Lognormal distribution [SA05]

$$p_{S_L}(s_L) = \frac{\xi}{\sqrt{2\pi}\sigma s_L} \exp\left(-\frac{(10\log_{10} s_L - \mu_{dB})^2}{2\sigma_{dB}^2}\right),$$

where $\xi = \frac{10}{\ln 10}$. For the powers in deciBels $Z_L = g(S_L) = 10\log_{10} S_L$ one can find

$$\begin{aligned} p_{Z_L}(z_L) &= \frac{\xi}{\sqrt{2\pi}\sigma_{dB} s_L |g'(s_L)|} \exp\left(-\frac{(10\log_{10} s_L - \mu_{dB})^2}{2\sigma_{dB}^2}\right)\Bigg|_{s_L = 10^{\frac{z_L}{10}}} \\ &\overset{(a)}{=} \frac{\xi}{\sqrt{2\pi}\sigma_{dB} s_L \xi \frac{1}{s_L}} \exp\left(-\frac{(z_L - \mu_{dB})^2}{2\sigma_{dB}^2}\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma_{dB}} \exp\left(-\frac{(z_L - \mu_{dB})^2}{2\sigma_{dB}^2}\right) \sim \mathcal{N}(\mu_{dB}, \sigma_{dB}) \end{aligned}$$

where the (a) follows from

$$g'(s_L) = (\xi \ln s_L)' = \xi \frac{1}{s_L}.$$

Normally we estimate, for example in a calibration stage, the mean power in dB μ_{dB} and the SNR, which allow the computation of σ_{dB}. In fact, the value of σ_{dB} can be achieved by putting the

Lognormal distribution of S_L in its standard form

$$p_{S_L}(s_L) = \frac{\xi}{\sqrt{2\pi}\sigma_{dB}s_L}\exp\left(-\frac{(10\log_{10}s_L-\mu_{dB})^2}{2\sigma_{dB}^2}\right)$$
$$= \frac{1}{\sqrt{2\pi}\frac{\sigma_{dB}}{\xi}s_L}\exp\left(-\frac{\left(\ln s_L-\frac{\mu_{dB}}{\xi}\right)^2}{2\left(\frac{\sigma_{dB}}{\xi}\right)^2}\right)$$

that highlights the SNR

$$SNR = E[S_L] = \exp\left(\frac{\mu_{dB}}{\xi}+\frac{\left(\frac{\sigma_{dB}}{\xi}\right)^2}{2}\right)$$

and thus the log mean

$$SNR_{dB} = 10\log_{10}E[S_L] = 10\log_{10}\exp\left(\frac{\mu_{dB}}{\xi}+\frac{\left(\frac{\sigma_{dB}}{\xi}\right)^2}{2}\right)$$
$$= \xi\ln\exp\left(\frac{\mu_{dB}}{\xi}+\frac{\left(\frac{\sigma_{dB}}{\xi}\right)^2}{2}\right) = \mu_{dB}+\frac{\sigma_{dB}^2}{2\xi}$$

$$\sigma_{dB} = 2\xi\sqrt{SNR_{dB}-\mu_{dB}}.$$

The power μ_{dB} used to define the $E[r_L^2]$ is again calculated at a reference point; for d_0, one has

$$\mu_{dB} = P_0.$$

The AF is [SA05, [p. 33]]

$$AF = \exp\left(\frac{\sigma_{dB}^2}{\xi^2}\right)-1 = \exp\left[2\xi\left(SNR_{dB}-\mu_{dB}\right)\right]-1.$$

A.2.2 Lognormal distribution for amplitudes

Similarly, the Lognormal distribution is also used to describe the amplitudes of the signal [Has93]. In this case $r_L = \sqrt{s_L(d)}$ is characterized by the pdf

$$p_{R_L}(r_L) = \frac{2\xi}{\sqrt{2\pi}\sigma_{dB} r_L} \exp\left(-\frac{(20\log_{10} r_L - \mu_{dB})^2}{2\sigma_{dB}^2}\right)$$

so that the power calculated in dB assumes the form $Z = g(R_L) = 20\log_{10} r_L$, with a pdf

$$\begin{aligned} p_R(r_L) &= \frac{2\xi}{\sqrt{2\pi}\sigma_{dB} r_L |g'(r_L)|} \exp\left(-\frac{(20\log_{10} r_L - \mu_{dB})^2}{2\sigma_{dB}^2}\right)\Bigg|_{r_L = 10^{\frac{x}{10}}} \\ &\overset{(a)}{=} \frac{2\xi}{\sqrt{2\pi}\sigma_{dB} r_L 2\xi \frac{1}{r_L}} \exp\left(-\frac{(x-\mu_{dB})^2}{2\sigma_{dB}^2}\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma_{dB}} \exp\left(-\frac{(x-\mu_{dB})^2}{2\sigma_{dB}^2}\right) \sim \mathcal{N}(\mu_{dB}, \sigma_{dB}) \end{aligned}$$

where the (a) follows from

$$g'(r_L) = (2\xi \ln r_L)' = 2\xi \frac{1}{r_L}.$$

The value of σ_{dB} can be achieved by putting the Lognormal distribution of r_L in the standard form

$$\begin{aligned} p_{R_L}(r_L) &= \frac{2\xi}{\sqrt{2\pi}\sigma_{dB} r_L} \exp\left(-\frac{(20\log_{10} r_L - \mu_{dB})^2}{2\sigma_{dB}^2}\right) \\ &= \frac{1}{\sqrt{2\pi}\frac{\sigma_{dB}}{2\xi} r_L} \exp\left(-\frac{\left(\ln s_R - \frac{\mu_{dB}}{2\xi}\right)^2}{2\left(\frac{\sigma_{dB}}{2\xi}\right)^2}\right) \end{aligned}$$

By using [SA05, [p. 33]]

$$E\left[R_L^k\right] = \exp\left(\frac{k}{2\xi}\mu_{dB} + \frac{1}{2}\left(\frac{k}{2\xi}\right)^2 \sigma_{dB}^2\right)$$

one has

$$E\left[R_L^2\right] = \exp\left(\frac{2}{2\xi}\mu_{dB} + \frac{1}{2}\left(\frac{2}{2\xi}\right)^2 \sigma_{dB}^2\right) = \exp\left(\frac{\mu_{dB}}{\xi} + \frac{1}{2}\left(\frac{\sigma_{dB}}{\xi}\right)^2\right)$$

and thus

$$SNR_{dB} = 10\log_{10} E\left[R_L^2\right] = 10\log_{10}\exp\left(\frac{\mu_{dB}}{\xi} + \frac{\left(\frac{\sigma_{dB}}{\xi}\right)^2}{2}\right)$$

$$= \xi \ln\exp\left(\frac{\mu_{dB}}{\xi} + \frac{\left(\frac{\sigma_{dB}}{\xi}\right)^2}{2}\right) = \mu_{dB} + \frac{\sigma_{dB}^2}{2\xi}$$

The power μ_{dB} used to define the $E\left[r_L^2\right]$ is again calculated at a reference point; choosing d_0, one has

$$\mu_{dB} = P_0.$$

and thus, like before,

$$\sigma_{dB} = 2\xi\sqrt{E\left[r_L^2\right] - \mu_{dB}}.$$

The AF can be easily computed in

$$AF_R = \exp\left[2\xi\left(SNR_{dB} - \mu_{dB}\right)\right] - 1.$$

This shows that the same performance figures are achieved by modeling either the amplitudes or the powers like Lognormal r.v., provided the same calibration of the power parameters in dB.

A.3 Comparison of Rice and Lognormal

We need to build some rule to properly generate rician and Lognormal random variables, in order to compare localization algorithm

performance for both fading kinds. Moreover we know that we have fading models that describe both the power in dB Z and the channel fading amplitude R that are related by the relationship:

$$Z = \zeta \ln R \Leftrightarrow R = \exp\left(\frac{Z}{\zeta}\right) \tag{A.1}$$

where $\zeta = 20/\ln 10 = 2\xi$.

It is well known that when the shadowing phenomenon is predominant, the power in dB (say it Z_L) is described by a normal random variable with mean μ_{dB} and variance σ^2_{dB}, i.e. the pdf:

$$p_{Z_L}(z_L) = \frac{1}{\sigma_{dB}\sqrt{2\pi}} \exp\left(-\frac{(z_L - \mu_{dB})^2}{2\sigma^2_{dB}}\right) \tag{A.2}$$

the channel fading amplitude is well described by a Lognormal random variable R_L whose pdf is

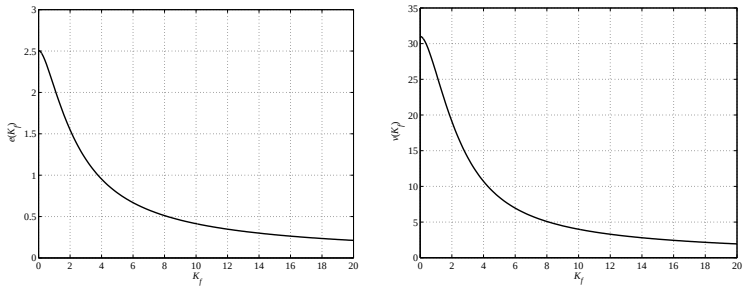
$$p_{R_L}(r_L) = \frac{\zeta}{r_L \sigma_{dB}\sqrt{2\pi}} \exp\left(-\frac{(\zeta \ln r_L - \mu_{dB})^2}{2\sigma^2_{dB}}\right). \tag{A.3}$$

Instead channel fading amplitude due to multipath is well described by a rician random variable R_R, whose pdf is:

$$p_{R_R}(r_R) = \frac{2(K_f+1)r_R}{\Omega} \exp\left(-\frac{(K_f+1)r_R^2}{\Omega} - K_f\right) I_0\left(2r\sqrt{\frac{K_f(K_f+1)}{\Omega}}\right). \tag{A.4}$$

In this formulation K_f is the *K-factor*, while Ω is the mean power in watt, i.e. $\Omega = E[R_R^2]$. The power in dB, by using the equation (A.1), can be indicated by a r.v. Z_R that has density function

$$p_{Z_R}(z_R) = \frac{2(K_f+1)}{\zeta\Omega} \exp\left(-\frac{(K_f+1)\exp\left(\frac{2z_R}{\zeta}\right)}{\Omega} - K_f + \frac{2z_R}{\zeta}\right) I_0\left(2\exp\left(\frac{z_R}{\zeta}\right)\sqrt{\frac{K_f(K_f+1)}{\Omega}}\right). \tag{A.5}$$



(a) (b)

Figure A.1 Plot of functions $e(K_f)$ (a) and $v(K_f)$ (b) vs. K_f.

If we convert Ω in dB, i.e. $\Omega_{dB} \triangleq (\zeta/2)\ln(\Omega)$, we can perform the substitution $\Omega \rightarrow \exp(2\Omega_{dB}/\zeta)$ in eq. (A.5) and we obtain

$$p_{Z_R}(z_R) = \frac{2(K_f+1)}{\zeta}\exp\left(-\frac{(K_f+1)\exp\left[\frac{2(z_R-\Omega_{dB})}{\zeta}\right]}{\Omega} - K_f\right. \tag{A.6}$$

$$\left. + \frac{2(z_R-\Omega_{dB})}{\zeta}\right) I_0\left(2\exp\left[\frac{(z_R-\Omega_{dB})}{\zeta}\right]\sqrt{K_f(K_f+1)}\right). \tag{A.7}$$

By inspection it is evident that Ω_{dB} is a shift parameter. Thus we can say that

$$E[Z_R] = \Omega_{dB} - e(K_f);$$
$$VAR[Z_R] = v(K_f).$$

Functions $e(K_f)$ and $v(K_f)$ can be numerically computed for $K_f \in [0, 20]$, because this range covers the typical value of the K factor (see also [GME99]), and they are shown in Fig. A.1.

A.3.1 From Decibel to Watt Domain

Let's introduce the coefficient of variation $c(\cdot)$, whose squared value is defined as

$$c^2(\cdot) = \frac{VAR[\cdot]}{E^2[\cdot]}. \tag{A.8}$$

The issue is to generate Lognormal and rician random variables such that their behavior in dB, is comparable. Thus, dealing with the random variables Z_L and Z_R, whose pdf is defined in eq.(A.2) and (A.7) respectively, we need that $c(Z_L) = c(Z_R)$ in each point of the surveilled area. Another constrain of the problem is that the mean power of e.m. signal in dB follow the (simplified) decay rule

$$E[P(d)] = E[Z] = P_0 - \frac{\zeta\alpha}{2}\ln(d), \tag{A.9}$$

where d is the distance between the *access point* (AP) and the *mobile user* (MU), α is the decay coefficient and P_0 is the power for $d = d_0 = 1$. Instead there is no known law that describes the power variance behaviour as function of the distance between AP and MU. Thus, for sake of simplicity, we will perform the comparison by setting to a constant the power variance on the whole surface, while the mean power will follow eq.(A.9). This idea is simple to apply for the gaussian random variable Z_L. Instead, when we want to simulate the power of the e.m. field emitted by an AP that is affected by multipath (i.e. Z_R), in order to match some value of mean μ_{dB} and variance σ^2_{dB} in dB, the procedure will be:

- compute $K_f = v^{-1}(\sigma^2_{dB})$;
- compute $\Omega_{dB} = \mu_{dB} + e(K_f)$;
- compute $\Omega = \exp(2\ \Omega_{dB}/\zeta)$.

It is important to point out that, when comparing Z_L and Z_R, we have to fix an upper bound for the variance in dB, that is:

$$\max \sigma^2_{dB} = v(K_f)|_{K_f=0} \approx 31.025. \tag{A.10}$$

Now the question is: what happens if we apply this procedure and then we want to compute the *Amount of Fading* (AF), that is tipically defined in Watt? We know (see [SA05]) that, if we indicate with S the power in watt, the AF is defined as

$$AF = \frac{VAR[S]}{E^2[S]}, \tag{A.11}$$

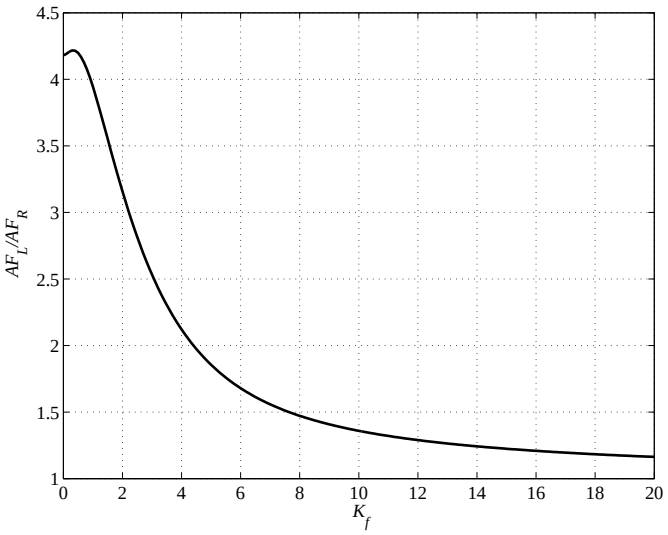


Figure A.2 Plot of the ratio AF_L/AF_R vs. K_f.

i.e. it is the square value of the coefficient of variation computed in watt. Moreover we know that, for a rician random variable R_R with pdf in eq.(A.4), the Amount of Fading is

$$AF_R = \frac{1 + 2K_f}{(1 + K_f)^2}, \tag{A.12}$$

while for a log-normal random variable R_L with pdf in eq.(A.3) it is

$$AF_L = \exp\left(\frac{4\sigma_{dB}^2}{\zeta^2}\right) - 1. \tag{A.13}$$

So we can compare AF_R and AF_L in the hypothesis that there is the relationship $\sigma_{dB}^2 = v(K_f)$ and, as shown in Fig. A.2, we have that $AF_L > AF_R$. Their ratio is close to 1 only for large values of K_f (or small values of σ_{dB}^2), i.e. if they are close to zero. Moreover we notice that while in the Rician case $AF_R \in [0, 1]$, in the Lognormal one AF_L is unbounded. In order to have a view of pdf shapes obtained by means of this method of comparison, let's give a look to Fig. A.3.

A.3.2 From Watt to Decibel Domain

In the previous section we have determined a simple procedure to compute the Rician parameters in order to have a fixed coefficient of variation in dB, for comparing, for example, the performance of a localization algorithm under multipath and under shadowing. In this section we want to replicate the same procedure in order that multipath and shadowing will produce the same AF. So the idea is to express σ^2_{dB} as function of K_f by putting $AF_R = AF_L$. The result is

$$\sigma^2_{dB} = \frac{\zeta^2}{4} \ln \left[1 + \frac{(1+2K_f)}{(1+K_f)^2} \right], \tag{A.14}$$

and the behaviour is shown in Fig. A.4. It is worth noting that also in this case we have an upper bound for $K_f \to 0$, i.e. $\max \sigma^2_{dB} = \frac{\zeta^2}{4} \ln(2) \approx 13.07$. Moreover we have to put equal also the mean value of the power in Watt, i.e. $E[R^2_L] = E[R^2_R]$. So we have

$$\Omega = \exp\left(\frac{2\mu_{dB}}{\zeta} + \frac{2\sigma^2_{dB}}{\zeta^2} \right), \tag{A.15}$$

that can be also expressed as

$$\Omega_{dB} = \mu_{dB} + \frac{\sigma^2_{dB}}{\zeta}. \tag{A.16}$$

Thus, when we want to simulate e.m. fields affected by multipath or shadowing with the same AF, the procedure will be:

- compute σ^2_{dB} as a function of K_f (or vice versa) according to eq.(A.14);
- compute $\Omega_{dB} = \frac{\zeta}{2} \ln(\Omega)$;
- compute $\mu_{dB} = \Omega_{dB} - \frac{\sigma^2_{dB}}{\zeta}$.

Now the question is: what happens to the square value of the coefficient of variation $c^2(\cdot)$ defined in eq.(A.8)? We have, by using eq.(A.16) that

$$c^2(Z_L) = \frac{\sigma^2_{dB}}{\mu^2_{dB}} = \frac{\sigma^2_{dB}}{\left(\Omega_{dB} - \frac{\sigma^2_{dB}}{\zeta}\right)^2}, \tag{A.17}$$

and that

$$c^2(Z_R) = \frac{v(K_f)}{(\Omega_{dB} - e(K_f))^2}. \tag{A.18}$$

Thus, also remembering eq.(A.14), the two coefficients will be function of both K_f and Ω_{dB}. In Fig. A.5 we show the behaviour of these coefficients through the ratio $c^2(Z_L)/c^2(Z_R)$. It is evident that there is a great variation for small absolute values of Ω_{dB}, while for big values the ratio saturates towards a limit behaviour. Also in this case we show several pdf shapes obtained by means of this method of comparison (see Fig. A.6). We point out that the plots for $\sigma^2_{dB} = 20$ are missing, because in this case the upper bound for the variance in dB is $\sigma^2_{dB} \approx 13.07$.

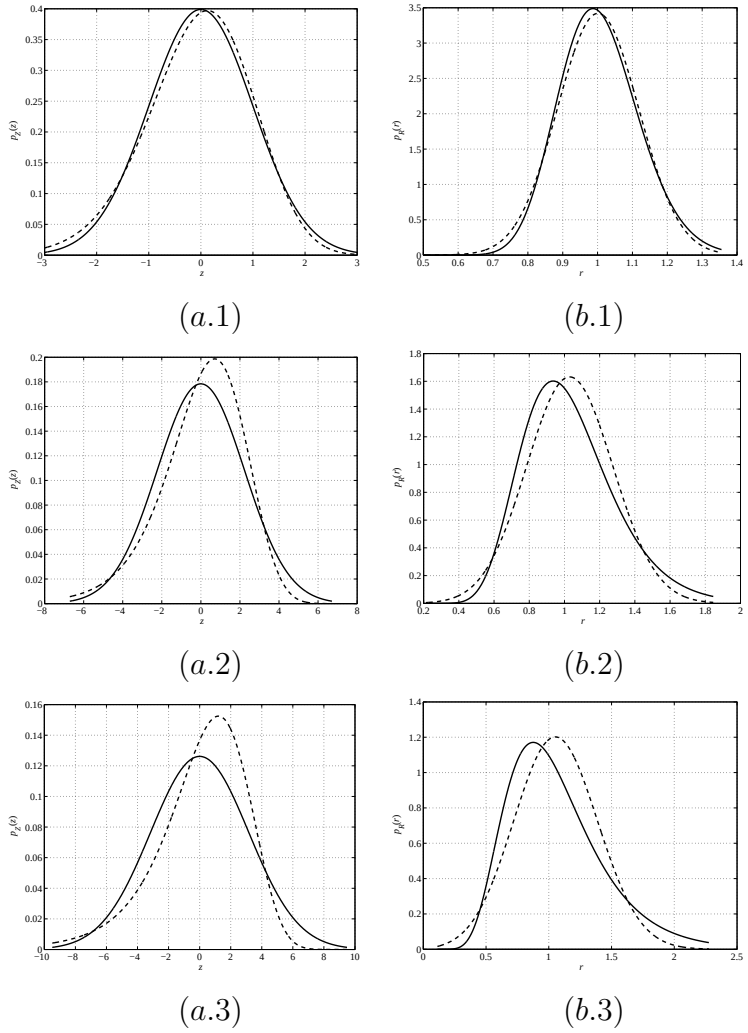


Figure A.3 Equal $c(\cdot)$ in dB. Plots of Lognormal (continuous line) and rician (dotted line) pdf in dB (a.*) and in the amplitude domain (b.*) for different values of σ^2_{dB}. In particular $\sigma^2_{dB} = 1$ (*.1), $\sigma^2_{dB} = 5$ (*.2) and $\sigma^2_{dB} = 10$ (*.3). The mean in dB is set to $\mu_{dB} = 0$.

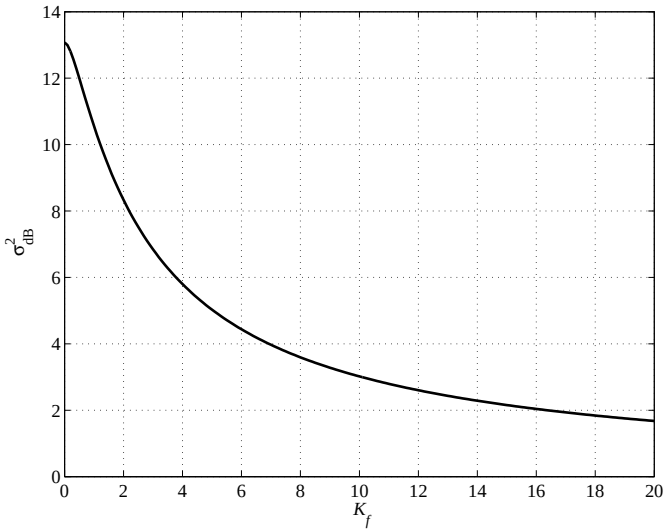


Figure A.4 Plot σ^2_{dB} vs. K_f (solid line) computed by putting $AF_R = AF_L$.

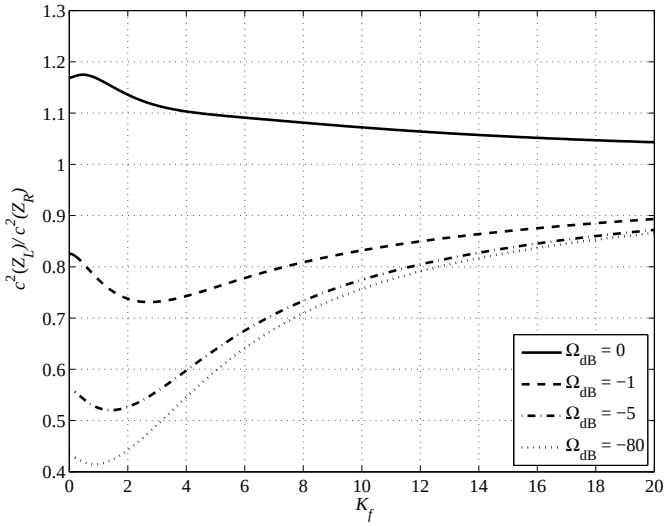


Figure A.5 Plot of the ratio $c^2(Z_L)/c^2(Z_R)$ vs. K_f for several values of Ω_{dB}.

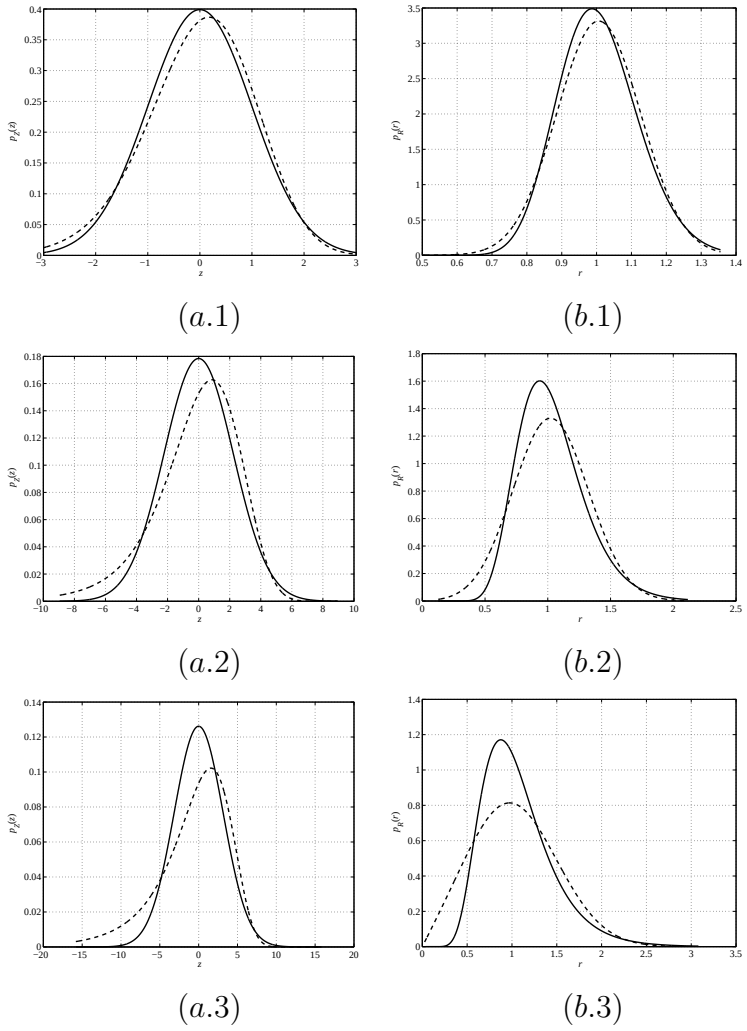


Figure A.6 Equal AF in Watt. Plots of Lognormal (continuous line) and rician (dotted line) pdf in dB (a.*) and in the amplitude domain (b.*) for different values of σ^2_{dB}. In particular $\sigma^2_{dB} = 1$ (*.1), $\sigma^2_{dB} = 5$ (*.2) and $\sigma^2_{dB} = 10$ (*.3). The mean in dB is set to $\mu_{dB} = 0$.

Appendix B

Matricial notation

Let $f(\mathbf{x})$ be a real function and $\mathbf{x} = [x_1, ..., x_d]^T$ a d dimensional column vector. We define the gradient of $f(\mathbf{x})$ like

$$\nabla_{\mathbf{x}} f = \frac{\partial f}{\partial \mathbf{x}} = \left[\frac{\partial f}{\partial x_1}, ..., \frac{\partial f}{\partial x_d} \right]^T . \tag{B.1}$$

If we define another column vector $\mathbf{y}$ and a matrix A of adequate dimensions, the following relations hold

$$\frac{\partial}{\partial \mathbf{x}} \left[\mathbf{y}^T \mathbf{x} \right] = \frac{\partial}{\partial \mathbf{x}} \left[\mathbf{x}^T \mathbf{y} \right] = \mathbf{y}; \tag{B.2}$$

$$\frac{\partial}{\partial \mathbf{x}} \left[A\mathbf{x} \right] = \frac{\partial}{\partial \mathbf{x}} \left[\mathbf{x}^T A^T \right] = A; \tag{B.3}$$

$$\frac{\partial}{\partial \mathbf{x}} \left[\mathbf{x}^T A \mathbf{x} \right] = \left[A + A^T \right] \mathbf{x}. \tag{B.4}$$

This last amounts in $2A\mathbf{x}$ in the case of symmetric A.

Let now $F(\mathbf{x}) = [f_1(\mathbf{x}), ..., f_n(\mathbf{x})]^T$ be a n dimensional function of $\mathbf{x}$. The Jacobian of $F(\mathbf{x})$ is the matrix of size n x d defined like:

$$\nabla_{\mathbf{x}} F = \begin{pmatrix} \partial f_1/\partial x_1 & ... & \partial f_1/\partial x_d \\ \vdots & \ddots & \vdots \\ \partial f_n/\partial x_1 & ... & \partial f_n/\partial x_d \end{pmatrix}.$$

Appendix C

Variance filter

The variance filter of interest is defined in [Hay01] and, among the others, in [Nel00]. Let us consider the scalar case along with the maximum likelihood score function, here defined for a set of N observations

$$J^{ML}(\sigma^2) = -\sum_{k=1}^{N}\left(\log(2\pi\sigma^2_{\epsilon_k}) + \frac{(\epsilon_k)^2}{\sigma^2_{\epsilon_k}}\right), \qquad \text{(C.1)}$$

where $\epsilon_k = y_k - g(\hat{\mathbf{x}}_{k|k-1})$ is the prediction error and its variance $\sigma^2_{\epsilon_k}$ is dependent upon σ^2.

Given a starting estimation of the variance $\hat{\sigma}^2_0 = E[\sigma^2]$, with $q_0 = E[(\sigma^2 - \hat{\sigma}^2_0)^2]$, we perform a running estimation of σ^2 as in Kalman, composed at the generic instant $k > 0$ of two steps:

- a prediction step

$$\begin{aligned} \hat{\sigma}^2_{k|k-1} &= \hat{\sigma}^2_{k-1|k-1}, \\ q_{k|k-1} &= \lambda^{-1} q_{k-1|k-1}, \end{aligned}$$

  where λ is the forgetting factor;

- an update step

$$\begin{aligned} q_{k|k-1} &= \left(q^{-1}_{k|k-1} + H_k^T \sigma_r^{-2} H_k\right), \\ \hat{\sigma}^2_{k|k} &= \hat{\sigma}^2_{k|k-1} + q_{k|k} H_k^T \sigma_r^{-2} e_k. \end{aligned}$$

In the previous for $k = 1$ we assume $\sigma^2_{k|k-1} = \sigma^2_0$ and $q_{k|k-1} = q_0$, and the maximum likelihood goal leads to

$$e_k = \begin{bmatrix} \sqrt{l_{\epsilon,k}} \\ \sigma_{\epsilon,k}\epsilon_k \end{bmatrix},$$

$$H_k = \begin{bmatrix} -\frac{1}{2}\frac{l_{\epsilon,k}^{-1/2}}{\sigma^2_{\epsilon,k}}\frac{\partial \sigma^2_{\epsilon,k}}{\partial \sigma^2} \\ -\frac{1}{\sigma_{\epsilon,k}}\frac{\partial \epsilon_k}{\partial \sigma^2} + \frac{\epsilon_k}{2\sigma^3_{\epsilon,k}}\frac{\partial \sigma^2_{\epsilon,k}}{\partial \sigma^2} \end{bmatrix},$$

where $l_{\epsilon,k} = \frac{\sigma^2_{\epsilon,k}}{3\epsilon_k^2 - 2\sigma^2_{\epsilon,k}}$, and accountig for the fact that the state-to-measurement matrix C_k is time varying

$$\sigma^2_{\epsilon,k} = \sigma^2 + C_k P_{k|k-1} C_k^T. \tag{C.2}$$

Also the derivatives in e_k and H_k can be expressed:

$$\begin{aligned} \frac{\partial \sigma^2_{\epsilon,k}}{\partial \sigma^2} &= 1 + C_k \frac{\partial P_{k|k-1}}{\partial \sigma^2} C_k^T, \\ \frac{\partial \epsilon_k}{\partial \sigma^2} &= -C_k \frac{\partial x_{k|k-1}}{\partial \sigma^2}, \end{aligned}$$

which need the computation of other derivatives by means of the following recursive equations

$$\begin{aligned} \frac{\partial P_{k|k-1}}{\partial \sigma^2} =& A\left[-\frac{\partial K_{k-1}}{\partial \sigma^2} C_{k-1} P_{k-1|k-2} + \right. \\ & \left. (I - K_{k-1}C_{k-1}) \frac{\partial P_{k-1|k-2}}{\partial \sigma^2}\right] A^T, \\ \frac{\partial K_k}{\partial \sigma^2} =& \frac{I - K_k C_k}{C_k P_{k|k-1} C_k^T + \sigma^2} \frac{\partial P_{k|k-1}}{\partial \sigma^2} C^T, \\ \frac{\partial x_{k+1|k}}{\partial \sigma^2} =& A\left[(I - K_k C_k) \frac{\partial x_{k|k-1}}{\partial \sigma^2} + \frac{\partial K_k}{\partial \sigma^2}\epsilon_k\right], \end{aligned} \tag{C.3}$$

given initial (arbitrary) values.

Supposing a stable behavior of the algorithm (the analysis is difficult because the involved matrices are time varying), these

derivatives tend to zero for $k \to \infty$. A chance is to set them to zero since $k = 0$. This simplified version of the filter, reported as unstable by some authors, has been tested and the results are available in the Sect. 3.5.

C.1 Minor issues

Some numerical issues were tackled:

- in (C.2) and in the second of (C.3) we use the current estimation $\hat{\sigma}^2_{k|k-1}$ instead of the actual σ^2;
- as explained in the references, it is better to filter the logarithm of the variances, mainly to avoid negative values (see [Hay01] for the modified equations);
- the starting values used for the recursive derivatives in the simulations are the following

$$\begin{aligned} \frac{\partial x_0}{\partial \sigma^2} &= [1,1,1,1]', \\ \frac{\partial P_0}{\partial \sigma^2} &= diag(1,1,1,1), \\ \frac{\partial K}{\partial \sigma^2} &= \mathbf{1}, \end{aligned} \tag{C.4}$$

  where $\mathbf{1}$ is the unitary matrix of suitable size;
- the fictitious random process (3.31) describing the variance is considered additive and Gaussian with the actual σ^2 like mean and variance $\sigma^2_r = 1/2$, as suggested in the above references.

Appendix D

CRLB computation

In this section we express in detail the computations of the matrices required for the CRLB; we will exploit the notation of App. B and the symmetry properties of R_v and R_n.

D.1 Gaussian measurements

In the case of fully known model as in Sect. 1.4.1, the matrices involved, here restated for clearness, are

$$\begin{cases} D_k^{11} &= E\left\{-\nabla_{\mathbf{x}_k}\left[\nabla_{\mathbf{x}_k}^T \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right]\right\}, \\ D_k^{12} &= E\left\{-\nabla_{\mathbf{x}_k}\left[\nabla_{\mathbf{x}_{k+1}}^T \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right]\right\}, \\ D_k^{21} &= E\left\{-\nabla_{\mathbf{x}_{k+1}}\left[\nabla_{\mathbf{x}_k}^T \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right]\right\}, \\ D_k^{22} &= E\left\{-\nabla_{\mathbf{x}_{k+1}}\left[\nabla_{\mathbf{x}_{k+1}}^T \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right]\right\} + \\ & \quad E\left\{-\nabla_{\mathbf{x}_{k+1}}\left[\nabla_{\mathbf{x}_{k+1}}^T \log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})\right]\right\}, \end{cases}$$

where the expectations are over $\mathbf{x}_k$, $\mathbf{x}_{k+1}$ and $\mathbf{y}_{k+1}$. We now assume the user to be far away from the obstacles and thus we use a linear model $\mathbf{x}_{k+1} = F\mathbf{x}_k$. Exploiting the Gaussian model in dB

we obatain

$$\begin{aligned}\log p(\mathbf{x}_{k+1}|\mathbf{x}_k) &\propto -\frac{1}{2}(\mathbf{x}_{k+1} - F\mathbf{x}_k)^T R_v^{-1}(\mathbf{x}_{k+1} - F\mathbf{x}_k)\\ \log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}) &\propto -\frac{1}{2}(\mathbf{y}_{k+1} - g(\mathbf{x}_{k+1}))^T R_n^{-1}(\mathbf{y}_{k+1} - g(\mathbf{x}_{k+1})).\end{aligned} \tag{D.1}$$

Now, applying the matricial operators, we can find

$$\nabla^T_{\mathbf{x}_k} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k) = [F^T R_v^{-1}(\mathbf{x}_{k+1} - F\mathbf{x}_k)]^T = (\mathbf{x}_{k+1} - F\mathbf{x}_k)^T R_v^{-1} F.$$

The above provides straightforwardly

$$\begin{aligned} D_k^{11} &= E\left\{-\nabla_{\mathbf{x}_k}\left[\nabla^T_{\mathbf{x}_k} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right]\right\} = F^T R_v^{-1} F,\\ D_k^{21} &= E\left\{-\nabla_{\mathbf{x}_{k+1}}\left[\nabla^T_{\mathbf{x}_k} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right]\right\} = -R_v^{-1} F,\end{aligned}$$

where we have exploited the symmetry of R_v, and the fact that the expectations vanish because R_v^{-1} and F are deterministic matrices. Similarly,

$$\nabla^T_{\mathbf{x}_{k+1}} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k) = -[R_v^{-1}(\mathbf{x}_{k+1} - F\mathbf{x}_k)]^T = -(\mathbf{x}_{k+1} - F\mathbf{x}_k)^T R_v^{-1}, \tag{D.2}$$

leading to

$$D_k^{12} = E\left\{-\nabla_{\mathbf{x}_k}\left[\nabla^T_{\mathbf{x}_{k+1}} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right]\right\} = -F^T R_v^{-1}.$$

As for D_k^{22}, it is composed of two terms, the former is easily obtained from eq. (D.2) and amounts to

$$E\left\{-\nabla_{\mathbf{x}_{k+1}}\left[\nabla^T_{\mathbf{x}_{k+1}} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right]\right\} = R_v^{-1};$$

the latter takes into account that

$$\nabla_{\mathbf{x}_{k+1}} \log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}) = \bar{g}^T(\mathbf{x}_{k+1}) R_n^{-1}(\mathbf{y}_{k+1} - \bar{g}(\mathbf{x}_{k+1})),$$

with $\bar{g}(\mathbf{x}_{k+1}) = \nabla_{\mathbf{x}_{k+1}} g(\mathbf{x}_{k+1})$, leading to

$$\begin{aligned}
&E\left\{-\nabla_{\mathbf{x}_{k+1}}\left[\nabla^T_{\mathbf{x}_{k+1}}\log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})\right]\right\}\\
&\quad= E\left\{\left[\nabla_{\mathbf{x}_{k+1}}\log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})\right]\left[\nabla_{\mathbf{x}_{k+1}}\log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})\right]^T\right\}\\
&\quad= E\left\{\bar{g}^T(\mathbf{x}_{k+1})R_n^{-1}(\mathbf{y}_{k+1}-\bar{g}(\mathbf{x}_{k+1}))\right.\\
&\qquad\left.(\mathbf{y}_{k+1}-\bar{g}(\mathbf{x}_{k+1}))^T R_n^{-1}\bar{g}(\mathbf{x}_{k+1})\right\}\\
&\quad= E_{\mathbf{x}_{k+1}}\left\{\bar{g}^T(\mathbf{x}_{k+1})R_n^{-1}\right.\\
&\qquad\left.E_{\mathbf{y}_{k+1}}\left[(\mathbf{y}_{k+1}-\bar{g}(\mathbf{x}_{k+1}))(\mathbf{y}_{k+1}-\bar{g}(\mathbf{x}_{k+1}))^T\right]R_n^{-1}\bar{g}(\mathbf{x}_{k+1})\right\}\\
&\quad= E_{\mathbf{x}_{k+1}}\left\{\bar{g}^T(\mathbf{x}_{k+1})R_n^{-1}E\left[\mathbf{n}_{k+1}\mathbf{n}_{k+1}^T\right]R_n^{-1}\bar{g}(\mathbf{x}_{k+1})\right\}\\
&\quad= E\left\{\bar{g}^T(\mathbf{x}_{k+1})R_n^{-1}\bar{g}(\mathbf{x}_{k+1})\right\}. \qquad \text{(D.3)}
\end{aligned}$$

The final result is thus

$$D_k^{22} = R_v^{-1} + E\left\{\bar{g}^T(\mathbf{x}_{k+1})R_n^{-1}\bar{g}(\mathbf{x}_{k+1})\right\}.$$

In the case of unknown parameters the matrices are here restated

$$\left\{\begin{array}{lcl}
H_k^{11} &=& E\left\{-\nabla_{\mathbf{x}_k}[\nabla^T_{\mathbf{x}_k}\log p_k]\right\},\\
H_k^{12} &=& E\left\{-\nabla_{\mathbf{x}_k}[\nabla^T_{\mathbf{w}_k}\log p_k]\right\},\\
H_k^{13} &=& E\left\{-\nabla_{\mathbf{x}_k}[\nabla^T_{\mathbf{x}_{k+1}}\log p_k]\right\},\\
H_k^{22} &=& E\left\{-\nabla_{\mathbf{w}_k}[\nabla^T_{\mathbf{w}_k}\log p_k]\right\},\\
H_k^{23} &=& E\left\{-\nabla_{\mathbf{w}_k}[\nabla^T_{\mathbf{x}_{k+1}}\log p_k]\right\},\\
H_k^{33} &=& E\left\{-\nabla_{\mathbf{x}_{k+1}}[\nabla^T_{\mathbf{x}_{k+1}}\log p_k]\right\},
\end{array}\right.$$

where the expectations are with respect to $\mathbf{x}_k$,$\mathbf{x}_{k+1}$,$\mathbf{p}_k$ and $\mathbf{y}_{k+1}$ and

$$p_k = p(\mathbf{x}_{k+1}|\mathbf{x}_k,\mathbf{p}_k)p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1},\mathbf{x}_k,\mathbf{p}_k) = p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1},\mathbf{p}_k).$$

Being it

$$\begin{aligned}
\log p_k =& -\frac{1}{2}(\mathbf{x}_{k+1}-F\mathbf{x}_k)^T R_v^{-1}(\mathbf{x}_{k+1}-F\mathbf{x}_k)\\
&-\frac{1}{2}(\mathbf{y}_{k+1}-g(\mathbf{x}_{k+1},\mathbf{p}_k))^T R_n^{-1}(\mathbf{y}_{k+1}-g(\mathbf{x}_{k+1},\mathbf{p}_k)),
\end{aligned} \qquad \text{(D.4)}$$

we can easily find

$$\begin{cases} \nabla^T_{\mathbf{x}_k} \log p_k &= = (\mathbf{x}_{k+1} - F\mathbf{x}_k)^T R_v^{-1} F, \\ \nabla^T_{\mathbf{x}_{k+1}} \log p_k &= -(\mathbf{x}_{k+1} - F\mathbf{x}_k)^T R_v^{-1} \\ & \quad +(\mathbf{y}_{k+1} - g(\mathbf{x}_{k+1}, \mathbf{p}_k))^T R_n^{-1} \bar{g}_x(\mathbf{x}_{k+1}, \mathbf{p}_k), \\ \nabla^T_{\mathbf{p}_k} \log p_k &= (\mathbf{y}_{k+1} - g(\mathbf{x}_{k+1}, \mathbf{p}_k))^T R_n^{-1} \bar{g}_p(\mathbf{x}_{k+1}, \mathbf{p}_k), \end{cases}$$

where again $\bar{g}_x(\mathbf{x}_{k+1}, \mathbf{p}_k) = \nabla_{\mathbf{x}_{k+1}} g(\mathbf{x}_{k+1}, \mathbf{p}_k)$ and $\bar{g}_p(\mathbf{x}_{k+1}, \mathbf{p}_k) = \nabla_{\mathbf{p}_k} g(\mathbf{x}_{k+1}, \mathbf{p}_k)$. Now, from the first of the above we can find by differentiating again with respect to $\mathbf{x}_k$ (the expectation vanishes)

$$H_k^{11} = F^T R_v^{-1} F.$$

From the second line we obtain

$$\begin{cases} H_k^{13} &= -F^T R_v^{-1}, \\ H_k^{23} &= E\left\{\bar{g}_p^T(\mathbf{x}_{k+1}, \mathbf{p}_k) R_n^{-1} \bar{g}_x(\mathbf{x}_{k+1}, \mathbf{p}_k)\right\}, \\ H_k^{33} &= R_v^{-1} + E\left\{\bar{g}_x^T(\mathbf{x}_{k+1}, \mathbf{p}_k) R_n^{-1} \bar{g}_x(\mathbf{x}_{k+1}, \mathbf{p}_k)\right\}, \end{cases}$$

and finally from the last one

$$\begin{cases} H_k^{12} &= 0, \\ H_k^{22} &= E\left\{\bar{g}_p^T(\mathbf{x}_{k+1}, \mathbf{p}_k) R_n^{-1} \bar{g}_p(\mathbf{x}_{k+1}, \mathbf{p}_k)\right\}. \end{cases}$$

D.2 Rice measurements

In the case of fully known models, assuming a Ricean model for the RSS from the n_{AP} nodes ($y_{k+1,i}$ is the RSS from the $i-th$ AP at the instant $k+1$)

$$\begin{aligned} \log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}) = \sum_{i=1}^{n_{AP}} \Bigg\{ & \log\left[\frac{2y_{k+1,i}(1+\gamma)}{P(d_{k+1,i})}\right] \\ & - \frac{(1+\gamma)y_{k+1,i}^2 + \gamma P(d_{k+1,i})}{P(d_{k+1,i})} \\ & + \log\left[I_0\left(2y_{k+1,i}\sqrt{\frac{\gamma(1+\gamma)}{P(d_{k+1,i})}}\right)\right]\Bigg\}. \end{aligned} \quad \text{(D.5)}$$

The only matrix to be affected is D_k^{22}, where

$$\begin{aligned}\nabla_{\mathbf{x}_{k+1}} \log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}) \\ = \sum_{i=1}^{n_{AP}} \left\{ \frac{\alpha \left(P(d_{k+1,i}) - y^2(\gamma+1)\right) I_0\left(2y_{k+1,i}\sqrt{\frac{\gamma(\gamma+1)}{P(d_{k+1,i})}}\right)}{d_{k+1,i}^2 P(d_{k+1,i}) I_0\left(2y_{k+1,i}\sqrt{\frac{\gamma(\gamma+1)}{P(d_{k+1,i})}}\right)} \right. \\ \left. + \frac{y_{k+1,i}\sqrt{\gamma(\gamma+1)} I_1\left(2y_{k+1,i}\sqrt{\frac{\gamma(\gamma+1)}{P(d_{k+1,i})}}\right)}{d_{k+1,i}^2 \sqrt{P(d_{k+1,i})} I_0\left(2y_{k+1,i}\sqrt{\frac{\gamma(\gamma+1)}{P(d_{k+1,i})}}\right)} \right\} (\mathbf{x}_{k+1} - \mathbf{x}_{AP,i}),\end{aligned} \tag{D.6}$$

leading to the computation of

$$E\left\{-\nabla_{\mathbf{x}_{k+1}}\left[\nabla^T_{\mathbf{x}_{k+1}} \log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})\right]\right\}$$

by performing a further gradient. This computation is staightforward but involved and thus is here omitted.

In the case of unknown parameters, we have for $H_k^{2,2}$

$$\begin{aligned} H_k^{2,2} &= E\left[\sum_{i=1}^{n_{AP}} \nabla_{\mathbf{p}_k}\nabla^T_{\mathbf{p}_k}\left[\alpha \log P_{0,i} - \alpha \log d_{k+1,i}\right.\right. \\ &\quad + \frac{(1+\gamma)y_{k+1,i}^2 + \gamma P_{0,i}(d_0/d_{k+1,i})^\alpha}{P_{0,i}(d_0/d_{k+1,i})^\alpha} \\ &\quad \left.\left. - \log I_0\left(2y_{k+1,i}\sqrt{\frac{\gamma(1+\gamma)}{P_{0,i}(d_o/d_{k+1,i})^\alpha}}\right)\right]\right], \end{aligned}$$

where the terms $\frac{\partial^2(\cdot)}{\partial P_{0,i}\partial P_{0,j}} = 0$, $i \neq j$. Similar arguments hold for the last two matrices and, in particular

$$H_k^{33} = R_v^{-1} - E\left[\nabla_{\mathbf{x}_{k+1}}\left[\nabla^T_{\mathbf{x}_{k+1}} \sum_{i=1}^{n_{AP}} \log p(\mathbf{y}_{k+1,i}|\mathbf{x}_{k+1}, \mathbf{p}_k)\right]\right].$$

Appendix E

GMM approximation for the peaks in WiSLAM

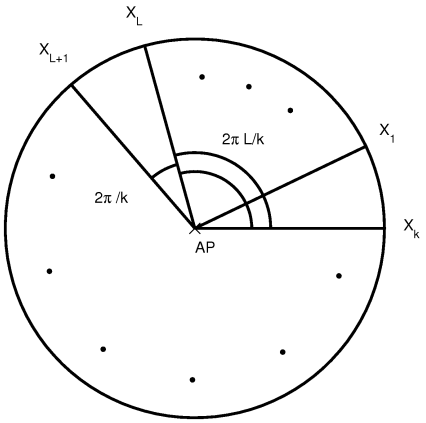
In Fig. E.1.a: the AP is located in the center and the user collects k RSS measurements in uniformly spaced positions of the circle. The symmetric version for Kullback Leibler (KL) divergence

$$D_s\left(p||\hat{p}\right) = \int p \log \frac{p}{\hat{p}} + \int \hat{p} \log \frac{\hat{p}}{p}, \tag{E.1}$$

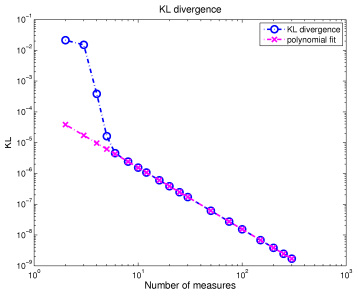
is used to evaluate the approximation error between the posterior AP's position pdf and a Gaussian pdf with same mean and covariance matrix for increasing values of k and the results are shown in Fig. E.1:

- for small k the log of the KL divergence decreases linearly with the square of k, as highlighted by the comparison with the fitting curve in Fig. E.1.b;
- the mean of the Gaussian peak is centered on the AP's position while the covariance matrix is diagonal with its entries asymptotically proportional to $1/k$ (see Fig. E.1.c).

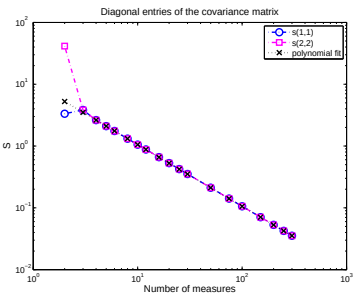
These results indicate that the Gaussian approximation is asymptotically suitable for the peaks but it is still reasonable for small k. However, in this case, we ensure more flexibility by allowing a generic nondiagonal covariance matrix.



(a) Framework



(b) KL divergence



(c) Variances

Figure E.1 Simulations confirming our guess about the asymptotical distribution for $\mathbf{W}$. Here $R = 20$, $k = 2 \div 300$. We show (a) the framework, (b) the symmetric KL distance as defined in eq. (E.1) with its polynomial fit and (c) the variances of the Gaussian approximation.

Appendix F

Proof of the Proposition 1

We will first show that

$$f_{p,k-1}(\mathbf{x}_{AP}, h_h) \cdot p\left(\mathbf{Z}_k^W | \mathbf{x}_{AP}, h_h, \mathbf{P}_k\right) = c_0 \cdot f_{p,k}(\mathbf{x}_{AP}, h_h), \quad \text{(F.1)}$$

where c_0 is a constant term and $f_{p,k}(\mathbf{x}_{AP}, h_h)$ a Gaussian pdf, and then we will compute the new GMM coefficients $\tilde{u}_{p,k}$.

The peak $f_{p,k-1}(\mathbf{x}_{AP}, h_h)$ is distributed as a bivariate Gaussian pdf, depending on AP's emitted power h_h and position $\mathbf{x}_{AP}$. This latter is expressed in terms of the Cartesian reference system (x, y) of Fig. 5.17, whose origin is in the center of the RSS likelihood ($\mathbf{P}_k = 0$). It is comfortable to rotate the axes by an angle α counter clockwise, such that in the new system of coordinates (a, b) the second component of the peak mean, say μ_{k-1}^R, is zero, i.e. $\mu_{k-1}^R = [\mu_{a,k-1}, 0]'$ and the resulting covariance matrix S_{k-1}^R is

$$S_{k-1}^R = \begin{bmatrix} \sigma_{a,k-1}^2 & \rho_{k-1}\sigma_{a,k-1}\sigma_{b,k-1} \\ \rho_{k-1}\sigma_{a,k-1}\sigma_{b,k-1} & \sigma_{b,k-1}^2 \end{bmatrix}.$$

In what follows we will consider just a peak at a given reference power h_h, denoting it with the simplified notation

$$f_{k-1}(a, b) \hat{=} f_{p,k-1}(\mathbf{x}_{AP}, h_h),$$

where a and b identify the coordinates of $\mathbf{x}_{AP}$ in the rotated reference system. With this notation in mind, we recall that the peak's

conditional pdfs $f(a|b)$ and $f(b|a)$ are both Gaussian. Similarly, we express the RSS likelihood function in the (a, b) system by a simplified notation

$$g_k(a) \hat{=} p\left(\mathbf{Z}_k^W | \mathbf{x}_{AP}, h_h, \mathbf{P}_k\right),$$

where we highlight that, if the Ass. 2 is employed, $g(\cdot)$ is uniform along the axis b and it is Gaussian distribution along a with parameters $(\hat{r}, \sigma_G^2)$. We are interested in

$$f_k(a, b) \propto f_{k-1}(a, b) \cdot g_k(a),$$

whose conditional pdfs

$$\begin{cases} f_k(a|b) \propto & f_{k-1}(a|b) \cdot g_k(a) \\ f_k(b|a) \propto & f_{k-1}(b|a) \end{cases}$$

are both Gaussian with easily computable parameters.

The problem of how to build up a compatible joint pdf from its conditionals is studied in the auto-models literature [Bes74]. Recalling the results regarding Gaussian auto-models [GG10, Ch. 2], the parameters of the new Gaussian joint pdf result in

$$\rho_k^2 = dd',\ \sigma_{a,k} = \sqrt{\frac{e'}{1-\rho_k^2}},\ \sigma_{b,k} = \sqrt{\frac{e}{1-\rho_k^2}}, \tag{F.2}$$

$$\mu_{a,k} = \frac{c' + c\rho_k\sqrt{e'/e}}{1-\rho_k^2},\ \mu_{b,k} = \frac{c + c'\rho_k\sqrt{e/e'}}{1-\rho_k^2}, \tag{F.3}$$

where

$$c = -\rho_{k-1}\frac{\sigma_{b,k-1}}{\sigma_{a,k-1}}\mu_{a,k-1},\ d = \rho_{k-1}\frac{\sigma_{b,k-1}}{\sigma_{a,k-1}}, \tag{F.4}$$

$$e = \sigma_{b,k-1}^2(1-\rho_{k-1}^2),\ c' = \frac{\sigma_G^2\mu_{a,k-1} + \sigma_{a,k-1}^2(1-\rho_{k-1}^2)\hat{r}}{\sigma_G^2 + \sigma_{a,k-1}^2(1-\rho_{k-1}^2)}, \tag{F.5}$$

$$d' = \frac{\rho_{k-1}\frac{\sigma_{a,k-1}}{\sigma_{b,k-1}}\sigma_G^2}{\sigma_G^2 + \sigma_{a,k-1}^2(1-\rho_{k-1}^2)},\ e' = \frac{\sigma_G^2\sigma_{a,k-1}^2(1-\rho_{k-1}^2)}{\sigma_G^2 + \sigma_{a,k-1}^2(1-\rho_{k-1}^2)}. \tag{F.6}$$

Notice that we compute ρ_k^2 and both its roots are feasible, since $-1 \leq \rho_k \leq 1$. Therefore the sign of ρ_k should be set using other physical considerations and here it must be the same as ρ_{k-1}, since one can show that $\rho_k^2 \leq \rho_{k-1}^2$.

After having rotated this pdf by the same angle α, but clockwise, finding the updated μ_k and S_k of the peak in the original coordinate system (x, y), the constant c_0 in eq. (F.1) is computed by fixing $\mathbf{x}_{AP}$, for example, on the mean μ_k:

$$c_0 = \frac{f_{p,k-1}(\mu_k, h_h) \cdot p\left(\mathbf{Z}_k^W | \mu_k, h_h, \mathbf{P}_k\right)}{f_{p,k}(\mu_k, h_h)}.$$

As for the coefficients $\tilde{u}_{p,k}$ of eq. (5.37), we can observe that employing (F.1) in eq. (5.34) we find

$$\begin{aligned}
&\hat{p}\left(\mathbf{x}_{AP} | h_h, \mathbf{P}_{0:k}, \mathbf{Z}_{1:k}^W\right) \\
&\propto \sum_{p=1}^{N_{peaks}} \tilde{u}_{p,k-1} \cdot f_{p,k-1}(\mathbf{x}_{AP}, h_h) \cdot p\left(\mathbf{Z}_k^W | \mathbf{x}_{AP}, h_h, \mathbf{P}_k\right) \\
&= \sum_{p=1}^{N_{peaks}} u_{p,k} \cdot f_{p,k}(\mathbf{x}_{AP}, h_h),
\end{aligned}$$

where the coefficients $u_{p,k} \hat{=} \tilde{u}_{p,k-1} \cdot c_0$ must be normalized over p to obtain the new $\tilde{u}_{p,k}$ of eq. (5.37):

$$\tilde{u}_{p,k} = \frac{u_{p,k}}{\sum_{p=1}^{N_{Peaks}} u_{p,k}}.$$

Bibliography

[ABG+10] Paolo Addesso, Luigi Bruno, Roberto Garufi, Maurizio Longo, Rocco Restaino, and Anton Luca Robustelli, *A model-based approach for wlan localization in indoor parking areas*, Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on, 2010, pp. 1 –10.

[ABR10] Paolo Addesso, Luigi Bruno, and Rocco Restaino, *Adaptive localization techniques in wifi environments*, Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on, May 2010, pp. 289 –294.

[AMGC02] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, *A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking*, IEEE Trans. On Signal Processing. **50** (2002), no. 2, 174–188.

[BB99] Sergio Benedetto and Ezio Biglieri, *Principles of digital transmission: With wireless applications*, Kluwer Academic Publishers, Norwell, MA, USA, 1999.

[BB05] R. Battiti and R. Brunato, *Statistical learning theory for location fingerprinting in wireless lans*, Computer Networks **47** (2005), no. 6.

[Bes74] J. Besag, *Spatial interaction and the statistical analysis of lattice systems*, Journal of the Royal Statistical

Society, Series B (Methodological) **36** (1974), no. 2, 192 – 236.

[BMM11] L. Bruno, V. Matta, and S. Marano, *Multisensor systems with variable sampling rate*, Proceeding Signal and Image Processing and Applications / 716: Artificial Intelligence and Soft Computing - 2011, 2011.

[BP00a] P. Bahl and V. Padmanabhan, *Enhancements to the radar user location and tracking systems*, Technical Report, MSR-TR-200-12, Microsoft Research (2000), 775–784.

[BP00b] _____, *Radar: An in-building rf-based user location and tracking system*, Proceedings of IEEE INFOCOM 2000 (2000), 775–784.

[BSLK01] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: Theory algorithms and software*, John Wiley And Sons, 2001.

[BU11] G. Battistello and M. Ulmke, *Exploitation of a priori information for tracking maritime intermittent data sources*, Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on, july 2011, pp. 1 –8.

[BWK08] S. Beauregard, Widyawan, and M. Klepal, *Indoor pdr performance enhancement using minimal map information and particle filters*, Position, Location and Navigation Symposium, 2008 IEEE/ION, May 2008, pp. 141 –147.

[CBC06] Marc Ciurana, Francisco Barceló, and Sebastiano Cugno, *Indoor tracking in wlan location with toa measurements*, Proceedings of the 4th ACM international workshop on Mobility management and wire-

less access (New York, NY, USA), MobiWac '06, ACM, 2006, pp. 121–125.

[CH67] T. Cover and P. Hart, *Nearest neighbor pattern classification*, IEEE Transactions on Information Theory **13** (1967), 21–27.

[CT01] Thomas M. Cover and Joy A. Thomas, *Frontmatter and index*, pp. i–xxiii, John Wiley & Sons, Inc., 2001.

[CWYS09] Y.-S. Chiou, C.-L. Wang, S.-C. Yeh, and M.-Y. Su, *Design of adaptive positioning system based on wifi radio signals*, Comput. Commun. **32** (2009), no. 7-10, 2097–2101.

[DFG01] A. Doucet, N. de Freitas, and N. Gordons, *Sequential monte carlo methods in practice*, Springer, 2001.

[DRFF10] F. Della Rosa, S. Frattasi, and J Figueiras, *Cooperative localization in a hybrid wimax/wifi system: A future-proof framework*, Springer, 2010.

[FFL07] Brian Ferris, Dieter Fox, and Neil Lawrence, *Wifi-slam using gaussian process latent variable models*, In Proceedings of IJCAI 2007, 2007, pp. 2480–2485.

[FH89] Evelyn Fix and Jr. Hodges, J. L., *Discriminatory analysis. nonparametric discrimination: Consistency properties*, International Statistical Review / Revue Internationale de Statistique **57** (1989), no. 3, pp. 238–247 (English).

[FHL+03] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, *Bayesian filtering for location estimation*, Pervasive Computing (2003), 24–33.

[Fox05] E. Foxlin, *Pedestrian tracking with shoe-mounted inertial sensors*, Computer Graphics and Applications, IEEE **25** (2005), no. 6, 38 –46.

[GABS10] J. Graefenstein, A. Albert, P. Biber, and A. Schilling, *Simultaneous mobile robot and radio node localization in wireless networks*, Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on, 2010, pp. 1 –6.

[GG10] C. Gaetan and X. Guyon, *Spatial statistics and modeling*, Springer, 2010.

[GME99] L.J. Greenstein, D.G. Michelson, and V. Erceg, *Moment-method estimation of the ricean k-factor*, Communications Letters, IEEE **3** (1999), no. 6, 175 –176.

[Has93] H. Hashemi, *The indoor propagation channel*, Proceedings of the IEEE **81** (1993), no. 7, 943–968.

[Hay01] S. Haykin, *Kalman filtering and neural networks*, John Wiley And Sons, 2001.

[HHS+99] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster, *The anatomy of a context-aware application*, Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (New York, NY, USA), MobiCom '99, ACM, 1999, pp. 59–68.

[HMQ+11] Joseph Huang, David Millman, Morgan Quigley, David Stavens, Sebastian Thrun, and Alok Aggarwal, *Efficient, generalized indoor wifi graphslam*, Robotics and Automation (ICRA), 2011 IEEE International Conference on, may 2011, pp. 1038 –1043.

[HVBW01] Jeffrey Hightower, Chris Vakili, Gaetano Borriello, and Roy Want, *Design and calibration of the spoton ad-hoc location sensing system*, Tech. report, 2001.

[ICB+06] F. Izquierdo, M. Ciurana, F. Barcelo, J. Paradells, and E. Zola, *Performance evaluation of a toa-based*

trilateration method to locate terminals in wlan, Wireless Pervasive Computing, 2006 1st International Symposium on, 2006, pp. 1 – 6.

[JUDW95] S.J. Julier, J.K. Uhlmann, and H.F. Durrant-Whyte, *A new approach for filtering nonlinear systems*, American Control Conference, 1995. Proceedings of the, vol. 3, jun 1995, pp. 1628 –1632 vol.3.

[KA07] Young-Chai Ko and Mohamed-Slim Alouini, *Local mean signal estimation over nakagami-m fading channels*, Wireless Communications and Mobile Computing **7** (2007), no. 3, 367–374.

[Kay93] S.M. Kay, *Fundamentals of statistical signal processing: Estimation theory*, Prentice Hall, 1993.

[KR08] B. Krach and P. Roberston, *Cascaded estimation architecture for integration of foot- mounted inertial sensors*, Position, Location and Navigation Symposium, 2008 IEEE/ION, May 2008, pp. 112 –119.

[LLP09] C. Liang, W. Lenan, and Piche, *Posterior cramer-rao lower bound for mobile tracking in mixed los/nlos conditions*, Eusipco 2009, 17th European Signal Processing Conference, 2009, pp. 90 –94.

[MEM+05] D. Madigan, E. Einahrawy, R.P. Martin, W.-H. Ju, P. Krishnan, and A.S. Krishnakumar, *Bayesian indoor positioning systems*, Proceedings of IEEE INFOCOM 2005 **2** (2005), 1217–1227.

[MS00] T.K. Moon and W.C. Stirling, *Mathematical methods and algorithms for signal processing*, Prentice Hall, 2000.

[MTKB02] M. Montemerlo, S. Thrun, D. Koller, and Wegbreit B., *Fastslam: A factored solution to the simultaneous localization and mapping problem*, Proceedings of AAAI 2002, 2002, pp. 593 – 598.

[MZZ+09] E. Menegatti, A. Zanella, S. Zilli, F. Zorzi, and E. Pagello, *Range-only slam with a mobile robot and a wireless sensor networks*, Robotics and Automation, 2009. ICRA '09. IEEE International Conference on, May 2009, pp. 8 –14.

[Nel00] Alex Tremain Nelson, *Nonlinear estimation and modeling of noisy time series by dual kalman filtering methods*, Ph.D. thesis, 2000, AAI9984703.

[NLLP04] Lionel M. Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P. Patil, *Landmarc: Indoor location sensing using active rfid*, Wireless Networks **10** (2004), 701–710, 10.1023/B:WINE.0000044029.06344.dd.

[NV05] Ruixin Niu and Pramod K. Varshney, *Distributed detection and fusion in a large wireless sensor network of random size*, EURASIP J. Wirel. Commun. Netw. **2005** (2005), 462–472.

[Par00] J. Parsons, *The mobile radio propagation channel, $2^n d$ ed.*, Wiley & Sons, November 2000.

[PAR11] L. Bruno P. Addesso and R. Restaino, *Integrating RSS from unknown access points in WLAN positioning*, Wireless Networking Symposium (IWCMC2011-Wireless Nets) (Istanbul, Turkey), 7 2011.

[PCB00] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan, *The cricket location-support system*, Proceedings of the 6th annual international conference on Mobile computing and networking (New York, NY, USA), MobiCom '00, ACM, 2000, pp. 32–43.

[PPY+12] J.J. Pan, S.J. Pan, Jie Yin, L.M. Ni, and Qiang Yang, *Tracking mobile users in wireless networks via semi-supervised colocalization*, Pattern Analysis and Ma-

chine Intelligence, IEEE Transactions on **34** (2012), no. 3, 587 –600.

[PW08] A.S. Paul and E.A. Wan, *Wi-fi based indoor localization and tracking using sigma-point kalman filtering methods*, Position, Location and Navigation Symposium, 2008 IEEE/ION, May 2008, pp. 646 –659.

[Rab89] L.R. Rabiner, *A tutorial on hidden markov models and selected applications in speech recognition*, Proceedings of the IEEE **77** (1989), no. 2, 257 –286.

[RAG04] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the kalman filter: particle filters for tracking applications*, Artech House Publishers, 2004.

[RAK09] P. Robertson, M. Angermann, and B. Krach, *Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors*, Proceedings of the 11th international conference on Ubiquitous computing, Ubicomp '09, ACM, 2009, pp. 93–96.

[RAK10] P. Robertson, M. Angermann, and M. Khider, *Improving simultaneous localization and mapping for pedestrian navigation and automatic mapping of buildings by using online human-based feature labeling*, Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION, May 2010, pp. 365 –374.

[Rap96] Theodore S. Rappaport, *Wireless communications: Principles and practice*, 1st ed., IEEE Press, Piscataway, NJ, USA, 1996.

[RGPA11] P. Roberston, M. Garcia Puyol, and M. Angermann, *Collaborative pedestrian mapping of buildings using inertial sensors and footslam*, Pedestrian Navigation, 2011 IEEE/ION, 2011.

[RMT+02] T. Roos, P. Myllymaki, H. Tirri, P. Misikangas, and J. Sievanen, *A probabilistic approach to wlan user location estimation*, International Journal of Wireless Information Networks **9** (2002), no. 3, 155–164.

[SA05] M. Simon and M. Alouini, *Digital communications over fading channels, 2^{n}d ed.*, Wiley & Sons, 2005.

[SGB05] C. Stachniss, G. Grisetti, and W. Burgard, *Recovering particle diversity in a rao-blackwellized particle filter for slam after actively closing loops*, Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, april 2005, pp. 655 – 660.

[SR04] M.L. Sichitiu and V. Ramadurai, *Localization of wireless sensor networks with a mobile beacon*, Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on, 2004, pp. 174 – 183.

[sti00] *Mathematical methods and algorithms for signal processing / t.k. moon, wynn c. stirling.*, Upper Saddle River, EUA : Prentice-Hall, 2000.

[TMN98] P. Tichavsky, C.H. Muravchik, and A. Nehorai, *Posterior cramer-rao bounds for discrete-time nonlinear filtering*, IEEE Trans. on Sig. Proc. **46** (1998).

[VSK+10] Thorsten Vaupel, Jochen Seitz, Frederic Kiefer, Stephan Haimerl, and Jorn Thielecke, *Wi-fi positioning: System considerations and device calibration*, Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on, 2010, pp. 1 –7.

[VT01] H.L. Van Trees, *Detection, estimation, and modulation theory, part i*, John Wiley $ Sons Inc., 2001.

[WH08] Oliver Woodman and Robert Harle, *Pedestrian localisation for indoor environments*, Proceedings of

the 10th international conference on Ubiquitous computing (New York, NY, USA), UbiComp '08, ACM, 2008, pp. 114–123.

[WHFaG92] Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons, *The active badge location system*, ACM Trans. Inf. Syst. **10** (1992), 91–102.

[YA08] Moustafa Youssef and Ashok Agrawala, *The horus location determination system*, Wireless Networks **14** (2008), 357–374, 10.1007/s11276-006-0725-7.

[YOT+05] R. Yamasaki, A. Ogino, T. Tamaki, T. Uta, N. Matsuzawa, and T. Kato, *Tdoa location system for ieee 802.11b wlan*, Wireless Communications and Networking Conference, 2005 IEEE, vol. 4, 2005, pp. 2338 – 2343 Vol. 4.

[YYN08] Jie Yin, Qiang Yang, and Lionel M. Ni, *Learning adaptive temporal radio maps for signal-strength-based location estimation*, IEEE Transactions on Mobile Computing **7** (2008), 869–883.

[ZZ69] J. Ziv and M. Zakai, *Some lower bounds on signal parameter estimation*, Information Theory, IEEE Transactions on **15** (1969), no. 3, 386 – 391.

[ZZ09] F. Zorzi and A. Zanella, *Opportunistic localization: Modeling and analysis*, Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th, 2009, pp. 1 –5.